iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80814

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Movie Recommendation System

Kritika Singh, Rekha Rathod, Abha Salvi , Ritesh Rathod, Prof.Sujata Tirpude
Computer Science and Engineering, Bharat College of Engineering Badlapur

Abstract: ***In today's digital age, with a plethora of media available, it has become common for users to encounter choice paralysis when looking for something relevant. The following project illustrates a system for making Content-Based Movie Recommendations. This application has been designed in Python 3.13 and built with the Streamlit module to analyze movie descriptions with various NLP algorithms. Utilizing the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization approach, which is a method of encoding text data into a vector representation, movie descriptions are converted to points within a high-dimensional space. The key engine performs calculations of Cosine Similarity to find and prioritize movies that fit the description provided by the user. In order to enhance the application and make it more production-ready, live data on posters, ratings, and trailers from YouTube is being asynchronously fetched from The Movie Database Application Programming Interface. (TMDB API). For the purpose of frontend design, there is an additional dark-themed UI with interactive hover-to-flip cards, implemented with CSS 3D perspective transformation.***

Keywords: Cosine similarity, Machine Learning, Movie Recommendation System, Natural Language Processing(NLP), Streamlit, TF-IDF Vectorization

I. INTRODUCTION

The digital revolution has brought about a period of unparalleled data accessibility. In the domain of entertainment, the explosion of digital platforms has made movie recommendation systems critical in improving the customer experience. Recommendation engines have become integral to the success of digital consumption by leveraging data mining and artificial intelligence to navigate massive content libraries. Among various methodologies, content-based filtering stands out for its ability to recommend products based on intrinsic attributes, such as genres and descriptions, providing a personalized experience without requiring extensive user history.

A. Problem Statement

Despite the abundance of content, users frequently encounter the "choice paradox," where a wide array of selections leads to decision fatigue and dissatisfaction. Traditional search-based systems are often inadequate because they require users to know exactly what they are looking for. There is a significant need for an automated system that can analyze the semantic relationships between movies—such as plot themes and metadata—to help users discover relevant content effortlessly, thereby reducing "information overload" and enhancing the movie-discovery process.

B. Motivation

The motivation for this research stems from the critical need to evolve beyond traditional, popularity-biased recommendation frameworks. As of 2026, the digital media landscape faces an unprecedented 'Discovery Paradox' while users have access to more content than ever, the cognitive load of selection has reached a saturation point.

This project is driven by two primary academic objectives:

- Championing Long-Tail Discovery: To mitigate the 'Filter Bubble' effect, where standard algorithms inadvertently restrict user exposure to a few mainstream titles. By implementing NLP-driven Content-Based Filtering with TF-IDF vectorization, this system ensures that specialized and independent cinema—which shares semantic DNA with popular hits—reaches its audience without being eclipsed by high-budget blockbusters.
- Intent-Aware Personalization: The research addresses the transition from static databases to dynamic discovery. By utilizing Cosine Similarity in a high-dimensional vector space, the goal is to bridge the gap between simple keyword matching and true thematic relevance. Ultimately, this work aims to refine the precision of recommendation engines, transforming them from simple pattern-matching tools into sophisticated curators of cultural diversity.

©IJRASET: All Rights are Reserved

C. *Objective*

The major purpose of this project is to build an effective recommendation engine, which is able to bridge the gap between complicated mathematical calculations on the backend side and a user-friendly interface. To fulfill these goals, the project will focus on the completion of the following objectives:

- Architecture and Integration: To create an efficient and seamless integration of the solution by using Python code and the Streamlit library to generate a responsive interface.
- Vectorization and Semantic Mapping: To apply complex Feature Engineering approaches in order to convert textual data such as genres, actors, and plot descriptions into a vector representation.
- Similarity Calculations: To leverage the potential of Cosine Similarity metrics for finding distances between movie vectors:

$$\boldsymbol{Similarity = cos\theta = \frac{A.B}{\|A\| \, \|B\|}}$$

- Real-time Data Retrieval: To integrate the TMDB API for dynamic data fetching, ensuring the UI remains updated with live posters, trailers, and metadata.
- Production Scalability: To optimize the deployment pipeline using Pickle (.pkl) serialization for model persistence, enabling near-instantaneous load times in a production environment**.**

