



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Πτυχιακή Εργασία
Σύστημα προσωποποιημένων προτάσεων
για ταινίες

Πέτρος – Αθανάσιος Παπαντωνόπουλος
(ΑΜ: 711171069)

Επιβλέπων: Χρήστος Τρούσσας

Αιγάλεω
Σεπτέμβριος 2022

Επιτροπή Εξέτασης

- Χρήστος Τρούσσας

- Παναγιώτα Τσελέντη

- Γεώργιος Μελετίου

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Παπαντωνόπουλος Πέτρος-Αθανάσιος, με αριθμό μητρώου 711171069 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο/Η Δηλών/ούσα



Ευχαριστίες – Αφιερώσεις

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω όλους όσους συνέβαλλαν με την βοήθεια και την υποστήριξη τους στην εκπόνηση αυτής της πτυχιακής εργασίας.

Αρχικά θα ήθελα να ευχαριστήσω τον Δρ. Τρούσσα Χρήστο για την επίβλεψη, την καθοδήγηση και την άμεση ανταπόκριση σε οποιοδήποτε ερώτημα μου, σε όλη την διάρκεια της εκπόνησης της πτυχιακής εργασίας.

Επιπλέον θα ήθελα να ευχαριστήσω όλους τους συνάδελφους που με στήριξαν κατά την διάρκεια της εκπόνησης της πτυχιακής εργασίας αλλά και καθ' όλη την διάρκεια των σπουδών μου.

Τέλος θα ήθελα να ευχαριστήσω τους φίλους μου και την οικογένεια μου για την μεγάλη τους υποστήριξη και κατανόηση καθ' όλη την διάρκεια των σπουδών μου.

Περίληψη στα ελληνικά

Στην παρούσα πτυχιακή εργασία μελετήθηκαν τα στάδια, και οι δύο βασικές τεχνικές-κατηγορίες (content-based, collaborative filtering) των συστημάτων προτάσεων, καθώς και το πώς οι ατομικές, προσωποποιημένες, προτάσεις των ατόμων μιας ομάδας μπορούν να προσαρμοστούν σε μια ενιαία λίστα με προτάσεις για τα άτομα της ομάδας αυτής. Ο τελικός στόχος ήταν η δημιουργία μιας εφαρμογής για κινητά τηλέφωνα Android, η οποία θα μπορούσε να προσφέρει προσωποποιημένες προτάσεις για ένα σύνολο από διαθέσιμες ταινίες σε μια ομάδα χρηστών. Η εφαρμογή δόθηκε για δοκιμή σε μερικούς χρήστες από τους οποίους και συλλέχθηκαν σχόλια για τις προτάσεις που τους έγιναν. Το συμπέρασμα που εξήχθη ήταν πως η εφαρμογή πέτυχε τον αρχικό σκοπό της, δηλαδή ήταν σε θέση να παράξει προτάσεις τέτοιες, ώστε να ικανοποιούνται οι προτιμήσεις όλων των χρηστών, ταυτόχρονα.

Λέξεις Κλειδιά: (Συστήματα προτάσεων, Ομαδικές προτάσεις, Collaborative filtering, Content-based filtering, Neighborhood-based model, Latent-factor model)

Abstract

In this thesis, the stages, both basic techniques-categories (content-based, collaborative filtering) of recommendation systems were studied, as well as how the individual, personalized, suggestions of the people of a group can be adapted to a single list of suggestions for the people in this group. The ultimate goal was to create an Android mobile phone application that could provide personalized recommendations for a set of available movies to a group of users. The application was given for testing to a few users from whom feedback was collected on the suggestions made to them. The conclusion drawn was that the application achieved its original purpose, that it was able to produce recommendations that all user preferences were satisfied at the same time.

Keywords: (Recommendation system, Group recommendation, Collaborative filtering, Content-based filtering, Neighborhood-based model, Latent-factor model)

Πίνακας περιεχομένων

| | |
|--|-----------|
| ΕΠΙΤΡΟΠΗ ΕΞΕΤΑΣΗΣ | II |
| ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ | III |
| ΕΥΧΑΡΙΣΤΙΕΣ – ΑΦΙΕΡΩΣΕΙΣ | IV |
| ΠΕΡΙΛΗΨΗ ΣΤΑ ΕΛΛΗΝΙΚΑ | V |
| ABSTRACT | VI |
| ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ | VII |
| ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ ΚΑΙ ΕΙΚΟΝΩΝ | IX |
| ΣΥΝΤΜΗΣΕΙΣ - ΑΡΚΤΙΚΟΛΕΞΑ - ΑΚΡΩΝΥΜΙΑ | XI |
| ΕΙΣΑΓΩΓΗ | 1 |
| ΣΚΟΠΟΣ ΠΤΥΧΙΑΚΗΣ..... | 4 |
| 1. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ | 6 |
| 1.1 ΤΙ ΕΙΝΑΙ ΕΝΑ RECOMMENDATION SYSTEM..... | 6 |
| 1.2 ΣΤΑΔΙΑ ΠΑΡΑΓΩΓΗΣ ΠΡΟΤΑΣΕΩΝ..... | 7 |
| 1.2.1 Στάδιο συλλογής πληροφοριών..... | 7 |
| 1.2.2 Στάδιο εκμάθησης | 9 |
| 1.2.3 Στάδιο παραγωγής προτάσεων | 10 |
| 1.3 ΤΕΧΝΙΚΕΣ - ΚΑΤΗΓΟΡΙΕΣ RECOMMENDATION SYSTEM | 11 |
| 1.3.1 Content-based filtering | 11 |
| 1.3.2 Collaborative filtering | 12 |
| 1.4 GROUP RECOMMENDATIONS | 20 |
| 2. ΜΕΘΟΔΟΛΟΓΙΑ ΈΡΕΥΝΑΣ | 22 |
| 3. ΑΛΓΟΡΙΘΜΙΚΕΣ ΤΕΧΝΙΚΕΣ | 25 |
| 3.1 ΔΗΜΙΟΥΡΓΙΑ DATASET | 25 |
| 3.2 ΜΟΝΤΕΛΟ SVD | 26 |
| 3.3 ΜΟΝΤΕΛΟ KNN-BASELINE | 29 |
| 3.4 GROUP RECOMMENDATIONS ΜΕ SVD..... | 31 |
| 4. ΕΠΙΣΚΟΠΗΣΗ ΕΦΑΡΜΟΓΗΣ | 32 |
| 4.1 ΟΘΟΝΗ ΑΡΧΙΚΟΠΟΙΗΣΕΙΣ ΠΡΟΦΙΛ ΧΡΗΣΤΗ..... | 32 |

| | | |
|-----------|--|-----------|
| 4.2 | ΟΘΟΝΗ ΛΙΣΤΑΣ ΦΙΛΩΝ | 33 |
| 4.3 | ΔΙΑΛΟΓΟΣ ΕΠΙΛΟΓΗΣ ΣΥΝΕΔΡΙΑΣ | 34 |
| 4.4 | ΟΘΟΝΗ ΕΠΙΛΟΓΗΣ ΤΑΙΝΙΑΣ ΓΙΑ ΣΥΝΕΔΡΙΑ | 35 |
| 4.5 | ΟΘΟΝΗ ΕΠΙΛΟΓΗΣ ΕΙΔΟΥΣ ΤΑΙΝΙΑΣ ΓΙΑ ΣΥΝΕΔΡΙΑ | 36 |
| 4.6 | ΟΘΟΝΗ ΨΗΦΟΦΟΡΙΑΣ ΣΥΝΕΔΡΙΑΣ | 37 |
| 4.7 | ΟΘΟΝΗ ΜΕ ΤΕΛΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ | 38 |
| 5. | ΑΞΙΟΛΟΓΗΣΗ & ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ | 40 |
| 5.1 | ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ ΓΙΑ ΣΥΝΕΔΡΙΑΣ | 40 |
| 5.1.1 | <i>Συνεδρίες παρόμοιων ταινιών</i> | <i>40</i> |
| 5.1.2 | <i>Συνεδρίας με βάση παλιές βαθμολογίες.....</i> | <i>43</i> |
| 5.2 | ΣΥΜΠΕΡΑΣΜΑΤΑ ΑΞΙΟΛΟΓΗΣΗΣ | 46 |
| 6. | ΠΕΡΙΟΡΙΣΜΟΙ & ΠΙΘΑΝΕΣ ΕΠΕΚΤΑΣΗΣ..... | 47 |
| 6.1 | ΠΕΡΙΟΡΙΣΜΟΙ | 47 |
| 6.2 | ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΗΣ | 48 |
| | ΒΙΒΛΙΟΓΡΑΦΙΑ | 49 |

Πίνακας Σχημάτων και Εικόνων

| | |
|--|----|
| Εικόνα 1. Προτάσεις για ταινίες στην εφαρμογή Netflix | 1 |
| Εικόνα 2. Προτάσεις για βιβλία στο Amazon..... | 2 |
| Εικόνα 3. Απονομή βραβείου Netflix prize | 3 |
| Εικόνα 4. Η ροή ενός recommendation system | 6 |
| Εικόνα 5. Netflix taste preferences, καταργήθηκε το 2015 (Falk, 2019). | 8 |
| Εικόνα 6. Η διαφορά μεταξύ της έμμεσης και άμεσης ανατροφοδοτήσεις και πώς μπορούν εξωτερικοί παράγοντές να επηρεάσουν το πώς βαθμολογεί ένας χρήστης, και τι του αρέσει πραγματικά (Falk, 2019) | 8 |
| Εικόνα 7. Στάδια ενός Recommendation system (Isinkaye, Folajimi, & Ojokoh, 2015)..... | 10 |
| Εικόνα 8. Παράδειγμα επεξήγησης προτάσεων | 11 |
| Εικόνα 9. Παράδειγμα Content-based filtering..... | 12 |
| Εικόνα 10. Παράδειγμα rating matrix (Elahi, Ricci, & Rubens, 2016) | 13 |
| Εικόνα 11. Διαδικασία προτάσεων μέσω Collaborative Filtering (Isinkaye, Folajimi, & Ojokoh, 2015)..... | 13 |
| Εικόνα 12. Παράδειγμα Collaborative filtering..... | 14 |
| Εικόνα 13. Παράδειγμα Neighborhood-based με βάση τα Items..... | 15 |
| Εικόνα 14. Παράδειγμα Neighborhood-based με βάση τους χρήστες..... | 16 |
| Εικόνα 15. Παράδειγμα προωθήσεις νέων ταινιών & σειρών στο Netflix | 18 |
| Εικόνα 16. Παράδειγμα αραιού rating matrix (Falk, 2019) | 19 |
| Εικόνα 17. Παράδειγμα Gray sheep χρήστη | 19 |
| Εικόνα 18. TMDb Logo | 23 |
| Εικόνα 19. Μεθοδολογία έρευνας..... | 24 |
| Εικόνα 20. Κώδικας συνάρτησης export_data_file..... | 25 |
| Εικόνα 21. Κώδικας συνάρτησης create_dataset | 26 |
| Εικόνα 22. Κώδικας συνάρτησης find_best_combination | 27 |
| Εικόνα 23. Αποτελέσματα GridSearchCV για SVD..... | 27 |
| Εικόνα 24. Κώδικας συνάρτησης train_svd..... | 28 |
| Εικόνα 25. Κώδικας για την πρόβλεψη μιας ταινίας | 28 |
| Εικόνα 26. Προβλεπόμενη βαθμολογία..... | 29 |
| Εικόνα 27. Κώδικας συνάρτησης train_knn | 29 |
| Εικόνα 28. Κώδικας ενδεικτικού παραδείγματος | 30 |
| Εικόνα 29. Αποτελέσματα παραδείγματος KNN..... | 30 |

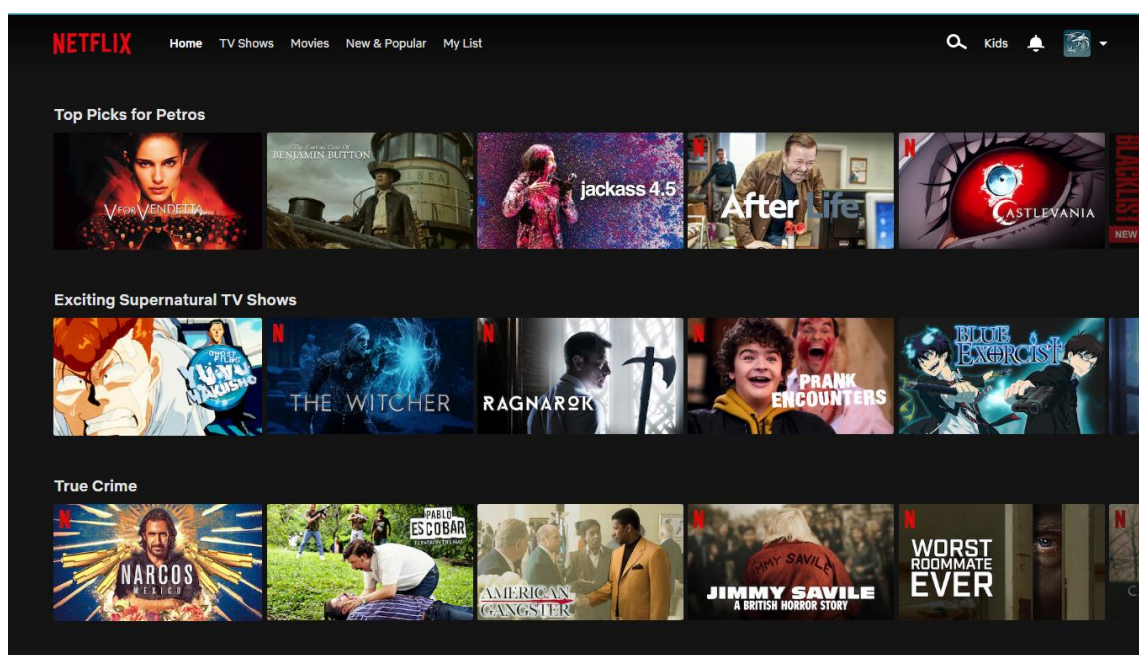
| | |
|--|----|
| Εικόνα 30. Κώδικας συνάρτησης <code>get_group_recommendations</code> | 31 |
| Εικόνα 31. Οθόνη αρχικοποιήσεις προφίλ χρήστη | 32 |
| Εικόνα 32. Οθόνη λίστας φίλων | 33 |
| Εικόνα 33. Διάλογος επιλογής συνεδρίας..... | 34 |
| Εικόνα 34. Οθόνη επιλογής ταινίας | 35 |
| Εικόνα 35. Οθόνη επιλογής είδους ταινίας | 36 |
| Εικόνα 36. Οθόνη ψηφοφορίας ταινιών..... | 37 |
| Εικόνα 37. Οθόνη με τελικά αποτελέσματα | 39 |

Συντμήσεις - Αρκτικόλεξα - Ακρωνύμια

| | |
|-----|-----------------------------------|
| RS | Recommendation System |
| IF | Implicit Feedback |
| EF | Explicit Feedback |
| API | Application Programming Interface |
| CF | Collaborative Filtering |
| SVD | Singular Value Decomposition |
| KNN | K-Nearest Neighbors |

Εισαγωγή

Στις μέρες μας, σε δραστηριότητες όπως οι αγορές σε ηλεκτρονικά καταστήματα σαν το Amazon¹ (Εικόνα 2), ή η παρακολούθηση ταινιών και σειρών σε συνδρομητικές υπηρεσίες σαν το Netflix² (Εικόνα 1), έχουν εφαρμοστεί κάποιας μορφής συστήματος προτάσεων για τα προϊόντα που διαθέτουν. Ο λόγος είναι ότι το πλήθος των διαθέσιμων προϊόντων είναι τόσο μεγάλο που συχνά δυσκολεύουν τους χρήστες στην προσπάθειά τους να επιλέξουν. Αυτό το πρόβλημα προσπαθούν να λύσουν τα συστήματα προτάσεων. Η προσέγγιση που ακολουθούν είναι φιλτράροντας τον μεγάλο όγκο προϊόντων με βάση τις προτιμήσεις ή τα ενδιαφέροντα των χρηστών.



Εικόνα 1. Προτάσεις για ταινίες στην εφαρμογή Netflix

¹ <https://www.amazon.com/>

² <https://www.netflix.com/>

Recommended for You

**Inside Apple: How America's Most Admired--and Secretive-- Company Really Works****Our Price: \$9.99**
Used & new from \$9.99[See all buying options](#)

Rate this item

 I own it Not interested

Because you purchased...

