Anime Recommendation System

Νικόλαος Σταυρινός και Κωνσταντίνος Κωνσταντινίδης

Εξόρυξη Δεδομένων Μεγάλου Όγκου 2021-22
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Πανεπιστήμιο Θεσσαλίας, Βόλος
{konkonstant,nstavrinos}@e-ce.uth.gr

Περίληψη Ένα σύστημα συστάσεων, είναι μια υποκατηγορία συστήματος φιλτραρίσματος πληροφοριών που επιδιώκει να προβλέψει τη ”βαθμολόγηση” ή την ”προτίμηση” που θα έδινε ένας χρήστης σε ένα στοιχείο. Τα συστήματα συστάσεων χρησιμοποιούνται σε διάφορους τομείς, με κοινά αναγνωρισμένα παραδείγματα που λαμβάνουν τη μορφή δημιουργίας λιστών αναπαραγωγής για υπηρεσίες βίντεο και μουσικής, συστάσεων προϊόντων για ηλεκτρονικά καταστήματα ή συστάσεων περιεχομένου για πλατφόρμες μέσων κοινωνικής δικτύωσης και συστάσεων ανοιχτού περιεχομένου ιστού.Στο δικό μας project αναπτύξαμε ενα recommendation system για anime.

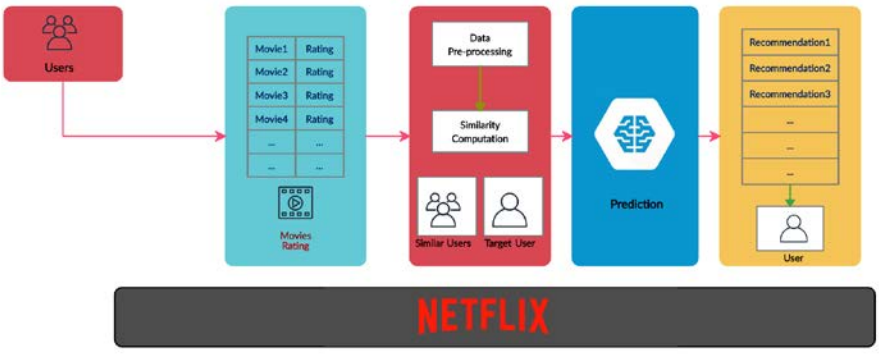
Λέξεις Κλειδιά: Recommendation system, Anime, Cosine Similarity

1 Εισαγωγή

Κάθε streaming περιεχόμενο έχει τους δικούς του θεατές και κάθε περιεχόμενο έχει τη βαθμολογία του. Οι θεατές αφήνουν μερικές καλές βαθμολογίες για το περιεχόμενο, αν τους αρέσει. Πού ισχύει όμως; Οι θεατές μπορούν να περάσουν ώρες ψάχνοντας ανάμεσα σε εκατοντάδες, μερικές φορές χιλιάδες anime χωρίς να βρίσκουν ποτέ το περιεχόμενο που τους αρέσει. Οι επιχειρήσεις πρέπει να λαμβάνουν προτάσεις με βάση τις προτιμήσεις και τις ανάγκες τους, προκειμένου να δημιουργήσουν ένα καλύτερο περιβάλλον αναπαραγωγής περιεχομένου που ενισχύει τα έσοδα και αυξάνει τον χρόνο που αφιερώνεται σε έναν ιστότοπο.

1.1 Recommendation System

Είναι ένας αλγόριθμος μάθησης χωρίς επίβλεψη (δηλ. δεν έχει μεταβλητή στόχου για τη μέτρηση της ακρίβειας) που χρησιμοποιείται κυρίως για να βοηθήσει στη λήψη αποφάσεων από τους καταναλωτές. Τέτοια συστήματα εμφανίζονται σε μέρη όπως εφαρμογές streaming (γνωστοί και ως Netflix και Amazon Prime) για να βοηθήσουν τους χρήστες να επιλέξουν μια τηλεοπτική εκπομπή ή ταινία για να παρακολουθήσουν και σε ιστότοπους δημοσιογραφίας/μέσα όπως το Medium για να προτείνουν άλλα άρθρα που μπορεί να θέλουν να διαβάσουν. Προφανώς, πολλοί ηλεκτρονικοί λιανοπωλητές όπως η Amazon χρησιμοποιούν ήδη αλγόριθμους συστάσεων για αρκετό καιρό, αλλά πολλοί μικρότεροι ή νεότεροι ιστότοποι εξακολουθούν να χρειάζονται ένα. Υπάρχουν διαφορετικές ποικιλίες συστάσεων που βασίζουν τις προβλέψεις τους σε διαφορετικά χαρακτηριστικά.



