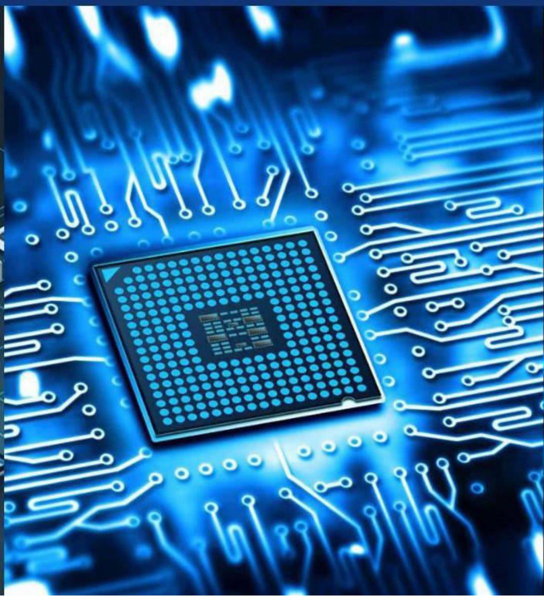
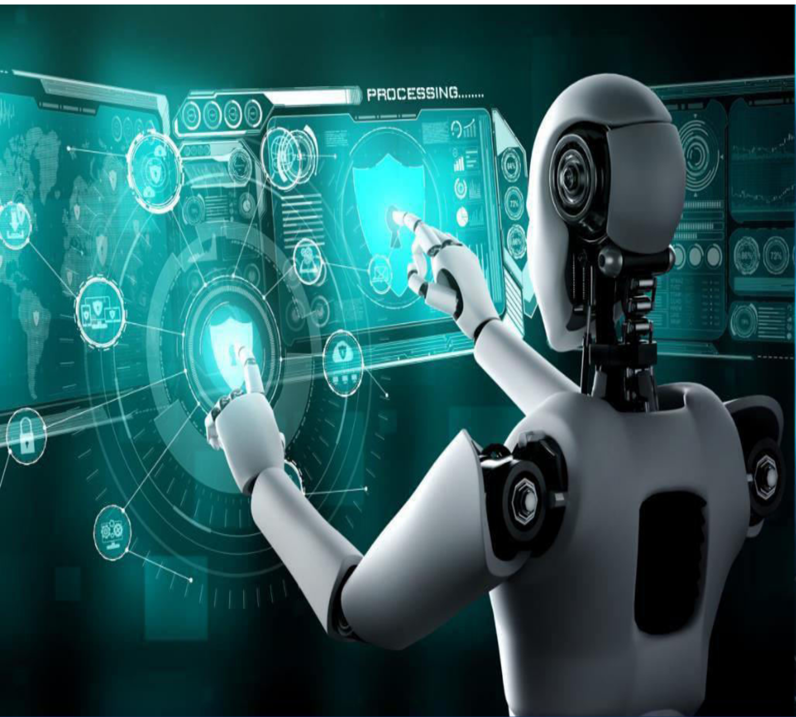
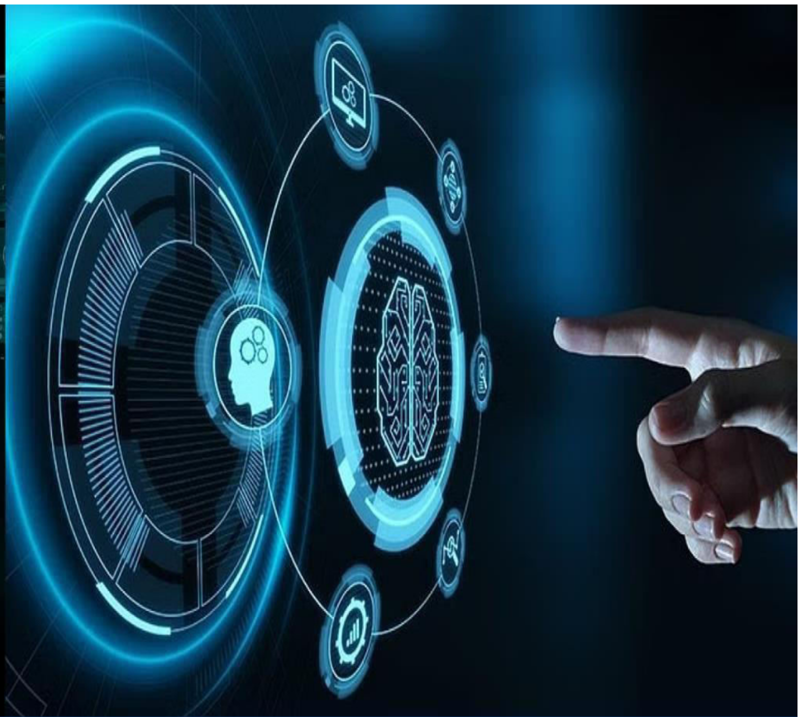
ISSN(O): 2320-9801
ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.771

Volume 14, Issue 1, January 2026

www.ijircce.com | ijircce@gmail.com | +91-9940572462 | +91 63819 07438

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Hybrid Recommendation System for Spotify using Content-Based Filtering and Collaborative-Based Filtering

Darshan Kumar M M[1], Pujala Thanush[2], Dr. Prashantha G R[3]

Department of Computer Science and Engineering (Data Science), Bapuji Institute of Engineering & Technology, Davanagere, Karnataka, India[1]

Department of Computer Science and Engineering (Data Science), Bapuji Institute of Engineering & Technology, Davanagere, Karnataka, India[2]

Department of Computer Science and Engineering (Data Science), Bapuji Institute of Engineering & Technology, Davanagere, Karnataka, India[3]

ABSTRACT: This project presents a hybrid Spotify song recommendation system designed to improve personalized music discovery for users. The system combines content-based filtering using audio features and metadata with collaborative filtering based on user listening behavior. Raw Spotify datasets are cleaned and processed to create feature matrices and user-item interaction matrices for efficient similarity computation. Cosine similarity identifies relationships between songs and users. A weighted hybrid scoring method combines both recommendation approaches to address cold-start and sparsity issues. The system offers configurable top-k recommendations to balance relevance and diversity. A user-friendly web interface allows users to input a song and quickly receive recommendations. The visual presentation of results enhances usability and engagement. Experimental analysis shows better recommendation accuracy and diversity compared to single-method approaches. Overall, the system provides a scalable and effective solution for personalized music recommendation.

