Πανεπιστήμιο Πειραιώς

Τμήμα Ψηφιακών Συστημάτων

Π.Μ.Σ. «Πληροφοριακά Συστήματα και Υπηρεσίες»

Κατεύθυνση: «Μεγάλα Δεδομένα και Αναλυτική»

Προσέγγιση συστάσεων βασισμένη στην έκπληξη

(A recommendation approach based on serendipity)

Διπλωματική Εργασία

ΧΑΡΙΚΛΕΙΑ ΡΑΠΤΗ

Επιβλέπουσα:

ΜΑΡΙΑ ΧΑΛΚΙΔΗ

Αναπληρώτρια Καθηγήτρια

Αθήνα, Μάρτιος 2022

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτρια της διπλωματικής μου Καθηγήτρια κ. Μαρία Χαλκίδη για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα, καθώς και για την καθοδήγηση και την βοήθεια που μου παρείχε καθ' όλη τη διάρκεια της εκπόνησης αυτής της διπλωματικής εργασίας.

Τέλος, είμαι ευγνώμων στην οικογένειά μου και τους φίλους μου για την συμπαράσταση τους και την στήριξη που μου παρείχαν καθ' όλη τη διάρκεια των σπουδών μου.

Περίληψη

Τα συστήματα συστάσεων χρησιμοποιούν προηγούμενες συμπεριφορές χρηστών για να προτείνουν αντικείμενα (π.χ ταινίες). Οι περισσότεροι τείνουν να προσφέρουν αντικείμενα παρόμοια με τα στοιχεία που ένας χρήστης-στόχος έχει υποδείξει ως ενδιαφέροντα. Ως αποτέλεσμα, οι χρήστες βαριούνται τις προφανείς προτάσεις που μπορεί να έχουν ήδη ανακαλύψει.

Για τη βελτίωση της ικανοποίησης των χρηστών, τα συστήματα συστάσεων θα πρέπει να προσφέρουν προτάσεις που θα προκαλέσουν έκπληξη: αντικείμενα όχι μόνο σχετικά και πρωτότυπα για τον χρήστη-στόχο, αλλά και σημαντικά διαφορετικά από τα αντικείμενα που έχει βαθμολογήσει ο χρήστης.

Ωστόσο, η έννοια του serendipity (έκπληξη) είναι πολύ υποκειμενική, και οι αιφνιδιαστικές συναντήσεις χρήστη-αντικειμένου είναι πολύ σπάνιες σε σενάρια του πραγματικού κόσμου, γεγονός που καθιστά τις συστάσεις που προκαλούν έκπληξη στον χρήστη, εξαιρετικά δύσκολο να μελετηθούν.

Μέχρι σήμερα, έχουν προταθεί διάφοροι ορισμοί και μετρήσεις αξιολόγησης για τη μέτρηση της έκπληξης και δυστυχώς δεν υπάρχει ευρεία συναίνεση σχετικά με τον ορισμό ή την αξιολόγηση της μετρικής που μπορεί να χρησιμοποιηθεί.

Στόχος της παρούσας εργασίας είναι η σχεδίαση και υλοποίηση προσέγγισης που θα λαμβάνει υπόψη το serendipity στη διαδικασία συστάσεων. Πιο συγκεκριμένα το σύστημα θα υπολογίζει το serendipity για ταινίες που έχει ήδη δει και βαθμολογήσει ο χρήστης αλλά και για νέες ταινίες που δεν έχει δει και θα κάνει τις ανάλογες προτάσεις. Τέλος, θα γίνεται αξιολόγηση των αλγορίθμων που χρησιμοποιήθηκαν και σύγκριση των εκάστοτε αποτελεσμάτων.

Abstract

Recommender systems use past behaviors of users to suggest items. Most tend to offer items similar to the items that a target user has indicated as interesting. As a result, users become bored with obvious suggestions that they might have already discovered. To improve user satisfaction, recommender systems should offer serendipitous suggestions: items not only relevant and novel to the target user, but also significantly different from the items that the user has rated. However, the concept of serendipity is very subjective and serendipitous encounters are very rare in real-world scenarios, which makes serendipitous recommendations extremely difficult to study. To date, various definitions and evaluation metrics to measure serendipity have been proposed, and there is no wide consensus on which definition and evaluation metric to use. Aim of this paper is to design and implement an approach that takes serendipity into account in the referral process. More specifically, the system that will evaluate the serendipity for movies that the user has already seen and rated, but also for new movies that he has not seen and will make the appropriate suggestions. Finally, the algorithms used will be evaluated and the results will be compared.

Περιεχόμενα

Κατάλογος Σχημάτων

Κεφάλαιο 1: Εισαγωγή

1.1 Αντικείμενο Διπλωματικής

Με την αύξηση της πληροφορίας στο Διαδίκτυο καθίσταται πλέον δύσκολο να βρεθεί κάποιο περιεχόμενο ενδιαφέρον για έναν χρήστη. Τα συστήματα συστάσεων έχουν σχεδιαστεί για να λύσουν αυτό το πρόβλημα. Ο όρος σύστημα συστάσεων αναφέρεται σε ένα εργαλείο λογισμικού που προτείνει αντικείμενα στους χρήστες. Για παράδειγμα, ένα αντικείμενο μπορεί να είναι μια ταινία, ένα τραγούδι ή ακόμα και έναν φίλο σε ένα διαδικτυακό κοινωνικό δίκτυο.

Τα συστήματα συστάσεων και οι μηχανές αναζήτησης είναι διαφορετικά είδη συστημάτων που στοχεύουν στην ικανοποίηση των αναγκών πληροφοριών των χρηστών. Παραδοσιακά, μια μηχανή αναζήτησης λαμβάνει ένα ερώτημα και, σε ορισμένες περιπτώσεις, ένα προφίλ χρήστη ως είσοδο και παρέχει ένα σύνολο από τα πιο κατάλληλα στοιχεία ως απάντηση. Αντίθετα, ένα σύστημα συστάσεων δεν λαμβάνει κανένα ερώτημα, αλλά το προφίλ ενός χρήστη και επιστρέφει ένα σύνολο αντικειμένων που θα απολάμβαναν οι χρήστες. Ο όρος προφίλ χρήστη αναφέρεται σε ενέργειες που έκανε ένας χρήστης με αντικείμενα στο παρελθόν.Το προφίλ ενός χρήστη συχνά αντιπροσωπεύεται από αξιολογήσεις που έκανε ένας χρήστης σε αντικείμενα.

Οι περισσότεροι αλγόριθμοι συστάσεων αξιολογούνται με βάση την ακρίβεια που υποδεικνύει πόσο καλός είναι ένας αλγόριθμος στο να προσφέρει ενδιαφέροντα στοιχεία ανεξάρτητα από το πόσο προφανείς και γνωστές σε έναν χρήστη είναι οι προτάσεις. Για να επιτευχθεί υψηλή ακρίβεια, τα συστήματα συστάσεων τείνουν να προτείνουν παρόμοια αντικείμενα σε ένα προφίλ χρήστη. Ως αποτέλεσμα, ο χρήστης λαμβάνει συστάσεις μόνο για παρόμοια αντικείμενα που ο χρήστης βαθμολόγησε αρχικά. Οι αλγόριθμοι που βασίζονται στην ακρίβεια περιορίζουν τον αριθμό των στοιχείων που μπορούν να προταθούν στον χρήστη (το λεγόμενο πρόβλημα της υπερεξειδίκευσης), το οποίο μειώνει την ικανοποίηση των χρηστών. Για να ξεπεραστεί το πρόβλημα της υπερεξειδίκευσης και να διευρυνθούν οι προτιμήσεις των χρηστών, ένα σύστημα συστάσεων θα πρέπει να προτείνει αντικείμενα που θα προκαλέσουν έκπληξη.

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί η εξέταση του ορισμού του serendipity. Επίσης εξετάζουμε και ταξινομούμε τις μετρικές αξιολόγησης για τη

μέτρηση της έκπληξης και αναφέρουμε τα πλεονεκτήματα και τα μειονεκτήματα τους.

1.2 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφονται τα συστήματα συστάσεων, τα βασικά κίνητρα για τη δημιουργία τους και διάφοροι τύποι συστημάτων συστάσεων. Στη συνέχεια παρουσιάζεται

1.3 Εισαγωγή στα συστήματα συστάσεων

1.3.1 Συστήματα συστάσεων

Υπάρχει μια εκτεταμένη κατηγορία εφαρμογών Ιστού που περιλαμβάνουν πρόβλεψη απαντήσεων του χρήστη σε επιλογές απαντήσεων. Μια τέτοια εφαρμογή ονομάζεται σύστημα συστάσεων. Ωστόσο, για να εστιάσουμε στο πρόβλημα, δύο καλά παραδείγματα συστήματα συστάσεων είναι:

1. Προσφορά ειδησεογραφικών άρθρων σε διαδικτυακούς αναγνώστες εφημερίδων, βάσει πρόβλεψης των ενδιαφερόντων των αναγνωστών.

2. Προσφορά στους πελάτες ενός on-line λιανοπωλητή, προτάσεις σχετικά με το τι μπορεί να θέλουν να αγοράσουν, με βάση το προηγούμενο ιστορικό αγορών ή/και αναζητήσεων προϊόντων.

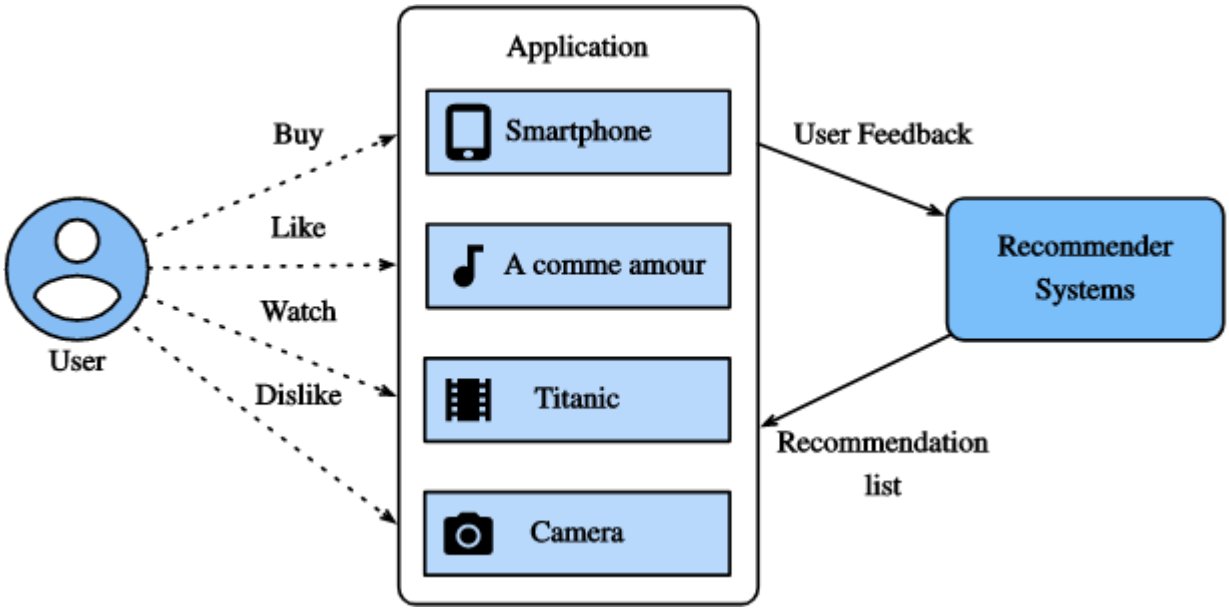
Τα συστήματα συστάσεων χρησιμοποιούν πολλές διαφορετικές τεχνολογίες. Μπορούμε να ταξινομήσουμε αυτά τα συστήματα σε δύο μεγάλες ομάδες.

• Τα συστήματα που βασίζονται σε περιεχόμενο και εξετάζουν τις ιδιότητες των συνιστώμενων στοιχείων.

Για παράδειγμα, εάν ένας χρήστης του Netflix έχει παρακολουθήσει πολλές ταινίες καουμπόι, τότε προτείνετε μια ταινία που έχει ταξινομηθεί στη βάση δεδομένων ως το είδος «καουμπόι».

• Τα συνεργατικά συστήματα φιλτραρίσματος που προτείνουν στοιχεία που βασίζονται σε μέτρα ομοιότητας μεταξύ χρηστών και/ή στοιχείων. Τα στοιχεία που προτείνονται σε έναν χρήστη είναι αυτά που προτιμούν παρόμοιοι χρήστες.

Ωστόσο, αυτές οι τεχνολογίες από μόνες τους δεν επαρκούν και υπάρχουν ορισμένοι νέοι αλγόριθμοι που έχουν αποδειχθεί πιο αποτελεσματικοί για τα συστήματα συστάσεων.



Εικόνα 1 : Παράδειγμα ενός Συστήματος Συστάσεων

1.3.2 Τρόπος λειτουργίας

Ένα σύστημα συστάσεων λειτουργεί με δύο διαφορετικούς τύπους τρόπων.

Πρώτον, θα δει τι θα αρέσει στους ανθρώπους που έχουν παρόμοια γούστα με εσάς. Αυτό γίνεται με τη συλλογή μεταδεδομένων, όπως αν προτείνουν μια συγκεκριμένη εκπομπή στο Netflix και βλέποντας τι τείνουν να παρακολουθούν οι άνθρωποι μετά την ολοκλήρωση αυτής της εκπομπής.

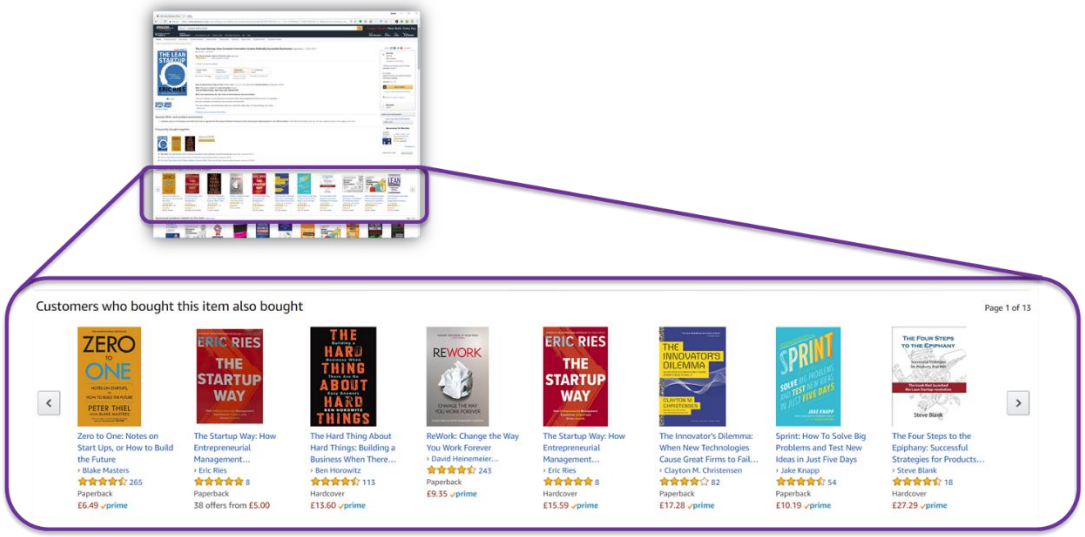
Ο άλλος τρόπος με τον οποίο θα λειτουργήσει ένα σύστημα προτάσεων είναι ότι θα εκχωρήσει μια συγκεκριμένη βαθμολογία «πιθανότητας» σε χρήστες και στοιχεία που μπορεί να τους αρέσουν. Στη συνέχεια, θα χρησιμοποιήσει το φιλτράρισμα για να αφαιρέσει τυχόν απίθανα στοιχεία ή ανωμαλίες και θα σας εμφανίσει μόνο στοιχεία που συνδέονται με αυτό για το οποίο έχετε ήδη δείξει ενδιαφέρον.

