ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μέτρα αξίας ARM και παραγωγή συστάσεων

Της φοιτήτριας
Φωτοπούλου Νικολέττας
Αρ. Μητρώου: 185307

Επιβλέπων
Δέρβος Δημήτριος
Καθηγητής

Ημερομηνία 08-09-2023

Τίτλος Δ.Ε. Μέτρα αξίας ARM και παραγωγή συστάσεων
Κωδικός Δ.Ε. 23119
Ονοματεπώνυμο φοιτήτριας Νικολέττα Φωτοπούλου
Ονοματεπώνυμο εισηγητή Δημήτριος Δέρβος
Ημερομηνία ανάληψης Δ.Ε. 23-02-2023
Ημερομηνία περάτωσης Δ.Ε. 08-09-2023

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

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Στην οικογένειά μου»

Πρόλογος

Στο μάθημα "Εισαγωγή στην Αναλυτική των Δεδομένων" γίνεται μία πρώτη προσέγγιση του θέματος της εξόρυξης πληροφορίας από σύνολα δεδομένων, μάλιστα: με τη χρήση του αλγορίθμου Apriori σε προγραμματιστικό περιβάλλον R/RStudio. Παράλληλα, γίνεται αναφορά στην αξιοποίηση της εν λόγω πληροφορίας για την παραγωγή συστάσεων που βρίσκουν εφαρμογή σε ένα ευρύ φάσμα τομέων στη σημερινή καθημερινότητα η οποία ολοένα και περισσότερο διαμορφώνεται από τη χρήση του διαδικτύου. Σε αυτό το πλαίσιο εντοπίζεται το ενδιαφέρον και το κίνητρο για την εκπόνηση της παρούσας πτυχιακής εργασίας. Πιο συγκεκριμένα, στη μελέτη των μεθόδων αξιολόγησης της ποιότητας ενός συστήματος παραγωγής συστάσεων με τη χρήση κανόνων συσχέτισης, ειδικά στην περίπτωση που χρησιμοποιούνται αριθμητικού τύπου δεδομένα.

Περίληψη

Η παρούσα πτυχιακή εργασία ερευνά τον τρόπο με τον οποίο ένα σύστημα παραγωγής συστάσεων εφαρμόζει την τεχνική Association Rule Mining (ARM) σε αριθμητικά σύνολα δεδομένων. Για την αξιολόγηση των αποδόσεων του συστήματος, χρησιμοποιούνται τα μέτρα επιδόσεων Precision και Recall. Οι συστάσεις προκύπτουν από την ταξινόμηση των κανόνων συσχέτισης με βάση τα μέτρα αξίας ARM lift και conviction, όπως επίσης και με τα συντελεστές συσχέτισης Pearson και Spearman. Ο διαχωρισμός των δεδομένων γίνεται με την χρήση της του μοντέλου αξιολόγησης cross-validation.

Η παραπάνω έρευνα γίνεται στο περιβάλλον R/RStudio. Χρησιμοποιείται η βιβλιοθήκη RecommenderLab για την παραγωγή των συστάσεων και για την αξιολόγησή τους.

Η συνεισφορά της πτυχιακής εργασίας έγκειται στην επέκταση του κώδικα R της βιβλιοθήκης RecommenderLab για την χρήση του μέτρου αξίας Conviction της ARM, καθώς επίσης και τους συντελεστές συσχέτισης Pearson και Spearman. Η απόδοση της παραγωγής συστάσεων υπολογίζεται από την κατασκευή του αντίστοιχου διαγράμματος Precision-Recall. Η χρήση των συντελεστών συσχέτισης Pearson και Spearman για την ταξινόμηση των κανόνων της ARM οδηγεί σε βελτιωμένα αποτελέσματα στις ειδικές περιπτώσεις όπου τα σύνολα δεδομένων περιλαμβάνουν αριθμητικά δεδομένα.

ARM Interestingness Measures for Recommendations

Nikoletta Fotopoulou

Abstract

This thesis investigates how a data mining recommender service operates using Association Rule Mining (ARM) with numeric datasets. To evaluate the service's performance, the Precision and Recall measures are used. Recommendations are generated by ranking the association rules output based on ARM interestingness measures such as lift and conviction, as well as by using the Pearson and Spearman correlation measures. Data splitting and cross-validation are employed to assess performance.

All processing takes place within the R/RStudio environment. RecommenderLab library functions are employed to carry out the recommendation generation and performance evaluation tasks.

This thesis makes an innovative contribution by extending the R code of the RecommenderLab library to accommodate the ARM Conviction measure, as well as Pearson and Spearman correlations. Recommender service performance is measured by constructing the corresponding Precision-Recall plots. The use of Pearson and Spearman correlation coefficients for ranking the ARM output is seen to lead to improved performance in special cases of transaction datasets that involve numeric data.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Δημήτριο Δέρβο για τον προσωπικό χρόνο που αφιέρωσε από την αρχή έως το τέλος της πτυχιακής μου εργασίας, για την πολύτιμη καθοδήγησή του, για την εμπιστοσύνη που μου έδειξε και για την συνεχή του στήριξη.

Ευχαριστώ τον κ. Κωνσταντίνο Κελεσίδη για την βοήθειά του και την συνεργασία μας.

Τέλος, ευχαριστώ την οικογένειά μου και τον σύντροφό μου για την στήριξή τους.

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Συντομογραφίες

ARM	Association Rule Mining
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
ΣΑΠ	Σύστημα Ανάκτησης Πληροφορίας
RS	Recommender System
ΑΤΕΙΘ	Αλεξάνδρειο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλονίκης

Κεφάλαιο 1ο : Αντικείμενο και στόχος της εργασίας

Είναι πολλές φορές απαραίτητο ένας άνθρωπος να λάβει αποφάσεις χωρίς να έχει κάποια προηγούμενη εμπειρία και γνώση σχετικά με τις επιλογές που του παρέχονται. Για αυτό στην καθημερινή ζωή θα βασιστεί στις συστάσεις άλλων ανθρώπων από στόμα σε στόμα, στα Μέσα Μαζικής Ενημέρωσης, θα διαβάσει κριτικές, έρευνες και οδηγούς [1].

Τα Recommender Systems (RSs - Συστήματα Παραγωγής Συστάσεων) παίζουν καθοριστικό ρόλο στην σύγχρονη εποχή, χάρη στην ανάπτυξη του διαδικτύου και της χρήσης υπηρεσιών μέσα από αυτό. Μέσα από το διαδίκτυο οι άνθρωποι διαμορφώνουν στρατηγικές για αγορές προϊόντων, για τη ψυχαγωγία τους (μουσική, ταινίες, κλπ.), και επικοινωνούν με άλλους ανθρώπους. Η καταγραφή σε ηλεκτρονική μορφή αυτού του είδους των δραστηριοτήτων του σύγχρονου καταναλωτή, συνιστά μία πρόκληση για τους ερευνητές να μελετήσουν αυτά τα τεράστια σύνολα δεδομένων και να εξορύξουν χρήσιμη πληροφορία από αυτά, και μεταξύ άλλων να δημιουργήσουν συστήματα παραγωγής συστάσεων. Ο σκοπός αυτών των συστημάτων είναι μέσα από το σύνολο των δεδομένων να μετατρέψουν τις προτιμήσεις των χρηστών σε προβλέψεις για τις μελλοντικές προτιμήσεις τους. Μεγάλες ιστοσελίδες και εφαρμογές όπως το Netflix, το Youtube, το Spotify και η Amazon κάνουν χρήση τέτοιων συστημάτων για την παραγωγή συστάσεων στους πελάτες τους [2].

Αυτό θα μπορούσε να επιτευχθεί με διάφορες προσεγγίσεις, για αυτό και δεν υπάρχουν κάποιες βασικές αρχές που θα έπρεπε να ακολουθούνται κάθε φορά, καθώς δεν θεωρείται κάποια τακτική ως ‘πιο σωστή’ σε σχέση με τις άλλες [2]. Για παράδειγμα, μπορεί μία προσέγγιση να υποστηρίζει πως προκειμένου να γίνει σύσταση σε κάποιον χρήστη θα ληφθεί υπόψη τι άρεσε στους χρήστες που παρουσιάζουν ομοιότητα με αυτόν τον χρήστη. Μία άλλη προσέγγιση θα μπορούσε να είναι να υπολογιστεί τι αρέσει στον συγκεκριμένο χρήστη με βάση το προφίλ που παρουσιάζει ο ίδιος και να του προταθούν παρόμοια αντικείμενα, ή ακόμα να αποφασιστεί με βάση τα δεδομένα που παρέχουν οι υπόλοιποι χρήστες αν η κατανάλωση ενός αντικειμένου που άρεσε στο παρελθόν στον συγκεκριμένο χρήστη συνεπάγεται την κατανάλωση και την αρέσκεια κάποιου άλλου αντικειμένου για να του γίνει αυτή η σύσταση.

Η παρούσα πτυχιακή εργασία εστιάζει στην γλώσσα R, στα συστήματα παραγωγής συστάσεων και στις μεθοδολογίες αξιολόγησης των επιδόσεών τους. Με την χρήση της βιβλιοθήκης RecommenderLab, στοχεύει στην σύγκριση μεταξύ τεχνικών παραγωγής συστάσεων και μεταξύ μεθόδων αξιολόγησης αυτών των τεχνικών. Ειδικότερα, αξιολογούνται οι επιδόσεις συστημάτων παραγωγής συστάσεων οι οποίες παράγονται με την χρήση Association Rules σε σχέση με τα μέτρα αξίας που σχετίζονται με τη συγκεκριμένη τεχνική εξόρυξης πληροφορίας και με την χρήση μέτρων που χρησιμοποιούνται στην αξιολόγηση των επιδόσεων ενός Συστήματος Ανάκτησης Πληροφορίας (ΣΑΠ / Information Retrieval). Τέλος, διερευνάται το ενδεχόμενο της χρήσης ενός επιπλέον μέτρου αξίας (αυτό του συντελεστή συσχέτισης) στην ειδική περίπτωση της χρήσης της τεχνικής Association Rules Mining (ARM) σε αριθμητικά δεδομένα.

Στο Κεφάλαιο 2 παρουσιάζονται οι βασικές έννοιες που είναι απαραίτητες για την κατανόηση του αντικειμένου, όπως μερικές Τεχνικές Παραγωγής Συστάσεων, τα μέτρα ομοιότητας, τους συντελεστές συσχέτισης και τα μέτρα αξιολόγησης των επιδόσεων ενός ΣΑΠ. Δίνεται έμφαση στις τεχνικές ARM, καθώς και στα μέτρα και στις μεθόδους που είναι απαραίτητα για την αξιολόγηση των αποτελεσμάτων των παραγόμενων συστάσεων.

Στο Κεφάλαιο 3 γίνεται αναφορά στο προγραμματιστικό περιβάλλον, καθώς και στις βιβλιοθήκες που χρησιμοποιήθηκαν, με ιδιαίτερη έμφαση στην βιβλιοθήκη RecommenderLab που είναι η βιβλιοθήκη που παράγει τις συστάσεις και εκτελεί την αξιολόγησή τους. Θα παρουσιαστούν αναλυτικά οι συναρτήσεις και οι μέθοδοί της.

Το Κεφάλαιο 4 συμπεριλαμβάνει την τροποποίηση του κώδικα RecommenderLab ώστε να χρησιμοποιείται και το conviction ως μέτρο αξίας που λαμβάνεται υπ΄όψιν στην ταξινόμηση των κανόνων ARM. Παρουσιάζονται οι αξιολογήσεις συστάσεων με τα μέτρα αξίας conviction και lift για τους Association Rules.

Το Κεφάλαιο 5 συνιστά την κύρια συνεισφορά της πτυχιακής εργασίας στο αντικείμενο. Διερευνάται η χρήση ενός επιπλέον μέτρου αξίας στη διαδικασία ARM, αυτό της συσχέτισης μεταξύ αριθμητικών διανυσμάτων. Είναι φανερό ότι η διερεύνηση στοχεύει στην ειδική περίπτωση όπου η τεχνική ARM χρησιμοποιείται επί αριθμητικού τύπου δεδομένων. Οι επιδόσεις της εν λόγω προσέγγισης συγκρίνονται με εκείνες των τυπικών μέτρων αξίας στο αποτέλεσμα της ARM.

Στο Κεφάλαιο 6 συνοψίζονται και σχολιάζονται η εμπειρία και τα οφέλη που προέκυψαν από την εκπόνηση της πτυχιακής εργασίας, όπως και των αποτελεσμάτων που έχουν προκύψει από την επεξεργασία δεδομένων των δύο συνόλων δεδομένων που χρησιμοποιήθηκαν: (α) τα δεδομένα MovieLens, και (β) τα βαθμολογικά δεδομένα των ετών 2009-2021 των τμημάτων Μηχανικών Πληροφορικής του πρώην ΑΤΕΙΘ, και του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙΠΑΕ για τους φοιτητές της πληροφορικής.

Στην Ενότητα "Βιβλιογραφία" παρατίθενται στοιχεία των πηγών που έχουν χρησιμοποιηθεί στο πλαίσιο της παρούσας έρευνας.

Κεφάλαιο 2ο : Βασικές έννοιες και τεχνικές

2.1 Εισαγωγή

Σε αυτό το Κεφάλαιο παρουσιάζονται κάποιες βασικές τεχνικές παραγωγής συστάσεων, καθώς και κάποιες μαθηματικές έννοιες. Θα υπάρξει εστίαση στην χρήση των Association Rules και στα μέτρα αξίας και επιδόσεων που αυτοί χρησιμοποιούν, και τέλος θα παρουσιαστούν οι μέθοδοι και οι μετρικές που μπορούν να αξιολογήσουν τα αποτελέσματα που παράγουν οι τεχνικές παραγωγής συστάσεων.

2.2 Τεχνικές παραγωγής συστάσεων

Οι αλγόριθμοι συστημάτων παραγωγής συστάσεων χρησιμοποιούνται για την παραγωγή εξατομικευμένων συστάσεων σε χρήστες. Υπάρχουν διάφοροι τύποι τέτοιων αλγορίθμων, συμπεριλαμβανομένου οι Collaborative Filtering, Content-Based Filtering και Session-Based Filtering.

2.2.1 Collaborative Filtering

Ο αλγόριθμος Collaborative Filtering (CF) αποτελεί μία δημοφιλή τεχνική για την ανάπτυξη συστημάτων παραγωγής συστάσεων. Η βασική ιδέα του είναι να ταυτοποιήσει την ομοιότητα μεταξύ των χρηστών ή των αντικειμένων με βάση παλαιότερες βαθμολογίες, και έπειτα να χρησιμοποιηθεί αυτή η ομοιότητα για την πρόβλεψη των βαθμολογιών που λείπουν από τον χρήστη για τον οποίο πρέπει να γίνει η παραγωγή συστάσεων, δηλαδή να προβλεφθούν οι βαθμολογίες που θα έβαζε στα αντικείμενα αν είχε αλληλεπιδράσει μαζί τους. Τέλος, του γίνονται συστάσεις με βάση τις μεγαλύτερες βαθμολογίες [3],[4].

Το σύνολο των χρηστών πλήθους m ορίζεται ως U = $\{u_1, u_2, \ldots, u_m\}$ και ως I = $\{i_1, i_2, \ldots, i_n\}$ το σύνολο πλήθους n των αντικειμένων. Οι βαθμολογίες αποθηκεύονται σε έναν πίνακα m x n χρήστη-αντικειμένου σε έναν πίνακα βαθμολογιών R = (r_{jl}), όπου η κάθε γραμμή αντιπροσωπεύει έναν χρήστη u_j με 1 ≤ j ≤ m και κάθε στήλη να αντιπροσωπεύει ένα αντικείμενο i_l με 1 ≤ l ≤ n. Κάθε χρήστης u_i έχει μία λίστα αντικειμένων I_u, για τα οποία ο χρήστης έχει εκφράσει το ενδιαφέρον ή την προτίμησή του. Αυτή η προτίμηση δίνεται από τον χρήστη στο αντικείμενο ως βαθμολογία, συνήθως σε ένα αριθμητικό πλαίσιο. Είναι πιθανό μία τιμή I_{ui} να είναι null, δηλαδή να μην έχει δοθεί κάποια βαθμολογία. Ο χρήστης για τον οποίο ο αλγόριθμος επιδιώκει να παράξει συστάσεις ονομάζεται active user και ορίζεται ως u_a∈U [3],[4].

Ένας αλγόριθμος CF χωρίζεται σε δύο στάδια [3],[4]:

- της πρόβλεψης, κατά το οποίο όλες οι βαθμολογίες του active user προβλέπονται ώστε όλο το διάνυσμα I_u να μην περιέχει null τιμή

- των συστάσεων, καθώς με βάση τις βαθμολογίες που προβλέφθηκαν για τον active user του συστήνεται μία λίστα μήκους N, τα οποία είναι τα αντικείμενα που είναι πιο πιθανό να αρέσουν στον χρήστη. Αυτές οι συστάσεις είναι γνωστές ως Top-N συστάσεις.

Η τεχνική αυτή μπορεί να υλοποιηθεί με δύο βασικούς τρόπους: βασισμένη στους χρήστες (UBCF) ή βασισμένη στα αντικείμενα (IBCF) [3],[5].

Ο αλγόριθμος UBCF (User-Based Collaborative Filtering), βασίζεται στην λογική ότι χρήστες με παρόμοια ενδιαφέροντα θα βαθμολογήσουν με παρόμοιο τρόπο τα αντικείμενα. Επομένως, οι null τιμές μπορούν να προβλεφθούν δεδομένου μία γειτονιά παρόμοιων χρηστών (k nearest neighbors) είτε όλων των χρηστών δεδομένου ενός κατωφλιού ομοιότητας. Δημοφιλής μετρικές για τον υπολογισμό την ομοιότητας μεταξύ των χρηστών είναι η pearson correlation coefficient και η cosine similarity [3].

Με την χρήση αυτής της ομοιότητας, ορίζεται η γειτονιά του active user $N(a) \subset U$. Αφού έχουν οριστεί οι χρήστες της γειτονιάς, οι βαθμολογίες τους συγκεντρώνονται για να σχηματίσουν τις προβλέψεις για τον active user. Η πιο εύκολη τακτική είναι να βρεθεί ο μέσος όρος των βαθμολογιών τους για κάθε βαθμολογία αντικειμένου που λείπει για τον active user. Για ένα αντικείμενο il ισχύει ο τύπος της παράστασης (2.1) για τον υπολογισμό της απούσας βαθμολογίας [3]:

$$\hat{r}_{al} = \frac{1}{|\mathcal{N}(a)|} \sum_{i \in \mathcal{N}(a)} r_{il}. \tag{2.1}$$

Ο αλγόριθμος UBCF είναι απλός, αλλά μπορεί να έχει το μειονέκτημα του cold-start, κατά το οποίο οι χρήστες έχουν ελάχιστες ή και καθόλου πληροφορίες για να γίνει η σύγκριση [17]. Δύο επιπλέον βασικά προβλήματα που διαθέτει είναι πως χρειάζεται να κρατάει όλη την βάση δεδομένων των χρηστών στην μνήμη και ή ακρίβεια του υπολογισμού της ομοιότητας μεταξύ του active user και όλων των χρηστών [3].

Ο αλγόριθμος IBCF (Item Based Collaborative Filtering) βασίζεται στην λογική πως οι χρήστες θα προτιμήσουν αντικείμενα που είναι παρόμοια με τα αντικείμενα που ήδη τους αρέσουν. Σε πρώτο βήμα υπολογίζεται ένας πίνακας με τις ομοιότητες μεταξύ όλων των αντικειμένων, δεδομένου ενός μέτρου ομοιότητας. Δημοφιλής επιλογές αποτελούν επίσης οι pearson correlation coefficient και cosine similarity [3].

Όλες οι ομοιότητες μεταξύ των αντικειμένων αποθηκεύονται σε έναν πίνακα n x n σε έναν πίνακα ομοιότητας S. Προκειμένου να μειωθεί το μέγεθος του μοντέλου, για κάθε αντικείμενο αποθηκεύεται στη μνήμη μόνο μία λίστα με τα k πιο όμοια αντικείμενα και τις ομοιότητές τους. Για να υπολογιστούν οι βαθμολογίες των αντικειμένων του active user χρησιμοποιείται ο τύπος της παράστασης (2.2) [3]:

$$\hat{r}_{al} = \frac{1}{\sum_{i \in \mathcal{S}(l)} s_{li}} \sum_{i \in \mathcal{S}(l)} s_{li} r_{ai} \tag{2.2}$$

Ο αλγόριθμος IBCF, διαθέτει επίσης το μειονέκτημα του cold-start στην περίπτωση που ένα αντικείμενο εισέρχεται για πρώτη φορά στο σύστημα [6]. Η αποθήκευση όμως μόνο k ομοιοτήτων για κάθε αντικείμενο βελτιώνει τον χρόνο απόδοσης και τον χώρο μνήμης σημαντικά, θυσιάζοντας όμως πιθανώς την ποιότητα των συστάσεων [3].

Το πακέτο RecommenderLab δίνει την δυνατότητα χρήσης των αλγορίθμων UBCF και IBCF με την μέθοδο των πλησιέστερων γειτόνων, που η προεπιλογή τους είναι για τον UBCF οι 25 πλησιέστεροι γείτονες και για τον IBCF οι 30. Για τον υπολογισμό της ομοιότητας μεταξύ τους μπορούν να χρησιμοποιηθούν οι cosine similarity που αποτελεί και την προεπιλεγμένη μέθοδο υπολογισμού ομοιότητας [7],[8], η pearson correlation και η jaccard similarity που χρησιμοποιείται την περίπτωση που τα δεδομένα είναι δυαδικής μορφής. Ο χρήστης της βιβλιοθήκης μπορεί να επιλέξει να τροποποιήσει αυτές τις τιμές έτσι ώστε να προσαρμόσει τα μοντέλα σύμφωνα με τις απαιτήσεις της έρευνας που επιθυμεί να πραγματοποιήσει [3].

2.2.2 Content Based Filtering

Ο αλγόριθμος Content-Based Filtering είναι ένας ακόμα δημοφιλής αλγόριθμος που χρησιμοποιείται στα συστήματα παραγωγής συστάσεων. Σε αντίθεση με τον Collaborative Filtering, που βασίζεται στις αλληλεπιδράσεις χρηστών-αντικειμένων για να δημιουργήσει συστάσεις, ο Content-Based Filtering προτείνει αντικείμενα με βάση την ομοιότητά τους με άλλα αντικείμενα για το οποία ο χρήστης στο παρελθόν έχει παρουσιάσει ενδιαφέρον [9-11].

Η βασική ιδέα του αλγορίθμου είναι να ταυτοποιήσει τα βασικά χαρακτηριστικά ενός αντικειμένου και να τα χρησιμοποιήσει για να δημιουργήσει στον χρήστη ένα προφίλ. Αυτό το προφίλ ορίζει την βαρύτητα των διαφόρων αυτών χαρακτηριστικών για τον συγκεκριμένο χρήστη. Έτσι, αυτή η βαρύτητα χρησιμοποιείται για να κατατάξει τα αντικείμενα που περιέχουν αυτά τα χαρακτηριστικά [9-11].

Για παράδειγμα, ένας χρήστης Α έχει παρουσιάσει ενδιαφέρον για ταινίες δράσεις στο παρελθόν. Με τη χρήση του Content-Based Filtering αλγορίθμου, θα μπορούσε να δημιουργηθεί ένα προφίλ για τον χρήστη Α με μεγάλη βαρύτητα στα χαρακτηριστικά “δράση”, “μυστήριο” και “συναρπαστικό”. Τότε θα γίνονταν η αναζήτηση στην βάση δεδομένων με ταινίες που έχουν αυτά τα χαρακτηριστικά και θα αποτελούσαν πιθανές συστάσεις για αυτόν τον χρήστη.

Για να υλοποιηθεί ο συγκεκριμένος αλγόριθμος σε σύστημα παραγωγής συστάσεων, είναι απαραίτητο να γίνει πρώτα η ταυτοποίηση των χαρακτηριστικών των αντικειμένων και η βαρύτητά τους. Μόλις οριστούν αυτά τα χαρακτηριστικά, χρειάζεται να υπολογιστεί η ομοιότητά τους με άλλα αντικείμενα με βάση αυτά τα χαρακτηριστικά και τις τιμές βαρύτητάς τους [9-11].

Ένα πλεονέκτημα του Content-Based Filtering αλγορίθμου είναι ότι μπορεί να δημιουργήσει συστάσεις ακόμα και σε καινούριους χρήστες ή χρήστες με πολύ αραιά δεδομένα. Εφόσον βασίζεται μόνο στην ομοιότητα μεταξύ των αντικειμένων και δεν απαιτεί κάποια σημαντική αλληλεπίδραση μεταξύ χρηστών και αντικειμένων, μπορεί να δημιουργήσει συστάσεις βασιζόμενος αποκλειστικά στις προτιμήσεις που έχει δηλώσει ο χρήστης [12].

Ωστόσο, ένα πιθανό μειονέκτημα του αλγορίθμου είναι το “over-specialization”. Εφόσον ο αλγόριθμος προτείνει αντικείμενα που είναι παρόμοια με αυτά που ο χρήστης έχει εκφράσει ενδιαφέρον στο παρελθόν, μπορεί να αποτύχει στο να παράξει προβλέψεις με αντικείμενα που είναι εκτός των γνωστών προτιμήσεων του χρήστη [12]. Για παράδειγμα, αν ένας χρήστης έχει παρουσιάσει ενδιαφέρον μόνο σε ταινίες δράσης, ίσως να μην του γίνει ποτέ πρόταση κάποιας κωμωδίας ή κάποιου δράματος, ακόμα κι αν του αρέσουν.

Ένα ακόμα πιθανό μειονέκτημα είναι το πρόβλημα του cold-start. Εφόσον ο αλγόριθμος βασίζεται στα χαρακτηριστικά των αντικειμένων για να παράξει συστάσεις, μπορεί να αντιμετωπίσει δυσκολία

στο να προτείνει νέα αντικείμενα τα οποία δεν έχουν κατηγοριοποιηθεί ακόμα με βάση τα χαρακτηριστικά τους [9],[10].

2.2.3 Session-Based Filtering

Τα Session-Based συστήματα παραγωγής συστάσεων είναι μία ειδική κατηγορία που στοχεύουν στην παροχή εξατομικευμένων συστάσεων σε χρήστες με βάση την τρέχουσα συνεδρία τους. Αυτά τα συστήματα είναι ιδιαίτερα χρήσιμα σε περιπτώσεις που τα ενδιαφέροντα και οι προτιμήσεις των χρηστών μπορούν να αλλάξουν σε σύντομο χρονικό διάστημα, όπως σε σελίδες διαδικτυακών αγορών, σε πλατφόρμες με άρθρα ή υπηρεσίες μουσικής. Αντί να βασίζονται στα παλαιότερα δεδομένα του χρήστη, τα Session-Based συστήματα αναλύουν την τωρινή συμπεριφορά του χρήστη και προσπαθούν να ταυτοποιήσουν μοτίβα ή συσχετίσεις μεταξύ των ενεργειών τους και των αντικειμένων με τα οποία αλληλεπιδρούν [13],[14].

Μία δημοφιλής μέθοδος που χρησιμοποιείται στα Session-Based συστήματα παραγωγής συστάσεων είναι ο αλγόριθμος συστάσεων Association Rules. Η εξόρυξη δεδομένων με Association Rules είναι μία τεχνική που χρησιμοποιείται για να ανακαλύψει ενδιαφέρονται και χρήσιμα μοτίβα ή σχέσεις μεταξύ αντικειμένων σε ένα σύνολο δεδομένων. Στα πλαίσια των συστημάτων παραγωγής συστάσεων, οι Association Rules μπορούν να χρησιμοποιηθούν για να ταυτοποιήσουν τα σύνολα αντικειμένων που εμφανίζονται συχνά μαζί στις συναλλαγές του χρήστη με το σύστημα και να προτείνουν σχετικά αντικείμενα που μπορεί να ενδιαφέρουν τον χρήστη [13],[14].

Ένα από τα πλεονεκτήματα των Session-Based συστημάτων παραγωγής συστάσεων είναι ότι μπορούν να προσαρμοστούν γρήγορα στις αλλαγές των προτιμήσεων και των ενδιαφερόντων του χρήστη. Καθώς αλλάζει η συμπεριφορά του μέσα στο σύστημα, οι συστάσεις του θα προσαρμοστούν αναλόγως. Επιπλέον, αυτά τα συστήματα δεν βασίζονται στα παρελθοντικά δεδομένα του χρήστη, επομένως μπορεί να είναι εξίσου αποτελεσματικά με τους νέους χρήστες που δεν διαθέτουν ακόμα ιστορικό αλληλεπίδρασης με το σύστημα [13],[14].

2.3 Similarity Measures

Τα μέτρα ομοιότητας (similarity measures) αποτελούν στην επιστήμη των δεδομένων έναν τρόπο για την ποσοτικοποίηση της ομοιότητας μεταξύ δύο μεταβλητών. Ο υπολογισμός της ομοιότητας στα συστήματα παραγωγής συστάσεων υπολογίζεται είτε μεταξύ των χρηστών, που ορίζονται ως τα διανύσματα των βαθμολογιών που έχουν αποδώσει, είτε μεταξύ των αντικειμένων, που ορίζονται ως τα διανύσματα των βαθμολογιών που τους έχουν αποδοθεί [3].

Η λογική ακολουθείται από τους δύο βασικότερους αλγορίθμους Collaborative Filtering που χρησιμοποιούνται από τα συστήματα παραγωγής συστάσεων, τους UBCF και IBCF. Για παράδειγμα, στην περίπτωση του UBCF, υπολογίζεται η ομοιότητα μεταξύ δύο χρηστών, ενώ στην περίπτωση του IBCF, υπολογίζεται η ομοιότητα μεταξύ δύο αντικειμένων [3],[6].

Για τον υπολογισμό αυτής της ομοιότητας μεταξύ δύο μεταβλητών μπορούν να χρησιμοποιηθούν διάφορες μετρικές υπολογισμού ομοιότητας, με τις πιο γνωστές να είναι η cosine similarity για αριθμητικά δεδομένα και η jaccard similarity για δυαδικά δεδομένα [3].

2.3.1 Cosine Similarity

Η cosine similarity είναι μία μετρική που χρησιμοποιείται για τον υπολογισμό της ομοιότητας μεταξύ δύο μεταβλητών. Πιο συγκεκριμένα, υπολογίζει την ομοιότητα στην κατεύθυνση των μεταβλητών. Η ομοιότητα των δύο μεταβλητών υπολογίζεται από το συνημίτονο της μεταξύ τους γωνίας [15].

Οι ομοιότητες μεταξύ δύο διανυσμάτων A και B ορίζονται για την cosine similarity στην παράσταση (2.3) [3],[16]:

$$cosine\ similarity = |A||B|cos\theta = \frac{A \cdot B}{|A||B|} = \frac{\sum_i^n A_i B_i}{\sqrt{\sum_i^n A_i^2}\sqrt{\sum_i^n B_i^2}} \tag{2.3}$$

Η μετρική μπορεί να πάρει τιμές από -1 μέχρι 1, όποιο το 1 δηλώνει ότι τα διανύσματα είναι εντελώς όμοια, το 0 δηλώνει πως δεν σχετίζονται μεταξύ τους και το -1 πως είναι αντίθετα μεταξύ τους [16].

Ένα από τα βασικά πλεονεκτήματα της cosine similarity είναι ότι δεν λαμβάνει υπόψη τα δεδομένα που είναι μηδενικά και στα δύο διανύσματα. Αυτό είναι σημαντικό, καθώς σε αραιά δεδομένα, ο υπολογισμός της ομοιότητας συμπεριλαμβάνοντας αυτά τα στοιχεία θα οδηγούσε σε αυξημένο βαθμό ομοιότητας χωρίς να είναι στην πραγματικότητα δεδομένα που προσφέρουν κάποια πληροφορία.

2.3.2 Jaccard Similarity

Η jaccard similarity είναι μία ακόμα μετρική ομοιότητας μεταξύ δύο μεταβλητών που χρησιμοποιείται από τα συστήματα παραγωγής συστάσεων. Σε αντίθεση όμως με τις παραπάνω μετρικές που απευθύνονταν σε αριθμητικά σύνολα δεδομένων, η jaccard similarity χρησιμοποιείται για να υπολογίσει ομοιότητες σε δεδομένα δυαδικής μορφής. Αυτό μπορεί να συμβαίνει για διάφορους λόγους, όπως για παράδειγμα οι χρήστες να μην επιθυμούν να φανερώσουν τις προτιμήσεις τους με την προσθήκη κάποιας βαθμολογίας ή απλά η φύση των δεδομένων να απαιτεί να είναι δυαδικά, όπως η περίπτωση του να έχει βάλει ή να μην έχει βάλει στο καλάθι αγορών του ο χρήστης ένα αντικείμενο, και επομένως η δυαδική μορφή των δεδομένων να δηλώνει εάν απλά το συγκεκριμένο αντικείμενο έχει χρησιμοποιηθεί από έναν συγκεκριμένο χρήστη [3].

Σε δεδομένα δυαδικής μορφής, η τιμή true δηλώνει πως ο χρήστης γνωρίζει και έχει προτίμηση προς το συγκεκριμένο αντικείμενο, ενώ η τιμή false όλες τις άλλες περιπτώσεις, όπως ότι ο χρήστης δεν γνωρίζει για το αντικείμενο, ότι το γνωρίζει και δεν το έχει χρησιμοποιήσει ακόμα, ή ότι δεν προτιμάει το συγκεκριμένο αντικείμενο. Η jaccard similarity διαχειρίζεται τον κάθε χρήστη ως ένα σύνολο που περιλαμβάνει όλα τα αντικείμενα που έχει βαθμολογήσει ως true, ενώ το κάθε αντικείμενο ως ένα σύνολο με όλους τους χρήστες που το έχουν βαθμολογήσει ως true [3].

Στην βιβλιοθήκη RecommenderLab, οι αλγόριθμοι Collaborative Filtering, UBCF και IBCF, μπορούν να χρησιμοποιηθούν και για την παραγωγή συστάσεων σε δυαδικά σύνολα δεδομένων, κάνοντας χρήση της jaccard similarity. Η jaccard similarity μεταξύ δύο συνόλων x και y, δηλαδή μεταξύ δύο χρηστών ή μεταξύ δύο αντικειμένων, υπολογίζεται ως τα κοινά στοιχεία των συνόλων προς την ένωση των στοιχείων τους, όπως φαίνεται στην παράσταση (2.4) [3]:

$$\mathrm{sim}_{\mathrm{Jaccard}}(\mathcal{X}, \mathcal{Y}) = \frac{|\mathcal{X} \cap \mathcal{Y}|}{|\mathcal{X} \cup \mathcal{Y}|} \tag{2.4}$$

Σε σχέση με την cosine similarity, την pearson correlation και την spearman correlation, η jaccard similarity έχει πιο περιορισμένες δυνατότητες χρήσης στα συστήματα παραγωγής συστάσεων, καθώς περιορίζεται σε δεδομένα δυαδικής μορφής [20]. Ωστόσο, παραμένει μία πολύ χρήσιμη μετρική στις περιπτώσεις που είναι κατάλληλη να χρησιμοποιηθεί.

2.4 Correlation Coefficients

Οι συντελεστές συσχέτισης (Correlation Coefficients) αποτελούν στην επιστήμη των δεδομένων έναν τρόπο για την ποσοτικοποίηση της εξάρτησης της σχέσης μεταξύ δύο μεταβλητών, δηλαδή κατά πόσο η μεταβολή της μίας μεταβλητής προκαλεί μεταβολή στην άλλη μεταβλητή. Ο υπολογισμός της συσχέτισης στα συστήματα παραγωγής συστάσεων υπολογίζεται είτε μεταξύ των χρηστών, που ορίζονται ως τα διανύσματα των βαθμολογιών που έχουν αποδώσει, είτε μεταξύ των αντικειμένων, που ορίζονται ως τα διανύσματα των βαθμολογιών που τους έχουν αποδοθεί [3].

Για τον υπολογισμό αυτής της συσχέτισης μεταξύ δύο μεταβλητών μπορούν να χρησιμοποιηθούν διάφορες μετρικές, με τις πιο γνωστές να είναι η pearson correlation και η spearman correlation [3].

2.4.1 Pearson Correlation

Η pearson correlation coefficient (r) είναι η πιο γνωστή μετρική υπολογισμού γραμμικού συσχετισμού. Περιγράφει την δύναμη και την κατεύθυνση στην σχέση μεταξύ δύο μεταβλητών [17].

Η συσχέτιση μεταξύ δύο διανυσμάτων x και y ορίζονται για την pearson correlation στην παράσταση (2.5), όπου x_i, y_i ένα στοιχείο των διανυσμάτων και $\bar{x}$, $\bar{y}$ ο μέσος όρος των διανυσμάτων [3],[16]:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \tag{2.5}$$

Η μετρική μπορεί επίσης να πάρει τιμές από -1 μέχρι 1, όπου το 1 δηλώνει ότι τα διανύσματα έχουν θετική συσχέτιση μεταξύ τους, το 0 δηλώνει πως δεν σχετίζονται μεταξύ τους και το -1 πως έχουν αρνητική συσχέτιση μεταξύ τους [16].

Η επιλογή της χρήσης μεταξύ της pearson correlation και ακόμα περισσοτέρων μετρικών, εξαρτάται από τα χαρακτηριστικά του συνόλου δεδομένων που εξετάζεται και τους στόχους που συστήματος παραγωγής συστάσεων, και η επιλογή της καθεμιάς από αυτές τις μετρικές θα μπορούσε να έχει σημαντικό αντίκτυπο στα αποτελέσματα του αλγόριθμου.

2.4.2 Spearman Correlation

Η spearman correlation coefficient (ρ), γνωστή και ως spearman's rank correlation coefficient, είναι μία ακόμα μετρική για να υπολογίσει την συσχέτιση στην σχέση μεταξύ δύο διανυσμάτων. Χρησιμοποιεί τάξεις και δεν βασίζεται στην κατανομή των δεδομένων. Αντιθέτως, αξιολογεί την δύναμη και την κατεύθυνση της μονοτονικής σχέσης μεταξύ των μεταβλητών. Η μονοτονική σχέση αναφέρεται στην συνεχή αύξηση ή μείωση στις τιμές μιας μεταβλητής σε σχέση με την άλλη [18].

Για τον υπολογισμό της spearman correlation, για δεδομένα πρώτα κατατάσσονται για κάθε διάνυσμα. Οι κατατάξεις αναπαριστούν την θέση των δεδομένων όταν κατατάσσονται από την μεγαλύτερη στην μικρότερη τιμή [18].

Αφού αποκτήσουν τις κατατάξεις, η διαφορά των κατατάξεων για κάθε ζευγάρι των δεδομένων των δύο μεταβλητών υπολογίζεται και τετραγωνίζεται, και η spearman correlation υπολογίζεται από τον τύπο στην παράσταση (2.6), όπου d^2 η διαφορά των κατατάξεων x_i-y_i στο τετράγωνο, και n το άθροισμα των παρατηρήσεων των διανυσμάτων [18],[19]:

$$\rho = 1 - \frac{6\Sigma\, \mathrm{d}_i^2}{n(n^2-1)} \tag{2.6}$$

Η τιμή της spearman correlation παίρνει τιμές από -1 μέχρι 1, όπως και στην pearson correlation. Ωστόσο, η spearman correlation αξιολογεί μονοτονικές σχέσεις αντί για γραμμικές. Μία τιμή r κοντά στο 1 δηλώνει ισχυρή μονοτονική σχέση, δηλαδή ότι οι υψηλές κατατάξεις μιας μεταβλητής αντιστοιχούν στις υψηλές κατατάξεις της άλλης μεταβλητής. Μία τιμή κοντά στο -1 δηλώνει αρνητική μονοτονική σχέση, δηλαδή ότι οι υψηλές κατατάξεις της μιας μεταβλητής αντιστοιχούν στις χαμηλές κατατάξεις της άλλης. Τέλος, μία τιμή ρ κοντά στο 0 δηλώνει αδύναμη ή όχι μονοτονική σχέση μεταξύ των δύο μεταβλητών [18].

Ένα πλεονέκτημα της είναι ότι, σε αντίθεση με την pearson correlation που απευθύνεται μόνο σε συνεχής τιμές, η spearman correlation μπορεί να χρησιμοποιηθεί για συνεχής αλλά και για διακριτές τιμές δεδομένων. Αποτελεί επίσης μετρική που μπορεί να χρησιμοποιηθεί όταν χρειάζεται να υπολογιστεί η συσχέτιση σε ποιοτικά χαρακτηριστικά που δεν μπορούν να υπολογιστούν ποσοτικά αλλά μπορούν να ταξινομηθούν σειριακά [19].

Ένα από τα μειονεκτήματα της spearman correlation coefficient είναι πως με την διαδικασία της απόδοσης κατάταξης στα δεδομένα μπορεί να χαθεί σημαντική πληροφορία, και σε περίπτωση που τα δεδομένα είναι κανονικής κατανομής, είναι λιγότερο ισχυρή από την pearson correlation [19].