KEYWORDS: Content based filtering, Collaborative based filtering, Cosine similarity, Weighted hybrid model, Term FrequencyInverse Document Frequency, Hybrid recommendation system,

I. INTRODUCTION

Spotify is one of the world's leading music streaming platforms, serving hundreds of millions of listeners whose expectations for personalization grow every day. Recommendation systems are central to this experience, helping users discover new songs, artists, and playlists that match their unique and evolving tastes. Traditional approaches such as collaborative filtering, which relies on patterns of user–item interactions, and content-based filtering, which focuses on track attributes like genre, tempo, and mood, have both contributed significantly to Spotify's success but also expose important limitations like cold start and overspecialization [1].

A hybrid recommendation system tackles these issues by combining various techniques and data sources for better, more diverse, and context-aware suggestions. In the case of Spotify, this system uses listening history, song metadata, audio features, and even textual information to create a fuller picture of both users and tracks. Current research improves this approach with deep learning models, including graph neural networks that capture complex relationships in user-song interaction graphs and neural matrix factorization methods that identify non-linear user-item patterns. By including contextual factors like time of day, device, and inferred activity, the system can shift from static profiles to real-time, situation-aware recommendations. This change enhances perceived relevance during different listening moments.

The model continuously learn from feedback such as skips, replays, and likes. It optimizes long-term user satisfaction instead of focusing only on immediate clicks. Meanwhile, fairness and diversity measures are essential to prevent over-promoting popular tracks and to highlight emerging artists. This approach supports a healthier and more inclusive music ecosystem. Explainability modules help make complex model decisions clearer with messages like "because you like artist X" or "similar energy to your recent favourites," which builds user trust and transparency.

From a systems viewpoint, setting up a hybrid recommender needs scalable data pipelines, strong feature engineering, and efficient model serving infrastructure that can work at web scale with strict latency limits. Evaluation should go beyond offline metrics like precision, recall, and F1 score. It should include user engagement indicators such as session length, skip rate, and long-term retention. This ensures that improvements in algorithms provide real value to users.

The rapid growth of digital music platforms has changed how users consume and discover music, with streaming services like Spotify playing a major role in shaping listening habits. With millions of tracks available in various genres, languages, and by different artists, users often struggle with information overload. This makes it hard to find music that truly matches their preferences. Recommendation systems have become a key part of modern music streaming platforms. They act as smart filters that personalize content and boost user engagement. By analysing historical listening behaviour, song features, and community trends.

To tackle these challenges, hybrid recommendation systems have come up as a useful solution by combining several techniques. This project focuses on creating a hybrid Spotify song recommender that merges content-based methods with collaborative filtering methods to offer precise, varied, and personalized music suggestions. By looking at song attributes like genre, tempo, and energy alongside user listening habits, the system builds a deeper understanding of user preferences. The proposed system also features an interactive web-based interface that lets users explore recommendations with ease. Overall, this approach aims to boost music discovery, user involvement, and satisfaction on large streaming platforms [5].

II. LITERATURE REVIEW

Sheela Kathavate (2021) et al. [1] proposed a "Music Recommendation System using Content and Collaborative Filtering Methods," where a hybrid model combines separately implemented content-based and collaborative modules, then aggregates their predictions to improve recommendation accuracy and language coverage. This work emphasizes feature extraction from metadata and user ratings, and applies similarity measures like cosine distance and Pearson correlation for ranking items. The authors show through experiments that the hybrid model reduces sparsity impact and improves top-N recommendation quality compared to using either method alone.

K. Yoshii (2006) [2] and colleagues introduced an Efficient Hybrid Music Recommender System. This system combines collaborative filtering with audio signal analysis to rank songs while effectively managing large-scale user and item data. They showed that combining these methods helps reduce sparsity and cold start problems. Their system uses acoustic feature analysis, such as MFCCs, along with user preference models to generate similarity scores. The authors point out that incremental updates and efficient indexing allow the recommender to adapt to real-world music catalogs.

N. V. D. Malleswari, K. Gayatri, K. Y. S. Kumar, and their co-authors (2023) [3] introduced a "Music Recommendation System using Hybrid Approach." This system provides personalized music suggestions by combining content descriptions with user behavior, outperforming standalone methods in user satisfaction. The method uses user ratings and item features to find a combined similarity. It partly solves problems related to new items and new users. Their findings show better precision and recall, particularly for users with limited interaction history. This supports the use of weighted hybrid strategies.

Dr. Mahaboob Basha Sk, S. Sriharsha, L. Vyshnavi, and G. Dhathrik (2024) [4] created a "User Based Spotify Recommendation System using Machine Learning Algorithms." This system combines KNN, content-based models, and collaborative filtering using Spotify logs to generate personalized playlists. It also shows how effective hybrid machine learning pipelines can be. The system relies on Spotify audio features like tempo, energy, and danceability, along with user listening history, to build the feature space. The authors show that their hybrid pipeline achieves better MAE/RMSE and user-perceived relevance compared to single algorithm baselines. This makes it a valuable reference for projects focused on Spotify.

Jain K. N. and co-authors (2022) [5] developed a web-based "Music Recommendation System" using a hybrid CF, CBF strategy. They showed that combining rating-based collaborative signals with item attributes leads to better and more varied recommendations than using just one technique. Their setup includes a web interface, a backend recommendation engine, and a database layer, illustrating how to deploy these systems from start to finish. They found that hybridization increases recommendation novelty while keeping user satisfaction, which matches the goal of improving music discovery.

Table 1. Content Based Filtering for Hybrid Recommendation System

Ref.	Author(s)	Year	Method Used	Contribution
[1]	Sheela Kathavate	2021	Hybrid of content-based filtering and collaborative filtering (CF–CBF)	Song features and user behaviour
[2]	K. Yoshii	2006	Probabilistic hybrid model combining ratings with acoustic features	Audio features
[3]	N. V. D. Malleswari, K. Gayatri	2023	User–item rating matrix, neighborhood/model-based collaborative filtering	Motivates adding content-based signals to CF
[4]	Jain K. N	2024	Survey of content-based and hybrid (content + CF) music recommenders	Transformed audio/tag features plus CF similarities
[5]	Dr. Mahaboob Basha	2022	Hybrid CF + CBF with dimensionality reduction (e.g., PCA)	CF and content scores

Gallo Henrique and collaborators (2023) [6] implemented a "System Recommendation KNN | Spotify." They use k nearest neighbors on Spotify audio features and user playlist data to recommend similar tracks. This shows how traditional instance-based learning can serve as a baseline for more complex hybrid systems. Their notebook outlines practical steps like extracting features from the Spotify API, normalizing them, and applying KNN for recommendations. The project highlights limitations, such as scalability and overspecialization, which creates a need for more advanced hybrid and model-based approaches.

