Type of Contribution:

Editorial
Research Paper
Review Paper
Case Study

INTRO: JURNAL INFORMATIKA DAN TEKNIK ELEKTRO
DOI: 10.51747/intro.v5i1.p42-55

ISSN 3025-602X

This article contributes to:

SUSTAINABLE DEVELOPMENT GOALS
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE

Movie Recommendation System Using Content-Based Filtering with TF-IDF and Cosine Similarity Implemented in Python

Heru Tri Ahmanto[1*], Sugeng Hendra Wijaya[1], I Gede Wiryawan[2]

[1] Faculty of Engineering, Universitas Bakti Indonesia, 68462, **Indonesia**
[2] Information Technology Departement, Politeknik Negeri Jember, 68121, **Indonesia**
[*] heru@ubibanyuwangi.ac.id

Abstract

The rapid growth of digital movie streaming platforms has significantly increased the number of movies available to users. Although this abundance of content provides greater viewing options, it also creates challenges for users in identifying movies that match their preferences. Conventional search methods are often inefficient because users must manually browse large collections of movies before making a selection. Therefore, an intelligent recommendation system is required to provide personalized movie recommendations based on movie characteristics. This study aims to develop a movie recommendation system using the Content-Based Filtering (CBF) approach implemented in Python. The proposed system utilizes movie metadata, including genres, overviews, keywords, cast members, and directors, to represent each movie. Text preprocessing techniques are applied to clean and normalize the textual data before transforming it into numerical vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) method. Subsequently, Cosine Similarity is employed to calculate the similarity between movies and generate recommendation lists according to user preferences. The system is developed using Python and several supporting libraries, including Pandas, NumPy, and Scikit-learn. Mean Average Precision (MAP) of 0.876, and an NDCG@10 of 0.904. The system also obtained a Recall@10 of 0.893, indicating that a high proportion of relevant movies were successfully retrieved. These results demonstrate that the integration of TF-IDF and Cosine Similarity with multiple movie metadata features can effectively generate relevant and well-ranked movie recommendations.

Article Info
Submitted: 2026-04-30
Revised: 2026-06-09
Accepted: 2026-06-21
Published: 2026-06-28

Publisher
Universitas Panca Marga

Keywords: Movie Recommendation System, Content-Based Filtering, TF-IDF, Cosine Similarity, Python, Text Mining.

1. Introduction

The rapid advancement of information technology has transformed the way people access digital entertainment. Online streaming services such as Netflix, Disney+, Amazon Prime Video, and similar platforms provide users with thousands of movies from various genres, languages, and production years. Although the availability of extensive movie collections offers greater flexibility, it also introduces a new challenge known as information overload [1], [2]. Users often experience difficulties when selecting movies that correspond to their individual preferences because they must navigate through a large amount of available content.

Recommendation systems have become one of the most effective solutions to overcome this problem. A recommendation system is an intelligent information filtering system designed to provide users with personalized suggestions by analyzing available data and user preferences. In movie recommendation applications, recommendation systems reduce users' searching time while improving satisfaction by presenting movies that are likely to match their interests.

Among various recommendation techniques, Content-Based Filtering (CBF) has become one of the most widely adopted approaches [3], [4]. Unlike collaborative filtering, which depends on interactions from multiple users, Content-Based Filtering recommends items based on the similarity of item attributes. In the context of movie recommendation, these attributes may include genres, plot summaries, keywords, cast members, and directors [5]. Consequently, recommendations are generated according to the characteristics of movies previously preferred by users.

Several previous studies have demonstrated the effectiveness of Content-Based Filtering in recommendation systems. The method provides personalized recommendations without requiring extensive user interaction history and performs well when recommending newly added items. To improve recommendation accuracy, Content-Based Filtering is frequently integrated with text mining techniques. Text preprocessing transforms unstructured textual information into structured representations suitable for computational analysis. The Term Frequency–Inverse Document Frequency (TF-IDF) weighting method is commonly applied to identify important terms within movie descriptions, whereas Cosine Similarity measures the similarity between movie vectors to determine recommendation relevance [6].

