ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΣΥΣΤΑΣΕΩΝ ΒΑΣΙΣΜΕΝΟΥ ΣΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΜΕ ΓΛΩΣΣΑ PYTHON

Διπλωματική Εργασία

Κοσμάς Γλαβάς

Γεώργιος Πράπας

Επιβλέπων: Βασιλακόπουλος Μιχαήλ

Βόλος 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΣΥΣΤΑΣΕΩΝ ΒΑΣΙΣΜΕΝΟΥ ΣΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΜΕ ΓΛΩΣΣΑ PYTHON

Διπλωματική Εργασία

Κοσμάς Γλαβάς

Γεώργιος Πράπας

Επιβλέπων: Βασιλακόπουλος Μιχαήλ

Βόλος 2021

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

DEVELOPMENT OF A CONTENT-BASED RECOMMENDER SYSTEM WITH PYTHON

Diploma Thesis

Kosmas Glavas

Georgios Prapas

Supervisor: Michael Vassilakopoulos

Volos 2021

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων **Βασιλακόπουλος Μιχαήλ**
Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Τουσίδου Ελένη**
μέλος ΕΔΙΠ, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Τσαλαπάτα Χαρίκλεια**
μέλος ΕΔΙΠ, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: 15-2-2021

Ευχαριστίες

Με την παρούσα διπλωματική εργασία ολοκληρώνονται οι σπουδές μας στο τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών . Επιθυμούμε να ευχαριστήσουμε τον καθηγητή και επιβλέποντα μας για την παρούσα διπλωματική εργασία, κο Μιχαήλ Βασιλακόπουλο, για την επιστημονική και συμβουλευτική καθοδήγηση που μας προσέφερε σε όλα τα στάδια εκπόνησης της εργασίας με τις εύστοχες και πολύ εποικοδομητικές παρατηρήσεις του. Τέλος, οφείλουμε να ευχαριστήσουμε τις οικογένειές μας, για τη συμπαράσταση και την υπομονή τους.

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

«Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής».

Οι Δηλώντες

Κοσμάς Γλαβάς και Γεώργιος Πράπας

14-2-2021

Περίληψη

Τα συστήματα συστάσεων είναι ένα εργαλείο που βοηθά τους χρήστες να βρουν περιεχόμενο. Προβλέπουν τα ενδιαφέροντα των χρηστών και κάνουν προτάσεις σύμφωνα με τις προτιμήσεις τους. Είναι από τα πιο ισχυρά συστήματα μηχανικής μάθησης που εφαρμόζονται ευρέως, με ενδεικτικό παράδειγμα τους διαδικτυακούς πωλητές με σκοπό να προωθήσουν τις πωλήσεις τους. Τα δεδομένα που απαιτούνται για τα Συστήματα Συστάσεων πηγάζουν από ρητές αξιολογήσεις χρηστών, όπως μετά από την παρακολούθηση μιας ταινίας ή την ακρόαση ενός τραγουδιού, έμμεσα από αναζητήσεις στις μηχανές αναζήτησης καθώς και το ιστορικό των αγορών ή από άλλες πληροφορίες σχετικά με τους ίδιους τους χρήστες και τα αντικείμενα που αυτοί επιλέγουν. Ουσιαστικά τα συστήματα συστάσεων βοηθούν τους χρήστες να ξεπεράσουν τον μεγάλο όγκο των πληροφοριών του διαδικτύου. Σε αυτό το πρόβλημα εντάσσεται και η επιλογή ταινίας για παρακολούθηση. Αυτό συμβαίνει διότι κάθε χρόνο υπάρχουν πάρα πολλές παραγωγές ταινίων και τηλεοπτικών σειρών με αποτέλεσμα να υπάρχει ένας τεράστιος όγκος πληροφοριών. Αφού παρατηρήσαμε αυτό το πρόβλημα δημιουργήσαμε έναν ιστότοπο που μπορείς να αναζητήσεις και να βρεις ταινίες χωρίς να χρειάζεται να σπαταλάς πολύ χρόνο. Ο σκοπός του ιστότοπου μας είναι να κάνει ελκυστικές προτάσεις στους χρήστες χωρίς αυτές να είναι πάντα οι αναμενόμενες. Για να γίνονται οι προτάσεις στους χρήστες δημιουργήσαμε ένα σύστημα συστάσεων που βασίζεται στο περιεχόμενο και υπολογίζει την ομοιότητα με την ομοιότητα συνημιτόνου για τα πιο σημαντικά στοιχεία της κάθε ταινίας. Τα στοιχεία που αξιοποιεί ο αλγόριθμος είναι οι ηθοποιοί, ο σκηνοθέτης, το είδος, τις λέξεις-κλειδιά και την εταιρία παραγωγής της ταινίας. Οι συστάσεις του συστήματος που δημιουργήσαμε δεν είναι προσωποποιημένες, δηλαδή δεν είναι διαφορετικές για κάθε χρήστη. Όμως αναλύοντας τις ταινίες με τον τρόπο που επιλέξαμε γίνονται πολύ εύστοχες και ενδιαφέρουσες συστάσεις που μπορούν πραγματικά να βοηθήσουν τους χρήστες του ιστότοπου να επιλέξουν μια ταινία που θα τους αρέσει. Κάνοντας χρήση της γλώσσας προγραμματισμού Python, της πλατφόρμας Django και του IMDB API η εφαρμογή

μας είναι σε θέση να παρέχει συστάσεις στο χρήστη παίρνοντας ως είσοδο μια ταινία από αυτόν. Ακόμα υπάρχουν αρκετές λίστες με ταινίας κάθε κατηγορίας για να διαλέξει ο χρήστης. Επίσης όταν αναζητά μια ταινία μπορεί να τη βαθμολογήσει και να ανταλλάξει απόψεις με άλλους χρήστες. Τέλος αν κάποιος χρήστης δεν μπορεί να βρει μια ταινία τις αρεσκίας του μπορεί να κοιτάξει προφίλ άλλων χρηστών που θεωρεί ότι έχουν παρόμοιες προτιμήσεις και να δεί ποιες ταινίες έχουν πρακαλουθήσει και έχουν δηλώσει ότι τους άρεσαν.

Abstract

Recommender systems are a tool that helps users find content and overcome a large amount of information. It anticipates the interests of the users and makes suggestions according to their preferences. They are among the most powerful machine learning systems widely used, as for example the online retailers who aim to promote their sales. The data required for recommendation systems comes from explicit user ratings, such as after watching a movie or listening to a song, indirectly from search engine searches and market history or other information about the users themselves, and the objects they choose. Recommender systems help users overcome the vast amount of information on the internet. Choosing a movie to watch is part of this problem. This is because every year there are too many productions of movies and TV series resulting in a huge amount of information. After noticing this problem, we created a website where you can search and find movies without wasting a lot of time. The purpose of our site is to make attractive suggestions to users without them always being what is expected. To make suggestions to users we created a Content base algorithm that calculates the similarity with cosine similarity for the most important features of each movie. The features used by the algorithm are the actors, the director, the genre, the keywords and the production company of the film. The recommendations of the system we created are not personalized, so they are not different for each user. But by analyzing the movies in the way we have chosen, the recommendations become really interesting and they can help the users of our site to choose a film that they will like. Using the Python programming language, the Django platform and the IMDB API, our application is able to provide recommendations to the user by taking an input movie from him. There are several movie lists of each genre for the user to choose from. Also when searching for a movie he can rate it and exchange views with other users. Finally, if users can not find a movie they like, they can look at the profiles of other users they think they have similar preferences and see which movies they have watched and liked.

Πίνακας περιεχομένων

Κατάλογος σχημάτων

Συντομογραφίες

κ.λπ.	και λοιπά
κ.α.	και άλλα

Κεφάλαιο 1

Εισαγωγή

Με την ανάπτυξη του διαδικτύου και των τεχνολογιών της πληροφορίας έχουμε εισέλθει σε μια εποχή υπερφορτωμένη από πληροφορίες. Είναι πλέον πολύ δύσκολο οι χρήστες να προσκομίσουν τις πληροφορίες τις οποίες θέλουν. Αντίστοιχα και για αυτούς που παρέχουν τις πληροφορίες, είναι πολύ δύσκολο να τις κάνουν να ξεχωρίσουν στο μεγάλο σύνολο. Το πρόβλημα αυτό λύνεται με την ανάπτυξη συστημάτων συστάσεων (Recommender Systems). Ο σκοπός των συστημάτων συστάσεων είναι να ενώσει τους ανθρώπους με την πληροφορία. Αυτό βοηθάει τόσο τους χρήστες που ψάχνουν για συγκρεκρίμενες πληροφορίες όσο και αυτούς που προωθούν τις πληροφορίες.

1.1 Αντικείμενο της διπλωματικής

Κάθε χρόνο υπάρχουν πάρα πολλές παραγωγές ταινίων και τηλεοπτικών σειρών με αποτέλεσμα να υπάρχει ένας τεράστιος όγκος πληροφοριών. Είναι πολύ δύσκολο να θύμασαι πληροφορίες για τις ταινίες που έχεις παρακολουθήσει στο παρελθόν και ακόμα πιο δύσκολο να θυμάσαι το έργο των αγαπημένων σου ηθοποιών-σκηνοθετών. Ακόμα, εξαιτίας της υπερφόρτωσης του κλάδου, οι χρήστες σπαταλάνε πολλές ώρες για να βρουν μια ταινία-σείρα που να θέλουν να παρακολουθήσουν. Αφού παρατηρήσαμε αυτό το πρόβλημα δημιουργήσαμε έναν ιστότοπο που μπορείς να αναζητήσεις όλες τις παραγωγές που υπάρχουν και να βρεις στοιχεία όπως ο σκηνοθέτης, ο παραγωγός, οι ηθοποιοί που πρωταγωνιστούν, η υπόθεση κ.α. . Επιπλέον στην ιστοσελίδα μας υπάρχει μια ξεχωριστή αναζήτηση που σου προτείνει ταινίες παρόμοιες με την ταινία που έβαλες στην αναζήτηση. Έτσι ο χρήστης μπορεί πλέον γρήγορα να βρίσκει ταινίες της αρεσκίας του χωρις να αναλώνεται σε άσκοπες και χρονοβό-

ρες αναζητήσεις.

1.2 Οργάνωση του τόμου

Η θεωρία σχετικά με τα συστήματα συστάσεων παρουσιάζεται στο Κεφάλαιο 2. Το Κεφάλαιο 3 αναφέρει τα μετρικά και τους τρόπους αξιολόγησης των συστημάτων προτάσεων. Το Κεφάλαιο 4 ασχολείται με το περιβάλλον εργασίας και τα εργαλεία που χρησιμοποιήσαμε. Το Κεφάλαιο 5 με τους αλγόριθμους συστάσεων που δοκιμάσαμε. Το Κεφάλαιο 6 περιγράφει αναλυτικά τον ιστότοπο της εργασίας και το Κεφάλαιο7 αναλύει τον κώδικα.

Κεφάλαιο 2

Θεωρία των συστημάτων συστάσεων

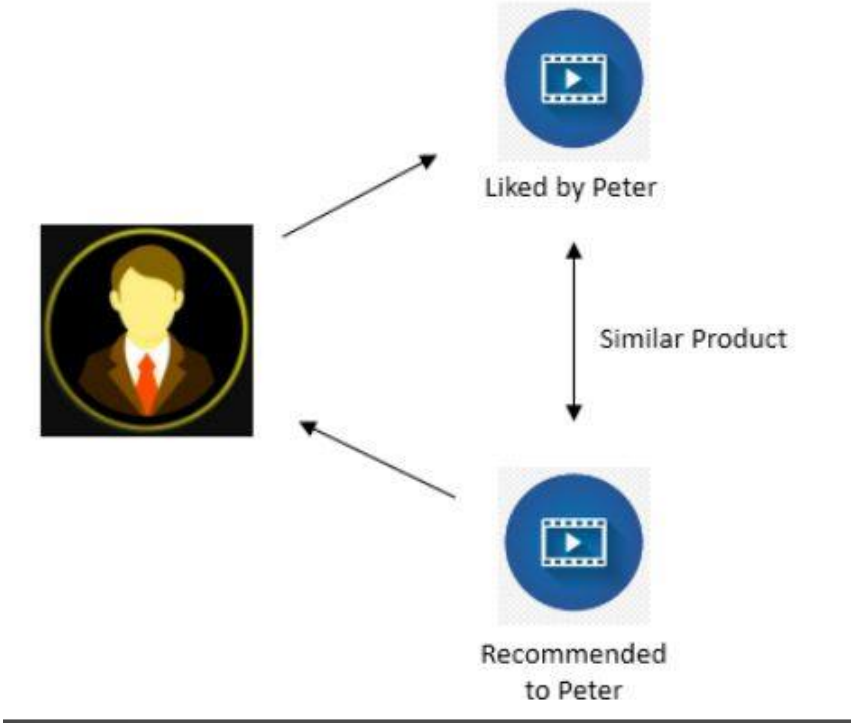
2.1 Εισαγωγή

Τα συστήματα προτάσεων είναι ένας ειδικός τύπος συστημάτων φιλτραρίσματος πληροφοριών. Είναι εφαρμογές λογισμικού που στοχεύουν στην υποστήριξη των χρηστών στη λήψη αποφάσεων, ενώ αλληλεπιδρούν με ένα τεράστιο όγκο πληροφοριών. Συνιστούν στοιχεία ενδιαφέροντος για τους χρήστες με βάση τις προτιμήσεις που έχουν εκφράσει, είτε ρητά είτε έμμεσα. Ο συνεχώς αυξανόμενος όγκος και η αυξανόμενη πολυπλοκότητα των πληροφοριών στον Ιστό έχουν καταστήσει επομένως τέτοια συστήματα απαραίτητα εργαλεία για χρήστες σε μια ποικιλία εφαρμογών, συνήθως συμπεριλαμβανομένων των δραστηριοτήτων αναζήτησης πληροφοριών ή ηλεκτρονικού εμπορίου. Συστήματα προτάσεων βοηθούν τους χρήστες να ξεπεράσουν το πρόβλημα υπερφόρτωσης πληροφοριών εκθέτοντάς τα στα πιο ενδιαφέροντα αντικείμενα και προσφέροντας καινοτομία, διορατικότητα και σχετικότητα. Η τεχνολογία προτάσεων είναι επομένως μια σημαντική λύση στο πρόβλημα αναζήτησης πληροφοριών που έχει προκύψει μαζί με το παγκόσμιο ιστό.

2.2 Συστήματα συστάσεων βασισμένα στο περιεχόμενο

Η βασική του ιδέα είναι να προτείνει στο χρήστη αντικείμενα τα οποία είναι όμοια με αντικείμενα που του είχαν αρέσει στο παρελθόν όπως φαίνεται και στο Σχήμα 2.1. Για παράδειγμα, εάν σε ένα χρήστη αρέσουν μόνο ταινίες δράσης, τότε το σύστημα του προτείνει μόνο ταινίες δράσης παρόμοιες με αυτήν που έχει υψηλή βαθμολογία. Η ευρύτερη εξήγηση είναι ότι αν σε ένα χρήστη αρέσει μόνο περιεχόμενο που σχετίζεται με την πολιτική, τότε

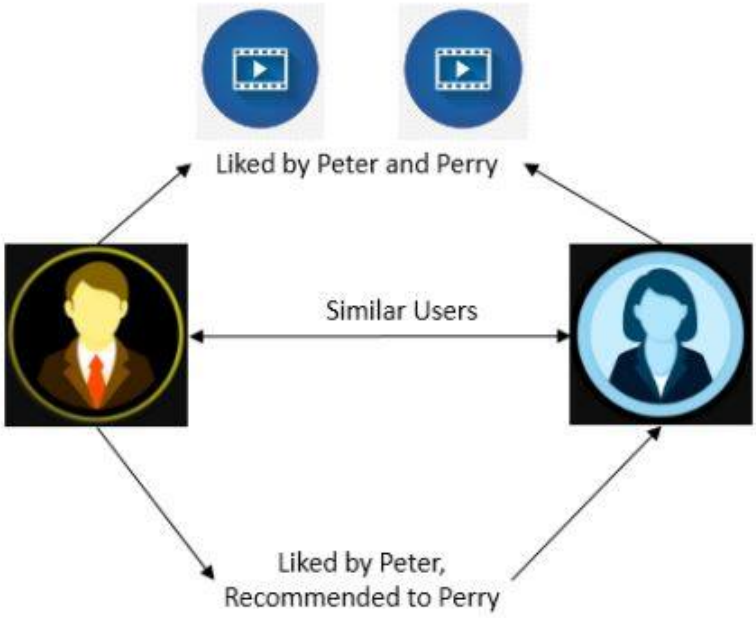
το σύστημα του προτείνει ιστότοπους, ιστολόγια ή νέα που αφορούν τα πολιτικά. Σε αντίθεση με τα συστήματα συστάσεων συνεργατικού φιλτραρίσματος, τα συστήματα βασισμένα στο περιεχόμενο δεν αντιμετωπίζουν πρόβλημα με τους νέους χρήστες. Υπάρχουν πολλές μέθοδοι για να μοντελοποιήσουμε το αντικείμενο και το το πιο διάσημο είναι το μοντέλο διανυσματικού χώρου. Όπως γίνεται αντιληπτό ο βασικός σκοπός του συστήματος αυτού είναι ο υπολογισμός της ομοιότητας μεταξύ των αντικειμένων. Υπάρχουν διάφορες τεχνικές και αλγόριθμοι που υπολογίζουν ομοιότητα όπως για παράδειγμα η Ευκλείδια απόσταση και η ομοιότητα συνημιτόνου. ’Ενα μεγάλο πλεονέκτημα της κατηγορίας αυτής είναι οτι το μοντέλο δεν χρειάζεται δεδομένα για άλλους χρήστες, καθώς οι προτάσεις είναι συγκεκριμένες για αυτόν τον χρήστη. Αυτό διευκολύνει την κλιμάκωση σε μεγάλο αριθμό χρηστών. Επιπρόσθετα, ένα σύστημα συστάσεων βασισμένο στο περιεχόμενο μπορεί να συλλάβει τα συγκεκριμένα ενδιαφέροντα ενός χρήστη και έτσι να είναι ικανό να προτείνει εξειδικευμένα αντικείμενα που ενδιαφέρονται πολύ λίγοι άλλοι χρήστες. Από την άλλη, για να βρεθεί κάποιο συγκεκριμένο χαρακτηριστικό, όπως εικόνες ή ταινίες ενός συγκεκριμένου είδους, μερικές φορές είναι πολύ δύσκολο και αναφέρεται ως το πρόβλημα υπερβολικής εξειδίκευσης. Μια βασική αδυναμία της κατηγορίας είναι οτι μπορεί να κάνει προτάσεις μόνο με βάση τα υπάρχοντα ενδιαφέροντα του χρήστη. Με άλλα λόγια, το μοντέλο έχει περιορισμένη ικανότητα επέκτασης στα υπάρχοντα ενδιαφέροντα των χρηστών. Γίνεται αντιληπτό λοιπόν ότι είναι εύκολο να παραλείψουμε να προτείνουμε κάποιο στοιχείο στον χρήστη, καθώς δεν υπάρχουν αρκετές πληροφορίες σχετικά με αυτό το στοιχείο. [12]



Σχήμα 2.1: Παράδειγμα για συστήματα βασισμένα στο περιεχόμενο [1]

2.3 Συστήματα συστάσεων συνεργατικού φιλτραρίσματος

Είναι η πιο διαδεδομένη κατηγορία συστάσεων. Το συνεργατικό φιλτράρισμα είναι η διαδικασία αξιολόγησης αντικειμένων χρησιμοποιώντας τις απόψεις άλλων ανθρώπων. Η λογική του είναι βασισμένη σε κάτι που κάνουν οι άνθρωποι εδώ και αιώνες, να μοιράζονται απόψεις με άλλους. Οι υπολογιστές και ο ιστός μας επιτρέπουν να προχωρούμε πέρα από το απλό στόμα σε στόμα. Αντί να περιοριστούμε σε δεκάδες ή εκατοντάδες άτομα, το διαδίκτυο μας επιτρέπει να εξετάσουμε τις απόψεις χιλιάδων ατόμων. Η ταχύτητα των υπολογιστών μας επιτρέπει να επεξεργαζόμαστε αυτές τις απόψεις σε πραγματικό χρόνο και να προσδιορίζουμε όχι μόνο τι σκέφτεται μια πολύ μεγαλύτερη κοινότητα για ένα αντικείμενο, αλλά και να αναπτύξουμε μια πραγματικά εξατομικευμένη άποψη αυτού του στοιχείου χρησιμοποιώντας τις απόψεις που είναι πιο κατάλληλες για έναν δεδομένο χρήστη ή ομάδα χρηστών. Στο Σχήμα 2.2 έχουμε μια αναπαράσταση του συνεργατικού φιλτραρίσματος. Υπάρχουν δύο μέθοδοι συνεργατικού φιλτραρίσματος η πρώτη είναι η μέθοδος με βάση τη μνήμη και η δεύτερη η μέθοδος με βάση το μοντέλο [13].



Σχήμα 2.2: Παράδειγμα συστήματος συστάσεων συνεργατικού φιλτραρίσματος [1]

2.3.1 Μέθοδος με βάση τη μνήμη

Αυτή η μέθοδος είναι επίσης γνωστή ως προσέγγιση που βασίζεται στη γειτονιά . Η μέθοδος που βασίζεται στη μνήμη χρησιμοποιεί μέτρα ομοιότητας που υπολογίζονται από τη ρητή βαθμολογία που δίνει ο χρήστης για την εύρεση γείτονα και τη δημιουργία προβλέψεων. Αυτός ο τύπος της μεθόδου βλέπει το ενδιαφέρον του χρήστη για οποιοδήποτε στοιχείο. Αφού αναλύσει την προβολή του χρήστη για ένα στοιχείο, ελέγχει ένα παρόμοιο χρήστη που έχει επίσης το ίδιο ενδιαφέρον με αυτόν. Έτσι, η εύρεση παρόμοιων χρηστών γίνεται με τη μελέτη ενός βοηθητικού πίνακα. Έτσι, αυτός ο τύπος προσέγγισης βασίζεται κυρίως στη μνήμη συστημάτων για την πρόβλεψη παρόμοιων χρηστών. Η προσέγγιση που βασίζεται στη μνήμη ταξινομείται περαιτέρω σε δύο τύπους. Προσέγγιση βάσει χρήστη και προσέγγιση βάσει αντικειμένων. Η προσέγγιση με βάση το χρήστη θεωρεί ότι σε ένα χρήστη θα άρεσει ένα αντικείμενο όταν αρέσει σε άλλους χρήστες που έχουν ίδιες προτιμήσεις.Οι χρήστες θεωρούνται όμοιοι όταν τους άρεσουν τα ίδια αντικείμενα. Αυτή η προσέγγιση είναι επίσης γνωστή ως φιλτράρισμα από χρήστη σε χρήστη. Σε αυτήν την τεχνική, δημιουργείται ένας πίνακας με βαθμολογίες ν χρηστών και κ αντικειμένων. Για να γίνει μια πρόταση για έναν νέο χρήστη, αυτή η προσέγγιση βρίσκει τον πλησιέστερο γείτονα χρησιμοποιώντας την προηγούμενη βαθμολογία του γείτονα και κάνει μια πρόβλεψη για ένα στοιχείο. Με άλλα λόγια, η πρόταση πραγματοποιείται ελέγχοντας ποιος χρήστης έχει παρόμοιες προτιμήσεις.

Για να υπολογιστεί η ομοιότητα μεταξύ χρηστών χρησιμοποιούνται διάφορα μέτρα ομοιότητας ή δημιουργούνται συστάδες. Από την άλλη, η προσέγγιση με βάση τα αντικείμενα έχει παρόμοια φιλοσοφία με τα συστήματα συστάσεων που βασίζονται στο περιεχόμενο καθώς προτείνει αντικείμενα όμοια με άλλα που στο παρελθόν άρεσαν στον χρήστη. Αυτή η προσέγγιση είναι επίσης γνωστή ως φιλτράρισμα αντικειμένων και χρησιμοποιείται ευρέως από κολοσσούς του διαδικτύου όπως το Netflix και το Youtube.

