**Phase-2 Submission Template**

**Student Name:** [Enter Your Name]

**Register Number:** [Enter Your Register Number]

**Institution:** [Insert College Name]

**Department:** [Enter Your Department Name]

**Date of Submission:** [Insert Date]

**Github Repository Link:** [Update the project source code to your Github Repository]

### **1. Problem Statement**

### *In today’s digital age, users face overwhelming choices when selecting movies. Traditional recommendation systems based on popularity or genre often fall short in providing personalized experiences. Our project addresses this issue by leveraging AI-driven matchmaking to offer tailored movie suggestions based on user preferences, historical ratings, and behavioral data.*

### ***Problem Type:*** *Classification + Clustering*

### *This problem blends classification (predicting user preferences) and clustering (grouping similar users/movies).*

### ***Relevance:*** *Personalized recommendations enhance user satisfaction, engagement, and platform retention, making this system highly relevant for OTT platforms and streaming services.*

### 

### **2. Project Objectives**

### *Develop a machine learning system that delivers highly personalized movie recommendations.*

### *Incorporate collaborative filtering and content-based filtering into a hybrid approach.*

### *Improve accuracy and relevance using clustering (e.g., K-Means) and deep learning models (e.g., Neural Networks).*

### *Optimize system performance using evaluation metrics like Precision, Recall, and F1-score.*

### *Adapt the recommendation based on real-time behavior and feedback.*

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

### *Data Collection → Data Cleaning → EDA → Feature Engineering → Model Selection →*

### *Training & Validation → Model Evaluation → AI Matchmaking Algorithm → Final Recommendations*

### 

### **4. Data Description**

### ***Dataset:*** *MovieLens 1M Dataset (from GroupLens)*

### ***Data Type:*** *Structured*

### ***Records:*** *1,000,209 ratings from 6,000 users on 4,000 movies*

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

### ***Target Variable:*** *Rating (for collaborative filtering) or Like/Dislike (for classification)*

### **5. Data Preprocessing**

* Handled missing values by imputing user demographics with mode or removing irrelevant rows.*

* Removed duplicate user-movie ratings.*

* Handled timestamp conversion to extract time-related features.*

* Used One-Hot Encoding for genre and label encoding for categorical variables.*

* Scaled numerical features using MinMaxScaler.*

* Created user-movie interaction matrix for collaborative filtering*

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

* *Analyzed rating distribution (most users rate movies between 3 and 4).*
* *Identified most/least rated movies.*
* *Heatmap of correlations between user behavior and genres.*
* *Observed clustering of user preferences around genre and time.*

***Insights:***

* *Users tend to be loyal to certain genres.*
* *Time of rating impacts rating behavior (e.g., weekend vs. weekday).*
* *Few users account for a majority of the ratings.*

### 

### **7. Feature Engineering**

* *Added “Rating Timestamp (hour, weekday)” as time-based features.*
* *Created binary target feature: "Liked" (rating ≥ 4).*
* *Created user genre preference profiles.*
* *Applied PCA for dimensionality reduction of user-movie matrix.*

### 

### **8. Model Building**

### ***Models Used:***

### ***Model 1:*** *K-Nearest Neighbors (Collaborative Filtering)*

### ***Model 2:*** *XGBoost Classifier (Content-Based Filtering)*

### ***Train-Test Split:*** *80/20 stratified*

### ***Evaluation Metrics:***

### *Accuracy: 0.86 (XGBoost), 0.82 (KNN)*

### *F1-Score: 0.83*

### *Precision: 0.81*

### *Recall: 0.85*

### ***Hybrid Approach:*** *Combined both models with weighted average prediction for final recommendation score.*

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

** ***Confusion Matrix:*** *Showed strong performance in predicting “Liked” category.*

** ***Feature Importance (XGBoost):*** *Genre, age, and user activity were top predictors.*

** ***Clustering Plot (K-Means):*** *Grouped users with similar tastes.*

** ***Recommendation Example:*** *Visual list of top-5 recommendations per user with explanation.*

### 

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

** ***Language:*** *Python*

** ***IDE:*** *Google Colab*

** ***Libraries:*** *pandas, numpy, matplotlib, seaborn, scikit-learn, surprise, XGBoost, TensorFlow (optional)*

** ***Visualization:*** *seaborn, matplotlib, Plotly*

### 

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

|  |  |
| --- | --- |
| ***Name*** | ***Responsibility*** |
| *[Your Name]* | *Data Cleaning, Feature Engineering, Reporting* |
| *[Teammate 1 Name]* | *EDA, Model Training* |
| *[Teammate 2 Name]* | *Model Evaluation, Visualization* |
| *[Teammate 3 Name]* | *Documentation, Presentation Prep* |