**PHASE-1**

**Student Name:** Navaneeth S  
**Register Number:** 422623106051  
**Institution:** University College of Engineering, Panruti  
**Department:** Electronic and Communication Engineering  
**Date of Submission:** 26.04.2025

**1. Problem Statement**

* In the era of digital streaming, the vast amount of content overwhelms users, making it difficult to choose the right movie.
* Traditional recommendation systems often fail to deliver highly personalized and socially interactive experiences.
* This project proposes an AI-driven system that recommends movies based on user preferences and matches users with similar tastes, enhancing the social dimension of movie discovery.

**2. Objectives of the Project**

* Develop a **hybrid movie recommendation system** combining collaborative filtering, content-based filtering, and deep learning.
* Build a **user profiling module** to analyze viewing history, genre preferences, emotional reactions, and behaviors.
* Implement an **AI-powered matchmaking system** to link users with psychological and behavioral similarity.
* Provide **explainable recommendations** that adapt based on real-time user feedback.
* Enhance **interactive and social movie discovery** through intelligent user matching.

**3. Scope of the Project**

**Features to Implement:**

* Hybrid recommendation system (content-based + collaborative filtering + deep learning)
* Dynamic user profiling
* User-matching algorithm based on similarity
* Command-Line Interface (CLI) application (expandable to GUI/Web later)
* Real-time feedback loop for continuous improvement

**Limitations:**

* Initial prototype as CLI (not full web app).
* Small dataset for prototyping (no large-scale deployment).
* Focus on movies only (no TV shows, music, books).
* No use of user demographic details (age, location, etc.).

**4. Data Resources**

* **Movie Data Source:** Kaggle MovieLens Dataset
  + Features: Movie metadata (genres, actors, directors, tags, ratings)
* **User Data Source:** Synthetic user profiles
  + Features: Viewing history, ratings, genre preferences, behavioral patterns
* **Dataset Nature:** Static (downloaded once and used locally)

**5. High-Level Methodology**

**Data Collection**

* Fetch movie data from Kaggle MovieLens dataset.
* Generate synthetic user profiles with logical viewing patterns.

**Data Cleaning**

* Handle missing values (like missing director names).
* Normalize text fields (e.g., lowercase genres).
* Remove duplicates.

**Exploratory Data Analysis (EDA)**

* Analyze most-watched genres.
* Study relationship between user ratings and genres.
* Explore viewing habits via user clustering.

**Feature Engineering**

* Create user preference vectors and movie feature vectors.
* Build user similarity matrices for matchmaking.

**Model Building**

* Content-Based Filtering for initial recommendations.
* Collaborative Filtering (Matrix Factorization) for user-based recommendations.
* Deep Learning models (Neural Networks) for non-linear user behavior.
* Clustering (K-means) to group users with similar preferences.

**Model Evaluation**

* RMSE for rating prediction accuracy.
* Precision@k and Recall@k for recommendation quality.
* Silhouette Score for clustering evaluation.

**Visualization & Interpretation**

* Precision-Recall curves.
* 2D visualizations using PCA/t-SNE for clustering.
* Flow diagrams showing user interactions.

**Deployment**

* Initially deploy as a CLI application.
* Future possibility: Upgrade to Streamlit or Flask-based web app.

**6. Tools and Technologies**

**Programming Language**

* Python

**Notebook/IDE**

* Google Colab,
* Jupyter Notebook,
* VS Code

**Libraries**

* Data Processing: pandas,numpy
* Visualization: matplotlib, seaborn, plotly
* Machine Learning: scikit-learn, surprise, TensorFlow/Keras
* (Optional) NLP: nltk, spaCy

**Optional Deployment Tools**

* Streamlit (Web UI)
* Flask/FastAPI (full-stack app)
* Docker (for advanced deployment)

**7. Results & Visualization (Future Phase)**

* Graphs showing recommendation accuracy.
* Visual clustering of users.
* Example recommendation lists per user.

**8. Conclusion & Future Work**

* Proposed a hybrid AI-based movie recommendation system.
* Enhanced recommendation quality through dynamic learning and user matchmaking.
* Future extensions:
  + Expand to real-time feedback learning.
  + Develop a full GUI/Web-based platform.
  + Integrate with larger streaming datasets.

**9. Team Members and Roles**

| **Team Member** | **Responsibilities** |
| --- | --- |
| **Aruvimani** | Design CLI interactions, test features |
| **Harikrishnan** | Design system architecture, integrate modules |
| **Balasubramani** | Manage timelines, document work, team communication |
| **Navaneeth** | Build AI models, develop matching algorithm, Data cleaning,Feature engineering, Data consistency |