Netflix-Clone DS5110 Project

Liam

October 26 2025

1 Project Introduction

This project aims to design and implement a **Netflix-clone movie recommendation system**, focusing on the creation of a minimal viable product (MVP) that demonstrates data-driven personalization through collaborative and content-based filtering techniques. The system enables users to browse, rate, and receive movie recommendations based on their preferences. The development of this project is part of a broader exploration of recommended systems, which combine database design, data engineering, and machine learning. Since this is an independent project, all development, analysis, and documentation were completed by the project author.

2 Scope and Objectives

2.1 Project Scope

The scope of this project focuses on creating a functional prototype rather than a fully commercial streaming platform. The MVP includes:

- A back-end service for managing users, movies, and ratings.
- A front-end interface for movie browsing and rating.
- Machine learning models for recommendation generation.
- Analytical SQL reports highlighting user and movie insights.

The project excludes copyright-protected video streaming, paid APIs, or production deployment. It would only use publicly available datasets and open-source tools.

2.2 Objectives

1. Build a scalable SQL database schema with key tables: `users`, `movies`, `ratings`, and `recommendations`.

2. Clean and preprocesses the data to handle missing values, normalize ratings, and extract relevant features.

3. Implement and evaluate at least three recommendation algorithms: Collaborative Filtering, Content-Based Filtering, and a hybrid model.

4. Develop at least eight analytical SQL reports that address user behavior and movie popularity.

5. Create a clear and reproducible development workflow tracked via version control and progress logs.

3 Team Contributions

As a solo developer, all responsibilities were handled independently. Contributions are categorized below:

- **Data Engineering:** Dataset cleaning, feature extraction, and SQL schema design.
- **Machine Learning:** Implementation of recommendation algorithms using Python.
- **Front-end Development:** Building a simple user interface using Flask and HTML templates.
- **Backend Development:** API creation and integration with database queries.
- **Reporting and Documentation:** Creation of analytical SQL reports, progress tracking, and final documentation.

4 Tools and Programming Languages

- **Programming Languages:** Python, SQL, JavaScript, HTML/CSS.
- **Frameworks:** Flask, Pandas, Scikit-Learn, Surprise.
- **Database:** PostgreSQL (alternatively SQLite for local testing).
- **Version Control:** Git and GitHub.
- **Development Environment:** Visual Studio Code and Jupyter Notebook.

5 Dataset and Data Access

The project utilizes the MovieLens 100K dataset provided by GroupLens, which contains 100,000 ratings from 943 users on 1,682 movies. The dataset is publicly available and can be accessed here: `https://grouplens.org/datasets/movielens/100k/`

6 Progress Tracking

Progress has been tracked throughout the development cycle using an Excel file. The tracker records milestones, dates, deliverables, and status updates. Ensure that project goals, from data preparation to algorithm testing, are effectively monitored.

7 Verification and Readiness for Review

All project requirements have been met according to the instructor's feedback and specifications:

- MVP implemented using open data sources.
- SQL schema with key relationships and indexes.
- Feature engineering and data normalization steps documented.

- Analytical SQL queries (at least eight) created and validated.
- Multiple recommendation algorithms implemented and evaluated.

8 GitHub Repository

All files are uploaded and maintained in the GitHub repository:

- `/src/` – Source code for back-end, front-end, and models.
- `/data/` – Dataset (link of the CSVs).
- `/docs/` – PDF report and LaTeX source.
- `/tracking/` – Excel progress tracker.