1.2 Εργαλεία

Η υλοποίηση της εργασίας μας έγινε σε jupyter notebook και σε γλώσσα python.

2 Dataset

Αυτό το σύνολο δεδομένων περιέχει πληροφορίες για δεδομένα προτιμήσεων χρήστη από 73.516 χρήστες σε 12.294 anime. Κάθε χρήστης μπορεί να προσθέσει anime στην ολοκληρωμένη λίστα του και να του δώσει μια βαθμολογία και αυτό το σύνολο δεδομένων είναι μια συλλογή αυτών των αξιολογήσεων.

Dataset Details

1)anime_data:

anime_id - myanimelist.net's unique id identifying an anime.
name - full name of anime.
genre - comma separated list of genres for this anime.
type - movie, TV, OVA, etc.
episodes - how many episodes in this show. (1 if movie).
rating - average rating out of 10 for this anime.
members - number of community members that are in this anime's "group".

2)rating_data:

user_id - non identifiable randomly generated user id.
anime_id - the anime that this user has rated.
rating - rating out of 10 this user has assigned (-1 if the user watched it but didn't assign a rating).

3 Προεπεξεργασία Δεδομένων

3.1 Καθαρισμός anime_title

Βρήκαμε πολλά σύμβολα στο anime_title και αφαιρούμε από το όνμα του κάθε anime ώστε να τα διαχειριστούμε σωστά.

3.2 Διαχείριση NaN τιμών

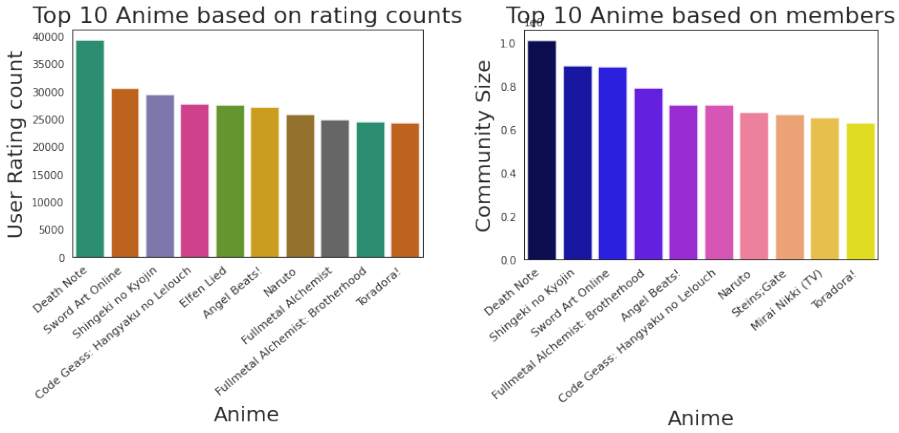
Αρχικά, θέτουμε το rating σε όσες ταινίες λείπει σε μηδεν.Στη συνέχεια, βρίσκουμε τις τιμές που λείπουν από τα επεισόδια και τον τύπο του anime και σε όσα γνωρίζουμε τα συμπληρώνουμε με το χέρι.Ενώ τα υπόλοιπα τα διαγράφουμε, εφόσον δε θα βοηθήσουν στη πρόταση επιλογής.Τέλος, όσων anime το genre δε το γνωρίζουμε το συμπληρώνουμε με Unknown τιμή.

3.3 Ένωση Δεδομένων

Για να υλόποιήσουμε το recommendation system μας χρειάζεται να ενώσουμε τα δυο dataset με τα οποία δουλεύουμε,ώστε να το σύστημα να μπορει να κάνει προβλέψεις.