Με πιο τεχνικούς όρους, οι αγορές ή τα ενδιαφέροντα των χρηστών θα αποθηκευτούν μέσα σε έναν πίνακα χρήστη και στοιχείου. Οι περισσότερες βαθμολογίες είτε μετρούνται χρησιμοποιώντας ένα αριθμητικό σύστημα όπως '1 στα 10', αλλά ορισμένες μπορεί να χρησιμοποιούν δυαδική βαθμολογία. Αυτό συνήθως μετριέται με έναν απλούστερο τρόπο, όπως «Κλικ» ή «Το έχω δει» ('Clicked' or 'Watched').

Τα περισσότερα στοιχεία σε μια μήτρα χρήστη/στοιχείου θα μείνουν κενά, καθώς αυτό έχει σχεδιαστεί για να συμπληρώσει το μηχάνημα. Αυτά τα κενά θα συμπληρωθούν τελικά από προτεινόμενα στοιχεία ή εκπομπές.

Αυτός ο πίνακας μπορεί στη συνέχεια να συγκριθεί με άλλους χρήστες για να βρει τον «πλησιέστερο γείτονα» ή τον πιο παρόμοιο χρήστη, και μέσω της βαθιάς μάθησης, ένα μηχάνημα μπορεί να μάθει περισσότερα για το τι μπορεί να αρέσει περισσότερο στους χρήστες.

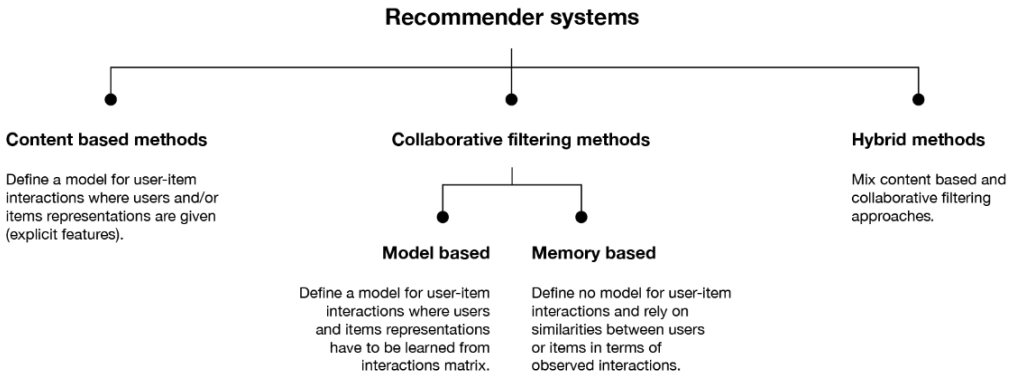
Έτσι, την επόμενη φορά που θα δείτε μια σύσταση βάσει περιεχομένου ή προτάσεις προϊόντων, απλώς γνωρίζετε ένα σύνθετο μοντέλο μηχανικής εκμάθησης (συνήθως χρησιμοποιώντας τη **μέθοδο αποσύνθεσης της μοναδικής αξίας** ή την **ομαδοποίηση**).



Εικόνα 2 : Παράδειγμα ενός Συστήματος Συστάσεων Βιβλίων

1.3.3 Τύποι συστημάτων συστάσεων

Υπάρχουν πολλοί διαφορετικοί τύποι τεχνικών και εφαρμογών εκεί έξω. Οι δύο κύριες κατηγορίες βασίζονται στη μνήμη και βασίζονται σε μοντέλα:



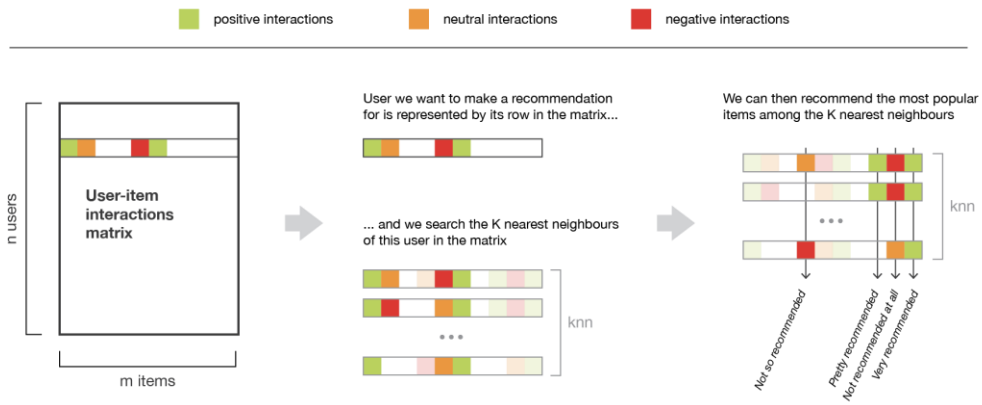
Εικόνα 3 : Τύποι συστημάτων συστάσεων

Προσεγγίσεις βασισμένες στη μνήμη

Το κύριο χαρακτηριστικό που ξεχωρίζει αυτές τις λύσεις είναι ότι υποθέτουν ότι δεν έχετε μοντέλο για να κάνετε προβλέψεις και απλώς κάνετε επιλογές με βάση τις πληροφορίες από τη μήτρα αλληλεπίδρασης χρήστη-στοιχείου

User-user

Αυτή η μέθοδος προσπαθεί να αντιστοιχίσει αυτό το άτομο με τα υπάρχοντα "προφίλ" που αντιστοιχούν στις φυσιολογικές αλληλεπιδράσεις και των δύο για να βρουν ομοιότητες. Ο στόχος είναι να προβλέψουμε το καλύτερο στοιχείο για αυτό το νέο άτομο που είναι δημοφιλές στους παρόμοιους χρήστες, με βάση τις αλληλεπιδράσεις με το στοιχείο.



Εικόνα 4 : User-user προσέγγιση

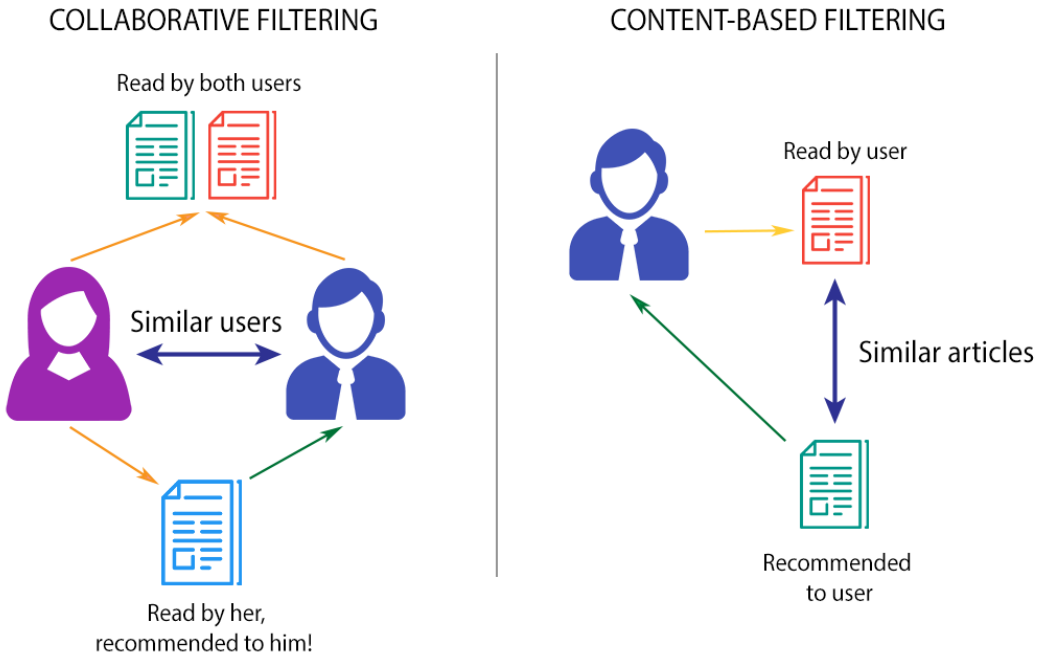
Item-item

Η ιδέα του στοιχείου στοιχείου είναι να επικεντρωθεί σε στοιχεία που μπορεί να αρέσουν σε έναν υποψήφιο με βάση άλλα στοιχεία με τα οποία αλληλεπίδρασε θετικά. Δύο διαφορετικά στοιχεία (προϊόν/σελίδα/email) θεωρούνται παρόμοια εάν η πλειοψηφία αλληλεπιδρούσε και με τα δύο με παρόμοιο τρόπο. Η κύρια διαφορά από τον χρήστη χρήστη είναι ότι τώρα επικεντρωνόμαστε σε ομοιότητες αλληλεπίδρασης μέσα σε ένα πίνακα στοιχείων, όχι σε διαφορετικούς χρήστες. Ένα από τα κύρια πλεονεκτήματα αυτής της τεχνικής είναι ότι συνήθως καταλήγουμε σε ένα διάνυσμα αντικειμένων, αναφέροντας τα πιο πιθανά αντικείμενα με τη σειρά και μας επιτρέπει να επικεντρωθούμε σε διαφορετικά με βάση την απόδοση επένδυσης ή την τιμή.

Content-Based Filtering Systems

Λύσεις που βασίζονται σε περιεχόμενο χρησιμοποιούνται από μια σειρά διαφορετικών επιχειρήσεων. Είναι ίσως η πιο συχνά χρησιμοποιούμενη μέθοδος και μία που όλοι έχουμε συναντήσει κάποια στιγμή. Ιστότοποι όπως το Amazon και το Google Play Store είναι μόνο μερικά από τα πολλά παραδείγματα που υπάρχουν.

Ένα σύστημα προτάσεων περιεχομένου θα βλέπει αυτό που ήδη ενδιαφέρει τον χρήστη και θα προτείνει παρόμοια προϊόντα ή στοιχεία.



Εικόνα 5 : Αναπαράσταση ενός συστήματος συνεργατικού φιλτραρίσματος βασιζόμενο στο χρήστη

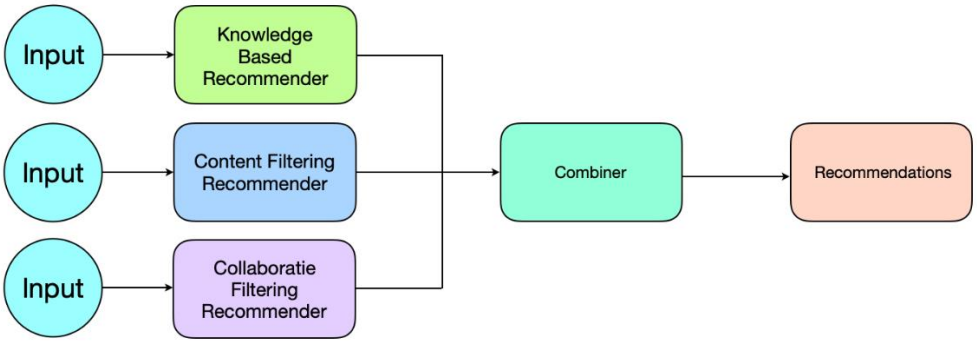
Collaborative Filtering Systems

Σε αντίθεση με ένα σύστημα προσανατολισμένο στο περιεχόμενο, τα συνεργατικά συστήματα φιλτραρίσματος θα χρησιμοποιούν αυτό που ενδιαφέρουν άλλους χρήστες παρόμοιους με εσάς. Συνήθως, μέθοδοι περιεχομένου και συνεργατικά συστήματα φιλτραρίσματος χρησιμοποιούνται σε συνδυασμό μεταξύ τους για να διασφαλιστεί ότι οι χρήστες βρίσκουν αυτό που ψάχνουν.

Μερικά από τα πιο δημοφιλή παραδείγματα είναι τα συστήματα του Netflix "What to Watch Next" και τα συστήματα "More Like" του Spotify.

Χρησιμοποιώντας αυτό που οι χρήστες με παρόμοια γούστα με εσάς απολάμβαναν στο παρελθόν, είναι περισσότερο από δυνατό να προσφέρετε ακριβείς συστάσεις. Ένα συνεργατικό σύστημα φιλτραρίσματος είναι ιδιαίτερα ακριβές καθώς λαμβάνει δεδομένα από πολλούς χρήστες, όχι μόνο από έναν. Στον κόσμο της μηχανικής μάθησης, όσο περισσότερα δεδομένα τόσο το καλύτερο.

Υβριδικό Σύστημα



Εικόνα 6 : Αναπαράσταση ενός υβριδικού συστήματος

Μία από τις πιο αποτελεσματικές μεθόδους είναι ο συνδυασμός περισσότερων από μία μεθόδων. Ισως το πιο συνηθισμένο μείγμα είναι ένα προτεινόμενο φιλτράρισμα βάσει περιεχομένου και συνεργασίας.

Καθώς αυτό θα συγκρίνει τα γούστα σας με άτομα που έχουν παρόμοιες προτιμήσεις με εσάς, το σύστημα μηχανικής μάθησης θα είναι σε θέση να προσφέρει πιο ακριβείς συστάσεις.

Το κύριο μειονέκτημα μιας υβριδικής τεχνικής είναι ότι συχνά θα είναι πιο ακριβό από την απλή εξειδίκευση σε μία. Ωστόσο, το κόστος είναι κάτι παραπάνω από αξιόλογο καθώς οι υβριδικές μέθοδοι έχουν αποτελέσματα.

Κεφάλαιο 2: Συστήματα συστάσεων βασισμένα στην έκπληξη (serendipity)

2.1 Ορισμός

Ο Merriam-Webster ορίζει τον όρο «serendipity» ως:

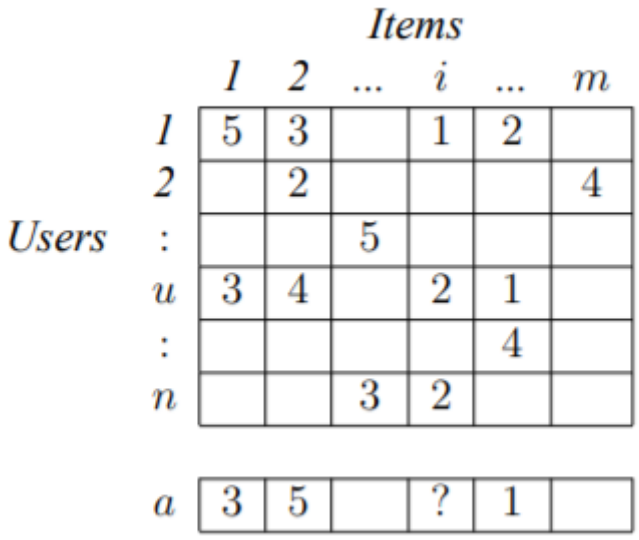
«Η ικανότητα ή το φαινόμενο της εύρεσης πολύτιμων ή ευχάριστων πραγμάτων που δεν αναζητούνται».

