Πανεπιστήμιο Πειραιώς

Τμήμα Ψηφιακών Συστημάτων

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

«Ψηφιακά Συστήματα και Υπηρεσίες»

Κατεύθυνση: Μεγάλα Δεδομένα και Αναλυτική

Μεταπτυχιακή Διπλωματική Εργασία :

«Μελέτη συστήματος συστάσεων βασιζόμενο στην έκπληξη των χρηστών»

Αθανάσιος Κομνός - Α.Μ.: ΜΕ1715

Επιβλέπων : Μαρία Χαλκίδη, Αναπληρώτρια Καθηγήτρια

Πειραιάς 2020

Αφιέρωση

Η διπλωματική εργασία αφιερώνεται στη μητέρα μου που δεν είναι πια μαζί μας...

Περίληψη

Οι μηχανές αναζήτησης και πλήθος άλλων εφαρμογών παρέχουν συστάσεις (recommendations) με βάση προτιμήσεις και συσχετίσεις χρηστών. Η μεροληψία (Bias) και η δικαιοσύνη (fairness) στις τεχνικές μηχανικής μάθησης είναι θέματα που έχουν προκαλέσει το ενδιαφέρον των ερευνητών.

Η καλή ακρίβεια πρόβλεψης από μόνη της δεν εγγυάται στους χρήστες μία αποτελεσματική και ικανοποιητική εμπειρία. Προς αυτήν την κατεύθυνση, ένας άλλος παράγοντας ο οποίος φαίνεται να παίζει σημαντικό ρόλο στο κατά πόσο οι χρήστες εκτιμούν το σύστημα συστάσεων είναι το serendipity (απρόσμενα ευχάριστη έκπληξη).

Στα πλαίσια της εργασίας θα μελετήσουμε διαφορετικές προσεγγίσεις δικαιοσύνης καθώς και την εφαρμογή μίας πολιτικής δικαιοσύνης στα συστήματα συστάσεων. Η δικαιοσύνη μπορεί να αφορά στη δίκαιη αντιμετώπιση των χρηστών από το σύστημα σε σχέση με την ποιότητα συστάσεων που παρέχει ή/και στη δίκαιη κατανομή των προτεινόμενων αντικειμένων (να μην υπάρχει μεροληψία σε ομάδες αντικειμένων που προτείνονται).

Ο στόχος της εργασία είναι η σχεδίαση και υλοποίηση προσέγγισης που θα λαμβάνει υπόψη τη δικαιοσύνη στη διαδικασία συστάσεων καθώς και το serendipity . Πιο συγκεκριμένα το σύστημα θα υπολογίζει το serendipity για νέες ταινίες που δεν έχει δει ο χρήστης και θα κάνει τις ανάλογες προτάσεις. Όσον αφορά τη δικαιοσύνη, οι ταινίες που θα ελέγχει και θα προτείνει το σύστημα είναι από διαφορετικές κατηγορίες και όχι μόνο από αυτές που ο χρήστης συνηθίζει να βλέπει.

Abstract

Search engines and a host of other applications provide recommendations based on user preferences and correlations. Bias and fairness in machine learning techniques are issues that have aroused the interest of researchers.

Good prediction accuracy alone does not guarantee users an effective and satisfying experience. In this regard, another factor that seems to play an important role in whether users value the recommendation system is serendipity (unexpectedly pleasant surprise).

In this postgraduate thesis we will study different approaches to fairness as well as the implementation of a political fairness in recommendation systems. Fairness may be related to the fair treatment of users by the system in relation to the quality of the recommendations it provides and / or the fair distribution of the proposed objects (no bias in the groups of objects proposed).

The aim of the postgraduate thesis is to design and implement an approach that takes into account justice in the recommendation process as well as serendipity. In particular, the system will calculate the serendipity for new movies that the user has not seen and make the appropriate suggestions. In terms of fairness, the movies that the system will monitor and recommend are of different categories and not just those that the user is used to watching.

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω την κα. Χαλκίδη Μαρία, καθηγήτρια του τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς. Αρχικά, για την επίβλεψη και τη πολύτιμη βοήθεια της , με πολύτιμες παρατηρήσεις και συμβουλές καθώς και για την υπομονή που έδειξε καθ' όλη τη διάρκεια της διπλωματικής μου εργασίας μέχρι την ολοκλήρωσή της. Ευχαριστώ πολύ και τους υπόλοιπους καθηγητές του τμήματος για τη βοήθεια που μου προσέφεραν και για τις χρήσιμες παρατηρήσεις τους, καθ' όλη τη διάρκεια του μεταπτυχιακού. Τέλος θα ήθελα να εκφράσω την ευγνωμοσύνη μου στους γονείς μου και στην αδελφή μου που με στήριξαν και με βοήθησαν σε όλη τη διάρκεια των μεταπτυχιακών σπουδών μου καθώς και στην σύντροφό μου που παρέμεινε στο πλευρό μου διακριτικά και υπομονετικά στηρίζοντας με αγάπη την προσπάθεια μου αυτή.

Περιεχόμενα

Πίνακας Εικόνων/Σχημάτων

Πίνακας Πινάκων

1. Εισαγωγή στα συστήματα συστάσεων

1.1 Συστήματα συστάσεων

Η ανάπτυξη των συστημάτων συστάσεων ξεκίνησε από την απλή παρατήρηση ότι ο κάθε άνθρωπος συχνά βασίζει μεγάλο μέρος της καθημερινότητάς του και διάφορες αποφάσεις που παίρνει σε προτάσεις που λαμβάνει από τον περίγυρό του. Ο αυξανόμενος όγκος online πληροφοριών λόγω της ραγδαίας εξέλιξης των ηλεκρονικών εφαρμογών και υπηρεσιών έχει δημιουργήσει την ανάγκη ύπαρξης προσεγγίσεων οι οποίες βοηθούν στην εξυπηρέτησή του.

Τα συστήματα συστάσεων είναι εργαλεία λογισμικού που παρέχουν προσωποποιημένες συστάσεις στους χρήστες που πρόκειται να λάβουν αποφάσεις για αγορές ακόμη και για ταινίες που θέλουν να δουν [1].Αντικείμενο ορίζουμε κάτι που προτείνει το σύστημα στους χρήστες. Το σύστημα συστάσεως επικεντρώνεται σε ένα είδος αντικειμένων , όπως είναι λόγου χάρη οι ταινίες και έχει ως στόχο την παροχή χρήσιμων συστάσεων γ'αυτό το αντικείμενο. Για την επίτευξη του στόχου το σύστημα συστάσεως θα πρέπει να επικεντρωθεί στα ενδιαφέροντα αντικείμενα για τον εκάστοτε χρήστη.Προκειμένου να γίνει αυτή η διαδικασία το σύστημα συστάσεως θα πρέπει να προβλέψει τη χρησιμότητα των αντικειμένων και στην συνέχεια να τα συγκρίνει. Αφού ολοκληρωθεί αυτή η διαδικασία θα είναι σε θέση να προτείνει στο χρήστη το αντικείμενο ενδιαφέροντος βάσει σύγκρισης.

Το πρόβλημα των συστάσεων ορίζεται ως η εκτίμηση της απόκρισης του χρήστη για νέα αντικείμενα με βάση παλαιότερες πληροφορίες που διαθέτει το σύστημα. Αυτό καθιστά την προβλεπόμενη απόκριση για το χρήστη υψιλή.Το είδος των αποκρίσεων των χρηστών διαφέρει από εφαρμογή σε εφαρμογή πχ η απόκριση του χρήστη μπορεί να είναι ενδιαφέρομαι/δεν ενδιαφέρομαι, μου αρέσει/δεν μου αρέσει,ή διαφορετικά να υπάρχει κλίμακα βαθμολόγησης με κατώτερη βαθμολογία το 1 και υψηλότερη βαθμολογία το 5 ως άριστα.Εαν λάβουμε την απόκριση του χρήστη ως βαθμολογία μπορούμε να δώσουμε το εξής παράδειγμα: ένας χρήστης παρακολουθεί ταινίες στο Netflix και με βάση αυτή βαθμολογεί με αστέρια την εκάστοτε ταινία.Όμως η άμεση ανατροφοδότηση (feedback) δεν είναι πάντα διαθέσιμη και έτσι η διαδικασία γίνεται λίγο πιο δύσκολη.Εκεί τα συστήματα προσπαθούν να συμπεράνουν τις προτιμήσεις των χρηστών από παράπλευρες πληροφορίες, όπως είναι το ιστορικό αγορών ή το ιστορικό περιήγησης.

Τα δύο πιο βασικά είδη συστημάτων συστάσεως είναι τα Συστήματα Συνεργατικής Διήθησης (collaborative filtering CF) και Διήθησης με Βάση το περιεχόμενο (Content –based filtering).Οι πιο διάσημες είναι οι CF. Σύμφωνα με αυτήν τα αντικείμενα που προτείνονται στο χρήστη είναι αυτά που άρεσαν στο παρελθόν σε χρήστες με παρόμοιες προτιμήσεις.Αυτό προκύπτει από την ομοιότητα των βαθμολογιών των χρηστών.

Στις τεχνικές διήθησης με βάση το περιεχόμενο το σύστημα συστάσεως προτείνει στους χρήστες αντικείμενα τα οποία είναι παρόμοια με αυτά που του άρεσαν στο παρελθόν.Στόχος αυτής της διαδικασίας είναι η αναγνώριση των κοινών χαρακτηριστικών που ο χρήστης έχει βαθμολογήσει θετικά και του προτείνει ένα νέο αντικείμενο με τα ίδια χαρακτηριστικά. Ένα παράδειγμα γ'αυτή την περίπτωση είναι το εξής: ένας χρήστης έχει βαθμολογήσει θετικά μία ταινία δράματος , άρα το σύστημα θα του προτείνει και άλλες

ταινίες δράματος.Στην ουσία εξετάζει το προφίλ του κάθε χρήστη , το οποίο φανερώνει και τις προτιμήσεις του για κάποιο αντικείμενο και του προτείνει κάτι αντίστοιχο [2].

Όμως τα συστήματα συστάσεως που βασίζονται μόνο στο περιεχόμενο έχουν να αντιμετωπίσουν και προβλήματα , δηλαδή περιορισμένη ανάλυση περιεχομένου καθώς και την υπερ- εξειδίκευση [3].Η περιορισμένη ανάλυση περιεχομένου οδηγεί με τη σειρά της στην έλλειψη πληροφοριών.Λόγου χάρη τα ζητήματα προσωπικών δεδομένων μπορεί να προβληματίζουν έναν χρήστη και το αποτέλεσμα που προκύπτει είναι πως αυτός ο χρήστης δεν θα παρέχει προσωπικές πληροφορίες. Το ακριβές περιεχόμενο των αντικειμένων μπορεί να είναι δύσκολο ή κοστοβόρο να αποκτηθει πχ (μουσική , φωτογραφίες). Επίσης δεν μπορεί να καθοριστεί η ποιότητά του.Ενδεχομένως να καθίσταται αδύνατος ο διαχωρισμός ανάμεσα σε ένα καλώς γραμμένο και ένα κακώς γραμμένο άρθρο εαν χρησιμοποιούνται οι ίδιοι όροι.

Βέβαια, από την άλλη πλευρά η υπερ-εξειδίκευση είναι παρενέργεια του τρόπου με τον οποίο τα συστήματα με βάση το περιεχόμενο συστήνουν στο χρήστη νέα αντικείμενα ορμώμενα από την υψιλή βαθμολογία και τις ομοιότητες των αντικειμένων. Για παράδειγμα σε μια εφαρμογή που συστήνει ταινίες το σύστημα θα προτείνει μια ταινία ίδιας κατηγορίας ,ή μια ταινία με ίδιους ηθοποιούς ταινιών που έχει παρακολουθήσει ο χρήστης.Μπορούμε να καταλήξουμε στο εξής: λόγω αυτής της συμπεριφοράς το σύστημα είναι πιθανόν να μην μπορεί να συστήσει αντικείμενα τα οποία είναι διαφορετικά αλλά ταυτόχρονα και ενδιαφέροντα για το χρήστη.Οι λύσεις που έχουν προταθεί γ'αυτό το πρόβλημα περιλαμβάνουν την προσθήκη τυχαιότητας [4] ή το φιλτράρισμα αντικειμένων τα οποία είναι ιδιαίτερα όμοια [5,6].

Τέλος, αξίζει να αναφέρουμε άλλο ένα είδος συστήματος συστάσεως που είναι τα Υβριδικά. Ο πυρήνας του είναι ο συνδιασμός τεχνικών. Ένα υβριδικό σύστημα συνδιάζει δύο διαφορετικές τεχνικές χρησιμοποιώντας τα πλεονεκτήματα της μίας για να επιλύσει τα μειονεκτήματα της άλλης.

1.2 Ανασκόπηση Συστημάτων Συστάσεων

Οι συνηθέστερες τεχνικές που χρησιμοποιούν τα συστήματα συστάσεων μέχρι σήμερα ανήκουν σε δύο κύριες κατηγορίες, το συνεργατικό φιλτράρισμα (collaborative filtering) και το φιλτράρισμα με βάση το περιεχόμενο (content-based filtering). Η διαφοροποίηση των δύο τεχνικών έγκειται στη λογική σύμφωνα με την οποία θα συσταθούν αντικείμενα στο κάθε χρήστη. Πέραν από αυτές τις δυο περιπτώσεις, τα συστήματα συστάσεων μπορούν να κατηγοριοποιηθούν και σε επιπλέον κατηγορίες λόγω κυρίως της διαφορετικής φύσης των προβλημάτων που επιλύουν.

Επίσης υπάρχουν και τα υβριδικά(hybrid) συστήματα συστάσεων. Ένα από τα πρώτα συστήματα συστάσεων που δημιουργήθηκαν ήταν το Fab [7]. Τέθηκε σε λειτουργία τον Δεκέμβριο του 1994 από ερευνητές του Πανεπιστημίου Stanford και παρείχε στους χρήστες του προτάσεις σχετικά με ποιες ιστοσελίδες θα τους ενδιέφερε να επισκεφθούν, με βάση κάποιες βαθμολογίες (ratings) που οι χρήστες είχαν δώσει. Το σύστημα ήταν υβριδικό, αφού συνδύαζε content-based και collaborative filtering.

Στη συνέχεια, παρουσιάζονται αναλυτικά αυτές οι κατηγορίες συστημάτων συστάσεων, για να κατανοηθούν τα πλεονεκτήματα και τα μειονεκτήματα της καθεμιάς.

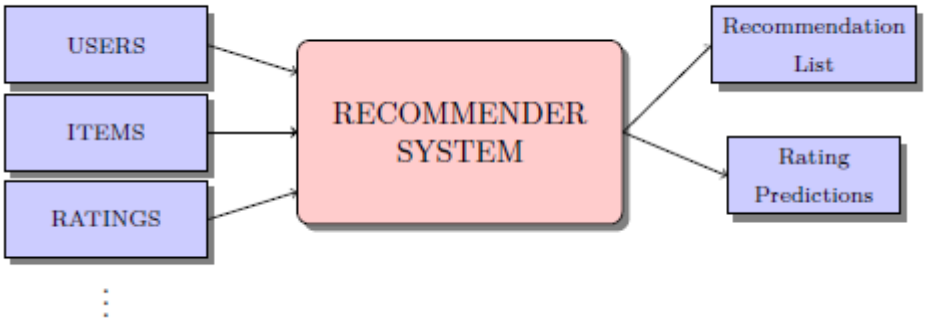
1.3 Συνεργατική Διήθηση ή Συνεργατικό φιλτράρισμα (Collaborative Filtering)

Το συνεργατικό φιλτράρισμα είναι μια δημοφιλής τεχνική σύμφωνα με την οποία οι συστάσεις για κάθε χρήστη προκύπτουν με βάση τις προτιμήσεις άλλων χρηστών με παρόμοια συμπεριφορά. Συνήθως, αυτές οι μέθοδοι δεν χρησιμοποιούν πληροφορίες που έχουν να κάνουν με το περιεχόμενο των αντικειμένων προς σύσταση αυτό καθ' αυτό, αλλά βασίζονται στις γνώμες των χρηστών (συνήθως αξιολογήσεις). Σε ένα τυπικό σύστημα συστάσεων που βασίζεται στο συνεργατικό φιλτράρισμα τα δεδομένα σχετικά με τις προτιμήσεις των χρηστών καταγράφονται σε ένα πίνακα χρηστών-αντικειμένων του οποίου οι γραμμές συνήθως αναφέρονται σε διαφορετικούς χρήστες και κάθε εγγραφή υποδεικνύει τη προτίμηση του χρήστη u για το αντικείμενο i.

