Phase-2 Submission

Student Name: Sai Mouleeshwar S.M

Register Number: 712523106015

Institution: PPG institute of technology

Department: B tech information technology

Date of Submission: 05/05/2025

Github Repository Link:

https://github.com/SaiMouleeshwar10/NM_SAIMOULEESHWAR_DS

---

1. Problem Statement

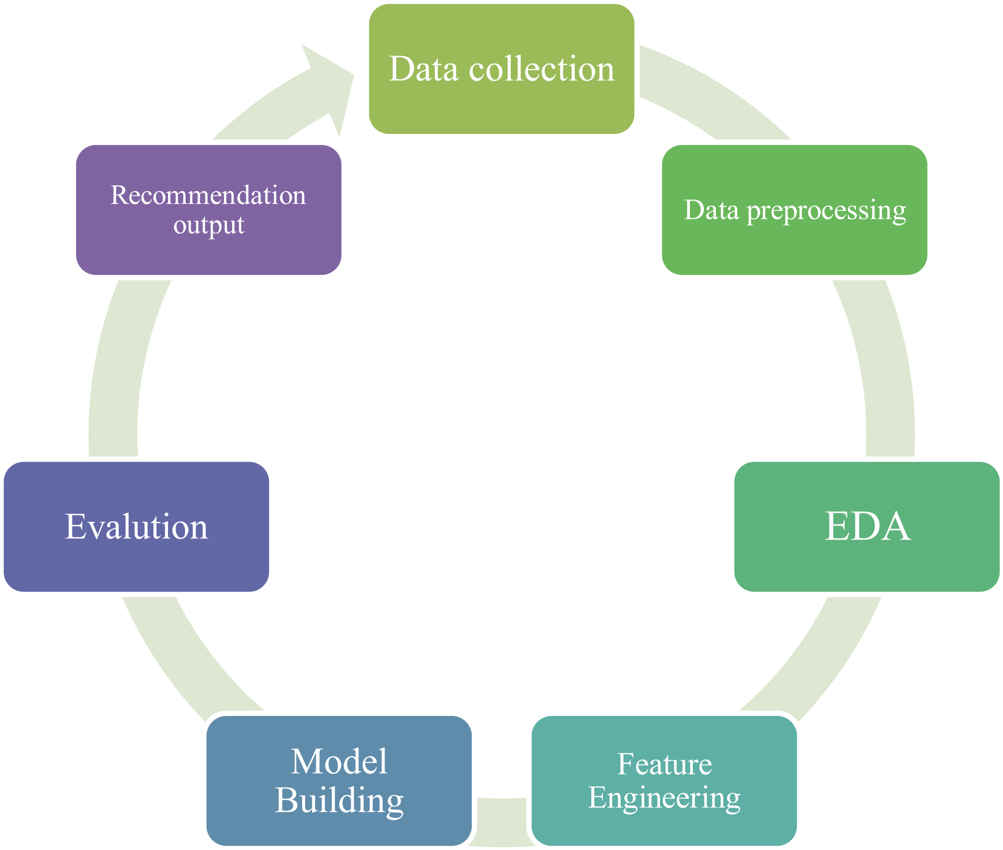
With the exponential growth of digital streaming platforms, users are often overwhelmed by the sheer volume of content available. Navigating this vast landscape without intelligent guidance leads to choice fatigue, resulting in unsatisfactory user experiences. This project addresses this issue by developing an AI-driven movie recommendation system that merges collaborative filtering, content-based filtering, and user-movie matchmaking techniques to deliver highly personalised and explainable movie suggestions.

- *Problem Type: Recommendation (Ranking), Regression (Rating Prediction), and Clustering (User Segmentation)*
- *Real-world Impact: Boosts engagement, satisfaction, and retention by enhancing user-content alignment.*

2. Project Objectives

- *Develop a robust machine learning-based recommendation system.*
- *Implement both collaborative and content-based filtering techniques.*
- *Compare and evaluate models such as KNN, SVD, and Matrix Factorization.*
- *Engineer features to improve recommendation accuracy.*
- *Deliver interpretable, real-time, and user-specific movie recommendations.*

