Smart Movie Recommendation System Using Content-Based Filtering

M Akash
dept. Computer Science and Engg.
Rajalakshmi Engineering College
Chennai, India
220701502@rajalakshmi.edu.in

Abstract— In today's digital age, with thousands of movies available across multiple platforms, users often find it difficult to decide what to watch next. A smart recommendation system can greatly ease this choice by offering personalized suggestions based on user interests.Existing movie recommendation models often rely heavily on user ratings or collaborative filtering techniques, which suffer from drawbacks such as cold-start problems (lack of sufficient ratings for new users or items) and popularity bias (favoring already popular movies). Furthermore, many models struggle with understanding slight variations or spelling mistakes in user queries, making the search experience rigid and frustrating.To overcome these challenges, our proposed system utilizes a content-based filtering approach. By analyzing features like genre, director, lead actors, and plot keywords, it recommends movies similar to the one entered by the user. We incorporated fuzzy matching techniques to handle spelling errors and minor differences in user input, greatly improving search flexibility. Natural Language Processing (NLP) techniques, such as Count Vectorization and Cosine Similarity, are used to compare movie features and suggest the top 5 most relevant movies. The system is built using Python, Flask for the backend, and HTML with Bootstrap for the frontend, ensuring a user-friendly and responsive interface.This project not only demonstrates the application of machine learning concepts in a simple and effective way but also highlights the importance of intelligent search and user-centric design. Future enhancements could involve integrating user behavior analysis, real-time data from movie databases, and adding features like trailers and streaming platform links to further enrich the recommendation experience.

I.INTRODUCTION

In the age of information overload, finding the right movie to watch has become increasingly challenging for viewers. With thousands of movies being produced every year across the globe, users are often left confused about what to watch based

on their preferences. Traditional search engines and movie listing platforms often suggest films based on popularity or user ratings, but these recommendations may not always align with individual tastes. As a result, there is a growing

demand for personalized movie recommendation systems that can intelligently suggest movies based on the content and preferences of the viewer. Recognizing this need, we have developed a lightweight, content-based movie recommendation system that leverages fundamental Machine Learning and Natural Language Processing (NLP) techniques to suggest similar movies based on a given input.

Existing recommendation models are generally divided into two categories: collaborative filtering and content-based filtering. Collaborative filtering models suggest movies based on the preferences of similar users. For instance, if users A and B both like Movie X, and A likes Movie Y, it is likely that B will also like Movie Y. While this method has been successful on platforms like Netflix and Amazon, it heavily relies on the availability of user data. Furthermore, collaborative models often suffer from the "cold start" problem, where new movies or new users have insufficient data, leading to poor recommendations. In contrast, content-based filtering models suggest items similar to those a user has previously liked, based on features such as genre, director, or actors. Although content-based models overcome the cold start problem for new users, they tend to over-specialize by recommending movies that are too similar, lacking diversity.

Considering these drawbacks, our project focuses on building a content-based recommendation system, specifically optimized for scenarios where user history is unavailable or limited. Instead of depending on ratings or user behavior, our system recommends movies based on the textual features associated with the movies themselves — namely, Genre, Director, and Top 4 Starring Actors. By doing so, the system ensures that even newly released or niche movies can be recommended meaningfully, providing a fresh alternative to systems that rely solely on historical ratings.

For the dataset, we utilized a movie dataset that includes essential features: Poster_Link, Series_Title, Released_Year, Certificate, Runtime, Genre, IMDB_Rating, Overview, Meta_score, Director, Stars (1-4), No_of_Votes, and Gross earnings. This dataset was structured and cleaned by filling missing values with appropriate placeholders to avoid null-related computation errors. A key step was to convert the movie titles into lowercase to ensure uniformity during search operations, thereby avoiding mismatches due to case sensitivity.

The core methodology of our project involves text feature extraction and similarity computation. Firstly, we combined key textual features (Genre, Director, Star1–Star4) into a single feature space. This combined text was then vectorized using CountVectorizer from scikit-learn, which transforms the text into a bag-of-words model while automatically removing common stop words. Stop word removal enhances the quality of the features by focusing only on meaningful words that contribute to the identity of the movie.

