CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE | DELHI NCR | PUNE

DEPARTMENT OF AIML & DS

MINI PROJECT REPORT

on

Movie Recommendations System Using User based Collaborative Filtering

Course Name:

Honours In Artificial Intelligence and Machine Learning

Course Code: CSHO432AIP

School of Engineering and Technology,

CHRIST (Deemed to be University),

Kumbalagodu, Bengaluru-560 074

March 2026

CHRIST

(DEEMED TO BE UNIVERSITY)

BANGALORE | DELHI NCR | PUNE

Certificate

This is to certify that Lakshmi Prakash, Kasturi S, Vismaya G (2460396, 2460390, 2460473) has successfully completed the Mini Project work for the course Honours In Artificial Intelligence and Machine Learning in partial fulfilment for the award of Bachelor of Technology during the year 2025-2026.

FACULTY- IN CHARGE

NAME: Lakshmi Prakash, Kasturi S, Vismaya G

REGISTER NO: 2460396, 2460390, 2460473

INDEX

1. ABSTRACT

With the rapid expansion of online streaming platforms, users are exposed to thousands of movies, making content selection challenging. Recommendation systems help users discover relevant content by analyzing their preferences and behavior.

This project presents a **Movie Recommendation System using User-Based Collaborative Filtering**. The system analyzes user rating patterns and identifies users with similar preferences. Movies liked by similar users are then recommended to the target user.

Cosine similarity is used to measure similarity between users based on their rating vectors. The system is implemented using Python, Pandas, Scikit-learn, and Flask to provide a web-based interface for user interaction.

The developed system demonstrates how user-based collaborative filtering can effectively generate personalized movie recommendations.

2. INTRODUCTION

Recommendation systems are intelligent systems designed to predict user preferences and suggest relevant items. They are widely used in platforms such as:

- Netflix
- Amazon
- Spotify
- YouTube

Collaborative Filtering is one of the most popular recommendation techniques. It works based on the assumption that:

Users who had similar interests in the past will likely have similar interests in the future.

There are two types of collaborative filtering:

- User-Based Collaborative Filtering
- Item-Based Collaborative Filtering

This project focuses specifically on **User-Based Collaborative Filtering**, where recommendations are generated by finding users with similar rating behavior

3. OBJECTIVES

The main objectives of this project are:

1. To design and implement a movie recommendation system using user-based collaborative filtering.
2. To compute similarity between users using cosine similarity.
3. To generate personalized movie recommendations for a selected user.
4. To build a simple web interface using Flask for user interaction.
5. To evaluate the effectiveness of the recommendation approach.

4. DATASET DESCRIPTION

This project uses the **MovieLens dataset**, which contains user ratings for movies.

Dataset attributes include:

- userId – Unique ID of user
- movieId – Unique ID of movie
- rating – Rating given by user (1–5 scale)
- timestamp – Time of rating

The dataset used for the mini project contains approximately 1000+ rating records.

The ratings dataset is used to build the user-item interaction matrix required for collaborative filtering

5. METHODOLOGY

The system is developed using the following steps:

5.1 Data Preprocessing

• Load dataset using Pandas
• Remove duplicates
• Handle missing values
• Convert ratings into numerical format

5.2 Creating User-Item Matrix

A pivot table is created:
• Rows → Users
• Columns → Movies
• Values → Ratings
Missing ratings are filled with 0.
This matrix represents user preferences.

5.3 User Similarity Calculation

Cosine similarity is used to compute similarity between users.

Similarity Formula:

Similarity(U1, U2) =(U1 · U2) / (||U1|| × ||U2||)

Users with higher similarity scores are considered similar.

5.4 Generating Recommendations

Steps:

1. *Select target user.*
2. *Find top N similar users.*
3. *Identify movies liked by similar users but not watched by target user.*

4. *Rank movies based on weighted average ratings.*
5. *Recommend top movies.*

5.5 Web Interface

A user-friendly interface is built using Streamlit.