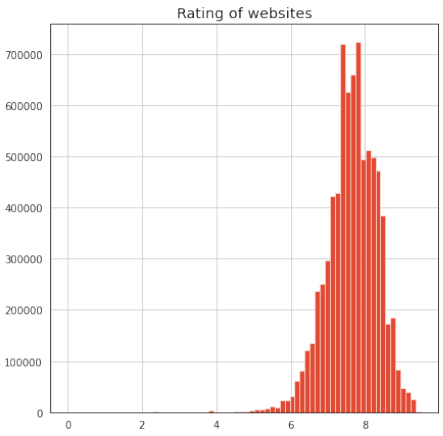
4 Ανάλυση Δεδομένων

4.1 Top 10 Anime on rating counts and members



4.2 Distribution of ratings

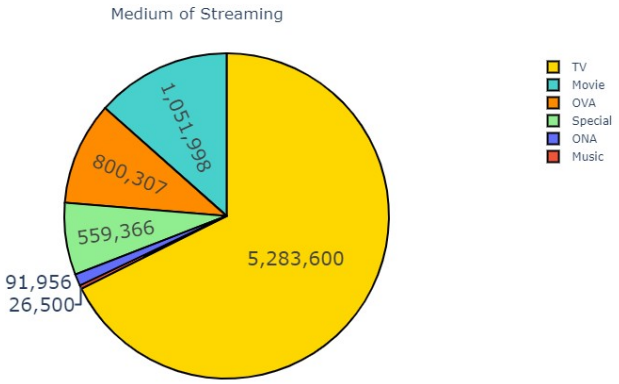
Αυτή είναι η κατανομή των αξιολογήσεων και στα δύο σύνολα δεδομένων. Η βαθμολογία από το anime.csv προέρχεται από ιστότοπους κριτικών και η βαθμολογία χρήστη στο rating.csv προέρχεται από το αναγνωριστικό των χρηστών



Συμπέρασμα

- Οι περισσότερες βαθμολογίες κατανέμονται μεταξύ 6-10
- Το μεγαλύτερο μέρος της κατανομής κυμαίνεται περίπου στο 7,5-8,0
- Και οι δύο κατανομές αφήνονται skewed
- Έχουμε βαθμολογία -1 ως ακραία τιμή στη βαθμολογία των χρηστών που μπορεί να γίνει NaN

4.3 Μέσο streaming



- Το 67,6% των anime προβλήθηκαν στην τηλεόραση και ακολούθησε το 13,5% μέσω ταινιών.
- Το 10,2% των anime μεταδίδεται ως OVA, το οποίο είναι μεγαλύτερο από το ONA (1,18%)

5 Recommendation building

5.1 Φιλτράρισμα χρηστών

Υπάρχουν χρήστες που έχουν βαθμολογήσει μόνο μία φορά, ακόμα κι αν το έχουν βαθμολογήσει με 5, δεν μπορεί να θεωρηθεί πολύτιμο ρεκόρ για σύσταση. Επομένως, έχουμε θεωρήσει ως κατώτατη τιμή τουλάχιστον 300 αξιολογήσεις από τον χρήστη.Αυτή η επιλογή έγινε τόσο για τον λόγο που αναφέραμε πριν, όσο για το περιορισμό μνήμης στο σύστημα μας(χρήστες με > 300 βαθμολογήσεις χρειάζονται 10 gb Ram). Μπορούμε να αλλάξουμε την τιμή του ορίου για να έχουμε καλύτερα αποτελέσματα, αν και η επιλογή που κάναμε ήταν αρκετή.

5.2 Types of Recommendation Systems

Content filtering recommender systems. Το φιλτράρισμα βάσει περιεχομένου, προτείνει στοιχεία που βασίζονται σε σύγκριση μεταξύ του περιεχομένου των στοιχείων και ενός προφίλ χρήστη. Είναι μια τεχνική μηχανικής μάθησης που χρησιμοποιεί ομοιότητες στα χαρακτηριστικά για τη λήψη αποφάσεων. Αυτή η τεχνική χρησιμοποιείται συχνά σε συστήματα συστάσεων, τα οποία είναι αλγόριθμοι που έχουν σχεδιαστεί για να διαφημίζουν ή να προτείνουν πράγματα στους χρήστες με βάση τη γνώση που έχει συσσωρευτεί για τον χρήστη.