2.3.2 Μέθοδος με βάση το μοντέλο

Η μέθοδος αυτή αναπτύσσει ένα μοντέλο χρήστη χρησιμοποιώντας βαθμολογίες κάθε χρήστη για να αξιολογήσει την αναμενόμενη τιμή των μη βαθμολογημένων αντικειμένων. Αυτή η μέθοδος χρησιμοποιεί γενικά αλγορίθμους μηχανικής εκμάθησης ή εξόρυξης δεδομένων για τη δημιουργία ενός μοντέλο. Το μοντέλο αναπτύσεται χρησιμοποιώντας ένα βοηθητικό πίνακα που δημιουργείται χρησιμοποιώντας τη βαθμολογία που δίνεται από τον χρήστη για οποιοδήποτε στοιχείο. Το μοντέλο εκπαιδεύεται με τη λήψη των πληροφοριών από το βοηθητικό πίνακα. Διαδεδομένοι αλγόριθμοι για αυτή την κατηγορία είναι η παραγοντοποίηση πινάκων και η Μοναδική Τιμή Αποσύνθεσης (Singular Value Decomposition).

2.4 Υβριδικά συστήματα συστάσεων

Είναι η ανερχόμενη κατηγορία συστήματων συστάσεων καθώς αποτελέσματα ερευνών έχουν δείξει πως ο συνδυασμός των συστημάτων συστάσεων βασισμένα στο περιεχομένο και των συστημάτων συνεργατικού φιλτραρίσματος είναι περισσοτέρο αποδοτικός σε σύγκριση με την μεμονομένη χρήση των συστημάτων αυτών. Η βασική αρχή του συνδυασμού των συστημάτων είναι η αποφυγή των ελατωμάττων της κάθε κατηγορίας. Μερικές μέθοδοι των υβριδικών συστημάτων είναι:

- Σταθμισμένο Βάρος: προσθέτει βαθμολογίες απο διαφορετικά προτεινόμενα στοιχεία.
- Εναλλαγή: επιλέγει μεθόδους με εναλλαγή σε διαφορετικά προτεινόμενα στοιχεία.
- Μίξη Αποτελεσμάτων: εμφανίζει αποτελέσματα συστάσεων απο διαφορετικά συστήματα.
- Συνδυασμός Χαρακτηριστικών : εξάγονται χαρακτηριστικά από διαφορετικές πηγές και συνδιάζονται ως μια είσοδος.

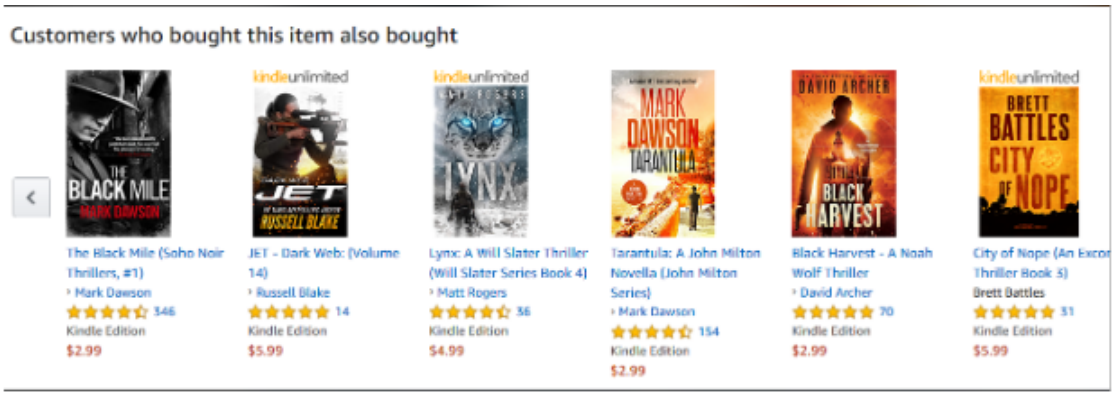
- Αύξηση Χαρακτηριστικών: υπολογίζονται τα χαρακτηριστικά από ένα σύστημα και χρησιμοποιείται το αποτέλεσμα στο επόμενο βήμα.
- Μετά-Επίπεδο: χρησιμοποιείται το μοντέλο που δημιουργήθηκε απο μια σύσταση ως είσοδος σε μια άλλη τεχνική σύστασης.

2.5 Σύγκριση

Κάθε προσέγγιση έχει τα πλεονεκτήματα και τα μειονεκτήματα της, και τα αποτελέσματα είναι διαφορετικά για διαφορετικά σύνολα δεδομένων. Μια προσέγγιση μπορεί να μην είναι κατάλληλη για όλα τα είδη προβλημάτων λόγω του ίδιου του αλγορίθμου. Για παράδειγμα, είναι δύσκολο να εφαρμοστεί η αυτοματοποιημένη εξαγωγή δεδομένων πολυμέσων με τα συστήματα που βασίζονται στο περιεχόμενο. Είναι επίσης πολύ δύσκολο να γίνουν προτάσεις σε χρήστες που δεν επιλέγουν ποτέ τίποτα. Η μέθοδος του συνεργατικού φιλτραρίσματος ξεπερνά το μειονέκτημα που αναφέρθηκε προηγουμένως. Αλλά τα συστήματα συνεργατικού φιλτραρίσματος βασίζονται σε μεγάλο αριθμό ιστορικών δεδομένων, οπότε υπάρχουν προβλήματα αραιοτήτων και το ”κρύο” ξεκίνημα. Ένα ακόμα πρόβλημα με τη μέθοδο αυτή είναι ότι όταν ένας μοναδικός χρήστης έχει μοναδικό γούστο, μπορεί να μην υπάρχουν παρόμοιες αντιστοιχίες άλλων χρηστών. Σε αυτή τη περίπτωση χρησιμοποιούμε τα συστήματα βασισμένα στο περιεχόμενο που δεν αντιμετωπίζουν τέτοιου είδους προβλήματα [14].

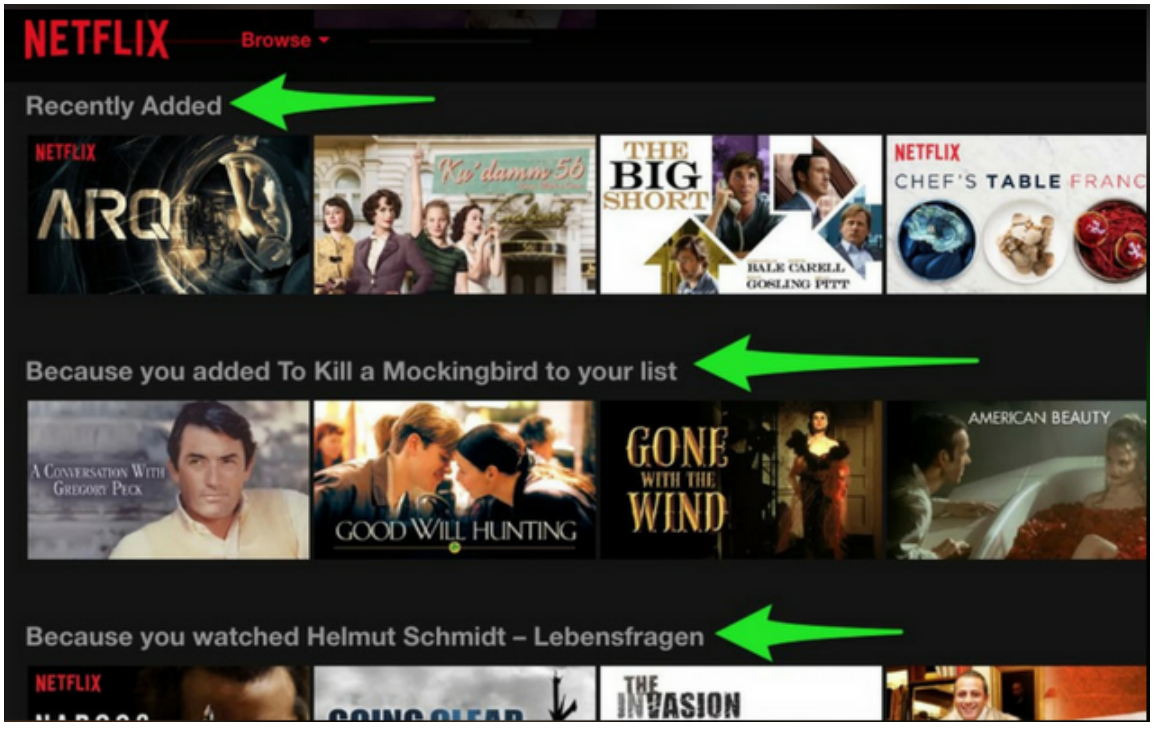
2.6 Γνωστά συστήματα συστάσεων

Υπάρχουν πολλές ιστοσελίδες που χρησιμοποιούν συστήματα συστάσεων. Αυτό συμβαίνει διότι οι ιστοσελίδες γίνονται πιο φιλικές προς το χρήστη και μπορούν με μεγαλύτερη ευκολία και επιτυχία να προωθήσουν τα προϊόντα τους. Τα πεδία που χρησιμοποιούν ευρέως τα συστήματα συστάσων είναι το ηλεκτρονικό εμπόριο, ιστοσελίδες με ταινίες, βίντεο και μουσική και ιστοσελίδες κοινωνικής δικτύωσης. Η πιο γνωστή ιστοσελίδα ηλεκτρονικού εμπορίου το Amazon χρησιμοποιεί εδώ και πολλά χρόνια συστήματα συστάσεων. Ένα από τα πολλά παραδείγματα που μπορούμε να δώσουμε είναι το ”πελάτες που αγόρασαν αυτό το προιόν, αγόρασαν επίσης” που ουσιαστικά προτείνει στους χρήστες παρόμοια προϊόντα με αυτά που αγόρασαν, όπως φαίνεται και από το Σχήμα 2.3.



Σχήμα 2.3: Προτάσεις Amazon [2]

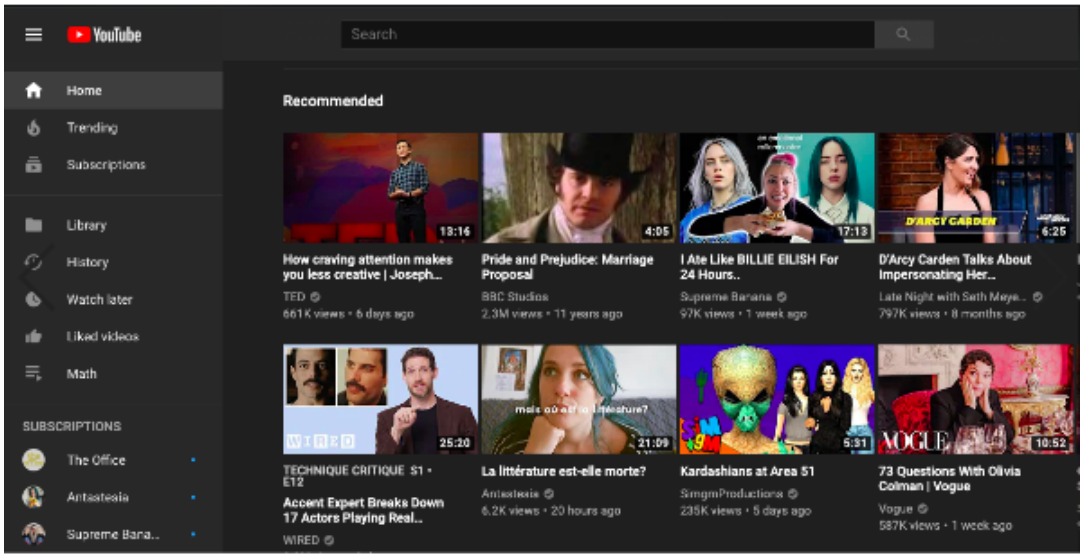
Ακόμα το Netflix έχει ενσωματώσει προτάσεις ταινιών που βοηθούν τους χρήστες να βρίσκουν ενδιαφέρον υλικό που θα τους κρατήσει στην ιστοσελίδα. Έχει επενδύσει τόσο πολύ στα συστήματα συστάσεων που πριν μερικά χρόνια είχε δημιουργήσει ενα διαγωνισμό για να βελτιώσουν το σύστημα τους με ένα πολύ μεγάλο χρηματικό έπαθλο. Στο Σχήμα 2.4 διακρίνουμε την αρχική σελίδα του Netflix με τις προτάσεις για τον χρήστη.



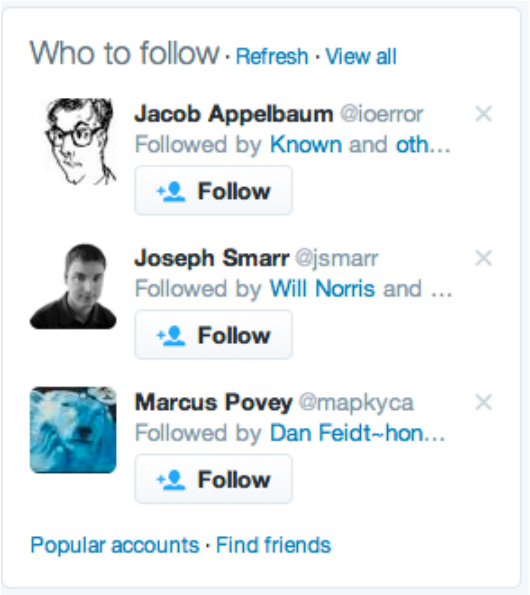
Σχήμα 2.4: Προτάσεις Netflix [3]

Τα μέσα κοινωνικής δικτύωσης είναι ένα σημαντικό κομμάτι της καθημερινότητας μας με δισεκατομμύρια ενεργούς χρήστες. Υπάρχει τεράστιος όγκος πληροφοριών στις πλατφόρμες αυτές, πράγμα που δυσκολεύει την εμπειρία των χρηστών κατά την χρησιμοποίηση

τους. Για την αποφυγή αυτής της κατάστασης οι πλατφόρμες ενσωμάτωσαν τα συστήματα συστάσεων, τα οποία αξιοποιούν αυτές τις πληροφορίες ώστε να προσφέρουν στον κάθε χρήστη αυτό που χρειάζεται. Τα παραδείγματα είναι πολλά αφού πλέον όλες οι πλατφορμες αξιοποιούν τα οφέλη των συστάσεων. Τέλος, ακολουθούν τα Σχήματα 2.5 και 2.6 με παραδείγματα συστάσεων από το YouTube και το Twitter προς τους χρήστες.



Σχήμα 2.5: Προτάσεις YouTube [4]



Σχήμα 2.6: Προτάσεις Twitter [5]

2.7 Συγγενικές εργασίες

Μια παρόμοια προσέγγιση για ένα σύστημα συστάσεων βασισμένο στο περιεχόμενο παρουσιάζεται στο άρθρο [6]. Η μέθοδος τους, αρχικά, περιλαμβάνει την μετατροπή του ιστορικού ενός χρήστη για κάθε ταινία σε έμμεση βαθμολογία (implicit rating). Χρησιμοποιούνται επίσης χαρακτηριστικά όπως ο σκηνοθέτης, οι ηθοποιοί, λέξεις κλειδιά κ.α. τα οποία ομαδοποιούνται. Για κάθε χαρακτηριστικό σε αυτές τις ομάδες, με βάση τις ταινίες που έχει παρακολουθήσει ο χρήστης και την βαθμολογία του για τις ταινίες αυτές, υπολογίζεται το βάρος του χαρακτηριστικού αυτού. Κάθε βάρος χαρακτηριστικού υπολογίζεται ξεχωριστά για κάθε χρήστη. Αν ένας χρήστης παρακολούθησε μια ταινία εντελώς ή μεγάλο μέρος της, τότε, τα χαρακτηριστικά αυτά αποκτούν μεγαλύτερο βάρος. Όταν μια ταινία χρειάζεται να αξιολογηθεί για έναν χρήστη, με βάση τα χαρακτηριστικά της ταινίας, παράγονται διαφορετικές αξιολογησεις για κάθε ομάδα χαρακτηριστικών οι οποίες και συγκρίνονται.

Μετατροπή διάρκειας προβολής ταινιών σε βαθμολογία

Στο σύστημα τους, οι χρήστες δεν αξιολογούν τις ταινίες άμεσα, οπότε πρέπει να υπολογιστούν έμμεσα χρησιμοποιώντας τη διάρκεια προβολής. Υποθέτοντας ότι ο χρήστης u βλέπει την ταινία i για $t(u, i)$ λεπτά κατά την διάρκεια ενός χρόνου, και t_i η συνολική διάρκεια της ταινίας i. Στην εξίσωση 2.1 παρουσιάζεται η κανονικοποιημένη διάρκεια προβολής του χρήστη u για την ταινία i.

$$r(u, i) = \frac{t(u, i)}{t_i} \tag{2.1}$$

Σημαντική παρατήρηση για αυτήν την φόρμουλα είναι οτι μπορεί να πάρει και τιμές μεγαλύτερες της μονάδας διότι ένας χρήστης μπορεί να παρακολουθήσει μια ταινία παραπάνω από μια φορά.

Μέθοδος υπολογισμού βάρους βάσει χαρακτηριστικών

Όπως αναφέραμε και παραπάνω, επειδή ο χρήστης δεν δίνει άμεση βαθμολογία για τις ταινίες, λαμβάνονται υπ' όψην μερικά χαρακτηριστικά τα οποία ομαδοποιούνται. Προκειμένου να προσδιοριστεί το βάρος κάθε χαρακτηριστικού για τον χρήστη χρησιμοποιούνται

τα δεδομένα από το σετ εκπαίδευσης. Έστω ότι ο χρήστης u παρακολουθεί τα αντικείμενα i_o,....,i_8, τα οποία αποτελούνται από τα χαρακτηριστικά j_o,..,j_3,..

user u	j_0	j_1	j_2	j_3	j_4	Puan
i_0	1	1	0	0	0	0.5
i_1	0	1	0	0	0	0.3
i_2	1	1	1	0	0	0.9
i_3	1	0	0	1	0	0.7
i_4	0	0	0	1	0	0.2
i_5	1	0	0	1	0	1.0
i_6	1	0	0	1	0	0.44
i_7	0	1	0	0	0	0.67
i_8	1	0	0	1	0	0.2
$w_k(u,j)$	0.42	0.26	0.1	0.26	0	-

Σχήμα 2.7: Πίνακας 1 [6]

Στον πίνακα του Σχήματος 2.7 βλέπουμε τα χαρακτηριστικά από την ομάδα k για τον χρήστη u. Το βάρος του χαρακτηριστικού j στην ομάδα k για τον χρήστη u υπολογίζεται από την φόρμουλα 2.2.

$$w_k(u,j) = \frac{1}{|I_u^{train}|} \sum_{i \in I_u^{train}} x_{k,u}(i,j) r(u,i) \tag{2.2}$$

Στην παραπάνω εξίσωση, το k αναπαριστά τον τύπο της ομάδας (ηθοποιός, κατηγορία, σκηνοθέτης). Το r(u,i) είναι η κανονικοποιημένη διάρκεια προβολής και το I_u^{train} είναι το σετ των ταινίων που είδε ο χρήστης u στην διάρκεια εκπαίδευσης.

Μέθοδος πρόβλεψης της αξιολόγησης

Αφού έχουν υπολογιστεί τα βάρη όλων των χαρακτηριστικών για κάθε χρήστη, τα χρησιμοποιούμε για να προβλέψουμε τις αξιολογήσεις των περιεχόμενων ταινιών. Υπολογίζουμε την αξιολόγηση για κάθε ομάδα χαρακτηριστικών ξεχωριστά χρησιμοποιώντας το βάρος $w_k(u,j)$ από τα δεδομένα εκπαίδευσης. Έπειτα υπάρχουν δύο μέθοδοι για τον υπολογισμό των προβλέψεων. Η πρώτη είναι με το άθροισμα από τα βάρη των χαρακτηριστικών (Εξί-

σωση 2.3) και η δεύτερη (Εξίσωση 2.4) είναι κανονικοποιώντας αυτό το άθροισμα, διαιρώντας το με τον αριθμό των χαρακτηριστικών.

$$r_k(u,i) = \sum_{j \in D_{k,i}} w_k(u,j) \tag{2.3}$$

$$r'_k(u,i) = \frac{1}{|D_{k,i}|} \sum_{j \in D_{k,i}} w_k(u,j) \tag{2.4}$$

Αξιολόγηση συστάσεων

Για την αξιολόγηση της απόδοσης του συστήματος χρησιμοποιήθηκαν οι μετρικές ακρίβεια (precision), ανάκληση (recall) και Μέτρο-F που περιγράφονται και από τις εξισώσεις 2.5, 2.6 και 2.7 αντίστοιχα. Ακόμα, για την μέτρηση της ακρίβειας του συστήματος χρησιμοποιήθηκαν οι πρώτες N επιλογές όπου σαν N μπαίνει ο αριθμός 10, καθώς οι προτάσεις που παράγει το σύστημα είναι 10.

$$Precision = \frac{hitCounts}{N} \tag{2.5}$$

$$Recall = \frac{hitCounts}{|I_u^{test}|} \tag{2.6}$$

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.7}$$

Μία ακόμα προσέγγιση για ένα συστήμα συστάσεων βασισμένο στο περιεχόμενο παρουσιάζεται στο άρθρο [15]. Το σύστημα συστάσεων που εφαρμόζεται κάνει προτάσεις ταινιών με βάση τα είδη των ταινιών. Ουσιαστίκα όταν ο χρήστης αξιολογεί με μεγάλο βαθμό μια ταινία που κατηγοριοποιείται σε συγκεκριμένο είδος τότε θα του προτείνονται ταινίες που ανήκουν σε αυτό το είδος. Για να επιτευχθεί αυτό χρησιμοποιήθηκε ένα σύνολο δεδομένων που χωρίστηκε σε δύο ενότητες. Η πρώτη ενότητα περιέχει τη λίστα των ταινιών μαζί με

τα είδη που ανήκουν. Το άλλο μέρος του συνόλου δεδομένων περιέχει μια λίστα με αξιολογήσεις ταινιών που έχουν βαθμολογηθεί από τον χρήστη σε κλίμακα 1-5, με 5 να είναι το υψηλότερο. Πρώτα, πρέπει να δημιουργηθεί ένα συνδυασμένο σύνολο δεδομένων ταινιών, ειδών και των αξιολογήσεών τους συσχετίζοντας τα είδη με τις βαθμολογίες. Για λόγους απλότητας, οι βαθμολογίες μετατρέπονται σε δυαδικές τιμές. Εάν η βαθμολογία που δίνεται από έναν συγκεκριμένο χρήστη είναι μεγαλύτερη από 3,η τιμή μετατρέπεται σε 1, διαφορετικά η τιμή μετατρέπεται σε -1. Τα είδη που χαρακτηρίζουν μια ταινία παίρνουν την τιμή 1 διαφορετικά λαμβάνουν την τιμή 0. Τα δύο σύνολα συνδιάζονται σε έναν ενιαίο πίνακα με τη βοήθεια τις κλιμακωτής συνάρτησης (scalar product).

Αλγόριθμος

Τα βήματα του αλγορίθμου είναι τα εξής:

- Κατασκευάστε ένα πλαίσιο δεδομένων του συνόλου δεδομένων του είδους με το αναγνωριστικό ταινίας ως τις σειρές και τα είδη ως στήλες διαχωρισμένες με κάθετη γραμμή.
- Δημιουργήστε μια λίστα με όλα τα είδη που είναι διαθέσιμα στο σύνολο δεδομένων.
- Επαναλάβετε μέσω του προηγούμενου πλαισίου δεδομένων είδους. Εάν υπάρχει ένα είδος σε μια ταινία, η τιμή 1 αντιστοιχεί στον πίνακα ειδών.
- Δημιουργήστε έναν πίνακα βαθμολογίας που αντιστοιχίζει 1 για ταινίες με βαθμολογία μεγαλύτερη από 3 και -1 για ταινίες με βαθμολογίες μικρότερο ή ίσο με 3.
- Υπολογίστε το προϊόν κουκκίδων των πινάκων είδους και βαθμολογιών. Αυτός είναι ο πίνακας αποτελεσμάτων.
- Μετατροπή του πίνακα αποτελεσμάτων σε δυαδική μορφή. Για ένα προϊόν με αρνητικές κουκκίδες, εκχωρείτε η τιμή 0, αλλιώς εκχωρείτε η τιμή 1.
- Υπολογίστε την ευκλείδια απόσταση μεταξύ του τρέχοντος χρήστη και άλλων χρηστών.
- Διατηρήστε τις σειρές που έχουν την ελάχιστη απόσταση. Αυτές είναι οι προτεινόμενες ταινίες για τον τρέχοντα χρήστη.

