ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Συστήματα συστάσεων με αναγνώριση και χρήση συναισθημάτων (Recommender systems with sentiment detection and exploitation)

Μιχαηλίδης Κυριάκος
Α.Μ. 2022201400136

Επιβλέπων:
Κωνσταντίνος Βασιλάκης
Καθηγητής

ΤΡΙΠΟΛΗ, ΝΟΕΜΒΡΙΟΣ 2021

Περίληψη

Τα συστήματα συστάσεων αποτελούν μία σημαντική τεχνολογία που έχει εισαχθεί δυναμικά στο διαδίκτυο προκειμένου να βοηθήσει τους χρήστες να αντιμετωπίσουν το φαινόμενο της πληροφοριακής υπερφόρτωσης. Ιδιαίτερα στον τομέα των ηλεκτρονικών αγορών, τα συστήματα συστάσεων χρησιμοποιούνται ευρύτατα, προκειμένου να προτείνουν στους καταναλωτές προϊόντα ή υπηρεσίες που είναι πιθανόν να τους αρέσουν, τα οποία τους προτείνονται υπό τη μορφή συστάσεων. Η αναγνώριση των προϊόντων ή υπηρεσιών που είναι πιθανόν να αρέσουν σε έναν χρήστη Χ πραγματοποιείται αξιοποιώντας τις βαθμολογίες που έχουν εισάγει άλλοι χρήστες: αρχικά εντοπίζονται χρήστες οι οποίοι έχουν εισάγει παρόμοιες βαθμολογίες με τον Χ, οι οποίοι κατηγοριοποιούνται ως «*κοντινοί γείτονες*» του Χ. Πρακτικά με αυτόν τον τρόπο εντοπίζονται χρήστες που έχουν παρόμοια ενδιαφέροντα και προτιμήσεις με τον χρήστη Χ. Στη συνέχεια, οι βαθμολογίες που έχουν εισάγει οι «κοντινοί γείτονες» για αντικείμενα που δεν έχει ήδη βαθμολογήσει ο χρήστης Χ, συνδυάζονται για να εκτιμήσουν τις βαθμολογίες που ο Χ θα έδινε για τα αντικείμενα αυτά. Τέλος, τα αντικείμενα που συγκεντρώνουν την υψηλότερη εκτιμώμενη βαθμολογία, είναι εκείνα που προτείνονται στον χρήστη Χ.

Στο ανωτέρω πλαίσιο υπάρχουν διάφορες παραλλαγές, τόσο ως προς τον υπολογισμό της εγγύτητας των χρηστών όσο και ως προς τον τρόπο διαμόρφωσης της εκτίμησης της βαθμολογίας που ο ένας χρήστης θα έδινε σε ένα αντικείμενο. Παραδείγματα τέτοιων παραλλαγών είναι η συνεκτίμηση του μέσου όρου των βαθμολογιών που έχει εισάγει ένας χρήστης ώστε να αντισταθμιστούν διαφορές μεταξύ χρηστών που βαθμολογούν αυστηρά με εκείνους που βαθμολογούν επιεικώς, η συνεκτίμηση του χρόνου που κατατέθηκε μία βαθμολογία ώστε να αντιμετωπίζεται το κύμα υψηλών βαθμολογιών που συνήθως συνοδεύει την αρχική εμφάνιση δημοφιλών προϊόντων ή ταινιών κινηματογράφου, η εξέταση της διακύμανσης των βαθμολογιών ώστε να εξομαλύνονται οι περιπτώσεις όπου κάποιοι χρήστες εισάγουν αξιολογήσεις κοντά στον μέσο όρο της κλίμακας, ενώ άλλοι χρήστες εισάγουν πιο ακραίες αξιολογήσεις (κοντά στα όρια της κλίμακας βαθμολογίας κ.ο.κ.).

Μία πρόσθετη διάσταση που μπορεί να ληφθεί υπ’ όψιν αφορά τα συναισθήματα που προκαλούν τα βαθμολογούμενα αντικείμενα στους χρήστες, και τα οποία εκτιμούμε ότι παίζουν ρόλο στη διαμόρφωση της βαθμολογίας. Για παράδειγμα, στην αξιολόγηση ενός εστιατορίου κάποιος χρήστης Υ μπορεί να έδωσε χαμηλή βαθμολογία γιατί αισθάνθηκε θυμωμένος από την κακή εξυπηρέτηση, ενώ ένας άλλος χρήστης Ζ να έδωσε χαμηλή βαθμολογία γιατί αισθάνθηκε αηδία λόγω της κακής καθαριότητας ή της κακής γεύσης του φαγητού. Μολονότι οι βαθμολογίες μπορεί να είναι εξ ίσου χαμηλές και για τους δύο χρήστες, οι λόγοι που τους οδήγησαν στη διαμόρφωση

της βαθμολογίας είναι διαφορετικοί, και έτσι -αντίστοιχα- διαφορετικός μπορεί να είναι ο βαθμός εγγύτητας που παρουσιάζουν οι χρήστες αυτοί με έναν τρίτο χρήστη Χ ο οποίος βαθμολόγησε με χαμηλό σκορ το εστιατόριο λόγω αηδίας: είναι εύλογο να υποθέσουμε ότι ο τρίτος αυτός χρήστης Χ βρίσκεται εγγύτερα προς τον δεύτερο χρήστη Z παρά προς τον πρώτο χρήστη Υ και έτσι η γνώμη του χρήστη Z πρέπει να ληφθεί ισχυρότερα υπ' όψιν, σε σχέση με αυτή του Υ, όταν διαμορφώνονται συστάσεις για τον χρήστη Χ. Η εξαγωγή των συναισθημάτων μπορεί να πραγματοποιηθεί εξετάζοντας τα κείμενα κριτικής που εισάγουν οι χρήστες παράλληλα με τις βαθμολογίες τους.

Στην παρούσα πτυχιακή εργασία πραγματοποιείται διερεύνηση σε σχέση με τον βαθμό που τα συναισθήματα που εκφράζουν οι χρήστες μπορούν να αξιοποιηθούν στη βελτίωση της ακρίβειας εκτίμησης των βαθμολογιών, και κατ' επέκταση στην επαύξηση της ποιότητας των συστάσεων. Για τον σκοπό αυτό αναπτύσσονται κατάλληλοι αλγόριθμοι, επεκτείνοντας τους αλγόριθμους του συνεργατικού φιλτραρίσματος, εξετάζονται παράμετροι που διέπουν τη λειτουργία τους, και ειδικότερα η βαρύτητα με την οποία ο παράγων των συναισθημάτων πρέπει να ληφθεί υπ' όψιν σε σχέση με αυτόν της αριθμητικής βαθμολογίας, και δοκιμάζονται σε σύνολα δεδομένων που περιέχουν τόσο αριθμητικές βαθμολογίες όσο και κειμενικές αξιολογήσεις.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Συστήματα συστάσεων με αναγνώριση και χρήση συναισθημάτων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Συστάσεις, αναγνώριση συναισθημάτων, χρήση συναισθημάτων, Υπολογισμός βαρών, Πρόβλεψη βαθμολογίας, Ομαδοποίηση

ABSTRACT

Recommendation systems are an important technology that has been introduced in order to help users deal with information overload. Especially in the field of e-shopping, recommendation systems are widely used to offer consumers products or services they are likely to like or use, which are offered as recommendations, which are offered in the form of recommendations. Identifying products or services that an X user might like is done using ratings entered by other users: first identifying users who have entered ratings similar to X are identified, who are categorized as "close neighbors" of user X. This identifies users who have similar interests and preferences to user X. Then, the scores entered by the "close neighbors" for items that user X has not already rated are combined to estimate the scores that X would give for those items. Finally, the items that collect the highest rated score are those that are suggested to user X.

In the above context, there are several variations, both in terms of calculating the proximity of users and in terms of how to configure the rating that a user would rate an item. Examples of such variations are the consideration of the average of the scores entered by a user to compensate for differences between users who score strictly with those who score leniently, the consideration of the time a score was submitted to deal with wave of high scores usually the initial appearance of popular products or movies, examining the fluctuation of ratings to normalize the cases where some users enter ratings close to the average of the scale, while other users enter more extreme ratings (close to the limits of the rating scale and so on.).

Additionally the emotions can be taken in concerns that the rated items evoke in the users, and which we believe play a role in shaping the rating. For example, in the rating of a restaurant, a user Y may have given a low rating because he felt angry about the poor service, while another user Z may have given a low rating because he felt disgusted due to poor cleanliness or bad taste of food. Although the scores may be equally low for both users, the reasons that led them to the rating are different, and so -respectively- the degree of closeness that these users have with a third user X may be different. who rated the restaurant with a low score due to disgust: it is reasonable to assume that this third user X is closer to the second user Z than to the first user Y and so the opinion of user Z should be taken more strongly into account, in relation to with that of Y, when recommendations are made for user X. Emotions can be extracted by examining the review texts that users enter alongside their ratings.

In this Thesis, we investigate the extent to which the emotions expressed by users can be used to improve the accuracy of ratings prediction, and consequently to increase the quality of recommendations. For this purpose, appropriate algorithms are developed, extending the algorithms of collaborative filtering, examining parameters that govern their operation, and in particular the importance with which the factor of emotions must be taken into account in relation to that of numerical scoring, and are tested. in datasets containing both numerical scores and textual evaluations.

1 Εισαγωγή

Η ταχύτατη ανάπτυξη του παγκόσμιου ιστού διευκόλυνε κατά πολύ τους χρήστες στο να έχουν πρόσβαση σε δισεκατομμύρια πληροφορίες και δεδομένα, καθώς όμως ταυτόχρονα δημιουργήθηκε ένα πρόβλημα, οι χρήστες δεν ήταν ικανοί να φιλτράρουν αποδοτικά τον τεράστιο όγκο πληροφοριών. Τα συστήματα συστάσεων εντοπίζουν με έξυπνο τρόπο τις προτιμήσεις του κάθε χρήστη και διαμορφώνουν συστάσεις, οι οποίες παραπέμπουν σε υλικό, μέσα στον τεράστιο όγκο πληροφοριών στον οποίο έχει πρόσβαση ο κάθε χρήστης, που εκτιμάται ότι θα εμπίπτει στις προτιμήσεις του. Από τις αρχές του 1990 έρευνες πάνω σε συστήματα συστάσεων έχουν δείξει μεγάλη αποτελεσματικότατα στη λύση του προβλήματος τις πληροφοριακής υπερφόρτωσης

Ένα σύστημα συστάσεων βασίζεται σε τρεις θεμελιώδη οντότητες, που συμπεριλαμβάνουν τους χρήστες, τα αντικείμενα (π.χ. ταινίες, ειδήσεις) και τη σχέση χρήστη-αντικειμένου (π.χ. αξιολογήσεις, σχόλια). Το κύριο καθήκον ενός συστήματος συστάσεων είναι να βρει ένα αποτελεσματικό και γενικευμένο μοτίβο που να περιγράφει τη σχέση χρήστη-αντικειμένων δηλαδή να εντοπίσει τους παράγοντες που προσδιορίζουν το πως συνδέεται ο χρήστης με τα αντικείμενα τα οποία προτιμά, λαμβάνοντας υπ' όψιν ενδεχομένως παραμέτρους όπως η χρονική εξέλιξη ή το πλαίσιο αναφοράς/την περιβάλλουσα πληροφορία (context). Από τη στιγμή που θα διαμορφωθεί το μοτίβο αυτό, μπορεί στη συνέχεια να χρησιμοποιηθεί για να προβλέπει άλλα αντικείμενα που θα άρεσαν στον χρήστη. Για την επίλυση του συγκεκριμένου προβλήματος προτάθηκαν πολλά μοντέλα τα τελευταία 30 χρόνια, κάθε ένα από τα οποία εφαρμόζει διαφορετικούς αλγορίθμους και εξετάζει ένα πλήθος από στοιχεία παραμέτρους.

1.1. Περιγραφή προβλήματος

Τα συστήματα συνεργατικού φιλτραρίσματος λαμβάνουν υπόψιν ως δεδομένα τα ενδιαφέροντα και τις αξιολογήσεις του του κάθε χρήστη και στη συνέχεια υπολογίζουν τον βαθμό ενδιαφέροντος του χρήστη για αντικείμενα που δεν έχει εξετάσει και διαμορφώνουν συστάσεις που μπορεί να ενδιαφέρουν τον συγκεκριμένο χρήστη. Τα συστήματα συστάσεων συνεργατικού φιλτραρίσματος αποθηκεύουν διάφορα στοιχεία όπως βαθμολογία που έχει εισάγει ένας χρήστης σε ένα προϊόν σε μια βάση δεδομένων αξιολογήσεων, ώστε να μπορούν στη συνέχεια να τα επεξεργαστούν και δώσουν συστάσεις παρόμοιων αντικειμένων που ίσως ενδιαφέρουν τον χρήστη. Η τεχνική συνεργατικού φιλτραρίσματος έχει αποδεχθεί ότι είναι επιτυχημένη σε συστήματα συστάσεων.

Αρχικά οι αλγόριθμοι συνεργατικού φιλτραρίσματος εντοπίζουν χρήστες με κοινά ενδιαφέροντα, για να γίνει αυτό θα πρέπει να εξετάσουμε αντικείμενα που έχουν ήδη βαθμολογηθεί από χρήστες. Όταν δύο χρήστες έχουν παραπλήσιες βαθμολογίες σε αρκετά αντικείμενα, μπορούμε να

θεωρήσουμε ότι έχουν κοινά ενδιαφέροντα. Αρχικά μπορούμε να θεωρήσουμε τον χρήστη Χ που έχει κάποια κοινά ενδιαφέροντα με έναν χρήστη Υ ως κοντινούς γείτονες, έπειτα για να εκτιμήσουμε τη βαθμολογία που θα μπορούσε να έχει αναθέσει ο χρήστης Χ σε ένα αντικείμενο που δεν έχει ήδη βαθμολογήσει, θα λάβουμε υπόψιν μας τον συνδυασμό των βαθμολογιών που έχουν αναθέσει στο ίδιο αντικείμενο οι κοντινοί γείτονες του Χ. Το συνεργατικό φιλτράρισμα έχει τη θεώρηση ότι αν τα ενδιαφέροντα χρηστών έχουν συμπέσει στο παρελθόν, πιθανότατα θα συμπέσουν και στο μέλλον.

