Penerbit Akademia Baru

Journal of Advanced Research Design

Journal homepage:
https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

Movie Description Feature Extraction for Movie Recommendation Model with Content Based Filtering Approach

Lili Ayu Wulandhari[1], Gabriela Janice Wijaya[1], Ika Nardianac[1], Sherly Graciad[1], Zirawani Baharum[2,*]

[1] Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia
[2] Malaysian Institute of Industrial Technology, Universiti Kuala Lumpur, Persiaran Sinaran Ilmu, Bandar Seri Alam, Johor, Malaysia

ARTICLE INFO

Article history:
Received 23 June 2025
Received in revised form 26 July 2025
Accepted 14 August 2025
Available online 1 November 2025

Keywords:
Movie Recommendation; Content Based Filtering; Text Mining; CountVectorizer; TF-IDF; Cosine Similarity

ABSTRACT

Recommendation is a method that simplifies the user's decision-making process when faced with multiple options. Recommendation might also be put into practice for the choice of movies. Prior academics have extensively proposed recommendation models employing various data and methodologies. The presence of data in the process of modeling recommendations plays a crucial role in determining the choice of approaches. The data employed in this study pertains to specific movie items that will be recommended. Therefore, the chosen method for analysis is content-based filtering. Due to the presence of descriptions in the available features, the text mining strategy is employed using vectorization methods such as TF-IDF and CountVectorizer for analysis. This approach is capable of generating effective features for the recommendation model, in addition to other movie-related features. The experimental results demonstrated that TF-IDF achieved precision@k values that were 0.6% superior to those of CountVectorizer for film type of show. Conversely, for television shows, CountVectorizer yielded 3.24% more accurate outcomes in providing relevant recommendations for the selected base movie.

1. Introduction

Recommendation is a useful feature that offers consumers a favorable experience across a wide range of applications. Recommendations facilitate the user's decision-making process by aligning with their product preferences, thereby simplifying the selection process among an extensive variety of options. Users do not personally generate recommendations; rather, scientists have developed models utilizing precise algorithms. A movie recommendation is one example of a recommendation system. A movie recommendation aims to present a curated selection of films that share common characteristics, minimizing user confusion while making choices.

Prior researchers, such as Pavita et. al., have conducted research on film recommendations. They utilized cosine similarity and incorporated sentiment prediction as supplementary features [1]. Hasan and Ferdous suggest a movie recommendation technique that involves converting the text into a numerical representation and utilizing cosine similarity to identify movies that are similar [2].

* *Izanoordina Ahmad.*
E-mail address: zirawani@unikl.edu.my

Another group of researchers created a film recommendation algorithm that tackles the cold start problem by combining user similarity with weighted trust propagation [3]. Prior studies have put forth several recommendation algorithms for films, with some employing a cosine similarity technique that has demonstrated its ability to offer insights on similarities between two movies. The selection of data is dependent upon its availability and the desired objectives to be accomplished. The features utilized in constructing recommendation models are diverse, encompassing both numerical and occasionally textual data. Extracting meaningful features from text data has its own issues in the processing phase. Therefore, the paper proposes a recommendation model that uses data from film descriptions that have been processed using text mining with TF-IDF and CountVectorizer vectorization. The vectorization result is integrated with additional features, and cosine similarity will be computed.

This paper is divided into five sections. The first section deals with the background of the development of the movie recommendation model, followed by an introduction to the recommendation problem in Section 2. Section 3 discusses the methodology carried out in this study, with the results and discussion described in Section 4. The end of this paper is closed with conclusions and references in Section 5 and Section References.

2. Recommendation Problem

The recommendation problem is a computational method that aims to offer customized suggestions based on the user's preferences and requirements. Prior knowledge of the reference is essential for providing a recommendation on an issue. The recommendation system is highly beneficial in managing the overwhelming amount of information present in the domains of e-commerce, entertainment, and social media [4]. Recommendation systems have demonstrated their ability to expedite the growth of well-known worldwide enterprises such as Netflix, Spotify, and TikTok. Netflix utilizes data on viewing patterns and user behavior to provide personalized suggestions to those who exhibit similar tendencies. This technique has demonstrated its efficacy in providing consumers with film recommendations in a more comprehensive and extensive manner [5]. Spotify, an online music portal, employs a reinforcement learning approach to offer music recommendations that are highly pertinent to its customers [6]. Tiktok as a global reach social media also implement recommendation system, researchers at the company have developed Monolith; a Real Time Recommendation System With Collisionless Embedding Table. This system generates a list of video arrangements that enhance user experience when surfing the app [7].

