

Hybrid Movie Recommendation System Project

Objective:

Develop a hybrid recommendation system that leverages both user-item interactions (collaborative filtering) and item attributes (content-based filtering) to suggest movies tailored to individual user preferences.

Requirements

Functional Requirements:

1. Data Ingestion and Preprocessing:

- Load and preprocess the MovieLens 100K dataset, which includes user ratings, movie details (e.g., genres), and user information.
- Handle missing values and ensure data consistency.

2. Content-Based Filtering Module:

- Extract features from movie metadata (e.g., genres, titles).
- Compute similarity scores between movies using techniques like TF-IDF and cosine similarity.
- Generate recommendations based on item similarity.

3. Collaborative Filtering Module:

- Implement collaborative filtering using matrix factorization techniques such as Singular Value Decomposition (SVD).
- Predict user ratings for unseen movies based on user-item interaction patterns.

4. Hybrid Recommendation Engine:

- Combine the outputs of both content-based and collaborative filtering models.

- Implement strategies like weighted averaging or model stacking to integrate recommendations

5. **User Interface:**

- Develop an interactive interface (e.g., using Streamlit) where users can input their preferences and receive movie recommendations.

6. **Evaluation Metrics:**

- Assess the performance of the recommendation system using metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Precision, Recall, and F1-Score.
-

Deliverables

1. **Data Processing Scripts:**

- Scripts to load, clean, and preprocess the dataset.

2. **Model Implementation:**

- Code for content-based filtering using TF-IDF and cosine similarity.
- Code for collaborative filtering using SVD or other matrix factorization techniques

3. **Hybrid Model Integration:**

- Mechanism to combine both models' outputs into a unified recommendation list.

4. **User Interface:**

- A Streamlit application allowing users to receive recommendations based on their inputs.

5. **Evaluation Report:**

- Documentation detailing model performance, evaluation metrics, and insights.

6. **Deployment [Bonus]:**

- Deployment of the Streamlit application on a cloud platform for public access.
-

Dataset Overview

- Dataset: [MovieLens 100K](#)
-

Implementation Tips

- **Content-Based Filtering:**
 - Use TF-IDF vectorization on movie genres to represent each movie as a feature vector.
 - Compute cosine similarity between movies to find similar items.
 - **Collaborative Filtering:**
 - Utilize the Surprise library's SVD algorithm for matrix factorization.
 - Train the model on user-item rating data to predict unseen ratings.
 - **Hybrid Approach:**
 - Combine the normalized scores from both models using a weighted average.
 - Adjust weights based on validation performance to optimize recommendations.
 - **Evaluation:**
 - Split the dataset into training and testing sets (e.g., 80/20 split).
 - Evaluate each model separately and the hybrid model to compare performance.
-