Ένας άλλος παράγοντας που μπορούμε να θεωρήσουμε στη διαδικασία των εκτιμήσεων είναι και τα συναισθήματα που έχει ένας χρήστης όταν αναθέτει μια βαθμολογία σε ένα προϊόν. Υπάρχουν 6 βασικά συναισθήματα (έκπληξη, αηδία, φόβος, θύμος, χαρά, λύπη), και τα συναισθήματα αυτά μπορεί να επηρεάσουν κατά μεγάλο ποσοστό την βαθμολογία του κάθε χρήστη άλλα και να χρησιμοποιηθούν για τον υπολογισμό της ομοιότητας μεταξύ των χρηστών, προσδιορίζοντας έτσι τους κοντινούς γείτονες του. Για παράδειγμα, στην αξιολόγηση ενός εστιατορίου κάποιος χρήστης Υ μπορεί να έδωσε χαμηλή βαθμολογία γιατί αισθάνθηκε θυμωμένος από την κακή εξυπηρέτηση, ενώ ένας άλλος χρήστης Ζ να έδωσε χαμηλή βαθμολογία γιατί αισθάνθηκε αηδία λόγω της κακής καθαριότητας ή της κακής γεύσης του φαγητού. Μολονότι οι βαθμολογίες μπορεί να είναι εξ ίσου χαμηλές και για τους δύο χρήστες, οι λόγοι που τους οδήγησαν στη διαμόρφωση της βαθμολογίας είναι διαφορετικοί, και έτσι -αντίστοιχα- διαφορετικός μπορεί να είναι ο βαθμός εγγύτητας που παρουσιάζουν οι χρήστες αυτοί με έναν τρίτο χρήστη Χ ο οποίος βαθμολόγησε χαμηλά το εστιατόριο λόγω αηδίας: είναι εύλογο να υποθέσουμε ότι ο τρίτος αυτός χρήστης Χ βρίσκεται εγγύτερα προς τον δεύτερο χρήστη Ζ παρά προς τον πρώτο χρήστη Υ και έτσι η γνώμη του χρήστη Ζ πρέπει να ληφθεί ισχυρότερα υπ' όψιν, σε σχέση με αυτή του Υ, όταν διαμορφώνονται συστάσεις για τον χρήστη Χ. Η εξαγωγή των συναισθημάτων μπορεί να πραγματοποιηθεί εξετάζοντας τα κείμενα κριτικής που εισάγουν οι χρήστες παράλληλα με τις βαθμολογίες τους.

Στην παρούσα πτυχιακή εργασία πραγματοποιείται διερεύνηση σε σχέση με τον βαθμό που τα συναισθήματα που εκφράζουν οι χρήστες μπορούν να αξιοποιηθούν στη βελτίωση της ακρίβειας της εκτίμησης των βαθμολογιών, και κατ' επέκταση στην επαύξηση της ποιότητας των συστάσεων. Για τον σκοπό αυτό αναπτύσσονται κατάλληλοι αλγόριθμοι, επεκτείνοντας τους αλγόριθμους του συνεργατικού φιλτραρίσματος, εξετάζονται παράμετροι που διέπουν τη λειτουργία τους, και ειδικότερα η βαρύτητα με την οποία ο παράγων των συναισθημάτων πρέπει να ληφθεί υπ' όψιν σε σχέση με αυτόν της αριθμητικής βαθμολογίας, και δοκιμάζονται σε σύνολα δεδομένων που περιέχουν τόσο αριθμητικές βαθμολογίες όσο και κειμενικές αξιολογήσεις.

Η παρούσα πτυχιακή εργασία διαρθρώνεται ως ακολούθως: στο κεφάλαιο 2 παρουσιάζονται οι ευρύτερα χρησιμοποιούμενες μετρικές ομοιότητας χρηστών και εισάγεται επέκταση των μετρικών

αυτών ώστε να λαμβάνεται υπ' όψιν το συναίσθημα. Επίσης δίνεται μία εισαγωγή στην τεχνική του συνεργατικού φιλτραρίσματος. Στο κεφάλαιο 3 παρουσιάζεται η ενσωμάτωση των συναισθημάτων στον υπολογισμό της ομοιότητας των χρηστών. Στο κεφάλαιο 4 παρουσιάζεται η υλοποίηση και τα τεχνολογικά στοιχεία που χρησιμοποιήθηκαν σε αυτή, ενώ στο κεφάλαιο 5 παρουσιάζονται τα αποτελέσματα των δοκιμών από την εκτέλεση του αλγόριθμου. Τέλος, στο κεφάλαιο 6 εξάγονται τα συμπεράσματα και σκιαγραφούνται πιθανές μελλοντικές επεκτάσεις.

2. Συνεργατικό φιλτράρισμα και μετρικές ομοιότητας χρηστών

Το συνεργατικό φιλτράρισμα είναι μία τεχνική η οποία, για κάθε χρήστη Χ, εντοπίζει τους χρήστες που επιδεικνύουν παρόμοιες προτιμήσεις με τον Χ, και στη συνέχεια χρησιμοποιεί τις αξιολογήσεις των χρηστών αυτών (οι οποίοι καλούνται *κοντινοί γείτονες*) για να εκτιμήσει τις αξιολογήσεις που θα έδινε ο Χ σε αντικείμενα που δεν έχει βαθμολογήσει ακόμη ο ίδιος (αλλά έχουν βαθμολογήσει οι κοντινοί γείτονές του). Στη συνέχεια, με βάση τις εκτιμήσεις διαμορφώνεται η σύσταση κάποιων αντικειμένων προς τον χρήστη. Στη διαδικασία αυτή διακρίνουμε τα εξής στάδια:

1. εύρεση των κοντινών γειτόνων,
2. υπολογισμός εκτιμήσεων αξιολογήσεων,
3. διαμόρφωση σύστασης.

Για την εύρεση των κοντινών γειτόνων είναι απαραίτητο να μπορούμε να ποσοτικοποιήσουμε το πόσο παρόμοιες είναι οι αξιολογήσεις δύο χρηστών. Αυτό πραγματοποιείται μέσω *μετρικών ομοιότητας*, οι οποίες είναι μαθηματικές συναρτήσεις που λαμβάνουν ως είσοδο τις αξιολογήσεις δύο χρηστών και παράγουν ως έξοδο μία τιμή που αντανακλά τον βαθμό ομοιότητας των αξιολογήσεων. Έχοντας ποσοτικοποιήσει την ομοιότητα των χρηστών, για κάθε χρήστη Χ στη συνέχεια επιλέγονται οι K χρήστες που έχουν τη μεγαλύτερη ομοιότητα με τον Χ, και αυτοί αποτελούν το σύνολο των κοντινών γειτόνων. Τυπικά μπορούμε να διατηρούμε 20-50 κοντινούς γείτονες.

Κατόπιν, στο στάδιο του υπολογισμού εκτιμήσεων αξιολογήσεων για τον χρήστη Χ, λαμβάνονται υπ' όψιν οι ομοιότητες του Χ με τους κοντινούς του γείτονες και οι βαθμολογίες των κοντινών γειτόνων με τα αντικείμενα που δεν έχει ήδη αξιολογήσει ο Χ, προκειμένου να διαμορφωθούν οι εκτιμήσεις των αξιολογήσεων.

Τέλος, στη διαμόρφωση της σύστασης, τυπικά επιλέγονται τα αντικείμενα που έχουν τις υψηλότερες εκτιμήσεις βαθμολογίας και προτείνονται στον χρήστη.

Στη συνέχεια του παρόντος κεφαλαίου, περιγράφουμε τις κυριότερες μετρικές ομοιότητας, εισάγουμε την ομοιότητα που λαμβάνει υπ' όψιν και τα συναισθήματα, και τέλος περιγράφουμε τον τρόπο υπολογισμού των εκτιμήσεων των αξιολογήσεων.

2.1. Μετρικές ομοιότητας χρηστών

Δύο από τις πιο διαδεδομένες μετρικές που χρησιμοποιούνται σε συστήματα συνεργατικού φιλτραρίσματος για την εύρεση κοντινών γειτόνων χρηστών είναι οι Pearson Correlation Coefficient (PCC) και Cosine Similarity (CS).

2.1.1. Η μετρική Cosine Similarity

Η μετρική Cosine Similarity θεωρεί τις αξιολογήσεις ν-αντικειμένων από έναν χρήστη ως ένα ν-διάστατο διάνυσμα, όπου κάθε διάσταση αντιστοιχεί στην αξιολόγηση ενός διακριτού αντικειμένου. Η ομοιότητα μεταξύ δύο τέτοιων διανυσμάτων X και Y υπολογίζεται ως το συνημίτονο της γωνίας που σχηματίζουν τα δύο διανύσματα. Δύο ταυτιζόμενα διανύσματα θα σχηματίζουν γωνία ίση με μηδέν και άρα το συνημίτονό της θα είναι 1, ενώ δύο αντίθετα διανύσματα θα σχηματίζουν γωνία ίση με π και άρα το συνημίτονό της θα είναι ίσο με -1. Ο υπολογισμός του συνημιτόνου της γωνίας πραγματοποιείται με χρήση της μαθηματικής εξίσωσης του εσωτερικού γινομένου, καθώς ισχύει ότι

$$\vec{x} * \vec{y} = |\vec{x}| * |\vec{y}| * \cos(\theta)$$

όπου $|\vec{x}|$ και $|\vec{y}|$ είναι τα μέτρα των διανυσμάτων $\vec{x}$ και $\vec{y}$ αντίστοιχα και θ είναι η γωνία που σχηματίζουν. Ο τύπος αυτός μετασχηματίζεται σε

$$\cos(\theta) = \frac{\vec{x} * \vec{y}}{|\vec{x}| * |\vec{y}|}$$

Το εσωτερικό γινόμενο δύο διανυσμάτων ορίζεται και ως το άθροισμα των γινομένων των επιμέρους συνιστωσών, δηλαδή αν έχουμε δύο διανύσματα $x=(x_1, x_2, \ldots, x_n)$ και $y=(y_1, y_2, \ldots, y_n)$, τότε

$$\vec{x} * \vec{y} = x_1 y_1 + x_2 y_2 + \ldots + x_v y_v$$

Επίσης, το μέτρο $|\vec{x}|$ ενός διανύσματος $\vec{x}$ προκύπτει ως εξής:

$$|\vec{x}| = \sqrt{\sum_{i=1}^{v} x_i^2}$$

Συνδυάζοντας τα ανωτέρω, έχουμε ότι η ομοιότητα συνημιτόνου μεταξύ δύο διανυσμάτων $\vec{x}$ και $\vec{y}$ υπολογίζεται ως:

$$CS(\vec{x}, \vec{y}) = \frac{x_1 y_1 + x_2 y_2 + \ldots + x_v y_v}{\sqrt{\sum_{i=1}^{v} x_i^2} * \sqrt{\sum_{i=1}^{v} y_i^2}} = \frac{\sum_{i=1}^{v} x_i * y_i}{\sqrt{\sum_{i=1}^{v} x_i^2} * \sqrt{\sum_{i=1}^{v} y_i^2}}$$

Καθώς τυπικά οι δύο χρήστες X και Y, στους οποίους αντιστοιχούν τα διανύσματα αξιολογήσεων $\vec{x}$ και $\vec{y}$ (α) δεν έχουν βαθμολογήσει όλα τα αντικείμενα και (β) δεν έχουν αξιολογήσει τα *ίδια αντικείμενα* (δηλαδή υπάρχουν αντικείμενα που έχει βαθμολογήσει ο X αλλά όχι ο Y και αντίστοιχα αντικείμενα που έχει βαθμολογήσει ο Y αλλά όχι ο X), στον παραπάνω τύπο χρησιμοποιούνται μόνο οι συνιστώσες των $\vec{x}$ και $\vec{y}$ που αντιστοιχούν σε αντικείμενα που έχουν βαθμολογήσει και οι δύο χρήστες. Εν τέλει, αν X και Y δύο χρήστες, R(X) και R(Y) τα αντικείμενα που έχουν αντίστοιχα βαθμολογήσει, και με $r_{U,i}$ συμβολίσουμε τη βαθμολογία που έδωσε ο χρήστης *U* στο αντικείμενο *i*, η ομοιότητα συνημιτόνου των δύο χρηστών X και Y υπολογίζεται ως εξής:

$$CS(X,Y)=\frac{\sum_{i\in R(X)\cap R(Y)} r_{X,i}*r_{Y,i}}{\sqrt{\sum_{i\in R(X)\cap R(Y)} r_{X,i}^2}*\sqrt{\sum_{i\in R(X)\cap R(Y)} r_{Y,i}^2}}$$

2.1.2. Η μετρική Pearson Correlation Coefficient

Η μετρική cosine similarity αδυνατεί να λάβει υπ' όψιν τις διαφορετικές πρακτικές βαθμολόγησης που μπορεί να ακολουθούν δύο χρήστες. Για παράδειγμα, εάν ένας χρήστης X βαθμολογήσει τρία αντικείμενα με τις βαθμολογίες (1, 5, 7) και ένας χρήστης Y βαθμολογήσει τα ίδια αντικείμενα με τις βαθμολογίες (5, 7, 9) είναι σαφές ότι οι βαθμολογίες αυτές είναι ταυτόσημες με τη διαφορά ότι ο πρώτος χρήστης βαθμολογεί συνολικά πιο αυστηρά από τον πρώτο (4 μονάδες πιο χαμηλά). Η μετρική cosine similarity θα υπολογίσει γι' αυτούς τους χρήστες ομοιότητα ίση με 0.955, η οποία είναι μεν αρκετά υψηλή, ωστόσο υπάρχει περιθώριο βελτίωσης.

Αυτή την αδυναμία έρχεται να καλύψει η μετρική Pearson Correlation Coefficient, η οποία υπολογίζει τον μέσο όρο των βαθμολογιών που έχει δώσει ο κάθε χρήστης και κατόπιν αφαιρεί από την κάθε βαθμολογία αυτόν τον μέσο όρο. Έτσι, αντικείμενα που έχουν βαθμολογηθεί κάτω από τον μέσο όρο θα καταλήξουν με αρνητική βαθμολογία, ενώ αντίστοιχα αντικείμενα που έχουν βαθμολογηθεί πάνω από τον μέσο όρο θα καταλήξουν με θετική βαθμολογία. Οι προσαρμοσμένες αυτές βαθμολογίες θα χρησιμοποιηθούν στη συνέχεια με τον ίδιο τρόπο που παρουσιάστηκε για τη μετρική cosine similarity, έτσι ώστε να υπολογιστεί η τιμή της μετρικής Pearson Correlation Coefficient.

Σύμφωνα με τα παραπάνω, η ομοιότητα μεταξύ δύο χρηστών σύμφωνα με τη μετρική Pearson Correlation Coefficient (PCC) υπολογίζεται ως εξής:

$$PCC(X,Y)=\frac{\sum_{i\in R(X)\cap R(Y)} (r_{X,i}-\bar{r_X})*(r_{Y,i}-\bar{r_Y})}{\sqrt{\sum_{i\in R(X)\cap R(Y)} (r_{X,i}-\bar{r_X})^2}*\sqrt{\sum_{i\in R(X)\cap R(Y)} (r_{Y,i}-\bar{r_Y})^2}}$$

όπου $\bar{r}_X$ και $\bar{r}_Y$ είναι αντίστοιχα η μέση τιμή των βαθμολογιών που έχουν εισάγει οι χρήστες X και Y.

Για τους δύο χρήστες X, Y του ανωτέρω παραδείγματος, η μετρική *PCC(X, Y)* θα είναι ίση με 1.