2.5 Association Rules

Η Association Rule Mining (ARM) αποτελεί μία σημαντική τεχνική στην εξόρυξη δεδομένων, που στοχεύει στην εξαγωγή χρήσιμων συσχετίσεων μεταξύ αντικειμένων σε ένα σύνολο δεδομένων. Εφαρμόζεται ευρέως σε τηλεπικοινωνιακά δίκτυα, στην αγορά, στην καταγραφή εμπορευμάτων κλπ. [21]

Η τεχνική διαχειρίζεται κατηγορικά δεδομένα σε μορφή transactions (συναλλαγών), τα οποία αναφέρονται σε μία συλλογή δεδομένων ή περιπτώσεων όπου η κάθε καταγραφή αναπαριστά ένα σύνολο αντικειμένων που σχετίζονται μεταξύ τους. Τυπικά, τα transactional data στην τεχνική ARM, όπως φαίνεται και στην Εικόνα 2.1, περιλαμβάνουν [22]:

- Τα transactions είναι μία λίστα αντικειμένων τα οποία αγοράστηκαν ή επιλέχθηκαν μαζί
- Τα items τα οποία είναι μεμονωμένα τα προϊόντα που εμφανίζονται σε ένα transaction
- Τα transaction IDs, που αποτελούν για κάθε συναλλαγή μία ξεχωριστή ταυτότητα

Transaction ID	Items
1	{bread, milk, eggs}
2	{milk, cheese}
3	{bread, eggs}
4	{milk, bread, eggs}
5	{cheese}

Πίνακας 2.1 Τυπική δομή transactional data κατηγορικού τύπου στην τεχνική ARM

Μια δημοφιλής εφαρμογή της ARM αποτελεί η Market Basket Analysis που επικεντρώνεται στην ανάλυση transactional data στο εμπόριο και στις διαδικτυακές αγορές. Στοχεύει συγκεκριμένα στην ανάλυση και την εύρεση διαφόρων μοτίβων στην συμπεριφορά των καταναλωτών, ταυτοποιώντας τα αντικείμενα που αγοράζονται συχνά μαζί. Τα αποτελέσματα της εφαρμογής αυτής χρησιμοποιούνται στη λήψη στρατηγικών αποφάσεων στην αγορά, όπως την βελτιστοποίηση της θέσης των αντικειμένων, cross-selling και την δημιουργία προσφορών προς όφελος του καταναλωτή [23].

Δεδομένου ενός συνόλου πλήθους m αντικειμένων I=I_1, I_2, … , I_m, το T δηλώνει ένα transaction που περιέχει ένα σύνολο αντικειμένων έτσι ώστε T⊆ I. Ένας Association Rule είναι της μορφής X⇒Y, όπου X, Y ⊂ I και είναι σύνολα αντικειμένων που ονομάζονται στοιχειοσύνολα (itemsets), και X∩ Y =∅. Το X ονομάζεται σώμα του κανόνα (antecedent/body) ενώ το Y ονομάζεται κεφαλή του κανόνα (consequent/head), και ο κανόνας υποδηλώνει πως το X προκαλεί το Y [21].

Η ARM βρίσκει τους Association Rules οι οποίοι ικανοποιούν ένα προκαθορισμένο κατώφλι support και confidence, δύο σημαντικών μέτρων αξίας των κανόνων, δεδομένου ενός σύνολο δεδομένων. Εφόσον οριστούν από τον χρήστη οι τιμές αυτών των δύο κατωφλιών, οι κανόνες οι οποίοι δεν τις πληρούν θεωρούνται πως δεν είναι χρήσιμοι και εγκαταλείπονται [21].

Ανάλογα με τις τιμές των διαφόρων μέτρων αξίας, ορίζεται η αξία και η χρησιμότητα των κανόνων.

2.5.1 Interestingness Measures

Μερικές από τις πιο γνωστές μετρικές για να αξιολογήσουν την αξία και το πόσο ισχυρός είναι ένας κανόνας είναι η support, η confidence, η lift και η conviction.

- **Support**

Το μέτρο support υπολογίζει πόσο συχνά ένα στοιχειοσύνολο εμφανίζεται στο σύνολο των δεδομένων. Για έναν κανόνα X⇒Y, ορίζεται ως ο αριθμός όλων των συναλλαγών που περιέχουν τα στοιχειοσύνολα X και Y μαζί, προς τον αριθμό όλων των συναλλαγών του συνόλου δεδομένων, όπως φαίνεται στην παράσταση (2.7) [24],[25] :

$$\text{Support}(X \Rightarrow Y) = \frac{\text{Number of transactions in which X and Y appear}}{\text{Total number of transactions}} \quad (2.7)$$

Η support είναι μία σημαντική μετρική στην ARM καθώς βοηθάει στην ταυτοποίηση των συχνά εμφανιζόμενων στοιχειοσυνόλων στο σύνολο των δεδομένων, και επομένως είναι πιθανό να είναι πολύ πιο χρήσιμα στο να προκύψουν συστάσεις [25].

Ωστόσο, είναι σημαντικό να σημειωθεί πως η υψηλή τιμή support δεν σημαίνει απαραίτητα πως υπάρχει ισχυρή συσχέτιση μεταξύ των αντικειμένων, καθώς κάποια συχνά εμφανιζόμενα στοιχειοσύνολα μπορεί να προκύπτουν τυχαία. Επομένως, η support συχνά χρησιμοποιείται σε συνδυασμό με άλλες μετρικές για να ταυτοποιηθούν οι ισχυρές συσχετίσεις [24].

- **Confidence**

Η confidence είναι ένα σημαντικό μέτρο στην ARM στα συστήματα παραγωγής συστάσεων. Υπολογίζει τις πιθανότητες των αντικειμένων να εμφανιστούν στην κεφαλή του κανόνα δεδομένου την παρουσία των αντικειμένων στο σώμα του κανόνα. Δηλαδή, υπολογίζει το ποσοστό των φορών που τα αντικείμενα της κεφαλής εμφανίστηκαν σε συναλλαγές μαζί με αντικείμενα του σώματος του κανόνα [25].

Η confidence για έναν κανόνα X⇒Y ορίζεται ως ο αριθμός των συναλλαγών που περιέχουν τα X και Y μαζί, προς τον αριθμό των συναλλαγών που περιέχουν μόνο το X, όπως φαίνεται στην παράσταση (2.8) [25],[26]:

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Number of transactions in which X and Y appear}}{\text{Number of transactions in which X appears}} \quad (2.8)$$

Υψηλή τιμή confidence δηλώνει πως τα αντικείμενα της κεφαλής είναι πολύ πιθανό να προκύψουν ως συστάσεις δεδομένου την παρουσία των αντικειμένων στο σώμα του κανόνα.

Όπως η support, έτσι και η confidence μπορεί να χρησιμοποιηθεί για να αποκλειστούν αδύναμοι κανόνες. Τυπικά, όταν ένα κατώφλι για την confidence ορίζεται, τότε θεωρούνται έγκυροι μόνο οι κανόνες που έχουν μεγαλύτερη τιμή confidence [25].

Παρόλα αυτά, οι τιμές support και confidence μπορεί να αποτελέσουν θόρυβο στην περίπτωση που πρέπει να αναζητηθεί κατά πόσο το σώμα προκαλεί την εμφάνιση της κεφαλής. Για παράδειγμα, ένας κανόνας {bread ⇒ eggs} μπορεί να έχει υψηλή τιμή support καθώς αυτά τα δύο αγαθά εμφανίζονται συχνά μαζί στο σύνολο των συναλλαγών σε μία βάση δεδομένων, και μπορεί ο κανόνας να έχει επίσης υψηλή τιμή confidence καθώς στην πλειοψηφία των συναλλαγών που έχει αγοραστεί το ψωμί, έχουν αγοραστεί μαζί και τα αυγά. Αυτό όμως δεν σημαίνει πως η αγορά του ψωμιού είναι αυτή που προκαλεί την αγορά των αυγών. Οι μετρικές οι οποίες μπορούν να δηλώσουν εξάρτηση μεταξύ του σώματος και της κεφαλής είναι η lift και η conviction.

- **Lift**

Η lift είναι ένα μέτρο που χρησιμοποιείται στην ARM για να υπολογίσει την δύναμη στην συσχέτισης μεταξύ δύο στοιχειοσυνόλων και σε τι βαθμό η παρουσία του ενός επηρεάζει την πιθανότητα να εμφανιστεί το άλλο σε μια συναλλαγή. Για έναν κανόνα X⇒Y ,ορίζεται ως το κλάσμα της support των

X και Y, προς το support του X πολλαπλασιασμένο με το support του Y, όπως φαίνεται στην παράσταση (2.9) [25],[27]:

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{support}(X,Y)}{\text{support}(X) * \text{support}(Y)} \tag{2.9}$$

Μία τιμή lift μεγαλύτερη του 1 δηλώνει πως δύο στοιχειοσύνολα συσχετίζονται θετικά και εμφανίζονται μαζί πιο συχνά από ότι αναμένεται. Μία τιμή lift μικρότερη του 1 δηλώνει ότι τα στοιχειοσύνολα συσχετίζονται αρνητικά μεταξύ τους και εμφανίζονται μαζί λιγότερο συχνά από ότι αναμένεται. Μία τιμή lift ίση με το 1 δηλώνει πως δεν υπάρχει συσχέτιση μεταξύ των δύο στοιχειοσυνόλων [25],[27].

Το μειονέκτημα του μέτρου lift είναι πως διαθέτει την αντιμεταθετική ιδιότητα, δηλαδή μία τιμή lift για έναν κανόνα X⇒Y είναι ίση με την lift ενός κανόνα Y⇒X. Αυτό σημαίνει πως η τιμή lift ενός κανόνα δεν δηλώνει κατεύθυνση ως προς την αιτιότητα (causality), επομένως είναι ελλιπής η πληροφορία για το αν προκαλεί το σώμα την κεφαλή ή η κεφαλή το σώμα. Μπορεί δηλαδή ένας ισχυρός σε αιτιότητα κανόνας να έχει την ίδια lift με έναν μη ισχυρό σε αιτιότητα κανόνα επειδή είναι αντιμεταθετικοί. Το μειονέκτημα αυτό θα φανεί στην απόδοση της ταξινόμησης των Association Rules με την τιμή lift έναντι της conviction.

- **Conviction**

Η conviction είναι ένα ακόμα μέτρο που χρησιμοποιείται στην ARM και θα χρησιμοποιηθεί επίσης στην συνέχεια της εργασίας. Υπολογίζει το πόσο σημαντικός είναι ένας κανόνας και αξιολογεί την δύναμη της συσχέτισης μεταξύ του σώματος και της κεφαλής [28].

Η conviction υπολογίζει τον βαθμό εξάρτησης μεταξύ σώματος και κεφαλής, λαμβάνοντας υπόψη την συχνότητα εμφάνισης του σώματος και την συχνότητα που δεν εμφανίζεται η κεφαλή. Για έναν κανόνα X⇒Y ορίζεται όπως στον τύπο της παράστασης (2.10) [28]:

$$\text{conviction}(X \Rightarrow Y) = \frac{1\text{-support}(Y)}{1\text{-confidence}(X \Rightarrow Y)} \tag{2.10}$$

Η τιμή της conviction μπορεί να ανήκει από το 0 ως το άπειρο, με την τιμή 1 να δηλώνει πως η κεφαλή και το σώμα είναι ανεξάρτητα μεταξύ τους. Με τιμή μεγαλύτερη του 1 δηλώνεται πως θετική συσχέτιση, δηλαδή σημαίνει πως η παρουσία του σώματος καθιστά πιθανή την παρουσία της κεφαλής και όσο μεγαλύτερη η τιμή της conviction, τόσο μεγαλύτερη είναι η εξάρτηση μεταξύ των δύο. Τέλος, τιμή conviction κάτω του 1 δηλώνει αρνητική συσχέτιση, δηλαδή ότι η παρουσία του σώματος καθιστά λιγότερο πιθανή την παρουσία της κεφαλής [29].

Η conviction είναι μία χρήσιμη μετρική, καθώς η υψηλή τιμή της μπορεί να ταυτοτοποιήσει ότι ένας κανόνας δεν έχει προκύψει τυχαία, και επομένως να μπορεί να θεωρηθεί ισχυρός κανόνας, και επίσης δηλώνει την κατεύθυνση της αιτιότητας [29].

2.5.2 Apriori Algorithm

Ο αλγόριθμος Apriori είναι ένας από τους πιο γνωστούς και δημοφιλής αλγορίθμους για την εξόρυξη συχνών στοιχειοσυνόλων και κανόνων συσχέτισης από σύνολα κατηγορικών δεδομένων με την μορφή transactions. Ο αλγόριθμος ανακαλύπτει τα συχνά εμφανιζόμενα στοιχειοσύνολα επαναλαμβάνοντας μία διαδικασία σε υποσύνολα αντικειμένων στο σύνολο των δεδομένων. Ονομάζεται Apriori καθώς χρησιμοποιεί μία εκ των προτέρων γνώση των συχνών στοιχειοσυνόλων για να περιορίσει τον χώρο της έρευνάς του και βελτιώσει την αποτελεσματικότητά του [25],[30].

Ο αλγόριθμος στην αρχή σκανάρει το σύνολο δεδομένων για να καθορίσει το μέτρο support του κάθε αντικειμένου. Τα αντικείμενα που δεν καταφέρνουν να φτάσουν το κατώφλι της support που έχει οριστεί, εξαλοίφονται από την διαδικασία. Τα αντικείμενα που έχουν παραμείνει αποτελούν τα συχνά εμφανιζόμενα αντικείμενα και είναι υποψήφια για να δημιουργήσουν στοιχειοσύνολα μεγέθους δύο αντικειμένων, τα οποία επίσης φιλτράρονται με βάση την κατώτατη τιμή support. Αυτή η διαδικασία επαναλαμβάνεται, δημιουργώντας υποψήφια στοιχειοσύνολα μεγέθους k από τα μεμονωμένα συχνά εμφανιζόμενα ακτικείμενα μέχρι να μην μπορούν να δημιουργηθούν άλλα στοιχειοσύνολα [25],[30].

Στον Apriori αλγόριθμο είναι σημαντική η χρήση της "Apriori" ιδιότητάς του, η οποία δηλώνει πως οποιοδήποτε υπερσύνολο ενός μη συχνά εμφανιζόμενου στοιχειοσυνόλου πρέπει επίσης να είναι μη συχνά εμφανιζόμενο. Αυτή η ιδιότητα επιτρέπει στον αλγόριθμο να αποφεύγει την δημιουργία υποψήφιων στοιχειοσυνόλων που εγγυημένα δεν θα είναι συχνά εμφανιζόμενα, το οποίο μπορεί να μειώσει σημαντικά τον αριθμό των στοιχειοσυνόλων που θα χρειαζόταν να ληφθούν υπόψη [30].

Αφού ταυτοποιηθούν τα συχνά εμφανιζόμενα στοιχειοσύνολα, το επόμενο βήμα είναι να εξορυχθούν κανόνες συσχέτισης από αυτά. Οι κανόνες παράγονται λαμβάνοντας υπόψη όλα τα πιθανά υποσύνολα των συχνά εμφανιζόμενων στοιχειοσυνόλων και υπολογίζοντας τις τιμές support και confidence. Οι κανόνες τότε φιλτράρονται με βάση τα κατώφλια των μετρικών που έχει επιλέξει ο χρήστης [30].

Ένα από τα βασικά πλεονεκτήματα του Apriori είναι η απλότητα και η ευκολία της υλοποίησής του, καθώς και το γεγονός πως οι κανόνες είναι εύκολα κατανοητοί από τους ανθρώπους. Επίσης, είναι ευέλικτος και είναι εύκολο να δημιουργηθούν επεκτάσεις ανάλογα με το πρόβλημα που πρέπει να επιλυθεί [44].

Κάποια από τα μειονεκτήματα του αλγορίθμου αποτελούν η υπολογιστική πολυπλοκότητα του, καθώς και ο υψηλός χρόνος και μνήμη που καταναλώνει κατά την εκτέλεσή του. Επιπλέον, οι ικανότητες του είναι περιορισμένες στην ανακάλυψη περίπλοκων μοτίβων καθώς και στην διαχείριση των αριθμητικών δεδομένων, καθώς διαχειρίζεται τα δεδομένα ως συναλλαγές κατηγορικού τύπου [30]. Τέλος, υπάρχει μία μεγάλη κλίση και εξάρτηση στην τιμή support των στοιχειοσυνόλων.

Ο αλγόριθμος Apriori χρησιμοποιείται ευρέως σε πολλές εφαρμογές, όπως και σε συστήματα παραγωγής συστάσεων. Χρησιμοποιείται επίσης από την βιβλιοθήκη RecommenderLab στην παραγωγή συστάσεων με την χρήση των Association Rules [3],[31].

2.6 Evaluation Schemes

Προκειμένου να κριθεί αν ένας αλγόριθμος παραγωγής συστάσεων που ακολουθήθηκε ήταν κατάλληλος ή όχι, χρειάζεται να περάσει από το στάδιο της αξιολόγησης. Σε αυτό το στάδιο, με την χρήση ενός μοντέλου αξιολόγησης (evaluation scheme), ο αλγόριθμος θα χρησιμοποιήσει κάποια από

τα δεδομένα ως training set για την εκπαίδευση του αλγόριθμου και θα παράξει συστάσεις για τα υπόλοιπα, νέα για τον αλγόριθμο δεδομένα που αποτελούν το test set. Η ακρίβειά του ως προς αυτές τις προβλέψεις είναι που θα καθορίσει τον βαθμό της επιτυχίας του [3].

Υπάρχουν πολλά μοντέλα που χρησιμοποιούνται. Τρία που χρησιμοποιούνται από την βιβλιοθήκη RecommenderLab είναι τα: splitting, cross-validation και bootstrap [3].

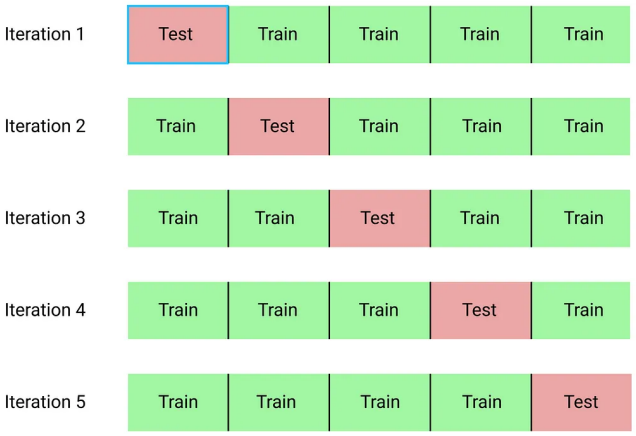
2.6.1 Splitting

Η splitting είναι ένα μοντέλο αξιολόγησης το οποίο χρησιμοποιείται συχνά καθώς είναι εύκολο να υλοποιηθεί και παρέχει έναν απλό τρόπο για να αξιολογηθεί η απόδοση ενός αλγορίθμου παραγωγής συστάσεων [3],[32].

Στον διαχωρισμό του συνόλου δεδομένων με splitting, δεδομένου ενός ποσοστού training set, ο αλγόριθμος θα επιλέξει τυχαία τόσα transactions όσα δηλώνει το ποσοστό. Αυτά θα είναι τα δεδομένα στα οποία θα εκπαιδευτεί ο αλγόριθμος, και τα υπόλοιπα θα αποτελέσουν το test set. Το training set είναι συνήθως μεγαλύτερο από το test set [3],[32].

2.6.2 Cross Validation

Η cross validation είναι ένα μοντέλο αξιολόγησης που χρησιμοποιείται στην μηχανική μάθηση. Υπάρχουν διάφορες διαφοροποιήσεις αυτής της τεχνικής, όμως αυτή που χρησιμοποιείται από την βιβλιοθήκη RecommenderLab είναι αυτή της k-fold cross validation. Η διαδικασία περιλαμβάνει την διαίρεση του συνόλου δεδομένων σε k υποσύνολα, ή folds, ίσου μεγέθους. Ο αλγόριθμος τότε εκπαιδεύεται και κάνει τεστ k φορές, με κάθε fold να έχει τον ρόλο του test set από μία φορά, και όλα τα υπόλοιπα k-1 folds να χρησιμοποιούνται ως το training set, όπως φαίνεται στην Εικόνα 2.1 ένα παράδειγμα με k=5 [25]. Τα αποτελέσματα από κάθε επανάληψη επιστρέφονται ως μέσος όρος στα αποτελέσματα για να αξιολογηθεί η ολική απόδοση [3],[32],[33].



Εικόνα 2.1 K-Fold Cross Validation με 5 folds

Η cross validation έχει πολλά πλεονεκτήματα σε σχέση με άλλα μοντέλα αξιολόγησης. Ένα από τα βασικά είναι ότι τα αποτελέσματά της είναι πιο αξιόπιστα καθώς χρησιμοποιεί όλα τα διαθέσιμα δεδομένα και για training και για test set. Αυτό είναι ιδιαίτερα χρήσιμο όταν το σύνολο δεδομένων είναι μικρό ή όταν τα δεδομένα δεν είναι ισορροπημένης κατανομής [33].

Ένα ακόμα πλεονέκτημα του μοντέλου είναι ότι παρέχει έναν τρόπο να εκτιμηθεί η σταθερότητα της απόδοσης του αλγορίθμου. Επαναλαμβάνοντας την διαδικασία k φορές με διαφορετικά test sets, μπορεί να διαπιστωθεί πόσο ποικίλουν τα αποτελέσματα του αλγορίθμου ανάλογα με τα δεδομένα στα οποία έχει εκπαιδευτεί [33].

2.6.3 Bootstrap Sampling

Η bootstrap sampling είναι ένα μοντέλο αξιολόγησης που χρησιμοποιείται συχνά στην στατιστική και στην μηχανική μάθηση. Αποτελεί μία τεχνική δειγματοληψίας με αντικατάσταση [3].

Κατά την εφαρμογή του μοντέλου σε ένα σύνολο δεδομένων, αφού οριστεί ποιο θα είναι το μέγεθος του training set, θα επιλέγονται τυχαία χρήστες από το σύνολο δεδομένων ένας-ένας και θα καταγράφονται, με κάθε καταγραφή τους όμως επιστρέφονται πίσω στο σύνολο των δεδομένων και είναι πιθανό να επιλεχθούν ξανά για το train set. Στο τέλος, όταν καταγραφεί ο επιθυμητός αριθμός χρηστών για το train set, αυτοί οι χρήστες που δεν επιλέχθηκαν ποτέ (out-of-bag observations) θα αποτελέσουν το test set [25],[34].

Αυτό το μοντέλο έχει πλεονέκτημα στα μικρού μεγέθους σύνολα δεδομένων, καθώς μπορεί να δημιουργηθεί μεγάλου μεγέθους train set και να υπάρχουν ακόμα χρήστες διαθέσιμοι για το test set [34].

Ένα από τα βασικά μειονεκτήματα της bootstrap sampling είναι ότι τα δείγματα μπορεί να μην αντιπροσωπεύουν με ακρίβεια την πραγματική διανομή των δεδομένων.

2.7 Evaluation Measures

Εφόσον έχουν χρησιμοποιηθεί αλγόριθμοι παραγωγής συστάσεων που διαχειρίζονται τα δεδομένα σε δυαδική μορφή, όπως ο αλγόριθμος των Association Rules ή οι αλγόριθμοι Collaborative Filtering με την χρήση της Jaccard Similarity, η αξιολόγηση των αποτελεσμάτων κρίνεται αν προέβλεψαν ότι ένας χρήστης θα αλληλεπιδράσει με ένα αντικείμενο ή όχι [3].

Μερικά από τα πιο δημοφιλή μέτρα επίδοσης (evaluation measures) είναι η precision, η recall, η accuracy και η specificity.

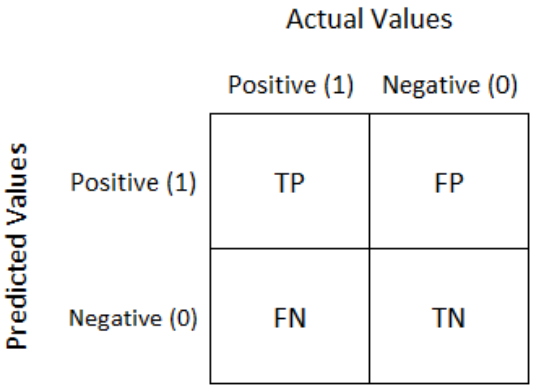
2.7.1 Confusion Matrix

Ο confusion matrix (πίνακας σύγχυσης) είναι ένας πίνακας που χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός αλγορίθμου ταξινόμησης στην μηχανική μάθηση. Είναι ένας τετραγωνικός πίνακας

που συγκρίνει τις προβλεπόμενες και τις πραγματικές τιμές στις προβλέψεις του αλγορίθμου. Αποτελεί ένα χρήσιμο εργαλείο για την κατανόηση της απόδοσης ενός αλγορίθμου και για την ταυτοποίηση των σημείων στα οποία κάνει λάθος [25],[35].

Ένας confusion matrix αποτελείται από τέσσερα βασικά στοιχεία: true positives (TP), true negatives (TN), false positives (FP), και false negatives (FN). TP είναι ο αριθμός των σωστών προβλεπόμενων θετικών περιπτώσεων, TN είναι ο αριθμός των σωστών προβλεπόμενων αρνητικών περιπτώσεων, FP είναι ο αριθμός των περιπτώσεων που προβλέφθηκαν ως θετικές αλλά στην πραγματικότητα ήταν αρνητικές, ενώ FN είναι ο αριθμός των περιπτώσεων που προβλέφθηκαν αρνητικές αλλά ήταν στην πραγματικότητα θετικές [25].

Οι γραμμές του confusion matrix αναπαριστούν τις προβλεπόμενες τιμές, ενώ οι στήλες αναπαριστούν τις πραγματικές τιμές. Σε έναν αλγορίθμου ταξινόμησης δυαδικών δεδομένων, ο confusion matrix θα αποτελείται από 2 γραμμές και 2 στήλες. Τα διαγώνια κελιά του πίνακα αναπαριστούν τις σωστά ταξινομημένες περιπτώσεις, ενώ τα υπόλοιπα κελιά αναπαριστούν τις λανθασμένα ταξινομημένες περιπτώσεις, όπως φαίνεται στην Εικόνα 2.2 [35].



Εικόνα 2.2 Confusion matrix

Ένας confusion matrix μπορεί να παρέχει διάφορα άλλα μέτρα επιδόσεων, όπως την precision, την recall, την accuracy και την specificity. Παρέχεται επίσης στα αποτελέσματα των αξιολογήσεων της βιβλιοθήκης RecommenderLab [3].

Ο confusion matrix είναι ένα απαραίτητο εργαλείο για την αξιολόγηση αλγορίθμων ταξινόμησης. Παρέχει ανάλυση για τις προβλέψεις ενός αλγορίθμου, επιτρέποντας στον χρήστη του να ταυτοποιήσει που εκτελεί λάθη. Αυτή η πληροφορία είναι κρίσιμη για την κατανόηση των ορίων του αλγορίθμου και στην βελτίωση της αποτελεσματικότητάς του [35].

2.7.2 Precision

Η precision είναι ένα δημοφιλές μέτρο επίδοσης στην μηχανική μάθηση και βοηθάει στην αξιολόγηση της ακρίβειας των προβλέψεων των αλγορίθμων. Είναι μία μετρική που υπολογίζει το ποσοστό των σωστών θετικών αποτελεσμάτων στον συνολικό αριθμό των θετικών περιπτώσεων, δηλαδή είναι το κλάσμα των true positives (TP) προς τον συνολικό αριθμό των θετικών προβλέψεων, όπως φαίνεται

στην παράσταση (2.11) [34]. Χρησιμοποιείται σε δυαδικά προβλήματα ταξινόμησης, όπου ο στόχος είναι να προβλεφθούν δύο πιθανά αποτελέσματα [3][35].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.11}$$

Η precision είναι μία σημαντική μετρική καθώς δηλώνει πόσο ακριβές ήταν οι θετικές προβλέψεις του μοντέλου. Σε πολλές εφαρμογές με προβλήματα του πραγματικού κόσμου, όπως μία ιατρική διάγνωση, τα false positives (FP), δηλαδή η πρόβλεψη ενός θετικού αποτελέσματος ενώ στην πραγματικότητα είναι αρνητικό) μπορεί να έχει σοβαρές επιπτώσεις. Επομένως, είναι σημαντικό η τιμή της precision να είναι υψηλή για να είναι όσο το δυνατόν μικρότερος ο αριθμός των false positives.

Η τιμή της precision κυμαίνεται από το διάστημα 0 μέχρι 1, όπου η τιμή 1 δηλώνει ότι όλες οι θετικά προβλεπόμενες περιπτώσεις είναι σωστές και δεν υπάρχουν false positives. Αντιθέτως, μία τιμή precision ίση με το 0 δηλώνει πως όλές οι θετικές προβλέψεις ήταν λανθασμένες, άρα δεν υπάρχουν true positives.

2.7.3 Recall

Η recall (ή αλλιώς γνωστή ως sensitivity και true positive rate) είναι ένα μέτρο επίδοσης που χρησιμοποιείται για να αξιολογήσει την αποτελεσματικότητα ενός αλγορίθμου ταξινόμησης. Η recall υπολογίζει το ποσοστό των σωστών θετικών προβλέψεων του συστήματος προς τον αριθμό όλων των πραγματικών θετικών περιπτώσεων, όπως φαίνεται στην παράσταση (2.12) [35],[36]:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.12}$$

Η recall, με άλλα λόγια, υπολογίσει την ικανότητα του συστήματος να ταυτοποιεί όλες τις πραγματικά θετικά περιπτώσεις. Αυτό είναι σημαντικό όταν αντιμετωπίζονται καταστάσεις όπου το να μην βρεθεί μία πραγματικά θετική περίπτωση θα μπορούσε να έχει σοβαρές επιπτώσεις, όπως σε σύστημα ιατρικών διαγνώσεων.

Η υψηλή τιμή recall είναι επιθυμητή καθώς διαβεβαιώνει πως το σύστημα είναι ικανό να ταυτοποιήσει όσο το δυνατόν περισσότερες θετικές περιπτώσεις. Ωστόσο, η υψηλή αυτή τιμή δεν σημαίνει πως δεν προτείνεται στον χρήστη μεγάλος αριθμός λανθασμένων αντικειμένων στον χρήστη, το οποίο θα μπορούσε να επηρεάσει αρνητικά την εμπειρία του χρήστη.

Σε τέτοιες περιπτώσεις, η precision έχει σημαντικό ρόλο καθώς υπολογίζει το ποσοστό των σωστών θετικών προβλέψεων από όλες τις θετικές προβλέψεις, δηλαδή από όλα τα αντικείμενα που προτάθηκαν.

Μαζί με την precision, προσφέρονται επίσης στα αποτελέσματα αξιολόγησης της βιβλιοθήκης Recommenderlab για τα σύνολα δεδομένων δυαδικού περιεχομένου [3],[32].

2.7.4 Accuracy

Η accuracy είναι ένα μέτρο επίδοσης που χρησιμοποιείται συχνά στην μηχανική μάθηση και στα συστήματα παραγωγής συστάσεων για να αξιολογήσει την αποτελεσματικότητα ενός αλγορίθμου ταξινόμησης. Η accuracy υπολογίζει το ποσοστό των σωστών προβλέψεων σε όλο το σύνολο των προβλέψεων που έγιναν από το σύστημα. Στα συστήματα παραγωγής συστάσεων, η accuracy χρησιμοποιείται για να υπολογίσει πόσο καλά το σύστημα μπορεί να προβλέψει τις προτιμήσεις των χρηστών και να προτείνει αντικείμενα σχετικά με τα ενδιαφέροντά τους [35],[37].

Η μετρική accuracy υπολογίζεται με τον αριθμό των σωστών προβλέψεων προς τον αριθμό όλων των προβλέψεων, όπως φαίνεται στην παράσταση (2.13). Είναι ένας απλός τρόπος για να αξιολογηθεί η απόδοση ενός συστήματος ταξινόμησης[37].

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.13}$$

Η σημασία της accuracy στα συστήματα παραγωγής συστάσεων βρίσκεται στην ικανότητά της να εκτιμήσει πόσο καλά τα σύστημα έχει καταφέρει να προβλέψει τις προτιμήσεις των χρηστών. Μία υψηλή τιμή accuracy μπορεί να δηλώνει πως το σύστημα μπορεί να προβλέψει με ακρίβεια τα αντικείμενα για τα οποίο ένας χρήστης είναι πιθανόν να ενδιαφέρεται και να του τα προτείνει. Αυτό μπορεί να οδηγήσει στην ικανοποίηση των χρηστών και στην προτίμησή του προς το σύστημα, καθώς θα είναι πιο πιθανό να βρει αντικείμενα που τον ενδιαφέρουν και θα συνεχίσει να το χρησιμοποιεί.

Ωστόσο, δεν είναι πάντα η accuracy το πιο κατάλληλο μέτρο επίδοσης για την αξιολόγηση των συστημάτων παραγωγής συστάσεων. Ένας περιορισμός της είναι ότι η υψηλή τιμή της accuracy δεν σημαίνει απαραίτητα πως έχουν προταθεί στον χρήστη αντικείμενα σχετικά με τα ενδιαφέροντά του, αλλά η τιμή της να οφείλεται σε όλες τις προτάσεις εκείνες που δεν ενδιαφέρουν τον χρήστη και όντως δεν του προτάθηκαν, δηλαδή τις true negative περιπτώσεις [37].

Για αυτό, είναι σημαντικό να συνδυάζεται μαζί με άλλα μέτρα επιδόσεων, όπως η precision και η recall. Αυτές οι μετρικές μπορούν να παρέχουν μία πιο αντικειμενική αξιολόγηση της απόδοσης του συστήματος και να ταυτοποιήσουν τα σημεία που χρειάζεται βελτίωση.

2.7.5 Specificity

Η specificity είναι ένα ακόμα μέτρο επίδοσης που χρησιμοποιείται συχνά σε δυαδικά προβλήματα ταξινόμησης, όπως σε συστήματα παραγωγής συστάσεων. Είναι μία μετρική που υπολογίζει το ποσοστό των σωστών πραγματικά αρνητικών περιπτώσεων σε όλες τις πραγματικά αρνητικές περιπτώσεις του συνόλου δεδομένων, όπως φαίνεται στην παράσταση (2.14). Στα συστήματα παραγωγής συστάσεων, η specificity χρησιμοποιείται για να αξιολογήσει πόσο καλά ένα συστήματα

έχει καταφέρει να ταυτοποιήσει τα αντικείμενα για τα οποία ο χρήστης δεν θα παρουσιάσει ενδιαφέρον [35].

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{2.14}$$

Ένα από τα βασικά πλεονεκτήματα της specificity είναι ότι μπορεί να αντιμετωπίσει το πρόβλημα των συστάσεων αντικειμένων που δεν ενδιαφέρουν τους χρήστες. Αξιολογόντας την ικανότητα του συστήματος να ταυτοποιεί σωστά τα αντικείμενα για τα οποία ένας χρήστης δεν θα ενδιαφερθεί, η μετρική αυτή μπορεί να βοηθήσει να μειωθεί ο αριθμός των αντικειμένων αυτών ως συστάσεις στον συγκεκριμένο χρήστη. Αυτό μπορεί να βελτιώσει την εμπειρία του χρήστη και να αυξήσει τις πιθανότητες να συνεχίσει να χρησιμοποιεί το σύστημα.

Ωστόσο, υπάρχουν επίσης κάποιοι περιορισμοί στην χρήση της specificity. Ένας από τους βασικούς περιορισμούς είναι ότι λαμβάνει υπόψη μόνο τις αρνητικές περιπτώσεις, όπως προκύπτει από τον τύπο της, και όχι τις θετικές. Αυτό σημαίνει ότι ένα σύστημα με υψηλή τιμή specificity μπορεί ακόμα να προτείνει αντικείμενα που δεν ενδιαφέρουν τους χρήστες, εφόσον είναι ικανό να ταυτοποιήσει έναν μεγάλο αριθμό από τα αντικείμενα που δεν τους ενδιαφέρουν.

Η specificity και η accuracy δεν προσφέρονται ως αποτελέσματα αξιολόγησης από την βιβλιοθήκη RecommenderLab, αλλά μπορούν να υπολογιστούν με την χρήση των αποτελεσμάτων του confusion matrix. Παρόλα αυτά, δεν θα γίνει η χρήση τους στην συνέχεια της πτυχιακής εργασίας, καθώς περιέχουν την τιμή των TN, τα οποία από την φύση του αλγορίθμου που συστήνει top-N-Lists θα είναι ένας πολύ μεγάλος αριθμός, καθώς περιλαμβάνει όλα τα αντικείμενα που δεν συστάθηκαν και ο χρήστης είναι σπάνιο να έχει καταναλώσει/προτιμήσει την πλειοψηφία από αυτα [32].

2.8 Επίλογος

Μερικές από τις πιο γνωστές τεχνικές παραγωγής συστάσεων αποτελούν οι Collaborative Filtering, Content-Based Filtering, και η Session-Based Filtering που κάνει χρήση των Association Rules. Μέτρα ομοιότητας όπως Cosine Similarity, Jaccard Similarity και συντελεστές συσχέτισης όπως Pearson Correlation Coefficient, Spearman Correlation Coefficient χρησιμοποιούνται για τον υπολογισμό της ομοιότητας και της συσχέτισης μεταξύ δύο μεταβλητών και μπορούν να χρησιμοποιηθούν από τις τεχνικές παραγωγής συστάσεων. Η ARM, στην οποία επικεντρώνεται η παρούσα πτυχιακή εργασία, κάνει χρήση διαφόρων μέτρων αξίας για την ταυτοποίηση της σημασίας ενός κανόνα, όπως είναι τα support, confidence, lift και conviction, με τα δύο τελευταία από αυτά να δηλώνουν αιτιότητα σε έναν κανόνα, με τον lift όμως να περιέχει θόρυβο λόγω της αντιμεταθετικής του ιδιότητας. Η ARM μπορεί να χρησιμοποιηθεί στην παραγωγή συστάσεων, και κατά το στάδιο της αξιολόγησης τα δεδομένα χωρίζονται σε training set και test set με την χρήση ενός μοντέλου αξιολόγησης, όπως είναι η splitting, η cross-validation και η bootstrap sampling, με την cross-validation να αποτελεί το μόνο μοντέλο που χρησιμοποιεί σίγουρα όλα τα δεδομένα από μία φορά ως test set. Τέλος, η αξιολόγηση των αποτελεσμάτων των συστάσεων γίνεται με την χρήση μέτρων επιδόσεων, όπως ο confusion matrix, η precision, η recall, η accuracy και η specificity, με τις

δύο τελευταίες να κάνουν χρήση των ΤΝ που στην περίπτωση των συστημάτων παραγωγής συστάσεων συνήθως είναι πολύ μεγάλοι αριθμοί.

Κεφάλαιο 3ο : Λογισμικό

3.1 Εισαγωγή

Σε αυτό το Κεφάλαιο γίνεται αναφορά στην γλώσσα προγραμματισμού R και στο προγραμματιστικό περιβάλλον RStudio που χρησιμοποιούνται στην παρούσα πτυχιακή εργασία, καθώς και οι απαραίτητες βιβλιοθήκες που συμβάλλουν στην παραγωγή των συστάσεων, στην αξιολόγησή τους και στην παρουσίαση των απαραίτητων γραφημάτων. Παρουσιάζονται επίσης αναλυτικά κάποιες από τις βασικότερες κλάσεις, συναρτήσεις και μεθόδους της βιβλιοθήκης RecommenderLab, ο τρόπος λειτουργίας και χρήσης της για την παραγωγή συστάσεων καθώς και κάποια κομμάτια του κώδικά της και η ανάλυσή τους. Τέλος, παρουσιάζονται επίσης κάποια αποτελέσματα των συστάσεων και των αξιολογήσεων της. Όλα τα παραπάνω εφαρμόζονται στο σύνολο δεδομένων MovieLens 100K που παρέχεται από την ίδια την βιβλιοθήκη.

3.2 R και RStudio

Η R είναι μία γλώσσα προγραμματισμού ανοιχτού κώδικα που χρησιμοποιείται για στατιστικούς υπολογισμούς και γραφήματα. Δημιουργήθηκε από τους στατιστικούς Ross Ihaka και Robert Gentleman και χρησιμοποιείται από επιστήμονες της εξόρυξης δεδομένων, από επιστήμονες της βιοπληροφορικής και από στατιστικούς για την στατιστικού και αναλυτικού τύπου επεξεργασία των δεδομένων. Είναι παρόμοια με την γλώσσα προγραμματισμού S στην οποία βασίστηκε για την δημιουργία της, που αναπτύχθηκε στα εργαστήρια Bell (παλαιότερα γνωστά ως AT&T, τώρα ως Lucent Technologies) από τον John Chambers και τους συναδέλφους του [38].