Μια ακόμα ενδιαφέρουσα προσέγγιση για ένα σύστημα συστάσεων που βασίζεται στο περιεχόμενο είναι αυτή που παρουσιάζεται στο αρθρο [16]. Σε αυτή τη δημοσίευση αναλύθηκε η συμπεριφορά αξιολόγησης των χρήστων πάνω σε ένα σύνολο δεδομένων και διαπιστώθηκε οτι οι συμπεριφορές των χρηστών στα συστήματα κοινωνικών μέσων επηρεάζεται και από χρονικές προτιμήσεις. Έτσι προτείνετε μια νέα τεχνική βασισμένη στο περιεχόμενο που χρησιμοποιεί ένα χρονικό μοντέλο χρήστη.

Μοντελοποίηση των χρονικών προτιμήσεων του χρήστη

Σε αυτό το μοντέλο το προφίλ του χρήστη αποτελείται από τις δραστηριότητες του, όπου κάθε δραστηριότητα δείχνει το περιεχόμενο και το χρόνο πρόσβασης των επιλεγμένων αντικειμένων. Αυτό το μοντέλο είναι επικεντρωμένο στο χρήστη. Το μοντέλο χρονικών προτιμήσεων βασίζεται στο μη παραμετρικό πλαίσιο Bayesian και έχει τρία κύρια στοιχεία: την εξαγωγή ενδιαφέροντος, τα συμπεράσματα προτιμήσεων και την πρόβλεψη.

Εξαγωγή ενδιαφέροντος

Αναλύεται το προφίλ του χρήστη για να βρεθούν τα αντικείμενα που τον ενδιαφέρουν. Χρησιμοποιείται η διαδικασία εξαρτώμενου κινεζικού εστιατορίου (Distance Dependent Chinese Restaurant Process) που είναι μια μέθοδος κατανομής πιθανοτήτων και ομαδοποιούνται τα αποτελέσματα. Οι συστάδες μπορούν να μεγαλώσουν κάθε φορά που υπάρχουν νέα δεδομένα. Κάθε συστάδα υποδεικνύει μια ομάδα παρόμοιων αντικειμένουν που ενδιαφέρουν τον κάθε χρήστη. Για να γίνει η ομαδοποίηση χρησιμοποιείται η εξίσωση 2.8. Οι τιμές κλιμακώνονται στο εύρος [0,1].

$$p(t_{a_i} = k|t, a_i, a) = \begin{cases} Sim(.) * \mathscr{L}(.) & k \in t \\ a & k > |t| \end{cases} \tag{2.8}$$

Συμπεράσματα προτιμήσεων

Εξάγοντας τα ενδιαφέροντα των χρηστών, λαμβάνεται υπόψη η διάρκεια και ο αριθμός των δραστηριοτήτων των χρηστών για να υπολογίστει το διάνυσμα προτίμησης. Κάθε στοιχείο του διανύσματος προτιμήσεων δείχνει την πιθανότητα για ένα χρήστη να επιλέξει ένα αντικείμενο για να πραγματοποιήσει νέες δραστηριότητες. Η εξίσωση 2.9 χρησιμοποιείται

για την κατασκευή του διανύσματος προτίμησης για έναν χρήστη.

$$Pref(cluster_i) \sim \sum_{j=1}^{|cluster_i|} Age(activity_j) \tag{2.9}$$

Πρόβλεψη

Η πρόβλεψη βασίζεται στην πιθανότητα που έχει ένα νέο αντικείμενο να επιλεχθεί από τον χρήστη. Αρχικά υπολογίζεται η πιθανότητα του νέου αντικειμένου να ανατεθεί σε μία συστάδα και αυτό γίνεται μέσω της συνάρτησης ομοιότητας Sim(). Έπειτα, το διάνυσμα προτιμήσεων χρησιμοποιείται για να καθορίσει την προτεραιότητα της συστάδας που εισήχθε το νέο αντικείμενο. Έτσι για να υπολογιστεί η πιθανότητα της επιλογής ενός νέου στοιχείου χρησιμοποιείται η φόρμουλα 2.10

$$p(a * |.) \sim \sum_{c_i \in Cluster} (Sim(a*, c_i, .) * Pref(c_i)) \tag{2.10}$$

Για να κατασκευαστεί το σύστημα συστάσεων , χρησιμοποιείται η εξαγωγή ενδιαφέροντος και τα συμπεράσματα προτιμήσεων για να μοντελοποιηθούν οι προτιμήσεις του χρήστη και η πρόβλεψη για τη δημιουργία της λίστας προτάσεων. Πιο αναλυτικά το προφίλ του χρήστη αποτελείται απο πληροφορίες για ταινίες τις οποίες αξιολόγησε μια καθορισμένη χρονική στιγμή. Οι πληροφορίες για τις ταινίες συλλέγονται απο το IMDB. Από αυτές μπορούν να βρεθούν τα ενδιαφέροντα των χρηστών. Κάθε αντικείμενο που τον ενδιαφέρει δείχνει μια ομάδα παρόμοιων ταινιών που έχουν επιλεχθεί απο τον χρήστη στο παρελθόν. Χρησιμοποιείται ο χρόνος αξιολόγησης και ο αριθμός των ταινιών σε κάθε ομάδα για να υπολογιστεί η πιθανότητα κάθε ομάδας. Η πιθανότητα αυτή δείχνει πόσο πιθανό είναι ο χρήστης να επιλέξει μία συγκεκριμένη ομάδα. Από την ομάδα με την μεγαλύτερη πιθανότητα δημιουργείται η λίστα ταινιών που θα προταθεί στο χρήστη.

Τα βήματα της προσέγγισης αυτής είναι τα εξής:

- Λήψη του αρχείου αξιολόγησης χρήστη και δημιουργία προφίλ.
- Εξαγωγή των ενδιαφερόντων του χρήστη και συμπερασμάτων για τις προτιμήσεις του.
- Δημιουργία του διανύσματος προτίμησης.
- Υπολογισμός της πιθανότητας να ανατεθεί μία ταινία σε μία συστάδα.

- Υπολογισμός της πιθανότητας μίας ταινίας m με δεδομένο ότι ανήκει σε μία συστάδα c.

- Δημιουργία της λίστας συστάσεων που προτείνεται στο χρήστη.

Κεφάλαιο 3

Αξιολόγηση συστημάτων

3.1 Εισαγωγή

Το επόμενο βήμα μετά την υλοποίηση ενός συστήματος συστάσεων είναι η αξιολόγηση της απόδοσης του. Για τον σκοπό αυτό χρησιμοποιούνται διάφορες μετρικές που χωρίζονται σε δύο κατηγορίες. Η μια κατηγορία αφορά τα σφάλματα των προβλέψεων και περιλαμβάνει το Απόλυτο σφάλμα(Mean Absolute Error) και το Σφάλμα τετραγωνικής ρίζας(Root Mean Square Error) ενώ η άλλη αφορά το ποσοστό επιτυχίας των προβλέψεων με τις πιο γνωστές μετρικές να είναι ο Αμοιβαίος μέσος όρος επιτυχίας(Average Reciprocal Hit Rate) και το Αθροιστικό ποσοστό επιτυχίας (cumulative Hit Rate).

3.2 Απόλυτο σφάλμα

Το απόλυτο σφάλμα, όπως μας προϊδεάζει και το όνομα του, πρόκειται για το απόλυτο σφάλμα στις προβλέψεις. Από την εξίσωση 3.1 βλέπουμε πως ο υπολογισμός του γίνεται από τον μέσο όρο των σφαλμάτων που προκύπτουν από τις βαθμολογίες του συστήματος συστάσεων.

$$\frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{3.1}$$

3.3 Σφάλμα τετραγωνικής ρίζας

Το σφάλμα τετραγωνικής ρίζας (εξίσωση 3.2) είναι προτιμότερη μετρική για την αξιολόγηση των συστημάτων συστάσεων διότι παίρνει τον μέσο όρο της ρίζας των τετραγώνων των σφαλμάτων από τις προβλέψεις, γεγονός που καθιστά το σφάλμα μεγαλύτερο όταν η πρόβλεψη είναι κακή, και μικρότερο όταν η πρόβλεψη είναι κοντά στην πραγματική τιμή.

$$\sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i)^2}{n}} \tag{3.2}$$

3.4 Ποσοστό επιτυχίας

Συνήθως οι πρώτες αξιολογήσεις των συστημάτων συστάσεων περιλαμβάνουν τις μετρικές των σφαλμάτων που αναλύσαμε στην προηγούμενη υποενότητα διότι είναι εύκολος ο υπολογισμός τους και μας δίνει μια πρώτη εικόνα για την απόδοση του συστήματος που δημιουργήσαμε. Όμως σε πιο πρακτικό κομμάτι εάν αναλογιστούμε την χρησιμότητα των μετρικών για τα σφάλματα συμπεραίνουμε πως είναι λίγο ‘’άστοχα’’ σε σχέση με τις μετρικές που σχετίζονται με το ποσοστό επιτυχημένων προβλέψεων, γνωστό και ως ποσοστό επιτυχίας. Η εξήγηση είναι απλή. Το σφάλμα που έχουμε αναφέρει τόσες φορές σε αυτήν την ενότητα είναι η διαφορά που έχει η πρόβλεψη μας από την πραγματική τιμή. Με απλά λόγια προσπαθούμε να προσεγγίσουμε ,όσο πιο πολύ γίνεται, μια τιμή που ήδη γνωρίζουμε. Ο χρήστης δεν ενδιαφέρεται για μια σωστή πρόβλεψη σε μια ταινία που έχει ήδη δει αλλά για μια πρόταση σε μια ταινία που θα του αρέσει. Γι αυτό το λόγο οι μετρικές που συνδέονται με το ποσοστό επιτυχίας δίνουν ένα πιο ορθό συμπέρασμα για την απόδοση του συστήματος.

3.5 Αμοιβαίος μέσος όρος επιτυχίας

Αυτή η μετρική είναι παρόμοια με το ποσοστό επιτυχημένων προβλέψεων (hit rate), αλλά για κάθε χρήστη για τον οποίο έχουμε επιτυχημένη πρόβλεψη λαμβάνουμε υπ’ όψην μας και την θέση στην οποία βρισκόταν η πρόταση μας (εξίσωση 3.3). Αυτή είναι μια μετρική εστιασμένη στο χρήστη, καθώς οι άνθρωποι τείνουν να επικεντρώνονται περισσότερο σε αυτό που βλέπουν στην αρχή των κορυφαίων λιστών. Δίνουμε λοιπόν μεγαλύτερο βάρος σε αυτές τις επιτυχίες που εμφανίζονται στην κορυφή της λίστας.

$$\frac{\sum_{i=1}^{n} \frac{1}{rank_i}}{Users} \tag{3.3}$$

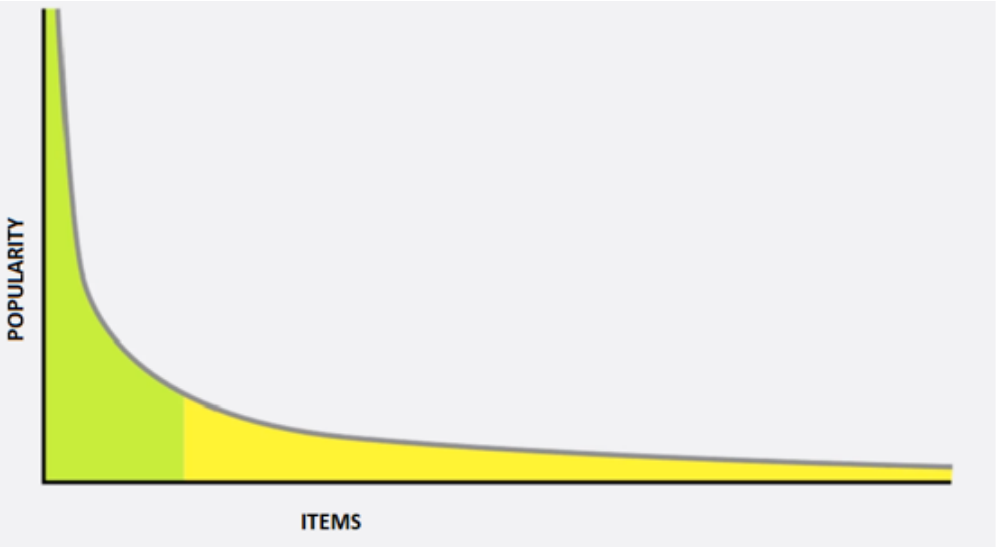
3.6 Αθροιστικό ποσοστό επιτυχίας

Το Αθροιστικό ποσοστό επιτυχίας (Σχήμα 3.1) είναι απλά η μετρική του ποσοστού επιτυχίας με ένα προκαθορισμένο, από τον δημιουργό του συστήματος, κατώφλι έτσι ώστε να χρησιμοποιούνται από το τεστ σετ ταινίες που πραγματικά άρεσαν στον χρήστη.

Σχήμα 3.1: Αθροιστικό ποσοστό επιτυχίας [7]

3.7 Κάλυψη

Τι κάνει μια πρόταση ή μια λίστα προτάσεων να είναι καλές; Η έρευνα σχετικά με τα συστήματα συστάσεων έχει παραδοσιακά επικεντρωθεί στην ακρίβεια, και ειδικότερα πόσο κοντά είναι οι προβλεπόμενες αξιολογήσεις του συνιστώμενου με τις πραγματικές αξιολογήσεις των χρηστών. Ωστόσο, έχει αναγνωριστεί ότι άλλες ιδιότητες προτάσεων - όπως το αν η λίστα των προτάσεων είναι διαφορετική (diversity) και αν περιέχει νέα στοιχεία (novelty) - μπορεί να έχουν σημαντικό αντίκτυπο στη συνολική ποιότητα ενός συστήματος συστάσεων. Κατά συνέπεια, τα τελευταία χρόνια, η εστίαση της έρευνας συστημάτων σύστασης έχει μετατοπιστεί ώστε να περιλαμβάνει ένα ευρύτερο φάσμα στόχων «πέρα από την ακρίβεια». Ο δημιουργός ενός συστήματος συστάσεων θέλει να ενσωματώσει διαφορετικά και μη δημοφιλή αντικείμενα για να επιτύχει τον όρο ”η βαριά ουρά” (the long tail). Όπως φαίνεται στο Σχήμα 3.2 ο συνδυασμός του διαφορετικού και του νέου προϊόντος πετυχαίνει το τέλειο αποτέλεσμα σε ένα σύστημα συστάσεων.



Σχήμα 3.2: ”Η βαριά ουρά” [7]

Κεφάλαιο 4

Περιβάλλον και εργαλεία

4.1 Python

Για την δημιουργία της εφαρμογής μας χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python και συγκεκριμένα η έκδοση Python 3.8.5. Πρόκειται για μια αντικειμενοστραφή γλώσσα υψηλού επιπέδου που δημιουργήθηκε το 1989 από τον Ολλανδό Γκίντο βαν Ρόσσουμ και κυκλοφόρησε το 1991. Αν και είναι από τις παλαίοτερες αντικειμενοστραφείς γλώσσες, η δημοτικότητα της ανέβηκε πολύ την τελευταία δεκαέτια. Αυτό οφείλεται στο γεγονός ότι ο κώδικας της είναι ευανάγνωστος, καθώς αντί για συντομογραφίες χρησιμοποιούνται ολόκληρες αγγλικές λέξεις, και στο ότι ο χρήστης δεν δυσκολεύεται στην εκμάθηση της. Αξιοσημείωτο επίσης είναι η πληθώρα από βιβλιοθήκες που υπάρχουν προεγκατεστημένες και βοηθούν σε συνηθισμένες εργασίες. Παρόλο που υστερεί σε ταχύτητα σε σχέση με τους κύριους ‘’ανταγωνιστές’’ της (Java, C++), αφού δεν πρόκειται για μεταγλωτισμένη γλώσσα, ο χρόνος που γλυτώνει ο προγραμματιστής στο κομμάτι της ανάπτυξης της εφαρμογής την καθιστά ως την πιο δημοφιλή γλώσσα προγραμματισμού.

4.2 Anaconda

Το Anaconda πρόκειται για μια διανομή των γλωσσών προγραμματισμού Python και R που στοχεύει στην απλοποίηση της διαχείρισης και της ανάπτυξης πακέτων. Είναι το κατάλληλο περιβάλλον για τον συνδυασμό Python και της επιστήμης των δεδομένων καθώς περιέχει εγκατεστημένα πακέτα και εργαλεία που διευκολύνουν τον χρήστη που έχει ως σκοπό να ασχοληθεί με μηχανική εκμάθηση ή με την επιστήμη των δεδομένων.

4.3 Spyder

Το Spyder είναι ένα ισχυρό επιστημονικό περιβάλλον γραμμένο σε Python, για την Python, και σχεδιάστηκε από και για επιστήμονες, μηχανικούς και αναλυτές δεδομένων. Διαθέτει έναν μοναδικό συνδυασμό της προηγμένης λειτουργικότητας επεξεργασίας, ανάλυσης, εντοπισμού σφαλμάτων και προφίλ ενός ολοκληρωμένου εργαλείου ανάπτυξης με την εξερεύνηση δεδομένων, τη διαδραστική εκτέλεση, τη βαθιά επιθεώρηση και τις όμορφες δυνατότητες οπτικοποίησης ενός επιστημονικού πακέτου. Επιπλέον, το Spyder προσφέρει ενσωματωμένα πολλά δημοφιλή επιστημονικά πακέτα, όπως τα NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy και άλλα. Πέρα από τις πολλές ενσωματωμένες δυνατότητές του, το Spyder μπορεί να επεκταθεί ακόμη περισσότερο μέσω προσθήκες τρίτων. Το Spyder μπορεί επίσης να χρησιμοποιηθεί ως βιβλιοθήκη επέκτασης PyQt5, επιτρέποντας την ενσωμάτωση του σε άλλο πρόγραμμα επεξεργασίας ή λογισμικό.

4.4 Django

Για τον σχεδιασμό της εφαρμογής χρησιμοποιήσαμε την πλατφόρμα σχεδίασης Django. Το Django είναι μια πλατφόρμα σχεδίασης ανοιχτού κώδικα γραμμένο σε Python που επιτρέπει την ταχεία ανάπτυξη ασφαλών και διατηρήσιμων ιστότοπων. Ο κύριος σκοπός του είναι να αποσυμφορεί τον χρήστη από την διαδικασία ανάπτυξης της ιστοσελίδας ώστε να επικεντρώνεται στην ανάπτυξη της εφαρμογής. Άξιο αναφοράς είναι η ευελιξία στην δυνατότητα κλιμάκωσης της πλατφόρμας. Το Django έχει σχεδιαστεί με τέτοιο τρόπο ώστε να μπορεί να δέχεται όσο hardware προσφέρει ο χρήστης με αποτέλεσμα να επεκτείνει την εφαρμογή πολύ εύκολα.

Το Django υποστηρίζει το πρότυπο MVC (Model View Controller) και ειδικότερα το πρότυπο MTV (Model Template View). Το πρότυπο αυτό ικανοποιεί την ανάγκη για διαχωρισμό του περιεχομένου από την παρουσίαση. Ακόμα καθιστά ευκολότερη την τροποποίηση και συντήρηση της εφαρμογής.

4.5 Βάσεις δεδομένων και Django

Επιπλέον το Django παρέχει την δυνατότητα πρόσβασης σε βάσεις δεδομένων καθώς και την δημιουργία ερωτημάτων σε SQL από τον προγραμματιστή. Υποστηρίζει πολλούς από

τους γνωστούς τύπους βάσεων δεδομένων όπως η PostgreSQL, MySQL, SQLite και Oracle Database. Στην παρούσα εργασία χρησιμοποιήσαμε την SQLite η οποία υπάρχει προεγκατεστημένη στις βιβλιοθήκες της Python. Η SQLite χαρακτηρίζεται από μια απλότητα στην χρήση καθώς όλη η βάση αποθηκεύεται σε ένα αρχείο γεγονός που καθιστά την μεταφορά της πολύ εύκολη. Επίσης είναι αρκετά γρήγορη και βολική για διαδικτυακές εφαρμογές μέτριας και μικρής επισκεψιμότητας όπως η δική μας.

4.6 Πακέτα στο Anaconda

Τα πακέτα (modules) που εγκαταστήσαμε στο περιβάλλον εργασίας του Anaconda προκειμένου να λειτουργήσει ο κωδικάς μας είναι τα εξής:

- Numpy: Είναι μια βιβλιοθήκη που χρησιμοποιείται για εργασία με πίνακες. Στην περίπτωση μας είναι αναγκαία για τη παραγοντοποιήση πινάκων.
- Pandas: Είναι η πιο δίασημη βιβλιοθήκη Python για ανάλυση δεδομένων. Παρέχει εξαιρετικά βελτιστοποιημένη απόδοση με πηγαίο κώδικα back-end γραμμένο καθαρά σε C ή Python.
- SurpriseLib: Πρόκειται για μια βιβλιοθήκη που βοηθάει στην δημιουργία και ανάλυση συστημάτων συστάσεων με αποκλειστικά δεδομένα.

4.7 Χρήση του API

Για την αναζήτηση των ταινιών χρησιμοποίησαμε το API της σελίδας Open Movie DataBase [8]. Δημιουργήσαμε έναν λογαριασμό στην ιστοσελίδα και είχαμε στην διάθεση μας το δωρεάν API κλειδί που επιτρέπει μέχρι 1000 αιτήματα την ημέρα. Όταν γίνεται ένα αίτημα μας επιστρέφει τα αποτελέσματα σε μορφή JSON. Το Σχήμα 4.1 δείχνει ένα παράδειγμα χρήσης του API.

```
if query:
    url = 'http://www.omdbapi.com/?apikey=d4a899e2&s=' + query
    response = requests.get(url)
    movie_info = response.json()
```

Σχήμα 4.1: Παράδειγμα χρήσης του API

Υπάρχουν δύο τρόποι αναζήτησης ταινίας με δύο διαφορετικές παραμέτρους. Με την παράμετρο αναζήτησης ”i” μπορείς να ψάξεις για μια ταινία με το Imdb ID της ενώ με την παράμετρο ”t” αναζητάς μια ταινία με βάση τον τίτλο της. Στο Σχήμα 4.2 ακολουθεί αναλυτικότερη εξήγηση των παραμέτρων.

By ID or Title

Parameter	Required	Valid Options	Default Value	Description
i	Optional*		<empty>	A valid IMDb ID (e.g. tt1285016)
t	Optional*		<empty>	Movie title to search for.
type	No	movie, series, episode	<empty>	Type of result to return.
y	No		<empty>	Year of release.
plot	No	short, full	short	Return short or full plot.
r	No	json, xml	json	The data type to return.
callback	No		<empty>	JSONP callback name.
v	No		1	API version (reserved for future use).

Σχήμα 4.2: Παράμετροι που υπάρχουν για αναζήτηση πληροφοριών [8]

Κεφάλαιο 5

Αλγόριθμος συστάσεων

5.1 Εισαγωγή

Στο παρακάτω κεφάλαιο θα αναλύσουμε τον αλγόριθμο συστάστεων που χρησιμοποιήσαμε για να την ιστοσελίδα μας. Όπως θα έχει γίνει ήδη αντιλυπτό είναι ένας αλγόριθμος που βασίζεται στο περιεχόμενο και στηρίζεται στην ομοιότητα για να κάνει τις τελικές συστάσεις. Η ιστοσελίδα που δημιουργήσαμε είναι για ταινίες όποτε καταλαβαίνουμε ότι ο αλγόριθμος χρησιμοποιείται για συστάσεις ταινιών. Επίσης θα αναλύσουμε τους αλγορίθμους που απορρίψαμε γιατί θεωρήσαμε ότι θα πετύχουμε καλύτερα αποτελέσματα με την χρήση του πρώτου

5.2 Σύνολο δεδομένων

Για τον αλγόριθμο των συστάσεων χρησιμοποιείται ένα σύνολο δεδομένων 5000 ταινιών από το Movielens [17]. Αποτελείται από δύο αρχεία όπου το πρώτο περιέχει τα παρακάτω:

- movieId: Ένα μοναδικό αναγνωριστικό για κάθε ταινία.
- cast: Το όνομα των πρωταγωνιστών και υποστηρικτικών ηθοποιών.
- crew: Το όνομα του σκηνοθέτη, συντάκτη, συνθέτη, συγγραφέα

Το δεύτερο αρχείο περιέχει τις εξής πληροφορίες:

- budget: Ο προϋπολογισμός της ταινίας.
- genre: Το είδος της ταινίας

- homepage: Ένας σύνδεσμος προς την αρχική σελίδα της ταινίας.
- id: Ένα μοναδικό αναγνωριστικό για κάθε ταινία.
- keywords: Οι λέξεις-κλειδιά ή οι ετικέτες που σχετίζονται με την ταινία.
- originalLanguage: Η γλώσσα στην οποία έγινε η ταινία.
- originalTitle: Ο τίτλος της ταινίας πριν από τη μετάφραση ή την προσαρμογή.
- overview: Μια σύντομη περιγραφή της ταινίας.
- popularity: Μια αριθμητική ποσότητα που καθορίζει τη δημοτικότητα της ταινίας.
- productionCompanies: Το όνομα της εταιρίας παραγωγής της ταινίας.
- productionCountries: Η χώρα που γυρίστηκε η ταινία.
- releaseDate: Η ημερομηνία κατά την οποία κυκλοφόρησε.
- revenue: Τα παγκόσμια έσοδα που δημιουργήθηκαν από την ταινία.
- runtime: Ο χρόνος διάρκειας της ταινίας σε λεπτά.
- status: Αν η ταινία έχει γυριστεί ή αν φημολογείτε οτι θα γυριστεί.
- tagline: Η ετικέτα της ταινίας.
- title: Τίτλος της ταινίας.
- voteAverage: Μέσες βαθμολογίες που έλαβε η ταινία.
- voteCount: Ο αριθμός των ψήφων που έχει λάβει η ταινία.

