JURNAL TEKNOINFO
Volume 20, Number 2, July 2026, Page 170-179
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i2.1731

Feature Optimization for a Content-Based Music Recommendation System on Spotify

Viga Laksa Hardjanto[1*], Nisrina Hanifa Setiono[2]
[1]Computer Science Department, BINUS Online Learning, Bina Nusantara University, Indonesia
[2]Data Science, Telkom University Purwokerto, Indonesia
[1*]viga.hardjanto@binus.ac.id, [2]nisrinahanifas@telkomuniversity.ac.id

Abstract

The growth of music streaming services like Spotify has encouraged users to explore millions of songs, making effective recommendation systems essential. This study examines content-based music recommendation systems by comparing feature configurations and similarity metrics. The system uses 11 Spotify audio features, both with and without genre features encoded using one-hot encoding. Similarity is calculated using cosine similarity and Euclidean distance on normalized features (MinMaxScaler) to generate the top 10 recommendations. Recommendations are considered relevant if the recommended song is by the same artist as the searched song. Performance is measured using Precision@10 and Recall@10 on 100 samples. The *audio-only model* yields Precision@10 of 3.80–3.90% and Recall@10 of 6.36–6.45%. The *audio-plus-genre model* improves performance to Precision@10 of 9.00–9.10% and Recall@10 of 9.06–9.89%. These results show that genre features significantly improve the relevance of recommendations, while both similarity metrics perform similarly when the features have been well normalized. This study contributes by demonstrating that feature representation plays a more critical role than similarity metrics in content-based music recommendation.
Keywords: Music recommendation; content-based filtering; audio features; cosine similarity; Euclidean distance.

1. INTRODUCING

The rapid growth of digital music streaming services such as Spotify, Apple Music, and YouTube Music has led to an explosion of audio data, resulting in information overload that makes it increasingly difficult for users to discover music aligned with their preferences without automated assistance. Music recommendation systems have therefore become essential components in Music Information Retrieval (MIR) and modern streaming platforms [1], [2].

In general, recommendation approaches can be categorized into content-based filtering, collaborative filtering, and hybrid methods [3], [4]. Content-based filtering relies on intrinsic item features such as genre, tempo, and timbre to measure similarity between items, making it particularly suitable for domains rich in structured content [5], [6], [7]. This approach has been widely applied in various domains, including movie recommendation systems, where user profiles and content features are utilized to generate personalized recommendations based on item similarity [8]. In contrast, collaborative filtering leverages user interaction patterns but often suffers from data sparsity issues when user data is limited [9].

To address similarity measurement, techniques such as cosine similarity and Euclidean distance are widely used to compare music embeddings or acoustic feature vectors [10]. These techniques are particularly effective when applied to large-scale datasets such as Spotify, which provide structured audio attributes including danceability, energy, valence,

Viga Laksa Hardjanto: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 2, July 2026, Page 170-179
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i2.1731

and tempo [11], [12]. Prior studies have demonstrated that Spotify audio features are effective for tasks such as classification, clustering, and popularity prediction [13], [14], while their application in content-based recommendation systems remains an active area of research.

Recommendation systems have also been successfully applied beyond multimedia domains, such as in travel itinerary generation, where optimization techniques are used to generate efficient recommendations based on user preferences and constraints. For example, [15] proposed an enhanced simulated annealing approach combined with greedy algorithms to generate optimized travel itineraries, demonstrating how recommendation and optimization techniques can be adapted across domains .

Although advanced approaches such as deep learning have been proposed to enhance feature representation and recommendation performance [16] and hybrid models have shown improved accuracy by combining multiple techniques [3], [4], these methods often require large-scale user interaction data and higher computational complexity. Recent advancements in artificial intelligence, particularly in language models, have also demonstrated strong capabilities in handling content-based and creative tasks. However, these approaches still face challenges related to computational efficiency, scalability, and deployment constraints [17]. In contrast, simpler content-based approaches remain relevant, especially when focusing on intrinsic characteristics of items without requiring extensive user data.