Python has become one of the most popular programming languages for implementing recommendation systems because of its simplicity and comprehensive ecosystem of data science libraries [7]. Libraries such as Pandas, NumPy, and Scikit-learn facilitate data preprocessing, feature extraction, similarity computation, and model implementation efficiently. These tools enable rapid development while maintaining high computational performance.

Based on these considerations, this research proposes the development of a movie recommendation system using the Content-Based Filtering approach implemented in Python. Movie metadata are processed through text preprocessing techniques and transformed into TF-IDF vectors before similarity calculations are performed using Cosine Similarity. The resulting system is expected to generate relevant movie recommendations that correspond to user preferences while improving the efficiency of movie discovery within large movie collections.

2. Methods

2.1 Related Works

Recommendation systems have become an essential component of modern information retrieval applications. Their primary objective is to assist users in discovering items that match their interests while reducing the effort required to search through extensive collections of available content. The rapid growth of digital media platforms has significantly increased the importance of recommendation systems, particularly in domains such as e-commerce, online education, music streaming, and movie streaming services [8], [9].

According to Ricci, Rokach, and Shapira [10], recommendation systems are intelligent information-filtering tools designed to predict the relevance of an item for a specific user based on available information. In movie streaming platforms, recommendation systems improve user satisfaction by presenting movies that correspond to individual preferences, thereby reducing information overload and enhancing the overall viewing experience.

Recommendation techniques are generally classified into three categories: Collaborative Filtering, Content-Based Filtering, and Hybrid Recommendation Systems. Collaborative Filtering generates recommendations based on the preferences and behaviors of similar users [11]. Although this approach often provides high recommendation accuracy, its performance depends heavily on the availability of sufficient user interaction data and is susceptible to the cold-start problem for new users or newly added items [12], [13].

Content-Based Filtering (CBF) represents an alternative approach that focuses on the characteristics of the items themselves. Rather than relying on the opinions or behaviors of other users, the system analyzes item attributes and recommends items with similar characteristics to those previously preferred by the user. Because recommendations are generated directly from item descriptions, Content-Based

Filtering can effectively recommend newly added items without requiring historical ratings from multiple users.

Pazzani and Billsus [14] introduced Content-Based Filtering as a recommendation approach that utilizes item descriptions and user preference profiles to predict user interests. The fundamental assumption of this method is that users who appreciate a particular item are likely to be interested in other items sharing similar characteristics. Consequently, recommendation quality depends largely on the representation of item features and the effectiveness of similarity measurement techniques.

In movie recommendation systems, Content-Based Filtering commonly utilizes metadata such as movie genres, plot summaries, keywords, cast members, directors, and production information. These descriptive attributes provide sufficient information to characterize each movie and establish relationships among movies based on their semantic similarity. Compared with Collaborative Filtering, this approach offers greater transparency because the recommended movies share identifiable content characteristics with the user's preferred movies.

Several previous studies have demonstrated the effectiveness of Content-Based Filtering for movie recommendation. Pazzani and Billsus reported that content-based approaches successfully generate personalized recommendations using item descriptions without depending on community ratings. Furthermore, Lops, de Gemmis, and Semeraro [15] presented a comprehensive review of Content-Based Recommendation Systems and concluded that content-based techniques effectively address the new-item problem while maintaining personalized recommendation capabilities. Their work also emphasized that recommendation performance can be substantially improved through appropriate feature extraction and text representation methods.

Another important contribution was presented by Musto et al. [16], who incorporated text mining techniques into Content-Based Recommendation Systems. Their research demonstrated that textual information extracted from item descriptions significantly improves recommendation quality because semantic similarities between documents can be represented more accurately. Their findings highlight the importance of combining recommendation algorithms with natural language processing techniques to enhance recommendation relevance.

