香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

CIE6006/MCE5918
Data Analytics

Learning Objectives

Objective: Apply collaborative filtering algorithms to real-world TV shows dataset, implement user-user and item-item recommendation systems, and evaluate recommendation performance for personalized content discovery.

Due Date: Sunday, November 30, 11:59:59 PM

Core Challenge: Building practical recommendation systems for TV shows and understanding the differences between collaborative filtering approaches in real-world scenarios.

Assignment Overview

Your Main Task: Implement and evaluate collaborative filtering systems on a real TV shows dataset containing 9985 users and 563 popular TV shows. You will analyze user viewing patterns, compute similarity matrices, and generate personalized recommendations.

Your Workspace:

1. You will work primarily within the following notebook: TVShowsRecommendation.ipynb
2. Complete the main supporting Python files: tv_recommendation.py (main recommendation engine), similarity_computation.py (similarity matrix calculations), evaluation_metrics.py (recommendation evaluation)

Core Task

What you need to code:

1. **Data Processing:** Load and analyze the user-shows.txt matrix (9985 users × 563 shows) where Rij = 1 if user i watched show j over a 3-month period.
2. **User Analysis:** Focus on the 500th user (Alex with Python's 0-based indexing: users[499]) as your test case for recommendation generation.
3. **Missing Data Simulation:** The first 100 entries of Alex's row are erased and replaced with 0s. This creates a realistic prediction scenario.
4. **MAIN OBJECTIVE::** Implement both user-user and item-item collaborative filtering to generate top-5 recommendations for Alex.

For sanity check, your highest recommendation score for user-user collaborative filtering should be above 900, and your highest recommendation score for movie-movie filtering should be above 31.

Implementation Requirements

1.Matrix Preprocessing:

- Compute matrices P (user degree matrix) and Q (item degree matrix) as diagonal matrices
- P[i,i] = number of shows user i watched
- Q[j,j] = number of users who watched show j

2. User-User Collaborative Filtering:

- Compute user similarity matrix using cosine similarity
- Generate recommendations for Alex using similar users' preferences

3. Item-Item Collaborative Filtering:

- Compute item similarity matrix using cosine similarity
- Generate recommendations for Alex based on item similarities

4. Results Analysis: Compare recommendation outputs and identify show names using shows.txt

Dataset Overview

Key Dataset Characteristics:

- 9985 users with diverse viewing preferences
- 563 popular TV shows across various genres
- Binary interaction data (1 = watched, 0 = not watched)
- Represents realistic sparse data common in recommendation systems

Target Analysis:

- Focus on user Alex (500th user in the dataset)
- Predict preferences for the first 100 shows in the catalog
- Generate top-5 recommendations using collaborative filtering methods
- Compare effectiveness of user-based vs. item-based approaches

Grading Criteria

THE CHINESE UNIVERSITY OF HONG KONG, SHENZHEN

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Component	Weight	Key Requirements
Correctness of Code	**40%**	All required components (similarity matrices, collaborative filtering algorithms, recommendation generation, etc) are correctly implemented. Code runs without error and passes the provided tests.
Completeness of Experiments	**30%**	All notebooks are completed and executable without error. Recommendation results are achieved and properly reported. Output.pdf is clear. **Important: Please make sure that the notebooks have been run and the cell outputs are visible.**
Quality and Depth of Report	**30%**	Clear analysis on recommendation system design and performance (especially in the collaborative filtering comparison); final report is concise, logical, and well-structured.
Extra Credit	**Bonus**	Meaningful attempts at advanced improvements (e.g., alternative similarity metrics, hybrid recommendation approaches, evaluation metrics, etc.) in the recommendation system, with clear implementation and analysis, will be considered for extra credit.

Submission Instructions

THE CHINESE UNIVERSITY OF HONG KONG, SHENZHEN

Prepare your codes: Once you have completed all notebooks and filled out the necessary code, you need to follow the below instructions to submit your work:

1. Open **collect_submission.ipynb** in Colab and execute the notebook cells.
2. This notebook/script will help you:

a) Generate a zip of your code (.py and .ipynb) called **a3_code_submission.zip**.

b) Convert all notebooks into **a single PDF file**.

Note: If you encounter issues (e.g., environment errors, missing content in the auto-generated PDF) with the above script, use manual screenshots to generate the PDF instead.

Submission Instructions

THE CHINESE UNIVERSITY OF HONG KONG, SHENZHEN

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

1. **Prepare the following 3 files:**
   1. a3_inline_submission.pdf: Your notebook exported to PDF, with all output cells visible.
   2. a3_code_submission.zip: Your Python implementation file and final Jupyter Notebook files.
   3. Student_id_report.pdf: Your assignment report (use the provided template).

2.Packaging and Naming:

1. Place all files into a single folder.
2. Compress this folder into a **ZIP file** named: **your_student_id_ass3.zip** (e.g., zhan1234_ass3.zip).
3. Please double check your submitted file to ensure that the correct assignment has been successfully uploaded to the BB system.

3.Deadline: [2025-11-30], 23:59:59

Provided Materials

Starter Code (.zip):

Click here to download the starter code zip file

Highly Recommended Resources:

Collaborative Filtering

Kaggle