ISSN 2356-5462
http://socj.telkomuniversity.ac.id/ijoict/

Intl. Journal on ICT
Vol. 9, No. 2, Dec 2023. pp. 1-14
doi: 10.21108/ijoict.v9i2.747

Movie Recommendation System Based on Synopsis Using Content-Based Filtering with TF-IDF and Cosine Similarity

Armadhani Hiro Juni Permana[1], *Agung Toto Wibowo[1]

[1]*School of Computing, Telkom University*
Jl. Telekomunikasi 1 Terusan Buah Batu, Bandung, Jawa Barat, Indonesia

*agungtoto@telkomuniversity.ac.id

Abstract

The recommendation systems have garnered significant attention in the fields of artificial intelligence and data analysis. In the current era, the entertainment industry has experienced substantial growth due to the widespread passion for film consumption. The sheer number and variety of films across different genres and titles pose a challenge for users in selecting suitable movies. The creation of a vital recommendation system that offers film suggestions aligned with user preferences and interests is crucial to aid users in this process. This research investigates the utilization of the TF-IDF algorithm and cosine similarity in the implementation of a content-based filtering approach. In this research, we examines the use of keyword extraction, and combination keyword extraction with genre to recommend movies in content-based flavored. The research employs a publicly dataset (MovieLens). Keywords are extracted from film synopses within the dataset. We use several performance evaluation metrics such as precision, recall, and f1-score. Experimental findings indicate limitations in the film recommendation system's ability to provide accurate and relevant suggestions to users. Specifically, the content-based filtering method with TF-IDF and cosine similarity, integrating keyword extraction using the Gensim library, is less effective in generating relevant film recommendations.

Keywords: Recommendation System; Keyword Extraction; Content-Based Filtering; TF-IDF; Cosine Similarity.

I. INTRODUCTION

THE world of entertainment, particularly the film industry, is experiencing rapid growth and popularity in recent times. The increasing number of films being produced on a daily basis presents a challenge for viewers who are faced with a wide range of choices [1]. This abundance of options is a common problem faced by many users, making it difficult for them to find films that align with their preferences. Each user has their own unique taste and interests, highlighting the importance of having a recommendation system to assist in narrowing down the selection process [2].

A film recommendation system is designed to filter and refine user preferences, providing personalized film recommendations based on individual interests and preferences [2]. User interests can be influenced by several factors, such as genre, ratings, or even the film's synopsis [3]. There are content-based recommender system which use item descriptions and genres [4] [5]. However, there is no research which use keyword extraction to make a recommendation. Numerous studies have been conducted using various approaches to develop film

Received on 23 Jun 2023. Revised on 24 Jul 2023. Accepted and Published on 08 Dec 2023.

recommendation systems. Collaborative filtering and content-based filtering are two widely adopted methodologies in recommendation systems. Content-based filtering involves filtering content based on specific information or attributes to generate recommendations tailored to user profiles or interactions [6]. Conversely, collaborative filtering focuses on identifying users with similar preferences and generating recommendations based on the preferences of these similar users [7].

In this research, we employ a content-based filtering approach utilizing the TF-IDF algorithm and cosine similarity. This approach is chosen due to its ability to anticipate user preferences and generate film recommendations based on content extracted from user histories or interactions. The dataset used in this study consists of publicly available data (MovieLens) containing comprehensive film information. One crucial piece of information is the film synopsis. We compare recommendation models based on genre, keywords, and a combination of both variables. The synopses are extracted into keywords using the Python library, Gensim. We use keyword extraction because it can take the essence of the story of a movie that users like. The essence can be used as an attribute to provide recommendations based on the essence of similar movie stories. The keywords are then processed using feature extraction utilizing the TF-IDF method. TF-IDF transforms each word in the document into numerical values [8]. The similarity between films is computed using cosine similarity based on these numerical values. The recommendation system generates film recommendations by considering the top-k films with the highest similarity scores, k represents the number of recommendations of items that have the most similarity.

II. LITERATURE REVIEW

A. Related Works

This study was grounded in a comprehensive review of relevant existing research sources, which encompassed both the methodological approaches and the specific topic of interest. The primary objective of incorporating these references was to establish a strong theoretical foundation and ascertain the limitations associated with the content-based filtering method in the context of film.

To ease human efforts in selecting movies, a content-based model has been built to provide movie recommendations by analyzing movie descriptions. For example, when the movie "GoldenEye" was tested, the model suggested "Skyfall". The model produced several similarities score: 66.73% (CountVectorizer), 13.14% (Jaccard Recommender), 14.34% (TF-IDF Keyword), and 9.87% (TF-IDF Plot Overview). The best performance is shown by the calculation of similarity using CountVectorizer with a similarity value of 66.73%.[4].