2.2 Serendipity in a Context

Τα περισσότερα συστήματα συστάσεων δεν λαμβάνουν υπόψη πληροφορίες σχετικά με τα συμφραζόμενα, όπως ο χρόνος, η τοποθεσία ή η διάθεση ενός χρήστη (Adomavicius and Tuzhilin, 2011). Εν τω μεταξύ, το πλαίσιο μπορεί να επηρεάσει σημαντικά τη συνάφεια των στοιχείων για έναν χρήστη (Adomavicius και Tuzhilin, 2011). Ένα στοιχείο που ήταν σχετικό για έναν χρήστη χθες ενδέχεται να μην είναι σχετικό αύριο. Ένα πλαίσιο μπορεί περιλαμβάνει οποιαδήποτε πληροφορία σχετική με συστάσεις. Για παράδειγμα, ένα σύστημα προτάσεων μπορεί να εξετάσει τον καιρό που προτείνει για ένα μέρος . Τα συστήματα προτάσεων Contextaware χρησιμοποιούν πληροφορίες σχετικά με τα συμφραζόμενα για να προτείνουν στοιχεία ενδιαφέροντα σε έναν χρήστη.

2.3 Δομή των Συστημάτων Συστάσεων

Το γενικότερο πλαίσιο μελέτης των συστημάτων συστάσεων που μελετάται φαίνεται στην εικόνα 7.



Εικόνα 7. Πίνακας αξιολογήσεων χρηστών όπου κάθε κελί $r_{u,i}$ αντιστοιχεί στην αξιολόγηση του αντικειμένου i από τον χρήστη u. Ο στόχος είναι η πρόβλεψη της αξιολόγησης $r_{a,i}$ για τον ενεργό χρήστη a.

Οι προτιμήσεις των χρηστών που είναι γνωστές, αναπαριστώνται ως ένας πίνακας n χρηστών και m αντικειμένων, όπου κάθε κελί $r_{u,i}$ αντιστοιχεί στην αξιολόγηση του αντικειμένου m από τον χρήστη n. Αυτός ο πίνακας είναι συνήθως αραιός, καθώς οι περισσότεροι χρήστες δεν αξιολογούν τα περισσότερα αντικείμενα. Ο στόχος είναι να γίνει η πρόβλεψη για το ποια αξιολόγηση θα έδινε ένας χρήστης σε ένα αντικείμενο που προηγουμένως δεν είχε αξιολογήσει. Συνήθως, γίνονται οι προβλέψεις των αξιολογήσεων για όλα τα αντικείμενα που δεν έχουν παρατηρηθεί από έναν χρήστη και τα αντικείμενα με την υψηλότερη αξιολόγηση παρουσιάζονται ως προτάσεις. Ο χρήστης για τον οποίο υπολογίζονται οι συστάσεις αναφέρεται ως ενεργός χρήστης.

Οι προσεγγίσεις στα συστήματα συστάσεων μπορούν να κατηγοριοποιηθούν ως εξής:

- **Συνεργατικό Φιλτράρισμα.** Στο συνεργατικό φιλτράρισμα προτείνονται σε έναν χρήστη αντικείμενα με βάση τις προηγούμενες αξιολογήσεις όλων των χρηστών συλλογικά.

- **Σύσταση Βάσει Περιεχομένου.** Σε αυτές τις προσεγγίσεις προτείνονται σε ένα χρήστη αντικείμενα που είναι παρόμοια ως προς το περιεχόμενο με αντικείμενα που άρεσαν στον συγκεκριμένο χρήστη στο παρελθόν ή ταιριάζουν με τα χαρακτηριστικά του χρήστη.
- **Υβριδικές προσεγγίσεις.** Αυτές οι προσεγγίσεις συνδυάζουν το συνεργατικό φιλτράρισμα με τις συστάσεις βάσει περιεχομένου.

2.4 Συνεργατικό Φιλτράρισμα

Στα συστήματα συνεργατικού φιλτραρίσματος τα δεδομένα είναι αξιολογήσεις αντικειμένων από χρήστες και η επεξεργασία τους αφορά την ανάλυση της ομοιότητας που παρουσιάζουν οι χρήστες στη συμπεριφορά τους καθώς αξιολογούν τα αντικείμενα. Ο τελικός στόχος είναι η εύρεση ενός αποτελεσματικού τρόπου σύστασης των αντικειμένων. Υπάρχουν δύο προσεγγίσεις συνεργατικού φιλτραρίσματος, μία που βασίζεται στη μνήμη και μία που βασίζεται στο μοντέλο.

Συνεργατικό Φιλτράρισμα με βάση τη μνήμη

Στη μέθοδο συνεργατικού φιλτραρίσματος με βάση τη μνήμη προκειμένου να γίνουν προβλέψεις για έναν συγκεκριμένο χρήστη επιλέγεται ένα σύνολο χρηστών που παρουσιάζουν αρκετή ομοιότητα με αυτόν και μέσω ενός συνδυασμού των αξιολογήσεών τους και με κάποια βάρη υπολογίζεται το ζητούμενο αποτέλεσμα. Παρακάτω παρουσιάζονται συνοπτικά τα βήματα αυτής της μεθόδου:

1. Ορίζεται ένα βάρος για όλους τους χρήστες, το οποίο εκφράζει την ομοιότητα με τον αρχικό χρήστη για τον οποίο θα γίνουν οι προβλέψεις.
2. Επιλέγεται ένα σύνολο k χρηστών που παρουσιάζουν την μεγαλύτερη ομοιότητα με τον αρχικό χρήστη. Το σύνολο αυτό μπορεί να ονομαστεί και ως «γειτονιά».

Γίνεται η πρόβλεψη μέσω του συνδυασμού των αξιολογήσεων και των βαρών των k χρηστών που επιλέχθηκαν.

Το βάρος $w_{a,u}$ είναι το μέτρο ομοιότητας μεταξύ του χρήστη a (ενεργός χρήστης) και του χρήστη u (χρήστης από το σύνολο που επιλέχθηκε). Ο συντελεστής συσχέτισης Pearson αποτελεί ένα μέτρο ομοιότητα μεταξύ δύο χρηστών και ορίζεται παρακάτω:

$$w_{a,u} = \frac{\Sigma_{i \in I}(r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\Sigma_{i \in I}(r_{a,i} - \bar{r}_a)^2 \, \Sigma_{i \in I}(r_{u,i} - \bar{r}_u)^2}} \quad (1)$$

όπου I είναι το σύνολο των αντικειμένων που έχουν βαθμολογηθεί και από τους δύο χρήστες, $r_{u,i}$ είναι η αξιολόγηση του αντικειμένου i από τον χρήστη u και $\bar{r}_u$ είναι η μέση αξιολόγηση που έχει γίνει από τον χρήστη u.

Η πρόβλεψη της αξιολόγησης του αντικειμένου i από έναν χρήστη a υπολογίζεται από την παρακάτω εξίσωση:

$$p_{a,i} = \bar{r}_a + \frac{\Sigma_{u \in K} (r_{u,i} - \bar{r}_u)\ x\ w_{a,u}}{\Sigma_{u \in K}\ w_{a,u}} \quad (2)$$

όπου $p_{a,i}$ είναι η πρόβλεψη για την αξιολόγηση που θα δώσει ο χρήστης a στο αντικείμενο i, $w_{a,u}$ είναι η ομοιότητα μεταξύ του χρήστη a και u και K είναι το σύνολο των χρηστών που παρουσιάζουν την μεγαλύτερη ομοιότητα με τον χρήστη a.

Το μέτρο ομοιότητας που παρουσιάστηκε παραπάνω εξετάζει τη γραμμική εξάρτηση μεταξύ δύο μεταβλητών. Ένα διαφορετικό μέτρο ομοιότητας είναι η ομοιότητα συνημίτονου. Σε αυτή την περίπτωση οι αξιολογήσεις των δύο χρηστών αντιμετωπίζονται ως διανύσματα με m διαστάσεις και υπολογίζεται το συνημίτονο της μεταξύ τους γωνίας. Στην παρακάτω εξίσωση ορίζεται αυτή η ομοιότητα:

$$w_{a,u} = cos(\vec{r}_a, \vec{r}_u) = \frac{\vec{r}_a \cdot \vec{r}_u}{\|\vec{r}_a\|_2\ x\ \|\vec{r}_u\|_2} = \frac{\Sigma_{i=1}^{m} r_{a,i} r_{u,i}}{\sqrt{\Sigma_{i=1}^{m} r_{a,i}^2} \sqrt{\Sigma_{i=1}^{m} r_{u,i}^2}} \quad (3)$$

Στον υπολογισμό της ομοιότητας συνημίτονων οι αξιολογήσεις πρέπει να έχουν μη αρνητικές τιμές, ενώ στις περιπτώσεις που δεν υπάρχει αξιολόγηση, τότε παίρνει την τιμή μηδέν.

Συνεργατικό φιλτράρισμα με βάση το αντικείμενο

Η μέθοδος συνεργατικού φιλτραρίσματος με βάση τη μνήμη δεν είναι αποτελεσματική όταν εφαρμόζεται σε εκατομμύρια χρήστες και αντικείμενα. Αυτό συμβαίνει λόγω της πολυπλοκότητας που έχει η εύρεση παρόμοιων χρηστών. Μία λύση σε αυτό το πρόβλημα αποτελεί η μέθοδος συνεργατικού φιλτραρίσματος με βάση το αντικείμενο, όπου αντί να υπολογίζονται παρόμοιοι χρήστες, υπολογίζονται παρόμοια αντικείμενα με αυτά που έχει αλληλοεπιδράσει θετικά ο χρήστης. Οι υπολογισμοί αυτής της μεθόδου είναι σημαντικά ταχύτεροι από ότι στο συνεργατικό φιλτράρισμα με βάση τη μνήμη.

Σε αυτή τη μέθοδο υπολογίζεται ο συντελεστής συσχέτισης Pearson ως μέτρο ομοιότητας μεταξύ δύο αντικειμένων i και j, όπως φαίνεται στην παρακάτω σχέση:

$$w_{i,j} = \frac{\Sigma_{u \in U} \left(r_{u,i} - \bar{r}_i\right)\left(r_{u,j} - \bar{r}_j\right)}{\sqrt{\Sigma_{u \in U}\left(r_{u,i} - \bar{r}_i\right)^2}\sqrt{\Sigma_{u \in U}\left(r_{u,j} - \bar{r}_j\right)^2}} \quad (4)$$

όπου U είναι το σύνολο όλων των χρηστών που έχουν αξιολογήσει και τα δύο αντικείμενα i και j, $r_{u,i}$ είναι η αξιολόγηση του αντικειμένου i από τον χρήστη u και $\bar{r}_i$ είναι η μέση αξιολόγηση του αντικειμένου i από όλους τους χρήστες του συνόλου U.

Η πρόβλεψη της αξιολόγησης του αντικειμένου i από έναν χρήστη a υπολογίζεται από την παρακάτω εξίσωση:

$$p_{a,i} = \frac{\Sigma_{j \in K} r_{a,j} w_{i,j}}{\Sigma_{j \in K} |w_{i,j}|} \quad (5)$$

όπου K είναι το σύνολο των k αντικειμένων που έχουν αξιολογηθεί από τον χρήστη a και παρουσιάζουν μεγαλύτερη ομοιότητα με το αντικείμενο i.

Παρακάτω παρουσιάζονται ορισμένες τεχνικές για την αντιμετώπιση πιθανών προκλήσεων κατά την διαδικασία του συνεργατικού φιλτραρίσματος με βάση τη μνήμη.

Συντελεστής Σημασίας Βάρους. Είναι πιθανό για έναν ενεργό χρήστη να υπολογιστούν ως αρκετά κοντινοί ‘γείτονές’ του, χρήστες με τους οποίους έχει πολύ λίγα κοινά αντικείμενα που έχουν αξιολογηθεί. Το γεγονός αυτό μπορεί να οδηγήσει σε λανθασμένες προβλέψεις. Ο συντελεστής σημασίας βάρους μπορεί να επιλύσει αυτό το πρόβλημα όταν πολλαπλασιαστεί με το βάρος ομοιότητας, καθώς δίνει αρκετά μικρή τιμή όταν τα κοινά αντικείμενα μεταξύ των δύο χρηστών είναι πολύ λίγα.

Προεπιλεγμένη Αξιολόγηση. Ένας διαφορετικός τρόπος επίλυσης του προβλήματος όπου τα κοινά αντικείμενα που έχουν αξιολογηθεί είναι πολύ λίγα, είναι να τεθεί μία προεπιλεγμένη τιμή στα αντικείμενα που δεν έχουν αξιολογηθεί. Έτσι, μπορεί να υπολογιστεί το μέτρο ομοιότητας $w_{a,u}$ μεταξύ δύο χρηστών χρησιμοποιώντας την ένωση των αντικειμένων που έχουν αξιολογηθεί από τον κάθε χρήστη αντί για την τομή τους.

Αντίστροφη Συχνότητα Χρήστη. Όταν υπολογίζεται η ομοιότητα μεταξύ χρηστών κάποια αντικείμενα που πιθανώς να έχουν αξιολογηθεί από όλους τους χρήστες

(ομόφωνα είτε θετικά ή αρνητικά) δεν είναι τόσο σημαντικά όσο αντικείμενα που έχουν αξιολογηθεί από λιγότερους χρήστες. Για την επίλυση αυτού του θέματος στο [9] παρουσιάστηκε η αντίστροφη συχνότητα χρήστη που ορίζεται παρακάτω:

$$f_i = log \frac{n}{n_i} \quad (6)$$

όπου n είναι ο συνολικός αριθμός των χρηστών και n_i ο αριθμός των χρηστών που έχουν αξιολογήσει το αντικείμενο i. Συνεπώς, όταν ένα αντικείμενο i έχει αξιολογηθεί σχεδόν από όλους τους χρήστες πολλαπλασιάζεται με την αντίστροφη συχνότητα χρήστη f_i.

Περίπτωση ενίσχυσης. Οι χρήστες που είναι αρκετά όμοιοι με τον ενεργό χρήστη θα πρέπει να λαμβάνονται περισσότερο υπόψη. Για το λόγο αυτό στο [9] παρουσιάστηκε η περίπτωση ενίσχυσης όπου τα αρχικά βάρη του $p_{a,i}$ τροποποιούνται ως εξής:

$$w'_{a,u} = w_{a,u} \cdot \left|w_{a,u}\right|^{\rho-1} \quad (7)$$

όπου ρ είναι ο παράγοντας ενίσχυσης και ισχύει ότι $\rho \geq 1$.

Συνεργατικό Φιλτράρισμα με Βάση το Μοντέλο

Οι μέθοδοι συνεργατικού φιλτραρίσματος με βάση το μοντέλο λειτουργούν υπολογίζοντας τις παραμέτρους στατιστικών μοντέλων για την πρόβλεψη των αξιολογήσεων των χρηστών. Συγκεκριμένα, όπως περιγράφεται στο το συνεργατικό φιλτράρισμα με βάση το μοντέλο μπορεί να αντιμετωπιστεί ως ένα πρόβλημα ταξινόμησης, όπου τα αντικείμενα αντιστοιχούν στο διάνυσμα των χαρακτηριστικών του κάθε χρήστη και οι υπάρχουσες αξιολογήσεις στις ετικέτες. Το αποτέλεσμα είναι οι προβλεπόμενες αξιολογήσεις του κάθε ενεργού χρήστη.