Η R παρέχει μεγάλη ποικιλία στατιστικών και γραφικών τεχνικών, όπως γραμμική και μη γραμμική μοντελοποίηση, κλασικές στατιστικές δοκιμές, ομαδοποίηση, ταξινόμηση κλπ. Προσφέρει δομές δεδομένων που χρησιμεύουν στην αποτελεσματική οργάνωση των δεδομένων όπως [39]:

- vectors
- arrays
- matrices
- lists
- data frames
- factors

Επίσης, η R μπορεί να επεκταθεί σε μεγάλο βαθμό από τους χρήστες της, δημιουργώντας έτσι πλήθος πακέτων προσφέροντας έτσι επιπλέον στατιστικές τεχνικές, εισαγωγή, εξαγωγή δεδομένων κλπ. Χάρη στην εύκολη εγκατάστασή τους, αυτά τα πακέτα έχουν υιοθετηθεί ευρέως από τις επιστήμες των δεδομένων [38].

Για τις ανάγκες της παρούσας πτυχιακής εργασίας χρησιμοποιήθηκαν τα εξής πακέτα:

- recommenderlab: Πακέτο για την δημιουργία εκπαιδευμένων μοντέλων, την παραγωγή συστάσεων με διάφορες τεχνικές και την αξιολόγηση αυτών των τεχνικών.

- ggplot2: Πακέτο για την δημιουργία γραφημάτων, για την καλύτερη παρουσίαση των αποτελεσμάτων της αξιολόγησης.

Η R διαθέτει κάποια πλεονεκτήματα όπως [40-42]:

- Είναι ανοιχτού κώδικα, που σημαίνει ότι δεν χρειάζεται κάποια άδεια ή χρέωση για να χρησιμοποιηθεί και μπορούν να προσφέρουν προγραμματιστικά σε αυτήν οι χρήστες της, βοηθώντας έτσι στην επέκτασή της.
- Τρέχει σε όλα τα λειτουργικά συστήματα, προσφέροντας την άνεση στους προγραμματιστές να είναι αρκετό να αναπτύξουν μόνο μία φορά τον κώδικα τους.
- Προσφέρει λειτουργίες για μηχανική μάθηση μέσα από διάφορα πακέτα για την ανάπτυξη τεχνητών νευρωνικών δικτύων.
- Παράγει ποιοτικά γραφήματα με απλοποιημένη διαδικασία. Βιβλιοθήκες όπως το ggplot2 προσφέρουν ελκυστικά γραφήματα που κάνουν την R να ξεχωρίζει.
- Διαθέτει πλήθος πακέτων, με πάνω από 19.000 από αυτά να προσφέρονται στο CRAN repository και ο αριθμός τους συνεχώς αυξάνεται.

Το RStudio είναι ένα ολοκληρωμένο περιβάλλον προγραμματισμού με την R. Είναι διαθέσιμο σε δύο μορφές, το RStudio Desktop που είναι η εφαρμογή για την επιφάνεια εργασίας, ενώ το R Studio Server τρέχει σε έναν απομακρυσμένο διακομιστή επιτρέποντας την πρόσβαση στο RStudio μέσω προγράμματος περιήγησης ιστού [42].

3.3 RecommenderLab

Το πακέτο RecommenderLab της γλώσσας προγραμματισμού R παρέχει ένα σύνολο αλγορίθμων για την δημιουργία και την αξιολόγηση συστημάτων παραγωγής συστάσεων [3].

Είναι χτισμένο με βάση το αντικειμενοστραφές σύστημα S4 στην R, το οποίο επιτρέπει την υλοποίηση αποτελεσματικών και ευέλικτων αλγορίθμων. Πιο συγκεκριμένα, το σύστημα S4 είναι ένα σύστημα όπου για να δημιουργηθεί ένα αντικείμενο χρησιμοποιείται μία constructor συνάρτηση η οποία είναι ορισμένη σε μία κλάση. Η κλάση αυτή ορίζει τις ιδιότητες αυτού του αντικειμένου, τις μεθόδους με τις οποίες θα γίνεται η διαχείρισή του και άλλα χαρακτηριστικά [43].

Το RecommenderLab αναπτύχθηκε από τον Michael Hahsler, ο οποίος είναι επιστήμονας των υπολογιστών με συνεισφορά στον τομέα της εξόρυξης δεδομένων, στην τεχνητή νοημοσύνη, στην μηχανική μάθηση και στην ανάπτυξη open-source πακέτων για την υποστήριξη ερευνών. Έχει εκδώσει περισσότερα από 90 τεχνικά papers, έχει οργανώσει αρκετά workshops και μαζί με την ομάδα του υποστηρίζουν 15 ευρέως διαδομένα πακέτα της R, μεταξύ των οποίων βρίσκονται και τα “arules” και “recommenderlab” [44].

Το πακέτο προσφέρει πλήθος αλγορίθμων για την δημιουργία μοντέλων παραγωγής συστάσεων και συμπεριλαμβάνει διάφορες τεχνικές. Επικεντρώνεται στην τεχνική Collaborative Filtering, αλλά παρέχονται και επιπλέον τεχνικές όπως η χρήση των Association Rules στους οποίους θα επικεντρωθεί η παρούσα πτυχιακή εργασία [3],[16].

Πιο αναλυτικά, το πακέτο υποστηρίζει τους εξής αλγορίθμους παραγωγής συστάσεων [32][45]:

- User-Based collaborative filtering (UBCF)
- Item-Based collaborative filtering (IBCF)
- SVD with column-mean imputation (SVD)
- Funk SVD (SVDF)
- Alternating Least Squares (ALS)
- Matrix factorization with LIMBF (LIMBF)
- Association rule-based recommender (AR)
- Popular Items (POPULAR)
- Randomly chosen items for comparison (RANDOM)
- Re-recommend liked items (RERECOMMEND)
- Hybrid recommendations (HybridRecommender)

Ένα από τα σημαντικότερα χαρακτηριστικά του RecommenderLab είναι η ευελιξία του, καθώς δίνει την δυνατότητα στους χρήστες να επεξεργαστούν τους αλγορίθμους και τις παραμέτρους που χρησιμοποιούνται στα μοντέλα παραγωγής συστάσεων. Έτσι, οι χρήστες μπορούν να προσαρμόσουν τα μοντέλα στις δικές τους ανάγκες και να πειραματιστούν με νέες προσεγγίσεις [3].

Ένα επιπλέον σημαντικό χαρακτηριστικό του είναι η ικανότητά του να διαχειρίζεται μεγάλα σύνολα δεδομένων. Το πακέτο παρέχει αλγορίθμους για την διαχείριση των sparse matrices, δηλαδή έναν τύπο πίνακα κατάλληλο για την αποθήκευση μεγάλων συνόλων δεδομένων που πολλά από τα δεδομένα τους θα αποτελούν μηδενικές τιμές και δεν υπάρχει περιορισμός στο πλήθος αυτών των μηδενικών. Αυτό το χαρακτηριστικό επιτρέπει στο RecommenderLab να διαχειρίζεται τεράστια σύνολα δεδομένων με πλήθος χρηστών και αντικειμένων [3],[32],[46].

Το RecommenderLab παρέχει επίσης εργαλεία για την αξιολόγηση των μοντέλων παραγωγής συστάσεων. Το πακέτο περιλαμβάνει για τα αριθμητικά δεδομένα τις μετρικές MSE (Mean Squared Error), RMSE (Root Mean Squared Error) και MAE (Mean Absolute Error), ενώ για τα δυαδικά δεδομένα υπολογίζονται μετρικές αξιολόγησης όπως οι precision, recall, TPR/FPR (True Positive Rating/ False Positive Rating). Αυτές οι μετρικές παρέχουν ένα μέτρο απόδοσης για για τα μοντέλα παραγωγής συστάσεων [32],[45].

Το πακέτο προσφέρει επίσης συναρτήσεις για την απεικόνιση των δεδομένων και των αποτελεσμάτων της αξιολόγησης των συστάσεων καθώς και συναρτήσεις για την μετατροπή των δεδομένων στην απαραίτητη μορφή που απαιτείται από τους αλγορίθμους [3].

Προσφέρεται επίσης η δυνατότητα διαχωρισμού των δεδομένων σε training set και σε test set με τα παρακάτω μοντέλα αξιολόγησης [3],[32],[45]:

- Train/test split
- Cross-Validation
- Repeated bootstrap sampling

Έναν περιορισμό αποτελεί το γεγονός πως το πακέτο δεν περιλαμβάνει μερικές από τις πιο προηγμένες τεχνικές που έχουν αναπτυχθεί τα τελευταία χρόνια, όπως Deep Learning-Based μοντέλα παραγωγής συστάσεων.

3.4 Συναρτήσεις και χρήση της βιβλιοθήκης

Για να μπορούν να χρησιμοποιηθούν οι συναρτήσεις της βιβλιοθήκης RecommenderLab, αρκεί να εγκατασταθεί και να φορτωθεί η βιβλιοθήκη στο RStudio με τις παρακάτω εντολές.

```
> install.packages('recommenderlab')
> library(recommenderlab)
```

Η βιβλιοθήκη περιέχει ένα σύνολο δεδομένων MovieLense, το οποίο προέρχεται από τα δεδομένα της ιστοσελίδας MovieLens κατά την διάρκεια μιας έρευνας που έγινε από τις 19 Σεπτεμβρίου 1997 μέχρι τις 22 Απριλίου του 1998. Το σύνολο δεδομένων περιέχει περίπου 100 χιλιάδες βαθμολογίες ταινιών με εύρος τιμών από 1 μέχρι 5 από 943 χρήστες για 1664 ταινίες [32].

```
> data("MovieLense")
> MovieLense
943 x 1664 rating matrix of class 'realRatingMatrix' with 99392 ratings.
```

Αφού φορτωθεί το MovieLense και κληθεί, εμφανίζονται πληροφορίες σχετικά με την κλάση του και τις διαστάσεις του. Η κλάση ratingMatrix είναι μία κοινή κλάση για βαθμολογικά δεδομένα που στην βιβλιοθήκη υλοποιείται με δύο μορφές, realRatingMatrix που περιέχει πραγματικές αριθμητικές βαθμολογίες και binaryRatingMatrix που περιέχει τις βαθμολογίες σε δυαδική μορφή [32].

3.4.1 Ο πίνακας realRatingMatrix

Στο παραπάνω αποτέλεσμα φαίνεται ότι η κλάση του συνόλου δεδομένων MovieLense είναι η realRatingMatrix, που είναι μία υλοποίηση της κλάσης ratingMatrix [32]. Είναι ένας πίνακας με 943 γραμμές που υποδηλώνουν τους χρήστες (users) των δεδομένων και 1664 ταινίες (items), και η μεταξύ τους σχέσεις είναι οι βαθμολογίες που έδωσαν οι χρήστες στις ταινίες.

Με την μέθοδο getRatingMatrix() που υλοποιείται στην κλάση ratingMatrix θα προβληθούν τα περιεχόμενα του realRatingMatrix. Στην περίπτωση του παραδείγματος, ζητούνται οι πρώτες 5 γραμμές και οι πρώτες 3 στήλες, με αποτέλεσμα να επιστρέφεται ένας πίνακας με τους πρώτους 5 χρήστες και τις πρώτες 3 ταινίες, τύπου sparseMatrix της κλάσης dgCMatrix [32].

```
> getRatingMatrix(MovieLense[1:5, 1:3])
5 x 3 sparse Matrix of class "dgCMatrix"
  Toy Story (1995) GoldenEye (1995) Four Rooms (1995)
1                5                3                 4
2                4                .                 .
3                .                .                 .
```

```
4                .                .                .
5                4                3                .
```

Στα αριστερά φαίνονται οι αριθμοί που αντιστοιχούν στην κωδικό του κάθε χρήστη, ενώ όλες οι υπόλοιπες στήλες αντιστοιχούν η καθεμία σε μία ταινία του συνόλου δεδομένων. Αν ένας χρήστης έχει βαθμολογήσει μία ταινία, αυτή η βαθμολογία θα εμφανιστεί στην γραμμή αυτού του χρήστη και στην στήλη της ανάλογης ταινίας. Αν ο χρήστης δεν έχει βαθμολογήσει μία ταινία, τότε εκείνη η θέση έχει μία τελεία, υποδηλώνοντας ότι δεν είναι διαθέσιμη η βαθμολογία.

3.4.2 Ο πίνακας binaryRatingMatrix

Σε κάποιες περιπτώσεις στην διαδικασία παραγωγής συστάσεων, μπορεί να χρησιμοποιηθούν προσεγγίσεις και τεχνικές οι οποίες διαχειρίζονται με διαφορετικό τρόπο τα δεδομένα. Στην μορφή του realRatingMatrix πίνακα, οι βαθμολογίες παρουσιάζονται αριθμητικές, όπως ακριβώς δόθηκαν οι βαθμολογίες από τους χρήστες. Υπάρχει όμως περίπτωση για να γίνει η επεξεργασία, τα δεδομένα να πρέπει να είναι σε δυαδική μορφή, καθώς η πληροφορία μπορεί για παράδειγμα να δείχνουν αν απλα ο χρήστης παρουσιάζει ενδιαφέρον για ένα συγκεκριμένο αντικείμενο. Αυτό επιτυγχάνεται με την χρήση της συνάρτησης binarize() [3], η οποία λαμβάνοντας ως παράμετρο έναν realRatingMatrix τον μετατρέπει σε μία άλλη μορφή της βιβλιοθήκης, τον binaryRatingMatrix που επίσης υλοποιεί την κλάση ratingMatrix, και αναπαριστά έναν πίνακα βαθμολογιών που πλέον είναι σε δυαδική μορφή [32].

```
> binarized <- binarize(MovieLense, minRating=4)
> binarized
943 x 1664 rating matrix of class 'binaryRatingMatrix' with 55024 ratings.
```

Η συνάρτηση binarize() χρειάζεται άλλη μία παράμετρο για να κάνει την μετατροπή του πίνακα, και αυτή είναι η **minRating**. Δίνοντας μία τιμή στη minRating που να είναι εντός των ορίων των τιμών των βαθμολογιών, όλες οι τιμές από αυτό το νούμερο και πάνω θα μετατραπούν σε true, ενώ όλες οι τιμές κάτω από αυτό το νούμερο, όπως και όλες οι περιπτώσεις απουσίας τιμής (.) θα μετατραπούν σε false [3].

Παρατηρείται στο παραπάνω αποτέλεσμα ότι ενώ στον realRatingMatrix ο αριθμός των βαθμολογιών ήταν 99392, στον binaryRatingMatrix αυτός ο αριθμός έχει πέσει στο 55024, καθώς περιλαμβάνονται μόνο οι βαθμολογίες που έχουν μετατραπεί σε true.

```
> as(binarized[1:5,1:3],"matrix")
  Toy Story (1995) GoldenEye (1995) Four Rooms (1995)
1             TRUE            FALSE             TRUE
2             TRUE            FALSE            FALSE
3            FALSE            FALSE            FALSE
```

```
4              FALSE             FALSE             FALSE
5               TRUE             FALSE             FALSE
```

Έτσι, παρουσιάζοντας τα περιεχόμενα του binaryRatingMatrix σε πίνακα για τις πρώτες 5 γραμμές και τις πρώτες 3 στήλες, τα ονόματα των γραμμών και των στηλών έχουν παραμείνει ίδια με του realRatingMatrix πίνακα, αναπαριστώντας τους χρήστες και τις ταινίες αντίστοιχα, αλλά η μεταξύ τους σχέση πλέον αντί για αριθμητικές τιμές και τελείες είναι τιμές true και false.

Είναι σημαντικό να παρατηρηθεί πως έτσι χάνεται η πληροφορία σχετικά με ποιες ταινίες έχει βαθμολογήσει χαμηλά ο χρήστης και ποιες δεν έχει παρακολουθήσει ή δεν έχει βαθμολογήσει.

3.4.3 Η συνάρτηση Recommender()

Η συνάρτηση Recommender() καλείται καθώς της δίνεται παραμετρικά ένας ratingMatrix, μία μέθοδος για την εκπαίδευση του αλγόριθμου, κάποιες προαιρετικές παράμετροι ανάλογα με την μέθοδο που χρησιμοποιείται και επιστρέφει ένα αντικείμενο Recommender. Αυτό το αντικείμενο είναι ένα εκπαιδευμένο μοντέλο και θα αξιοποιηθεί στην συνέχεια για να υπολογίσει τις topNLists, δηλαδή τις λίστες των συστάσεων μήκους N, για τους χρήστες στους οποίους πρέπει να γίνουν οι συστάσεις [3],[32].

Είναι σημαντικό να σημειωθεί πως ανάλογα με την μέθοδο που δίνεται ως παράμετρος, εκτελείται και μία διαφορετική υλοποίηση της συνάρτησης Recommender(). Για παράδειγμα, αν η μέθοδος που δοθεί είναι η "AR", τότε εκτελείται η υλοποίηση της Recommender() για την μέθοδο των Association Rules από το αρχείο RECOM_AR.R, ενώ αν δοθεί η "IBCF" τότε θα εκτελεστεί η υλοποίηση της συνάρτησης στο αρχείο RECOM_IBCF.R κλπ [31],[47].

Στην περίπτωση των Association Rules, τα δεδομένα πρέπει να έχουν την μορφή transactions κατηγορικού τύπου, προκειμένου να παραχθούν κανόνες συσχέτισης. Επομένως, τα δεδομένα στην είσοδο της επεξεργασίας οργανώνονται σε δομή τύπου binaryRatingMatrix, και όχι realRatingMatrix. Ο αλγόριθμος apriori θα επεξεργαστεί το δυαδικό πίνακα ως μία συλλογή από transactions κατηγορικού τύπου για τους χρήστες, δηλαδή για κάθε χρήστη θα θεωρήσει ότι οι 'αγορές' του είναι τα αντικείμενα τα οποία έχει βαθμολογίσει ως true [31].

Ορίζεται στο παρακάτω παράδειγμα η μεταβλητή trainset, η οποία αποτελεί το μέρος του συνόλου των δεδομένων και περιέχει τις πρώτες 500 συναλλαγές. Στην συνέχεια, θα τροφοδοτηθεί στην είσοδο της συνάρτησης Recommender() ως το training set, μαζί με την μέθοδο που θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου η οποία θα είναι η Association Rules ("AR") και επομένως χρειάζεται παραμέτρους όπως support, confidence, max length και sort_measure ως παράμετροι για τους κανόνες που πρόκειται να παραχθούν. Το αντικείμενο Recommender που επιστρέφεται αποθηκεύεται στην μεταβλητή recommenderAR για να χρησιμοποιηθεί στην συνέχεια για την παραγωγή των συστάσεων.

```
> trainset<- binarized[1:500]
> trainset
500 x 1664 rating matrix of class 'binaryRatingMatrix' with 31600 ratings.
```

```
> recommenderAR<- Recommender(trainset, "AR", param=list(supp=0.2, conf =0.3,
maxlen=3, sort_measure="confidence"))
> recommenderAR
Recommender of type 'AR' for 'binaryRatingMatrix'
learned using 500 users.
```

Με αυτόν τον τρόπο, η δομή recommenderAR με το περιεχόμενό της αποτελεί ένα μοντέλο το οποίο έχει εκπαιδευτεί με βάση τις παραμέτρους που δέχτηκε. Δίνεται η δυνατότητα να προβληθούν τα περιεχόμενά του με την μέθοδο getModel() που υλοποιείται στην κλάση Recommender() [3],[32].

```
> getModel(recommenderAR)
$description
[1] "AR: rule base"
$rule_base
set of 195 rules
$support
[1] 0.2
$confidence
[1] 0.3
$maxlen
[1] 3
$maxtime
[1] 5
$sort_measure
[1] "confidence"
$sort_decreasing
[1] TRUE
$apriori_control
$apriori_control$verbose
[1] FALSE
$verbose
[1] FALSE
```

Η δομή recommenderAR αποτελείται από τα εξής πεδία: description, rule_base, support, confidence, maxlen, maxtime, sort_measure, sort_decreasing, apriori_control και verbose. Από το support μέχρι και το verbose είναι οι παράμετροι που μπορούν να δοθούν όταν καλείται η συνάρτηση Recommender, αλλιώς θα μπουν αυτόματα οι προεπιλεγμένες τιμές που έχουν οριστεί από την

βιβλιοθήκη RecommenderLab. Στο πεδίο rule_base αποθηκεύονται όλοι οι κανόνες που παράγονται από την κλήση της apriori αφού έχει γίνει η κατάταξή τους με βάση το sort_measure, δηλαδή το μέτρο αξίας με το οποίο θα γίνει η κατάταξη των κανόνων σε φθίνουσα τάξη [31].

Οι κανόνες που παρήγαγε ο αλγόριθμος είναι 195 στο συγκεκριμένο παράδειγμα και γίνεται να προβληθούν τα περιεχόμενα του rule_base ζητώντας τα από το αντικείμενο Recommender ως dataframe με την συνάρτηση as() [3]. Όπως φαίνεται και στο παρακάτω αποτέλεσμα, τα περιεχόμενα του rule_base αποτελούνται από τον αριθμό του κανόνα και στήλες με τον κανόνα και τις τιμές του για support, confidence, coverage, lift και count. Αυτοί οι κανόνες θα χρησιμοποιηθούν στην συνέχεια για την παραγωγή των συστάσεων σε νέους χρήστες.

```
> as(getModel(recommenderAR)$rule_base, "data.frame")
```

	rules	support	confidence	coverage	lift	count
157	{Empire Strikes Back, The (1980),Return of the Jedi (1983)} => {Star Wars (1977)}	0.252	0.9692308	0.260	1.815039	126
145	{Toy Story (1995),Return of the Jedi (1983)} => {Star Wars (1977)}	0.200	0.9615385	0.208	1.800634	100
175	{Silence of the Lambs, The (1991),Return of the Jedi (1983)} => {Star Wars (1977)}	0.228	0.9579832	0.238	1.793976	114
163	{Empire Strikes Back, The (1980),Raiders of the Lost Ark (1981)} => {Star Wars (1977)}	0.252	0.9545455	0.264	1.787538	126
172	{Raiders of the Lost Ark (1981),Return of the Jedi (1983)} => {Star Wars (1977)}	0.252	0.9545455	0.264	1.787538	126
160	{Pulp Fiction (1994),Empire Strikes Back, The (1980)} => {Star Wars (1977)}	0.200	0.9523810	0.210	1.783485	100
151	{Godfather, The (1972),Return of the Jedi (1983)} => {Star Wars (1977)}	0.200	0.9433962	0.212	1.766660	100
169	{Fargo (1996),Return of the Jedi (1983)} => {Star Wars (1977)}	0.224	0.9411765	0.238	1.762503	112
166	{Silence of the Lambs, The (1991),Empire Strikes Back, The (1980)} => {Star Wars (1977)}	0.216	0.9310345	0.232	1.743510	108
113	{Empire Strikes Back, The (1980)} => {Star Wars (1977)}	0.296	0.9308176	0.318	1.743104	148
123	{Return of the Jedi (1983)} => {Star Wars (1977)}	0.376	0.9170732	0.410	1.717365	188
3	{Terminator, The (1984)} => {Raiders of the Lost Ark (1981)}	0.212	0.9137931	0.232	2.392129	106

3.4.4 Η μέθοδος predict()

Η predict() είναι η μέθοδος που δηλώνεται στην κλάση Recommender() και υλοποιείται στην συνάρτηση Recommender() [3],[32] και θα δημιουργήσει τις συστάσεις χρησιμοποιώντας ένα αντικείμενο Recommender και τα δεδομένα από τους νέους χρήστες. Στο παρακάτω παράδειγμα δέχεται παραμετρικά το μοντέλο που δημιουργήθηκε προηγουμένως το οποίο εκπαιδεύτηκε και περιέχει τους κανόνες της μεθόδου Association Rules στο πεδίο rule_base, δίνονται ως νέοι χρήστες από τον δυαδικό πίνακα οι συναλλαγές από 501 έως και 503 και ορίζεται οι λίστες με τις προτάσεις να περιέχουν το πολύ 5 ταινίες στην παράμετρο n. Επομένως θα παραχθούν top5Lists για 3 χρήστες.

```
> predictionsAR <- predict(recommenderAR, binarized[501:503], n=5)
> predictionsAR
Recommendations as 'topNList' with n = 5 for 3 users.
```

Τα αποτελέσματα της predict() έχουν αποθηκευτεί σε μία μεταβλητή με όνομα predictionsAR, και ζητώντας να προβληθούν τα αποτελέσματά της ως λίστα παρουσιάζονται για κάθε έναν από τους 3 χρήστες οι ταινίες που προτείνει ο αλγόριθμος.

```
> as(predictionsAR, "list")
```

```
$`0`
[1] "Star Wars (1977)"                  "Pulp Fiction (1994)"
[3] "Raiders of the Lost Ark (1981)"    "Empire Strikes Back, The (1980)"
[5] "Silence of the Lambs, The (1991)"

$`1`
character(0)

$`2`
[1] "Pulp Fiction (1994)"               "Shawshank Redemption, The (1994)"
[3] "Terminator, The (1984)"            "Usual Suspects, The (1995)"
[5] "Back to the Future (1985)"
```

Στην περίπτωση των Association Rules, αυτό το αποτέλεσμα επιτυγχάνεται με τον εξής τρόπο που παρουσιάζεται στην Εικόνα 3.1 που αποτελεί κομμάτι του κώδικα του αρχείου RECOM_AR.R (που περιέχει την υλοποίηση της συνάρτησης Recommender() για τους Association Rules) [31]: Στην παράμετρο newdata@data της εντολής is.subset(lhs(model$rule_base), newdata@data), τα newdata αποτελούνται από τους χρήστες που πέρασαν παραμετρικά στην μέθοδο predict() ως το test set για να παραχθούν συστάσεις. Ο πίνακας αυτός έχει ως γραμμές τους χρήστες και ως στήλες όλες τις ταινίες, ενώ τα μεταξύ τους κελιά είναι συμπληρωμένα με τιμές true-false ανάλογα με το αν ο κάθε χρήστης έχει βαθμολογήσει την συγκεκριμένη ταινία.

Για παράδειγμα, γίνεται κλήση της παραμέτρου ως εντολή να εμφανιστούν από τους 3 χρήστες οι πρώτες 4 στήλες.

```
> as(binarized[501:503]@data[,1:4],"matrix")
    Toy Story (1995) GoldenEye (1995) Four Rooms (1995) Get Shorty (1995)
501            FALSE            FALSE             FALSE             FALSE
502            FALSE            FALSE             FALSE             FALSE
503             TRUE            FALSE             FALSE             FALSE
```

```
m <- is.subset(lhs(model$rule_base), newdata@data)
reclist <- list()
ratings <- list()
for(i in 1:nrow(newdata)) {
  ars <- sort(model$rule_base[m[,i]], by = sort_measure)
  ar_rhs <- unlist(LIST(rhs(ars), decode=FALSE))
  ar_qualities <- quality(ars)[[sort_measure]]
  ar_duplicates <- duplicated(ar_rhs)
  recom_item <- ar_rhs[!ar_duplicates]
  recom_qual <- ar_qualities[!ar_duplicates]

  reclist[[i]] <- if(!is.null(recom_item)) recom_item else integer(0)
  ratings[[i]] <- if(!is.null(recom_qual)) recom_qual else integer(0)
}
```

Εικόνα 3.1 RECOM_AR: Παραγωγή συστάσεων μέσα στην μέθοδο predict()

Στην παράμετρο lhs(model$rule_base), το model είναι αυτό που παρουσιάστηκε και στην Ενότητα 3.2.3, το οποίο μεταξύ άλλων πεδίων περιέχει και το rule_base που περιέχει όλους τους κανόνες που παρήγαγε η κλήση της συνάρτησης Recommender(). Για παράδειγμα, γίνεται κλήση της παραμέτρου αυτής ως εντολή με το μοντέλο του αντικειμένου recommenderAR που έχει δημιουργηθεί στην Ενότητα 3.2.3. Το αποτέλεσμα είναι ένας πίνακας με γραμμές που αντιστοιχούν στο πλήθος των κανόνων που υπάρχουν στο rule_base και στήλες που αντιστοιχούν στο πλήθος όλων των ταινιών.

```
> lhs(getModel(recommenderAR)$rule_base)
itemMatrix in sparse format with
 195 rows (elements/transactions) and
 1664 columns (items)
```

Αν ζητηθούν τα αποτελέσματα για τις πρώτες 4 γραμμές (κανόνες) και τις πρώτες 3 στήλες (ταινίες) με μορφή matrix, εμφανίζεται ένας πίνακας με τις σχέσεις μεταξύ κανόνων και

ταινιών να είναι τιμές true-false. Οι τιμές true δηλώνουν πως η συγκεκριμένη ταινία είναι μέλος του σώματος του συγκεκριμένου κανόνα.

Σύμφωνα με το αποτέλεσμα, φαίνεται η ταινία “Toy Story (1995)” να βρίσκεται στο σώμα του δεύτερου κανόνα.

```
> as(lhs(getModel(recommenderAR)$rule_base)[1:4,1:3], "matrix")
     Toy Story (1995) GoldenEye (1995) Four Rooms (1995)
[1,]            FALSE            FALSE             FALSE
[2,]             TRUE            FALSE             FALSE
[3,]            FALSE            FALSE             FALSE
```

```
[4,]              FALSE              FALSE              FALSE
```

Αυτό επιβεβαιώνεται με την κλήση του δεύτερου κανόνα από το μοντέλο ως dataframe. Στο σώμα του συγκεκριμένου κανόνα βρίσκεται όντως η συγκεκριμένη ταινία.

```
> as(getModel(recommenderAR)$rule_base[2],"data.frame")
                                                                    rules  support  confidence  coverage       lift
count
145 {Toy Story (1995),Return of the Jedi (1983)} => {Star Wars (1977)}     0.2  0.9615385     0.208 1.800634
100
```

Με την κλήση της εντολής is.subset(lhs(model$rule_base), newdata@data) επιστρέφεται ένας πίνακας με γραμμές το πλήθος των κανόνων του rule_base και στήλες ίσες με τον αριθμό των χρηστών του newdata, όπως φαίνεται στο παρακάτω παράδειγμα.

```
>is.subset(lhs(getModel(recommenderAR)$rule_base), binarized[501:503]@data)
195 x 3 sparse Matrix of class "ngCMatrix"
```

Οι μεταξύ τους τιμές είναι true-false, με την τιμή true να δηλώνει πως το αριστερό μέρος του συγκεκριμένου κανόνα περιέχει τις ταινίες που έχει βαθμολογήσει ο συγκεκριμένος χρήστης και επομένως μπορεί να χρησιμοποιηθεί για να του παράξει κάποια σύσταση. Στο τέλος, αυτός ο πίνακας αποθηκεύεται στην μεταβλητή m.

Στην συνέχεια δημιουργούνται δύο κενές λίστες, η reclist που στην συνέχεια θα περιέχει τις ταινίες που θα προταθούν για κάθε χρήστη, και την ratings που είναι η λίστα που θα περιέχει τις τιμές του sort_measure που είχε ο κάθε κανόνας από τον οποίο προέκυψαν οι συστάσεις της λίστας reclist.

Έπειτα, ξεκινάει μία εντολή for με κάθε επανάληψη να εκτελείται για κάθε έναν χρήστη του πίνακα newdata. Για κάθε χρήστη, θα απομονώσει τους κανόνες του rule_base που ο πίνακας m δηλώνει πως θα χρησιμοποιηθούν για την παραγωγή συστάσεων του συγκεκριμένου χρήστη και τους κατατάσει με βάση το sort_measure που έχει δοθεί, αποθηκεύοντας τους στην μεταβλητή ars.

Με την εντολή LIST(rhs(ars)) απομονώνονται οι κεφαλές αυτών των κανόνων σε μία λίστα, δηλαδή οι συστάσεις που έχουν προκύψει, και με την συνάρτηση unlist() αυτές οι συστάσεις μετατρέπονται σε ένα διάνυσμα (vector) [48] και αποθηκεύονται στην μεταβλητή ar_rhs.

Στην Ενότητα 3.2.3 προβάλλονται τα περιεχόμενα του rule_base και οι στήλες που περιέχει, όπως η στήλη rules που περιέχει τους κανόνες με το σώμα και την κεφαλή τους, το support που περιέχει τις τιμές support για κάθε κανόνα κλπ. Με την χρήση της συνάρτησης quality() γίνεται η πρόσβαση στα δεδομένα του ars και επιλέγεται η στήλη της οποίας το όνομα αντιστοιχεί σε αυτό του sort_measure και αποθηκεύεται στην μεταβλητή ar_qualities. Για παράδειγμα, αν το sort_measure που δόθηκε στις παραμέτρους ήταν το support, τότε στην μεταβλητή θα αποθηκεύονταν όλες οι τιμές support των κανόνων που περιέχονται στην μεταβλητή ars.

Καθώς πλέον οι κεφαλές των κανόνων βρίσκονται μεταβλητή ar_rhs, κάποιες από αυτές τις κεφαλές μπορεί να παρουσιάζονται περισσότερες από μία φορές, όμως δεν είναι επιθυμητό να προταθεί η ίδια ταινία πολλαπλές φορές. Η duplicated() είναι μία συνάρτηση, η οποία αν δεχτεί ένα διάνυσμα με

τιμές, θα επιστρέψει ένα διάνυσμα με τιμές true-false ίδιας διάστασης με το διάνυσμα που της δόθηκε [49]. Σε κάθε θέση του δυνανίσματος που επιστρέφεται, εμφανίζεται η τιμή false αν η αντίστοιχη τιμή του διανύσματος που δέχτηκε εμφανίζεται πρώτη φορά (δηλαδή δεν ειναι διπλότυπη) ενώ τιμή true θα αντιστοιχίζεται στις τιμές εκείνες που δεν εμφανίζονται για πρώτη φορά (δηλαδή είναι διπλότυπες).

Στο παρακάτω παράδειγμα η συνάρτηση δέχεται ένα διάνυσμα 7 αριθμητικών τιμών, και επιστρέφεται ένα διάνυσμα 7 λογικών τιμών. Όταν οι τιμές 1,2,3 εμφανίζονται στις πρώτες 3 θέσεις του διανύσματος αντίστοιχα, τους αντιστοιχίζεται η τιμή false καθώς εμφανίζονται πρώτη φορά και δεν είναι διπλότυπες. Αμέσως μετά όμως επαναλαμβάνονται για τις θέσεις 4-6 και τους αντιστοιχίζεται η τιμή true, καθώς είναι πλέον διπλότυπα. Το 4 που εμφανίζεται για πρώτη φορά στην έβδομη θέση έχει επίσης την τιμή false, καθώς δεν είναι διπλότυπο.

```
> duplicated(c(1,2,3,1,2,3,4))
[1] FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
```

Έτσι, και για τις ταινίες του διανύσματος ar_rhs θα γίνει η ίδια διαδικασία, με την μεταβλητή ar_duplicates να περιέχει ένα διάνυσμα μεγέθους ίσου του ar_rhs με λογικές τιμές που δείχνουν αν η κάθε κεφαλή εμφανίζεται για πρώτη φορά ή εμφανίζεται για πολλαπλή φορά.

Ακολουθεί η εντολή ar_rhs[!ar_duplicates] της οποίας το αποτέλεσμα αποθηκεύεται την μεταβλητή recom_item. Όπως αναφέρθηκε παραπάνω, η ar_duplicates περιέχει το λογικό διάνυσμα των λογικών τιμών που αντιστοιχίζονται στο διάνυσμα των ταινιών ar_rhs. Με την χρήση του θαυμαστικού μπροστά στο λογικό διάνυσμα ar_duplicates θα αντιστρέψει τιμές από true σε false και αντίστροφα. Αυτό σημαίνει πως όπου τιμή true πλέον θα σημαίνει ότι η αντίστοιχη ταινία εμφανίζεται πρώτη φορά, ενώ false πως η ταινία εμφανίζεται για δεύτερη ή παραπάνω φορές. Έτσι, αποθηκεύονται στην μεταβλητή recom_item το διάνυσμα των ταινιών εκείνων οι οποίες έχουν τιμή true, με τις διπλότυπες να αφαιρούνται πλέον.

Φαίνεται και στο παρακάτω παράδειγμα πως αντιστρέφοντας τις λογικές τιμές που επιστρέφει η συνάρτηση duplicated() παραμένουν πλέον μόνο οι αριθμητικές τιμές του διανύσματος που στην αντίστοιχη θέση του αντιστραμμένου λογικου διανύσματος είχαν τιμή true.

```
> !duplicated(c(1,2,3,1,2,3,4))
[1]  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE
> c(1,2,3,1,2,3,4)[!duplicated(c(1,2,3,1,2,3,4))]
[1] 1 2 3 4
```

Επομένως, η μεταβλητή recom_item περιέχει ένα διάνυσμα ταινιών που προέκυψαν από τις κεφαλές των κανόνων που χρησιμοποιήθηκαν για την παραγωγή συστάσεων για τον συγκεκριμένο χρήστη. Η κάθε ταινία αυτού του διανύσματος εμφανίζεται μόνο από μία φορά και είναι ταξινομημένες με βάση το sort_measure που έχει δοθεί.

Στην συνέχεια, η ίδια διαδικασία με το αντίστροφο λογικό διάνυσμα εκτελείται και για την μεταβλητή recom_qual, η οποία αντί για τα ονόματα των ταινιών, είναι ένα διάνυσμα ίδιου μήκους που στις

αντίστοιχες θέσεις περιέχει τις αριθμητικές τιμές του sort_measure, καθώς το διάνυσμα ar_qualities περιέχει αυτές.

Στο τέλος, στις λίστες reclist και ratings αποθηκεύονται τα δύο διανύσματα ταινιών και αριθμητικών τιμών sort_measure αντίστοιχα. Καθώς όλη η παραπάνω διαδικασία θα επαναληφθεί για τον επόμενο χρήστη του newdata, θα αποτελέσουν λίστες πολλών διανυσμάτων και ανάλογα την τιμή του n, δηλαδή το μήκος της topNList που έχει ζητηθεί, θα επιστρέφονται από την μέθοδο και οι ανάλογες top ταινίες των διανυσμάτων αυτών.

Αφού παρουσιάστηκε αναλυτικά ο κώδικας που εκτελείται για την παραγωγή αυτών των συστάσεων, εκτελώντας ξανά το παράδειγμα από την αρχή του κεφαλαίου, παρατηρείται πως για τον δεύτερο χρήστη του test set (χρήστης 502), η μέθοδος predict() δεν έχει επιστρέψει καμία σύσταση, δηλαδή είναι κενή η λίστα των συστάσεων. Αυτό μπορεί να συμβεί στο στάδιο της απομόνωσης των κανόνων. Ο συγκεκριμένος χρήστης μπορεί είτε να μην είχε βαθμολογήσει ως true καμία ταινία και επομένως να μην ήταν εφικτό να φιλτραριστούν οι κανόνες αφού δεν υπήρχε κανένα δεδομένο επιλογής τους, ή θα μπορούσε να μην υπήρχε κανένας κανόνας που στο αριστερό του μέρος να είχε τις ταινίες που είχε βαθμολογήσει ως true.

```
> predictionsAR <- predict(recommenderAR, binarized[501:503], n=5)
> predictionsAR
Recommendations as 'topNList' with n = 5 for 3 users.
> as(predictionsAR, "list")
$`0`
[1] "Star Wars (1977)"                  "Pulp Fiction (1994)"
[3] "Raiders of the Lost Ark (1981)"    "Empire Strikes Back, The (1980)"
[5] "Silence of the Lambs, The (1991)"

$`1`
character(0)

$`2`
[1] "Pulp Fiction (1994)"               "Shawshank Redemption, The (1994)"
[3] "Terminator, The (1984)"            "Usual Suspects, The (1995)"
```

3.4.5 Η συνάρτηση evaluationScheme()

Ένα από τα σημαντικότερα εργαλεία που προσφέρει η βιβλιοθήκη Recommenderlab είναι αυτό της αξιολόγησης των διαφόρων αλγορίθμων που χρησιμοποιούνται για την παραγωγή συστάσεων. Η evaluationScheme() είναι μία συνάρτηση που επιστρέφει ένα αντικείμενο evaluationScheme, το οποίο αποτελεί ένα σχέδιο αξιολόγησης, στο οποίο έχει οριστεί ποιο θα είναι το σύνολο των δεδομένων το οποίο αργότερα θα χωριστεί σε training set και σε test set, ποια μέθοδος θα χρησιμοποιηθεί για αυτόν τον διαχωρισμό, το given, που είναι ένας αριθμός απαραίτητος για την διαδικασία της επιλογής των

κατάλληλων κανόνων για κάθε χρήστη του test set για την παραγωγή των συστάσεων, και όποια άλλη παράμετρος σχετική με την μέθοδο είναι απαραίτητη [3],[32].

- **Given**

Το given είναι από τις πιο σημαντικές παραμέτρους της evaluationScheme(), καθώς η τιμή του μπορεί να αλλάξει καθοριστικά τα αποτελέσματα της αξιολόγησης.