- Select user ID
- Fetch recommendations
- Display recommended movies

6. SYSTEM ARCHITECTURE

The system follows this pipeline:

1. Load dataset
2. Preprocess data
3. Create user-item matrix
4. Compute user similarity matrix
5. Select target user
6. Identify similar users
7. Generate recommendations
8. Display results via Flask interface

7. TOOLS AND TECHNOLOGIES

Programming Language:

- Python

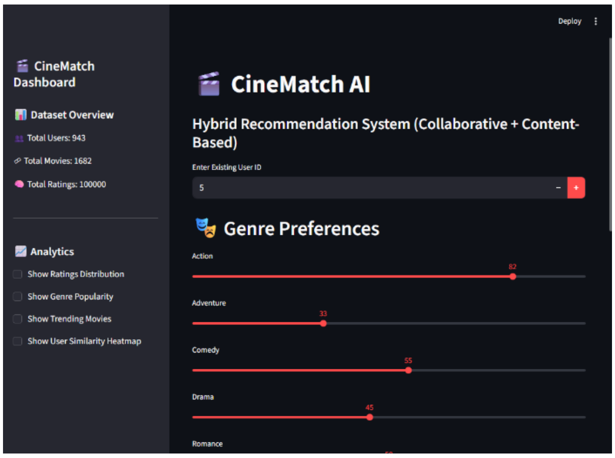
Libraries:

- Pandas (data handling)
- NumPy (numerical computing)
- Scikit-learn (machine learning)
- Matplotlib (visualization)

- Streamlit (GUI development)

Development Environment:

- Visual Studio Code



Total Users: 943

Total Movies: 1682

Total Ratings: 100000

CineMatch AI

Hybrid Recommendation System (Collaborative + Content-Based)

Enter Existing User ID: 5

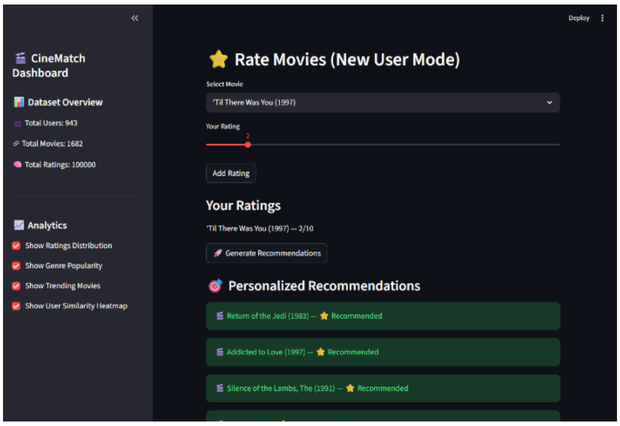
Action: 82

Adventure: 33

Comedy: 55

Drama: 45

Romance



Total Users: 943

Total Movies: 1682

Total Ratings: 100000

Rate Movies (New User Mode)

Select Movie: 'Til There Was You (1997)

Your Rating: 2

Your Ratings

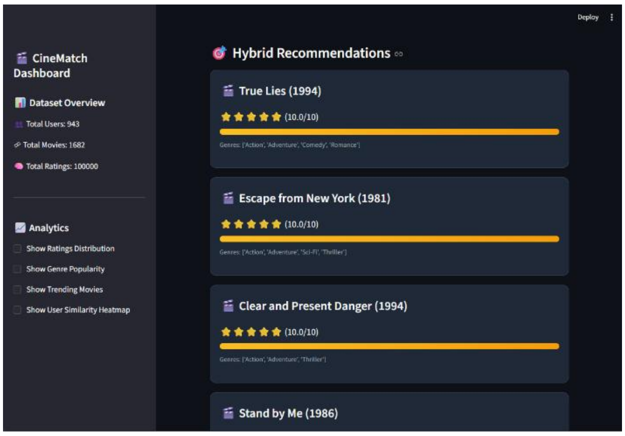
'Til There Was You (1997) — 2/10

Personalized Recommendations

Return of the Jedi (1983) — ⭐ Recommended

Addicted to Love (1997) — ⭐ Recommended

Silence of the Lambs, The (1991) — ⭐ Recommended



Total Users: 943

Total Movies: 1682

Total Ratings: 100000

Hybrid Recommendations

True Lies (1994)
⭐⭐⭐⭐⭐ (10.0/10)
Genres: ['Action', 'Adventure', 'Comedy', 'Romance']

Escape from New York (1981)
⭐⭐⭐⭐⭐ (10.0/10)
Genres: ['Action', 'Adventure', 'Sci-Fi', 'Thriller']

Clear and Present Danger (1994)
⭐⭐⭐⭐⭐ (10.0/10)
Genres: ['Action', 'Adventure', 'Thriller']