Για να μπορούμε να διαβάσουμε και να αξιοποιήσουμε τα παραπάνω δεδομένα χρησιμοποιήσαμε τις βιβλιοθήκες pandas και numpy.

5.3 Αλγόριθμος ένα

Ο πρώτος αλγόριθμος που θα αναφέρουμε είναι και αυτός που χρησιμοποιήσαμε τελικά. Είναι αλγόριθμος που κάνει συστάσεις με βάση τους τρεις βασικούς ηθοποιούς μια ταινίας, τον σκηνοθέτη, το είδος, τις βασικές λέξεις της πλοκής και το όνομα της παραγωγής. Κάθε μια επιλογή έχει βάρος όπου αν αλλάξει τότε οι συστάσεις αλλάζουν και αυτές.

5.3.1 Συχνότητα όρου - αντίστροφη συχνότητα εγγράφου

Από τη διαίσθησή μας, πιστεύουμε ότι οι λέξεις που εμφανίζονται πιο συχνά θα πρέπει να έχουν μεγαλύτερο βάρος στην ανάλυση δεδομένων κειμένου, αλλά αυτό δεν ισχύει πάντα. Λέξεις όπως ”το”, ”θα” και ”και” - που ονομάζονται λέξεις-κλειδιά - εμφανίζονται πιο συχνά σε ένα σώμα κειμένου, αλλά έχουν πολύ μικρή σημασία. Αντίθετα, οι λέξεις που είναι σπάνιες είναι αυτές που πραγματικά βοηθούν στη διάκριση μεταξύ των δεδομένων και έχουν μεγαλύτερο βάρος. Για να καταφέρουμε να μειώσουμε τη σημασία των λέξων κλειδιών θα χρησιμοποιήσουμε τη συχνότητα όρου - αντίστροφη συχνότητα εγγράφου (TF-IDF) [18] που είναι μια από τις πιο ευρέως χρησιμοποιούμενες τεχνικές για την επεξεργασία δεδομένων κειμένου. Πρώτον, θα μάθουμε τι σημαίνει αυτός ο όρος μαθηματικά.

Συχνότητα Όρου

Μας δίνει τη συχνότητα της λέξης σε κάθε έγγραφο του σώματος. Είναι ο λόγος των φορών που η λέξη εμφανίζεται σε ένα έγγραφο σε σύγκριση με τον συνολικό αριθμό λέξεων σε αυτό το έγγραφο (Εξίσωση 5.1). Αυξάνεται καθώς αυξάνεται ο αριθμός των εμφανίσεων αυτής της λέξης στο έγγραφο. Κάθε έγγραφο έχει τη δική του συχνότητα όρου.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}} \tag{5.1}$$

Αντίστροφη συχνότητα εγγράφου

Χρησιμοποιείται για τον υπολογισμό του βάρους των σπάνιων λέξεων σε όλα τα έγγραφα του σώματος. Οι λέξεις που εμφανίζονται σπάνια στο σώμα έχουν υψηλή αντίστροφη συχνότητα εγγράφου (Εξίσωση 5.2).

$$idf(w) = \log(\frac{N}{df_t}) \tag{5.2}$$

Συνδυάζοντας αυτά τα δύο καταλήγουμε στη βαθμολογία συχνότητα όρου - αντίστροφη συχνότητα εγγράφου (w) για μια λέξη σε ένα έγγραφο στο σώμα. Η μαθηματική εξίσωση είναι η 5.3.

$$w_{i,j} = tf_{i,j} \times \log(\frac{N}{df_t}) \tag{5.3}$$

Χρησιμοποιήσαμε τη συχνότητα όρου - αντίστροφη συχνότητα εγγράφου που έχει ενσωματωμένη η βιβλιοθήκη scikit-learn για να μειώσουμε το βάρος των απλών λέξεων στη πλοκή. Για να περιγράψουν τις 4800 ταινίες χρησιμοποιήθηκαν πάνω από 20.000 διαφορετικές λέξεις.

5.3.2 Μεταδεδομένα

Είναι αυτονόητο ότι η ποιότητα του συστήματός μας θα αυξηθεί με τη χρήση καλύτερων μεταδεδομένων. Πρόκειται να δημιουργήσουμε έναν αλγόριμθο με βάση τα ακόλουθα μεταδεδομένα: τους 3 κορυφαίους ηθοποιούς, τον σκηνοθέτη, τα είδη, τις λέξεις-κλειδιά και την εταιρία παραγωγής της ταινίας. Από το ”cast”, ”crew” και τα ”keywords”, πρέπει να εξαγάγουμε τους τρεις πιο σημαντικούς ηθοποιούς, τον σκηνοθέτη και τις λέξεις-κλειδιά για κάθε ταινία. Τα δεδομένα μας υπάρχουν με τη μορφή αυστηροποιημένων λιστών, οπότε πρέπει να τα μετατρέψουμε σε ασφαλή και χρησιμοποιήσιμη δομή. Για να πάρουμε και να αποθηκεύσουμε τις πληροφορίες που χρειαζόμαστε δημιουργήσαμε τις συναρτήσεις getDirector(x) για να βρίσκουμε τον σκηνοθέτη και getList(x) για να υπολογίζουμε τους 3 βασικούς ηθοποιούς. Το επόμενο βήμα είναι να μετατρέψουμε τα ονόματα και τις εμφανίσεις λέξεων-κλειδιών σε πεζά και να αφαιρέσουμε όλα τα κενά μεταξύ τους ώστε ηθοποιοί με το ίδιο πρώτο όνομα να μην θεωρούνται ως ο ίδιος άνθρωπος. Εφόσον έχουν γίνει τα παραπάνω βήματα με επιτυχία μπορούμε πλέον να δημιουργήσουμε τη συνάρτηση που ενώνει τα χαρακτηριστικά που έχουμε αναφέρει. Η συνάρτηση ονομάζεται createTheMix(x) και επιστρέφει μια λίστα με τα στοιχεία αυτά. Κάθε μια επιλογή έχει βάρος όπου αν αλλάξει τότε οι συστάσεις αλλάζουν και αυτές. Για παράδειγμα βάζοντας ειδικό βάρος δύο στο είδος της ταινίας και στο σκηνοθέτη ο αλγόριθμος κατανοεί ότι αυτά τα δύο χαρακτηριστικά είναι πιο σημαντικά στην αναζήτηση όμοιων ταινιών.Μετά από πληθώρα δοκιμών στο ειδικό βάρος σε κάθε χαρακτηριστικό και σε αρκετούς συνδιασμούς τους παρατηρήσαμε ότι οι καλύτερες συστάσεις γίνονται δίνοντας σε όλα τα χαρακτηριστικά βάρος ένα. Τέλος αναφέραμε παραπάνω ότι όταν προσθέτεις στοιχεία στο σύστημα συστάσεων πετυχαίνεις συνήθως καλύτερα μεταδεδομένα. Αυτό βέβαια δεν συμβαίνει πάντα. Αυτό το καταλάβαμε όταν προσπαθήσαμε να προσθέσουμε τη δημοτικότητα (popularity) στη συνάρτηση createTheMix(x) που σαν λογική φαινόταν σωστή και θεωριτικά θα αναβάθμιζε αρκετά το σύστημα μας. Αντίθετα με τις προσδοκίες μας αυτή η προσθήκη δεν βελτίωσε καθόλου τον αλγόριθμο και παρατηρήσαμε ότι μερικές προτάσεις ήταν παράλογες. Αυτό μας οδήγησε να κατανοήσουμε ότι τα συστήματα συστάσεων είναι

πιο πολύ ”τέχνη” και ότι πολλές φορές το καλύτερο είναι υποκειμενικό. Ο μόνος σίγουρος τρόπος για να δοκιμάσεις τις δυνατότητες του συστήματος σου ειναι τα live A/B Testing που έμεις δυστυχώς δεν είχαμε αυτή την δυνατότητα. Πλέον πορούμε να υπολογίσουμε την ομοιότητα των ταινιών με την ομοιότητα συνημιτόνου.

5.3.3 Ομοιότητα συνημιτόνου

Η ομοιότητα συνημιτόνου μετρά το συνημίτονο της γωνίας μεταξύ δύο μη μηδενικών διανυσμάτων. Αυτή η μέτρηση ομοιότητας αφορά ιδιαίτερα τον προσανατολισμό και όχι το μέγεθος. Εν ολίγοις, δύο διανύσματα που ευθυγραμμίζονται στον ίδιο προσανατολισμό θα έχουν μέτρηση ομοιότητας 1, ενώ δύο διανύσματα ευθυγραμμισμένα κάθετα θα έχουν ομοιότητα 0 [19]. Η ομοιότητα συνημιτόνου είναι μια αναπαράσταση ομοιότητας στον προσανατολισμό Προτιμούμε την ομοιότητα συνημιτόνου απο την Ευκλίδια απόσταση γιατί πολλές φορές η απόσταση δυο διανυσμάτων είναι μεγάλη (λόγο μεγέθους) ενώ η γωνία τους είναι μικρή που ουσιαστικά δείχνει την ομοιότητα τους. Όταν σχεδιάζεται σε έναν πολυδιάστατο χώρο, η ομοιότητα του συνημίτονου καταγράφει τον προσανατολισμό (τη γωνία) των αντικειμένων και όχι το μέγεθος.

Πιο συσκεκριμένα υπολογίζεται η ομοιότητα της ταινίας για την οποία θέλουμε να κάνουμε σύσταση με όλες τις άλλες ταινίες του συνόλου δεδομένων και τελικά προτείνουμε τις πρώτες 10 ταινίες με το μεγαλύτερο βαθμό ομοιότητας.

5.3.4 Συστάσεις

Όταν έχει υπολογιστεί η ομοιότητα με όλες τις άλλες ταινίες τότε επιστρέφεται μια ταξινομημένη λίστα με τις 10 ταινίες που είναι πιο όμοιες με την ταινία που ζητήσαμε σύσταση. Για να εμφανίσουμε τα αποτελέσματα στην ιστοσελίδα μας δημιουργήσαμε τη συνάρτηση ”recommendations” όπου χρησιμοποιεί αυτή τη λίστα για να πάρει τις κατάλληλες πληροφορίες από το API του IMDB. Περισσότερα για την εμφάνιση των συστάσεων στην ιστοσελίδα θα αναφερθούν στο παρακάτω κεφάλαιο. Ακολουθούν τα Σχήματα 5.1 και 5.2 από τις συστάσεις του αλγορίθμου για τη ταινία ”Star Trek” και για την ταινία ”The Godfather”.

```
47                      Star Trek Into Darkness
2317       Star Trek III: The Search for Spock
56                             Star Trek Beyond
52               Transformers: Dark of the Moon
228                                    Oblivion
507                           Independence Day
1750    Star Trek VI: The Undiscovered Country
2815          Star Trek II: The Wrath of Khan
242                              Fantastic Four
1959              Star Trek IV: The Voyage Home
Name: title, dtype: object
```

Σχήμα 5.1: Προτάσεις του αλγόριθμου για την ταινία Star Trek

```
867     The Godfather: Part III
2731     The Godfather: Part II
2649          The Son of No One
1525             Apocalypse Now
4124          This Thing of Ours
1018             The Cotton Club
1170    The Talented Mr. Ripley
1209              The Rainmaker
1394              Donnie Brasco
1850                   Scarface
Name: title, dtype: object
```

Σχήμα 5.2: Προτάσεις του αλγόριθμου για την ταινία The Godfather

Παρακάτω θα παραθέσουμε και ένα στιγμιότυπο οθόνης από τις συστάσεις του αλγορίθμου με διαφορετικά ειδικά βάρη. Αυξάνουμε το βάρος των ηθοποιών και του είδους της ταινίας σε δύο έτσι ώστε να γίνει κατανοητό το πόση μεγαλή διαφορά μπορεί να προκύψει στα αποτελέσματα μόνο με την αλλαγή του ειδικού βάρους στα χαρακτηριστικά που χρησιμοποιεί η συνάρτηση createTheMix(x). Ακολουθεί το αποτέλεσμα του αλγορίθμου (Σχήμα 5.3) για την ταινία ”The Godfather”.

```
2649          The Son of No One
1394              Donnie Brasco
801         The Devil's Advocate
1024                 Dick Tracy
867     The Godfather: Part III
2731     The Godfather: Part II
1850                   Scarface
2280                Sea of Love
2792         Glengarry Glen Ross
1525              Apocalypse Now
Name: title, dtype: object
```

Σχήμα 5.3: Προτάσεις του αλγόριθμου με διαφορετικό βάρος για την ταινία The Godfather

Όπως μπορούμε να δούμε από το παράδειγμα παραπάνω αλλάζοντας το ειδικό βάρος μόνο σε δυο χαρακτηριστικά κατα μια μόνο μονάδα οι συστάσεις διαφοροποιήθηκαν σε

πολύ μεγάλο βαθμό. Αρχικά υπάρχουν πέντε διαφορετικές ταινίες στο δεύτερο παράδειγμα. Ακόμα, ταινίες που ήταν και στο πρώτο παράδειγμα παρατηρούμε ότι άλλαξαν σειρά πράγμα που σημαίνει ότι άλλαξε ο βαθμός ομοιότητας τους με την ταινία ”The Godfather”. Αυτό ίσως να φαίνεται ασήμαντο ή μια λεπτομέρια αλλά στην πραγματικότητα η σειρά που εμφανίζονται οι ταινίες στο χρήστη παίζει πολύ σημαντικό ρόλο σε ένα επιτυχημένο σύστημα συστάσεων.

5.4 Αλγόριθμος δύο

Ο δεύτερος αλγόριθμος που δοκιμάσαμε είναι παρόμοιος με αυτόν που αναλύσαμε παραπάνω. Είναι ένα σύστημα συστάσεων το οποιο όταν ο χρήστης αναζητά μια ταινία, θα του προτείνονται δέκα όμοιες ταινίες. Χρησιμοποιούνται οι βιβλιοθήκες pandas και numpy και η ομοιότητα υπολογίζεται με την ομοιότητα συνημιτόνου. Είναι ένα σύστημα που χρησιμοποιεί τους ηθοποιούς, τον σκηνοθέτη, το είδος , την εταιρία και χώρα παραγωγής και τις λέξεις κλειδία από την πλοκή. Η διαφορά του με το προηγούμενο σύστημα είναι ότι αυτό εκμεταλεύται την χώρα που έχει γίνει η παραγωγή και οτι χρησιμοποιεί του πέντε βασικούς ηθοποιούς αντί για τους τρεις που χρησιμοποιούσε το προηγούμενο. Τα δεδομένα μας υπάρχουν με τη μορφή αυστηροποιημένων λιστών, οπότε πρέπει να τα μετατρέψουμε σε ασφαλή και χρησιμοποιήσιμη δομή. Μερικές στήλες του συνόλου δεδομένων είναι σε JSON μορφή οπότε πρέπει να τα μετατρέψουμε και για αυτό χρησιμοποιείται η συνάρτηση convertJson(x). Οι στήλες που χρειάζονται αυτή τη μετατροπή είναι το είδος της ταινίας, οι λέξεις κλειδία, η εταιρία και η χώρα παραγωγής. Στη συνέχεια παίρνουμε τους πέντε πιο σημαντικούς ηθοποιούς από την ταινία μέσω της συνάρτησης getCast(x). Έπειτα θέλουμε να βρούμε ένα τρόπο να επεξεργαστούμε τις λέξεις κλειδιά οι οποιές είναι στα Αγγλικά. Το μοντέλο μας μπορεί να κατανοήσει μόνο αριθμούς, οπότε θα μετατρέψουμε τις λέξεις-κλειδιά σε έναν αραιό πίνακα χρησιμοποιώντας είτε CountVectorizer είτε TfidfVectorizer. Το CountVectorizer μετράει ακριβώς τις λέξεις που εμφανίζονται, οπότε υπάρχουν μεγάλες πιθανότητες να χάσουμε τις σπάνιες λέξεις που θα μπορούσαν να είχαν βοηθήσει στην αποτελεσματική πρόβλεψη του μοντέλου. Έτσι θα χρησιμοποιήσουμε το TfidfVectorizer που μετρά τη συχνότητα των λέξεων και τις ομαλοποιεί .Αυτή είναι και η τεχνική που συνιστάται ως επί το πλείστον. Με αυτό τον τρόπο αχρηστεύονται οι λέξεις χωρίς μεγάλη σημασία όπως το ”like”, ”a”, ”the” , ”will” οι οποίες δέν προσφέρουν στο ειδικό νόημα και ως αποτέλεσμα δημιουργούν

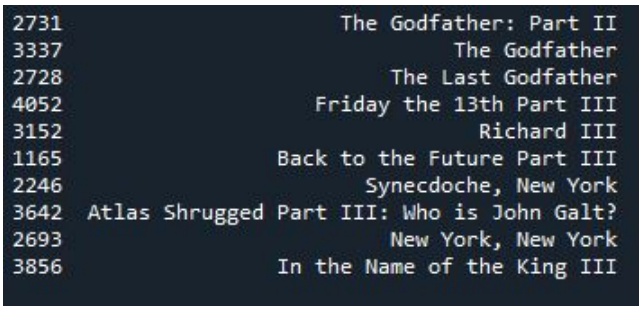
θόρυβο στο συστημά μας. Δοκιμάσαμε στον αλγόριθμο αυτόν να υπολογίζουμε την ομοιότητα μεταξύ των ταινιών με την ευκλείδεια απόσταση αλλά βλέποντας τα αποτελέσματα των συστάσεων καταλάβαμε ότι όταν έχουμε αραιούς πίνακες η ομοιότητα συνημιτόνου δίνει καλύτερα αποτελέσματα οπότε αντικαταστήσαμε την ευκλείδεια απόσταση με αυτή. Στη συνέχεια δημιουργήσαμε την συνάτηση που κάνει τις συστάσεις η οποία ονομάζεται getMovieRecommendation(x) και δέχεται ως όρισμα τον τίτλο της ταινίας για την οποία θέλουμε να δεχτούμε προτάσεις. Είναι ένας αλγόριθμος που προτείνει 10 ταινίες στο χρήστη. Όπως και στον πρώτο αλγόριθμο οι ταινίες που προτείνονται είναι αυτές με το μεγαλύτερο βαθμό ομοιότητας.

5.4.1 Συστάσεις

Όπως μπορούμε να καταλάβουμε το σύστημα ειναι παρόμοιο με το πρώτο με τις ελάχιστες διαφορές τους να εντοπίζονται στον αριθμό των ηθοποιών που αξιοποιούν και οτι χρησιμοποιείται ένα επιπλέον χαρατηριστικό το οποίο είναι η χώρα παραγωγής της ταινίας. Όταν έχει υπολογιστεί η ομοιότητα με όλες τις άλλες ταινίες τότε επιστρέφεται μια ταξινομημένη λίστα με τις 10 ταινίες που είναι πιο όμοιες με την ταινία που ζητήσαμε σύσταση. Ακολουθούν δύο στιγμιότυπα οθόνης (Σχήμα 5.4 και Σχήμα 5.5) από τις συστάσεις του αλγορίθμου για τη ταινία ”Star Trek” και για την ταινία ”The Godfather”.

```
56                          Star Trek Beyond
1959          Star Trek IV: The Voyage Home
1367          Star Trek: The Motion Picture
158                                Star Trek
2317     Star Trek III: The Search for Spock
1583         Star Trek V: The Final Frontier
1750  Star Trek VI: The Undiscovered Country
2815         Star Trek II: The Wrath of Khan
581                  Star Trek: Insurrection
755                      Star Trek: Nemesis
```

Σχήμα 5.4: Προτάσεις του αλγόριθμου με για την ταινία Star Trek



Σχήμα 5.5: Προτάσεις του αλγόριθμου για την ταινία The Godfather

Το σύστημα παρατηρούμε ότι κάνει αρκετά καλές προτάσεις για τις δύο αυτές τις ταινίες που χρησιμοποιήσαμε ως παράδειγμα. Για την ταινία ”Star Trek” βλέπουμε ότι προτείνει να δούμε και τις υπόλοιπες ταινίες της σείρα αλλά και τις παλαιότερες. Το ίδιο συμβαίνει και με την ταινία ”The Godfather”. Ενώ οι συστάσεις είναι αρκετά καλές μας προβλημάτισε το γεγονός ότι για μια σειρά ταινιών δεν έχει την ικανότητα να προτείνει και μερικές άλλες ταινίες. Θεωρήσαμε ότι οι προτάσεις είναι αρκετά προφανής και ότι ο χρήστη μπορεί να δυσκολευτεί μερικές φορές να βρεί μια ταινία που να μην έχει σκεφτεί από μόνος του να παρακολουθήσει. Αυτός είναι και ο λόγος που προτιμήσαμε το πρώτο σύστημα συστάσεων για την ιστοσελίδα μας.

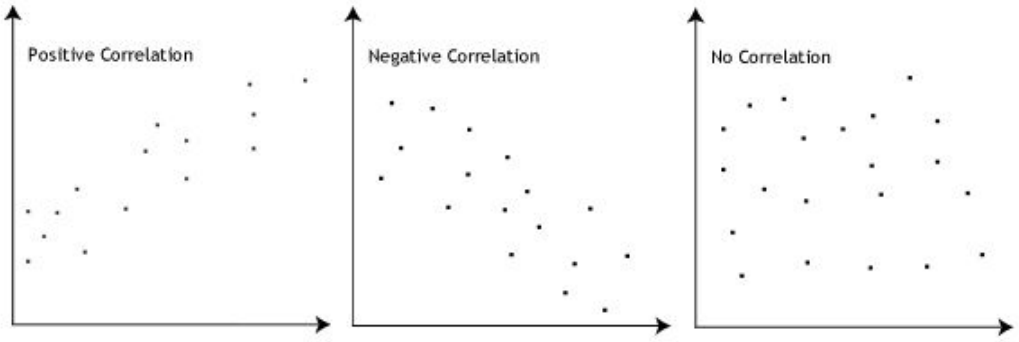
5.5 Αλγόριθμος τρία

Ο αλγόριθμος αυτός είναι πιο προσωποποιημένος από τους δύο προηγούμενους που αναλύσαμε παραπάνω. Είναι ένα αρκετά απλό σύστημα συστάσεων που κάνει τις προτάσεις του με βάση την βαθμολογία μιας ταινίας. Είναι πιο προσωποποιημένο σύστημα γιατί δεν προτείνει ταινίες στο χρήστη που έχει ξαναδεί στο παρελθόν έτσι ώστε οι χρήστες να βρίσκουν συνεχώς καινούργιο υλικό και να μην αναλώνονται σε κουραστικές και χρονοβόρες αναζητήσεις. Μια ακόμα διαφορά του με τους προηγούμενους είναι ότι υπολογίζει την ομοιότητα μεταξύ των ταινιών με τη συσχέτιση του Pearson.

5.5.1 Η συσχέτιση του Pearson

Ο συντελεστής συσχέτισης Pearson [9] είναι ένα μέτρο της ισχύος μιας γραμμικής συσχέτισης μεταξύ δύο μεταβλητών και δηλώνεται με r. Βασικά, μια συσχέτιση προϊόντος-ροής Pearson προσπαθεί να σχεδιάσει μια γραμμή καλύτερης προσαρμογής μέσω των δεδομένων

δύο μεταβλητών και ο συντελεστής συσχέτισης Pearson, r, δείχνει πόσο μακριά όλα αυτά τα σημεία δεδομένων βρίσκονται σε αυτήν τη γραμμή βέλτιστης προσαρμογής (δηλαδή, πόσο καλά τα σημεία δεδομένων ταιριάζουν σε αυτό το νέο μοντέλο / γραμμή βέλτιστης εφαρμογής. Ο συντελεστής συσχέτισης Pearson, r, μπορεί να πάρει ένα εύρος τιμών από +1 έως -1. Η τιμή 0 δείχνει ότι δεν υπάρχει συσχέτιση μεταξύ των δύο μεταβλητών. Μια τιμή μεγαλύτερη από 0 υποδεικνύει θετική συσχέτιση. Δηλαδή, καθώς αυξάνεται η τιμή μιας μεταβλητής, το ίδιο ισχύει και για την τιμή της άλλης μεταβλητής. Μια τιμή μικρότερη από 0 υποδηλώνει αρνητική συσχέτιση. Δηλαδή, καθώς η τιμή μιας μεταβλητής αυξάνεται, η τιμή της άλλης μεταβλητής μειώνεται. Αυτό φαίνεται στο Σχήμα 5.6.