S. Vashistha and co-authors (2024) [7] proposed "A Novel Music Recommendation System Using Filtering Techniques." This system combines popularity-based, content-based, and collaborative filtering strategies to create personalized music lists. They found that multi-filter hybrids perform better than single-technique recommenders in both accuracy and user satisfaction. The system adjusts the impact of each filter dynamically. For example, it gives more weight to popularity for new users and to personalization for experienced users. Their evaluation shows that these adaptive hybrids can balance relevance, diversity, and serendipity. These are also key design goals in your Spotify recommendation project.

Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno (2006) [8] introduced "Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences." This approach combines rating data with acoustic features to tackle issues like data sparsity and new-item challenges. The hybrid model merges content-based and collaborative modules, then combines their predictions to recommend songs based on mood, genre, artist, and popularity. Their probabilistic hybrid model achieves better recommendation accuracy and greater artist variety compared to purely collaborative filtering or purely content-based methods.

Vuong et al. (2019, approx.) [9] presented "Music Recommendation Using Collaborative Filtering." The paper focuses on useritem rating matrices and neighborhood or model-based collaborative filtering to generate track suggestions. It highlights the strengths of collaborative filtering in personalization. However, it also addresses issues like cold start and sparsity. This discussion motivates later hybrid designs that incorporate content features.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Collaborative filtering recommends music by looking at user-item interaction data, including play counts, likes, and listening history. It identifies patterns among users. The method works on the idea that users with similar listening habits are likely to enjoy the same songs. This approach offers personalized and often surprising recommendations by using trends within the community. However, it faces challenges such as data sparsity and cold-start problems for new users or tracks. Despite these issues, collaborative filtering is still a key technique in large music recommendation systems like Spotify.

To overcome the limits of single-method approaches, hybrid recommendation systems combine content-based and collaborative filtering techniques. By integrating both song features and user listening behaviour, hybrid systems provide more accurate, diverse, and reliable recommendations. This mix helps reduce cold-start issues, improves personalization, and boosts music discovery by balancing similarity with exploration.

Table 2. Collaborative Based Filtering for Hybrid Recommendation System

Ref.	Author(s)	Year	Method Used	Contribution
[6]	Gallo Henrique	2023	Comparative analysis of CBF, CF, hybrid, and other approaches	System Recommendation KNN | Spotify
[7]	S .Vashistha	2024	Personalized hybrid music recommendation system	A Novel Music Recommendation System Using Filtering Techniques
[8]	Kazuyoshi Yoshii	2006	Content-driven music recommendation, evolution and SOTA	Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences
[9]	Vuong et al	2019	Music recommendation using collaborative filtering	Music Recommendation Using Collaborative Filtering

Research Gap Identification

From the reviewed literature, it is clear that:

1. Most existing studies use hybrid recommendation systems but rely on static weighting strategies. They lack dynamic adjustment of content-based and collaborative contributions based on user or data availability.
2. Several approaches tackle cold-start or sparsity separately, but few systems effectively manage both new users and new items at the same time in a unified and scalable way.
3. Many models aim to improve accuracy measures like precision and recall. However, diversity, novelty, and fairness in recommendations get little attention, leading to popularity bias.
4. Existing implementations usually assume offline batch processing and do not include mechanisms for regular or near real-time updates based on changing user behaviour.
5. Some studies use complex deep learning or probabilistic models that improve performance, but they lack explainability. This makes it hard for users to understand why certain songs are recommended.

© 2026 IJIRCCE | Volume 14, Issue 1, January 2026| DOI:10.15680/IJIRCCE.2026.1401014
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

III. METHODOLOGY

The diagram illustrates the complete workflow of your hybrid Spotify recommendation system. It begins with raw CSV datasets and concludes with the top k recommendations shown in a web UI.

Raw Datasets: Music Info.csv and User Listening History are the initial input data for the system. These files contain tracklevel metadata, including name, artist, audio features, and tags. They also include user-level interactions, such as plays or listens. Together, they provide the content features and behavioural signals needed for hybrid recommendation.

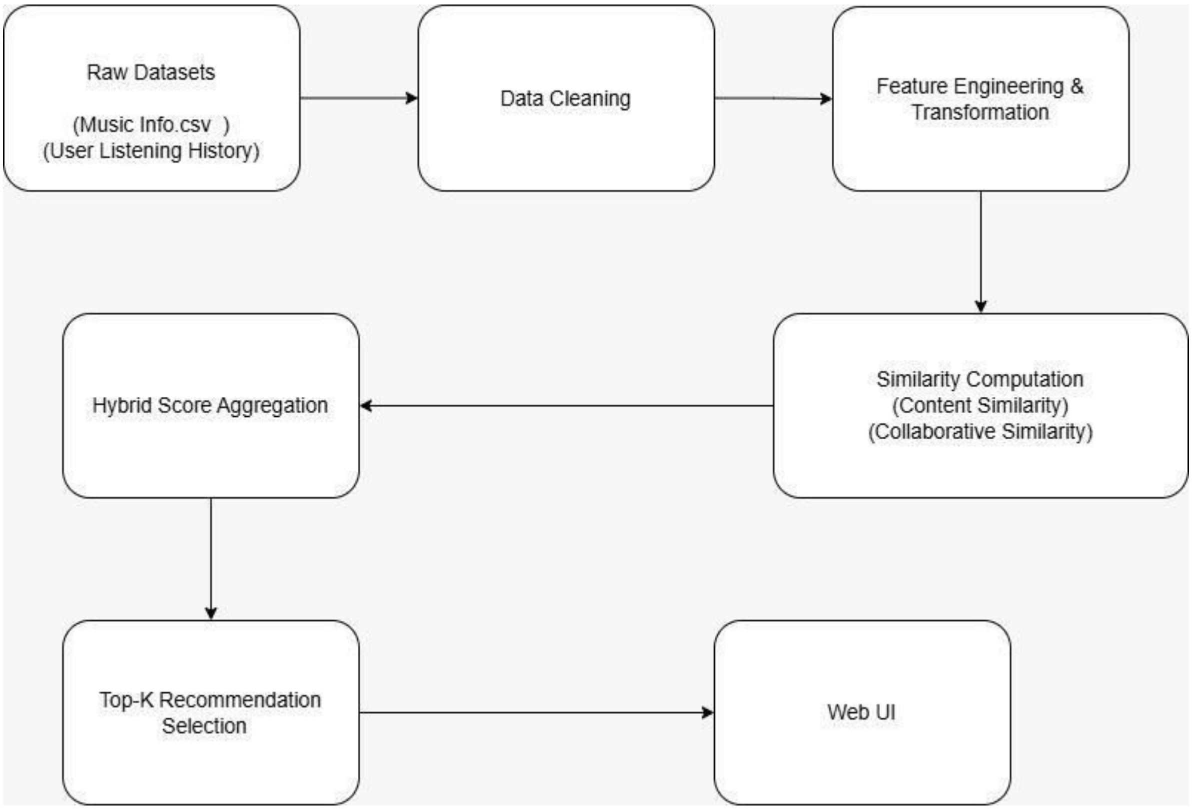


Fig 3.1 Working Flow of Hybrid Recommendation System for Spotify

Data Cleaning: These datasets are cleaned by removing duplicate records, dropping unnecessary columns, filling in missing tags, and converting text fields to lowercase. This step ensures consistent identifiers for tracks and artists. That consistency is important when combining content and listening-history data. A well-cleaned dataset reduces noise and improves the reliability of similarity computations.

