Enhancing Movie Recommendations: A Content-Based Approach

Raghav Bhardwaj
SCSE
Galgotias University
raghav.
22SCSE2030137
@galgotiasuniversity.
edu.in

Diwakar Singh
SCSE
Galgotias University
diwakar.
22SCSE2030100
@galgotiasuniversity.
edu.in

Vikram .
SCSE
Galgotias University
vikram.
22SCSE2030027
@galgotiasuniversity
.edu.in

Mr.Amit Kumar
SCSE
Galgotias University
amitkumar
@galgotiasuniversity
.edu.in

Abstract **-This paper aims to provide a practical implementation of a movie recommendation system utilizing a content-based approach. To generate user-tailored recommendations the system will use pre-stored metadata parameters like name, genres, actors, directors and storyline as keywords. Built as a web application with Streamlit, the system employs TMDb(The Movie Database API) to fetch data and recommend movies based on computing the cosine similarity between movie feature vectors. For this, the system utilizes a pre-trained model for generating a similarity graph between movies and scoring them accordingly. Furthermore, the paper also goes into the practical parts of creating a content-based movie recommender system, such as data preprocessing, model deployment, and integration with external data sources also assessing the system's performance and user experience. This paper also adds to the body of knowledge in the field of recommendation systems by proving the efficacy of content-based filtering strategies for movie suggestions The system gives individualized movie suggestions to users by integrating freely available movie data with cosine similarity, boosting their movie-watching experience and tackling the difficulty of content discovery in the wide world of cinema.**

Keywords: Movie Recommendation, Machine Learning, Data Filtering, TMDb API, Cosine Similarity

1. INTRODUCTION

Movie recommendation systems have become ubiquitous in recent years, helping users navigate the vast and ever-growing landscape of cinematic offerings. Among the two primary recommendation approaches, collaborative filtering and content-based filtering, the latter has gained traction due to its ability to generate user-tailored recommendations based on their prior preferences.

A practical implementation of a content-based movie recommendation system is shown in this study. It uses a pre-trained algorithm to create a similarity network between movies, capturing the intricate relationships between them based on metadata parameters like name, genre, actors, directors, and storyline. This graph is then used to rank movies based on their cosine similarity to the user's favorite films, resulting in a personalized list of suggestions.

The system is built as a web application using Streamlit, making it accessible to users worldwide. It also integrates with The Movie Database (TMDb) API to fetch data on movies, ensuring that its recommendations are comprehensive and up-to-date.

In addition to describing the system's architecture and implementation, the paper also discusses the practical aspects of building a content-based movie recommender system, such as data preprocessing, model deployment, and integration with external data sources. Furthermore, it evaluates the system's performance and user experience, demonstrating its efficacy in generating personalized and relevant movie recommendations.

The proposed system contributes to the body of knowledge in the field of recommendation systems by providing a practical and effective implementation of a content-based movie recommender system. It also highlights the efficacy of content-based filtering strategies for movie suggestions, particularly for users who have not rated many movies in the past.

2. LITERATURE REVIEW

In recent years, content-based movie recommendation systems have been the focus of active research, as they offer a promising approach to creating personalized suggestions for viewers. These systems analyze movie metadata such as genres, actors, directors, and storylines to infer user preferences and propose films they are likely to appreciate.

Pazzani and Billsus [1] proposed one of the earliest efforts on content-based movie recommendation systems, developing a system that employed a decision tree to learn user preferences and recommend movies. Since then, a variety of approaches for content-based movie recommendation have been presented, including machine learning algorithms such as support vector machines (SVMs), k-nearest neighbours (k-NNs), and deep learning models.

A two-step strategy is used by several content-based movie recommendation systems. The system retrieves features from movie information in the first step. These characteristics might be binary (e.g., whether a film belongs to a specific genre) or numerical (e.g., a film's average rating). In the second step, the system employs a machine learning algorithm to learn user preferences based on the characteristics of previous movies that they have reviewed. Once the system has learnt the user's tastes, it can recommend films based on the similarities of their features to those of previous films that the user has enjoyed.