**The Toyota Way : 14 Management Principles from the World's Greatest Manufacturer**
(Kindle Edition) This was a gift Don't use for recommendations

Εικόνα 2. Προτάσεις για βιβλία στο Amazon

Ιδιαίτερο ενδιαφέρον προκαλεί το γεγονός πως το Netflix διοργάνωσε το 2006 έναν διαγωνισμό με το όνομα Netflix Prize με σκοπό τη βελτίωση του τρέχοντος αλγορίθμου του παραγωγής προτάσεων (Cinematch) κατά ένα ποσοστό τουλάχιστον 10%. Για τον σκοπό αυτό, το Netflix παρέιχε στα μέλη του διαγωνισμού ένα data set προκειμένου να δοκιμάσουν τα συστήματα που θα ανέπτυσαν με δεδομένα του πραγματικού κόσμου. Ο διαγωνισμός διήρκεσε 3 χρόνια μέχρι που το 2009, μια ομάδα με το όνομα "BellKor's Pragmatic Chaos", κατάφερε να ξεπεράσει τον Cinematch κατά 10,05% κερδίζοντας έτσι το έπαθλο του 1 εκατομμύριου δολαρίων. Το γεγονός ότι το Netflix έδωσε ένα τόσο μεγάλο ποσό για την βελτίωση του recommendation system του αποτελεί αποδείξει της σημαντικότητας των recommendation system για τέτοιου είδους εφαρμογές.



Εικόνα 3. Απονομή βραβείου Netflix prize

Παρά τον καίριο ρόλο που διαθέτουν τα συστήματα αυτά, το μεγαλύτερο ενδιαφέρον έχει δοθεί για την παραγωγή ατομικών προτάσεων, χωρίς να έχει εξερευνηθεί η πτυχή των προτάσεων σε μία ομάδα ατόμων. Με βάση τα ανωτέρω, και λαμβάνοντας υπόψιν την έλλειψη αυτή, το σύστημα που θα αναπτυχθεί θα προσπαθήσει να προσεγγίσει τη λύση αυτής της έλλειψης, παράγοντας προτάσεις για μία ομάδα δύο ατόμων, όπου οι προτάσεις αυτές να συμφωνούν με τις προτιμήσεις και των δύο μελών. Η έλλειψη αυτή οδηγεί τα άτομα αυτά να προχωρήσουν σε εύρεση και διαλογή υλικού με βάσει τις προτιμήσεις και των δύο, διαδικασία η οποία μπορεί να είναι χρονοβόρα και ίσως κουραστική.

Σκοπός πτυχιακής

Σκοπός αυτής της πτυχιακής εργασίας είναι, αρχικά, η εμβάθυνση στο αντικείμενο των συστημάτων προτάσεων (recommendation systems), η κατανόηση ενός συστήματος προτάσεων, σε ποιες κατηγορίες χωρίζεται, ποια είναι τα θετικά και τα αρνητικά αυτών των κατηγοριών, καθώς και τι μετατροπές χρειάζεται για να μπορέσει να κάνει προτάσεις σε ομάδες από χρήστες.

Έπειτα έγινε έρευνα προκειμένου να κατανοηθεί το είδος των πληροφοριών που χρειάζεται ένα σύστημα προτάσεων, αφενός για να κατανοήσει τις προτιμήσεις των χρηστών και αφετέρου προκειμένου να διαπιστώσει αν μία ταινία είναι κατάλληλη ως πρόταση με βάση τις προτιμήσεις αυτές, ώστε να είναι σε θέση να παράξει εξατομικευμένες προτάσεις για ταινίες στους χρήστες μιας εφαρμογής.

Τέλος, δημιουργήθηκε μια εφαρμογή για κινητά τηλεφωνά με λειτουργικό σύστημα Android, η οποία καλείται, δοθείσας μιας λίστα με ταινίες, να παράξει προσωποποιημένες προτάσεις στους χρήστες της. Ο τρόπος με τον οποίο επιτυγχάνεται αυτό είναι μέσω δημιουργίας μίας συνεδρίας (session) μεταξύ δύο χρηστών όπου ο ένας από τους δύο μπορεί να επιλέξει τον τύπο συνεδρίας, ανάμεσα σε δύο διαθέσιμους.

Ο πρώτος τύπος συνεδρίας είναι η πρόταση παρόμοιων ταινιών με βάση μια ταινία της επιλογής τους χρησιμοποιώντας τον αλγόριθμο KNN (K-Nearest Neighbor), ένας αλγόριθμος του οποίου δουλειά είναι να ομαδοποιεί παρόμοια μεταξύ τους αντικείμενα, και στη συνέχεια δοθέντος ενός νέου αντικειμένου, να βρίσκει με ποια ομάδα έχει τις μεγαλύτερες ομοιότητες. Με τον τρόπο αυτό θα ανακαλυφθούν ομοιότητες μεταξύ των ταινιών, μέσα από παλαιότερες βαθμολογίες των χρηστών της εφαρμογής.

Ο δεύτερος τύπος συνεδρίας είναι η παραγωγή προτάσεων στις οποίες, σε αντίθεση με τον πρώτο τύπο, θα λαμβάνονται υπόψιν οι παλαιότερες βαθμολογίες που έχουν κάνει οι ίδιοι οι χρήστες που συμμετέχουν στη συνεδρία, αλλά και άλλοι χρήστες μέσα στην εφαρμογή. Αυτό υλοποιήθηκε χρησιμοποιώντας ένα latent-factor μοντέλο πρόβλεψης χρησιμοποιώντας τον αλγόριθμο SVD (Singular Value Decomposition), ο οποίος, δοθέντος ενός συνόλου χαρακτηριστικών, αναλαμβάνει να πραγματοποιεί προβλέψεις. Στην περίπτωση αυτού του συστήματος, αναλαμβάνει να προβλέψει τι βαθμολογία θα έδιναν οι δύο χρήστες σε ταινίες τις οποίες δεν έχουν παρακολουθήσει, λαμβάνοντας υπόψιν παλαιότερες βαθμολογίες τους, ώστε να παραχθεί μια λίστα από προτάσεις για τους χρήστες που βρίσκονται στην συνεδρία.

Παράλληλα με την εφαρμογή δημιουργήθηκε ένα πρόγραμμα σε python, χρησιμοποιώντας την βιβλιοθήκη SciPy surprise, που ήταν υπεύθυνο για την εκπαίδευση των δύο μοντέλων με βάση τις βαθμολογίες των χρηστών.

1. Θεωρητικό υπόβαθρο

Στο συγκεκριμένο κεφάλαιο παρουσιάζονται οι βασικές έννοιες ενός recommendation system, ποια είναι τα στάδια που ακολουθεί ένα recommendation system ώστε να είναι σε θέση να παράγει προτάσεις, ποιες είναι οι δύο βασικές κατηγορίες – τεχνικές ενός recommendation system, καθώς και πως μπορούμε να συναθροίσουμε τις ατομικές προτάσεις των χρηστών μιας ομάδα, σε μια λίστα με ομαδικές προτάσεις.

1.1 Τι είναι ένα recommendation system

Ένα recommendation system είναι μια υποκατηγορία των συστημάτων φιλτραρίσματος πληροφοριών (Information Filtering System). Η κύρια διαφορά είναι ότι τα συστήματα φιλτραρίσματος πληροφοριών έχουν σαν στόχο να αφαιρούν ανεπιθύμητες ή μη σχετικές πληροφορίες από κάποια ροή πληροφοριών η οποία πρόκειται να εμφανιστεί στον χρήστη. Τα recommendation systems αντ' αυτού, έχουν ως σκοπό να προωθούν items³ όπως ταινίες, βιβλία, μουσική, φαγητό που μπορεί να ενδιαφέρουν τον χρήστη (Ricci, Rokach, & Shapira, 2010). Ένας τρόπος με τον οποίο το επιτυγχάνουν είναι συλλέγοντας πληροφορίες από τους χρήστες για να κάνουν predict (προβλέψουν) τα items που θεωρούν ότι θα τους ενδιαφέρουν.



Εικόνα 4. Η ροή ενός recommendation system

³ Item: Με τον όρο item εννοούμε οποιοδήποτε προϊόν ή αντικείμενο μπορεί να προτείνει ένα recommendation system.

1.2 Στάδια παραγωγής προτάσεων

1.2.1 Στάδιο συλλογής πληροφοριών

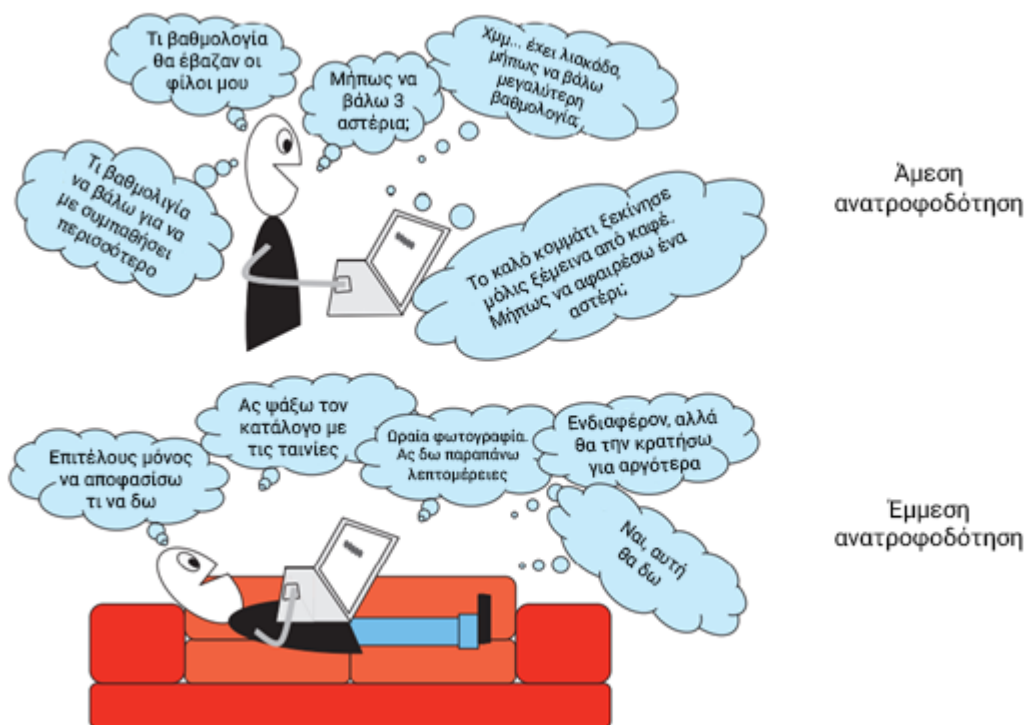
Σε αυτό το στάδιο συλλέγονται πληροφορίες σχετικές με τους χρήστες ώστε να παραχθεί ένα προφίλ χρήστη, το οποίο έπειτα θα χρησιμοποιηθεί από το RS (Recommendation System) για να παράγει προτάσεις. Οι πληροφορίες που μπορεί να συλλέξει μπορεί να είναι είτε ιδιαιτερότητες ενός χρήστη, όπως για παράδειγμα τι είδους ταινίες του αρέσουν ή πως βαθμολόγησε μια ταινία που παρακολούθησε. Μπορούν ωστόσο να εξαχθούν χρήσιμες πληροφορίες και από τον τρόπο που χρησιμοποιεί την εφαρμογή. Για παράδειγμα, η γνωστή συνδρομητική υπηρεσία Netflix χρησιμοποιεί πληροφορίες όπως τη συσκευή στην οποία ο χρήστης παρακολουθεί μια ταινία ή την ώρα της ημέρας που την παρακολουθεί ή αναζητά ταινία προκειμένου να παρακολουθήσει, για να του προτείνει ακόμα πιο ακριβείς προτάσεις για εκείνη την χρονική στιγμή (Falk, 2019). Το σημαντικό ωστόσο είναι ότι χωρίς καθόλου πληροφορίες για τον χρήστη ένα RS δεν μπορεί να κάνει ακριβείς εξατομικευμένες προτάσεις. Όσο πιο πολλές πληροφορίες έχει και μπορεί να αξιοποιήσει ένα RS τόσο πιο στοχευμένες θα είναι και οι προτάσεις που θα κάνει (Isinkaye, Folaajimi, & Ojokoh, 2015).

Τα παραδείγματα που μόλις αναφερθήκαν ανάγονται αντίστοιχα σε δύο κατηγορίες. Η πρώτη ονομάζεται άμεση ανατροφοδότηση (explicit feedback) η οποία όπως προαναφέρθηκε περιλαμβάνει τις προτιμήσεις ενός χρήστη για ένα item και η δεύτερη ονομάζεται έμμεση ανατροφοδότηση (implicit feedback) η οποία παρατηρώντας την συμπεριφορά του χρήστη όσο χρησιμοποιεί την εφαρμογή μπορεί να εξάγει πληροφορίες που θα είναι χρήσιμες για το RS. Υπάρχει και η περίπτωση χρήσης ένα τρόπο υβριδικής ανατροφοδότησης. Για παράδειγμα, το Netflix μέχρι και το 2015 σε συνδυασμό με το IF (Implicit Feedback) είχε μία κατηγορία στην εφαρμογή που ονόμαζαν «Taste profile». Εκεί ο χρήστης μπορούσε να βαθμολογήσει ταινίες, να αλλάξει τι κατηγορίες ταινιών του αρέσουν ή ακόμα και κατά ποσό του αρέσει να βλέπει παρόμοιες ταινίες με βάση κάποια άλλη ταινία που είχε παρακολουθήσει (Falk, 2019).



Εικόνα 5. Netflix taste preferences, καταργήθηκε το 2015 (Falk, 2019).

Ο λόγος για τον οποίο το Netflix, ωστόσο, αφαίρεσε το Taste profile, είναι ότι τις περισσότερες φορές, η άμεση ανατροφοδότηση που έπαιρνε από τους χρήστες δεν ήταν ακριβής, διότι επηρεαζόταν από εξωτερικούς παράγοντες. Για παράδειγμα μπορεί ένας χρήστης να βαθμολογούσε μια ταινία που δεν του αρέσει υψηλά, άπλα και μόνο προκειμένου να είναι αποδεκτός στον κοινωνικό του κύκλο.



Εικόνα 6. Η διαφορά μεταξύ της έμμεσης και άμεσης ανατροφοδοτήσεων και πώς μπορούν εξωτερικοί παράγοντες να επηρεάσουν το πώς βαθμολογεί ένας χρήστης, και τι του αρέσει πραγματικά (Falk, 2019) .

Το συμπέρασμα από το παραπάνω παράδειγμα είναι ότι οι πληροφορίες που συλλέγονται για τις προτιμήσεις των χρηστών, εκτός από πολλές, πρέπει να είναι και ακριβείς ώστε το RS να αντικατοπτρίζει τις πραγματικές προτιμήσεις τους και να είναι σε θέση να παράγει ενδιαφέροντες προτάσεις στους χρήστες της εκάστοτε εφαρμογής.

1.2.1.1 Άμεση ανατροφοδότηση

Μέσω της αλληλεπίδρασης του χρήστη με την εκάστοτε πλατφόρμα μπορεί να πραγματοποιηθεί συλλογή πληροφοριών, όπως βαθμολογίες από προϊόντα (items) που έχει αγοράσει. Το ποσό ακριβές είναι το RS αντικατοπτρίζεται στο πλήθος των βαθμολογιών που έχει κάνει ο χρήστης.

- **Θετικά:** Είναι αρκετά εύκολο να υλοποιηθεί σαν τρόπος συλλογής ανατροφοδότησης, για παράδειγμα μέσα από ένα ερωτηματολόγιο για τι είδους ταινίες προτιμάει ή με την δημιουργία μιας μπάρας βαθμολογίας όπου μπορεί να αξιολογήσει τα προϊόντα που καταναλώνει.
- **Αρνητικά:** Με την άμεση ανατροφοδότηση απαιτείται επιπλέον προσπάθεια από τον χρήστη, βαθμολογώντας κάθε προϊόν που καταναλώνει από την εφαρμογή, πράγμα το οποίο πολλές φορές δεν συμβαίνει και έχει ως αποτέλεσμα να μην υπάρχουν αρκετές πληροφορίες προκειμένου να παραχθούν ακριβής προτάσεις.

1.2.1.2 Έμμεση ανατροφοδότηση

Μέσω της αλληλεπίδρασης του χρήστη με την εφαρμογή δίνεται η δυνατότητα συλλογής χρήσιμων πληροφοριών από τον τρόπο που περιηγείται μέσα στην εφαρμογή, πώς αλληλοεπιδρά με ένα προϊόν, για παράδειγμα αν παρακολουθεί μια ταινία που του αρέσει μέχρι το τέλος ή αν την σταματάει σε περίπτωση που δεν του αρέσει.

