Phase-2

Github Link: https://github.com/Raguram2005/Ebpl-DS-Delivering-personalized-movie-recommendations-with-an-AI-driven-matchmaking-system.git

Phase-2 Project Overview

Student Name: RAGURAM.R

Register Number: 511523205041

Institution: P.T.Lee Chengalvaraya Naicker College of Engineering &Technology

Department: INFORMATION TECHNOLOGY

Date of Submission: 07-05-2025

Project Title: Delivering Personalized Movie Recommendations with an AI-driven Matchmaking System

1. Problem Statement

In today's digital entertainment world, users often face difficulty finding movies they would truly enjoy due to the overwhelming volume of content available. Our project focuses on building a personalized movie recommendation system that leverages AI-driven matchmaking techniques to suggest movies tailored to individual tastes. By analyzing user preferences, historical viewing patterns, and movie metadata, the system aims to deliver highly accurate recommendations, enhancing user satisfaction and engagement.

2. Project Objectives

● Develop a recommendation engine that suggests movies personalized to user preferences.
● Utilize collaborative filtering, content-based filtering, and hybrid methods.
● Ensure scalability and adaptability with dynamic user profiles.
● Provide a user-friendly interface for interaction and feedback collection.
● Continuously refine recommendations using user feedback and behavior.

3. Flowchart of the Project Workflow

(Flowchart Image to be inserted manually)

4. Data Description

● Dataset Name: MovieLens Dataset (or custom dataset from online sources)
● Source: GroupLens Research, IMDb metadata
● Type of Data: Structured tabular data and text data
● Records and Features: Movie titles, genres, ratings, user preferences
● Static or Dynamic: Static dataset with periodic updates
● Attributes Covered: Title, genre, director, cast, user ratings, timestamps

5. Data Preprocessing

● Removed missing or null entries.
● Merged movie metadata with ratings data.
● Encoded categorical features such as genre and tags.
● Normalized user rating scales.
● Generated user-item interaction matrices.

6. Exploratory Data Analysis (EDA)

● Analyzed rating distribution among users.
● Identified most popular movies and genres.

- Examined sparsity of the user-item matrix.
- Studied genre-based user preferences.

7. Feature Engineering

- Created genre similarity scores using TF-IDF.
- Built user profiles based on historical ratings.
- Computed movie-to-movie similarity matrices.
- Introduced timestamp-based weighting to favor recent ratings.

8. Model Building

- Algorithms Used:
  - Collaborative Filtering (User-User and Item-Item)
  - Content-Based Filtering
  - Hybrid Approach (weighted combination)

- Model Selection Rationale:
  - Collaborative Filtering for leveraging user patterns
  - Content-Based for handling cold-start users
  - Hybrid model for better accuracy

- Evaluation Metrics:
  - Precision@K
  - Recall@K
  - F1-Score
  - RMSE for rating prediction

9. Visualization of Results & Model Insights

- Precision-Recall curves for different models.
- User satisfaction based on feedback.
- Genre-wise recommendation effectiveness.
- Word clouds of recommended movie tags.

10. Tools and Technologies Used

- Programming Language: Python 3
- Notebook Environment: Google Colab / Jupyter Notebook
- Key Libraries:
  - pandas, numpy for data handling
  - scikit-learn for machine learning models
  - surprise for recommendation models
  - matplotlib, seaborn for visualizations
  - streamlit/gradio for user interface

11. Team Members and Contributions

- **RAGURAM.R** - Data collection, preprocessing, and EDA.
- **PRIYADHASHAN.A** - Feature engineering and model building.
- **UGENDRAN.R** - Evaluation, visualization.
- **SANTHASEELAN.R** - visualization, and documentation.