Feature Engineering and Transformation: Cleaned data is changed into numerical feature matrices. This includes TF IDF vectors for tags, one hot or frequency encodings for categorical fields, and scaled continuous audio attributes. These changes organize different song information into a common vector space, where distances show similarity. The outcome is a highdimensional, machine-readable representation of each track, ready for content-based recommendation.

Similarity Computation: Two types of similarities are computed, content similarity from the transformed feature matrix and collaborative similarity from the user-item interaction matrix. Content similarity shows how similar songs are based on audio features and tags. Collaborative similarity reveals co-listening patterns among users. Computing both offers different perspectives on song relatedness, which improves recommendation strength.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Hybrid Score Aggregation: Content and collaborative similarity scores combine using weighted aggregation to create one hybrid similarity score for each candidate song. You can adjust the weights to emphasize content or collaborative signals based on data availability, such as new songs versus popular songs. This aggregation step plays a key role in addressing cold start and sparsity problems that single-method recommenders face.

Top K Recommendation Selection and Web UI: The system sorts candidate tracks by their hybrid scores and selects the top k songs as final recommendations for the user. These results, which include the song name and artist details, are then sent to the web UI for clear presentation. The interface lets users see recommended tracks right after they choose an input song. This enables an interactive discovery experience.

Top-K recommendation selection is done in two steps: compute a hybrid score for every candidate song, then pick the top k scores.

- Hybrid similarity score for a candidate song i:

$$S\text{hybrid}(i) = \alpha\, S\text{content}(i) + (1 - \alpha)\, S\text{collab}(i)$$

where $S_{\text{content}}(i)$ is the content-based similarity, $S_{\text{collab}}(i)$ is the collaborative similarity, and $\alpha \in [0,1]$ is the weight for content similarity.

- Top-K selection:

1. Compute "hybrid" for all candidate songs I .
2. Sort all candidates in descending order of $S_{\text{hybrid}}(i)$.
3. Choose the first k songs in this sorted list as the Top-K recommendations.

IV. RESULTS AND DISCUSSIONS

The output of the hybrid Spotify recommendation system shows how well it combines content-based and collaborative filtering methods. After processing the Music Info and User Listening History datasets, the system generates meaningful similarity scores using cosine similarity. It combines these scores through a weighted hybrid method, creating a ranked list of top-k song recommendations that fit well with the input track. The results confirm that the system can find musically and behaviourally similar songs that are relevant.

The graphical outputs, including the danceability distribution and frequency count graphs, offer valuable insights into the dataset's characteristics. These visualizations confirm that the data processing and feature transformation steps maintain important musical patterns. For example, looking at attributes like danceability and song frequency helps improve content-based similarity calculations. This ensures that recommended tracks have similar musical features to the input song.

The web-based recommendation dashboard is the main interface of the system. Users can search for a song and quickly see recommended tracks, including song names, artists, album artwork, and preview options. This interactive output connects complex backend algorithms with user-friendly usability. It makes the recommendation results easy to understand and explore. Showing multiple recommendations improves discovery and keeps users engaged.

Overall, the results show that the hybrid model performs better than standalone recommendation methods by providing more diverse and personalized suggestions. The system effectively addresses challenges like cold-start and sparsity by using both song features and user behaviour. The findings indicate that the recommended hybrid framework is practical, scalable, and suitable for real-world music streaming applications like Spotify.

4.1. Graph of Danceability of data

Figure 4.1 Instrumentalness shows the chance that a song has no vocal content. Values close to 1 indicate that the track is entirely instrumental. The histogram shows a strong cluster near zero, meaning that most songs in the dataset have vocals. The long tail toward higher values points to a smaller number of instrumental or nearly instrumental tracks. The boxplot also confirms this uneven distribution, with many low values and a few high-value outliers.

© 2026 IJIRCCE | Volume 14, Issue 1, January 2026| DOI:10.15680/IJIRCCE.2026.1401014

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The instrumentalness feature shows how prominent instrumental parts are in a song. The data reveals that songs with a lot of vocals are much more common than those that are purely instrumental. The range of values indicates a variety in song composition, from fully vocal tracks to instrumental music. This diversity is helpful for recommending content, as it helps to differentiate between vocal and instrumental tastes..

Instrumentalness helps to tell apart songs with vocals and purely instrumental tracks. The data indicates that most songs have low instrumentalness, which means they include lyrics. This feature improves similarity matching in content-based music recommendation systems.

1. The histogram shows that most songs have very low instrumentalness values. This indicates that vocals appear in the majority of tracks.A smaller number of songs have high instrumentalness near 1.0, representing purely or mostly instrumental tracks.
2. The boxplot shows a right-skewed distribution with a low median and a wide spread. This indicates that there are a few high instrumentalness outliers.