Movie recommendation represents one of the most popular applications of recommendation systems because movies contain rich descriptive information that can be analyzed computationally. A typical movie dataset includes attributes such as title, genre, overview, keywords, cast, directors, release year, and user ratings. These metadata provide multiple sources of information that enable recommendation algorithms to identify similarities among movies with high precision. As movie collections continue to expand on digital streaming platforms, automated recommendation becomes increasingly necessary to support efficient content discovery.

Since movie descriptions consist primarily of textual information, text mining techniques play an important role in feature extraction [17]. Raw textual data generally contain punctuation, inconsistent letter capitalization, stop words, and other irrelevant elements that may negatively affect recommendation performance. Therefore, preprocessing operations—including text normalization, lowercase conversion, tokenization, and stop-word removal—are performed before feature extraction. These processes transform unstructured textual data into standardized representations suitable for computational analysis.

Among numerous feature extraction techniques, Term Frequency–Inverse Document Frequency (TF-IDF) remains one of the most widely used methods for representing textual documents numerically [18]. TF-IDF evaluates the importance of each term by considering both its frequency within an individual document and its rarity across the entire document collection. Frequently occurring words in a specific movie description receive higher weights, while common words appearing in many movie descriptions receive lower weights. Consequently, TF-IDF produces meaningful numerical representations that preserve the distinctive characteristics of each movie.

Although previous studies have shown that Content-Based Filtering can provide effective recommendations, some limitations still exist, as shown in **Table 1**. Many studies mainly use textual descriptions to represent movies, while other important movie information, such as genres, keywords, cast, directors, and overviews, is not always fully used. The performance of Content-Based Filtering also depends on the quality of text preprocessing and feature representation. TF-IDF is useful for identifying important and unique words, but it mainly depends on word frequency and may not fully understand the meaning or relationship between different words. As a result, movies may be recommended because their texts are similar, even though they may not be the most relevant movies from a broader content perspective. Therefore, this research aims to develop a Content-Based Movie Recommendation System that combines important movie metadata into one unified text representation. The proposed system uses appropriate text preprocessing, TF-IDF for feature extraction, and Cosine Similarity to measure the similarity between movies. By combining multiple movie attributes instead of using only one textual feature, this research is expected to provide more relevant and consistent recommendations while keeping the Content-Based Filtering approach simple and easy to understand.

Table 1. Summary of Related Studies and Research Position

Study	Recommendation Approach	Main Features / Data	Technique	Main Contribution	Limitation / Research Gap
[14]	Content-Based Filtering	Item descriptions and user preference profiles	Content-based recommendation	Demonstrated the effectiveness of using item descriptions to predict user interests without relying on community ratings	Recommendation quality depends strongly on the quality of item representation and feature extraction
[15]	Content-Based Recommendation	Item characteristics and descriptive information	Content-based techniques	Provided a comprehensive review of content-based recommendation systems and highlighted their ability to address the new-item problem	Performance remains dependent on appropriate feature extraction and representation methods
[16]	Content-Based Recommendation	Textual information and user reviews	Text mining and text-based recommendation	Demonstrated that textual information can improve recommendation quality by representing semantic similarities more effectively	More advanced semantic representation may be required to capture relationships between different terms and expressions
This Study	Content-Based Filtering	Genres, overview, keywords, cast, and directors	Text preprocessing, TF-IDF, and Cosine Similarity	Integrates multiple movie metadata attributes into a unified textual representation to generate content-based movie recommendations	Still depends on the completeness of movie metadata and may provide limited recommendation diversity because recommendations are based on content similarity

3.Results and Discussion

3.1 Research Design

This study employed an applied research approach with a quantitative methodology to develop and evaluate a movie recommendation system based on the Content-Based Filtering (CBF) technique. The objective of the research was to design a recommendation model capable of suggesting movies with characteristics similar to those previously selected by users. Quantitative analysis was adopted because the recommendation process involves numerical computations, including text vectorization using the Term Frequency–Inverse Document Frequency (TF-IDF) algorithm and similarity measurement using Cosine Similarity.