The field of recommendation system engineering has experienced significant advancements in recent times. Two commonly employed algorithms in various domains are content-based filtering and collaborative filtering. Content-based filtering uses item features to recommend items that are similar to what the user likes based on previous actions or explicit feedback [8]. A film recommendation, for example, will be based on attributes specific to the film, such as genre, actor or actress, and film topics. On the other hand, the collaborative filtering strategy chooses items by considering the correlation between users and their similar tastes [9].

Collaborative filtering often involves two matrices: one representing the user's preferences for recommended items, and another providing the specific information about the items to be recommended. Collaborative filtering is feasible when there is access to both user preference data and the specific information about the item that is being recommended. Researchers often integrate multiple recommendation methods, such as collaborative and fuzzy expert systems, to create hybrid approaches. This approach yields the strength of each method in achieving improved recommendation results [10]. Reinforcement learning is another technique included in the most

recent recommendation. Reinforcement learning is a subfield of machine learning that focuses on analyzing problems and finding solutions. In this approach, agents are trained to optimize a numerical reward by interacting with their environment. Reinforcement learning is capable of addressing the recommendation problem by effectively managing sequential and dynamic interactions between users and systems. Additionally, it considers the long-term engagement of users [11].

The decision to use one of the recommendation techniques is significantly affected by the accessibility of the data and the desired goals to be accomplished. This research utilizes data on the specific characteristics of the item that will be recommended, consequently employing the content-based filtering strategy.

3. Movie Recommendation using Content Based Filtering and Text Features

This study utilized the content-based filtering approach to generate movie recommendations. Content-based filtering is employed due to the fact that the available data exclusively pertains to the film. The feature utilized is solely depending on the fundamental characteristics of each film item. This study utilized movie data that included the following features: ID, title, show type, movie description, release year, age certification, runtime in minutes, IMDb score, and number of IMDb votes. The data was collected from Kaggle [12]. A total of 4222 data points were utilized for constructing and assessing the recommendation model. A subset of these data points is displayed in Table 1.

Table 1
The Example of movie data and its attributes

	index	id	title	type	description	release_year	age_certification	runtime	imdb_id	imdb_score	imdb_votes
0	0	tm84618	Taxi Driver	MOVIE	A mentally unstable Vietnam War veteran works ...	1976	R	113	tt0075314	8.3	795222.0
1	1	tm127384	Monty Python and the Holy Grail	MOVIE	King Arthur, accompanied by his squire, recrui...	1975	PG	91	tt0071853	8.2	530877.0
2	2	tm70993	Life of Brian	MOVIE	Brian Cohen is an average young Jewish man, bu...	1979	R	94	tt0079470	8.0	392419.0
3	3	tm190788	The Exorcist	MOVIE	12-year-old Regan MacNeil begins to adapt an e...	1973	R	133	tt0070047	8.1	391942.0
4	4	ts22164	Monty Python's Flying Circus	SHOW	A British sketch comedy series with the shows ...	1969	TV-14	30	tt0063929	8.8	72895.0

In order to construct a movie recommendation system utilizing the content-based filtering technique, the movie description attribute is analyzed using the text mining methodology to extract a textual characteristic that can offer a comprehensive summary of the movie's content, which may not be conveyed by other attributes. Figure 1 illustrates the sequential process involved in constructing a movie recommendation model.

This movie description undergoes a cleansing process that involves transforming all capital letters to lowercase, eliminating symbols, numbers, and punctuation, as well as reducing double spaces. Table 2 displays a summary of the cleansed movie description data. The outcome of the text that has been cleaned through tokenization is the segmentation of the phrase into its smallest constituent, known as a token. Each token in this scenario corresponds to a single word. The tokenization process generated 18,452 unique tokens from the data used in this study.

