A Unified Approach to Movie and Music Recommendations with Hybrid ML

[1]P. BINDHU PRIYA,[2] MOGALATHURTHI BHARATH RAJU

[1]Assistant Professor, Department Of MCA, 2MCA Final Semester,
[1]Master of Computer Applications,
[1]Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India

Abstract:

In the era of exponential digital content growth, users are often overwhelmed by the vast range of available options, leading to decision fatigue and decreased satisfaction. Effective recommendation systems have therefore become a cornerstone for enhancing user engagement, retention, and overall platform value. This project presents a unified hybrid recommendation system designed to deliver highly relevant movie and music suggestions tailored to individual user preferences. The proposed system integrates content-based filtering, collaborative filtering, and feature similarity analysis to maximize recommendation accuracy. For movies, the content-based component employs TF-IDF (Term Frequency–Inverse Document Frequency) vectorization of movie overviews to capture semantic similarities, enabling recommendations of movies with closely related themes, plots, and genres. Complementing this, a collaborative filtering approach using Truncated Singular Value Decomposition (SVD) predicts personalized ratings by analyzing patterns in historical user–item interactions, thus capturing latent preference factors. For the music recommendation module, the system processes structured audio feature datasets containing parameters such as danceability, energy, loudness, tempo, and valence. By applying cosine similarity to these numerical feature vectors, it can recommend either artists or genres that closely match the acoustic profile of a selected choice. This dual-level recommendation—artist-based and genre-based—allows for flexible exploration and discovery of new music aligned with a listener's tastes. The entire solution is implemented using Python and Streamlit, offering an intuitive and interactive web-based interface. Users can seamlessly switch between movie and music recommendations, select the type of filtering, and instantly receive results in a visually organized format. The hybrid methodology ensures the system is capable of handling both textual (movie metadata) and numerical (music audio features) data, making it a versatile framework adaptable to other multimedia domains such as books, podcasts, or videos. Overall, this project demonstrates how combining multiple recommendation strategies within a unified architecture can lead to a richer, more personalized user experience, improving both content discoverability and user satisfaction while maintaining scalability and adaptability for future enhancements.

Keywords: Hybrid Recommendation System, Content-Based Filtering, Collaborative Filtering, TF-IDF, Truncated Singular Value Decomposition (SVD), Cosine Similarity, Movie Recommendation, Music Recommendation, Audio Feature Analysis, Streamlit, Personalized Content Discovery, Multimedia Recommendation, User Preference Modelling, Feature Similarity Analysis, Digital Content Overload.

INTRODUCTION:

In the digital age, the vast availability of movies, music, and other entertainment content has led to "choice overload," making it difficult for users to find content that matches their preferences. Recommendation systems address this challenge by suggesting relevant items, typically using either content-based filtering, which relies on item similarity, or collaborative filtering, which leverages the preferences of similar users. However, each method has limitations, such as lack of novelty in content-based systems and cold start issues in collaborative filtering**.** This project presents a unified hybrid recommendation system that integrates both movies and music. For movies, TF-IDF vectorization captures semantic similarities in plot overviews, while Truncated SVD predicts ratings based on collaborative patterns. For music, cosine similarity on audio feature datasets enables recommendations by artist and genre. Built with Python and Streamlit, the system offers an interactive interface and delivers more accurate, diverse, and personalized suggestions—showcasing the potential of hybrid approaches for multimedia content discovery.

1.1.1 Existing System:

Current movie recommendation systems primarily use the following methods:

1. **Content-Based Filtering**
   Recommends movies based on item features like genre, plot, or cast.
   - Technique: TF-IDF, Cosine Similarity
   - Limitation: Over-specialization, struggles with cold start for new users.
2. **Collaborative Filtering**
   Suggests movies based on user behaviour and preferences.
   - Technique: User-Item Matrix, SVD
   - Limitation: Requires large rating data, cold start for new items/users.

1.2. Challenges:

Building an effective movie recommendation system involves several technical and practical challenges, particularly when combining both Content-Based and Collaborative Filtering techniques. [11] Key challenges include**:**

1. Cold Start Problem

- User Cold Start: Difficulty in recommending movies to new users with no interaction history.
- Item Cold Start: Inability to recommend newly added movies that lack user ratings or metadata.

