**Phase-2 Submission Template**

**Student Name:** [Shivam Keshari]

**Register Number:** [410823104030]

**Institution:** [GKMCET]

**Department:** [BE CSE]

**Date of Submission:** [14-05-2025]

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

### **1. Problem Statement**

### **1.Low user engagement**: Users may not find relevant movies, leading to decreased viewing time and satisfaction.

### **2.Information overload**: Users are overwhelmed by the vast number of movie options, making it difficult to discover new movies.

### **3.Poor recommendation accuracy**: Recommendations may not align with user preferences, leading to frustration and disappointment.

### **2. Project Objectives**

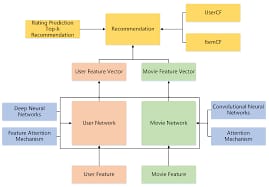
**1.Develop an AI-driven match making system**: Create a system that uses machine learning algorithms to match users with movies based on their profiles and preferences.

**2.Improve recommendation accuracy**: Increase the accuracy of movie recommendations by leveraging user data and movie features.

**3.Enhance user experience**: Provide users with personalized movie recommendations that increase engagement and satisfaction..

### 

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



### **4. Data Description**

### **1.User ID**: Unique identifier for each user.

### **2.Demographics**: Age, gender, location, etc.

### **3.Viewing history**: List of movies watched by the user.

### **4.Ratings**: User-provided ratings for movies (e.g., 1-5 stars).

### **5.Search queries**: Search queries entered by the user.

### **Movie Data**

### **1.Movie ID**: Unique identifier for each movie.

### **2.Title**: Movie title.

### **3.Genre**: Movie genre (e.g., action, comedy, drama).

### **4.Director**: Movie director.

### **5. Data Preprocessing**

### **Data Preprocessing**

### **1.Data cleaning**: Remove missing or duplicate data.

### 2.Data normalization: Scale numerical features to a common range.

### **3.Data transformation**: Convert categorical features into numerical features.

### **Feature Engineering**

### **1.Movie feature extraction**: Extract relevant features from movie metadata (e.g., genre, director, cast).

### **2.User feature extraction**: Extract relevant features from user data (e.g., viewing history, ratings).

### **3.Creating interaction matrix**: Create a matrix of user-item interactions (e.g., ratings, watches).

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

### Goals

### **1.Understand data distribution:** Examine the distribution of variables in the dataset.

### **2.Identify patterns and relationships:** Discover relationships between variables and identify patterns in the data.

### **3.Detect anomalies:** Identify outliers or anomalies in the data.

### Techniques

### **1.Summary statistics:** Calculate summary statistics (e.g., mean, median, standard deviation) for numerical variables.

### **2.Data visualization**: Use plots (e.g., histograms, scatter plots) to visualize the data.

### **7. Feature Engineering**

### **Potential Directions**

### **1.Multimodal recommendations:** Incorporate multiple data sources (e.g., text, images, audio) to improve recommendation accuracy.

### **2.Explainable recommendations**: Develop techniques to provide transparency into recommendation decisions.

### **3.Real-time recommendations:** Generate recommendations in real-time based on user behavior and preferences**.**

### **4.Personalized explanations:** Provide personalized explanations for recommendations based on user preferences and behavior**.**

### **5.Context-aware recommendations**: Incorporate contextual information (e.g., location, time of day) to improve recommendation accuracy**.**

### **8. Model Building**

### **Model Types**

### **1.Collaborative Filtering (CF)**: Recommend movies based on the behavior of similar users.

### **2.Content-Based Filtering (CBF)**: Recommend movies based on their features and user preferences.

### **3.Hybrid Models**: Combine CF and CBF techniques to leverage the strengths of both approaches.

### 

### 

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

**Visualization Techniques**

**1.Bar charts**: Compare the performance of different models or algorithms.

**2.Heatmaps**: Visualize the similarity between users or movies.

**3.Scatter plots:** Examine the relationship between different variables (e.g., user ratings and movie features).

**4.Histograms**: Visualize the distribution of user ratings or movie features.4. Histograms: Visualize the distribution of user ratings or movie features.

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

Programming Languages

1. Python: A popular language for machine learning and data science.

2. R: A language for statistical computing and data visualization.

Machine Learning Libraries

1. TensorFlow: An open-source machine learning library.

2. PyTorch: An open-source machine learning library.

3. Scikit-learn: A library for machine learning in Python.

Data Processing Libraries

1. Pandas: A library for data manipulation and analysis.

2. NumPy: A library for numerical computing. Programming Languages

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

**1.RAHUL TEJA P [EDA,Model Development]**

**2.TAMIZHMANI[Future engineering]**

**3.SIVA C [Documentation and reporting]**

**4.SHIVAM KESHARI [Data cleaning]**