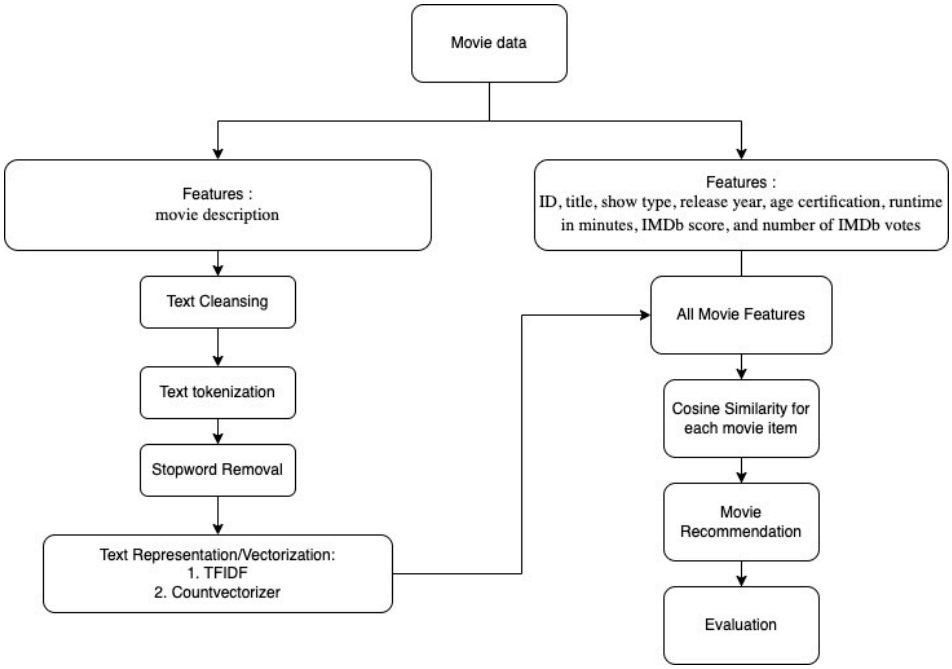

Fig. 1. Procedure a Movies Recommendation model

Table 2
Raw and Clean Movie Description

Original Movie Description	Loving parodies of some of the world's best-known documentaries. Each episode is shot in a different style of documentary filmmaking, and honors some of the most important stories that didn't actually happen.
Movie Description after Cleansing	loving parodies of some of the world's best known documentaries each episode is shot in a different style of documentary filmmaking and honors some of the most important stories that didn't t actually happen

Every token generated by tokenization will go through the process of stopwords removal. A stopword is a frequently used word that does not affect the meaning of a phrase if it is ignored. This study utilizes the typical English stopwords provided in the NLTK library. The removal of stopwords led to a reduction in the number of processed tokens from 18,452 to 18,314, indicating that 0.74% of the tokens contained stopwords. The description data has been processed to remove stopwords following the feature extraction using the TF-IDF (term frequency-inverse document frequency) and the CountVectorizer techniques. The results will be compared later when producing film recommendations. TF-IDF is a method used to assess the significance of a word in a document within a collection of texts known as a corpus. TF-IDF of a word in a document is expressed by Equation 1.

$$tf\_idf_{t,d} = tf_{t,d} * idf \quad (1)$$

Where tf is term frequency refers to the number of times a word or term appears in a document.

$$tf_{t,d} = \frac{n}{T} \quad (2)$$

where n is number of times word or term t appears in a document d. The T is Total number of terms in the document d. While idf is inverse document frequency, means measurement how important the term is.

$$idf = log\left(\frac{TD}{D_t}\right) \quad (3)$$

where TD is total number of documents in corpus, D_t is number of documents that contains word or term t.

The CountVectorizer, on the other hand, determines how frequently each word appears in each sentence, creating a number matrix whose dimensions are the quantity of data multiplied by the number of tokens it contains. Because TF-IDF and CountVectorizer are executed after stopword removal, these two text representations use a total of 18,314 tokens.