A content-based filtering method is proposed with recommendations based on specific topics that represent aspects depicted in movies. WordNet and topic modeling are utilized to assign topics to movies based on their plot summaries. The accuracy of this assignment is then manually calculated using popular Telugu movies on IMDb. This approach represents a simplistic yet effective method for discerning the underlying topics within a movie [5].

A study using the Cosine Similarity and K-Nearest Neighbors (KNN) algorithms with the aim of providing accurate recommendations with reduced complexity. This recommendation system will recommend movies based on cosine similarity. Subsequently, a normalized popularity score will be utilized, incorporating KNN functionality to obtain the nearest neighbors recommended to the user. The KNN algorithm is implemented in this model along with the cosine similarity principle because it provides higher accuracy compared to other distance metrics, while maintaining relatively low complexity [7].

The issues in Collaborative Filtering approaches, namely cold start, trust, and privacy. These three weaknesses can sometimes be problematic, and important opportunities may be missed as a result. They created a content-based filtering recommendation system using the cosine similarity algorithm. The process begins with cleaning and constructing the dataset, then obtaining numerical features from the data. Next, the cosine similarity function is applied to determine the similarity between users' previous preferences and available jobs. Finally, the top job recommendations are obtained based on the scores derived from this similarity measure [9].

The recommendation system implemented by Netflix utilizes similarity analysis through the application of the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm and Cosine Similarity, but the relevance was limited due to the scarcity of occurrences in the corpus used. This recommendation system only utilized the title and description as data. In other words, applying TF-IDF and Cosine Similarity algorithms on a limited corpus like this would not produce meaningful outcomes [10].

A topic-based news recommendation system based on keyword extraction, experimental results unequivocally support the notion that topic words can be derived effectively from daily keywords. Theoretical analysis, complemented by extensive numerical experiments, substantiates the system's ability to enhance the anticipated weights of keywords and successfully identify pertinent keyword occurrences within a defined time period. However, addressing the challenge of identifying and eliminating recurrent words, particularly precise nouns, remains an area that warrants additional research efforts. Moreover, it is important to acknowledge that the dataset utilized in this study is relatively constrained, primarily focusing on the political domain [11].

A recommendation system for Facebook photo posts based on a topic model, which can enhance the engagement level of fan page posts. They devised an automated approach to streamline the process of selecting posts for ad creation, effectively replacing manual selection tasks previously undertaken by social media managers and alleviating their daily workload. By leveraging the Gensim module within the topic modeling framework, they transformed words into vectors and integrated them into the TF-IDF model. Through this conversion, individual weight vectors were assigned to each word. Subsequently, these weighted vectors were employed in conjunction with the Gensim library to determine the optimal number of topics, thereby generating distinct LSI/LDA models for evaluating cosine similarity in relation to subsequent target posts [12].

B. Recommendation Systems

The recommendation system functions to provide product recommendations based on user preferences, past history, and information to obtain desired products [13]. The commonly used methods in building recommendation systems are collaborative filtering and content-based filtering. Collaborative filtering is a method used to evaluate and predict items based on the opinions and similarities of other users. On the other hand, content-based filtering is a method used to recommend a product based on the availability of similar content [14].

The recommendation systems have been implemented across diverse domains, demonstrating their positive influence. For example, within the e-commerce landscape in Indonesia, exemplified by Amazon and eBay, recommender systems play a crucial role in presenting customers with relevant products [15]. Similarly, the music industry benefits from personalized recommendation algorithms on platforms like Spotify and Pandora, which suggest music based on song genres, artists, and playlists to individual users [16]. Notably, platforms like Netflix employ recommendation algorithms to provide movie suggestions aligned with users' interests, viewing history, and ratings [10]. This can facilitate users in easily discovering content that aligns with their interests, thus enhancing their satisfaction in their film-watching experience on these platforms.

C. Content-Based Filtering

Content-Based Filtering is a method of recommendation systems that relies on data or information about users' preferences based on their past interactions. The algorithms used in content-based filtering attempt to recommend items by measuring their similarity to items that users have rated positively in the past. The most suitable items are then recommended by comparing the similarity between the items previously rated by the user and other items in the dataset [17].

Several algorithms are commonly used in content-based recommendation systems. These include TF-IDF (Term Frequency-Inverse Document Frequency), which calculates the importance of words in a film's description or synopsis, and cosine similarity, which measures the similarity between films based on their feature vectors. Other algorithms such as decision trees and Bayesian classifiers can also be utilized for classification and prediction tasks in content-based systems [18].

The input for content-based recommendation systems includes various film-related features. This ranges from textual data such as film titles, genres, synopses, and user-generated tags to visual features like film posters or even audio features like soundtracks. However, in this study, we will specifically focus on movies domain with the features of genres, keywords, and the combination of both. The combination of genre and keyword refers to the merging of both features using concatenation operation. For example, if a film i has genre = {"action", "comedy"} and keyword = {"woody", "birthday"}, the combination of features would be combination = {"action", "comedy", "woody", "birthday"}.This is conducted to compare the performance of different models in generating optimal recommendations based on user preferences. By leveraging these inputs, content-based systems can capture the unique characteristics of films and generate recommendations that are tailored to the user's preferences and interests.