2.2. Υπολογισμός εκτιμήσεων αξιολογήσεων

Η εκτίμηση των αξιολογήσεων που θα δώσει ένας χρήστης U για αντικείμενα που δεν έχει ήδη αξιολογήσει, πραγματοποιείται με τη βοήθεια των αξιολογήσεων που έχουν εισάγει οι κοντινοί γείτονες του U για τα αντικείμενα αυτά και την ποσοτικοποίηση της ομοιότητας του U με τους κοντινούς του γείτονες.

Συνηθισμένοι μαθηματικοί τύποι για τον υπολογισμό της εκτίμησης $p_{U,i}$ για την αξιολόγηση που θα έδινε ο χρήστης U στο αντικείμενο i είναι οι ακόλουθοι:

2.2.1. Μέσος όρος

Η τεχνική αυτή υπολογίζει ως εκτίμηση $p_{U,i}$ τον μέσο όρο των αξιολογήσεων που έχουν εισάγει οι κοντινοί γείτονες του U για το αντικείμενο i. Τυπικά αυτό εκφράζεται ως:

$$p_{U,i} = \frac{1}{|V \in NN(U) \wedge r_{V,i} \neq NULL|} \sum_{V \in NN(U) \wedge r_{V,i} \neq NULL} r_{V,i}$$

όπου $r_{V,i}$ είναι η βαθμολογία που έχει δώσει ο χρήστης V στο αντικείμενο i. Για τον υπολογισμό της εκτίμησης για το αντικείμενο *ι* λαμβάνονται υπ' όψιν μόνο οι κοντινοί γείτονες που έχουν αξιολογήσει το αντικείμενο αυτό.

2.2.2. Σταθμισμένος μέσος όρος

Η τεχνική αυτή είναι ανάλογη με την προηγούμενη, με τη διαφορά ότι η βαθμολογία του κάθε κοντινού γείτονα *V* του χρήστη *U* (δηλαδή του χρήστη για τον οποίο υπολογίζεται η εκτίμηση) που έχει αξιολογήσει το αντικείμενο *i* σταθμίζεται με έναν συντελεστή βάρους ίσο με την ομοιότητα των *U* και *V*. Τυπικά αυτό εκφράζεται ως:

$$p_{U,i} = \frac{\sum_{V \in NN(U) \wedge r_{V,i} \neq NULL} s_{U,V} r_{V,i}}{\sum_{V \in NN(U) \wedge r_{V,i} \neq NULL} s_{U,V}}$$

όπου $r_{V,i}$ είναι η βαθμολογία που έχει δώσει ο χρήστης V στο αντικείμενο I και $s_{U,V}$ είναι η ομοιότητα μεταξύ των χρηστών U και V. Και πάλι, τον υπολογισμό της εκτίμησης για το αντικείμενο *ι* λαμβάνονται υπ' όψιν μόνο οι κοντινοί γείτονες που έχουν αξιολογήσει το αντικείμενο αυτό.

2.2.3. Σταθμισμένος μέσος όρος με προσαρμογή μέσου όρου

Η τεχνική αυτή ακολουθεί τη λογική της στάθμισης της συνεισφοράς κάθε κοντινού γείτονα *V* του χρήστη *U* (δηλαδή του χρήστη για τον οποίο υπολογίζεται η εκτίμηση) με βάση την ομοιότητα μεταξύ των χρηστών U και V, επιπρόσθετα όμως προσπαθεί να αντιμετωπίσει το φαινόμενο όπου διαφορετικοί χρήστες εφαρμόζουν διαφορετικά επίπεδα "αυστηρότητας" στη βαθμολογία τους. Έτσι, οι βαθμολογίες δεν εξετάζονται με όρους απόλυτης τιμής, αλλά "κανονικοποιούνται" αφαιρώντας από κάθε βαθμολογία τον μέσο όρο βαθμολογιών του χρήστη στον οποίο η εκάστοτε βαθμολογία ανήκει. Με τον τρόπο αυτό, αν η βαθμολογία $r_{U,I1}$ ενός χρήστη U για ένα αντικείμενο I_1 βρίσκεται πάνω από τον μέσο όρο των βαθμολογιών $\bar{r}_U$ που έχει εισάγει ο χρήστης U, τότε η βαθμολογία $r_{U,I1}$ θα κανονικοποιηθεί σε μία τιμή μεγαλύτερη του μηδενός. Αντίστροφα, αν η βαθμολογία $r_{U,I2}$ του χρήστη U για ένα αντικείμενο I_2 βρίσκεται κάτω από τον μέσο όρο των βαθμολογιών $\bar{r}_U$ που έχει εισάγει ο χρήστης U, τότε η βαθμολογία $r_{U,I2}$ θα κανονικοποιηθεί σε μία τιμή μικρότερη του μηδενός.

Θεωρώντας τις κανονικοποιημένες τιμές, προκειμένου να υπολογιστεί η εκτίμηση $p_{U,i}$ για την αξιολόγηση που θα έδινε ο χρήστης U για το αντικείμενο i, υπολογίζεται ο σταθμισμένος μέσος των *κανονικοποιημένων* αξιολογήσεων που έχουν δώσει οι κοντινοί γείτονες του U για το αντικείμενο i, καταλήγοντας έτσι σε μία ποσότητα που είναι μεγαλύτερη του μηδενός, αν εκτιμάται ότι ο U θα βαθμολογούσε το i με βαθμό πάνω από τον προσωπικό του μέσο όρο $\bar{r}_U$, ενώ καταλήγουμε σε μία ποσότητα που είναι μικρότερη του μηδενός, αν εκτιμάται ότι ο U θα βαθμολογούσε το i με βαθμό κάτω από τον προσωπικό του μέσο όρο $\bar{r}_U$. Σε αυτή την ποσότητα, προσθέτουμε τέλος τον προσωπικό μέσο όρο $\bar{r}_U$ του χρήστη U, προκειμένου να εξάγουμε την τελική εκτίμηση $p_{U,i}$. Συνθέτοντας τα παραπάνω, η εκτίμηση $p_{U,i}$ υπολογίζεται ως:

αυτό εκφράζεται ως:

$$p_{U,i} = \bar{r}_U + \frac{\sum_{V \in NN(U) \wedge r_{V,i} \neq NULL} s_{U,V}(r_{V,i} - \bar{r}_V)}{\sum_{V \in NN(U) \wedge r_{V,i} \neq NULL} s_{U,V}}$$

όπου $r_{V,i}$ είναι η βαθμολογία που έχει δώσει ο χρήστης V στο αντικείμενο i, $s_{U,V}$ είναι η ομοιότητα μεταξύ των χρηστών U και V και $\bar{r}_U$ (αντίστοιχα $\bar{r}_V$) είναι ο μέσος όρος των αξιολογήσεων που έχει εισάγει ο χρήστης U (αντίστοιχα V). Και πάλι, τον υπολογισμό της εκτίμησης για το αντικείμενο *ι* λαμβάνονται υπ' όψιν μόνο οι κοντινοί γείτονες που έχουν αξιολογήσει το αντικείμενο αυτό.

3. Ενσωμάτωση συναισθημάτων στις μετρικές ομοιότητας

Οι μετρικές ομοιότητας, όπως περιγράφηκαν στο προηγούμενο κεφάλαιο, βασίζονται μόνο στην τελική αξιολόγηση του χρήστη. Ωστόσο, οι χρήστες σε πολλές πλατφόρμες εισάγουν παράλληλα με την αριθμητική αξιολόγηση και κριτικές σε μορφή κειμένου. Τα κείμενα εκφράζουν με πλούσιο τρόπο τις απόψεις του χρήστη για το αντικείμενο ή συγκεκριμένα χαρακτηριστικά αυτού (π.χ. την πλοκή, την ηθοποιία και τη σκηνοθεσία σε μία ταινία, την οθόνη, τη μπαταρία και την ευχρηστία σε ένα έξυπνο τηλέφωνο) και έτσι τα κείμενα μπορούν να αποτελέσουν σημαντική πηγή πληροφοριών για τη διαδικασία του συνεργατικού φιλτραρίσματος.

Μία διάσταση που μπορεί να εξαχθεί από τα κείμενα και που μπορεί να ληφθεί υπ' όψιν, αφορά τα συναισθήματα που προκαλούν τα βαθμολογούμενα αντικείμενα στους χρήστες, και τα οποία εκτιμούμε ότι παίζουν ρόλο στη διαμόρφωση της βαθμολογίας. Για παράδειγμα, στην αξιολόγηση ενός εστιατορίου κάποιος χρήστης Y μπορεί να έδωσε χαμηλή βαθμολογία γιατί αισθάνθηκε θυμωμένος από την κακή εξυπηρέτηση, ενώ ένας άλλος χρήστης Z να έδωσε χαμηλή βαθμολογία γιατί αισθάνθηκε αηδία λόγω της κακής καθαριότητας ή της κακής γεύσης του φαγητού. Μολονότι οι βαθμολογίες μπορεί να είναι εξ ίσου χαμηλές και για τους δύο χρήστες, οι λόγοι που τους οδήγησαν στη διαμόρφωση της βαθμολογίας είναι διαφορετικοί, και έτσι -αντίστοιχα- διαφορετικός μπορεί να είναι ο βαθμός εγγύτητας που παρουσιάζουν οι χρήστες αυτοί με έναν τρίτο χρήστη X ο οποίος βαθμολόγησε με χαμηλό σκορ το εστιατόριο λόγω αηδίας: είναι εύλογο να υποθέσουμε ότι ο τρίτος αυτός χρήστης X βρίσκεται εγγύτερα προς τον δεύτερο χρήστη Z παρά προς τον πρώτο χρήστη Y και έτσι η γνώμη του χρήστη Z πρέπει να ληφθεί ισχυρότερα υπ' όψιν, σε σχέση με αυτή του Y, όταν διαμορφώνονται συστάσεις για τον χρήστη X. Η εξαγωγή των συναισθημάτων μπορεί να πραγματοποιηθεί εξετάζοντας τα κείμενα κριτικής που εισάγουν οι χρήστες παράλληλα με τις βαθμολογίες τους.

Στην παρούσα πτυχιακή εργασία διερευνούμε την αξιοποίηση των συναισθημάτων για τη διαμόρφωση των συστάσεων και ειδικότερα στο στάδιο του υπολογισμού της ομοιότητας χρηστών. Ειδικότερα, η προσέγγιση έχει ως εξής:

1. τα κείμενα των αξιολογήσεων υπόκεινται σε επεξεργασία ώστε να εξαχθούν εκτιμήσεις για έξι βασικά συναισθήματα: θυμό, αηδία, φόβο, χαρά, λύπη και έκπληξη. Οι εκτιμήσεις εξάγονται με τη βοήθεια λογισμικού συναισθηματικής ανάλυσης κειμένου, και για κάθε συναίσθημα υπολογίζεται μία τιμή από 0 έως 1 που απεικονίζει τη σχετική συνεισφορά του κάθε συναισθήματος στη συναισθηματική ένταση του κειμένου. Το άθροισμα όλων των τιμών που αντιστοιχίζονται στα συναισθήματα είναι πάντα ίσο με 1. Αυτό σημαίνει ότι αν ένα συναίσθημα είναι απόλυτα κυρίαρχο σε μία κριτική, τότε η τιμή γι' αυτό θα είναι κοντά

στο 1 και για τα υπόλοιπα συναισθήματα θα έχουν τιμές κοντά στο μηδέν. Ανάλογα προσαρμόζονται οι τιμές σε περίπτωση που κυριαρχεί μία πλειάδα συναισθημάτων (π.χ. χαρά και ενθουσιασμός).

2. Τα συναισθήματα που εξάγονται για κάθε κειμενική κριτική χρησιμοποιούνται για να εμπλουτιστεί ο πίνακας αξιολογήσεων των χρηστών για τα αντικείμενα. Πρακτικά, οι εκτιμήσεις των συναισθημάτων προστίθενται ως επιπλέον διαστάσεις σε κάθε στοιχείο $r_{U,i}$ του πίνακα αξιολογήσεων, και έτσι κάθε τέτοιο στοιχείο από απλή αριθμητική τιμή μετατρέπεται σε ένα διάνυσμα 7 διαστάσεων (συνολικήΒαθμολογία, θυμός, αηδία, φόβος, χαρά, λύπη, έκπληξη).
3. Στον εμπλουτισμένο πίνακα εφαρμόζεται μία τεχνική υπολογισμού ομοιότητας, η οποία επεκτείνει μία από τις προαναφερθείσες τεχνικές (cosine similarity, Pearson correlation coefficient) ώστε να μπορεί να χειριστεί πολυδιάστατους χώρους. Στο σημείο αυτό γεννώνται τα εξής ζητήματα:
    - οι αριθμητικές αξιολογήσεις (βαθμολογίες) είναι εκφρασμένες σε κάποια κλίμακα, συνήθως [1, 5] ή [1, 10], ενώ τα συναισθήματα εκφράζονται σε κάποια διαφορετική κλίμακα, [0, 1]. Έτσι, θα πρέπει να ληφθεί μέριμνα ώστε να μην κυριαρχείται (dominated) ο υπολογισμός της τιμής ομοιότητας από την τιμή της βαθμολογίας. Για τον σκοπό αυτό, πραγματοποιείται κλιμάκωση των αριθμητικών βαθμολογιών ώστε να έρθουν σε μία συγκρίσιμη κλίμακα με αυτή στην οποία εκφράζονται τα συναισθήματα.
    - Κάποια συναισθήματα εκτιμάται ότι θα έχουν πολύ σημαντικότερη συνεισφορά στη διαμόρφωση της τελικής αξιολόγησης. Για παράδειγμα, εκτιμάται ότι ο φόβος δεν θα διαδραματίσει σημαντικό ρόλο, καθώς η βαθμολογία προκύπτει περισσότερο από την εμπειρία, ενώ ο φόβος εκφράζει μία αβεβαιότητα για μία δυσάρεστη μελλοντική εξέλιξη. Για τον λόγο αυτό, σε κάθε συναίσθημα ανατίθεται μία βαρύτητα και η βαρύτητα αυτή χρησιμοποιείται στον μαθηματικό τύπο υπολογισμού της μετρικής ομοιότητας μεταξύ χρηστών.

Στις επόμενες υποενότητες περιγράφονται οι επεκτάσεις των μετρικών cosine similarity και Pearson correlation coefficient, έτσι ώστε στη διαδικασία υπολογισμού ομοιότητας να λαμβάνουν υπ’ όψιν (α) τα συναισθήματα και (β) τις παραμέτρους κλιμάκωσης και ανάθεσης βαρύτητας που αναφέρθηκαν ανωτέρω. Στις υποενότητες που ακολουθούν, χρησιμοποιούνται οι εξής συμβολισμοί:

Συμβολισμός	Επεξήγηση
D	Το σύνολο των διαστάσεων που εξετάζονται, οι οποίες είναι {συνολικήΒαθμολογία, θυμός, αηδία, φόβος, χαρά, λύπη, έκπληξη}
$r_{U,i}(d)$	Η διάσταση $d \in D$ της αξιολόγησης του χρήστη U για το αντικείμενο i.
w_d	Το βάρος (weight) που ανατίθεται στη διάσταση d.
sc_d	Ο συντελεστής κλιμάκωσης (scaling) για τη διάσταση d.
$\overline{r_U(d)}$	Η μέση τιμή της διάστασης d για τις αξιολογήσεις του χρήστη U.