Collaborative filtering based recommender systems. Η ιδέα πίσω από το συλλογικό φιλτράρισμα είναι να ληφθούν υπόψη οι απόψεις των χρηστών για διαφορετικά anime και να προτείνουμε το καλύτερο σε κάθε χρήστη με βάση τις προηγούμενες κατατάξεις του χρήστη και τη γνώμη άλλων παρόμοιων τύπων χρηστών.

5.3 Content filtering

Πρόβλεψη βάση κατηγοριών και είδους Για τη πρόβλεψη βάση κατηγοριών και είδους, δημιουργούμαι έναν πίνακα δεδομένων που δημιουργήθηκε με τον αλγόριθμο

TF-IDF. Το TF είναι απλώς η συχνότητα μιας λέξης σε ένα έγγραφο. Το IDF είναι το αντίστροφο της συχνότητας εγγράφων μεταξύ ολόκληρου του συνόλου των εγγράφων. Η στάθμιση TF-IDF αναιρεί την επίδραση των λέξεων υψηλής συχνότητας στον προσδιορισμό της σημασίας ενός αντικειμένου.Στη δική μας περίπτωση, ο πίνακας μας δείχνει ττη συχνότητα της κάθε κατηγορίας αλλά και είδους για το κάθε τίτλο άνιμε.

Για να υπολογίσουμε τη συσχέτιση μεταξύ των άνιμε, χρησιμοποιούμε 3 διαφορετικούς τρόπους(linear, sigmoid kernel και cosine similarity) για να δούμε αν υπάρχει διαφορά στα recommendation.

Linear Kernel Recommendation Η συνάρτηση linear_kernel υπολογίζει τον γραμμικό πυρήνα, δηλαδή μια ειδική περίπτωση πολυωνύμου πυρήνα με βαθμό=1 και συντελεστή0=0 (ομογενής). Εάν τα x και y είναι διανύσματα στηλών, ο γραμμικός πυρήνας τους είναι:

$$k(x, y) = x^\top y$$

	Anime name	Similarity	Type
0	Naruto	1.000000	TV
1	Boruto: Naruto the Movie	0.854500	Movie
2	Naruto: Shippuuden Movie 4 - The Lost Tower	0.854500	Movie
3	Naruto: Shippuuden Movie 3 - Hi no Ishi wo Tsu...	0.854500	Movie
4	Naruto Soyokazeden Movie: Naruto to Mashin to ...	0.854500	Movie
5	Naruto x UT	0.853885	OVA
6	Boruto: Naruto the Movie - Naruto ga Hokage ni...	0.848793	Special
7	Naruto Shippuuden: Sunny Side Battle	0.848793	Special
8	Rekka no Honoo	0.648278	TV
9	Battle Spirits: Ryuuko no Ken	0.624475	OVA

Εικ. 1. Linear recommendations for Naruto: Shippuden

Sigmoid Kernel Recommendation Η συνάρτηση sigmoid_kernel υπολογίζει τον σιγμοειδές πυρήνα μεταξύ δύο διανυσμάτων. Ο σιγμοειδής πυρήνας είναι επίσης γνωστός ως υπερβολική εφαπτομένη. Ορίζεται ως:

$k(x, y) = \tanh(\gamma x^\top y + c_0)$

που:

x, y είναι τα διανύσματα εισόδου

γ είναι γνωστή ως κλίση

c_0 είναι γνωστή ως intercept

	Anime name	Similarity	Type
0	Naruto	0.761641	TV
1	Boruto: Naruto the Movie	0.761634	Movie
2	Naruto: Shippuuden Movie 4 - The Lost Tower	0.761634	Movie
3	Naruto: Shippuuden Movie 3 - Hi no Ishi wo Tsu...	0.761634	Movie
4	Naruto Soyokazeden Movie: Naruto to Mashin to ...	0.761634	Movie
5	Naruto x UT	0.761634	OVA
6	Boruto: Naruto the Movie - Naruto ga Hokage ni...	0.761634	Special
7	Naruto Shippuuden: Sunny Side Battle	0.761634	Special
8	Rekka no Honoo	0.761624	TV
9	Battle Spirits: Ryuuko no Ken	0.761623	OVA