Σχήμα 5.6: Οι τίμες που μπορεί να πάρει η συσχέτιση Pearson [9]

Όσο ισχυρότερη είναι η συσχέτιση των δύο μεταβλητών, τόσο πιο κοντά θα είναι ο συντελεστής συσχέτισης Pearson, r, είτε στο +1 είτε στο -1 ανάλογα με το αν η σχέση είναι θετική ή αρνητική, αντίστοιχα. Η επίτευξη τιμής +1 ή -1 σημαίνει ότι όλα τα σημεία δεδομένων περιλαμβάνονται στη γραμμή της βέλτιστης εφαρμογής - δεν υπάρχουν σημεία δεδομένων που να δείχνουν παραλλαγές μακριά από αυτήν τη γραμμή. Οι τιμές για r μεταξύ +1 και -1 (για παράδειγμα, r = 0,8 ή -0,4) υποδηλώνουν ότι υπάρχει διακύμανση γύρω από τη γραμμή της βέλτιστης εφαρμογής. Όσο πιο κοντά είναι η τιμή του r στο 0, τόσο μεγαλύτερη είναι η διακύμανση γύρω από τη γραμμή της βέλτιστης εφαρμογής.

5.5.2 Ανάλυση συστήματος

Αφού έχουμε πλέον όλοι μια ίδεα για τη συσχέτιση Pearson μπορουμε να αναλύσουμε με μεγαλύτερη ευκολία το τρίτο σύστημα συστάσεων που δοκιμάσαμε. Όπως έχουμε ήδη αναφέρει αυτός ο αλγόριθμος κάνει συστάσεις ταινιών με βάση τη βαθμολογία τους. Για να το πετύχουμε αυτό δημιουρούμε ένα πίνακα με τους χρήστες να είναι οι γραμμές, τις ταινίες

να είναι οι στήλες και με τιμές τη βαθμολογία του χρήστη για κάθε ταινία. Στη συνέχεια διώχνουμε τις NaN (τιμές που δεν υπάρχουν) βαθμολογίες. Το επόμενο βήμα είναι να δημιουργήσουμε τη συνάρτηση υπολογισμού της ομοιότητας με βάση τη συσχέτιση Pearson. Τέλος φτιάξαμε την συνάρτηση recommend η οποία δέχεται ως ορίσματα το τίτλο της ταινίας, τον πίνακα που αναφέραμε παραπάνω και τον αριθμό των συστάσεων που θέλουμε να πραγματοποιήσει. Για να μην προτείνει ταινίες που ο χρήστης έχει παρακολουθήσει στο παρελθόν δημιουργήσαμε την συνάρτηση notWatchedRec η οποία αφαιρεί τις ταινίες που θα προτείνονταν στο χρήστη αν τις έχει δει στο παρελθόν.

5.5.3 Συστάσεις

Πρόκειται για έναν αρκετά απλό αλγόριθμο που έχει την ικανότητα να σου προτείνει συνεχώς διαφορετικές ταινίες εφόσον βέβαια ενημερώνεις το προφίλ σου για τις ταινίες που έχεις ήδη παρακολουθήσει. Παρακάτω παραθέτουμε τα Σχήματα 5.7 και 5.8 από τις συστάσεις του συστήματος για τις ταινίες ”The GodFather” και ”Iron Man” αντίστοιχα.

```
('The Life Aquatic with Steve Zissou', 0.15455567090339464)
('Poltergeist', 0.14794120778157321)
('Face/Off', 0.1443299810701065)
('High Fidelity', 0.13840014671474227)
('The Silence of the Lambs', 0.1321740835023221)
('Predator 2', 0.12674693952287186)
('Harold and Maude', 0.12175638865398528)
('The Bridges of Madison County', 0.11215831479255829)
('Eternal Sunshine of the Spotless Mind', 0.11147807424894121)
('Forrest Gump', 0.10913621389688172)
```

Σχήμα 5.7: Προτάσεις του αλγόριθμου για την ταινία The Godfather

```
('Escape from the Planet of the Apes', 0.08413230890416898)
('Stuck on You', 0.07840032136977679)
('Sneakers', 0.07631763813771965)
('Rent', 0.07525349248350914)
('Abraham', 0.06753967983198243)
('Clean, Shaven', 0.06631340700109448)
('Secret Window', 0.06584136684862986)
('When Saturday Comes', 0.06513810906976945)
('The Next Best Thing', 0.06444608595625413)
('The Men', 0.06424462636648709)
```

Σχήμα 5.8: Προτάσεις του αλγόριθμου για την ταινία Iron Man

Βλέπουμε ότι οι συστάσεις του συστήματος αυτού είναι αρκετά μέτριες και ίσως αδιάφορες. Για την ταινία ”The Godfather” για παράδειγμα δεν προτείνονται οι υπόλοιπες ταινίες που ακολούθησαν αυτή. Προτείνονται ταινίες που έχουν μεγάλη βαθμολογία το οποίο είναι

αναμενόμενο άφου η συγκεκριμένη ταινία είναι μια απο τις πιο υψηλά βαθμολογημένες ταινίες διαχρονικά. Επίσης παρατηρούμε ότι προτείνονται ταινίες διαφορετικού είδους και με τελείως διαφορετικούς ηθοποιούς. Το ίδιο βλέπουμε και από τις προτάσεις που έγιναν για την ταινία ”Iron Man” όπου δεν προτάθηκε καμία άλλη ταινία της εταιρίας παραγωγής ούτε κάποια ταινία που να πρωταγωνιστούν οι ηθοποιοί της εν λόγο ταινίας. Ως συμπέρασμα, καταλαβαίνουμε ότι πρόκειται για ένα σύστημα συστάσεων που δεν θα μπορούσε να χρησιμοποιηθεί σε μια ιστοσελίδα για να βοηθήσει τους χρήστες να βρουν ταινίες της αρεσκίας τους. Για αυτό ακριβώς το λόγο το απορρίψαμε.

Κεφάλαιο 6

Περιγραφή του ιστότοπου

6.1 Ανάλυση των σελίδων της εφαρμογής

Αρχική σελίδα (/)

Στο κέντρο της σελίδας υπάρχουν δύο πεδία για την εισαγωγή στοιχείων και δύο κουμπιά, ένα το login και ένα το register.

Σελίδα εγγραφής (/signup)

Στο κεντρικό μέρος της σελίδας υπάρχουν υποδοχείς κειμένου για της πληροφορίες του χρήστη (Username, Email, First name, Last name, Password, Re-type Password) και από κάτω τους το κουμπί Create account.

Σελίδα αναζήτησης (/rec)

Στο επάνω δεξιά μέρος της σελίδας υπάρχει το κουμπί με το όνομα χρήστη το οποίο είναι ένα αναπτυσσόμενο κουμπί και εμφανίζει τα κουμπιά Profile, Home και Log out. Στο επάνω αριστερά μέρος υπάρχει το κουμπί Get recommendations. Στην κορυφή της σελίδας βρίσκεται το κουμπί Visit our blog. Στο κέντρο της σελίδας υπάρχει μια μπάρα αναζήτησης για ταινίες. Στο κάτω μέρος της σελίδας υπάρχουν τέσσερα κουτιά κειμένου που περιγράφουν τις δυνατότες της εφαρμογής.

Σελίδα προφίλ (/"username")

Στο πάνω μέρος της σελίδας υπάρχει η επιλεγμένη εικόνα προφίλ του χρήστη. Κάτω αριστερά υπάρχει ένας πίνακας με τις προσωπικές πληροφορίες του χρήστη. Και κάτω δεξιά οι λίστες Movies Liked, Movies in watchlist, Search history με τα δικά τους κουμπία Show list.

Επεξεργασία προφίλ (/profile/edit)

Επάνω αριστερά υπάρχει το κουμπί File. Από κάτω υπάρχουν υποδοχείς κειμένου (First name, Last name, Location, URL, About). Στο κάτω μέρος υπάρχει το κουμπί Update.

Σελίδα προτάσεων (/rec/recommendation)

Στο πάνω αριστερά μέρος υπάρχει ένα αναπτυσσόμενο μενού που περιέχει δεκαεπτά υποκουμπιά για κάθε είδος ταινίας. Στο κεντρικό μέρος υπάρχει μια μπάρα για την εισαγωγή της ταινίας που θέλει συστάσεις ο χρήστης, καθώς και τα κουμπία Search for similar movies και Return to home. Το κάτω μέρος της σελίδας είναι όμοιο με την σελίδα αναζήτησης.

Πληροφορίες ταινίας (/rec/"movieid")

Στο πάνω μέρος της σελίδας υπάρχουν οι λεπτομέριες της ταινίας μαζί με την αφίσα της. Κάτω από τις πληροφορίες υπάρχουν τα κουμπία rate, watchlist, liked, return to home, visit imdb. Στο κάτω μέρος υπάρχουν αξιολογήσεις που αφορούν την ταινία.

Αξιολόγηση ταινίας (rec/"movieid"/rate)

Στο κέντρο της σελίδας υπάρχει το όνομα και η αφίσα της ταινίας καθώς και η σύντομη πλοκή της. Από κάτω υπάρχει ένας υποδοχέας κειμένου και το κουμπί Rate.

Σχόλια στις αξιολογήσεις ("username"/review/"movieid")

Στο πάνω μέρος βλέπουμε το προφίλ του χρήστη που έκανε την αξιολόγηση. Στο επόμενον επόμενο τομέα υπάρχει η αξιολόγηση καθώς και η ημερομηνία και η ώρα που έγινε. Από κάτω είναι ο τομέας των σχολίων με έναν υποδοχέα κειμένου και το κουμπί Send.

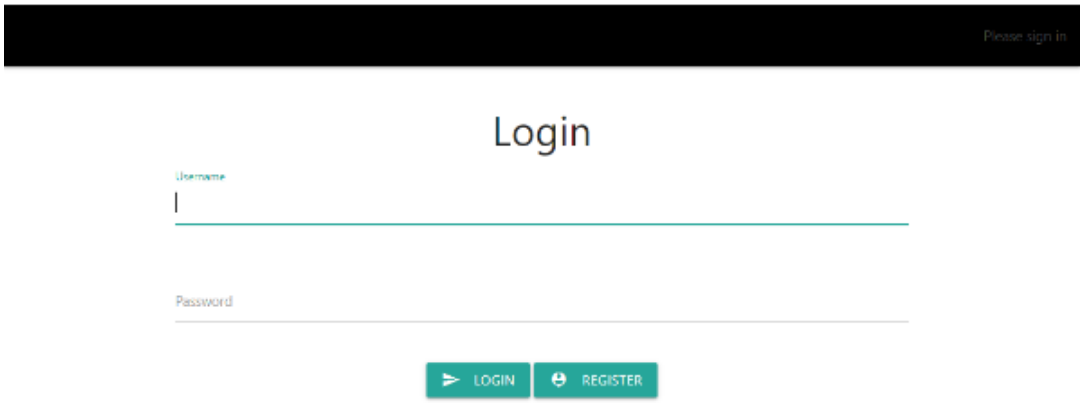
Σελίδα του blog (/blog)

Επάνω αριστερά υπάρχει το κουμπί CinemaBlend που ανακατευθύνει τον χρήστη στην σελίδα CinemaBlend όπου μπορεί να ενημερωθεί για τα τελευταία νέα στο χώρο των ταινίων και των τηλεοπτικών σειρών. Στο κέντρο της σελίδας υπάρχουν σύνδεσμοι-τίτλοι για άρθρα που ανεβαίνουν από τους διαχειριστές της σελίδας και όταν κλικάρεις πάνω τους σε οδηγούν στην σελίδα /article ”number of article”.

6.2 Ανάλυση της λειτουργικότας της σελίδας με στιγμιότυπα οθόνης

Login

Πρόκειται για την πρώτη σελίδα που θα αντικρίσει ο χρήστης καθώς για τις λειτουργίες της εφαρμογής χρειάζεται να δημιουργήσει ή να συνδεθεί στον λογαριασμό του (Σχήμα 6.1).



Σχήμα 6.1: Login page

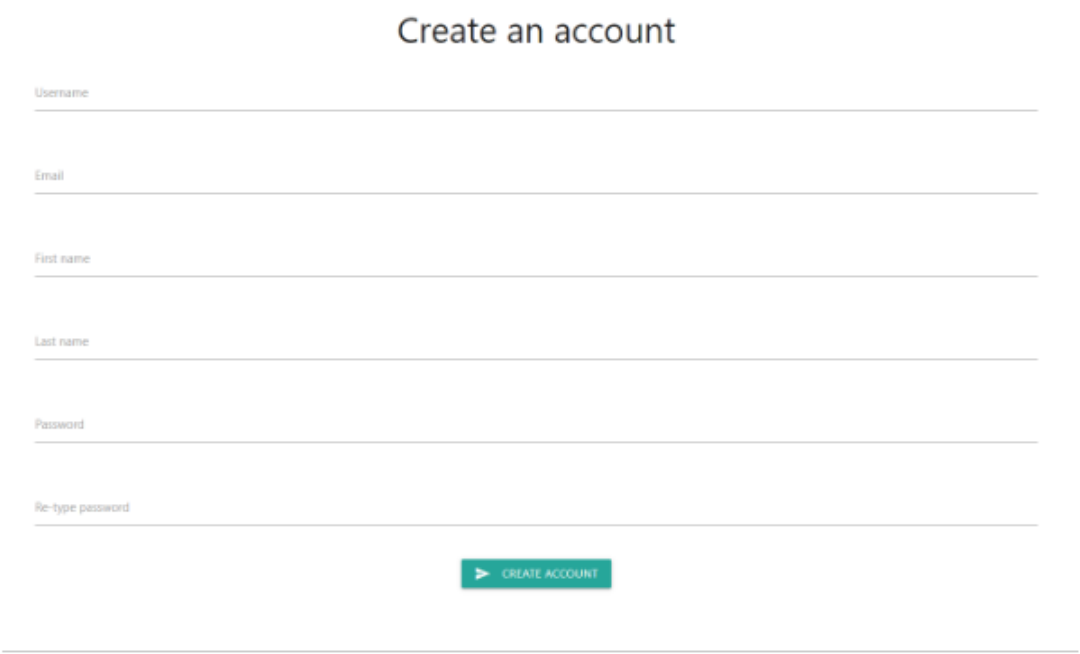
Register

Στην σελίδα του register ο χρήστης πραγματοποιεί την εγγραφή του στο site δίνοντας τα ακόλουθα στοιχεία:

- Username

- Email (Πρέπει να τηρεί το σωστό format ενός email αλλίως δεν το δέχεται. πχ someone@sth.com/gr)
- First name
- Last name
- Password
- Re-type Password

Έπειτα, πατώντας το κουμπί Create account δημιουργεί τον λογαριασμό του και τον ανακατευθύνει στην σελίδα σύνδεσης (Σχήμα 6.2).

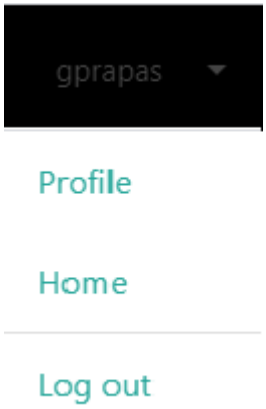


Σχήμα 6.2: Register page

Home page

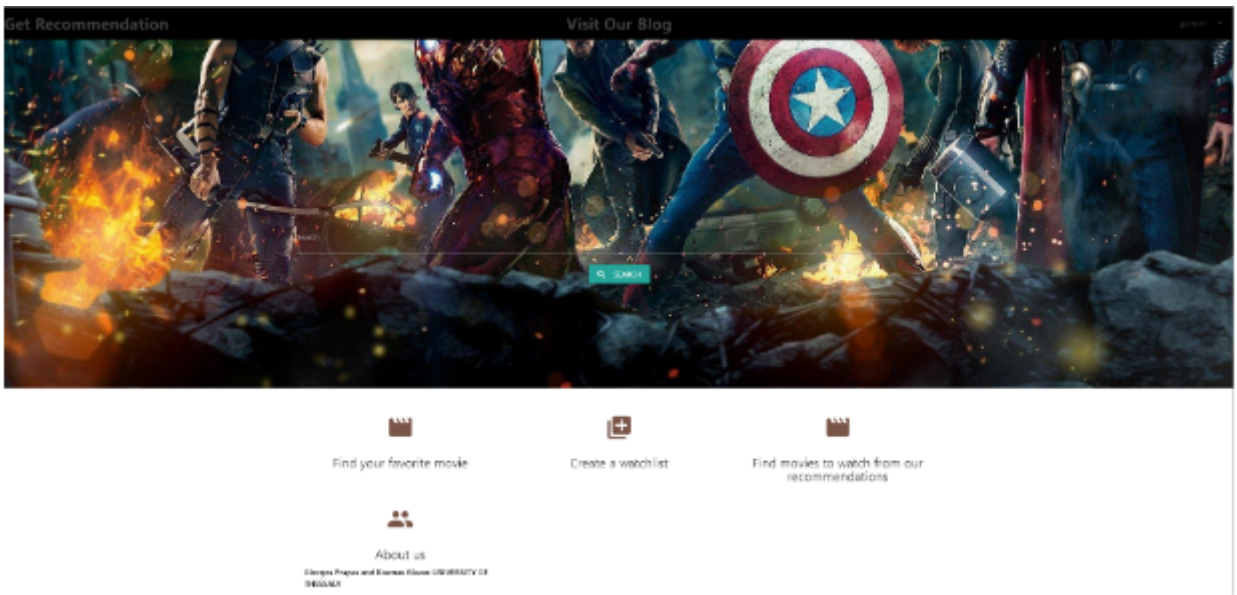
Ξεκινώντας από πάνω αριστερά υπάρχει το κουμπί Get Recommendation που οδηγεί τον χρήστη στην σελίδα /rec/recommendation όπου μπορεί να βρει συστάσεις για μια ταινία της αρεσκίας του. Συνεχίζοντας, στο κέντρο της κορυφής το κουμπί Visit our blog κατευθύνει τον χρήστη στην σελιδα /blog για να διαβάσει τα άρθρα των διαχειριστών της σελίδας. Επάνω

δεξιά υπάρχει το κουμπί με το όνομα του χρήστη το οποίο του δίνει την επιλογή να επισκεφτεί το προφίλ του, να πάει στην αρχική σελίδα ή να αποσυνδεθεί (Σχήμα 6.3).



Σχήμα 6.3: Dropdown button

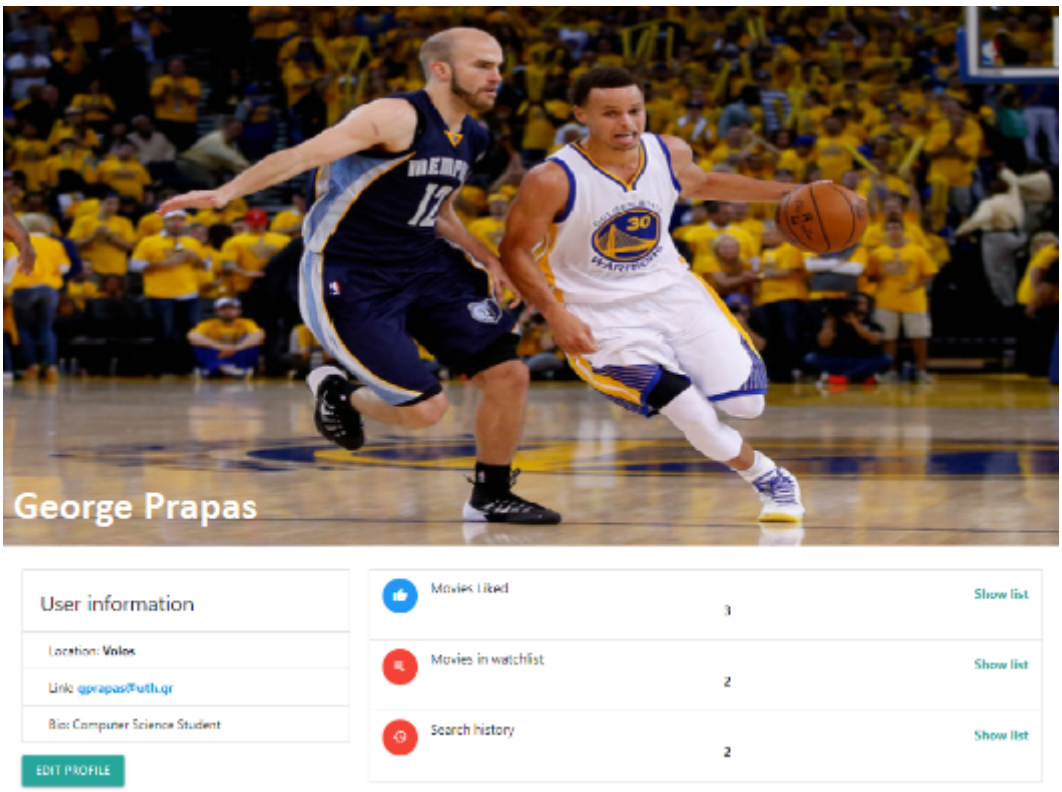
Στο κέντρο της αρχικής σελίδας υπάρχει μια μπάρα αναζήτησης όπου ο χρήστης μπορεί να ψάξει για ταινίες που ενδιαφέρεται. Στο κάτω μέρος υπάρχουν πληροφορίες για το τι μπορεί να κάνει ο χρήστης χρησιμοποιώντας την σελίδα μας καθώς και μερικά λόγια για τους διαχειριστές (Σχήμα 6.4).



Σχήμα 6.4: Home page

Profile page

Πρόκειται για την σελίδα προφίλ του χρήστη (Σχήμα 6.5). Στο πάνω μέρος φαίνεται η φωτογραφία προφίλ του χρήστη με το ονοματεπώνυμο του. Κάτω αριστερά υπάρχουν επιπρόσθετες πληροφορίες για τον χρήστη (Location, Link, Bio) τις οποίες μπορεί να επεξεργαστεί κλικάροντας στο κουμπί Edit profile που υπάρχει. Δίπλα από τις πληροφορίες υπάρχουν οι τρεις λίστες, Movies liked, Movies in watchlist, Search history με έναν μετρητή στο τέλος του ονόματος τους έτσι ώστε ο χρήστης να μπορεί να δει πόσες ταινίες έχει αποθηκεύσει στην κάθε λίστα. Επιπλέον πατώντας στο κουμπί Show list μπορεί να δει και ποιες είναι αυτές οι ταινίες.