Furthermore, previous studies highlight the importance of explainability and user experience in recommendation systems [2], [18], as well as the effectiveness of structured features in supporting various analytical tasks such as clustering and classification [19]. Based on these considerations, this study proposes a content-based music recommendation system using cosine similarity and Euclidean distance applied to a Spotify dataset from Kaggle, utilizing eleven audio features and genre information represented numerically. The system evaluates recommendation relevance using an artist-based approach, as artist information tends to be more consistent than genre labels across songs [20].

The contribution of this study is not limited to implementing a content-based music recommendation system. This study also provides an empirical analysis of how feature representation affects recommendation quality. By comparing the *audio-only model* and the *audio-plus-genre model* under cosine similarity and Euclidean distance, this study highlights the methodological importance of combining numerical acoustic features with categorical semantic features. The findings are expected to provide insight for recommendation system research, particularly in designing effective item representations for content-based music recommendation. This study aims to (1) implement a recommendation model based on audio features and genre, (2) analyze the impact of incorporating genre features on recommendation quality using Precision@10 and Recall@10, and (3) compare the performance of cosine similarity and Euclidean distance under different normalization schemes. The findings are expected to provide insights into effective feature configurations and similarity measures for building efficient and accurate content-based music recommendation systems.

2. METHODOLOGY

This study employs a content-based recommendation system approach by measuring similarity between songs using cosine similarity and Euclidean distance. This approach is selected because it does not rely on user interaction data, making it suitable for recommendation scenarios with limited user behavior data while maintaining computational simplicity. The overall research workflow consists of data collection, data preprocessing, feature engineering, model construction, similarity computation, and performance evaluation, as illustrated in Figure 1.

Viga Laksa Hardjanto: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 2, July 2026, Page 170-179
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i2.1731

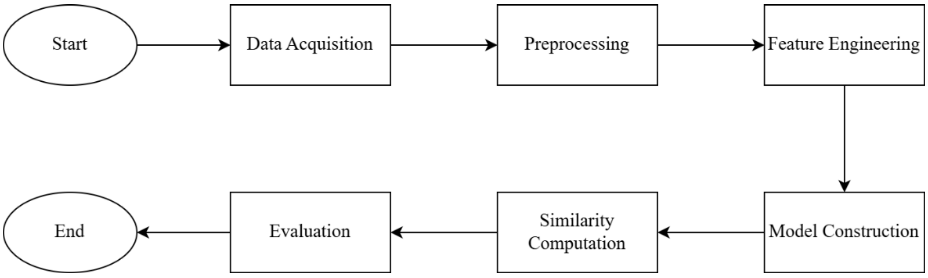

Figure 1. Research Design

2.1. Dataset and Data Preprocessing

The dataset used in this study is the *Spotify Tracks Dataset* obtained from Kaggle, which contains 114,000 song entries and 21 columns. The dataset includes metadata such as *track_id*, *track_name*, *artists*, *album_name*, *popularity*, *duration_ms*, *explicit*, and *track_genre*, as well as audio features extracted from the Spotify API. The audio features used in this study consist of *danceability*, *energy*, *key*, *loudness*, *mode*, *speechiness*, *acousticness*, *instrumentalness*, *liveness*, *valence*, and *tempo*. These features provide numerical representations of the acoustic, rhythmic, and emotional characteristics of each song, making them suitable for content-based recommendation. Figure 2 shows a sample of the data included in the dataset.

track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	energy	key	loudness
5SuOikwiRyPMVoIQDJUgSV	Gen Hoshino	Comedy	Comedy	73	230666	False	0.676	0.4610	1	-6.746
4qPNDBW1i3p13qLCt0Ki3A	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic	55	149610	False	0.420	0.1660	1	-17.235
1iJBSr7s7jYXzM8EGcbK5b	Ingrid Michaelson;ZAYN	To Begin Again	To Begin Again	57	210826	False	0.438	0.3590	0	-9.734
6lfxq3CG4xtTiEg7opyCyx	Kina Grannis	Crazy Rich Asians (Original Motion Picture Sou...	Can't Help Falling In Love	71	201933	False	0.266	0.0596	0	-18.515
5vjLSffimiIP26QG5WcN2K	Chord Overstreet	Hold On	Hold On	82	198853	False	0.618	0.4430	2	-9.681

