NextGen Movie Recommendation

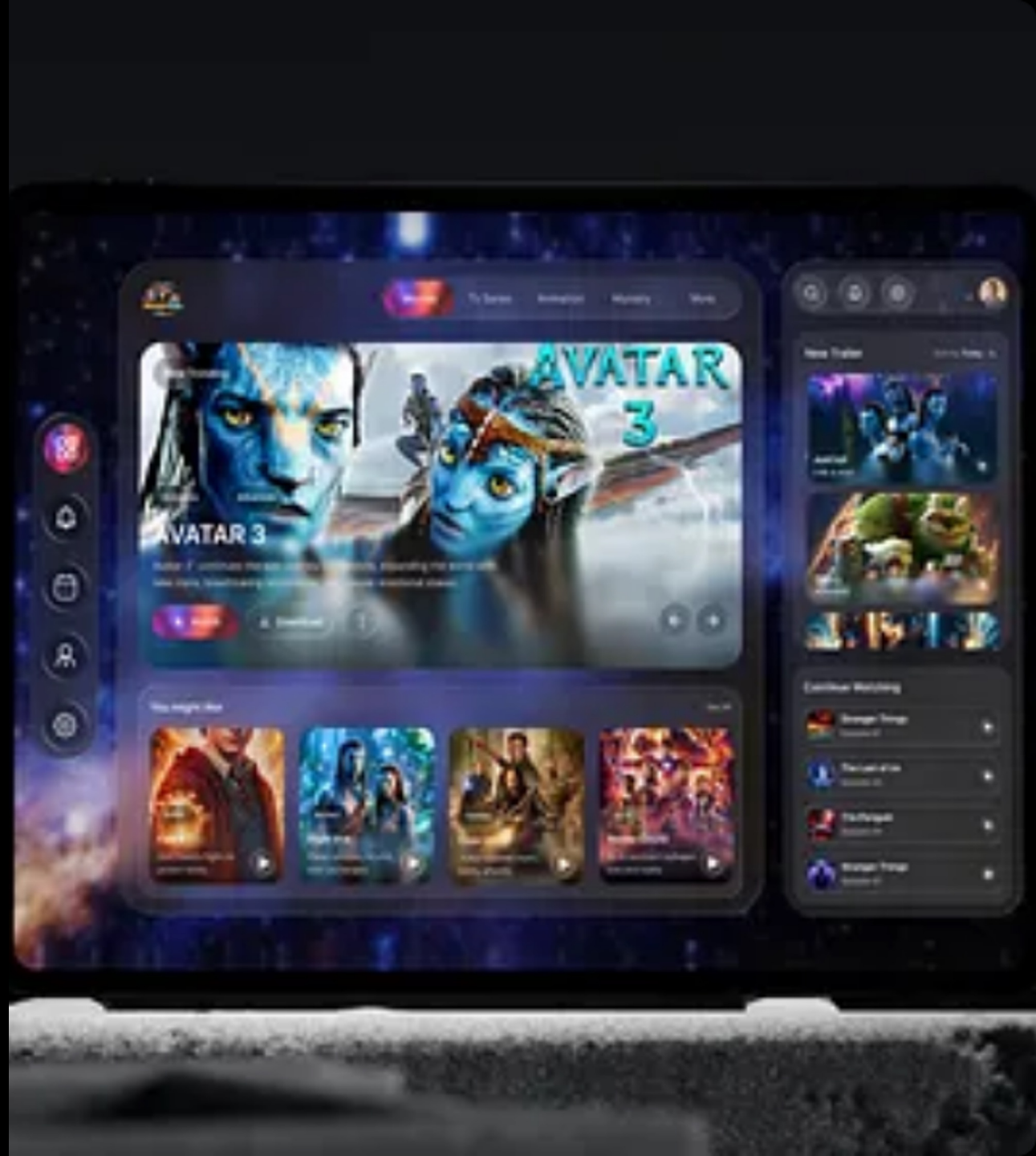
Discovering your next action

Movie Recommendation Platform

Personalized Movie Discovery

This system demonstrates a movie recommendation experience inspired by modern streaming platforms.

It suggests relevant movies using user preferences, viewing history, and similarity-based recommendation methods, while presenting the results through an intuitive and visual dashboard.



System Features & Goals

Personalization

Tailoring movie suggestions based on user preferences, viewing history, ratings, and interaction patterns.

Smart Search

A metadata-aware search component that matches user queries with movie titles, genres, keywords, and descriptive attributes.

Discovery

Encouraging exploration through diversity and serendipity mechanisms that expose users to relevant but less obvious recommendations.