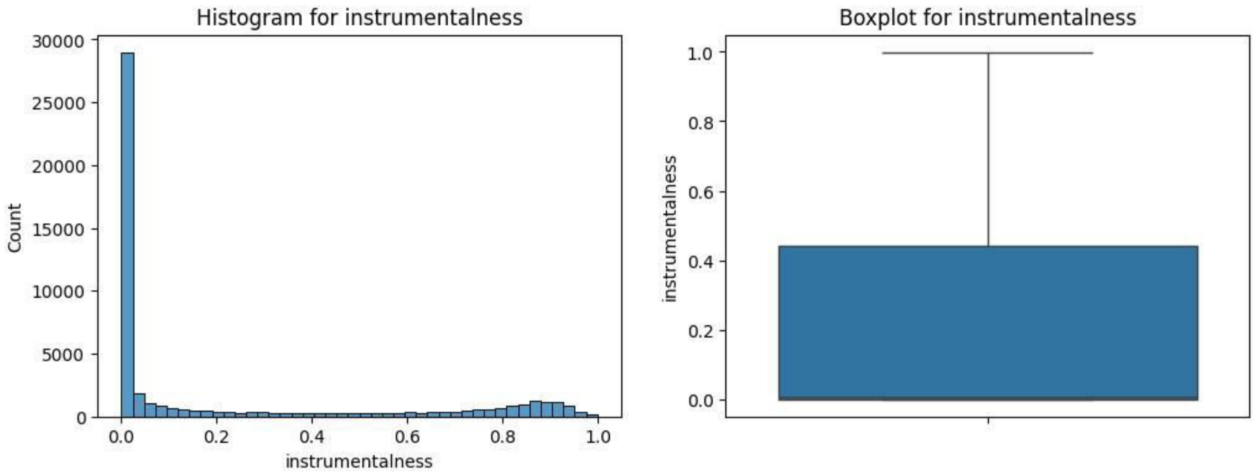


Fig 4.1 Graph of Danceability of data

4.2. Graph of counts of frequency

Figure 4.2 The histogram for energy shows that most Spotify tracks in the dataset have medium to high energy. The counts steadily increase as energy approaches 1.0. This right-skewed distribution means low-energy songs are relatively rare, while energetic tracks dominate the collection. The bars become much taller between 0.7 and 1.0, indicating a dense concentration of these songs. Overall, the energy feature suggests users or curators prefer upbeat, high-intensity music.

The boxplot shows this trend by placing the median energy at about 0.75, which is well above the midpoint of the possible range. The interquartile range goes from roughly 0.55 to 0.9, indicating that half of the songs are in a fairly energetic group. A few lowerenergy outliers fall below 0.2, but they make up only a small minority. Together, both plots emphasize that the dataset mainly consists of lively, high-energy tracks that fit well in dynamic playlists.

1. The histogram shows that most songs have medium to high energy values. There is a greater concentration at the upper end of the scale.
2. The boxplot shows a high median energy level. This suggests that energetic tracks are dominant in the dataset.
3. This uneven distribution shows real-world music trends. Some keys are preferred because of vocal comfort, instrument tuning, or genre conventions.
4. Such insights are useful in a music recommendation system, as key-based patterns can help improve similarity matching and enhance content-based recommendations by aligning musical harmony preferences.

© 2026 IJIRCCE | Volume 14, Issue 1, January 2026| DOI:10.15680/IJIRCCE.2026.1401014
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

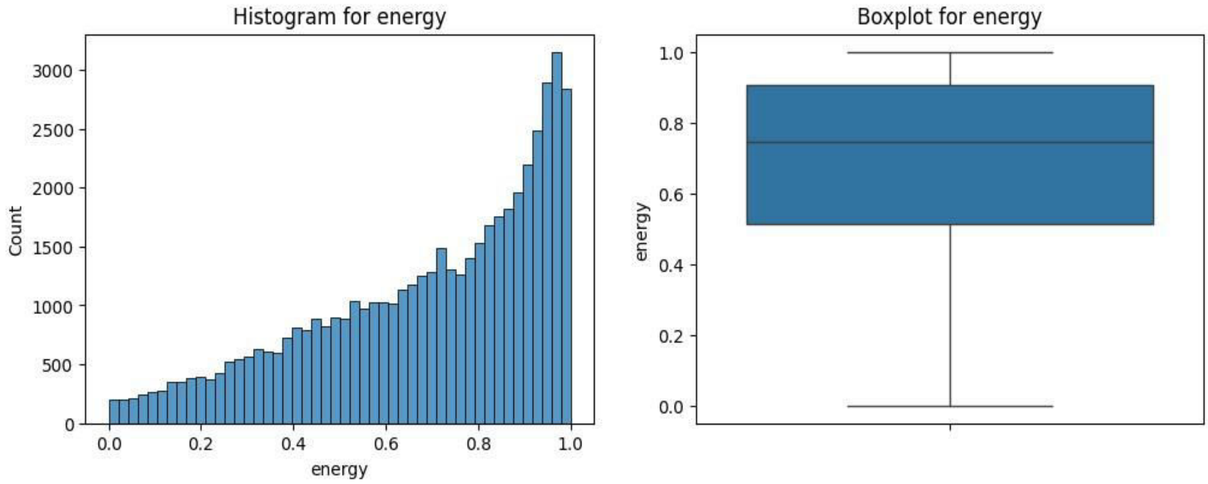


Fig 4.2 Graph of counts of frequency

4.3. Users Data Path Connection

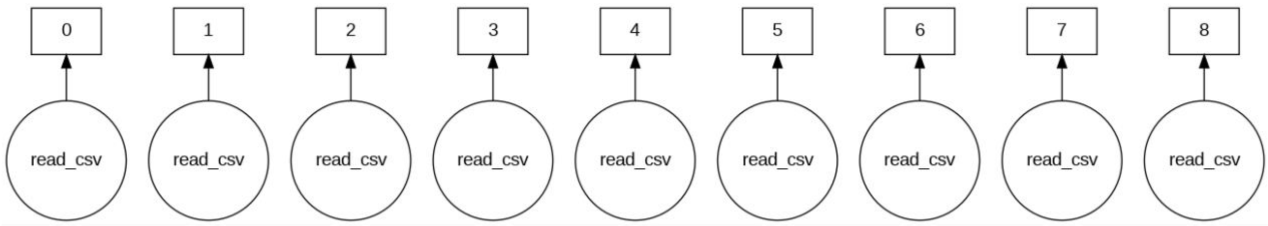


Fig 4.3 Users Data Path Connection

Figure 4.3 shows a parallel data ingestion workflow. In this setup, multiple CSV files, or parts of a large CSV, are read at the same time using repeated read_csv operations. Each numbered block, from 0 to 8, represents a separate task that loads a part of the dataset into memory. This design is often used in distributed or parallel computing frameworks, like Dask or Spark, to speed up data loading and preprocessing. By breaking the data into chunks and reading them simultaneously, the system improves scalability, reduces I/O bottlenecks, and allows for better handling of large datasets. This method is ideal for projects like your Spotify hybrid recommendation system, which needs to process large music metadata and user listening history files efficiently.

This diagram shows a parallel data loading architecture. Here, multiple CSV files (indexed 0 through 8) are read at the same time using the read_csv function. Each numbered box represents a different file or partition, while the oval below shows the read operation being performed on that specific file. The arrows indicate the direction of data flow from the read operations to their related file indices. This method demonstrates multi-threaded or distributed data processing. Instead of reading files one by one, all nine files are loaded at once. This architecture is often used in big data frameworks and data engineering pipelines to improve performance by using parallel I/O operations and cutting down total processing time.

4.4. Percentage of songs wrt to key

Figure 4.4 The chart shows the percentage distribution of songs with respect to musical key (0–11) in the dataset. It highlights how songs are spread across different pitch classes used in music theory.