2. Data Sparsity

- User-item rating matrices are often sparse, as most users rate only a small number of movies.
- Sparse data leads to unreliable similarity measures and poor prediction accuracy.

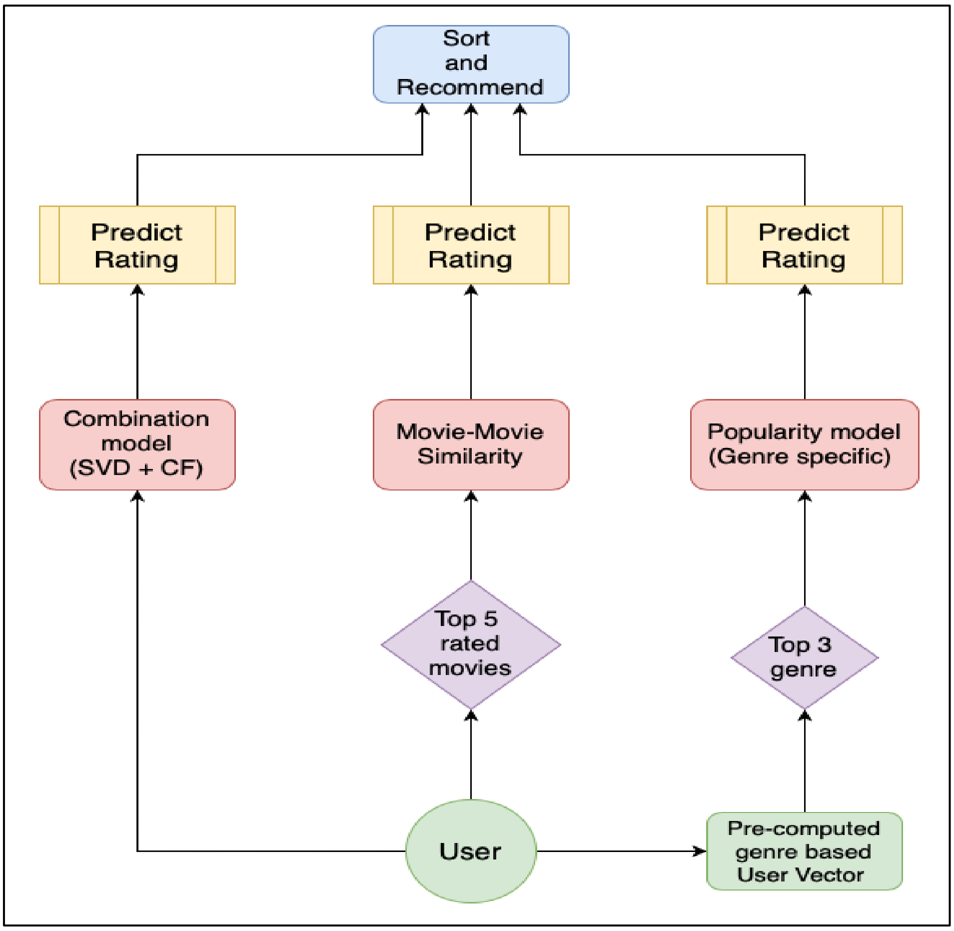
3. Scalability

- As the number of users and movies increases, the computational complexity of similarity calculations and matrix factorization grows.
- Real-time recommendations become slower without optimization and caching techniques.

1.2 Proposed system:

The proposed system is a hybrid movie recommendation engine that integrates both Content-Based Filtering and Collaborative Filtering techniques within a user-friendly Streamlit web application.[2] This dual-approach system is designed to overcome the limitations of traditional recommendation models by combining the strengths of both content and user interaction data**.**