Κάποιες από τις πιο δημοφιλείς τεχνικές συνεργατικού φιλτραρίσματος με βάση το μοντέλο είναι τα μοντέλα παραγοντοποίησης πίνακα και τα μοντέλα λανθάνουσας μεταβλητής. Στα μοντέλα λανθάνουσας μεταβλητής υπολογίζεται και αναπαρίσταται η ομοιότητα των χρηστών και των αντικειμένων από κάποια απλούστερη δομή δεδομένων, ενώ στα μοντέλα συνεργατικού φιλτραρίσματος με βάση τη μνήμη υπολογίζεται η ομοιότητα είτε μεταξύ των χρηστών ή μεταξύ των αντικειμένων. Η μέθοδος παραγοντοποίησης πίνακα αποτελεί μία κατηγορία της μεθόδου λανθάνουσας μεταβλητής, όπου οι χρήστες και τα αντικείμενα αναπαρίστανται από διανύσματα χαρακτηριστικών w_u και h_i αντίστοιχα, διάστασης k. Η εκπαίδευση γίνεται με σκοπό το εσωτερικό γινόμενο αυτών των διανυσμάτων $w_u^T h_i$ να

προσεγγίζει τις υπάρχουσες αξιολογήσεις $r_{u,i}$ με κάποια συνάρτηση απώλειας. Μία επιλογή συνάρτησης απώλειας είναι το τετραγωνικό σφάλμα, όπου ελαχιστοποιείται η παρακάτω συνάρτηση:

$$J(W, H, \{b_u\}_{u=1}^{n}, \{b_i\}_{i=1}^{m}) = \sum_{(u,i)\in L}\left(r_{u,i} - w_u^T h_i\right)^2 \quad (8)$$

όπου ο $W = [w_1 \dots w_n]^T$ είναι ένας πίνακας διαστάσεων $n \times k$, ο $H = [h_1 \dots h_m]$ είναι ένας πίνακας $k \times m$ και L είναι το σύνολο όλων των σημείων (u, i) όπου ο χρήστης u έχει αξιολογήσει το αντικείμενο i. Στην περίπτωση που όλοι οι χρήστες έχουν αξιολογήσει όλα τα αντικείμενα η παραπάνω συνάρτηση γίνεται:

$$J(W, H) = \|R - WH\|_{fro}^2 \quad (9)$$

όπου R είναι ο πίνακας $n \times m$, n ο αριθμός των χρηστών, m ο αριθμός των αντικειμένων και τα στοιχεία του πίνακα που εκφράζουν τις αντίστοιχες αξιολογήσεις είναι όλα γνωστά. Σε αυτή την περίπτωση μπορεί να εφαρμοστεί ο αλγόριθμος SVD στον πίνακα R με:

$$R = UDV^T, \quad W = U_k D_k^{\frac{1}{2}}, \quad H = D_k^{\frac{1}{2}} V_k^T \quad (10)$$

όπου U_k, D_k, V_k εμπεριέχουν τις μεγαλύτερες τιμές στον πίνακα R.

Όμως, στις περισσότερες περιπτώσεις οι πιο πολλές αξιολογήσεις είναι άγνωστες και οι προσεγγίσεις με βάρη είναι προτιμότερες. Στην περίπτωση αυτή η συνάρτηση απώλειας εμπεριέχει βάρη και ορίζεται ως εξής:

$$J(W, H) = \|S \odot (R - WH)\|_{fro}^2 \quad (11)$$

όπου το σύμβολο ⊙ ορίζει την πράξη του πολλαπλασιασμού κατά στοιχείο και ο S είναι ένας δυαδικός πίνακας που παίρνει την τιμή 1 όταν η αξιολόγηση είναι γνωστή και 0 διαφορετικά. Οι συνήθεις τεχνικές βελτιστοποίησης είναι gradient-based, όπως για παράδειγμα η εναλλαγή ελαχίστων τετραγώνων, όπου λύνεται ως προς το διάνυσμα H, θεωρώντας το διάνυσμα W σταθερό και αντίστροφα μέχρι να ικανοποιηθεί κάποιο κριτήριο σύγκλισης. Καθώς υπολογίζεται το ένα διάνυσμα W ή

H διατηρώντας το άλλο σταθερό, η διαδικασία είναι πλέον γραμμική παλινδρόμηση με βάρη.

Μία τεχνική για να μην υπερπροσαρμοστεί ένα μοντέλο στα δεδομένα, είναι η ελαχιστοποίηση της συνάρτησης $J(W,H)$ με κανονικοποίηση όπως φαίνεται παρακάτω:

$$J(W,H) + \gamma\|W\|^2 + \lambda\|H\|^2 \quad (12)$$

όπου γ και λ είναι παράμετροι κανονικοποίησης που καθορίζονται μέσω Διασταυρωμένης Επικύρωσης (Cross Validation). Όταν ολοκληρωθεί η εκπαίδευση του μοντέλου με τα διανύσματα W και H, το γινόμενο WH επιστρέφει ένα νέο πίνακα που περιέχει τις προβλεπόμενες αξιολογήσεις ώστε να γίνουν οι κατάλληλες συστάσεις.

2.5 Σύσταση βάσει Περιεχομένου

Στις συστάσεις συνεργατικού φιλτραρίσματος χρησιμοποιείται μόνο ο πίνακας αξιολογήσεων των χρηστών. Αυτές οι προσεγγίσεις αντιμετωπίζουν τους χρήστες και τα αντικείμενα ως ατομικές μονάδες, όπου γίνονται προβλέψεις χωρίς να λαμβάνονται υπόψη οι ιδιαιτερότητες των μεμονωμένων χρηστών ή αντικειμένων. Ωστόσο, είναι εφικτή μία καλύτερη εξατομικευμένη πρόταση αν είναι γνωστές περισσότερες πληροφορίες για ένα χρήστη όπως τα δημογραφικά στοιχεία ή για ένα αντικείμενο όπως ο σκηνοθέτης και το είδος μίας ταινίας. Οι συστάσεις βάσει περιεχομένου αναφέρονται σε τέτοιες προσεγγίσεις που παρέχουν συστάσεις συγκρίνοντας τις αναπαραστάσεις του περιεχομένου ενός αντικειμένου με τις αναπαραστάσεις περιεχομένου που ενδιαφέρουν το χρήστη.

Αρκετή από την έρευνα σε αυτόν τον τομέα έχει επικεντρωθεί στη σύσταση αντικειμένων με περιεχόμενο κειμένου, όπως ιστοσελίδες, βιβλία και ταινίες όπου οι ιστοσελίδες ή κάποιο σχετικό περιεχόμενο για παράδειγμα κάποια περιγραφή είναι διαθέσιμα μαζί με τις κριτικές των χρηστών. Επομένως, αρκετές προσεγγίσεις αντιμετώπισαν αυτό το πρόβλημα ως εργασία ανάκτησης πληροφοριών, όπου το περιεχόμενο που σχετίζεται με τις προτιμήσεις του χρήστη αντιμετωπίζεται ως ένα ερώτημα (query) και τα μη αξιολογημένα έγγραφα αξιολογούνται βάσει αυτού του ερωτήματος.

2.6 Υβριδικές προσεγγίσεις

Προκειμένου να αξιοποιηθούν τα προτερήματα του συνεργατικού φιλτραρίσματος και των συστάσεων με βάση το περιεχόμενο έχουν προταθεί αρκετές υβριδικές προσεγγίσεις που συνδυάζουν και τα δύο. Μία απλή μέθοδος είναι να δημιουργηθούν ξεχωριστές λίστες συστάσεων από τη μέθοδο συνεργατικού φιλτραρίσματος και την μέθοδο συστάσεων βάσει περιεχομένου και να συγχωνευθούν δημιουργώντας μία τελική λίστα συστάσεων .

Αρκετές άλλες υβριδικές προσεγγίσεις βασίζονται στο συνεργατικό φιλτράρισμα αλλά διατηρούν επίσης ένα προφίλ βασισμένο στο περιεχόμενο για κάθε χρήστη. Αυτά τα προφίλ βάσει του περιεχομένου χρησιμοποιούνται για την εύρεση παρόμοιων χρηστών, αντί για την αξιολόγηση αντικειμένων. Στο κάθε προφίλ χρήστη αναπαρίσταται από ένα διάνυσμα σταθμισμένων λέξεων που προέρχονται από θετικά παραδείγματα εκπαίδευσης χρησιμοποιώντας τον αλγόριθμο Winnow. Οι προβλέψεις γίνονται εφαρμόζοντας το συνεργατικό φιλτράρισμα κατευθείαν στον πίνακα των προφίλ των χρηστών (σε αντίθεση με τον πίνακα των αξιολογήσεων των χρηστών). Ορισμένες υβριδικές προσεγγίσεις προσπαθούν να συνδυάσουν άμεσα το περιεχόμενο και τα συνεργατικά δεδομένα κάτω από ένα πιθανοτικό πλαίσιο. Επεκτάθηκε το μοντέλο πτυχών του Hofmann ώστε να ενσωματώνονται τρισδιάστατα δεδομένα όπου συνυπάρχουν οι χρήστες, τα αντικείμενα και το περιεχόμενο. Το μοντέλο δημιουργίας τους προϋποθέτει ότι οι χρήστες επιλέγουν λανθάνοντα θέματα και τα έγγραφα και οι λέξεις περιεχομένου τους δημιουργούνται από αυτά τα θέματα.

2.7 Ορισμός προβλήματος

Το serendipity αφορά την καινοτομία των συστάσεων ως προς την θετική έκπληξη των χρηστών στις συστάσεις που τους γίνονται . 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.

Τα συστήματα συστάσεων που χρησιμοποιούνται κυρίως στην αγορά είναι συστήματα που χρησιμοποιούν την ομοιότητα για να προτείνουν ταινίες π.χ collaborative filtering (CB). 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 (1) ,με βάση τον περιορισμό που τίθεται από την εξίσωση (2) και (3) .

$$\max_{\mathbf{x}} \frac{1}{|\mathcal{U}|} \sum_{c=1,\ldots,C} \sum_{i \in \mathcal{I}_c} \sum_{u \in \mathcal{U}} s_{iu} x_{iu} \tag{1}$$

$$\sum_{c} \sum_{i \in \mathcal{I}_c} x_{iu} = L_s \text{ for each user } u \tag{2}$$

$$\overline{S}_c = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} \sum_{u \in \mathcal{U}} s_{iu} x_{iu} \geq \theta \text{ for all categories } c \tag{3}$$

Όπου το ϑ είναι ένα ελάχιστο ποσό της μέσης έκπληξης ανά στοιχείο που επιτυγχάνεται για τα στοιχεία της κατηγορίας *c* για κάθε χρήστη. Ισχύει για $x_{iu} \in \{0, 1\}$.

Το μέγιστο ποσό του ϑ είναι $5|u|$,το οποίο επιτυγχάνεται εάν το στοιχείο συνιστάται σε όλους τους χρήστες και επιτυγχάνει μέγιστη έκπληξη σε όλους τους.

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.

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

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 mllatest-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/. Παίρνουμε 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 mllatest-small.zip ή από το σύνολο δεδομένων ml-latest.zip . Μπορούμε να δούμε το κλάσμα:

$$p_i = \frac{\text{number of users that have rated item } i}{\text{number of users that have rated movies}} \tag{4}$$

Στη συνέχεια, μπορούμε να πάρουμε το -log10 pi για να υπολογίσουμε τη δημοτικότητα. Να σημειωθεί ότι για να υπολογίσουμε την παραπάνω ομοιότητα, Sij μεταξύ μιας ταινίας i και άλλων ταινιών που ο χρήστης έχει αξιολογήσει, θα 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).

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.

$$S_i = \frac{1}{3}(s5 + s6 + s7)\,. \tag{5}$$

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.

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 (1),(2),(3).

Κεφάλαιο 3: Πειραματική Μελέτη και Αξιολόγηση

3.1 Εισαγωγή

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 ποιες ταινίες θα προτείνει στον χρήστη. Για την τελική αξιολόγηση του συστήματος μπορούμε να δούμε τη βαθμολογία που παίρνουν οι ταινίες αυτές από τους χρήστες. Η άλλη προσέγγιση που ακολουθούμε είναι να προτείνουμε ταινίες με ένα κλασσικό αλγόριθμο χωρίς να λάβουμε υπόψιν το serendipity..Θα αξιολογήσουμε και τις δύο προσεγγίσεις.

3.2 Η Γλώσσα Προγραμματισμού Python

Η γλώσσα προγραμματισμού που επιλέχθηκε για την πειραματική μελέτη της διπλωματικής εργασίας είναι η Python (Python 3). Όπως αναφέρεται στο [19], η Python είναι μία διαδραστική αντικειμενοστραφής γλώσσα προγραμματισμού. Παρέχει δομές δεδομένων υψηλού επιπέδου όπως λίστες, λεξικά, modules, classes, αυτόματη διαχείριση μνήμης κλπ. Έχει μία εξαιρετικά απλή σύνταξη, ωστόσο είναι μία ισχυρή και γενικής χρήσης γλώσσα προγραμματισμού. Σχεδιάστηκε το 1990 από τον Guido van Rossum. Όπως και άλλες γλώσσες προγραμματισμού είναι δωρεάν, ακόμα και για εμπορικούς σκοπούς και μπορεί να εκτελεστεί σε οποιονδήποτε σύγχρονο υπολογιστή. Ένα πρόγραμμα Python μεταγλωττίζεται αυτόματα από τον διερμηνευτή (interpreter) σε ανεξάρτητη πλατφόρμα κώδικα byte που στη συνέχεια ερμηνεύεται.

3.3 Βιβλιοθήκες

Παρακάτω παρουσιάζονται οι βασικότερες βιβλιοθήκες της γλώσσας προγραμματισμού Python που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας διπλωματικής εργασίας, όπως αναφέρονται στο Python Package Index:

- **Pandas.** Η βιβλιοθήκη Pandas παρέχει γρήγορες και ευέλικτες δομές δεδομένων που έχουν σχεδιαστεί για να λειτουργούν με δομημένα δεδομένα (πολυδιάστατα, δυνητικά ετερογενή, πίνακες) και δεδομένα χρονοσειρών. Στόχος είναι το θεμελιώδες δομικό στοιχείο υψηλού επιπέδου για την πρακτική και πραγματική ανάλυση δεδομένων στην Python.
- **Numpy.** Η βιβλιοθήκη Numpy παρέχει υποστήριξη για μεγάλους πολυδιάστατους πίνακες καθώς και μαθηματικές συναρτήσεις υψηλού επιπέδου.
- **Matplotlib.** Η βιβλιοθήκη Matplotlib είναι μία ολοκληρωμένη βιβλιοθήκη για τη δημιουργία στατικών, κινούμενων και διαδραστικών απεικονίσεων.
- **Pymprog.** Η βιβλιοθήκη PyMathProg παρέχει μια εύκολη και ευέλικτη σύνταξη μοντελοποίησης χρησιμοποιώντας *Python* για τη δημιουργία και τη βελτιστοποίηση μοντέλων μαθηματικού προγραμματισμού.
- **Scikit-learn.** Το Scikit-learn είναι ένα module της Python για μηχανική μάθηση.

3.4 Περιγραφή Πειραματικής Μελέτης

Σε αυτό το κεφάλαιο παρουσιάζεται η πειραματική μελέτη που έγινε για την υλοποίηση αυτής της διπλωματικής εργασίας. Συγκεκριμένα, προτείνουμε και εφαρμόζουμε έναν νέο αλγόριθμο σε ένα σύνολο δεδομένων αναφορικά με αξιολογήσεις χρηστών για ταινίες, για τη δημιουργία μιας λίστας των αντικειμένων με βάση το serendipity, χρησιμοποιώντας διάφορες τεχνικές. Το serendipity αφορά την καινοτομία των συστάσεων ως προς την θετική έκπληξη των χρηστών στις συστάσεις που τους γίνονται . Αρχικά, παρουσιάζονται τα τεχνικά χαρακτηριστικά που επιλέχθηκαν (π.χ. η γλώσσα προγραμματισμού), έπειτα το σύνολο δεδομένων που επιλέχθηκε, στη συνέχεια τα αποτελέσματα που προέκυψαν και τέλος γίνεται η αξιολόγηση αυτών των αποτελεσμάτων.

3.5 Αξιολόγηση συστημάτων συστάσεων

3.5.1 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 :

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

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.

