

Movie Recommendation System Project

Haidy Mohamed 2305309

Arwa Ayman 2305137

Hagar Amr 2305197

Movie Recommendation System Project Report

Based on the code you've shared, I'll provide a comprehensive overview of this hybrid movie recommendation system project.

Project Overview

This project implements a sophisticated movie recommendation system that combines both content-based and collaborative filtering approaches. The resulting hybrid model leverages the strengths of both recommendation strategies to provide personalized movie suggestions to users. The system is implemented as a Streamlit web application with an interactive user interface, complete with data visualizations.

System Architecture

The recommendation system follows a modular architecture:

1. **Core Components:**
 - `HybridRecommender`: Combines content-based and collaborative filtering recommendations
 - `ContentBasedRecommender`: Recommends movies based on content similarity (genres)
 - `CollaborativeFilteringRecommender`: Recommends movies based on user-item interactions
 - `Evaluation`: Tools for measuring model performance
2. **Data Processing:**
 - Data loading and preprocessing utilities
 - Cleaning and transformation operations
3. **User Interface:**
 - Streamlit web application
 - Interactive controls and visualizations
 - Real-time recommendation generation

Implementation Details

Data Sources

The system uses the MovieLens dataset, which includes:

- Movie information (title, genres)
- User ratings
- Tags (optional)

- External links

Recommendation Algorithms

1. **Content-Based Filtering:**
 - Utilizes TF-IDF vectorization on movie genres
 - Computes cosine similarity between movies
 - Recommends movies similar to a specified title
2. **Collaborative Filtering:**
 - Implements Singular Value Decomposition (SVD) using Surprise library
 - Predicts user ratings for unwatched movies
 - Handles the cold-start problem when a user has no history
3. **Hybrid Approach:**
 - Combines scores from both models with adjustable weights
 - Normalizes and merges recommendations
 - Provides a balanced set of suggestions

User Interface Features

The Streamlit web application offers:

1. **Input Controls:**
 - User ID selection
 - Movie title search with autocomplete
 - Genre filtering
 - Model weight adjustment sliders
 - Number of recommendations selector
2. **Visualizations:**
 - Recommendation scores bar chart
 - Genre distribution pie chart
 - Popularity vs. recommendation score scatter plot
 - Content vs. collaborative contribution stacked bar chart
 - User ratings distribution histogram
 - Top genres by user preference bar chart
 - User activity over time line chart
 - Model evaluation metrics visualization
3. **Additional Features:**
 - Downloadable recommendations as CSV
 - Model performance metrics display
 - Interactive data exploration capabilities

Evaluation Metrics

The system evaluates recommendation quality using:

- Root Mean Square Error (RMSE)

- Mean Absolute Error (MAE)
- Precision (for ratings ≥ 4)
- Recall (for ratings ≥ 4)
- F1-Score

Technical Implementation

The project is implemented in Python with these key libraries:

- **Streamlit**: Web application framework
- **Pandas**: Data manipulation
- **NumPy**: Numerical operations
- **Scikit-learn**: Feature extraction and similarity computation
- **Surprise**: Collaborative filtering algorithms
- **Plotly**: Interactive visualizations

Strengths of the System

1. **Hybrid Approach**: Combines the strengths of both recommendation paradigms
2. **Interactive UI**: Allows users to tune parameters and explore recommendations
3. **Rich Visualizations**: Provides insights into recommendation patterns
4. **Modular Design**: Facilitates extension and maintenance
5. **Performance Evaluation**: Includes metrics to assess recommendation quality

Potential Enhancements

Based on the codebase, some potential improvements could include:

1. Adding more features beyond genres (actors, directors, plot keywords)
2. Implementing more advanced algorithms (neural networks, matrix factorization)
3. Adding time-based recommendation decay
4. Incorporating explicit user feedback mechanisms
5. Adding A/B testing capabilities to compare recommendation strategies

Conclusion

This movie recommendation system demonstrates a comprehensive approach to personalizing movie suggestions using hybrid techniques. The interactive Streamlit interface makes the system accessible to users while providing transparency into the recommendation process through visualizations and metrics.