CptS 315: Introduction to Data Mining
Homework 2

(Due date: Mar 5, midnight PST)

Instructions

- Please use a word processing software (e.g., Microsoft word) to write your answers and submit a printed copy to me at the beginning of the class on Feb 8. The rationale is that it is sometimes hard to read and understand the hand-written answers.
- All homeworks should be done individually.

Analytical Part (40 points)

Q1. Consider the following ratings matrix with three users and six items. Ratings are on a 1-5 star scale. Compute the following from data of this matrix: (20 points)

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	4	5		5	1	
User 2		3	4	3	1	2
User 3	2		1	3		4

Table 1: Data of ratings from three users for six items.

a) Treat missing values as 0. Compute the jaccard similarity between each pair of users.

b) Treat missing values as 0. Compute the cosine similarity between each pair of users.

c) Normalize the matrix by subtracting from each non-zero rating, the average value for its user. Show the normalized matrix.

d) Compute the (centered) cosine similarity between each pair of users using the above normalized matrix.

Q2. Please read the following two papers and write a brief summary of the main points in at most TWO pages. (20 points)

Brent Smith, Greg Linden: Two Decades of Recommender Systems at Amazon.com. IEEE Internet Computing 21(3): 12-18 (2017)
`https://www.computer.org/csdl/mags/ic/2017/03/mic2017030012.pdf`

Greg Linden, Brent Smith, Jeremy York: Industry Report: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Distributed Systems Online 4(1) (2003) `https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf`

Programming and Experimental Part (60 points)

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.

Instructions for Code Submission and Output Format.
Please follow the below instructions. It will help us in grading your programming part of the homework.

- Supported programming languages: Python, Java, C++
- Store all the relevant files in a folder and submit the corresponding zipfile named after your student-id, e.g., 114513209.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 ``../hw2/filename.txt''
  ```

- 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
  ```

- Zip the entire folder and submit it as

  ```
  <student_id>.zip
  ```

Grading Rubric

Each question in the students work will be assigned a letter grade of either A,B,C,D, or F by the Instructor and TAs. This five-point (discrete) scale is described as follows:

- **A) Exemplary (=100%).**
  Solution presented solves the problem stated correctly and meets all requirements of the problem.
  Solution is clearly presented.
  Assumptions made are reasonable and are explicitly stated in the solution.
  Solution represents an elegant and effective way to solve the problem and is not overly complicated than is necessary.

- **B) Capable (=75%).**
  Solution is mostly correct, satisfying most of the above criteria under the exemplary category, but contains some minor pitfalls, errors/flaws or limitations.

- **C) Needs Improvement (=50%).**
  Solution demonstrates a viable approach toward solving the problem but contains some major pitfalls, errors/flaws or limitations.

- **D) Unsatisfactory (=25%)**
  Critical elements of the solution are missing or significantly flawed.
  Solution does not demonstrate sufficient understanding of the problem and/or any reasonable directions to solve the problem.

- **F) Not attempted (=0%)**
  No solution provided.

The points on a given homework question will be equal to the percentage assigned (given by the letter grades shown above) multiplied by the maximum number of possible points worth for that question. For example, if a question is worth 6 points and the answer is awarded a B grade, then that implies 4.5 points out of 6.