Figure 2. Sample Data

Data preprocessing was conducted to ensure data quality and consistency. First, duplicate records were removed based on the *track_id* column to ensure that each Spotify track identifier appeared only once in the dataset. This process reduced the dataset from 114,000 records to 89,741 records. Second, missing values were examined across all columns. Missing values were found only in the *artists*, *album_name*, and *track_name* columns, with one missing value in each column. These incomplete records were removed because these columns are essential for identifying songs and generating recommendations. After this process, the final cleaned dataset consisted of 89,740 records.

Although duplicate records were removed based on *track_id*, the dataset may still contain songs with the same *track_name* and *artists* appearing in different rows, for example, due to different albums, versions, or genre assignments. Therefore, this study treats *track_id* as the primary identifier for preprocessing, while the combination of *track_name* and *artists* is considered when handling duplicate recommendation outputs. This step helps prevent the same song-artist pair from appearing repeatedly in the final recommendation list.

Furthermore, feature normalization was applied to numerical audio attributes to avoid scale bias in similarity computation. Two normalization techniques were used: *MinMaxScaler*, which scales features to the range [0, 1], and *StandardScaler*, which standardizes features to have zero mean and unit variance. These techniques were applied to examine the effect of feature scaling on cosine similarity and Euclidean distance.

Viga Laksa Hardjanto: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 2, July 2026, Page 170-179
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i2.1731

2.2. Feature Engineering and Data Representation

Feature engineering was performed to construct song vectors for similarity computation. Two modeling scenarios were used in this study. The first model used only 11 Spotify audio features, resulting in an 11-dimensional numerical feature vector. The second model added genre information from the *track_genre* column using one-hot encoding and combined it with the normalized audio features. As a result, the *audio-only* model consisted of 11 features, while the audio-plus-genre model consisted of 124 features, including 11 audio features and 113 genre dummy variables. Because one-hot encoding produces a sparse representation, most genre columns contain 0 values for each song, while only the column corresponding to the available *track_genre* value contains 1.

Each row in the dataset contains one value in the *track_genre* column. However, the same song identity, such as the same *track_name* and *artists*, may appear in different rows with different genre labels, albums, or track identifiers. In this study, the genre feature was encoded based on the available row-level *track_genre* value. Therefore, multi-genre aggregation into a multi-hot representation was not applied. Genre was used only as an additional input feature, while recommendation relevance was evaluated using an artist-based approach.

2.3. Index Mapping and System Preparation

An index mapping mechanism was implemented to retrieve song vectors efficiently based on user input. The mapping connects *track_name* with the corresponding dataset index, allowing the system to locate the selected song before similarity computation. Since several songs may have the same title, the combination of *track_name* and *artists* was also considered to reduce ambiguity.

In the recommendation output, duplicate results were removed based on the combination of *track_name* and *artists*. The system first retrieves more than 10 candidate songs, removes repeated song-artist pairs, and then returns the final Top-10 unique recommendations. This step helps prevent the same song from appearing repeatedly due to different albums, versions, or genre assignments.

2.4. Similarity Computation

2.4.1. Cosine Similarity

This study employs cosine similarity as one of the primary similarity metrics in the recommendation system. Cosine similarity measures the similarity between two songs by calculating the cosine of the angle between their feature vectors, as defined in Equation (1):

$$\text{cosine}(A, B) = \frac{A \cdot B}{\| A \| \| B \|} \tag{1}$$

This method emphasizes the similarity in feature patterns rather than their magnitudes. When two vectors have a similar orientation, the cosine similarity value approaches 1, indicating a high degree of similarity. Conversely, values close to 0 indicate dissimilar feature patterns. Due to its robustness to scale variation and effectiveness in high-dimensional data, cosine similarity is widely used in content-based recommendation systems. As illustrated in Figure 3, differences in vector length may represent variations in intensity (e.g., louder or faster songs), while similar directions indicate similar musical characteristics. Therefore, the system focuses on capturing similarities in musical patterns rather than absolute feature values.

