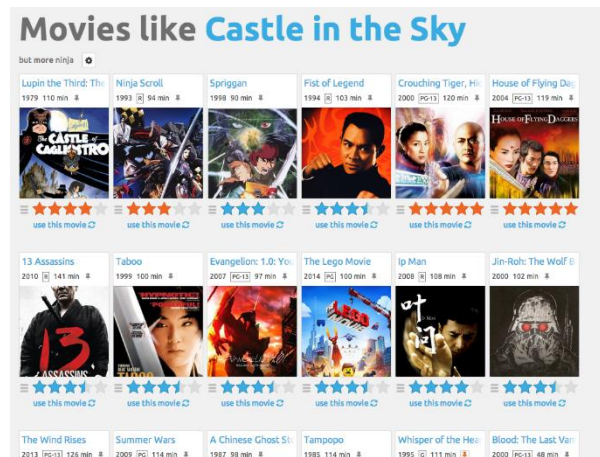


Movie Recommendation System

Objective

The goal of this project is to develop a movie recommendation system that can generate personalized recommendations based on user preferences and prior movie-watching history. Students will choose an appropriate algorithm, build the system in R, and evaluate it using a custom similarity-based scoring method.



Project Guidelines

1. Dataset

- Use the [MovieLens](#) Dataset as the primary source for user-movie interaction data. The dataset provides movie metadata and user ratings, suitable for various recommendation techniques. It contains 20000263 ratings and 465564 tag applications across 27278 movies.
- Each group is responsible for loading and preprocessing the dataset as needed for their chosen algorithm.

2. Algorithm Selection

Students may select any technique from the data mining methods covered in this class, including classification, association mining, or clustering techniques that are suitable for recommendation tasks. Examples include:

- Collaborative filtering (user-based, item-based)
- Content-based filtering
- Hybrid recommendation models
- Neural networks or matrix factorization techniques

3. System Requirements

Design a gateway, such as a simple user interface (UI) or command-line function, where users can input previously watched movies and current interests to receive three personalized movie recommendations. Available gateway options in R include Plumber, Shiny, RDS files, Streamlit, and others.

- This will be used to Test and Demo how the recommendation system works in real-time.
- Ensure that the gateway can process multiple input sets in bulk. For each input set, the system should return three movie recommendations.
- Each input set should simulate how well their recommendation matches expected preferences in the unseen test data.

4. Evaluation and Scoring Parameters

- X: Input list of movies
- Y: Actual movies that complete the recommendation (Output list)
- Z: Recommended movies by the algorithm
- $C(X, Y)$ = Items common in both lists X, Y.
- Scoring for each recommendation = $\frac{C(Z, Y) - C(Z, X)}{3}$
- Model Score is the average of all the scores.

5. Peer Review

- After completion, each group will record a 3-minutes video that demos the system and upload to canvas.
- Other groups will review the work and provide feedback on both the algorithm and result based on the following criteria:
 - Creativity and soundness of approach
 - Quality of recommendations (result)
 - Presentation clarity
- Each group will grade their peers on a scale from 1 to 7 focusing on

6. Submission

Ensure that the following are submitted:

- A detailed report explaining the algorithm choice, implementation, and the result. (20 points)
- Provide a GitHub link on a well-organized code (20 points)
- A 3-minute demo video submitted on canvas and **in-class presentation** (20 points)
- Peer Review (30 points)
- Instructors Review (10 points)