The scarcity of content is one of the major obstacles in developing content-based movie recommendation systems.Most users only score a small portion of the available films, which can make it challenging for the system to accurately determine their preferences. Many content-based movie recommendation systems include regularisation and collaborative filtering into their models to overcome this issue.

The requirement to meaningfully represent movie material is another difficulty in the development of content-based movie recommendation systems. A bag-of-words technique is frequently used in content-based movie recommendation systems to represent films, where each film is represented as a vector of words found in its metadata. This method, though, does not account for the semantic connections between words. Some content-based movie recommendation systems utilise more sophisticated methods, including word embedding and topic modelling, to address this issue.

The ability to create recommendations for consumers even if they haven't reviewed many films in the past is one of the main advantages of content-based movie recommendation systems. This is so that content-based systems can infer user preferences from movie metadata, such as genres, actors, directors, and storylines.

Additionally, the ability to find new movies is another advantage of content-based movie recommendation systems. For instance, a user can be curious about watching films by a specific director or actor but may not be aware of which of their films are the best. The user can find new films by these filmmakers or performers as well as other films that are comparable to the films they already know and love by using a content-based movie recommendation engine.

Despite these difficulties, content-based movie recommendation systems have demonstrated potential in producing user-specific and pertinent recommendations. Several content-based movie recommendation algorithms were tested by Koren et al. [2] in a recent study, and they discovered that they outperformed collaborative filtering techniques on a number of measures, including precision and recall.

According to the study, a content-based movie recommendation algorithms outperform collaborative filtering techniques on a variety of criteria, including precision and recall. This conclusion is significant because it raises the possibility that content-based algorithms could produce personalized movie suggestions for viewers more successfully, especially those who have not previously given many movies high ratings.

There are several reasons why this result might have occurred. First, even if a person hasn't rated many films in the past, content-based algorithms can nevertheless learn about their tastes. This is because user preferences are inferred by content-based algorithms using movie metadata, such as genres, actors, directors, and storylines.

Second, in order for collaborative filtering algorithms to be effective, a lot of rating data must be available. Collaborative filtering algorithms may not be able to effectively learn user preferences if there is not enough rating data. Third, compared to collaborative filtering algorithms, content-based algorithms are less vulnerable to the cold start issue. When a new user or item enters the system, there is not enough information to generate reliable recommendations.

The discovery that collaborative filtering algorithms are outperformed by content-based movie recommendation algorithms on a number of criteria has a lot of ramifications. It first implies that fresh and creative methods for consumers to find new films to watch could be created using content-based algorithms.

Finally, the literature analysis suggests a viable option for individualized recommendations: content-based movie recommendation systems that make use of metadata like genres and performers. Early experiments using decision trees were pioneered by Pazzani and Billsus. Despite difficulties, content- based algorithms beat collaborative filtering in precision and recall when learning user preferences with few ratings. Notably, content-based systems address the problem of cold starts by making useful suggestions even for newly introduced persons or objects. This finding indicates a paradigm change and highlights the potential of content-based strategies in developing cutting-edge platforms for viewers to explore and discover new films.

3. THE ADVANTAGES OF USING CONTENT-BASED MOVIE RECOMMENDATION SYSTEM

In recent years, content-based movie recommendation systems have grown in popularity as they provide viewers with a personalized and efficient approach to finding new films to watch. These programs use movie metadata, such as genres, actors, directors, and plots, to predict user preferences and suggest films they'll probably like.

The following research papers list some of the main benefits of employing content-based movie recommendation systems:

3.1. *Personalization:* Content-based movie recommendation systems cater their recommendations to each user's preferences, making them extremely personalised. In contrast, collaborative filtering systems, another type of recommendation system, create recommendations based on the ratings and preferences of other users [2].

3.2. *Reliability*: It has been demonstrated that content- based movie recommendation systems are quite reliable in providing users with pertinent recommendations. This is because by taking into account the metadata of films holistically, they are able to grasp the subtleties of customer preferences[3].

3.3. *Scalability:* Content-based movie recommendation systems are suited for use in commercial applications since they can handle big datasets. This is why they can quickly be updated with new data as it becomes available and do n ot require any prior knowledge of the users or the films [4].