Οι τεχνικές που ανήκουν σε αυτή τη κατηγορία λοιπόν χρησιμοποιούν τις βαθμολογήσεις των χρηστών για να καθορίσουν τη σχέση μεταξύ τους και να εξάγουν προβλέψεις σε βαθμολογήσεις που θα έδινε κάθε χρήστης σε κάποιο νέο αντικείμενο. Οι τεχνικές συνεργατικού φιλτραρίσματος χρησιμοποιούνται σε πολλές εφαρμογές συστημάτων συστάσεων. Δημοφιλείς ιστοσελίδες που χρησιμοποιούν τέτοιες τεχνικές είναι οι LinkedIn, Facebook, Twitter και Amazon.

Τα περισσότερα συστήματα συστάσεων υιοθετούν δύο βασικές προσεγγίσεις: συνεργατικό φιλτράρισμα(collaborative filtering) ή φιλτράρισμα με βάση το περιεχόμενο(content-based filtering) . Το συνεργατικό φιλτράρισμα (CF) είναι μία από τις πιο επιτυχημένες προσεγγίσεις για τη δημιουργία συστημάτων συστάσεων [8] .

Αντίθετα με τις προσεγγίσεις με βάση το περιεχόμενο, οι οποίες για να παράξουν συστάσεις χρησιμοποιούν το περιεχόμενο των αντικειμένων εκείνων που έχουν βαθμολογηθεί στο παρελθόν από ένα συγκεκριμένο χρήστη, οι προσεγγίσεις συνεργατικής διήθησης βασίζονται μεν στις βαθμολογίες του συγκεκριμένου χρήστη αλλά βασίζονται και σε αυτές από άλλους χρήστες. Η βασική ιδέα είναι ότι η βαθμολογία ενός χρήστη u για ένα νέο αντικείμενο i είναι πιθανό να είναι παρόμοια με αυτή ενός άλλου χρήστη v, αν οι u και v έχουν βαθμολογήσει άλλα αντικείμενα με παρόμοιο τρόπο. Παρόμοια, ο u είναι πιθανό να βαθμολογήσει δύο αντικείμενα i και j με παρόμοιο τρόπο, αν άλλοι χρήστες έχουν δώσει παρόμοιες βαθμολογίες σε αυτά τα δύο αντικείμενα.



Εικόνα 1 : Παράδειγμα ενός Συστήματος Συστάσεων [9]

Προκειμένου να θεσπιστούν συστάσεις, τα συστήματα CF πρέπει να συνδέσουν δύο βασικές διαφορετικές οντότητες: αντικείμενα και χρήστες. Υπάρχουν δύο βασικές προσεγγίσεις για τη διευκόλυνση μιας τέτοιας σύγκρισης, οι οποίες αποτελούν τις δύο κύριες τεχνικές του CF: η προσέγγιση της γειτνίασης(neighborhood-based) ή αλλιώς προσέγγιση βασισμένη στη μνήμη(memory-based) και η βασισμένη στο μοντέλο(model-based) προσέγγιση.

Συνήθως τα συστήματα που χρησιμοποιούν το συνεργατικό φιλτράρισμα διαχωρίζονται με βάση τους αλγορίθμους που χρησιμοποιούν σε βασιζόμενα στη μνήμη (memory-based) και βασιζόμενα στο μοντέλο (model-based) συστήματα. Τα συστήματα που ανήκουν στη πρώτη κατηγορία εμπεριέχουν αλγορίθμους που χρησιμοποιούν το σύνολο των αντικειμένων που έχει βαθμολογήσει κάθε χρήστης στο παρελθόν για να εξάγουν για αυτόν συστάσεις. Η δεύτερη κατηγορία model-based συστημάτων χρησιμοποιεί αλγόριθμους οι οποίοι εκμεταλλεύονται μόνο ένα μέρος των αντικειμένων που έχει βαθμολογήσει ο κάθε χρήστης για να δημιουργήσουν ένα μοντέλο το οποίο μετά χρησιμοποιείται για να γίνουν προβλέψεις βαθμολογιών του χρήστη σε νέα αντικείμενα.

Οι μέθοδοι γειτνίασης επικεντρώνονται σε τεχνικές βασιζόμενες στο χρήστη (user-based) και βασιζόμενες στο αντικείμενο (item-based). Οι βασιζόμενες στο χρήστη τεχνικές υπολογίζουν ομοιότητα μεταξύ χρηστών, ενώ οι βασιζόμενες στο αντικείμενο τεχνικές υπολογίζουν ομοιότητα μεταξύ αντικειμένων, χρησιμοποιώντας τις βαθμολογήσεις των χρηστών. Με άλλα λόγια οι μέθοδοι γειτνίασης επικεντρώνονται στις σχέσεις μεταξύ χρηστών (user-user CF) ή μεταξύ αντικειμένων (item-item CF).

1.3.1 Μέθοδος user-user CF

Μια προσέγγιση γειτνίασης μεταξύ χρηστών μοντελοποιεί την προτίμηση ενός χρήστη σε ένα αντικείμενο βάσει αξιολογήσεων παρόμοιων χρηστών για το ίδιο αντικείμενο. Το user-user CF είναι μια απλή αλγοριθμική ερμηνεία του CF: βρίσκει άλλους χρήστες των οποίων η προηγούμενη συμπεριφορά αξιολόγησης είναι παρόμοια με αυτή του τρέχοντος χρήστη και χρησιμοποιεί τις αξιολογήσεις του σε άλλα αντικείμενα για να προβλέψει τι επιθυμεί ο τρέχων χρήστης [10] . Τα βασικά συστατικά για το CF είναι: (i) ο πίνακας αξιολογήσεων R ο οποίος καθορίζει το αντικείμενο, τον χρήστη, την αξιολόγηση /

προτίμηση, (ii) μια συνάρτηση ομοιότητας sim (u,υ) μεταξύ των χρηστών u και υ και (iii) μιας μεθόδου για τη χρήση ομοιοτήτων και αξιολογήσεων για τη δημιουργία προβλέψεων. Ο πίνακας αξιολογήσεων R είναι μια είσοδος χρήστη και ένα παράδειγμα εμφανίζεται στον Πίνακα 2.2.1.

Items →

Users ↓	1	2	...	i	...	N_{items}	Average rating
1	3	1					$\bar{r}_1$
2	2				3	4	$\bar{r}_2$
v	5	2		$r_{v,i}$	5	1	$\bar{r}_v$
⋮							⋮
u	3			$r_{u,i}$	5	1	$\bar{r}_u$
⋮							⋮
N_{users}	4			3	1	5	$\bar{r}_{N_{users}}$

Εικόνα 2 : Πίνακας αξιολογήσεων

Η έννοια της ομοιότητας μπορεί να καθοριστεί με πολλούς τρόπους. Τα δύο πιο διαδεδομένα μέτρα ομοιότητας είναι το μέτρο cosine similarity και ο συντελεστής συσχέτισης Pearson [11] ,τα οποία δίδονται από τις εξισώσεις 1.3.1-1 και 1.3.1-2 αντίστοιχα. Όπως φαίνεται στην εξίσωση 1.3.1-1 η τιμή της ομοιότητας μεταξύ δύο χρηστών u,u' εξαρτάται από τις βαθμολογήσεις R(u,i) και R(u',i) που έχουν δώσει οι χρήστες στο παρελθόν,για κάθε αντικείμενο i που ανήκει στο σύνολο των κοινών αντικειμένων I(u,u'). Το σύνολο I(u,u') αναφέρεται σε όλα τα αντικείμενα που έχουν βαθμολογήσει και οι δύο χρήστες.

$$\mathrm{simil}(u,u') = \frac{\sum_{i \in I(u,u')} R(u,i) \cdot R(u',i)}{\sqrt{\sum_{i \in I(u,u')} R(u,i)^2} \cdot \sqrt{\sum_{i \in I(u,u')} R(u',i)^2}}$$

Εξίσωση 1.3.1-1

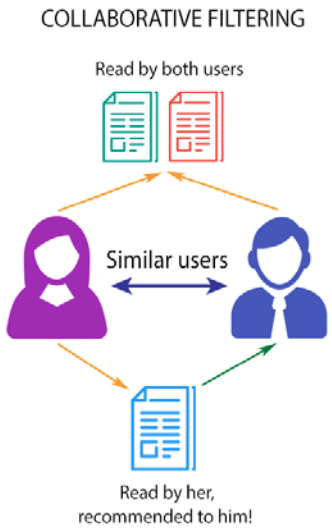
Ομοίως χρησιμοποιώντας το συντελεστή Pearson υπολογίζουμε το simil(u,u') όπως φαίνεται στην εξίσωση 1.3.1-2.

$$simil(u,u') = \frac{\sum_{i \in I(u,u')} (R(u,i) - \bar{R}(u)) \cdot (R(u',i) - \bar{R}(u'))}{\sqrt{\sum_{i \in I(u,u')} (R(u,i) - \bar{R}(u))^2} \cdot \sqrt{\sum_{i \in I(u,u')} (R(u',i) - \bar{R}(u'))^2}}$$

Εξίσωση 1.3.1-2

Το μέτρο ομοιότητας με βάση το συντελεστή του Pearson κυμαίνεται από 1 (θετική συσχέτιση), για χρήστες με απόλυτη συμφωνία, έως -1 (αρνητική συσχέτιση) για χρήστες με απόλυτη διαφωνία.

Με βάση τις τεχνικές του συνεργατικού φιλτραρίσματος και σε συνδυασμό με μεθόδους εξόρυξης γνώσης από δεδομένα, μηχανικής μάθησης (data mining, machine learning) και στατιστικής, έχουν αναπτυχθεί πολλοί αλγόριθμοι συστάσεων.



Εικόνα 3 : Αναπαράσταση ενός συστήματος συνεργατικού φιλτραρίσματος βασιζόμενο στο χρήστη
Πηγή: www.marutitech.com/recommendation-engine-benefits

1.3.2 Μέθοδος item-item CF

Παρόλο που οι τεχνικές φιλτραρίσματος user-user CF είναι δημοφιλείς, προκύπτουν προβλήματα επεκτασιμότητας(scalability) που συνδέονται με τον συχνό υπολογισμό της ομοιότητας μεταξύ των χρηστών. Όταν ένας χρήστης αλλάζει συχνά την αξιολόγηση των αντικειμένων, το διάνυσμα αξιολόγησης ενός τέτοιου χρήστη αλλάζει, το οποίο τροποποιεί την ομοιότητα με τους άλλους. Ως εκ τούτου, η γειτνίαση χρήστη N για έναν συγκεκριμένο χρήστη δεν μπορεί να υπολογιστεί εκ των προτέρων, αλλά πρέπει να υπολογίζεται όποτε χρειάζονται συστάσεις. Αυτό μπορεί να είναι ένα μεγάλο υπολογιστικό εμπόδιο για τα μεγάλα σύνολα δεδομένων. Αυτό το αποτέλεσμα ενισχύεται όταν υπάρχουν περισσότεροι χρήστες από τα αντικείμενα που είναι χαρακτηριστικά για πολλές ιστοσελίδες ηλεκτρονικού εμπορίου.

Για να εξαλειφθεί αυτό το ζήτημα επεκτασιμότητας, προτάθηκε μία μέθοδος item-item CF από την Linden et al. [12] (βλ., [13], [14]). Αυτοί οι αλγόριθμοι αναλύουν την ομοιότητα μεταξύ αντικειμένων αντί να προβλέπουν τις αξιολογήσεις βάσει της ομοιότητας μεταξύ χρηστών. Εάν δύο αντικείμενα τείνουν να έχουν τις προτιμήσεις από τους ίδιους χρήστες, τότε αυτά είναι παρόμοια και οι χρήστες αναμένεται να έχουν παρόμοιες προτιμήσεις για παρόμοια αντικείμενα. Σε συστήματα με υψηλή αναλογία χρήστη προς αντικείμενο, συχνή αλλαγή των αξιολογήσεων ενός στοιχείου από έναν χρήστη είναι απίθανο να αλλάξει τις ομοιότητες μεταξύ αντικειμένων, καθώς κάθε αντικείμενο έχει πολύ περισσότερες αξιολογήσεις από πολλούς χρήστες που δεν αλλάζουν. Ως εκ τούτου, αλλαγή των αξιολογήσεων από ένα πολύ μικρό σύνολο χρηστών θα αλλάξει ελαφρώς την ομοιότητα μεταξύ αντικειμένων [10] και οι χρήστες θα συνεχίσουν να δέχονται καλές συστάσεις. Η μέθοδος item-item CF είναι παρόμοια με τη μέθοδο user-user CF εκτός από το ότι η ομοιότητα ενός αντικειμένου συνάγεται από τα πρότυπα προτιμήσεων χρήστη αντί να εξάγονται από δεδομένα αντικειμένου.

1.3 Μέθοδος model-based

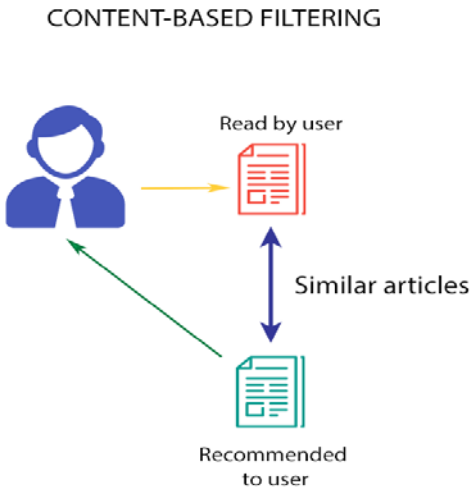
Οι model-based μέθοδοι, προσαρμόζουν ένα παραμετρικό μοντέλο με τα δεδομένα εκπαίδευσης που μπορούν να χρησιμοποιηθούν αργότερα για την πρόβλεψη αξιολογήσεων και συστάσεων. Τα μοντέλα παραγοντοποίησης μήτρας(matrix factorization models) εμφανίστηκαν ως μια μεθοδολογία τελευταίας τεχνολογίας. Στη βασική του μορφή, ο παραγοντισμός της μήτρας χαρακτηρίζει τα αντικείμενα και τους χρήστες ως διανύσματα παραγόντων που συνάγονται από τα πρότυπα αξιολόγησης αντικειμένων. Η μεγάλη αντιστοιχία μεταξύ αντικειμένων και χρηστών οδηγεί σε μια σύσταση. Αυτές οι μέθοδοι έχουν γίνει δημοφιλείς τα τελευταία χρόνια συνδυάζοντας την καλή επεκτασιμότητα με την προβλεπτική ακρίβεια. Επιπλέον, προσφέρουν μεγάλη ευελιξία για τη μοντελοποίηση διαφόρων καταστάσεων της πραγματικής ζωής [15]. Άλλες μέθοδοι περιλαμβάνουν συσταδοποίηση(cluster-based CF) [16], μπεϋζιανούς κατηγοριοποιητές (Bayesian classifiers) [17]και μεθόδους με βάση την παλινδρόμηση(regression-based) [18]. Η μέθοδος slope-one [19] δημιουργεί ένα γραμμικό μοντέλο στη μήτρα αξιολογήσεων, επιτυγχάνοντας γρήγορο υπολογισμό και λογική ακρίβεια.

1.4 Φιλτράρισμα με βάση το περιεχόμενο (Content-based Filtering)

Τα συστήματα που χρησιμοποιούν φιλτράρισμα με βάση το περιεχόμενο αναλύουν ένα σύνολο από πληροφορίες, για τα χαρακτηριστικά των αντικειμένων που έχουν βαθμολογηθεί προηγουμένως από τους χρήστες. Τέτοιες πληροφορίες μπορεί να είναι π.χ. ο τίτλος ή ο σκηνοθέτης σε μία ταινία. Στη συνέχεια χτίζουν ένα μοντέλο ή προφίλ των ενδιαφερόντων του χρήστη που βασίζεται στα χαρακτηριστικά αυτά. Σε πολλές περιπτώσεις το προφίλ αυτό δίδεται άμεσα από το χρήστη. Το προφίλ αποτελεί μία δομημένη παρουσίαση των προτιμήσεων του χρήστη και προσαρμόζεται ανάλογα με τα νέα αντικείμενα για τα οποία δείχνει ενδιαφέρον. Η διαδικασία σύστασης συνίσταται ουσιαστικά στην αντιστοίχιση των χαρακτηριστικών του προφίλ των χρηστών έναντι των χαρακτηριστικών ενός αντικειμένου.

Ανάλογα με το τομέα που εξετάζεται, τα δεδομένα που περιγράφουν τα αντικείμενα μπορεί να έχουν μια συγκεκριμένη δομή, π.χ. για το τομέα των ταινιών μια βάση δεδομένων μπορεί να έχει χαρακτηριστικά όπως σκηνοθέτης,είδος,τίτλος, είτε να βρίσκονται σε μια αδόμητη μορφή. Στη περίπτωση της αδόμητης μορφής των δεδομένων χρησιμοποιούνται τεχνικές επεξεργασίας φυσικής γλώσσας και επεξεργασίας κειμένων (natural language processing , text mining) για την εξαγωγή της απαραίτητης πληροφορίας.