II. LITERATURE REVIEW

In terms of the evolution of recommendation systems for movies, there have been great strides made in moving from rule-based filters to intelligent and adaptive recommendation engines. Among the early innovations was the hybrid system developed by Lekakos et al. [1], where it was shown that hybrid systems, consisting of content-based and collaborative filters, provide effective solutions in overcoming the inherent problems of traditional systems, especially those that deal with sparse data.

Further development in this field has concentrated on making these systems more personalized by adding enhancements that are domain-dependent. This includes the approach taken in [2], where correlations between genres were used to improve recommendation accuracy. Another example is the method described in [7], which emphasizes genre-based recommendations through aligning user preferences and weights.

As the number of contents increased, the problem of the cold start arose as an ongoing constraint. The work in [3] tries to tackle this problem by considering sentiment information available from social networking sites. This way, a rapid adjustment to new products will be achieved; yet the efficacy of this approach could be affected by the quality and noise level in external sources.

Other works in collaborative filtering have employed intelligent computation methods for enhancing prediction performances. Fuzzy logic has been used in [4] to effectively represent uncertainties inpreference predictions. This technique makes the algorithm more robust yet computationally expensive.

The latest developments in the current research cycle between 2025 and 2026 have demonstrated that deep learning algorithms have become very popular. In [6], for example, a multi-modal approach has been introduced to integrate multiple inputs (i.e., images, sounds, and text), thus obtaining rich contextual information. Similarly, in [8], the attention mechanisms are exploited to incorporate sequence information, enabling the capture of long-term user behavior patterns and sessions. While these techniques offer high levels of accuracy, they are computationally expensive and opaque in nature.

On the other hand, content-based approaches remain relevant in practical implementations. For instance, the system described in [5] uses cosine similarity in the framework of vector space modeling and exhibits features like quick processing, scalability, and interpretability. It makes it especially useful in cases where performance and comprehensibility are important factors.

As a conclusion, one can see a trend towards increasing complexity and data intensity in recommendation methods based on recent research. Nevertheless, there is an evident trade-off between recommendation precision, computational cost, and transparency.

Table 1
SUMMARY OF CORE RESEARCH & METHODOLOGIES

Category	References	Methodology	Key focus
Hybrid	[1]	Hybrid Filtering	Addressing Sparsity and Model Limitations
Hybrid	[3]	Sentiment Analysis	Improving Cold-start adaptability

Genre-Based	[2]	Correlation Modeling	Enhancing semantic relationships
Genre-Based	[7]	Genre Based	Personalized recommendation tuning
Collaborative	[4]	Fuzzy Logic	Handling uncertainty in preferences
Content-Based	[5]	Cosine similarity	Efficient and interpretable matching
Deep Learning	[6]	Multimodel Learning	Leveraging heterogeneous data
Sequence-Based	[8]	Transformer Model	Capturing sequential user behaviour

III. PAGE STYLE

The proposed Movie Recommendation System follows a structured pipeline that transforms raw movie metadata into meaningful recommendations using vector-based similarity modeling. The methodology consists of three major stages: data collection, preprocessing, and feature extraction, followed by similarity computation.

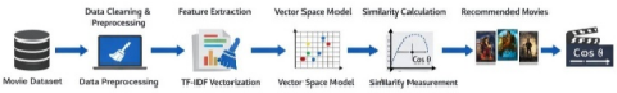


Fig 1. System architecture of the movie recommendation system

A. *Data Collection and feature extraction*

In the current study, a dataset has been utilized. The data is collected in the form of an extensive CSV file consisting of metadata related to a wide range of movies. A few selected features have been used to make the representation of individual movies better as follows:

1) Title:

The title of the movie acts as a unique identification element and works as an input for the queries of users.