The resulting text representation is subsequently combined with other features, including the show type, release year, age certification, runtime in minutes, IMDb score, and number of IMDb votes, in two distinct sets, TF-IDF and CountVectorizer, respectively. The label encoder for the release year is determined by the year in which the movie was released. It consists of 62 distinct values ranging from 1953 to 2022, with 7 years having no available movie data. The show type is limited to two values, "show" and "movie", and is encoded using binary encoding with the values 0 and 1. The age certification consists of 12 values, specifically 'TV-14', 'other', 'G', 'R', 'TV-Y', 'PG', 'tv-PG', 'TV-MA', 'Tv-G', 'PG-13', 'TV-Y7', and 'NC-17'. These values are encoded using one hot encoding, resulting in the addition of 12 extra columns. The total number of features that will be considered for cosine similarity is 18,331(Table 3). Table 3. Sample of Complete Features for Movie Recommendation.

Table 3
Sample of Complete Features for Movie Recommendation

	type	release_year	runtime	imdb_score	imdb_votes	aardman	aaron	aback	abad	abah	...	age_certification_PG	age_certification_PG-13	age_certification_R
0	1	60	53	5.9	14714.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
1	1	54	22	8.1	6445.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
2	0	56	167	5.7	4690.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
3	0	56	120	6.8	178.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
4	0	54	181	7.7	48.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0

5 rows × 18331 columns

The cosine similarity method is selected to measure the proximity between two films based on predetermined features established in the previous step. The cosine similarity is computed using Equation 4, where $A, B \in movie\ list$ from the data. And n is number of movie recommendation features, in this case n equal to 18,331.

$$\cos\theta = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2}\sqrt{\sum_{i=1}^{n} {B_i}^2}} \quad (4)$$

It generates a square matrix with dimensions of 4222 x 4222, representing the total number of distinct movies in our dataset. The cosine similarity value is distributed along the range of [0,1], where a value of 1 indicates a high similarity between the films, and a value of 0 indicates rising dissimilarity that is shown in Table 4.

Table 4
Sample of Cosine Similarity Matrix

	0	1	2	3	4	5	6	7	8	9	...	4212	4213	4214	4215	4216	4217	4218	4219	4220	4221
0	1.000000	0.999990	0.999457	0.804907	0.250582	0.999537	0.992802	0.999990	0.993709	0.990208	...	0.560531	0.999994	0.997992	0.999991	0.999748	0.999995	0.999413	0.998333	0.490418	0.999986
1	0.999990	1.000000	0.999476	0.805895	0.251625	0.999588	0.992995	0.999964	0.993930	0.990505	...	0.562844	0.999972	0.998256	0.999967	0.999777	0.999974	0.999509	0.998445	0.492289	0.999960
2	0.999457	0.999476	1.000000	0.823651	0.282296	0.999960	0.996132	0.999333	0.996705	0.994026	...	0.584221	0.999375	0.998159	0.999346	0.999929	0.999394	0.999815	0.999603	0.517428	0.999299
3	0.804907	0.805895	0.823651	1.000000	0.770113	0.822532	0.870178	0.802557	0.865994	0.879135	...	0.920361	0.803280	0.822622	0.802768	0.818028	0.803596	0.823276	0.837589	0.907881	0.802012
4	0.250582	0.251625	0.282296	0.770113	1.000000	0.279196	0.363453	0.247109	0.354314	0.378117	...	0.863521	0.248231	0.273711	0.247438	0.271988	0.248758	0.279064	0.304185	0.936302	0.246212

The recommendation will be determined based on the cosine similarity value. A movie with a similarity value close to 1 indicates strong similarities, and a recommendation will be made based on these. A comprehensive description of the recommendation model evaluation will be provided in Section 4.

4. Result and Discussion

Evaluation of a recommendation system cannot be done directly, as it does not involve a straightforward classification or regression problem with a clearly defined label. The evaluation of the recommendation model utilizes precision@value, as indicated in Equation 5. In this equation, TP represents the true positive, and FP represents the false negative.

$$precision@k = \frac{TP}{TP + FP} \tag{5}$$

Precision@k measures the number of recommended items generated by a model that are in the top k order compared to the total recommendations. Specifically, TP represents the recommendations given by the model that are relevant to the base movie. On the other hand, FP represents the recommendations provided by the model that are not relevant to the base movie.