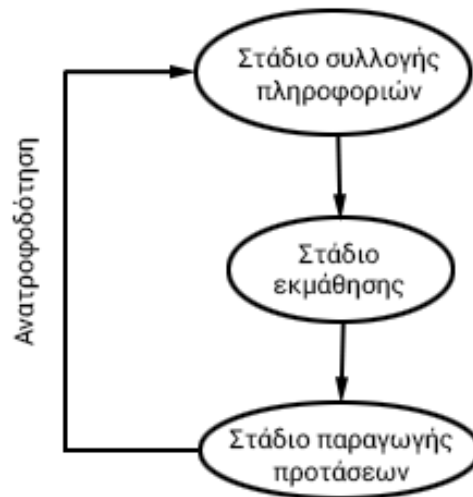
- **Θετικά:** Με την έμμεση ανατροφοδότηση δεν χρειάζεται κάποια προσπάθεια από τον χρήστη για τη συλλογή πληροφοριών· ο χρήστης απλώς πρέπει να χρησιμοποιεί την εφαρμογή για να συλλεχθούν πληροφορίες για αυτόν.
- **Αρνητικά:** Δεν είναι εύκολο να εξαχθούν σίγουρα συμπεράσματα από την παρακολούθησή της συμπεριφοράς του χρήστη.

1.2.2 Στάδιο εκμάθησης

Με είσοδο τα δεδομένα από το στάδιο συλλογής ανατροφοδότησης, γίνεται η εφαρμογή κάποιου αλγόριθμου εκμάθησης προκειμένου να εξαχθούν οι προτιμήσεις των χρηστών.

1.2.3 Στάδιο παραγωγής προτάσεων

Σε αυτό το στάδιο παράγει προτάσεις για προϊόντα που μπορεί να ενδιαφέρουν τους χρήστες.



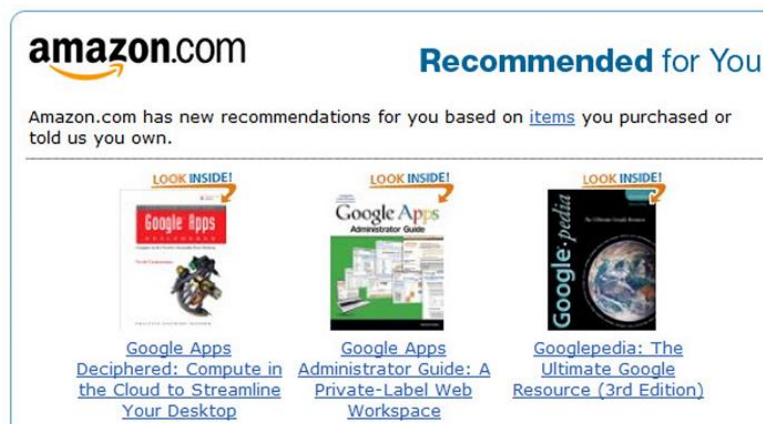
Εικόνα 7. Στάδια ενός Recommendation system (Isinkaye, Folajimi, & Ojokoh, 2015).

1.3 Τεχνικές - Κατηγορίες Recommendation system

1.3.1 Content-based filtering

Τα content-based recommendation systems επιλέγουν τα items που θα προτείνουν με βάση την συσχέτιση των ιδιοτήτων των items που υπάρχουν σε μια εφαρμογή (Mathew, Kuriakose, & Hegde, 2016). Για παράδειγμα σε ένα RS για ταινίες, μια πληροφορία που θα μπορούσε να αξιοποιηθεί θα μπορούσε να είναι σε τι είδη ανήκει η ταινία (δράμα, κωμωδία, περιπέτεια). Έπειτα, φιλτράροντας παλιές ταινίες που έχει βαθμολογήσει υψηλά ο χρήστης, θα μπορούσε να γίνει φιλτράρισμα με βάση τα είδη, και να προταθούν παρόμοιες ταινίες (Troussas, Krouska, & Vίνου, 2018).

Θετικά: Είναι αρκετά εύκολο να προσαρμοστεί στις προτιμήσεις των χρηστών, στην περίπτωση που αλλάζουν. Επίσης, στην περίπτωση που εισαχθούν νέα προϊόντα μέσα στην εφαρμογή μπορούν να αρχίσουν να προτείνονται γρηγορά στους χρήστες αφού μπορούν να συσχετιστούν άμεσα με παλιά προϊόντα που μπορεί να έχουν κατανάλωση (Zhongqi, 2015). Επιπλέον είναι εύκολο να εξηγηθεί στους χρήστες για ποιο λόγο βλέπουν τις συγκεκριμένες προτιμήσεις που παράγει το recommendation system (Isinkaye, Folajimi, & Ojokoh, 2015).



Εικόνα 8. Παράδειγμα επεξήγησης προτάσεων

Αρνητικά: Για να μπορέσουν να λειτουργήσουν σωστά οι content-based τεχνικές, θα πρέπει να υπάρχουν πολλές πληροφορίες για τα προϊόντα ώστε να μπορέσουν να βρεθούν ακριβής συσχετίσεις μεταξύ αυτών. Για αυτό οι content-based τεχνικές λειτουργούν καλύτερα με προϊόντα όπως βιβλία και ταινίες που μπορούν να συγκεντρωθούν αρκετά μετα-δεδομένα για αυτά. Επιπλέον με τις τεχνικές αυτές δεν είναι εφικτό να προταθούν νέα είδη προϊόντων στον χρήστη, όπως ένα νέο είδος ταινίας, μιας και λαμβάνει υπόψιν μονάχα τα προϊόντα που ο χρήστης έχει ήδη αγοράσει. Αυτό σημαίνει ότι αν ένας χρήστης έχει αγοράσει από ένα

ηλεκτρονικό κατάστημα μόνο ένα είδος από βιβλία, ο recommender δεν θα του προτείνει κάποιο άλλο είδος. Σαν αποτέλεσμα οι προτάσεις του recommendation system μπορεί να γίνουν μονότονες (Balabanović & Shoham, 1997).

CONTENT-BASED FILTERING



Εικόνα 9. Παράδειγμα Content-based filtering

1.3.2 Collaborative filtering

Τα CF (Collaborative Filtering) recommendation systems επιλέγουν τα items που θα προτείνουν σε έναν χρήστη παρατηρώντας την ανατροφοδότηση και βρίσκοντας παρόμοιους χρήστες. Το αν δύο χρήστες είναι παρόμοιοι μπορεί να υπολογιστεί κοιτάζοντας απλά παλιές βαθμολογίες σε items που έχουν βαθμολογήσει. Οι χρήστες είναι όμοιοι εάν βαθμολογούν τις ίδιες ταινίες με τον ίδιο τρόπο. Έπειτα το CF μπορεί να προτείνει ταινίες στον χρήστη που δεν έχει δει, αλλά έχουν δει και έχουν βαθμολογήσει υψηλά όμοιοι χρήστες στην εφαρμογή (Koren & Bell, 2015) (Desrosiers & Karypis, 2010) (Tsolakidis, Triperina, Sgouroulou, & Christidis, 2016).

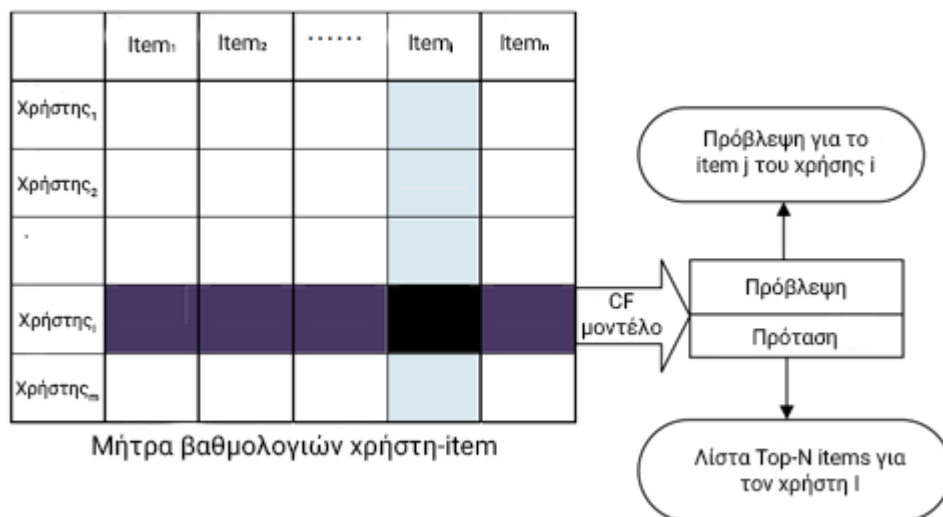
Οι βαθμολογίες των χρηστών αποθηκεύονται σε ένα πίνακα δύο διαστάσεων $m \times n$ όπου m είναι ο αριθμός των χρηστών, και n είναι ο αριθμός των items. Κάθε κελί μέσα στον πίνακα έχει την βαθμολογία του χρήστη που βρίσκεται στην i -οστή σειρά, και για το item που

βρίσκεται στην j-οστή στήλη (Εικόνα 10). Για τα items που ο χρήστης δεν έχει βαθμολογήσει ακόμα, τα κελιά είναι κενά. Αυτός ο πίνακας είναι και η είσοδος σε ένα CF recommendation system.

| | Πόλεμος των Άστρων | Τιτανικός | Toy Story | Star Trek |
|-------|--------------------|-----------|-----------|-----------|
| Alice | 3 | 2 | 3 | ∅ |
| Bob | 5 | 1 | 5 | 2 |
| Cindy | ∅ | 4 | 1 | 3 |
| David | 1 | ∅ | 2 | 2 |

Εικόνα 10. Παράδειγμα rating matrix (Elahi, Ricci, & Rubens, 2016)

Ένα CF υπολογίζει τις προτάσεις που θα εμφανίσει προβλέποντας τι βαθμολογία θα έβαζε ο χρήστης στα κελιά που δεν έχει βαθμολογήσει ακόμα (κενά κελιά), αξιοποιώντας πάντα τις σχέσεις μεταξύ των χρηστών. Έπειτα σύμφωνα με τις προβλέψεις που έχει κάνει το RS, οι προτάσεις ταξινομούνται, και αυτές με την μεγαλύτερη βαθμολογία προτείνονται τελικά στον χρήστη. Η ακρίβεια του CF ενώ βασίζεται κυρίως στην επιλογή του αλγόριθμου πρόβλεψης που θα υιοθετηθεί, μπορεί να επηρεαστεί σημαντικά και από τα δεδομένα των βαθμολογιών που παίρνει σαν είσοδο (Elahi, Ricci, & Rubens, 2016).



Εικόνα 11. Διαδικασία προτάσεων μέσω Collaborative Filtering (Isinkaye, Folajimi, & Ojokoh, 2015)

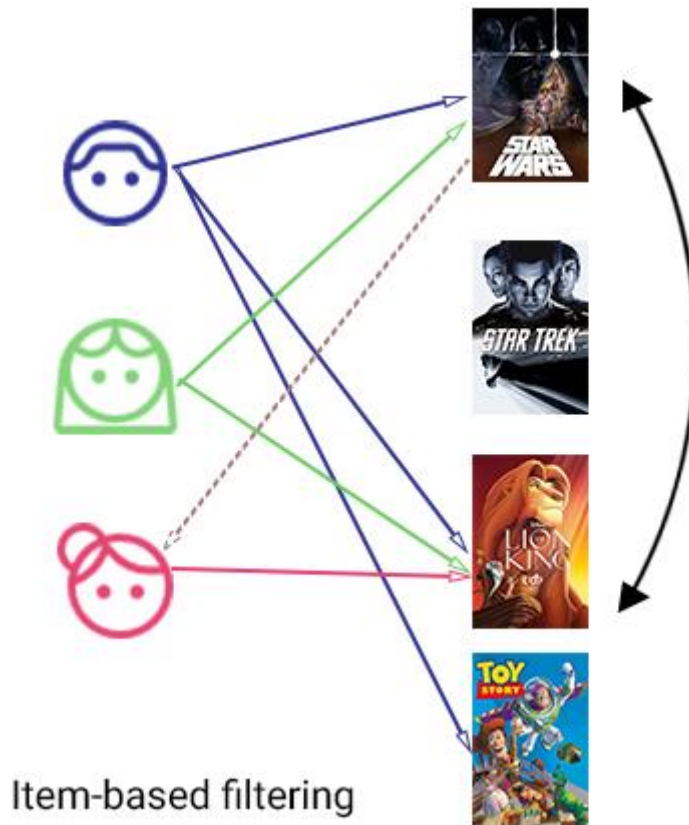
COLLABORATIVE FILTERING



Εικόνα 12. Παράδειγμα Collaborative filtering

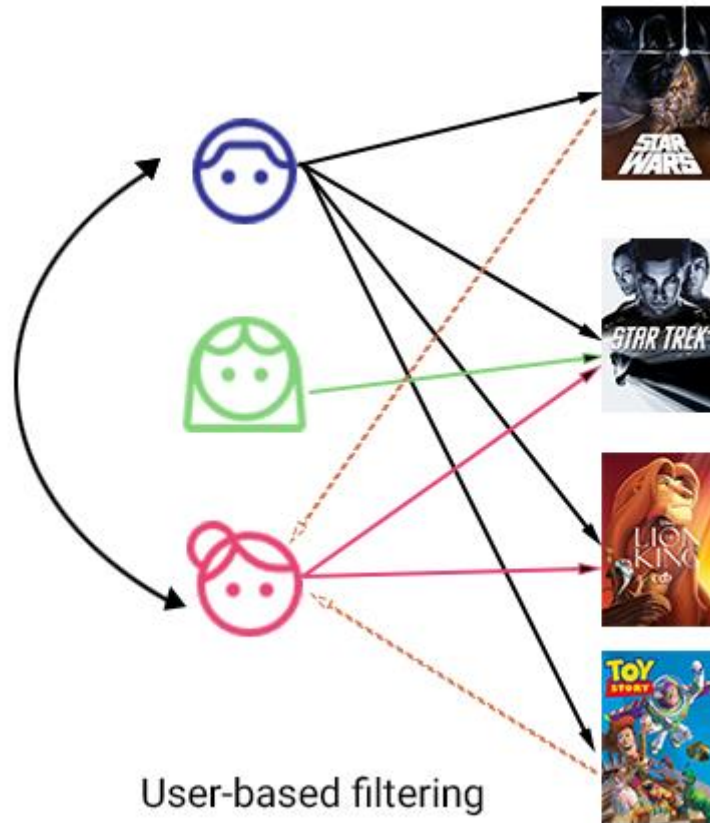
1.3.2.1 Neighborhood-based μοντέλα

Τα Neighborhood-based μοντέλα μπορούν να είναι είτε με βάση το item ή τον χρήστη (Elahi, Ricci, & Rubens, 2016). Τα μοντέλα που είναι με βάση το item, βρίσκουν την βαθμολογία για ένα item i ενός χρήστη u με βάση τις βαθμολογίες που έχει κάνει ο χρήστης σε παρόμοια item με το i (Desrosiers & Karypis, 2010).



Εικόνα 13. Παράδειγμα Neighborhood-based με βάση τα Items

Τα μοντέλα που είναι με βάση τον χρήστη υπολογίζουν τις βαθμολογίες για τα items χρησιμοποιώντας δύο είδη δεδομένων: τις βαθμολογίες του χρήστη, καθώς και τις βαθμολογίες άλλων όμοιων χρηστών. Όμοιοι χρήστες είναι αυτοί οι οποίοι βαθμολογούν τις ίδιες ταινίες με τον ίδιο τρόπο. Άρα η βαθμολογία για ένα item για έναν συγκεκριμένο χρήστη είναι βασισμένη στο πώς έχουν βαθμολογήσει το συγκεκριμένο item παρόμοιοι χρήστες με αυτόν (Elahi, Ricci, & Rubens, 2016).



Εικόνα 14. Παράδειγμα Neighborhood-based με βάση τους χρήστες

1.3.2.2 Latent-factor μοντέλα

Τα latent factor μοντέλα αποτελούν μια διαφορετική προσέγγιση στην κατηγορία του collaborative filtering για το πώς υπολογίζονται οι βαθμολογίες - προβλέψεις από το RS. Ένας διάσημος τρόπος ονομάζεται matrix factorization (Koren, Bell, & Volinsky, Matrix Factorization Techniques for Recommender Systems, 2009). Το matrix factorization βρίσκει μέσα από το learning phase ένα διάνυσμα από latent features που αποκαλούνται factors, για κάθε χρήστη και item μέσα στη μήτρα με τις βαθμολογίες των χρηστών για τα items. Κάθε factor μέσα στο διάνυσμα αντιπροσωπεύει πόσο καλά ένας χρήστης ή ένα item κατέχει ένα latent feature. Για παράδειγμα, σε ένα RS όπου τα items είναι ταινίες, τα latent factors θα μπορούσαν να αντιπροσωπεύουν είδη ταινιών: πόσο δράμα, κωμωδία ή τρόμου είναι μια ταινία. Μπορεί ωστόσο να είναι ανερμήνευτα. Όσο για τους χρήστες, το διάνυσμα με τα factor αντιπροσωπεύει τις προτιμήσεις των χρηστών για τα κάθε factor. Ο στόχος των factorization αλγορίθμων είναι να χωρίσει τον αρχικό πίνακα βαθμολογιών R σε δύο υποπίνακες S και M με τέτοιο τρόπο ώστε το γινόμενο τους να προσεγγίζει τον αρχικό πίνακα και επιπλέον να μπορεί να προβλέψει τις τιμές των κενών κελίων:

$$R \approx SM^T$$

S είναι η μήτρα με διαστάσεις $|U| \times F$ και M η μήτρα με διαστάσεις $|I| \times F$ όπου $|U|$ ο αριθμός των χρηστών, και $|I|$ ο αριθμός των items. F είναι ο αριθμός των factors, ο οποίος είναι και η παράμετρος που πρέπει να βελτιστοποιηθεί κατά το στάδιο της εκμάθησης. Η παραπάνω μέθοδος ονομάζεται Regularized SVD (Singular Value Decomposition) και προτάθηκε από τον Simon Funk κατά την διάρκεια του Netflix prize στο οποίο και κέρδισε την 3 θέση (Funk, 2006). Σαν αλγόριθμο εκμάθησης προτάθηκε ο stochastic gradient descent ο οποίος θα βελτιστοποιούσε τα factors με το να ελαχιστοποιήσει το εξής σφάλμα: $e_{ui} = r_{ui} - \widehat{r}_{ui}$ όπου r_{ui} το πραγματικό rating για τον χρήστη u και item i και \widehat{r}_{ui} η προβλέψη του μοντέλου για το rating του χρήστη u και item i . Σταματώντας όμως νωρίς για να μην γίνει over fitting στα δεδομένα εκπαίδευσης.