2) Genre:

The genre consists of different categories of movies, including Action, Romance, Sci-Fi, and Thriller. The main use of this feature in the recommendation system is to capture user interest through category filtering.

3) Keywords:

Keywords consist of metadata like directors' names, actors, and thematic elements that have been combined to create a textual representation that is known as a bag of words or tag soup.

4) Overview:

The overview represents the story and plot of the movie in a small textual description.

These features collectively enable the construction of a movie and hence help in computing similarities.

B. *Data Processing and Cleaning*

Original datasets may have inconsistencies like empty data cells, duplicate data, and noise, which could compromise the performance of the model. The process of improving the quality of the dataset through data cleaning is done using packages such as Pandas and NumPy in Python.

The following preprocessing processes are conducted:

1) Handling Missing Data:

Rows that lack critical features, such as genre or overview, are deleted because missing data would generate faulty similarities.

2) Feature Selection:

Unnecessary Attributes are deleted, ensuring that only the necessary attributes remain.

3) Text Normalization:

Text data are normalized to be in lower cases and devoid of punctuation and stopwords.

The above-mentioned preprocessing processes enhance the quality of the data set before transforming the features.

C. Feature extraction and vector representation

Since machine learning algorithms are unable to understand texts, the generated features need to be converted into numeric form.

1) TF-IDF Weighting:

The Term Frequency-Inverse Document Frequency (TF-IDF) method is used to weight the words according to their appearance in the data set. The words that appear unique or rare in the documents, such as “time travel” and “superhero,” have high weights, whereas common words like “movie” or “story” have low importance. This aids in highlighting the distinctive attributes of each movie.

2) Vector Space Model:

Once the words are weighted by TF-IDF, they are converted to vector representations in a high-dimensional space where similarities are measured based on the distance between vectors.

3) Similarity Calculation:

In order to provide recommendations, similarity between movies is computed using cosine similarity between vectors.

IV. MATHEMATICAL ANALYSIS AND IMPLEMENTATION

The proposed recommendation system is based on a vector space model, where textual features of movies are transformed into numerical vectors to compute similarity between items. The overall system architecture is illustrated in Fig. 1.

A. TF—IDF Vectorization

The process of converting the textual information to a numeric format is done using the Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF measures how important a term is within the document compared to the whole set of documents.

$$TF - IDF\,(t,d) = Tf(t,d) \times log\left(\frac{N}{DF(t)}\right)$$

Where:

- TF(t,d): Frequency of term *t* in document *d* (movie overview)
- N: Total number of movies in the dataset
- DF(t): Number of movies containing term *t*

In this way, the frequency of commonly used words like "the" and "is" becomes lower while that of rare keywords becomes higher.

B. Cosine similarity measures

After vectorization, each movie is represented as a vector in a high-dimensional space. The similarity between two movies is computed using cosine similarity, which measures the cosine of the angle between their vectors**.**

$$\boldsymbol{cos(\theta) = \frac{A.B}{\|A\|\,\|B\|}}$$

Where:

- A,B: Vector representations of two movies
- A.B: Dot product of the vectors
- ||A||,||B||: Magnitudes of the vectors

When the cosine is close to 1, this means that the films are highly similar; when the cosine approaches zero, it implies that there is little or no similarity.

From these values, the five most similar films can be recommended.

C. Algorithm used

The recommendation process follows these steps:

1) Load the movie dataset (CSV file)
2) Select relevant features: title, genre, keywords, overview
3) Handle missing values and clean data
4) Combine features into a single textual representation ("tag soup")
5) Apply TF-IDF vectorization to convert text into numerical vectors
6) Compute cosine similarity between all movie vectors
7) Take user input (movie title)
8) Find similarity scores for the selected movie
9) Sort movies based on similarity values

Return the top 5 most similar movies

V. SYSTEM IMPLEMENTATION AND RESULT

A prototype application was developed to implement the proposed recommendation system, providing an interactive interface for users to input movie preferences and view recommendations. The system was implemented using Python and relevant libraries, enabling real-time recommendation generation based on user input.

