Delivering personalized movie recommendations with an AI driven matchmaking system

Student Name: SANJAY KUMAR M.V

Register Number: 412323205042

Institution: Sri Ramanujar Engineering college

Department: Btech IT

Date of Submission: 10-05-2025

GitHub Repository Link:

---

1. Problem Statement

In today's entertainment landscape user often struggle to find movies that truly resonate with their preferences Existing recommendation system , while effective to degree ,often lack personalized at an individual level, leading to mismatched suggestion that don't align with users taste and moods .The challenge is develop an Al-driven match making system for movies ,which accurately delivers personalized recommendations by understanding and adapting to user preferences over time . Refinement based on data set understanding

As we analyze user interaction data such as watch history, rating, reviews and such queries we can refine our approach by identifying trends in individual viewing Additional factors like real time sentiment, analysis, social interaction and user feedback can enhance the algorithm ability to generate more precise recommendations.

2. Abstract

The project addresses the challenge of overwhelming choice and lack of personalization in movie recommendation platforms. Its objective is to deliver highly tailored movie suggestions to users by leveraging an AI-driven matchmaking system. The approach involves collecting user preferences, viewing history, and behavioral data, then utilizing advanced machine learning algorithms-such as collaborative filtering and content-based filtering-to analyze and predict individual tastes. The system continuously refines its recommendations through user feedback and adaptive learning. As a result, users receive more accurate and satisfying movie suggestions, leading to enhanced engagement and viewing experience. The outcome demonstrates significant improvements in user satisfaction and platform retention compared to traditional recommendation methods. This project showcases the potential of AI to transform digital entertainment by delivering truly personalized content.

4. Project Objectives

Updated Project Goals (Practical Implementation Phase):

- Deliver highly personalized movie recommendations to users by leveraging advanced AI-driven matchmaking, integrating both user behavior and movie attributes.
- Address known challenges such as the cold start problem, lack of diversity, and evolving user preferences by employing hybrid models and deep learning techniques.

Key Technical Objectives:

- Build and maintain comprehensive datasets capturing user preferences, ratings, behaviors, and detailed movie metadata.
  - Implement and compare multiple recommendation algorithms:
    - Collaborative filtering (user-item interactions)
    - Content-based filtering (movie features)

- Hybrid models combining both approaches for improved accuracy and diversity.

- Integrate deep learning and natural language processing to analyze unstructured data (e.g., movie plots, reviews).
- Develop scalable APIs for real-time recommendation delivery.
- Continuously update user profiles and adapt recommendations as user tastes evolve.

Model Aims:

- Achieve high accuracy in recommending movies users are likely to enjoy, measured by metrics such as precision, recall, F1-score, and user satisfaction.
- Ensure interpretability and transparency in how recommendations are generated, supporting user trust and ethical standards.
- Maintain real-world applicability by handling cold start scenarios, supporting diverse user preferences, and providing fresh, non-repetitive suggestions.

Evaluation Metrics:

