

# Assignment: Item Ranking Recommendation System

## Objective

Build a simple **recommendation engine** that, given a set of input items and a user, outputs a **ranked list of recommendations**. The goal is to simulate a personalized ranking task using structured data.

## Task Instructions

You are tasked with building a **ranking recommendation system** for a movie platform using **MovieLens 100K dataset**:

Dataset: <https://grouplens.org/datasets/movielens/100k/>

Your engine should be able to output a **ranked list of movies** for a given user and a given subset of candidate movies.

---

## Your Task

### 1. Data Exploration & Preprocessing

- Load and explore the data.
- Merge relevant tables to create a training-ready dataset (user features, movie metadata, ratings).

### 2. Modeling User Preferences

- Build a model that predicts the **relevance or rating** of a movie for a given user.
- Use features like user demographics, movie genres, or past ratings.

### 3. Ranking Logic

- Given a user ID and a list of candidate movie IDs, output a **ranked list of those movies**.
- Sort by predicted rating or affinity score.
- Optionally, provide a short explanation of **why** a movie is ranked highly

#### 4. Code Organization & Reproducibility (30 mins)

- Modularize your code into logical sections: preprocessing, modeling, scoring.
  - Include a sample script or notebook to:
    - Train the model
    - Provide ranked movie recommendations for a sample user + candidate set
- 

#### Optional

- Implement a **baseline model vs. your ML model**, and compare on ranking metrics.
  - Wrap the ranking logic in a small **FastAPI app** that accepts user ID + movie list and returns rankings.
  - Use embeddings to power the ranking.
- 

#### Submission Instructions

Please submit:

- A GitHub repo or zip containing:
  - Scripts or notebooks
  - `README.md` explaining:
    - Your approach
    - Model choice rationale
    - How to run the code and reproduce results
  - `requirements.txt` or `conda.yaml`