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

Η μετρική που επιλέχθηκε για την αξιολόγηση του συγκεκριμένου μοντέλου είναι το Mean Squared Error (MSE). Το Mean Squared Error υπολογίζει το μέσο τετραγωνικό σφάλμα, δηλαδή τον μέσο όρο του τετραγώνου της διαφοράς μεταξύ των προβλεπόμενων και των πραγματικών τιμών. Η τιμή του MSE δίνεται από τον παρακάτω τύπο:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

όπου n είναι ο συνολικός αριθμός των όρων από τους οποίους υπολογίζεται το MSE, y_i είναι οι πραγματική τιμή και $\hat{y}_i$ είναι η τιμή που πρόβλεψε το μοντέλο. Το μέσο

τετραγωνικό σφάλμα παίρνει μη αρνητικές τιμές και όσο πιο κοντά στο μηδέν βρίσκεται τόσο πιο αποδοτικό είναι το μοντέλο.

3.5.2 Ακρίβεια (Precision)

Η μονάδα μέτρησης Ακρίβεια (Precision) μετράει το ποσοστό των συστάσεων που ορθά προτάθηκαν από τον αλγόριθμο (True Positive) σε σχέση με το σύνολο των συστάσεων. Το μεγαλύτερο ποσοστό είναι το καλύτερο. Η Ακρίβεια ορίζεται λοιπόν ως ο λόγος των ορθών συστάσεων προς το σύνολο των συστάσεων, ορθών και λανθασμένων που παρήγαγε ο αλγόριθμος (True Positive + False Positive).

$$precision = \frac{|\{relevant\ recommendations\} \cap \{retrieved\ recommendations\}|}{|\{retrieved\ recommendations\}|}$$

3.5.3 Ανάκληση (Recall)

Η μονάδα μέτρησης Ανάκληση (Recall) μετράει το ποσοστό των συστάσεων που ορθά προτάθηκαν από τον αλγόριθμο (True Positive) σε σχέση με το (ιδανικό) σύνολο όλων των συστάσεων που θα μπορούσαν να προταθούν. Η Ανάκληση ορίζεται λοιπόν ως ένας λόγος που ο αριθμητής της είναι ίδιος με την Ακρίβεια (είναι και πάλι το πλήθος True Positive), αλλά διαφέρει στον παρονομαστή, όπου έχει το σύνολο των συστάσεων που προτάθηκαν ορθά και αυτών που λανθασμένα δεν προτάθηκαν (True Positive + False Negative). Επίσης οι μεγαλύτερες τιμές στην Ανάκληση, είναι οι καλύτερες.

$$recall = \frac{|\{relevant\ recommendations\} \cap \{retrieved\ recommendations\}|}{|\{relevant\ recommendations\}|}$$

3.6 Σύνολο δεδομένων

Το σύνολο δεδομένων που επιλέχθηκε για την συγκεκριμένη διπλωματική εργασία είναι το MovieLens (όπου από το link https://grouplens.org/datasets/serendipity-2018/ επιλέχθηκε το dataset `serendipity-sac2018.zip`). Από αυτό το σύνολο δεδομένων χρησιμοποιήθηκαν τα αρχεία **movies.csv** και **answers.csv** τα οποία περιέχουν 43018 και 418 εγγραφές αντίστοιχα. Το αρχείο movies.csv αυτό αποτελείται από πέντε στήλες οι οποίες είναι:

- 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».

Στόχος μας είναι να δημιουργήσουμε ένα μοντέλο που θα προβλέπει την έκπληξη του χρήστη (serendipity) για κάθε ταινία.

	movieId	title	directedBy	starring	genres
0	1	Toy Story (1995)	John Lasseter	Tim Allen, Tom Hanks, Don Rickles, Jim Varney,...	Adventure,Animation,Children,Comedy,Fantasy
1	2	Jumanji (1995)	Joe Johnston	Jonathan Hyde, Bradley Pierce, Robin Williams,...	Adventure,Children,Fantasy
2	3	Grumpier Old Men (1995)	Howard Deutch	Jack Lemmon, Walter Matthau, Ann-Margret , Sop...	Comedy,Romance
3	4	Waiting to Exhale (1995)	Forest Whitaker	Angela Bassett, Loretta Devine, Whitney Housto...	Comedy,Drama,Romance
4	5	Father of the Bride Part II (1995)	Charles Shyer	Steve Martin, Martin Short, Diane Keaton, Kimb...	Comedy

Εικόνα 8. Πρώτες εγγραφές του συνόλου δεδομένων movies

	userId	movieId	timestamp	predictedRating	s5	s6	s7
0	205229	108979	1486127833000	4.882299	2.0	2.0	5.0
1	205229	6947	1486121212000	3.253348	4.0	2.0	5.0
2	205229	117444	1486127837000	4.922837	2.0	2.0	4.0
3	205229	150548	1486127824000	4.428912	4.0	2.0	4.0
4	205229	136542	1486128075000	4.101256	1.0	2.0	5.0

Εικόνα 9. Πρώτες εγγραφές του συνόλου δεδομένων answers

3.7 Υπολογισμός ομοιότητας κάθε ταινίας για κάθε χρήστη (similarity)

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 jaccard ομοιότητας. Η κάθε ταινία έχει διαφορετική ομοιότητα για κάθε χρήστη. Χρησιμοποιούμε την jaccard ομοιότητα (jaccard similarity) με τα

χαρακτηριστικά από το αρχείο movies.csv, όπως οι κοινές λέξεις στα είδη(genres), ίδιοι πρωταγωνιστές(starring), ίδιοι σκηνοθέτες(directedBy).

3.8 Υπολογισμός δημοτικότητας (popularity)

Όπως προαναφέρθηκε η δημοτικότητα (popularity) ενός αντικειμένου μπορεί να υπολογιστεί μέσω της (κανονικοποιημένης) πιθανότητας ότι 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. Έτσι χρησιμοποιώντας την εξίσωση (4) βρίσκουμε την πιθανότητα και στη συνέχεια, χρησιμοποιούμε το -log10pi 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.

Παρακάτω βλέπουμε τις πρώτες 5 εγγραφές του συνόλου δεδομένων ratings.

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

Εικόνα 10. Πρώτες εγγραφές του συνόλου δεδομένων ratings

3.9 Training dataset

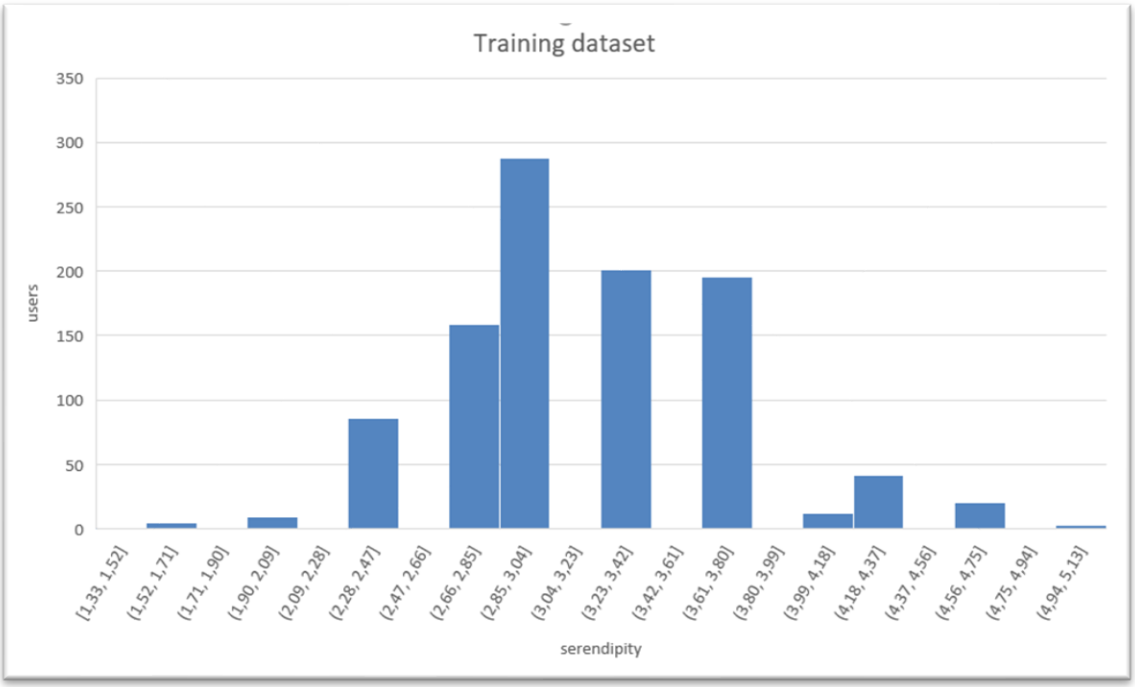
Το training dataset όπως προαναφέρθηκε αποτελείται από τα εξής χαρακτηριστικά: movieId, το similarity, το popularity και το serendipity. Στην παρακάτω εικόνα φαίνονται οι 5 πρώτες εγγραφές του:

	movieId	similarity	popularity	serendipity	userId
0	2	0.125000	0.743937	3.000000	111751
1	7	0.077381	1.052936	4.000000	118409
2	10	0.043478	0.664756	2.333333	202038
3	11	0.097744	0.940232	2.666667	144406
4	12	0.000000	1.506576	2.666667	109289

Εικόνα 11. Training dataset

3.10 Υπολογισμός serendipity μέσω της εξίσωσης

Αρχικά υπολογίζουμε το serendipity σύμφωνα με την εξίσωση (5) για όλες τις ταινίες που έχουν δει οι χρήστες. Στο παρακάτω γράφημα βλέπουμε πως κυμαίνεται το serendipity στο training dataset. Όπως παρατηρούμε η τιμή του κυμαίνεται για τους περισσότερους χρήστες/ταινίες γύρω στο 3 (από περίπου 2,85 έως 3,04). Η μικρότερη τιμή που παίρνει περίπου το 1,52 και η μεγαλύτερη το 5,13.



Εικόνα 12. Γράφημα Training dataset

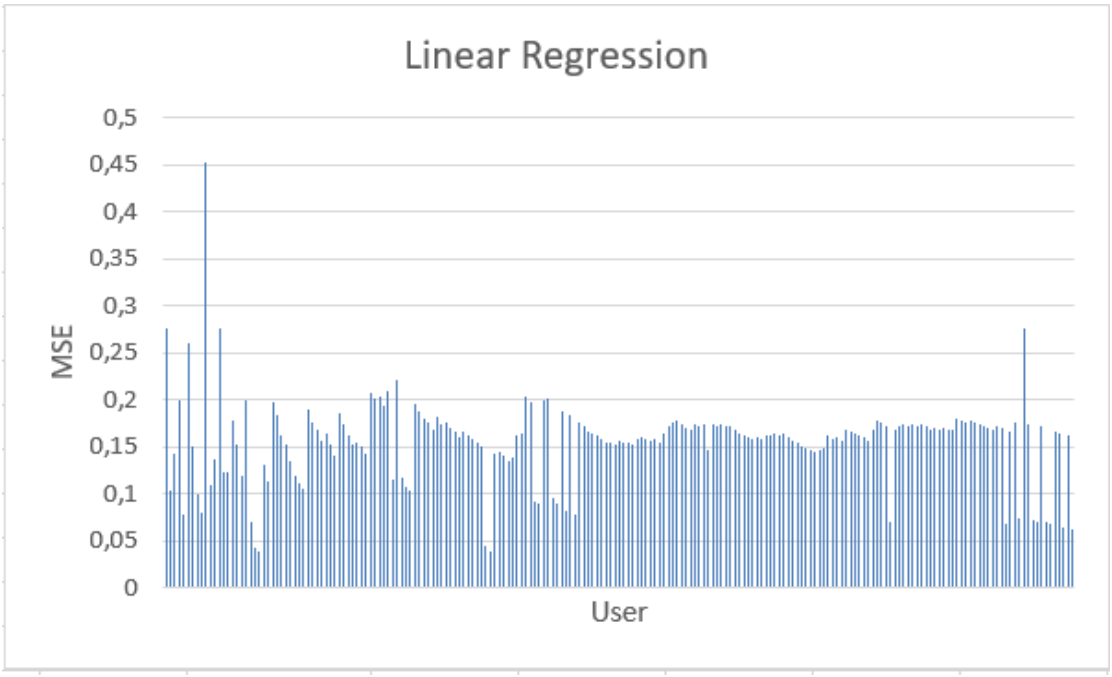
3.11 Πρόβλεψη serendipity με μοντέλο γραμμικής παλινδρόμησης

Στη στατιστική, η γραμμική παλινδρόμηση είναι μια προσέγγιση για τη μοντελοποίηση της σχέσης μεταξύ μιας βαθμωτής εξαρτημένης μεταβλητής Y και μία ή περισσότερες επεξηγηματικές μεταβλητές (ή ανεξάρτητη μεταβλητή) συμβολίζεται X. Περίπτωση μιας επεξηγηματικής μεταβλητής ονομάζεται απλή γραμμική παλινδρόμηση. Για περισσότερες από μία επεξηγηματικές μεταβλητές, η διαδικασία ονομάζεται πολλαπλή γραμμική παλινδρόμηση. (Ο όρος αυτός θα πρέπει να διακρίνεται από πολυμεταβλητή γραμμική παλινδρόμηση, όπου πολλαπλά προβλέπουν συσχέτιση με εξαρτημένες μεταβλητές , αντί για μία ενιαία βαθμωτή μεταβλητή.)

Εδώ θα χρησιμοποιήσουμε ένα μοντέλο πολλαπλής γραμμικής παλινδρόμησης με δύο χαρακτηριστικά, την ομοιότητα (similarity) και τη δημοτικότητα (popularity) της ταινίας για να προβλέψουμε την ευχάριστη έκπληξη (serendipity).

Τα δεδομένα x και y χωρίζονται σε δεδομένα εκπαίδευσης που αποτελούνται από το 80% των αρχικών δεδομένων και δεδομένα ελέγχου που αποτελούνται από το 20% των αρχικών δεδομένων. Συνεπώς, τα νέα σύνολα που προκύπτουν είναι τα σύνολα x_train και y_train που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου και τα x_test και y_test που θα χρησιμοποιηθούν για την αξιολόγηση του μοντέλου, κατά την οποία τα δεδομένα x_test θα αποτελέσουν την είσοδο του μοντέλου και τα αποτελέσματα που θα προκύψουν θα είναι οι προβλεπόμενες αξιολογήσεις. Τέλος, αυτές οι προβλεπόμενες αξιολογήσεις συγκρίνονται με τις πραγματικές τιμές (τιμές του serendipity από την εξίσωση) από το σύνολο y_test και έτσι αξιολογείται η αποτελεσματικότητα του μοντέλου που δημιουργήθηκε.

Το μέσο τετραγωνικό σφάλμα (MSE) που προκύπτει είναι 0.161, που σημαίνει ότι ο αλγόριθμός μας ήταν αρκετά ακριβής, και μπορεί να κάνει αρκετά καλές προβλέψεις. Επομένως έχει καλή εφαρμογή το μοντέλο μας, τα παρατηρούμενα σημεία δεδομένων βρίσκονται πολύ κοντά στις προβλεπόμενες τιμές του μοντέλου. Όπως βλέπουμε και στην παρακάτω εικόνα το mse ανά χρήστη είναι κυρίως κοντά στο 1,5. Ακραίες τιμές όπως 0,45 είναι απειροελάχιστες και δεν επηρεάζουν την απόδοση του μοντέλου μας.