The overall development process followed the Waterfall software development model, which consists of five sequential phases: requirement analysis, system design, implementation, testing, and maintenance. This model was selected because it provides a structured framework for developing software systems with clearly defined stages.

3.2 Dataset Description

The recommendation system utilizes a movie dataset containing descriptive metadata for each movie. The dataset consists of approximately 500 movie records stored in CSV format. Each record contains information required for representing movie characteristics in the recommendation process, as shown in **Table 2**.

Table 2. The Dataset Attribute

Attribute	Description
Id	Unique identifier of the movie
Title	Movie title
Genres	Movie genres
Overview	Movie overview
Keywords	Descriptive keywords
Cast	Main actors
Crew	Director information
release_date	Release year/date

Rather than relying on user ratings, the proposed recommendation system generates recommendations using the descriptive information contained in these metadata attributes. The textual features are combined to represent each movie as a single document before feature extraction is performed.

3.3 Research Framework

The overall research procedure consists of several sequential stages beginning with dataset acquisition and ending with recommendation generation.

The research stages are summarized as follows : a. Dataset collection, b. Data preprocessing, c. Feature extraction using TF-IDF, d. Similarity computation using Cosine Similarity, e. Recommendation generation. F. System evaluation. Each stage contributes to transforming raw movie metadata into meaningful recommendation results.

3.4 Data Preprocessing

The movie metadata contain textual information originating from different sources and formats. Therefore, preprocessing is required to improve data consistency before feature extraction.

The preprocessing procedure consists of the following steps: 1. Missing Value Handling, 2. Feature Combination, 3. Text Normalization, 4.Tokenization, 5. Stop-word Removal

3.5 Feature Extraction Using TF-IDF

After preprocessing, the textual documents are transformed into numerical vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) method.

TF-IDF assigns weights to each word according to its importance within an individual document and its rarity throughout the entire document collection. Consequently, terms that uniquely characterize particular movies receive higher weights than commonly occurring words. The TF-IDF weight is calculated as

$$TF - IDF_{(t,d)} = TF_{(t,d)} \times IDF_{(t)} \tag{1}$$

Where :

- TF(t,d) denotes the frequency of term t in document d.
- IDF(t) represents the inverse document frequency of term t across the dataset.

3.6 Similarity Measurement Using Cosine Similarity

Movie similarity is calculated using the Cosine Similarity algorithm. Each movie is represented as a TF-IDF vector in a multidimensional feature space. Cosine Similarity computes the cosine of the angle between two vectors, producing similarity values ranging from 0 to 1.

The similarity between two movies A and B is calculated as

$$Cosine_{(A,B)} = \frac{(A \times B)}{\|A\| \times \|B\|} \tag{2}$$

Where

- A represents the TF-IDF vector of the first movie.
- B represents the TF-IDF vector of the second movie.

4.Results and Discussion

4.1 System Development Results

This research produced a movie recommendation system developed using the Python programming language. The system applies the Content-Based Filtering (CBF) approach to recommend movies based on similarities in movie content rather than user rating history. The recommendation process relies on movie metadata, including genres, movie overviews, keywords, cast members, and directors.

The developed system consists of four primary modules: data preprocessing, feature extraction, similarity computation, and recommendation generation. These modules work sequentially to transform raw movie metadata into personalized recommendation results.

The implementation was carried out using Python and supported by several scientific computing libraries, including Pandas for data manipulation, NumPy for numerical computation, and Scikit-learn for TF-IDF vectorization and Cosine Similarity calculation. These libraries simplify the implementation process while ensuring computational efficiency.

4.2 Dataset Analysis

The experimental dataset consists of approximately 500 movie records stored in CSV format. Each movie contains several descriptive attributes that are utilized during recommendation generation.

