Programming Project 2 – Recommender System

The goal of this project is the implementation and evaluation of a collaborative item-item recommender system for movies based on the use of the pearson coefficient as its similarity measure. The system may be implemented in any programming language of your choice (but the use of python is recommended). You will use the MovieLens dataset (https://grouplens.org/datasets/movielens/latest/), and in particular, the small dataset with 100000 ratings (only the ratings file is required).

You will implement four prediction functions that given the number of nearest neighbors *N* they will calculate their prediction based on:

1. the weighted average
2. the weighted average with adjustment of the average user rating and minus the neighbors bias (lec. 11, slide 53)
3. the weighted average where the weights will be based on the number of common users that have rated the two items. That is, the more common users have rated a neighbor of the item for which we are predicting its rating, the greater this neighbors weight. Define your own weight function
4. the weighted average where the weights will be based on the variance of the ratings of each item. That is, the greater the variance in the ratings of an item (neighbor), the greater its weight in the weighted average. Define your own weight function.

Note: In all cases, the nearest N items (neighbors) that contribute to the prediction are chosen based on pearson similarity and they must have a rating by the user for which we are giving a prediction.

Your program will split the data to training set T% and testing set (100-T)%. Then, for its evaluation it will show the confusion matrix and it calculate: (a) the mean absolute error (MAE), (b) macro average precision and (c) macro average recall. For the evaluation of the binary measures consider that a movie is considered relevant if its rating is >=3.

Run the following experiment:

- For a fixed training set of 80%, test set 20% and for 5 values of *N* compare the 4 prediction functions.
- The experiment should be run 5 times and the reported evaluation measures should be the average of all runs.

What you will submit:

a) your code (your code must include the code that runs the experiment)

b) a report in which you will present and discuss your results. In the report, regarding your code, discuss any optimization or assumption you made that is not stated in the assignment, and also describe the weight functions 3 and 4 you have defined. For the experiment, show a corresponding table of results and plots of the measures.

Note: If you have efficiency problems with the data size, choose a way to reduce their size (e.g., sampling, filtering). Describe how you reduced your data and report the new sizes.

Στόχος της εργασίας είναι η υλοποίηση και η αποτίμηση ενός συστήματος συστάσεων συνεργατικού φιλτραρίσματος αντικειμένου-αντικειμένου για ταινίες με χρήση του συντελεστή pearson ως μέτρο ομοιότητας. Η υλοποίηση μπορεί να γίνει σε όποια γλώσσα προγραμματισμού προτιμάτε (συνιστάται η χρήση python). Θα χρησιμοποιήσετε τα δεδομένα του MovieLens (https://grouplens.org/datasets/movielens/latest/) και συγκεκριμένα το μικρό σύνολο με τις 100000 βαθμολογίες (απαιτείται μόνο το αρχείο με τα ratings).
Υλοποιήστε τέσσερις συναρτήσεις πρόβλεψης που δεδομένου του πλήθους κοντινότερων γειτόνων που λαμβάνουν υπόψη *N* χρησιμοποιούν για την πρόβλεψη:

1. σταθμισμένο μέσο όρο
2. σταθμισμένο μέσο όρο με προσαρμογή της μέσης βαθμολογίας του χρήστη και αφαίρεση του bias των γειτόνων (διαλ. 11, διαφ. 53)
3. σταθμισμένο μέσο όρο στον οποίο όμως η στάθμιση θα βασίζεται στο πλήθος των κοινών χρηστών που έχουν βαθμολογήσει τα δύο αντικείμενα. Δηλαδή, όσο περισσότεροι κοινοί χρήστες έχουν βαθμολογήσει έναν γείτονα με το αντικείμενο για το οποίο παράγουμε την πρόβλεψη, τόσο μεγαλύτερο το βάρος του γείτονα. Ορίστε δική σας συνάρτηση στάθμισης.
4. σταθμισμένο μέσο όρο στον οποίο η στάθμιση θα βασίζεται στην διακύμανση των βαθμολογιών του κάθε αντικειμένου. Δηλαδή, όσο μεγαλύτερη η διακύμανση που υπάρχει στη βαθμολογία ενός αντικειμένου, τόσο μεγαλύτερο το βάρος του στον σταθμισμένο μέσο όρο. Ορίστε δική σας συνάρτηση στάθμισης.

Προσοχή: Σε όλες τις περιπτώσεις τα κοντινότερα N αντικείμενα (γείτονες) που συμμετέχουν στην πρόβλεψη επιλέγονται με την ομοιότητα pearson, και θα πρέπει να έχουν βαθμολογία από τον χρήστη για τον οποίο βγάζουμε την πρόβλεψη.
Το πρόγραμμα σας θα χωρίζει τα δεδομένα σε σύνολο εκπαίδευσης Τ% και σύνολο ελέγχου (100-Τ)%. Στη συνέχεια, για την αποτίμηση του συστήματος θα εμφανίζει τον πίνακα σύγχυσης και θα υπολογίζει (α) μέσο απόλυτο σφάλμα (MAE), (β) μέση ακρίβεια (macro average precision) και (γ) μέση ανάκληση (macro average recall). Για τον υπολογισμό των δυαδικών μέτρων αποτίμησης θεωρείστε μια ταινία ως σχετική αν ο βαθμός της είναι >=3.
Πραγματοποιήστε το εξής πείραμα:

- Για σταθερό σύνολο εκπαίδευσης 80%, σύνολο ελέγχου 20% και για 5 τιμές *N* συγκρίνετε τις 4 συναρτήσεις πρόβλεψης.
- Το πείραμα θα πρέπει να εκτελεστεί 5 φορές και τα μέτρα αποτίμησης που θα αναφερθούν να είναι οι μέσοι όροι όλων των εκτελέσεων.

Θα παραδώσετε:
α) τον κώδικα σας (ο κώδικας θα πρέπει να περιλαμβάνει τον κώδικα εκτέλεσης των δύο πειραμάτων)
β) μια αναφορά στην οποία θα παρουσιάζετε και θα σχολιάζετε τα αποτελέσματα σας.
Στην αναφορά σε σχέση με τον κώδικα σας, σχολιάστε όποια βελτιστοποίηση ή υπόθεση κάνατε για οτιδήποτε δεν καθορίζεται από την εκφώνηση και περιγράψτε τις συναρτήσεις στάθμισης 3 και 4. Για τα πειράματα εμφανίστε σχετικούς πίνακα με τα αποτελέσματα και γραφικές παραστάσεις των μέτρων αποτίμησης.

Σημείωση: Αν υπολογιστικά αντιμετωπίσετε πρόβλημα με το μέγεθος των δεδομένων, επιλέξτε έναν τρόπο να μειώσετε το μέγεθος τους (π.χ. δειγματοληψία, φιλτράρισμα). Περιγράψτε πώς κάνατε την μείωση και αναφέρετε τα νέα μεγέθη αρχείων.