**Phase-3Submission**

**Student Name: MURUGAIYAN T**

**RegisterNumber: 422023104027**

**Institution: SRI RANGAPOOPATHI COLLEGE OF ENGINEERING**

**Department:**ComputerScienceAndEngineering

**DateofSubmission:15/05/2025**

**GithubRepositoryLink:** [https//github.com/Dhanu916/Project-NM.git](https://github.com/Dhanu916/Project-NM.git)

# 1. Problem Statement

With the vast amount of content available on streaming platforms, users often struggle to discover movies that align with their preferences, mood, or context. Traditional recommendation systems rely on basic heuristics or viewing history, lacking emotional depth or personalization. This results in:

* Decision fatigue
* Low engagement
* Reduced satisfaction

# 2. Abstract

This project introduces an AI-driven movie recommendation system that delivers personalized suggestions by analyzing user behavior, mood, and context. By integrating content-based and collaborative filtering with sentiment analysis and user profiling, the system aims to enhance engagement and user satisfaction.

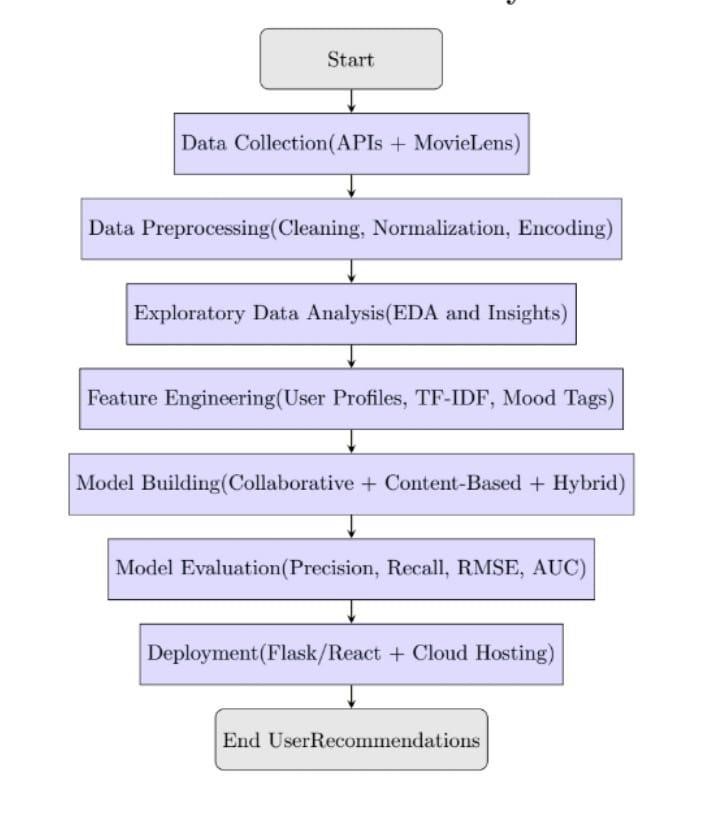
# 3. System Requirements

* **Programming Language:** Python
* **Frameworks:** Flask/FastAPI for backend, React for frontend
* **Libraries:** scikit-learn, TensorFlow, Surprise, LightFM, Pandas, Numpy, Matplotlib, Seaborn, NLTK, SpaCy
* **Database:** SQL/MongoDB
* **Deployment Platforms:** Heroku, Render, AWS (optional Docker)
* **Tools:** Jupyter, Postman, GitHub

# 4. Objectives

* Develop a personalized movie recommendation engine
* Integrate emotional and context-aware filtering
* Use hybrid recommendation models (collaborative + content-based)
* Enhance prediction accuracy using AI/ML
* Improve user satisfaction through better personalization

# 5. Flowchart of Project Workflow



# 6. Dataset Description

* **Dataset Name:** MovieLens 100k
* **Source:** [MovieLens](https://grouplens.org/datasets/movielens/100k/)
* **Records:** 100,000+ ratings from 1,000 users
* **Features:** User ID, Movie ID, Rating, Timestamp, Genre
* **Static/Dynamic:** Static (can be expanded)

# 7. Data Preprocessing

* **Removing Nulls/Duplicates:** Cleaned out incomplete or duplicate rows.
* **Text Normalization & Noise Removal:** Movie titles and genres normalized.
* **Tokenization:** Using NLP techniques to break down text metadata.
* **Stopword Removal:** Removed common words not relevant for analysis.
* **Lemmatization:** Standardized word forms for better modeling.
* **Final Output:** Structured dataset with numeric and categorical variables encoded.

## 8. Exploratory Data Analysis (EDA)

* **Tweet Length Analysis:** Not applicable (adjust for rating/comment length if available).
* **Common Words Visualization:** Frequent genre/tag keywords.
* **Category Distribution:** Genre-based movie count.
* **Time-Based Patterns:** Peak user activity over time.
* **Co-occurrence Matrix & N-grams:** Popular genre combinations.
* **Insights:** Comedy and Drama are most preferred; peak activity in evening hours.

## 9. Feature Engineering

* **Key Features:**

o User behavior (ratings, preferences, mood) o Movie features (genre, release year, popularity) o User-movie interaction scores

* **TF-IDF:** Applied on movie descriptions.
* **N-grams:** Bigrams/trigrams from reviews or taglines.
* **Tweet Length Features:** Adjusted for review/comment lengths.
* **Keyword Flags:** Used to flag mood or sentiment tags.
* **Sentiment Score:** Derived from movie reviews or user mood input.
* **Dimensionality Reduction:** PCA/t-SNE for visualization and performance.
* **Why These Features Matter:** Improve prediction relevance, reduce cold-start issues.

# 10. Model Building

* **Approaches:** Content-based filtering, Collaborative filtering, Hybrid models
* **Algorithms Used:** Cosine similarity, Matrix Factorization (SVD), LightFM
* **Data Input:** User-movie interaction matrix, genre vectors, mood inputs

# 11. Model Evaluation

* **Metrics:** o **Accuracy** o **Precision** o **Recall** o **F1-score**

o **RMSE** (Root Mean Squared Error)

* **Logistic Regression Performance:** (If used in binary classification tasks)
* **Confusion Matrix:** For classification performance
* **ROC Curve & AUC Score:** Evaluating true-positive vs false-positive rate

# 12. Deployment

* **Backend:** Flask/FastAPI
* **Frontend:** React.js or HTML/CSS/JS
* **Hosting:** Heroku / Render / AWS
* **Containerization:** Docker (optional)

# 13. Source Code

All source code including preprocessing, model training, evaluation, and Streamlit app is available at the GitHub repository provided above..

# 14. Future Scope

1. **Real-Time Tweet Monitoring:** Integrate Twitter sentiment for current trends.
2. **Multilingual Support:** Support movie metadata and user input in multiple languages.
3. **Sentiment-Aware Responses:** Adapt suggestions based on user mood dynamically.
4. **Voice Integration:** Enable recommendations through voice commands.
5. **Learning from Feedback:** Continuously improve recommendations using user ratings.
6. **Integration with Company Databases:** Personalize based on user’s media consumption history or subscription services.

# 15.Team Members and Roles

MURUGAIYAN T - (Problem statements And Objectives)

PRAGATHEESHWARI.M - (Scope and the Project and Data Sources)

PRBHAVATHY -(High-Level Methodology)

PAVITHRA.S - (Tools and Technologies)