Οι συναρτήσεις είναι:

$$s_{uf} = s_{uf} + \gamma(e_{ui}m_{if} - \lambda s_{uf})$$

$$m_{if} = m_{if} + \gamma(e_{ui}s_{uf} - \lambda m_{if})$$

Όπου το γ και το λ είναι οι παράμετροι για τον ρυθμό εκμαθήσεις και το regulation αντίστοιχα.

Θετικά & Αρνητικά Collaborative Filtering

Ένα από τα θετικά του collaborative filtering είναι πως δεν χρειάζεται να διατίθεται η περιγραφή των items για τα οποία κάνει προτάσεις το recommendation system μιας και οι collaborative filtering τεχνικές το μόνο που χρειάζονται είναι οι βαθμολογίες από τους χρήστες της εφαρμογής. Επιπλέον, σε αντίθεση με τα content base όπου κάνουν προτάσεις με βάση τις προτιμήσεις των χρηστών, τα collaborative filtering μπορούν να προτείνουν items τα οποία δεν είναι στις προτιμήσεις των χρηστών, αλλά παρόλα αυτά να καταλήξουν να τους αρέσουν εξίσου (Krouska, Troussas, & Sgouroulou, 2020).

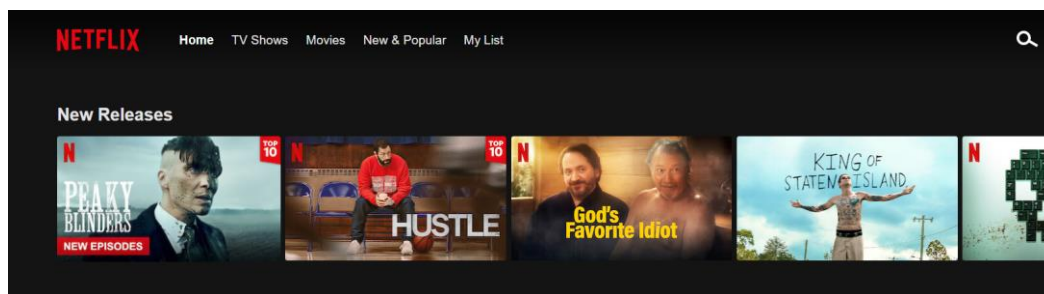
Ωστόσο, παρόλο την μεγάλη επιτυχία που έχουν οι Collaborative filtering τεχνικές, έχουν αναδείξει και πιθανά προβλήματα όπως:

Cold start πρόβλημα

Όταν ένας νέος χρήστης ή ένα νέο item προστίθεται στο rating matrix, όπως είναι φυσικό, η σειρά ή στήλη αντίστοιχα είναι κενή. Αυτό έχει ως αποτέλεσμα το RS να μην διαθέτει

πληροφορίες για τον νέο χρήστη ή item και να μην μπορεί να παράξει καλές προβλέψεις για τον χρήστη ή να προτείνει το νέο item σε άλλους χρήστες.

Ένας τρόπος να ξεπεραστεί αυτό το πρόβλημα για έναν νέο χρήστη είναι να του ζητηθεί κατά την εισαγωγή του στην εφαρμογή την πρώτη φορά να βαθμολογήσει μερικά items ώστε να μπορεί να ενταχθεί στον πίνακα και να βρεθούν παρόμοιοι χρήστες με αυτόν. Όσο για τα καινούρια items, ένας τρόπος είναι να τα προωθεί σε κάποιο ξεχωριστό σημείο της εφαρμογής (Παράδειγμα: Εικόνα 15) μέχρι να αποκτήσουν αρκετές βαθμολογίες, ώστε να αρχίσουν να προτείνονται και μέσω του RS (Liu, et al., 2014). Το cold start problem αποτελεί ένα από τα μεγαλύτερα προβλήματα των recommendation systems επηρεάζοντας την ακρίβεια τους σε μεγάλο βαθμό (Isinkaye, Folaajimi, & Ojokoh, 2015).



Εικόνα 15. Παράδειγμα προωθήσεις νέων ταινιών & σειρών στο Netflix

Data sparsity πρόβλημα

Σε μεγάλες εμπορικές εφαρμογές όπου ο αριθμός των χρηστών και των items είναι πολύ μεγάλος, είναι φυσικό ο rating matrix που χρησιμοποιείται στα CF ως είσοδο, να είναι αραιός δηλαδή ένα πολύ μεγάλο μέρος του πίνακα να έχει κενά κελιά. Ως αποτέλεσμα το RS δυσκολεύεται να προβλέψει τις βαθμολογίες από τα κενά κελιά με ακρίβεια. Μια υποκατηγορία του είναι και το cold start που αναφέρθηκε παραπάνω.

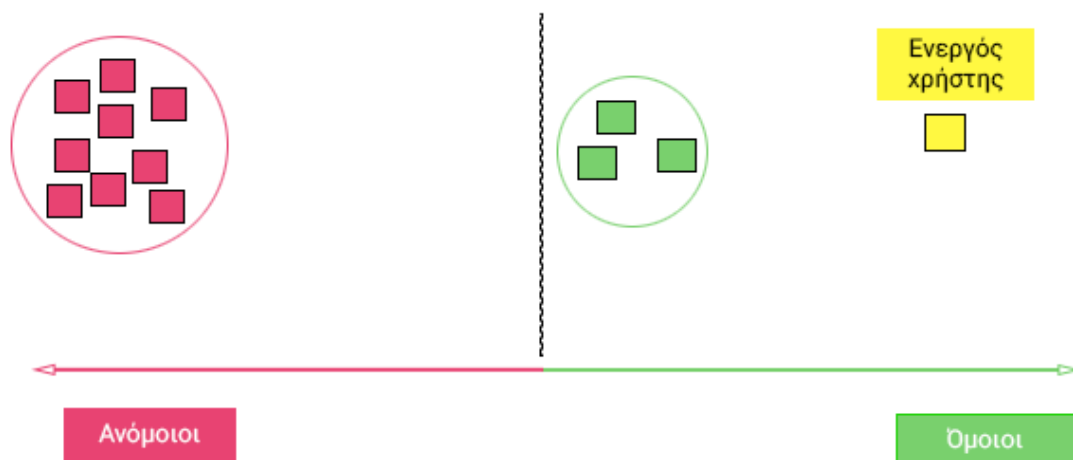


Εικόνα 16. Παράδειγμα αραιού rating matrix (Falk, 2019)

Gray sheep – Black sheep πρόβλημα

Το πρόβλημα του gray sheep αναφέρετε σε ένα μέρος των χρηστών της εφαρμογής, όπου οι προτιμήσεις τους είναι τέτοιες, που δεν συμφωνούν ή διαφωνούν σταθερά με κάποια συγκεκριμένη ομάδα από χρήστες, με αποτέλεσμα να μην επωφελούνται από τις τεχνικές του Collaborative filtering (Εικόνα 17).

Το πρόβλημα του black sheep αναφέρεται σε ένα μέρος από χρήστες όπου οι ιδιότυπες προτιμήσεις τους, κάνουν τις προτάσεις από ένα RS σχεδόν αδύνατες. Ωστόσο το πρόβλημα του black sheep υπάρχει και ευρύτερα από κλάδο της πληροφορικής, πράγμα που καθιστά το πρόβλημα μια αποδεκτή αποτυχία των RS (Su & Khoshgoftaar, 2009).



Εικόνα 17. Παράδειγμα Gray sheep χρήστη

1.4 Group Recommendations

Με τον όρο *group recommendations* εννοούμε την ιδιότητα ενός *recommendation system* να κάνει προτάσεις σε μια ομάδα από χρήστες, αντί των γνωστών ατομικών προτάσεων που αναφέρθηκαν μέχρι σε αυτό το σημείο.

Το κύριο πρόβλημα που πρέπει να λυθεί είναι ο τρόπος με τον οποίο το RS θα προσαρμόσει τις ατομικές προτάσεις των χρηστών μιας ομάδας σε μια ενιαία λίστα με προτάσεις που θα απευθύνονται σε όλους (Masthoff, 2010).

Ένας τρόπος με τον οποίο μπορεί να λυθεί το παραπάνω πρόβλημα είναι με κάποια στρατηγική συνάθροισης, η οποία δίνοντας για παράδειγμα τις λίστες με τα ατομικά *predict ratings*, θα τις συνάθροισα και θα μας επιστρέψει τελικά μία μοναδική λίστα με *predict ratings*.

Για τα παρακάτω παραδείγματα έστω ότι έχουμε τον ακόλουθο πίνακα για μια ταινία X:

| Χρήστες | Βαθμολογίες RS |
|---------|----------------|
| A | 4 |
| B | 2 |
| Γ | 5 |

Κάποιες από τις στρατηγικές συναθροίσεις βαθμολογιών είναι:

- **Least misery:** Επιλέγοντας για κάθε item το μικρότερο *predict rating*. Παράδειγμα:
Για την ταινία X επιλέχθηκε η βαθμολογία $\text{Min}(4, 2, 5) = 2$.
- **Most pleasure:** Επιλέγοντας για κάθε item το μεγαλύτερο *predict rating*.
Παράδειγμα: Για την ταινία X επιλέχθηκε η βαθμολογία $\text{Max}(4, 2, 5) = 5$.
- **Average:** Η βαθμολογία για κάθε item προκύπτει από το μέσω όρο των *predict rating*.
Παράδειγμα: Για την ταινία X επιλέχθηκε η βαθμολογία $(4+2+5)/3 = 3.6$
- **Multiplicative:** Για κάθε item παράγεται ένα είδος score που μπορεί να χρησιμοποιεί για την επιλογή των προτάσεων.
Παράδειγμα: Για την ταινία X επιλέχθηκε το score $(4*2*5) = 40$

- **Fairness:** Για κάθε item επιλέγεται κάθε φορά εκ περιτροπής η βαθμολογία ενός από τους τρεις χρήστες.
Παράδειγμα: Εάν ήταν η σειρά του πρώτου χρήστη, η ταινία X θα έπαιρνε βαθμολογία ίση με 4.

Αφότου εφαρμοστεί μία από τις παραπάνω στρατηγικές συναθροίσεις, θα προκύψει ένας πίνακας με predict ratings ή scores (στην περίπτωση του Multiplicative). Έπειτα μπορούν να ταξινομηθούν και τελικά να δοθούν οι n καλύτερες ταινίες με βάση τη βαθμολογία τους σαν προτάσεις στην ομάδα των χρηστών (Jameson & Smyth, 2007).

2. Μεθοδολογία Έρευνας

Σε αυτό το κεφάλαιο επεξηγείται η μεθοδολογία έρευνας που ακολουθήθηκε για την διεκπεραίωση της πτυχιακής εργασίας.

Όπως αναφέρθηκε και στον σκοπό της πτυχιακής στόχοι ήταν η κατανόηση του θεωρητικού υπόβαθρου που χρειαζόταν για την δημιουργία ενός recommendation system, καθώς και η δημιουργία μιας εφαρμογής Android που θα ήταν σε θέση να κάνει προσωποποιημένες προτάσεις για ταινίες.

Αρχικά, πραγματοποιήθηκε αναζήτηση σχετικής βιβλιογραφίας για τα συστήματα προτάσεων προκειμένου να διαπιστωθεί τι είναι ένα recommendation system, ποιες είναι οι φάσεις παραγωγής προτάσεων, τι δεδομένα χρειάζεται, ποιες είναι οι δύο βασικές τεχνικές καθώς και ποια είναι τα θετικά και τα αρνητικά αυτών.

Έπειτα έγινε εύρεση της κατάλληλης βιβλιοθήκης της γλώσσα python που θα βοηθούσε στην δημιουργία του recommendation system. Ως βιβλιοθήκη επιλέχθηκε η SciPy Surprise, η οποία είχε μια μεγάλη λίστα με ετοιμους υλοποιημένους αλγορίθμους για collaborative filtering όπως ο SVD και ο KNN, καθώς και άλλα χρήσιμα εργαλεία όπως η υλοποίηση μιας κλάσης με όνομα GridSearchCV που χρησιμοποιείται για την βελτιστοποίηση των παραμέτρων που χρησιμοποιούνται κατά την εκπαίδευση των μοντέλων.

Ένα από τα προβλήματα που έπρεπε να ξεπεραστεί, δεδομένου πως θα γινόταν χρήση τεχνικής collaborative filtering, είναι το πρόβλημα Cold Start. Η ύπαρξη μονάχα μιας μεγάλης λίστας από νέες ταινίες θα προκαλούσε τον πίνακα rating matrix που χρησιμοποιείται σαν είσοδο στο collaborative filtering, να ήταν άδειος, με αποτέλεσμα οι προτάσεις να μην είναι ακριβής. Για τους σκοπούς της πτυχιακής αποφασίστηκε να χρησιμοποιηθεί σαν βάση για την εφαρμογή ένα dataset με βαθμολογίες για ταινίες με το όνομα Movielens 1M. Το movielens αποτελείται από μια λίστα με 6000 χρήστες, 4000 ταινίες και περιέχει 1 εκατομμύριο βαθμολογίες. Με αυτόν τον τρόπο θα υπήρχαν αρκετές βαθμολογίες για τις ταινίες του movielens οι οποίες και τελικά κατέληξαν να είναι η λίστα με τις ταινίες που θα είχε διαθέσιμες η εφαρμογή. Το δεύτερο πρόβλημα που έπρεπε να αντιμετωπιστεί είναι ότι κατά την εισαγωγή ενός νέου χρήστη θα έπρεπε να γίνει κάποιου είδους αρχικοποίηση, δηλαδή ο χρήστης θα έπρεπε να βαθμολογήσει έναν αριθμό από ταινίες αρχικά ώστε το μοντέλο να μπορεί να βρει παρόμοιους χρήστες μέσα από το dataset. Το πρόβλημα αυτό επιλύθηκε με την προσθήκη μίας επιπλέον οθόνης στην εφαρμογή όπου ο χρήστης πρέπει να βαθμολογήσει έναν αριθμό από αντιπροσωπευτικές ταινίες.

Έπειτα έγινε αναζήτηση για κάποιο σχετικό API από το οποίο θα γινόταν ανάκτηση των πληροφοριών για τις ταινίες που θα εμφανίζονταν στον χρήστη ώστε να διαπιστωθεί εάν τον ενδιαφέρουν οι προτάσεις που θα εμφανιζόντουσαν στις συνεδρίες. Τελικά επιλέχθηκε το API TMDb⁴ από το οποίο έγινε συλλογή πληροφοριών όπως περιγραφή ταινίας, είδος ταινίας, ημερομηνία κυκλοφορίας καθώς και ένα μέσο ορό βαθμολογίας χρηστών στο site του TMDb. Ωστόσο δεν ήταν εφικτό να βρεθούν πληροφορίες για έναν αριθμό από τις ταινίες που υπήρχαν στο Movielen. Ο αριθμός των ταινιών που υπάρχουν διαθέσιμες στην εφαρμογή είναι 3383.



Εικόνα 18. TMDb Logo

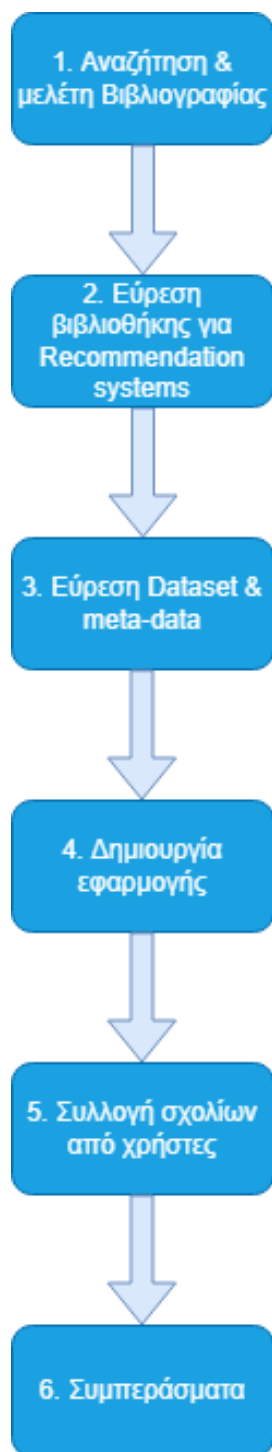
Έπειτα ακολούθησε μια σειρά από εκπαιδευτικά προγράμματα για την εκμάθηση της ανάπτυξης μια εφαρμογής για κινητά Android με την γλώσσα Kotlin.