Fig 1

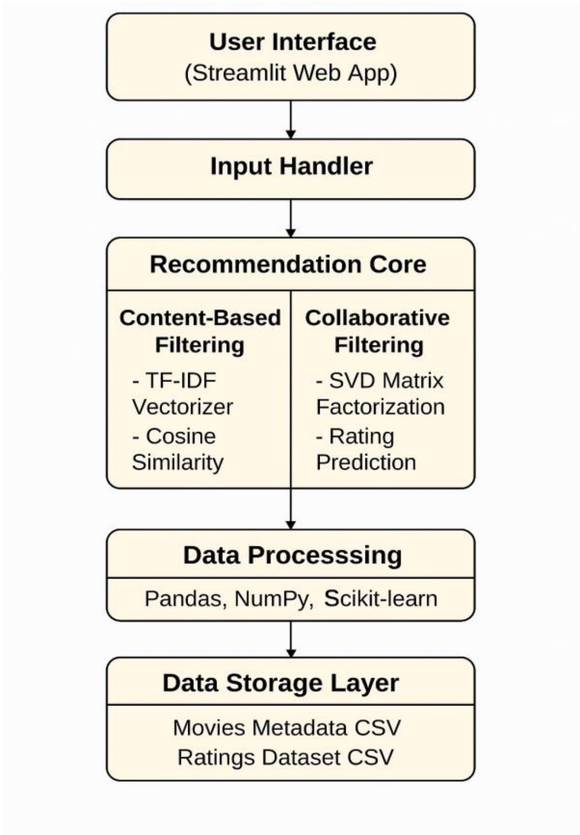


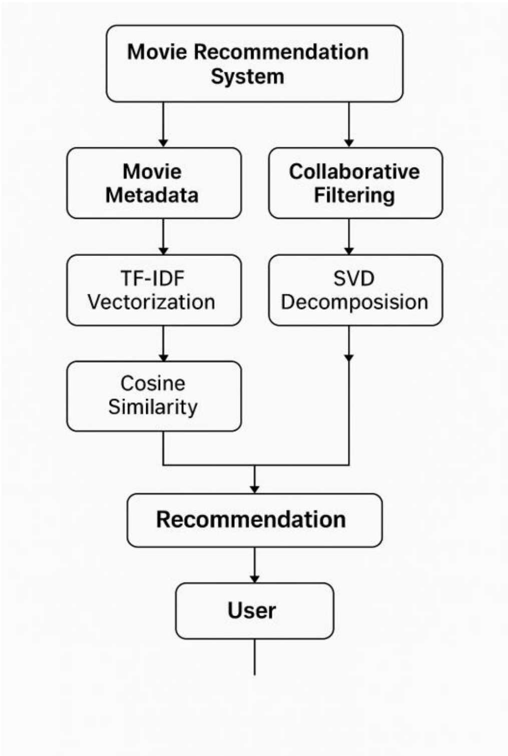
Advantages of the Proposed System

- Reduces cold start issues by using both user ratings and content data.
- Improves recommendation accuracy through hybrid logic.
- Enhances scalability and performance using caching and optimized computations.
- Offers an intuitive front-end for real-time user interaction.

2.1 Architecture:

The architecture of the proposed Movie Recommendation System is designed to integrate both Content-Based Filtering and Collaborative Filtering into a seamless web application using Streamlit.[4] It follows a modular and layered design for better maintainability, scalability, and performance.


User Interface
(Streamlit Web App)
Input Handler
Recommendation Core
Content-Based Filtering
- TF-IDF Vectorizer
- Cosine Similarity
Collaborative Filtering
- SVD Matrix Factorization
- Rating Prediction
Data Processsing
Pandas, NumPy, Scikit-learn
Data Storage Layer
Movies Metadata CSV
Ratings Dataset CSV