D. Keyword Extraction for The Synopsis

The utilization of keywords in recommendation systems can provide several advantages. Keywords offer a concise representation of the content and essence of an item, such as capturing the synopsis of a film. By incorporating keywords, recommendation algorithms can capture the core features and themes of an item, enabling more precise matching with user preferences. Furthermore, keywords facilitate efficient and scalable processing of large datasets. They provide a structured and standardized approach to represent the characteristics of films, making it easier to retrieve and analyze relevant information [19].

Keyword extraction is the process of extracting keywords from a synopsis. The sentences in the synopsis are often lengthy, requiring a process of extracting the essence and important words from those sentences. Keyword extraction is performed using the Gensim library in Python. Gensim is an open-source natural language processing library in Python that can automatically extract keywords from documents and discover text similarity. In this study, Gensim will be utilized as the keyword extractor [20].

E. TF-IDF

TF-IDF stands for Term Frequency Inverse Document Frequency of Records. TF-IDF calculates the relevance of a word within a document. The TF-IDF value of a word indicates its level of characterization in the document. A higher value of a word signifies its importance in the document [10]. Based on the above explanation, TF-IDF is calculated using equation (1) as follows::

$$tfidf_i = tf_i \times idf_i \tag{1}$$

In equation (1), $tfidf_i$ represents the calculation of Term Frequency Inverse Document Frequency for a term (i), which is obtained by multiplying the Term Frequency (tf_i) in equation (2) with the Inverse Document Frequency (idf_i) in equation (3). In this study, the term referred to can be a genre (e.g., "action") or a keyword (e.g., "woody").

$$tf_i = \frac{n_{i,u}}{|u|} \tag{2}$$

Where $n_{i,u}$ represents the number of occurrences of the term i in the film u term list. Meanwhile, $|u|$ denotes the length of the term possessed by film u.

$$idf_i = log\left(\frac{|D|}{df_i}\right) \tag{3}$$

Where $|D|$ represents the total number of films in the dataset, and df_i represents the number of documents that contain the term i.

F. Cosine Similarity

Cosine similarity is an algorithm used to determine the similarity between a target data and a given set of data. It measures the similarity between two documents based on their differences in size and calculates the cosine angle between the two vectors in a multidimensional space. In this study, we leverage cosine similarity as a computational measure to determine the presence or absence of specific terms in an item. Recommendations are then formulated based on the degree of similarity between the respective item and the active item associated with a specific user. The probability of recommending an item to a user increases with higher levels of similarity exhibited towards the active item. The application of cosine similarity requires treating the data objects in the dataset as vectors, enabling comprehensive vector-based analysis and comparison. This approach facilitates the efficient and accurate generation of recommendations in the research context [21]. Based on the above explanation, cosine similarity is computed using the following formula:

$$Cos(x, y) = \frac{x \cdot y}{\|x\| * \|y\|} \tag{4}$$

Where $Cos(x, y)$ represent the calculation of cosine similarity between x and y, x is the vector obtained from the TF-IDF calculation on the film chosen as the active item, and y is the vector obtained from the TF-IDF calculation on the film chosen as the reference item. $||x||$ denotes the unit length of vector x, while $||y||$ denotes the unit length of vector y.

III. RESEARCH METHOD

In this phase, the recommendation method will be implemented by constructing the top-k similar item using TF-IDF algorithm and cosine similarity. Fig. 1 provides an overview of the system flow developed in this study.

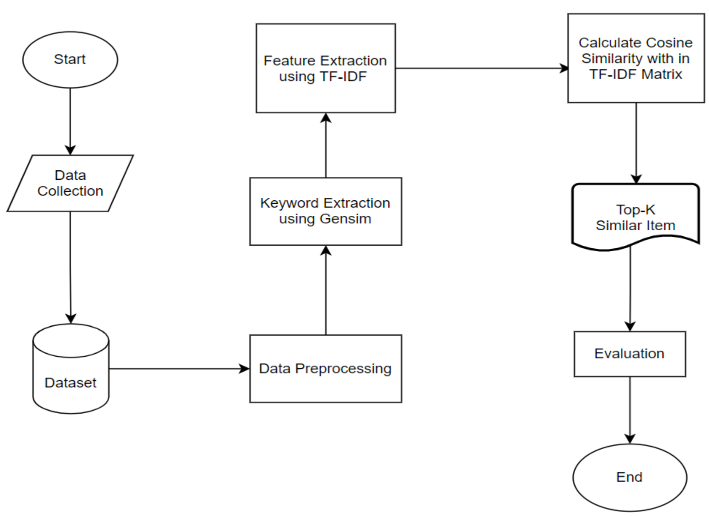

Fig. 1. Recommendation System Flowchart

Based on Fig. 1, the system flow commences with the collection of data in CSV format, containing synopses. The subsequent stage involves preprocessing, where the data is cleansed from noise and other disturbances. Next, keyword extraction from synopses is performed using Gensim. Subsequently, feature extraction is conducted using the TF-IDF method. The purpose of this feature extraction is to transform words into vectors and obtain weights for each word.