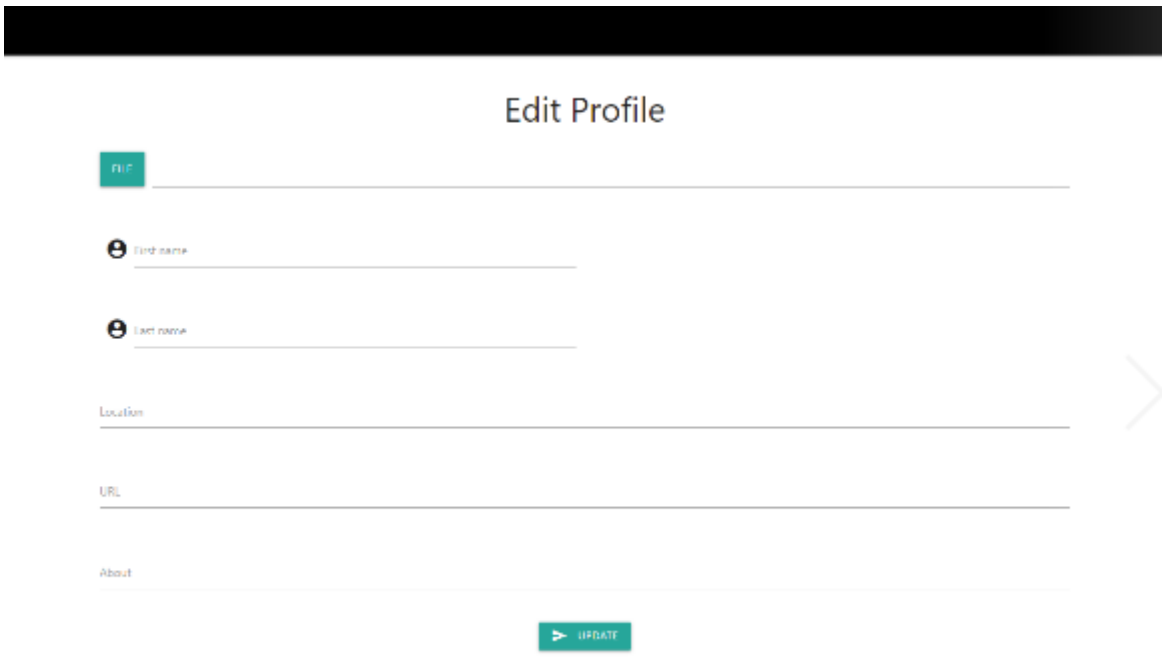
Σχήμα 6.5: Profile page

Edit profile page

Σε αυτή τη σελίδα (Σχήμα 6.6) ο χρήστης μπορεί να επεξεργαστεί τα εξής στοιχεία του προφίλ του:

- Εικόνα προφίλ με το κουμπί File (Η εικόνα πρέπει να βρίσκεται σε .jgp format)
- First name

- Last name
- Location
- URL
- About

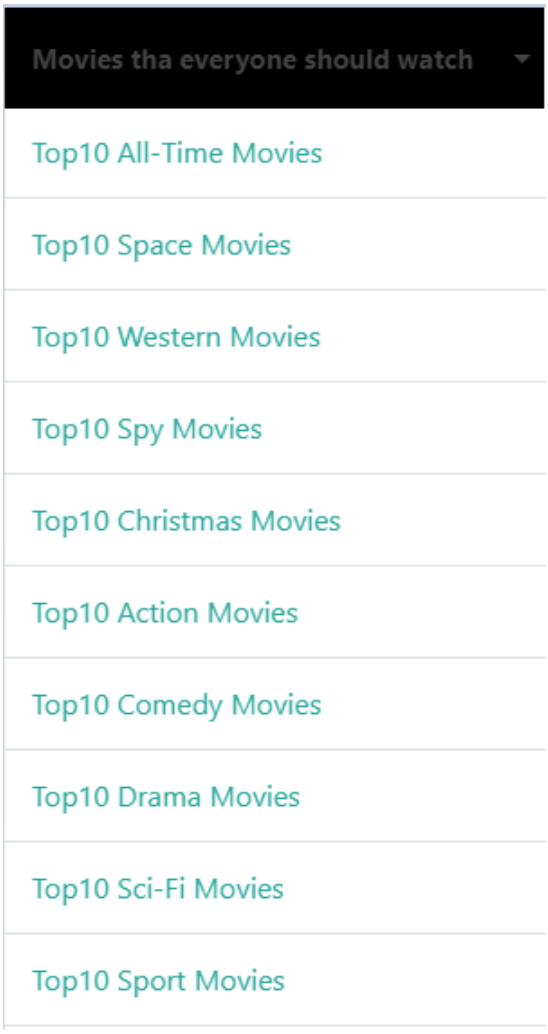


Σχήμα 6.6: Edit profile page

Recommendation page

Σε αυτή τη σελίδα ο χρήστης μπορεί να πάρει συστάσεις με βάση μια ταινία γράφοντας το όνομα της στην μπάρα αναζήτησης που βρίσκεται στο κέντρο της σελίδας και πατώντας το κουμπί Search for similar movies (Σχήμα 6.7). Ακόμα, στο πάνω αριστερά μέρος της σελίδας υπάρχει ένα dropdown κουμπί με τις κορυφαίες δέκα ταινίες κάθε είδους σε περίπτωση που θέλει μια γρηγορότερη επιλογή (Σχήμα 6.8). Τέλος με το κουμπί Return to home μπορεί να επιστρέψει στην αρχική σελίδα.

Σχήμα 6.7: Recommendation page



Σχήμα 6.8: Μερικές από τις 17 λίστες των κορυφαίων ειδών

Movie details page

Περιέχει όλες τις διαθέσιμες πληροφορίες για την ταινία (Σχήμα 6.9). Αυτές είναι:

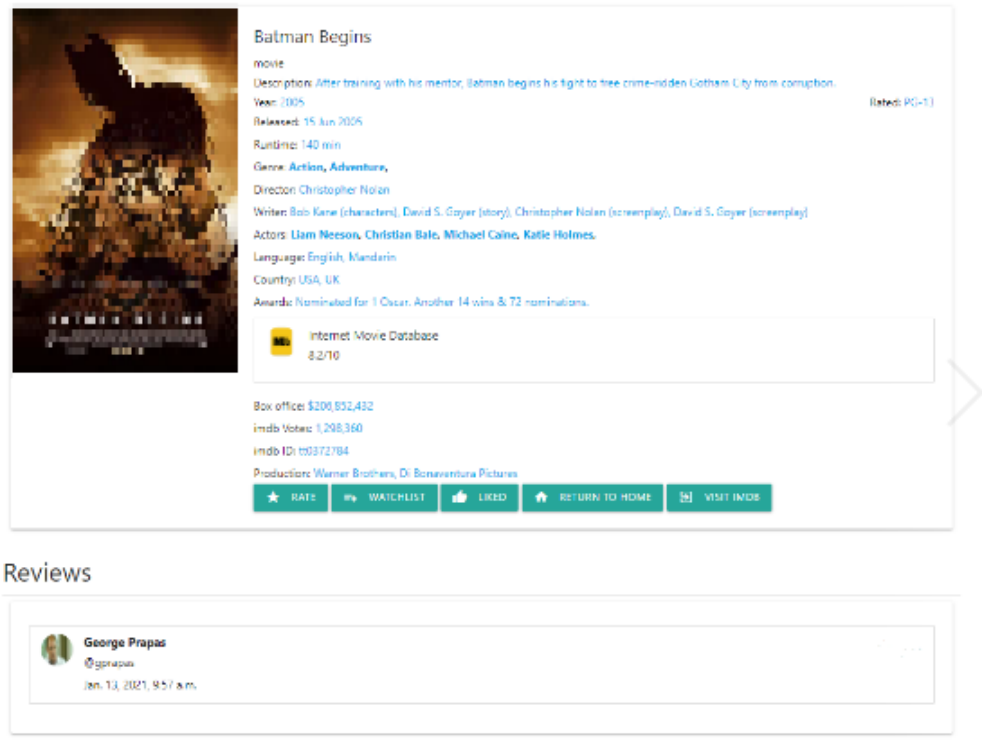
- Η αφίσα της ταινίας
- Τον τίτλο
- Την περιγραφή της ταινίας
- Την χρονιά κυκλοφορίας
- Την διάρκεια
- Το είδος
- Τον σκηνοθέτη
- Τον σεναριογράφο
- Τους πρωταγωνιστές
- Την γλώσσα
- Την χώρα παραγωγής
- Τα βραβεία
- Την βαθμολογία στο imdb
- Το box office (εάν είναι διαθέσιμο)
- Το imdb id της ταινίας
- Την παραγωγή

Ο χρήστης στη συνέχεια έχει τις εξής επιλογές για την συγκεκριμένη ταινία:

- Το κουμπί Rate, ώστε να αξιολογήσει την ταινία
- Το κουμπί Watchlist, που προσθέτει την ταινία στην λίστα για παρακολούθηση αργότερα

- Το κουμπί Liked, που προσθέτει την ταινία στην λίστα με τις αγαπημένες ταινίες του χρήστη

- Το κουμπί Visit imdb που σε ανακατευθύνει στην σελίδα της ταινίας στο imdb για περισσότερες λεπτομέρειες και για να μπορείς να βλέπεις το trailer της ταινίας.



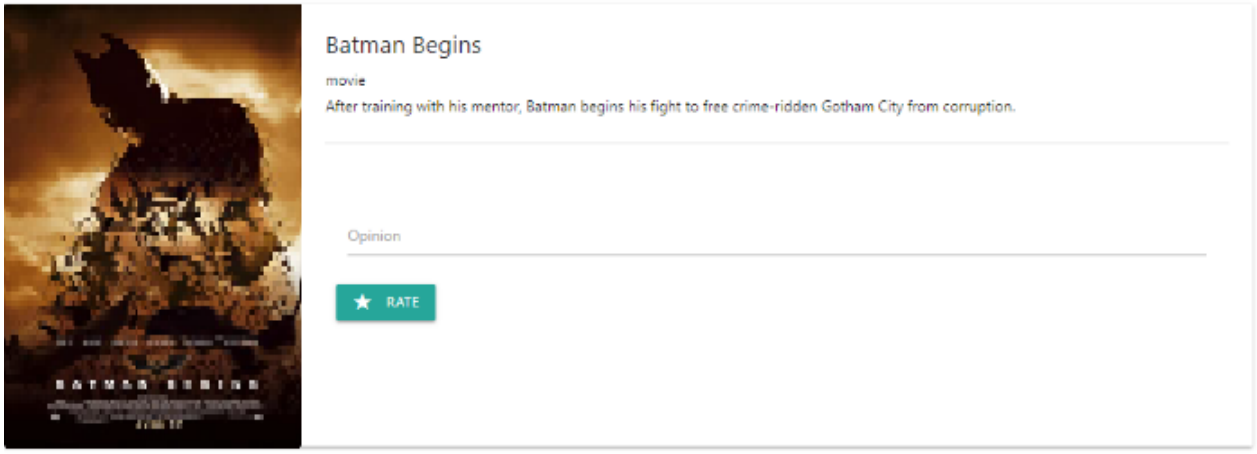
Σχήμα 6.9: Movie details [10]

Ακόμα, στο κάτω μέρος της σελίδας υπάρχει το κομμάτι των Reviews, όπου εμφανίζονται οι αξιολογήσεις των χρηστών. Ο χρήστης μπορεί να κλικάρει σε μια αξιολόγηση και να οδηγηθεί στην σελίδα των comments όπου μπορεί να ανταλλάξει μηνύματα για την ταινία με άλλους χρήστες. Τέλος τα ονόματα των ηθοποιών και το είδος της ταινίας είναι γραμμένα σε έντονη γραφή δίοτι πρόκειται για υπερσυνδέσμους που ανακατεθευθύνουν τον χρήστη σε άλλες ταινίες των συγκεκριμένων ηθοποιών ή άλλες ταινίες του ίδιου είδους.

Movie rate

Στην σελίδα αυτή ο χρήστης μπορεί να αξιολογήσει την ταινία, γράφοντας στο λευκό κουτί την γνώμη του. Έπειτα πατώντας το κουμπί Rate αποθηκεύεται στην βάση και τον

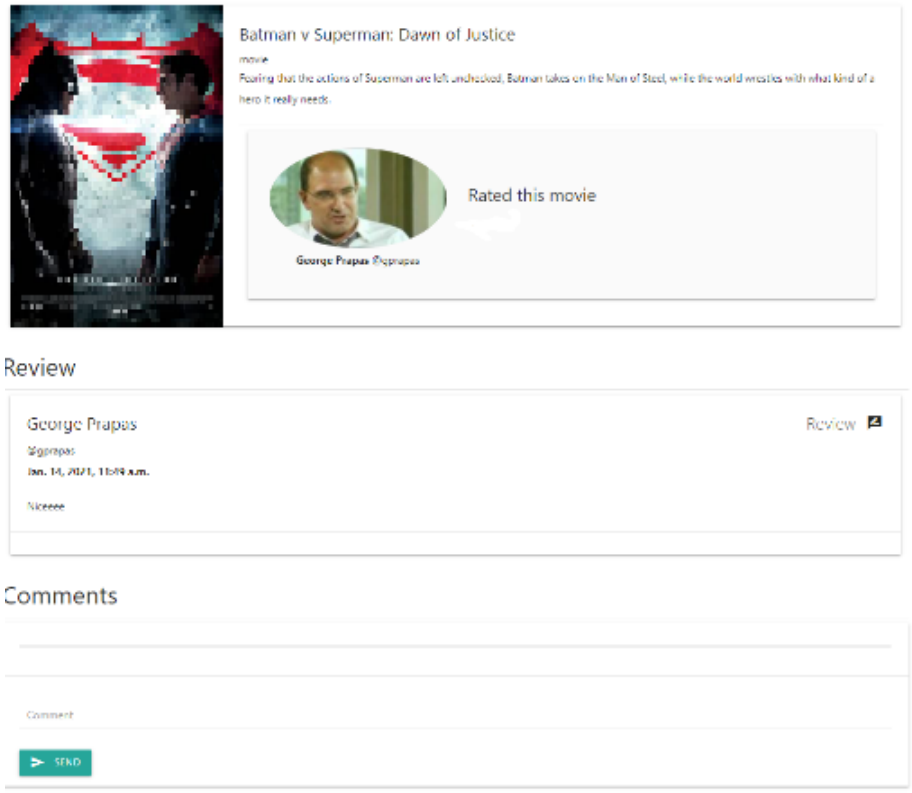
ανακατευθύνει στην σελίδα με της πληροφορίες της ταινίας όπου μπορεί στο κάτω μέρος να δει την αξιολόγηση του (Σχήμα 6.10).



Σχήμα 6.10: Movie rate [10]

Comments section

Εδώ ο χρήστης μπορεί να δει την αξιολόγηση κάποιου άλλου χρήστη(ή ακόμη και την δική του), να προσθέσει σχόλια και να κουβεντιάσει με άλλους χρήστες για μια ταινία (Σχήμα 6.11).



Σχήμα 6.11: Comments page [10]

Blog

Εδώ ο χρήστης μπορεί να διαβάσει άρθρα για ταινίες και σειρές που έχει γράψει ο διαχειριστής του ιστότοπου που τον ενδιαφέρουν. Κλικάροντας πάνω στον τίτλο του άρθρου οδηγείται στη σελίδα /article όπου υπάρχει το κείμενο που θέλει να διαβάσει (Σχήματα 6.12 και 6.13).

Σχήμα 6.12: Blog page

Σχήμα 6.13: Article page

Κεφάλαιο 7

Ανάλυση κώδικα και βάσης δεδομένων

Σε αυτό το κεφάλαιο θα αναλύσουμε μερίκες απο τις βασικές συναρτήσεις που δημιουργήσαμε για τον ιστότοπο μας, θα αναφέρουμε πως αντλούμε τα δεδομένα μας για τις ταινίες, θα σας παρουσιάσουμε τη βάση δεδομένων που στηρίζει τον ιστότοπο και από που βρήκαμε το πρότυπο (template) και όλα τα βασικά γραφικά (π.χ. κουμπιά) που βλέπουμε στις παραπάνω φωτογραφίες.

7.1 Γραφικά

Τα βασικά συστατικά όπως και το πρότυπο (template) τα πήραμε από το Materialize [20]. Είναι ένας ιστότοπος που δημιουργήθηκε και σχεδιάστηκε από την Google. Το Material Design είναι μια σχεδιαστική γλώσσα που συνδυάζει τις κλασικές αρχές του επιτυχημένου σχεδιασμού μαζί με την καινοτομία και την τεχνολογία. Στόχος της Google είναι να αναπτύξει ένα σύστημα σχεδίασης που επιτρέπει μια ενοποιημένη εμπειρία χρήστη σε όλα τα προϊόντα τους σε οποιαδήποτε πλατφόρμα. Το πρότυπο που χρησιμοποιήσαμε ονομάζεται Parallax Template και είναι μια πολύ απλή σελίδα με ελάχιστα χαρακτηριστικά. Ακόμα χρησιμοποιήσαμε τα κουμπιά που προσφέρουν, κάρτες για να μπορούμε να εμφανίζουμε τις ταινίες μας, και διαδραστικά εικονίδια. Αφού έχουμε πλέον αναφέρει πως κάναμε τη σχεδίαση του ιστότοπου μπορούμε να συνεχίσουμε με την ανάλυση του κώδικα βασικών κομματιών της εφαρμογής.

7.2 Εφαρμογές

Το Django [21] περιέχει ένα μητρώο εγκατεστημένων εφαρμογών που αποθηκεύει τη διαμόρφωση και παρέχει ενδοσκόπηση. Διατηρεί επίσης μια λίστα διαθέσιμων μοντέλων. Ο όρος εφαρμογή περιγράφει ένα πακέτο Python που παρέχει κάποια σειρά δυνατοτήτων. Οι εφαρμογές μπορούν να επαναχρησιμοποιηθούν σε διάφορα έργα. Οι εφαρμογές περιλαμβάνουν κάποιο συνδυασμό από μοντέλα (models), προβολές (views), πρότυπα (templates), ετικέτες προτύπων (template tags), στατικά αρχεία (static files),διευθύνσεις (URLs), ενδιάμεσο υλικό (middleware) κ.α.. Γενικά συνδέονται σε έργα με τη ρύθμιση INSTALLED APPS και προαιρετικά με άλλους μηχανισμούς όπως το URLconfs, τη ρύθμιση MIDDLEWARE ή την κληρονομιά προτύπου . Είναι σημαντικό να κατανοήσουμε ότι μια εφαρμογή Django είναι ένα σύνολο κώδικα που αλληλεπιδρά με διάφορα μέρη του πλαισίου. Δεν υπάρχει αντικείμενο εφαρμογής. Ωστόσο, υπάρχουν μερικά μέρη όπου το Django πρέπει να αλληλεπιδρά με εγκατεστημένες εφαρμογές, κυρίως για διαμόρφωση και επίσης για ενδοσκόπηση. Γι ’αυτό το μητρώο εφαρμογών διατηρεί μεταδεδομένα σε μια παρουσία AppConfig για κάθε εγκατεστημένη εφαρμογή.

7.2.1 Actor

Η εφαρμογή αυτή δημιουργήθηκε για να συλλέγει πληροφορίες σχετικά με τους ηθοποιούς των ταινιών που αναζητά ο χρήστης. Το μοντέλο (model) που χρησιμοποιεί αποθηκεύει το όνομα του ηθοποιού, μια φωτογραφία του και τις ταινίες στις οποίες έχει παίξει. Επίσης έχουμε φτιάξει μια συνάρτηση στις προβολές (views) η οποία ονομάζεται actors info και ουσιαστικά στέλνει τις απαιτούμενες πληροφορίες για τον εκάστοτε ηθοποιό. Αυτές οι πληροφορίες εμφανίζονται στη σελίδα του κάθε ηθοποιού (/actor/όνομα ηθοποιού).

7.2.2 Authent

Χρησιμοποιούμε την εφαρμογή Authent για να μπορούμε να δημιουργούμε προφίλ για τους χρήστες μας. Το μοντέλο (model) αποτελείται από μια κλάση που ονομάζεται Profile. Τα στοιχεία που αποθηκεύει για κάθε χρήστη είναι το όνομα, το επίθετο, η τοποθεσία, το url, μερικές πληροφορίες για αυτόν και μια φωτογραφία. Ακόμα αποθηκευεί ολες τις ταινίες που έχει αναζητήσει ο χρήστης . Σε αυτή την εφαρμογή χρησιμοποιούμε φόρμες σε HTML. Μια φόρμα είναι μια συλλογή στοιχείων μέσα στο <form> ... </form> που επιτρέπουν στον χρή-

στη να κάνει πράγματα όπως εισαγωγή κειμένου, επιλογή στοιχείων, χειρισμό αντικειμένων ή στοιχείων ελέγχου και ούτω καθεξής και στη συνέχεια να στείλει αυτές τις πληροφορίες πίσω στον διακομιστή. Η λειτουργικότητα της φόρμας του Django μπορεί να απλοποιήσει και να αυτοματοποιήσει τεράστια τμήματα ενός έργου και μπορεί επίσης να το κάνει με μεγαλύτερη ασφάλεια από ό, τι οι περισσότεροι προγραμματιστές θα μπορούσαν να κάνουν σε κώδικα που έγραψαν οι ίδιοι. Πιο συγκεκριμένα δημιουργήσαμε δύο κλάσεις-φόρμες την SignupForm και την EditProfileForm. Στη πρώτη τα στοιχεία που ζητάει να συμπληρωθούν για να υπάρξει επιτυχής σύνδεση του χρήστη στον ιστότοπο είναι το όνομα χρήστη , η ηλεκτρονική διεύθυνση ,το πραγματικό του όνομα και επίθετο , ένας κωδικός πρόσβασης και η επαλήθευση του κωδικού αυτού. Για το πεδίο του κωδικού χρησιμοποιήσαμε το forms.PasswordInput όπου ουσιαστικά κρύβει τον κωδικό του χρήστη. Ακόμα για το όνομα χρήστη έχουμε απαγορεύσει την χρησιμοποιήση ορισμένων λέξεων όπως για παράδειγμα το ”admin”, το ”login”, το ”email” κ.α.. Τέλος έχουμε κάνει δύο συναρτήσεις που αποτρέπουν την δημιουργία δύο όμοιων προφίλ (πρέπει όλοι να έχουν διαφορετικό όνομα χρήστη και διαφορετική ηλεκτρονική διεύθυνση). Η δεύτερη φόρμα επιτρέπει την επεξεργασία των προφίλ. Χάρη σε αυτή, ο χρήστης έχει τη δυνατότητα να προσθέσει παραπάνω πληροφορίες για τον ίδιο όπως μια φωτογραφία, λίγα λόγια για τον χαρακτήρα του και τις ταινίες που τον ενδιαφέρουν, την τοποθεσία του και το url του. Εφόσον αναλύσαμε τις φόρμες της εφαρμογής μπορούμε να εξηγήσουμε και τις βασικές της συναρτήσεις στις προβολές (Views). Για να χρησιμοποιήσει κάποιος αυτές τις συναρτήσεις θα πρέπει να έχει ήδη συνδεθεί στο προφίλ του. Η πρώτη συνάρτηση που θα αναλύσουμε είναι η UserProfileWatchList η οποία είναι υπεύθυνη για να εμφανίζει τη λίστα ταινιών που έχει φτιάξει ο κάθε χρήστης. Ουσιαστικά κάθε φορά που ένας χρήστης κλικάρει το κουμπί watchlist στο movie details ενεργοποιείται αυτή η συνάρτηση προσθέτοντας την ταινία στην αντίστοιχη λίστα της βάσης δεδομένων του ιστότοπου με τη σωστή σείρα και αυξάνει τον μετρητή του showlist κατά ένα. Ακριβώς την ίδια λειτουργία έχει και η συνάρτηση UserProfileMoviesLiked με την μόνη διαφορά να είναι ότι αναφέρεται στις ταινίες που έχει δηλώσει ο χρήστης ότι του άρεσαν.

7.2.3 Comments

Η εφαρμογή αυτή χρησιμοποιείται για να μπορούν οι χρήστες να επικοινωνούν μεταξύ τους και να ανταλλάσουν απόψεις για τις ταινίες. Τα Comments αφορούν κάθε ταινία ξεχωριστά. Πιο συγκεκριμένα το μοντέλο (model) της εφαρμογής αποτελείται από μια κλάση

που ονομάζεται chat που αποθηκεύει τη κριτική και τον χρήστη που την έκανε, το κείμενο και την ημερομηνία συγγραφής του. Υπάρχει επιπλέον μια φόρμα που ονομάζεται chatForm και ο σκοπός της είναι να αποθηκεύει το σχόλιο του χρήστη στη βάση έτσι ώστε να γίνεται ορατό στους υπόλοιπους χρήστες.

7.2.4 Movie recommendations

Είναι η βασική εφαρμογή του ιστότοπου που δημιουργήθηκε όταν ξεκινήσαμε την εργασία μας στο Django. Σε αυτή την εφαρμογή περιέχονται τα πρότυπα (templates), οι ρυθμίσεις (settings) και τα βασικά urls του ιστότοπου. Ένα αρχείο ρυθμίσεων Django περιέχει όλες τις ρυθμίσεις της εγκατάστασης του Django. Αυτό το έγγραφο εξηγεί πώς λειτουργούν οι ρυθμίσεις (settings) και ποιες ρυθμίσεις είναι διαθέσιμες. Κάθε φορά που κάνουμε μια νέα εφαρμογή πρέπει να την δηλώνουμε στις ρυθμίσεις (settings) στη λίστα εγκατεστημένες εφαρμογές (INSTALLED APPS). Αν δεν δηλωθεί μια εφαρμογή τότε δεν αναγνωρίζεται και δεν μπορεί να χρησιμοποιηθεί. Ακόμα μια σημαντική διαδικασία που πρέπει να δηλωθεί είναι ο καθορισμός των μονοπατιών των αρχείων που θα χρησιμοποιήσουμε στο έργο μας. Πιο συγκεκριμένα τέτοιου είδους αρχεία είναι τα πρότυπα (templates), οι στατικές διευθύνσεις url και τα media urls. Στο φάκελο static υπάρχουν τα απαραίτητα java scripts για την ομαλή λειτουργία των προτύπων. Τέλος σε αυτό το αρχείο γίνεται η σύνδεση με την βάση δεδομένων δηλώνοντας ποιο DBMS χρησιμοποιούμε καθώς και το μονοπάτι του αρχείου της βάσης.