Datasets & Integration

Data Source	Type	Key Attributes	Volume
MovieLens 25M	User-Item Interactions	Ratings, UserID, MovieID, Timestamps	400k+ users, 100k+ items, 25m+ events
TMDb Metadata	Movie Metadata	Genres, Cast, Crew, Keywords, Budget	500k+ Titles
IMDb Ratings	External Ratings	Average Score, Vote Count, Certification	300k+ entries

Data Preprocessing Pipeline

- Cleaning: Removing duplicate entries, handling missing metadata values, and aligning movie identifiers across datasets.
- Feature Vectorization: Converting textual and categorical metadata into numerical feature vectors for similarity computation.
- Rating Normalization: Applying mean normalization to reduce user-specific rating bias, such as users who consistently rate movies higher or lower than average.

System Architecture Overview

Full-Stack Integration

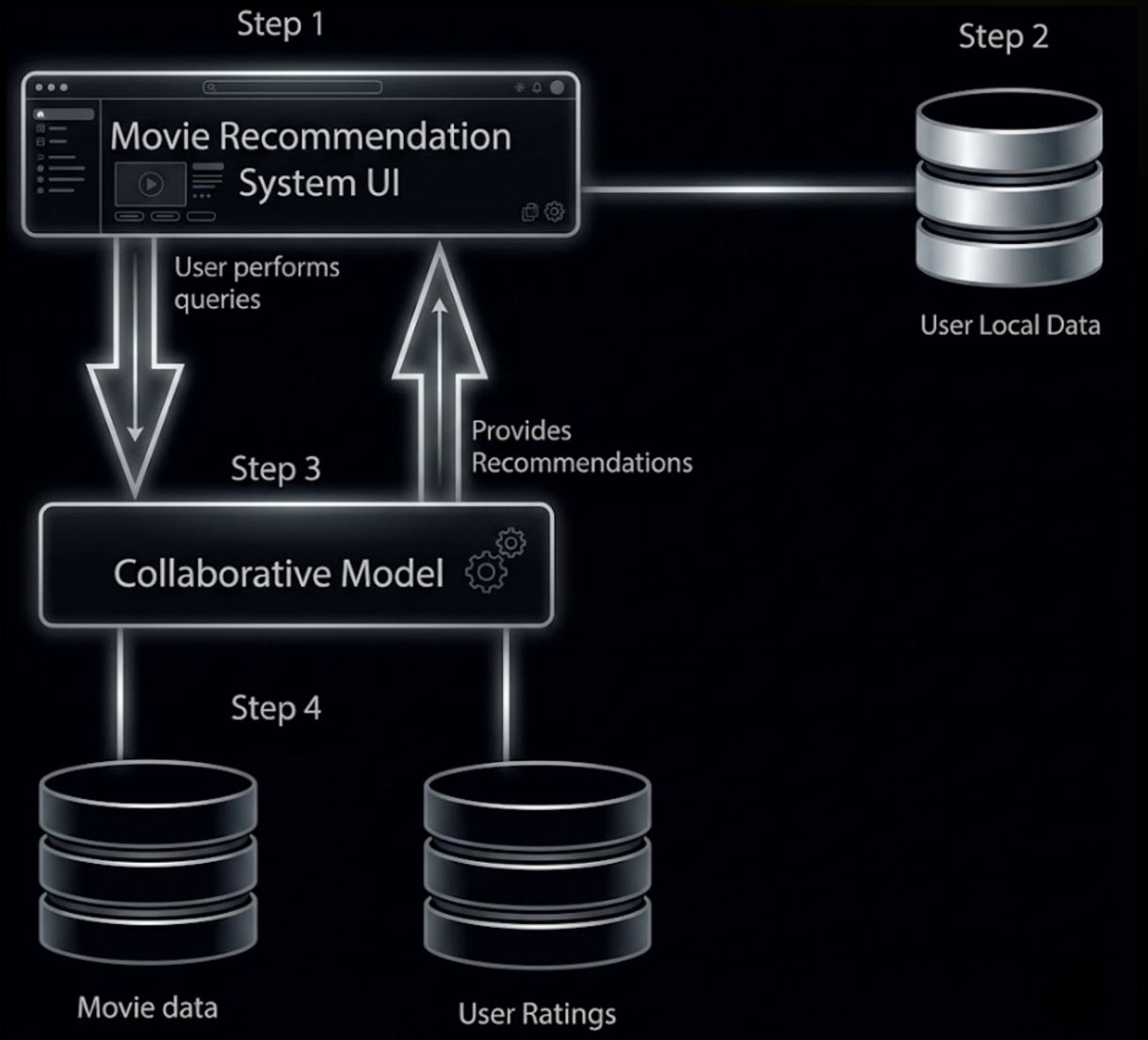
The system utilizes a **Client-Server** architecture designed for scalability:

Frontend: React/Streamlit for responsive UI and poster rendering.