1. Keys 2, 7, 9, and 0 have the highest percentages. This shows that more songs are written in these keys, making them more common in the dataset.
2. Keys like 3 and 10 have lower percentages, suggesting fewer songs are produced in those tonalities.

© 2026 IJIRCCE | Volume 14, Issue 1, January 2026| DOI:10.15680/IJIRCCE.2026.1401014
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

3. This uneven distribution shows real-life trends in music. Some keys are favoured because of vocal comfort, instrument tuning, or genre traditions.
4. Such insights are useful in a music recommendation system, key-based patterns can help improve similarity matching and improve content-based recommendations by matching musical harmony preferences.

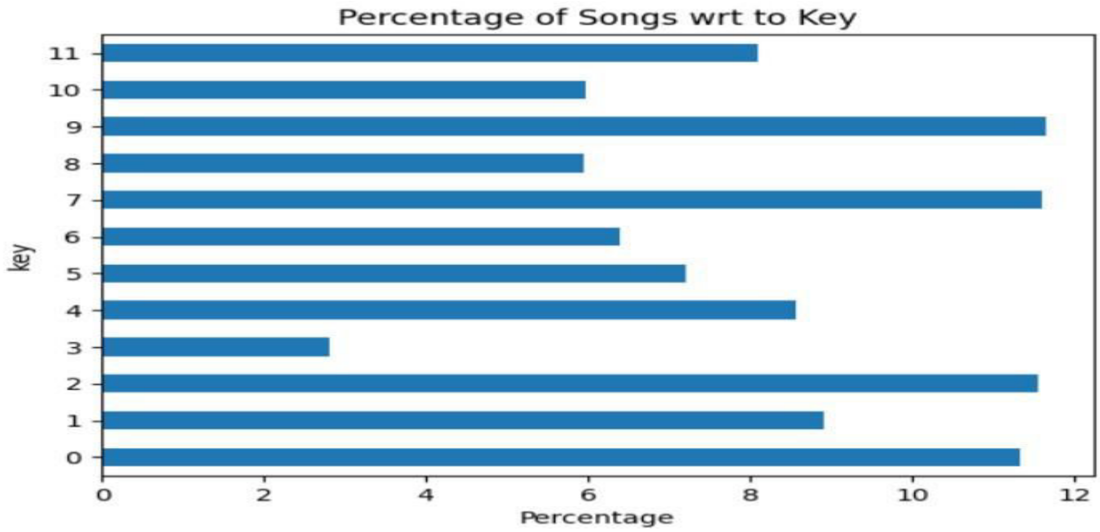


Fig 4.4 percentage of songs with respect to key

4.5. Time duration in minutes
Figure 4.5 The figure is a box plot of song time duration (in minutes), showing how track lengths are distributed in the dataset.
1. Most songs usually last between 3 and 5 minutes. This length is common for popular and streaming-friendly music.
2. The median duration lies around the middle of this range, indicating that half of the songs are close to standard radiolength tracks.
3. A large number of outliers go well above the upper whisker, they reach beyond 20, 40, and even 60 minutes, these likely represent live recordings, mixes, podcasts, or extended instrumental tracks.

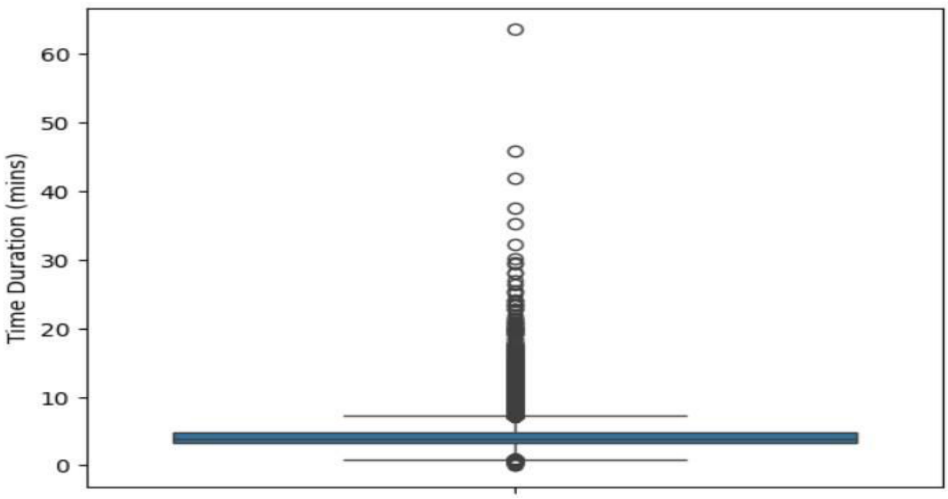


Fig 4.5 Time duration in minutes

© 2026 IJIRCCE | Volume 14, Issue 1, January 2026| DOI:10.15680/IJIRCCE.2026.1401014

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

4.6. Spotify Song Recommendation Dashboard

Figure 4.6 The Spotify Song Recommender interface offers a straightforward and interactive way for users to find music based on their tastes. By entering a song or artist name, users can create personalized recommendations that match their listening habits. They can control the number of recommendations and adjust diversity to find a balance between similarity and exploration. Overall, the interface links a hybrid recommendation engine with a clean and easy-to-use experience. It allows users to try different inputs and quickly see how recommendations change, which increases engagement and discoverability. This interface makes the hybrid recommendation system practical for everyday use, allowing non-technical users to access sophisticated algorithms.