Ο πιο διαδεδομένος τρόπος για να αντιστοιχηθεί μια αριθμητική τιμή σε κάθε αντικείμενο, είναι η τεχνική συχνότητας όρου – αντίστροφης συχνότητας όρου ή tf-idf αλγόριθμος. Ως όρος (term) συνήθως θεωρείται μία λέξη, λέξη-κλειδί ή και μια ολόκληρη πρόταση ανάλογα με την εφαρμογή. Ένα αρχείο για παράδειγμα, μπορεί να αναπαρασταθεί ως διάνυσμα σταθμισμένων όρων. Ο αλγόριθμος tf-idf αποδίδει σε κάθε όρο ενός εγγράφου ένα συντελεστή βάρους ανάλογα με το πόσες φορές εμφανίζεται αυτός στο έγγραφο. Ο συντελεστής βάρους είναι βασικά ένα στατιστικό μέτρο που χρησιμοποιείται για να μετρηθεί πόσο σημαντική είναι μια λέξη μέσα σε ένα έγγραφο.



Εικόνα 4 : Αναπαράσταση ενός συστήματος βασιζόμενο στο φιλτράρισμα με βάση το περιεχόμενο

Πηγή: www.marutitech.com/recommendation-engine-benefits

1.5 Μειονεκτήματα Κάθε Συστήματος

Τόσο τα συστήματα συστάσεων με βάση το περιεχόμενο, όσο και τα συνεργατικά συστήματα συστάσεων έχουν κάποια μειονεκτήματα.

Ένα αμιγώς content-based σύστημα έχει για παράδειγμα το πρόβλημα της υπερ-συγκεκριμενοποίησης. Όταν δηλαδή το σύστημα μπορεί μόνο να προτείνει προϊόντα σχετικά με το προφίλ του χρήστη, ο χρήστης περιορίζεται στο να βλέπει μόνο προϊόντα σχετικά με αυτά που έχει ήδη βαθμολογήσει [20]. Επιπλέον, τέτοια συστήματα απαιτούν να γίνεται εξαγωγή χαρακτηριστικών (feature extraction), πράγμα που πολλές φορές δεν είναι εύκολο, είτε επειδή η εφαρμογή δεν παρέχει προφανή τέτοια χαρακτηριστικά, είτε επειδή για να εξαχθούν απαιτούνται ειδικές και κοστοβόρες τεχνικές. Τέλος, πρόβλημα αποτελεί και η εκμαίευση βαθμολόγησης από τον χρήστη. Συνήθως οι χρήστες δεν είναι πρόθυμοι να βαθμολογούν και η συλλογή βαθμολογιών είναι μια επίπονη διαδικασία. Εφόσον όμως οι βαθμολογίες που δίνουν οι χρήστες είναι ο μόνος παράγοντας που επηρεάζει την απόδοση, η έλλειψη πλήθους βαθμολογιών καθιστά πολύ δύσκολη την ικανοποιητική απόδοση των συστημάτων αυτών [7].

Από την άλλη, ένα συνεργατικό σύστημα συστάσεων, που προτείνει στον χρήστη προϊόντα που άλλοι παρόμοιοι χρήστες έχουν επιλέξει, παρουσιάζει άλλου είδους προβλήματα. Ένα καινούργιο προϊόν καταρχάς δεν μπορεί με κανένα τρόπο να συσταθεί σε χρήστη μέχρις ότου κάποιος χρήστης το βαθμολογήσει ή προσδιορίσει με ποια άλλα προϊόντα είναι παρόμοιο. Έτσι, αν το πλήθος των χρηστών είναι πολύ μικρό σε σχέση με τον όγκο της πληροφορίας που υπάρχει στο σύστημα, τότε υπάρχει ο κίνδυνος ο utility matrix να γίνει υπερβολικά αραιός και τα προϊόντα που έχουν πιθανότητα να συσταθούν να περιοριστούν πάρα πολύ. Άλλο πρόβλημα προκύπτει όταν κάποιος χρήστης δεν έχει ενδιαφέροντα παρόμοια με κάποιον άλλο χρήστη στον πληθυσμό, οπότε σε εκείνον τον χρήστη θα παρέχονται ανεπιτυχείς συστάσεις [7] [20].

1.6 Υβριδικές προσεγγίσεις

Όπως έγινε φανερό προηγουμένως, και τα δύο προαναφερθέντα είδη συστημάτων συστάσεων έχουν το καθένα από μόνο του σημαντικά μειονεκτήματα. Παρόλα αυτά, συστήματα που προσπαθούν να συνδυάσουν τους δύο παραπάνω τύπους καταφέρνουν να ξεπεράσουν αυτά τα προβλήματα σε μεγάλο βαθμό. Τα συστήματα αυτά, που συνδυάζουν content-based και collaborative filtering, λέγονται συστήματα υβριδικών συστάσεων (hybrid recommendation systems).

Στα συστήματα υβριδικών συστάσεων, ένας χρήστης λαμβάνει συστάσεις προϊόντων που είτε βρίσκονται «κοντά» στα ενδιαφέροντα/στις προηγούμενες επιλογές του ίδιου (όπως στο content-based filtering), είτε έχουν λάβει υψηλή βαθμολογία από ένα χρήστη με παρόμοιο προφίλ. Έτσι, αφενός μεν, κάνοντας συνεργατικές συστάσεις χρησιμοποιείται η εμπειρία των άλλων χρηστών και οι συστάσεις δεν περιορίζονται μόνο στα όμοια προϊόντα με αυτά που έχει ήδη διαλέξει ο χρήστης. Αφετέρου δε, κάνοντας συστάσεις με βάση το περιεχόμενο, μπορούν να παρουσιαστούν και προϊόντα που δεν έχουν δει άλλοι χρήστες (γενικά όμοιοι με τον χρήστη στον οποίο γίνονται οι συστάσεις). Τα καινούργια προϊόντα δεν αποτελούν πια πρόβλημα, καθώς λαμβάνονται υπόψη από το content-based σκέλος. Παράλληλα, ένας χρήστης με πολύ ιδιαίτερα ενδιαφέροντα θα μπορεί επίσης να βρει προϊόντα που τον ενδιαφέρουν μέσω του content-based σκέλους [7].

Συνοπτικά, τα συστήματα υβριδικών συστάσεων πλεονεκτούν σε σχέση με τα content-based και collaborative συστήματα όταν το καθένα λειτουργεί μόνο του, γιατί συνδυάζει τα δύο σκέλη, και το ένα σκέλος αντιμετωπίζει τα προβλήματα του άλλου. Αυτός είναι και ο λόγος που στην πράξη τα συστήματα συστάσεων που χρησιμοποιούνται είναι υβριδικά.

1.7 Αξιολόγηση συστημάτων συστάσεων

Το πρόβλημα συστάσεων μπορεί να επιλυθεί είτε με προσδιορισμό μιας βαθμολογίας R(u,i) για κάθε αντικείμενο i που δεν έχει αξιολογήσει ο χρήστης u, είτε με τη κατηγοριοποίηση των αντικειμένων σε ομάδες «προτείνονται» και «δεν προτείνονται».Για την αξιολόγηση λοιπόν των συστημάτων συστάσεων, έχουν προταθεί διάφοροι τρόποι [11], καθώς πλέον τα συστήματα συστάσεων αξιολογούνται με διαφορετικούς τρόπους, ανάλογα με το τομέα και το λόγο για τον οποίο πρόκειται να εφαρμοσθούν. Σε αυτή την ενότητα θα αναλυθούν μέτρα ακρίβειας για την αξιολόγηση των συστημάτων συστάσεων.

1.7.1 Στατιστικά μέτρα ακρίβειας (Statistical accuracy metrics)

Τα μέτρα που ανήκουν σε αυτή τη κατηγορία υπολογίζουν πόσο κοντά είναι η εκτιμώμενη από το σύστημα βαθμολογία R’(u,i) σε σχέση με τη πραγματική βαθμολογία R(u,i). Το πιο διαδεδομένο μέτρο αυτής της κατηγορίας είναι το μέσο απόλυτο σφάλμα (Mean Absolute Error-MAE) , το οποίο υπολογίζει για κάθε αντικείμενο i την απόλυτη

διαφορά της εκτιμώμενης βαθμολογίας από την πραγματική βαθμολογία που έχει εισάγει ο χρήστης και στη συνέχεια τις σταθμίζει ως εξής :

$$\mathrm{MAE_u} = \frac{1}{n}\sum_{i=1}^{n}\left|\mathrm{r_{ui}} - \mathrm{r'_{ui}}\right| \qquad \textbf{1.7.1-1}$$

Όπου n το πλήθος των αντικειμένων που έχει βαθμολογήσει ο χρήστης u. Το μέσο απόλυτο σφάλμα για όλο το σύστημα υπολογίζεται βρίσκοντας το μέσο όρο των MAEu όλων των χρηστών.

Άλλο ένα διαδεδομένο μέτρο αυτής της κατηγορίας είναι η ρίζα του μέσου τετραγωνικού σφάλματος(Root Mean Squared Error- RMSE). Η διαφορά του RMSE και MAE είναι ότι στο MAE όλες οι επιμέρους διαφορές είναι εξίσου σταθμισμένες. Στη περίπτωση του RMSE οι διαφορές υψώνονται στο τετράγωνο προτού βρεθεί ο μέσος όρος, οπότε δίδεται σχετικά μεγαλύτερο βάρος σε μεγάλες διαφορές.

$$\mathrm{RMSE_u} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\mathrm{r_{ui}} - \mathrm{r'_{ui}}\right)^2} \qquad \textbf{1.7.1-2}$$

1.7.2 Μέτρα ακρίβειας βασισμένα στη ποικιλία, κάλυψη και έκπληξη

Στα συστήματα συστάσεων, ο παραδοσιακός στόχος απόδοσης που πρέπει να βελτιστοποιηθεί είναι το μέσο τετραγωνικό σφάλμα (MSE) που μετρά την τετραγωνική διαφορά μεταξύ των προβλεπόμενων βαθμολογιών ενός αλγόριθμου συστάσεων (π.χ. συνεργατικό φιλτράρισμα(CF)) και των πραγματικών αξιολογήσεων που έχουν δώσει οι χρήστες. Το MSE καταγράφει τη συνολική ποιότητα της πρόβλεψης του αλγορίθμου συστάσεων. Ωστόσο η σύσταση στοιχείου ώστε να ελαχιστοποιείται μόνο το MSE, δεν συμπεριλαμβάνει άλλες μετρήσεις απόδοσης όπως η ποικιλία(diversity), η κάλυψη(coverage) και η έκπληξη(serendipity) που έχουν σημαντική επίδραση στην ποιότητα της εμπειρίας των χρηστών. Η ποικιλία, η κάλυψη και η έκπληξη αναγνωρίζονται ως κρίσιμα γεγονότα για την ποιότητα της εμπειρίας των χρηστών σε ένα σύστημα συστάσεων πέρα από την ακρίβεια, με διάφορους ορισμούς και ερμηνείες [26], [27].

Ας υποθέσουμε ένα σύνολο στοιχείων I και ένα σύνολο χρηστών U και υποθέτουμε ότι κάποιο σύστημα συστάσεων βασικής γραμμής (π.χ. στοιχείο-στοιχείο CF) δημιουργεί μια λίστα των συνιστώμενων στοιχείων L_u για κάθε χρήστη u, όπου $|L_u| = L$, με το L να λαμβάνει τιμές, π.χ. 5, 10, 20. Έστω C ένα σύνολο κατηγοριών, όπου κάθε κατηγορία αντιπροσωπεύει διαφορετική οντότητα, π.χ. ένα διαφορετικό εκδότη βιβλίων ή παραγωγό ταινιών, έτσι ώστε κάθε στοιχείο να ανήκει σε ακριβώς μια κατηγορία. Έστω I_c είναι το σύνολο των στοιχείων της κατηγορίας c, για c = 1,. . . , |C|. Για κάθε χρήστη u και κάθε

στοιχείο i μιας κατηγορίας, το r_{iu} υποδηλώνει την προβλεπόμενη βαθμολογία με τον αλγόριθμο βασικού συστήματος συστάσεων (baseline Rs algorithm).

1.7.2.1 Απόκλιση από τις βασικές συστάσεις (baseline recommendations)

Θα θέλαμε να βρούμε μια νέα λίστα των συνιστώμενων αντικειμένων L'_u για κάθε χρήστη u, με $|L'_u| = L$, έτσι ώστε τα στοιχεία κάθε κατηγορίας που συνιστάται στους χρήστες, να πληρούν ορισμένους περιορισμούς.

Έστω U_i το υποσύνολο των χρηστών στο οποίο συνιστάται το στοιχείο i ∈ I_c σύμφωνα με τον αλγόριθμο βασικού συστήματος συστάσεων και $|U_i|$ να είναι η καρδιανότητα του(μοναδικότητα των τιμών). Έστω x = (x_{iu}: i ∈ I, u ∈ U) υποδηλώνει τη πολιτική σύστασης 0-1. Δηλαδή, για κάθε στοιχείο i και χρήστη u η δυαδική μεταβλητή x_{iu}=1 αν το στοιχείο i συνιστάται στο χρήστη u στις νέες λίστες των συνιστώμενων στοιχείων L'_u και x_{iu} = 0 διαφορετικά.

Το κόστος απόκλισης για τα στοιχεία της κατηγορίας c ορίζεται ως η διαφορά της μέσης βαθμολογίας των στοιχείων της κατηγορίας c στις λίστες του αλγόριθμου βασικού συστήματος συστάσεων των συνιστώμενων στοιχείων $\{L_u\}$ και των νέων $\{L'_u\}$,

$$Cost_c(x) = \frac{\sum_{i \in I_c} r_{iu}}{\sum_{i \in I_c} |U_i|} - \frac{\sum_{i \in I} \sum_u r_{iu} x_{iu}}{\sum_{i \in I} \sum_u x_{iu}} \qquad \textbf{1.7.2.1-1}$$

1.7.2.2 Ποικιλία(diversity)

Έστω w_{uv} η ομοιότητα προφίλ μεταξύ ενός ζεύγους χρηστών (u, υ), η οποία μπορεί να υπολογιστεί με μετρήσεις όπως η ομοιότητα συνημιτόνου(cosine similarity) πάνω σε ένα σύνολο χαρακτηριστικών των χρηστών. 'Εστω d_{uv} η ανομοιότητα μεταξύ των χρηστών u και υ , που ορίζεται ως d_{uv} = 1 - w_{uv} , έτσι ώστε 0 ≤ d_{uv} ≤ 2.

Για κάθε στοιχείο i στο σύνολο των στοιχείων I_c της κατηγορίας c, εξετάζουμε το σύνολο των χρηστών στο οποίο το στοιχείο i συνιστάται και υπολογίζουμε ανά ζεύγη τη συνολική ανομοιότητα μεταξύ αυτών των χρηστών, κατά μέσον όρο του αριθμού των ζευγών χρηστών. Έτσι έχουμε το μέσο όρο των στοιχείων στο I_c. Επομένως, η μέση ποικιλία ορίζεται ως εξής:

$$Div_c(x) = \frac{2}{|I_c|} \sum_{i \in I_c} \frac{\sum_{u \in U} \sum_{v \in U : v \neq u} d_{uv} x_{iu} x_{iv}}{(\sum_{u \in U} x_{iu}) \times (\sum_{u \in U} x_{iu} - 1)} \qquad \textbf{1.7.2.1-2}$$

η οποία λαμβάνει επίσης τιμές στο [0, 2].

1.7.2.3 Κάλυψη(coverage)

Μια άλλη απαίτηση είναι ότι ο αλγόριθμος συστάσεων πρέπει να προτείνει στοιχεία από κάθε κατηγορία c σε επαρκή αριθμό των χρηστών, δηλαδή ότι στοιχεία από κάθε κατηγορία εμφανίζονται στις λίστες συστάσεων αρκετών χρηστών. Η κάλυψη για μια κατηγορία c μπορεί να οριστεί με διαφορετικούς τρόπους, π.χ. ο αριθμός των χρηστών του οποίου συνιστώνται στοιχεία της κατηγορίας c, ορίζεται ως

$$Cov_c(x) = \sum_{u \in U} \min\{1, \sum_{i \in I_c} x_{iu}\} \qquad \textbf{1.7.2.1-3}$$

Η κάλυψη της κατηγορίας c είναι ο αριθμός των χρηστών στους οποίους τουλάχιστον ένα στοιχείο της κατηγορίας c συνιστάται. Εάν υπάρχουν περισσότερα από ένα στοιχεία της κατηγορίας c που συνιστώνται σε ένα χρήστη u, δηλαδή αν $\sum_{i \in I_c} x_{iu} > 1$, αυτός ο χρήστης μετράει ως ένας στην κάλυψη. Εμείς υιοθετούμε μια απλούστερη μορφή κάλυψης. Για κάθε στοιχείο i ξεχωριστά, υπολογίζουμε τον αριθμό των χρηστών στους οποίους συνιστάται το στοιχείο. Η μέση κάλυψη χρήστη ανά στοιχείο για στοιχεία της κατηγορίας c είναι

$$Cov_c(x) = \frac{1}{|I_c|} \sum_{i \in I_c} \sum_{u \in U} x_{iu} \qquad \textbf{1.7.2.1-4}$$

1.7.2.3 Έκπληξη(*serendipity*)

Υπάρχουν αρκετοί διαδεδομένοι ορισμοί του serendipity.Εδώ επιλέγουμε μια σειρά χαρακτηριστικών, τα οποία κατά την άποψή μας περιλαμβάνουν καλύτερα το serendipity για ένα στοιχείο και έναν χρήστη:

• Ο βαθμός στον οποίο το στοιχείο διαφέρει από άλλα στοιχεία που ο χρήστης έχει βιώσει μέχρι τότε. Ένα στοιχείο που δεν είναι τόσο παρόμοιο με άλλα στοιχεία που ο χρήστης έχει δει στο παρελθόν, έχει μεγαλύτερες πιθανότητες να εκπλήξει το χρήστη.