Εικ. 2. Sigmoid recommendations for Naruto: Shippuden

Cosine Similarity Recommendation Η ομοιότητα συνημιτόνου υπολογίζει το κανονικοποιημένο με L2 γινόμενο διανυσμάτων.Δηλαδή, αν τα x και y είναι διανύσματα σειρών, η ομοιότητα συνημιτόνου τους ορίζεται ως:

$$k(x, y) = \frac{xy^\top}{\|x\|\|y\|}$$

Ονομάζεται ομοιότητα συνημιτόνου, επειδή η κανονικοποίηση της Ευκλείδειας (L2) προβάλλει τα διανύσματα σε μοναδιαία σφαίρα και το γινόμενο τους είναι τότε το συνημίτονο της γωνίας μεταξύ των σημείων που συμβολίζονται με τα διανύσματα.

	Anime name	Similarity	Type
0	Naruto	1.000000	TV
1	Boruto: Naruto the Movie	0.854500	Movie
2	Naruto: Shippuuden Movie 4 - The Lost Tower	0.854500	Movie
3	Naruto: Shippuuden Movie 3 - Hi no Ishi wo Tsu...	0.854500	Movie
4	Naruto Soyokazeden Movie: Naruto to Mashin to ...	0.854500	Movie
5	Naruto x UT	0.853885	OVA
6	Boruto: Naruto the Movie - Naruto ga Hokage ni...	0.848793	Special
7	Naruto Shippuuden: Sunny Side Battle	0.848793	Special
8	Rekka no Honoo	0.648278	TV
9	Battle Spirits: Ryuuko no Ken	0.624475	OVA

Εικ. 3. Cosine Similarity recommendations for Naruto: Shippuden

Σύγκριση Απ' ότι βλέπουμε από τα αποτελέσματα των τριών kernel, περιέχουν τις ίδιες καταχωρήσεις απλά με διαφορετική σειρά και αυτο συμβαίνει γιατί ο κάθε kernel υπολογίζει με διαφορετικό τρόπο τις συσχετίσεις μεταξύ των ανιμε.Άρα με όποιον αλγόριθμο και να δουλέψουμε, οι προτάσεις θα είναι παρόμοιες.

5.4 Collaborative filtering

Pivot Matrix Αυτός ο Pivot Matrix αποτελείται από τίτλους για γραμμές και αναγνωριστικά χρηστών για στήλες, αυτό θα μας βοηθήσει να δημιουργήσουμε ένα sparse matrix που μπορεί να είναι πολύ χρήσιμος για την εύρεση της ομοιότητας συνημιτόνου.

Υπολογισμός όμοιων άνιμε(Item-Item) Για να κάνουμε μια νέα σύσταση, η ιδέα της μεθόδου Item-Item είναι να βρούμε αντικείμενα παρόμοια με αυτά με τα οποία ο χρήστης έχει ήδη «θετικά» αλληλεπιδράσει. Δύο στοιχεία θεωρούνται παρόμοια εάν οι περισσότεροι από τους χρήστες που έχουν αλληλεπιδράσει και με τα δύο το έκαναν με παρόμοιο τρόπο. Αυτή η μέθοδος λέγεται ότι είναι «στοιχειοκεντρική», καθώς αντιπροσωπεύει στοιχεία με βάση τις αλληλεπιδράσεις που είχαν οι χρήστες μαζί τους και αξιολογεί τις αποστάσεις μεταξύ αυτών των στοιχείων.

anime_title	0	001	009 Re:Cyborg	009-1	009-1: RandB	00:08	07-Ghost	1+2=Paradise	100%	100-man-nen Chikyuu no Tabi: Bander Book	1000-nen Joou: Queen Millennia	1001 Nights	11-nin Iru!	11eyes
anime_title														
0	1.000000	0.206862	0.031740	0.000000	0.000000	0.005087	0.002259	0.030909	0.000000	0.000000	-0.000546	-0.005825	0.057495	0.039000
001	0.206862	1.000000	0.009038	0.000000	0.000000	0.228795	0.062647	0.000000	0.000000	0.000000	0.000000	0.000000	0.016959	0.060775
009 Re:Cyborg	0.031740	0.009038	1.000000	0.071822	0.008172	0.007358	0.028168	0.063855	0.000000	0.130601	0.098002	-0.030331	0.013718	0.029751
009-1	0.000000	0.000000	0.071822	1.000000	0.513067	0.000000	0.064231	0.109574	0.132193	0.021769	-0.045496	0.063864	0.030671	0.027219
009-1: RandB	0.000000	0.000000	0.008172	0.513067	1.000000	0.000000	0.017089	0.100047	0.144983	0.000000	-0.110478	0.136589	0.014154	0.011943