Δεδομένου ότι, στην περίπτωση των Association Rules, από το training set έχουν προκύψει κάποιοι κανόνες, στην συνέχεια αυτοί οι κανόνες θα χρησιμοποιηθούν από τον αλγόριθμο για να κάνει προβλέψεις για τους χρήστες του test set και να αξιολογήσει αν οι προβλέψεις του ήταν σωστές. Όμως, για τους χρήστες του test set, ο αλγόριθμος δεν γνωρίζει καμία πληροφορία και επομένως χρειάζεται κάποιο κριτήριο για να μπορέσει να διαλέξει κανόνες από το rule_base και κατά συνέπεια να παράξει συστάσεις.

Σε αυτό το σημείο έρχεται το given ως παράμετρος. Είναι ένας ακέραιος αριθμός, θετικός ή αρνητικός, και ορίζει πριν από το στάδιο των προβλέψεων πόσες ταινίες θα αποσπαστούν από τον κάθε χρήστη για να τις χρησιμοποιήσει ο αλγόριθμος ως κριτήρια για την επιλογή των κανόνων [3],[32].

Για παράδειγμα, αν μετά το στάδιο του binarize() ένας χρήστης έχει βαθμολογήσει 10 ταινίες ως true και αργότερα αποτελέσει χρήστη του test set, αν οριστεί το given να είναι 3, τότε ο αλγόριθμος θα αποσπάσει τυχαία 3 από τις βαθμολογημένες ταινίες του χρήστη και θα αναζητήσει κανόνες που να έχουν αυτές τις ταινίες στο σώμα τους ώστε να παράξει συστάσεις. Στην περίπτωση που δοθεί αρνητικός αριθμός, για παράδειγμα -3, τότε θα αποσπαστούν όλες οι ταινίες του χρήστη εκτός από 3, δηλαδή θα αποσπαστούν 10-3=7.

Η σημασία του given γίνεται αντιληπτή όταν στο παραπάνω παράδειγμα ληφθούν υπόψη τα εξής: δεδομένου ότι οι ταινίες που θα αποσπαστούν δεν μπορούν να χρησιμοποιηθούν αργότερα στις προβλέψεις, δηλαδή δεν μπορούν να προκύψουν ως συστάσεις για τον συγκεκριμένο χρήστη, αν ζητηθεί η αξιολόγηση να γίνει για top5List, τότε στο πρώτο παράδειγμα θα μπορούσαν να παραχθούν θετικά αποτελέσματα καθώς με την απόσπαση των τριών ταινιών έχουν απομείνει άλλες 7 για να μπορέσει ο αλγόριθμος να τις συμπεριλάβει στην λίστα.

Στο δεύτερο παράδειγμα όμως, μετά την απόσπαση των 7 ταινιών, έχουν παραμείνει μόνο 3. Έτσι στην περίπτωση της top5List, ακόμα κι αν ο αλγόριθμος πετύχαινε και τις 3 ταινίες σωστά, θα έπρεπε να συμπεριλάβει αναγκαστικά άλλες 2 τις οποίες ο χρήστης έχει βαθμολογήσει ως false, ρίχνοντας έτσι την απόδοση του αλγόριθμου.

- **Methods**

Η evaluationScheme() δέχεται ως παράμετρο ένα σύνολο δεδομένων, το οποίο θα χωρίσει σε training set και σε test set προκειμένου να παράξει κανόνες και αργότερα να κάνει προβλέψεις. Ο τρόπος με τον οποίον θα διαχωρίσει τα δεδομένα καθορίζεται από την μέθοδο που θα δοθεί επίσης παραμετρικά, και οι επιλογές είναι τρεις [3],[45].

a. Split

Η split, όπως αναφέρθηκε και στην Ενότητα 2.6.1, είναι η πιο απλή μέθοδος διαχωρισμού των δεδομένων η οποία συνοδεύεται από την παράμετρο train. Η τιμή της train είναι ένας δεκαδικός αριθμός που ορίζει τι ποσοστό από τους χρήστες των δεδομένων που δόθηκαν θα χρησιμοποιηθούν για το training set και την παραγωγή των κανόνων, ενώ το υπόλοιπο ποσοστό των χρηστών θα πάρει μέρος στο test set για την διαδικασία της αξιολόγησης. Με την μεταβλητή k μπορεί να οριστεί πόσες φορές θα εκτελεστεί αυτή η διαδικασία (folds/runs) και η προεπιλογή της για την συγκεκριμένη μέθοδο είναι το 1, ενώ η τιμή της train είναι 0.9 [32].

```
> schemeAR<- evaluationScheme(binarized[1:600], method = "split", train=0.7, given=1)
> schemeAR
Evaluation scheme with 1 items given
Method: 'split' with 1 run(s).
Training set proportion: 0.700
Good ratings: NA
Data set: 600 x 1664 rating matrix of class 'binaryRatingMatrix' with 37054 ratings.
```

b. Cross Validation

Η cross validation, όπως αναφέρθηκε και στην Ενότητα 2.6.2, αναφέρεται στην περίπτωση της k-fold cross validation και συνοδεύεται από την παράμετρο k, η οποία ορίζει τον αριθμό των folds που θα πραγματοποιηθούν. Η λογική αυτής της μεθόδου είναι ότι θα διαιρέσει τα δεδομένα σε όσα ίσα τμήματα δηλώνει το k, και θα πραγματοποιήσει k φορές την εξής διαδικασία: θα χρησιμοποιήσει ένα από τα k τμήματα ως test set και όλα τα υπόλοιπα ως training set, έτσι ώστε να έχει υπάρξει από μία φορά που το κάθε τμήμα θα έχει χρησιμοποιηθεί ως test set. Η προεπιλεγμένη τιμή του k για το cross validation είναι το 10 [32].

```
> schemeAR<- evaluationScheme(binarized[1:600], method = "cross", k=5, given=1)
> schemeAR
Evaluation scheme with 1 items given
Method: 'cross-validation' with 5 run(s).
Good ratings: NA
Data set: 600 x 1664 rating matrix of class 'binaryRatingMatrix' with 37054 ratings.
```

c. Bootstrap

Η bootstrap, όπως αναφέρθηκε και στην Ενότητα 2.6.3, συνοδεύεται από την μεταβλητή train που ορίζει το ποσοστό των δεδομένων που θα χρησιμοποιηθούν για το training set. Η bootstrap αποτελεί δειγματοληψία με αντικατάσταση, επομένως από αυτό το σύνολο των δεδομένων θα γίνεται τυχαία δειγματοληψία μεγέθους όσο ορίζει το ποσοστό του training set. Ένας χρήστης που επιλέχθηκε στο training set μπορεί να επιλεχθεί ξανά στην συγκεκριμένη δειγματοληψία, καθώς μετά την επιλογή του σημειώνεται από τον αλγόριθμο ως χρήστης του train set και επιστρέφεται πίσω στο σύνολο των δεδομένων. Στο τέλος, όσοι χρήστες δεν επιλέχθηκαν για να αποτελέσουν κομμάτι του training set μία ή περισσότερες φορές, θα είναι μέρος του test set. Με την μεταβλητή k μπορεί να οριστεί πόσες φορές να εκτελεστεί αυτή η διαδικασία και η προεπιλεγμένη της τιμή είναι το 10, ενώ η τιμή της train είναι 0.9 [32].

```
> schemeAR<- evaluationScheme(binarized[1:600], method = "bootstrap", train=0.8, k=4, given=1)
> schemeAR
Evaluation scheme with 1 items given
Method: 'bootstrap' with 4 run(s).
Training set proportion: 0.800
Good ratings: NA
Data set: 600 x 1664 rating matrix of class 'binaryRatingMatrix' with 37054 ratings.
```

3.4.6 Η συνάρτηση evaluate()

Το evaluationScheme αντικείμενο που δημιουργήθηκε θα χρησιμοποιηθεί στην συνέχεια από την συνάρτηση evaluate(), που θα αξιολογήσει τις συστάσεις. Η evaluate() δέχεται ως παραμέτρους το evaluationScheme που έχει δημιουργηθεί, την μέθοδο που θα εφαρμόσει στα δεδομένα, που στην περίπτωση του παραδείγματος είναι η Association Rules, το type που καθορίζει σε τι τύπου δεδομένα γίνεται αξιολόγηση και μπορεί να πάρει τις τιμές 'topNList' (που αποτελεί την προεπιλογή), δηλαδή δηλώνει ότι θα αξιολογηθεί η ικανότητα του αλγόριθμου να παράγει top-N συστάσεις, ή 'ratings', δηλαδή δηλώνει ότι θα αξιολογηθεί η ικανότητα του αλγόριθμου να προβλέπει βαθμολογίες που θα έδιναν οι χρήστες στις ταινίες. Στην περίπτωση των Association Rules πρακτική σημασία έχει η αξιολόγηση σε topNLists, καθώς δεν προβλέπονται αριθμητικές βαθμολογίες, καθώς το σύνολο των δεδομένων έχει δυαδική μορφή. Επίσης, παραμετρικά στην evaluate() δίνεται το n, που ισχύει μόνο για την περίπτωση που η αξιολόγηση γίνεται σε topNLists, και είναι ένα διάνυσμα ακεραίων τιμών, με την κάθε τιμή από αυτές να δηλώνει το N από μία topNList για την οποία θα αξιολογηθούν οι συστάσεις, και η param που δέχεται τις παραμέτρους της μεθόδου που επιλέχθηκε [3],[32].

Στο παράδειγμα που ακολουθεί, η evaluationScheme() συνάρτηση δέχεται τους πρώτους 600 χρήστες του πίνακα binarized ως δεδομένα, την μέθοδο cross-validation με 5 folds και 1 given. Το αποτέλεσμα αποθηκεύεται στη μεταβλητή schemeAR και στην συνέχεια χρησιμοποιείται από την συνάρτηση evaluate() και παρουσιάζονται τα 5 folds. Στην κλήση της συνάρτησης evaluate() δίνονται το schemeAR που δημιουργήθηκε παραπάνω, η μέθοδος AR για να εφαρμοστεί στα δεδομένα, ο τύπος

των προβλέψεων που θα ειναι topNList, το n που δηλώνει πως θα παραχθούν προβλέψεις για top 1-2-3-4-5 λίστες, και το param που περιέχει τις παραμέτρους εκείνες που θα χρειαστεί η μέθοδος AR, όπως confidence, support και maxlength, που όμως όταν παραλείπονται, χρησιμοποιούνται οι προεπιλεγμένες τους τιμές για την παραγωγή των κανόνων. Τα αποτελέσματα που παράγονται αποθηκεύονται την μεταβλητή results.

```
> schemeAR<- evaluationScheme(binarized[1:600], method = "cross", k=5, given=1)
> results <- evaluate(schemeAR, method= "AR", type="topNList", n=c(1,2,3,4,5), param=list(sup=0.2, conf=0.3, maxlen=3))
AR run fold/sample [model time/prediction time]
        1  [0sec/0.27sec]
        2  [0.02sec/0.26sec]
        3  [0sec/0.28sec]
        4  [0sec/0.26sec]
        5  [0.02sec/0.27sec]
> results
Evaluation results for 5 folds/samples using method 'AR'.
```

Τα αποτελέσματα μπορούν να προβληθούν με την συνάρτηση getConfusionMatrix() [3], τα οποία όμως στην συγκεκριμένη περίπτωση θα περιέχουν τα αποτελέσματα που προέκυψαν από κάθε fold.

```
> getConfusionMatrix(results)
[[1]]
            TP         FP       FN       TN    N precision      recall         TPR          FPR n
[1,] 0.04166667 0.01666667 62.46667 1600.475 1663 0.7142857 0.000555051 0.000555051 1.025020e-05 1
[2,] 0.07500000 0.04166667 62.43333 1600.450 1663 0.6428571 0.001022012 0.001022012 2.590252e-05 2
[3,] 0.10833333 0.05833333 62.40000 1600.433 1663 0.6428571 0.001456290 0.001456290 3.623979e-05 3
[4,] 0.13333333 0.08333333 62.37500 1600.408 1663 0.6071429 0.001738964 0.001738964 5.201076e-05 4
[5,] 0.14166667 0.12500000 62.36667 1600.367 1663 0.5285714 0.001806169 0.001806169 7.830067e-05 5


[[2]]
           TP         FP       FN       TN    N precision      recall         TPR          FPR n
[1,] 0.1166667 0.03333333 58.74167 1604.108 1663 0.7777778 0.002685909 0.002685909 2.030696e-05 1
[2,] 0.1833333 0.11666667 58.67500 1604.025 1663 0.6111111 0.004084697 0.004084697 7.147360e-05 2
[3,] 0.2250000 0.19166667 58.63333 1603.950 1663 0.5370370 0.004912320 0.004912320 1.174696e-04 3
[4,] 0.2750000 0.25833333 58.58333 1603.883 1663 0.5138889 0.006038307 0.006038307 1.583624e-04 4
[5,] 0.3250000 0.32500000 58.53333 1603.817 1663 0.5000000 0.007229416 0.007229416 1.993753e-04 5
```

```
[[3]]
           TP         FP       FN       TN    N precision      recall         TPR          FPR n
[1,] 0.1250000 0.04166667 67.20833 1595.625 1663     0.750 0.003477209 0.003477209 2.570821e-05 1
[2,] 0.2000000 0.11666667 67.13333 1595.550 1663     0.600 0.004697119 0.004697119 7.233369e-05 2
[3,] 0.2666667 0.17500000 67.06667 1595.492 1663     0.575 0.006002103 0.006002103 1.087916e-04 3
[4,] 0.3416667 0.22500000 66.99167 1595.442 1663     0.575 0.006923331 0.006923331 1.393007e-04 4
[5,] 0.4000000 0.26666667 66.93333 1595.400 1663     0.570 0.007966403 0.007966403 1.648652e-04 5


[[4]]
            TP         FP       FN       TN    N precision      recall         TPR          FPR n
[1,] 0.07500000 0.03333333 54.28333 1608.608 1663 0.6923077 0.002044699 0.002044699 2.031098e-05 1
[2,] 0.09166667 0.09166667 54.26667 1608.550 1663 0.5384615 0.002599237 0.002599237 5.632535e-05 2
[3,] 0.14166667 0.10833333 54.21667 1608.533 1663 0.5897436 0.004970624 0.004970624 6.662104e-05 3
[4,] 0.17500000 0.12500000 54.18333 1608.517 1663 0.6153846 0.006122499 0.006122499 7.674673e-05 4
[5,] 0.19166667 0.15833333 54.16667 1608.483 1663 0.6000000 0.006540746 0.006540746 9.713732e-05 5


[[5]]
            TP         FP       FN       TN    N precision      recall         TPR          FPR n
[1,] 0.05833333 0.05833333 60.66667 1602.217 1663 0.5000000 0.001771218 0.001771218 3.543179e-05 1
[2,] 0.08333333 0.11666667 60.64167 1602.158 1663 0.4642857 0.002414532 0.002414532 7.087270e-05 2
[3,] 0.11666667 0.15833333 60.60833 1602.117 1663 0.5000000 0.005089914 0.005089914 9.630264e-05 3
[4,] 0.14166667 0.20833333 60.58333 1602.067 1663 0.5000000 0.006716473 0.006716473 1.267741e-04 4
[5,] 0.14166667 0.27500000 60.58333 1602.000 1663 0.4607143 0.006716473 0.006716473 1.673434e-04 5
```

Επομένως, για να προβληθεί μία συνολική εικόνα για όλα τα folds μπορεί να χρησιμοποιηθεί η μέθοδος avg() για να εμφανίσει τους μέσους όρους των τιμών [3].

```
> avg(results)
            TP         FP       FN       TN    N precision      recall         TPR          FPR n
[1,] 0.08333333 0.03666667 60.67333 1602.207 1663 0.6868742 0.002106817 0.002106817 2.240163e-05 1
[2,] 0.12666667 0.09666667 60.63000 1602.147 1663 0.5713431 0.002963520 0.002963520 5.938157e-05 2
[3,] 0.17166667 0.13833333 60.58500 1602.105 1663 0.5689276 0.004486250 0.004486250 8.508493e-05 3
[4,] 0.21333333 0.18000000 60.54333 1602.063 1663 0.5622833 0.005507915 0.005507915 1.106389e-04 4
[5,] 0.24000000 0.23000000 60.51667 1602.013 1663 0.5318571 0.006051841 0.006051841 1.414044e-04 5
```

Μέσα στον κώδικα της συνάρτησης evaluate(), η πρώτη διαδικασία που εκτελείται είναι αυτή του διαχωρισμού των χρηστών για να καθοριστεί ποιοι θα πάρουν μέρος στο training set και ποιοί στο test set, όπως φαίνεται στην Εικόνα 3.4.2, και αυτός ο διαχωρισμός γίνεται με την βοήθεια του αντικειμένου evaluationScheme που έχει περάσει παραμετρικά στην evaluate() [50] κατά την κλήση της.

Το training set αποτελείται από τους χρήστες που θα πάρουν μέρος στην εκπαίδευση του αλγόριθμου, δηλαδή στην παραγωγή των κανόνων, για αυτό και θα δοθεί στη συνάρτηση Recommender() που θα κληθεί στην συνέχεια για την δημιουργία του μοντέλου που περιέχει τους κανόνες συσχέτισης.

Το test set όμως, σύμφωνα με την Εικόνα 3.2 [50], φαίνεται να αποτελείται από 2 μέρη λόγω του given που αναφέρθηκε παραπάνω: στο test_known και στο test_unknown, που το πρώτο θα χρησιμοποιηθεί ως το σύνολο δεδομένων για την επιλογή των κατάλληλων κανόνων για κάθε χρήστη του test set προκειμένου να προκύψουν συστάσεις, ενώ το δεύτερο θα χρησιμοποιηθεί ως το σύνολο δεδομένων με βάση το οποίο θα αξιολογηθεί αν οι προβλέψεις που παρήγαγε ο αλγόριθμος ήταν σωστές ή όχι. Τα δύο αυτά σύνολα δεδομένων περιέχουν ακριβώς τους ίδιους χρήστες, αλλά διαφέρουν στις βαθμολογικές τιμές που περιέχουν [32].

Καθώς έχει δοθεί το given στο evaluation scheme, ο αλγόριθμος πλέον ξέρει ότι πρέπει να εκτελέσει την διαδικασία να αποσπάσει για κάθε χρήστη όσες ταινίες ορίζει το given από αυτές που έχει βαθμολογήσει ως true και να κάνει προβλέψεις με βάση αυτές. Επομένως, το test_known είναι το σύνολο των δεδομένων στο οποίο οι χρήστες έχουν βαθμολογίες (τιμές true) στις ταινίες που έχουν επιλεχθεί τυχαία από το given, ενώ όλες οι υπόλοιπες θα έχουν τιμές false [50]. Ο αλγόριθμος δεν γνωρίζει τις υπόλοιπες ταινίες που έχει βαθμολογήσει ο χρήστης και φτιάχνει για τον κάθε χρήστη του test set τις top-N συστάσεις. Το test_known είναι που θα δοθεί στην μέθοδο predict() για να την παραγωγή των συστάσεων, με το αντικείμενο Recommender που δημιουργήθηκε παραπάνω.

```r
## prepare data
train <- getData(scheme, type = "train", run = run)
test_known <- getData(scheme, type = "known", run = run)
test_unknown <- getData(scheme, type = "unknown", run = run)
given <- getData(scheme, type = "given", run = run)

## train recommender
time_model <- system.time(rec <-
    Recommender(train, method, parameter = parameter),
  gcFirst = FALSE)


time_predict <- system.time(pre <-
    predict(rec, test_known, n = max(n), type = type),
  gcFirst = FALSE)
```

Εικόνα 3.2 evaluate.R: Διαχωρισμός δεδομένων και κλήση της συνάρτησης Recommender() και της μεθόδου predict()

Επομένως, το test_unknown περιέχει ακριβώς τους ίδιους χρήστες που περιέχει και το test_known, και περιέχει τις βαθμολογίες του χρήστη, χωρίς όμως να υπάρχουν στήλες των ταινιών εκεινων που χρησιμοποιήθηκαν στο test_known για την παραγωγή των συστάσεων στον συγκεκριμένο χρήστη. Όταν καλείται η συνάρτηση evaluate() δέχεται ως παράμετρο το διάνυσμα n που περιέχει τις topNLists που πρέπει να αξιολογήσει. Για την κάθε τιμή αυτού του διανύσματος θα απομονώσει από τις λίστες των προβλέψεων που προέκυψαν από την μέθοδο predict() τις ανάλογες top ταινίες και εκτελεί την αξιολόγηση για την κάθε μία από αυτές. Η απομόνωση των top-N ταινιών από την

συνολική λίστα γίνεται με την χρήση της μεθόδου bestN() [3],[32] που δέχεται μία topNList και τον αριθμό των αντικειμένων που θα απομονώσει από αυτή τη λίστα και θα τις επιστρέψει. Για παράδειγμα, αν οι συστάσεις που παρήχθησαν από την predict() για έναν χρήστη ήταν 10 όμως έχει ζητηθεί από τον αλγόριθμο να αξιολογήσει top-5-list, τότε θα δοθούν παραμετρικά στην bestN() η λίστα με τις 10 ταινίες και ο αριθμός 5, και αυτή θα επιστρέψει την λίστα μόνο με τις πρώτες 5 ταινίες.

Αυτά τα αποτελέσματα μαζί με το test unknown θα χρησιμοποιηθούν για να κληθεί η calcPredictionAccuracy(), όπως φαίνεται και στην Εικόνα 3.3 [50], που είναι η συνάρτηση που ευθύνεται για όλα τα αποτελέσματα που επιστρέφονται από την evaluate().

```
if (is(pre, "topNList")) {
  res <- NULL
  for (i in 1:length(n)) {
    NN <- n[i]

    ## get best N
    topN <- bestN(pre, NN)

    r <-  calcPredictionAccuracy(
      topN,
      test_unknown,
      byUser = FALSE,
      given = given,
      goodRating = scheme@goodRating
    )
    res <- rbind(res, r)
  }
  res <- cbind(res, n)
```

Εικόνα 3.3 evaluate.R: Κλήση της συνάρτησης calcPredictionAccuracy() από την συνάρτηση evaluate() για κάθε n λίστα που πρέπει να αξιολογήσει

3.4.7 Η συνάρτηση calcPredictionAccuracy()

Η συνάρτηση calcPredictionAccuracy() δέχεται ως παραμέτρους τις topNLists για κάθε έναν από τους χρήστες του test set και αποθηκεύονται στην μεταβλητή x, αποτελώντας τις προβλέψεις που θα αξιολογηθούν, το test_unknown που αποτελεί τα πραγματικά δεδομένα και αποθηκεύεται στην μεταβλητή data, το byUser που δηλώνει αν θα παρουσιαστούν τα αποτελέσματα εξατομικευμένα για τον κάθε χρήστη του test set ή ως ο μέσος όρος όλων και έχει προκαθορισμένη τιμή false, και το given για να γνωρίζει πόσες ταινίες να αφαιρέσει από τους υπολογισμούς. Οι παραπάνω παράμετροι φαίνονται στην Εικόνα 3.4 [51].

```
setMethod("calcPredictionAccuracy", signature(x = "topNList",
  data = "binaryRatingMatrix"),

  function(x,
    data,
    byUser = FALSE,
    given = NULL,
    ...) {
    if (is.null(given))
      stop("You need to specify how many items were given for the
prediction!")
```

Εικόνα 3.4 calcPredictionAccuracy.R: Παράμετροι της συνάρτησης για binaryRatingMatrix

Ακολουθεί ο κώδικας της Εικόνας 3.5 [51], που φαίνονται οι υπολογισμοί για τα αποτελέσματα που παράγει η συνάρτηση, που στην περίπτωση των δυαδικών δεδομένων είναι ο confusion matrix. Το N είναι το σύνολο των ταινιών που παίρνουν μέρος στην αξιολόγηση, επομένως αφαιρείται το given από αυτά καθώς για κάθε χρήστη έχουν αποσπαστεί given ταινίες και δεν μπορούν να προκύψουν ως συστάσεις.

Στην συνέχεια, οι προβλέψεις και τα πραγματικά δεδομένα οργανώνονται σε πίνακα ngCMatrix με δυαδικό περιεχόμενο (TRUE, FALSE) και αθροίζονται οι τιμές true που συμπίπτουν μεταξύ των δύο. Αυτές οι τιμές θα αποτελέσουν τα true positives (TP).

Υπολογίζονται τα predicted positives ως το άθροισμα των true των προβλέψεων, και τα πραγματικά positives ως το άθροισμα των true των πραγματικών δεδομένων.

Για να τον υπολογισμό των false positives (FP), δηλαδή αυτά που προέβλεψε ως positives αλλά ήταν false, αφαιρεί από τα predicted positives τα true positives, τις πραγματικές δηλαδή προβλέψεις που είχαν true τιμές.

Τα false negatives (FN), δηλαδή οι ταινίες που προβλέφθηκαν ως false αλλά ήταν στην πραγματικότητα true, υπολογίζονται με την αφαίρεση των true positives από τις πραγματικές positive τιμές.

Τέλος, τα true negatives (TN) υπολογίζονται με την αφαίρεση των TP, FP, FN από το σύνολο των ταινιών N.

```
# given show up as a prediction of NA and a test data FALSE (TN)
N <- ncol(data) - given

TP <- rowSums(as(x, "ngCMatrix") * as(data, "ngCMatrix"))
PredPositives <- rowSums(as(x, "ngCMatrix"))
Positives <- rowSums(as(data, "ngCMatrix"))
FP <- PredPositives - TP

FN <- Positives - TP
TN <- N - TP - FP - FN
```

Εικόνα 3.5 calcPredictionAccuracy.R: Υπολογισμός confusion matrix

Μαζί με τον confusion matrix επιστρέφονται και κάποιες επιπλέον μετρικές, όπως τα precision, recall, true positive rating και false positive rating και υπολογίζονται με την βοήθεια των τιμών του confusion matrix όπως φαίνεται από τον κώδικα της συνάρτησης calcPredictionAccuracy() στην Εικόνα 3.6 [51]. Όλες οι τιμές ενώνονται σε έναν ενιαίο πίνακα και αν δεν έχουν ζητηθεί τα αποτελέσματα εξατομικευμένα για κάθε χρήστη τότε επιστρέφονται οι μέσοι όροι αυτών.

```
    ## calculate some important measures
    precision <- TP / (TP + FP)
    recall <- TP / (TP + FN)
    TPR <- recall
    FPR <- FP / (FP + TN)

    res <- cbind(TP, FP, FN, TN, N, precision, recall, TPR, FPR)

    #Average over test users
    if (!byUser)
      res <- colMeans(res, na.rm = TRUE)

    res
  })
```

Εικόνα 3.6 calcPredictionAccuracy.R: Υπολογισμός precision, recall, TPR, FPR

3.4.8 Η συνάρτηση plot()

Η βιβλιοθήκη RecommenderLab προσφέρει την συνάρτηση plot() που μπορεί να δεχτεί τα αποτελέσματα μιας αξιολόγησης ως παράμετρο και να τα εμφανίσει σε γράφημα. Δίνονται δύο επιλογές για τις καμπύλες του γραφήματος: η ROC (receiver operating characteristic curve) που είναι η προεπιλογή, και η precision/recall [3],[32],[45].

Στην ROC καμπύλη, στις δύο διαστάσεις του γραφήματος βρίσκονται τα TPR (true positive rating) και τα FPR (false positive rating) αποτελέσματα. Η TPR είναι ένα συνώνυμο της recall και της sensitivity και ορίζεται από τον τύπο της παράστασης (3.1) ενώ η FPR από τον τύπο της παράστασης (3.2) [52],[53]:

$$TPR = \frac{TP}{TP + FN} \tag{3.1}$$

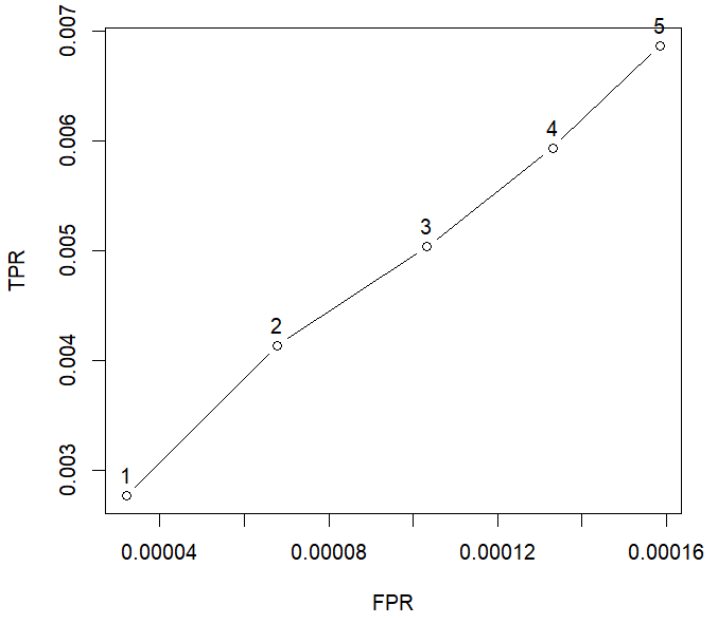
$$FPR = \frac{FP}{FP + TN} \tag{3.2}$$

Η ROC είναι μία καμπύλη που δείχνει την επίδοση ενός μοντέλου ταξινόμησης σε διάφορα κατώφλια ταξινόμησης. Καθώς μειώνεται ένα κατώφλι, τόσο περισσότερα αντικείμενα θα προβλέπονται ως αληθή αυξάνοντας έτσι τα false positives και τα true positives [52].

Η περιοχή κάτω από την ROC καμπύλη ονομάζεται AUC-ROC (area under the ROC curve) και παρέχει έναν τρόπο αξιολόγησης ενός μοντέλου σε όλα τα πιθανά κατώφλια ταξινόμησης. Η AUC μπορεί να πάρει τιμές από 0 έως 1. Στην περίπτωση που το μοντέλο προβλέψει όλες τις περιπτώσεις λάθος έχει τιμή 0, ενώ αν προβλέψει όλες τις περιπτώσεις σωστά έχει τιμή 1. Η τιμή προκύπτει από το εμβαδόν της δισδιάστατης περιοχής κάτω από την ROC καμπύλη. Η ROC καμπύλη είναι μία από τις διαδεδομένες μετρικές για να υπολογιστεί η απόδοση ενός αλγορίθμου μηχανικής μάθησης, και μπορεί να βοηθήσει στην επιλογή του κατάλληλου κατωφλιού, ενώ με την AUC-ROC μπορούν να συγκριθούν διάφορες ROC καμπύλες μεταξύ τους και να επιλεχθεί η πιο αποτελεσματική προσέγγιση στο εκάστοτε πρόβλημα που πρέπει να επιλυθεί [52],[53].

Στα αποτελέσματα που έχουν παραχθεί από την Ενότητα 3.2.6 στην μεταβλητή results για την οποία καλείται η συνάρτηση plot() για την ROC καμπύλη, που φαίνεται στην Εικόνα 3.7, τα διαφορετικά κατώφλια είναι οι topNLists για τις οποίες έχει εκτελεστεί η αξιολόγηση. Η επιλογή του κατάλληλου κατωφλιού γίνεται με βάση τους στόχους του κάθε μοντέλου. Τυπικά, πρέπει να γίνεται η επιλογή εκείνου του κατωφλιού που προσφέρει μεγάλη τιμή TPR και μικρή FPR.

```
> plot(results, annotate=TRUE)
```



Εικόνα 3.7 Καμπύλη ROC με την χρήση της συνάρτησης plot() για top 1 μέχρι 5 λίστες στην περίπτωση του “results”

Η PR (precision-recall) είναι μία καμπύλη που δείχνει την σχέση μεταξύ precision και recall για διάφορα κατώφλια [54],[55]. Όπως προαναφέρθηκε και στην Ενότητα 2.5.4, η precision είναι οι θετικές προβλέψεις που ήταν ήταν όντως θετικές προς όλες τις προβλέψεις που θεωρήθηκαν θετικές. Η recall είναι οι θετικές προβλέψεις που ήταν όντως θετικές προς όλες τις πραγματικές θετικές τιμές.

Όπως και η ROC καμπύλη, έτσι και η PR καμπύλη χρησιμοποιείται για να αξιολογήσει την απόδοση δυαδικών αλγορίθμων ταξινόμησης. Συνήθως χρησιμοποιείται σε περιπτώσεις όπου τα δεδομένα

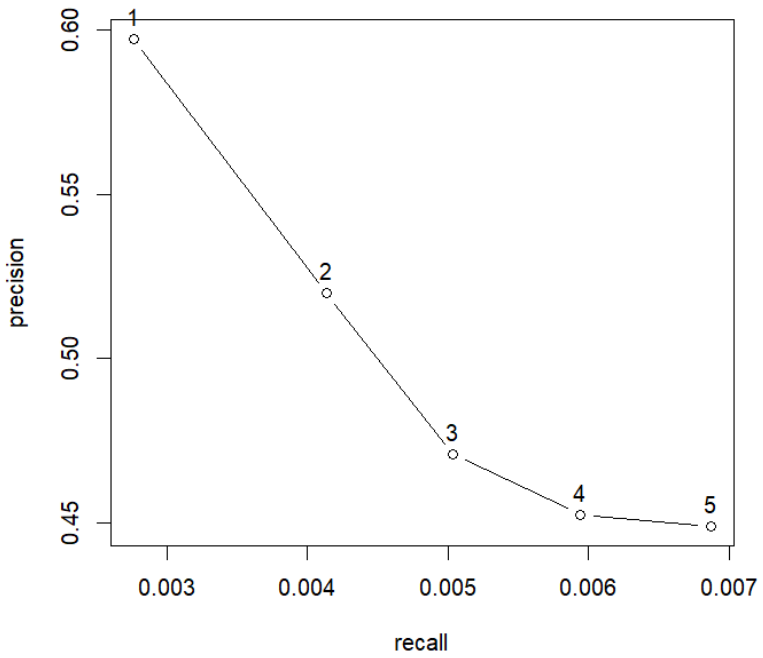
παρουσιάσουν ανισορροπίες ως προς τις κλάσεις τους. Τέλος, όπως και η ROC καμπύλη, έτσι και αυτή παρουσιάζει οπτική αναπαράσταση της απόδοσης του ταξινομητή για διάφορες τιμές κατωφλιού [56].

Η PR καμπύλη δημιουργείται με τον υπολογισμό την precision και της recall για έναν ταξινομητή σε διάφορες τιμές κατωφλιών. Η καμπύλη βοηθάει στην απεικόνιση του πως τα διαφορετικά κατώφλια επηρεάζουν την απόδοση του ταξινομητή, και μπορεί να γίνει καλύτερη η επιλογή ενός από αυτά για την επίλυση ενός συγκεκριμένου προβλήματος. Ένας καλός ταξινομητής θα έπρεπε να υψηλή τιμή precision και υψηλή τιμή recall στο γράφημα [56].

Όπως και στις ROC καμπύλες, έτσι και στην PR η περιοχή κάτω από την καμπύλη ονομάζεται AUC-PR (area under the precision-recall curve). Όσο μεγαλύτερη η τιμή της AUC-PR, τόσο καλύτερα έχει αποδώσει ένας ταξινομητής για ένα συγκεκριμένο πρόβλημα, και η τιμή της προκύπτει από δισδιάστατο εμβαδόν κάτω από την καμπύλη. Μπορεί να πάρει τιμές από 0 μέχρι 1, με το 1 να δηλώνει πως ο ταξινομητής πραγματοποιεί πάντα σωστές προβλέψεις και το 0 το αντίθετο [56].

Στα αποτελέσματα που έχουν παραχθεί από την Ενότητα 3.2.6 στην μεταβλητή results για την οποία καλείται η συνάρτηση plot() για την PR καμπύλη, που φαίνεται στην Εικόνα 3.8, τα διαφορετικά κατώφλια είναι οι topNLists για τις οποίες έχει εκτελεστεί η αξιολόγηση. Η επιλογή του κατάλληλου κατωφλιού γίνεται με βάση τους στόχους του κάθε μοντέλου. Τυπικά, πρέπει να γίνεται η επιλογή εκείνου του κατωφλιού που προσφέρει μεγάλη τιμή precision και μεγάλη recall.

```
> plot(results, "prec/rec", annotate=TRUE)
```



Εικόνα 3.8 Καμπύλη precision/recall με την χρήση της plot() για top 1 μέχρι 5 λίστες στην περίπτωση του “results”

3.5 Επίλογος

Στην παρούσα πτυχιακή εργασία γίνεται χρήση του περιβάλλοντος R/RStudio για την παραγωγή συστάσεων με την χρήση της βιβλιοθήκης RecommenderLab, η οποία παρέχει διάφορες τεχνικές παραγωγής συστάσεων, μεταξύ αυτών και η ARM. Η βιβλιοθήκη διαθέτει ειδικούς πίνακες για την διαχείριση των βαθμολογιών σε αριθμητικά (realRatingMatrix) και δυαδικά (binaryRatingMatrix) δεδομένα. Επιπλέον, διαθέτει ένα εξειδικευμένο αντικείμενο το οποίο περιέχει την εκπαίδευση του μοντέλου παραγωγής συστάσεων, τον Recommender, την συνάρτηση για την δημιουργία του, την Recommender(), και την μέθοδο για την παραγωγή συστάσεων με την χρήση αυτού του αντικειμένου, την predict(). Δίνει βάση στην αξιολόγηση των τεχνικών παραγωγής συστάσεων, για αυτό διαθέτει το αντικείμενο EvaluationScheme το οποίο αποθηκεύει το μοντέλο αξιολόγησης, το σύνολο των δεδομένων που θα χρησιμοποιηθεί για την αξιολόγηση και την τιμή given. Δημιουργείται με την συνάρτηση evaluationScheme() και η συνάρτηση evaluate() επιστρέφει τα αποτελέσματα της αξιολόγησης δεδομένου μιας συγκεκριμένης τεχνικής παραγωγής συστάσεων. Τα αποτελέσματα έχουν την μορφή ενός confusion matrix, μαζί με άλλες τιμές όπως precision, recall, TPR και FPR. Τέλος, η βιβλιοθήκη διαθέτει επίσης την συνάρτηση plot() η οποία επιστρέφει γραφήματα ROC καμπύλων και precision/recall.

Κεφάλαιο 4ο : Σύγκριση μέτρων και μεθόδων με αποτελέσματα συνόλων δεδομένων

4.1 Εισαγωγή

Σε αυτό το Κεφάλαιο γίνεται χρήση της βιβλιοθήκης RecommenderLab για την παραγωγή συστάσεων με τον αλγόριθμο των Association Rules για ταξινόμηση των κανόνων που παρήγαγε ο αλγόριθμος apriori με τα μέτρα αξίας lift και conviction, αφού παρουσιαστούν οι απαραίτητες τροποποιήσεις που έγιναν στον κώδικα της βιβλιοθήκης για την χρήση της conviction. Γίνεται η χρήση αυτών των δύο μέτρων αξίας, καθώς όπως προαναφέρθηκε στην Ενότητα 2.4.1 είναι αυτά που δηλώνουν βαθμό εξάρτησης μεταξύ σώματος και κεφαλής, δηλαδή κατά πόσο η παρουσία του σώματος προκαλεί και επηρεάζει την παρουσία της κεφαλής. Εκτελείται η αξιολόγησή τους και η σύγκρισή τους με το μοντέλο αξιολόγησης cross validation για διαφορετικές τιμές top-N-Lists, min rating και given και τα αποτελέσματα εμφανίζονται για precision και recall σε γραφήματα με την χρήση της βιβλιοθήκης ggplot2. Χρησιμοποιούνται δύο σύνολα δεδομένων: τα βαθμολογικά δεδομένα της σχολής και βαθμολογίες χρηστών σε ταινίες.

4.2 Conviction

Ένα από τα μέτρα αξίας συνδυαστικών κανόνων που θα χρησιμοποιηθούν στην συνέχεια της εργασίας είναι το conviction. Αυτό σημαίνει ότι κατά την χρήση της βιβλιοθήκης, η συνάρτηση Recommender() που θα κληθεί θα πρέπει να δεχτεί ως παράμετρο το sort_measure = “conviction”.

```
> recommenderAR<- Recommender(trainset, "AR", param=list(supp=0.2, conf =0.3,
maxlen=3, sort_measure="conviction"))
Error in .basicRuleMeasure(x, measure, transactions = transactions, ...) :
  unused argument (method = "conviction")
In addition: Warning message:
In interestMeasure(rule_base, method = p$sort_measure, transactions = data) :
interestMeasure: parameter method is now deprecated! Use measure instead!
```

Ωστόσο, ως απάντηση επιστρέφεται ένα σφάλμα στην κονσόλα, το οποίο οδηγεί στον κώδικα του αρχείου RECOM_AR.R, υπεύθυνο για την δημιουργία του Recommender και την κλήση του predict για την μέθοδο των Association Rules.

```
## additional measures for sorting the rulebase
if(p$sort_measure == "cxs") quality(rule_base) <- cbind(quality(rule_base),
  cxs = quality(rule_base)$confidence * quality(rule_base)$support)

if(!p$sort_measure %in% names(quality(rule_base))) quality(rule_base) <-
  cbind(quality(rule_base), interestMeasure(rule_base, method = p$sort_measure,
    transactions = data))

## sort rule_base
rule_base <- sort(rule_base, by = p$sort_measure, decreasing=p$sort_decreasing)
```

Εικόνα 4.1 Κώδικας του RECOM_AR.R για την προσθήκη επιπλέον sort measure

Καθώς το rule_base περιέχει από προεπιλογή κάποιες περιορισμένες στήλες, όπως παρουσιάστηκε στο Κεφάλαιο 3.4.3, μέσα στον κώδικα δίνεται η δυνατότητα να χρησιμοποιηθούν και κάποια άλλα μέτρα αξίας για την κατάταξη των κανόνων, όπως φαίνεται στην Εικόνα 4.1 [31]. Στην αρχή φαίνεται πως έχει προστεθεί η περίπτωση confidence*support, και το αποτέλεσμα γίνεται αργότερα cbind στον rule_base.