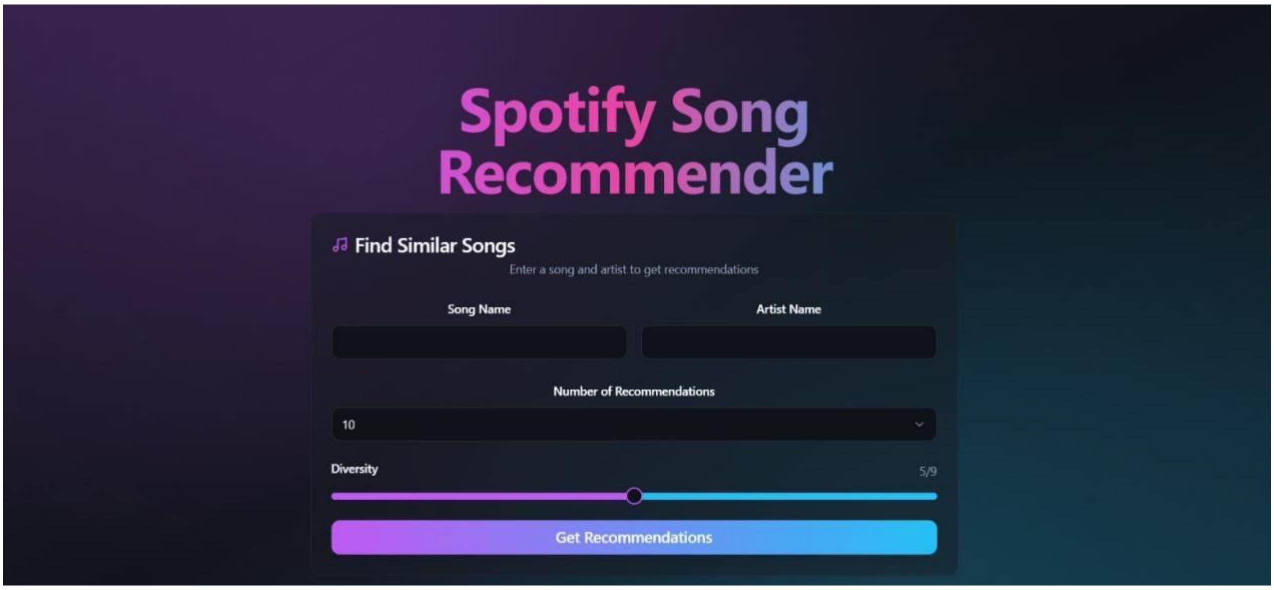


Fig 4.6 Spotify Song Recommendation Dashboard

4.9. Searching for Song as Input

Figure 5.8 shows a music recommendation system based on the input song "Whenever Wherever" by Shakira. It features beautiful album artwork with stars that creates a nice atmosphere. Below the song title and artist name, there is a large purple preview button. This button lets users listen to the selected track before getting recommendations. The dark background and contrasting purple accents give it a modern and sleek look, similar to other music streaming platforms. This layout emphasizes the main selection process, clearly indicating that users need to choose or confirm their input song first. The preview feature adds an interactive touch, allowing users to check their song choice before the recommendation algorithm takes over.

The recommended songs section features a horizontal carousel of four different album covers. Each cover represents a unique musical suggestion based on the input track's audio traits and data. The varied images include an intense sunset, a peaceful landscape with people, a forest scene, and a calm lakeside structure. These visuals hint at different music genres and moods within the recommendations. The thumbnails are both visually appealing and helpful for users looking through suggestions. The design shows that the recommendation engine looks at factors like energy, danceability, acousticness , and mood from Shakira's song to find tracks that are thematically and acoustically similar. This creates a more personalized music discovery experience.

© 2026 IJIRCCE | Volume 14, Issue 1, January 2026| DOI:10.15680/IJIRCCE.2026.1401014

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

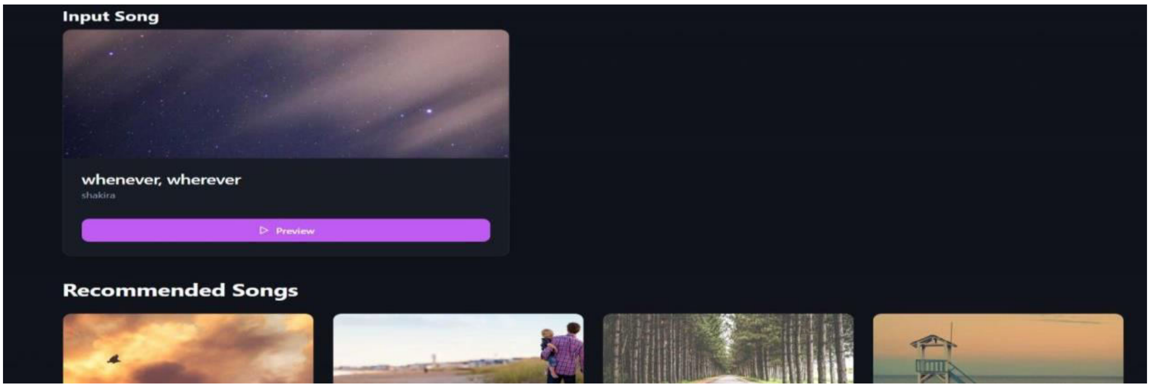


Fig 4.7 Searching for Song as Input

4.8. Recommended Songs for Input

Figure 5.8 The image shows the output screen of the Spotify Song Recommender application after a user submits an input track. At the top, the chosen input song, “Whenever Wherever” by Shakira, is displayed along with its artwork and a preview option. This confirms the reference used for recommendations. Below it, a grid of suggested songs appears, featuring album images, song titles, artist names, and preview buttons for quick listening. This layout shows how the hybrid recommendation system provides personalized, visually appealing, and easy-to-explore music suggestions to improve user discovery.

The recommendations feature tracks from the same artists and related ones. This shows that both similarity in content and shared listening habits are taken into account. The card-based layout lets users compare several suggestions at once without cluttering the interface. Preview buttons promote quick interaction and help users check out recommendations before adding them to their playlists. Overall, the screen strikes a good balance between smart algorithms and user-friendly design.

The Recommended Songs section features a grid of eye-catching album cards. Each card shows artwork, a track title, and the artist's name to help users explore. The top row highlights songs like “why wait” and “hips don’t lie” by Shakira, along with “oops!... I did it again” and “perfect lover” by Britney Spears. This setup indicates that the recommendations are related to the input Latin-pop track in both theme and style. Each card has a bold purple Preview button, making it easy to sample songs without leaving the page. The uniform card layout and dark background create an engaging browsing experience that focuses attention on the music. Overall, this interface shows how a recommendation system brings forward familiar artists and similar songs.

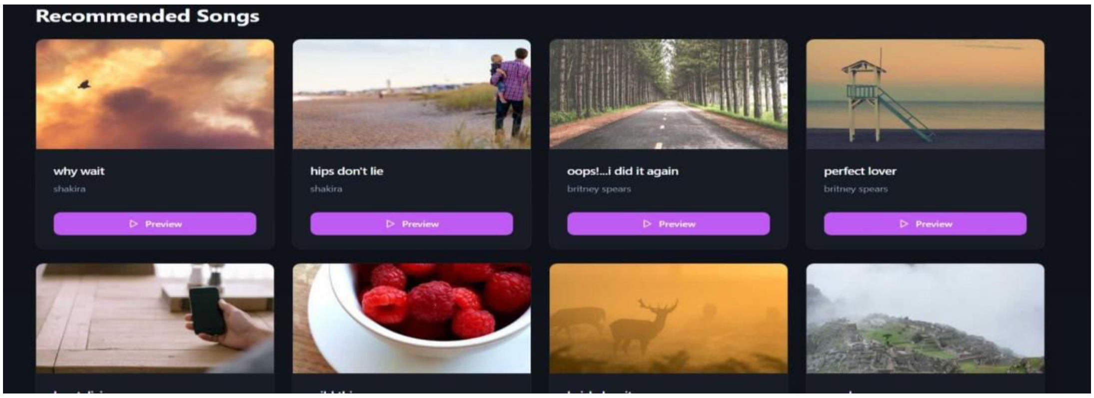


Fig 4.8 Recommended Songs for Input

The Recommended Songs grid displays each track as a clear card that includes cover art, song name, and artist. This design makes it easy to scan and visually appealing. A dark background and bright purple Preview buttons create strong contrast and draw attention to the main action, which is sampling each recommendation. By featuring well-known

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

artists like Shakira and Britney Spears, the interface helps users feel confident that the recommendation model understands their listening preferences.