Table 5
Overview of Evaluation Table

No	Base Movie	Recommendation	Evaluator 1	Evaluator 2	Evaluator 3	Result
1	Let Me In	A Walk Among the Tombstones	TRUE	FALSE	TRUE	TRUE
		The Ballad of Buster Scruggs	FALSE	FALSE	FALSE	FALSE
		Savages	TRUE	FALSE	FALSE	FALSE
		Phantom Thread	TRUE	FALSE	TRUE	TRUE
		Halloween	TRUE	TRUE	TRUE	TRUE
		I Care a Lot	FALSE	FALSE	FALSE	FALSE
		Triple Frontier	FALSE	FALSE	FALSE	FALSE
		The Other Boleyn Girl	FALSE	FALSE	FALSE	FALSE
		Hairspray	FALSE	FALSE	FALSE	FALSE
		What Happened to Monday	TRUE	FALSE	TRUE	TRUE
2	Prince	Mujrim	TRUE	TRUE	TRUE	TRUE
		Beirut, Oh Beirut	FALSE	FALSE	FALSE	FALSE
		Bye Bye London	TRUE	TRUE	TRUE	TRUE
		Lal Patthar	TRUE	TRUE	TRUE	TRUE

The Little Wars	FALSE	TRUE	TRUE	TRUE
A Lion in the House	TRUE	FALSE	FALSE	FALSE
Yaar Gaddar	TRUE	FALSE	TRUE	TRUE
Manoranjan	TRUE	FALSE	FALSE	FALSE
Quiet Victory: The Charlie Wedemeyer Story	FALSE	FALSE	FALSE	FALSE
Dushmani	TRUE	TRUE	TRUE	TRUE

In order to conduct this evaluation, 60 base movies were selected randomly, with 30 films and 30 television shows, respectively. This is because there are significant differences in the attributes of a television show and a film. Three people conducted a manual evaluation of the top 10 recommendations made by the algorithm for each base movie. Table 5 provides a summary of the data evaluation form. The evaluator assesses the consistency between the base movie and the recommendations given by the model. The evaluators conducted a manual examination of several aspects of the movie, including posters, synopsis, actors, and genres. They subsequently assigned a label to a table, with label TRUE if the base movie and recommendation are relevant (True Positive TP), and FALSE vice versa (False Positive FP). Due to the fact that three data labelers evaluate each base movie pair and its suitability suggestions, there is potential for each labeler to provide a distinct assessment. At the conclusion of the review procedure, the final result will be chosen through majority voting, where the outcome with the highest frequency will emerge.

Assessments, like the one shown in Table 5, are conducted for each of the vectorization techniques utilizing TF-IDF and CountVectorizer. The evaluation results indicated that the TF-IDF approach outperformed the CountVectorizer method by 0.6% in terms of recommendation relevance for films. Regarding the television show, CountVectorizer exhibits a 3.24% superior performance compared to TF-IDF. Table 6 provides a summary of the performance of the recommendations.

Table 6
Performance Evaluation of Recommendation Model

Show Type	Vectorization Method	Precision@k
Film	TF-IDF	**55.3%**
	CountVectorizer	55%
Television Show	TF-IDF	41%
	CountVectorizer	**42.3%**

Table 6 presents a precision@k analysis that compares the recommendation model using various methods of extracting description features for different genres of films and television episodes. This table demonstrates that CountVectorizer performs better on television show descriptions, potentially because the length of the television program description is 13.3% shorter compared to the film description. The calculation is based on the median length description for each show type. Given this scenario, while utilizing CountVectorizer with a frequency-based methodology, it effectively captures significant information from the description. In order to handle film types with longer descriptions, a TF-IDF approach is necessar. This approach considers the importance of each word in a sentence with respect to a significant feature in a recommendation model.