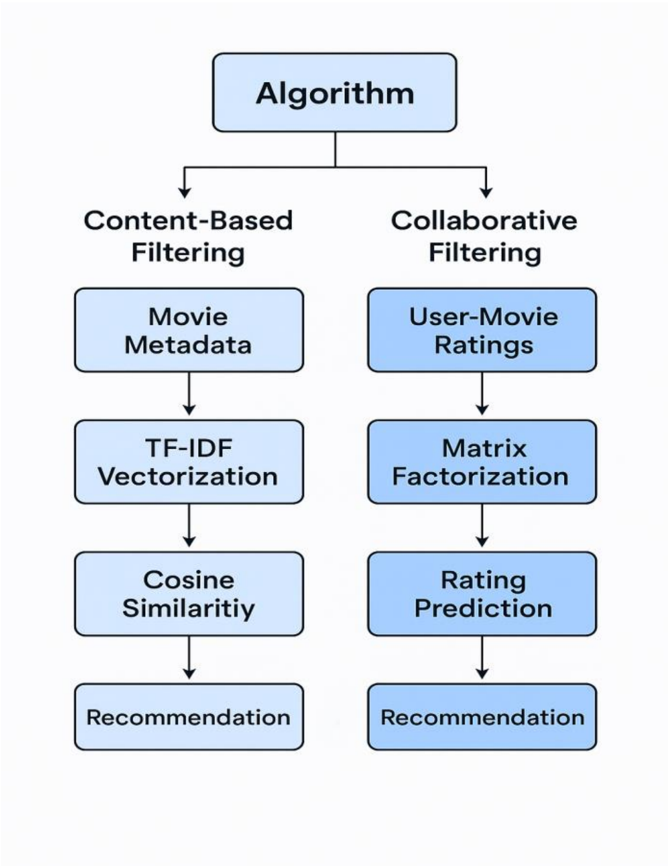

Movie Recommendation System
Movie Metadata
Collaborative Filtering
TF-IDF Vectorization
SVD Decomposision
Cosine Similarity
Recommendation
User

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04880
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

2.2 Algorithm:

Technologies Used:

- TF-IDF & Cosine Similarity: scikit-learn
- Matrix Factorization (SVD): scikit-learn or Surprise library
- Data Handling: pandas, numpy
- Interface: streamlit



2.3 Techniques:

Content-Based Filtering:

- **TF-IDF Vectorization** – Converts movie overviews into feature vectors.
- **Cosine Similarity** – Measures similarity between movie descriptions.
- **Text Cleaning** – Preprocessing like stopword removal and tokenization.[16]

Collaborative Filtering:

- **SVD (Singular Value Decomposition)** – Factorizes user-movie matrix to predict missing ratings.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04880
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

- **User-Item Matrix** – Represents ratings given by users to movies.
- **Predicted Rating = Dot Product** of user and movie feature vectors.[7]

Other Techniques:

- **Caching (@st.cache_data)** – Speeds up loading and similarity calculations.
- **Pandas, NumPy** – For data handling and computation.
- **Scikit-learn** – For TF-IDF, cosine similarity, and SVD implementation.

2.4 Tools:

- Python – Main programming language.
- Streamlit – Builds the interactive web interface.
- Pandas – For data handling and preprocessing.
- NumPy – For numerical and matrix operations.
- Scikit-learn – Used for:
  - TF-IDF Vectorizer (Content-Based)
  - Cosine Similarity
  - SVD (Collaborative Filtering)
- CSV Files – Movie metadata and user ratings as datasets.[10]
- @st.cache_data – Speeds up app by caching computations.

2.5 Methods:

Content-Based Filtering:

- Uses movie overviews and genres.
- Steps:
  1. Preprocess text (clean, lowercase, etc.)
  2. Convert to TF-IDF vectors.
  3. Compute cosine similarity.
  4. Recommend similar movies.

◆ **Collaborative Filtering:**

- Based on user ratings.
- Steps:
  1. Create user-movie rating matrix.
  2. Apply SVD (matrix factorization).
  3. Predict unseen ratings.
  4. Recommend top-rated movies.

◆ **Web Integration:**

- Built with Streamlit.
- Takes movie title or user/movie ID as input.