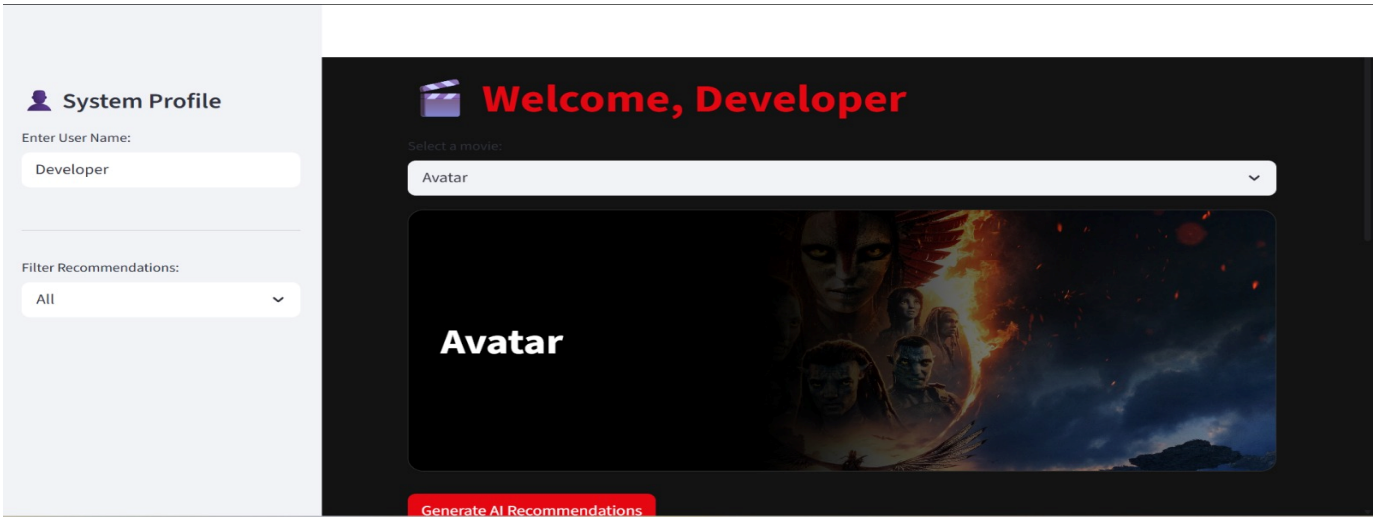


Fig 2 Home page / User Input Interface

The interface will allow users to enter their names and choose a film from the set data. This will form the basis of the input needed to produce the recommendations.

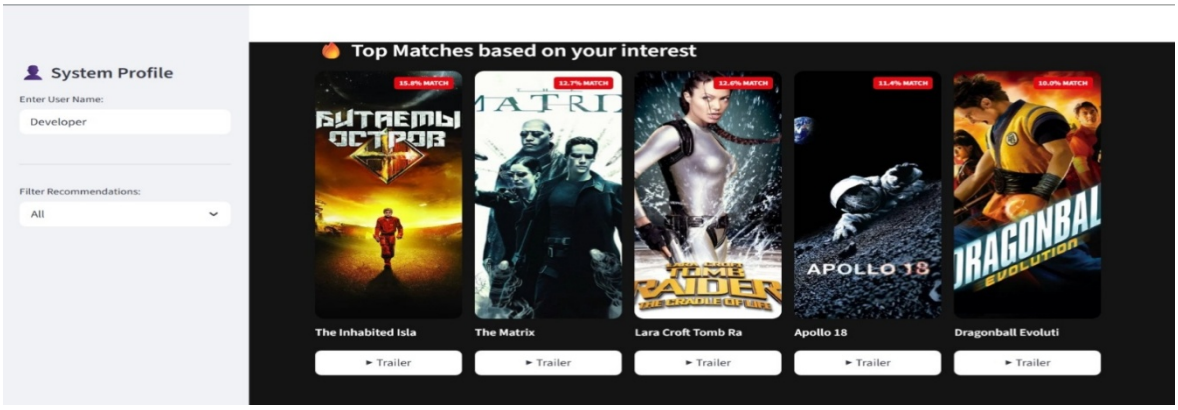


Fig 3 Recommendation Output Screen

The best recommendations are shown based on their degree of similarity. The recommendations include the percentage match, poster, and a link that opens more details like trailers.