Εικόνα 13. Γράφημα MSE/User για Linear Regression

3.12 Πρόβλεψη serendipity με μοντέλο μη γραμμικής παλινδρόμησης

Εδώ θα διερευνήσουμε την περίπτωση να προβλέψουμε το serendipity χρησιμοποιώντας ένα μοντέλο μη γραμμικής παλινδρόμησης. Επιλέξαμε το Random Forest Regressor γιατί μας έδινε τα καλύτερα αποτελέσματα. Δοκιμάσαμε και άλλους αλγόριθμους όπως το Decision Tree αλλά όχι με την ίδια επιτυχία.

Το Random Forest είναι ένας αλγόριθμος εποπτευόμενης μάθησης που χρησιμοποιεί τη μέθοδο εκμάθησης συνόλου για ταξινόμηση και παλινδρόμηση. Λειτουργεί κατασκευάζοντας ένα πλήθος δέντρων αποφάσεων κατά το χρόνο εκπαίδευσης και βγάζοντας την κλάση που είναι ο τρόπος των κλάσεων (ταξινόμηση) ή η μέση πρόβλεψη (παλίνδρομος) των μεμονωμένων δέντρων.

Ένας Random Forest αλγόριθμος είναι ένας μετα-εκτιμητής (δηλαδή συνδυάζει το αποτέλεσμα πολλαπλών προβλέψεων) που συγκεντρώνει πολλά δέντρα απόφασης , με ορισμένες χρήσιμες τροποποιήσεις:

Ο αριθμός των χαρακτηριστικών που μπορούν να διαχωριστούν σε κάθε κόμβο περιορίζεται σε κάποιο ποσοστό του συνόλου (το οποίο είναι γνωστό ως υπερπαράμετρος). Αυτό διασφαλίζει ότι το μοντέλο συνόλου δεν βασίζεται σε μεγάλο βαθμό σε κανένα μεμονωμένο χαρακτηριστικό και κάνει ορθή χρήση όλων των πιθανών προγνωστικών χαρακτηριστικών .

Κάθε δέντρο αντλεί ένα τυχαίο δείγμα από το αρχικό σύνολο δεδομένων κατά τη δημιουργία των διαχωρισμών του, προσθέτοντας ένα επιπλέον στοιχείο τυχαίας που αποτρέπει την υπερβολική προσαρμογή .

Οι παραπάνω τροποποιήσεις βοηθούν στην αποφυγή υπερβολικής συσχέτισης των δέντρων.

Και εδώ χρησιμοποιούμε σαν εξαρτημένη μεταβλητή το 'serendipity' και σαν ανεξάρτητες τα 'popularity' και 'similarity'. Διαχωρίζουμε τα δεδομένα σε train και test σε ποσοστά 80% και 20% αντίστοιχα των αρχικών δεδομένων.

Το μέσο τετραγωνικό σφάλμα (MSE) βγήκε 0.285, που σημαίνει ότι ο αλγόριθμός μας δεν είναι και πολύ ακριβής, άρα δεν μπορεί να κάνει αρκετά καλές προβλέψεις. Επομένως δεν έχει και τόσο έχει καλή εφαρμογή το μοντέλο μας, τα παρατηρούμενα σημεία δεδομένων δεν βρίσκονται τόσο κοντά στις προβλεπόμενες τιμές του μοντέλου.

Τέλος, συγκρίνοντας τους δυο αλγόριθμους βλέπουμε ότι ο Linear Regression είναι πιο ακριβής και συνεπώς αυτόν θα χρησιμοποιήσουμε και παρακάτω για τις προβλέψεις μας.

Αλγόριθμος	**MSE**
Linear Regression	0,161
Random Forest	0,285

Εικόνα 14. MSE τιμή σε Linear και random forest regression

3.13 Πρόβλεψη serendipity για ταινίες που δεν έχουν δει οι χρήστες με το μοντέλο γραμμικής παλινδρόμησης

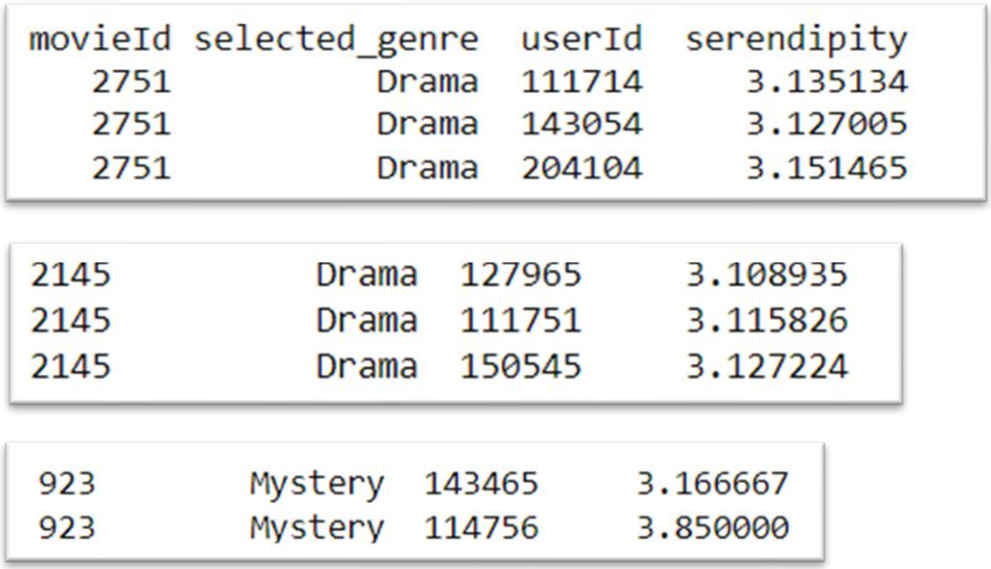
Σε αυτό το σημείο θα χρησιμοποιήσουμε ένα νέο dataset το οποίο έχουμε δημιουργήσει, το 'recommended' dataset ,και περιλαμβάνει ταινίες που δεν έχουν δει οι χρήστες, για να εκπαιδεύσουμε το μοντέλο μας (Linear Regression), με σκοπό την εύρεση του serendipity για τις ταινίες που δεν έχουν δει οι χρήστες.

Για σκοπούς έρευνας και για να μπορέσουμε να αξιολογήσουμε τα αποτελέσματα μας παρακάτω θα χρησιμοποιήσουμε το 10% του συνόλου δεδομένων μας (48 χρήστες) και θα υποθέσουμε για αυτό το ποσοστό ότι δεν έχουν δει τις ταινίες που έχουν βαθμολογήσει. Αυτοί οι 48 χρήστες μαζί με τις ταινίες που δεν έχουν δει αλλά και τις ταινίες που έχουν πραγματικά δει και βαθμολογήσει (αλλά υποθέτουμε ότι δεν έχουν δει) θα είναι το σύνολο ελέγχου των δεδομένων μας. Αυτό, θα μας βοηθήσει παρακάτω να δούμε κατά πόσο θα είμαστε επιτυχείς στο να προτείνουμε ταινία που τελικά αρέσει στον χρήστη. Δηλαδή να δούμε αυτές τις ταινίες που του προτείνουμε πόσο τις είχε βαθμολογήσει?

Το recommended dataset αποτελείται όπως και το training dataset που χρησιμοποιήσαμε παραπάνω, από τις στήλες:

- 'movieId', κωδικός ταινίας
- 'selected_genre', είδος ταινίας (π.χ κωμωδία, δράμα κτλ.)
- 'popularity', υπολογίστηκε ξανά σύμφωνα με την εξίσωση (4)
- 'similarity', υπολογίστηκε ξανά με το Jaccard similarity
- 'UserId', κωδικός χρήστη

Χρησιμοποιώντας λοιπόν το μοντέλο μας όπως στην παράγραφο 4.11, προβλέπουμε το serendipity, και το αποτέλεσμα φαίνεται παρακάτω στο τελικό μας dataset:



movieId	selected_genre	userId	serendipity
2751	Drama	111714	3.135134
2751	Drama	143054	3.127005
2751	Drama	204104	3.151465
2145	Drama	127965	3.108935
2145	Drama	111751	3.115826
2145	Drama	150545	3.127224
923	Mystery	143465	3.166667
923	Mystery	114756	3.850000

Εικόνα 15. Recommended dataset

Βλέπουμε πως αλλάζει το serendipity στις ίδιες ταινίες για διαφορετικούς χρήστες. Όπως και πως αλλάζει και για διαφορετικές ταινίες.

3.14 Πρόβλεψη ταινιών με βάση το serendipity χρησιμοποιώντας τον solver (Πείραμα 1)

Όπως αναφέρθηκε και παραπάνω 4.1 , σκοπός είναι να κάνουμε συστάσεις ταινιών με στόχο να βελτιστοποιήσουμε την έκπληξη του χρήστη. Το serendipity εκφράζει το

βαθμό της ευχάριστης έκπληξης που προκαλείται όταν ένα αντικείμενο (π.χ ταινία) προτείνεται σε ένα χρήστη.

Θα επιλύσουμε το πρόβλημα της βελτιστοποίησης (εξισώσεις (1),(2),(3)) για ακέραιες και για συνεχείς τιμές που ανήκουν ανάμεσα στο 0 και στο 1.

Το πρόβλημα είναι γραμμικό. Ο Γραμμικός Προγραμματισμός (LP) είναι μια μέθοδος για να φτάσουμε σε μια βέλτιστη λύση ενός προβλήματος με την επίλυση μιας γραμμικής εξίσωσης.

Ο Γραμμικός Προγραμματισμός θα εξετάσει ένα πρόβλημα και θα το μετατρέψει σε μια μαθηματική εξίσωση χρησιμοποιώντας μεταβλητές όπως *x* και *y*. Μετά από αυτό, είναι θέμα δοκιμής αριθμών για αυτές τις μεταβλητές μέχρι να φτάσουμε στην καλύτερη λύση.

Υπάρχουν διάφορα πακέτα επίλυσης LP διαθέσιμα στην Python. Μεταξύ αυτών είναι τα SciPy, PuLP, CVXOPT. Σε αυτό το πείραμα, εργαζόμαστε με GLPK.

Ο σκοπός μας είναι να κάνουμε τέτοιες προτάσεις ταινιών στους χρήστες, ώστε να τους προκαλέσουμε την μεγαλύτερη δυνατή ευχάριστη έκπληξη (serendipity). Αυτό θα επιτευχθεί με την εξίσωση (1). Υπάρχουν όμως κάποιοι περιορισμοί σύμφωνα με τις εξισώσεις (2) και (3). Το Ls, που είναι ο αριθμός των ταινιών που θα προτείνουμε σε κάθε χρήστη, και το ϑ.

Μπορούμε να ορίσουμε το ϑ ως το κάτω όριο του μέσου serendipity που απαιτούμε το σύστημα να επιτύχει για κάθε κατηγορία ταινίας. Στην ουσία το ϑ εξασφαλίζει ότι θα μας προταθούν ταινίες από όλα τα είδη που δεν έχουμε δει. Ένα χαμηλό ϑ (min) είναι στην ουσία το μέσο serendipity μιας ταινίας σε μία κατηγορία (genre). Αντίθετα, υψηλό ϑ (max), δηλαδή κοντά στο 3,2, σημαίνει ότι σε όλες τις κατηγορίες ταινιών θα επιτευχθεί υψηλό serendipity και άρα καλύτερες λύσεις το οποίο και επιδιώκουμε.

Θα πρέπει να χωρίσουμε τα δεδομένα μας σε κατηγορίες. Επιλέγουμε αυτές οι κατηγορίες να είναι το selected_genres, δηλαδή τα είδη των ταινιών (κωμωδία, δράμα κτλ.). Έχουμε προβλέψει ήδη παραπάνω το serendipity για τις ταινίες που δεν έχει δει ο κάθε χρήστης.

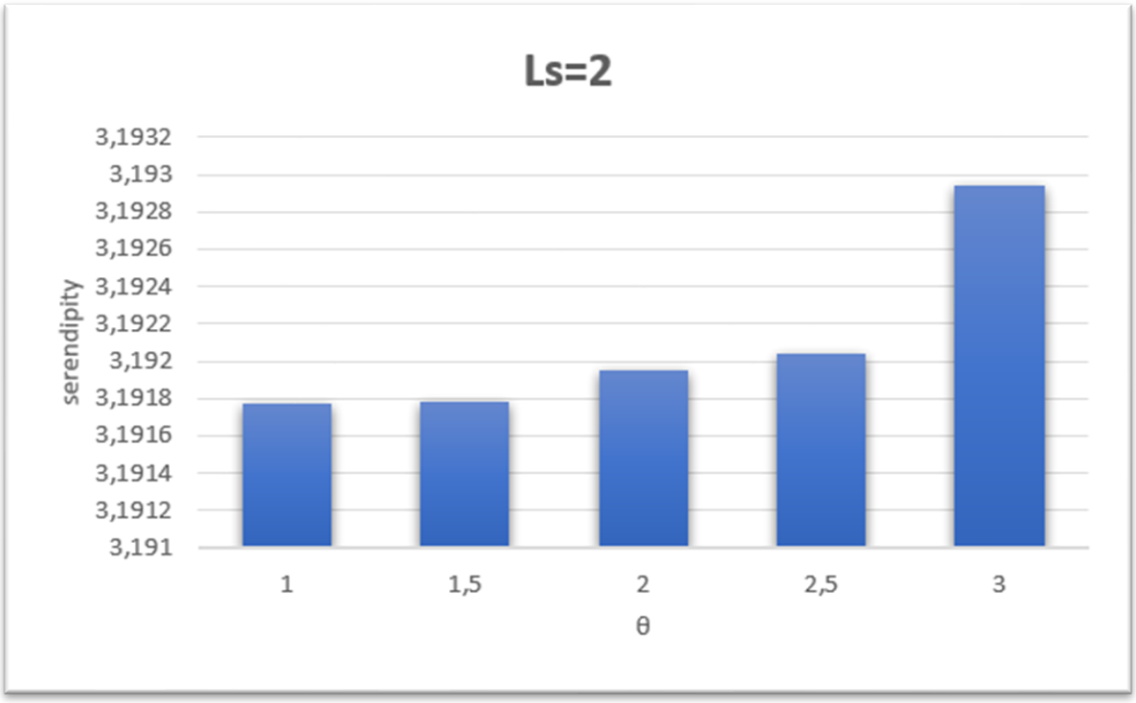
Θα χρησιμοποιήσουμε τον solver για Ls = 2 και για Ls = 3, δηλαδή θα προτείνουμε σε κάθε χρήστη τη μια φορά 2 ταινίες και την άλλη 3. Τέλος, θα επιλύσουμε το παραπάνω πρώτα για ακέραιες τιμές Xiu ϵ {0,1} ,και ορίζουμε ότι όταν x=1 προτείνω την ταινία ενώ όταν x=0 δεν την προτείνω.