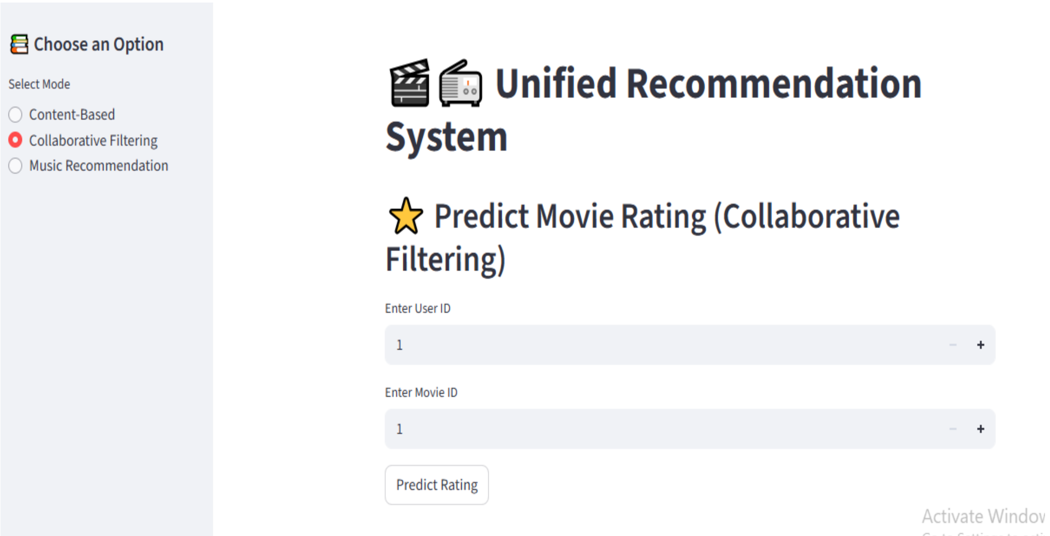
International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04880
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

- Displays recommendations or predicted ratings instantly.

III. METHODOLOGY

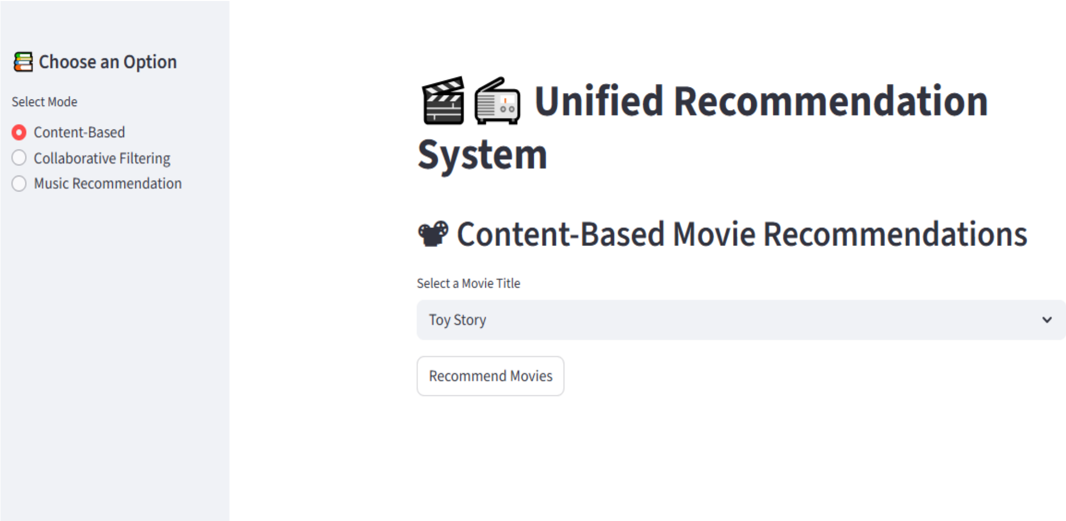
3.1 Input:

In Collaborative Filtering, the input consists of a user ID and a movie ID, which identify a specific user and movie in the system. [6]Using the user's historical ratings and patterns of similar users, the system predicts how much the selected user would like the chosen movie. The output is a numerical predicted rating, often on a scale of 0 to 5. For example, if User 1 selects Movie ID 1,[15] the system might output a rating of 2.79, indicating a moderate preference. This rating helps in recommending movies that the user is more likely to enjoy. It is widely used in platforms like Netflix and Amazon Prime.