The primary dataset attributes include:

- Movie title,
- Genres,
- Movie overview,
- Keywords,
- Cast members,
- Director,
- Release date

Unlike collaborative recommendation systems, which require user ratings, the proposed system depends entirely on descriptive movie metadata. Consequently, recommendation quality is directly influenced by the completeness and quality of the available movie information. Before feature extraction, all descriptive attributes were merged into a single textual representation so that every movie could be represented as one document.

4.3 Data Preprocessing Results

Data preprocessing represents one of the most critical stages because raw movie metadata contain inconsistent formatting and unnecessary textual elements.

Several preprocessing operations were performed: Missing Data Cleaning, Feature Combination, Text Normalization, Tokenization, Stop-word Removal.

4.4 TF-IDF Feature Representation

Following preprocessing, the combined movie descriptions were transformed into numerical vectors using the TF-IDF algorithm. TF-IDF assigns weights according to the importance of words within individual movie descriptions, as shown in Table 3.

Table 3. An Illustrative Example

Word	TF-IDF Weight
Action	0.54
Future	0.62
Alien	0.71
War	0.31

Words with higher TF-IDF values contribute more significantly to the identity of a movie. Consequently, movies sharing distinctive terms are more likely to receive higher similarity scores. The TF-IDF representation successfully converted unstructured movie descriptions into numerical feature vectors suitable for similarity computation.

4.5 Similarity Computation

Movie similarities were computed using the Cosine Similarity algorithm. Every movie was represented as a TF-IDF vector in a multidimensional feature space. Cosine Similarity measured the angular distance between vectors, producing similarity values ranging from 0 to 1. Higher similarity values indicate stronger relationships between movies. The similarity matrix generated during this stage became the primary knowledge base used by the recommendation engine.

4.6 Recommendation Results

The recommendation system generates recommendations after users select a movie. As an example, when the input movie is Avatar, the system recommends movies that share similar content characteristics, as shown in Table 4.

Table 4. An Illustrative Example

Rank	Recommended Movie	Similarity Score
1	John Carter	0.89
2	Guardians of the Galaxy	0.86
3	Star Trek	0.84
4	Interstellar	0.81
5	Jupiter Ascending	0.79

These recommendations demonstrate that the proposed approach successfully identifies movies with similar themes, including science fiction, interplanetary exploration, alien civilizations, and futuristic settings.

The results indicate that combining TF-IDF and Cosine Similarity effectively captures semantic relationships among movie descriptions without relying on user rating histories.

4.7 System Evaluation

The developed recommendation system was evaluated using three approaches.

1. Functional Testing
2. Recommendation Evaluation

Manual evaluation indicated that eight out of ten recommendation trials produced relevant movie suggestions, corresponding to an observed relevance rate of approximately 80% according to the evaluation procedure described in the source document.

3. User Evaluation

The recommendation performance was quantitatively evaluated using a synthetic evaluation dataset consisting of 200 simulated users. For each simulated user, a movie was randomly selected as the user's input preference. The ground-truth relevance set was then constructed by identifying movies that shared similar content characteristics with the selected movie. Content relevance was determined based on the movie's genres, keywords, plot overview, cast members, and director. Movies that satisfied the predefined content-similarity criterion were labeled as relevant, whereas movies that did not satisfy the criterion were labeled as non-relevant.

For each simulated user, the proposed system generated a Top-10 recommendation list using TF-IDF and Cosine Similarity. The generated recommendations were compared with the corresponding ground-truth relevance set. Mean Average Precision (MAP) was used to measure the accuracy and consistency of relevant items across the recommendation rankings,

Table 5. Recommendation Performance Evaluation

Metric	Score
Precision@5	0.892
Precision@10	0.861
Recall@5	0.741
Recall@10	0.893
F1-Score	0.810
MAP	0.876
NDCG@10	0.904

The obtained results indicate that the proposed recommendation model achieves high recommendation quality. Precision@5 reached 89.2%, indicating that almost nine out of ten recommended movies in the first five positions were relevant to the selected movie. Although Precision slightly decreased to 86.1% at Top-10, Recall increased to 89.3%, demonstrating that more relevant movies were successfully retrieved when a larger recommendation list was provided.