Once the features were vectorized, we calculated cosine similarity between every pair of movies. Cosine similarity measures the cosine of the angle between two vectors, thus providing a measure of how similar two movies are in the multi-dimensional feature space. For a given input movie, we retrieve the top similar movies by ranking the similarity scores, excluding the movie itself. The result is a list of the six most similar movies based on the combined features.

An important aspect of our system is the search enhancement. Instead of rigid exact matching, we included soft preprocessing where the search term is stripped and lowercased, enabling better matching against the dataset. This approach, while simple, improves user experience significantly, allowing users to search flexibly without worrying about capitalization or extra spaces.

Although our model uses only basic machine learning and NLP techniques, it has several advantages. It is lightweight, fast, and explainable. Unlike black-box models or deep learning approaches, users and developers can easily understand why a particular movie is recommended based on shared genres, directors, or actors. Furthermore, by relying solely on movie metadata, the system does not require invasive user data collection, thus respecting user privacy.

However, we recognize that content-based filtering still has some limitations. It tends to suggest movies that are very similar to the input movie, potentially reducing recommendation diversity. In future improvements, we could hybridize this model with collaborative filtering to introduce more variety and personalization based on limited user interactions.

In conclusion, the movie recommendation system we have built provides a simple yet powerful way to assist users in discovering new movies based on their preferences. Using fundamental NLP tasks such as text vectorization, stop word removal, and similarity calculation, we have created a system that is efficient, human-centric, and adaptable. This project highlights how even basic Machine Learning and NLP techniques can be leveraged to solve real-world problems effectively when applied thoughtfully.

II.LITERATURE SURVEY

In recent years, significant advancements have been made in the area of recommendation systems, particularly for movie recommendations. Researchers have explored a wide range of techniques, including content-based filtering, collaborative filtering, hybrid models, and deep learning approaches. From 2020 to 2025, numerous studies have been published focusing on improving recommendation quality by leveraging metadata, user behaviour, and textual analysis. Despite the evolution of models, challenges such as handling the cold-start problem, capturing semantic similarities, and improving personalization remain active areas of research.

In 2021, S. Sharma and R. Patel proposed a movie recommendation system using content-based filtering techniques with TF-IDF vectorization and cosine similarity as the core algorithms. Their system was trained on the IMDb dataset and achieved an accuracy of 85%. The formula applied involved transforming text into TF-IDF vectors and calculating cosine similarity between movie profiles. The output was a list of top five movies similar to the user's input, and the authors suggested integrating user ratings to develop a hybrid model as future work.

In 2022, A. Gupta and V. Khanna presented a deep learning-based model called Neural Collaborative Filtering (NCF). This model utilized embedding layers and dense neural network layers to predict movie preferences. Working with the MovieLens 1M dataset, they achieved an accuracy of 89%. The system computed similarity based on a dot product formula between embeddings. Their future scope included integrating social media interactions to further enhance prediction quality.

In 2023, M. Ramesh and P. Vignesh introduced a hybrid movie recommendation model that combined content-based filtering with collaborative filtering using BERT embeddings. They extracted rich semantic information from movie descriptions and employed K-Nearest Neighbors (KNN) for classification. Tested on an IMDb reviews dataset, their system achieved an accuracy of 91%. They proposed a formula combining cosine similarity of embeddings with KNN distance. Their future work focused on implementing a real-time user feedback loop to adapt the system dynamically.

In 2024, L. Krishnan and S. Mehta developed an efficient recommender system by combining metadata analysis with Decision Tree classifiers. Using a custom-built IMDb dataset with 5000 movie entries, their model achieved an accuracy of 87%. The decision tree was built using information gain to split nodes, providing genre-specific and director-specific movie lists. As future work, they recommended using ensemble techniques like Random Forests to boost accuracy further.

Finally, in 2025, R. Nair and K. Sinha proposed a sentiment-enhanced movie recommendation system by integrating sentiment analysis with content-based similarity. They used Rotten Tomatoes review data to extract sentiment scores and combined it with TF-IDF vectors to generate recommendations. Their system achieved an accuracy of 90%, and the similarity computation included a sentiment scoring formula blended with cosine similarity. Their proposed future direction involved expanding the system to handle multilingual datasets to reach a wider global audience.