In the following stage, cosine similarity calculations are performed between every pair of items. The results are then utilized to provide top-ranked film recommendations (top-k similarity) based on the calculated synopsis similarity. K represents the number of recommendations of items that have the most similarity. Subsequently, evaluation is carried out to measure the performance of the constructed model using precision, recall, and f1-score metrics.

A. Data Collection

Data collection is conducted using a public dataset (MovieLens) consisting of seven CSV files: credits, keywords, links, links_small, movies_metadata, ratings, and ratings_small. However, only the data from movies_metadata, links_small, and ratings_small are utilized in this study. In the movies_metadata dataset, the selected features include id, imdbId, title, genres, and overview. This dataset comprises 45,466 rows and 5 columns, providing valuable information about the films. As for the links_small and ratings_small data, they are merged to represent the complete film history, encompassing 99,806 rows and 10 columns, capturing the viewing records of all users. The combined dataset incorporates features such as userId, movieId, rating, timestamp, imdbId, tmdbId, id, title, genres, and synopsis. Notably, the keywords data available in the dataset is not used in its original form, as it will be replaced with keywords obtained through the keyword extraction process conducted during the preprocessing stage. This data collection phase ensures the availability of comprehensive and relevant information required for subsequent analysis and the development of an effective recommendation system.

B. Data Pre-processing

The pre-processing stage is crucial in transforming raw data into a more usable format. It involves various steps aimed at addressing incompleteness, inconsistencies, and irregularities within the data. The first step in pre-processing is case folding, which involves converting all uppercase letters to lowercase. This normalization process ensures consistency and facilitates subsequent analysis and text mining tasks. Additionally, the elimination of irrelevant words is performed using the nltk and sastrawi libraries. By removing stopwords and non-relevant terms, the dataset becomes more focused and informative. Furthermore, data cleansing techniques are applied to remove symbols and non-alphanumeric characters that may hinder the accuracy and reliability of subsequent analyses. This step contributes to data standardization and ensures consistency in the dataset.

Table I provides an example of pre-processing result made to the data. By implementing these pre-processing techniques, the research aims to improve the quality, integrity, and usability of the data, ultimately leading to more accurate and insightful findings.

TABLE I
EXAMPLE OF DATA PRE-PROCESSING

imdbId	Input	Output
862	Under the guidance of Woody, the toys belonging to Andy enjoy a contented existence within his room, until the arrival of Buzz Lightyear on Andy's birthday. Overwhelmed by the fear of losing his cherished position in Andy's affections, Woody devises a scheme to undermine Buzz. However, as fate intervenes and separates Buzz and Woody from their owner, the two eventually realize the importance of setting aside their conflicts and working together.	guidance woody toys belonging andy enjoy contented existence within room arrival buzz lightyear andys birthday overwhelmed fear losing cherished position andys affections woody devises scheme undermine buzz however fate intervenes separates buzz woody owner two eventually realize importance setting aside conflicts working together

C. Keyword Extraction

In the keyword extraction process, the dataset will be tagged using the Gensim library in Python. This process involves identifying keywords that effectively describe the content of each synopsis. By extracting meaningful keywords, the system aims to capture the essence of the synopses and improve the accuracy of the recommendation process. Table II showcases a sample of the extracted keywords obtained through this methodology.

TABLE II
EXAMPLE OF KEYWORD EXTRACTION

imdbId	Synopsis	Keyword
862	guidance woody toys belonging andy enjoy contented existence within room arrival buzz lightyear andys birthday overwhelmed fear losing cherished position andys affections woody devises scheme undermine buzz however fate intervenes separates buzz woody owner two eventually realize importance setting aside conflicts working together	woody andys buzz birthday

D. Feature Extraction

In this phase, the preprocessed data will be transformed into vectors or matrices. In this study, the method used is TF-IDF (Term Frequency - Inverse Document Frequency). TF-IDF is a widely used method in natural language processing to measure the importance of terms in a document corpus. This method aims to highlight the significance of terms by considering their frequency within a specific document and their inverse frequency across the entire corpus. Through TF-IDF, relevant features are extracted from textual data, enabling subsequent analysis and modeling stages to effectively capture inherent patterns and semantic relationships. Table III presents an example of the feature extraction results obtained through the TF-IDF method from the dataset used in the program.