• Η δημοτικότητα ενός στοιχείου. Όσο λιγότερο δημοφιλές είναι το στοιχείο, τόσο πιθανότερο είναι να εκπλήξει τον χρήστη.

Υποθέτουμε ότι ένας δείκτης serendipity s_{iu} μπορεί να υπολογιστεί για ένα στοιχείο i και έναν χρήστη u. Στη συνέχεια περιγράφεται πώς γίνεται αυτός ο υπολογισμός.

2. Παρουσίαση προσέγγισης συστάσεων με βάση το serendipity

Το Serendipity έχει οριστεί ως το ατύχημα να βρεθεί κάτι καλό ή χρήσιμο ενώ δεν το αναζητεί κάποιος. Με άλλα λόγια, το serendipity αφορά την καινοτομία των συστάσεων ως προς την θετική έκπληξη των χρηστών στις συστάσεις που τους γίνονται [21]. Στα συστήματα συστάσεων, ορίζεται ως μέτρο που δηλώνει πώς το σύστημα συστάσεων μπορεί να προσφέρει απρόσμενα και χρήσιμα αντικείμενα στους χρήστες. Σε αυτό το κεφάλαιο προτείνουμε και εφαρμόζουμε έναν νέο αλγόριθμο για τη δημιουργία μιας λίστας των αντικειμένων με βάση το serendipity,χρησιμοποιώντας διάφορες τεχνικές.

Ο αλγόριθμος και η υλοποίησή του εξηγούνται λεπτομερώς παρακάτω.

2.1 Ορισμός του προβλήματος

Σε ένα σύστημα συστάσεων ταινιών οι χρήστες συχνά παραπονιούνται για τις συστάσεις ότι είναι βαρετές ή πολύ προφανείς. Με την πάροδο του χρόνου, τέτοια συστήματα οδηγούν τους χρήστες μακριά από αυτά. Αν και η πλειονότητα των προηγούμενων συστημάτων έχει επιτύχει την υψηλότερη δυνατή ακρίβεια, ο χρήστης εξακολουθεί να διαμαρτύρεται για την έλλειψη ποικιλομορφίας. Ωστόσο, το να προτείνουμε απλώς ποικίλες ταινίες στους χρήστες δεν λύνει το πρόβλημα, καθώς μπορεί να τους απομακρύνουμε από τις προτιμήσεις τους.

Όπως προαναφέρθηκε το serendipity εκφράζει το στοιχείο της έκπληξης ενός αντικειμένου που συνιστάται σε ένα χρήστη. Ας υποθέσουμε ότι υπάρχουν μερικά αντικείμενα που θα πρέπει να συνιστώνται για το serendipity π.χ. Ls = 2, 3.

Θα υποθέσουμε C κατηγορίες αντικειμένων και το Ic θα είναι το σύνολο των στοιχείων της κατηγορίας c. Ας υποθέσουμε τώρα ότι έχουμε υπολογίσει κάπως ένα δείκτη Siu για κάθε χρήστη u και αντικείμενο i , ο οποίος υποδηλώνει την έκταση στην οποία το αντικείμενο i θα εκπλήξει το χρήστη u όταν του συστήνεται.Στη συνέχεια, ο στόχος είναι να μεγιστοποιηθεί ο συνολικός μέσος όρος του serendipity σε όλους τους χρήστες όπως φαίνεται στην εξίσωση 2.1-1 ,με βάση τους περιορισμό που τίθεται απο την εξίσωση 2.1-2.

$$\max_{x} \frac{1}{|U|} \sum_{i \in \mathrm{I}} \sum_{u \in \mathrm{U}} s_{iu} \; x_{iu} \qquad \textbf{Εξίσωση 2.1-1}$$

$\sum_{i \in \mathrm{I}} \; x_{iu} = L_s$ για κάθε χρήστη u **Εξίσωση 2.1-2**

2.1.1 Προτεινόμενος υπολογισμός του serendipity

Ο δείκτης serendipity Siu ώστε ένα αντικείμενο i να εκπλήξει τον χρήστη u όταν του προταθεί υπολογίζεται ως εξής. Επιλέγουμε μερικά χαρακτηριστικά που κατά την άποψή μας περιλαμβάνουν το serendipity. Αυτά είναι:

- Ο βαθμός στον οποίο η ταινία διαφέρει από άλλες ταινίες τις οποίες ο χρήστης έχει παρακολουθήσει / αξιολογήσει μέχρι τότε. Μια ταινία που δεν είναι τόσο παρόμοια με άλλες ταινίες που ο χρήστης έχει δει / αξιολογήσει στο παρελθόν έχει περισσότερες πιθανότητες να εκπλήξει το χρήστη.

- Η δημοτικότητα(popularity) ενός αντικειμένου. Όσο λιγότερο δημοφιλές είναι το αντικείμενο, τόσο πιο πιθανό είναι να εκπλαγεί ο χρήστης.

Προκειμένου να υπολογιστεί ο δείκτης του serendipity, χρειαζόμαστε πρώτα ένα σύνολο δεδομένων εκπαίδευσης, με το οποίο εκπαιδεύουμε ένα μοντέλο γραμμικής παλινδρόμησης. Επισκεπτόμαστε τον ιστοχώρο: https://grouplens.org/datasets/serendipity-2018/ και κάνουμε λήψη του συνόλου δεδομένων serendipity-sac2018.zip. Στη συνέχεια διαβάζουμε το αρχείο README.txt που μας δίνει τις απαραίτητες πληροφορίες καθώς και την εργασία [22].

Από το σύνολο δεδομένων serendipity-sac2018.zip, χρειαζόμαστε τα αρχεία movies.csv και answers.csv.

Το αρχείο movies.csv έχει όλα τα αναγνωριστικά(ids) ταινιών, τα διάφορα χαρακτηριστικά(features) των ταινιών και λέξεις-κλειδιά(keywords). Εμείς ενδιαφερόμαστε για τα ακόλουθα χαρακτηριστικά:

- movieId (αναγνωριστικό ταινίας)
- title (τίτλος ταινίας)
- directedBy (σκηνοθέτες)
- starring (πρωταγωνιστές)
- genres (είδη ταινίας)

Από το αρχείο answers.csv μας ενδιαφέρουν τα ακόλουθα:

- userId (481 χρήστες)
- movieId (1678 ταινίες)
- timestamp (δείχνει πότε ο χρήστης έδωσε την βαθμολογία)
- s1 ‘Η πρώτη φορά που άκουσα αυτή την ταινία ήταν όταν το μου την πρότεινε το MovieLens.’
- s2 ‘Το MovieLens με επηρέασε στην απόφασή μου να παρακολουθήσω αυτή την ταινία.’

- s3 ‘Αναμένω να απολαύσω αυτή την ταινία πριν την παρακολουθήσω για πρώτη φορά’
- s4 ‘Αυτός είναι ο τύπος της ταινίας που κανονικά δεν θα ανακάλυπτα από μόνος μου. Χρειάζομαι ένα σύστημα συστάσεων όπως το MovieLens, το οποίο να βρίσκει ταινίες όπως αυτή.’
- s5 ‘Αυτή η ταινία είναι διαφορετική (π.χ. στο στυλ, είδος, θέμα) από τις ταινίες που συνήθως παρακολουθώ.’
- s6 ‘ Ήμουν (ή, θα ήμουν) έκπληκτος που το MovieLens επέλεξε να μου προτείνει αυτή την ταινία’
- s7 ‘Χαίρομαι που είδα αυτή την ταινία.’
- s8 ‘Η παρακολούθηση αυτής της ταινίας διευρύνει τις προτιμήσεις μου. Τώρα με ενδιαφέρει ένα ευρύτερο φάσμα ταινιών. ‘

Τα πεδία s1-s8 αντιστοιχούν στις αξιολογήσεις χρηστών της έρευνας μας.Οι αξιολογήσεις δίνονται χρησιμοποιώντας την κλίμακα, όπου 1 αντιστοιχεί σε «διαφωνώ έντονα», 2 «διαφωνώ», 3 «ούτε συμφωνώ ούτε διαφωνώ», 4 «συμφωνώ», 5 «συμφωνώ απόλυτα», NA «δεν θυμάμαι».

Θα δημιουργήσουμε ένα μοντέλο που θα προβλέψει τον δείκτη του serendipity για οποιαδήποτε ταινία και για οποιοδήποτε χρήστη.

Κοιτάζοντας την κάθε γραμμή του αρχείου answers.csv, θα πρέπει να βρούμε για κάθε ταινία i που έχει δει ένας χρήστης u τα εξής:

- Την ομοιότητα (similarity) Sij μεταξύ αυτής της ταινίας i και των άλλων ταινιών j που ο χρήστης u έχει αξιολογήσει. Οι ταινίες που έχει βαθμολογήσει ο χρήστης u μπορούν επίσης να βρεθούν στο ίδιο αρχείο, answers.csv. Η ομοιότητα μπορεί να βρεθεί με διάφορους τρόπους, όπως (i) (ένας εύκολος τρόπος): χρησιμοποιώντας την ομοιότητα των Jaccard με τα χαρακτηριστικά από το αρχείο movies.csv, όπως οι κοινές λέξεις στα είδη, ίδιοι πρωταγωνιστές, ίδιοι σκηνοθέτες, (ii) (ένας όχι τόσο εύκολος τρόπος): χρησιμοποιώντας το αρχείο ratings.csv (είτε από το σύνολο δεδομένων ml-latest-small.zip είτε από το σύνολο δεδομένων ml-latest.zip. Για την λήψη του συνόλου δεδομένων ml-latest-small.zip ή ml-latest.zip, επισκεπτόμαστε τον ιστοχώρο: https://grouplens.org/datasets/movielens/. Πέρνουμε τον μέσο όρο αυτής της ομοιότητας σε σχέση με τον αριθμό των ταινιών που έχει αξιολογήσει ο χρήστης.

- Τη δημοτικότητα(popularity) ενός αντικειμένου. Αυτή μπορεί να υπολογιστεί μέσω της (κανονικοποιημένης) πιθανότητας όταν μία ταινία αξιολογείται από έναν χρήστη. Αυτό μπορεί να εξαχθεί μέσω του συνόλου δεδομένων ml-latest-small.zip ή από το σύνολο δεδομένων ml-latest.zip . Μπορούμε να δούμε το κλάσμα:

$$p_i = \frac{\text{αριθμός των χρηστών που έχουν βαθμολογήσει το αντικείμενο i}}{\text{αριθμός των χρηστών που έχουν βαθμολογήσει ταινίες}} \quad \textbf{Εξίσωση 2.1.1-1}$$

Στη συνέχεια, μπορούμε να πάρουμε το -$\log_{10} p_i$ ή οποιαδήποτε άλλη μετρική για να έχουμε τη δημοτικότητα.

Προαιρετικά, θα μπορούσατε επίσης να χρησιμοποιήσετε ως τρίτο χαρακτηριστικό την προβλεπόμενη βαθμολογία ενός αντικειμένου (π.χ. η μέση βαθμολογία), η οποία περιλαμβάνεται επίσης στο αρχείο answers.csv.

Να σημειωθεί ότι για να υπολογίσουμε την παραπάνω ομοιότητα, Sij μεταξύ μιας ταινίας i και άλλων ταινιών που ο χρήστης έχει αξιολογήσει, θα μπορούσαμε να χρησιμοποιήσουμε το timestamp. Για μια δεδομένη ταινία i που έχει βαθμολογηθεί από ένα χρήστη, θα μπορούσαμε να υπολογίσουμε την ομοιότητα με τις ταινίες που έχει αξιολογήσει ο ίδιος χρήστης και έχουν την μικρότερη διαφορά timestamp μεταξύ τους.

Το σύνολο δεδομένων εκπαίδευσης θα πρέπει να έχει ως εξής:

Ταινία, ομοιότητα με όλες τις προηγούμενες ταινίες ενός χρήστη, δημοτικότητα, δείκτης serendipity ταινίας(Si).

Εάν χρησιμοποιηθεί η προβλεπόμενη βαθμολογία μιας ταινίας ως τρίτο χαρακτηριστικό, θα υπάρχουν τρία χαρακτηριστικά από πάνω.

Ο δείκτης serendipity μιας ταινίας μπορεί να υπολογιστεί μέσω π.χ. την λήψη του μέσου όρου των απαντήσεων των χρηστών στις ερωτήσεις s5, s6, s7, δηλ.

$$S_i = \frac{1}{3}(\mathrm{s5} + \mathrm{s6} + \mathrm{s7}) \quad \textbf{Εξίσωση 2.1.1-2}$$

Άλλες δυνατότητες για τον προσδιορισμό του δείκτη serendipity είναι δυνατές. Όπως φαίνεται και στη βιβλιογραφία υπάρχουν αρκετοί ορισμοί του serendipity. Με αυτό το σύνολο δεδομένων, μπορούμε να χτίσουμε ένα μοντέλο πρόβλεψης. Οι δυνατότητες που χρησιμοποιεί το μοντέλο μηχανικής μάθησης είναι:

- Μοντέλο γραμμικής παλινδρόμησης(linear regression model) με δύο χαρακτηριστικά, την ομοιότητα(similarity) με προηγούμενες ταινίες που έχει δει ο χρήστης και τη δημοτικότητα(popularity) της ταινίας.

- Μη γραμμικά μοντέλα παλινδρόμησης, με τα δύο παραπάνω χαρακτηριστικά. Για παράδειγμα, θα μπορούσαμε να προσπαθήσουμε να δώσουμε στα δεδομένα μια συνάρτηση της μορφής:

$$serendipity(u,i) = 1 - popularity(i) + \frac{1}{|Iu|}\sum_{j\varepsilon Iu} \mathrm{dist(i,j)} \quad \textbf{Συνάρτηση 2.1.1-3}$$

Όπου Iu τα αντικείμενα που έχει βαθμολογήσει ο χρήστης u και dist(i,j) το similarity μεταξύ των αντικειμένων i και j.

Άλλες επιλογές είναι επίσης δυνατές. Λαμβάνοντας υπόψη αυτό το σύνολο δεδομένων, θα προβλέψουμε για μία ταινία i που δεν την έχει δει ο χρήστης u τον δείκτη serendipity (Siu). Πρέπει να αναφερθεί ότι μία άγνωστη ταινία ή μία ταινία που δεν έχει

προβληθεί i, θα έχει διαφορετικό προβλεπόμενο δείκτη serendipity για κάθε χρήστη u, λόγω του γεγονότος ότι η ταινία έχει διαφορετική ομοιότητα με τις ήδη προβεβλημένες ταινίες αυτού του χρήστη. Θα δοκιμάσουμε το μοντέλο με ένα συγκεκριμένο σετ χρηστών και ταινιών, για να λύσουμε το διατυπωμένο πρόβλημα της εξίσωσης 2.1-1 ,με βάση τον περιορισμό που τίθεται απο την εξίσωση 2.1-2.

Όλα τα παραπάνω είναι βασισμένα στην παρακάτω τεχνική αναφορά [25].

3. Υλοποίηση συστήματος και πειραματική μελέτη

Για να μπορέσουμε να προτείνουμε ταινίες στους χρήστες έγιναν κάποια βήματα. Πιο συγκεκριμένα έπρεπε να φτιαχτεί ένα σύνολο δεδομένων εκπαίδευσης(training dataset) με συγκεκριμένα χαρακτηριστικά. Αρχικά υπολογίστηκε η ομοιότητα νέων ταινιών με βάση τις ταινίες που είχε δει ο κάθε χρήστης και η δημοτικότητα της κάθε ταινίας. Στο επόμενο βήμα έγινε υπολογισμός του serendipity με δύο τρόπους. Στην συνέχεια θα αναλυθούν και οι δύο. Τέλος το σύστημα αποφασίζει ποιές ταινίες θα προτείνει στον χρήστη. Για την τελική αξιολόγηση του συστήματος μπορούμε να δούμε τη βαθμολογία που παίρνουν οι ταινίες αυτές από τους χρήστες. Θα αξιολογήσουμε και τις δύο προσεγγίσεις(δύο τρόποι υπολογισμού του serendipity).