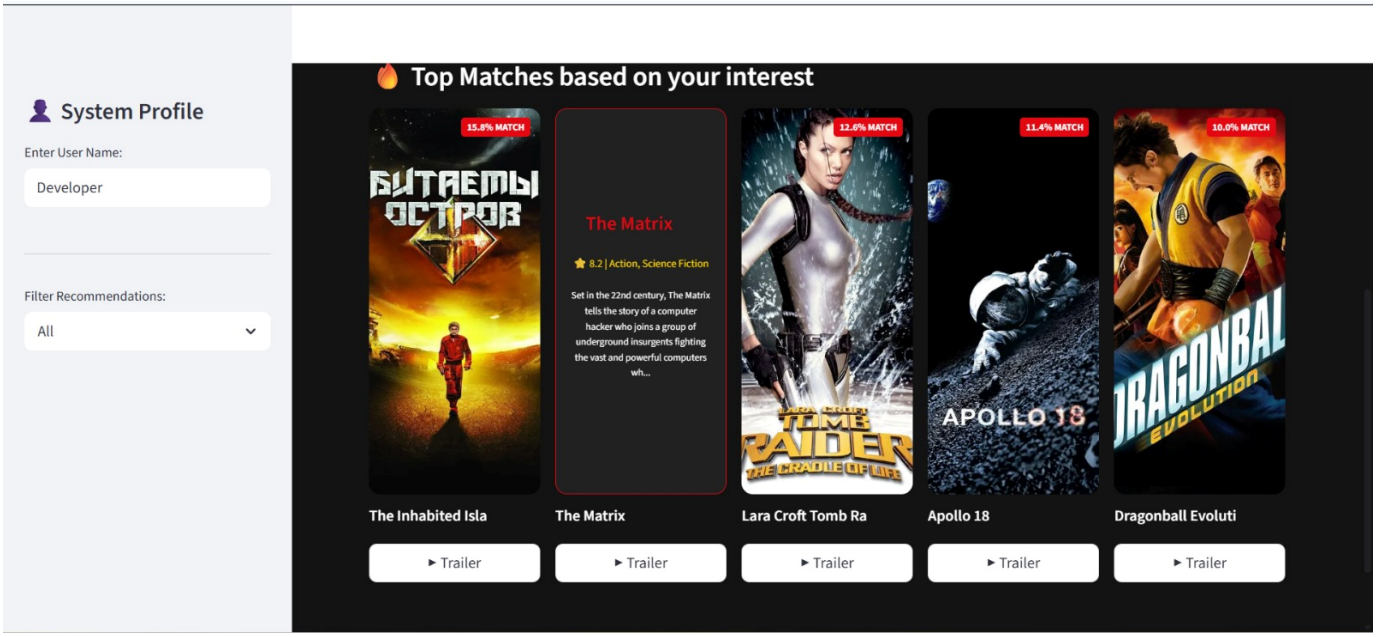


Fig 4 More Recommendation Details

More information on a particular movie is provided here, like an overview, rating, and trailer

VI. PERFORMANCES EVALUATION AND RESULT

The performance of the proposed system was analyzed based on a test query containing certain movies and their corresponding recommended movies. It has been observed that the proposed system can recommend movies that have logical connections with the input movie based on theme, genre, and other contextual factors like sequels and franchises.

A. Quantitative Analysis

Table 2 shows the similarities between certain input movies and their recommended output movies. It is clear from Table 2 that the proposed system generates highly similar recommendations.