Πίνακας 1: Συμβολισμοί για τις εκτεταμένες μορφές των μετρικών ομοιότητας

3.1. Η εκτεταμένη μορφή της μετρικής Cosine Similarity

Η εκτεταμένη μορφή της μετρικής Cosine Similarity φαίνεται στον μαθηματικό τύπο που ακολουθεί:

$$CS_{sent}(X,Y)=\frac{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * r_{X,i}(d) * r_{Y,i}(d)}{\sqrt{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * r_{X,i}^2} * \sqrt{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * r_{Y,i}^2}}$$

Όπως παρατηρούμε, ο εκτεταμένος τύπος διατρέχει όλες τις διαστάσεις, συνυπολογίζοντάς τες στο αποτέλεσμα, εφαρμόζοντας τόσο την κλιμάκωση όσο και τη βάρυνση.

3.2. Η εκτεταμένη μορφή της μετρικής Pearson Correlation Coefficient

Η εκτεταμένη μορφή της μετρικής Cosine Similarity φαίνεται στον μαθηματικό τύπο που ακολουθεί:

$$PCC_{sent}(X,Y)=\frac{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * (r_{X,i}(d)-\overline{r_X(d)}) * (r_{Y,i}(d)-\overline{r_Y(d)})}{\sqrt{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * (r_{X,i}(d)-\overline{r_X(d)})^2} * \sqrt{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * (r_{Y,i}(d)-\overline{r_Y(d)})^2}}$$

Όπως παρατηρούμε, ο εκτεταμένος τύπος διατρέχει όλες τις διαστάσεις, συνυπολογίζοντάς τες στο αποτέλεσμα, εφαρμόζοντας τόσο την κλιμάκωση όσο και τη βάρυνση. Και στην εκτεταμένη του μορφή, ο τύπος της Pearson Correlation Coefficient διαφέρει από τον τύπο της Cosine Similarity κατά το ότι προσαρμόζει την κάθε βαθμολογία αφαιρώντας τον μέσο όρο του συγκεκριμένου χρήστη. Αυτό συμβαίνει σε όλες τις εξεταζόμενες διαστάσεις.

4. Συστήματα συστάσεων: Εφαρμόζοντας την τεχνική συνεργατικού φιλτραρίσματος με συνεκτίμηση των συναισθημάτων χρήστη

Σκοπός τις πτυχιακής ήταν να υλοποιηθεί λογισμικό το οποίο (α) θα υπολογίζει μετρικές για το συναίσθημα από τα κείμενα αξιολογήσεων που συνοδεύουν τις βαθμολογίες, (β) θα αξιοποιεί τις μετρικές για το συναίσθημα στον υπολογισμό της ομοιότητας μεταξύ των χρηστών, με βάση τις εκτεταμένες μορφές των μετρικών Cosine Similarity και Pearson Correlation Coefficient που παρατέθηκαν στο κεφάλαιο 3, (γ) θα υπολογίζει τις εκτιμήσεις για βαθμολόγηση αντικειμένων με βάση τις εκτεταμένες μετρικές ομοιότητας και τις βαθμολογίες των κοντινών γειτόνων και (δ) θα υπολογίζει στατιστικά για την ακρίβεια των εκτιμήσεων.

Για την ανάπτυξη του λογισμικού χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python και συγκεκριμένα η έκδοση 3.7 της Python και το περιβάλλον για την ανάπτυξη του κώδικα ήταν το PyCharm 2020.2.1 (Community Edition). Ο υπολογισμός των μετρικών για τα συναισθήματα από τα κείμενα των αξιολογήσεων έγινε με το λογισμικό "Twitter Emotion Recognition" (https://github.com/nikicc/twitter-emotion-recognition), ενώ για την αξιολόγηση χρησιμοποιήθηκαν τα datasets της Amazon που είναι διαθέσιμα στο https://jmcauley.ucsd.edu/data/amazon/. Για την επεξεργασία των δεδομένων αξιοποιήθηκε η βιβλιοθήκη Pandas[1]. Στις ακόλουθες παραγράφους περιγράφονται τα βασικότερα χαρακτηριστικά των στοιχείων που χρησιμοποιήθηκαν στην υλοποίηση, καθώς και τα βασικά χαρακτηριστικά της υλοποίησης.

4.1. Γλώσσα προγραμματισμού και περιβάλλον ανάπτυξης

Όπως αναφέρθηκε, για την υλοποίηση του λογισμικού χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python και συγκεκριμένα η έκδοση 3.7 της Python και το περιβάλλον για την ανάπτυξη του κώδικα ήταν το PyCharm 2020.2.1 (Community Edition).

Για την επιλογή της γλώσσας προγραμματισμού πραγματοποιήθηκε μία αρχική έρευνα για το συγκεκριμένο θέμα, και οι γλώσσες προγραμματισμού που φάνηκαν πιο κατάλληλες για τη διεκπεραίωση του έργου ήταν η Python και η R. Η επιλογή έγινε ανάμεσα σε αυτές τις δύο και τελικά επιλέχτηκε η Python, και οι λόγοι για αυτή την επιλογή Python ήταν οι ακόλουθοι:

- εμπειρία χρήσης της συγκεκριμένης γλώσσας,
- ευκολία εκμάθησης νέων χαρακτηριστικών της γλώσσας
- διαθεσιμότητα κατάλληλων βιβλιοθηκών, ιδίως για τον υπολογισμό μετρικών συναισθημάτων από κείμενα

1 https://pandas.pydata.org/

- καλή τεκμηρίωση και υλικό από την κοινότητα (community)

Το αμέσως επόμενο βήμα ήταν να βρεθεί το περιβάλλον που θα υλοποιηθεί ο κώδικας. Στο σημείο αυτό δεν καταβλήθηκε ιδιαίτερη προσπάθεια, καθώς τα απαραίτητα χαρακτηριστικά (χρωματισμός κώδικα, υποστήριξη εσοχών, αυτόματη συμπλήρωση/προτάσεις, επισήμανση συντακτικών λαθών, χρήση βιβλιοθηκών και υποστήριξη αποσφαλμάτωσης) ήταν διαθέσιμα σε όλα τα περιβάλλοντα ανάπτυξης. Εν τέλει το περιβάλλον που επιλέχθηκε για την ανάπτυξη του κώδικα ήταν PyCharm 2020.2.1 (Community Edition) (https://www.jetbrains.com/pycharm/).

4.2. Η βιβλιοθήκη Twitter Emotion Recognition

Η βιβλιοθήκη που χρησιμοποιήθηκε στην πτυχιακή είχε σκοπό την εκπαίδευση επαναλαμβανόμενων νευρωνικών δικτύων (RNN), για την εκτίμηση συναισθημάτων αγγλικών tweets. Πιο συγκεκριμένα όλα τα μοντέλα δουλεύουν σε συμβολοσειρές και ως εκ τούτου περνάμε ολόκληρη τη συμβολοσειρά χωρίς καμία επεξεργασία ως είσοδο στο μοντέλο RNN. Τα διαθέσιμα μοντέλα της βιβλιοθήκης είναι:

- Ekman's six basic emotions
- Plutchik's eight basic emotions
- Profile of Mood States (POMS) six mood states.

Στην παρούσα πτυχιακή εργασία χρησιμοποιήθηκαν τα έξι βασικά συναισθήματα του Ekman. Τα βασικά έξι συναισθήματα με βάση τον Ekman είναι (Bosco, Patti, & Bolioli, 2013):

- φόβος – Fear
- θυμός – Anger
- Χαρά – Joy
- Θλίψη – Sadness
- Αηδία (απέχθεια) – Disgust
- Έκπληξη – Surprise

Κάποια από τα συναισθήματα έχουν κάποιες αντίστοιχες εκφράσεις προσώπου, κάτι που δεν χρειάστηκε να λάβουμε υπόψη στην πτυχιακή εργασία.

Η βιβλιοθήκη μας έδινε την δυνατότητα να χρησιμοποιήσουμε τα οκτώ συναισθήματα του Robert Plutchik. Τα 8 συναισθήματα του Plutchik βασίζονται στα πρωταρχικά συναισθήματα και το κάθε συναίσθημα έναυσμα συμπεριφοράς με υψηλή αξία επιβίωσης και η θεωρία των βασικών συναισθημάτων έχει δέκα αξιώσεις που βασίζονται σε ζώα και ανθρώπους, κάτι που δεν το

χρειαζόμαστε στην πτυχιακή εργασία. Όπως και προηγουμένως, θα αναφέρουμε τα συναισθήματα παρακάτω.

Συναίσθημα	Συμπεριφορά	Αποτέλεσμα
Φόβος, τρόμος	Τρέξιμο, ή πέταγμα μακριά	Προστασία
Θυμός, οργή	Δάγκωμα, χτύπημα	Καταστροφή
Χαρά, έκσταση	Φλερτάρισμα, ζευγάρωμα	Αναπαραγωγή
Θλίψη	Αναζήτηση για βοήθεια	Επανένταξη
Αποδοχή, εμπιστοσύνη	Περιποίηση, κοινή χρήση	Δεσμός
Αηδία	Εμετός, απομάκρυνση	Απόρριψη
Προσμονή	Εξέταση, χαρτογράφηση	Εξερεύνηση
Έκπληξη	Διακοπή, ειδοποίηση	Προσανατολισμός

Η τρίτη επιλογή που μπορούσαμε να χρησιμοποιήσουμε απο την βιβλιοθήκη ήταν Profile of Mood States που είναι μια κλίμακα διαβάθμισης που χρησιμοποιείται για την αξιολόγηση παροδικών, καταστάσεων διάθεσης, και αυτή η επιλογή δεν κρίθηκε άμεσα αξιοποιήσιμη γιατί τα reviews έχουν να κάνουν κυρίως με το προϊόν και όχι με την ψυχοσύνθεση και την κατάσταση διάθεσης του χρήστη. Για παράδειγμα ένας χρήστης μπορεί να έχει καλή διάθεση άλλα να γράψει κακή κριτική για το προϊόν που αγόρασε γιατί το προϊόν δεν κάλυπτε τις ανάγκες του. Τα συναισθήματα με βάση Profile of Mood States είναι:

- Ένταση ή άγχος
- Οργή ή εχθρότητα
- Δυνατότητα ή Δραστηριότητα
- Κούραση ή αδράνεια
- Κατάθλιψη ή απογοήτευση
- Σύγχυση ή σύγχυση

Ωστόσο, είναι πολύ πιθανό η βαθμολογία του χρήστη να επηρεάζεται και από την τρέχουσα κατάτασή του, και αυτός ο παράγοντας θα ληφθεί υπ’ όψιν σε μελλοντική εργασία.

4.3. Σύνολα δεδομένων

Στις αρχικές εκδόσεις της πτυχιακής εργασίας επιχειρήθηκε η ανάπτυξη λογισμικού το οποίο θα αντλούσε τις βαθμολογίες και τις κειμενικές αξιολογήσεις από σχετικούς ιστοχώρους. Προς αυτή

την κατεύθυνση χρησιμοποιήθηκε η βιβλιοθήκη της Python “Beautiful Soup”[2] για να υλοποιηθεί web scraping για την ανάκτηση των δεδομένων. Ο κώδικας που αναπτύχθηκε φαίνεται στη συνέχεια.

```
import csv
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.chrome.options import Options


# drivers path on this computer
PATH = "/usr/bin/chromedriver"
driver = webdriver.Chrome(r"C:\\Users\\DS-pc\\PycharmProjects\\Follow_bot\\chromedriver.exe")

# gets the url (amazon) and automatically opens it
url = 'https://www.amazon.com/'

chrome_options = Options()
chrome_options.add_experimental_option("detach", True)
driver.get(url)


def get_url(search_text):
    # generate url from text with patern from amazon, the link the base template will stay the same
    # but in {} we will have the search term and because the link uses the symbol + instead of ' '
    # we will replace the ' ' with '+' symbol
    # also this template will work for 2 terms only seperated with space
    base_template = 'https://www.amazon.com/s?k={}&ref=nb_sb_noss_1'
    search_term = search_text.replace(' ', '+')
    # add term query url
    url = base_template.format(search_term)
    # add page query placeholder
    url += '&page{}'
    return url

# getting the records from search term on the all pages
def extract_record(item):
     # in this case we are using the h2 tag because it have all the items info and its easy to
identify,
    # we assign the h2 to the atag to get different kind of info about the item
    atag = item.h2.a
    # in this case we get the text from the a tag and the text show us the item description
    description = atag.text.strip()
    # then we will need the static url of the site ( amazon.com) plus the link from the item we are
trying
    # to the the data, the link will be different for every item but the first part will be the same
cause
    # the are on the same site (amazon.com)
    url = 'https://www.amazon.com' + atag.get('href')

    # we are finding and scraping the product data from the "inspect element"
```

2 https://www.crummy.com/software/BeautifulSoup/

```
    # every time we are trying to find something unique to identify the item to get the data
    try:
        # product price
        price_parent = item.find('span', 'a-price')
        price = price_parent.find('span', 'a-offscreen').text
    except AttributeError:
        return

    try:
        # rating and review count
        rating = item.i.text
        review_count = item.find('span', {'class': 'a-size-base', 'dir': 'auto'}).text
    except AttributeError:
        rating = ''
        review_count = ''

    try:
        # full review
        # in full review the is a problem i tried to identifie the data but i cant export the
        # text in csv file for some reason,
         # ( maybe because its not on the first page and its on every item and if someone wants to
see
        # the full user reviews he must click on another hyperlink)
         full_review = item.find.all('div', {'data-hook':'review-collapsed','aria-expanded': 'true',
'class':'a-expander-content reviewText review-text-content a-expander-partial-collapse-content a-
expander-content-expanded'}).text
    except AttributeError:
        full_review = ''


    # the resault with our data
    result = (description, price, rating, review_count, url, full_review)

    return result


def main(search_term):
    """Run main program routine"""

    records = []
    url = get_url(search_term)

    # the range can be lower the number is just for testing ( 21 pages)
    for page in range(1, 21):
        driver.get(url.format(page))

         # for some strange reason now the chrome closes, i did some changes in the code and now it
close
        # after the program finishes for no reason :(
        chrome_options = Options()
        chrome_options.add_experimental_option("detach", True)
        #parse html content from the page source
        soup = BeautifulSoup(driver.page_source, 'html.parser')
```

```
        #getting number of the resaults from the first page with
        results = soup.find_all('div', {'data-component-type': 's-search-result'})
        # just printing the amount of results for testing
        print(len(results))

        # we call the function extract record for every items to get all the data we want form
        # the records of all pages
        for item in results:
            record = extract_record(item)
            if record:
                records.append(record)

    driver.close()
    # save data to csv file
    with open('results.csv', 'w', newline='', encoding='utf-8') as f:
        writer = csv.writer(f)
        writer.writerow(['Description', 'Price', 'Rating', 'ReviewCount', 'Url','Full review'])
        writer.writerows(records)

    # run the main program

# we are giving the arguments from here (didnt ask the user give the search term inside main)
main('nvidia rtx')
```

