

Movie Recommender: Whitepaper

Introduction

The Movie Recommender is a web application designed to provide users with personalized movie suggestions. By analyzing user preferences, the application employs a hybrid approach of content-based and collaborative filtering to deliver relevant and engaging recommendations. This whitepaper outlines the methodology, features, and technical architecture of the Movie Recommender.

Methodology

The Movie Recommender combines content-based and collaborative filtering techniques to generate movie recommendations:

Content-Based Filtering: The system analyzes the intrinsic properties of movies, such as genres, to identify similarities. When a user likes a movie, the system identifies other movies with similar attributes.

Collaborative Filtering: The system also uses a simplified form of collaborative filtering by analyzing a user's "liked" movies. It doesn't compare users to each other, but it does use the ratings of liked movies in its scoring.

Hybrid Approach: The system combines both methods. Similarity between movies is calculated based on a weighted average of content-based similarity (70%) and the movie's rating (30%). This hybrid approach leverages the strengths of both methodologies, resulting in more accurate and personalized recommendations.

Features

Movie Browsing: Users can browse a list of available movies, filterable by genre. The application displays movie titles, release years, genres, and ratings.

Like/Dislike Functionality: Users can "like" movies, and these preferences are stored locally.

Personalized Recommendations: Based on the user's liked movies, the application generates a list of recommended movies, sorted by relevance. The recommendation algorithm calculates a "match" percentage, indicating how closely a movie aligns with the user's taste.

Responsive Design: The application is designed to be responsive and accessible across various devices.

Dynamic Content Loading: The application dynamically loads and displays movies, with a "Show More" button to load additional movies as needed.

Technical Architecture

Frontend:

HTML: Provides the structure of the web application.

CSS (via Tailwind CSS): Handles the styling and visual presentation of the user interface.

JavaScript: Implements the application's logic, interactivity, and recommendation algorithm. The Lodash library is used to simplify data manipulation.

Data:

The movie database is stored as a JavaScript array. Each movie object contains information such as title, year, genres, and rating.

Algorithm:

Genre Analysis: When a user likes a movie, the application analyzes the genres of that movie.

Similarity Scoring: The application calculates a similarity score for each available movie based on its genre similarity to the user's liked movies and its rating.

Recommendation Generation: The application sorts the movies by their similarity score and displays the top 5 as recommendations.

User Interface:

The user interface consists of two main sections: "Browse Movies" and "Likes & Recommendations."

The "Browse Movies" section displays a list of available movies, filterable by genre, with a "Like" button for each movie.

The "Likes & Recommendations" section displays the user's liked movies and a list of personalized recommendations.

Conclusion

The Movie Recommender provides a user-friendly and effective way to discover new movies. By combining content-based and collaborative filtering, the application delivers personalized recommendations that align with individual user preferences. The application's responsive design and dynamic content loading ensure a seamless and engaging user experience.