**Phase-2 Submission**

**Student Name:** Navaneeth S  
**Register Number:** 422623106051  
**Institution:** University College of Engineering, Panruti  
**Department:** Electronic and Communication Engineering  
**Date of Submission:** 27.04.2025  
**Title:** Delivering personalized movie recommendation with an AI driven matchmaking system

**GITHUB Repository Link:** https://github.com/Navaneeth2608/oracle-nm-datascience

**1. Problem Statement**

In today's digital streaming era, users face overwhelming choices when selecting content. Traditional recommendation systems either suggest obvious choices or lack a social discovery aspect.

This project focuses on building a hybrid recommendation and user matchmaking system by combining clustering and recommendation techniques. It recommends movies based on both individual preferences and social similarity to other users.

Problem Type: Clustering + Recommendation (Hybrid System)  
Impact: Enhances personalized content experience and fosters social discovery

**2. Project Objectives**

* Create a hybrid movie recommendation system using content-based filtering, collaborative filtering, and deep learning
* Develop dynamic user profiles capturing preferences, viewing patterns, and emotional reactions
* Implement an AI-driven matchmaking system connecting users with similar preferences
* Build a real-time feedback loop to adapt and refine recommendations
* Deliver an engaging and socially enriched movie discovery platform

**3. Flowchart of the Project Workflow**

**┌────────────────────────────────────────────────────┐**

**│ Start: Project Initialization │**

**└────────────────────────────────────────────────────┘**

**↓**

**┌────────────────────────────────────────────────────┐**

**│ Data Collection │**

**│ (MovieLens Dataset + Synthetic User Profiles) │**

**└────────────────────────────────────────────────────┘**

**↓**

**┌────────────────────────────────────────────────────┐**

**│ Data Cleaning and Preprocessing │**

**│ (Handling Missing Values, Encoding, Normalization) │**

**└────────────────────────────────────────────────────┘**

**↓**

**┌────────────────────────────────────────────────────┐**

**│ Exploratory Data Analysis (EDA) and Feature Engineering │**

**│ (User Preferences, Movie Vectors, Similarity Matrices) │**

**└────────────────────────────────────────────────────┘**

**↓**

**┌────────────────────────────────────────────────────┐**

**│ Model Development │**

**│ - Content-Based Filtering │**

**│ - Collaborative Filtering (Matrix Factorization) │**

**│ - Deep Learning Models │**

**└────────────────────────────────────────────────────┘**

**↓**

**┌────────────────────────────────────────────────────┐**

**│ User Matchmaking │**

**│ (User Clustering using K-Means / Cosine Similarity) │**

**└────────────────────────────────────────────────────┘**

**↓**

**┌────────────────────────────────────────────────────┐**

**│ Model Evaluation │**

**│ (RMSE, Precision@k, Recall@k, Silhouette Score) │**

**└────────────────────────────────────────────────────┘**

**↓**

**┌────────────────────────────────────────────────────┐**

**│ Deployment │**

**│ (Command-Line Interface Application) │**

**└────────────────────────────────────────────────────┘**

**↓**

**┌────────────────────────────────────────────────────┐**

**│ End │**

**└────────────────────────────────────────────────────┘**

**4. Data Description**

Dataset Name and Source:

* Movies: Kaggle - MovieLens Dataset
* Users: Synthetic User Data (Self-Generated)

Type of Data: Structured (Tabular)

Number of Records and Features:

* Movies: ~10,000 movies
* Users: ~500 synthetic users

Dataset Nature: Static (locally used)

Target Variables:

* For Recommendation: Predict movie ratings
* For Matchmaking: Form user clusters (unsupervised)

**5. Data Preprocessing**

* Handled missing values by filling missing directors/genres with "Unknown"
* Removed duplicate movie entries
* Outliers were not critical; ratings were already bounded between 0 and 5
* Standardized genre names using lowercase formatting
* Encoded genres using multi-hot encoding
* Applied Min-Max scaling on numerical features where necessary

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

Univariate Analysis:

* Plotted histograms for ratings distribution
* Bar plots for most popular genres

Bivariate/Multivariate Analysis:

* Heatmaps to visualize user similarity
* Scatterplots showing the relationship between watch time and rating

Insights Summary:

* Action and Drama were the most frequently watched genres
* Higher-rated movies were generally watched for longer durations

**7. Feature Engineering**

* Constructed user preference vectors based on genre affinity, watch time, and emotional feedback
* Created movie feature vectors based on genre, cast, and keywords
* Built user similarity matrices using cosine similarity
* Applied K-Means clustering to group users into similar segments

Justification:  
These engineered features directly enhance the recommendation and matchmaking performance

**8. Model Building**

Models Implemented:

1. Content-Based Filtering: Recommends similar movies to highly-rated ones
2. Collaborative Filtering (Matrix Factorization): Recommends based on user-user and item-item similarity
3. Deep Neural Networks: Captures complex non-linear user behavior patterns

Data Split: 80% training and 20% testing

Evaluation Metrics:

* RMSE for rating prediction accuracy
* Precision@k, Recall@k for recommendation quality
* Silhouette Score for clustering quality

**9. Visualization of Results and Model Insights**

Recommendation System Evaluation:

* RMSE graphs comparing different models
* Precision and Recall curves

Matchmaking Evaluation:

* Silhouette scores visualized
* 2D PCA or t-SNE projections of clustered users

Insights:

* Collaborative Filtering performed better for sparse users
* Deep Learning models significantly improved recommendations for new (cold start) users

**10. Tools and Technologies Used**

Programming Language: Python

IDE/Notebook: Google Colab, Jupyter Notebook, VS Code

Libraries:

* Data Processing: pandas, numpy
* Visualization: matplotlib, seaborn, plotly
* Machine Learning: scikit-learn, Surprise, TensorFlow/Keras
* Natural Language Processing (optional): nltk, spaCy

Deployment Platform: Command-Line Interface (CLI) (Future expansion to Streamlit or Flask planned)

**11. Team Members and Contributions**

* **Balasubramani G**  
  • Deep Learning Implementation  
  • Deployment (CLI Application)  
  • Documentation and Reporting
* **Harikrishnan A**   
  • Data Cleaning  
  • Data Preprocessing
* **Aruvimani M**  
  • Exploratory Data Analysis (EDA)  
  • Visualization of Insights

• User Profile Creation

* **Navaneeth S**  
  • Model Development (Recommendation Systems and Matchmaking)  
  • Model Evaluation

• Feature Engineering