# Spotify Music Recommendation System

Shravani Gurram
*Dept of Computer Science and Artificial Intellegence*
*SR University*
Warangal, India
shravanigurram04@gmail.com

Amanta Phelix
*dept.of Computer Science*
*Viswajyothi College of Engineering and Technology*
Ernakulam, India
amantaphelix@gmail.com

Anjali Garrepalli
*dept of Computer Science*
*SR University*
Warangal, India
garrepallianjali93@gmail.com

Vijay Pantula
*Dept of Computer Science and Artificial Intellegence*
*VKR, VNB & AGK College of Engineering*
Gudivada, India
pantulavijay0@gmail.com

***Abstract*—The recommendation of music is critical in improving the experiences of users with streaming services. This project converts a Spotify Music Recommendation System where inputs as artists, genres, or tracks of the user can be easily inputted by the users. In contrast to conventional systems having discouraging manual coding procedures, the considered approach is based on powerful data-oriented approaches and content-based paradigm to determine users' preferences and propose appropriate tracks. Due to these additions, the system guarantees the recommendations of existential listening histories and other metadata. The approach proves that the system is capable of providing useful and easily digestible recommendations and thus presents a light but powerful solution to today's music recommendation problems.**

***Index Terms*—Content-Based Filtering, Personalized Recommendations, Streaming Platforms, User Preferences, Music Metadata, Data-Driven Algorithms.**

## I. Introduction

With the increase in consumption of digital music streaming, companies like Spotify offer people such a huge catalog of songs that it may be difficult for them to find songs that really suit their preference. Among all the elements, personalization of the music recommendation is vitally important for improving the customers' experiences. Due to using millions of users, A recommendation system is an important feature in order to keep track with a large amount of content and helping the users to find the track of their interest according to users' preferences. The objective of this project is to develop a use case of a recommendation system for Spotify that compiles customized lists with songs based on the user's preferences regarding artists, genres or tracks for a more efficient and user oriented streaming.

In contrast, the proposed system avoids using a more complicated set of features such as metadata and rather works with the input parameters provided by the user like artists, genres, and tracks. The purpose is to give suggestions which correspond to the user's orientations on the music without employing further tags or detailed activity data. Using these few but critical inputs, the system is designed to return recommendations that anyone from the casual listener to the loyal music lover, may find useful and convenient to navigate the sea of sound.

The system applies rules-based algorithms to accomplish a correlation between users, their preferences and music. Recommendation by similarity relies on the preferences of like-minded individuals for tracks, while rule-based filtering can instantly match the user's inputs, such as the genre of the recommended tracks mirrors the specific genre of the user's favorite tracks. This way, they are highly personalized and yet based on the user's deliberate choices; thus, the system is maximally open and easy to comprehend.

The main objective of Music Recommendation System is to design an appropriate music search process. When people want to listen to their favorite songs, the system has this to offer but when people want new music from a niche interest, the system surprises you with recommendation that feels both new and appropriate. Thus, by covering the distance between the overabundance of input and the listener's expectation and preference in terms of the music selection, the system guarantees that every session becomes engaging and fun.

## II. Literature Review

This paper [5] presents an insight on the use of collaborative filtering techniques in e-commerce systems with major emphasis on using other users' preference to recommend products to users. It outlines the use of collaborative filtering in actual systems and further examines different issues and approaches to the recommendation systems.

This chapter [4] from the "Recommender Systems Handbook" provides a most recent survey of the recommendation system techniques concentrating on two main categories: collaborative filtering and content-based recommendation. It includes the background and issues on recommendation systems, and provides visions on the historical development and possible futures of recommender science.

In this paper [3]focuses on Netflix recommender system which works based on collaborative method filtering as well

as others to suggest movies and shows for users to watch according to their history. The authors describe the system and explain how it affects businesses, citing how it utilises data to improve user experience in a novel manner.