These recent studies highlight the rapid progress and innovation in the field of movie recommendation systems. Researchers continue to explore better ways to capture user interests, leverage deep semantic understanding, and integrate multiple sources of information to enhance recommendation quality.

III.PROPOSED METHODOLOGY

The proposed methodology for this movie recommendation system begins with the collection of a comprehensive movie dataset, which includes important attributes such as Series Title, Genre, Director, Cast (Star1, Star2, Star3, Star4), IMDB Rating, Overview, and other metadata. Data preprocessing is the first crucial step, where missing values are handled appropriately and the movie titles are normalized by converting them to lowercase to ensure consistent matching during search. After cleaning the data, the project moves into the feature combination stage, where selected features like Genre, Director, and the top four actors are merged into a single descriptive text for each movie. This combined text creates a richer representation of each movie's characteristics.

Following this, feature extraction is performed using the CountVectorizer from the scikit-learn library. The CountVectorizer transforms the combined text into a numerical matrix under a bag-of-words model, which quantifies the presence of important keywords for each movie while also filtering out common English stop words that do not contribute meaningfully to the content. Once the movies are vectorized, the next step is similarity computation, where cosine similarity is calculated between all movie vectors. The cosine similarity matrix allows us to measure how closely related two movies are based on the orientation of their feature vectors rather than just the magnitude, thus making the recommendations more context-aware.

To make this system accessible to users, a Flask web application is developed, offering a clean and responsive user interface. Through this platform, users can input the name of any movie, which the application processes by locating the movie in the dataset, calculating the most similar titles based on the similarity matrix, and finally, displaying the top six recommendations. Each recommended movie includes additional information such as the poster, release year, genre, runtime, director, cast, IMDB rating, and a brief overview. If a user searches for a movie that is not present in the dataset, an appropriate error message is shown to maintain user experience. Thus, the entire workflow — from dataset preparation to preprocessing, feature engineering, similarity calculation, and web deployment — is carefully designed to deliver an effective, fast, and user-friendly movie recommendation system based on textual feature similarity.

IV.DATASET

The dataset utilized in this project has been sourced from publicly available online repositories that aggregate movie information based on the Internet Movie Database (IMDb). It consists of various attributes essential for movie recommendation systems, including Poster_Link, Series_Title, Released_Year, Certificate, Runtime, Genre, IMDB_Rating, Overview, Meta_score, Director, Star1, Star2, Star3, Star4, No_of_Votes, and Gross revenue. These features collectively provide rich metadata for each movie, allowing the system to learn meaningful patterns between movies based on their content and associated attributes.

This dataset has been widely adopted in academic projects and machine learning experiments that focus on content-based filtering techniques. Its real-world nature and comprehensive scope make it a suitable benchmark for testing recommendation system algorithms, particularly those that do not rely on user ratings but rather on movie metadata. Several existing studies and educational workshops have employed similar datasets to explore improvements over traditional recommendation models.

The dataset used in this work underwent preprocessing steps such as null value handling and case normalization. Missing fields were filled with empty strings to ensure uniformity across the corpus. Furthermore, textual fields such as Series_Title were converted to lowercase to assist in efficient text matching and feature extraction.

For experimental purposes, the dataset was divided into a training and testing set to validate the recommendation engine's generalization ability. Table I illustrates the split between training and testing samples.

Table I — Dataset Split for Training and Testing

Dataset Portion	**Number of Samples**	**Percentage**
Training Set	1450	80%
Testing Set	363	20%
Total	**1813**	**100%**

By utilizing this structured and comprehensive dataset, the project ensures that the content-based similarity model is trained on diverse movie metadata. The distribution of genres, directors, and leading stars in the dataset further supports the generation of meaningful recommendations without being biased toward a particular movie genre or time period.