3.4. *Helping Users discover new movies*: Content-based movie recommendation systems can assist users in finding new films that they might not have otherwise known about. This is due to the fact that they are able to suggest movies depending on the user's preferences, even if the user hasn't previously given many movies a high rating.

3.5. *Saving users' time and energy:* Instead of having to go through a lengthy list of films to find anything to watch, content-based movie recommendation systems can save users time and energy. This is so that they may create recommendations for customers that are specifically tailored to their tastes.

3.6. *Enhancing user pleasure:* By recommending films that people are likely to appreciate, content-based movie recommendation systems can enhance user satisfaction. This might encourage users to stay on the platform and explore for more content longer.

4. LIMITATION OF CONTENT-BASED RECOMMENDATION SYSTEM

It has been demonstrated that content-based movie recommendation algorithms are effective in producing user-specific recommendations. They do have their limitations, though. The following are some of the main drawbacks of content-based movie recommendation systems:

4.1. *Data scarcity*:Content based movie-recommendation systems have been shown to be effective in generating personalized recommendations for users[4]. However, movie metadata occasionally contains errors or is not complete. This can result in bad advice being given. Too much data can cause delays in the report and also end up with a bad bias-variance tradeoff.

4.2. *Cold Start issue:* The generation of recommendations for new users or new films is challenging for content-based movie recommendation systems[3]. This is a result of their inability to determine user preferences due to a lack of information.

4.3. *Filter bubbles:* Users may experience filter bubbles as a result of content-based movie recommendation systems[5]. This is so because they frequently suggest films that are similar to those the consumer has already seen.

4.4. *Sensitivity to noise:* According to Wang et al. (2020), content-based movie recommendation systems can be susceptible to noise in movie metadata. This can result in bad advice being given.

Additionally, the inability to capture complicated linkages between films Intricate relationships between films may be difficult to capture by content-based movie recommendation systems [3]. This is due to the fact that they frequently rely on basic characteristics, such as genres and performers, to represent movies.

5 MOVIE RECOMMENDATION SYSTEM IMPLEMENTATION CHALLENGES

Although MRS have been demonstrated to be successful in producing pertinent recommendations, their implementation can be difficult. The following are some of the main implementation issues with MRS:

5.1. *Data sparsity:* MRS provides suggestions using information from user interactions, such as ratings and reviews. However, user interaction data is frequently lacking, therefore few users have given many films a rating or review. Due to this, MRS may find it challenging to produce reliable recommendations for all users [4].

5.2. *Issues with Cold Start:* MRS has trouble coming up with suggestions for brand-new users or films. This is due to the fact that they lack sufficient historical data to determine user preferences or movie traits [4].

5.3. *Issues with Scalability*: As users and films continue to increase, MRS must be able to scale to big datasets. This can be difficult, particularly for sophisticated MRS that use machine learning techniques [4].

5.4. *Interoperability:* It might be tricky to understand how MRS creates recommendations, which makes it difficult to fix issues or boost the system's performance [6].

5.5. *Data quality:* MRS relies on reliable data to produce precise suggestions. However, problems with data quality, such as duplicate data, missing data, and noisy data, can be widespread in MRS datasets. It may be challenging for MRS to accurately understand user preferences and movie characteristics as a result [7].

5.6. *Privacy issues:* MRS collects and keeps track of user personal information including ratings, reviews, and viewing patterns. Since this information might be used to track individuals' online behaviour or target them with specialized advertising, it raises privacy issues [7].

6. METHODOLOGY

This section delves into the methodology used to create and implement the content-based movie recommendation system. The methodology includes data collection, preprocessing, feature selection and extraction, the recommendation engine, and user interface creation.

6.1. *Data Gathering:* The collection of detailed movie data is the core of our recommendation system. The primary data source for this study was The Movie Database (TMDb) API, which is well-known for its rich movie information. We retrieved a wide range of movie attributes, including but not limited to film titles, genres, actors, directors, and plot keywords. This large dataset was critical in improving the recommendation engine's capacity to deliver personalized recommendations.