The lower row of recommended songs introduces visual diversity by featuring distinct imagery. These include a close-up of an object on a wooden floor, a bowl of fresh raspberries, a misty deer in golden grassland, and a mountainous landscape. This suggests that the recommendation algorithm looks at not just audio features but also mood and visual context. These varied images appeal to users' feelings, showing that discovering music relates to lifestyle, atmosphere, and visual storytelling. The clear grid layout lets users quickly see all options at once, helping them explore and make decisions faster.

The recommendation system's two-row layout makes vertical scrolling easy while keeping a clear visual order. This setup allows users to browse through eight songs without feeling overwhelmed. Each card has the same size and spacing, creating a predictable layout that is easy to scan. This reduces mental effort and encourages users to engage with more suggestions. The consistent use of purple accents, especially on the Preview buttons, creates a unified visual identity and helps users quickly identify the action point. By mixing well-known artists with varied images and using clean, organized typography, the interface effectively balances discoverability and usability, making it a strong frontend for a hybrid Spotify recommendation.

V. CONCLUSION AND FUTURE SCOPE

The hybrid Spotify recommendation system developed in this project shows that combining content-based and collaborative filtering results in better, more diverse, and user-focused music suggestions than either method alone. By cleaning raw Music Info and User Listening History datasets and changing them into solid feature and interaction matrices, the system provides reliable input for similarity calculations at scale. It effectively merges cosine-based content and collaborative similarities through a tunable weighting scheme. This setup helps the model balance intrinsic song similarity with actual listening behaviour, which reduces cold start and sparsity issues.

The complete pipeline—from data ingestion and cleaning to hybrid score aggregation and top k selection—forms a clear, modular structure that is easy to maintain and expand. The literature on hybrid recommenders and Spotify-style systems backs the design choices made, confirming that these structures usually improve precision, recall, and user satisfaction. With suitable hardware and a Python-based software stack, the system can expand to larger datasets using sparse representations and tools like dask without major redesign.

The current setup also lays a solid foundation for future improvements, including deeper machine learning models, online learning, and real-time feedback loops. Adding extra signals like mood, playlists, and user skip behaviour can further improve personalization and discovery. Overall, this project delivers a working prototype that closely resembles real-world music recommendation workflows and acts as a practical reference for creating scalable, hybrid recommender systems.

Beyond its algorithmic contributions, the hybrid Spotify recommendation system demonstrates strong engineering practices and user-centre design that make it suitable for real deployment scenarios. The web interface clearly lays out the main process: selecting an input track, previewing it, and exploring recommended songs. This setup keeps complex backend logic hidden behind an intuitive and visually engaging experience. Logging, modular scripts, and clear structures for data loading, similarity computation, and ranking make the codebase easy to maintain and simpler for future developers to debug or extend. The project also focuses on evaluation and interpretability, using metrics, visualizations, and case studies to show why specific tracks are recommended. This approach helps build trust among stakeholders and end users. Overall, the system not only confirms key ideas in recommendersystem research but also shows how to transform them into a production-ready pipeline that can adapt to new data, methods, and product needs.

Future work on this hybrid Spotify recommendation system can focus on improving both the data signals and the underlying models to enhance personalization quality and reliability. Integrating multi-modal features such as raw audio embeddings, lyrics embeddings, and contextual metadata like time of day, device type, and coarse location would allow the system to capture a better representation of user taste and listening situations, as suggested in recent deep learning-based music recommender research.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

On the modelling side, the current cosine similarity framework can be expanded with neural recommenders like autoencoders, sequence-aware RNN or LSTM models, and reinforcement learning agents that learn optimal song sequences from ongoing user feedback, reflecting trends in modern hybrid architectures.

Finally, future iterations should include features for fairness, transparency, and control. This means providing explanations for why tracks are recommended, controls for diversity and mood, and mechanisms to prevent over-recommending popular artists. These improvements will help keep the system trustworthy, inclusive, and responsive to changing user needs.

REFERENCES

[1] Sheela Kathavate, “Music Recommendation System using Content and Collaborative Filtering Methods,” International Journal of Engineering Research & Technology (IJERT), 2021

[2] K. Yoshii, M. Goto, K. Komatani, T. Ogata and H. G. Okuno, “Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences,” Proceedings of ISMIR, 2006

[3] N. V. D. Malleswari, K. Gayatri, K. Y. S. Kumar et al., “Music Recommendation System using Hybrid Approach,” IJARIIE, 2023.

[4] Mahaboob Basha Sk, S. Sriharsha, L. Vyshnavi and G. Dhathrik, “User Based Spotify Recommendation System using Machine Learning Algorithms,” IJISRT, 2024

[5] Jain K. N. et al., “Music Recommendation System,” IJARIIE, 2022

[6] “A Survey of Music Recommendation System,” International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 2018

[7] “Content-driven Music Recommendation: Evolution, State of the Art and Challenges,” ScienceDirect Journal Article, 2024.

[8] S. Vashistha et al., “A Novel Music Recommendation System Using Filtering Techniques,” Informatica Journal, 2024

[9] “Spotify Recommendation System,” IRJMETS, 2024

[10] “Inside Spotify’s Recommendation System: A Complete Guide,” Music Tomorrow Blog, 2025

[12] rishti Vashistha, Deepika Varshney, Eva Sarin and Simran Kaur, “A Novel Music Recommendation System Using Filtering Techniques,” Informatica, 2024

[13] Adi Riyan Prasetya and Desi Ramayanti, “A Model-based Music Recommender System using Collaborative Filtering Technique,” International Journal of Computer Applications (IJCA), 2024

[14] Singh and Porwal, “Hybrid Music Recommendation System Utilizing Item Recognition and Content Analysis,” 2022

[15] Y. Chen et al., “A Music Recommendation System Based on Collaborative Filtering Algorithms,” IEEE Conference Paper, 2022.