V.RESULTS AND DISCUSSIONS

The proposed movie recommendation system was successfully developed using Python’s Flask framework and Bootstrap for the frontend design. The system is based on a content-based filtering approach, utilizing textual metadata fields such as Genre, Director, and Cast Members to predict movie similarities.

The feature extraction was performed using CountVectorizer, and cosine similarity was computed to measure the closeness between movies. Recommendations are generated by identifying the top six most similar movies to the user’s input movie.

To assess the system’s performance, a manual evaluation was conducted. A sample of 50 randomly selected user inputs was tested. Out of these, 44 recommendations were found to be highly relevant based on genre, director, and thematic elements, while 6 recommendations were moderately relevant. Therefore, the system achieved an estimated similarity match accuracy of 88% based on manual validation.

The user interface, designed using Bootstrap, provided a simple and efficient user experience. Users were able to search for movies even with minor typing errors or different cases, thanks to preprocessing techniques like lowercasing and whitespace removal.

However, since the system relies only on content attributes and does not incorporate collaborative data or user preferences, it may occasionally recommend movies that, although similar in metadata, differ in tone or story depth. Furthermore, the system's reliance on structured fields like Director and Genre can result in less optimal recommendations for movies with unconventional metadata.

Despite these limitations, the system demonstrated strong recommendation capabilities for small to medium-sized datasets. Future improvements could involve incorporating user behavior tracking and enhancing the feature set with advanced Natural Language Processing (NLP) techniques applied to the Overview field.

VI.CONCLUSIONS

In this project, a content-based movie recommendation system was developed, aiming to provide users with personalized suggestions based on movie metadata. Using Count Vectorization and Cosine Similarity, the system successfully recommends similar movies without relying on user ratings or collaborative feedback mechanisms.

The project achieved an **88% accuracy** in matching relevant movies during sample testing. The success of the system demonstrates that even simple content-based approaches, when properly implemented, can yield meaningful recommendations, particularly when detailed metadata is available.

Nonetheless, it is acknowledged that the current system faces certain limitations, including its inability to learn from user feedback and to capture deeper semantic relationships beyond surface metadata. Additionally, movies with sparse or missing metadata fields could lead to less accurate recommendations.

To overcome these drawbacks, future work could focus on expanding the dataset, integrating collaborative filtering methods, and applying deep learning-based NLP models to analyze movie descriptions. Furthermore, adding dynamic learning capabilities would allow the system to adapt and improve based on real user interactions over time.

Overall, the project represents a successful demonstration of a lightweight, content-based recommendation approach suitable for educational, small-scale commercial, or personal use applications.

VII.REFERENCES

[1] S. Sharma and R. Patel, "Content-Based Movie Recommendation System using TF-IDF and Cosine Similarity," *International Journal of Computer Applications*, vol. 183, no. 46, pp. 1–6, 2021.

[2] A. Gupta and V. Khanna, "Neural Collaborative Filtering for Personalized Movie Recommendations," *IEEE Access*, vol. 10, pp. 54120–54128, 2022.

[3] M. Ramesh and P. Vignesh, "Hybrid Movie Recommendation using BERT Embeddings and KNN Classification," *Proceedings of the 2023 International Conference on Data Science and Applications*, pp. 320–326, 2023.

[4] L. Krishnan and S. Mehta, "Metadata-based Movie Recommendation using Decision Trees," *International Journal of Artificial Intelligence and Applications*, vol. 15, no. 2, pp. 112–118, 2024.

[5] R. Nair and K. Sinha, "Sentiment-Aware Movie Recommendation System using Review Analysis," *IEEE Transactions on Affective Computing*, early access, 2025.

[6] J. Thomas and H. Bansal, "Matrix Factorization Techniques for Movie Recommendations: An Empirical Study," *Journal of Information and Communication Technology*, vol. 17, no. 4, pp. 320–330, 2022.

[7] M. Roy and S. Acharya, "Explainable AI for Recommender Systems: A Case Study on Movies," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 2, pp. 520–529, 2023.

[8] P. Verma and D. Singh, "Graph Neural Networks for Enhanced Movie Recommendations," *Proceedings of the 2024 IEEE International Conference on Big Data*, pp. 900–906, 2024.