3.1 Λόγοι για την επιλογή της python

Η Python είναι μια γλώσσα υψηλού επιπέδου που έχει εύκολη ανάγνωση του συντακτικού της. Η εγγενής ικανότητα αλληλεπίδρασης με δομές δεδομένων και αντικείμενα με ευρύ φάσμα της ενσωματωμένης λειτουργικότητας διευκολύνει την εγγραφή επιστημονικών προγραμμάτων. Επιπλέον βοηθάει στο να μην χρειάζεται να θυμάται δύσκολα κομμάτια κώδικα ο τελικός χρήστης. Αυτό διευκολύνει τους ερευνητές και τους επιστήμονες να αφιερώνουν περισσότερο χρόνο στην εξερεύνηση διαφόρων ιδεών από το πώς να τις κωδικοποιήσουν. Ο γρήγορος χρόνος ανάπτυξης κώδικα στην Python κάνει πολύ πιο εύκολη την δοκιμή νέων ιδεών με πρωτότυπα.

Ένα παράδειγμα που παρέχεται στο [23] για ένα πρόγραμμα "Hello World " εξηγεί την διαφορά ευκολίας ανάγνωσης μεταξύ της Python και της C ++.

```
/*
A C++ program to print "Hello World"
*/
#include <iostream.h>
void main()
{
cout << "Hello World" << endl;
}
```

Το παραπάνω κομμάτι κώδικα στην Python είναι:

```
print "Hello World"
```

Για τους παραπάνω λόγους, η Python επιλέχθηκε ως η γλώσσα που πρέπει να εφαρμοστεί για τη δημιουργία του serendipity συστήματος συστάσεων στην παρούσα εργασία. Επίσης, η Python είναι δωρεάν και ανοικτού κώδικα.

Το NumPy είναι μια μονάδα επέκτασης(extension module) της Python που παρέχει αποτελεσματική λειτουργία σε συστοιχίες ομοιογενών δεδομένων. Επιτρέπει στην Python να «υπηρετεί» ως γλώσσα υψηλού επιπέδου για χειρισμών αριθμητικών δεδομένων. Τόσο το NumPy όσο και άλλα modules χρησιμοποιήθηκαν στην παρούσα εργασία.

3.2 Περιγραφή πειραματικής μελέτης

Όπως προαναφέρθηκε στο κεφάλαιο 2.1.1 χρησιμοποιήθηκαν διάφορα αρχεία από το σύνολο δεδομένων serendipity-sac2018.zip. Το σύνολο δεδομένων που χρησιμοποιήθηκε στην παρούσα εργασία ήταν τριάντα χρήστες, οι οποίοι είχαν βαθμολογήσει τουλάχιστον πέντε ταινίες. Οι τριάντα χρήστες επιλέχθηκαν από το αρχικό σύνολο δεδομένων, το οποίο είχε όλες τις απαντήσεις των χρηστών(answers.csv).Αρχικά υπολογίζουμε το popularity της κάθε ταινίας και στη συνέχεια το similarity της κάθε ταινίας με βάση τις υπόλοιπες ταινίες που έχει δει ο εκάστοτε χρήστης. Το popularity υπολογίζεται με τον τρόπο που περιγράφτηκε στο κεφάλαιο 2.1.1 και στην εξίσωση 2.1.1-1. Το similarity υπολογίζεται βάσει της εξίσωσης 1.3.1-1. Στη συνέχεια υπολογίζουμε το serendipity της κάθε ταινίας για κάθε χρήστη με βάση τις απαντήσεις(s5-s7) που έχει δώσει στην έρευνα(answers.csv). Με αυτόν τον τρόπο δημιουργήθηκε το αρχικό dataset.

Από το αρχικό dataset βγάλαμε το 60% των ταινιών για κάθε χρήστη και θεωρούμε ότι δεν τις έχει δει. Πιο συγκεκριμένα ο κάθε χρήστης είχε βαθμολογήσει πέντε ταινίες, από αυτές θεωρούμε ότι έχει δει τις δύο και τις άλλες τρεις όχι. Επομένως δημιουργήθηκαν δύο καινούργια datasets. Το training dataset και το unseen_movies. Το unseen_movies έχει τις ταινίες που δεν έχουν δει οι χρήστες και το training dataset τις ταινίες που θεωρούμε πως έχει δει ο εκάστοτε χρήστης. Στο training dataset υπάρχουν επίσης το movieId, το similarity, το popularity και το serendipity.

Με το training dataset θα προβλέψουμε το serendipity για αυτές τις ταινίες. Για παράδειγμα εάν υπάρχουν οι ταινίες 1,2,3,4,5 στο σύστημα και ο χρήστης 1 έχει δει τις 1 και 3 τότε θα χρησιμοποιήσουμε αυτές τις δύο για να υπολογίσουμε το serendipity των ταινιών 2,4,5.

Τέλος το σύστημα θα προτείνει ταινίες στον χρήστη και θα αξιολογήσουμε τις προτάσεις αυτές με βάση τις βαθμολογίες που έχει δώσει. Στόχος της παρούσας εργασίας είναι να πετύχει το σύστημα να δώσει όσο το δυνατόν περισσότερες σωστές συστάσεις-προτάσεις με βάση το serendipity.

3.3 Υπολογισμός ομοιότητας κάθε ταινίας για κάθε χρήστη

Όπως προαναφέρθηκε δημιουργήσαμε ένα αρχείο με τις ταινίες που δεν έχουν δει οι χρήστες. Στο κεφάλαιο πέντε φαίνεται ένα μέρος του.

Επίσης δημιουργήσαμε και ένα δεύτερο αρχείο με τις ταινίες που έχουν δει οι χρήστες και έχει ακριβώς την ίδια δομή με το προηγούμενο. Το μέγεθος του πρώτου αρχείου είναι 90 εγγραφές και του δεύτερου 60. Αυτό προκύπτει γιατί όπως προαναφέρθηκε επιλέξαμε 30 χρήστες που έχουν ψηφίσει από πέντε ταινίες. Θεωρούμε λοιπόν ότι ο κάθε χρήστης έχει δει από δύο ταινίες.

Στη συνέχεια υπολογίζουμε την ομοιότητα της κάθε καινούργιας ταινίας με όλες τις άλλες που έχουν δει οι χρήστες. Ο υπολογισμός της ομοιότητας γίνεται όπως προαναφέρθηκε βάσει της εξίσωσης 1.3.1-1, δηλαδή της cosine ομοιότητας. Η κάθε ταινία έχει διαφορετική ομοιότητα για κάθε χρήστη. Αρχικά χωρίζουμε το αρχείο των ταινιών που έχουν δει οι χρήστες ανά χρήστη και για κάθε ταινία του υπολογίζουμε το similarity της με αυτές που δεν έχει δει, βάσει κάποιων χαρακτηριστικών. Χρησιμοποιούμε την cosine ομοιότητα(cosine similarity) με τα χαρακτηριστικά από το αρχείο movies.csv, όπως οι κοινές λέξεις στα είδη(genres), ίδιοι πρωταγωνιστές(starring), ίδιοι σκηνοθέτες(directedBy). Τέλος αποθηκεύουμε τα αποτελέσματα σε ένα καινούργιο αρχείο για μελλοντική χρήση.

3.4 Υπολογισμός δημοτικότητας

Όπως προαναφέρθηκε στο προηγούμενο κεφάλαιο η δημοτικότητα(popularity) ενός αντικειμένου μπορεί να υπολογιστεί μέσω της (κανονικοποιημένης) πιθανότητας όταν μία ταινία αξιολογείται από έναν χρήστη. Έτσι χρησιμοποιώντας την εξίσωση 2.1.1-1 βρίσκουμε την πιθανότητα και στη συνέχεια, χρησιμοποιούμε το -$\log_{10} p_i$ για να έχουμε τη δημοτικότητα. Ο υπολογισμός της πιθανότητας για κάθε ταινία γίνεται από στοιχεία που εξάγουμε από το αρχείο answers.csv. Πιο συγκεκριμένα βρίσκουμε τον αριθμό των χρηστών που βαθμολόγησαν την κάθε ταινία και τον διαιρούμε με τον αριθμό των χρηστών που έχουν βαθμολογήσει γενικά ταινίες. Με αυτόν τον τρόπο δημιουργήθηκε ένα καινούργιο αρχείο(βλέπε κεφάλαιο 5) που περιέχει τα χαρακτηριστικά: userId , MovieId, similarity , popularity.

3.5 Training dataset

Για να ολοκληρωθεί το training_dataset.csv πρέπει να βρεθεί το serendipity. Ο δείκτης serendipity μιας ταινίας μπορεί να υπολογιστεί μέσω της λήψης του μέσου όρου των απαντήσεων των χρηστών στις ερωτήσεις s5, s6, s7. Οι απαντήσεις βρίσκονται στο αρχείο answers.csv. Χρησιμοποιώντας την εξίσωση 2.1.1-2 γίνεται ο υπολογισμός του serendipity. Το training dataset όπως προαναφέρθηκε αποτελείται από τα εξής χαρακτηριστικά: movieId, το similarity, το popularity και το serendipity. Στην παρακάτω εικόνα φαίνεται ένα μέρος του.

	movield	similarity	Popularity	serendipity
0	108979	0.249055	2.205024	3
1	6947	0.233624	2.682145	4
2	117444	0.252277	2.682145	3
3	150548	0.230435	2.381115	3
4	136542	0.240925	2.682145	3
5	77455	1.000000	2.682145	4
6	1303	0.227817	2.381115	3
7	103306	0.246150	2.682145	2
8	2060	0.213561	2.682145	2
9	135534	0.252540	2.682145	2

Εικόνα 5 : training_dataset.csv

3.6 Υπολογισμός serendipity

Σκοπός μας είναι να υπολογίσουμε το serendipity για όλες τις ταινίες που δεν έχουν δει οι χρήστες, για να μπορέσουμε στη συνέχεια να κάνουμε την σύσταση ταινιών. Ο υπολογισμός του serendipity έγινε με τρεις τρόπους. Στη συνέχεια γίνεται ανάλυση του κάθε τρόπου.

3.6.1 Υπολογισμός με μοντέλο γραμμικής παλινδρόμησης

Η πολλαπλή γραμμική παλινδρόμηση(multiple linear regression) μπορεί να μοντελοποιήσει τη σχέση μεταξύ δύο ή περισσοτέρων χαρακτηριστικών εφαρμόζοντας μια γραμμική εξίσωση στα παρατηρούμενα δεδομένα.

Η απλή γραμμική παλινδρόμηση(simple linear regression) έχει μία εξαρτώμενη και μία ανεξάρτητη μεταβλητή, αλλά στην πολλαπλή γραμμική παλινδρόμηση η εξαρτημένη μεταβλητή είναι μία, αλλά μπορεί να υπάρχουν δύο ή περισσότερες ανεξάρτητες μεταβλητές. Τα βήματα για την εκτέλεση της απλής και της πολλαπλής γραμμικής παλινδρόμησης είναι σχεδόν τα ίδια, η διαφορά έγκειται στην αξιολόγηση(evaluation). Με τη χρήση της αξιολόγησης μπορούμε να διαπιστώσουμε ποιος παράγοντας έχει τον μεγαλύτερο αντίκτυπο στην προβλεπόμενη απόδοση.

Στην περίπτωσή μας χρειαζόμαστε ένα μοντέλο πολλαπλής γραμμικής παλινδρόμησης με δύο χαρακτηριστικά, την ομοιότητα(similarity) με προηγούμενες ταινίες που έχει δει ο χρήστης και τη δημοτικότητα(popularity) της ταινίας.

Ακολουθούν συνοπτικά τα βήματα δημιουργίας του μοντέλου.

Βήμα 1 : Εισαγωγή βιβλιοθηκών

- Το NumPy είναι το θεμελιώδες πακέτο για την επιστημονική πληροφορική.
- Το scikit-learn είναι μια ευρέως χρησιμοποιούμενη βιβλιοθήκη της Python για μηχανική μάθηση, που είναι βασισμένη πάνω στο NumPy και μερικά άλλα πακέτα.
- Το Matplotlib είναι μία βιβλιοθήκη σχεδίασης Python 2D.
- Η βιβλιοθήκη pandas παρέχει εύχρηστες δομές δεδομένων και εργαλεία ανάλυσης δεδομένων για τη γλώσσα προγραμματισμού Python.

Βήμα 2 : Διάβασμα του συνόλου δεδομένων

Σε αυτό το σημείο θα χρησιμοποιήσουμε το training dataset για να εκπαιδεύσουμε το μοντέλο μας, με σκοπό την εύρεση του serendipity για τις ταινίες που δεν έχουν δει οι χρήστες.

Βήμα 3: Διαχωρισμός των χαρακτηριστικών εισόδου και εξόδου από το σύνολο δεδομένων

Στο σύνολο δεδομένων x1 είναι η μεταβλητή εξόδου. Επομένως, θεωρούμε όλες τις στήλες εκτός από τη x1 ως είσοδο X και την 0η στήλη (x1) ως έξοδο Y.

Στη συνάρτηση train_test_split (), χρησιμοποιήσαμε test_size ως 0.2, που σημαίνει ότι χρησιμοποιούμε το 20% των δεδομένων για έλεγχο και το υπόλοιπο 80% για την εκπαίδευση.

Βήμα 4: Εκπαίδευση του μοντέλου

Χρησιμοποιούμε τα X_train και Y_train για να εκπαιδεύσουμε το μοντέλο γραμμικής παλινδρόμησης. Η χρήση του X_test είναι για την πρόβλεψη της εξόδου Y.

Όταν κάνει τις προβλέψεις το μοντέλο μας τις αποθηκεύουμε σε ένα αρχείο. Ενώνωντας(merge) αυτό το αρχείο με το προηγούμενο που είχαμε υπολογίσει το similarity και το popularity παράγουμε το τελικό dataset(βλέπε κεφάλαιο 5) που θα χρησιμοποιήσουμε για να κάνουμε την σύσταση των ταινιών.

3.6.2 Υπολογισμός με βάση τη συνάρτηση εκτίμησης

Σε αυτή την προσέγγιση δημιουργούμε ένα μοντέλο υπολογισμού του serendipity, με βάση δύο χαρακτηριστικά. Την ομοιότητα(similarity) με προηγούμενες ταινίες που έχει δει ο χρήστης και τη δημοτικότητα(popularity) της ταινίας. Ο υπολογισμός του serendipity γίνεται χρησιμοποιώντας την συνάρτηση 2.1.1-3. Σκοπός μας είναι και πάλι η εύρεση του serendipity για το αρχείο με τις ταινίες που δεν έχουν δει οι χρήστες. .

Ακολουθούν συνοπτικά τα βήματα δημιουργίας του μοντέλου και αποσπάσματα του κώδικα υλοποίησης.

Βήμα 1 : Εισαγωγή βιβλιοθηκών και διάβασμα του συνόλου δεδομένων

Βήμα 2 : Δημιουργία μοντέλου

Σε αυτό το σημείο γίνεται η υλοποίηση της συνάρτησης 2.1.1-3.

Όταν κάνει τις προβλέψεις το μοντέλο μας τις αποθηκεύουμε σε ένα αρχείο. Ενώνωντας(merge) αυτό το αρχείο με το προηγούμενο που είχαμε υπολογίσει το similarity και το popularity παράγουμε το τελικό dataset που θα χρησιμοποιήσουμε για να κάνουμε την σύσταση των ταινιών. Η δομή του τελικού dataset είναι η ίδια με αυτή του κεφαλαίου 3.6.1.

3.6.3 Υπολογισμός με βάση τις απαντήσεις των χρηστών

Ο δείκτης serendipity μιας ταινίας μπορεί να υπολογιστεί μέσω της λήψης του μέσου όρου των απαντήσεων των χρηστών στις ερωτήσεις s5, s6, s7. Οι απαντήσεις βρίσκονται στο αρχείο answers.csv. Χρησιμοποιώντας την εξίσωση 2.1.1-2 γίνεται ο υπολογισμός του serendipity.

3.7 Πρόβλεψη ταινιών

Η πρόβλεψη των ταινιών θα γίνει με δύο προσεγγίσεις. Μία με βάση το serendipity που υπολογίστηκε από το γραμμικό μοντέλο παλινδρόμησης και μία με βάση το serendipity που υπολογίστηκε από το μοντέλο της συνάρτησης εκτίμησης. Στη συνέχεια θα παρουσιαστούν τα αποτελέσματα και θα αξιολογηθούν.