Αμέσως μετά ξεκίνησε η δημιουργία της εφαρμογής. Δημιουργήθηκε αρχικά το fragment που θα γινόταν η αρχικοποίησή του προφίλ του νέου χρήστη. Επιπλέον δημιουργήθηκε η λίστα με τους φίλους όπου ο χρήστης θα μπορούσε να στείλει αιτήματα φιλίας σε άλλους χρήστες όπου και εκεί τελικά έγινε προσθήκη της δυνατότητας δημιουργίας συνεδρίας.

Παράλληλα αναπτύχθηκε ένα πρόγραμμα σε python όπου παράγει τα δύο μοντέλα με βάση το dataset του movielens συγχωνευμένο με τις βαθμολογίες των πραγματικών χρηστών.

Το τελευταίο δίλημα ήταν ποια μέθοδος συνάθροισης θα χρησιμοποιηθεί για την ομαδική λίστα με προτάσεις για κάθε συνεδρία στην περίπτωση χρήσης SVD, όπου τελικά επιλέχθηκε η multiplicative μέθοδος συνάθροισης. Τέλος, συλλέχθηκαν γενικά σχόλια αναδράσεις σχετικά με την εφαρμογή.

⁴ <https://www.themoviedb.org/>



Εικόνα 19. Μεθοδολογία έρευνας

3. Αλγοριθμικές τεχνικές

Σε αυτό το κεφάλαιο θα εξηγηθούν οι αλγοριθμικές τεχνικές που χρησιμοποιήθηκαν στην εφαρμογή για την εκπαίδευση των δύο μοντέλων, και για την παραγωγή των προτάσεων στις συνεδρίες των χρηστών.

3.1 Δημιουργία Dataset

Ο παρακάτω κώδικας python χρησιμοποιείται στην εφαρμογή για την δημιουργία του dataset το οποίο δίνεται σαν είσοδος και στους δύο αλγορίθμους SVD και KNN-Baseline.

Αρχικά χρησιμοποιείται η παρακάτω συνάρτηση `export_data_file()` για τη δημιουργία του αρχείου με τα ratings.

```
def export_data_file():
    # file path to movielens ratings
    filePath = "resources/initial_ratings.dat"

    ratings = []
    with io.open(filePath, 'r', encoding='ISO-8859-1') as f:
        for line in f:
            line_split = line.split('::')
            ratings.append(line_split)

    # get the movie ids from the Data base
    movies = list(db.Movies.find())
    movies_ids = [movie["movielens_id"] for movie in movies]

    user_ratings = list(db.Ratings.find())

    for user in user_ratings:
        for movie_id in user["ratings"].keys():
            ratings.append([user['uid'], movie_id, user["ratings"][movie_id], "00000000"])

    with open("resources/ratings.dat", 'w') as f:
        for rating in ratings:
            if rating[1] in movies_ids:
                uid = rating[0]
                movie_id = rating[1]
                rating_value = int(rating[2])
                timestamp = rating[3].strip()
                f.write(f"{uid}::{movie_id}::{rating_value}::{timestamp}\n")
```

Εικόνα 20. Κώδικας συνάρτησης `export_data_file`

Η συνάρτηση `create_dataset()` φορτώνει το αρχείο `ratings` που παράγει η `export_data_file()` ώστε να δημιουργήσει και να επιστρέψει το `dataset` χρησιμοποιώντας τις βοηθητικές κλάσεις του SciPy Surprise Dataset και Reader:

```
def create_dataset():
    file_path_large = "resources/ratings.dat"
    reader_large = Reader(line_format='user item rating timestamp', sep=':::')
    dataset = Dataset.load_from_file(file_path=file_path_large, reader=reader_large)
    return dataset
```

Εικόνα 21. Κώδικας συνάρτησης `create_dataset`

3.2 Μοντέλο SVD

Το μοντέλο που παράγεται παρακάτω γίνεται εξαγωγή σε ένα αρχείο από το οποίο έπειτα μπορεί να χρησιμοποιηθεί για να παράγει προτάσεις για τους χρήστες της εφαρμογής που θα παρουσιαστεί στο κεφάλαιο 4. Πιο συγκεκριμένα, χρησιμοποιείται στην δημιουργία του πρώτου είδους συνεδρίας μεταξύ δύο φίλων για την παραγωγή εξατομικευμένων προτάσεων με βάση παλαιότερες βαθμολογίες τους.

Η παρακάτω συνάρτηση χρησιμοποιήθηκε για να βρεθεί ο καλύτερος συνδυασμός παραμέτρων για την εκπαίδευση του SVD. Πιο συγκεκριμένα χρησιμοποιήθηκε η κλάση `GridSearchCV` από την βιβλιοθήκη του SciPy Surprise η οποία δίνοντας της μια λίστα με τιμές για κάθε παράμετρο του SVD, παράγει ένα μοντέλο με κάθε δυνατό συνδυασμό ώστε να βρει αυτό με το μικρότερο RMSE (Root Mean Square Error) Score.

```

def find_best_combination():
    # Use custom dataset
    data = create_dataset()

    param_grid = {'n_factors': [100, 150],
                  'n_epochs': [20, 25, 30],
                  'lr_all': [0.005, 0.01],
                  'reg_all': [0.02, 0.05, 0.1]}

    gs = GridSearchCV(SVD, param_grid, measures=['rmse', 'mae'], cv=3)

    gs.fit(data)

    # best RMSE score
    print(gs.best_score['rmse'])

    # combination of parameters that gave the best RMSE score
    print(gs.best_params['rmse'])

```

Εικόνα 22. Κώδικας συνάρτησης find_best_combination

GridSearchCV Output:

```

0.8630248433773101
{'n_factors': 150, 'n_epochs': 25, 'lr_all': 0.01, 'reg_all': 0.05}

Process finished with exit code 0

```

Εικόνα 23. Αποτελέσματα GridSearchCV για SVD

Η παρακάτω συνάρτηση train_svd(), χρησιμοποιώντας τις τιμές που πήραμε για τις παραμέτρους, εκπαιδεύει και αποθηκεύει το μοντέλο του SVD σε ένα αρχείο. Το αρχείο αυτό θα χρησιμοποιηθεί έπειτα για τη δημιουργία της λίστας με τις προτάσεις των ταινιών στο πρώτο είδος συνεδρίας της εφαρμογής.

```

def train_svd():
    # load custom dataset
    dataset = create_dataset()
    trainset = dataset.build_full_trainset()

    algo = SVD(_n_epochs=25, n_factors=150, lr_all=0.01, reg_all=0.5, verbose=True)
    algo.fit(trainset)

    # dump trained algorithm
    file_name = os.path.expanduser('TrainedModels/trainedSVDAlgo.model')
    dump.dump(file_name=file_name, algo=algo)
    print("SVD Training done!")

```

Εικόνα 24. Κώδικας συνάρτησης train_svd

Παράδειγμα για χρήση του μοντέλου:

Το παρακάτω ενδεικτικό πρόγραμμα χρησιμοποιεί το μοντέλο που εκπαιδεύσαμε, για την πρόβλεψη της βαθμολογίας για μια ταινία, για έναν συγκεκριμένο χρήστη από το dataset, χρησιμοποιώντας την μέθοδο predict του μοντέλου με είσοδο το id του χρήστη και το id της ταινίας.

```

# load a tuple with (prediction, trained-algorithm)
algo = dump.load('TrainedModels/trainedSVDAlgo.model')
# get just the algorithm
algo = algo[1]
# Read the mappings raw id <-> movie name
rid_to_name, name_to_rid = read_item_names()
# Retrieve inner id of the selected Movie
movieName = "Toy Story (1995)"
# movie id used in the dataset
raw_id = name_to_rid[movieName]
# user id used in the dataset
uid = "jB0MVvnKcPhj2GcJpa86utf8req1"
# make prediction
pred = algo.predict(uid, raw_id)

print(f"{movieName}: est. {pred.est}")

```

Εικόνα 25. Κώδικας για την πρόβλεψη μιας ταινίας

Αποτέλεσμα παραπάνω συνάρτησης:

```
Toy Story (1995): est. 3.745124306923437  
Process finished with exit code 0
```

Εικόνα 26. Προβλεπόμενη βαθμολογία

3.3 Μοντέλο KNN-Baseline

Το μοντέλο που παράγεται παρακάτω γίνεται εξαγωγή σε ένα αρχείο από το οποίο έπειτα χρησιμοποιείται από την εφαρμογή ώστε να βρεθούν οι K κοντινότεροι γείτονες μιας ταινίας. Πιο συγκεκριμένα χρησιμοποιείται στο δεύτερο είδος συνεδρίας όπου ο χρήστης κατά την δημιουργία της συνεδρίας επιλέγει μια ταινία. Έπειτα η λίστα με τις προτάσεις θα περιέχει τις 20 πιο όμοιες ταινίες (δηλαδή τους 20 κοντινότερους γείτονες για την συγκεκριμένη ταινία).

Η παρακάτω συνάρτηση `train_knn()` εκπαιδεύει και έπειτα αποθηκεύει το μοντέλο ώστε να μπορεί να χρησιμοποιηθεί για τον δεύτερο τύπο συνεδρίας.

```
def train_knn():  
    # load custom dataset  
    dataset = create_dataset()  
    trainset = dataset.build_full_trainset()  
  
    # train the algorithm to compute the similarities between items  
    sim_options = {'name': 'pearson_baseline',  
                  'user_based': False}  
  
    algo = KNNBaseline(k=40, min_k=15, sim_options=sim_options)  
    algo.fit(trainset)  
  
    # dump trained algorithm  
    file_name = os.path.expanduser('TrainedModels/trainedKNNBaseline.model')  
    dump.dump(file_name=file_name, algo=algo)
```

Εικόνα 27. Κώδικας συνάρτησης `train_knn`

Ενδεικτικό παράδειγμα χρήσης του παραπάνω μοντέλου:

```
# load a tuple with (prediction, trained-algorithm)
algo = dump.load('TrainedModels/trainedKNNBaseline.model')
# get just the algorithm
algo = algo[1]
# Read the mappings raw id <-> movie name
rid_to_name, name_to_rid = read_item_names()
# Retrieve inner id of the selected Movie
movieName = "Aladdin (1992)"
raw_id = name_to_rid[movieName]
input_movie_inner_id = algo.trainset.to_inner_iid(raw_id)
# Retrieve inner ids of the nearest neighbors of Aladdin.
input_movie_neighbors = algo.get_neighbors(input_movie_inner_id, k=20)
# Convert inner ids of the neighbors into names.
input_movie_neighbors = (algo.trainset.to_raw_iid(inner_id) for inner_id in input_movie_neighbors)
input_movie_neighbors = (rid_to_name[rid] for rid in input_movie_neighbors)

# Print Results
print(f'\nThe 10 nearest neighbors of {movieName} are:')
for movie in input_movie_neighbors:
    print(movie)
```

Εικόνα 28. Κώδικας ενδεικτικού παραδείγματος

Αποτέλεσμα παραδείγματος:

```
The 10 nearest neighbors of Aladdin (1992) are:
Beauty and the Beast (1991)
Little Mermaid, The (1989)
Lion King, The (1994)
Toy Story (1995)
Bug's Life, A (1998)
Cinderella (1950)
Tarzan (1999)
Mulan (1998)
Toy Story 2 (1999)
Hercules (1997)
101 Dalmatians (1961)
Lady and the Tramp (1955)
Pinocchio (1940)
Rescuers, The (1977)
Mary Poppins (1964)
Father of the Bride (1950)
Rescuers Down Under, The (1990)
It's a Wonderful Life (1946)
Great Mouse Detective, The (1986)
Sleeping Beauty (1959)

Process finished with exit code 0
```

Εικόνα 29. Αποτελέσματα παραδείγματος KNN

3.4 Group Recommendations με SVD

Η συνάρτηση `get_group_recommendations` χρησιμοποιείται από την εφαρμογή για την παραγωγή της λίστας με τις τελικές προτάσεις για το πρώτο είδος συνεδρίας. Πιο αναλυτικά η συνάρτηση παίρνει σαν είσοδο τα `user id` των δύο χρηστών καθώς και μια λίστα με `movie ids`. Υπολογίζει τα `prediction ratings` των δύο χρηστών, και έπειτα τα συναθροίζει με την στρατηγική `multiplicative` (βλ. κεφάλαιο 2) για να παράγει ένα είδους `score` για κάθε ταινία. Τέλος ταξινομεί τις ταινίες με βάση το `score` τους, και επιστρέφει τις 50 με το υψηλότερο `score`.

```
def get_group_recommendations(vid: str, friend_vid: str, list_of_movies):
    # load a tuple with (prediction, trained-algorithm)
    algo = dump.load("TrainedModels/trainedSVDAlgo.model")
    # get just the algorithm
    algo = algo[1]

    user_combined_predictions = []
    for movie_id in list_of_movies:
        # get first user predict rating
        user_pred = algo.predict(vid, movie_id)
        # get second user predict rating
        friend_user_pred = algo.predict(friend_vid, movie_id)
        # aggregate predicted ratings with Multiplicative strategy
        user_combined_predictions.append((movie_id, (user_pred.est * friend_user_pred.est)))

    # sort list of ratings
    user_combined_predictions.sort(key=lambda x: x[1], reverse=True)

    # get top 50 movies
    user_combined_predictions = user_combined_predictions[0:50]

    return user_combined_predictions
```

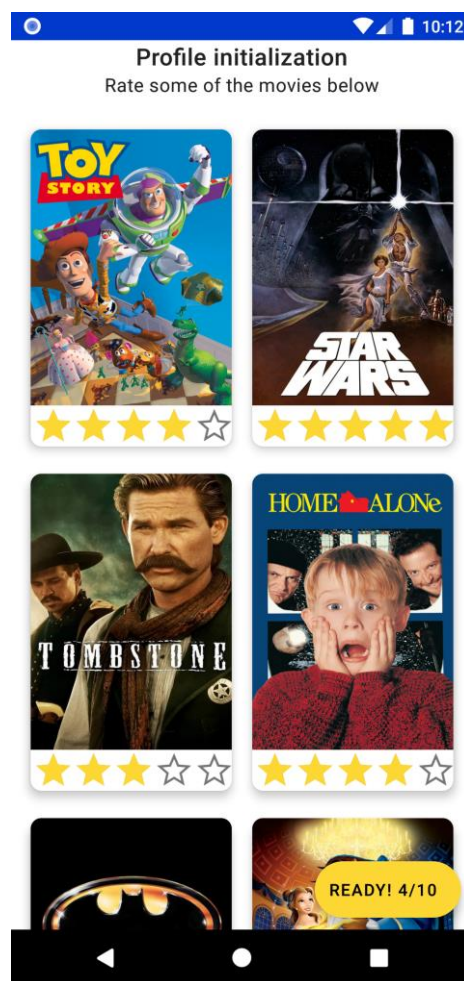
Εικόνα 30. Κώδικας συνάρτησης `get_group_recommendations`

4. Επισκόπηση Εφαρμογής

Σε αυτό το κεφάλαιο θα γίνει επισκόπηση της εφαρμογής που δημιουργήθηκε με αφορμή την πτυχιακή εργασία.