Υπάρχουν όμως πολλά ακόμα μέτρα αξίας που μπορούν να χρησιμοποιηθούν. Όλα αυτά περιέχονται στην συνάρτηση interestMeasure(), στην οποία δίνεται ως παράμετρος το sort_measure που έχει δοθεί. Όμως, το σφάλμα που λαμβάνεται από την κονσόλα του RStudio δηλώνει πως η ονομασία method της συγκεκριμένης παραμέτρου έχει καταργηθεί και πως πρέπει πλέον να χρησιμοποιείται η παράμετρος measure. Επομένως, γίνεται η απαραίτητη αλλαγή στον κώδικα για να λειτουργήσει.

```
if(!p$sort_measure %in% names(quality(rule_base))) quality(rule_base) <-
cbind(quality(rule_base), interestMeasure(rule_base, measure =
p$sort_measure, transactions = data))
```

Το RecommenderLab δίνει την δυνατότητα να προσθέσουν οι χρήστες τις δικές τους μεθόδους για την διαχείριση των δεδομένων με την χρήση της εντολής recommenderRegistry$set_entry() [10]. Έτσι, έγινε η επιλογή να δημιουργηθεί μία μέθοδος αντίστοιχη της “AR” με το όνομα “AR_MOD” που περιέχει τον ίδιο κώδικα αλλά με τις τροποποιήσεις που έγιναν για την παρούσα εργασία και θα χρησιμοποιηθεί επίσης στην συνέχεια, όποτε η χρήση της κρίνεται απαραίτητη.

```
recommenderRegistry$set_entry(
  method="AR_MOD", dataType = "binaryRatingMatrix", fun=BIN_AR_MOD,
  description="Recommender based on association rules MODIFIED.",
  parameters=.BIN_AR_MOD_param
)
```

Αφού έχει προστεθεί στις μεθόδους της βιβλιοθήκης που μπορούν πλέον να χρησιμοποιηθούν, πλέον μπορεί να κληθεί από την Recommender().

```
> recommenderAR<- Recommender(trainset, "AR_MOD", param=list(supp=0.2, conf
=0.3, maxlen=3, sort_measure="conviction"))
 Error in .local(x, decreasing, ...) :
Unknown interest measure to sort by.
```

Όμως, παρουσιάζεται ξανά σφάλμα. Αυτό συμβαίνει καθώς η στήλη που δημιουργείται με τις τιμές του conviction δεν έχει τίτλο όταν γίνεται cbind με τον υπόλοιπο rule_base, σε αντίθεση με το μέτρο αξίας confidence*support που του έχει δοθεί το όνομα cxs. Έτσι, στην συνέχεια όταν πρέπει να γίνει η κατάταξη των κανόνων με βάση την στήλη του conviction, αυτή δεν μπορεί να βρεθεί.

Επομένως, μαζί με την αλλαγή της μεταβλητής από method σε measure, η στήλη που θα δημιουργηθεί από το νέο sort_measure θα πάρει το όνομα placeholder_name, το οποίο αμέσως μετά θα αλλάξει στο όνομα του sort_measure με την χρήση της names.

```
#--- changed method to measure and created placeholder name----
   if(!p$sort_measure %in% names(quality(rule_base))) quality(rule_base) <-
cbind(quality(rule_base), placeholder_name = interestMeasure(rule_base,
measure = p$sort_measure, transactions = data))

#----- change the placeholder name to sort measure-------
   names(quality(rule_base))[names(quality(rule_base)) == "placeholder_name"]
<- p$sort_measure
```

Έτσι, η μέθοδος μπορεί πλέον να χρησιμοποιηθεί με όποιο μέτρο αξίας είναι επιθυμητό και βρίσκεται στις διαθέσιμες επιλογές. Επιπλέον, στα αποτελέσματα φαίνεται πλέον η νέα στήλη με το ανάλογο όνομα και τιμές, που στην συγκεκριμένη περίπτωση είναι το conviction.

```
> recommenderAR<- Recommender(trainset, "AR_MOD", param=list(supp=0.2, conf =0.3, maxlen=3, sort_measure="conviction"))
> as(getModel(recommenderAR)$rule_base, "data.frame")
                                                                                                    rules support confidence coverage     lift count conviction
157             {Empire Strikes Back, The (1980),Return of the Jedi (1983)} => {Star Wars (1977)}   0.252  0.9692308    0.260 1.815039   126  15.145000
145                          {Toy Story (1995),Return of the Jedi (1983)} => {Star Wars (1977)}   0.200  0.9615385    0.208 1.800634   100  12.116000
175            {Silence of the Lambs, The (1991),Return of the Jedi (1983)} => {Star Wars (1977)}   0.228  0.9579832    0.238 1.793976   114  11.090800
163         {Empire Strikes Back, The (1980),Raiders of the Lost Ark (1981)} => {Star Wars (1977)}   0.252  0.9545455    0.264 1.787538   126  10.252000
172              {Raiders of the Lost Ark (1981),Return of the Jedi (1983)} => {Star Wars (1977)}   0.252  0.9545455    0.264 1.787538   126  10.252000
160                  {Pulp Fiction (1994),Empire Strikes Back, The (1980)} => {Star Wars (1977)}   0.200  0.9523810    0.210 1.783485   100   9.786000
151                      {Godfather, The (1972),Return of the Jedi (1983)} => {Star Wars (1977)}   0.200  0.9433962    0.212 1.766660   100   8.232667
```

4.3 Σύνολο Δεδομένων MovieLens 1M

Για την εξαγωγή των παρακάτω αποτελεσμάτων χρησιμοποιήθηκε το σύνολο δεδομένων MovieLens 1M. Τα δεδομένα συλλέχθηκαν από την ιστοσελίδα MovieLens και περιλαμβάνουν 1000000 βαθμολογίες από 6000 χρήστες για 4000 ταινίες. Οι βαθμολογίες κυμαίνονται στις ακέραιες τιμές από 1 μέχρι 5.

4.3.1 Διερεύνηση παραμέτρων

Προκειμένου να γίνει η σύγκριση των μέτρων αξίας και των μεθόδων αξιολόγησης, είναι απαραίτητο να προσδιοριστούν οι τιμές διαφόρων παραμέτρων που χρησιμοποιούνται από τις συναρτήσεις και τις μεθόδους της βιβλιοθήκης για την εξαγωγή των αποτελεσμάτων, όπως το given, οι τιμές της support και της confidence για την κλήση του αλγορίθμου apriori, καθώς και ποιο θα είναι το κατώφλι για την διαδικασία του binarization του συνόλου δεδομένων.

Σε πρώτο στάδιο, για να μπορέσει η βιβλιοθήκη να διαβάσει ένα σύνολο δεδομένων και να το μετατρέψει στην κλάση που μπορεί να διαχειριστεί, δηλαδή την realRatingMatrix, θα πρέπει αυτό να έχει μία συγκεκριμένη μορφή όπως φαίνεται στην Εικόνα 4.2. Το σύνολο δεδομένων σε αυτή τη μορφή αποτελείται από τρεις στήλες. Δεδομένου ότι η βιβλιοθήκη διαχειρίζεται βαθμολογίες μεταξύ χρήστη και αντικειμένου, η πρώτη στήλη αποτελείται αυστηρά από την ταυτότητα του χρήστη. Η δεύτερη στήλη αποτελεί τα αντικείμενα, και η τρίτη στήλη αποτελεί την βαθμολογία. Μία ταυτότητα ενός χρήστη μπορεί να επαναλαμβάνεται σε άλλες γραμμές, καθώς η κάθε γραμμή αντιπροσωπεύει μία βαθμολογία και ένας χρήστης μπορεί να βαθμολογήσει παραπάνω από μία ταινία.

Κατεβάζοντας τα αρχεία του MovieLens 1M dataset, το αρχείο ratings.csv είναι αυτό που περιέχει τις επιθυμητές πληροφορίες. Διαβάζοντας το αρχείο των βαθμολογιών ως CSV (Comma-Separated Values αρχείο) παρατηρείται πως υπάρχει μία τέταρτη στήλη, η οποία αποτελεί το timestamp της βαθμολογίας. Προκειμένου να μετατραπούν τα δεδομένα σε realRatingMatrix, η συγκεκριμένη στήλη θα αφαιρεθεί.

```
> movieLens1M <-read.csv('ratings.csv', header = FALSE)
> movieLens1M
    V1   V2 V3        V4
1    1 1193  5 978300760
2    1  661  3 978302109
3    1  914  3 978301968
4    1 3408  4 978300275
5    1 2355  5 978824291
6    1 1197  3 978302268
7    1 1287  5 978302039 ...

> movieLens1M <- movieLens1M[,-4]
> movieLens1M
    V1   V2 V3
1    1 1193  5
2    1  661  3
3    1  914  3
4    1 3408  4
5    1 2355  5
6    1 1197  3
7    1 1287  5 ...
```

	V1	V2	V3
1	1	1193	5
2	1	661	3
3	1	914	3
4	1	3408	4
5	1	2355	5
6	1	1197	3
7	1	1287	5
8	1	2804	5
9	1	594	4
10	1	919	4
11	1	595	5
12	1	938	4
13	1	2398	4
14	1	2918	4
15	1	1035	5

Εικόνα 4.2 Μορφή συνόλου δεδομένων για να μετατραπεί σε realRatingMatrix

Εφόσον το σύνολο δεδομένων έχει την επιθυμητή μορφή πλέον, μπορεί να μετατραπεί σε πίνακα realRatingMatrix. Αν ζητηθεί να προβληθούν τα περιεχόμενά του, όπως φαίνεται παρακάτω, παρατηρείται πως η μορφή του πίνακα πλέον είναι στην μορφή ενός realRatingMatrix, με την κάθε γραμμή να αντιστοιχεί σε έναν χρήστη, τις στήλες στις ταινίες και η μεταξύ τους σχέση να είναι οι βαθμολογίες των χρηστών, με την απώλεια βαθμολογίας να ορίζεται ως μία τελεία.

```
> dataset<-as(movieLens1M, "realRatingMatrix")
> dataset
6040 x 3706 rating matrix of class 'realRatingMatrix' with 1000209 ratings.
> getRatingMatrix(dataset[1:10, 1:5])
10 x 5 sparse Matrix of class "dgCMatrix"
   1 2 3 4 5
1  5 . . . .
2  . . . . .
3  . . . . .
4  . . . . .
5  . . . . .
6  4 . . . .
7  . . . . .
8  4 . . 3 .
9  5 . . . .
10 5 5 . . .
```

Φαίνεται πως ο realRatingMatrix αποτελείται από 6040 χρήστες, οι οποίοι θα χρησιμοποιηθούν για την διαδικασία της αξιολόγησης. Προκειμένου όμως να οριστούν τα κατώφλια για την διαδικασία του

binarization, πρέπει να ταυτοποιηθεί ο μέσος όρος βαθμολογιών. Με την συνάρτηση rowMeans() υπολογίζεται ο μέσος όρος βαθμολογιών για κάθε χρήστη, και με την συνάρτηση mean() ο μέσος όρος αυτών.

```
> mean(rowMeans(dataset))
[1] 3.702705
```

Καθώς ο στόχος είναι να προσδιοριστεί για κάθε χρήστη ποιες ταινίες θα του αρέσουν με βάση τις ταινίες που ήδη του αρέσουν, θα δοκιμαστούν περιπτώσεις με βαθμολογίες 3 και 4.

Με την διαδικασία του binarization ο αρχικός πίνακας των βαθμολογιών της μορφης realRatingMatrix μετατρέπεται σε binaryRatingMatrix με κάθε βαθμολογία να έχει μετατραπεί σε τιμή true ή false, με true αυτές που είναι μεγαλύτερες ή ίσες του κατωφλιού που δόθηκε στην παράμετρο minRating, και false όλες τις υπόλοιπες.

Έτσι, από τις αρχικές 1000209 βαθμολογίες του αρχικού βαθμολογικού πίνακα, με binarization με κατώφλι 3 περιορίζονται στις 836478 βαθμολογίες, ενώ με κατώφλι 4 στις 575281 βαθμολογίες.

```
> binarized <- binarize(dataset, minRating=3)
> binarized
6040 x 3706 rating matrix of class 'binaryRatingMatrix' with 836478 ratings.
> binarized <- binarize(dataset, minRating=4)
> binarized
6040 x 3706 rating matrix of class 'binaryRatingMatrix' with 575281 ratings.
```

Για να προσδιοριστεί η τιμή της παραμέτρου given πρέπει να υπολογιστούν πόσες περίπου ταινίες έχουν βαθμολογημένες ως true οι χρήστες.

Ο μέσος όρος είναι περίπου 138 ταινίες με κατώφλι 3, επομένως οι τιμές given που θα δοκιμαστούν θα μπορούσαν να είναι σε ένα διάστημα από -150 μέχρι και 150. Με κατώφλι 4 και μέσο όρο 95 ταινιών περίπου, το given θα μπορούσε να έχει τιμές από –110 μέχρι και 110.

Οι top-N λίστες που θα προταθούν ορίζεται να κυμαίνονται από top-1 μέχρι και top-5.

```
> binarized <- binarize(dataset, minRating=3)
> mean(rowCounts(binarized))
[1] 138.4897
> binarized <- binarize(dataset, minRating=4)
> mean(rowCounts(binarized))
[1] 95.2452
```

Τέλος, δοκιμάστηκαν διάφορες τιμές κατωφλίου για τις παραμέτρους support και confidence.

Για minRating ίσο 3 το κατώφλι για support ορίζεται 0.05, για confidence 0.9 με πλήθος κανόνων περίπου 24628. Για minRating ίσο 4 το κατώφλι για support ορίζεται 0.03, για confidence 0.83 με πλήθος κανόνων περίπου 26464.

Οι περιπτώσεις και τα κατώφλια που προαναφέρθηκαν παρουσιάζονται αναλυτικά στον Πίνακα 4.4.1.

minRating	Μ.Ο. Πλήθους Ταινιών	Εύρος Given	Κατώφλια Apriori
3	138.48	[-150,150]	sup=0.05, conf=0.9
4	95.24	[-110,110]	sup=0.03, conf=0.83

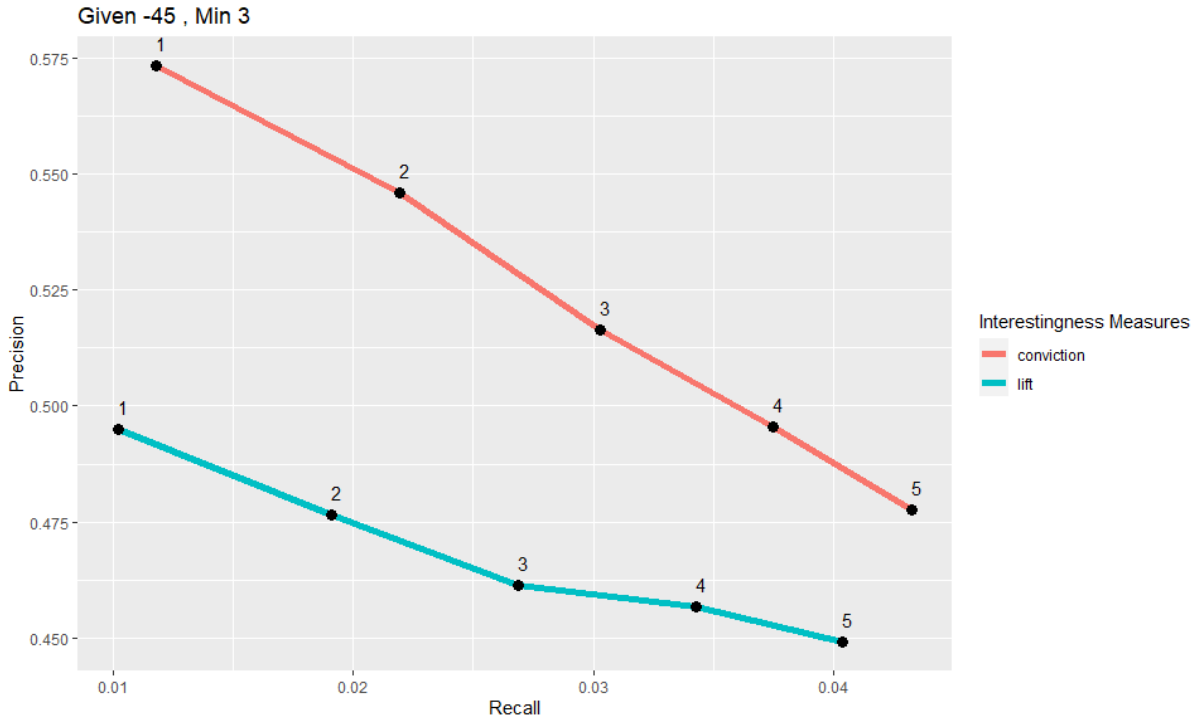
Πίνακας 4.1 Τιμές κατωφλιών για αξιολόγηση των δεδομένων MovieLens 1M

4.3.2 Σύγκριση μέτρων αξίας conviction-lift

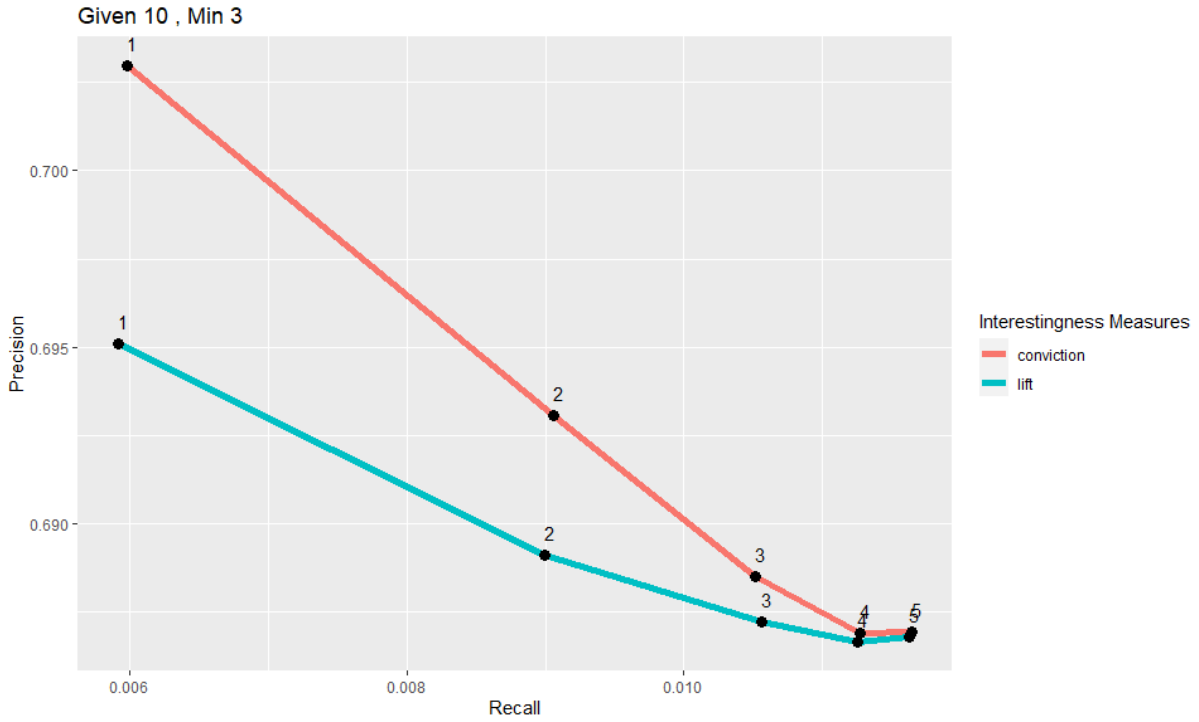
Σε αυτή την Ενότητα παρουσιάζονται τα αποτελέσματα των αξιολογήσεων της παραγωγής συστάσεων με την ταξινόμηση των κανόνων του αλγορίθμου apriori ως προς lift και ως προς conviction για το συνολο δεδομένων των ταινιών MovieLens 1M. Το μοντέλο αξιολόγησης είναι αυτό της cross-validation. Τα γραφήματα δημιουργήθηκαν με την χρήση της βιβλιοθήκης ggplot2, με τον άξονα x να αντιπροσωπεύει τις τιμές recall και τον άξονα y τις τιμές precision των συστάσεων, και δοκιμάζονται όλες οι ακέραιες τιμές binarize από το 3 έως το 4, όπως αναφέρθηκε και στην προηγούμενη Ενότητα.

Στα παρακάτω γραφήματα, η αξιολόγηση έγινε πολλαπλές φορές για διαφορετικές τιμές given, με τις τιμές τους να μεταβάλλεται στα διαστήματα που αναφέρθηκαν στην προηγούμενη Ενότητα, ανάλογα με την τιμή του minRating. Παρουσιάζονται 2 γραφήματα για κάθε minRating, με αρνητική και θετική τιμή given. Σε κάθε ένα από αυτά τα γραφήματα, οι γραμμές των δύο μέτρων αξίας μεταβάλλονται από την αλλαγή της τιμής top-N, που όπως προαναφέρθηκε, παίρνει τιμές από 1 μέχρι και 5. Δηλαδή, έγινε παραγωγή συστάσεων για 1 μέχρι και 5 ταινίες στους χρήστες , και από αυτές τις συστάσεις αξιολογείται κάθε φορά πόσα από τις ταινίες που συστάθηκαν ο χρήστης τα είχε όντως βαθμολογήσει με βαθμό από το minRating και πάνω.

Για την περίπτωση που το minRating είναι 3, δοκιμάστηκαν 18 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-150,150], με όλες από αυτές να παρουσιάζουν τα ίδια αποτελέσματα που συνοψίζονται στις Εικόνες 4.3 και 4.4. Σε κάθε περίπτωση, οι τιμές της conviction υπερείχαν αυτές της lift.



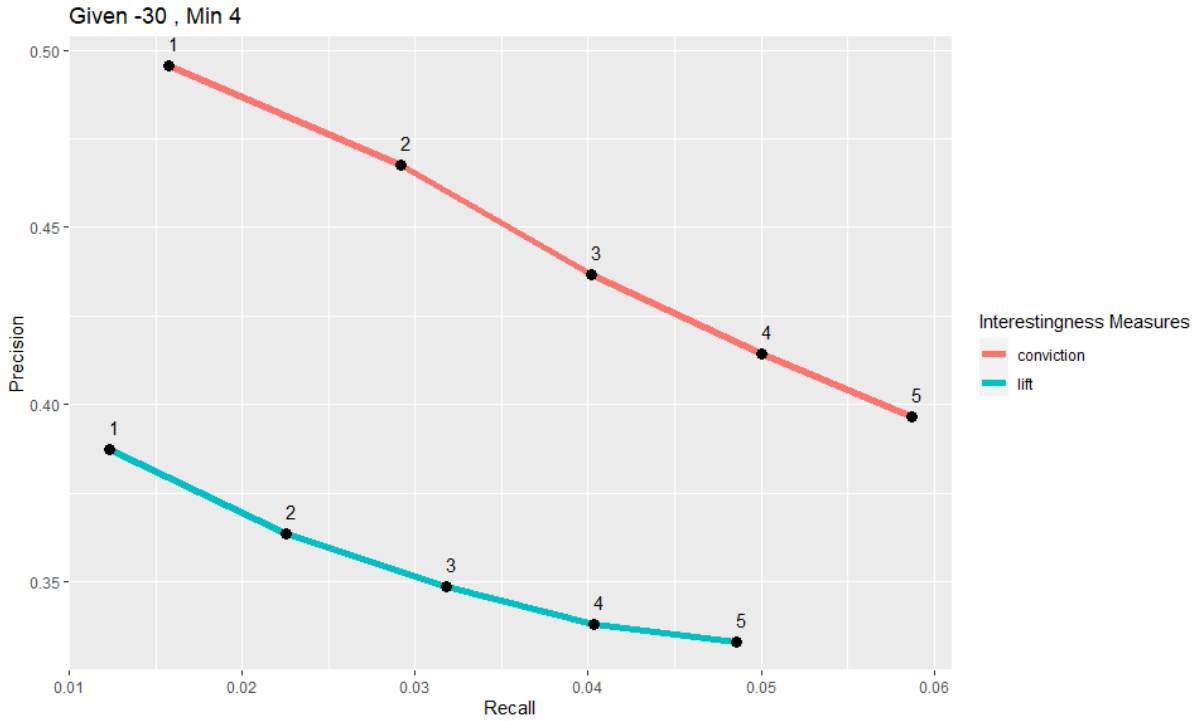
Εικόνα 4.3 MovieLens 1M: τιμές 1-5 για Top-N, given= -45 και minRating= 3



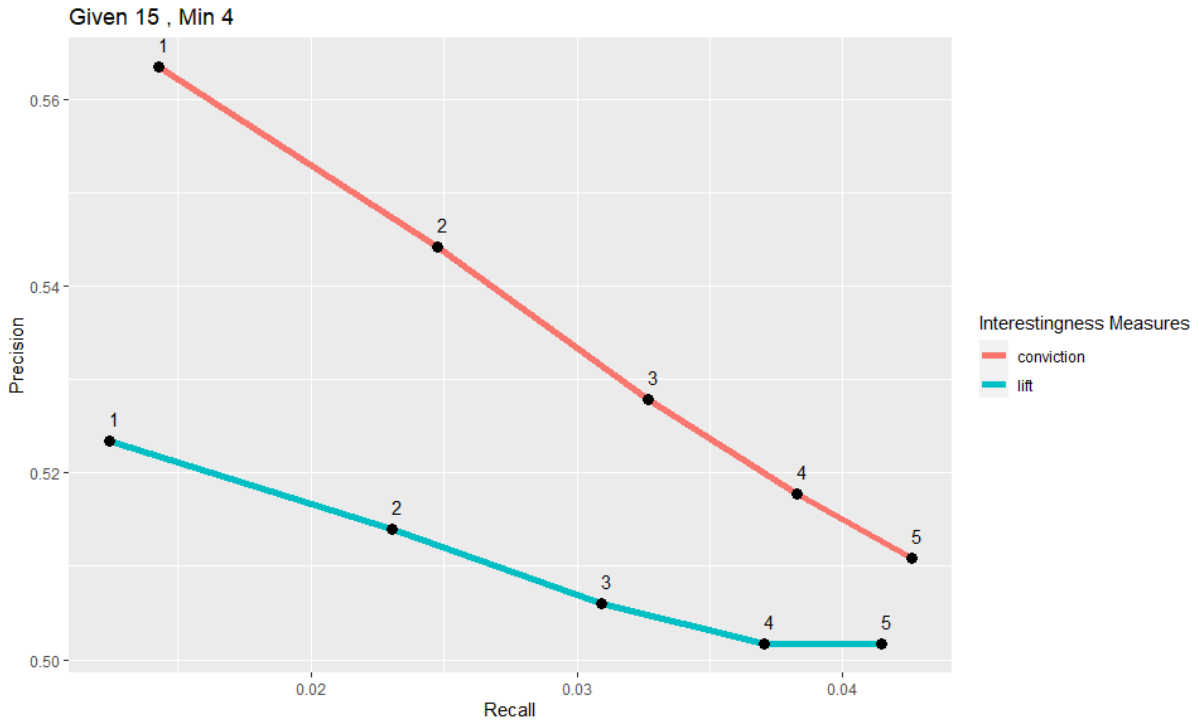
Εικόνα 4.4 MovieLens 1M: τιμές 1-5 για Top-N, given= 10 και minRating= 3

Για την περίπτωση που το minRating είναι 4, δοκιμάστηκαν 16 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-110,110], με όλες από αυτές να παρουσιάζουν τα ίδια αποτελέσματα που

συνοψίζονται στις Εικόνες 4.5 και 4.6. Τα προηγούμενα αποτελέσματα επαναλαμβάνονται, με τις τιμές της conviction να υπερέχουν της lift.



Εικόνα 4.5 MovieLens 1M: τιμές 1-5 για Top-N, given= 30 και minRating= 4



Εικόνα 4.6 MovieLens 1M: τιμές 1-5 για Top-N, given= 15 και minRating= 4

4.4 Βαθμολογικά δεδομένα του τμήματος

Για την εξαγωγή των παρακάτω αποτελεσμάτων χρησιμοποιήθηκε ένα σύνολο με βαθμολογικά δεδομένα του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, φιλτραρισμένο μόνο για τους φοιτητές της Πληροφορικής.

Πιο συγκεκριμένα, το αρχικό σύνολο δεδομένων περιέχει βαθμολογίες για τα έτη από το 2009 έως το 2021, με 522841 εγγραφές (βαθμολογίες), 366 μαθήματα και 7276 φοιτητές. Στο φιλτραρισμένο σύνολο δεδομένων το οποίο χρησιμοποιείται στην συνέχεια της πτυχιακής εργασίας, αποτελείται από 186127 εγγραφές για τα έτη από 2009 μέχρι 2021, με 40886 από αυτές τις βαθμολογίες να είναι με προβιβάσιμο βαθμό εξετάσεων, 79 μαθήματα και 2687 φοιτητές. Οι βαθμολογίες κυμαίνονται στο εύρος τιμών από 0 μέχρι και 10.

4.4.1 Διερεύνηση παραμέτρων

Όπως αναφέρθηκε και στην Ενότητα 4.3.1, προκειμένου να γίνει η σύγκριση των μέτρων αξίας και των μεθόδων αξιολόγησης, είναι απαραίτητο να προσδιοριστούν οι τιμές διαφόρων παραμέτρων που χρησιμοποιούνται από τις συναρτήσεις και τις μεθόδους της βιβλιοθήκης για την εξαγωγή των αποτελεσμάτων, όπως το given, οι τιμές της support και της confidence για την κλήση του αλγορίθμου apriori, καθώς και ποιο θα είναι το κατώφλι για την διαδικασία του binarization του συνόλου δεδομένων.

Η μορφή του συνόλου δεδομένων προκειμένου να μετατραπεί σε μορφή realRatingMatrix, θα πρέπει να αποτελείται από τρεις στήλες: η πρώτη να περιέχει την ταυτότητα των φοιτητών, η δεύτερη την ταυτότητα των μαθημάτων και η τρίτη την αντίστοιχη βαθμολογία που δόθηκε.

Εφόσον το σύνολο δεδομένων έχει αυτή τη μορφή, μπορεί να διαβαστεί ως CSV μορφή και στην συνέχεια να μετατραπεί σε realRatingMatrix μορφή. Αν ζητηθεί να προβληθούν τα περιεχόμενά του, όπως φαίνεται παρακάτω, παρατηρείται πως η μορφή του πίνακα πλέον είναι στην μορφή ενός realRatingMatrix, με τις γραμμές να αντιστοιχούν σε έναν φοιτητή, τις στήλες στα μαθήματα και η μεταξύ τους σχέση να είναι οι βαθμολογίες των φοιτητών, με την απώλεια βαθμολογίας να ορίζεται ως μία τελεία.

```
> subjectsDataset<-read.csv('subjects.csv')
> dataset<-as(subjectsDataset, "realRatingMatrix")
> getRatingMatrix(dataset[1:10, 1:5])
10 x 5 sparse Matrix of class "dgCMatrix"
                                    1625-1101 1625-1102 1625-1103 1625-1104 1625-1105
00216EE1-6090-48C3-9BAC-F9E84239D44B       5.0       7.5       5.0         .         .
0024E3EA-8088-40EE-BD6F-698008867449       .         .         8.5         .         .
0029318C-2F8B-485A-AC44-A4B44C45E010       6.0       7.0       6.0         .         .
00351CED-2D99-465D-84A6-56C4ABFC753E       2.0       3.2       3.5         .         .
0063EB35-5AC8-4EB3-B7F5-236A83FCF64A       8.5       7.0      10.0         .         .
0079C8F8-6B1B-412C-BB0C-1C428BE736BA       6.5       .         .           .         .
00A770D5-452C-48B2-81C5-E87A8F72238F       6.5      16.1       5.0         .         .
00A9C844-0371-4689-A016-D17A08AECB65       9.0       6.6       6.0         .         2
00AF1FB3-1EC7-44E3-AE14-C859ADA0B5A7       .         .         .           .         .
00D6B80E-CECE-4563-AE06-CFBF4E15E4DA       6.5       7.0       .           .         .
```

```
> dataset
2687 x 80 rating matrix of class ‘realRatingMatrix’ with 47935 ratings.
```

Φαίνεται πως ο realRatingMatrix αποτελείται από 2687 φοιτητές, οι οποίοι θα χρησιμοποιηθούν για την διαδικασία της αξιολόγησης. Προκειμένου όμως να οριστούν τα κατώφλια για την διαδικασία του binarization, πρέπει να προσδιοριστεί ο μέσος όρος βαθμολογιών. Με την συνάρτηση rowMeans() υπολογίζεται ο μέσος όρος βαθμολογιών για κάθε φοιτητή, και με την συνάρτηση mean() ο μέσος όρος αυτών.

```
> mean(rowMeans(dataset))
[1] 6.083009
```

Καθώς ο στόχος είναι να προσδιοριστεί για κάθε φοιτητή σε ποια μαθήματα θα τα πάει καλά με βάση σε ποια μαθήματα τα έχει πάει ήδη καλά, θα πρέπει να δοκιμαστούν περιπτώσεις με βαθμολογίες όπως 6, 7, 8 και 9.

Με την διαδικασία του binarization ο αρχικός πίνακας των βαθμολογιών της μορφης realRatingMatrix μετατρέπεται σε binaryRatingMatrix με κάθε βαθμολογία να έχει μετατραπεί σε τιμή true ή false, με true αυτές που είναι μεγαλύτερες ή ίσες του κατωφλιού που δόθηκε στην παράμετρο minRating, και false όλες τις υπόλοιπες.

Έτσι, από τις αρχικές 47935 βαθμολογίες του αρχικού βαθμολογικού πίνακα, με binarization με κατώφλι 6 περιορίζονται στις 29264 βαθμολογίες, με κατώφλι 7 στις 21874, με κατώφλι 8 στις 15566 και με κατώφλι 9 στις 10314. Παρατηρείται ότι από το κατώφλι 6 έως το 9 οι διαθέσιμες βαθμολογίες μειώνονται παραπάνω από τις μισές.

```
> binarized <- binarize(dataset, minRating=6)
> binarized
2687 x 80 rating matrix of class 'binaryRatingMatrix' with 29264 ratings.
> binarized <- binarize(dataset, minRating=7)
> binarized
2687 x 80 rating matrix of class 'binaryRatingMatrix' with 21874 ratings.
> binarized <- binarize(dataset, minRating=8)
> binarized
2687 x 80 rating matrix of class 'binaryRatingMatrix' with 15566 ratings.
> binarized <- binarize(dataset, minRating=9)
> binarized
2687 x 80 rating matrix of class 'binaryRatingMatrix' with 10314 ratings.
```

Για να προσδιοριστεί η τιμή της παραμέτρου given πρέπει να υπολογιστούν πόσα περίπου μαθήματα έχουν βαθμολογημένα ως true οι φοιτητές. Αυτό υπολογίζεται με την συνάρτηση rowCounts() που αθροίζει τις true τιμές του κάθε φοιτητή και με την συνάρτηση mean() υπολογίζεται ο μέσος όρος αυτών.

Ο μέσος όρος με κατώφλι 6 είναι περίπου 10 μαθήματα και οι τιμές given που θα δοκιμαστούν θα μπορούσαν να είναι σε ένα διάστημα από -15 μέχρι και 15. Με κατώφλι 7 και μέσο όρο 8 μαθημάτων περίπου, το given θα μπορούσε να έχει τιμές από -12 μέχρι και 12, με κατώφλι 8 και μέσο όρο μαθημάτων περίπου στο 6 οι τιμές given θα μπορούσαν να είναι από -10 μέχρι και 10, και τέλος με κατώφλι 9 και μέσο όρο περίπου 4 μαθημάτων, οι τιμές given ορίζονται στο διάστημα από -8 μέχρι 8.

Το μεγάλο εύρος τιμών given αποσκοπεί στην διερεύνηση της μεταβολής της απόδοσης του αλγορίθμου ανάλογα με την αλλαγή των τιμών αυτών.

Οι top-N-Lists, με βάση αυτους τους μέσους όρους, ορίζονται για top-1 μέχρι και top5. Είναι αναμενόμενο ότι σε top-N γραφήματα το min rating ίσο με 6 θα τα πάει καλύτερα, καθώς οι φοιτητές έχουν παρακολουθήσει περισσότερα μαθήματα κατά μέσο όρο και επομένως είναι πιο πιθανό να τα πετύχει ο αλγόριθμος στις προβλέψεις.

```
> binarized <- binarize(dataset, minRating=6)
> mean(rowCounts(binarized))
[1] 10.89096
> binarized <- binarize(dataset, minRating=7)
> mean(rowCounts(binarized))
[1] 8.140677
> binarized <- binarize(dataset, minRating=8)
> mean(rowCounts(binarized))
[1] 5.793078
> binarized <- binarize(dataset, minRating=9)
> mean(rowCounts(binarized))
[1] 3.838482
```

Τέλος, δοκιμάστηκαν διάφορες τιμές κατωφλίου για τις παραμέτρους support και confidence ώστε να είναι ικανοποιητικός ο αριθμός των διαθέσιμων κανόνων σε κάθε περίπτωση, αλλά ταυτόχρονα δόθηκε βάση στην υψηλή τιμή της confidence καθώς στόχος είναι οι κεφαλές και τα σώματα των κανόνων να έχουν μεγάλη εξάρτηση μεταξύ τους.

Για minRating ίσο 6 το κατώφλι για support ορίζεται 0.01, για confidence 0.85 με πλήθος κανόνων περίπου 695. Για minRating ίσο 7 το κατώφλι για support ορίζεται 0.01, για confidence 0.73 με πλήθος κανόνων περίπου 471. Για minRating ίσο 8 το κατώφλι για support ορίζεται 0.01, για confidence 0.65 με πλήθος κανόνων περίπου 349. Τέλος, για minRating ίσο με 9 το κατώφλι για support ορίζεται στην τιμή 0.01 ενώ για confidence στην 0.55 με αριθμό κανόνων περίπου 207.