5 rows × 9719 columns

Εικ. 4. Cosine Similarity for item-item matrix

Για να κάνουμε μια σύσταση για έναν συγκεκριμένο χρήστη. Αρχικά, θεωρούμε το στοιχείο που άρεσε περισσότερο σε αυτόν τον χρήστη και το αντιπροσωπεύουμε (όπως όλα τα άλλα στοιχεία) με το διάνυσμα αλληλεπίδρασής του με κάθε χρήστη («η στήλη του» στον πίνακα αλληλεπίδρασης). Στη συνέχεια, μπορούμε να υπολογίσουμε τις ομοιότητες μεταξύ του «καλύτερου αντικειμένου» και όλων των άλλων στοιχείων. Αφού υπολογιστούν οι ομοιότητες, μπορούμε να διατηρήσουμε τα επιλεγμένα «καλύτερα στοιχεία» που είναι νέα για τον χρήστη που μας ενδιαφέρει και να προτείνουμε αυτά τα στοιχεία.

Υπολογισμός όμοιων άνιμε(User-User) Προκειμένου να γίνει μια νέα σύσταση, η μέθοδος User-User προσπαθεί κατά προσέγγιση να προσδιορίσει τους χρήστες με το πιο παρόμοιο ”προφίλ αλληλεπιδράσεων” προκειμένου να προτείνει στοιχεία που είναι τα πιο δημοφιλή (και που είναι ”νέα” για τον χρήστη μας). Αυτή η μέθοδος λέγεται ότι είναι «επικεντρωμένη στο χρήστη», καθώς αντιπροσωπεύει τους χρήστες με βάση τις αλληλεπιδράσεις τους με αντικείμενα και αξιολογεί τις αποστάσεις μεταξύ των χρηστών.

user_id	5	7	17	46	123	160	210	226	248	256	271	294	321	392	39
user_id															
5	1.000000	0.066920	0.121267	0.078992	0.083271	0.058897	0.058062	-0.036328	0.024286	0.002557	-0.003032	0.050512	0.077466	0.161167	0.03965
7	0.066920	1.000000	0.131181	0.020610	0.016341	0.035607	0.010191	0.016517	0.128195	0.024388	0.075284	0.039792	0.077781	0.077336	0.10287
17	0.121267	0.131181	1.000000	0.061866	0.129657	0.108788	0.077822	0.028507	0.117661	0.098422	0.096431	0.116546	0.108458	0.147343	0.14554
46	0.078992	0.020610	0.061866	1.000000	0.144306	0.086872	0.035595	0.046569	0.040115	-0.008156	0.021238	0.053318	0.107638	0.104803	0.05659
123	0.083271	0.016341	0.129657	0.144306	1.000000	0.041138	0.058971	0.005630	0.032607	0.006792	0.077136	0.068480	0.058127	0.103337	0.06966

5 rows × 4349 columns

Εικ. 5. Cosine Similarity for user-user matrix

Για να κάνουμε μια σύσταση για έναν συγκεκριμένο χρήστη. Πρώτον, κάθε χρήστης μπορεί να αναπαρασταθεί από το διάνυσμα αλληλεπιδράσεων με τα διαφορετικά στοιχεία («η γραμμή του» στον πίνακα αλληλεπίδρασης). Στη συνέχεια, μπορούμε να υπολογίσουμε κάποιο είδος «ομοιότητας» μεταξύ του χρήστη που μας ενδιαφέρει και κάθε άλλου χρήστη. Αυτό το μέτρο ομοιότητας είναι τέτοιο ώστε δύο χρήστες με παρόμοιες αλληλεπιδράσεις στα ίδια στοιχεία θα πρέπει να θεωρούνται ως κοντινοί.

Μόλις υπολογιστούν οι ομοιότητες με όλους τους χρήστες, μπορούμε να διατηρήσουμε τους πιο παρόμοιους χρήστες και στη συνέχεια να προτείνουμε τα πιο δημοφιλή στοιχεία μεταξύ τους (εξετάζοντας μόνο τα στοιχεία με τα οποία ο χρήστης αναφοράς μας δεν έχει αλληλεπιδράσει ακόμη).