TABLE III
THE RESULT OF FEATURE EXTRACTION

Title	action	action adventure	action adventure animation	…	western war	western war history	western war romance
tekkonkinkreet	0.233515	0.369521	0.607958	…	0	0	0
the sky crawlers	0.233515	0.369521	0.607958	…	0	0	0
superman/batman: apocalypse	0.233515	0.369521	0.607958	…	0	0	0
justice league: crisis on two earths	0.233515	0.369521	0.607958	…	0	0	0
batman vs. robin	0.233515	0.369521	0.607958	…	0	0	0
operator 13	0	0	0	…	0.359418	0	0.411079
the plainsman	0	0	0	…	0.428157	0.489698	0
for a few extra dollars	0	0	0	…	0.825822	0	0
adios sabata	0.168587	0	0	…	0.554825	0	0
the deerslayer	0	0	0	…	0.526346	0	0

E. Calculate Cosine Similarity

In the cosine similarity calculation phase, the similarity level between two document representation vectors is measured. In the previous stage, the documents are transformed into numerical vector representations using techniques like TF-IDF (Term Frequency-Inverse Document Frequency). Then, each document vector is normalized by dividing it by its Euclidean norm. This is done to ensure that each document vector has a consistent length, enabling objective comparisons of similarity between documents.

For example, we have a corpus of 10 documents based on the results of the previous tf-idf calculation. Next, using the cosine similarity formula, the cosine angle between the two document representation vectors is computed. This process yields a value ranging from -1 to 1, where a value of 1 indicates a high similarity between the two documents, while a value of -1 indicates a high dissimilarity:

$$Cos(adios\ sabata, the\ plainsman) = \frac{adios\ sabata\ .the\ plainsman}{\|adios\ sabata\| * \|the\ plainsman\|} \tag{5}$$

$$= \frac{(0.1687587 * 0.0) + (0.0 * 0.0) + (0.0 * 0.0) + \cdots + (0.554825 * 0.428157) + (0.0 * 0.489698) + (0.0 * 0.0)}{\left(\sqrt{0.168587^2 + 0.0^2 + 0.0^2 + \cdots + 0.554825^2 + 0.0^2 + 0.0^2}\right) * \left(\sqrt{0.0^2 + 0.0^2 + 0.0^2 + \cdots + 0.428157^2 + 0.489698^2 + 0.0^2}\right)}$$

$$= 0.348326$$

The calculation above yields a value of 0.348326. This indicates that there is a low similarity between the films titled "Adios Sabata" and "The Plainsman". Table IV presents the results of the cosine similarity calculation based on the tfidf matrix calculation in the previous stage.

TABLE IV
THE RESULT OF COSINE SIMILARITY CALCULATION

Title	**tekkonkin kreet**	**the sky crawlers**	**superman/ batman: apocalypse**	**justice league: crisis on two earths**	**batman vs. robin**	**operator 13**	**the plainsman**	**for a few extra dollars**	**adios sabata**	**the deerslayer**
tekkonkin kreet	1	1	1	1	1	0	0	0	0.039368	0.055279
the sky crawlers	1	1	1	1	1	0	0	0	0.039368	0.055279
superman/ batman: apocalypse	1	1	1	1	1	0	0	0	0.039368	0.055279
justice league: crisis on two earths	1	1	1	1	1	0	0	0	0.039368	0.055279
batman vs. robin	1	1	1	1	1	0	0	0	0.039368	0.055279
operator 13	0	0	0	0	0	1	0.272873	0.435224	0.292404	0.277394
the plainsman	0	0	0	0	0	0.272873	1	0.518461	0.348326	0.330446

Title	tekkonkin kreet	the sky crawlers	superman/ batman: apocalypse	justice league: crisis on two earths	batman vs. robin	operator 13	the plainsman	for a few extra dollars	adios sabata	the deerslayer
for a few extra dollars	0	0	0	0	0	0.435224	0.518461	1	0.671846	0.637360
adios sabata	0.039368	0.039368	0.039368	0.039368	0.039368	0.292404	0.348326	0.671846	1	0.428208
the deerslayer	0.055279	0.055279	0.055279	0.055279	0.055279	0.277394	0.330446	0.637360	0.428208	1

F. Evaluation

In the evaluation phase, the performance of the recommendation system will be assessed. In this research, performance evaluation metrics like precision, recall, and f1-score are employed. As previously mentioned, these evaluation metrics will be computed using the equations (6) for precision, (7) for recall, and (8) for the f1-score [22]:

$$\text{precision} = \frac{TP}{TP + FP} \tag{6}$$

Where TP represents the number of true positives, referring to the correct prediction of actual positive data as positive. Meanwhile, FP represents the number of false positives, signifying the erroneous prediction of actual negative data as positive. Precision is a measure that quantifies the ratio of TP to the total number of data instances predicted as positive ($TP + FP$).

$$\text{recall} = \frac{TP}{TP + FN} \tag{7}$$

Where FN signifies the count of false negatives, indicating the cases where the actual positive data is mistakenly predicted as negative. On the other hand, recall is a measure that calculates the ratio of TP to the total number of actual positive data instances ($TP + FN$).

$$\text{f1-score} = 2 * \frac{(recall * precision)}{(recall + precision)} \tag{8}$$

The f1-score is a metric that signifies the harmonic mean of recall and precision. It provides a balanced measure that considers both precision and recall in a single metric.