In Content-Based Filtering , the input is a movie title chosen by the user as a reference. The system analyzes the movie's features, such as genre, actors, and storyline, and finds similar movies with matching characteristics. The output is a ranked list of similar movie titles based on content similarity. For example, choosing "Grumpier Old Men" may produce recommendations like "An Extremely Goofy Movie" and "Rushmore." This helps users discover movies similar to the ones they already like. [20] It is ideal for personalized content exploration.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04880
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

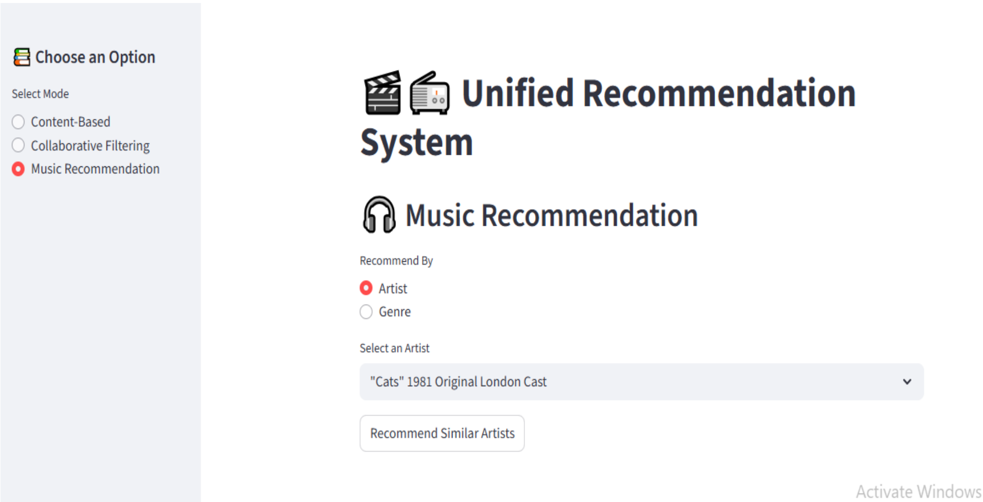


Music Recommendation:

In this unified recommendation system, the **music recommendation** module is designed to suggest **similar artists** or **similar genres** based on a user's selection. The recommendations are generated using **content-based filtering** techniques applied to numerical audio features (e.g., danceability, energy, loudness, tempo) extracted from the dataset.

The system works on **two levels**:

1. **Artist-based recommendations** – Finds artists that have similar musical characteristics to the chosen artist.
2. **Genre-based recommendations** – Finds genres that share similar aggregated audio features with the chosen genre.



3.2 **Method of Process:**

Data Collection

- Collect movie metadata and user ratings from datasets like MovieLens.

Data Preprocessing

- Clean data, handle missing values, merge movie and rating datasets.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04880
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Feature Extraction

- Use TF-IDF or CountVectorizer on genres, tags, or descriptions (for content-based).

Similarity Calculation

- Compute cosine similarity between movies (content-based) or users/items (collaborative).[23]

Model Building

- Content-Based: Recommend similar movies using similarity matrix.
- Collaborative: Train model (e.g., SVD) on user-item matrix to predict ratings.

Recommendation Generation

- Recommend top-N movies based on similarity scores or predicted ratings.

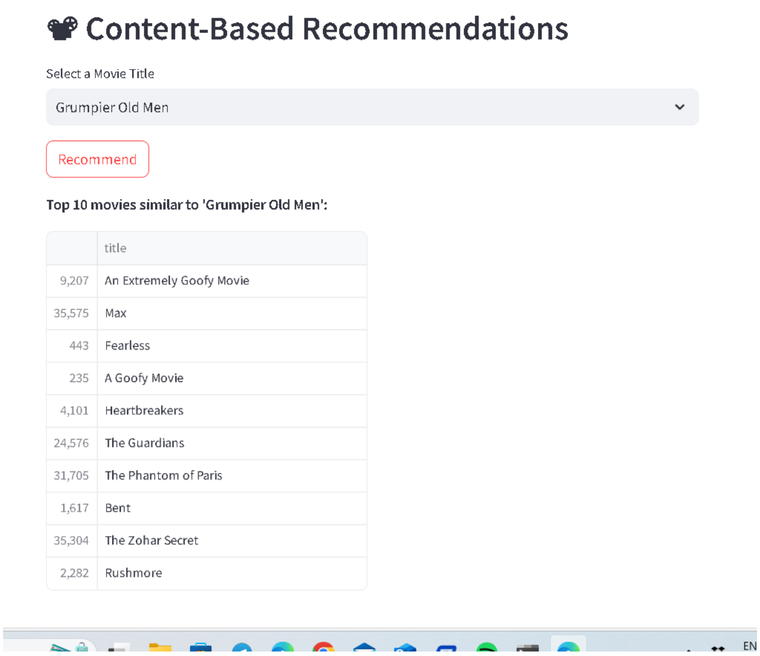
Evaluation

- Use RMSE, Precision, recall to measure performance.