Παρακάτω, βρίσκουμε το total average serendipity για θ = 1, 1.5, 2, 2.5, 3 και αναλύουμε τα αποτελέσματα:

Ls = 2

θ	S
1	3,191772
1,5	3,191786
2	3,191955
2,5	3,192042
3	3,192944

Εικόνα 16. Πίνακας - Συνολικό Μέσο serendipity για κάθε τιμή του θ (Ls = 2)



Εικόνα 17. Γράφημα - Συνολικό Μέσο serendipity για κάθε τιμή του θ (Ls = 2)

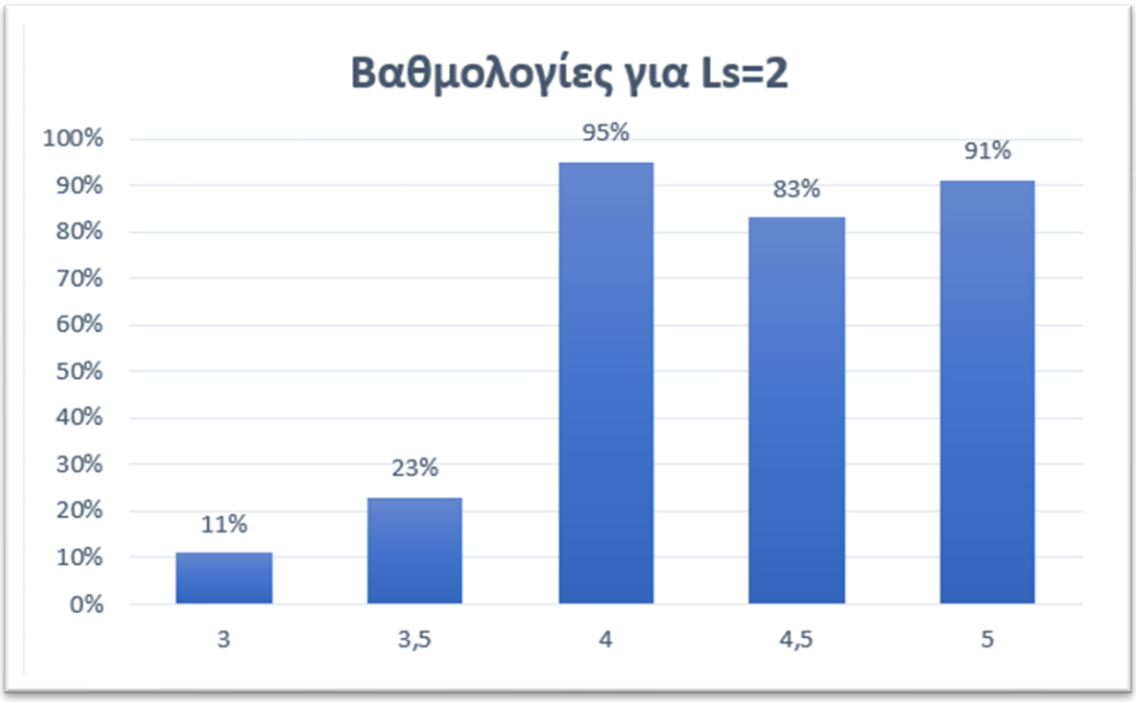
Όπως παρατηρούμε και από τον πίνακα και το γράφημα παραπάνω, το συνολικό μέσο serendipity για Ls = 2, δεν διαφέρει σχεδόν καθόλου για τις διάφορες τιμές του θ. Η μικρότερη τιμή του είναι 3,191772 για θ = 1 και η μεγαλύτερη 3,192944 για θ = 3. Επιπλέον έστω και ελάχιστα όσο μεγαλώνει το θ μεγαλώνει και το serendipity, αφού όπως είπαμε και παραπάνω, όσο πιο μεγάλο το θ τόσο καλύτερα τα αποτελέσματα. Τέλος, οι τιμές του serendipity πάλι είναι κοντά στο 3 όλες. Είναι λογικό αφού ο solver δεν πειράζει την τιμή του serendipity , παρά χρησιμοποιεί τις

ήδη προβλεπόμενες τιμές του serendipity από το σύνολο δεδομένων recommended και αναλόγως τα Ls και θ μας προτείνει ταινίες σύμφωνα με το μέσο serendipity που υπολογίζει. Όπως είδαμε και στην προηγουμένως στην εικόνα 14, σχεδόν όλες οι τιμές του serendipity στο recommender dataset είναι κοντά στο 3.

Ένα πράγμα ακόμα που θέλουμε να εξετάσουμε είναι αν τελικά οι συστάσεις ταινιών ικανοποίησαν τους χρήστες. Αυτό θα υπολογιστεί με βάση τις βαθμολογίες που είχαν δώσει οι χρήστες στις ταινίες τους προτείναμε. Οι βαθμολογίες των χρηστών κυμαίνονται που εξετάζουμε κυμαίνονται από 3 έως 5. Θα χρησιμοποιήσουμε ξανά τον παρακάτω τύπο:

$$Ποσοστό\ επιτυχίας = \frac{αριθμός\ προτάσεων\ που\ ικανοποιούν\ την\ συνθήκη}{συνολικός\ αριθμός\ προτάσεων} * 100$$

Στην Εικόνα 18 φαίνονται τα ποσοστά επιτυχίας όταν το μοντέλο προτείνει δύο ταινίες σε κάθε χρήστη (Ls=2). Πιο συγκεκριμένα τα ποσοστά επιτυχούς πρόβλεψης ανάλογα με τη βαθμολογία που έχει δώσει. Όπως φαίνεται το ποσοστό επιτυχίας ανέρχεται στο 95%, όταν προτείνονται ταινίες που έχει δει ο χρήστης και έχουν πάρει βαθμολογία τέσσερα, για βαθμολογία πέντε είναι 91%. Για ταινίες που έχουν βαθμολογηθεί με τρεισήμισι το ποσοστό φτάνει το 23%. Αντίστοιχα για ταινίες που έχουν βαθμολογηθεί με τρία το ποσοστό φτάνει το 11%. Τέλος για ταινίες με βαθμολογία 4.5 το ποσοστό είναι 83%.



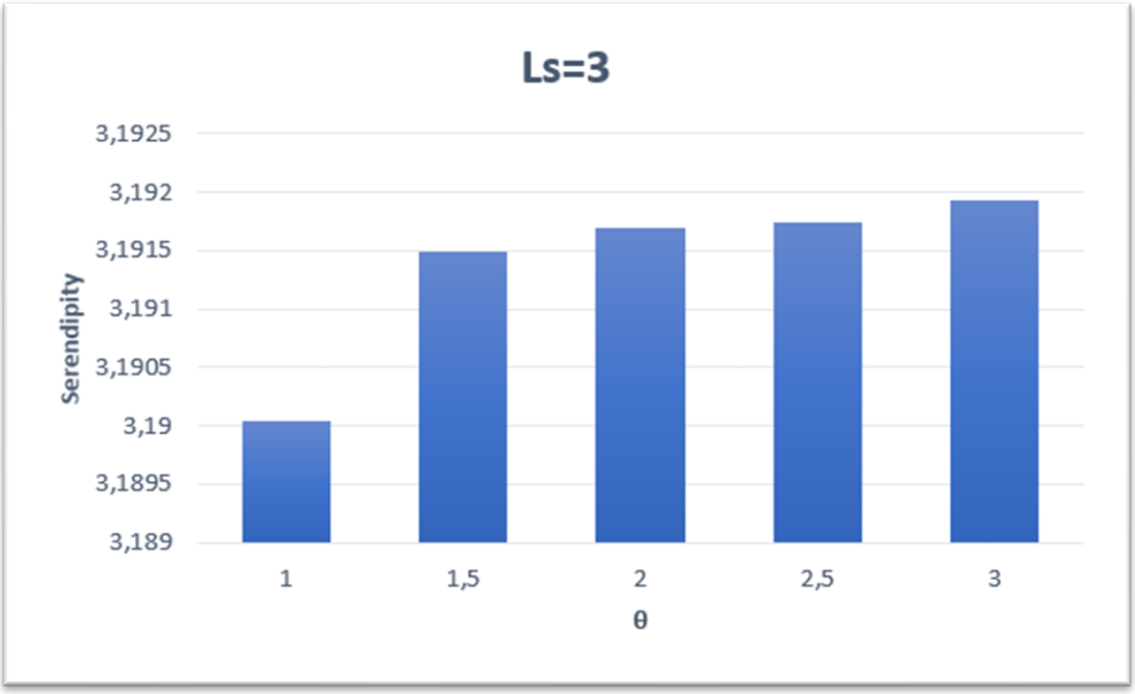
Εικόνα 18. Γράφημα – Βαθμολογίες ταινιών για Ls = 2

Ομοίως για Ls = 3:

Ls = 3

θ	S
1	3,190043
1,5	3,191482
2	3,191700
2,5	3,191741
3	3,191925

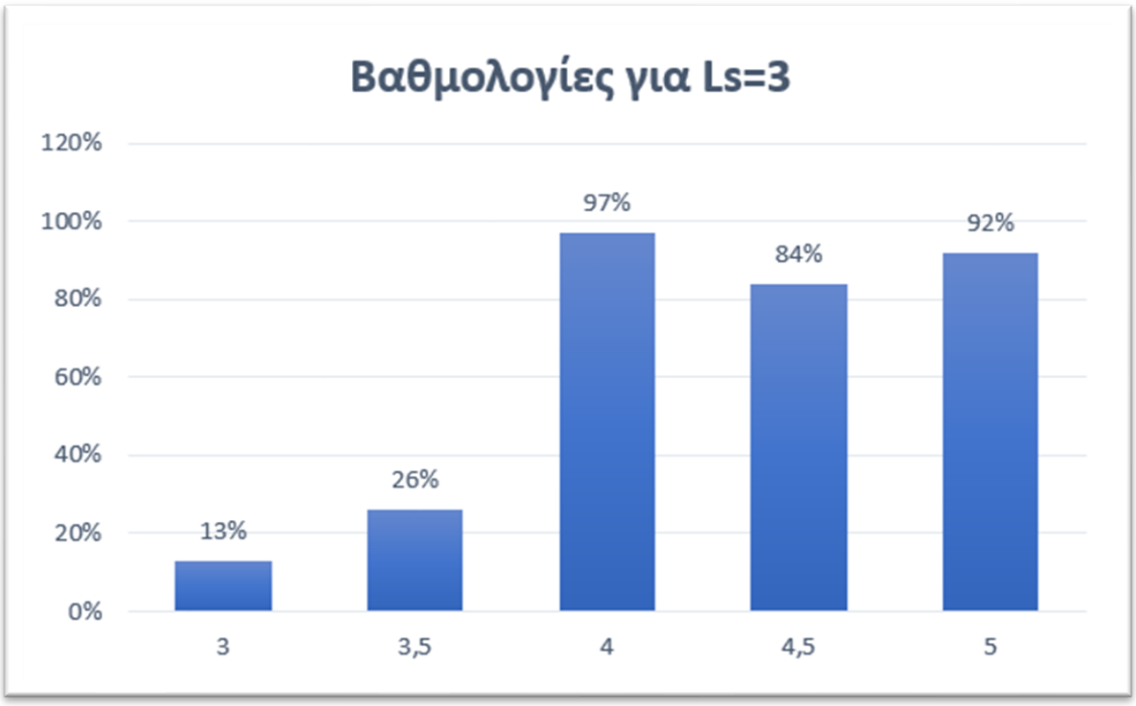
Εικόνα 19. Πίνακας - Συνολικό Μέσο serendipity για κάθε τιμή του θ (Ls = 3)



Εικόνα 20. Γράφημα - Συνολικό Μέσο serendipity για κάθε τιμή του θ (Ls = 3)

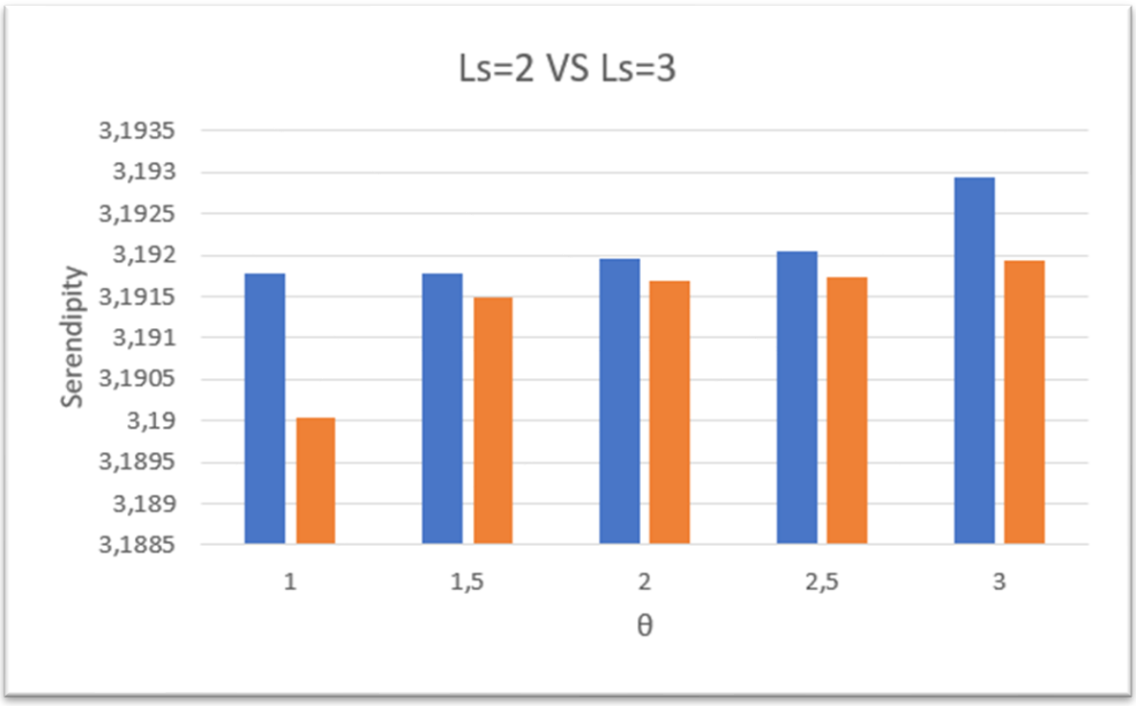
Το ίδιο συμβαίνει και όταν το Ls = 3. Οι τιμές του serendipity είναι πολύ παρόμοιες και κυμαίνονται όπως πριν για Ls=2 για τα διάφορα θ. Η μικρότερη τιμή του είναι 3,190043 για θ = 1 και η μεγαλύτερη 3,191925 για θ = 3. Και εδώ λοιπόν ακολουθείται το ίδιο μοτίβο όπως προηγουμένως.

Στην Εικόνα 21 παρακάτω φαίνονται τα αποτελέσματα όταν το μοντέλο προτείνει τρεις ταινίες σε κάθε χρήστη(Ls=3). Όπως φαίνεται το ποσοστό επιτυχίας ανέρχεται στο 97%, όταν προτείνονται ταινίες που έχει δει ο χρήστης και έχουν πάρει βαθμολογία τέσσερα, για βαθμολογία πέντε είναι 92%. Για ταινίες που έχουν βαθμολογηθεί με τρεισήμισι το ποσοστό φτάνει το 26%. Αντίστοιχα για ταινίες που έχουν βαθμολογηθεί με τρία το ποσοστό φτάνει το 13%. Τέλος για ταινίες με βαθμολογία 4.5 το ποσοστό είναι 84%.