3. Flowchart of the Project Workflow



4. Data Description

- ***Dataset***: *MovieLens 100K*
- ***Source***: *https://grouplens.org/datasets/movielens/*
- ***Type***: *Structured (ratings.csv, movies.csv, users.csv)*
- ***Records***: *100,000 ratings from 943 users on 1682 movies*
- ***Target Variable***: *rating (1–5 scale)*
- ***Static/Dynamic***: *Static*

5. Data Preprocessing

- *Merged multiple datasets (ratings, movies, users).*
- *Removed duplicates and handled missing/null values.*
- *Encoded genres using multi-hot encoding.*
- *Applied label encoding to user and movie IDs.*
- *Normalized rating scale.*
- *Transformed datasets for input into matrix-based algorithms.*

Key Data Fields

- *userId: Unique anonymous user identifier.*
- *movieId: Identifier linked to movie metadata.*
- *rating: User-assigned score (float, 0.5–5.0).*
- *timestamp: Unix timestamp indicating when a rating was submitted.*

timestamp

- ***Type**: Integer (Unix time format — seconds since Jan 1, 1970)*
- ***Description**: The time when the rating was recorded.*
- ***Properties**:*
    - *Useful for **time-based analysis**, like:*
        - *Detecting trends over time,*
        - *Implementing **time decay** (recent ratings get more weight),*
        - *Splitting data chronologically for **train/test**.*

- *Can be converted into readable formats (e.g., using Python's datetime module).*

6. Exploratory Data Analysis (EDA)

Univariate Analysis

- *Ratings skewed toward higher values (mostly 4s and 5s).*
- *Most watched genres: Romance, Action.*
- *Top-rated movies based on average score and volume.*

Bivariate/Multivariate Analysis

- *Heatmaps showing correlation between genres and average user ratings.*
- *Segmented user behavior: identification of "power users".*
- *Genre preference clusters help inform better personalized suggestions.*

7. Feature Engineering

- ***Genre Embeddings:*** *Capturing deeper genre characteristics.*
- ***User Profiles:*** *Vector representations of past preferences.*
- ***Dimensionality Reduction:*** *PCA applied to user-genre matrices.*
- ***Temporal Features:*** *Time decay functions applied to rating values to prioritize recent data.*
- *Built user profile vectors based on past preferences.*
- *Applied PCA on user-genre matrix for dimensionality reduction.*
- *Engineered temporal features (e.g., rating time decay).*

8. Model Building

Models Implemented

- *KNN (Collaborative Filtering): Based on user-user or item-item similarities.*

- *SVD (Matrix Factorization): Decomposes user-item matrix to discover latent factors.*

- *Content-Based Filtering: Recommends based on genre and movie metadata similarity.*

Training and Evaluation

- *Train/Test Split: 80/20*
- *Metrics Used:*
  - *RMSE (Root Mean Squared Error)*
  - *MAE (Mean Absolute Error)*
  - *Precision@K, Recall@K*
  - *NDCG (Normalized Discounted Cumulative Gain), MAP (Mean Average Precision)*

9. Visualization of Results & Model Insights

Performance Charts:

- *A comparison chart of RMSE across different models is mentioned.*
- *Bar plots are used to display Precision/Recall at K, illustrating the quality of top-N recommendations.*

***Confusion Matrix**:*

- ***RMSE (Root Mean Squared Error)** and **MAE (Mean Absolute Error)** — for evaluating rating predictions.*
- ***Precision@k** and **Recall@k** — for evaluating the quality of top-N recommendations.*
- ***Ranking correlation metrics** (like NDCG or MAP) — for assessing the relevance order of recommendations.*

- *Not applicable for rating regression*

Feature Importance:

- *SVD outperforms KNN in sparse data conditions.*
- *Temporal features improve recent movie prediction relevance.*
- *Content-based filtering boosts cold-start performance for new users/movies.*

10. Tools and Technologies Used

- ***Language:*** *Python*
- ***IDE:*** *Jupyter Notebook*

❖ ***Libraries:***

- ***Data Handling:*** *pandas, numpy*
- ***Visualization:*** *matplotlib, seaborn, Plotly*
- ***Modeling:*** *scikit-learn, Surprise, LightFM*

11. Team Members and Contributions

S NO	***TEAM MEMBERS***	***CONTRIBUTIONS***
1.	***Sai Mouleeshwar S.M***	*Data preprocessing*
2.	Vignesh V.	*EDA*, Evaluation
3.	Madhan raj R.	*Feature engineering*
4.	Selvam A.	*Modeling, documentation*