5.5 Σύγκριση user-user and item-item

Η μέθοδος user-user βασίζεται στην αναζήτηση παρόμοιων χρηστών όσον αφορά τις αλληλεπιδράσεις με αντικείμενα. Καθώς κάθε χρήστης έχει αλληλεπιδράσει μόνο με λίγα στοιχεία, καθιστά τη μέθοδο αρκετά ευαίσθητη σε τυχόν καταγεγραμμένες αλληλεπιδράσεις (υψηλή διακύμανση). Από την άλλη πλευρά, καθώς η τελική σύσταση βασίζεται μόνο σε αλληλεπιδράσεις που έχουν καταγραφεί για χρήστες παρόμοιους με τους χρήστες που μας ενδιαφέρουν, λαμβάνουμε πιο εξατομικευμένα αποτελέσματα (χαμηλή προκατάληψη).

Εικ. 6. user-user vs item-item method

Αντίθετα, η μέθοδος item-item βασίζεται στην αναζήτηση παρόμοιων αντικειμένων όσον αφορά τις αλληλεπιδράσεις χρήστη-αντικειμένου. Καθώς πολλοί χρήστες έχουν αλληλεπιδράσει με ένα αντικείμενο, η αναζήτηση είναι πολύ λιγότερο ευαίσθητη σε μεμονωμένες αλληλεπιδράσεις (χαμηλότερη διακύμανση). Αντίστοιχα, οι αλληλεπιδράσεις που προέρχονται από κάθε είδους χρήστες (ακόμη και χρήστες πολύ διαφορετικοί από τον χρήστη αναφοράς μας) λαμβάνονται υπόψη στη σύσταση, καθιστώντας τη μέθοδο λιγότερο εξατομικευμένη (πιο προκατειλημμένη). Έτσι, αυτή η προσέγγιση είναι λιγότερο εξατομικευμένη από την προσέγγιση user-user, αλλά πιο ισχυρή.

6 Συμπεράσματα

Μέσω της εργασίας μας μπορέσαμε να κατανοήσουμε πως υπάρχουν πολλοί και διαφορετικοί τρόποι για να κάνεις σύσταση περιεχομένου. Οι αλγόριθμοι συστάσεων μπορούν να χωριστούν σε δύο μεγάλα παραδείγματα: collaborative προσεγγίσεις (όπως user-user, item-item) που βασίζονται μόνο στην αλληλεπίδραση στοιχείου, χρήστη και σε content προσεγγίσεις που χρησιμοποιούν προηγούμενες πληροφορίες σχετικά με χρήστες ή/και στοιχεία. Οι περισσότεροι μέθοδοι οδηγούν στο ίδιο αποτέλεσμα,απλά με διαφορετική σειρά σύστασης,το οποίο επαληθεύεται και μέσω των πειραμάτων μας.Ο αλγορίθμος που χρησιμοποιήσαμε για να κάνουμε μια σύσταση για έναν συγκεκριμένο χρήστη,παρόλο που βγάζει συστάσεις, χρειάστηκε υπερβολικά πολύ χρόνο για μια εκτέλεση με αποτέλεσμα να είναι ανούσια η χρήση του.Επίσης, κρατήσαμε ότι η εξόρυξη σε μεγάλο όγκο δεδομένων χρειάζεται αρκετό χρόνο μεγάλη υπολογιστική δύναμη.Στη δικιά μας περίπτωση, τα δεδομένα που χρησιμοποιήσαμε είχαν μέγεθος 100 mb και για μια εκτέλεση όλης της εργασίας χρειαζόταν 2 ώρες και επειδή χρειαζόταν μεγάλη υπολογιστική δύναμη, το εκτελέσαμε στο google collab.

Αναφορές

1. Recommendation System Tutorial with Python using Collaborative Filtering https://pub.towardsai.net/recommendation-system-in-depth-tutorial-with-python-for-netflix-using-collaborative-filtering-533ff8a0e444
2. Recommendation systems: Principles, methods and evaluation https://www.sciencedirect.com/science/article/pii/S1110866515000341