Η λογική στον κώδικα είναι να ξεκινάει η αναζήτηση από ένα σημείο αφετηρίας (seed) και στη συνέχεια για κάθε σελίδα να εξάγονται τα στοιχεία των αξιολογήσεων που περιέχονται εκεί και στη συνέχεια να αναζητούνται άλλες σελίδες στις οποίες μπορούμε να πλοηγηθούμε για να εξάγουμε και άλλες αξιολογήσεις. Η διαδικασία ωστόσο απεδείχθη αρκετά περίπλοκη και έπρεπε να αντιμετωπιστεί ένα μεγάλο πλήθος ζητημάτων που αφορούσαν τη συγκεκριμένη δόμηση και μορφοποίηση της κάθε σελίδας, τον χρόνο που χρειαζόταν για να πραγματοποιηθεί η εξαγωγή της πληροφορίας, καθώς και τους μηχανισμούς ελέγχου ρυθμού πρόσβασης που διαθέτει η Amazon, όπως και άλλες αντίστοιχες εταιρείες και ιστοχώροι. Για τους λόγους αυτούς, και επιπρόσθετα διότι η ανάπτυξη λογισμικού για ιστοσυλλογή δεν ήταν ο κύριος στόχος της πτυχιακής, η προσπάθεια ανάπτυξης σχετικού λογισμικού εγκαταλείφθηκε και στραφήκαμε προς τη χρήση έτοιμων συνόλων δεδομένων. Έτσι, χρησιμοποιήθηκαν τα σύνολα δεδομένων που διαθέτει η Amazon στον σύνδεσμο https://jmcauley.ucsd.edu/data/amazon/. Συγκεκριμένα χρησιμοποιήθηκαν τα ακόλουθα σύνολα δεδομένων:

Πίνακας 2: Σύνολα δεδομένων

Όνομα συνόλου δεδομένων	Πλήθος χρηστών	Πλήθος αντικειμένων	Πλήθος αξιολογήσεων	Πυκνότητα
Instant video	5130	1685	37126	0.429%
Automotive	2928	1835	20473	0.381%
Musical instruments	1429	900	10261	0.798%
Digital music	5541	3568	64706	0.327%
Office products	4905	2420	53258	0.449%
Video games	24000	11000	232000	0.089%

Σε όλες τις περιπτώσεις χρησιμοποιήθηκαν "5-core" σύνολα δεδομένων, όπου κάθε αντικείμενο και κάθε χρήστης εμφανίζεται σε τουλάχιστον 5 αξιολογήσεις εντός του συνόλου δεδομένων. Αυτό διασφαλίζει ότι για κάθε αντικείμενο θα υπάρχει ένα ελάχιστο πλήθος χρηστών που μπορούν να συνεισφέρουν στη διαμόρφωση εκτιμήσεων, και αυξάνει την πιθανότητα να υπάρξει "κοντινός γείτονας" για κάθε χρήστη.

Η στήλη "Πυκνότητα" που εμφανίζεται στον πίνακα ορίζεται ως

$$Πυκνότητα = \frac{πλήθος\ αξιολογήσεων}{πλήθος\ χρηστών * πλήθος\ αντικειμένων}$$

και μας δίνει μία μετρική του ποσοστού του πίνακα αξιολογήσεων (ένας πίνακας που έχει ως γραμμές τους χρήστες, ως στήλες τα αντικείμενα, και κάθε κελί έχει ως τιμή την αξιολόγηση του αντίστοιχου χρήστη για το αντίστοιχο αντικείμενο) περιέχει τιμές, έναντι του ποσοστού που είναι κενό. Όσο μεγαλύτερη είναι η πυκνότητα, τόσο πιο πιθανό είναι να μπορούμε να υπολογίσουμε μια εκτίμηση για την αξιολόγηση που θα έδινε ένας χρήστης σε ένα αντικείμενο.

4.4. Η βιβλιοθήκη Pandas στην Python

Η βιβλιοθήκη Pandas είναι μια βιβλιοθήκη ανοιχτού κώδικα και μας επιτρέπει να εκτελέσουμε τον χειρισμό και ανάλυση δεδομένων σε Python. Πιο συγκεκριμένα, μας προσφέρει χειρισμό δεδομένων και λειτουργίες δεδομένων για αριθμητικούς πίνακες και χρονοσειρές. Το σημαντικό πλεονέκτημα αυτής της βιβλιοθήκη είναι ότι μας παρέχει χωρίς ιδιαίτερο κόπο εύκολο χειρισμό δεδομένων, μέσω των ακόλουθων χαρακτηριστικών:

- Χειρίζεται εύκολα τα δεδομένα που λείπουν,
- Χρησιμοποιεί σειρές για μονοδιάστατη δομή δεδομένων και DataFrame για πολυδιάστατη δομή δεδομένων,
- Παρέχει έναν αποτελεσματικό τρόπο περικοπής των δεδομένων,

- Παρέχει έναν ευέλικτο τρόπο συγχώνευσης, συνένωσης ή αναδιαμόρφωσης των δεδομένων.

4.5. Οι ενότητες του κώδικα που αναπτύχθηκε

4.5.1. Παράμετροι των διαδικασιών υπολογισμού ομοιότητας

Όπως αναφέρθηκε στις ενότητες 3.1 και 3.2, στον υπολογισμό των μετρικών ομοιότητας δεν λαμβάνονται υπ΄ όψιν το ίδιο έντονα όλα τα συναισθήματα, καθώς εκτιμάται ότι παίζουν σε διαφορετικό βαθμό ρόλο στη διαμόρφωση της τελικής εικόνας και αξιολόγησης του χρήστη. Για παράδειγμα, εκτιμάται ότι ο φόβος δεν θα διαδραματίσει σημαντικό ρόλο, καθώς η βαθμολογία προκύπτει περισσότερο από την εμπειρία, ενώ ο φόβος εκφράζει μία αβεβαιότητα για μία δυσάρεστη μελλοντική εξέλιξη. Για τον λόγο αυτό, σε κάθε συναίσθημα ανατίθεται μία βαρύτητα και η βαρύτητα αυτή χρησιμοποιείται στον μαθηματικό τύπο υπολογισμού της μετρικής ομοιότητας μεταξύ χρηστών.

Επίσης, οι βαθμολογίες εκφράζονται στην κλίμακα [1, 5], ενώ οι εκτιμήσεις των συναισθημάτων στην κλίμακα [0, 1] με τον επιπρόσθετο περιορισμό το άθροισμα όλων των εκτιμήσεων συναισθημάτων να είναι ίσο με 1. Κατά συνέπεια, αν οι τιμές αυτές χρησιμοποιούνταν ως είχαν, η βαθμολογία θα έπαιζε δεσπόζοντα ρόλο στην διαμόρφωση της τελικής τιμής ομοιότητας, με τις τιμές των εκτιμήσεων των συναισθημάτων να διαδραματίζουν οριακό ρόλο. Προκειμένου να αντιμετωπισθεί αυτό το θέμα, εισήχθη μία διαδικασία κλιμάκωσης (scaling) των τιμών.

Οι τιμές για τους συντελεστές βαρύτητας και κλιμάκωσης για κάθε διάστασης αξιολόγησης παρουσιάζονται στον πίνακα που ακολουθεί. Στην παρούσα φάση οι συντελεστές τέθηκαν κατ' εκτίμηση, ενώ η διερεύνηση και εύρεση των βέλτιστων τιμών για κάθε συντελεστή είναι ένα ερώτημα που μπορεί να εξετασθεί στο μέλλον.

Πίνακας 3: Τιμές συντελεστών βαρύτητας και κλιμάκωσης για κάθε διάσταση αξιολόγησης

Διάσταση	**Συντελεστής βαρύτητας**	**Συντελεστής κλιμάκωσης**
Συνολική βαθμολογία	0.5	0.1
Θυμός	0.6	1
Αηδία	0.3	1
Φόβος	0.1	1
Χαρά	0.6	1
Λύπη	0.4	1
Έκπληξη	0.2	1

Στη συνέχεια παρουσιάζεται και ο κώδικας Python που θέτει τις παραμέτρους αυτές.

```
# weights help us to change the impact of a specific emotion or the overall rating for our simulation, for example some emotions
# we can assume that some emotions have bigger impact on out simulation same goes to the overall rating
weights = {'overall': 0.5, 'anger': 0.6, 'disgust': 0.3, 'fear': 0.1, 'joy': 0.6, 'sadness': 0.4, 'surprise': 0.2}

# we need the scaling because our emotion function have fields from [0,1] with sum of 1 that mean its not possible to impliment it
# on the amazon dataset (like most datasets) because the range is [1 ,5] thats why we need the scaling, on different datasets it must change
scalings = {'overall': 0.1, 'anger': 1, 'disgust': 1, 'fear': 1, 'joy': 1, 'sadness': 1, 'surprise': 1}
```

4.5.2. Άνοιγμα αρχείου δεδομένων

Το πρώτο βήμα που εκτελείται στον κώδικα είναι να ανοίξουμε το αρχείο και να διαβάσουμε τα δεδομένα. Εν τέλει δημιουργήθηκε μια συνάρτηση για το άνοιγμα του αρχείου .json, η οποία μπορεί είτε να διαβάσει είτε το αρχείο με σύνολο δεδομένων, όπως έγινε η λήψη του από τη σελίδα https://jmcauley.ucsd.edu/data/amazon/, είτε μία επεξεργασμένη μορφή του στην οποία έχει ήδη πραγματοποιηθεί συναισθηματική ανάλυση των κειμένων και έχουν εξαχθεί οι βαθμολογίες για τις επιμέρους διαστάσεις. Η δυνατότητα αυτή ήταν σημαντική στη διαδικασία αποσφαλμάτωσης, καθώς απαιτείται σημαντικός χρόνος για την εξαγωγή μετρικών για τα συναισθήματα από τις κειμενικές αξιολογήσεις.

```
# read cached JSON file
def readJSONFile(filespec, filetype):
    # if file type = 'anew', we use this code
    if (filetype == 'anew'):
        df0 = pd.read_json(filespec, lines = True)
                                                df      =      pd.DataFrame(df0,      columns      =
['reviewerName','reviewerID','reviewText','overall','asin'])
    else:
        # else, if file type = cached, use this code
        with open(filespec)as json_data:
            data = json.load(json_data)
        df = pd.DataFrame(data['data'])
        df.columns = data['columns']

    return df
```

Το πρότυπο της συνάρτησης είναι `def readJSONFile(filespec, filetype)`. Η πρώτη παράμετρος της συνάρτησης (filespec) δίνει το αρχείο το οποίο θα διαβαστεί, ενώ η δεύτερη παράμετρος (filetype) δίνει το εάν θα διαβαστεί το αρχείο όπως ελήφθη (τιμή “anew”) ή στην επεξεργασμένη μορφή (τιμή “cached”).

Στην γραμμή 71 βλέπουμε τα πεδία που χρησιμοποιηθήκαν απο το αρχείο .json, τα οποία είναι reviewerName (όνομα αξιολογητή), reviewerID (κωδικός αξιολογητή), reviewText (κείμενο αξιολόγησης), overall (αριθμητική βαθμολογία), asin (κωδικός προϊόντος).

Τα δεδομένα επιστρέφονται σε ένα data frame, για περαιτέρω επεξεργασία.

4.5.3. Εξαγωγή μετρικών για τα συναισθήματα από τις κειμενικές αξιολογήσεις

Η εξαγωγή των μετρικών για τα συναισθήματα γίνεται με βάση τους μαθηματικούς τύπους που αναφέρθηκαν στις ενότητες 3.1 και 3.2. Ο τρόπος υπολογισμού της κάθε μετρικής αναλύεται στις επόμενες παραγράφους,

4.5.4. Υπολογισμός της Cosine Similarity

Για τον υπολογισμό του cosine similarity εφαρμόζεται ο τύπος της εκτεταμένης μορφής της μετρικής Cosine Similarity, που αναφέρθηκε στην ενότητα 3.1.

$$CS_{sent}(X,Y)=\frac{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * r_{X,i}(d) * r_{Y,i}(d)}{\sqrt{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * r_{X,i}^2}} * \sqrt{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * r_{Y,i}^2}$$

Όπως παρατηρούμε, ο εκτεταμένος τύπος διατρέχει όλες τις διαστάσεις, συνυπολογίζοντάς τες στο αποτέλεσμα, εφαρμόζοντας τόσο την κλιμάκωση όσο και τη βάρυνση.

Ο κώδικας για την υλοποίηση παρατίθεται στη συνέχεια. Ο κώδικας διατρέχει τα ζεύγη χρηστών ($user_i$, $user_j$) και για κάθε ζεύγος υπολογίζει την cosine similarity. Το τελικό αποτέλεσμα αποθηκεύεται σε έναν δισδιάστατο πίνακα CS_similarities. Για την επιτάχυνση της διαδικασίας, πραγματοποιήθηκαν οι εξής βελτιστοποιήσεις:

1. για κάθε χρήστη X, ισχύει ότι *cosineSimilarity(X, X) = 1*, συνεπώς σε αυτές τις περιπτώσεις δεν πραγματοποιούνται υπολογισμοί αλλά χρησιμοποιείται απ' ευθείας η τιμή 1.
2. για κάθε ζεύγος χρηστών X, Y, ισχύει ότι *cosineSimilarity(X, Y) = cosineSimilarity(Y, X)* (αντιμεταθετική ιδιότητα), συνεπώς για κάθε ζεύγος χρηστών γίνεται ένας μόνο υπολογισμός.
3. επειδή ο υπολογισμός των ομοιοτήτων για όλα τα ζεύγη χρηστών ήταν πολύ χρονοβόρος (της τάξης των πολλών ημερών), έγινε η σύμβαση να υπολογίζονται μόνο οι ομοιότητες των 500 πρώτων χρηστών του κάθε συνόλου δεδομένων με όλους τους υπόλοιπους, και στη συνέχεια στη διαδικασία του υπολογισμού εκτιμήσεων να πραγματοποιούνται διαμορφώσεις εκτίμησης μόνο για αξιολογήσεις των χρηστών αυτών. Αυτό πιστεύεται ότι

δεν βλάπτει τη γενικότητα και ακρίβεια της διαδικασίας αξιολόγησης, καθώς δεν εισάγεται οποιαδήποτε στρέβλωση (skew) στα δεδομένα.

