

Phase-2 Submission By

Student Name: VIGNESH V

Register Number: 622423121057

Institution: Salem College of Engineering and Technology

Department: Biomedical Engineering

Date of Submission: 26-04-2025

GitHub Repository: <https://github.com/VIGNESH1562006/Delivering-Personalized-movie-recommendations-with-an-AI-driven-matchmaking-system.git>

TOPIC: Delivering Personalized movie recommendations with an AI-driven matchmaking system

1.Problem Statement

*Current movie recommendation systems primarily rely on user history or general popularity trends. They often miss nuanced factors like mood, preferred story arcs, niche genres, or evolving tastes over time. A more dynamic, "matchmaking" approach could bridge this gap and deliver recommendations that feel more **personal and relevant**.*

2.Objectives of the Project

- ☐ **Develop** an AI model that **learns user preferences** beyond simple genres or ratings.

- **Design a matchmaking engine** that pairs users with movies based on detailed profiling (e.g., emotional tone, themes, pacing).
- **Continuously update recommendations** as user interactions evolve.
- **Provide an engaging user interface** that makes discovery enjoyable.

3.Scope of the Project

In-Scope:

- Building a **hybrid AI model** that combines content-based and collaborative filtering for personalized recommendations.
- Creating **detailed user profiles** based on explicit (ratings, preferences) and implicit (watch time, likes) feedback.
- Integrating a **matchmaking algorithm** that finds the best movie-user matches based on dynamic profiling.
- Developing a **prototype web interface** (optional) for users to interact with the system and receive recommendations.
- Using **public movie datasets** like **MovieLens** or TMDB API for data.

Out-of-Scope:

- Building a full production-ready streaming platform (e.g., like Netflix).
- Collecting private, real-world user data beyond public datasets (for privacy reasons).
- Creating an extensive mobile app — the focus will be on a simple prototype web version if needed.

4.Data Sources

1. MovieLens Dataset

- **Description:**

One of the most popular datasets for movie recommendation research. Contains millions of movie ratings, user IDs, movie IDs, timestamps, and tags (like genres, moods, etc.).

- **Link:**
<https://grouplens.org/datasets/movielens/>
 - **Why use it?**
 - Rich user-movie interaction data.
 - Available in multiple sizes (small: 100K ratings, medium: 1M ratings, large: 20M+ ratings).
-

2. The Movie Database (TMDB) API

- **Description:**
TMDB provides detailed movie metadata — such as genres, posters, descriptions, release dates, cast, crew, and reviews — via a free API.
 - **Link:** <https://developer.themoviedb.org/docs>
 - **Why use it?**
 - To **enrich** your movie data with more attributes (e.g., director, popularity score, language, poster images).
 - Helps in building a better **content-based** recommendation engine.
-

3. IMDb Datasets (via IMDb Official or IMDbPY Library)

- **Description:**
IMDb offers public downloadable datasets, including title basics, ratings, genres, actors, and crew.
 - **Link:**
<https://datasets.imdbws.com/>
 - **Why use it?**
 - Massive database of global movies and TV shows.
 - Good for cross-referencing additional movie features.
-

4. Kaggle Movie Datasets □ Examples:

- *Netflix Movies and TV Shows Dataset*
- *IMDb 5000 Movie Dataset*
- *Movies Metadata from TMDb*
- **Link:**
<https://www.kaggle.com/datasets> (Search "Movies" or "Movie Recommendations")
- **Why use it?**
 - *Quickly get curated datasets for faster prototyping.*
 - *Some Kaggle datasets are already cleaned and combined.*

5.High-Level Methodology

- ☐ **Data Collection:** Use open-source datasets (e.g., *MovieLens*) containing user ratings, genres, tags, and movie metadata.
- ☐ **User Profiling:** Create detailed user profiles based on past interactions, preferences, and implicit feedback.
- ☐ **Content-Based + Collaborative Filtering:** Use a hybrid recommendation approach.
- ☐ **Deep Learning:** Train a neural network model (like an *Autoencoder* or *DeepFM*) to predict user-movie matches.
- ☐ **Feedback Loop:** Continuously refine recommendations based on real-time feedback (like/dislike, watch time).

6.Tools and Technologies ☐

Languages: Python

- ☐ **Libraries:** TensorFlow / PyTorch, Scikit-Learn, Surprise (for recommender systems)
- ☐ **Database:** PostgreSQL or MongoDB for storing profiles and movies
- ☐ **Frontend (Optional):** ReactJS or simple Flask web app
- ☐ **APIs:** TMDB (The Movie Database) API for movie metadata and posters

7.Team Members and Roles

Name	Role	Key Responsibilities
Vignesh V	Project Manager + Data Scientist	Manage team, design AI models
Vinisha M	Machine Learning Engineer	Build and optimize recommendation engine
Vishalini S	Frontend Developer/Backend Developer	Create a simple web interface for users, Handledatabase, APIintegration,serversetup