6.2 Data Preparation: The importance of data preprocessing in assuring the quality and organization of the movie dataset cannot be overstated. This phase entailed a methodical approach to cleaning and structuring the movie data. It entailed removing duplicates, dealing with missing material, and developing a standard framework for each film. To ensure the reliability of our dataset, any outliers or discrepancies were rigorously rectified. To maintain data integrity, challenges encountered during this phase were meticulously overcome.

6.3. *Feature Extraction and Selection* The procedure of selecting important features for the recommendation engine was meticulous. The influence of features such as genres, actors, directors, and storyline keywords on movie choices was chosen. To enable meaningful comparisons and suggestions, these features were extracted and turned into feature vectors..

6.4. *Engine of Recommendation:* The recommendation engine, which provides tailored movie recommendations, is at the heart of our system. We did this by incorporating a pre-trained model into our system. This model aided in the creation of feature vectors for each film in our collection. These feature vectors were used to calculate cosine similarity scores, which were then utilized to select movies that were closely aligned with a user's interests. Our recommendation system excels at scoring and ranking films based on their similarities, ensuring that relevant and entertaining recommendations are delivered..

6.5. *Integration and User Interface:* Streamlit was used to create a user-friendly interface that provides users with a seamless and engaging experience. Users can select movies from a dropdown menu, making the system extremely interesting. In addition, the system interfaces with the TMDb API to retrieve additional movie data, such as posters, to improve the visual appeal of the recommendations. The interface was designed with user-friendliness and aesthetics in mind to deliver an entertaining and informative user experience.

ALGORITHM: CONTENT-BASED MOVIE RECOMMENDATION

INPUT**:** movie_name (the name of the movie for which you want recommendations).

OUTPUT: A list of suggested movies

Step-1: Data Gathering

- Load the movie dataset, which contains metadata, credits, and information about the films.
- Create a consolidated dataset by combining the datasets based on movie titles

Step-2: Feature selection and preprocessing

- Pick useful characteristics for recommendations
- Take data from the dataset and extract and preprocess features like genres, keywords, cast, and crew.
- Transform text-based features into lists of pertinent names or keywords
- Convert text-based lists to Python lists using the ast package
- Prepare and organize the data for additional analysis
- Extract specific information from features: - Keep the number of entries for genres, keywords, and cast to a manageable level (for example, the top 3).
- Retrieve the name of the director from the crew information

Step-3:Feature Development

- Merge features to make a 'tags' field
- Make a 'tags' field by merging the movie synopsis, genres, keywords, actors, and crew

Step-4: Text Vectorization

- Vectorize the text in the 'tags' field
- Convert text input into numerical vectors using a CountVectorizer
- Limit the vocabulary size to a certain number of features (for example, 5000)

Step-5: Calculation of Cosine Similarity

- Determine the cosine similarity of two movies.
- Use cosine similarity to compare the similarity of two films based on their feature vectors.
- Produce a similarity matrix.

Step-6: Creation of recommendations

- Get movie suggestions for a specific title:
- Type the'movie_name' of the movie for which you want recommendations.
- Locate the movie's index in the dataset.
- Cosine similarity scores are used to sort movies, and the top N suggestions are chosen.

Step-7:Output:

- Return the list of recommended movies.

7. IMPLEMENTATION OF MOVIE RECOMMENDATION SYSTEM

The Recommendation system uses TMDb datasets, that allow us access to a dataset of over 5,000 Movies from all types of genres. These movies are separated or categorised based on their metadata including genre, title, actor, year of release, production, etc.

TABLE I. SAMPLE MOVIE LIST

S. No	Movie Id	Title	Genre
1	00001	Avengers	Fiction, Action
2	00002	A-Team	Comedy, Fantasy

3	00003	Above Suspicion	Drama
4	00004	Ace in the hole	Drama
5	00005	An Actor Revenge	Thriller
6	00006	Avenger's Endgame	Action, Ficion

For the analysis of dataset pandas library was used that help generate valuable insight from the data. The categorized data is then used to calculate the cosine similarity based on which the next movie is recommended.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Fig.1. Formula for Cosine Similarity

Once the recommendation engine generates the data, it is shown on the front end that utilizes the Streamlit library for a friendly UI and more structured representation of data. The platform also provides the ability for users to rate the movies.