Οι περιπτώσεις και τα κατώφλια που προαναφέρθηκαν παρουσιάζονται αναλυτικά στον Πίνακα 4.3.1.

minRating	Μ.Ο. Πλήθους Μαθημάτων	Εύρος Given	Κατώφλια Apriori
6	10.8	[-15,15]	sup=0.01, conf=0.85
7	8.14	[-12,12]	sup=0.01, conf=0.73
8	5.79	[-10,10]	sup=0.01, conf=0.65
9	3.38	[-8,8]	sup=0.01, conf=0.55

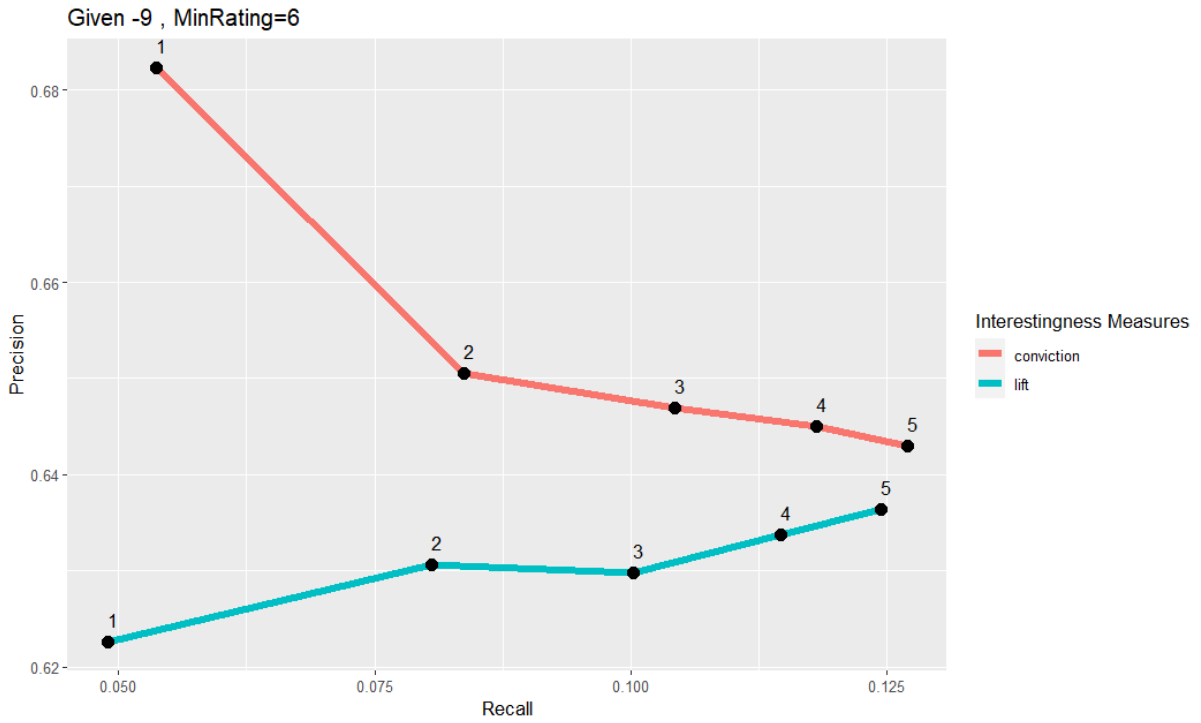
Πίνακας 4.2 Τιμές κατωφλιών για αξιολόγηση των βαθμολογικών δεδομένων

4.4.2 Σύγκριση μέτρων αξίας conviction-lift

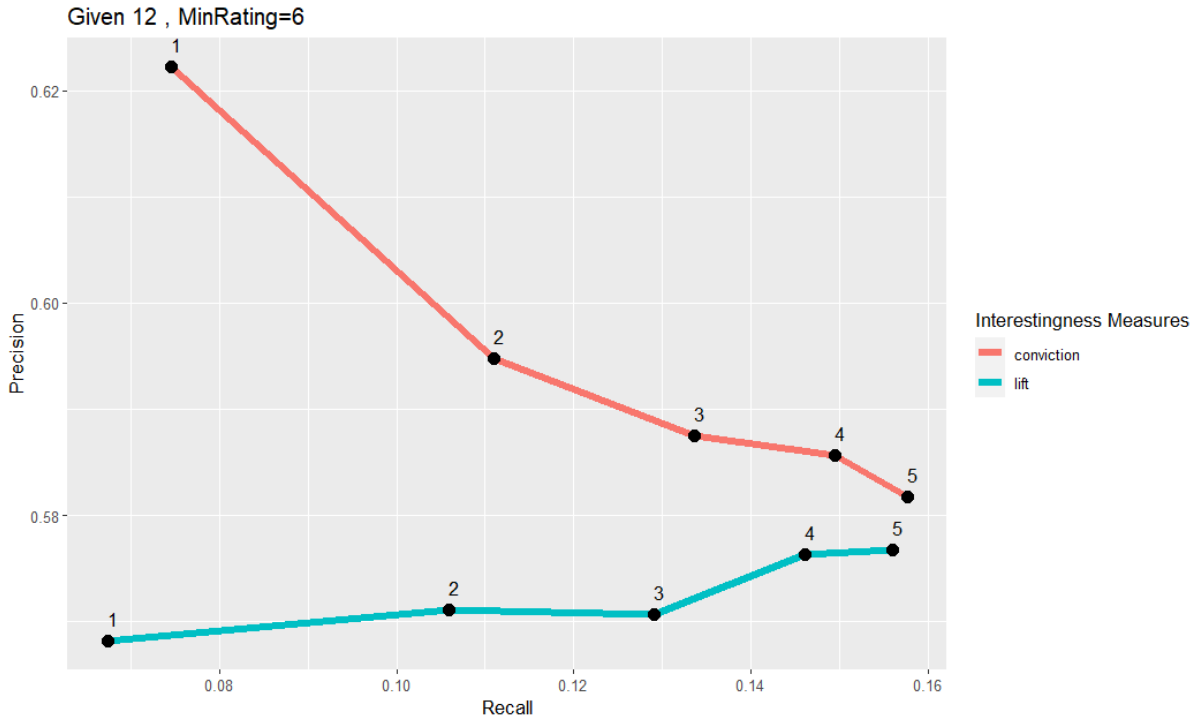
Σε αυτή την Ενότητα παρουσιάζονται τα αποτελέσματα των αξιολογήσεων της παραγωγής συστάσεων με την ταξινόμηση των κανόνων του αλγορίθμου apriori ως προς lift και ως προς conviction, για το συνολο των βαθμολογικών δεδομένων του τμήματος Το μοντέλο αξιολόγησης που χρησιμοποιείται είναι αυτό της cross-validation καθώς εγγυάται την χρήση όλων των συναλλαγών ως test sets από μία φορά, και τα folds ορίζονται στα 5. Τα γραφήματα δημιουργήθηκαν με την χρήση της βιβλιοθήκης ggplot2, με τον άξονα x να αντιπροσωπεύει τις τιμές recall και τον άξονα y τις τιμές precision των συστάσεων, και δοκιμάζονται όλες οι ακέραιες τιμές binarize από το 6 έως το 9, όπως αναφέρθηκε και στην προηγούμενη Ενότητα.

Στα παρακάτω γραφήματα, η αξιολόγηση έγινε πολλαπλές φορές για διαφορετικές τιμές given, με τις τιμές τους να μεταβάλλεται στα διαστήματα που αναφέρθηκαν στην προηγούμενη Ενότητα, ανάλογα με την τιμή του minRating. Παρουσιάζονται 2 γραφήματα για κάθε minRating, με αρνητική και θετική τιμή given. Σε κάθε ένα από αυτά τα γραφήματα, οι γραμμές των δύο μέτρων αξίας μεταβάλλονται από την αλλαγή της τιμής top-N, που όπως προαναφέρθηκε, παίρνει τιμές από 1 μέχρι και 5. Δηλαδή, έγινε παραγωγή συστάσεων για 1 μέχρι και 5 μαθήματα στους φοιτητές, και από αυτές τις συστάσεις αξιολογείται κάθε φορά πόσα από τα μαθήματα που συστάθηκαν ο φοιτητής τα είχε όντως περάσει με βαθμό από το minRating και πάνω.

Για την περίπτωση που το minRating είναι 6, δοκιμάστηκαν 14 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-15,15], με όλες από αυτές να παρουσιάζουν τα ίδια αποτελέσματα που συνοψίζονται στις Εικόνες 4.7 και 4.8. Σε κάθε περίπτωση, οι τιμές της conviction υπερείχαν αυτές της lift.

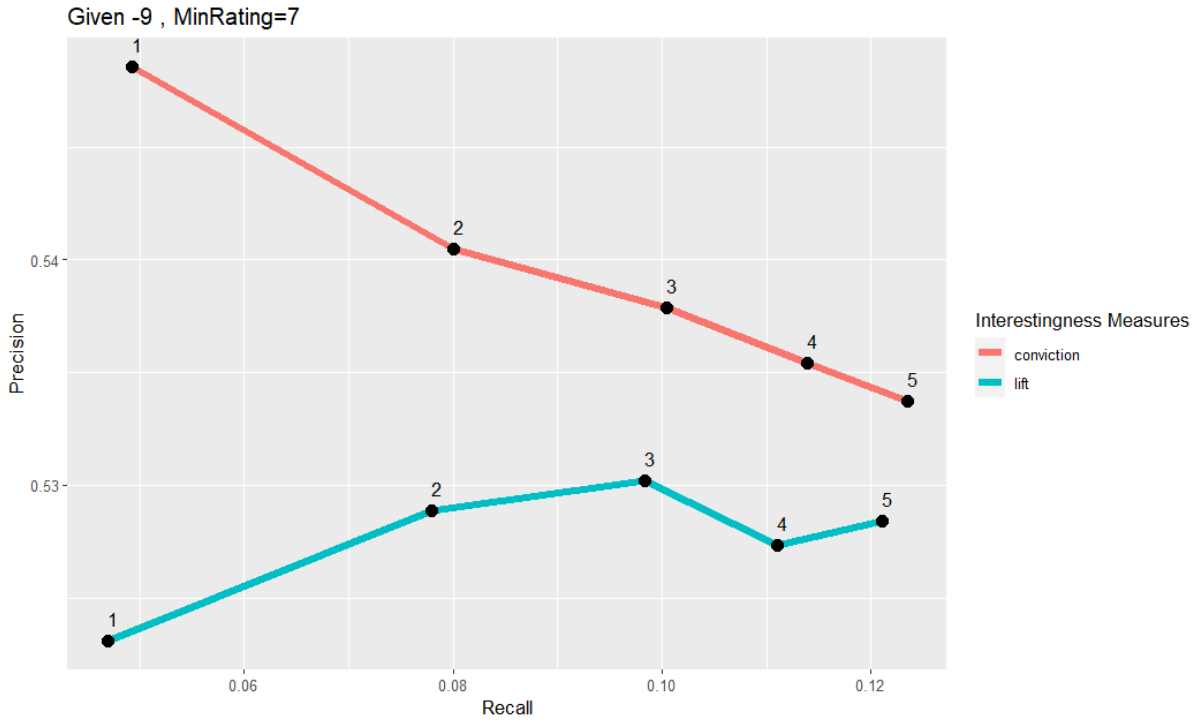


Εικόνα 4.7 Βαθμολογικά Δεδομένα: τιμές 1-5 για Top-N, given= -9 και minRating= 6

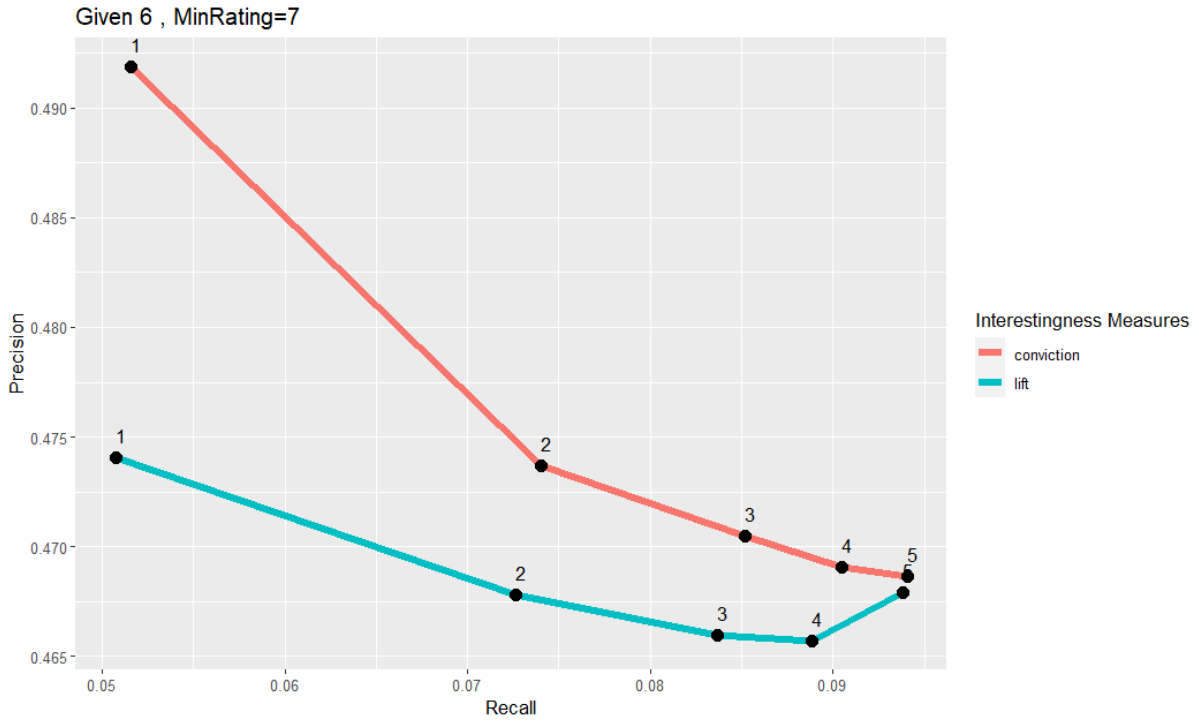


Εικόνα 4.8 Βαθμολογικά Δεδομένα: τιμές 1-5 για Top-N, given= 12 και minRating= 6

Για την περίπτωση που το minRating είναι 7, δοκιμάστηκαν 12 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-12,12], με όλες από αυτές να παρουσιάζουν τα ίδια αποτελέσματα που συνοψίζονται στις Εικόνες 4.9 και 4.10. Τα προηγούμενα αποτελέσματα επαναλαμβάνονται, με τις τιμές της conviction να υπερέχουν της lift.

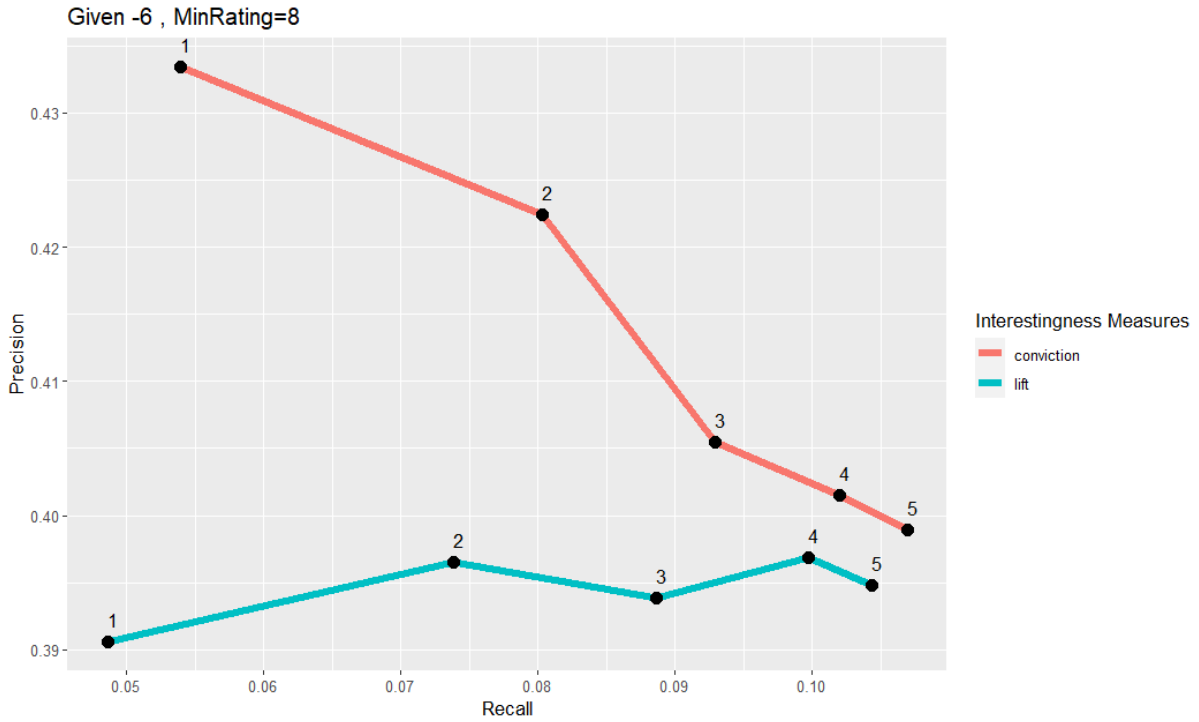


Εικόνα 4.9 Βαθμολογικά Δεδομένα: τιμές 1-5 για Top-N, given= -9 και minRating= 7

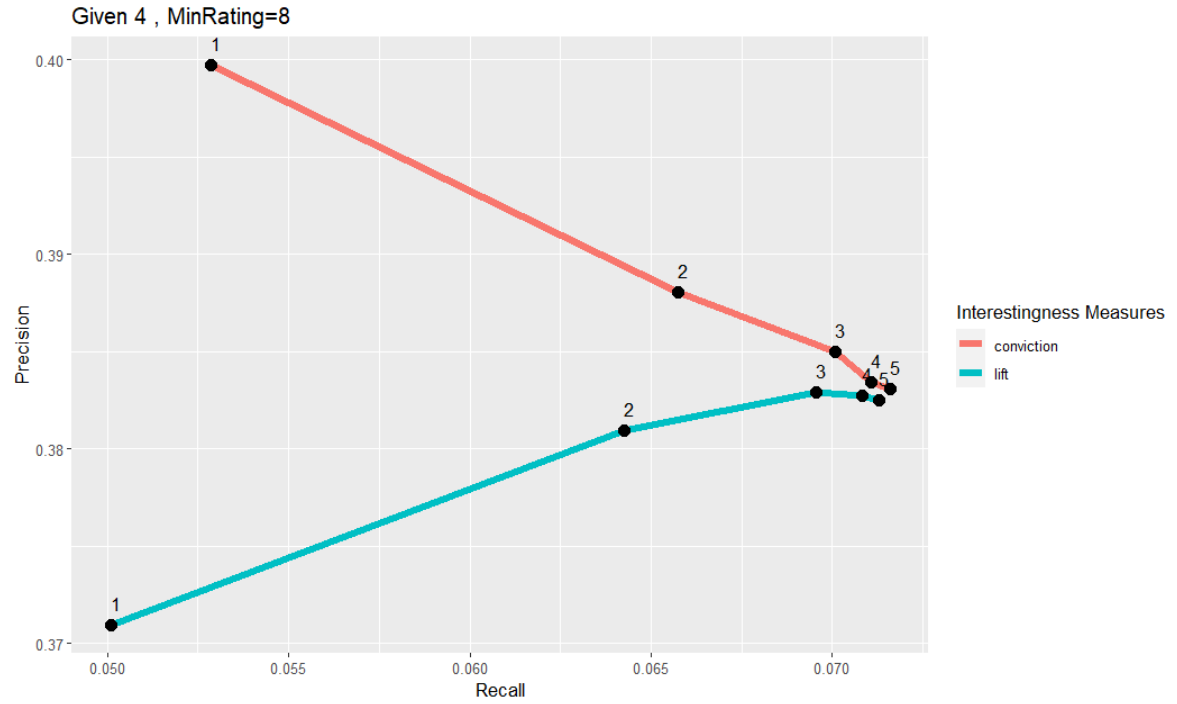


Εικόνα 4.10 Βαθμολογικά Δεδομένα: τιμές 1-5 για Top-N, given= 6 και minRating= 7

Για την περίπτωση που το minRating είναι 8, δοκιμάστηκαν 10 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-10,10], με όλες από αυτές να παρουσιάζουν τα ίδια αποτελέσματα που συνοψίζονται στις Εικόνες 4.11 και 4.12. Τα προηγούμενα αποτελέσματα επαναλαμβάνονται, με τις τιμές της conviction να υπερέχουν της lift.

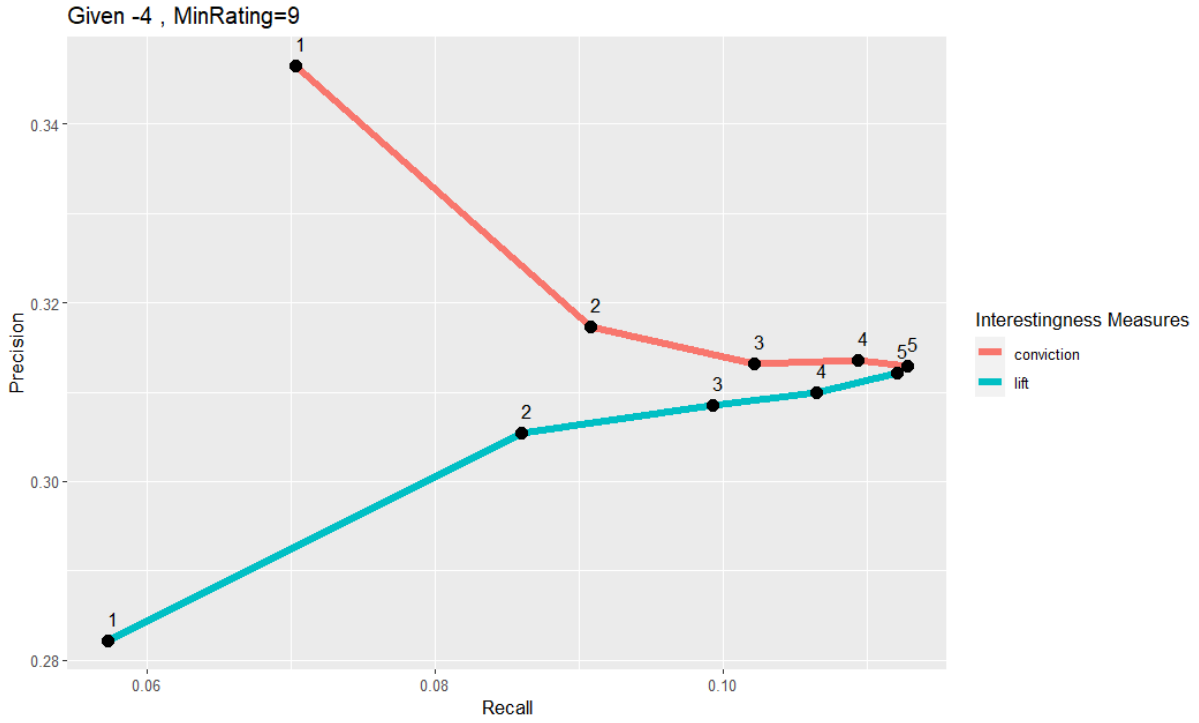


Εικόνα 4.11 Βαθμολογικά Δεδομένα: τιμές 1-5 για Top-N, given= -6 και minRating= 8

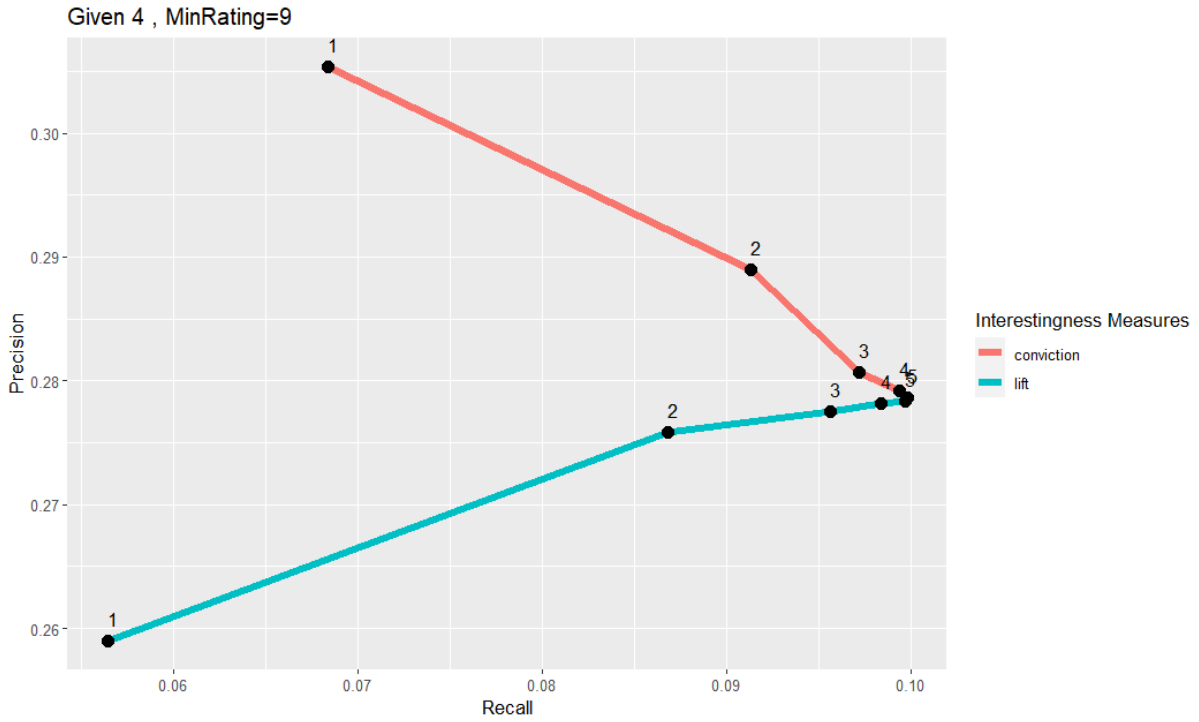


Εικόνα 4.12 Βαθμολογικά Δεδομένα: τιμές 1-5 για Top-N, given= 4 και minRating= 8

Τέλος, για την περίπτωση που το minRating είναι 9, δοκιμάστηκαν 8 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-8,8], με όλες από αυτές να παρουσιάζουν τα ίδια αποτελέσματα που συνοψίζονται στις Εικόνες 4.13 και 4.14. Τα προηγούμενα αποτελέσματα επαναλαμβάνονται, με τις τιμές της conviction να υπερέχουν της lift.



Εικόνα 4.13 Βαθμολογικά Δεδομένα: τιμές 1-5 για Top-N, given= -4 και minRating= 9



Εικόνα 4.14 Βαθμολογικά Δεδομένα: τιμές 1-5 για Top-N, given= 4 και minRating= 9

4.5 Συμπεράσματα

Συνοψίζοντας, σε όλες τις περιπτώσεις που δοκιμάστηκαν και στα δύο σύνολα δεδομένων, το μέτρο αξίας conviction φαίνεται να υπερέχει του lift. Δηλαδή, για κάθε top-N συστάσεις που προέκυψαν με ταξινόμηση των κανόνων ως προς conviction είχαν μεγαλύτερες τιμές precision και recall έναντι αυτών του lift. Όπως αναφέρθηκε και στην Ενότητα 2.4.1, αυτό ήταν αναμενόμενο, καθώς για την lift προκύπτει από τον τύπο της ότι η τιμή της για έναν κανόνα X⇒Y ισχύει και για τον κανόνα Y⇒X και επομένως δεν δηλώνει την κατεύθυνση με την οποία το ένα στοιχειοσύνολο επηρεάζει την εμφάνιση του άλλου. Αντιθέτως, η conviction δεν διαθέτει την αντιμεταθετική ικανότητα, δηλαδή την ενδιαφέρει η κατεύθυνση του κανόνα και αν είναι το σώμα που προκαλεί την εμφάνιση της κεφαλής ή το αντίθετο. Επομένως, ήταν αναμενόμενο να έχει μεγαλύτερες τιμές precision και recall.

Επιπλέον, παρατηρείται πως η άνοδος στην τιμή του minRating μειώνει τις τιμές precision και recall, καθώς μειώνεται σημαντικά ο αριθμός των διαθέσιμων δεδομένων.

4.6 Επίλογος

Μετά από μερικές τροποποιήσεις στο αρχείο RECOM_AR.R σε ένα ξεχωριστό αρχείο AR_MOD.R, το μέτρο αξίας conviction έγινε διαθέσιμο για να χρησιμοποιηθεί στην ταξινόμηση των Association Rules που παρήγαγε ο apriori. Για δύο διαφορετικά σύνολα δεδομένων, αυτό των βαθμολογικών δεδομένων του Τμήματος και αυτό των ταινιών MovieLens 1M, καθορίστηκαν οι διάφορες τιμές των παραμέτρων που χρησιμοποιήθηκαν στην συνέχεια και δημιουργήθηκαν γραφήματα με την χρήση της βιβλιοθήκης ggplot2 που αναπαριστούν τα αποτελέσματα των αξιολογήσεων για παραγωγή

συστάσεων με Association Rules και με ταξινόμηση των κανόνων ως προς conviction και ως προς lift, με το conviction να υπερέχει συντριπτικά του lift.

Κεφάλαιο 5ο : Επιπλέον μέτρα αξίας

5.1 Εισαγωγή

Σε αυτό το Κεφάλαιο παρουσιάζεται η προσωπική συνεισφορά στον κώδικα της βιβλιοθήκης RecommenderLab για την υλοποίηση των επιπλέον μέτρων αξίας που θα διερευνηθούν στην συνέχεια, βασισμένα στους συντελεστές συσχέτισης, pearson correlation coefficient και spearman correlation coefficient. Παράγονται συστάσεις για ταξινόμηση των κανόνων που παρήγαγε ο αλγόριθμος apriori με τα μέτρα αξίας conviction, το οποίο φανερά υπερέχει στα γραφήματα του Κεφαλαίου 4, pearson και spearman και εκτελείται η αξιολόγηση με το μοντέλο αξιολόγησης cross validation με 5 folds για διαφορετικές top-N-Lists, min rating και τιμές given. Γίνεται η προσθήκη μιας ακόμα παραμέτρου, ενός φίλτρου για την βελτίωση της απόδοσης των μέτρων αξίας που χρησιμοποιούν αριθμητικά δεδομένα. Τα αποτελέσματα της αξιολόγησης παρουσιάζονται σε γραφήματα με την χρήσης της βιβλιοθήκης ggplot2 για precision και recall.

5.2 Συντελεστές Συσχέτισης

Το μέτρο αξίας που θα διερευνηθεί στην παρούσα πτυχιακή εργασία βασίζεται στην συσχέτιση (correlation) μεταξύ δύο μεταβλητών. Είναι μία μέθοδος που ταυτοποιεί αν οι αλλαγές σε μία μεταβλητή προκαλούν αλλαγές σε μία άλλη μεταβλητή. Δηλώνει αν υπάρχει μία απλή σχέση μεταξύ των δύο μεταβλητών, την κατεύθυνση αυτής της σχέσης (αν είναι θετική ή αρνητική) και την δύναμη αυτής.

Η διερευνηση του ως μέτρο αξίας οφείλεται στην ικανότητά του να υπολογίσει την συσχέτιση μεταξύ του σώματος και της κεφαλής ενός κανόνα, χρησιμοποιώντας όμως αριθμητικές τιμές, πληροφορία που χάνεται κατά την διαδικασία της μετατροπής των δεδομένων σε δυαδική μορφή για την εισαγωγή των δεδομένων στον apriori ως transactions κατηγορικού τύπου. Αποσκοπεί στην διερεύνηση της αξιοποίησης των Association Rules, που εκφύσεων διαχειρίζονται δυαδικά δεδομένα, να χρησιμοποιηθούν και σε προβλήματα με αριθμητικά δεδομένα μέσα από τους συντελεστές συσχέτισης.

Υπάρχουν δύο κύριες τεχνικές που χρησιμοποιούνται για εκτέλεση της συσχετιστικής ανάλυσης, η pearson correlation coefficient και η spearman correlation coefficient.

Η pearson correlation coefficient αποτελεί την πιο διαδεδομένη τεχνική που χρησιμοποιείται ευρέως στην αγορά και στην έρευνα. Είναι μία μετρική της δύναμης μιας γραμμικής σχέσης μεταξύ δύο μεταβλητών για να ταυτοποιηθεί αν υπάρχει δυνατή συσχέτιση μεταξύ τους, όπως αναφέρθηκε πιο αναλυτικά στην Ενότητα 2.4.1.

Η spearman correlation coefficient είναι επίσης μία μετρική στατιστικής που υπολογίζει την δύναμη μιας μονοτονικής σχέσης μεταξύ δύο δεδομένων, όπως αναφέρθηκε πιο αναλυτικά στην Ενότητα 2.4.2.

Προκειμένου να μπορέσουν να χρησιμοποιηθούν αυτές οι δύο τεχνητικές να χρησιμοποιηθούν από την βιβλιοθήκη Recommenderlab κατά την διαδικασία της αξιολόγησης, έπρεπε να τροποποιηθεί το

αρχείο που καλεί τους συνδιαστικούς κανόνες, που όπως αναφέρθηκε στην Ενότητα 5.2 για τους σκοπούς της πτυχιακής εργασίας οι τροποποιήσεις γίνονται στο αρχείο AR_MOD.

Αρχικά, προκειμένου το σώμα και η κεφαλή ενός κανόνα να αποτελέσουν δύο διανύσματα τιμών, θα πρέπει το αρχείο AR_MOD.R να έχει πρόσβαση στον αρχικό realRatingMatrix που έχει δημιουργηθεί και έπειτα να τον μετατρέψει σε matrix για την καλύτερη διαχείριση του. Έτσι, αφού διαβαστεί ένα σύνολο δεδομένων και αφού μορφοποιηθεί, αν χρειαστεί, στην απαραίτητη μορφή για να μετατραπεί σε realRatingMatrix, αποθηκεύεται στην μεταβλητή dataset. Με την χρήση του αρχικού πίνακα βαθμολογιών, το κάθε αντικείμενο μπορεί να προσδιοριστεί ως το διάνυσμα των βαθμολογιών όλων των χρηστών του συνόλου δεδομένων. Επομένως, πριν ακόμα γίνει η κλήση της συνάρτησης apriori(), προστίθεται ο κώδικας που διαβάζει το σύνολο των δεδομένων όπως φαίνεται στην Εικόνα 5.1.

```
18  BIN_AR_MOD <- function(data, parameter = NULL) {
19
20    matrix <- as(dataset, "matrix")
21
```

Εικόνα 5.1 Εισαγωγή αρχικού αριθμητικού συνόλου δεδομένων στο αρχείο AR_MOD

Έπειτα, μετά την κλήση την συνάρτησης apriori(), τα αποτελέσματα αποθηκεύονται στην μεταβλητή rule_base, η οποία περιέχει τους κανόνες μαζί με διάφορες άλλες τιμές, όπως φάνηκε στην Ενότητα 3.4.3, μεταξύ αυτών και οι τιμές support και confidence που αντιστοιχούν σε κάθε κανόνα. Παρατηρείται όμως πως οι κανόνες βρίσκονται σε μία στήλη με το όνομα rules, επομένως θα πρέπει να διασπαστούν το σώμα και η κεφαλή του κάθε κανόνα προκειμένου να εντοπιστούν στον αρχικό πίνακα των βαθμολογιών.

Επομένως, μετά την αποθήκευση των κανόνων στην μεταβλητή rule_base, ακολουθεί ο διαχωρισμός του σώματος και της κεφαλής των κανόνων με τον κώδικα με τον οποίο προόριζε ο Michael Hahsler, δημιουργός της βιβλιοθήκης RecommenderLab και Arules, να γίνεται αυτή η διαδικασία, όπως φαίνεται στην Εικόνα 5.2.

```
34  rule_base <-  suppressWarnings(
35    apriori(data,
36            parameter=list(support=p$support, confidence=p$confidence,
37                           minlen=2, maxlen=p$maxlen, maxtime=p$maxtime), control=p$apriori_control)
38  )
39
40
41  frame <- data.frame(lhs = labels(lhs(rule_base), setStart = "", setEnd = ""),
42            rhs = labels(rhs(rule_base), setStart = "", setEnd = ""))
```

Εικόνα 5.2 Διαχωρισμός σώματος και κεφαλής των κανόνων

Ο διαχωρισμός γίνεται σε μία δομή dataframe, με την πρώτη στήλη να ονομάζεται lhs και να περιέχει το αριστερό μέρος των κανόνων, και στην στήλη με όνομα rhs την κεφαλή του κάθε κανόνα. Τα αποτελέσματα αποθηκεύονται στην μεταβλητή frame, τα περιεχόμενα της οποίας φαίνονται στην Εικόνα 5.3. Φαίνεται ότι η στήλη rhs περιέχει πάντα μόνο ένα μάθημα καθώς είναι η κεφαλή του κανόνα, ενώ η στήλη lhs μπορεί να περιέχει και δύο μαθήματα, ανάλογα με το πόσα βρίσκονταν στο σώμα του κανόνα.

	lhs	rhs
1	1625-17422,1625-1945	1625-1742
2	1625-17422,1625-1942	1625-1742
3	1625-1401,1625-17422	1625-1742
4	1625-17422,1625-1801	1625-1742
5	1625-1503,1625-17422	1625-1742
6	1625-1601,1625-17422	1625-1742
7	1625-1305,1625-17422	1625-1742
8	1625-1301,1625-17422	1625-1742
9	1625-1641,1625-17422	1625-1742
10	1625-17422,1625-1941	1625-1742
11	1625-17422,1625-1743	1625-1742
12	1625-1701,1625-17422	1625-1742
13	1625-1405,1625-17422	1625-1742
14	1625-1102,1625-17422	1625-1742
15	1625-1101,1625-17422	1625-1742
16	1625-1302,1625-17422	1625-1742

Εικόνα 5.3 Περιεχόμενα της μεταβλητής frame

Στην συνέχεια δημιουργείται μία κενή λίστα με όνομα similarities() η οποία σε κάθε υπολογισμό συσχέτισης θα αποθηκεύει την ανάλογη τιμή για κάθε κανόνα, και στο τέλος θα αποτελέσει την στήλη που θα προστεθεί στην μεταβλητή rule_base για να γίνει η ταξινόμηση των κανόνων με βάση αυτή.

Έπειτα, καθώς στην κλήση της συνάρτησης cor() που θα χρησιμοποιηθεί στην συνέχεια για τον υπολογισμό της συσχέτισης μεταξύ των δύο διανυσμάτων θα πρέπει να δοθεί ως παράμετρος ποια τεχνική θα χρησιμοποιηθεί, ανάλογα με το ποιό sort_measure έχει οριστεί κατά την κλήση της παρούσας συνάρτησης, σε περίπτωση που η τιμή του είναι “pearson” ή “spearman”, η μεταβλητή similarityName θα έχει το ανάλογο περιεχόμενο, όπως φαίνεται στην Εικόνα 5.4.

```
63   similarities <- list()
64   if(p$sort_measure == "pearson"){
65     similarityName <- "pearson"
66   }
67   if(p$sort_measure == "spearman"){
68     similarityName <- "spearman"
69   }
```

Εικόνα 5.4 Δημιουργία κενής λίστας και ορισμός συντελεστή συσχέτισης

Στην συνέχεια, μόνο στην περίπτωση που ως μέθοδος ταξινόμησης έχει δοθεί μία από τις δύο τιμές συσχέτισης, είναι επιθυμητό να ακολουθηθεί η παρακάτω διαδικασία:

Για κάθε κανόνα της μεταβλητής frame, δημιουργείται μία μεταβλητή lhs_items η οποία μετατρέποντας σε string τα περιεχόμενα της lhs στήλης της μεταβλητής frame για τον συγκεκριμένο κανόνα, διαχωρίζει τα περιεχόμενά της με διαχωριστή το κόμμα και τα αποθηκεύει. Έπειτα, στην μεταβλητή lhs_columns αποθηκεύονται οι αριθμοί των στηλών του αρχικού συνόλου δεδομένων που είναι αποθηκευμένα στον πίνακα matrix, οι οποίες στήλες έχουν στο όνομά τους τα αντικείμενα που διαχωρίστηκαν στο προηγούμενο βήμα. Επομένως, αυτή η μεταβλητή περιέχει αριθμούς, το πλήθος των οποίων είναι ίσο με το πλήθος των αντικειμένων που υπάρχουν στο σώμα του κάθε κανόνα.

Αν το πλήθος αυτών των αριθμών είναι μόνο ένα, δηλαδή ο κανόνας είχε μόνο ένα αντικείμενο στο σώμα του, τότε το διάνυσμα που θα αντιπροσωπεύει το σώμα του κανόνα στην κλήση της συνάρτησης cor() θα είναι η στήλη του αρχικού πίνακα βαθμολογιών της οποίας η θέση είναι στον αριθμό που έχει

αποθηκευτεί στην μεταβλητή lhs_items. Διαφορετικά, αν το σώμα του κανόνα περιέχει παραπάνω αντικείμενα, θα αποθηκευτούν όλα τα διανύσματα που εντοπίζονται στον αρχικό πίνακα των δεδομένων στις θέσεις που έχουν αποθηκευτεί στην μεταβλητή lhs_columns και στην συνέχεια υπολογίζεται ο μέσος όρος τους, που θα είναι ένα διάνυσμα ίσων διαστάσεων. Με την παράμετρο na.rm= FALSE κατά τον υπολογισμό του μέσου όρου ορίζονται πως οι τιμές που έχουν προκύψει ως NA (απώλεια τιμής) δεν πρέπει να αφαιρεθούν από το διάνυσμα, και αυτό είναι το επιθυμητό καθώς οι διαστάσεις του διανύσματος πρέπει να παραμείνουν αυστηρά στο μήκος του αριθμού των χρηστών προκειμένου να έχουν ίδιο μήκος με το διάνυσμα της κεφαλής του κανόνα. Και στις δύο περιπτώσεις το αποτέλεσμα είναι η μεταβλητή body_vector που περιέχει ένα διάνυσμα βαθμολογικών τιμών και απώλειας τιμής, με την μορφή NA, μήκους ίσου με τον αριθμό των χρηστών του αρχικού συνόλου δεδομένων. Αυτή η διαδικασία φαίνεται στην Εικόνα 5.5.

```
71  if(p$sort_measure == "pearson" || p$sort_measure == "spearman"){
72    for (i in 1:nrow(frame)){
73    lhs_items <- strsplit(as.character(frame$lhs[i]), ",")[[1]]
74    lhs_columns <- match(lhs_items, colnames(matrix))
75
76    if(length(lhs_columns) == 1){
77      body_vector <- matrix[, lhs_columns]
78    } else {
79      body_vectors <- matrix[, lhs_columns]
80      body_vector <- rowMeans(body_vectors, na.rm = FALSE)
81    }
82
83    head_vector <- matrix[,frame[i,2]]
84    similarity<- cor(body_vector, head_vector, method=similarityName, use = "complete.obs")
85    similarities[[i]] <- similarity
86    }
87  }
```

Εικόνα 5.5 Υπολογισμός συσχέτισης μεταξύ σώματος και κεφαλής ενός κανόνα

Στην συνέχεια υπολογίζεται το διάνυσμα της κεφαλής του κανόνα, που αντιστοιχεί στην στήλη του πίνακα του αρχικού συνόλου δεδομένων που έχει στον τίτλο της την ονομασία του αντικειμένου που βρίσκεται στην δεύτερη στήλη την μεταβλητής frame για τον συγκεκριμένο κανόνα. Αποθηκεύεται στην μεταβλητή head_vector.