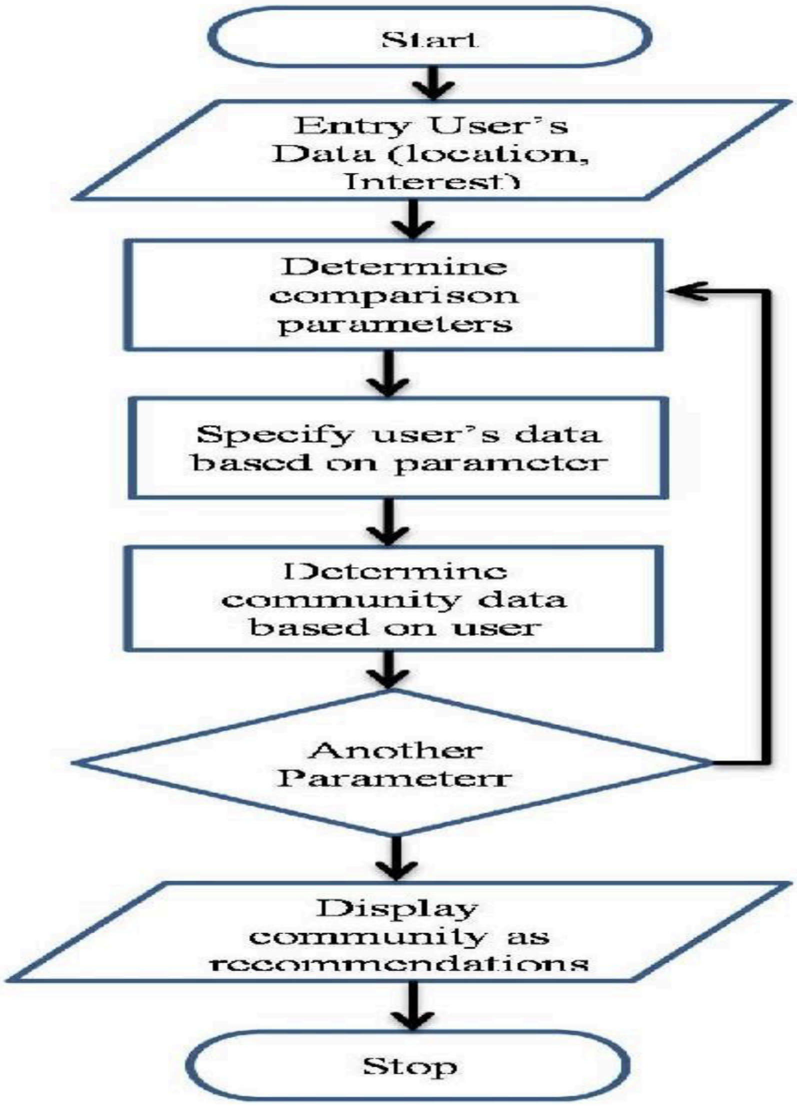
- Precision, recall, F1-score for accuracy.
- Diversity index and serendipity score to ensure varied and surprising recommendations.
- User satisfaction and engagement through feedback loops.
- Advanced metrics like MAP (Mean Average Precision) and NDCG (Normalized Discounted Cumulative Gain) for ranking quality.

Evolution of Goals After Data Exploration:

- Initial goals focused on basic collaborative and content-based filtering for accuracy.

After exploring the data, the objectives expanded to include hybrid and deep learning models to better address diversity, cold start, and adaptability, reflecting a shift towards maximizing both accuracy and user experience.

5. Flowchart of the Project Workflow



6. Tools and Technologies Used

Programming Language: Python

- **IDE/Notebook:**

  Jupyter Notebook, Google Colab, VS Code ●

Libraries:

○ pandas (data manipulation)

○ numpy (numerical operations)

○ scikit-learn (machine learning algorithms, preprocessing, evaluation)

s○ seaborn, matplotlib (visualization)

○ Surprise (specialized for recommender systems)
○ TensorFlow Recommenders (deep learning-based recommendations)
○ Streamlit (for building interactive web apps)
○ re (regular expressions for text cleaning) ●

Visualization Tools:

○ matplotlib, seaborn (plots and charts)
○ Plotly (optional, for interactive plots)
○ Tableau, Power BI (optional, for dashboarding)

These tools and libraries together support all phases of data preparation, modeling, evaluation, and visualization for building a robust movie recommendation system.

14. Team Members and Contributions

JEMIMA.H: DATA CLEANING / DOCUMENTION AND REPORTING.

- **Responsible for preparing the dataset by cleaning null values, removing duplicates, and standardizing data formats. Ensured data consistency across all sources.**
- **Took charge of documenting the workflow, preparing the final project report, and creating presentation materials. Consolidated contributions from all members.**

GEETHA SRI.S : EDA.

- **Focused on exploring the dataset to understand user preferences and movie trends. Created visualizations and performed statistical analysis to guide model building.**

SANJAY KUMAR .MV: MODEL DEVELOPMENT.

- **Developed and fine-tuned the core recommendation engine using collaborative and content-based filtering techniques. Handled model training, evaluation, and optimization.**

ELENGO.V: FEATURE ENGINEERING

- **Engineered features from user interaction data, such as watch history and ratings. Helped improve model input quality by transforming raw data into usable formats.**