Εικόνα 21. Γράφημα – Βαθμολογίες ταινιών για Ls = 3

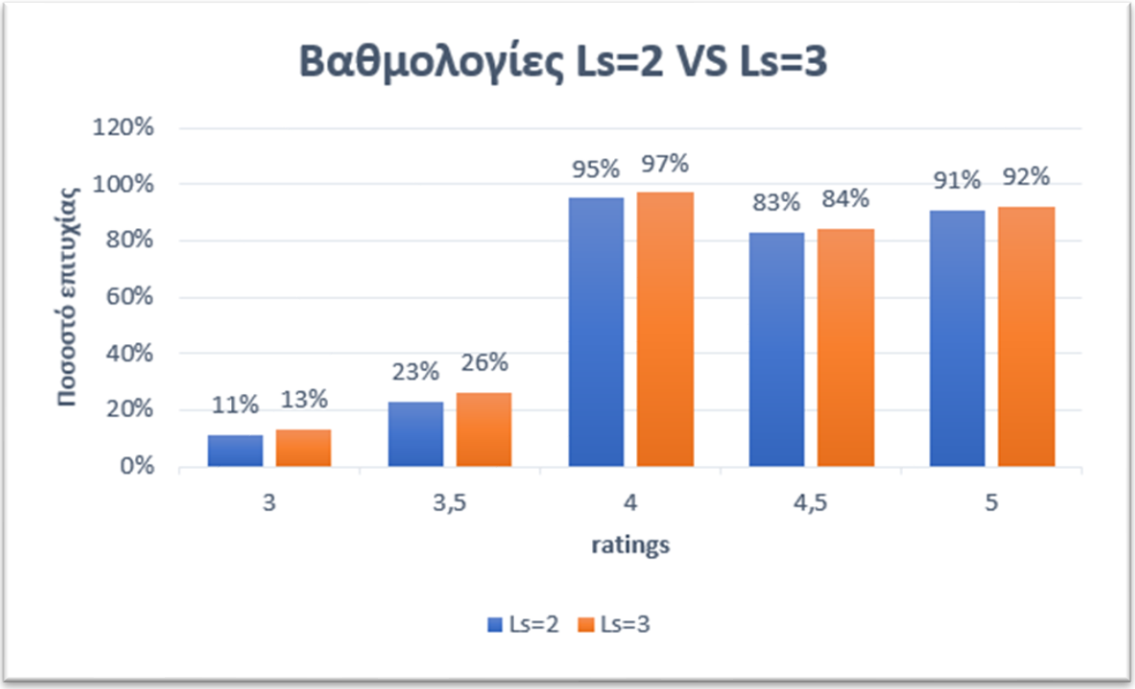
Έχει ενδιαφέρον να κάνουμε σύγκριση των αποτελεσμάτων του solver μεταξύ τους για Ls = 2 και Ls = 3 , όταν δηλαδή προτείνει δυο ή τρεις ταινίες στον χρήστη.



Εικόνα 22. Γράφημα - Συνολικό Μέσο serendipity για κάθε τιμή του θ (Ls = 2 VS Ls = 3)

Όπως φαίνεται στην Εικόνα 23, το μέσο serendipity, ανάλογα με τον αν το σύστημα προτείνει δυο ή τρεις ταινίες στο χρήστη είναι πολύ κοντά, ειδικά όταν θ = 3, που είναι και πολύ κοντά στο μέσο serendipity όλων των ταινιών που το βρήκαμε παραπάνω 3,2. Στο μόνο σημείο που υπάρχει κάποια ίσως μεγαλύτερη διαφορά είναι για θ=1. Επομένως είτε το σύστημα προτείνει δύο είτε τρεις ταινίες μας φέρνει σχεδόν τα ίδια αποτελέσματα όσον αφορά το θ.

Στη συνέχεια κάνουμε σύγκριση των αποτελεσμάτων του solver, όταν προτείνει δύο ή τρεις ταινίες(Ls=2 ή Ls=3) με βάση το serendipity που υπολογίστηκε από το γραμμικό μοντέλο παλινδρόμησης. Στο παρακάτω διάγραμμα φαίνονται αυτά τα αποτελέσματα.



Εικόνα 23. Γράφημα – Βαθμολογίες ταινιών (Ls = 2 VS Ls = 3)

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

Τέλος, να σημειώσουμε ότι κάνοντας όλη την ίδια διαδικασία του πειράματος για συνεχείς τιμές μας φέρνει τα ίδια αποτελέσματα.

3.15 Πρόβλεψη ταινιών με collaborative filtering (Πείραμα 2)

Εδώ θα εξετάσουμε μια κλασσική μηχανή σύστασης ταινιών με που λειτουργεί με βάση την ομοιότητα μεταξύ δύο οντοτήτων και σύμφωνα με αυτό μας δίνει το αποτέλεσμα.

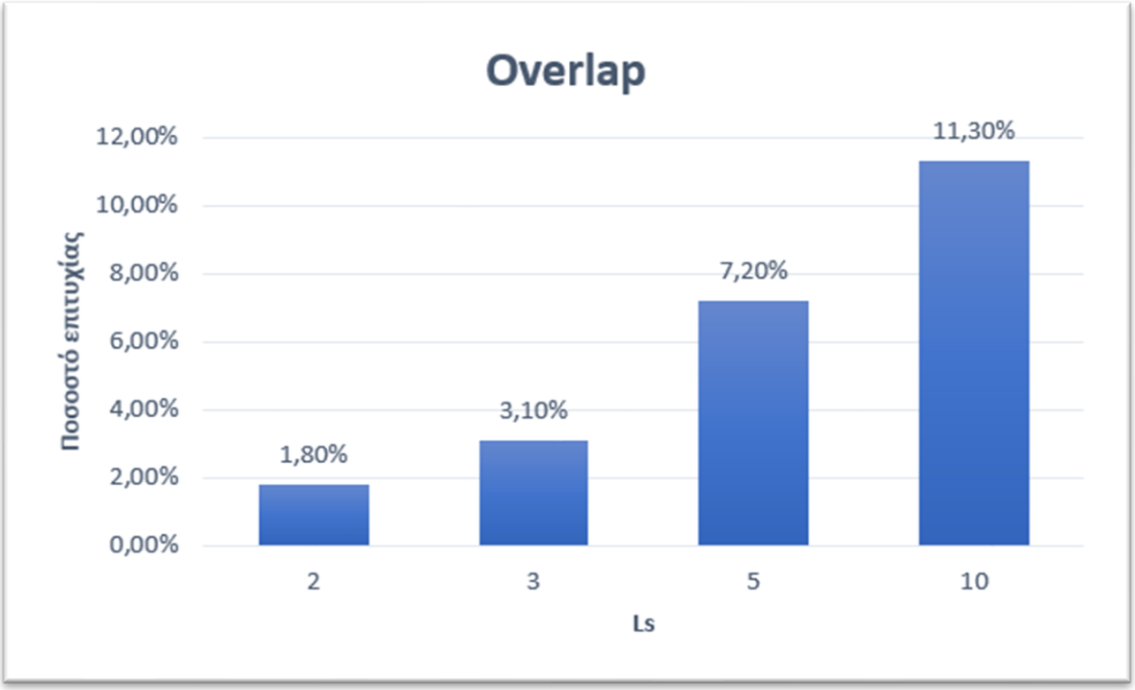
Πιο συγκεκριμένα, θα εξετάσουμε το Συνεργατικό φιλτράρισμα και μάλιστα βάσει χρήστη. Στο συνεργατικό φιλτράρισμα βάσει χρήστη, ανακαλύπτουμε τη βαθμολογία ομοιότητας μεταξύ των δύο χρηστών. Με βάση τη βαθμολογία ομοιότητας, προτείνουμε τα στοιχεία που αγόρασε/αρέσουν ένας χρήστης σε άλλο χρήστη, υποθέτοντας ότι μπορεί να του αρέσουν αυτά τα στοιχεία βάσει ομοιότητας. Αυτό θα είναι πιο ξεκάθαρο όταν προχωρήσουμε και το εφαρμόσουμε. Η μεγάλη διαδικτυακή υπηρεσία ροής, το **Netflix** έχει τη μηχανή συστάσεων που βασίζεται σε συνεργατικό φιλτράρισμα βάσει χρήστη.

Θα χρησιμοποιήσουμε το ίδιο σύνολο δεδομένων (48 χρήστες) μόνο που αυτή την φορά δεν θα «κρύψουμε» τις ταινίες που έχουν δει και βαθμολογήσει. Έχουμε ήδη βρει στο προηγούμενο πείραμα ποιες ταινίες δεν έχουν δει πραγματικά αυτοί οι χρήστες (recommended dataset). Θα χρησιμοποιήσουμε τον solver για να μας προτείνει ταινίες με βάση το serendipity που θα τους προκαλέσουν έκπληξη. Περιμένουμε πως αυτές οι ταινίες θα είναι τελείως διαφορετικές σε σχέση με το είδος των ταινιών που έχουν ήδη δει και βαθμολογήσει.

Στη συνέχεια, χρησιμοποιώντας το συνεργατικό φιλτράρισμα και μάλιστα τον SVD αλγόριθμο (Singular Value Decomposition) πάλι με το ίδιο σύνολο δεδομένων θα εφαρμόσουμε διάφορα είδη τεχνικών για να ανακαλύψουμε ομοιότητες μεταξύ των χρηστών, τις πιο δημοφιλείς ταινίες και εξατομικευμένες προτάσεις για τον στοχευμένο χρήστη. Εδώ περιμένουμε συστάσεις παρόμοιες με τις ταινίες που έχει δει ο χρήστης.

Εν κατακλείδι, αυτό που θέλουμε να δούμε είναι τι ποσοστό επικαλύψεων υπάρχει στις ταινίες που προτείνονται ανάμεσα στα δύο συστήματα.

Θα συμφωνήσουμε ότι έστω και μία ταινία να βρεθεί κοινή στις προτεινόμενες ταινίες για κάθε χρήστη, ανάμεσα στα δυο συστήματα συστάσεων θα υποθέσουμε ότι υπάρχει overlap. Κάναμε τα πειράματα μας και για τους 48 χρήστες και τους προτείναμε 2,3,5 ή 10 ταινίες. Παρακάτω βλέπουμε τα αποτελέσματα:



Εικόνα 24. Overlap ταινιών για Ls = 2, 3, 5, 10.

Παρατηρούμε ότι όταν προτείνουμε 2 ταινίες υπάρχει overlap 1.8%, στις 3 ταινίες 3,1%, στις 5 ταινίες 7,2% και στις 10 ταινίες 11,3%. Δηλαδή όσο αυξάνεται ο αριθμός των προτεινόμενων ταινιών τόσο αυξάνεται και το overlap.Είναι λογικό αφού όσο περισσότερες οι συστάσεις τόσο μεγαλύτερη η πιθανότητα να πετύχουμε έστω μία κοινή ταινία. Είναι σημαντικό επίσης να αναφερθεί ότι μόνο μια φορά πετύχαμε για ένα χρήστη πάνω από μια κοινή πρόταση ταινίας, κι αυτό ήταν 2 κοινές ταινίες. Ήταν αναμενόμενο αφού με το collaborative filtering προτείνουμε παρόμοιες ταινίες με αυτές που έχει δει ο χρήστης ενώ χρησιμοποιώντας τον solver και το serendipity προτείνουμε εντελώς διαφορετικές από αυτές που έχει δει, που θα του προκαλέσουν όμως ευχάριστη έκπληξη.

Κεφάλαιο 4: Συμπεράσματα και Μελλοντική Εργασία

Το κεφάλαιο αυτό περιλαμβάνει τα συμπεράσματα που εξήχθησαν κατά την εκπόνηση αυτής της διπλωματικής εργασίας και παρουσιάζει πιθανές μελλοντικές επεκτάσεις.

4.1 Σύνοψη

Σε αυτή την διπλωματική εργασία εξετάστηκε η προσέγγιση του serendipity (έκπληξης) στα συστήματα συστάσεων.

Στη συνέχεια, παρουσιάστηκε η υλοποίηση που έγινε σε αυτή την εργασία.

Τέλος, περιγράφηκε το πλαίσιο της πειραματικής μελέτης που έγινε, τα τεχνικά μέσα που χρησιμοποιήθηκαν και ο αλγόριθμος που αναπτύχθηκε, παρουσιάστηκαν τα αποτελέσματα που προέκυψαν και έγινε η αξιολόγηση.

4.2 Συμπεράσματα

Συμπερασματικά, μέσα από το θεωρητικό υπόβαθρο της διπλωματικής εργασίας, την υλοποίηση την πειραματική μελέτη αλλά και την τελική αξιολόγηση προσπαθήσαμε να διερευνήσουμε και να αξιολογήσουμε μέσω διαφόρων μετρικών το serendipity και πως αυτό επηρεάζει τα συστήματα συστάσεων. Δουλέψαμε πάνω σε πραγματικά δεδομένα. Δημιουργήσαμε το dataset με βάση το οποίο υπολογίσαμε το serendipity αρχικά για τις ταινίες που έχουν δει οι χρήστες. Συγκρίναμε τα αποτελέσματα αυτά με ένα μοντέλο γραμμικής και ένα μη γραμμικής παλινδρόμησης και παρατηρήθηκε ότι η τιμή του Μέσου Τετραγωνικού Σφάλματος ήταν μεγαλύτερη στο μη γραμμικό μοντέλο. Όσον αφορά το γραμμικό μοντέλο, τα αποτελέσματα που προέβλεψε ήταν πολύ κοντά στα πραγματικά. Στη συνέχεια, υλοποιήσαμε 2 πειράματα. Στο πρώτο πείραμα διαλέξαμε 48 χρήστες και για αυτούς υποθέσαμε ότι δεν έχουν δει τις ταινίες που έχουν δει και μαζί με αυτές που δεν έχουν δει πραγματικά οι χρήστες δημιουργήσαμε ξανά ένα νέο dataset. Χρησιμοποιώντας το μοντέλο γραμμικής παλινδρόμησης ξανά, προβλέψαμε το serendipity για ταινίες αυτές. Τέλος, προτείναμε ταινίες που θα μεγιστοποιήσουν την έκπληξη στον χρήστη μέσω του Glpk solver. Συγκεκριμένα, κάναμε συστάσεις είτε δυο είτε τριών ταινιών και

παρατηρήσαμε ότι δεδομένου διαφόρων τιμών του θ το μέσο συνολικό serendipity των χρηστών είναι σχεδόν το ίδιο για όλες τις τιμές όπως επίσης και το ποσοστό επιτυχίας πρόβλεψης ταινιών . Αξιολογήσαμε τις προτεινόμενες ταινίες βάση των ratings και παρατηρήσαμε ότι από τις ταινίες που προτείναμε για Ls=2 το μεγαλύτερο ποσοστό είχαν αυτές με βαθμολογία από 4 και πάνω. Αυτό συνέβη γιατί όταν του προτείνουμε ταινία που έχει ήδη δει (κι εμείς την έχουμε «κρύψει») η βαθμολογία που της είχε δώσει να είναι μεγάλη. Το ποσοστό ανέβαινε για Ls=3. 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. Στο δεύτερο πείραμα χρησιμοποιήσαμε ένα κλασσικό σύστημα συστάσεων (collaborative filtering) για να κάνουμε συστάσεις 2,3,5 και 10 ταινιών σε κάθε χρήστη. Το ίδιο κάναμε και με τον solver. Τα αποτελέσματα που πήραμε συγκρίνοντας τα 2 συστήματα είναι ότι όσο αυξάνεται ο αριθμός συστάσεων τόσο αυξάνεται το overlap.Παρόλα αυτά το overlap γινόταν σε πολύ μικρό ποσοστό , πράγμα αναμενόμενο αφού τα δυο συστήματα προτείνουν ταινίες με εντελώς αντίθετο τρόπο.

4.3 Μελλοντικές επεκτάσεις

Μία μελλοντική επέκταση αυτής της πειραματικής μελέτης θα μπορούσε να είναι να χρησιμοποιηθεί σαν τρίτο χαρακτηριστικό το predicted rating μιας ταινίας στο training dataset και κατ’ επέκταση και στην πρόβλεψη του serendipity είτε από το μοντέλο (π.χ linear regression) είτε από τη συνάρτηση εκτίμησης. Θα μπορούσαμε να δούμε κατά πόσο επηρεάζονται τα αποτελέσματα, και να τα συγκρίνουμε με τα ratings που έχουν δώσει οι χρήστες. Επιπλέον θα μπορούσαμε να χρησιμοποιήσουμε κάποιον άλλο solver π.χ τον PuLP ,για να δούμε κατά πόσο διαφέρουν τα αποτελέσματα του σε σχέση με τον GLPK solver.

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].46 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].47https://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,