Deployment (Optional)

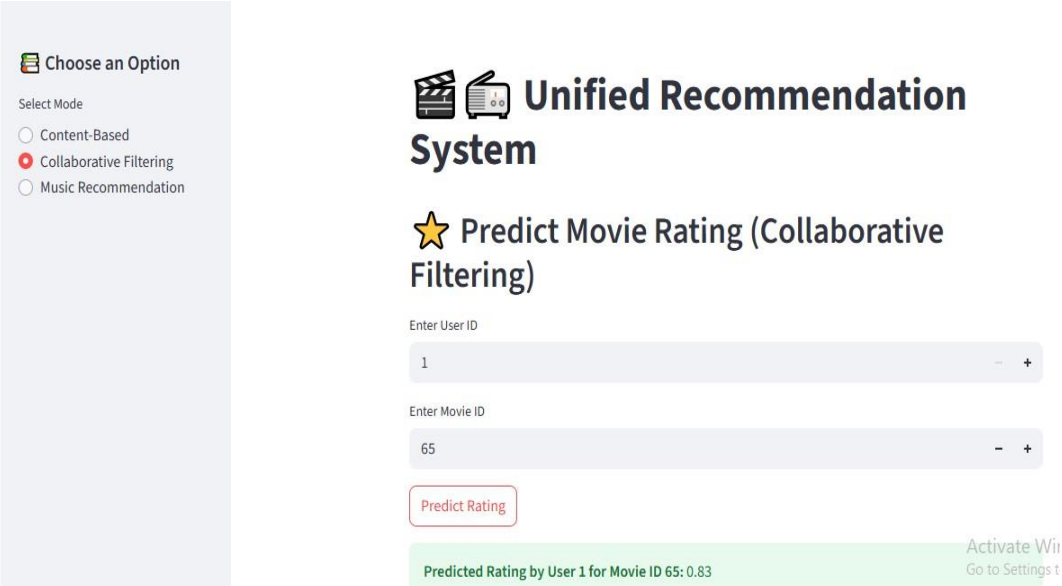
- Deploy using a web app (Flask, Streamlit) for user interaction.

3.3 Output:



V. RESULTS:

The Movie Recommendation System was successfully implemented using both Content-Based Filtering and Collaborative Filtering techniques. The system analyzed user preferences, movie metadata, and user-item rating patterns to generate personalized movie suggestions.**[19]**



V. DISCUSSIONS:

The development of the Movie Recommendation System revealed several important insights regarding the behavior, performance, and applicability of different recommendation techniques. This approach provided accurate recommendations for users with well-defined preferences.It relied heavily on movie metadata (e.g., genres, plot, director), [16]which made it effective in recommending movies similar in content. However, it lacked diversity in recommendations, often suggesting movies that were too similar to each other.A significant limitation was its inability to recommend new types of movies outside a user's known preferences**.**

VI. CONCLUSION:

In this project, a Movie Recommendation System was successfully developed using two primary machine learning techniques: Content-Based Filtering and Collaborative Filtering. The system aimed to enhance user experience by providing personalized movie suggestions based on user preferences, movie metadata, and rating patterns.The content-based filtering approach proved effective in recommending movies with similar characteristics, making it suitable for users with limited rating history. On the other hand, collaborative filtering utilized the collective behavior of users to generate meaningful recommendations, demonstrating its strength in capturing hidden patterns in user interactions**.**

VII. FUTURE SCOPE:

While the current system effectively provides personalized movie recommendations using Content-Based and Collaborative Filtering techniques, there are several opportunities to enhance and expand its capabilities in the future With the continuous advancement of data science, artificial intelligence, and user behavior modeling, movie recommendation systems can evolve into highly intelligent, context-aware platforms that significantly enhance user engagement and satisfaction.