7.2.5 Movie blog

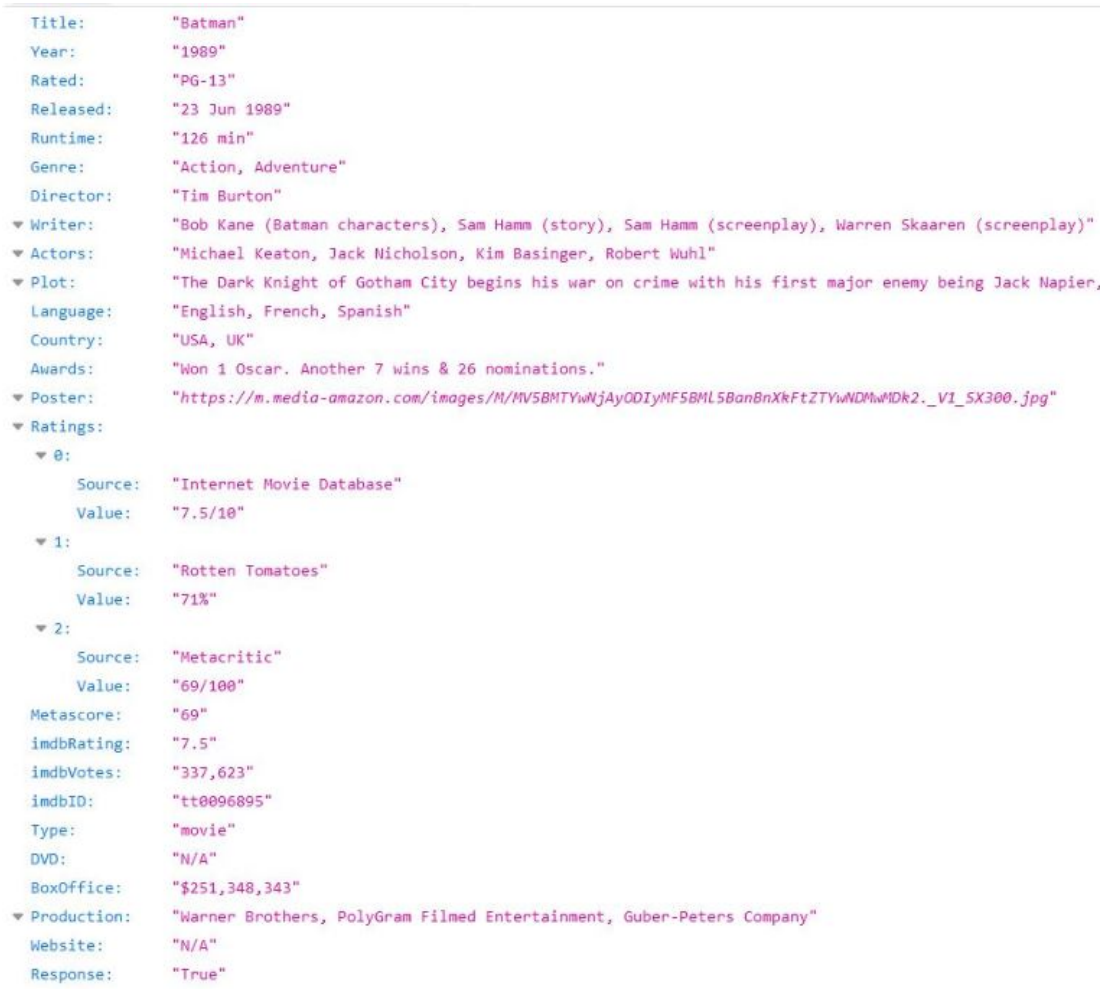
Χάρη σε αυτή την εφαρμογή δημιουργήσαμε ένα ιστολόγιο που δημοσιεύουμε άρθρα για ταινίες και σειρές για να μπορούν οι χρήστες να ενημερώνονται για τις τελευταίες εξελίξεις. Το μοντέλο (model) της εφαρμογής αποτελείται από μόνο μια κλάση την Post που αποθηκεύει τον τίτλο , τον συγγραφέα, το κείμενο και την αφίσα του άρθρου. Υπάρχουν δύο συναρτήσεις το movie blog και το article info που στη πρώτη χρησιμοποιούμε το query List View για να πάρουμε τις βασικές πληροφορίες από τη βάση μας για τα αποθηκευμένα άρθρα και στη δεύτερη χρησιμοποιούμε το Detail View για να πάρουμε όλες τις πληροφορίες για ένα συγκεκριμένο άρθρο. Η κάθε συνάρτηση στέλνει τον χρήστη στη σωστή σελίδα του ιστότοπου movie blog και article info αντίστοιχα.

7.2.6 Rec

Είναι η τελευταία εφαρμογή του ιστότοπου και αυτή που ασχολείται με τις συστάσεις. Το μοντέλο (model) της αποτελείται από τις κλάσεις Movie, Genre, και Review. Στην κλάση Movie πρέπει να αποθηκεύουμε τις πληροφορίες της κάθε ταινίας όπως αυτές στέλνονται από το API του omdb. Παρακάτω ακολουθεί μια φωτογραφία ως παράδειγμα για δεδομένα που στέλνει το API (Σχήμα 7.1).

- Τα πεδία που πρέπει να δημιουργηθούν είναι:
  - Title
  - Year
  - Rated
  - Released
  - Runtime
  - Genre
  - Director
  - Writer
  - Actors
  - Plot
  - Language
  - Country
  - Awards
  - Poster
  - Poster url
  - Ratings
  - Metascore
  - imdbRating
  - imdbVotes
  - imdbID
  - Type
  - DVD
  - Boxoffice
  - Production
  - Website
  - totalSeasons

Στην κλάση Review δημιουργούμε τα πεδία user, movie, date και text έτσι ώστε να μπορούμε να αποθηκεύουμε τις αξιολογήσεις του χρήστη για κάθε ταινία. Αυτή η κλάση συνδέεται με την φόρμα RateForm που είναι υπεύθυνη για την μεταφορά της αξιολόγησης του χρήστη στην βάση μας. Όσον αφορά στις συναρτήσεις της εφαρμογής, οι βασικότερες συναρτήσεις που δημιουργήσαμε είναι οι index, recommendations και movie information. Ξεκινώντας με την index, ο σκοπός της είναι να εμφανίζει τα αποτελέσματα της απλής αναζήτησης ταινιών. Για να το πετύχει αυτό, χρησιμοποιεί το API με παράμετρο ”s” που ουσιαστικά στέλνει όλες τι ταινίες με τον τίτλο ή μέρος του τίτλου που έδωσε ο χρήστης. Περνάει αυτά



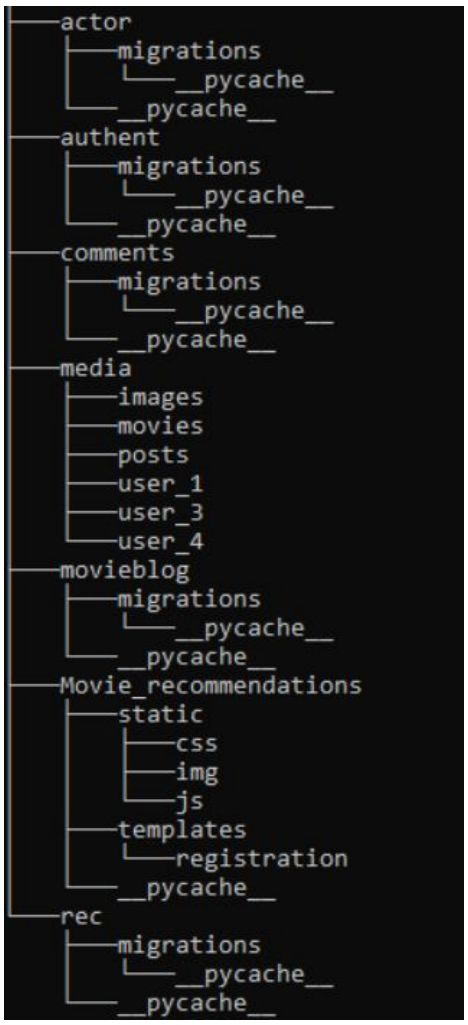
Σχήμα 7.1: Παράδειγμα των δεδομένων που στέλνονται από το API για την ταινία ”Batman”

τα δεδομένα σε ένα Django αντικείμενο και τα στέλνει στην σελίδα searchResult.html μαζί με την αναζήτηση που έκανε ο χρήστης και τον αριθμό της σελίδας.

Συνεχίζοντας, υπάρχει η συνάρτηση recommendations η οποία καλείται για την εμφάνιση των συστάσεων. Μέσα σε αυτή καλούμε την συνάρτηση get recommendations που αναλύσαμε παραπάνω για να πάρουμε την λίστα με τις δέκα ομοίοτερες ταινίες με αυτή που αναζήτησε ο χρήστης. Το επόμενο βήμα είναι να πάρουμε τις απαιτούμενες πληροφορίες για κάθε ταινία από το API. Τέλος στέλνουμε αυτές τις πληροφορίες για εμφάνιση στην σελίδα recommendationResult.html.

Τελευταία μας συνάρτηση είναι το movieInformation η οποία εκτός από το βασικό όρισμα request, δέχεται και το imdbID. Ο ρόλος της είναι να εμφανίζει όλες τις λεπτομέριες για μια ταινία. Οι λεπτομέριες αυτές είναι τα πεδία που αναφέραμε στην κλάση Movie παραπάνω. Για να βρει αυτές τις λεπτομέριες, αρχικά ελέγχει εάν αυτή η ταινία υπάρχει στην τοπική βάση. Σε αυτήν την περίπτωση συλλέγει και στέλνει τις πληροφορίες της ταινίας

για εμφάνιση. Διαφορετικά χρησιμοποιεί το API με παράμετρο 'i' για να προσκομίσει τις απαραίτητες πληροφορίες. Τέλος οι πληροφορίες αυτές μεταφέρονται στην σελίδα movieInformations.html για εμφάνιση. Στο Σχήμα 7.2 βλέπουμε τις εφαρμογές που υπάρχουν στην εργασία μας.

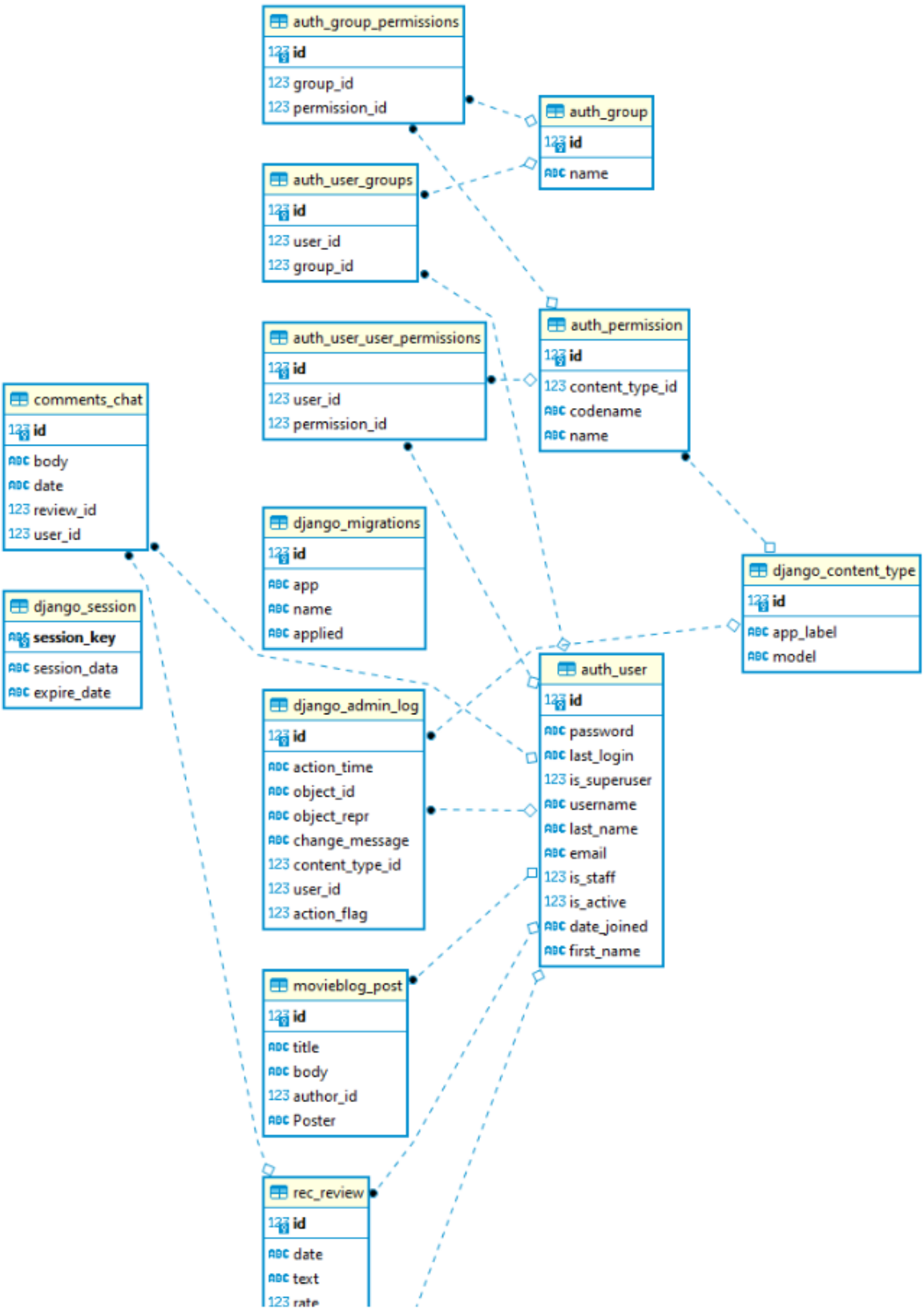


Σχήμα 7.2: Αναπαράσταση των εφαρμογών της εργασίας μας

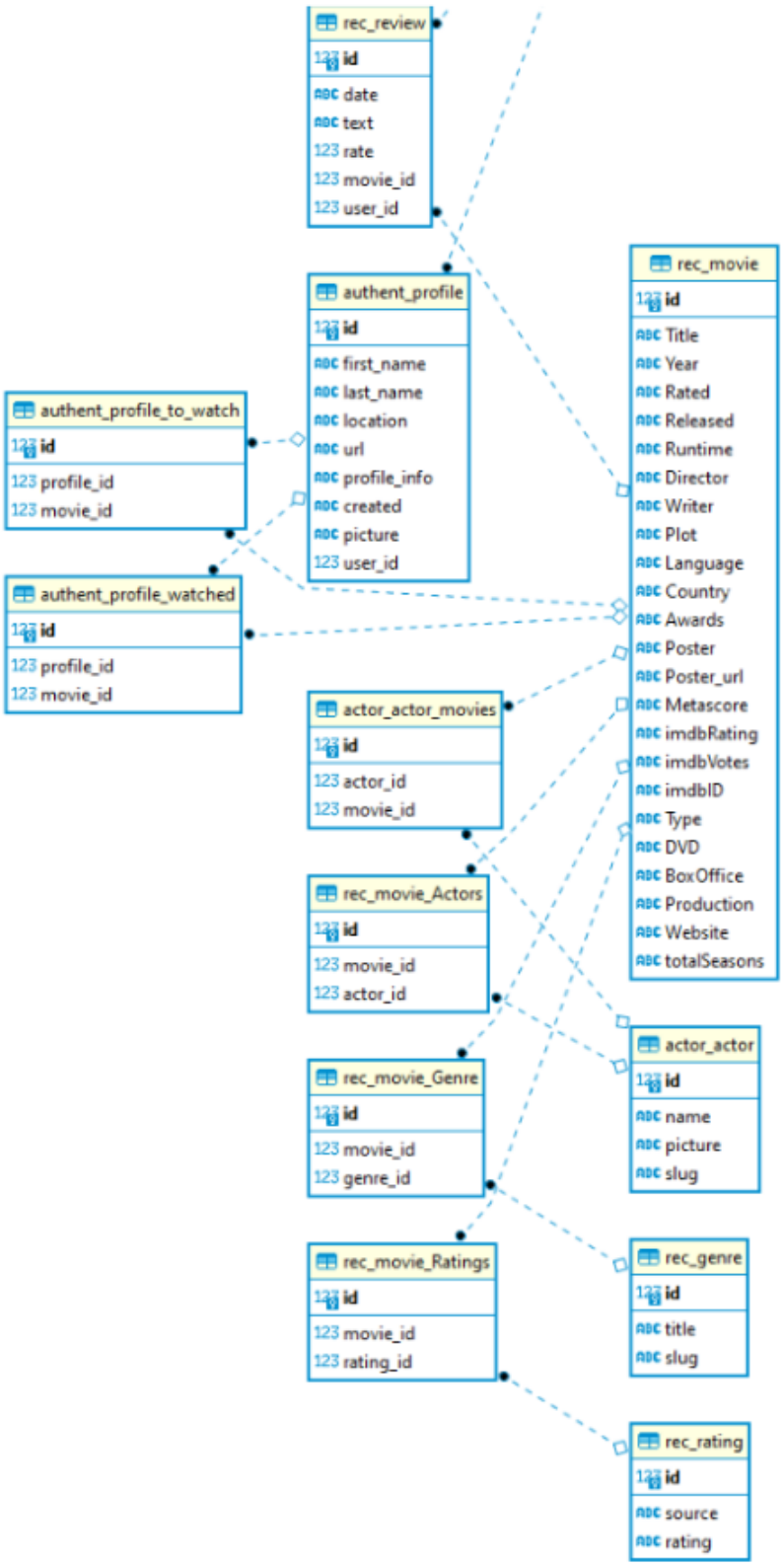
7.3 Βάση δεδομένων

Η βάση δεδομένων στο Django δημιουργείται από τα μοντέλα που έχει φτίαξει ο προγραμματιστής. Ένα μοντέλο είναι η μοναδική, οριστική πηγή πληροφοριών για τα δεδομένα μας. Περιέχει τα βασικά πεδία και τις συμπεριφορές των δεδομένων που αποθηκεύονται. Γενικά, κάθε μοντέλο αντιστοιχεί σε έναν πίνακα βάσης δεδομένων. Το DBMS που χρησιμοποιήσαμε είναι η SQLite η οποία είναι η προκαθορισμένη επιλογή για το Django. Όταν ξεκινάς το έργο σου δημιουργείται αυτόματα μια άδεια βάση .

7.3.1 Μοντέλο οντοτήτων-συσχετίσεων



Σχήμα 7.3: Α' μέρος μοντέλου οντοτήτων-συσχετίσεων (Ενώνεται με το κάτω Σχήμα 7.4)



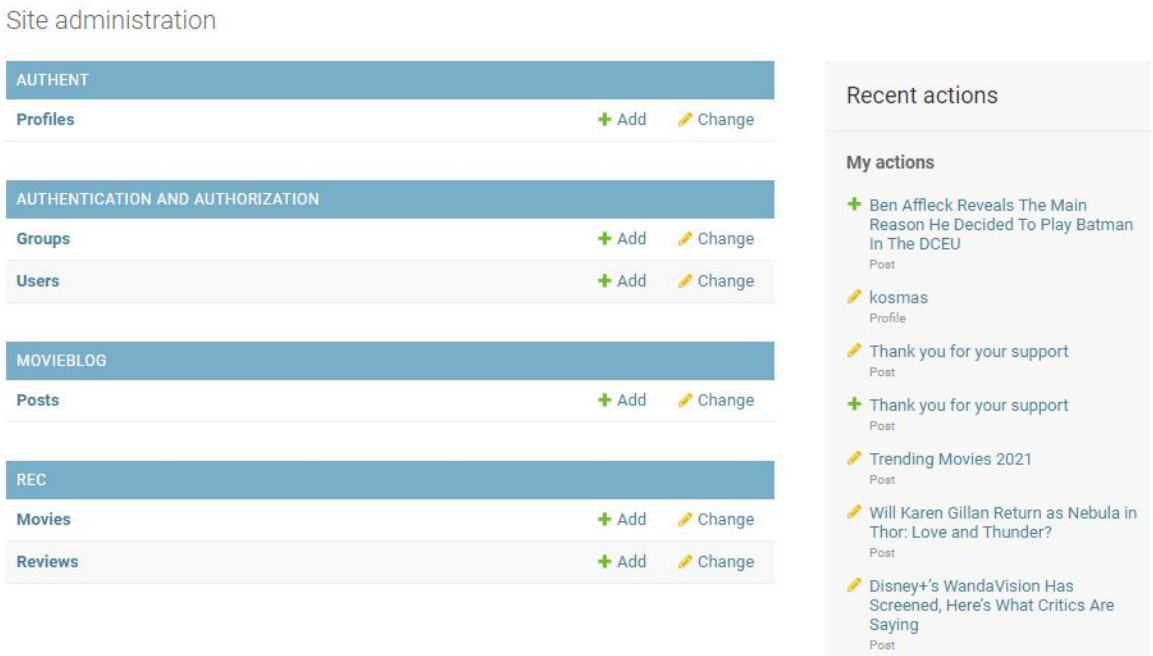
Σχήμα 7.4: Συνέχεια μοντέλου οντοτήτων-συσχετίσεων

Δημιουργήσαμε το μοντέλο οντοτήτων-συσχετίσεων με τη βοήθεια του DBeaver (Σχήμα 7.3 και Σχήμα 7.4). Το DBeaver είναι ένα δωρεάν εργαλείο διαχείρησης βάσης δεδομένων για

προγραμματιστές και διαχειριστές βάσεων δεδομένων. Οι πίνακες του ER Model που βλεπουμε στις παραπάνω εικόνες αντιστοιχούν στις κλάσεις των μοντέλων των εφαρμογών που αναλύσαμε στην προηγούμενη ενότητα. Ουσιαστικά κάθε κλάση που δημιουργούμε μέσα σε ένα μοντέλο αντιστοιχεί σε πίνακα τις βάσεις δεδομένων μας. Κάθε φορά που θέλουμε να ενημερώσουμε την βάση μας λόγω αλλαγών στα μοντέλα μας πρέπει να χρησιμοποιήσουμε την εντολή migrate για να ενημερωθεί και η βάση. Για αυτό το λόγο υπάρχει και ένας πίνακας στη βάση μας που ονομάζεται django migrations.

7.3.2 Σελίδα διαχείρισης

Η διαχείρηση της βάσης δεδομένων εκτός από κάποιο τρίτο πρόγραμμα, όπως το DBeaver, γίνεται από την σελίδα διαχείρησης. Για να αποκτήσεις πρόσβαση σε αυτή τη σελίδα απαιτείται η δημιουργία super-user από τη γραμμή εντολών. Έπειτα ο σύνδεσμος για την σελίδα διαχείρησης είναι ο http://127.0.0.1:8000/admin. Με την είσοδο του ο διαχειριστής θα πρέπει να εισάγει το όνομα και τον κωδικό του super-user προφίλ του. Ακολουθεί φωτογραφία με τα περιεχόμενα της σελίδας. (Σχήμα 7.5)



Σχήμα 7.5: Σελίδα διαχείρισης

Οι δυνατότητες που έχει ο διαχειριστής μέσω της σελίδας διαχείρησης είναι η προβολή και η επεξεργασία της βάσης (INSERT, UPDATE, DELETE). Ειδικότερα, μπορεί να επε-

ξεργαστεί τα προφίλ των χρηστών, να αλλάξει τα στοιχεία που έχουν δώσει οι ίδιοι και να δει ποιες ταινίες έχουν προσθέσει στην λίστα τους. Ακόμα μπορεί να αλλάξει τη κατάσταση ενός χρήστη και να του δώσει παραπάνω δικαιώματα. Μπορεί να δει πότε συνδέθηκε κάποιος χρήστης τελευταία φορά και πότε δημιούργησε το προφίλ του. Μέσω της σελίδας διαχείρησης, οι διαχειρίστες τις εφαρμογής μας μπορούν να γράφουν τα άρθρα για το ιστολόγιο μας. Τέλος έχουν πρόσβαση για να επεξεργαστούν τις ταινίες που υπάρχουν στη βάση απο τις αναζητήσεις των χρηστών και μπορούν να ελέγχουν τα σχόλια που παραθέτουν οι χρήστες έτσι ώστε να σβήσουν ακατάλληλα σχόλια όπως για παράδειγμα ρατσιστικές επιθέσεις και μηνύματα μίσους.