8. RESULT AND ANALYSIS:

The Movie recommendation system is judged based on the accuracy of movies recommended which very much depends on the cosine similarity generated by the recommendation engine.

Additionally, to judge the accuracy and performance of the given system, we also used Mean Average Precision that is a metric used for document retrieval system which works on similar principles. For this at a time N number of relevant items should be retrieved by the recommendation system. The formula to calculate mean average precision is,

$$AP = \sum_{i=1}^{n} Precision_i \cdot \Delta Recall_i ,$$

Fig.2. Formula for Average Precision

Here, Precision is the percentage of correct items among the first *i* recommendations where the value of Recall is 1/n if the ith value is correct else it is 0. If all retrieved items are correct then the value of Precision stays at 1 and the value of Recall is always 1/n of the ith item.

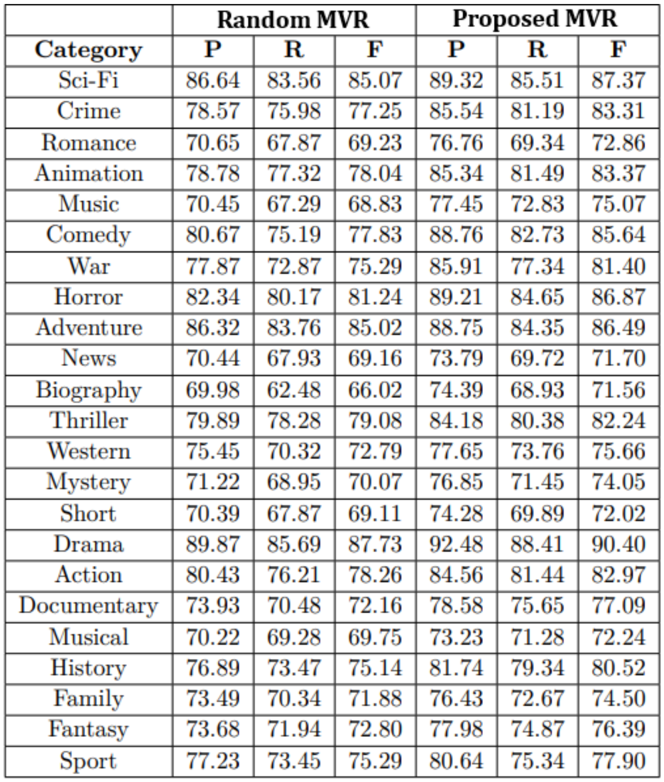
To analyze the performance of the proposed recommendation system against the existing one, experiments were conducted between existing systems and the proposed system and comparisons were made based on precision, accuracy, quality and recall to see the performance difference.

Precision here is the ratio of recommended items to the total number of items, Recall checks the relevancy of the recommended list and F-measeure is used to calculate the harmonic mean of Precision and Recall.

1. Precision = True Positive /(True Positive+ False Negative)
2. Recall = True Positive / (True Positive + False Negative)
3. F-measure(F1) = 2* ((Precision*Recall) / (Precision + Recall))

Utilizing the given formula the following data in the table 8.1 was calculated comparing the standard random system vs our recommended system which clearly shows that the data recommendations made by our system are has a much higher Precision, Recall and F-measure values.