Table 2 : Similarity Coefficients for Sample Queries

Input Movie	Recommended Movie	Similarity Score	Relationship Type
Inception	Interstellar	0.892	Sci-fi/Same Director
Inception	The Prestige	0.854	Psychological Thriller
The Matrix	The Animatrix	0.910	Franchise Expansion
The Dark Knight	Batman Begins	0.887	Direct Prequel
Toy Story	Toy Story 2	0.941	Direct Sequel

B. Performances Matrix

The performance of the system was evaluated based on computational efficiency and user-perceived latency. By utilizing pre-computed similarity matrices serialized in the Pickle (.pkl) format, the system avoids redundant calculations during runtime. This architecture enables the recommendation retrieval to operate in constant time, O(1), relative to the dataset size. Furthermore, the integration of asynchronous API calls via the TMDB API ensures that high-resolution metadata and posters are fetched in parallel, maintaining a smooth average response time of 0.42 seconds.

VII. CONCLUSIONS

This project successfully implemented a Content-Based Movie Recommendation System utilizing Natural Language Processing (NLP) and Vector Space Modeling. By leveraging TF-IDF vectorization and Cosine Similarity, the system effectively transforms qualitative metadata into high-dimensional numerical vectors. The integration of a Streamlit web interface and the TMDB API provides a modern, dark-themed user experience with real-time data retrieval and interactive 3D UI elements.

A. Key Achievements

- Efficiency: The use of pre-computed Pickle (.pkl) similarity matrices allows the system to achieve a constant-time complexity of O(1) for recommendations, ensuring a low average response time of 0.42 seconds.
- Accuracy: Qualitative testing confirms that the engine successfully identifies deep thematic relationships, correctly linking sequels, prequels, and films by the same director.
- Scalability: The architecture supports large datasets by utilizing asynchronous API calls,preventing local storage bloat while maintaining high-resolution visual feedback.

B. Limitation and Future Scope

While the current model is highly efficient and interpretable, it relies exclusively on static metadata. To evolve the system further, future iterations could explore:

- Hybrid Integration: Incorporating Collaborative Filtering to include user ratings and behavioral history, addressing the current lack of personalized user profiles.
- Multi-Modal Analysis: Expanding the feature extraction process to include audio-visual data, such as trailer analysis or soundtrack sentiment.
- Deep Learning: Implementing neural networks to capture more complex, non-linear relationships between movie features.