A film suggestion with a precision value of 55.33% indicates that, on average, 10 recommendations are provided, out of which 5 films are relevant to the base movie. On the other hand, a television show with an average of 4 recommendations is considered accurate since it has a precision value of 42.3%. It remains highly reasonable and rational for a recommendation system.

Consider the scenario when we browse through a catalog of films on an internet platform and receive approximately 5 or 4 suggestions that align with our previous viewing history. It will be quite beneficial within the numerous available choices.

5. Conclusions

Recommendation models have improved by incorporating extraction features into feature descriptions through the use of text mining methodologies. The recommendation model was successfully developed using the content-based filtering method, utilizing various features such as show type, release year, age certification, runtime in minutes, IMDb score, and number of IMDb votes. This approach was chosen because the available data specifically pertains to the characteristics of the movies to be recommended. The feature extraction methods chosen for description were TF-IDF and CountVectorizer. Experimental findings indicated that TF-IDF yielded superior results for film types compared to CountVectorizers. The television show CountVectorizer outperformed TF-IDF in providing more relevant recommendations.

The performance of this approach can be further improved, considering that one of the limitations of TF-IDF and CountVectorizer is the presence of a sparse matrix where there are more zero values than non-zero values. A. comparison of the keyword extraction method to using the full token from the description as a feature in the recommendation model is being used to see how well it works.

Acknowledgement

This work is funded by the Research and Innovation, Universiti Kuala Lumpur. Also, the authors thank to Bina Nusantara University and Universiti Kuala Lumpur for supporting this research.

References

[1] N. Pavitha et al., “Movie recommendation and sentiment analysis using machine learning,” Global Transitions Proceedings, vol. 3, no. 1, pp. 279–284, 2022.
[2] R. Hasan MBA and J. Ferdous, “Dominance of AI and Machine Learning Techniques in Hybrid Movie Recommendation System Applying Text-to-number Conversion and Cosine Similarity Approaches,” 2024, doi: 10.32996/jcsts.
[3] S. S. Choudhury, S. N. Mohanty, and A. K. Jagadev, “Multimodal trust based recommender system with machine learning approaches for movie recommendation,” International Journal of Information Technology (Singapore), vol. 13, no. 2, pp. 475–482, Apr. 2021, doi: 10.1007/s41870-020-00553-2.
[4] A. Da’u and N. Salim, “Recommendation system based on deep learning methods: a systematic review and new directions,” Artif Intell Rev, vol. 53, no. 4, pp. 2709–2748, 2020.
[5] S. D. Lamkhede and C. Kofler, “Recommendations and results organization in netflix search,” RecSys 2021
- 15th ACM Conference on Recommender Systems, vol. 1, no. 1, pp. 577–579, 2021, doi: 10.1145/3460231.3474602.
[6] F. Tomasi, J. Cauteruccio, S. Kanoria, K. Ciosek, M. Rinaldi, and Z. Dai, “Automatic Music Playlist Generation via Simulation-based Reinforcement Learning,” Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 4948–4957, 2023, doi: 10.1145/3580305.3599777.
[7] Z. Liu et al., Monolith: Real Time Recommendation System With Collisionless Embedding Table, vol. 3303, no. 1. Association for Computing Machinery, 2022.
[8] J. Son and S. B. Kim, “Content-based filtering for recommendation systems using multiattribute networks,” Expert Syst Appl, vol. 89, pp. 404–412, 2017, doi: https://doi.org/10.1016/j.eswa.2017.08.008.
[9] R. Van Meteren and M. Van Someren, “Using content-based filtering for recommendation,” in Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop, Barcelona, 2000, pp. 47–56.
[10] B. Walek and V. Fojtik, “A hybrid recommender system for recommending relevant movies using an expert system,” Expert Syst Appl, vol. 158, Nov. 2020, doi: 10.1016/j.eswa.2020.113452.
[11] M. M. Afsar, T. Crump, and B. Far, “Reinforcement learning based recommender systems: A survey,” ACM Comput Surv, vol. 55, no. 7, pp. 1–38, 2022.
[12] S. Victor, “Netflix TV Shows and Movies,” kaggle.com. 2022.