IV. RESULTS AND DISCUSSION

This research utilizes a movie dataset consisting of 45,443 films and a total movie history of 99,806 records from all users who have watched movies. These two datasets are merged using the Pandas Dataframe library in the Python programming language. After combining the two datasets, the data will be evaluated by computing precision and recall based on the viewing history of each user for those movies.

A. Performance Evaluation

Throughout the evaluation process, we conducted experiments using two scenarios with an active item as the query. The first scenario involves recommendations based on the active item's initial user rating, while the second scenario focuses on recommendations based on the active item's user preferences.

1) Testing Based on the First-Rated Active Item

In this testing phase, the researcher obtained user history data related to their film-watching activities. The utilized data consisted of the initial film-watching histories of the users. This data will be used as a testing sample for the developed recommendation model. Subsequently, precision, recall, and f1-score will be calculated by comparing the recommendations provided by the model with the users' film-watching histories. Table V displays an example of films that are considered as active items based on the first given rating.

TABLE V
EXAMPLE OF FILM SCENARIOS BASED ON THE FIRST GIVEN RATING

Name	Title	UserID	Rating
Hiro	Toy Story	1	1
	Jumanji	1	1
	Grumpier Old Men	1	4
	Waiting to Exhate	1	1
	Father of the Bride Part II	1	4
	Subdue	1	5
	Century of Birthing	1	1
Budi	Grumpier Old Men	2	4
	Father of the Bride Part II	2	3
	Toy Story	2	3
	Century of Birthing	2	4
	Jumanji	2	2

2) Testing Based on the Active Item Preferred by Users

In this testing phase, the researchers collected user history data related to their film-watching activities. The data used consisted of film-watching history data that had been liked by users, taking into account the ratings given by those users. This study assumes that users who like a film will give ratings of 4 or 5. This data was used as a testing sample for the evaluation of the system's performance. Table VI displays examples of films selected as active items based on user preferences.

TABLE VI
EXAMPLE OF FILM SCENARIOS BASED ON USER PREFERENCES

Name	Title	UserID	Rating
Hiro	Toy Story	1	1
	Jumanji	1	1
	Grumpier Old Men	1	4
	Waiting to Exhate	1	1
	Father of the Bride Part II	1	4
	Subdue	1	5
	Century of Birthing	1	1
Budi	Grumpier Old Men	2	4
	Father of the Bride Part II	2	3
	Toy Story	2	3
	Century of Birthing	2	4
	Jumanji	2	2

B. Experiment Results

In this study, a content-based filtering system was tested to evaluate the performance of the recommendation model in providing film recommendations to users. The testing was conducted using a film dataset consisting of genres and related keyword information. This system testing involved selecting relevant parameters to measure the quality of recommendations. The parameters used include precision, recall, and f1-score. Precision measures the extent to which the recommendations align with user preferences, while recall describes how many relevant recommendations can be found by the users. The F1-Score combines these two metrics to provide an overall view of the system's performance.

Furthermore, the testing process involved several different testing scenarios. Firstly, testing was conducted by considering the utilization of genres or keywords to provide film recommendations. Next, the recommendation model was evaluated using a combination of genre and keyword information. Subsequently, a comparison was made between the top-10 recommendations and the top-20 recommendations for all scenarios. The top-10 recommendations represent a model that consists of 10 recommendations based on attribute similarity, and the same applies to the top-20 recommendations. In order to obtain relevant benchmarks, the results were also compared with previous testing experiments. Table III presents the results of the recall, precision, and f1-score metrics calculation.

TABLE VII
SYSTEM EVALUATION RESULTS

Input	Scenario	Top-10 Recommendation			Top-20 Recommendation		
		Recall	Precision	F1-Score	Recall	Precision	F1-Score
Genres	Recommendation based on initial rating user.	0.06	0.40	0.09	0.09	0.34	0.13
Keyword		0.05	0.06	0.01	0.09	0.04	0.01
Genres + Keyword		0.01	0.14	0.02	0.01	0.11	0.02
Genres	Recommendation based on user preferences	0.10	0.30	0.12	0.15	0.25	0.15
Keyword		0.06	0.08	0.09	0.01	0.08	0.01
Genres + Keyword		.02	0.10	0.03	0.03	0.08	0.04