Viga Laksa Hardjanto: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 2, July 2026, Page 170-179
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i2.1731

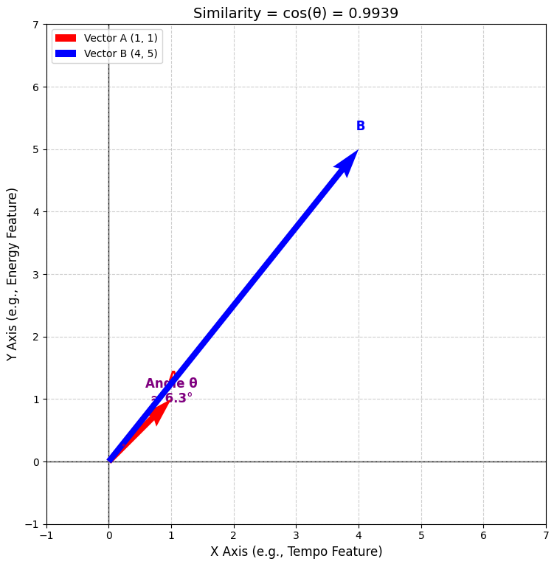

Figure 3. Example of a Cosine Similarity Visualization

2.4.2. Euclidean Distance

In addition to cosine similarity, Euclidean distance is used to measure the direct distance between two songs based on their feature values. This metric computes the straight-line distance between two vectors in a multidimensional feature space, as shown in Equation (2):

$$d(A,B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \tag{2}$$

A smaller Euclidean distance indicates greater similarity between songs. Unlike cosine similarity, this method is sensitive to feature scale, making normalization an essential preprocessing step. Euclidean distance defines similarity based on the absolute numerical difference of feature values, meaning that songs are considered similar only when their feature values are close in magnitude. As illustrated in Figure 4, this approach captures the shortest distance between two points in the feature space, providing a complementary perspective to cosine similarity. In this study, Euclidean distance is used as a comparative method to analyze how different similarity measures affect recommendation performance.

2.5. Recommendation Algorithm

The recommendation process follows a systematic pipeline. First, the user inputs a song title, and the system retrieves the corresponding feature vector using the predefined index mapping. Next, similarity scores between the selected song and all other songs in the dataset are computed using either cosine similarity or Euclidean distance. The songs are then ranked based on similarity (highest similarity for cosine, smallest distance for Euclidean). The system selects the top-N recommendations (e.g., Top-10), removes duplicate entries based on the combination of *track_name* and *artists*, and finally returns the recommendation list to the user.

2.6. Model Evaluation

The evaluation of the recommendation system is conducted using an *artist-based relevance* approach. A recommendation is considered relevant if the recommended song is performed by the same artist as the input song. This criterion is chosen because artist information is more consistent and reliable compared to genre labels, which may vary significantly and include highly specific subgenres in the Spotify dataset.

Viga Laksa Hardjanto: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 2, July 2026, Page 170-179
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i2.1731

The performance is measured using *Precision@10* and *Recall@10*. Precision@10 evaluates the proportion of relevant songs within the top 10 recommendations, while Recall@10 measures the proportion of all relevant songs successfully retrieved within those recommendations. The formulas are defined as follows:

$$\text{Precision@10} = \frac{\text{Number of songs with the same artist}}{10} \tag{3}$$

$$\text{Recall@10} = \frac{\text{Number of songs with the same artist}}{\text{Total songs by the artist} - 1} \tag{4}$$

The choice of $K = 10$ reflects typical user behavior in consuming top recommendations. The parameter K is used to define the number of top recommendations presented to users, ensuring that the system focuses on the most relevant items while reflecting typical user interaction behavior. Evaluation is performed on two scenarios (1000 and 100 sampled songs) to ensure the robustness and consistency of the results.