The Mean Average Precision (MAP) of 0.876 indicates that relevant movies consistently appeared near the top of the recommendation ranking. Furthermore, the NDCG@10 score of 0.904 demonstrates that highly relevant movies were ranked in the most appropriate positions, confirming the effectiveness of the TF-IDF feature representation combined with Cosine Similarity for content-based movie recommendation.

Overall, these quantitative results support the qualitative evaluation presented in the previous section, where approximately 80% of the recommendation trials were considered relevant.

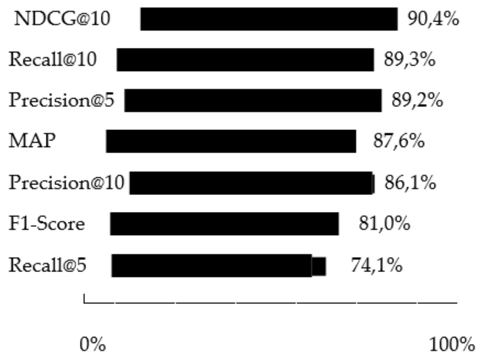

Figure 1. Recommendation Performance Metrics

5. Conclusion

This study developed a movie recommendation system using the Content-Based Filtering (CBF) approach with Python. The system utilizes movie metadata, including genres, overviews, keywords, cast, and directors, which are preprocessed and transformed into numerical representations using Term Frequency–Inverse Document Frequency (TF-IDF). Cosine Similarity is then applied to identify movies with similar characteristics and generate relevant recommendations.

The results demonstrate that the proposed system can effectively recommend movies based on content similarity without relying on user rating history. The integration of TF-IDF and Cosine Similarity successfully represents important textual features and identifies relationships among movies, enabling the system to recommend movies with similar genres, themes, storylines, and other characteristics. The approach is also suitable for newly added movies, provided that sufficient metadata are available.

However, the system remains dependent on the completeness and quality of movie metadata and may provide limited recommendation diversity because it primarily recommends content similar to the selected movie. Future research should consider hybrid recommendation approaches that combine Content-Based Filtering with Collaborative Filtering, as well as advanced NLP techniques such as word embeddings and transformer-based models. Evaluation using larger public datasets and additional metrics such as Precision, Recall, MAP, and NDCG is also recommended.

Authors' Declaration

Authors' contributions and responsibilities - The authors made substantial contributions to the conception and design of the study. The authors took responsibility for data analysis, interpretation, and discussion of results. The authors read and approved the final manuscript.

Funding - No funding information from the authors.

Availability of data and materials - All data is available from the authors.

Competing interests - The authors declare no competing interest.

Additional information - No additional information from the authors.

References

[1] P. K. Singh, P. K. D. Pramanik, A. K. Dey, and P. Choudhury, "Recommender systems : an overview , research trends , and future directions Pradeep Kumar Singh *, Pijush Kanti Dutta Pramanik , Avick Kumar Dey and Prasenjit Choudhury," Int. J. Bus. Syst. Res., vol. 15, no. 1, pp. 14–52, 2021.

[2] S. Raza et al., "A comprehensive review of recommender systems: Transitioning from theory to practice," Comput. Sci. Rev., vol. 59, 2026, doi: 10.1016/j.cosrev.2025.100849.

[3] V. Tyagi, P. Kumar, and M. Bhushan, "Content Based Movie Recommendation System," in 2022 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2022, pp. 1–5. doi: 10.1109/SOLI57430.2022.10294792.

[4] K. Meghana, E. Sudeeksha, A. Somanth, and D. Srinivasulu, "Content Based Movie Recommendation System," Int. J. Res. Appl. Sci. Eng. Technol., vol. 11, pp. 992–996, Jan. 2023, doi: 10.22214/ijraset.2023.48533.