C. Analysis of Experiment Results

In this evaluation, we assessed the performance of the system by considering performance metrics, namely recall, precision, and F1-score, to evaluate the system's ability to provide relevant recommendations to users. The test results reveal that there is a need to enhance the performance of the recommendation system. In the genre-based recommendation scenario, we observed a recall of 0.06, precision of 0.4, and F1-score of 0.09 for the top 10 recommendations, and a recall of 0.09, precision of 0.34, and F1-score of 0.13 for the top 20 recommendations. Despite precision being relatively higher than recall, these figures remain generally low. Similar results were obtained in the keyword-based recommendation scenario, where the recall, precision, and F1-score were 0.05, 0.06, and 0.01, respectively, for the top 10 recommendations, and 0.09, 0.04, and 0.01, respectively, for the top 20 recommendations. In the combined scenario involving both genre and keywords, the evaluation results demonstrated a slight improvement compared to the keyword-only scenario, but it still fell short compared to the genre-only scenario. The recall, precision, and F1-score for the top 10 recommendations were 0.01, 0.14, and 0.02, respectively, and for the top 20 recommendations were 0.01, 0.11, and 0.02, respectively. These results indicate that the system still has limitations in accurately remembering the correct items and providing accurate recommendations to users.

In the second evaluation, when user preferences were incorporated into the recommendation scenarios, the results showed an increase in performance. In the genre-based recommendation scenario with user preferences, the recall, precision, and F1-score for the top 10 recommendations were 0.1, 0.3, and 0.12, respectively, and for the top 20 recommendations were 0.15, 0.25, and 0.15, respectively. These results exhibited significant improvement compared to the previous scenario. However, further improvements are necessary to deliver more relevant recommendations. In the keyword-based recommendation scenario with user preferences, the evaluation results showed a recall of 0.06, precision of 0.08, and F1-score of 0.09 for the top 10 recommendations, and a recall of 0.01, precision of 0.08, and F1-score of 0.01 for the top 20 recommendations. Although there was a slight improvement compared to the previous scenario, these figures still indicate limitations in accurately recalling the relevant items and providing accurate recommendations. The combined scenario involving genre, keywords, and user preferences also demonstrated slightly better results than using keywords alone. In this scenario, the recall, precision, and F1-score for the top 10 recommendations were 0.02, 0.1, and 0.03, respectively, and for the top 20 recommendations were 0.03, 0.08, and 0.04, respectively, showing a significant improvement compared to the previous scenario. However, despite the notable performance improvement, the system still requires further refinement to provide better recommendations.

Overall, the performance of the f1-score calculations did not reach 0.1. In the scenario using keywords, the overall precision, recall, and f1-score values did not exceed 0.1. This is due to the small overlap of keywords generated during the keyword extraction process between different films. As a result, it becomes challenging to find film recommendations with similar keywords. Meanwhile, in the scenario using genres, only the f1-score did not reach 0.1. This is because many films share similar genres but have significant differences in terms of plot, storyline, and other aspects. As a result, there are multiple high similarity scores that lead to suboptimal recommendations.

From the above testing results, it is evident that scenarios involving user preferences tend to yield better performance compared to scenarios relying solely on genre or keyword information. The combination of genre and keyword in the recommendation scenario also produces slightly better results than the keyword-only scenario. However, all scenarios still have room for improvement, particularly in terms of recall and f1-score. Therefore, it can be concluded that the use of keywords with the Gensim library is not suitable for building a film recommendation system.

V. CONCLUSION

The results of this study highlight the limitations of the evaluated recommendation system in providing accurate and relevant recommendations to users. Several factors contribute to the observed challenges, including the scarcity of relevant information in the reference dataset, the inherent complexity of user preferences and patterns, and the need for more advanced data processing and analysis techniques. To improve the performance of the recommendation system, future research should focus on expanding and enriching the reference dataset with comprehensive and pertinent information. This will enable the system to generate more precise and tailored recommendations.

Additionally, the adoption of advanced algorithms and techniques can enhance the system's ability to capture intricate user preferences and patterns, leading to improved recommendation accuracy. Furthermore, integrating multiple recommendation approaches, such as combining content-based and collaborative filtering methods, holds promise for enhancing the recommendation system's capabilities. This hybrid approach can leverage the strengths of each method and address the limitations of individual approaches. Moreover, incorporating additional personalization factors into the recommendation process can further refine the recommendations and cater to the unique preferences of each user. To gain a comprehensive understanding of the recommendation system's performance, future studies should undertake rigorous testing using diverse datasets and carefully consider various scenarios. This will provide valuable insights into the system's strengths, weaknesses, and its performance across different user profiles and contexts. Ultimately, the overarching objective of future research endeavors should center on developing a robust and effective recommendation system that consistently delivers accurate and relevant recommendations to users. By enhancing the user experience and satisfaction in accessing recommended content, such a system can significantly contribute to the advancement of the entertainment industry and improve the overall user engagement and enjoyment.