4.1 Οθόνη αρχικοποιήσεις προφίλ χρήστη

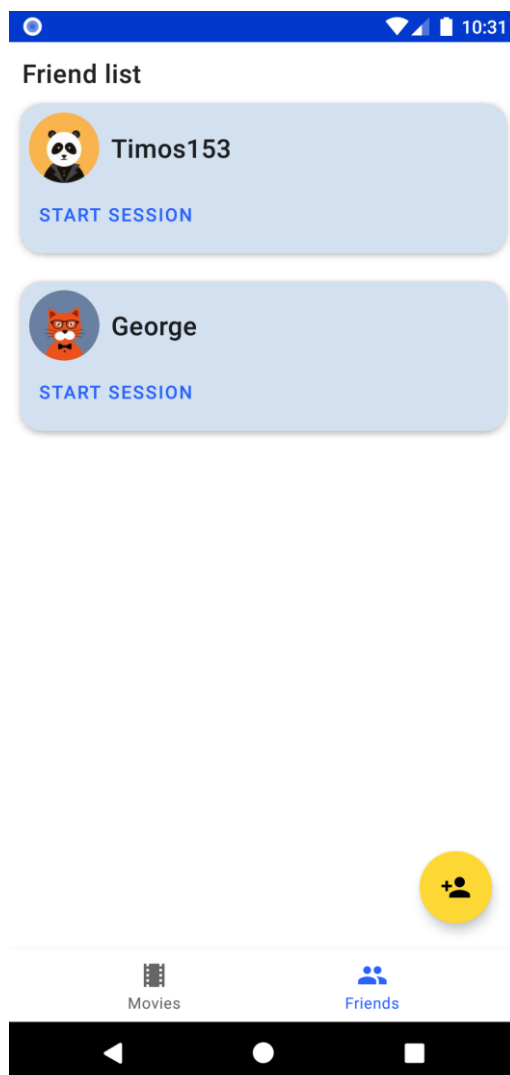
Η πρώτη οθόνη που δημιουργήθηκε για την εφαρμογή, ήταν η οθόνη αρχικοποιήσεις για το προφίλ χρήστη. Μετά την δημιουργία λογαριασμού ο χρήστης κατευθύνεται στην παρακάτω οθόνη όπου, από μια λίστα με 17 αντιπροσωπευτικές, διάσημες, ταινίες από διάφορα είδη (π.χ. κωμωδία, δράσης), ο χρήστης πρέπει να βαθμολογήσει τουλάχιστον 10 ταινίες ώστε να εισαχθούν οι βαθμολογίες στο rating matrix και να μπορεί το μοντέλο να παράξει προτάσεις μετά την επόμενη εκπαίδευση των μοντέλων.



Εικόνα 31. Οθόνη αρχικοποιήσεις προφίλ χρήστη

4.2 Οθόνη λίστας φίλων

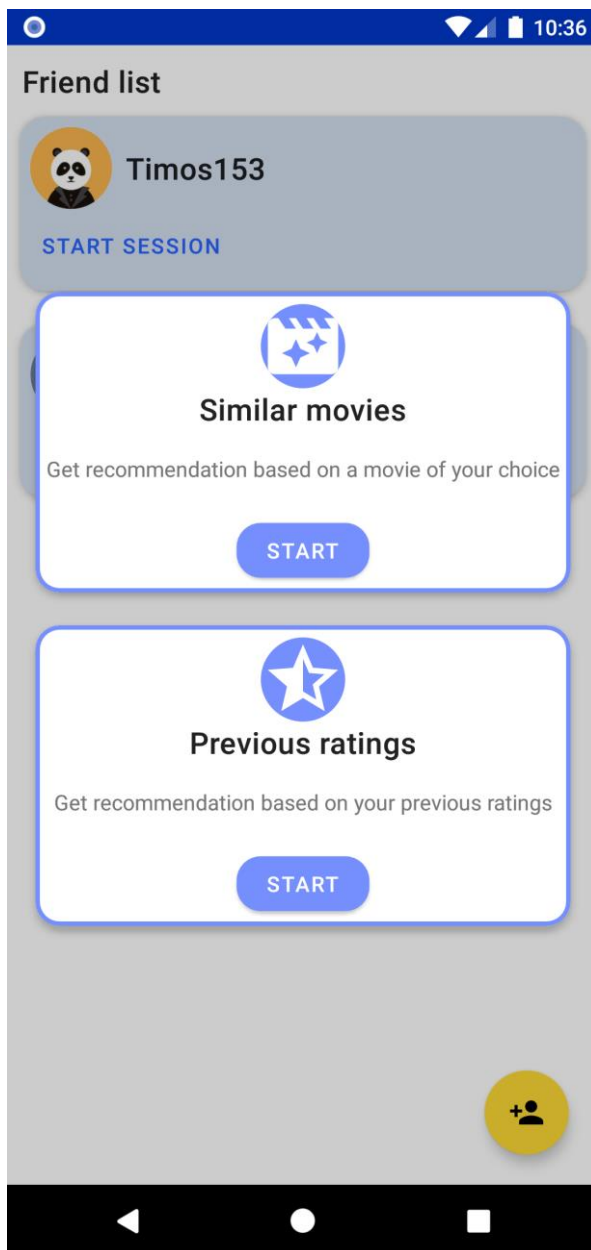
Η επόμενη οθόνη που δημιουργήθηκε ήταν η οθόνη με την λίστα των φίλων. Σε αυτήν την οθόνη ο χρήστης μπορεί να στείλει αιτήματα φιλίας σε άλλους χρήστες πληκτρολογώντας το username τους. Εάν ο χρήστης αποδεχτεί το αίτημα τότε θεωρούνται φίλοι και μπορούν να δημιουργήσουν συνεδρία μεταξύ ώστε να αποφασίσουν τι ταινία θέλουν να δουν με την βοήθεια του recommendation system. Μάλιστα στέλνεται και ειδοποιήσει στο κινητό του χρήστη που δέχτηκε το αίτημα ώστε να ενημερωθεί κατάλληλα.



Εικόνα 32. Οθόνη λίστας φίλων

4.3 Διάλογος επιλογής συνεδρίας

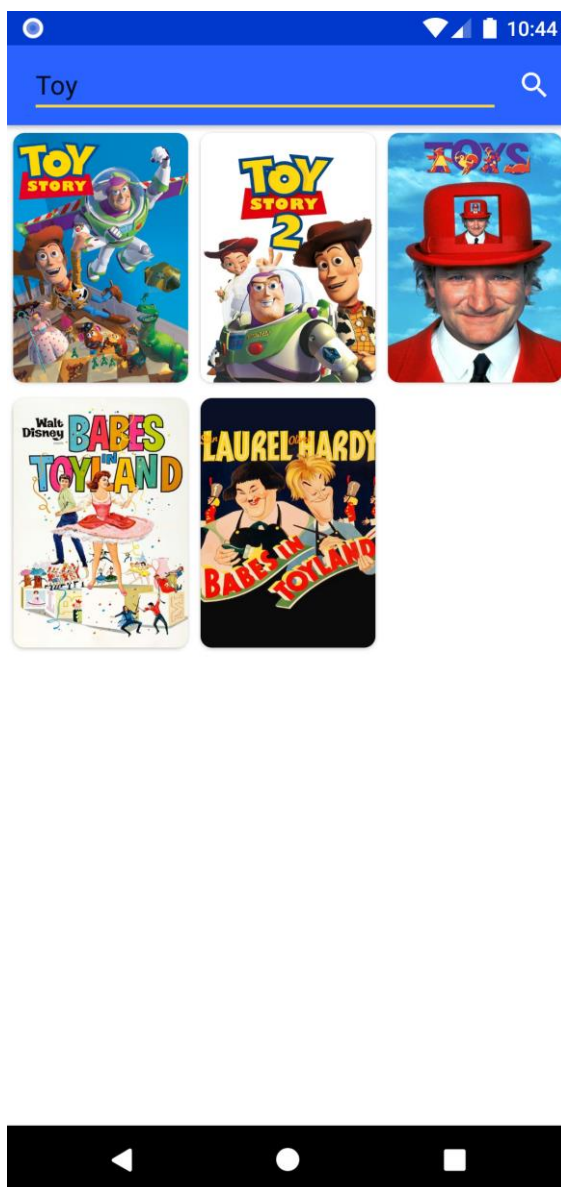
Πατώντας το κουμπί start session σε κάποιον από την λίστα φίλων ο χρήστης έχει την δυνατότητα να επιλέξει τι είδους συνεδρία θέλουν να ξεκινήσουν.



Εικόνα 33. Διάλογος επιλογής συνεδρίας

4.4 Οθόνη επιλογής ταινίας για συνέδρια

Σε αυτή την οθόνη ο χρήστης πληκτρολογεί το όνομα της ταινίας για την οποία θέλει να του βρει παρόμοιες ταινίες η εφαρμογή. Το σύστημα του επιστρέφει τις ταινίες που ο τίτλος ταιριάζει με το όνομα που πληκτρολόγησε. Έπειτα αφού ο χρήστης επιλέξει την επιθυμητή ταινία, η εφαρμογή με βάση την ταινία αυτή, καλεί το μοντέλο KNN που έχει εκπαιδεύσει ώστε να βρει παρόμοιες ταινίες με αυτήν που επέλεξε ο χρήστης. Οι πρώτες 20 αποτελούν και τη λίστα με τις ταινίες από τις οποίες θα ψηφίσουν οι χρήστες.



Εικόνα 34. Οθόνη επιλογής ταινίας

4.5 Οθόνη επιλογής είδους ταινίας για συνεδρία

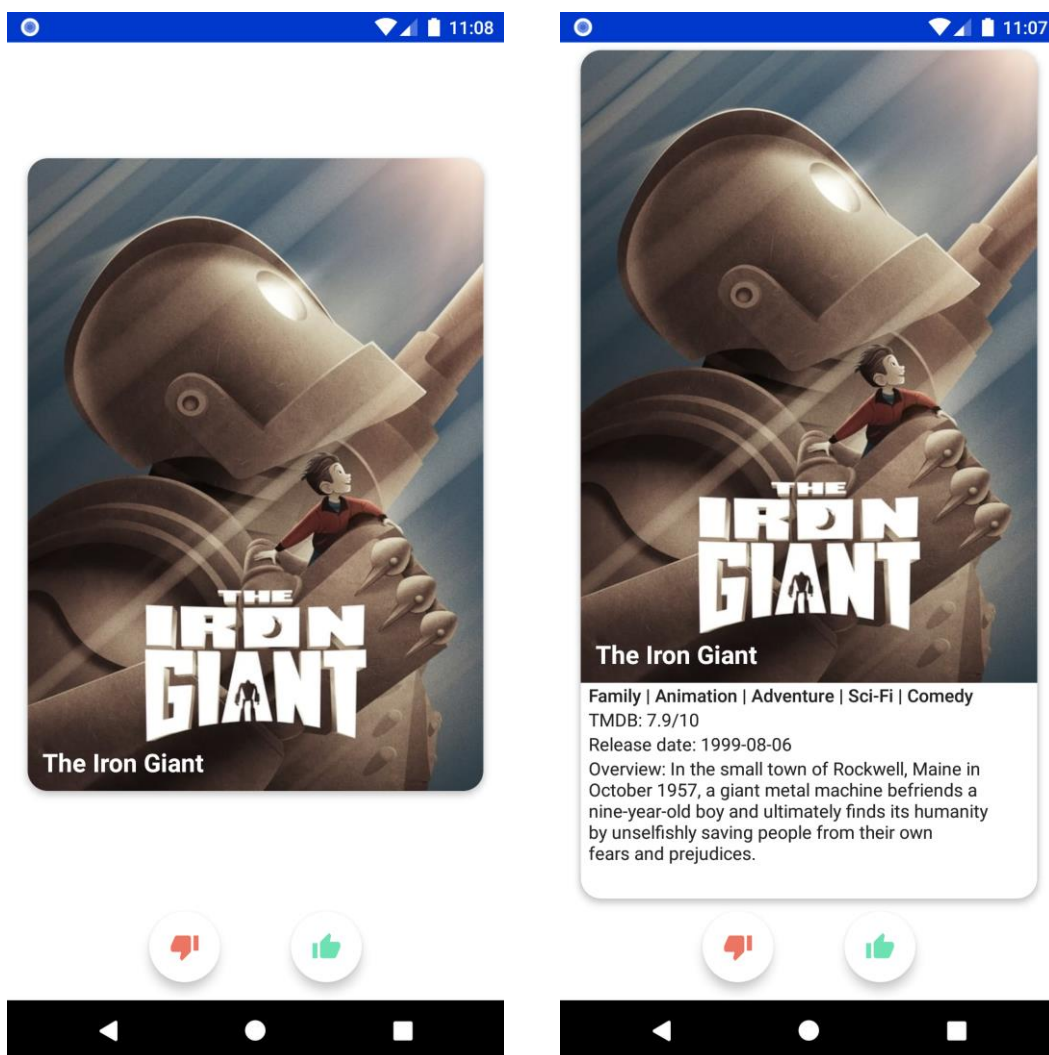
Σε αυτή την οθόνη ο χρήστης έχει την δυνατότητα να επιλέξει με βάση τι είδος ταινίας θέλουν το recommendation system να τους προτείνει ταινίες. Η συγκεκριμένη οθόνη εμφανίζεται κατά το δεύτερο είδος συνεδρίας, όπου κάνει προτάσεις με βάση τις παλιές βαθμολογίες που διαθέτει για τους δύο χρήστες. Στην περίπτωση που ο χρήστης δεν επιλέξει κάποιο είδος, λαμβάνονται υπόψη όλα τα είδη των ταινιών. Στην περίπτωση που επιλέξουν παραπάνω από ένα είδος, τότε συναθροίζονται οι ταινίες που περιέχουν κάποιο από τα επιλεγμένα είδη. Η συνεδρία θα περιέχει τις πρώτες 50 ταινίες που προτείνει το μοντέλο του SVD.

The screenshot shows a mobile application interface for selecting movie filters. At the top, there is a blue status bar with a back arrow, signal strength, Wi-Fi, battery, and the time 10:53. Below the status bar, the title "Choose filters" is centered, followed by the question "What kind of movie do you want to see?". The main area contains a grid of 18 light blue rounded rectangular buttons, each with a checkbox and a genre name. The genres are: Action, Adventure, Animation (checked), Comedy, Crime, Document, Drama, Family (checked), Fantasy (checked), History, Horror, Music, Mystery, Romance, Sci-Fi, TV Movie, Thriller, War, and Western. At the bottom center, there is a yellow rounded rectangular button with the text "START". Below the "START" button is a black navigation bar with a white back arrow, a white circle, and a white square.

Εικόνα 35. Οθόνη επιλογής είδους ταινίας

4.6 Οθόνη ψηφοφορίας συνεδρίας

Έπειτα από την επιλογή ταινίας, ή την επιλογή είδους, οι χρήστες εισέρχονται στην συνεδρία και μεταφέρονται στην παρακάτω οθόνη όπου και πρέπει να ψηφίσουν τις προτάσεις που τους έχει δώσει το recommendation system. Αφού ψηφίσουν τουλάχιστον 10 ταινίες, έχουν την δυνατότητα να στείλουν τις ψήφους τους στο server. Αφού ψηφίσουν και οι δύο χρήστες, στην περίπτωση που βρεθούν ταινίες στις οποίες έχουν ψηφίσει θετικά και οι δύο, μεταφέρονται στην τελική οθόνη όπου τους εμφανίζονται τα τελικά αποτελέσματα. Στην περίπτωση που δεν βρεθεί καμία κοινή ταινία, έχουν την δυνατότητα να συνεχίσουν ψηφίζοντας τις υπόλοιπες ταινίες. Στην περίπτωση που ψηφίσουν όλες τις ταινίες χωρίς να βρεθεί κάποια κοινή, εμφανίζεται κατάλληλο μήνυμα στην εφαρμογή.



Εικόνα 36. Οθόνη ψηφοφορίας ταινιών

4.7 Οθόνη με τελικά αποτελέσματα

Αφού ψηφίσουν οι χρήστες θα μεταφερθούν στην τελική οθόνη όπου θα τους εμφανιστεί ένα από τα παρακάτω μηνύματα ανάλογα με: αν πρέπει να περιμένουν τον άλλο χρήστη να ψηφίσει, να δουν τα τελικά αποτελέσματα, να συνεχίσουν την ψηφοφορία στην περίπτωση που δεν βρέθηκε κοινή ταινία ή ότι η συνεδρία απέτυχε εφόσον δεν βρέθηκε κάποια κοινή ταινία που να συμφωνήσαν.



You must wait for the others to vote
Come back later!



Results



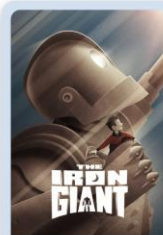
The Wrong Trousers

Animation | Comedy | Family

TMDB: 7.7/10

Release date: 1993-12-17

Wallace rents out Gromit's former bedroom to a penguin, who takes up an interest in the techno pants created by Wallace. However, Gromit later learns that the penguin is a wanted criminal.



