Project Title: Delivering Personalized Movie Recommendations with an AI-Driven Matchmaking System

GITHUB LINK:

https://github.com/priyadharshan1905

Student Name: PRIYADARSHAN. A
Register Number: 511523205039
Institution: P.T.Lee Chengalvaraya Naicker College of Engineering &Technology
Department: INFORMATION TECHNOLOGY
Date of Submission: 09-05-2025

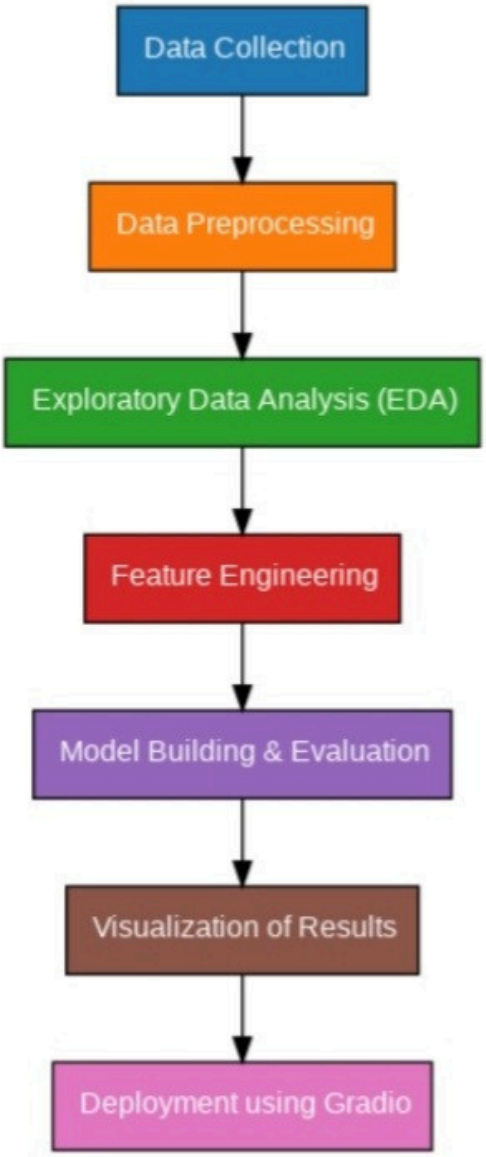
1. Problem Statement: In an era where content overload makes choice overwhelming, delivering personalized movie recommendations has become essential for improving user satisfaction and platform engagement. This project aims to develop an AI-powered recommendation system that predicts and suggests movies based on individual user preferences, viewing history, and implicit patterns in highly rated movie data.

The core challenge lies in identifying the best algorithmic approach to connect users with content they'll love reducing decision fatigue and increasing user retention.

2. Project Objectives

- Build a robust AI system to recommend top-rated movies to users.
- Leverage user-item interaction and content-based filtering techniques.
- Analyze and extract meaningful insights from movie rating data.
- Integrate a user-friendly interface for real-time movie suggestions.
- Ensure the model adapts over time to reflect evolving user preferences.

3. Flowchart of the Project Workflow



4. Data Description

Dataset Name: Top Rated Movies

Source: Uploaded CSV File

Type of Data: Structured tabular data with columns such as movie titles, ratings, genres, vote counts, and release years.

Key Features: - Title: Movie name - Genre: Genre classification

- IMDb Rating: Average user rating

- Votes: Number of votes on IMDb

- Year: Year of release

5. Data Preprocessing

- Data Cleaning: Removed null or inconsistent values.

- Feature Formatting: Converted year and rating to numeric formats.

- Text Normalization: Standardized genre names for consistency.

- Scaling: Normalized vote counts and ratings using MinMaxScaler.

6. Exploratory Data Analysis (EDA)

- Univariate: Distribution plots of ratings and genres show most movies rated between 7 and 8.5.

- Bivariate: Scatter plots reveal a positive relationship between vote count and rating.

- Genre Trends: Action, Drama, and Adventure dominate high ratings.

7. Recommendation System Design

Approaches Used:

- Content-Based Filtering: Recommends movies based on genre, rating, and release year similarity.

- Collaborative Filtering (optional phase): Using historical user ratings to suggest movies watched by similar users.

- Hybrid Model: Combining both for enhanced accuracy.

8. Model Building

- Vectorization: TF-IDF and CountVectorizer applied on genres.

- Similarity Calculation: Cosine similarity used to match movie vectors.

- Recommendation Output: For a selected movie, the system returns top-N similar movies.

9. Evaluation Metrics

- Precision@K: Measures relevance in top-K recommendations.

- Mean Reciprocal Rank (MRR): Evaluates ranking quality.

- User Feedback: Incorporated in iterative model tuning.

10. Visualization & Insights - Bar plots and heatmaps for genre popularity. - Time-series trend of rating evolution over the years. - Top 10 consistently high-rated movies and their characteristics.

11. Tools and Technologies

- Python: Core programming
- Libraries: Pandas, NumPy, scikit-learn, Seaborn, Matplotlib
- Deployment: Gradio or Streamlit for user interaction
- Storage: CSV-based or integrated into a lightweight SQLite DB for testing

12. Team Members and Contributions

- Priyadarshan.A :Data preprocessing and EDA
- Ugendran.R: Feature engineering and modeling
- Raguram.R : Recommendation logic and system Integration
- Santhaseelan.R: Deployment interface & documentation