Στο σημείο αυτό θα γίνει αναφορά στον τρόπο με τον οποίο έγιναν οι προβλέψεις. Πρέπει να αναφερθεί ότι μία άγνωστη ταινία ή μία ταινία που δεν έχει προβληθεί i, θα έχει διαφορετικό προβλεπόμενο δείκτη serendipity για κάθε χρήστη u, λόγω του γεγονότος ότι η ταινία έχει διαφορετική ομοιότητα με τις ήδη προβεβλημένες ταινίες αυτού του χρήστη. Ο τρόπος με τον οποίο υπολογίζεται το similarity των νέων ταινιών είναι ο ίδιος με αυτόν της παραγράφου 3.2.Θα δοκιμάσουμε το μοντέλο με ένα συγκεκριμένο σετ χρηστών και ταινιών, για να λύσουμε το διατυπωμένο πρόβλημα της εξίσωσης 2.1-1 ,με βάση τον περιορισμό που τίθεται απο την εξισώση 2.1-2.

Το πρόβλημα μας είναι γραμμικό.Ο γραμμικός προγραμματισμός (LP, που ονομάζεται επίσης γραμμική βελτιστοποίηση) είναι μια μέθοδος για την επίτευξη του καλύτερου αποτελέσματος (όπως το μέγιστο κέρδος ή το χαμηλότερο κόστος) σε ένα μαθηματικό μοντέλο των οποίων οι απαιτήσεις αντιπροσωπεύονται από γραμμικές σχέσεις. Ο γραμμικός προγραμματισμός είναι μια ειδική περίπτωση του μαθηματικού προγραμματισμού (επίσης γνωστή ως μαθηματική βελτιστοποίηση).

Πιο τυπικά, ο γραμμικός προγραμματισμός είναι μια τεχνική για τη βελτιστοποίηση μιας γραμμικής αντικειμενικής συνάρτησης, που υπόκειται σε γραμμικούς κανόνες ισότητας και γραμμικών ανισοτήτων [24].

Για την επίλυση τέτοιων προβλημάτων υπάρχουν OR solutions (δηλαδή λύσεις λογισμικού που βασίζονται σε μοντέλα βελτιστοποίησης) για την επίλυση τέτοιων προγραμμάτων. Μια καλή και δημοφιλής γλώσσα προγραμματισμού που συνιστάται από πολλούς στην κοινότητα OR και Data Science είναι η Python. Είναι εύκολη, ευέλικτη και ισχυρή και διαθέτει μεγάλες βιβλιοθήκες για τη Μηχανική Μάθηση, Βελτιστοποίηση και Στατιστική Μοντελοποίηση..

Πολλοί λύτες βελτιστοποίησης (optimization solvers) έχουν διεπαφές Python για τη μοντελοποίηση LPs.Υπάρχουν αρκετοί optimization solvers, όπως ο CPLEX και το Gurobi. Στην παρούσα εργασία χρησιμοποιήθηκε το PuLP, το οποίο είναι ένα ισχυρό πακέτο μοντελοποίησης ανοικτού κώδικα στην Python.

Σε αυτό το σημείο θα δούμε τη μοντελοποίηση βελτιστοποίησης με PuLP. Ως μια γρήγορη ανασκόπηση, ένα μοντέλο βελτιστοποίησης είναι ένα πρόβλημα το οποίο έχει έναν στόχο (ή ένα σύνολο στόχων στο πολυ-αντικειμενικό προγραμματισμό), ένα σύνολο περιορισμών και ένα σύνολο μεταβλητών απόφασης.

Οι εξισώσεις 2.1-1 και 2.1-2 μας δίνουν ένα μοντέλο βελτιστοποίησης. Ακολουθούν συνοπτικά τα βήματα δημιουργίας του μοντέλου.

Βήμα 1 : Εισαγωγή βιβλιοθηκών και διάβασμα του συνόλου δεδομένων

Βήμα 2 : Ορισμός παραμέτρων

Σε αυτό το σημείο ορίζουμε τις απαραίτητες παραμέτρους με βάση το πρόβλημά μας.

Βήμα 3 : Δημιουργία μοντέλου

Δημιουργούμε το μοντέλο μας στη Python. Όταν θέλουμε να προγραμματίσουμε ένα μοντέλο βελτιστοποίησης, τοποθετήσαμε ένα placeholder για αυτό το μοντέλο (όπως έναν κενό καμβά) και στη συνέχεια προσθέτουμε τα στοιχεία του (μεταβλητές απόφασης και περιορισμούς) σε αυτό.

Μετά από αυτό το βήμα, έχουμε ένα μοντέλο αντικειμένου με το όνομα opt_model. Στη συνέχεια, πρέπει να προσθέσουμε μεταβλητές απόφασης. Είναι στάνταρ η αποθήκευση μεταβλητών απόφασης σε λεξικά Python (ή Pandas Series) όπου τα λεξικά κλειδιών (dictionary keys) είναι μεταβλητές απόφασης και οι τιμές είναι αντικείμενα μεταβλητής απόφασης. Μια μεταβλητή απόφασης ορίζεται από τρεις κύριες ιδιότητες: τον τύπο της (συνεχής, δυαδική ή ακέραιος), το κατώτατο όριο της (0 από προεπιλογή) και το ανώτερο όριο (το άπειρο από προεπιλογή).

Στη δική μας περίπτωση η μεταβλητή απόφασης είναι δυαδική.

Βήμα 4 : Ορισμός μεταβλητών

Μετά τη ρύθμιση των μεταβλητών απόφασης και την προσθήκη τους στο μοντέλο μας, θέτουμε τους περιορισμούς. Σε αυτό το σημείο γίνεται και η δήλωση των εξισώσεων 2.1-1 και 2.1-2.

Βήμα 5 : Ορισμός περιορισμών

Το επόμενο βήμα είναι ο ορισμός ενός στόχου(objective), ο οποίος είναι μια γραμμική έκφραση.

Βήμα 6 : Ορισμός στόχου για maximize

Βήμα 7: Κάλεσμα Solver

Τέλος, καλούμε τον solver να λύσει το μοντέλο βελτιστοποίησης. Στο PuLP, ο προεπιλεγμένος solver είναι ο CBC, αλλά μπορεί να λειτουργήσει και με άλλους solvers. Αυτό είναι το τελευταίο βήμα για την επίλυση του μοντέλου μας.

Όταν λυθεί το πρόβλημα αποθηκεύουμε τις τιμές που το βελτιστοποιούν. Στην περίπτωση μας οι ταινίες με το υψηλότερο serendipity ανά χρήστη. Καλούμε τον solver να λύσει το πρόβλημα και με τις δύο προσεγγίσεις όπως προαναφέρθηκε. Επιπλέον και για τις δύο προσεγγίσεις δοκιμάζουμε να προτείνουμε στον χρήστη δύο και στη συνέχεια τρία αντικείμενα(ταινίες). Αποθηκεύουμε όλα τα δεδομένα για να μπορούμε στη συνέχεια να συγκρίνουμε τις προτάσεις με τις βαθμολογίες των χρηστών. Με αυτόν τον τρόπο θα κάνουμε την τελική αξιολόγηση του συστήματος μας.

Σε αυτό το σημείο αξίζει να αναφερθεί ένα παράδειγμα, το οποίο θα συνοψίσει όλα τα παραπάνω και θα κάνει εύκολα κατανοητή τη λύση του προβλήματος. Έστω ότι έχουμε το παρκάτω dataset:

user	item	serendipity
1	1	3
1	2	4
1	3	2
2	5	2
2	4	3
2	2	1
3	1	3
3	4	2
3	3	5

Πίνακας 1 : dataset επεξήγησης προβλήματος

Όπως προαναφέρθηκε θέλουμε να κάνουμε maximize την εξίσωση 2.1-1. Για να γίνει αυτό θα πρέπει οι προτάσεις στους χρήστες(Xiu) να είναι αυτές με το μεγαλύτερο serendipity. Να υπενθυμίσουμε ότι όταν προτείνουμε ένα αντικείμενο στον χρήστη το Xiu

παίρνει την τιμή ένα, αλλιώς την τιμή μηδέν. Επομένως αν είχαμε τον περιορισμό της πρότασης μίας μόνο ταινίας ανά χρήστη, η εξίσωση 2.1-1 θα έπαιρνε την μέγιστη τιμή όταν στον χρήστη ένα θα προτείναμε το αντικείμενο δύο, στον χρήστη δύο το αντικείμενο τέσσερα και στον χρήστη τρία το αντικείμενο τρία. Αυτό γίνεται εύκολα κατανοητό αν σκεφτούμε ότι αν δεν είχαμε κανένα περιορισμό το πρόβλημα που θα είχαμε να επιλύσουμε θα ήταν το εξής: $3\chi_1$+$4\chi_1$+$2\chi_1$+$2\chi_2$+$3\chi_2$+$1\chi_2$+$3\chi_3$+$2\chi_3$+$5\chi_3$. Οπότε αν προτείναμε όλα τα αντικείμενα(χ=1) στους χρήστες θα είχαμε την μέγιστη τιμή του αθροίσματος γινομένων. Σε περίπτωση περιορισμών όπως πριν γίνονται οι ανάλογες προτάσεις αντικειμένων στους χρήστες, έχοντας υπόψιν την μεγιστοποίηση της εξίσωσης. Πιο συγκεκριμένα αν είχαμε τον περιορισμό δύο προτάσεων αντικειμένων στους χρήστες , θα προτείναμε στον χρήστη ένα τα αντικείμενα δύο και ένα, στον χρήστη δύο τα αντικείμενα τέσσερα και πέντε και στον χρήστη τρία τα αντικείμενα τρία και ένα.

3.8 Αποτελέσματα και αξιολόγηση

Στο κεφάλαιο αυτό θα γίνει αναλυτική παρουσίαση των αποτελεσμάτων καθώς και η τελική αξιολόγηση. Όπως προαναφέρθηκε υλοποιήθηκαν δύο προσεγγίσεις. Η πρώτη είναι με βάση το serendipity που υπολογίστηκε από το γραμμικό μοντέλο παλινδρόμησης και η δεύτερη με βάση το serendipity που υπολογίστηκε από το μοντέλο της συνάρτησης εκτίμησης . Επίσης όπως προαναφέρθηκε στο κεφάλαιο 2.1 υποθέσαμε ότι υπάρχουν μερικά αντικείμενα που θα πρέπει να συνιστώνται για το serendipity ανά χρήστη π.χ. Ls = 2, 3. Οπότε θα δούμε τα αποτελέσματα που είχαμε όταν προτείναμε δύο ταινίες και τρεις ταινίες αντίστοιχα σε κάθε χρήστη.

Όπως προαναφέρθηκε ο αριθμός των χρηστών που συμμετέχουν στο συγκεκριμένο πείραμα είναι τριάντα και ο αριθμός των ταινιών ενενήντα.

Η τελική αξιολόγηση θα γίνει με βάση τις βαθμολογίες που έχουν δώσει οι χρήστες. Οι βαθμολογίες των χρηστών κυμαίνονται με βάση τη κλίμακα βαθμολόγησης με κατώτερη βαθμολογία το 1 και υψηλότερη βαθμολογία το 5 ως άριστα. Επομένως για να δούμε αν τελικά οι προτάσεις ικανοποίησαν τους χρήστες, θα εξετάσουμε πόσες από τις προτάσεις του συστήματος έχουν βαθμολογηθεί με βαθμό από τρια και πάνω. Πιο συγκεκριμένα για να υπολογίσουμε το ποσοστό επιτυχίας προβλέψεων του συστήματος μας, θα πρέπει να δούμε πόσες από τις προτάσεις ικανοποιούν την συνθήκη μας. Για παράδειγμα αν έγιναν πενήντα προτάσεις στους χρήστες και οι σαράντα ικανοποιούν την συνθήκη που ορίσαμε(έστω βαθμολογίες >=3) το ποσοστό επιτυχίας είναι 80%. Αυτό υπολογίζεται με βάση τον ακόλουθο τύπο(3.8-1):

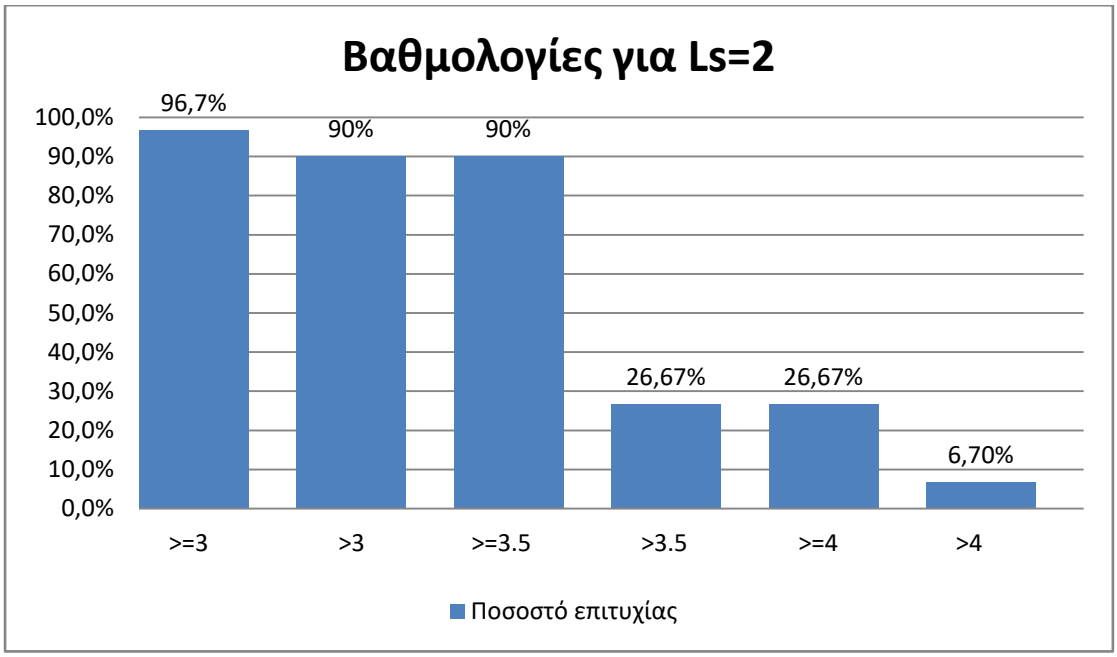
$$Ποσοστό\ επιτυχίας = \frac{αριθμός\ προτάσεων\ που\ ικανοποιούν\ την\ συνθήκη}{συνολικός\ αριθμός\ προτάσεων} * 100 \qquad \textbf{3.8-1}$$

Το ποσοστό επιτυχίας υπολογίζεται στο σύνολο των χρηστών και όχι ανά χρήστη. Πιο συγκεκριμένα αν έχω N χρήστες και έχω προτείνει σε κάθε χρήστη 2 ταινίες, ο παρανομαστής(συνολικός αριθμός προτάσεων) του τύπου 3.8-1 θα είναι 2×N. Αντίστοιχα αν προτείνω τρεις ταινίες θα είναι 3×N.

Στο σημείο αυτό να αναφερθεί ότι η ρίζα του μέσου τετραγωνικού σφάλματος(Root Mean Squared Error- RMSE) εκτίμησης του serendipity από το γραμμικό μοντέλο παλινδρόμησης ήταν : 0,38. Όταν το μήδεν είναι το τέλειο και ένα το χειρότερο.

3.8.1 Αποτελέσματα solver από serendipity γραμμικού μοντέλου παλινδρόμησης

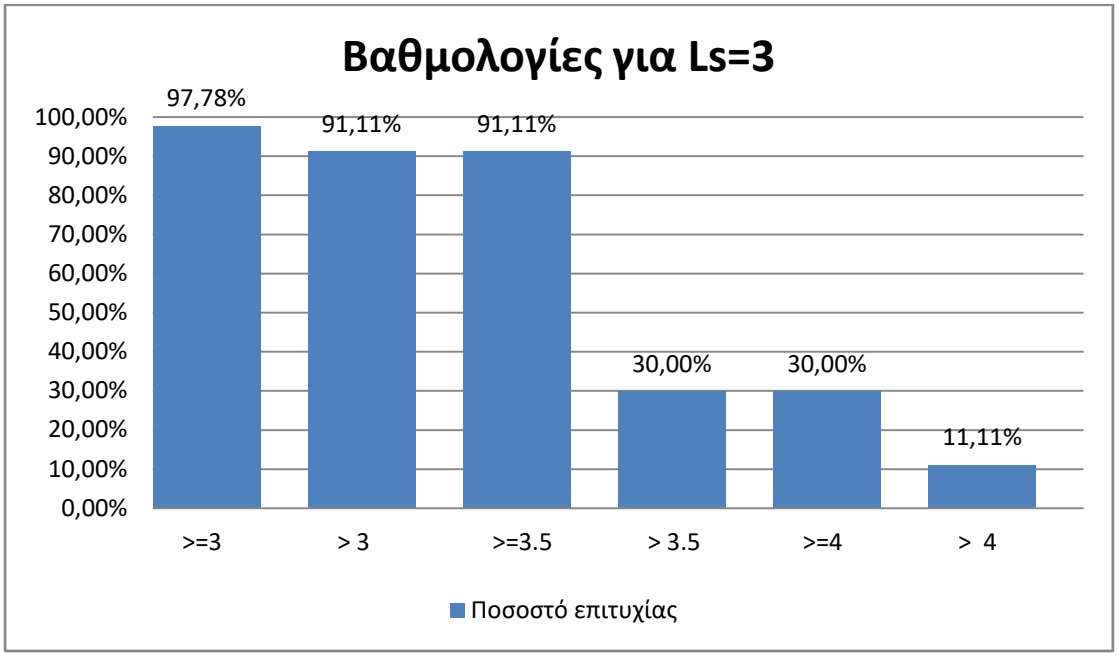
Στο παρακάτω διάγραμμα φαίνονται τα αποτελέσματα όταν το μοντέλο προτείνει δύο ταινίες σε κάθε χρήστη(Ls=2). Πιο συγκεκριμένα τα ποσοστά επιτυχούς πρόβλεψης ανάλογα με τη βαθμολογία.