VIII. ACKNOWLEDGEMENT:

Miss P. Bindhu priya working as an Assistant professor in MCA in sanketika vidya parishad engineering college, Visakhapatnam, ap with 2.5 yrs teaching experience and member of IAENG, accredited by Naac with her areas of interests in C, Data warehouse and data mining, Design and analysis of algorithm, python, software engineering

Mogalathurthi Bharath Raju is pursuing her final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Machine learning Mogalathurthi Bharath Raju has taken up her PG project on Movie Recommendation system content and collaborative filtering using mechine learning and published the paper in connection to the project under the guidance of PEETHALA BINDHU PRIYA, Assistant Professor, Training and Placement officer, SVPEC.

REFERENCES

1. Movie recommendation based on ALS collaborative filtering recommendation algorithm with deep learning model
   https://www.sciencedirect.com/science/article/abs/pii/S1875952124000831
2. Collaborative filtering and deep learning-based recommendation system for cold start items
   https://www.sciencedirect.com/science/article/abs/pii/S0957417416305309
3. An Efficient Deep Learning Approach for Collaborative Filtering Recommender System
   https://www.sciencedirect.com/science/article/pii/S1877050920310619
4. An Efficient Deep Learning Approach for Collaborative Filtering Recommender System
   https://www.sciencedirect.com/science/article/pii/S1877050920310619
5. A comprehensive analysis on movie recommendation system employing collaborative filtering
   https://link.springer.com/article/10.1007/s11042-021-10965-2
6. Building Accurate and Practical Recommender System Algorithms Using Machine Learning Classifier and Collaborative Filtering
   https://link.springer.com/article/10.1007/s13369-016-2410-1
7. Movie recommendation system using machine learning
   https://www.riejournal.com/article_106395.html
8. A Novel Movie Recommendation System Based on Collaborative Filtering and Neural Networks
   https://link.springer.com/chapter/10.1007/978-3-030-15032-7_75
9. Comparative study of recommender system approaches and movie recommendation using collaborative filtering
   https://link.springer.com/article/10.1007/s13198-021-01087-x
10. Collaborative Filtering with Temporal Features for Movie Recommendation System
   https://www.sciencedirect.com/science/article/pii/S1877050923001151

11. A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System https://iopscience.iop.org/article/10.1088/1742-6596/1000/1/012101/meta

12. A machine learning based recommendation system using different filtering techniques https://pubs.aip.org/aip/acp/article-abstract/2512/1/020034/3052418/A-machine-learning-based-recommendation-system
13. A Collaborative Filtering Movies Recommendation System based on Graph Neural Network https://www.sciencedirect.com/science/article/pii/S1877050923005938
14. An improved collaborative movie recommendation system using computational intelligence https://www.sciencedirect.com/science/article/abs/pii/S1045926X14000901
15. Deep Learning Based Collaborative Filtering Recommendation System https://www.sciencedirect.com/science/article/pii/S1877050925016035
16. integrating collaborative filtering and content-based methods for enhanced movie recommendations https://link.springer.com/article/10.1007/s11042-025-20895-y
17. Movie Recommender System Using Machine Learning Algorithms https://iopscience.iop.org/article/10.1088/1742-6596/1916/1/012052/meta

18. Content based movie recommendation system

https://www.riejournal.com/article_121501.html

19. A Movie Recommender System Based on Topic Modeling using Machine Learning Methods https://ijwr.usc.ac.ir/article_164092.html

20. Efficient machine learning model for movie recommender systems using multi-cloud environment

https://link.springer.com/article/10.1007/s11036-019-01387-4

21.Machine learning based recommendation system on movie reviews using KNN classifiers https://link.springer.com/article/10.1007/s11042-023-16056-8

22. A survey of active learning in collaborative filtering recommender systems

Author links open overlay panel

https://www.sciencedirect.com/science/article/abs/pii/S1574013715300150

23. Hybrid collaborative filtering using matrix factorization and XG Boost for movie recommendation

https://www.sciencedirect.com/science/article/abs/pii/S0920548924000163