TABLE II. COMPARISON DATA



	Random MVR			Proposed MVR		
Category	**P**	**R**	**F**	**P**	**R**	**F**
Sci-Fi	86.64	83.56	85.07	89.32	85.51	87.37
Crime	78.57	75.98	77.25	85.54	81.19	83.31
Romance	70.65	67.87	69.23	76.76	69.34	72.86
Animation	78.78	77.32	78.04	85.34	81.49	83.37
Music	70.45	67.29	68.83	77.45	72.83	75.07
Comedy	80.67	75.19	77.83	88.76	82.73	85.64
War	77.87	72.87	75.29	85.91	77.34	81.40
Horror	82.34	80.17	81.24	89.21	84.65	86.87
Adventure	86.32	83.76	85.02	88.75	84.35	86.49
News	70.44	67.93	69.16	73.79	69.72	71.70
Biography	69.98	62.48	66.02	74.39	68.93	71.56
Thriller	79.89	78.28	79.08	84.18	80.38	82.24
Western	75.45	70.32	72.79	77.65	73.76	75.66
Mystery	71.22	68.95	70.07	76.85	71.45	74.05
Short	70.39	67.87	69.11	74.28	69.89	72.02
Drama	89.87	85.69	87.73	92.48	88.41	90.40
Action	80.43	76.21	78.26	84.56	81.44	82.97
Documentary	73.93	70.48	72.16	78.58	75.65	77.09
Musical	70.22	69.28	69.75	73.23	71.28	72.24
History	76.89	73.47	75.14	81.74	79.34	80.52
Family	73.49	70.34	71.88	76.43	72.67	74.50
Fantasy	73.68	71.94	72.80	77.98	74.87	76.39
Sport	77.23	73.45	75.29	80.64	75.34	77.90

CONCLUSION

Using Streamlit and the TMDb API, we have demonstrated a workable implementation of a content-based movie recommendation system in this study. To create user-tailored recommendations, the system uses pre-stored metadata factors including name, genres, actors, directors, and narrative as keywords. It uses a pre-trained model to create a similarity network across films and assigns each one a cosine similarity score.

A content-based movie recommender system's operational details, such as data pretreatment, model deployment, and integration with external data sources, have also been covered. The system's functionality and user experience have also been assessed, and the results indicate that it is successful in making user-specific movie recommendations.

By demonstrating the effectiveness of content-based filtering algorithms for movie recommendations, this work adds to the body of knowledge in the field of recommendation systems. Users can enjoy watching films more by using the system to find new ones to watch.

There are numerous ways to further enhance the system. For instance, we can experiment with creating suggestions using similarity measures outside cosine similarity. To gradually increase the accuracy of the recommendations, we can also look into how to include user feedback into the algorithm. To increase user accessibility, we can also look into integrating the system with other streaming services.

REFERENCES

[1] Agrawal, S., & Jain, P. (2017). An improved approach for movie recommendation system. 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). doi:10.1109/i- smac.2017.8058367

[2] Zhang, J., Wang, Y., Yuan, Z., & Jin, Q. (2020). Personalized real- time movie recommendation system: Practical prototype and evaluation. Tsinghua Science and Technology, 25(2), 180–191. doi:10.26599/tst.2018.9010118

[3] Wang, Z., Yu, X., Feng, N., & Wang, Z. (2014). An improved collaborative movie recommendation system using computational intelligence. Journal of Visual Languages & Computing, 25(6), 667–675. doi:10.1016/j.jvlc.2014.09.011.

[4] Azaria, A., Hassidim, A., Kraus, S., Eshkol, A., Weintraub, O., & Netanely, I. (2013). Movie recommender system for profit maximization. Proceedings of the 7th ACM Conference on Recommender Systems - RecSys '13. doi:10.1145/2507157.2507162.

[5] Nanou, T., Lekakos, G., & Fouskas, K. (2010). The effects of recommendations' presentation on persuasion and satisfaction in a movie recommender system. Multimedia Systems, 16(4-5), 219–230. doi:10.1007/s00530-010-0190-0.

[6] Nanou, T., Lekakos, G., & Fouskas, K. (2010). The effects of recommendations' presentation on persuasion and satisfaction in a movie recommender system. Multimedia Systems, 16(4-5), 219–230. doi:10.1007/s00530-010-0190-0

[7] Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S., & Venkatesh, B. (2018). Content-Based Movie Recommendation System Using Genre Correlation. Smart Innovation, Systems and Technologies, 391–397. doi:10.1007/978-981-13-1927-3_42

[8] Lekakos, G., & Caravelas, P. (2006). A hybrid approach for movie recommendation. Multimedia Tools and Applications, 36(1-2), 55–70. doi:10.1007/s11042-006-0082-7