REFERENCES

[1] S. Chakrabarti *et al.*, "Annual Computing and Communication Workshop and Conference (CCWC)", *2018 IEEE 8th*, 8th-10th January, 2018, University of Nevada, Las Vegas, USA.

[2] S. Halder, A. M. J. Sarkar, and Y. K. Lee, "Movie recommendation system based on movie swarm," in *Proceedings - 2nd International Conference on Cloud and Green Computing and 2nd International Conference on Social Computing and Its Applications, CGC/SCA 2012*, 2012, pp. 804–809. doi: 10.1109/CGC.2012.121.

[3] Khan, E. M., Mukta, M. S. H., Ali, M. E., & Mahmud, J. (2020). "Predicting users' movie preference and rating behavior from personality and values". *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *10*(3), 1-25

[4] S. Aditya Narayan, H. Kumaar, D. Sathya Narayanan, S. Srikumaran, and S. Veni, "Content-based Movie Recommender System Using Keywords and Plot Overview," *2022 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2022*, pp. 49–53, 2022, doi: 10.1109/WISPNET54241.2022.9767172.

[5] P. Sirisha, G. L. Devi, and N. Ramesh, "Plot-Topic based Movie Recommendation System using WordNet," *Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development, INDIACom 2022*, pp. 45–49, 2022, doi: 10.23919/INDIACOM54597.2022.9763244.

[6] Lops, P., De Gemmis, M., & Semeraro, G. (2011). 'Content-based recommender systems: State of the art and trends'. Recommender systems handbook, 73-105.

[7] R. H. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav, "Movie Recommendation System using Cosine Similarity and KNN," *Int J Eng Adv Technol*, vol. 9, no. 5, pp. 556–559, Jun. 2020, doi: 10.35940/ijeat.E9666.069520.

[8] H. Bhowmick, A. Chatterjee, and J. Sen, "Comprehensive Movie Recommendation System."

[9] Domeniconi, G., Moro, G., Pagliarani, A., Pasini, K., & Pasolini, R. (2016, February). "Job recommendation from semantic similarity of LinkedIn users' skills". In *International Conference on Pattern Recognition Applications and Methods* (Vol. 2, pp. 270-277). SciTePress

[10] M. Chiny, M. Chihab, O. Bencharef, and Y. Chihab, "Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms," Scitepress, May 2022, pp. 15–20. doi: 10.5220/0010727500003101.

[11] Z. Wang, K. Hahn, Y. Kim, S. Song, and J. M. Seo, "A News-Topic Recommender System Based on Keywords Extraction," *Multimed Tools Appl*, vol. 77, no. 4, pp. 4339–4353, Feb. 2018, doi: 10.1007/s11042-017-5513-0.

[12] C. H. Liao, L. X. Chen, J. C. Yang, and S. M. Yuan, "A Photo Post Recommendation System Based on Topic Model for Improving Facebook Fan Page Engagement," *Symmetry (Basel)*, vol. 12, no. 7, Jul. 2020, doi: 10.3390/sym12071105.

[13] M. Mahendra Wiputra and Y. Jauhari Shandi, "Perancangan Sistem Rekomendasi Menggunakan Metode Collaborative Filtering Dengan Studi Kasus Perancangan Website Rekomendasi Film", 2021.

[14] F. Batari Agung Larasati and H. Februariyanti,“Sistem Rekomendasi Product Emina Cosmetics Dengan Menggunakan Metode Content - Based Filtering”, *Jurnal Manajemen Informatika & Sistem Informasi*, 2021, doi: https://doi.org/10.36595/misi.v4i1.250.

[15] J. Ben Schafer, J. Konstan, and J. Riedl, “Recommender Systems in E-Commerce.” [Online]. Available: www.reel.com

[16] H.-C. Chen and A. L. P. Chen, “A Music Recommendation System Based on Music and User Grouping”, 2005.

[17] P. B. Thorat, R. M. Goudar, and S. Barve, “Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System,” 2015.

[18] Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. Egyptian informatics journal, 16(3), 261-273.

[19] X. Li, “A Method for Extracting Keywords from English Literature Based on Location Feature Weighting,” in *International Conference on Communication Technology Proceedings, ICCT*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 1457–1460. doi: 10.1109/ICCT50939.2020.9295829.

[20] Řehůřek, Radim, and Petr Sojka. "Software framework for topic modelling with large corpora." (2010).

[21] Su, X., & Khoshgoftaar, T. M. (2009). “A survey of collaborative filtering techniques”. *Advances in artificial intelligence*, 2009.

[22] M. J. Mokarrama, S. Khatun, and M. S. Arefin, “A content-based recommender system for choosing universities,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 28, no. 4, pp. 2128–2142, Jul. 2020, doi: 10.3906/ELK-1911-37.