Hybrid Recommender System Proposal

# Introduction

In this project, we propose a hybrid recommender system that integrates multiple recommendation approaches to provide personalized movie suggestions. The system leverages the strengths of different techniques to enhance accuracy and robustness. The approaches include item-based collaborative filtering, user-based collaborative filtering, content-based filtering, and matrix factorization.

# Item-Based Collaborative Filtering

Item-based collaborative filtering recommends items by analyzing similarities between items based on user ratings.

Pros:

* Captures item similarity effectively.
* Works well for users with sparse ratings.

Cons:

* Suffers from cold start problem for new items.
* Requires a large number of user ratings to be effective.

# User-Based Collaborative Filtering

User-based collaborative filtering finds similar users and recommends items they liked.

Pros:

* Simple and intuitive approach.
* Can provide diverse recommendations based on community tastes.

Cons:

* Not scalable for large datasets.
* Suffers from the cold start problem for new users.

# Content-Based Filtering

Content-based filtering recommends items based on the features of items that a user has previously liked. In this project, handcrafted features such as year, actors, writers, directors, country, and language were extracted using external APIs. One-hot encoding (OHE) was applied to categorical variables and MinMaxScaler was used to normalize numerical features.

Pros:

* Can provide recommendations for new users (no cold start issue).
* Personalizes recommendations to user preferences.

Cons:

* Limited to known features.
* Cannot exploit collaborative information from other users.

# Matrix Factorization

Matrix factorization techniques (such as SVD) decompose the user-item interaction matrix into lower-dimensional matrices to uncover latent factors representing users and items.

Pros:

* Handles sparse data well.
* Uncovers hidden features influencing user preferences.

Cons:

* Requires sufficient data for accurate factorization.
* Might lack interpretability of latent factors.

# Benefits of the Hybrid Approach

By combining these approaches, the system benefits from the strengths of each while compensating for their weaknesses. Collaborative filtering exploits community wisdom, content-based filtering captures personal preferences using item features, and matrix factorization models hidden relationships. The hybrid model enhances accuracy, diversity, and robustness of recommendations.

# Hybrid Recommender System Summary

The hybrid recommender system combines multiple recommendation techniques to leverage the strengths of each and compensate for their individual limitations. The table below outlines how each approach contributes to the overall performance of the hybrid model.

|  |  |  |
| --- | --- | --- |
| Method | Benefits | Limitations Addressed |
| Content-Based Filtering | Utilizes metadata such as actors, directors, writers, year, etc. Extracted features using APIs; normalized using One-Hot Encoding and MinMaxScaler. | Cold start problem for new users, limited diversity. |
| Item-Based Collaborative Filtering | Captures item similarity based on user ratings, good for identifying similar movies. | Sparse user-item matrix can affect similarity computation. |
| User-Based Collaborative Filtering | Identifies similar users to suggest unseen items, helpful in personalizing results. | Can struggle with scalability and data sparsity. |
| Matrix Factorization | Learns latent features from the rating matrix, handles sparse data efficiently. | Requires training and tuning, lacks interpretability. |

## Final Hybrid Scoring Equation

The final predicted rating for an unrated movie is computed using a weighted sum of the scores from each of the individual models. The weights are chosen to balance the influence of each component based on empirical performance or domain knowledge.

**Hybrid Score Equation:**

Hybrid\_Rating = **w1** \* ContentBased\_Rating + **w2** \* ItemBased\_Rating + **w3** \* UserBased\_Rating + **w4** \* MatrixFactorization\_Rating  
  
**Where:** - w1, w2, w3, w4 ∈ [0, 1] and w1 + w2 + w3 + w4 = 1  
 - Each rating is normalized to the same scale before combining