Stand by Me (1986)

8. RESULTS

The system successfully:

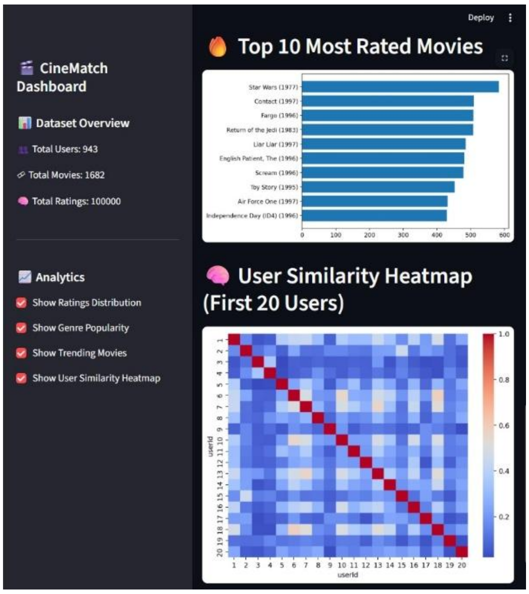
Identifies similar users
- Computes similarity scores
- Generates personalized movie recommendations
- Displays output through web interface

Example:

Target User: User 5

Recommended Movies:
- Movie A
- Movie B
- Movie C
- Movie D
- Movie E

The recommendations are based on ratings from users with similar preferences.



CineMatch Dashboard

Dataset Overview

Total Users: 943

Total Movies: 1682

Total Ratings: 100000

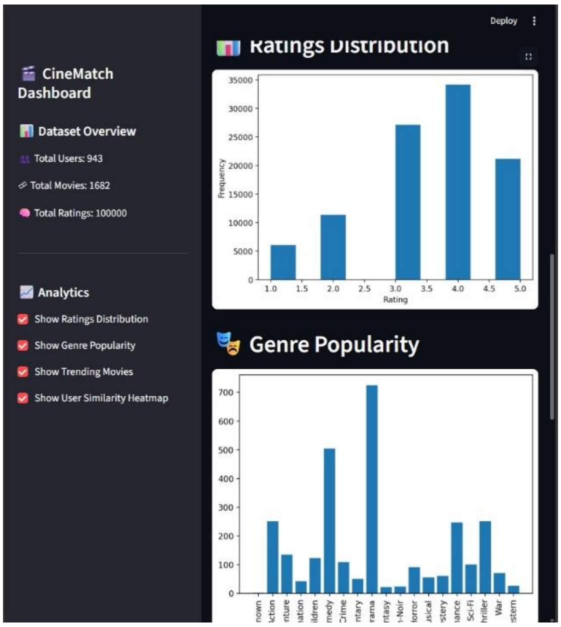
Analytics

☑ Show Ratings Distribution

☑ Show Genre Popularity

☑ Show Trending Movies

☑ Show User Similarity Heatmap



CineMatch Dashboard

Dataset Overview

Total Users: 943

Total Movies: 1682

Total Ratings: 100000

Analytics

☑ Show Ratings Distribution

☑ Show Genre Popularity

☑ Show Trending Movies

☑ Show User Similarity Heatmap

9. APPLICATIONS

This system can be applied in:

- Online movie streaming platforms
- E-commerce websites
- Music recommendation systems
- Online learning platforms
- News recommendation systems

10. ADVANTAGES

- Provides personalized recommendations
- Easy to implement
- Improves user satisfaction
- Scalable to large datasets
- Industry-relevant approach

11. LIMITATIONS

- Cold start problem (new users)
- Requires sufficient rating data

- Scalability issues with very large datasets
- Does not consider movie content features

12. FUTURE ENHANCEMENTS

The system can be improved by:

- Adding item-based filtering
- Building hybrid recommendation system
- Using deep learning models
- Deploying on cloud platform
- Adding real-time recommendation updates

13. CONCLUSION

The Movie Recommendation System using User-Based Collaborative Filtering demonstrates how machine learning techniques can be applied to generate personalized recommendations.

By identifying similar users and analyzing their preferences, the system effectively suggests relevant movies to target users. The integration of collaborative filtering with a Flask-based web interface provides a practical implementation of real-world recommendation systems.

This project highlights the importance of personalized recommendation technologies in modern digital platforms.