Backend: Flask API serving as the orchestration layer.

Recommendation Engine: scikit -learn and PyTorch models computing similarity scores.

Database: PostgreSQL for user metadata and Redis for caching recommendations.



Algorithmic Strategies

Content-Based Filtering

- which algorithm was selected for each recommendation approach, and why

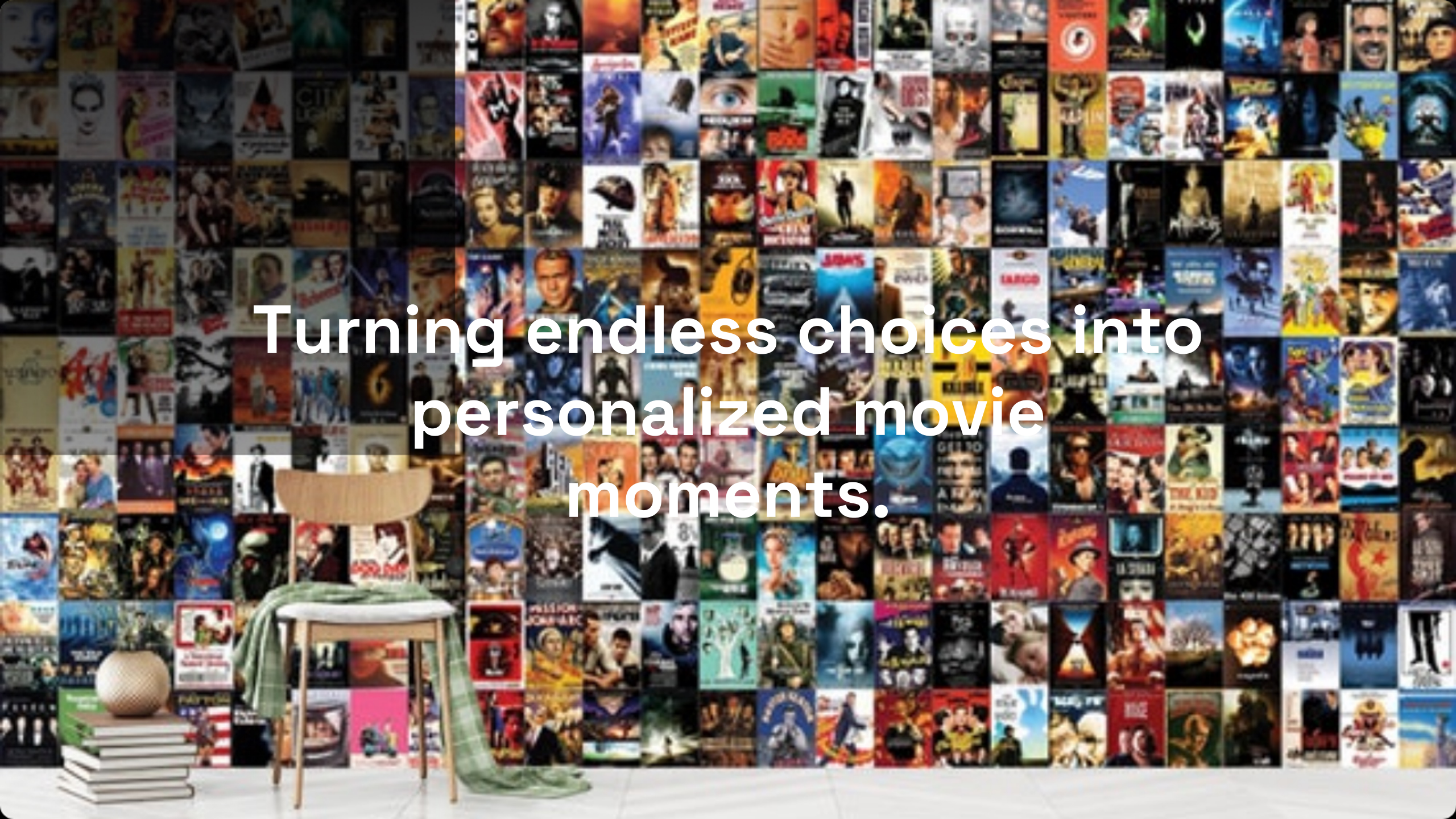
Collaborative Filtering

which algorithm was selected for each recommendation approach, and why

Roadmap till the final presentation

- [x] Implement feature 1
- [ ] Implement feature 2
- [ ] Add DB support
- [ ] Fix data entity matching

...


Turning endless choices into personalized movie moments.