The Iron Giant

Family | Animation | Adventure |

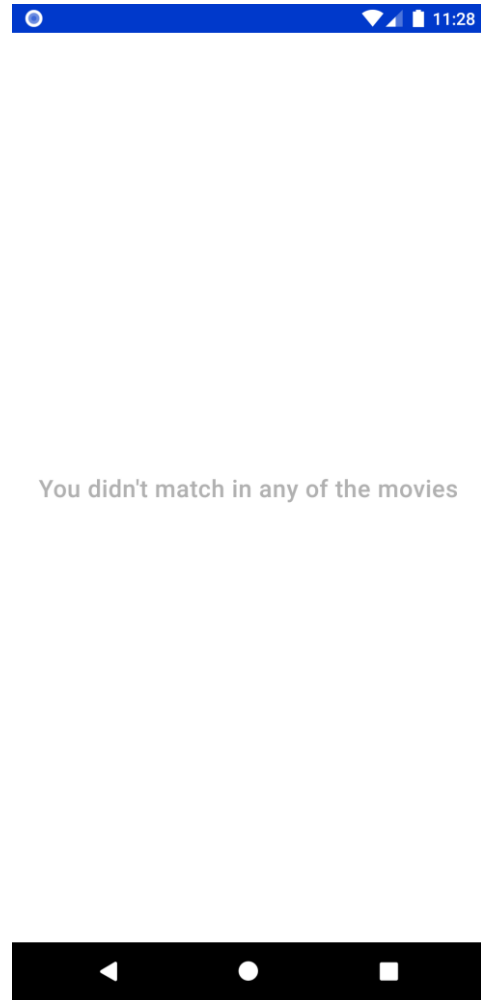
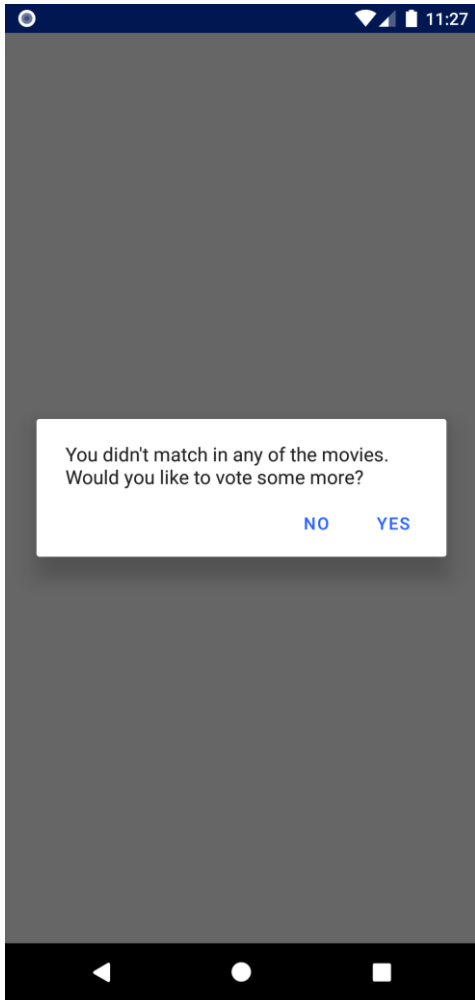
Sci-Fi | Comedy

TMDB: 7.9/10

Release date: 1999-08-06

In the small town of Rockwell, Maine in October 1957, a giant metal machine befriends a nine-year-old boy and ultimately finds its humanity by unselfishly saving people from their own fears and prejudices.





Εικόνα 37. Οθόνη με τελικά αποτελέσματα

5. Αξιολόγηση & Περιπτώσεις χρήσης

Η εφαρμογή δόθηκε σε δύο ζευγάρια από χρήστες ηλικίας 22-26 ετών. Ζητήθηκε από τους χρήστες η δημιουργία ατομικού προφίλ και έπειτα η δοκιμή των δύο τύπων συνεδριών, ώστε να διαπιστωθεί εάν θα κατάφερνε η εφαρμογή να προσφέρει ενδιαφέρον προτάσεις.

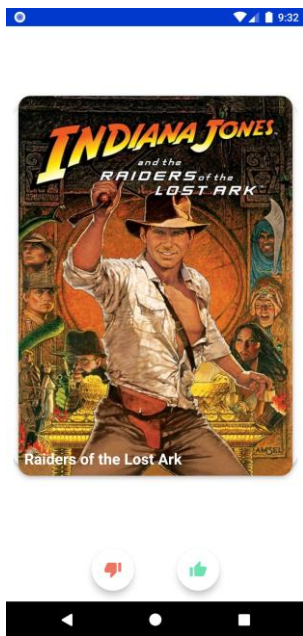
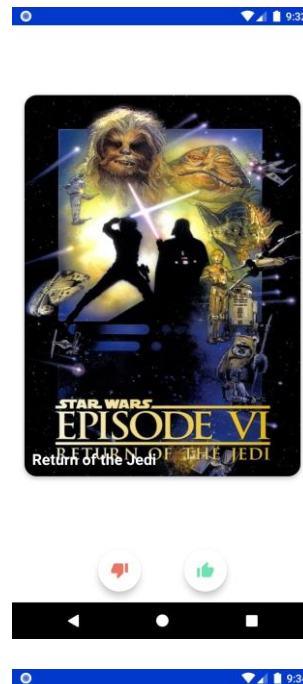
5.1 Περιπτώσεις χρήσης για συνεδρίες

5.1.1 Συνεδρίες παρόμοιων ταινιών

Τα αποτελέσματα που παρουσιάζονται παρακάτω αποτελούν τις 4 πρώτες ταινίες από τις 20 που πρότεινε το RS για την αντίστοιχα ταινία που επιλέχθηκε από το κάθε ζευγάρι.

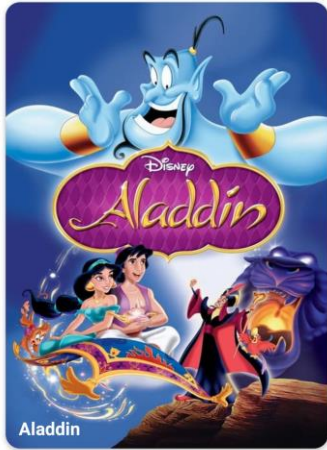
Τα αποτελέσματα που παρουσιάζονται σε αυτήν την ενότητα βασίζονται στην εκπαίδευση του μοντέλου με το dataset του movielens και τον αλγόριθμο KNN. Μετά την επιλογή της επιθυμητής ταινίας (από τον χρήστη που ξεκίνησε τη συνεδρία) το σύστημα τροφοδοτεί αυτήν την ταινία στο μοντέλο, το οποίο επιστρέφει τους 20 κοντινότερους γείτονες της ταινίας. Στη συνέχεια αυτές οι ταινίες παρουσιάζονται με φθίνουσα σειρά με βάση το σκορ στους χρήστες.

Για το πρώτο ζευγάρι χρηστών όπου επιλέχθηκε η ταινία **Star wars**, οι 4 πρώτες προτάσεις ήταν:

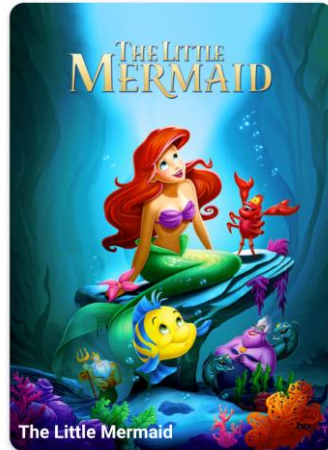


Το πρώτο ζευγάρι έκανε μία επιλογή ταινίας επιστημονικής φαντασίας. Οπότε δεν προκαλεί εντύπωση το γεγονός πως οι πρώτες 4 ταινίες με το μεγαλύτερο σκορ εμπίπτουν κι αυτές στην ίδια κατηγορία. Για την ακρίβεια, ιδιαίτερο ενδιαφέρον παρουσιάζει το γεγονός πως 3 από τις 4 προτάσεις είναι άμεσα σχετιζόμενες ταινίες με αυτήν που επιλέχθηκε, αποτελώντας είτε sequel είτε prequel αυτής.

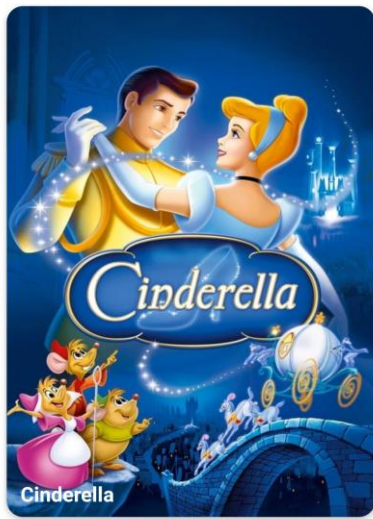
Για το δεύτερο ζευγάρι χρηστών όπου επιλέχθηκε η ταινία **Beauty and the Beast**, οι 4 πρώτες προτάσεις ήταν:



Aladdin



The Little Mermaid



Cinderella



The Lion King



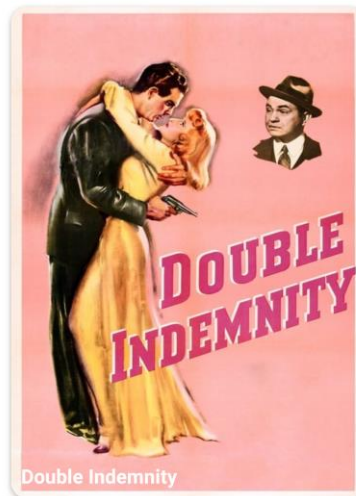
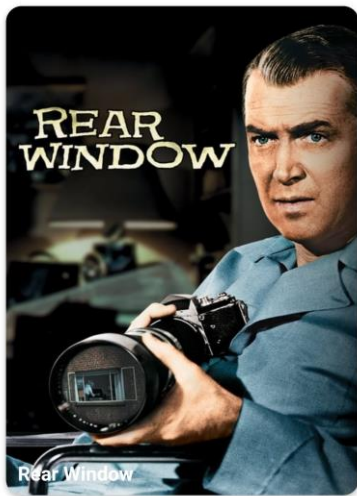
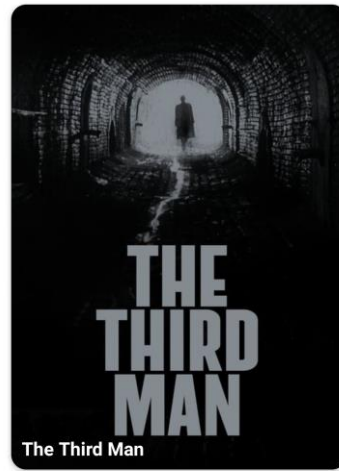
Αντίστοιχα, στην περίπτωση αυτή, η επιλεχθείσα ταινία ήταν ταινία κινουμένων σχεδίων, οπότε και το σύστημα πρότεινε τέτοιες ταινίες. Αυτό που παρουσιάζει ενδιαφέρον σε αυτήν την περίπτωση είναι που, στην περίπτωση του Beauty and The Beast, η ιστορία περιτριγυρίζεται γύρω από μία από τις πριγκίπισσες τις Disney, όπως συμβαίνει και με τις 3 πρώτες προτάσεις! Η 4^η πρόταση, αν και ταινία κινουμένων σχεδίων, δεν έχει τέτοια θεματολογία, γι' αυτό και παρουσιάζεται πιο κάτω στην κατάταξη.

5.1.2 Συνεδρίες με βάση παλιές βαθμολογίες

Τα αποτελέσματα που παρουσιάζονται παρακάτω αποτελούν τις 4 πρώτες ταινίες από τις 50 που πρότεινε το RS για τον αντίστοιχο συνδυασμό από είδη που επέλεξε το κάθε ζευγάρι. Επιπλέον, κάθε ζευγάρι για το οποίο έγιναν οι προτάσεις, είχε βαθμολογήσει μόνο τις ταινίες από την οθόνη αρχικοποιήσεις προφίλ του χρήστη.

Σε αυτόν τον τύπο συνεδρία οι προτάσεις προήλθαν με βάση το μοντέλο που εκπαιδεύτηκε με τον αλγόριθμο SVD. Στην περίπτωση αυτή, σημαντικό ρόλο παίζουν οι βαθμολογίες των χρηστών για την εκάστοτε ταινία κι όχι η ταινία αυτή καθ' αυτή. Αρχικά, το σύστημα φιλτράρει τις ταινίες σε δύο στάδια: πρώτα βρίσκει τις ταινίες αυτές που εμπίπτουν στο είδος που επέλεξε ο χρήστης που ξεκίνησε τη συνεδρία και στη συνέχεια από τη λίστα αυτήν αφαιρεί τις ταινίες οι χρήστες έχουν ήδη παρακολουθήσει. Στη συνέχεια, λαμβάνοντας υπόψιν τις βαθμολογίες που έχουν κάνει μέχρι στιγμής οι δύο χρήστες, προβλέπει τις βαθμολογίες των χρηστών για κάθε μία ταινία στη λίστα. Τέλος, με βάση τη multiplicative στρατηγική συνάθροισης, βρίσκει τις 50 καλύτερες και για τους δύο.

Για το πρώτο ζευγάρι χρηστών όπου επιλέχθηκαν τα είδη **Θρίλερ** ή **Τρόμου**, οι 4 πρώτες προτάσεις ήταν:



Στην περίπτωση αυτή οι χρήστες επέλεξαν το είδος να είναι θρίλερ ή τρόμου και πράγματι, οι ταινίες που το σύστημα τους πρότεινε εμπίπτουν στις κατηγορίες αυτές.

Για το δεύτερο ζευγάρι χρηστών όπου επιλέχθηκε το είδος **Επιστημονικής φαντασίας**, οι 4 πρώτες προτάσεις ήταν:



Λόγο του γεγονότος πως σε αυτήν την περίπτωση έγινε επιλογή του είδους επιστημονικής φαντασίας, είναι εφικτή μία σύγκριση με τα αποτελέσματα της πρώτης συνεδρίας. Αυτό που φαίνεται είναι πως παρόλο που και στις δύο περιπτώσεις το είδος ήταν το ίδιο, τα αποτελέσματα ήταν πολύ διαφορετικά. Αυτό οφείλεται ότι στη δεύτερη συνεδρία λαμβάνονται υπόψιν και οι βαθμολογίες των ίδιων των χρηστών, σε αντίθεση με την πρώτη που η επιλογή γίνεται με βάση την ταινία και μόνο.

5.2 Συμπεράσματα αξιολόγησης

Τα σχόλια που συλλέχθηκαν από τις δύο ζευγάρια μετά την χρήση της εφαρμογής ήταν ικανοποιητικά καθώς τα δύο ζευγάρια κατάφεραν να βρουν αρκετές ενδιαφέρουσες προτάσεις όπου σε αρκετές από αυτές συμφώνησαν και οι δύο πως φαίνονται ενδιαφέροντες να παρακολουθήσουν. Ωστόσο υπήρξαν και σχόλια και από τα δύο ζευγάρια, πως πολλές από τις ταινίες που τους πρότεινε η εφαρμογή ήταν αρκετά παλιές σε χρονολογία. Αυτό ωστόσο ήταν κάτι αναμενόμενο μιας και οι διαθέσιμες ταινίες ήταν περιορισμένες από το dataset movielens. Επιπλέον ανέφεραν πως θα τους άρεσε να έχει περισσότερες πληροφορίες όπως κάποιο trailer, ώστε να είναι πιο εύκολο να αποφασίσουν αν μια ταινία τους ενδιαφέρει ή όχι. Αυτό ήταν κάτι που αρχικά ήταν στα σχέδια της εφαρμογής, ωστόσο παρατηρήθηκε πως για μεγάλο πλήθος ταινιών δεν ήταν δυνατή η εύρεση trailer, οπότε, προκειμένου να υπάρξει μία συνέπεια, τελικά δεν συμπεριλήφθηκαν.

6. Συμπεράσματα, Περιορισμοί & Μελλοντικές Επεκτάσεις

6.1 Συμπεράσματα

Μέσω της εκπόνησης της πτυχιακής έγινε αρκετά φανερός ο ρόλος που έχουν τα recommendation systems στην εμπειρία των χρηστών σε περιπτώσεις που καλούνται να επιλέξουν μέσα από μία μεγάλη λίστα επιλογών. Κατά την ανάπτυξη του συγκεκριμένου συστήματος έγινε αντιληπτός ο τρόπος λειτουργίας ενός RS και οι διάφορες τεχνικές που χρησιμοποιούνται προκειμένου να παράγουν εξατομικευμένες προτάσεις για τους χρήστες. Έγινε ιδιαίτερα αντιληπτή η συνεισφορά τους στην ανακάλυψη νέων προϊόντων τα οποία οι χρήστες πιθανόν δε θα είχαν βρει με προσωπική αναζήτηση. Το σύστημα, παρότι όχι τέλειο, ήταν σε θέση να ανταποκριθεί στις απαιτήσεις των χρηστών και να παράξει σχετικά καλές προτάσεις και με τους δύο μηχανισμούς που αξιοποιεί. Όπως, όμως, έγινε εξίσου αντιληπτό από τα σχόλια των χρηστών που δοκίμασαν την εφαρμογή, στην περίπτωση των προτάσεων ταινιών, σημαντικό ρόλο έχει και η παλαιότητα των ταινιών, κάτι στο οποίο η εφαρμογή παρουσίαζε έναν ιδιαίτερο περιορισμό.

Εκτός από το σύστημα προτάσεων έγινε εξίσου αντιληπτές και οι δυσκολίες που μπορεί να συναντήσει κάποιος κατά την ανάπτυξη εφαρμογών για κινητές συσκευές, όπου διάφορες παράμετροι πρέπει να ληφθούν υπόψιν.