2.7. Implementation

The system is implemented in Python using libraries such as Pandas, NumPy, and Scikit-learn. All numerical audio features are normalized prior to similarity computation to ensure consistent scaling. Similarity calculations are efficiently performed using vectorized operations provided by Scikit-learn, specifically the *cosine_similarity* and *euclidean_distances* functions. All experiments are conducted in a local computing environment, and the recommendation results are analyzed using the defined evaluation metrics. While similarity computation utilizes either audio features alone or a combination of audio and genre features, system performance is consistently evaluated based on artist similarity to assess the model's ability to identify songs from the same source.

3. RESULT AND DISCUSSIONS

This section presents the experimental results to address the research objectives outlined in the Introduction, namely the evaluation of feature configurations, the impact of genre inclusion, and the comparison of similarity metrics and normalization schemes in a content-based music recommendation system.

3.1. Recommendation Results Based on a Single Query Song

An initial experiment was conducted using the song *"Knockin' On Heaven's Door"* by Bob Dylan (genre: country) as the query input. The results reveal a clear distinction between the *audio-only model* and the *audio-plus-genre model*. As shown in Table 1, the *audio-only* model produces recommendations spanning a wide range of genres, such as indie-pop, trip-hop, techno, and rock, which are not aligned with the country genre of the input song. This indicates that relying solely on audio features is insufficient to capture genre consistency or artist-related characteristics.

Table 1. Recommendations for Using the Audio Feature (Cosine Similarity)

No	Track Name	Artist	Genre
1	We fell in love in October	girl in red	indie-pop
2	Sweet Love	Phillip LaRue	acoustic
3	The Vagabond	Air	trip-hop
4	Vou Te Encontrar	Nando Reis; Péricles	brazil
5	Besos De Sal	Federico Aubele	trip-hop
6	Addio Addio - Original	Kollektiv Turmstrasse	minimal-techno
7	Shine	Collective Soul	grunge
8	Null Pointer	Monolake	idm
9	Through the Barricades	Spandau Ballet	disco
10	Amanda	Boston	hard-rock

In contrast, the model that combines audio features with genre information generates highly consistent recommendations, all belonging to the country genre, as presented in Table 2. This demonstrates that the inclusion of genre significantly improves the semantic relevance of the recommendations by guiding the system toward more contextually appropriate music.

Viga Laksa Hardjanto: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 2, July 2026, Page 170-179
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i2.1731

Table 2. Recommendations Using the Audio + Genre Feature (Cosine Similarity)

No	Track Name	Artist	Genre
1	Critic	Avery Anna	country
2	Ain't Over You Yet	Flatland Cavalry	country
3	Breathe	Faith Hill	country
4	Knowing You	Kenny Chesney	country
5	Driveway	Cody Johnson	country
6	Home	Blake Shelton	country
7	Austin	Blake Shelton	country
8	From This Moment On (On-Tour Version)	Shania Twain	country
9	Oklahoma Smokeshow	Zach Bryan	country
10	Holy Water	Brett Eldredge	country

3.2. Model Evaluation Using MinMaxScaler (1000 Samples)

The evaluation was conducted on 1,000 randomly selected songs using an *artist-based relevance* approach. The results are summarized in Table 3. The findings indicate that incorporating genre features leads to a substantial improvement in both *Precision@10* and *Recall@10*. Specifically, the performance of the *audio-only* model remains relatively low, with Precision@10 around 2.9% and Recall@10 around 4.2%. However, when genre is included, Precision@10 increases to approximately 7.3% and Recall@10 to around 6.6% for both cosine similarity and Euclidean distance. These results confirm that genre information enhances the system's ability to identify relevant songs from the same artist, thereby improving recommendation quality.

Table 3. Results of the Cosine and Euclidean Evaluations (MinMaxScaler, 1,000 Samples)

Model	Precision@10	Recall@10
Cosine (Audio Only)	2.91%	4.18%
Cosine (Audio + Genre)	7.31%	6.56%
Euclidean (Audio Only)	2.97%	4.21%
Euclidean (Audio + Genre)	7.35%	6.63%

3.3. Evaluation Using 100 Samples

To verify the consistency of the results, the evaluation was repeated using a smaller subset of 100 songs. The results, presented in Table 4, exhibit the same performance pattern as the larger experiment. The *audio-only* model achieves Precision@10 values of approximately 3.8–3.9% and Recall@10 around 6.3–6.4%, while the audio-plus-genre model significantly outperforms it, reaching Precision@10 values of around 9.0–9.1% and Recall@10 up to 9.9%. This consistency across different sample sizes indicates that the observed improvements are robust and not dependent on dataset scale.

