Μαρίνος Μάριος (Dai17147) Recommender Systems in Python.

# Σχολιασμός Κώδικα.

Όσον αφορά τον κώδικα, χρησιμοποιήσα Python και την βιβλιοθήκη pandas συγκεκριμένα. Πέρα από την προεπεξεργασία που αναφέρεται, έκανα και κάποια πράματα ακόμη. Αρχικά, αφαίρεσα τις στήλες 'imageUrlS', 'imageUrlM', 'imageUrlL' μιας και δεν χρειάζονται πουθενά. Έπειτα, κοίταξα αν οι βαθμολογίες που υπάρχουν στο books\_ratings csv υπάρχουν και τα αντίστοιχα τους βιβλία στο books.csv οπότε όποιες βαθμολογίες δεν έχουν το αντίστοχο βιβλίο τις αφαίρεσα από το ratings dataframe. Το ίδιο έκανα και για τους χρήστες αλλά εκεί όλοι αντιστοιχούσαν. Έπειτα σε δύο συγκεκριμένα βιβλία είχε year = 'DK Publishing Inc' ως λάθος και βλέποντας τις αντίστοιχες εγγραφές παρατήρησα οτι ήταν όλα μία στήλη πιο δεξιά (δηλαδή στην στήλη bookAuthor είχε την τιμή του bookTitle κ.ο.κ.) από όσο έπρεπε άρα διόρθσα τις αντίστοιχες γραμμές. Το ίδιο ίσχυε για μία γραμμή με year = Gallimard και έκανα τα ίδια. Συνεχίζοντας, διόρθωσα τις τιμές των years που υπάρχουν και είναι άκυρες, δηλαδή τις τιμές πάνω από το 2005 μιας και το dataset δημιουργήθηκε τότε και το θεωρούμε ως άνω όριο ενώ και τα βιβλιά που έχουν year = 0 και τα αντικατέστησα με τον μέσο όλων των υπόλοιπων years. Έπειτα κοίταξα για ποιές γραμμές δεν έχουν τιμή στο bookAuthor και βρήκα οτι ήταν μία μόνο οπότε έψαξα ποιός είναι ο συγγραφέας και το αντικατέστησα. Για την εξαγωγή keywords από τα bookTitles χρησιμοποιήσα την βιβλιοθήκη rake\_nltk και πιο συγκεκριμένα το Rake.

# Αποτελέσματα πειράματος.

Για τα 10 αρχεία των 5 χρηστών (τα οποία έχουν ως όνομα Jaccard αν χρησιμοποιήθηκε η πρώτη συνάρτηση ή DiceCoef αν χρησιμοποιήθηκε η δεύτερη και δίπλα το userID του χρήστη), η ομοιότητα μεταξύ των λιστών είναι 100% μιας και είναι **ακριβώς ίδιες** οι δύο λίστες με **ίδια σειρά** και συνεπώς η golden standard λίστα αποτελεσμάτων κάθε ένα χρήστη είναι η ίδια άρα δεν χρειάζεται να προβούμε σε κάποια περαιτέρω διαδικασία.

# Σχολιασμός και άλλες προσεγγίσεις.

Αρχικά, για τους 5 χρήστες υπάρχει μία τάση στο να υπάρχουν υψηλότερες βαθμολογίες χρησιμοποιηώντας την συνάρτηση με την Jaccard similarity σε αντίθεση με την DiceCoef, το οποίο είναι και αναμενόμενο μιας και για παράδειγμα με την Jaccard μπορούμε να έχουμε **80% ομοιότητα** σε δύο βιβλία χωρίς να περιέχεται **καμία** λέξη κλειδί των βιβλίων του χρήστη με κάποιο από τα βιβλία ( αν το έτος και ο συγγραφέας είναι ακριβώς ίδιοι.), είναι προφανές οτι δίνεται πολύ βαρύτητα στο έτος και στον συγγραφέα ενώ από την άλλη στην συνάρτηση με το DiceCoef δίνεται πολύ βαρύτητα στις λέξεις-κλειδιά και για αυτό το λόγο έχουμε αρκετά μικρές βαθμολογίες μιας όπου φυσιό είναι να μην ανήκουν **όλες** οι λέξεις κλειδιά ενός βιβλίου στις λέξεις κλειδιά των 3 ‘καλύτερων’ βιβλιών του χρήστη. Φαίνεται άρα οτι οι δύο συναρτήσεις είναι βασίζονται πάρα πολύ στον συγγραφέα και στο έτος οπότε αποφάσισα να ισσοροπήσω την εξίσωση βάζοντας και τον εκδοτικό οίκο του κάθε βιβλίου όπου είναι η ίδια προσέγγιση με τον συγγραφέα (δηλ. αν είναι ο ίδιος εκδοτικός οίκος 1 αλλιώς 0), με **ίσους συντελεστές βαρύτητας 25%** στο καθένα (Author, publisher, Keywords, year) ώστε να έχουν όλα ισάξια βαρύτητα χρησιμοποιώντας και την jaccard αλλά και το dicecoef(τα αρχεία αυτά είναι στον φάκελο publisher 0.25 all ενώ τα αρχικά αρχεία που ζητάτε στον φάκελο original functions) . Βλέποντας τα αποτελέσματα όπως ήταν αναμενόμενο διαφέρουν, αλλά επίσης φαίνεται οτι σε γενικές γραμμές συμβαδίζουν και κυρίως αλλάζει η κατάταξη των 10 βιβλίων οπότε ίσως μπορούμε να πούμε οτι έχουμε αρκετά ακριβείς προτάσεις όσον αφορά το ‘προφιλ’ του χρήστη. Επίσης θα μπορούσαν να γίνουν πολλές ακόμα παραλλαγές βάζοντας μέσα και σαν παράμετρο την ηλικία των χρηστών. Επίσης, θα μπορούσαμε να ‘παίξουμε’ με τους συντελεστές κάθε παραμέτρου και να δοκιμάσουμε διάφορα μέχρι να πάρουμε ένα ικανοποιητικό αποτέλεσμα ή ακόμη και να αλλάξουμε τον τρόπο υπολογισμού της ομοιότητας των λέξεων-κλειδιών με κάποια άλλη μετρική ομοιότητας και όχι την Jaccard ή την DiceCoef.