Εικόνα 6 : Αποτελέσματα solver από serendipity γραμμικού μοντέλου παλινδρόμησης για Ls=2

Όπως φαίνεται στο παραπάνω διάγραμμα το ποσοστό επιτυχίας ανέρχεται στο 96,7%, όταν προτείνονται ταινίες που έχουν πάρει βαθμολογία μεγαλύτερη ή ίση από τρια. Για ταινίες που έχουν βαθμολογηθεί παραπάνω από τρια ή με βαθμολογία μεγαλύτερη ή ίση του τρεισήμισι το ποσοστό φτάνει το 90%. Αντίστοιχα για ταινίες που έχουν βαθμολογηθεί παραπάνω από τρεισήμισι, παραπάνω απο τέσσερα ή και με τέσσερα το ποσοστό φτάνει το 26,67%. Τέλος για ταινίες με βαθμολογία πάνω απο τέσσερα το ποσοστό είναι 6,7%.

Στο παρακάτω διάγραμμα φαίνονται τα αποτελέσματα όταν το μοντέλο προτείνει τρεις ταινίες σε κάθε χρήστη(Ls=3).

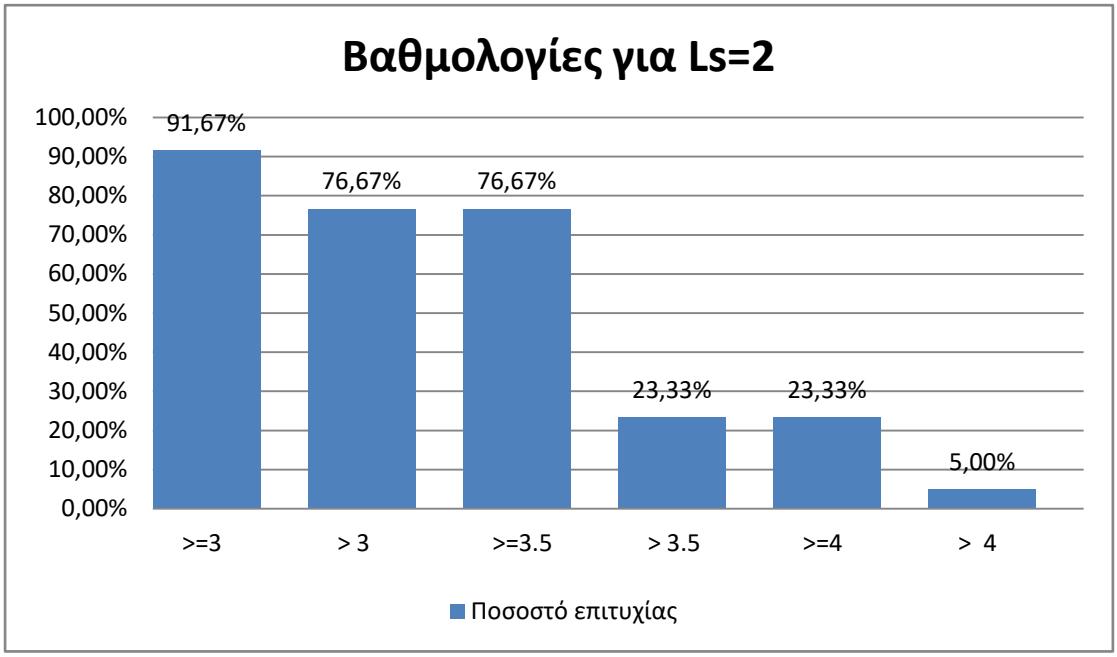


Εικόνα 7 : Αποτελέσματα solver από serendipity γραμμικού μοντέλου παλινδρόμησης για Ls=3

Όπως φαίνεται στο παραπάνω διάγραμμα το ποσοστό επιτυχίας ανέρχεται στο 97,78%, όταν προτείνονται ταινίες που έχουν πάρει βαθμολογία μεγαλύτερη ή ίση από τρια. Για ταινίες που έχουν βαθμολογηθεί παραπάνω από τρια ή με βαθμολογία μεγαλύτερη ή ίση του τρεισήμισι το ποσοστό φτάνει το 91,11%. Αντίστοιχα για ταινίες που έχουν βαθμολογηθεί παραπάνω από τρεισήμισι, παραπάνω απο τέσσερα ή και με τέσσερα το ποσοστό φτάνει το 30%. Τέλος για ταινίες με βαθμολογία πάνω απο τέσσερα το ποσοστό είναι 11,11%.

3.8.2 Αποτελέσματα solver από serendipity μοντέλου της συνάρτησης εκτίμησης

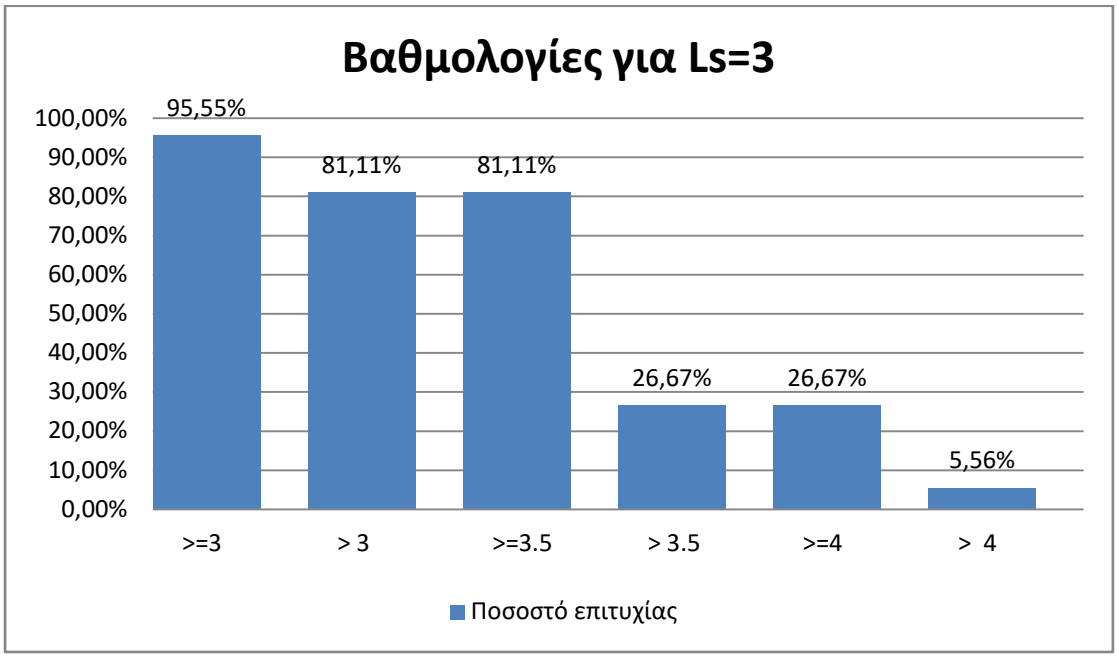
Στο παρακάτω διάγραμμα φαίνονται τα αποτελέσματα όταν το μοντέλο προτείνει δύο ταινίες σε κάθε χρήστη(Ls=2). Πιο συγκεκριμένα τα ποσοστά επιτυχούς πρόβλεψης ανάλογα με τη βαθμολογία.



Εικόνα 8 : Αποτελέσματα solver από serendipity μοντέλου της συνάρτησης εκτίμησης για Ls=2

Όπως φαίνεται στο παραπάνω διάγραμμα το ποσοστό επιτυχίας ανέρχεται στο 91,67%, όταν προτείνονται ταινίες που έχουν πάρει βαθμολογία μεγαλύτερη ή ίση από τρια. Για ταινίες που έχουν βαθμολογηθεί παραπάνω από τρια ή με βαθμολογία μεγαλύτερη ή ίση του τρεισήμισι το ποσοστό φτάνει το 76,67%. Αντίστοιχα για ταινίες που έχουν βαθμολογηθεί παραπάνω από τρεισήμισι, παραπάνω απο τέσσερα ή και με τέσσερα το ποσοστό φτάνει το 23,33%. Τέλος για ταινίες με βαθμολογία πάνω απο τέσσερα το ποσοστό είναι 5%.

Στο παρακάτω διάγραμμα φαίνονται τα αποτελέσματα όταν το μοντέλο προτείνει τρεις ταινίες σε κάθε χρήστη(Ls=3).

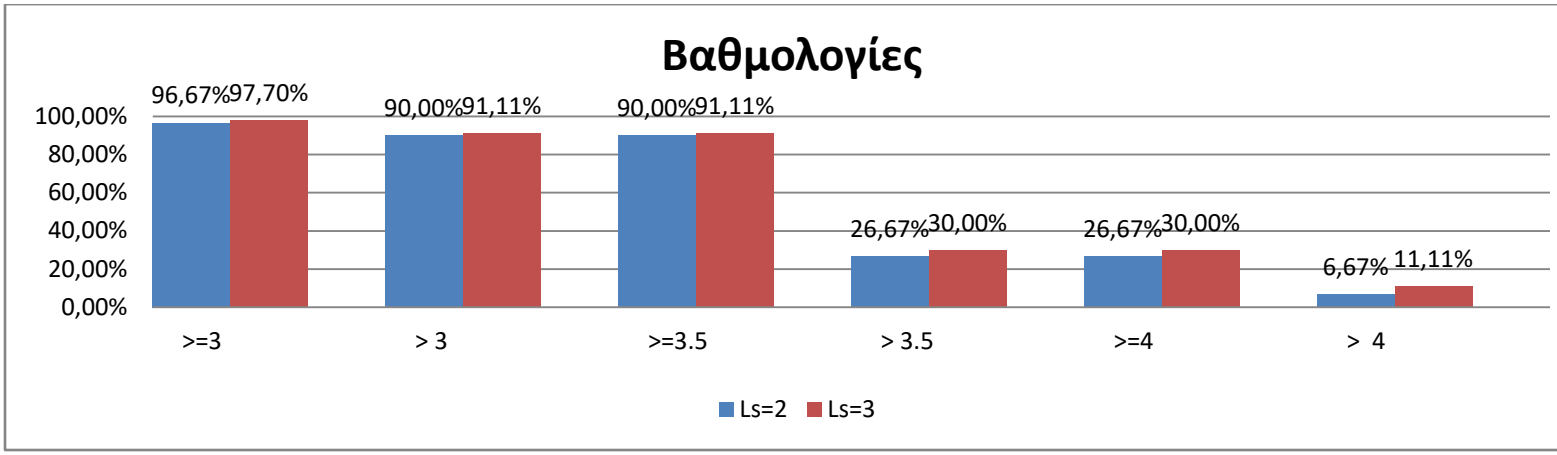


Εικόνα 9 : Αποτελέσματα solver από serendipity μοντέλου της συνάρτησης εκτίμησης για Ls=3

Όπως φαίνεται στο παραπάνω διάγραμμα το ποσοστό επιτυχίας ανέρχεται στο 95.55%, όταν προτείνονται ταινίες που έχουν πάρει βαθμολογία μεγαλύτερη ή ίση από τρια. Για ταινίες που έχουν βαθμολογηθεί παραπάνω από τρια ή με βαθμολογία μεγαλύτερη ή ίση του τρεισήμισι το ποσοστό φτάνει το 81.11%. Αντίστοιχα για ταινίες που έχουν βαθμολογηθεί παραπάνω από τρεισήμισι, παραπάνω απο τέσσερα ή και με τέσσερα το ποσοστό φτάνει το 26.67%. Τέλος για ταινίες με βαθμολογία πάνω απο τέσσερα το ποσοστό είναι 5.56%.

3.8.3 Σύγκριση αποτελεσμάτων solver από serendipity γραμμικού μοντέλου παλινδρόμησης για Ls=2 και Ls=3

Στο κεφάλαιο αυτό θα γίνει σύγκριση των αποτελεσμάτων του solver, όταν προτείνει δύο ή τρεις ταινίες(Ls=2 ή Ls=3) με βάση το serendipity που υπολογίστηκε απο το γραμμικό μοντέλο παλινδρόμησης. Στο παρακάτω διάγραμμα φαίνονται αυτά τα αποτελέσματα.

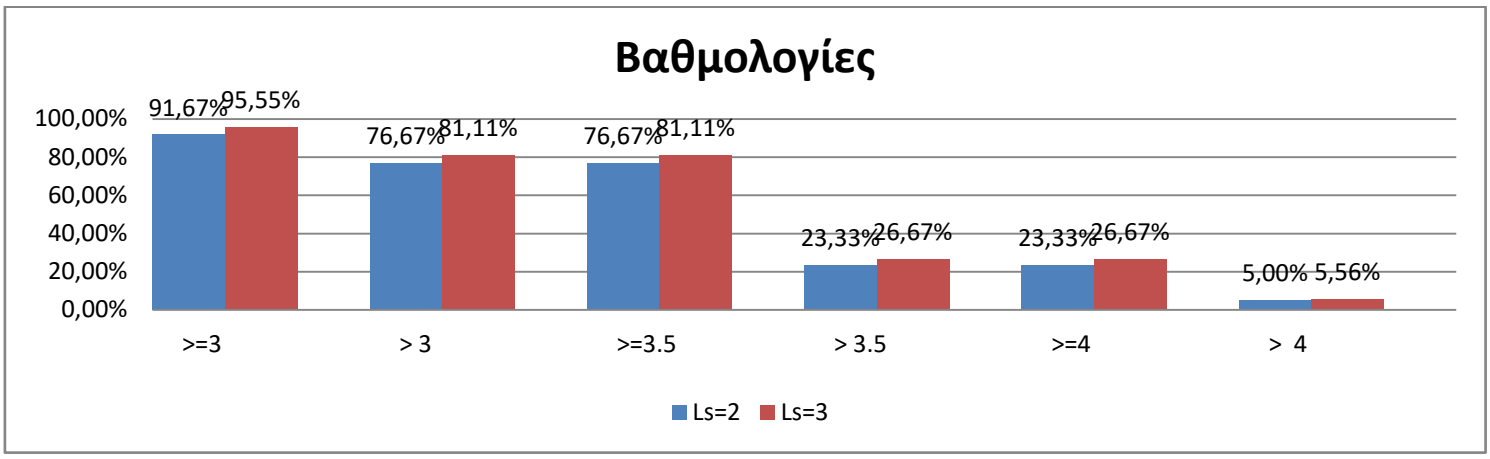


Εικόνα 10 : Σύγκριση αποτελεσμάτων solver από serendipity γραμμικού μοντέλου παλινδρόμησης για Ls=2 και Ls=3

Όπως φαίνεται στο παραπάνω διάγραμμα τα ποσοστά επιτυχίας, ανάλογα με το αν το σύστημα θα προτείνει δύο ή τρεις ταινίες στον χρήστη είναι πολύ κοντά. Στο μόνο σημείο που υπάρχει μεγάλη διαφορά είναι για βαθμολογία πάνω απο τέσσερα ,όπου το ποσοστό για Ls=3 είναι σχεδόν διπλάσιο από το αντίστοιχο για Ls=2. Επίσης παρατηρούμε ότι τα ποσοστά είναι καλύτερα όταν το σύστημα προτείνει τρεις ταινίες.

3.8.4 Σύγκριση αποτελεσμάτων solver από serendipity μοντέλου της συνάρτησης εκτίμησης για Ls=2 και Ls=3

Στο κεφάλαιο αυτό θα γίνει σύγκριση των αποτελεσμάτων του solver, όταν προτείνει δύο ή τρεις ταινίες(Ls=2 ή Ls=3) με βάση το serendipity που υπολογίστηκε απο το μη γραμμικό μοντέλο παλινδρόμησης. Στο παρακάτω διάγραμμα φαίνονται αυτά τα αποτελέσματα.