Ο υπολογισμός της ομοιότητας δύο χρηστών X και Y γίνεται μέσω της συνάρτησης *simCFweights*, η οποία δέχεται ως παραμέτρους τις αξιολογήσεις των δύο χρηστών καθώς και τους πίνακες κλιμάκωσης και βάρυνσης, και επιστρέφει την τιμή της μετρικής ομοιότητας μεταξύ των χρηστών, όπως αυτή εκφράζεται από τον αντίστοιχο μαθηματικό τύπο.

```
for i in range(numSimilaritiestocompute):
  for j in range(numUsers):
    if (i == j):
      CS_similarities[i][j] = 1
    elif (i < j):
# the list have the similarity between users[i][j]
# now here we are calling the function cosine similarity to find the nearest_neighbours
      nn_cosine_similarities = simCFweights(reviews[i], reviews[j], weights, scalings)
      CS_similarities[i][j] = nn_cosine_similarities
      CS_similarities[j][i] = nn_cosine_similarities
```

```
# in this functon we calculate the cosine similarity of all pair of the dataframe
def simCFweights(reviews1, reviews2, weights, scalings):
  numCommon = 0
  nominator = 0
  denom1 = 0
  denom2 = 0
  for index, row in reviews1.iterrows():
# check if the same item has been rated by user2
    otherUserReview = reviews2[reviews2.asin.eq(row.asin)]
    if (len(otherUserReview) > 0):
# same product rated
      numCommon = numCommon + 1
      for dimension in weights:
        nominator = nominator + row[dimension] * otherUserReview.iloc[0][dimension] *
weights[dimension] * scalings[dimension]
        denom1 = denom1 + weights[dimension] * row[dimension] * row[dimension] * scalings[dimension]
        denom2 = denom2 + weights[dimension] * otherUserReview.iloc[0][dimension] *
otherUserReview.iloc[0][dimension] * scalings[dimension]
  try:
      return nominator / math.sqrt(denom1) / math.sqrt(denom2)
  except ZeroDivisionError:
      return 0
```

4.5.5. Υπολογισμός της Pearson Correlation Coefficient

Περιγραφή της εξαγωγής των μέσων όρων και κατόπιν της simPFweights και του κώδικα που την καλεί.

Και για τον υπολογισμό του Pearson Correlation Coefficient εφαρμόζεται ο τύπος της εκτεταμένης μορφής της μετρικής Pearson Correlation Coefficient, που αναφέρθηκε στην ενότητα 3.2

$$PCC_{sent}(X,Y)=\frac{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * (r_{X,i}(d)-\overline{r_X(d)}) * (r_{Y,i}(d)-\overline{r_Y(d)})}{\sqrt{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * (r_{X,i}(d)-\overline{r_X(d)})^2} * \sqrt{\sum_{i\in R(X)\cap R(Y)}\sum_{d\in D} w_d * sc_d * (r_{Y,i}(d)-\overline{r_Y(d)})^2}}$$

Όπως παρατηρούμε, και εδώ ο εκτεταμένος τύπος διατρέχει όλες τις διαστάσεις, συνυπολογίζοντάς τες στο αποτέλεσμα, εφαρμόζοντας τόσο την κλιμάκωση όσο και τη βάρυνση. Η διαφορά στην εκτεταμένη του μορφή της Pearson Correlation Coefficient από τον τύπο της Cosine Similarity είναι ότι προσαρμόζει την κάθε βαθμολογία αφαιρώντας τον μέσο όρο του συγκεκριμένου χρήστη. Αυτό συμβαίνει σε όλες τις εξεταζόμενες διαστάσεις.

Ο κώδικας για την υλοποίηση παρατίθεται στη συνέχεια. Ο κώδικας διατρέχει τα ζεύγη χρηστών ($user_i$, $user_j$) σε αντίθεση με την cosine similarity στην Pearson Correlation Coefficient έπρεπε να λάβουμε υπόψη μας και και τον μέσο όρο του κάθε χρήστη ($user_{i,m}$, $user_{j,m}$) για κάθε ζεύγος υπολογίζει την Pearson Correlation Coefficient. Το τελικό αποτέλεσμα αποθηκεύεται σε έναν αντίστοιχο δισδιάστατο πίνακα PEARSON_similarities. Για την επιτάχυνση της διαδικασίας, πραγματοποιήθηκαν οι εξής βελτιστοποιήσεις όπως και στο κομμάτι στις Cosine Similarity:

1. για κάθε χρήστη X, ισχύει ότι PEARSON_s*imilarities(X, X) = 1*, συνεπώς σε αυτές τις περιπτώσεις δεν πραγματοποιούνται υπολογισμοί αλλά χρησιμοποιείται απ' ευθείας η τιμή 1.
2. για κάθε ζεύγος χρηστών X, Y, ισχύει ότι PEARSON_s*imilarities(X,Y)* = PEARSON_s*imilarities(Y, X)* (αντιμεταθετική ιδιότητα), συνεπώς για κάθε ζεύγος χρηστών γίνεται ένας μόνο υπολογισμός.
3. επειδή ο υπολογισμός των ομοιοτήτων για όλα τα ζεύγη χρηστών ήταν πολύ χρονοβόρος (της τάξης των πολλών ημερών), έγινε η σύμβαση να υπολογίζονται μόνο οι ομοιότητες των 500 πρώτων χρηστών του κάθε συνόλου δεδομένων με όλους τους υπόλοιπους, και στη συνέχεια στη διαδικασία του υπολογισμού εκτιμήσεων να διαμορφώνονται εκτιμήσεις τιμές μόνο για αξιολογήσεις των χρηστών αυτών. Αυτό πιστεύεται ότι δεν βλάπτει τη γενικότητα και ακρίβεια της διαδικασίας αξιολόγησης, καθώς δεν εισάγεται οποιαδήποτε στρέβλωση (skew) στα δεδομένα.

Ο υπολογισμός της ομοιότητας δύο χρηστών X και Y γίνεται μέσω της συνάρτησης *simPFweights*, η οποία δέχεται ως παραμέτρους τις αξιολογήσεις των δύο χρηστών καθώς και τους πίνακες κλιμάκωσης και βάρυνσης και τους μέσους όρους των αξιολογήσεων του κάθε χρήστη, και

επιστρέφει την τιμή της μετρικής ομοιότητας μεταξύ των χρηστών, όπως αυτή εκφράζεται από τον αντίστοιχο μαθηματικό τύπο.

```
for i in range(numSimilaritiestocompute):
  for j in range(numUsers):
      if (i == j):
          PEARSON_similarities[i][j] = 1
      elif (i < j):
          # the list have the similarity between users[i][j]
           # new we are importing pearson similarity between users[i][j] to our dataframe to the new column "user[i][j]_similarities"
                nn_pearson_similarities = simPFweights(reviews[i], reviews[j], PEARSON_means[i], PEARSON_means[j], weights, scalings)
              #print(simpfweights(reviews_pearson[i], reviews_pearson_mean[k], reviews_pearson[j], reviews_pearson_mean[z], weights, scalings))
    #here we are saying the result of the function to df_sim_pearson
          PEARSON_similarities[i][j] = nn_pearson_similarities
          PEARSON_similarities[j][i] = nn_pearson_similarities



#in this functon we calculate the pearson similarity of all pair of the dataframe
# in reviews1_mean,reviews2_mean we have the mean values of the rating
def simPFweights(reviews1, reviews2, reviews1_mean, reviews2_mean, weights, scalings):
    numCommon = 0
    nominator = 0
    denom1 = 0
    denom2 = 0
    for index, row in reviews1.iterrows():
        # check if the same item has been rated by user2
        otherUserReview = reviews2[reviews2.asin.eq(row.asin)]
        # check if the same item mean has been rated by user2
        if (len(otherUserReview) > 0):
          # same product rated
          numCommon = numCommon + 1
          #implimitation of the function according to sim-metrics
          for dimension in weights:
                      nominator = nominator + (((row[dimension] - reviews1_mean[dimension]) * (otherUserReview.iloc[0][dimension] - reviews2_mean[dimension]))) * weights[dimension] * scalings[dimension]
                      denom1 = denom1 + ((row[dimension] - reviews1_mean[dimension]) ** 2) * weights[dimension] * scalings[dimension]
              denom2 = denom2 + ((otherUserReview.iloc[0][dimension] - reviews2_mean[dimension]) ** 2) * weights[dimension] * scalings[dimension]
    try:
      return nominator / (math.sqrt(denom1 * denom2))
    except ZeroDivisionError:
      return 0
```

4.6. Διαμόρφωση εκτιμήσεων

Μετά τον υπολογισμό των μετρικών ομοιότητας Cosine Similarity και Pearson Correlation Coefficient, το επόμενο βήμα είναι να υπολογισθούν οι εκτιμήσεις για την βαθμολογία που θα έδιναν οι χρήστες στα αντικείμενα. Ο υπολογισμός της βαθμολογίας γίνεται σύμφωνα με τον μαθηματικό τύπο

$$p_{U,i} = \bar{r}_U + \frac{\sum_{V \in NN(U) \wedge r_{V,i} \neq NULL} s_{U,V}(r_{V,i} - \bar{r}_V)}{\sum_{V \in NN(U) \wedge r_{V,i} \neq NULL} s_{U,V}}$$

όπου $\bar{r}_U$ ($\bar{r}_V$) είναι ο μέσος όρος των αξιολογήσεων που έχει δώσει ο χρήστης U (V) και $s_{U,V}$ η εκάστοτε ομοιότητα μεταξύ των χρηστών U και Ω. Οι μέσες τιμές των αξιολογήσεων ανά χρήστη είναι αποθηκευμένες στον πίνακα *PEARSON_means*. Όπως παρατηρούμε, ο τύπος αυτός υπολογίζει πόσο αποκλίνει η αξιολόγηση του υπό θεώρηση αντικειμένου από τον μέσο όρο του υπό θεώρηση χρήστη ($r_{V,i} - \bar{r}_V$), ενώ η απόκλιση αυτή του κάθε κοντινού γείτονα σταθμίζεται με τη μετρική ομοιότητάς του ως προς τον χρήστη για τον οποίο διαμορφώνεται η εκτίμηση. Τέλος, η "σταθμισμένη συνολική απόκλιση" προστίθεται στον μέσο όρο των αξιολογήσεων του χρήστη για τον οποίο διαμορφώνεται η εκτίμηση.

Σε πολλές επιστημονικές εργασίες αλλά και υλοποιήσεις, από το δυνητικό σύνολο των κοντινών γειτόνων λαμβάνονται υπ' όψιν μόνο εκείνοι που ο βαθμός ομοιότητάς τους με τον χρήστη για τον οποίο διαμορφώνεται η εκτίμηση ξεπερνά ένα κατώφλι. Στην παρούσα πτυχιακή εργασία εξετάζουμε την επίδραση της τιμής του κατωφλίου στην κάλυψη και την ποιότητα των εκτιμήσεων, θεωρώντας τις ακόλουθες περιοχές τιμών για το κατώφλι ομοιότητας:

- Για την Cosine Similarity, το κατώφλι κυμαίνεται από 0.2 έως 0.8.
- Για την Pearson Correlation Coefficient, το κατώφλι κυμαίνεται από 0.0 μέχρι και 0.7.

Η διαφορά στο εύρος των θεωρούμενων τιμών οφείλεται στη διαφορετική φύση των δύο μετρικών, καθώς η Cosine Similarity δίνει τιμή κοντά στο 0 για πλήρως αντίθετες αξιολογήσεις και 1 για πλήρως ταυτιζόμενες όταν οι βαθμολογίες είναι θετικές (συνθήκη που ισχύει στα σύνολα δεδομένων βαθμολογιών), ενώ η Pearson Correlation Coefficient δίνει τιμή -1 για πλήρως αντίθετες αξιολογήσεις και 1 για πλήρως ταυτιζόμενες.

4.7. Υπολογισμός στατιστικών μεγεθών

Οι επιδόσεις της τεχνικής που περιγράψαμε και υλοποιήσαμε αξιολογήθηκαν ως προς δύο παραμέτρους:

1. *την ακρίβεια των* εκτιμήσεων, δηλαδή πόσο κοντά βρίσκεται η εκτιμώμενη τιμή αξιολόγησης στην πραγματική και
2. *την κάλυψη*, δηλαδή το ποσοστό των περιπτώσεων για τις οποίες μπορεί να διαμορφωθεί εκτίμηση.

Σε σχέση με την ακρίβεια των εκτιμήσεων, αυτή ποσοτικοποιείται με τον ακόλουθο τρόπο: επιλέγονται τυχαίες αξιολογήσεις χρηστών και αυτές αποκρύπτονται (αφαιρούνται από το υπό επεξεργασία σύνολο δεδομένων), και κατόπιν για κάθε μία από αυτές χρησιμοποιείται ο ανωτέρω αλγόριθμος διαμόρφωσης εκτιμήσεων για τη διαμόρφωση της τιμής της εκτίμησης και θεωρείται η διαφορά ανάμεσα στην εκτιμηθείσα τιμή και την "πραγματική"/κρυμμένη τιμή. Τέλος, όταν έχουμε το σύνολο των διαφορών ανάμεσα στις εκτιμηθείσες και τις "πραγματικές"/κρυμμένες τιμές υπολογίζουμε τη μετρική του μέσου απόλυτου σφάλματος (Mean Absolute Error – MAE). Το μέσο απόλυτο σφάλμα είναι μια μετρική που υπολογίζει αρχικά την απόλυτη τιμή του κάθε μεμονωμένου σφάλματος (διαφορά ανάμεσα στην προβλεφθείσα τιμή και την "πραγματική"/κρυμμένη τιμή αξιολόγησης) και ακολούθως εξάγει τον μέσο όρο τους. Ο μαθηματικός τύπος για τη μετρική του μέσου απόλυτου σφάλματος είναι

$$Mean\ Absolute\ Error = \frac{1}{|Predictions|} \sum_{p_{U,i} \in Predictions} |(p_{U,i} - rating_{U,i})|$$

όπου *Predictions* είναι το σύνολο των εκτιμήσεων που υπολογίσθηκαν, $p_{U,i}$ είναι μία εκτίμηση που ανήκει στο σύνολο *Predictions* και αφορά τη βαθμολογία που θα έδινε ο χρήστης *U* στο αντικείμενο *ι*, $rating_{U,i}$ είναι η *πραγματική βαθμολογία* που έχει δώσει ο χρήστης *U* στο αντικείμενο *ι* και |*Predictions*| είναι το πλήθος των εκτιμήσεων που διαμορφώθηκαν.

Προφανώς είναι επιθυμητό η μετρική MAE να βρίσκεται κοντά στο 0, καθώς αυτό καταδεικνύει ότι οι εκτιμήσεις για τις αξιολογήσεις ταυτίζονται με τις πραγματικές τιμές των αξιολογήσεων, και σε αυτή την περίπτωση ο αλγόριθμος είναι ακριβής.