Αφού πλέον έχουν υπολογιστεί τα δύο διανύσματα για το σώμα και την κεφαλή ενός κανόνα, υπολογίζεται η μεταξύ τους συσχέτιση με την συνάρτηση cor() που δέχεται ως παραμέτρους τα δύο διανύσματα, την μέθοδο που θα χρησιμοποιήσει, η οποία έχει αποθηκευτεί παραπάνω στην μεταβλητή similarityName ανάλογα με το ποιο μέτρο ταξινόμησης έχει δοθεί, και την παράμετρο use=”complete.obs” η οποια θα αφαιρέσει από τα διανύσματα τους χρήστες οι οποίοι έχουν βαθμολογήσει μόνο ένα ή σε κανένα αντικείμενο. Για τον υπολογισμό συσχέτισης μεταξύ δύο διανυσμάτων, είναι απαραίτητο ένας χρήστης να έχει βαθμολογήσει και στα δύο διανύσματα που συμμετέχουν στον υπολογισμό της συσχέτισης του κάθε κανόνα.

Στο τέλος, για τον συγκεκριμένο κανόνα αποθηκεύεται στην λίστα similarities ο αριθμός της συσχέτισης που προέκυψε από τους παραπάνω υπολογισμούς.

Αφού ολοκληρωθούν οι επαναλήψεις του παραπάνω κώδικα για όλους τους κανόνες της μεταβλητής frame, γίνεται έλεγχος για το ποια ήταν η τιμή της παραμέτρου sort_measure και αν ήταν η ονομασία μιας από τις δύο τεχνικές συσχέτισης θα προστεθεί η ανάλογη στήλη στα περιεχόμενα της rule_base, με τιμές συσχέτισης για κάθε κανόνα τις τιμές που έχουν αποθηκευτεί στην λίστα similarities, όπως φαίνεται στην Εικόνα 5.6.

Έτσι, όταν στην συνέχεια ταξινομηθούν οι κανόνες με βάση την τιμή της sort_measure, θα υπάρχει η στήλη με το αντίστοιχο όνομα όπως φαίνεται στην Εικόνα 5.7 όπου η τιμή της sort_measure ηταν "spearman".

```
90    ## add similarity to rule_base
91    if(p$sort_measure == "pearson") quality(rule_base) <-
92      cbind(quality(rule_base), pearson = unlist(similarities))
93
94    ## add similarity to rule_base
95    if(p$sort_measure == "spearman") quality(rule_base) <-
96      cbind(quality(rule_base), spearman = unlist(similarities))
```

Εικόνα 5.6 Προσθήκη συσχέτισης ως στήλη στην μεταβλητή rule_base

```
                                  rules   support confidence  coverage     lift count  spearman
336  {1625-1204,1625-1405} => {1625-1403} 0.1685895  0.7961336 0.2117603 1.734964   453 0.3361519
12  {1625-1305,1625-17422} => {1625-1742} 0.1016003  0.9820144 0.1034611 2.517817   273 0.2617650
339  {1625-1405,1625-1601} => {1625-1403} 0.1678452  0.7870855 0.2132490 1.715246   451 0.2543956
41   {1625-1742,1625-1842} => {1625-1969} 0.1161146  0.7898734 0.1470041 2.265091   312 0.2508404
377  {1625-1205,1625-1969} => {1625-1742} 0.1994790  0.8208270 0.2430220 2.104544   536 0.2379439
263  {1625-1401,1625-1405} => {1625-1403} 0.1559360  0.8026820 0.1942687 1.749235   419 0.2370438
376  {1625-1505,1625-1969} => {1625-1742} 0.1853368  0.8150573 0.2273911 2.089751   498 0.2326475
334  {1625-1301,1625-1405} => {1625-1403} 0.1581690  0.7988722 0.1979903 1.740932   425 0.2315803
16  {1625-1505,1625-17422} => {1625-1742} 0.1030889  0.9719298 0.1060662 2.491961   277 0.2302795
151  {1625-1203,1625-1403} => {1625-1102} 0.1659844  0.8527725 0.1946409 1.965180   446 0.2179278
25   {1625-1945,1625-1969} => {1625-1742} 0.1187198  0.8985915 0.1321176 2.303927   319 0.2163983
346  {1625-1301,1625-1969} => {1625-1742} 0.1711946  0.8348457 0.2050614 2.140487   460 0.2163750
5             {1625-17422} => {1625-1742} 0.1455154  0.9606880 0.1514700 2.463138   391 0.2159612
```

Εικόνα 5.7 Περιεχόμενα μεταβλητής rule_base με μέτρο ταξινόμησης "spearman"

5.3 Το κάτω_φίλτρο

Κατά την εφαρμογή των μέτρων αξίας με τους συντελεστές συσχέτισης όπως παρουσιάστηκαν στην Ενότητα 5.2, θα εφαρμοστεί η προσθήκη μιας ακόμα παραμέτρου, του **κάτω_φίλτρου (low_f)**.

Το φίλτρο αυτό εφαρμόζεται αμέσως μετά την ανάγνωση των δεδομένων από το αρχείο csv, όταν ακόμα έχουν την μορφή των τριών στηλών χρήστης-αντικείμενο-βαθμολογία. Το κάτω_φίλτρο θα διαγράψει όλες εκείνες τις καταχωρήσεις οι οποίες βρίσκονται κάτω από μία συγκεκριμένη αριθμητική τιμή, όπως φαίνεται παρακάτω με παράδειγμα από τα αριθμητικά δεδομένα.

```
> subjectsDataset<-read.csv('subjects.csv')
> subjectsDataset[1:5,]
                          Student_ID Course_ID Grade
1 25CFE594-0589-4380-85A6-DE2BC106FCE3 1625-1103   5.0
2 25CFE594-0589-4380-85A6-DE2BC106FCE3 1625-1102   1.0
3 25CFE594-0589-4380-85A6-DE2BC106FCE3 1625-1204   0.0
4 25CFE594-0589-4380-85A6-DE2BC106FCE3 1625-1204   2.5
5 25CFE594-0589-4380-85A6-DE2BC106FCE3 1625-1203   6.0
```

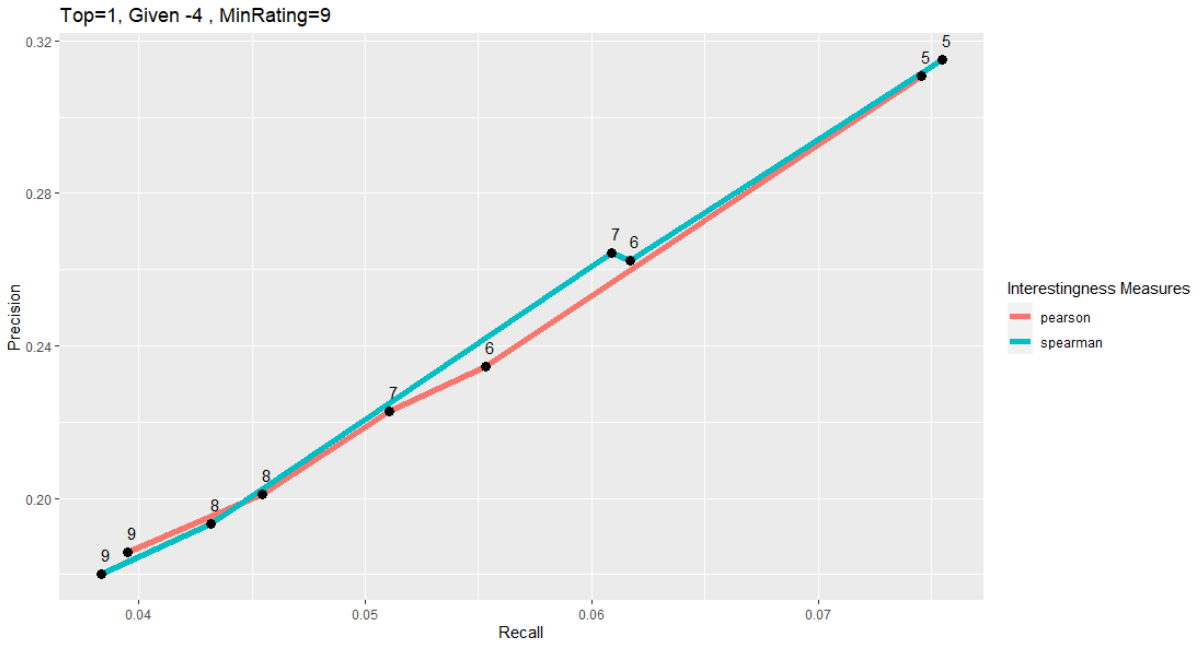
Επιλέγεται να διαγραφούν όλες οι βαθμολογίες που βρίσκονται κάτω από την βαθμολογία 5.0, και όπως φαίνεται και παρακάτω, κοιτάζοντας ξανά τις πρώτες 5 βαθμολογίες, έχουν διαγραφεί όσες δεν ικανοποιούσαν την συνθήκη.

```
> subjectsDataset<-subjectsDataset[subjectsDataset$Grade>=5,]
> subjectsDataset[1:5,]
                              Student_ID Course_ID Grade
1  25CFE594-0589-4380-85A6-DE2BC106FCE3 1625-1103     5.0
5  25CFE594-0589-4380-85A6-DE2BC106FCE3 1625-1203     6.0
10 25CFE594-0589-4380-85A6-DE2BC106FCE3 1625-1102     5.0
11 25CFE594-0589-4380-85A6-DE2BC106FCE3 1625-1204     5.0
13 25CFE594-0589-4380-85A6-DE2BC106FCE3 1625-1101     8.0
```

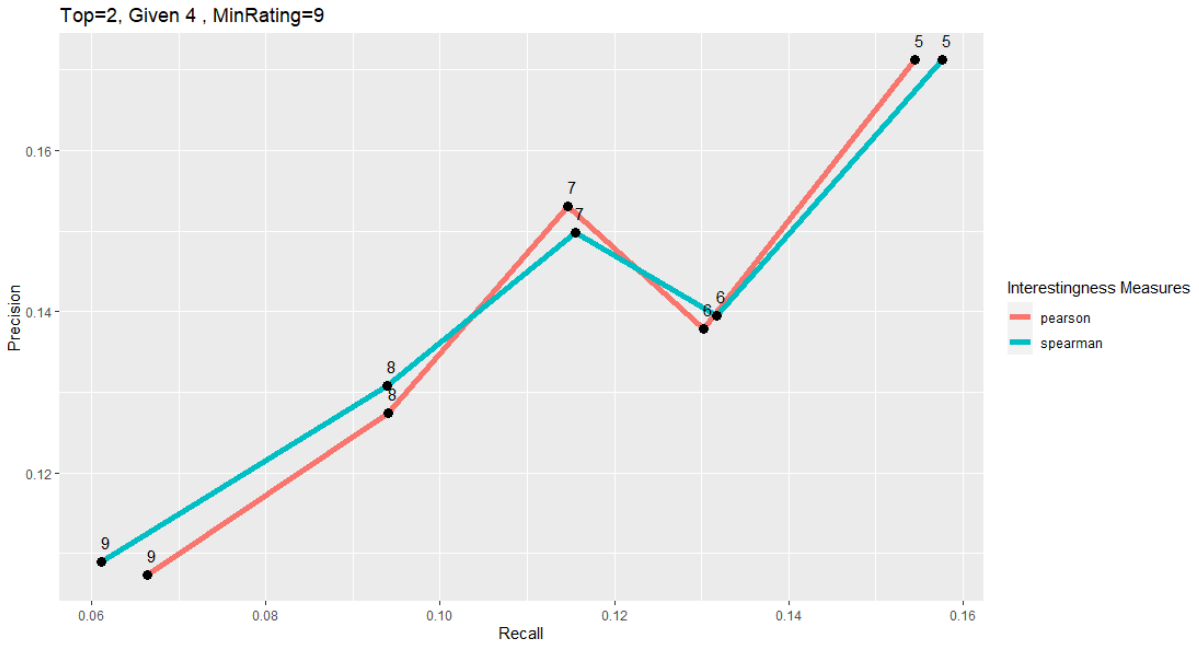
Η διαδικασία αυτή εφαρμόζεται καθώς στην περίπτωση της χρήσης των συντελεστών συσχέτισης, κατά τον υπολογισμό της συσχέτισης μεταξύ του σώματος και της κεφαλής, ο σκοπός είναι να ληφθούν υπόψη οι βαθμολογίες από φοιτητές που τα έχουν πάει καλά σε αυτά τα μαθήματα.

Ωστόσο, καθώς οι συντελεστές συσχέτισης χρειάζονται ένα ικανό εύρος τιμών για να μπορέσουν να παρουσιάσουν συσχέτιση και μεταβολή τιμής ανάλογα με την αλλαγή των βαθμολογιών, στην περίπτωση των βαθμολογικών δεδομένων είναι προτιμότερο να χρησιμοποιηθεί ως κάτω_φίλτρο το 5, που δηλώνει ο φοιτητής έχει περάσει ένα μάθημα και δίνει περιθώριο βαθμολογιών μέχρι και το 10. Η θεωρία αυτή επιβεβαιώνεται από τα 2 παρακάτω γραφήματα των Εικόνων 5.8 και 5.9, που αντιπροσωπεύουν τα αποτελέσματα πολλαπλών δοκιμών για περιπτώσεις με minRating τιμή ίση με 9 για να δοκιμαστεί το μεγαλύτερο εύρος τιμών για κάτω_φίλτρο και διάφορες τιμές given και διάφορες τιμές top-N που να αρμόζουν στους περιορισμούς minRating, που στην συγκεκριμένη περίπτωση η τιμή 9 διαθέτει πλέον ελάχιστα μαθήματα για κάθε φοιτητή.

Σημαντικό είναι να επισημανθεί πως μία τιμή minRating πρέπει να έχει τιμή μεγαλύτερη ή ίση του κάτω_φίλτρου, προκειμένου να μην επηρεαστεί η διαδικασία του binarization από το κάτω_φίλτρο. Υπενθυμίζεται πως το binarization θα αποδώσει τιμή false είτε ο χρήστης έχει μία βαθμολογία σε ένα αντικείμενο που δεν ικανοποιεί το minRating, είτε υπάρχει απώλεια τιμής. Η μόνη επίπτωση που μπορεί να έχει το κάτω_φίλτρο για τα υπόλοιπα μέτρα αξίας, είναι η διαγραφή κάποιων αντικειμένων ή χρηστών, καθώς υπάρχει περίπτωση στην διαγραφή των γραμμών στο στάδιο της εφαρμογής του να διαγραφούν όλες οι γραμμές στις οποίες θα εμφανίζοντας ένας χρήστης ή ένα αντικείμενο, και επομένως στην διαμόρφωση του realRatingMatrix να μην είναι γνωστή η ύπαρξή τους. Βέβαια, αυτοί οι χρήστες και αυτά τα αντικείμενα δεν θα αποτελούσαν χρήσιμη πληροφορία έτσι κι αλλιώς, καθώς κατά το binarization θα είχαν όλες τις τιμές τους false και δεν θα αποτελούσαν ποτέ μέρος της διαδικασίας παραγωγής κανόνων από τον apriori ή ως συστάσεις στην συνέχεια.



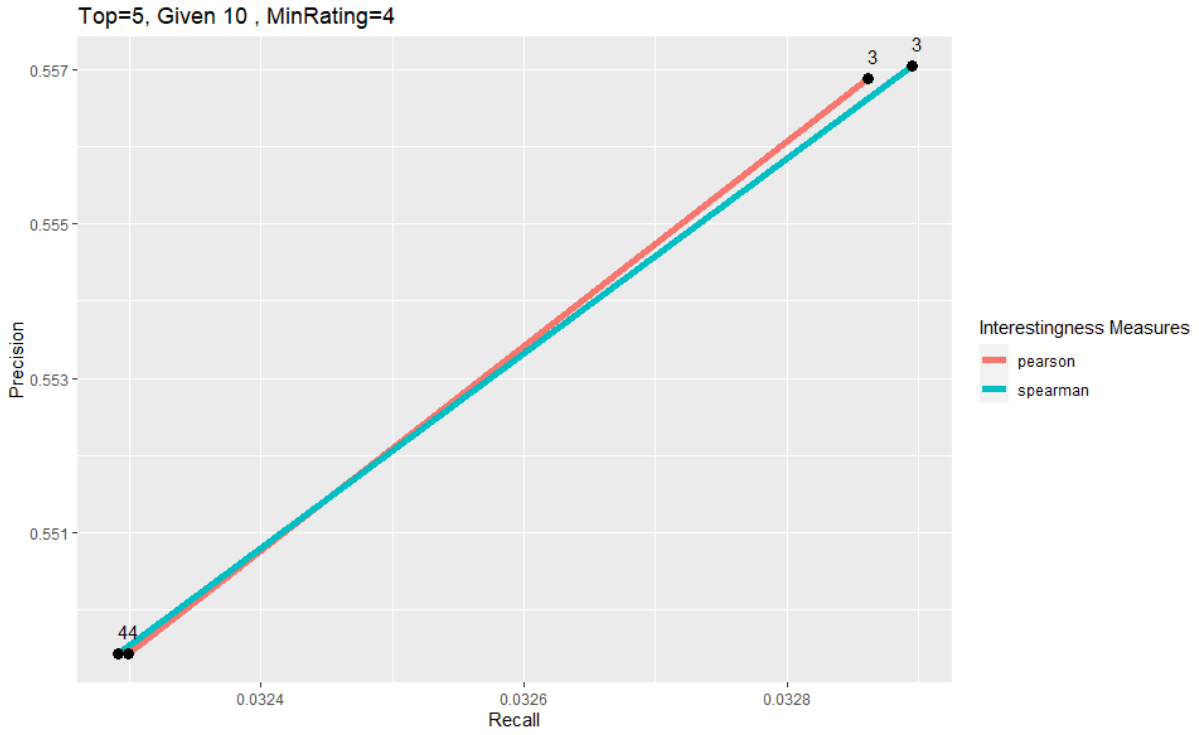
Εικόνα 5.8 Βαθμολογικά Δεδομένα: τιμές 5-9 για low_f, λίστα Top-1, given= -4 και minRating= 9



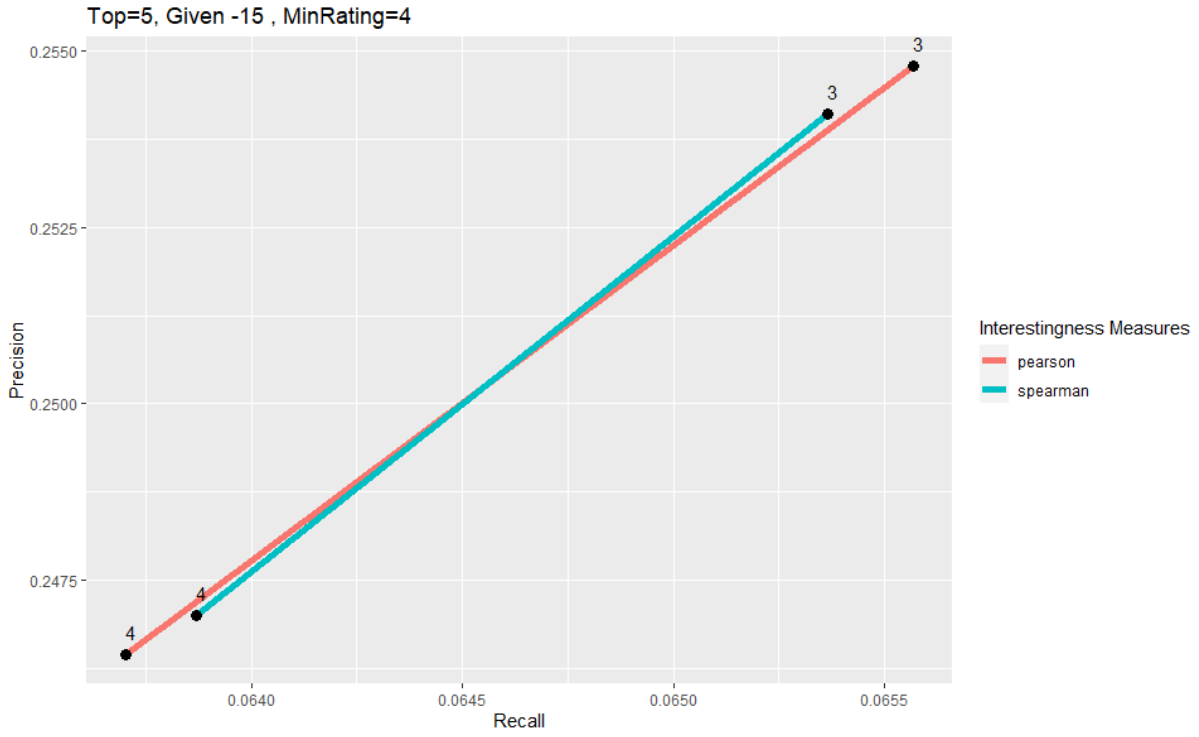
Εικόνα 5.9 Βαθμολογικά Δεδομένα: τιμές 5-9 για low_f, λίστα Top-2, given= 4 και minRating= 9

Όπως φαίνεται και από τα αποτελέσματα, η τιμή κάτω_φίλτρου ίση με 5 υπερέχει και σε τιμή precision αλλά και recall έναντι όλων των υπόλοιπων φίλτρων, και η τιμή κάτω_φίλτρου 9, που έχει αφήσει διαθέσιμη την μεταβολή των συσχετίσεων μόνο από 9 έως 10 παρουσιάσει τις χαμηλότερες τιμές precision και recall.

Ωστόσο, στα σύνολο δεδομένων των ταινιών, τα δεδομένα παρουσιάζουν μικρότερο εύρος τιμών, με τις αρχικές βαθμολογίες να κυμαίνονται από 1 μέχρι 5 και με τις καλές βαθμολογίες που να μπορούν να χρησιμοποιηθούν ως φίλτρα να θεωρούνται οι τιμές 3 και 4. Αυτό φαίνεται και τις παρακάτω Εικόνες 5.10 και 5.11.



Εικόνα 5.10 MovieLens 1M: τιμές 3-4 για low_f, λίστα Top-5, given= 10 και minRating= 4



Εικόνα 5.11 MovieLens 1M: τιμές 3-4 για low_f, λίστα Top-5, given= -15 και minRating= 4

5.4 Εφαρμογή και μετρήσεις

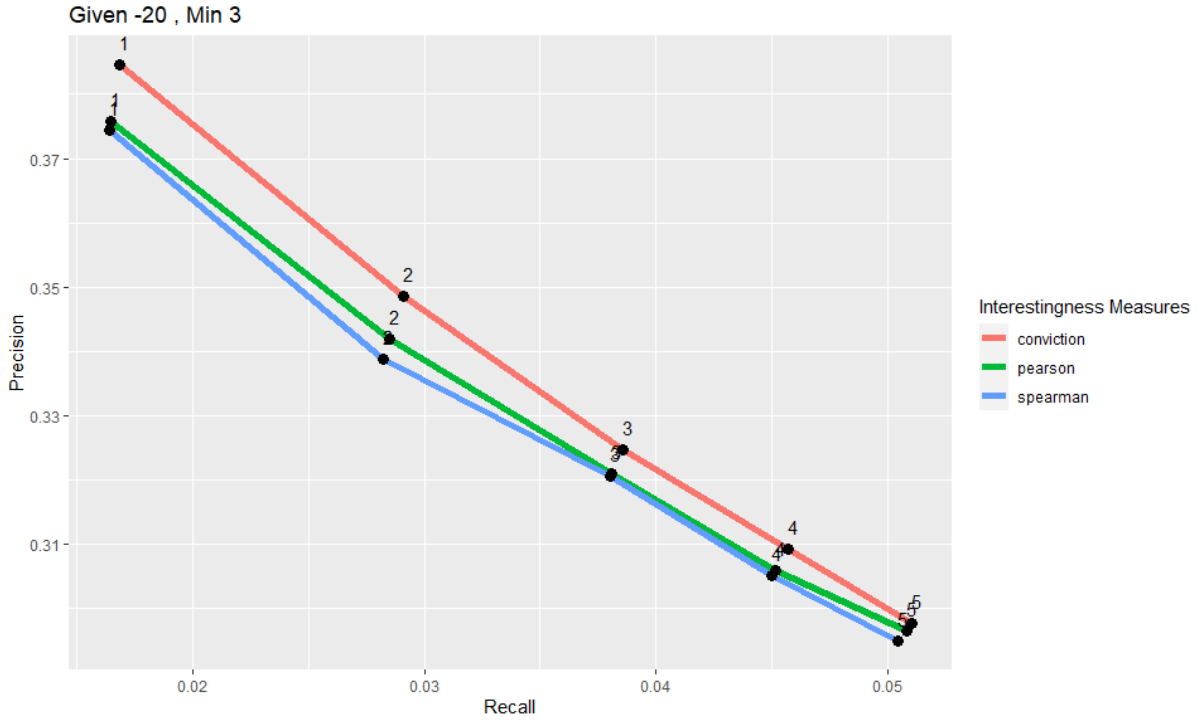
Σε αυτή την Ενότητα παρουσιάζονται τα αποτελέσματα των αξιολογήσεων της παραγωγής συστάσεων με την ταξινόμηση των κανόνων του αλγορίθμου apriori ως προς conviction, pearson και spearman, για το συνολο των βαθμολογικών δεδομένων του Τμήματος και για το σύνολο δεδομένων MovieLens 1M. Το μοντέλο αξιολόγησης που χρησιμοποιείται είναι αυτό της cross-validation καθώς εγγυάται την χρήση όλων των συναλλαγών ως test sets από μία φορά, και τα folds ορίζονται στα 5. Τα γραφήματα δημιουργήθηκαν με την χρήση της βιβλιοθήκης ggplot2, με τον άξονα x να αντιπροσωπεύει τις τιμές recall και τον άξονα y τις τιμές precision των συστάσεων.

5.4.1 Σύνολο MovieLens 1M

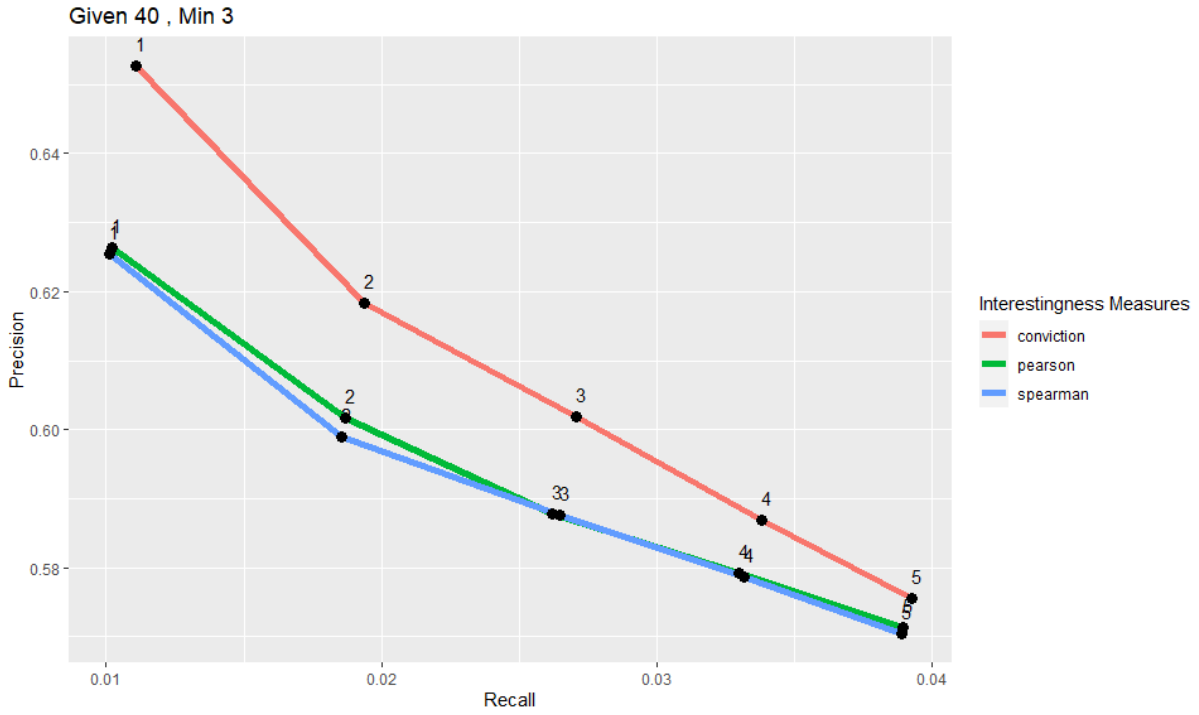
Για το σύνολο δεδομένων MovieLens 1M, δοκιμάζονται όλες οι ακέραιες τιμές binarize από το 3 έως το 4, και το φίλτρο low ορίζεται στο 3.

Στα παρακάτω γραφήματα, η αξιολόγηση έγινε πολλαπλές φορές για διαφορετικές τιμές given, με τις τιμές τους να μεταβάλλεται στα διαστήματα που αναφέρθηκαν στην Ενότητα 4.4.1, ανάλογα με την τιμή του minRating. Παρουσιάζονται 2 γραφήματα για κάθε minRating, με αρνητική και θετική τιμή given. Σε κάθε ένα από αυτά τα γραφήματα, οι γραμμές των δύο μέτρων αξίας μεταβάλλονται από την αλλαγή της τιμής top-N, που όπως προαναφέρθηκε, παίρνει τιμές από 1 μέχρι και 5.

Για την περίπτωση που το minRating είναι 3, δοκιμάστηκαν 18 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-150,150], με την πλειοψηφία αυτών να παρουσιάζουν τα ίδια αποτελέσματα που συνοψίζονται στις Εικόνες 5.12 και 5.13. Πιο συγκεκριμένα, από τις 18 περιπτώσεις, οι 14 παρουσίασαν υπεροχή του conviction έναντι των συντελεστών συσχέτισης, ενώ οι υπόλοιπες 4 περιπτώσεις παρουσίασαν μία ελάχιστη εναλλαγή μεταξύ των τριών μέτρων.

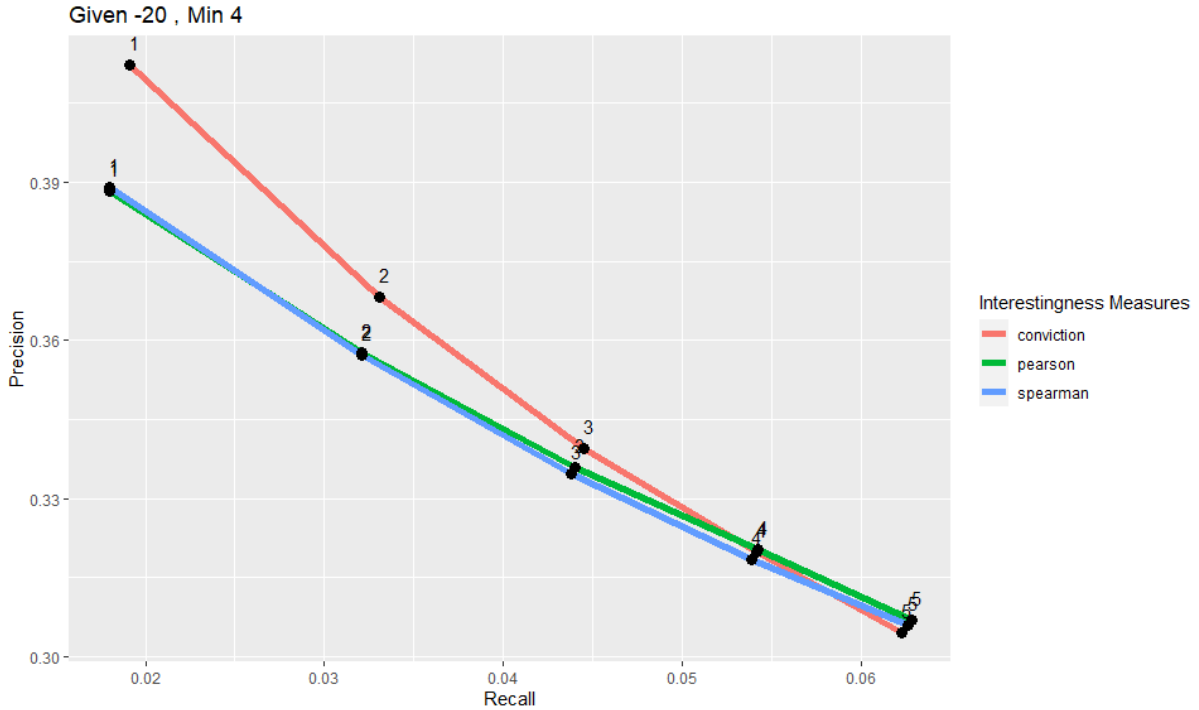


Εικόνα 5.12 MovieLens 1M: τιμές 1-5 για Top-N, given= -20, low_f= 3 και minRating= 3

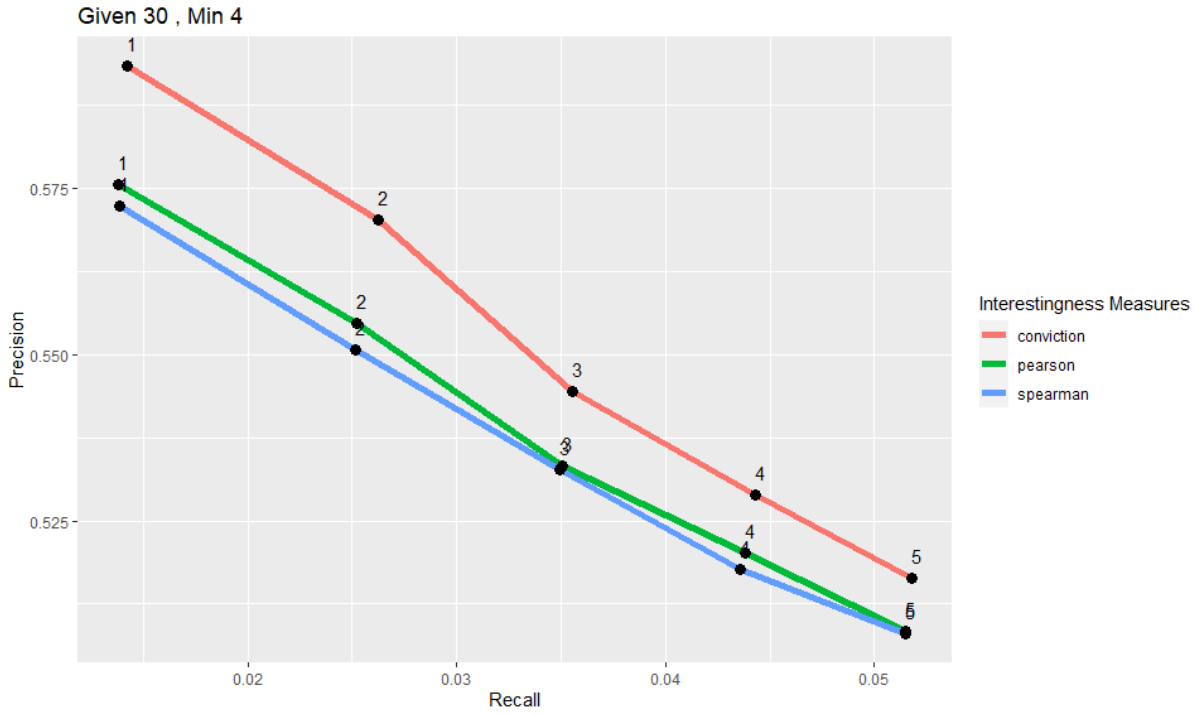


Εικόνα 5.13 MovieLens 1M: τιμές 1-5 για Top-N, given= 40, low_f= 3 και minRating= 3

Για την περίπτωση που το minRating είναι 4, δοκιμάστηκαν 16 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-150,150], με την πλειοψηφία αυτών να παρουσιάζουν τα ίδια αποτελέσματα που συνοψίζονται στις Εικόνες 5.14 και 5.15. Πιο συγκεκριμένα, από τις 16 περιπτώσεις οι 14 παρουσίασαν υπεροχή του μέτρου conviction έναντι των συντελεστών συσχέτισης, με τις υπόλοιπες 2 να παρουσιάζουν μια μικρή εναλλαγή μεταξύ των τριών μέτρων.



Εικόνα 5.14 MovieLens 1M: τιμές 1-5 για Top-N, given= -20, low_f= 3 και minRating= 4



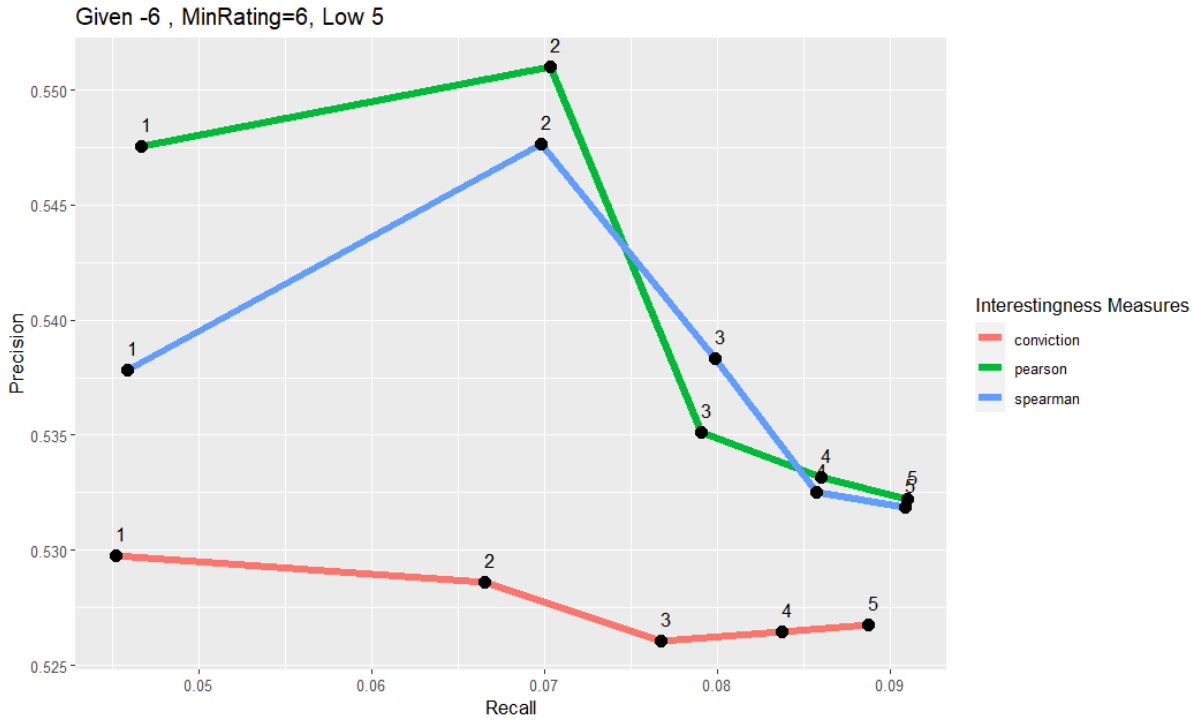
Εικόνα 5.15 MovieLens 1M: τιμές 1-5 για Top-N, given= 30, low_f= 3 και minRating= 4

5.4.2 Σύνολο βαθμολογικών δεδομένων του Τμήματος

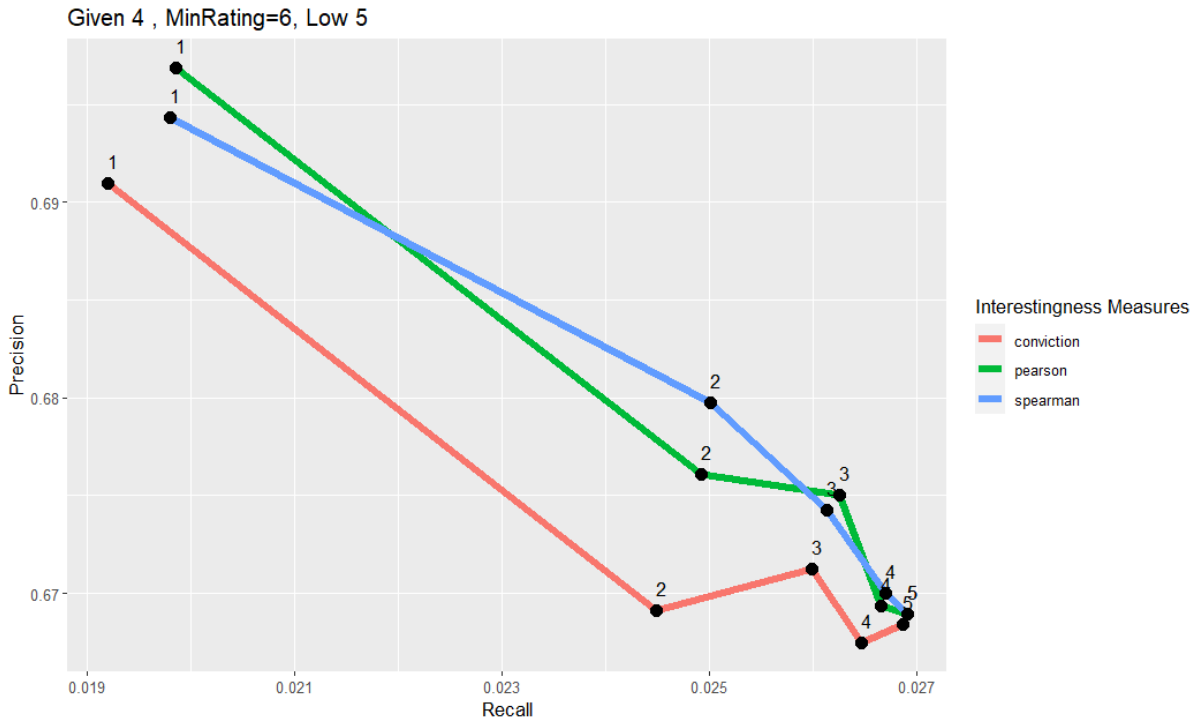
Για το σύνολο των βαθμολογικών δεδομένων του Τμήματος, δοκιμάζονται όλες οι ακέραιες τιμές binarize από το 6 έως το 9, και το φίλτρο low ορίζεται στο 5.