Table 4. Results of the Cosine and Euclidean Evaluations (MinMaxScaler, 100 Samples)

Model	Precision@10	Recall@10
Cosine (Audio Only)	3.80%	6.36%
Cosine (Audio + Genre)	9.10%	9.89%
Euclidean (Audio Only)	3.90%	6.45%
Euclidean (Audio + Genre)	9.00%	9.06%

3.4. Evaluation Using StandardScaler

To analyze the effect of normalization, an additional experiment was conducted using *StandardScaler*. The results, shown in Table 5, indicate a slight improvement in performance, particularly for the *audio-only* model.

For instance, Precision@10 increases to approximately 4.2–4.3% and Recall@10 to around 6.6% in the *audio-only* scenario. However, for the audio-plus-genre model, the performance remains relatively similar to that achieved with MinMaxScaler, suggesting that normalization has a limited effect when richer feature representations are used.

Viga Laksa Hardjanto: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 2, July 2026, Page 170-179
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i2.1731

Table 5. Evaluation Results Using StandardScaler (100 Samples)

Model	Precision@10	Recall@10
Cosine (STD Audio Only)	4.20%	6.57%
Cosine (STD Audio + Genre)	8.40%	9.38%
Euclidean (STD Audio Only)	4.30%	6.60%
Euclidean (STD Audio + Genre)	7.70%	7.90%

3.5. Discussion

Overall, the experimental results demonstrate that the *audio-only* model produces less accurate recommendations due to its limited ability to capture higher-level musical context. Audio features alone may vary significantly even within songs from the same artist, leading to dispersed and less relevant recommendations across different genres. This limitation is reflected in the relatively low Precision@10 and Recall@10 values. In contrast, the inclusion of genre information significantly enhances system performance. Genre provides an additional semantic layer that helps group songs with similar musical characteristics, resulting in more coherent and relevant recommendations. This improvement is evident both qualitatively, as shown in the recommendation examples, and quantitatively, with performance metrics increasing by more than twofold.

Furthermore, the comparison between cosine similarity and Euclidean distance shows that both metrics yield comparable results across all experiments. This suggests that the choice of similarity metric has a relatively minor impact when features are properly normalized. The use of StandardScaler also slightly improves performance in the *audio-only* model, although it does not fundamentally alter the overall trend. These findings indicate that feature representation, particularly the inclusion of genre, is the dominant factor in improving the effectiveness of content-based music recommendation systems, while the choice of similarity metric and normalization technique plays a secondary role. This evaluation approach provides an objective proxy for relevance, although it may not fully capture subjective user preferences. From a methodological perspective, these findings indicate that feature representation plays a more dominant role than the choice of similarity metric. The comparable results between cosine similarity and Euclidean distance suggest that, when features are properly normalized, the semantic richness of the item representation becomes more important than the distance function itself. This implies that researchers and system developers should carefully evaluate how content features are selected, encoded, and combined before adopting more complex recommendation architectures.

3.6. Limitations

Although the proposed system shows that genre features can improve the performance of content-based music recommendation, this study has several limitations. First, the system does not use user interaction data such as listening history, ratings, skips, likes, or playlist behavior. Therefore, the recommendations are based only on item-to-item similarity and are not fully personalized. Second, the system depends on the quality and consistency of metadata, especially the *track_genre* label and Spotify audio features. Inaccurate or inconsistent metadata may affect the quality of the recommendation results. Third, although content-based filtering can reduce dependency on user interaction data, the system may still face item cold-start constraints when a song has incomplete metadata or missing audio features. Fourth, the evaluation uses artist-based relevance as an objective proxy, where a recommendation is considered relevant if it has the same artist as the query song. This approach provides a measurable evaluation strategy, but it may not fully represent real user preferences. Future studies should incorporate user behavior data, hybrid recommendation models, lyrical features, popularity indicators, and user-based evaluation to produce more personalized and realistic recommendations.