Σε σχέση με την κάλυψη, ελέγχουμε ποιο είναι το ποσοστό των περιπτώσεων για τις οποίες κατέστη εφικτό να διαμορφωθεί εκτίμηση στο σύνολο των προσπαθειών για διαμόρφωση εκτίμησης αξιολόγησης. Μία προσπάθεια διαμόρφωσης εκτίμησης της αξιολόγησης που θα έδινε ο χρήστης *U* στο αντικείμενο *i* μπορεί να μην καρποφορήσει εάν δεν υπάρχουν κοντινοί γείτονες του *U* οι οποίοι να έχουν βαθμολογήσει το αντικείμενο *i*. Η πιθανότητα να έχουμε αποτυχία αυξάνεται σε συνάρτηση με τους εξής παράγοντες:

1. *πόσο "αραιό" είναι το σύνολο δεδομένων*, δηλ. πόσα στοιχεία του πίνακα αξιολογήσεων είναι κενά. Η τυπική μετρική για το πόσο "αραιό" είναι ένα σύνολο δεδομένων δίνεται από

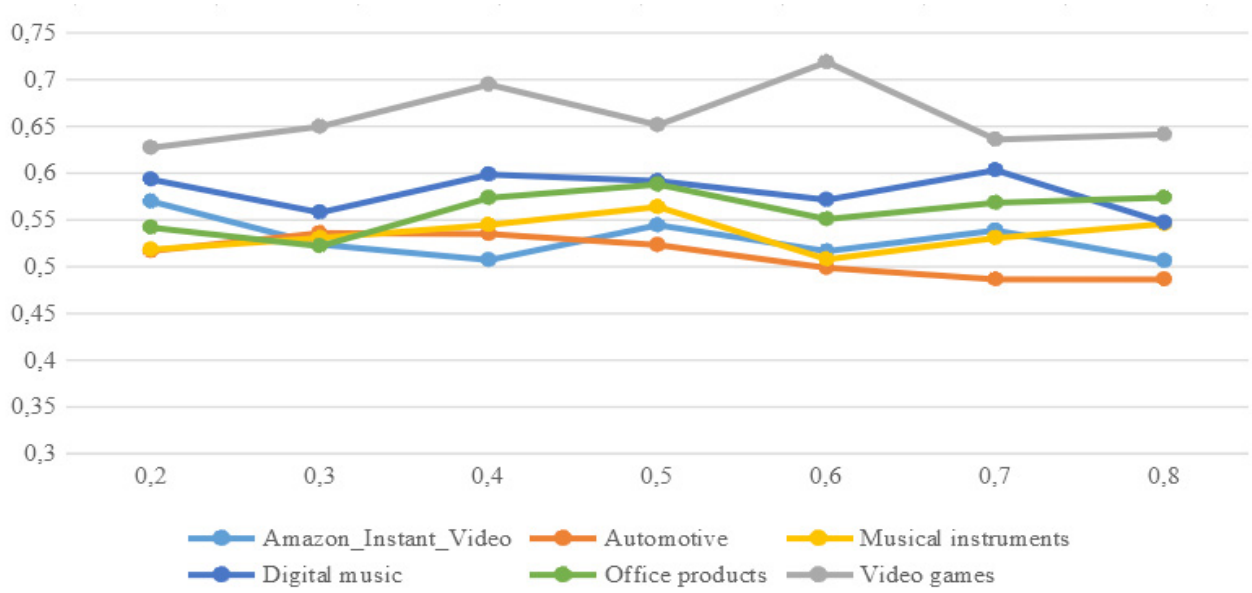
τον τύπο $\frac{\#ratings}{\#users * \#items}$, όπου *#ratings* είναι το πλήθος των αξιολογήσεων, *#users* το πλήθος των χρηστών και *#items* το πλήθος των αντικειμένων.

2. *Το κατώφλι ομοιότητας για τους κοντινούς γείτονες.* Όσο πιο υψηλό είναι το κατώφλι, τόσο αυξάνεται η πιθανότητα να μην ληφθεί υπ' όψιν ένας κοντινός γείτονας *V* ενός χρήστη *U*, και συνακόλουθα να μην συνυπολογισθούν οι αξιολογήσεις του *V* όταν διαμορφώνονται εκτιμήσεις για τις αξιολογήσεις του *U*. Έτσι, είναι πιο πιθανό να έχουμε περιστάσεις όπου κανένας από τους θεωρούμενους κοντινούς γείτονες του *U* δεν έχει αξιολογήσει ένα αντικείμενο *i*, και κατά συνέπεια να είναι αδύνατο να υπολογισθεί η εκτίμηση $p_{U,i}$.

5. Πειραματική αξιολόγηση

Στο παρόν κεφάλαιο παρουσιάζουμε τα αποτελέσματα της πειραματικής αξιολόγησης που πραγματοποιήθηκε για τον αλγόριθμο, χρησιμοποιώντας τις δύο μετρικές ομοιότητας που αναφέρθηκαν στο κεφάλαιο 3, καθώς και τα σύνολα δεδομένων που αναφέρθηκαν στην ενότητα 4.3.

Στο Σχήμα 1 παρουσιάζονται τα πειραματικά αποτελέσματα για το σφάλμα εκτίμησης των αξιολογήσεων των χρηστών (MAE), όταν για τη μέτρηση της εγγύτητας των χρηστών χρησιμοποιείται η μετρική Cosine Similarity. Ο Πίνακας 4 παρουσιάζει τα ίδια δεδομένα σε αριθμητικές τιμές. Ο Πίνακας 5 παρουσιάζει το ποσοστό κάλυψης της διαδικασίας παραγωγής εκτίμησης προβλέψεων με χρήση της μετρικής Cosine Similarity.



Σχήμα 1, Πειραματικά αποτελέσματα σφάλματος εκτίμησης (MAE) με χρήση Cosine Similarity

Πίνακας 4, Πειραματικά αποτελέσματα σφάλματος εκτίμησης (MAE) με χρήση Cosine Similarity

Κατώφλι ομοιότητας χρηστών	Instant Video	Automotive	Musical instruments	Digital music	Office products	Video games
0,2	0,570	0,517	0,518	0,593	0,542	0,627
0,3	0,524	0,536	0,531	0,558	0,522	0,650
0,4	0,507	0,535	0,545	0,598	0,574	0,695
0,5	0,544	0,523	0,564	0,592	0,588	0,651
0,6	0,517	0,499	0,508	0,571	0,551	0,719
0,7	0,539	0,486	0,531	0,603	0,568	0,636
0,8	0,506	0,486	0,546	0,547	0,574	0,641

Πίνακας 5, Ποσοστό κάλυψης της διαδικασίας παραγωγής εκτίμησης προβλέψεων με χρήση Cosine Similarity

Κατώφλι ομοιότητας χρηστών	Instant Video	Automotive	Musical instruments	Digital music	Office products	Video games
0,2	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
0,3	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
0,4	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
0,5	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
0,6	100,0%	100,0%	100,0%	100,0%	100,0%	99,8%
0,7	99,6%	100,0%	100,0%	100,0%	100,0%	99,8%
0,8	99,4%	99,2%	99,4%	100,0%	99,2%	99,0%

Στη συνέχεια θα εξετάσουμε και θα σχολιάσουμε πιο αναλυτικά τα αποτελέσματα για κάθε σύνολο δεδομένων, όταν χρησιμοποιείται η μετρική Cosine similarity.

Amazon Instant Video. Σε αυτό το σύνολο δεδομένων οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,8 (τιμή MAE: 0,506) και 0,4 (τιμή MAE: 0,507). Οι δύο τιμές κατωφλίου είναι πρακτικά ισοδύναμες όσον αφορά την ακρίβεια, ωστόσο για την τιμή κατωφλίου 0,4 έχουμε κάλυψη 100%, δηλαδή μπορούμε να διαμορφώσουμε προβλέψεις για το σύνολο των αντικειμένων για κάθε χρήστη, ενώ για την τιμή κατωφλίου 0,8 η κάλυψη μειώνεται σε 99,4%, συνεπώς η τιμή κατωφλίου 0,4 είναι προτιμητέα. Στην εργασία (Wu et al., 2017) αναφέρεται ότι για το συγκεκριμένο σύνολο δεδομένων οι αλγόριθμοι που εξετάζονται

επιτυγχάνουν MAE από 0,899 (matrix factorization) έως 0,7215 (ο αλγόριθμος που προτείνει η εργασία), συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

Automotive: Σε αυτό το σύνολο δεδομένων οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,7 (τιμή MAE: 0,486) και 0,8 (τιμή MAE: 0,486). Οι δύο τιμές κατωφλίου είναι πρακτικά ισοδύναμες όσον αφορά την ακρίβεια, ωστόσο για την τιμή κατωφλίου 0,7 έχουμε κάλυψη 100%, δηλαδή μπορούμε να διαμορφώσουμε προβλέψεις για το σύνολο των αντικειμένων για κάθε χρήστη, ενώ για την τιμή κατωφλίου 0,8 η κάλυψη μειώνεται σε 99,2%, συνεπώς η τιμή κατωφλίου 0,7 είναι προτιμητέα. Στην εργασία (Wu et al., 2017) αναφέρεται ότι για το συγκεκριμένο σύνολο δεδομένων οι αλγόριθμοι που εξετάζονται επιτυγχάνουν MAE από 0,897 (matrix factorization) έως 0,627 (ο αλγόριθμος που προτείνει η εργασία), συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

Musical instruments: Σε αυτό το σύνολο δεδομένων οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,6 (τιμή MAE: 0,508) και 0,2 (τιμή MAE: 0,518). Και για τις δύο τιμές κατωφλίου έχουμε κάλυψη 100%, συνεπώς προτιμητέα είναι η τιμή κατωφλίου 0,6, δεδομένου ότι δίνει τις ακριβέστερες προβλέψεις. Στην εργασία (Khan et al., 2021) αναφέρεται ότι για το συγκεκριμένο σύνολο δεδομένων οι αλγόριθμοι που εξετάζονται επιτυγχάνουν MAE από 1,005 (matrix factorization) έως 0,880 (ο αλγόριθμος DAML), συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

Digital music: Σε αυτό το σύνολο δεδομένων οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,3 (τιμή MAE: 0,547) και 0,2 (τιμή MAE: 0,558). Και για τις δύο τιμές κατωφλίου έχουμε κάλυψη 100%, συνεπώς προτιμητέα είναι η τιμή κατωφλίου 0,8, δεδομένου ότι δίνει τις ακριβέστερες προβλέψεις. Στην εργασία (Margaris et al., 2020) αναφέρεται ότι για το συγκεκριμένο σύνολο δεδομένων οι αλγόριθμοι που εξετάζονται επιτυγχάνουν MAE περίπου ίσο με 0,7 όταν χρησιμοποιείται το cosine similarity ως μετρική ομοιότητας, συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

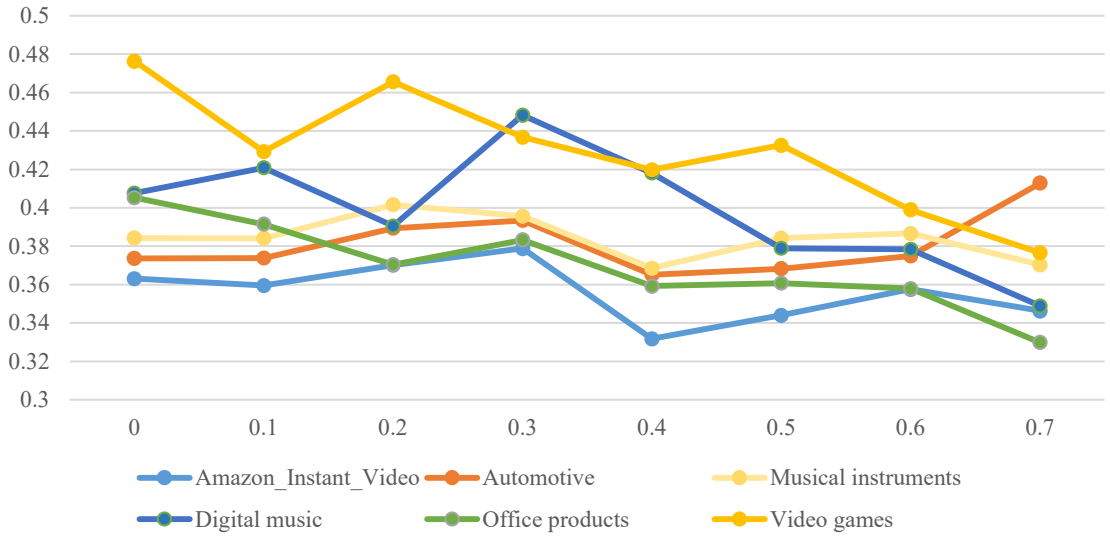
Office products: Σε αυτό το σύνολο δεδομένων οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,3 (τιμή MAE: 0,522) και 0,2 (τιμή MAE: 0,542). Και για τις δύο τιμές κατωφλίου έχουμε κάλυψη 100%, συνεπώς προτιμητέα είναι η τιμή κατωφλίου 0,3, δεδομένου ότι δίνει τις ακριβέστερες προβλέψεις. Στην εργασία (Margaris et al., 2020) αναφέρεται ότι για το συγκεκριμένο σύνολο δεδομένων οι αλγόριθμοι που εξετάζονται επιτυγχάνουν MAE

περίπου ίσο με 0,62 όταν χρησιμοποιείται το cosine similarity ως μετρική ομοιότητας, συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

Video games: Σε αυτό το σύνολο δεδομένων οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,2 (τιμή MAE: 0,627) και 0,7 (τιμή MAE: 0,636). Για τις δύο τιμές κατωφλίου έχουμε κάλυψη 100% και 99%,8 αντίστοιχα, συνεπώς προτιμιτέα είναι η τιμή κατωφλίου 0,2, δεδομένου ότι δίνει τις ακριβέστερες προβλέψεις και πλήρη κάλυψη. Στην εργασία (Margaris et al., 2020) αναφέρεται ότι για το συγκεκριμένο σύνολο δεδομένων οι αλγόριθμοι που εξετάζονται επιτυγχάνουν MAE περίπου ίσο με 0,8 όταν χρησιμοποιείται το cosine similarity ως μετρική ομοιότητας, συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

Στην μετρική Cosine similarity παρατηρούμε ότι σε όλα τα σύνολα δεδομένων υπάρχει μία διακύμανση σε σχέση με την τιμή κατωφλίου στην οποία παρατηρούνται τα ελάχιστα MAE. Επίσης παρατηρούμε ότι η αύξηση του κατωφλίου δεν επιφέρει απαραίτητα αύξηση της ακρίβειας: αυτό οφείλεται στο γεγονός ότι από τη μία πλευρά οι γείτονες που θεωρούνται σε κάθε διαμόρφωση πρόβλεψης είναι πιο "κοντινοί" στον χρήστη για τον οποίο διαμορφώνεται η πρόβλεψη (κάτι που συμβάλλει θετικά στην ακρίβεια των προβλέψεων), από την άλλη πλευρά ωστόσο είναι *λιγότεροι* (κάτι που συμβάλλει με αρνητικό τρόπο στην ακρίβεια των προβλέψεων), και η ακριβής επίδραση εξαρτάται από τα συγκεκριμένα περιεχόμενα του συνόλου δεδομένων. Σε όλα τα σύνολα δεδομένων υπάρχει μία διακύμανση και οι τιμές στο μεγαλύτερο μέρος τους βρίσκονται στην περιοχή [0,5, 0,6], με την εξαίρεση του συνόλου δεδομένων "Video games", όπου οι περισσότερες τιμές βρίσκονται στην περιοχή [0,6, 0,7], κάτι που αποδίδεται στο γεγονός ότι το συγκεκριμένο σύνολο δεδομένων είναι σημαντικά πιο αραιό σε σχέση με τα υπόλοιπα σύνολα δεδομένων.

