Phase-1

Student Name: MEGARAJ S

Register Number:712523121014

Institution: PPG INSTITUTE OF TECHNOLOGY

Department: BE BIO MEDICAL ENGINEERING

Date of Submission: 25-04-2025

Delivering personalized movie recommendations with an AI-driven matchmaking system

1.Problem Statement

Creating an AI-driven personalized movie recommendation system is a compelling project that combines machine learning, data analysis, and user experience design. Here's a comprehensive guide to help you develop such a system, inspired by industry practices and academic research.

2.Objectives of the Project

Primary Objectives

- Personalized Recommendations: Provide accurate and relevant movie recommendations tailored to individual user preferences.
- Enhanced User Experience: Improve user satisfaction and engagement through a personalized movie-watching experience.

Secondary Objectives

- Increased User Retention: Encourage users to return to the platform by offering relevant and engaging content.
- Discovery of New Content: Help users discover new movies, genres, and directors that align with their interests.
- Improved Recommendation Accuracy: Continuously refine and improve the accuracy of movie recommendations.

Key Benefits

- Increased User Engagement: Personalized recommendations lead to increased user engagement and time spent on the platform.
- Better Content Discovery: Users discover new content that resonates with them, increasing overall satisfaction.
- Competitive Advantage: A personalized recommendation system sets the platform apart from competitors.

Technical Objectives

- Develop Effective Algorithm: Design and develop an AI-driven matchmaking algorithm that accurately matches users with movies.
- Data Analysis: Collect and analyze user data to inform recommendation decisions.
- Scalability: Ensure the system can handle large volumes of user data and provide recommendations in real-time.

3.Scope of the Project

Functional Scope

- User Profiling: Create user profiles based on their movie-watching history, ratings, and preferences.
- Movie Database: Develop a comprehensive database of movies, including metadata such as genre, director, cast, and plot summary.
- Recommendation Algorithm: Design and implement an AI-driven matchmaking algorithm that provides personalized movie recommendations to users.
- User Interface: Develop a user-friendly interface that allows users to interact with the system, view recommendations, and provide feedback.

Technical Scope

- Data Collection: Collect user data from various sources, such as ratings, reviews, and viewing history.

- Data Analysis: Analyze user data to identify patterns and preferences.
- Machine Learning: Implement machine learning algorithms to improve the accuracy of movie recommendations.
- Integration: Integrate the system with existing movie streaming platforms or develop a standalone application.

Data Scope

- User Data: Collect and analyze user data, including ratings, reviews, and viewing history.
- Movie Metadata: Collect and integrate metadata for a vast collection of movies.
- Data Storage: Design and implement a data storage solution to manage user data and movie metadata.

Performance Scope

- Recommendation Accuracy: Ensure that the system provides accurate and relevant movie recommendations to users.
- Response Time: Optimize the system to provide recommendations in real-time.
- Scalability: Ensure that the system can handle a large volume of user data and provide recommendations to a large user base.

Limitations and Assumptions

- Data Quality: The system assumes that user data is accurate and reliable.
- Data Privacy: The system must comply with data privacy regulations and ensure the security of user data.

- Content Availability: The system is limited by the availability of movie content and metadata.

4.Data Sources

User Data

- User Ratings: Collect ratings provided by users for movies they have watched.

- User Reviews: Collect text reviews written by users about movies.
- Watch History: Collect data on the movies users have watched.
- User Profiles: Collect demographic information about users, such as age, location, and preferences.

Movie Data

- Movie Metadata: Collect metadata about movies, such as title, genre, director, cast, and plot summary.
- Movie Ratings: Collect aggregate ratings for movies from various sources.
- Movie Tags: Collect tags or keywords associated with movies.

External Data Sources

- Movie Databases: Utilize public movie databases, such as IMDB or The Movie Database (TMDb).
- Social Media: Collect data from social media platforms about users' movie preferences and opinions.
- Streaming Platforms: Integrate with streaming platforms to collect user data and movie metadata.

Data Collection Methods

- User Input: Collect data directly from users through ratings, reviews, and profiles.
- Web Scraping: Collect data from public websites and databases.
- API Integration: Integrate with APIs from streaming platforms and movie databases to collect data.

Data Types

- Structured Data: Collect structured data, such as ratings and metadata.
- Unstructured Data: Collect unstructured data, such as text reviews and social media posts.

Data Quality

- Data Accuracy: Ensure that the data collected is accurate and reliable.
- Data Completeness: Ensure that the data collected is comprehensive and covers a wide range of movies and users.
- Data Consistency: Ensure that the data is consistent across different sources and formats.

5.High-Level Methodology

- **Data Collection**
- Data Sources Identification: Identify relevant data sources, such as user ratings, reviews, and movie metadata.
- Data Collection: Collect data from identified sources, ensuring data quality and integrity.

- **Data Preprocessing**
- Data Cleaning: Clean and preprocess data to handle missing values, outliers, and inconsistencies.
- Data Transformation: Transform data into a suitable format for analysis and modeling.

- **Model Development**

Feature Engineering: Extract relevant features from preprocessed data.

- Model Selection: Select suitable machine learning algorithms for matchmaking, such as collaborative filtering or content-based filtering.
- Model Training: Train the model using preprocessed data and evaluate its performance.

- **Model Evaluation**

- Evaluation Metrics: Define evaluation metrics, such as precision, recall, and F1-score.
- Model Testing: Test the model on a holdout dataset to evaluate its performance.

Deployment

- System Design: Design a system architecture to deploy the model and integrate it with a user interface.
- Model Deployment: Deploy the trained model in the system.
- User Interface Development: Develop a user-friendly interface to interact with the system.

- **Maintenance and Improvement**
- Model Monitoring: Continuously monitor the model's performance and update it as needed.
- Data Updates: Regularly update the data to ensure the model remains accurate and relevant.

Model Refining: Refine the model based on user feedback and performance metrics.

6.Tools and Technologies

- **Programming Languages**: Python
- **Libraries**: Pandas, NumPy, Scikit-learn, Surprise, TensorFlow/Keras
- **Web Framework**: Flask for building the user interface
- **Deployment**: Heroku or AWS for hosting the application

7.Team Members and Roles

Team Member	Role	Responsibilities
MEGARAJ S	Team Lead & Coordinator	Oversees task management, roadmap planning, deployment, helps in chatbot logic and code review. Ensures team collaboration.

GOKUL R	Data & Research Specialist	Finds datasets, handles EDA, assists in data cleaning and feature engineering. Documents research.
PRIYADHARSHINI L	NLP & Chatbot Logic Assistant	Helps in intent classification setup, supports response logic design, assists Dharanidaran in data tasks
MITHRA S	Frontend & Documentation Support	Builds chatbot UI using Streamlit (with help), documents project work, maintains ticket records in CSV.
All Members	**Reviewers & Testers**	**Participate in testing, debugging, giving feedback, preparing final presentation and report.**