Στα παρακάτω γραφήματα, η αξιολόγηση έγινε πολλαπλές φορές για διαφορετικές τιμές given, με τις τιμές τους να μεταβάλλεται στα διαστήματα που αναφέρθηκαν στην Ενότητα 4.3.1, ανάλογα με την τιμή του minRating. Παρουσιάζονται 2 γραφήματα για κάθε minRating, με αρνητική και θετική τιμή given. Σε κάθε ένα από αυτά τα γραφήματα, οι γραμμές των δύο μέτρων αξίας μεταβάλλονται από την αλλαγή της τιμής top-N, που όπως προαναφέρθηκε, παίρνει τιμές από 1 μέχρι και 5.

Για την περίπτωση που το minRating είναι 6, δοκιμάστηκαν 14 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-15,15], με την πλειοψηφία αυτών να παρουσιάζουν τα ίδια αποτελέσματα που συνοψίζονται στις Εικόνες 5.16 και 5.17. Πιο συγκεκριμένα, από τις 14 περιπτώσεις, οι 11 από αυτές παρουσίασαν υπεροχή των pearson και spearman έναντι του conviction, όπως φαίνεται στις Εικόνες, ενώ 3 παρουσίασαν εναλλαγή μεταξύ των τριών μέτρων.

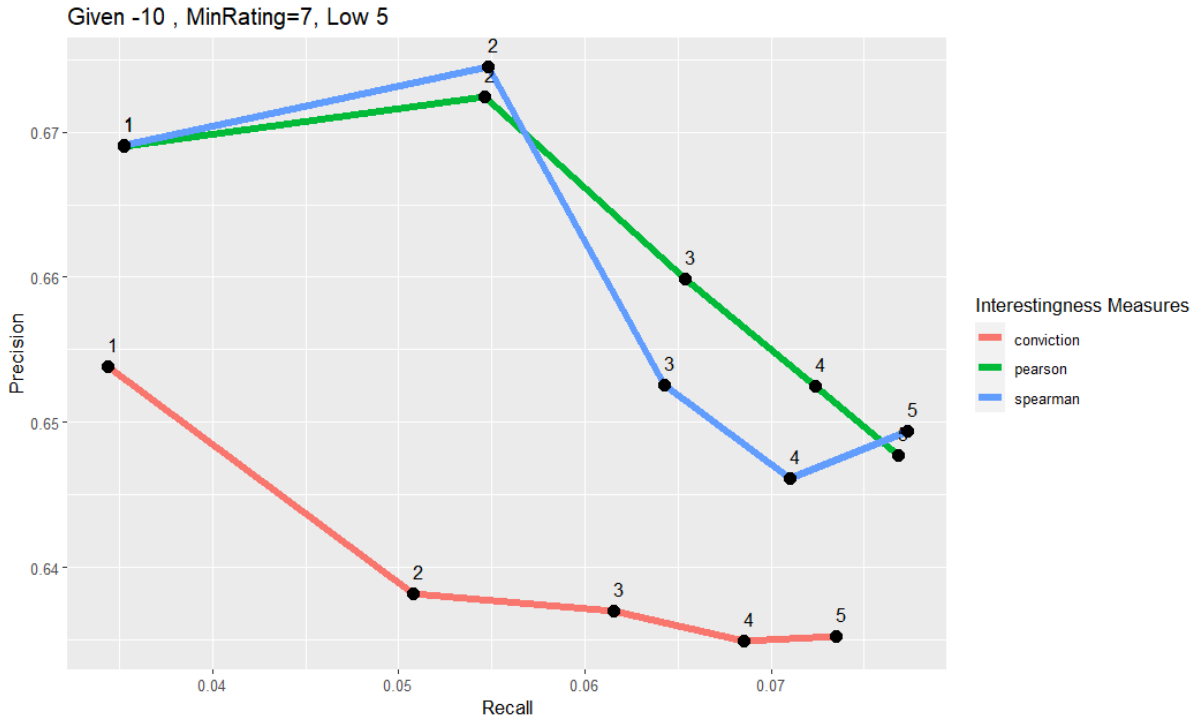


Εικόνα 5.16 Βαθμολογικά δεδομένα: τιμές 1-5 για Top-N, given= -6, low_f= 5 και minRating= 6

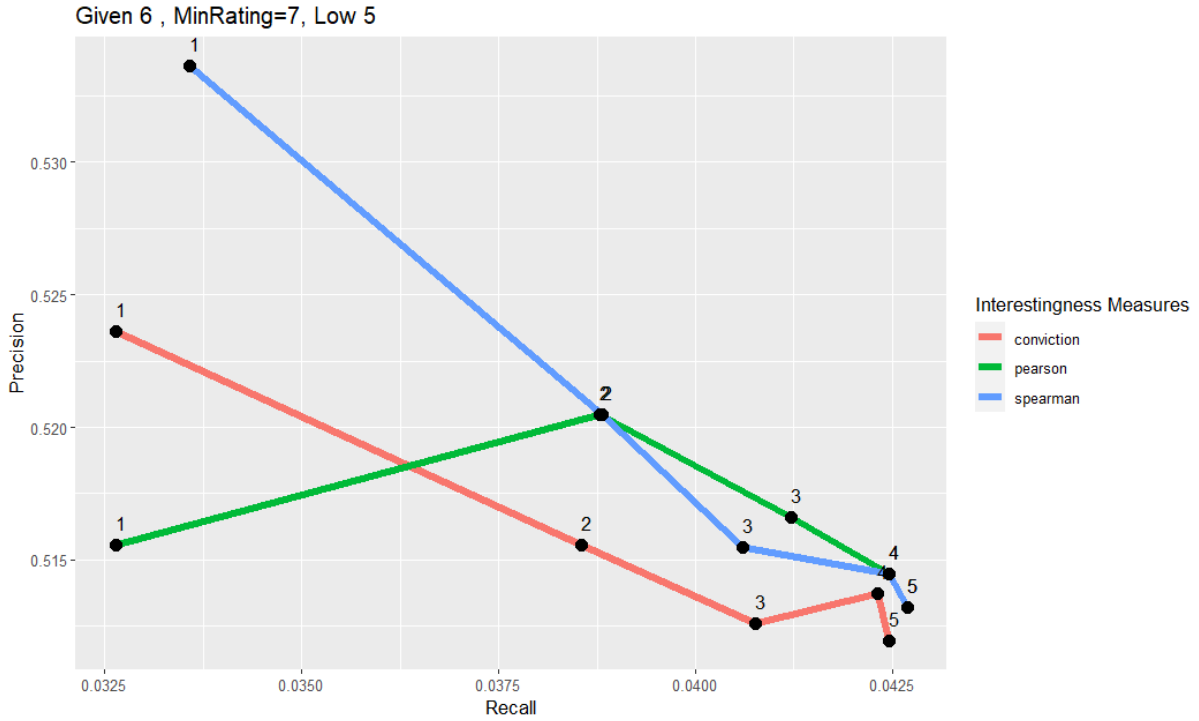


Εικόνα 5.17 Βαθμολογικά δεδομένα: τιμές 1-5 για Top-N, given= 4, low_f= 5 και minRating= 6

Για την περίπτωση που το minRating είναι 7, δοκιμάστηκαν 12 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-12,12], με την πλειοψηφία αυτών να παρουσιάζουν τα ίδια αποτελέσματα που συνοψίζονται στις Εικόνες 5.18 και 5.19. Πιο συγκεκριμένα, από τις 12 περιπτώσει, οι 10 παρουσίασαν υπεροχή των pearson και spearman, 1 περίπτωση όπου εναλλάσσονται και 1 που υπερέχει το conviction.

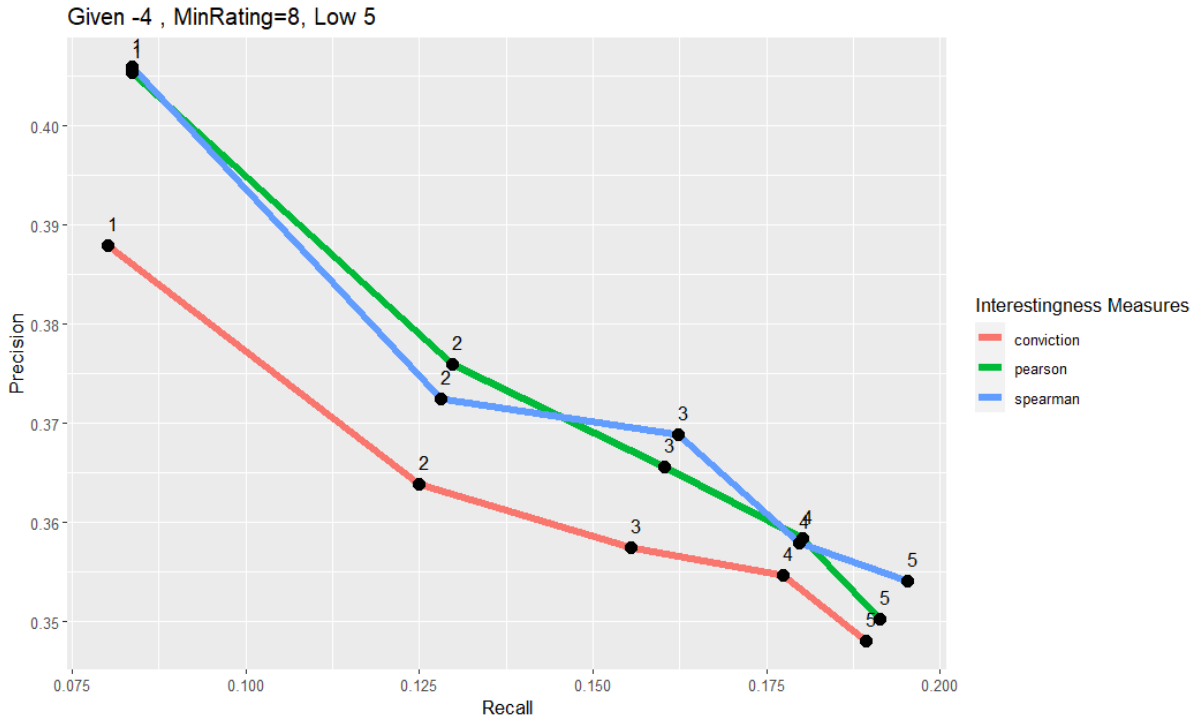


Εικόνα 5.18 Βαθμολογικά δεδομένα: τιμές 1-5 για Top-N, given= -10, low_f= 5 και minRating= 7

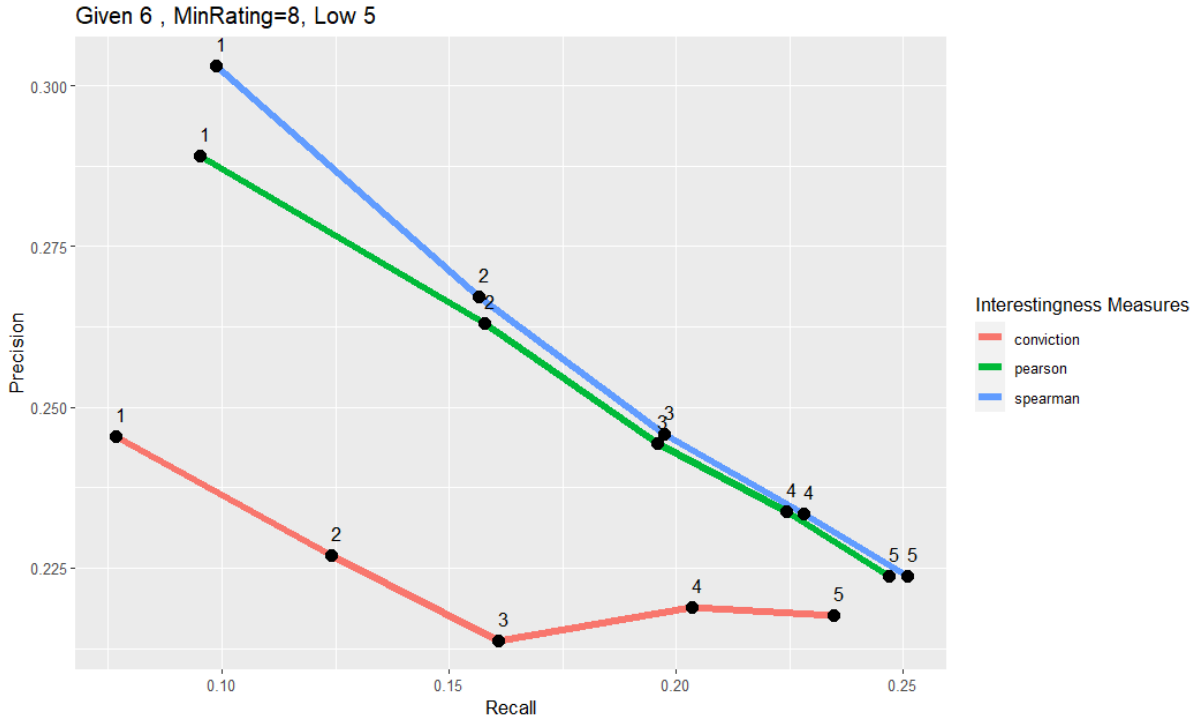


Εικόνα 5.19 Βαθμολογικά δεδομένα: τιμές 1-5 για Top-N, given= 6, low_f= 5 και minRating= 7

Για την περίπτωση που το minRating είναι 8, δοκιμάστηκαν 10 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-10,10], με την πλειοψηφία αυτών να παρουσιάζουν τα ίδια αποτελέσματα που συνοψίζονται στις Εικόνες 5.20 και 5.21. Πιο συγκεκριμένα, από τις 10 περιπτώσεις οι 7 παρουσίασαν υπεροχή των pearson και spearman ενώ οι εναλλαγή των τριών μέτρων.

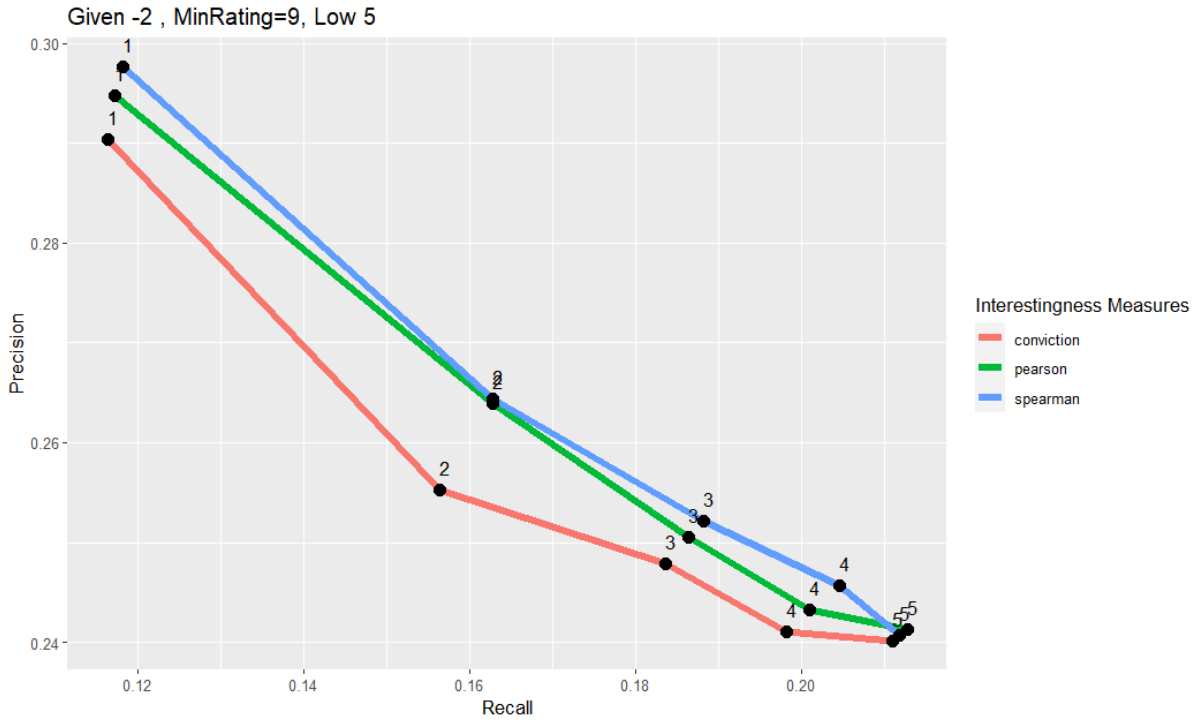


Εικόνα 5.20 Βαθμολογικά δεδομένα: τιμές 1-5 για Top-N, given= -4, low_f= 5 και minRating= 8

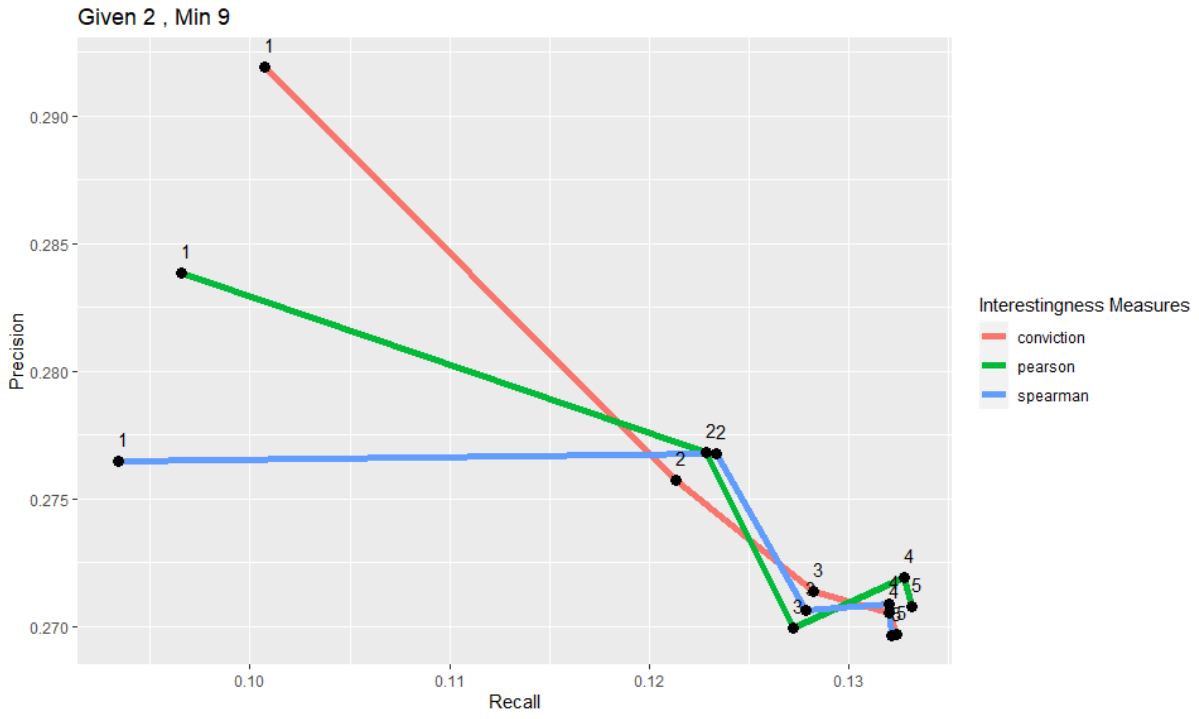


Εικόνα 5.21 Βαθμολογικά δεδομένα: τιμές 1-5 για Top-N, given= 6, low_f= 5 και minRating= 8

Για την περίπτωση που το minRating είναι 9, δοκιμάστηκαν 8 περιπτώσεις διαφορετικών τιμών given στο εύρος τιμών [-10,10], με την πλειοψηφία αυτών να παρουσιάζουν τα ίδια αποτελέσματα που συνοψίζονται στις Εικόνες 5.22 και 5.23. Πιο συγκεκριμένα, από τις 8 περιπτώσεις, οι 4 παρουσίασαν υπεροχή των pearson και spearman ενώ οι υπόλοιπες 4 εναλλαγή των τριών μέτρων.



Εικόνα 5.22 Βαθμολογικά δεδομένα: τιμές 1-5 για Top-N, given= -2, low_f= 5 και minRating= 9



Εικόνα 5.23 Βαθμολογικά δεδομένα: τιμές 1-5 για Top-N, given= 2, low_f= 5 και minRating= 9

5.5 Συμπεράσματα

Συνοψίζοντας, σε όλες τις περιπτώσεις που δοκιμάστηκαν και στα δύο σύνολα δεδομένων, το μέτρο αξίας που κάνει χρήση των συντελεστών συσχέτισης pearson correlation και spearman correlation

φαίνεται να υπερέχει του conviction στις περισσότερες περιπτώσεις στο σύνολο των βαθμολογικών δεδομένων του Τμήματος, ενώ το conviction φαίνεται να υπερέχει στο σύνολο δεδομένων των ταινιών MovieLens 1M. Τα αποτελέσματα ήταν αναμενόμενα, καθώς όπως αναφέρθηκε και στην Ενότητα 5.3, οι συντελεστές συσχέτισης παρόλο που μπορούν να κάνουν χρήση αριθμητικών δεδομένων και να αξιοποιήσουν πληροφορία που χάνεται κατά την εφαρμογή της ARM, χρειάζονται ένα εύρος τιμών ώστε να μπορέσουν να ταυτοποιήσουν την συσχέτιση μεταξύ του σώματος και της κεφαλής των κανόνων. Στην περίπτωση των δεδομένων του Τμήματος, αυτά τα δεδομένα έχουν μεγαλύτερο εύρος, σε αντίθεση με τα δεδομένα των ταινιών που το εύρος των βαθμολογιών είναι πολύ περιορισμένο.

5.6 Επίλογος

Έγινε πρόταση προς διερεύνηση η ταξινόμηση των Association Rules ως προς ένα μέτρο αξίας που κάνει χρήση αριθμητικών δεδομένων και χρησιμοποιεί συντελεστές συσχέτισης για την εύρεση της συσχέτισης μεταξύ του σώματος και της κεφαλής των κανόνων, και επιλέγονται οι συντελεστές Pearson Correlation και Spearman Correlation. Έγινε η προσθήκη μιας επιπλέον παραμέτρου κατά των συντελεστών συσχέτισης, ενός κάτω_φίλτρου που δεν λαμβάνει υπόψη τις βαθμολογίες εκείνες που είναι χαμηλές, καθώς οι ομοιότητες πρέπει να συμπεριλάβουν τις βαθμολογίες των χρηστών που προτίμησαν/τα πήγαν καλά στα αντικείμενα των κανόνων, δίνοντας όμως περιθώριο στους συντελεστές συσχέτισης για να ταυτοποιήσουν την σωστή κατεύθυνση της συσχέτισης. Έπειτα εκτελείται η αξιολόγηση των συστάσεων με την χρήση των συντελεστών συσχέτισης ως μετρικές ταξινόμησης των Association Rules, έναντι της conviction που υπερέχει στο Κεφάλαιο 4. Σύμφωνα με τα αποτελέσματα που παρουσιάστηκαν ως γραφήματα με την χρήση της βιβλιοθήκης ggplot2, στο σύνολο των βαθμολογικών δεδομένων φαίνονται να υπερέχουν οι συντελεστές συσχέτισης στην πλειοψηφία των γραφημάτων, ενώ στο σύνολο δεδομένων των MovieLens 1M φαίνεται να υπερέχει η conviction.

Κεφάλαιο 6ο : Συμπεράσματα και προτάσεις για την συνέχεια

Στην παρούσα πτυχιακή εργασία διερευνήθηκαν οι τεχνικές παραγωγής συστάσεων, με εστίαση σε αυτή που κάνει χρήση των Association Rules. Πιο συγκεκριμένα, μελετήθηκαν τα διάφορα μέτρα αξίας που υπολογίζουν την σημασία ενός κανόνα, τον μοντέλων αξιολόγησης που διαχωρίζουν το σύνολο των δεδομένων σε training set και test set για την διαδικασία της αξιολόγησης, και τα μέτρα επιδόσεων που παρέχουν τα αποτελέσματα της αξιολόγησης των συστάσεων.

Η εργασία επικεντρώνεται την περίπτωση της εφαρμογής της ARM στην παραγωγή συστάσεων σε σύνολα δεδομένων που περιέχουν αριθμητικές τιμές. Πιο συγκεκριμένα, στην χρήση των συνόλων δεδομένων MovieLens 1M εφαρμόζεται η αξιολόγηση στις συστάσεις του τύπου “Αν σε έναν χρήστη άρεσε η Α ταινία, τότε θα του αρέσει και η Β ταινία”, ενώ στο σύνολο των βαθμολογικών δεδομένων ότι “Αν ένας μαθητής τα πήγε καλά στο Α μάθημα, τότε θα τα πάει καλά και στο Β μάθημα”. Τα μέτρα αξίας conviction και lift εκφράζουν την δύναμη της συσχέτισης μεταξύ σώματος και κεφαλής και το πόσο η παρουσία του ενός επηρεάζει την παρουσία του άλλου, με το conviction να υπερέχει σύμφωνα με τα αποτελέσματα καθώς δεν περιέχει θόρυβο. Όμως, αυτά τα μέτρα δεν αξιοποιούν τις αριθμητικές τιμές του συνόλου δεδομένων.

Διερευνάται η χρήση ενός μέτρου αξίας το οποίο αξιοποιεί τα αρχικά αριθμητικά δεδομένα και υπολογίζει την συσχέτιση μεταξύ του σώματος και της κεφαλής των κανόνων, με την περίπτωση της Pearson Correlation και της Spearman Correlation, και εφαρμόζεται η αξιολόγησή τους σε σύγκριση με το μέτρο αξίας conviction. Στα αποτελέσματα των βαθμολογικών δεδομένων, οι συντελεστές συσχέτισης φαίνονται να υπερέχουν καθώς υπάρχει μεγάλο διάστημα τιμών προκειμένου να ταυτοποιηθεί ο βαθμός συσχέτισης μεταξύ των μαθημάτων. Αντιθέτως, στα δεδομένα MovieLens 1M η conviction υπερέχει, καθώς τα περιθώρια να εμφανιστεί συσχέτιση μεταξύ των ταινιών ήταν πολύ μικρά.

Συμπερασματικά, ένα μέτρο αξίας των Association Rules που αξιοποιεί αριθμητικά δεδομένα με τον υπολογισμό ομοιοτήτων μεταξύ σώματος και κεφαλής θα μπορούσε να αποτελέσει ένα αντικείμενο προς ακόμα μεγαλύτερη διερεύνηση, καθώς τα αποτελέσματα ήταν ενθαρρυντικά στην περίπτωση που υπήρχε το ανάλογο περιθώριο ανάδειξης των συσχετίσεων. Το κάθε μέτρο αξίας έχει διαφορετική σημασία και σκοπό που χρησιμοποιείται, επομένως είναι απαραίτητη η κατανόηση ενός προβλήματος προκειμένου να ταυτοποιηθεί το κατάλληλο μέτρο αξίας που αρμόζει στην επίλυση ενός προβλήματος με την χρήση των Association Rules.

Τα αποτελέσματα της πτυχιακής εργασίας θα ενταχθούν ως εν δυνάμη ενίσχυση της προσέγγισης που γίνεται στο ζήτημα της επεξεργασίας των βαθμολογικών δεδομένων στο πλαίσιο της διδακτορικής διατριβής του κ. Κώστα Κελεσίδη.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Paul Resnick, Hal R. Varian, “Recommender Systems”, Communications of the ACM, vol. 40, 1 Mar., 1997, pp. 56-58 (doi: https://doi.org/10.1145/245108.245121)

[2] Linyuan L., Matus M., Chi Ho Y., Yi-Cheng Z., Zi-Ke Z., Tao Z., “Recommender systems”, Physics Reports 519, 7 Feb., 2012, pp. 1-49 (doi: http://dx.doi.org/10.1016/j.physrep.2012.02.006)

[3] Michael Hahsler, “recommenderlab: An R Framework for Developing and Testing Recommendation Algorithms”, Feb. 2015

[4] Sarwar B., Karypis G., Konstan J., Riedl J., “Item-Based Collaborative Filtering Recommendation Algorithms”, Proceedings of the 10th international conference on World Wide Web, 1 April, 2001, pp. 285-295 (doi: http://dx.doi.org/10.1145/371920.372071)

[5] Priyank T., Krunal V., Vijay U., Sapan M., Sudeep T., “Combining User-Based and Item-Based Collaborative Filtering Using Machine Learning”, Proceeding of ICTIS 2018, vol.2, pp. 173-180 (doi: https://doi.org/10.1007/978-981-13-1747-7_17)

[6] Schafer B., Frankowski D., Herlocker J., Sen S., “Collaborative Filtering Recommender Systems”, The Adaptive Web, 24 April 2007, pp 291-324 (doi: https://doi.org/10.1007/978-3-540-72079-9_9)

[7] Recommenderlab source listing, RECOM_UBCF.R. [Online]. Available: https://rdrr.io/cran/recommenderlab/src/R/RECOM_UBCF.R (last accessed 30/08/2023)

[8] Recommenderlab source listing, RECOM_IBCF.R. [Online]. Available: https://rdrr.io/cran/recommenderlab/src/R/RECOM_IBCF.R (last accessed 30/08/2023)

[9] Pasquale Lops, Marco de Gemmis, Giovanni Semeraro, “Content-based Recommender Systems: State of the Art and Trends”, Recommender Systems Handbook, 1 Jan., 2010, pp. 73-105 (doi: https://doi.org/10.1007/978-0-387-85820-3_3)

[10] Charu C. Aggarwal, Recommender Systems, 2016 (doi: https://doi.org/10.1007/978-3-319-29659-3)

[11] Google Developers Machine Learning, “Content Based Filtering”. [Online]. Available: https://developers.google.com/machine-learning/recommendation/content-based/basics (last accessed 30/08/2023)

[12] Google Developers Machine Learning, “Content Based Filtering Advantages and Disadvantages”. [Online]. Available: https://developers.google.com/machine-learning/recommendation/content-based/summary (last accessed 30/08/2023)

[13] M. Ludewig, N. Mauro, S. Lafiti, D. Jannach, “Empirical Analysis of Session-Based Recommendation Algorithms”, User Modeling and User-Adapted Interaction (2020), pp. 149-181 (doi: https://doi.org/10.1007/s11257-020-09277-1)

[14] Shoujin W., Longbing C., Yan W., Wuan Z. S., Mehmet A. O., Defu L., “A Survey on Session-based Recommender Systems”, ACM Computing Surveys, vol. 57, pp. 1-38, 2021 (doi: http://dx.doi.org/10.1145/3465401)

[15] R. Harbir Singh, S. Maurya, T. Tripathi, T. Narula, G. Srivastav, “Movie Recommendation System using Cosine Similarity and KNN”, International Journal of Engineering and Advanced Technology (IJEAT), vol. 9, June 2020, pp. 556-559 (doi: http://dx.doi.org/10.35940/ijeat.E9666.069520)

[16] Suja Cherukullapurath Mana, T. Sasipraba, “Research on Cosine Similarity and Pearson Correlation based Recommender Models”, International Conference on Mathematical Sciences (ICMS), March 2021 (doi: 10.1088/1742-6596/1770/1/012014)

[17] Leily Sheugh, Sasan H. Alizadeh, “A note on Pearson Correlation Coefficient as a metric of similarity in recommender system”, 2015 AI & Robotics (IRANOPEN), 12 April 2015 (doi: http://dx.doi.org/10.1109/RIOS.2015.7270736)

[18] C. Xiao, J. Ye, R. Maximo Esteves, C. Rong, “Using Spearman’s correlation coefficients for exploratory data analysis on big data”, Special Issue: Combined Special Issues on SORT 2014 and Social Network analysis and its application 2015, 25 Sep. 2016, pp. 3866-3878 (doi: https://doi.org/10.1002/cpe.3745)

[19] Thomas D. Gauthier, “Detecting Trends Using Spearman’s Rank Correlation Coefficient”, Environmental Forensics (2001) 2 , pp. 359-362 (doi: https://doi.org/10.1006/enfo.2001.0061)

[20] Sujoy B., Sri Krishna K., Manoj Kumar T., “An efficient recommendation generation using relevant Jaccard similarity”, Information Sciences, 9 Jan., 2019, pp. 53-64 (doi: https://doi.org/10.1016/j.ins.2019.01.023)

[21] Sotiris Kotsiantis, Dimitris Kanellopoulos, “Association Rules Mining: A Recent Overview”, GESTS International Transactions on Computer Science and Engineering, Vol 32, 2006, pp. 71-82

[22] Muhamad Brilliant, DwiHandoko, Sriyanto, “Implementation of Data Mining Using Association Rules for Transactional Data Analysis”, Proceeding International Conference on Information Technology and Business, 7 Dec., 2017, pp. 177- 180

[23] Manpreet Kaur, Shivani Kang, “Market Basket Analysis: Identify the changing trends of market using association rule mining”, International Conference on Computational Modelling and Security (CMS 2016), pp. 78-85 (doi: https://doi.org/10.1016/j.procs.2016.05.180)

[24] IBM Documentation, “Support in an Association Rule”, 1 Mar. 2021, [Online]. Available: https://www.ibm.com/docs/da/db2/9.7?topic=associations-support-in-association-rule (last accessed 30/08/2023)

[25] Jiawei Han, Micheline Kamber, Jian Pei, Data Mining Concepts and Techniques, 2012 (doi: https://doi.org/10.1016/C2009-0-61819-5)

[26] IBM Documentation, “Confidence in an Association Rule”, 1 Mar. 2021, [Online]. Available: https://www.ibm.com/docs/da/db2/9.7?topic=associations-confidence-in-association-rule (last accessed 30/08/2023)

[27] IBM Documentation, “Lift in an Association Rule”, 1 Mar. 2021, [Online]. Available: https://www.ibm.com/docs/da/db2/9.7?topic=associations-lift-in-association-rule (last accessed 30/08/2023)

[28] Dinesh J. Prajapati, Sanjay Garg, N.C. Chauhan, “Interesting association rule mining with consistent and inconsistent rule detection from big sales in distributed environment”, Future Computing and Informatics Journal (2017), pp. 19-30 (doi: https://doi.org/10.1016/j.fcij.2017.04.003)

[29] Paulo J. Azevedo, Alipio M. Jorge, “Comparing Rule Measures for Predictive Association Rules”, Machine Learning: ECML 2007, pp.510-517 (doi: https://doi.org/10.1007/978-3-540-74958-5_47)

[30] Jugendra Dongre, Gend Lal Prajapati, “The Role of Apriori Algorithm for Finding the Association Rules in Data Mining”, 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 8 Feb. 2014, pp. 657-660 (doi: https://doi.org/10.1109/ICICICT.2014.6781357)

[31] Recommenderlab source listing, RECOM_AR.R. [Online]. Available: https://rdrr.io/cran/recommenderlab/src/R/RECOM_AR.R (last accessed 30/08/2023)

[32] Michael Hahsler, “Package ‘recommenderlab’”, 20 June 2023

[33] Daniel Berrar, “Cross Validation”, Encyclopedia of Bioinformatics and Computational Biology, 2019 (doi: http://dx.doi.org/10.1016/B978-0-12-809633-8.20349-X)

[34] J. Martin Bland, Douglas G. Altman, “Statistics Notes: Bootstrap resampling methods”, 2 June 2015. [Online]. Available: https://www.bmj.com/content/350/bmj.h2622 (last accessed 30/08/2023)

[35] Hossin b. Mohammad, Md Nasir Sulaiman, “A review on evaluation metrics for data classification evaluations”, International Journal of Data Mining & Knowledge Management Process (IJDKP) vol. 5, March 2015 (doi: http://dx.doi.org/10.5121/ijdkp.2015.5201)

[36] Google Developers Machine Learning, “Classification: Precision and Recall”. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall (last accessed 30/08/2023)

[37] Google Developers Machine Learning, “Classification: Accuracy”. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/accuracy (last accessed 30/08/2023)

[38] The R Foundation, ‘What is R?’. [Online]. Available: https://www.r-project.org/about.html (last accessed 30/08/2023)

[39] The R Foundation, “R Language Definition”, 16 June 2023. [Online]. Available: https://cran.r-project.org/doc/manuals/r-release/R-lang.pdf (last accessed 30/08/2023)

[40] Datacamp, “Python vs R for Data Science: Which should you learn?”, Dec. 2022. [Online]. Available: https://www.datacamp.com/blog/python-vs-r-for-data-science-whats-the-difference (last accessed 30/08/2023)

[41] The R Foundation, “Contributed Packages”. [Online]. Available: https://cran.r-project.org/web/packages/index.html (last accessed 30/08/2023)

[42] Kent State University, “Statistical & Qualitative Data Analysis Software: About R and RStudio”. [Online]. Available: https://libguides.library.kent.edu/statconsulting/r (last accessed 30/08/2023)

[43] Christophe Genolini, “A (Not So) Short Introduction to S4”, 20 August, 2008

[44] Michael Hahsler. [Online]. Available: https://michael.hahsler.net/ (last accessed 30/08/2023)

[45] RDocumentation, “recommenderlab- Lab for Developing and Testing Recommender Algorithms - R package”. [Online]. Available: https://www.rdocumentation.org/packages/recommenderlab/versions/0.2-7 (last accessed 30/08/2023)

[46] Roger Koenker, Pin Ng, “SparseM: A Sparse Matrix Package for R”, 18 Feb., 2021 (doi: http://dx.doi.org/10.18637/jss.v008.i06)

[47] Recommenderlab source listing, RECOM_IBCF.R. [Online]. Available: https://rdrr.io/cran/recommenderlab/src/R/RECOM_IBCF.R (last accessed 30/08/2023)

[48] RDocumentation, “unlist() function”. [Online]. Available: https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/unlist (last accessed 30/08/2023)

[49] RDocumentation, “duplicated: Determine Duplicated Elements”. [Online]. Available: https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/duplicated (last accessed 30/08/2023)

[50] Recommenderlab source listing, evaluate.R. [Online]. Available: https://rdrr.io/cran/recommenderlab/src/R/evaluate.R (last accessed 30/08/2023)

[51] Recommenderlab source listing, calcPredictionAccuracy.R. [Online]. Available: https://rdrr.io/cran/recommenderlab/src/R/evaluate.R (last accessed 30/08/2023)

[52] Google Developers, “Classification: ROC Curve and AUC”. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (last accessed 30/08/2023)

[53] Andrew P. Bradley, “The use of area under the ROC curve in the evaluation of machine learning algorithms”, Pattern Recognition, vol. 30, July 1997, pp. 1145-1159 (doi: https://doi.org/10.1016/S0031-3203(96)00142-2)

[54] Scikit-learn, “Precision-Recall”. [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html (last accessed 30/08/2023)

[55] Datacamp, “Precision-Recall Curve in Python Tutorial”, Jan. 2023. [Online]. Available: https://www.datacamp.com/tutorial/precision-recall-curve-tutorial (last accessed 30/08/2023)

[56] Kendrick Boyd, Kevin H. Eng, C. David Page, “Area Under the Precision-Recall Curve: Point Estimates and Confidence Intervals”, Machine Learning and Knowledge Discovery in Databases, 2013, pp. 451-466 (doi: http://dx.doi.org/10.1007/978-3-642-40994-3_55)