Hackathon Submission (Level-1-Solution)

Use Case Title: AI-Powered Movie Recommendation System

Student Name: Pachaiyammal P

Register Number: 410723104057

Institution: Dhanalakshmi college of Engineering

Department: Computer Science and Engineering

Date of Submission: 18.05.2025

1.Problem Statement

- In today's digital age, users are overwhelmed with an abundance of content across various streaming platforms. With thousands of movies and TV shows available, it becomes increasingly difficult for viewers to find content that matches their preferences.
- This results in wasted time, poor user experience, and underutilized content libraries.
- There is a clear need for a more intelligent, personalized, and context-aware movie recommendation system that can deliver accurate and engaging suggestions.

2.Proposed Solution

- The proposed solution is an AI-powered movie recommendation system that leverages advanced machine learning techniques, including natural language processing (NLP) and collaborative filtering, to offer personalized movie suggestions.
- This system will analyze a combination of user data—such as watch history, ratings, reviews, and behavioral patterns—as well as movie metadata including genre, cast, director, plot summaries, and user-generated content like reviews and tags.

3.Technologies & Tools Considered

Programming Language: Python

Libraries: Pandas, NumPy, Scikit-learn, Requests

Frameworks: Streamlit (for UI), Flask (alternative option)

APIs: TMDB API (for posters and movie metadata)

Dataset: MovieLens Dataset

ML Techniques: Cosine Similarity, TF-IDF Vectorization, Matrix Factorization

4.Solution Architecture & Workflow

- **Data Collection:** Use the MovieLens dataset and TMDB API.

- **Preprocessing:** Clean and merge movie metadata (genres, overview, keywords, etc.)

- **Feature Extraction:** Apply TF-IDF or CountVectorizer for content analysis.

- **Similarity Calculation:** Use cosine similarity to find similar movies.

- **Recommendation Engine:** Given a movie title, fetch and display top 5–10 similar movies.

- **UI Layer:** Streamlit interface for search, display of results with zposters and details.

- **Deployment:** Host the app on Streamlit Cloud or Render.com.

5.Feasibility & Challenges

Feasibility:

- Building an AI-powered movie recommendation system is practical with today's technology.
- Readily available datasets (like IMDb and MovieLens), open-source tools (e.g., TensorFlow, Scikit-learn), and cloud platforms make development and deployment manageable.
- The system can also be integrated into existing apps or streaming services.

Challenges:

- **Cold Start Problem**: Difficult to recommend for new users or new movies with little data.

- **Privacy Concerns**: Handling user data must comply with laws like GDPR.
- **Scalability**: Must process large volumes of data quickly and efficiently.
- **Changing Preferences**: The system must adapt to users' evolving tastes over time.
- **Language/Culture Differences**: Recommending across different cultures adds complexity.

6.Expected Outcome & Impact

Expected Outcome:

- The system will deliver highly personalized movie recommendations to users, improving content discovery and user satisfaction.
- It will reduce time spent searching for movies and increase user engagement on streaming platforms.

Impact:

- **Users** will enjoy tailored content suggestions that match their interests.
- **Streaming platforms** will benefit from increased watch time, user retention, and content utilization.
- Overall, the system will create a more engaging, efficient, and satisfying viewing experience for all stakeholders.

7.Future Enhancements

- Add voice command support.
- Sync recommendations across devices.
- Use real-time feedback (e.g., mood-based suggestions).
- Include social features (suggest what friends are watching).
- Make recommendations explainable.
- Support multiple languages and cultures.