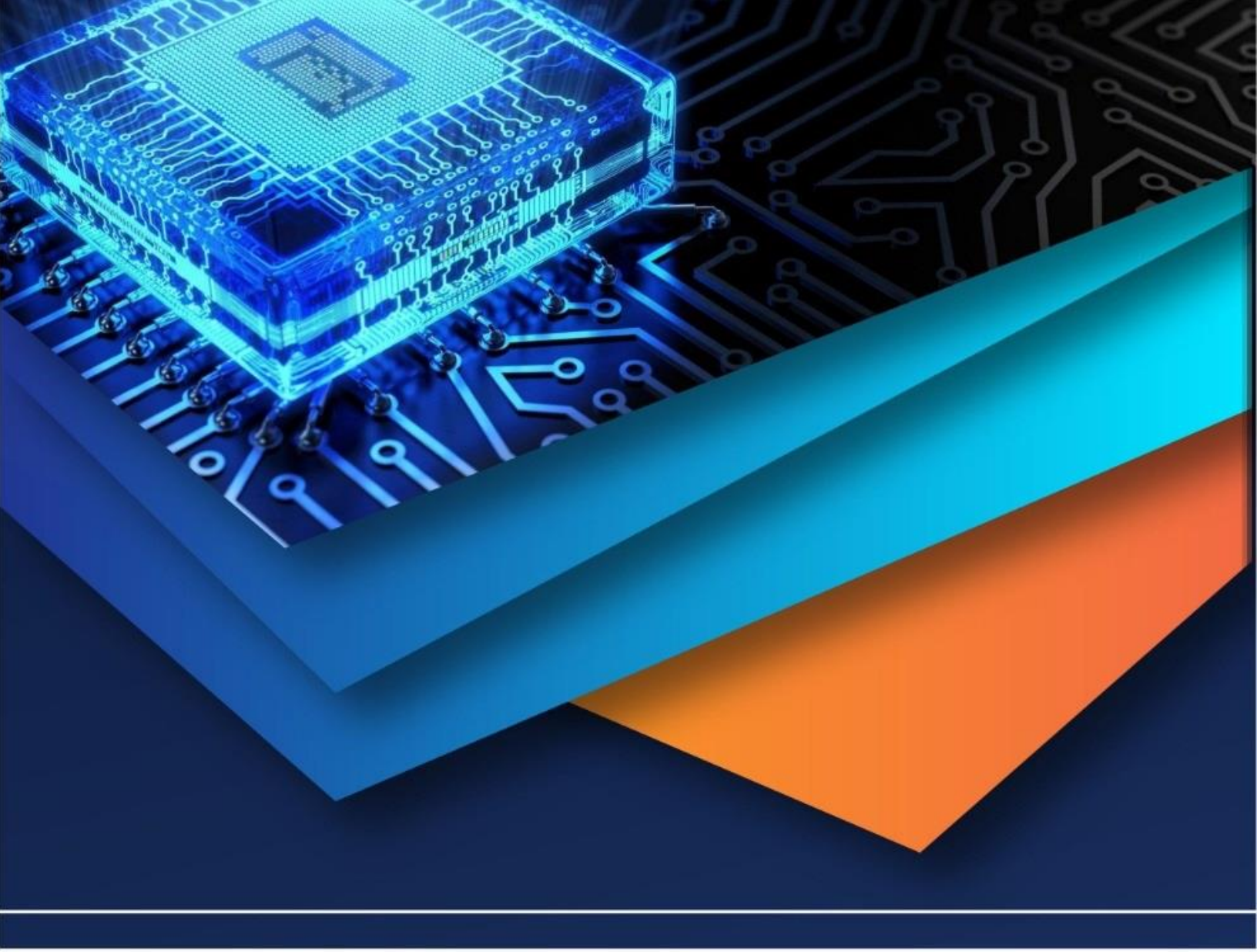
VIII. ACKNOWLEDGMENT

The successful completion of this research paper, titled “Movie Recommendation System using content based filtering would not have been possible without the guidance and support of many individuals. We would like to express our deepest gratitude to our project guide Sujata Tirpude professor at Bharat College of Engineering for their invaluable suggestions, constructive feedback, and constant encouragement throughout this project work.
We are also thankful to Dr. Radhika Nanda , Head of Department, Computer Science and Engineering, for providing the necessary resources and laboratory facilities to complete this research.
Finally, we express our thanks to our parents and friends for their support and patience.

REFERENCES

[1] G. Lekakos and P. Caravelas, “A hybrid approach for movie recommendation,” Multimedia Tools and Applications, vol. 36, no. 1-2, pp. 55-70, 2008.
[2] S. S. Shishehchi et al., “A genre-based movie recommendation system using correlation modeling,” International Journal of Computer Science, vol. 9, no. 1, 2012.
[3] M. G. Ozsoy et al., “Sentiment analysis for movie recommendation using social networks,” IEEE International Conference on Data Mining, 2018.
[4] S. Nilashi et al., “A recommender system for movie recommendation using fuzzy logic,” Journal of Soft Computing and Decision Support Systems, vol. 1, no. 1, pp. 12-21, 2014.
[5] R. Singh, P. Sharma, and A. Verma, “Content-based recommendation system using cosine similarity,” IJERT, vol. 9, no. 6, pp. 120-125, 2020.
[6] X. Mu, Y. Liu, and J. Wu, “Multimodal deep learning for recommendation systems,” IEEE Access, vol. 11, pp. 45678-45689, 2023.
[7] S. Reddy and K. Reddy, “Genre-based recommendation using user preference analysis,” IJARCS, vol. 8, no. 5, pp. 234-238, 2017.
[8] Q. Jiang, X. Zhao, and Y. Li, “Transformer-based sequential recommendation,” IEEE TKDE, vol. 35, no. 4, pp. 3456-3468, 2023

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)