International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

A Content-Based Movie Recommendation System Using Machine Learning Techniques

Nishant Singh, Sudhanshu Kumar,Shushant Mani Tripathi, Manisha Pundir
Department of Computer Science & Engineering
Noida Institute of Engineering & Technology
Greater Noida, India

Abstract— With the rapid growth of digital streaming platforms, users are exposed to a vast amount of movie content, making it difficult to identify relevant choices. This paper presents a Content-Based Movie Recommendation System that suggests movies based on their inherent features such as genre, cast, and keywords. The proposed system utilizes Machine Learning techniques, including TF-IDF (Term Frequency–Inverse Document Frequency) or Count Vectorization for feature extraction and Cosine Similarity for measuring similarity between movies. Unlike collaborative filtering methods, the system does not rely on user interaction data, thereby effectively addressing the cold start problem for new users. The model processes a structured movie dataset, converts textual data into numerical vectors, and generates recommendations based on similarity scores. The system is implemented using Python and deployed using Streamlit, providing an interactive and user-friendly interface. Experimental results demonstrate that the proposed system can efficiently generate accurate and relevant movie recommendations in real time. This approach highlights the effectiveness of content-based filtering techniques in enhancing user experience and improving content discovery in modern digital platforms.

Keywords—Movie Recommendation System, Content-Based Filtering, Machine Learning, TF-IDF, Cosine Similarity, Data Preprocessing, Information Retrieval, Streamlit, Artificial Intelligence.

I. INTRODUCTION

The rapid growth of digital media and online streaming platforms has significantly increased the availability of movie content, creating challenges for users in identifying relevant and personalized choices. With thousands of movies available across multiple genres and platforms, users often experience information overload, leading to inefficient content discovery. To address this issue, recommendation systems have emerged as essential tools for filtering and suggesting relevant items based on user preferences or content features.

This paper focuses on the development of a Content-Based Movie Recommendation System that recommends movies by analyzing their inherent attributes such as genre, cast, and keywords. Unlike collaborative filtering approaches, which rely on user interaction data, content-based methods utilize item features to generate recommendations, making them effective in handling the cold start problem.

The proposed system employs machine learning techniques, including TF-IDF or Count Vectorization for feature extraction and Cosine Similarity for measuring similarity between movies. The system is designed to provide accurate, real-time recommendations through an interactive interface. This approach demonstrates the practical application of artificial intelligence in enhancing user experience and improving content discovery in modern entertainment platforms.

II. LITERATURE REVIEW

Recommendation systems have been widely studied as an effective solution for filtering large volumes of information and providing personalized suggestions. In the domain of movie recommendation, various approaches have been proposed, including content-based filtering, collaborative filtering, and hybrid methods. Each technique offers unique advantages while also presenting certain limitations.

Content-based filtering is one of the earliest and most commonly used approaches. It recommends items based on their attributes and the similarity between them. Several studies have demonstrated that content-based systems can effectively utilize features such as genre, cast, and textual descriptions to generate personalized recommendations. Techniques like TF-IDF (Term Frequency–Inverse Document Frequency) and Cosine Similarity are frequently employed to convert textual data into numerical representations and measure similarity between items. This approach is particularly useful when user interaction data is limited or unavailable. However, it may suffer from limited diversity, as it

tends to recommend items similar to those already selected.
Collaborative filtering, on the other hand, relies on user behavior, such as ratings and viewing history, to identify patterns and recommend items. Research indicates that this method can provide highly accurate recommendations when sufficient user data is available. However, it faces challenges such as the cold start problem, data sparsity, and scalability issues, especially in systems with a large number of users and items.

To overcome the limitations of individual approaches, hybrid recommendation systems have been developed by combining content-based and collaborative techniques. These systems aim to improve recommendation accuracy and diversity. While hybrid models show better performance, they often require more complex implementation and higher computational resources.

Recent advancements in machine learning and natural language processing have further enhanced recommendation systems by enabling better feature extraction and understanding of user preferences. Despite these improvements, there remains a need for simple, efficient, and scalable systems that can provide accurate recommendations without heavy dependency on user data.

Based on the analysis of existing research, content-based filtering remains a suitable approach for this project due to its simplicity, effectiveness, and ability to handle scenarios with limited user information.

III. METHODOLOGY

The proposed system follows a content-based filtering approach that utilizes machine learning techniques to recommend movies based on their inherent features such as genre, cast, and keywords. The system is designed using a modular architecture, where each component performs a specific function, ensuring scalability, efficiency, and ease of maintenance.

A. System Overview

The overall system consists of four major components: the dataset module, feature extraction module, similarity computation module, and user interface. The dataset module provides structured movie data, while the feature extraction module converts textual data into numerical vectors using techniques such as TF-IDF or Count Vectorization. The similarity computation module calculates the similarity between movies using Cosine Similarity, and the user interface, developed using Streamlit, enables user interaction.

As shown in Fig. 1, the system follows a pipeline-based architecture where movie data is processed, transformed, and used to generate recommendations. The system focuses on analyzing content features rather than relying on user behavior, making it effective in handling new users and items.