Viga Laksa Hardjanto: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 2, July 2026, Page 170-179
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i2.1731

4. CONCLUSION

This study shows that feature representation plays an important role in improving content-based music recommendation systems. The *audio-only* model produced relatively low relevance, which supports previous findings that a single type of information is often insufficient to represent item characteristics comprehensively [3]. In contrast, adding genre information improved both Precision@10 and Recall@10, confirming the importance of categorical and structural content features in recommendation and music analysis systems [11].

The results also show that cosine similarity and Euclidean distance produced comparable performance when features were properly normalized. This indicates that feature representation has a stronger influence than the choice of similarity metric [16]. Compared with collaborative filtering, hybrid approaches [4], [9], and deep learning-based methods [21], this study demonstrates that optimizing content-based features, especially by adding genre and applying proper normalization, can improve recommendation quality without using complex model architectures or user interaction data.

However, this study is limited by the absence of user interaction modeling and its dependence on metadata quality. Future research should incorporate user behavior, lyrical features, popularity indicators, and hybrid recommendation techniques to develop more adaptive, explainable, and personalized music recommendation systems.

5. ACKNOWLEDGMENT

The authors would like to express their gratitude to Universitas Bina Nusantara and Universitas Telkom for providing the facilities and support that enabled the successful completion of this research.

6. REFERENCES

[1] Y. Mu dan Y. Wu, "Multimodal Movie Recommendation System Using Deep Learning," *Mathematics*, vol. 11, no. 4, hal. 1–12, 2023, doi: 10.3390/math11040895.

[2] M. Martijn, C. Conati, dan K. Verbert, *"Knowing me, knowing you": personalized explanations for a music recommender system*, vol. 32, no. 1–2. Springer Netherlands, 2022. doi: 10.1007/s11257-021-09304-9.

[3] Y. Afoudi, M. Lazaar, dan M. Al Achhab, "Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network," *Simul. Model. Pract. Theory*, vol. 113, no. July, hal. 102375, 2021, doi: 10.1016/j.simpat.2021.102375.

[4] R. Widayanti, M. H. R. Chakim, C. Lukita, U. Rahardja, dan N. Lutfiani, "Improving Recommender Systems using Hybrid Techniques of Collaborative Filtering and Content-Based Filtering," *J. Appl. Data Sci.*, vol. 4, no. 3, hal. 289–302, 2023, doi: 10.47738/jads.v4i3.115.

[5] X. Wen, "Using deep learning approach and IoT architecture to build the intelligent music recommendation system," *Soft Comput.*, vol. 25, no. 4, hal. 3087–3096, 2021, doi: 10.1007/s00500-020-05364-y.

[6] P. N., K. K. R. Mangalore, R. B., P. N., dan S. R., "Content Based Movie Recommendation System," *Int. J. Res. Ind. Eng.*, 2020, doi: 10.1109/SOLI57430.2022.10294792.

[7] S. Sahu, R. Kumar, M. S. Pathan, J. Shafi, Y. Kumar, dan M. F. Ijaz, "Movie Popularity and Target Audience Prediction Using the Content-Based Recommender System," *IEEE Access*, vol. 10, hal. 42030–42046, 2022, doi: 10.1109/ACCESS.2022.3168161.

[8] S. Sridhar, D. Dhanasekaran, dan G. C. P. Latha, "Content-Based Movie Recommendation System Using MBO with DBN," *Intell. Autom. Soft Comput.*, vol. 35, no. 3, hal. 3241–3257, 2022, doi: 10.32604/iasc.2023.030361.

