CptS 315: Introduction to Data Mining
Programming Assignment 2 (PA2)

Instructions

- Supported programming languages: Python, Java, C++
- Store all the relevant files in a folder and submit the corresponding zipfile (.zip).
- This folder should have a script file named

  `run_code.sh`

  Executing this script should do all the necessary steps required for executing the code including compiling, linking, and execution
- Assume relative file paths in your code. Some examples:

  ‘‘`./filename.txt`’’ or ‘‘`../hw1/filename.txt`’’
- You should submit your zipfile to Blackboard by the stated due date.

Programming Assignment Explanation

Movie Recommendations via Item-Item Collaborative Filtering. You are provided with real-data (Movie-Lens dataset) of user ratings for different movies. There is a *readme* file that describes the data format. In this project, you will implement the *item-item collaborative filtering* algorithm that we discussed in the class. The high-level steps are as follows:

a) Construct the profile of each item (i.e., movie). At the minimum, you should use the ratings given by each user for a given item (i.e., movie). Optionally, you can use other information (e.g., genre information for each movie and tag information given by user for each movie) creatively. If you use this additional information, you should explain your methodology in the submitted report.

b) Compute similarity score for all item-item (i.e., movie-movie) pairs. You will employ the *centered cosine* similarity metric that we discussed in class.

c) Compute the neighborhood set N for each item (i.e. movie). You will select the movies that have highest similarity score for the given movie. Please employ a neigborhood of size 5. Break ties using lexicographic ordering over movie-ids.

d) Estimate the ratings of other users who didn't rate this item (i.e., movie) using the neighborhood set. Repeat for each item (i.e., movie).

e) Compute the recommended items (movies) for each user. Pick the top-5 movies with highest estimated ratings. Break ties using lexicographic ordering over movie-ids.

Your program should output top-5 recommendations for each user.

Output Format:

- The output of your program should be dumped in a file named "output.txt" in the following format. One line for each user.

  User-id1 movie-id1 movie-id2 movie-id3 movie-id4 movie-id5
  User-id2 movie-id1 movie-id2 movie-id3 movie-id4 movie-id5
  ...
  ...

  Explanation.

  - Line 1 should have the first user-id followed by the movie-ids of recommended movies.
  - Line 2 should have the second user-id followed by the movie-ids of recommended movies.

- Make sure the output.txt file is dumped when you execute the script

  `run_code.sh`