6.2 Περιορισμοί

Καθ' όλη την διάρκεια της εκπόνησης της πτυχιακής παρατηρηθήκαν μερικοί περιορισμοί που αναλύονται παρακάτω. Αρχικά, όπως αναφέρεται και στο Κεφάλαιο 2, για να μπορέσουμε να ξεπεράσουμε το πρόβλημα του Cold start έπρεπε να χρησιμοποιηθεί το dataset movielens το οποίο περιόρισε σε μεγάλο βαθμό στο πλήθος των ταινιών που ήταν διαθέσιμες στην εφαρμογή. Αυτό περιόρισε και τις προτάσεις που εμφανίζονται στους χρήστες, καθώς και το πλήθος των metadata που συγκεντρώθηκαν για τις ταινίες μιας και πρόκειται για ταινίες αρκετά παλιάς χρονολογίας.

Επιπλέον δεν ήταν δυνατή η διόρθωση ενός bug το οποίο προκύπτει όταν ο χρήστης βαθμολογεί πολύ γρήγορα τις ταινίες, με αποτέλεσμα κάποια ταινία να εμφανιστεί δύο

φορές. Επίσης δεν έχει ληφθεί υπόψιν η περίπτωση κατά την οποία θα λαμβάνονται πολλά ίδια request από τον χρήστη, το οποίο θα οδηγήσει σε crash της εφαρμογής.

6.3 Μελλοντικές επέκτασης

Κάποιες μελλοντικές επεκτάσεις που μπορούν να υλοποιηθούν είναι:

- Υλοποίηση συνεδρίας για παραπάνω άτομα

Οι κατάλληλες μετατροπές στο κομμάτι της εφαρμογής θα μπορούσαν να προσθέσουν την δυνατότητα της δημιουργίας συνεδριών για παραπάνω από δύο άτομα.

- Παραγωγή προτάσεων με τεχνική βασισμένη στο περιεχόμενο (Content-based)

Η δημιουργία ενός τρίτου τύπου συνεδρίας το οποίο θα ήταν σε θέση να κάνει προτάσεις με βάση τα metadata που έχουμε για τις ταινίες.

Βιβλιογραφία

- Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40, 66-72.
- Desrosiers, C., & Karypis, G. (2010). A Comprehensive Survey of Neighborhood-based Recommendation Methods. Στο *Recommender Systems Handbook* (σσ. 107-144). Springer US. Ανάκτηση από http://dx.doi.org/10.1007/978-0-387-85820-3_4
- Elahi, M., Ricci, F., & Rubens, N. (2016). A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 29-50. doi:10.1016/j.cosrev.2016.05.002
- Falk, K. (2019). *Practical Recommender Systems*. Simon and Schuster.
- Funk, S. (2006). Netflix Update: Try This at Home. Ανάκτηση από <https://sifter.org/~simon/journal/20061211.html>
- Isinkaye, F., Folajimi, Y., & Ojokoh, B. (2015). *Recommendation systems: Principles, methods and evaluation*. Elsevier BV. doi:10.1016/j.eij.2015.06.005
- Jameson, A., & Smyth, B. (2007). Recommendation to Groups. Στο *The Adaptive Web* (σσ. 596-627). Springer Berlin Heidelberg. Ανάκτηση από http://dx.doi.org/10.1007/978-3-540-72079-9_20
- Koren, Y., & Bell, R. (2015). Advances in Collaborative Filtering. Στο *Recommender Systems Handbook* (σσ. 77-118). Springer US. Ανάκτηση από http://dx.doi.org/10.1007/978-1-4899-7637-6_3
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, 30-37. doi:10.1109/mc.2009.263
- Krouska, A., Troussas, C., & Sgouropoulou, C. (2020). Applying Genetic Algorithms for Recommending Adequate Competitors in Mobile Game-Based Learning Environments. Στο *Intelligent Tutoring Systems* (σσ. 196-204). Springer International Publishing. Ανάκτηση από http://dx.doi.org/10.1007/978-3-030-49663-0_23
- Liu, J.-H., Zhou, T., Zhang, Z.-K., Yang, Z., Liu, C., & Li, W.-M. (2014). Promoting Cold-Start Items in Recommender Systems. *PLoS ONE*(12), e113457. doi:10.1371/journal.pone.0113457

- Masthoff, J. (2010). Group Recommender Systems: Combining Individual Models. Στο *Recommender Systems Handbook* (σσ. 677-702). Springer US. Ανάκτηση από http://dx.doi.org/10.1007/978-0-387-85820-3_21
- Mathew, P., Kuriakose, B., & Hegde, V. (2016). Book Recommendation System through content based and collaborative filtering method. *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*. doi:10.1109/sapience.2016.7684166
- Ricci, F., Rokach, L., & Shapira, B. (2010). *Introduction to Recommender Systems Handbook*. Springer US. doi:http://dx.doi.org/10.1007/978-0-387-85820-3_1
- Su, X., & Khoshgoftaar, T. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 1-19. doi:10.1155/2009/421425
- Troussas, C., Krouska, A., & Virvou, M. (2018). Multi-Algorithmic Techniques and a Hybrid Model for Increasing the Efficiency of Recommender Systems. *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, 184-188. doi:10.1109/ictai.2018.00037
- Tsolakidis, A., Triperina, E., Sgouropoulou, C., & Christidis, N. (2016). Research Publication Recommendation System based on a Hybrid Approach. *Proceedings of the 20th Pan-Hellenic Conference on Informatics*, 1-6. doi:10.1145/3003733.3003805
- Zhongqi, L. (2015). Content-based collaborative filtering for news topic recommendation. Στο *Twenty-ninth AAAI conference on artificial intelligence*.
- Christos Troussas, Akrivi Krouska, Cleo Sgouropoulou: Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education, *Computers & Education*, Volume 144, 2020, 103698, <https://doi.org/10.1016/j.compedu.2019.103698>.
- Troussas C, Krouska A, Sgouropoulou C, Voyiatzis I. Ensemble Learning Using Fuzzy Weights to Improve Learning Style Identification for Adapted Instructional Routines. *Entropy*. 2020; 22(7):735. <https://doi.org/10.3390/e22070735>.
- Krouska A., Troussas C., Sgouropoulou C. (2020) A Personalized Brain-Based Quiz Game for Improving Students' Cognitive Functions. In: Frasson C., Bamidis P., Vlamos P. (eds) *Brain Function Assessment in Learning*. BFAL 2020. Lecture Notes in Computer Science, vol 12462. Springer, Cham. https://doi.org/10.1007/978-3-030-60735-7_11

Papakostas, C., Troussas, C., Krouska, A. et al. User acceptance of augmented reality welding simulator in engineering training. *Educ Inf Technol* (2021). <https://doi.org/10.1007/s10639-020-10418-7>.

Troussas, C., Krouska, A. & Sgouropoulou, C. Impact of social networking for advancing learners' knowledge in E-learning environments. *Educ Inf Technol* 26, 4285–4305 (2021). <https://doi.org/10.1007/s10639-021-10483-6>.

Krouska A., Troussas C., Sgouropoulou C. (2020) Applying Genetic Algorithms for Recommending Adequate Competitors in Mobile Game-Based Learning Environments. In: Kumar V., Troussas C. (eds) *Intelligent Tutoring Systems. ITS 2020. Lecture Notes in Computer Science*, vol 12149. Springer, Cham. https://doi.org/10.1007/978-3-030-49663-0_23

Christos Troussas, Cleo Sgouropoulou: Innovative Trends in Personalized Software Engineering and Information Systems - The Case of Intelligent and Adaptive E-learning Systems. *Frontiers in Artificial Intelligence and Applications* 324, IOS Press 2020, ISBN 978-1-64368-096-5, pp. 1-96

Troussas C., Krouska A., Sgouropoulou C. (2020) Towards a Reference Model to Ensure the Quality of Massive Open Online Courses and E-Learning. In: Frasson C., Bamidis P., Vlamos P. (eds) *Brain Function Assessment in Learning. BFAL 2020. Lecture Notes in Computer Science*, vol 12462. Springer, Cham. https://doi.org/10.1007/978-3-030-60735-7_18

Troussas C., Krouska A., Sgouropoulou C. (2020) Dynamic Detection of Learning Modalities Using Fuzzy Logic in Students' Interaction Activities. In: Kumar V., Troussas C. (eds) *Intelligent Tutoring Systems. ITS 2020. Lecture Notes in Computer Science*, vol 12149. Springer, Cham. https://doi.org/10.1007/978-3-030-49663-0_24

Christos Troussas, Akrivi Krouska, Filippos Giannakas, Cleo Sgouropoulou, Ioannis Voyiatzis: Automated reasoning of learners' cognitive states using classification analysis. *PCI 2020-24th Pan-Hellenic Conference on Informatics, November 2020, Pages 103–106*. <https://doi.org/10.1145/3437120.3437285>

Christos Troussas, Akrivi Krouska, Filippos Giannakas, Cleo Sgouropoulou, Ioannis Voyiatzis: Redesigning teaching strategies through an information filtering system. *PCI 2020-24th Pan-Hellenic Conference on Informatics, November 2020, Pages 111-114*. <https://doi.org/10.1145/3437120.3437287>

Akrivi Krouska, Christos Troussas, Cleo Sgouropoulou: Usability and Educational Affordance of Web 2.0 tools from Teachers' Perspectives. PCI 2020-24th Pan-Hellenic Conference on Informatics, November 2020, Pages 107-110. <https://doi.org/10.1145/3437120.3437286>

Christos Troussas, Filippos Giannakas, Cleo Sgouropoulou & Ioannis Voyiatzis (2020) Collaborative activities recommendation based on students' collaborative learning styles using ANN and WSM, Interactive Learning Environments, DOI: 10.1080/10494820.2020.1761835

F. Giannakas, C. Troussas, I. Voyiatzis, C. Sgouropoulou, A deep learning classification framework for early prediction of team-based academic performance, Applied Soft Computing, Volume 106, 2021, 107355. <https://doi.org/10.1016/j.asoc.2021.107355>.

Christos PAPAKOSTAS, Christos TROUSSAS, Akrivi KROUSKA, Cleo SGOUROPOULOU, Exploration of Augmented Reality in Spatial Abilities Training: A Systematic Literature Review for the Last Decade, Informatics in Education 20 (2021), no. 1, 107-130, DOI 10.15388/infedu.2021.06

C. Troussas, A. Krouska and C. Sgouropoulou, "A Novel Teaching Strategy Through Adaptive Learning Activities for Computer Programming," in IEEE Transactions on Education, vol. 64, no. 2, pp. 103-109, May 2021, doi: 10.1109/TE.2020.3012744.

Krouska, A., Troussas, C. and Sgouropoulou, C. 2019. Fuzzy Logic for Refining the Evaluation of Learners' Performance in Online Engineering Education. European Journal of Engineering and Technology Research. 4, 6 (Jun. 2019), 50-56. DOI:<https://doi.org/10.24018/ejers.2019.4.6.1369>.

Troussas C, Krouska A, Sgouropoulou C. Improving Learner-Computer Interaction through Intelligent Learning Material Delivery Using Instructional Design Modeling. Entropy. 2021; 23(6):668. <https://doi.org/10.3390/e23060668>

Papakostas C, Troussas C, Krouska A, Sgouropoulou C. Measuring User Experience, Usability and Interactivity of a Personalized Mobile Augmented Reality Training System. Sensors. 2021; 21(11):3888. <https://doi.org/10.3390/s21113888>

Giannakas F., Troussas C., Krouska A., Sgouropoulou C., Voyiatzis I. (2021) XGBoost and Deep Neural Network Comparison: The Case of Teams' Performance. In: Cristea A.I., Troussas C. (eds) Intelligent Tutoring Systems. ITS 2021. Lecture Notes in Computer Science, vol 12677. Springer, Cham. https://doi.org/10.1007/978-3-030-80421-3_37

Krouska, A., Troussas, C. & Sgouropoulou, C. Mobile game-based learning as a solution in COVID-19 era: Modeling the pedagogical affordance and student interactions. *Educ Inf Technol* (2021). <https://doi.org/10.1007/s10639-021-10672-3>

C. Troussas, A. Krouska, C. Sgouropoulou, "Enhancing Human-Computer Interaction in Digital Repositories through a MCDA-Based Recommender System", *Advances in Human-Computer Interaction*, vol. 2021, Article ID 7213246, 7 pages, 2021. <https://doi.org/10.1155/2021/7213246>

Krouska A, Troussas C, Sgouropoulou C. A Cognitive Diagnostic Module Based on the Repair Theory for a Personalized User Experience in E-Learning Software. *Computers*. 2021; 10(11):140. <https://doi.org/10.3390/computers10110140>

Troussas, C., Krouska, A. and Sgouropoulou, C., A User-centric System for Improving Human-Computer Interaction through Fuzzy Logic-based Assistive Messages, *Proceedings of the 17th International Conference on Web Information Systems and Technologies (WEBIST 2021)*, pages 365-370. DOI: 10.5220/0010702800003058.

Akrivi KROUSKA, Christos TROUSSAS, Filippos GIANNAKAS, Cleo SGOUROPOULOU, and Ioannis VOYIATZIS, Enhancing the Effectiveness of Intelligent Tutoring Systems Using Adaptation and Cognitive Diagnosis Modeling, *Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021)*, Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 40-45). IOS Press, doi:10.3233/FAIA210073.

Kanetaki, Z., Stergiou, C., Bekas, G., Troussas, C., & Sgouropoulou, C. (2021). Data Mining for Improving Online Higher Education Amidst COVID-19 Pandemic: A Case Study in the Assessment of Engineering Students. *Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021)*, Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 157-165). doi:10.3233/FAIA210088.

Kanetaki, Z., Stergiou, C., Troussas, C., & Sgouropoulou, C. (2021, October). Development of an Innovative Learning Methodology Aiming to Optimise Learners' Spatial Conception in an Online Mechanical CAD Module During COVID-19 Pandemic. In *Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021)*, Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 31-39). IOS Press. doi:10.3233/FAIA210072.

Kanetaki, Z., Stergiou, C., Bekas, G., Troussas, C., & Sgouropoulou, C. (2021). Creating a Metamodel for Predicting Learners' Satisfaction by Utilizing an Educational Information System During COVID-19 Pandemic. In *Novelties in Intelligent Digital Systems: Proceedings of*

the 1st International Conference (NIDS 2021), Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 127-136). IOS Press. doi:10.3233/FAIA210085.

Kapetanaki, A., Krouska, A., Troussas, C., & Sgouropoulou, C. (2021). A Novel Framework Incorporating Augmented Reality and Pedagogy for Improving Reading Comprehension in Special Education. In *Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021)*, Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 105-110). IOS Press. doi:10.3233/FAIA210081.

C. Papakostas, C. Troussas, A. Krouska, C. Sgouropoulou (2021, October). On the Development of a Personalized Augmented Reality Spatial Ability Training Mobile Application. In *Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021)*, Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 75-83). IOS Press. doi:10.3233/FAIA210078.

Kontellis, E., Troussas, C., Krouska, A., & Sgouropoulou, C. (2021). Real-time face mask detector using convolutional neural networks amidst COVID-19 pandemic. In *Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021)*, Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 247-255). IOS Press. doi:10.3233/FAIA210102

Troussas, C., Krouska, A., Giannakas, F., Sgouropoulou, C., & Voyiatzis, I. (2021). An Alternative Educational Tool Through Interactive Software over Facebook in the Era of COVID-19. In *Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021)*, Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 3-11). IOS Press. doi:10.3233/FAIA210069.

Troussas C., Krouska A., Giannakas F., Sgouropoulou C., Voyiatzis I. (2021) Representation of Generalized Human Cognitive Abilities in a Sophisticated Student Leaderboard. In: Cristea A.I., Troussas C. (eds) *Intelligent Tutoring Systems. ITS 2021. Lecture Notes in Computer Science*, vol 12677. Springer, Cham. https://doi.org/10.1007/978-3-030-80421-3_44

A. Marougkas, C. Troussas, A. Krouska, C. Sgouropoulou (2021, October). A Framework for Personalized Fully Immersive Virtual Reality Learning Environments with Gamified Design in Education. In *Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021)*, Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 95-104). IOS Press. doi:10.3233/FAIA210080.

Kanetaki Z, Stergiou C, Bekas G, Troussas C, Sgouropoulou C. Evaluating Remote Task Assignment of an Online Engineering Module through Data Mining in a Virtual Communication

Platform Environment. Electronics. 2022; 11(1):158.

<https://doi.org/10.3390/electronics11010158>

Kanetaki, Z., Stergiou, C., Bekas, G., Troussas, C., & Sgouropoulou, C. (2021). Analysis of Engineering Student Data in Online Higher Education During the COVID-19 Pandemic. *International Journal of Engineering Pedagogy (IJEP)*, 11(6), pp. 27–49. <https://doi.org/10.3991/ijep.v11i6.23259>.

Kanetaki, Z.; Stergiou, C.; Bekas, G.; Troussas, C.; Sgouropoulou, C. The Impact of Different Learning Approaches Based on MS Teams and Moodle on Students' Performance in an on-Line Mechanical CAD Module. *Global Journal of Engineering Education (GJEE)*. 2021, 23, 185–190.