Εικόνα 11 : Σύγκριση αποτελεσμάτων solver από serendipity μοντέλου της συνάρτησης εκτίμησης για Ls=2 και Ls=3

Όπως φαίνεται στο παραπάνω διάγραμμα τα ποσοστά επιτυχίας, ανάλογα με το αν το σύστημα θα προτείνει δύο ή τρεις ταινίες στον χρήστη είναι και πάλι πολύ κοντά. Επίσης παρατηρούμε ξανά ότι τα ποσοστά είναι καλύτερα όταν το σύστημα προτείνει τρεις ταινίες.

3.8.5 Σύγκριση όλων των αποτελεσμάτων

Στο κεφάλαιο αυτό θα γίνει σύγκριση όλων των αποτελεσμάτων του solver, όταν προτείνει δύο ή τρεις ταινίες(Ls=2 ή Ls=3) με βάση το serendipity που υπολογίστηκε απο τις δύο προσεγγίσεις. Στο παρακάτω πίνακα φαίνονται αυτά τα αποτελέσματα.

Ποσοστά επιτυχίας ανα βαθμολογία	**ML**		**Συνάρτηση εκτίμησης**	
	Ls=2	**Ls=3**	**Ls=2**	**Ls=3**
>=3	**96,70%**	**97,78%**	**91,67%**	**95,55%**
> 3	**90%**	**91,11%**	**76,67%**	**81,11%**
>=3.5	**90%**	**91,11%**	**76,67%**	**81,11%**
> 3.5	**26,67%**	**30,00%**	**23,33%**	**26,67%**
>=4	**26,67%**	**30,00%**	**23,33%**	**26,67%**
> 4	**6,70%**	**11,11%**	**5,00%**	**5,56%**

Πίνακας 2 : Πίνακας συγκρίσεως όλων των αποτελεσμάτων

Στον παραπάνω πίνακα το ML υποδηλώνει το γραμμικό μοντέλο πολλαπλής παλινδρόμησης. Επίσης το Ls υποδηλώνει των αριθμό των ταινιών που προτάθηκαν από το σύστημα στον χρήστη. Τέλος η αριστερή στήλη είναι οι βαθμολογίες των χρηστών.

Αυτό που παρατηρούμε κοιτάζοντας τον πίνακα είναι ότι και τα δύο μοντέλα έχουν υψηλά ποσοστά για βαθμολογία μεγαλύτερη ή ίση με το τρια. Όσο αυξάνουν οι βαθμολογίες παρατηρούμε ότι και τα δύο μοντέλα μειώνουν το ποσοστό τους, με το μοντέλο ML να κάνει καλύτερες συστάσεις. Πιο συγκεκριμένα για βαθμολογίες μεγαλύτερες από τρια ή με βαθμολογία μεγαλύτερη ή ίση του τρεισήμισι το ML μοντέλο ξεπερνά το μοντέλο της συνάρτησης εκτίμησης κατά δέκα και δεκατρείς μονάδες για Ls=3 και Ls=2 αντίστοιχα. Για ταινίες που έχουν βαθμολογηθεί παραπάνω από τρεισήμισι, παραπάνω απο τέσσερα ή και με τέσσερα πάλι το μοντέλο ML δίνει καλύτερες συστάσεις. Τέλος για βαθμολογία μεγαλύτερη του τέσσερα και τα δύο μοντέλα έχουν πολύ χαμηλό ποσοστό, με το ML μοντέλο όμως να έχει διπλάσιο ποσοστό έναντι του το μοντέλου της συνάρτησης εκτίμησης όταν προτείνουν τρεις ταινίες.

4. Συμπεράσματα και μελλοντική εργασία

Στο κεφάλαιο αυτό θα γίνει αναφορά των συμπερασμάτων πάνω στα αποτελέσματα καθώς και ιδέες για μελλοντική εργασία.

4.1 Συμπεράσματα

Το συμπέρασμα που απορρέει από τα αποτελέσματα είναι ότι τις καλύτερες συστάσεις ταινιών στους χρήστες, τις κάνει το μοντέλο ML όταν προτείνει τρεις ταινίες. Τις χειρότερες συστάσεις τις κάνει το μοντέλο της συνάρτησης εκτίμησης, όταν προτείνει δύο ταινίες. Αυτό οφείλεται στο γεγονός ότι το serendipity που υπολογίζουμε από την συνάρτηση 2.1.1-3 είναι προσεγγιστικό, ενώ το serendipity που υπολογίζουμε από τη πολλαπλή γραμμική παλινδρόμηση(multiple linear regression) είναι πιο κοντά στην πραγματικότητα. Ο λόγος που συμβαίνει αυτό είναι γιατί το serendipity υπολογίζεται βάσει πραγματικών τιμών.

Είναι λογικό όταν το σύστημα θα προτείνει παραπάνω ταινίες να επιτυγχάνει μεγαλύτερα ποσοστά, λόγω του ότι αυξάνονται οι πιθανότητες να προτείνει κάτι που θα αρέσει στον χρήστη. Αυτός είναι και ο λόγος που και τα δύο μοντέλα έχουν μεγαλύτερα ποσοστά όταν κάνουν συστάσεις για τρεις ταινίες έναντι δύο.

Επίσης παρατηρούμε ότι και οι δύο προσεγγίσεις για πολύ μεγάλες βαθμολογίες(>4) δεν επιτυγχάνουν καλά ποσοστά. Αυτό συμβαίνει γιατί είναι πολύ δύσκολο να προτείνεις μία ταινία σε ένα χρήστη, η οποία είναι απο κατηγορία που δεν συνηθίζει να βλέπει και να τη βαθμολογήσει με πολύ μεγάλο βαθμό(4,5 ή 5). Επιπλέον αν σκεφτούμε ότι το μέσο τετραγωνικό σφάλμα εκτίμησης του serendipity από το γραμμικό μοντέλο παλινδρόμησης ήταν 0,38 , γίνεται κατανοητός ο λόγος ύπαρξης μικρών ποσοστών.

4.2 Μελλοντική εργασία

Σαν μελλοντική εργασία πάνω στη παρούσα διπλωματική θα μπορούσε να είναι η βελτίωση των συστάσεων του μοντέλου για υψηλές βαθμολογίες. Επίσης θα μπορούσε να υπολογιστεί η ομοιότητα(similarity), Sij μεταξύ μιας ταινίας i και άλλων ταινιών που ο χρήστης έχει αξιολογήσει με βάση το timestamp. Δηλαδή για μια δεδομένη ταινία i που έχει βαθμολογηθεί από ένα χρήστη, θα μπορούσε να υπολογιστεί η ομοιότητα της με τις ταινίες που έχει αξιολογήσει ο ίδιος χρήστης και έχουν την μικρότερη διαφορά timestamp μεταξύ τους. Τέλος θα είχε ενδιαφέρον το τελικό αποτέλεσμα, αν χρησιμοποιηθεί σαν τρίτο χαρακτηριστικό το predicted rating στο μοντέλο (π.χ. γραμμική παλινδρόμηση).

5. Παράρτημα αρχείων

Σε αυτό το κεφάλαιο παρουσιάζονται τα αρχεία που χρησιμοποιήθηκαν και δημιουργήθηκαν για την υλοποίηση της εργασίας.

Στην παρακάτω εικόνα φαίνεται το dataset που αναφέρθηκε στο κεφάλαιο 3.3.

	movieId	title	directedBy	starring	genres
0	117444	Song of the Sea (2014)	Tomm Moore	Brendan Gleeson,Fionnula Flanagan,David Rawle,...	Animation,Fantasy,Children
1	150548	Sherlock: The Abominable Bride (2016)	Douglas Mackinnon	Benedict Cumberbatch,Martin Freeman,Amanda Abb...	Thriller,Drama,Mystery,Action,Crime
2	136542	Mickey's Christmas Carol (1983)	Burny Mattinson	Alan Young,Wayne Allwine,Clarence Nash,Hal Smith	Children,Animation
3	2060	BASEketball (1998)	David Zucker	Trey Parker, Matt Stone, Dian Bachar, Yasmine ...	Comedy
4	135534	Krampus (2015)	Michael Dougherty	Adam Scott,Toni Collette,Allison Tolman,David ...	Comedy,Fantasy,Horror
5	128542	Wyrmwood (2015)	Kiah Roache-Turner	Jay Gallagher,Bianca Bradey,Leon Burchill,Luke...	Action,Horror,Sci-Fi
6	65514	Ip Man (2008)	Wilson Yip	Donnie Yen, Simon Yam, Siu-Wong Fan, Ka Tung L...	Action,Drama,War
7	1291	Indiana Jones and the Last Crusade (1989)	Steven Spielberg	Harrison Ford, Sean Connery, John Rhys-Davies,...	Action,Adventure
8	162606	The Accountant (2016)	Gavin O'Connor	Anna Kendrick,Ben Affleck,J.K. Simmons,Jeffrey...	Crime,Thriller,Drama
9	127096	Project Almanac (2015)	Dean Israelite	Amy Landecker,Ginny Gardner,Jonny Weston,Katie...	Sci-Fi,Thriller
10	4699	Original Sin (2001)	Michael Cristofer	Antonio Banderas, Angelina Jolie, Thomas Jane,...	Drama,Romance,Thriller
11	80831	Let Me In (2010)	Matt Reeves	Kodi Smit-McPhee, Chloë Grace Moretz, Richard ...	Drama,Horror,Mystery
12	110882	Locke (2013)	Steven Knight	Tom Hardy, Olivia Colman, Ruth Colman, Andrew ...	Drama
13	159401	Kóblic (2016)	Sebastián Borensztein	Ricardo Darín,Inma Cuesta,Oscar Martínez	Crime
14	131502	A Wolf at the Door (2013)	Fernando Coimbra	Milhem Cortaz,Fabiula Nascimento,Leandra Leal,...	Drama,Thriller

Εικόνα 12 : Μέρος του αρχείου με τις ταινίες που δεν έχουν δει οι χρήστες

Στην παρακάτω εικόνα φαίνεται το dataset που αναφέρθηκε στο κεφάλαιο 3.4. Δηλαδή το αρχείο που περιέχει το υπολογισμένο similarity και popularity για τις ταινίες που δεν έχουν δει οι χρήστες.

	userId	MovieId	similarity	popularity
0	101889	117444	0.352578	2.682145
1	101889	150548	0.351411	2.682145
2	101889	136542	0.354850	2.682145
3	101889	2060	0.353162	2.682145
4	101889	135534	0.372323	2.682145

Εικόνα 13 : Μέρος του αρχείου με similarity και popularity

Στην παρακάτω εικόνα φαίνεται το dataset που αναφέρθηκε στο κεφάλαιο 3.6.1. Δηλαδή το αρχείο που περιέχει το υπολογισμένο serendipity. Ίδια δομή με αυτό έχει και το αρχείο που αναφέρθηκε στο κεφάλαιο 3.6.2.

	userId	MovieId	similarity	popularity	serendipity
0	101889	117444	0.352578	2.682145	3.100615
1	101889	150548	0.351411	2.682145	3.100963
2	101889	136542	0.354850	2.682145	3.099939
3	101889	2060	0.353162	2.682145	3.100442
4	101889	135534	0.372323	2.682145	3.094740
5	101889	128542	0.333333	2.682145	3.106341
6	101889	65514	0.347678	2.682145	3.102073
7	101889	1291	0.365863	2.682145	3.096662
8	101889	162606	0.432936	2.682145	3.076706
9	101889	127096	0.333333	2.682145	3.106341

Εικόνα 14 : Μέρος του αρχείου με υπολογισμένο serendipity

Βιβλιογραφία

[1] Francesco Ricci, Lior Rokach, and Bracha Shapira, "Introduction to Recommender Systems Handbook," *Recommender Systems Handbook*, pp. 1-35, 2011. [Online]. https://academic.microsoft.com/paper/1486317198

[2] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro, "Content-based Recommender Systems: State of the Art and Trends," *Recommender Systems Handbook*, pp. 73-105, 2011. [Online]. https://academic.microsoft.com/paper/2116206254

[3] Upendra Shardanand and Pattie Maes, "Social information filtering: algorithms for automating “word of mouth”," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1995, pp. 210-217. [Online]. https://academic.microsoft.com/paper/2124591829

[4] B. Sheth and P. Maes, "Evolving agents for personalized information filtering," in *Proceedings of 9th IEEE Conference on Artificial Intelligence for Applications*, 1993, pp. 345-352. [Online]. https://academic.microsoft.com/paper/2137719099

[5] Daniel Billsus and Michael J. Pazzani, "User Modeling for Adaptive News Access," *User Modeling and User-adapted Interaction*, vol. 10, no. 2, pp. 147-180, 2000. [Online]. https://academic.microsoft.com/paper/1582340466

[6] Yi Zhang, Jamie Callan, and Thomas Minka, "Novelty and redundancy detection in adaptive filtering," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002, pp. 81-88. [Online]. https://academic.microsoft.com/paper/1981825277

[7] Marko Balabanović and Yoav Shoham, "Fab: content-based, collaborative recommendation," *Communications of The ACM*, vol. 40, no. 3, pp. 66-72, 1997. [Online]. https://academic.microsoft.com/paper/2043403353

[8] Xiaoyuan Su and Taghi M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, p. 4, 2009. [Online]. https://academic.microsoft.com/paper/2100235918

[9] Athanasios N. Nikolakopoulos and John D. Garofalakis, "Top-N recommendations in the presence of sparsity: An NCD-based approach," *web intelligence*, vol. 13, no. 4, pp. 247-265, 2015. [Online]. https://academic.microsoft.com/paper/796211527

[10] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan, "Collaborative Filtering Recommender Systems," *Foundations and Trends in Human-Computer Interaction*, vol. 4, no. 2, pp. 81-173, 2011. [Online]. https://academic.microsoft.com/paper/2105953200

[11] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5-53, 2004. [Online].

https://academic.microsoft.com/paper/1971040550

[12] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76-80, 2003. [Online]. https://academic.microsoft.com/paper/2159094788

[13] George Karypis, "Evaluation of Item-Based Top- N Recommendation Algorithms," in *Proceedings of the tenth international conference on Information and knowledge management*, 2001, pp. 247-254. [Online]. https://academic.microsoft.com/paper/2128629010

[14] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285-295. [Online]. https://academic.microsoft.com/paper/2042281163

[15] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *IEEE Computer*, vol. 42, no. 8, pp. 30-37, 2009. [Online]. https://academic.microsoft.com/paper/2054141820

[16] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering," *in Proceedings of the fifth international conference on computer and information technology*, vol. 1, 2002, http://glaros.dtc.umn.edu/gkhome/fetch/papers/clusterICCIT02.pdf.

[17] Koji Miyahara and Michael J. Pazzani, "Collaborative filtering with the simple Bayesian classifier," in *PRICAI'00 Proceedings of the 6th Pacific Rim international conference on Artificial intelligence*, 2000, pp. 679-689. [Online]. https://academic.microsoft.com/paper/1853953842

[18] Slobodan Vucetic and Zoran Obradovic, "Collaborative Filtering Using a Regression-Based Approach," *Knowledge and Information Systems*, vol. 7, no. 1, pp. 1-22, 2005. [Online]. https://academic.microsoft.com/paper/2038901440

[19] Daniel Lemire and Anna Maclachlan, "Slope One Predictors for Online Rating-Based Collaborative Filtering," in *SDM*, 2005, pp. 471-475. [Online]. https://academic.microsoft.com/paper/2152208379

[20] Eleni Stai, Vasileios Karyotis, and Symeon Papavassiliou, "A hyperbolic space analytics framework for big network data and their applications," *IEEE Network*, vol. 30, no. 1, pp. 11-17, 2016. [Online]. https://academic.microsoft.com/paper/2285309634

[21] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach, "Beyond accuracy: evaluating recommender systems by coverage and serendipity," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 257-260. [Online]. https://academic.microsoft.com/paper/2161676175

[22] Denis Kotkov, Joseph A. Konstan, Qian Zhao, and Jari Veijalainen, "Investigating serendipity in recommender systems based on real user feedback," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, pp. 1341-1350. [Online].

https://academic.microsoft.com/paper/2811138351

[23] Jeffrey Elkner, Allen B. Downey, and Chris Meyers, *How To Think Like A Computer Scientist: Learning With Python*., 2002. [Online]. https://academic.microsoft.com/paper/1514942850

[24] https://en.wikipedia.org/wiki/Linear_programming.

[25] I. Koutsopoulos, M. Halkidi. *"Recommender systems optimization for coverage, diversity, and serendipity". Technical report*, 2019.

[26] M. Kaminskas and D. Bridge, "*Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems*", ACM Trans. on Interactive Intelligent Systems 7(1):1-42, Dec. 2016.

[27] D. Kotkov, S. Wang, and J. Veijalainen, "*A survey of serendipity in recommender systems*", Elsevier Knowledge-Based Systems, vol. 111, 180–192, 2016.