Στο Σχήμα 2 παρουσιάζονται τα πειραματικά αποτελέσματα για το σφάλμα εκτίμησης των αξιολογήσεων των χρηστών (MAE), όταν για τη μέτρηση της εγγύτητας των χρηστών χρησιμοποιείται η μετρική Pearson Correlation Coefficient. Ο Πίνακας 6 παρουσιάζει τα ίδια δεδομένα σε αριθμητικές τιμές, Ο Πίνακας 7 παρουσιάζει το ποσοστό κάλυψης της διαδικασίας παραγωγής εκτίμησης προβλέψεων με χρήση της μετρικής Pearson Correlation Coefficient.



Σχήμα 2. Πειραματικά αποτελέσματα σφάλματος εκτίμησης (MAE) με χρήση Pearson Correlation Coefficient

Πίνακας 6, Πειραματικά αποτελέσματα σφάλματος εκτίμησης (MAE) με χρήση Pearson Correlation Coefficient

Κατώφλι ομοιότητας χρηστών	**Instant Video**	**Automotive**	**Musical instruments**	**Digital music**	**Office products**	**Video games**
0,2	0,3631	0,3737	0,3843	0,4077	0,4053	0,4763
0,3	0,3595	0,3738	0,3841	0,4208	0,3915	0,4293
0,4	0,3701	0,3893	0,4016	0,3905	0,3704	0,4656
0,5	0,3788	0,3935	0,3957	0,4482	0,3833	0,4368
0,6	0,3318	0,3651	0,3685	0,4181	0,3592	0,4198
0,7	0,3440	0,3683	0,3842	0,3789	0,3608	0,4325
0,8	0,3576	0,3750	0,3866	0,3784	0,3579	0,3990

Πίνακας 7, Ποσοστό κάλυψης της διαδικασίας παραγωγής εκτίμησης προβλέψεων με χρήση Pearson Correlation Coefficient

Κατώφλι ομοιότητας χρηστών	**Instant Video**	**Automotive**	**Musical instruments**	**Digital music**	**Office products**	**Video games**
0,2	99,6%	98,6%	99,0%	99,4%	98,8%	99,0%
0,3	99,0%	98,2%	99,0%	99,6%	98,8%	98,6%
0,4	99,0%	96,0%	97,0%	97,8%	98,2%	98,4%
0,5	97,2%	96,2%	94,6%	97,2%	96,6%	97,2%
0,6	95,4%	89,4%	94,0%	94,6%	95,6%	94,8%
0,7	92,6%	86,8%	89,0%	93,4%	94,6%	90,2%
0,8	90,2%	81,2%	83,8%	86,4%	89,2%	88,8%

Στη συνέχεια θα εξετάσουμε και θα σχολιάσουμε πιο αναλυτικά τα αποτελέσματα για κάθε σύνολο δεδομένων, όταν χρησιμοποιείται η μετρική Pearson Correlation Coefficient.

Amazon Instant Video: Σε αυτό το σύνολο δεδομένων οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,4 (τιμή MAE: 0,332) και 0,5 (τιμή MAE: 0,344) με αντίστοιχα ποσοστά κάλυψης 95,4% και 92,6%. Μεταξύ των δύο τιμών, η τιμή 0,4 είναι προτιμητέα, καθώς δίνει μεγαλύτερη ακρίβεια και υψηλότερο ποσοστό κάλυψης, ωστόσο θα μπορούσε να χρησιμοποιηθεί και η τιμή 0,1, η οποία δίνει ελαφρά χειρότερο MAE (0,356) αυξάνοντας την κάλυψη σε 99%. Στην εργασία (Wu et al., 2017) αναφέρεται ότι για το συγκεκριμένο σύνολο δεδομένων οι αλγόριθμοι που εξετάζονται επιτυγχάνουν MAE από 0,899 (matrix factorization) έως 0,7215 (ο αλγόριθμος που προτείνει η εργασία), συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

Automotive: Σε αυτό το σύνολο δεδομένων οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,4 (τιμή MAE: 0,365) και 0,5 (τιμή MAE: 0,368), με αντίστοιχα ποσοστά κάλυψης 89,4% και 86,8%. Μεταξύ των δύο τιμών, η τιμή 0,4 είναι προτιμητέα, καθώς δίνει μεγαλύτερη ακρίβεια και υψηλότερο ποσοστό κάλυψης, ωστόσο θα μπορούσε να χρησιμοποιηθεί και η τιμή 0, η οποία δίνει ελαφρά χειρότερο MAE (0,374) αυξάνοντας την κάλυψη σε 98,6%. Στην εργασία (Wu et al., 2017) αναφέρεται ότι για το συγκεκριμένο σύνολο δεδομένων οι αλγόριθμοι που εξετάζονται επιτυγχάνουν MAE από 0,897 (matrix factorization) έως 0,627 (ο αλγόριθμος που προτείνει η εργασία), συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη

διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

Musical instruments: Σε αυτό το σύνολο δεδομένων οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,4 (τιμή MAE: 0,369) και 0,7 (τιμή MAE: 0,370), όντας πρακτικά ισοδύναμες. Μεταξύ των δύο, προτιμητέα είναι η τιμή 0,4 καθώς δίνει κάλυψη 94% έναντι κάλυψης 74% που δίνει η τιμή 0,7. Παραδεκτή επιλογή θα ήταν επίσης και η τιμή κατωφλίου 0, η οποία δίνει ελαφρά χειρότερη ακρίβεια (MAE: 0,385), αυξάνοντας την κάλυψη σε 99%. Στην εργασία (Khan et al., 2021) αναφέρεται ότι για το συγκεκριμένο σύνολο δεδομένων οι αλγόριθμοι που εξετάζονται επιτυγχάνουν MAE από 1,005 (matrix factorization) έως 0,880 (ο αλγόριθμος DAML), συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

Digital music: Σε αυτό το σύνολο δεδομένων οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,7 (τιμή MAE: 0,349) και 0,6 (τιμή MAE: 0,378), με αντίστοιχα ποσοστά κάλυψης 79% και 86,4%. Καθώς οι τιμές κάλυψης είναι αρκετά χαμηλές, θα μπορούσε να χρησιμοποιηθεί η τιμή κατωφλίου 0, η οποία δίνει MAE 0,407 και κάλυψη 99,4%, ή και ως ενδιάμεση επιλογή η τιμή κατωφλίου 0,2, η οποία δίνει MAE 0,391 και κάλυψη 97,8%. Στην εργασία (Margaris et al., 2020) αναφέρεται ότι για το συγκεκριμένο σύνολο δεδομένων οι αλγόριθμοι που εξετάζονται επιτυγχάνουν MAE περίπου ίσο με 0,8 όταν χρησιμοποιείται το PCC ως μετρική ομοιότητας, συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

Office products: Σε αυτό το σύνολο δεδομένων, οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,7 (τιμή MAE: 0,330) και 0,6 (τιμή MAE: 0,358), με αντίστοιχα ποσοστά κάλυψης 83,6% και 89,2%. Καθώς οι τιμές κάλυψης είναι αρκετά χαμηλές, θα μπορούσε να χρησιμοποιηθεί η τιμή κατωφλίου 0,2, η οποία δίνει MAE 0,370 και κάλυψη 98,2%. Στην εργασία (Margaris et al., 2020) αναφέρεται ότι για το συγκεκριμένο σύνολο δεδομένων οι αλγόριθμοι που εξετάζονται επιτυγχάνουν MAE περίπου ίσο με 0,68 όταν χρησιμοποιείται το PCC ως μετρική ομοιότητας, συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

Video games: Σε αυτό το σύνολο δεδομένων οι μικρότερες τιμές σφάλματος παρατηρούνται όταν το κατώφλι ομοιότητας είναι 0,7 (τιμή MAE: 0,377) και 0,6 (τιμή MAE: 0,399). Για τις δύο τιμές κατωφλίου έχουμε κάλυψη 82,0% και 88,8% αντίστοιχα. Καθώς οι τιμές κάλυψης είναι αρκετά χαμηλές, θα μπορούσε να χρησιμοποιηθεί η τιμή κατωφλίου 0,1, η οποία δίνει MAE 0,429 και κάλυψη 98,6%. Στην εργασία (Margaris et al., 2020) αναφέρεται ότι για το συγκεκριμένο σύνολο

δεδομένων οι αλγόριθμοι που εξετάζονται επιτυγχάνουν MAE περίπου ίσο με 0,87 όταν χρησιμοποιείται το PCC ως μετρική ομοιότητας, συνεπώς παρατηρούμε ότι η χρήση των συναισθημάτων στη διαδικασία εκτίμησης αξιολογήσεων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων.

Στην μετρική Pearson Correlation Coefficient παρατηρούμε ότι σε όλα τα σύνολα δεδομένων υπάρχει μία τάση μείωσης του MAE καθώς αυξάνεται η τιμή κατωφλίου, με την εξαίρεση του συνόλου δεδομένων Automotive, όπου για μεγάλες τιμές του κατωφλίου υπάρχει αύξηση του MAE και άρα μείωση της ακρίβειας. Σε όλα τα σύνολα δεδομένων υπάρχει μία διακύμανση αναφορικά με τις τιμές της μετρικής MAE, και οι τιμές στο μεγαλύτερο μέρος τους βρίσκονται στην περιοχή [0,34, 0,42], με την εξαίρεση του συνόλου δεδομένων "Video games", όπου παρατηρούνται υψηλότερες τιμές (έως 0,48), κάτι που αποδίδεται στο γεγονός ότι το συγκεκριμένο σύνολο δεδομένων είναι σημαντικά πιο αραιό σε σχέση με τα υπόλοιπα σύνολα δεδομένων.

Σύγκριση επιδόσεων Cosine Similarity και Pearson Correlation Coefficient

Λαμβάνοντας υπ' όψιν τις τιμές του MAE μπορούμε να συνάγουμε ότι καλύτερη και πιο αποδοτική μετρική εκτίμησης ομοιότητας χρηστών είναι η **Pearson Correlation Coefficient,** επειδή δίνει υψηλότερη ακρίβεια στις προβλέψεις, η οποία μάλιστα έχει μειωτική τάση καθώς αυξάνεται η τιμή του κατωφλίου. Ωστόσο, όταν χρησιμοποιείται η μετρική Pearson Correlation Coefficient υπάρχει και μείωση της κάλυψης καθώς αυξάνεται το κατώφλι, συνεπώς χρειάζεται να βρεθεί η χρυσή τομή μεταξύ επιπέδου κάλυψης και ακρίβειας.

6. Σύνοψη – Συμπεράσματα

Στην παρούσα πτυχιακή εργασία παρουσιάσαμε έναν αλγόριθμο ο οποίος χρησιμοποιεί συναισθηματική ανάλυση κειμένων αξιολογήσεων χρηστών προκειμένου να επιτύχει καλύτερη ακρίβεια στις εκτιμήσεις των αριθμητικών αξιολογήσεων των χρηστών, οι οποίες διαμορφώνονται μέσω της προσέγγισης του συνεργατικού φιλτραρίσματος. Τα αποτελέσματα της συναισθηματικής ανάλυσης χρησιμοποιούνται στον υπολογισμό εγγύτητας των χρηστών. Η προσέγγιση αξιολογήθηκε με χρήση έξι συνόλων δεδομένων και καταδείχθηκε ότι η χρήση των συναισθημάτων επιφέρει σημαντική βελτίωση στην ακρίβεια των προβλέψεων. Από τα αποτελέσματα επίσης φαίνεται ότι ανάμεσα στις δύο μετρικές εγγύτητας που θεωρήθηκαν στην πτυχιακή εργασία, δηλαδή την **Cosine Similarity** και την **Pearson Correlation Coefficient**, η Pearson Correlation Coefficient δίνει καλύτερα αποτελέσματα σε ό,τι αφορά την ακρίβεια των προβλέψεων, αν και ειδικά για υψηλές τιμές κατωφλίου ομοιότητας παρουσιάζει μειωμένη κάλυψη στις προβλέψεις.

Πιθανές επεκτάσεις της παρούσας εργασίας θα ήταν η χρήση διαφορετικών μοντέλων συναισθημάτων (Bosco, Patti & Bolioli, 2013), πρόσθετων μετρικών ομοιότητας χρηστών, καθώς και ο συνδυασμός του προτεινόμενου αλγόριθμου με άλλους αλγόριθμους για επαύξηση ακρίβειας, όπως π.χ. οι προτεινόμενοι στα (Margaris et al. 2020b) και (Margaris et al., 2021).

7. Βιβλιογραφία

Bosco, C., Patti, V., & Bolioli, A. (2013). Developing Corpora for Sentiment Analysis: The Case of Irony and Senti-TUT. In IEEE Intelligent Systems (Vol. 28, Issue 2, pp. 55–63). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/mis.2013.28

He, R., & McAuley, J. (2016). Ups and Downs. In Proceedings of the 25th International Conference on World Wide Web. WWW '16: 25th International World Wide Web Conference. International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/2872427.2883037

Khan, Z. Y., Niu, Z., Nyamawe, A. S., & Haq, I. ul. (2021). A Deep Hybrid Model for Recommendation by jointly leveraging ratings, reviews and metadata information. In Engineering Applications of Artificial Intelligence (Vol. 97, p. 104066). Elsevier BV. https://doi.org/10.1016/j.engappai.2020.104066

Margaris, D., Spiliotopoulos, D., Karagiorgos, G., & Vassilakis, C. (2020). An Algorithm for Density Enrichment of Sparse Collaborative Filtering Datasets Using Robust Predictions as Derived Ratings. In Algorithms (Vol. 13, Issue 7, p. 174). MDPI AG. https://doi.org/10.3390/a13070174

Margaris, D., Spiliotopoulos, D., Vassilakis, C., & Vasilopoulos, D. (2020b). Improving collaborative filtering's rating prediction accuracy by introducing the experiencing period criterion. In Neural Computing and Applications. Springer Science and Business Media LLC. https://doi.org/10.1007/s00521-020-05460-y

Margaris, D., Spiliotopoulos, D., Karagiorgos, G., Vassilakis, C., & Vasilopoulos, D. (2021). On Addressing the Low Rating Prediction Coverage in Sparse Datasets Using Virtual Ratings. In SN Computer Science (Vol. 2, Issue 4). Springer Science and Business Media LLC. https://doi.org/10.1007/s42979-021-00668-8

McAuley, James. Amazon product data. https://jmcauley.ucsd.edu/data/amazon/ (τελευταία πρόσβαση στις 20 Νοεμβρίου 2021)

Nikicc, "Twitter Emotion Recognition", https://github.com/nikicc/twitter-emotion-recognition (τελευταία πρόσβαση στις 20 Νοεμβρίου 2021)

Wu, H., Zhang, Z., Yue, K., Zhang, B., & Zhu, R. (2017). Content Embedding Regularized Matrix Factorization for Recommender Systems. In 2017 IEEE International Congress on Big Data (BigData Congress). 2017 IEEE International Congress on Big Data (BigData Congress). IEEE. https://doi.org/10.1109/bigdatacongress.2017.36