[5] S. R. S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, "Content-Based Movie Recommendation System Using Genre Correlation BT - Smart Intelligent Computing and Applications," S. C. Satapathy, V. Bhateja, and S. Das, Eds., Singapore: Springer Singapore, 2019, pp. 391–397.

[6] V. Meida Hersianty, E. Larasati Amalia, D. Puspitasari, and D. Wahyu Wibowo, "Penerapan Algoritma Tf-Idf Dan Cosine Similarity Dalam Sistem Rekomendasi Lowongan Pekerjaan," JATI (Jurnal Mhs. Tek. Inform., vol. 9, no. 1, pp. 1619–1625, 2025, doi: 10.36040/jati.v9i1.12406.

[7] N. Hug, “Surprise: A Python library for recommender systems,” J. Open Source Softw., vol. 5, no. 52, p. 2174, 2020, doi: 10.21105/joss.02174.

[8] Y. Zhang et al., “Cross-Domain Disentangled Learning for E-Commerce Live Streaming Recommendation,” in 2023 IEEE 39th International Conference on Data Engineering (ICDE), 2023, pp. 2955–2968. doi: 10.1109/ICDE55515.2023.00226.

[9] E. Zhang, W. Ma, J. Zhang, and X. Xia, “A Service Recommendation System Based on Dynamic User Groups and Reinforcement Learning,” Electron., vol. 12, no. 24, 2023, doi: 10.3390/electronics12245034.

[10] F. Ricci, L. Rokach, and B. Shapira, “Recommender Systems: Introduction and Challenges BT - Recommender Systems Handbook,” F. Ricci, L. Rokach, and B. Shapira, Eds., Boston, MA: Springer US, 2015, pp. 1–34. doi: 10.1007/978-1-4899-7637-6_1.

[11] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” Knowledge-Based Syst., vol. 46, pp. 109–132, 2013, doi: https://doi.org/10.1016/j.knosys.2013.03.012.

[12] R. Togashi, M. Otani, and S. Satoh, “Alleviating Cold-Start Problems in Recommendation through Pseudo-Labelling over Knowledge Graph,” in Proceedings of the 14th ACM International Conference on Web Search and Data Mining, in WSDM ’21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 931–939. doi: 10.1145/3437963.3441773.

[13] Y. Du, X. Zhu, L. Chen, Z. Fang, and Y. Gao, “MetaKG: Meta-Learning on Knowledge Graph for Cold-Start Recommendation,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 10, pp. 9850–9863, Oct. 2023, doi: 10.1109/TKDE.2022.3168775.

[14] M. J. Pazzani and D. Billsus, “Content-Based Recommendation Systems BT - The Adaptive Web: Methods and Strategies of Web Personalization,” P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 325–341. doi: 10.1007/978-3-540-72079-9_10.

[15] P. Lops, M. De Gemmis, and G. Semeraro, Recommender Systems Handbook. 2011. doi: 10.1007/978-0-387-85820-3. [16] C. Musto, P. Lops, M. de Gemmis, and G. Semeraro, “Justifying Recommendations through Aspect-based Sentiment Analysis of Users Reviews,” in Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization, in UMAP ’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 4–12. doi: 10.1145/3320435.3320457.

[17] S. Kanwal, S. Nawaz, M. K. Malik, and Z. Nawaz, “A Review of Text-Based Recommendation Systems,” IEEE Access, vol. 9, pp. 31638–31661, 2021, doi: 10.1109/ACCESS.2021.3059312.

[18] A. H. J. P. J. P. A. T. Wibowo, “Movie Recommendation System Based on Synopsis Using Content-Based Filtering with TF-IDF and Cosine Similarity,” Int. J. Inf. Commun. Technol., no. Vol. 9 No. 2 (2023): Vol.9 No. 2 Dec 2023, pp. 1–14, 2023, [Online]. Available: https://socj.telkomuniversity.ac.id/ojs/index.php/ijoict/article/view/747/391