Κεφάλαιο 8

Συμπεράσματα και μελλοντική εργασία

Στα πλαίσια της διπλωματικής καταφέραμε να δημιουργήσουμε από το μηδέν έναν ιστότοπο που έχει ως σκοπό να προτείνει ταινίες στους χρήστες. Ακόμα, εκτός από τις προτάσεις μπορούνε να παίρνουνε πληροφορίες για διάφορες ταινίες, ηθοποιούς και να ανταλλάσουν απόψεις με άλλους χρήστες. Χρησιμοποιήσαμε τρεις διαφορετικούς αλγορίθμους συστάσεων και επιλέξαμε τον καλύτερο με βάση τα μετρικά και τις συστάσεις που έκανε στο χρήστη. Για τον αλγόριθμο που επιλέξαμε για τις συστάσεις δοκιμάσαμε αρκετές τεχνικές ομοιότητας όπως η Ευκλείδια απόσταση, η συσχέτιση Pearson και η ομοιότητα συνημιτόνου. Δοκιμάζοντας αυτές τις τρεις προσεγγίσεις διαπιστώσαμε ότι η ομοιότητα συνημιτόνου ήταν το ιδανικό για την εφαρμογή μας γιατί έχουμε αραιά δεδομένα. Ολοκληρώνοντας την εργασίας μας είμαστε πολύ ικανοποιημένοι με τα αποτελέσματα και με τις γνώσεις που αποκτήσαμε. Αρχικά μάθαμε πως να χειριζόμαστε μεγάλες εργασίες με μία ομάδα σε ένα αυστηρό χρονικό περιθώριο. Ακόμα είδαμε πως δημιουργείς έναν ιστότοπο και πως είναι να δουλεύεις για ένα καθαρά προγραμματιστικό έργο. Τέλος θα θέλαμε να αναφέρουμε κάποιους μελλοντικούς μας στόχους :

- Χρησιμοποίηση της βάσης δεδομένων του ιστότοπου για τις συστάσεις. Αυτή τη στιγμή οι συστάσεις που γίνονται, βασίζονται και αντλούν τα δεδομένα τους από ένα csv αρχείο. Όμως με κάθε αναζήτηση ταινίας στον ιστότοπο τα δεδομένα της ταινίας αποθηκεύονται στην βάση μας. Έτσι αν ο ιστότοπος λειτουργούσε κανονικά με μερικούς χρήστες θα μπορούσε να αποθηκεύσει πάρα πολλές ταινίες και να αποκτήσουμε αυτονομία για τις συστάσεις μας. Αν συμβεί αυτό τότε πιστεύουμε ότι οι προτάσεις μας θα βελτιωνόντουσαν σε μεγάλο βαθμό γιατί θα μπορούσαμε να χρησιμοποιήσουμε περισσότερα χαρακτηριστικά όπως για παράδειγμα τη βαθμολογία του IMDB. Ακόμα θα

μπορούσαμε να κάνουμε πιο προσωποποιημένες προτάσεις αφού θα γνωρίζαμε καλύτερα τους χρήστες μας και τις προτιμήσεις τους.

- Αλλαγή του συστήματος προτάσεων από σύστημα που βασίζεται στο περιεχόμενο σε υβριδικό. θα θέλαμε να συνδυάσουμε διαφορετικούς αλγορίθμους με ένα νευρωνικό δίκτυο όπως το RBM (περιορισμένη μηχανή boltzmann) για να δούμε τα αποτελέσματα του και κατά πόσο θα αναβάθμιζε τον ιστότοπο μας με καλύτερες συστάσεις.

- Εισαγωγή λίστας ταινιών που δεν αρέσουν στους χρήστες. Τα δεδομένα χρήστη είναι πάντοτε χρήσιμα σε συστήματα σύστασης. Στο μέλλον εμείς θα συλλέξουμε περισσότερα δεδομένα χρήστη και θα προσθέσουμε μια λίστα από ταινίες που δεν θα τους αρέσουν. Έτσι θα βελτιώσουμε τις συστάσεις μας και δεν θα προτείνουμε στους χρήστες ταινίες που με χαρακτηριστικά που δεν θα τους αρέσουν.

Βιβλιογραφία

[1] Rajeev Kumar, Guru Basava, and Felicita Furtado. An efficient content, collaborative – based and hybrid approach for movie recommendation engine. Apr. 2020.

[2] Amazon. https://www.amazon.com/. Ημερομηνία πρόσβασης: 12-2-2021.

[3] Netflix. https://www.netflix.com/. Ημερομηνία πρόσβασης: 12-2-2021.

[4] Youtube. https://www.youtube.com/. Ημερομηνία πρόσβασης: 12-2-2021.

[5] Twitter. https://twitter.com/. Ημερομηνία πρόσβασης: 12-2-2021.

[6] Mahiye Uluyagmur, Z. Cataltepe, and Esengul Tayfur. Content-based movie recommendation using different feature sets. Oct. 2012.

[7] Building recommender systems with machine learning and ai. https://www.udemy.com/course/building-recommender-systems-with-machine-learning-and-ai/?fbclid=IwAR21XXkxBc3xddd0OG8quzh-yTaZr4jnYVT_qnIsvQ4ZYqPW9YKjCFSWURw. Ημερομηνία πρόσβασης: 12-2-2021.

[8] Omdb api. http://www.omdbapi.com/. Ημερομηνία πρόσβασης: 22-1-2021.

[9] Pearson correlation. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient. Ημερομηνία πρόσβασης: 22-1-2021.

[10] Imdb. https://www.imdb.com/. Ημερομηνία πρόσβασης: 12-2-2021.

[11] Anaconda. https://www.anaconda.com/. Ημερομηνία πρόσβασης: 12-2-2021.

[12] Charu C. Aggarwal. *Recommender Systems: The Textbook*. 1 edition, 2016.

[13] J.Ben Schafer, Dan Frankowski, and Jon Herlocker. Collaborative filtering recommender systems. Dec. 2007.

[14] Kim Falk. *Practical Recommender Systems*. 1 edition, 2019.

[15] SRS Reddy, Sravani Nalluri, Subramanyam Kunisetti, S. Ashok, and B. Venkatesh Herlocker. Content-based movie recommendation system using genre correlation. Dec. 2019.

[16] Bagher Rahimpour Cami, Hamid Hassanpour, and Hoda Mashayekhi. A content-based movie recommender system based on temporal user preferences. Dec. 2017.

[17] Movielens. `https://grouplens.org/datasets/movielens/`. Ημερομηνία πρόσβασης: 22-1-2021.

[18] Tf-idf. `https://en.wikipedia.org/wiki/Tf%E2%80%93idf`. Ημερομηνία πρόσβασης: 22-1-2021.

[19] Cosine similarity. `https://en.wikipedia.org/wiki/Cosine_similarity`. Ημερομηνία πρόσβασης: 22-1-2021.

[20] Materialize. `https://materializecss.com/`. Ημερομηνία πρόσβασης: 22-1-2021.

[21] Django documentation. `https://docs.djangoproject.com/en/3.1/`. Ημερομηνία πρόσβασης: 22-1-2021.

[22] Rounak Banik. *Hands-On Recommendation Systems with Python: Start building powerful and personalized, recommendation engines with Python*. 1 edition, 2018.

[23] Python. `https://www.python.org/downloads/`. Ημερομηνία πρόσβασης: 22-1-2021.

[24] Django. `https://www.djangoproject.com/download/`. Ημερομηνία πρόσβασης: 22-1-2021.

[25] Github code. `https://github.com/gprapas/Movie_Rec.git`. Ημερομηνία πρόσβασης: 12-2-2021.

ΠΑΡΑΡΤΗΜΑΤΑ

Παράρτημα Α

Απαραίτητες Εγκαταστάσεις

Σε αυτό το παράρτημα ακολουθούν βήμα βήμα οι απαραίτητες εγκαταστάσεις που πρέπει να πραγματοποιηθούν ώστε να τρέχει σωστά η εφαρμογή.

Α.1 Εγκατάσταση Python

Θα αναφερθούμε στην εγκατάσταση της Python σε Windows, καθώς στα λειτουργικά Mac OS X υπάρχει πλέον μια προεγκατεστημένη έκδοση Python 2 και στις περισσότερες εκδόσεις Linux υπάρχει προεγκατεστημένη ακόμα και η Python 3. Αρχικά ελέγχουμε αν και ποια έκδοση της Python είναι εγκατεστημένη στον υπολογιστή μας. Αυτό γίνεται με την εντολή python στην γραμμή εντολών όπως φαίνεται στο Σχήμα Α.1.

```
C:\Users\George>python
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Σχήμα Α.1: Έλεγχος Python

Στην συγκεκριμένη εφαρμογή χρησιμοποιήθηκε η έκδοση Python 3.8.5. Εάν δεν υπάρχει εγκατεστημένη η Python στο μηχάνημα σας υπάρχουν δύο τρόποι για την εγκάτασταση της. Ο πρώτος τρόπος είναι κατεβάζοντας την επιθυμητή έκδοση από την επίσημη ιστοσελίδα της Python [23]. Για την εγκάτασταση υπάρχει στο αρχείο διαθέσιμος Windows Installer. Όμως για τους χρήστες Windows υπάρχει ένας πιο βέλτιστος τρόπος για την εγκατάσταση της Python ο οποίος περιλαμβάνει την εγκατάσταση της ActivePython. Τα πλεονεκτήματα αυτής της διαδικασίας είναι τα εξής:

1. Δεν χρειάζεται να προσθέσουμε την Python στο ”μονοπάτι” της γραμμής εντολών

2. Εγκαθίστανται παράλληλα και κάποια χρήσιμα πακετά (packages), όπως το ”easyinstall”, το οποίο είναι ένα πακέτο που επιτρέπει την επιτυχή και άμεση εγκατάσταση πακέτων Python, χωρίς να απαιτείται από το χρήστη να κατεβάσει και να εγκαταστήσει κάποιο πακέτο.

Τέλος, με την εγκατάσταση της Python εγκαθίσταται παράλληλα και το IDLE. Το IDLE είναι το περιβάλλον ανάπτυξης κώδικα Python. Μέρη του IDLE είναι επίσης το Python shell, ένας Python debugger, το editor και το documentation της γλώσσας.

Α.2 Εικονικό περιβάλλον

Χρησιμοποιήσαμε εικονικό περιβάλλων για την εφαρμογή μας. Ο κύριος σκοπός ενός εικονικού περιβάλλοντος της Python είναι να δημιουργήσει ένα απομονωμένο περιβάλλον για έργα Python. Αυτό σημαίνει ότι κάθε έργο μπορεί να έχει τις δικές του εξαρτήσεις, ανεξάρτητα από το τι εξαρτάται από κάθε άλλο έργο. Με τον πιο απλό τρόπο, ένα εικονικό περιβάλλον παρέχει ένα περιβάλλον ανάπτυξης ανεξάρτητο από το λειτουργικό σύστημα του κεντρικού υπολογιστή. Μπορούμε λοιπόν να εγκαταστήσουμε και να χρησιμοποιήσουμε το απαραίτητο λογισμικό στο φάκελο / bin του virtualenv, αντί να χρησιμοποιήσουμε το λογισμικό που είναι εγκατεστημένο στον κεντρικό υπολογιστή. Όλα τα πακέτα που χρησιμοποιήσαμε τα εγκαταστήσαμε στο εικονικό μας περιβάλλον. Ακόμα εκεί εγκαταστήσαμε και το Django.

Η εγκατάσταση του εικονικού περιβάλλοντος είναι απλή και αποτελείται από τα εξής βήματα:

1. Αρχικά εγκαθιστούμε το εργαλείο virtualenv με την ακόλουθη εντολή

```
pip install virtualenv
```

2. Δημιουργούμε έναν φάκελο στον οποίο θα δουλέψουμε

```
mkdir python-virtual-environments
```

3. Μεταφερόμαστε σε αυτόν τον φάκελο

```
cd python-virtual-environments
```

4. Δημιουργούμε το εικονικό περιβάλλον

```
python3 -m venv env
```

Μετά από αυτήν την διαδικασία αυτό που έχει διαμορφωθεί φαίνεται στο Σχήμα Α.2.

```
├── bin
│   ├── activate
│   ├── activate.csh
│   ├── activate.fish
│   ├── easy_install
│   ├── easy_install-3.5
│   ├── pip
│   ├── pip3
│   ├── pip3.5
│   ├── python -> python3.5
│   ├── python3 -> python3.5
│   └── python3.5 -> /Library/Frameworks/Python.framework/Versions/3.5/bin/python3.5
├── include
├── lib
│   └── python3.5
│       └── site-packages
└── pyvenv.cfg
```

Σχήμα Α.2: Φάκελοι εικονικού περιβάλλοντος

Τελευταίο βήμα είναι η ενεργοποίηση του εικονικού περιβάλλοντος που γίνεται με την εντολή:

```
source env/bin/activate
```

A.3 Εγκατάσταση των πακέτων

Ο βασικός λόγος της ύπαρξης των πακέτων συναρτήσεων είναι η επαναχρησιμοποίηση συναρτήσεων και κώδικα γενικότερα. Η επαναχρησιμοποίηση αυτή μπορεί να γίνει τόσο από τον δημιουργό των συναρτήσεων, όσο και από τρίτους εφόσον ο δημιουργός ανεβάσει τον κώδικα του σε κάποιες από τις πολλές «αποθήκες» κώδικα στο διαδίκτυο.

Εγκατάσταση με το `pip_install`

Για τη χρήση της εντολής "pip" πρέπει προηγουμένως να έχει εγκατασταθεί το pip. Για την εγκατάστασή του κατεβάζουμε και εκτελούμε το "get-pip.py". Έπειτα για να εγκαταστήσουμε ένα πακέτο χρησιμοποιούμε την εξής εντολή:

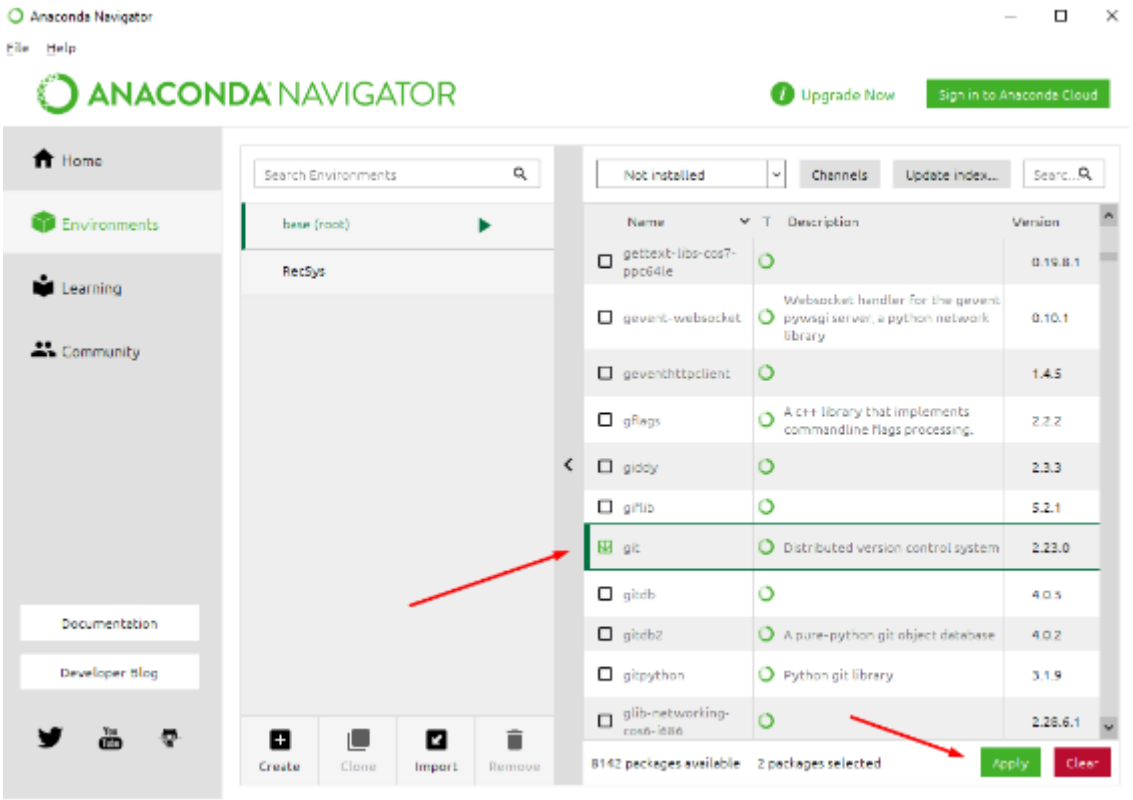
```
pip_install + package name
```

Για παράδειγμα, για την εγκάτασταση του πακέτου numpy χρησιμοποιούμε την παρακάτω εντολή:

```
pip_install numpy
```

Εγκατάσταση από το περιβάλλον του Anaconda

Αυτός ο τρόπος είναι πιο φιλικός προς τον χρήστη καθώς δεν χρειάζεται να αλληλεπιδράσει με την γραμμή εντολών που πολλές στεκέται εμποδίο στην ομαλή εγκατάσταση των πακέτων. Μέσω του Anaconda ο χρήστης μπορεί να αναζητήσει το επιθυμητό πακέτο από μια λίστα με τα διαθέσιμα πακέτα και να το εγκαταστήσει πατώντας το κουμπί apply όπως φαίνεται στο Σχήμα Α.3.



Σχήμα Α.3: Εγκατάσταση των πακέτων από Anaconda [11]

Στην συγκεκριμένη εργασία τα πακέτα που εγκαταστήσαμε είναι αυτά που απεικονίζονται στο Σχήμα Α.4 με τις ακόλουθες εκδόσεις τους.

```
asgiref==3.3.1
certifi==2020.11.8
chardet==3.0.4
Django==3.1.3
idna==2.10
joblib==0.17.0
numpy==1.19.4
pandas==1.1.4
Pillow==8.0.1
python-dateutil==2.8.1
pytz==2020.4
requests==2.25.0
scikit-learn==0.23.2
scipy==1.5.4
six==1.15.0
sklearn==0.0
sqlparse==0.4.1
threadpoolctl==2.1.0
urllib3==1.26.2
```

Σχήμα Α.4: Τα πακέτα της εφαρμογής

Α.3.1 Εγκατάσταση Django

Θα αναφέρουμε σύντομα πως μπορεί κανείς να εγκαταστήσει το Django σε Windows. Ούτως ή άλλως πολλές από τις διανομές Linux και Unix περιέχουν εξ αρχής το Django. Τα βήματα που απαιτούνται για την εγκατάσταση είναι:

1. Κατεβάζουμε την έκδοση που επιθυμούμε από την ιστοσελίδα του Django [24]. Εμείς επιλέξαμε την τελευταία επίσημη 3.1.3.

2. Αποσυμπιέζουμε το αρχείο που κατεβάζουμε

3. Εκτελούμε από τη γραμμή εντολών, αφού πλοηγηθούμε στο σωστό φάκελο, την εντολή:

   ```
   setup.py install
   ```

Δημιουργία νέου έργου

Πριν κάνουμε οποιαδήποτε ενέργεια πρέπει να έχουμε ενεργοποιήσει το εικονικό μας περιβάλλον. Έπειτα μεταφερόμαστε στον φάκελο που θέλουμε να δημιουργήσουμε το έργο μας και τρέχουμε την εντολή:

```
django-admin.py startproject ourproject
```

Έπειτα από την γραμμή εντολών εισερχόμαστε στον φάκελο του έργου μας όπου συναντάμε τα εξής αρχεία:

```
__init__.py
manage.py
urls.py
settings.py
```

Τρέχοντας την ακόλουθη εντολή στον φάκελο του έργου ενεργοποιούμε τον server στην IP του localhost μας.

```
python manage.py runserver
```

Εάν όλα πάνε καλά το αποτέλεσμα θα είναι το Σχήμα Α.5.

```
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
January 18, 2021 - 14:46:05
Django version 3.1.3, using settings 'Movie_recommendations.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Σχήμα Α.5: Django runserver

Τώρα που ο server τρέχει, επισκεπτόμενοι τη τοποθεσία http://127.0.0.1:8000/ με ένα πρόγραμμα περιήγησης και βλέπουμε την σελίδα ”Welcome to Django”.

Α.3.2 Ρύθμιση της βάσης δεδομένων

Δεδομένου ότι επιλέξαμε να χρησιμοποιήσουμε την SQLite, η οποία είναι προεγκατεστημένη μαζί με τη Python, απαιτείται μόνο μία ρύθμιση πριν τη χρήση της στο έργο μας. Για την ρύθμιση της βάσης δεδομένων θα χρειαστεί να επεξεργαστούμε το αρχείο settings.py (Σχήμα Α.6). Στην προκειμένη περίπτωση η βάση δεδομένων μας αποτελείται από ένα αρχείο με όνομα db.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

Σχήμα Α.6: Ρυθμίσεις της βάσης δεδομένων

Παράρτημα Β

Οδηγίες για τη χρησιμοποίηση της εφαρμογής

Σε αυτό το παράρτημα υπάρχουν οι οδηγίες για το κατέβασμα του κώδικα καθώς και τα απαραίτητα βήματα για την λειτουργία του.

Αρχικά κατεβάζουμε τον φάκελο της εφαρμογής από τον σύνδεσμο στο github. [25] Απαραίτητη προϋπόθεση για την σωστή λειτουργία του κώδικα αποτελεί η ολοκλήρωση των εγκαταστάσεων που περιγράφονται στο Παράρτημα Α. Εφόσον πραγματοποιηθούν τα βήματα αυτά, ανοίγουμε το αρχείο που κατεβάσαμε από τον σύνδεσμο. Μέσα στον φάκελο υπάρχει ένα αρχείο μορφής κειμένου (txt) με το όνομα modules το οποίο πρέπει να εγκατασταθεί στο εικονικό περιβάλλον. Από την γραμμή εντολών, αφού έχουμε ενεργοποιήσει το εικονικό περιβάλλον, πληκτρολογούμε την ακόλουθη εντολή:

```
pip install -r modules.txt
```

Ακόμα πρέπει να ανοίξουμε τον κώδικα και να πάμε στο MovieRecommendations/rec/views.py και να αλλάξουμε το μονοπάτι που έχουμε αποθηκεύσει τα csv αρχεία (Σχήμα Β.1).

```
df1=pd.read_csv('C:/Users/eglav/Documents/diplwmatiki/Movie_Recommendations/Movie_recommendations/tmdb_5000_credits.csv')
df2=pd.read_csv('C:/Users/eglav/Documents/diplwmatiki/Movie_Recommendations/Movie_recommendations/tmdb_5000_movies.csv')
```

Σχήμα Β.1: Το μονοπάτι των αρχείων csv στην εφαρμογή

Για να διασφαλίσουμε πως η βάση που υπάρχει στο αρχείο είναι ενημερωμένη πρέπει να εκτελέσουμε την ακόλουθη εντολή μέσα στον αρχικό φάκελο της εφαρμογής (Movie Recommendations).

```
python manage.py migrate
```

Έτσι ολοκληρώνονται οι απαραίτητες εγκαταστάσεις για την λειτουργία του κώδικα. Το επόμενο βήμα είναι η δημιουργία ενός διαχειριστή για την εφαρμογή. Αυτό γίνεται με την εντολή:

```
python manage.py createsuperuser
```

Στα πεδία που θα εμφανιστούν θα ζητηθεί από τον χρήστη ένα όνομα χρήστη, μια ηλεκτρονική διεύθυνση καθώς και τον κωδικό που επιθυμεί (Σχήμα Β.2).

```
(snik) C:\Users\George\Documents\kosmakos\Movie_Recommendations>python manage.py createsuperuser
Username (leave blank to use 'george'): GeorgePrapas
Email address: gprapas@uth.gr
Password:
Password (again):
```

Σχήμα Β.2: Δημιουργία διαχειριστή

Τελευταίο βήμα είναι η ενεργοποίηση του σέρβερ, το οποίο επιτυγχάνεται με την εντολή:

```
python manage.py runserver
```

Εάν ακολουθήσουμε πιστά τα βήματα το αποτέλεσμα θα είναι αυτό στο παρακάτω Σχήμα Β.3.

```
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
January 18, 2021 - 14:46:05
Django version 3.1.3, using settings 'Movie_recommendations.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Σχήμα Β.3: Ενεργοποίηση σέρβερ