**Phase-2 Submission**

**Student Name:** [A. Vetri vendhan ]

**Register Number:** [422223106039]

**Institution:** [surya group of institutions ]

**Department:** [BE-ECE]

**Date of Submission:** [08/05/2025]

**Github Repository Link:** [https://github.com/vetrivendhan007-vendhan/Delivering-personalized-an-AI-Driven-Matchmaking-System.git]

### **1. Problem Statement**

### *In the digital age of entertainment, users are often overwhelmed by the vast number of available movies. Traditional recommendation systems often fall short in capturing nuanced user preferences. This project aims to build an AI-driven matchmaking system that delivers personalized movie recommendations based on user behavior, preferences, and similarity with other users or items. The problem is framed as a recommendation system (using collaborative filtering, content-based filtering, or hybrid methods). It aims to enhance user experience by increasing engagement, satisfaction, and time-on-platform.*

### **2. Project Objectives**

To develop a recommendation engine that accurately suggests movies to users.

To analyze user-item interactions and build collaborative and content-based models.

To incorporate hybrid approaches that combine multiple recommendation strategies.

To optimize recommendation quality using evaluation metrics like RMSE and Precision@K.

To create a scalable system that can work with large datasets and real-time updates.

---

### **3. Flowchart of the Project**

**1. Data Collection (User ratings, movie metadata)**

**2. Data Cleaning & Preprocessing**

**3. Exploratory Data Analysis (EDA)**

**4. Feature Engineering (Genres, tags, ratings)**

**5. Building Recommendation Models (Collaborative, Content-Based, Hybrid)**

**6. Evaluation & Tuning**

**7. Deployment-ready output & Visualization**

**(I can create a visual version of this upon request.)**

**--**

### **4. Data Description**

*Dataset: MovieLens 100k*

*Source: GroupLens Research*

*Type: Structured data (user ratings, movie metadata)*

*Size: ~100,000 ratings from 943 users on 1,682 movies*

*Features: User ID, Movie ID, Rating, Timestamp, Genres*

*Target: Recommend top-N movies per user*

*---*

### **5. Data Preprocessing**

**Removed duplicate ratings**

**Converted timestamps to readable formats (year, month)**

**Merged movie metadata with user ratings**

**Encoded genres using multi-hot encoding**

**Normalized user rating distributions**

**Constructed user-item matrix for collaborative filtering**

### **6. Exploratory Data Analysis (EDA)**

*Analyzed rating distribution and user behavior*

*Identified popular and highly-rated movies*

*Detected trends in genres over time*

*Visualized user activity and movie frequency using barplots, heatmaps, and line graphs*

*Found genre-specific preferences among users*

### **7. Feature Engineering**

*Created genre vectors for content-based filtering*

*Calculated movie similarity scores using cosine similarity*

*Generated user profiles based on genre affinity*

*Developed hybrid features combining content and rating-based insights*

*Considered metadata like year of release, popularity*

*---*

### **8. Model Building**

*Collaborative Filtering (User-Based and Item-Based)*

*Matrix Factorization (SVD) using Surprise library*

*Content-Based Filtering using cosine similarity*

*Hybrid Recommender combining collaborative and content-based outputs*

*Evaluation Metrics: RMSE, MAE, Precision@K, Recall@K*

*SVD and Hybrid models outperformed others in terms of accuracy and personalization*

*---*

### **9. Visualization of Results & Model Insights**

*RMSE and MAE comparison bar plots*

*Precision/Recall curves*

*Heatmaps showing user-item prediction matrices*

*Top-N recommendations per user*

*Insights: Hybrid models offer balanced personalization and diversity*

*---*

### **10. Tools and Technologies Used**

*Programming: Python*

*IDE: Jupyter Notebook*

*Libraries: pandas, numpy, scikit-learn, Surprise, matplotlib, seaborn*

*Visualization: seaborn, matplotlib, Plotly*

*Recommendation Techniques: Cosine Similarity, SVD, Hybrid Fusion*

*---*

### **11. Team Members and Contributions**

[A. Vetri vendhan]: Data Preparation, Content-Based Filtering

[M. Thamizharasan]: EDA, Collaborative Filtering

[P. Jagankumar]: Model Tuning, Evaluation

[S. Mohanakrishnan]: Visualization, Report Writing

---