Viga Laksa Hardjanto: *Corresponding Author

[9] I. Yoshua dan H. Bunyamin, "Pengimplementasian Sistem Rekomendasi Musik dengan Metode Collaborative Filtering," *J. Strateg.*, vol. 3, no. 1, hal. 1–16, 2021.
[10] C. Liu dan Z. Chao, "Supervised learning and unsupervised learning on music data with different genres," in *Proceedings - 2021 IEEE 7th International Conference on Big Data Intelligence and Computing, DataCom 2021*, IEEE, 2021, hal. 7–12. doi: 10.1109/DataCom53700.2021.00008.
[11] S. Shinde, S. Kulkarni, P. Kulkarni, dan A. Bhatt, "Machine learning Based Clustering Using Spotify Audio Features," in *2023 1st DMIHER International Conference on Artificial Intelligence in Education and Industry 4.0, IDICAIEI 2023*, IEEE, 2023, hal. 1–5. doi: 10.1109/IDICAIEI58380.2023.10406332.
[12] J. S. Gulmatico, J. A. B. Susa, M. A. F. Malbog, A. Acoba, M. D. Nipas, dan J. N. Mindoro, "SpotiPred: A Machine Learning Approach Prediction of Spotify Music Popularity by Audio Features," in *ICPC2T 2022 - 2nd International Conference on Power, Control and Computing Technologies, Proceedings*, IEEE, 2022, hal. 1–5. doi: 10.1109/ICPC2T53885.2022.9776765.
[13] Z. Dai dan X. Huang, "Electronic Dance Music Classification Based on Machine Learning Methods," in *Proceedings - 2022 International Conference on Electronics and Devices, Computational Science, ICEDCS 2022*, IEEE, 2022, hal. 351–354. doi: 10.1109/ICEDCS57360.2022.00084.
[14] N. B. Bahadure, M. Vishwakarma, M. P. Kurhekar, J. C. Patni, dan P. D. Patil, "Machine Learning-Based Music Classification and Recommendation System from Spotify," *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 17, hal. 143–154, 2025, doi: 10.70917/2025011.
[15] N. Hanafiah, I. Wijaya, S. Xavier, C. G. Grolyus, D. Adrianto, dan M. Shodiq, "Itinerary Recommendation Generation using Enhanced Simulated Annealing Algorithm," in *Procedia Computer Science*, Elsevier B.V., 2019, hal. 605–612. doi: 10.1016/j.procs.2019.09.020.
[16] M. Sheikh Fathollahi dan F. Razzazi, "Music similarity measurement and recommendation system using convolutional neural networks," *Int. J. Multimed. Inf. Retr.*, vol. 10, no. 1, hal. 43–53, 2021, doi: 10.1007/s13735-021-00206-5.
[17] A. Ridwansyah dan I. P. Saputro, "Implementation of Small Language Model for On-Device Creative Lyrics Generation Using LoRA and 8-bit Quantization Methods," in *2025 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, IEEE, 2025, hal. 551–556. doi: 10.1109/ICIMCIS68501.2025.11327303.
[18] D. Afchar, A. B. Melchiorre, M. Schedl, R. Hennequin, E. V. Epure, dan M. Moussallam, "Explainability in music recommender systems," *AI Mag.*, vol. 43, no. 2, hal. 190–208, 2022, doi: 10.1002/aaai.12056.
[19] T. Jain, N. S. Kumar, V. Tiwari, dan P. K. Tiwari, "Music Recommendation System Using Machine Learning," in *2023 2nd International Conference on Futuristic Technologies, INCOFT 2023*, IEEE, 2023, hal. 1–6. doi: 10.1109/INCOFT60753.2023.10425522.
[20] A. D. Hartono dan J. Sutopo, "Implementasi Algoritma Upper Confidence Bound Untuk Sistem Rekomendasi Musik," *J. Algoritm.*, vol. 21, no. 2, hal. 82–90, 2024, doi: 10.33364/algoritma/v.21-2.2095.
[21] A. Poulose, C. S. Reddy, S. Dash, dan B. J. R. Sahu, "Music recommender system via deep learning," *J. Inf. Optim. Sci.*, vol. 43, no. 5, hal. 1081–1088, 2022, doi: 10.1080/02522667.2022.2094546.

Viga Laksa Hardjanto: *Corresponding Author