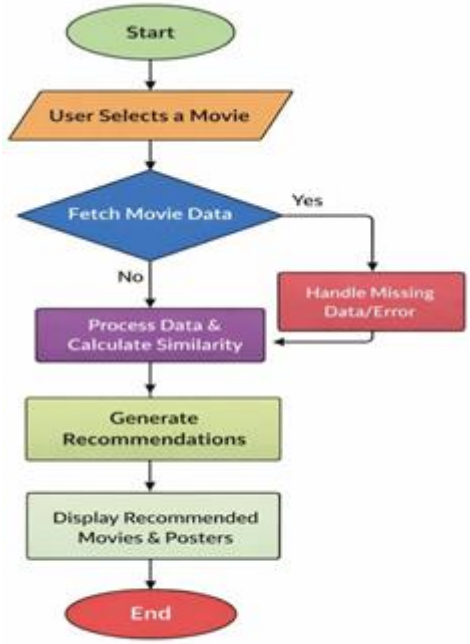

Fig. 1. System Architecture of the Proposed Movie Recommendation System

B. System Workflow

The system operates through a structured pipeline that begins with user input and ends with recommendation output. Initially, the user selects a movie through the interface. The system retrieves the corresponding movie data from the dataset and processes it using precomputed feature vectors.

The textual features of movies are converted into numerical vectors using TF-IDF or Count Vectorization. These vectors represent movies in a high-dimensional space. The system then applies Cosine Similarity to compute similarity scores between the selected movie and all other movies in the dataset.

Based on these scores, the system identifies the top similar movies and generates recommendations. The results are displayed along with movie titles and posters, ensuring an enhanced user experience.

C. Feature Extraction Module

The system operates through a structured pipeline that begins with user input and ends with recommendation output. Initially, the user selects a movie through the interface. The system retrieves the corresponding movie data from the dataset and processes it using precomputed feature vectors.

The textual features of movies are converted into numerical vectors using TF-IDF or Count Vectorization. These vectors represent movies in a high-dimensional space. The system then applies Cosine Similarity to compute similarity scores between the selected movie and all other movies in the dataset.

Based on these scores, the system identifies the top similar movies and generates recommendations. The results are displayed along with movie titles and posters, ensuring an enhanced user experience.

D. Similarity Computation

The similarity computation module uses Cosine Similarity to measure the closeness between movie vectors. This metric calculates the cosine of the angle between two vectors, indicating how similar the movies are based on their features.
Movies with higher similarity scores are considered more relevant and are selected for recommendation. This approach ensures that recommended movies closely match the characteristics of the selected input.

E. System Integration

The system is implemented using Python, with libraries such as Pandas, NumPy, and Scikit-learn handling data processing and machine learning tasks. The user interface is developed using Streamlit, providing an interactive and user-friendly environment.

Preprocessed data and similarity matrices are stored using Pickle, ensuring faster execution during runtime. The modular design allows seamless interaction between components, enabling real-time recommendation generation.

The integration of these modules results in an efficient and scalable recommendation system that enhances content discovery and user experience.

IV. RESULTS AND DISCUSSION

The proposed Content-Based Movie Recommendation System was successfully implemented and evaluated using a movie dataset containing features such as genre, cast, and keywords. The system was able to generate relevant and accurate movie recommendations based on the selected input. By applying TF-IDF/Count Vectorization and Cosine Similarity, the model effectively identified similarities between movies and produced meaningful suggestions.

The results demonstrate that the system can generate recommendations in real time, with minimal response delay. The use of preprocessed data and stored similarity matrices significantly improved performance and reduced computational overhead during execution. The integration of a Streamlit-based interface further enhanced usability by providing a simple and interactive platform for users to select movies and view recommendations along with posters.

From an accuracy perspective, the system performed well in recommending movies that share similar attributes with the selected movie. For example, movies with similar genres, themes, or cast members were frequently recommended, indicating that the feature extraction and similarity computation methods were effective. However, the system tends to recommend movies with closely related content, which may limit diversity in recommendations.

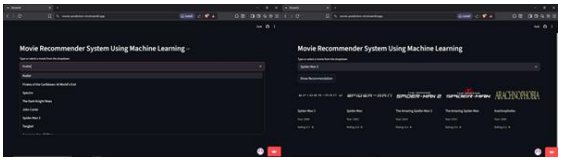

Fig. 2. User Interface showing Movie Recommendation System

One of the key advantages observed is that the system does not rely on user data, making it suitable for scenarios where user interaction history is unavailable. This effectively addresses the cold start problem. However, the absence of user preferences may reduce personalization compared to collaborative or hybrid approaches.

Overall, the system demonstrates a balance between accuracy, efficiency, and simplicity, making it a practical solution for content-based recommendation. Future improvements can focus on enhancing diversity and personalization by

integrating additional techniques such as hybrid recommendation models or user feedback mechanisms.

V. CONCLUSION

This paper presented a Content-Based Movie Recommendation System that utilizes machine learning techniques to provide relevant and personalized movie suggestions. By analyzing movie features such as genre, cast, and keywords, and applying methods like TF-IDF/Count Vectorization and Cosine Similarity, the system effectively identifies similar movies and generates accurate recommendations.

The proposed system demonstrates efficient real-time performance and does not rely on user interaction data, making it suitable for addressing the cold start problem. The integration of a Streamlit-based interface ensures ease of use and enhances user experience.

Overall, the system highlights the effectiveness of content-based filtering in recommendation systems. Future work can focus on incorporating hybrid approaches and user feedback to further improve recommendation accuracy and diversity.

REFERENCES

1. Scikit-learn, —Machine Learning in Python,‖ Available: https://scikit-learn.org
2. Pandas, —Python Data Analysis Library,‖ Available: https://pandas.pydata.org
3. NumPy, —Numerical Computing in Python,‖ Available: https://numpy.org
4. Streamlit, —Streamlit Documentation,‖ Available: https://streamlit.io
5. TMDB, —The Movie Database,‖ Available: https://www.themoviedb.org
6. C. C. Aggarwal, Recommender Systems: The Textbook. Springer, 2016.
7. F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook. Springer, 2015.
8. G. Salton and C. Buckley, —Term-weighting approaches in automatic text retrieval,‖ Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988.