Logan Laszewski - Movie Recommendations – README

A user-based collaborative filtering system that recommends movies based on your ratings and similar users' preferences.

Open in Colab:
https://colab.research.google.com/drive/1mq2hmqPqFY6gWli_IPyTiJCV-bKac8Kr?usp=sharing

Project Goal

I've always enjoyed discovering new movies, and I wanted to explore how data can make recommendations more personal and meaningful. While streaming platforms like Netflix already recommend films, they often rely on what's been *watched*, not necessarily what someone *enjoyed*.
This project applies a **user-based collaborative filtering** approach to identify people with similar movie preferences and suggest films they're likely to enjoy. It also served as a great opportunity to practice **data wrangling, similarity analysis, and recommendation logic** while working with a massive dataset of movie ratings from MovieLens

Project Overview

The app takes a user's ratings for 8–20 movies and compares them to millions of ratings from anonymous MovieLens users. It then finds users with similar "taste profiles" and recommends new movies that those users enjoyed.
This demonstrates how personalized recommendations can be generated using Python and statistical methods, **no deep learning or complex model training required**.

How It Works

1. Datasets

- **MovieLens** dataset (32+ million anonymous ratings).
- Two key files are used:
- movies.csv - movie titles and genres
- Ratings.csv. - user ratings (1-5 stars)

2. Merge Datasets

- Combine movie titles, genres, and user ratings into a single dataset.
- The merged data contains **84,000+ movies** across nearly **1,800 genres** (definitely some overlap)

3. User Input

- The user enters between 8–20 movies they've seen and rates them on a 1–5 scale. This provides enough data to identify similar users while keeping comparisons efficient.

4. Find Similar Users

- Filter to users who have rated at least **70%** of the same movies. This ensures overlap for meaningful comparisons and avoids unnecessary computations.

5. Compute Similarity

- Use the Pearson correlation coefficient to measure how similar two users' rating patterns are:
    - **+1.0** → identical taste
    - **0.0** → no relationship
    - **–1.0** → opposite taste

- Keep only users with **positive correlations**, and select the **top 10 most similar "neighbors."**

6. Recommend Movies

From these neighbors, recommend movies that:

- Have an average rating of **4.0 or higher**
- Were rated by **at least 4 neighbors**

Each recommendation's score is a **weighted average**, where more similar users have greater influence.
To reduce "mainstream bias," a **popularity penalty** was added — slightly lowering scores for widely rated, blockbuster movies so that more unique recommendations can surface.

7. Fuzzy Matching (RapidFuzz)

To make the system user-friendly, **fuzzy title matching** allows users to input movie names naturally (e.g., typing *"Knives Out Mystery"* should match *"Glass Onion: A Knives Out Mystery (2022)"*). If it doesn't, the user has the option to retype in the target movie or go on to the next. This improves accuracy when searching movie titles from user input.

Packages Needed

This project relies on Python's data analysis and computation ecosystem:

Pandas - Data manipulation, loading CSV files (ratings.csv, movies.csv), merging datasets, and tabular operations.

Numpy - Numerical operations, vector math, and rating normalization.

scipy.stats (pearsonr) - Computes Pearson correlation to measure similarity between users based on their movie ratings.

Rapidfuzz - Enables fuzzy string matching for movie title lookups and user input handling.

Os - File and directory handling for organizing datasets.

Zipfile - Extracts the downloaded MovieLens dataset from compressed .zip files.

urllib.request (urlretrieve) - Downloads the dataset directly from GroupLens if it's not already present.

Install Requirements:

```
pip install pandas numpy scipy rapidfuzz
```

Notes

If you don't want to download the dataset programmatically, you can manually grab it from: https://grouplens.org/datasets/movielens/32m/

Then upload movies.csv and ratings.csv into the directory.

A Google Colab notebook is provided for quick testing.