This paper [2]begins with a general description of the item based CF algorithms, with specific emphasis on how the technique works by filtering based on items as opposed to users. The authors describe that the item-based techniques have the efficiency of scalability and higher precision than the traditional user-based techniques in the big data set, especially in e commerce and media systems.

Here in this survey paper [1], a comprehensive literature of state-of-the-art recommender systems and both collaborative and content-based filtering techniques have been discussed. It covers the several concepts of personalization, the problem for each type of method and the foreseeable next-step in personalization such as combined and contextual based.

This article [6]focuses on the growth of the Spotify recommendation system and the core principles based on which recommendation is produced using machine learning algorithms and a user's profile. The outlined paper concerns the issue of scalability and accuracy of recommendation system for the music and the significance of the collaborative filtering in increasing the traffic of the target audience on the website.

The system [7] presented in this paper is a user-based collaborative filtering approach for recommending music. The authors also develop a music recommendation system in which recommendations of music tracks in the music database are made and influenced by other similar users in term of user ratings as well as listening history. The work is centered around enhancing the quality and qualities of scalability when it comes to recommendation systems with special regard (but not limited) to music streaming.

## III. Methodology

The music recommendation system entails a well-defined procedure of training as well as development of k-Nearest Neighbors (k-NN) to generate personalized recommendations concerning the feature that may contain track properties, artist behavioral patterns, and genre details.

### A. Data Processing

The first and crucial step of the presented methodology involves data pre-processing for model preparation. The data set used combines several features: track names, artists, genres, and other numerical characteristics of the songs (tempo, loudness, and key). Since features of two datasets need to be on the same scale, the scikit-learn standard scaling tool is used to scale the data. This is essential for the algorithm in the k-NN to not allow some feature with increased numerical values to have a much larger distance compared to the rest of the features for the desired k number of results. After scaling, both training and test datasets are prepared, and the features are standardized for the training and test set.

### B. Training the k-NN Model

The first and the crucial step of the presented methodology involves data preprocessing for model preparation. The dataset used combines several features: track names, artists, genres, and other numerical characteristics of the songs (tempo, loudness, and key). Since features of two datasets need to be on the same scale, the StandardScaler from scikit-learn is used to scale the data. This is essential so that the algorithm in the k-NN does not allow some feature with increased numerical values to have a way larger distance compared to the rest of the features for the desired k number of results. After scaling, both training and test datasets are prepared and features are standardize for training and test set.

### C. Evaluation of the Recommendation System

The primary recommendation system in place is the k-Nearest Neighbors (k-NN) algorithm which is performed on the newly scaled features of the dataset. It is a non-parametric technique for predicting or classifying a point on a space to its neighboring points in the feature space. Here the distance measure is Cosine as it is well implemented among the distance measures for comparing the data points of form Music track vectors. The k-NN model is trained with ten neighbors (parameters n-neighbors=20) to find the nearest tracks from given test samples. The model is trained on the training data, where it becomes necessary for the model to learn the nearest neighbor based on the generality of features of tracks with similar music type.

*1) Track Recommendation Based on Similarity:* The recommendation functionality is developed based on the k-NN model that is used to recommend tracks similar to a particular input track. For particular track in the database the index of necessary record is received, and using the k-nearest neighbors model distances to neighbors are calculated. The artists of these recommended tracks are also returned to the user and their distances or similarities from the query songs. Two fields of a plain text string are shown – the name of the track, as well as the name of the artist, while distance measure is used to evaluate the proximity of the recommendation. These suggestions are provided in an easily understandable order, with tracks that in some way are most similar to the input track listed first, thus allowing the user to navigate through music similar to what the user already likes in an efficient manner.

*2) Genre-Based Recommendation:* The genre-based recommendation system employs the k-Nearest Neighbors (k-NN) approach to generate tracks as well as the selected track of the user or similar genre. Complementarily, training the model to consider only specific features which are associated with a given genre allows the model to learn similarities of the tracks from the same genre. To make sure genre-related features are nicely compared, the dataset is preprocessed and scaled. If a user chooses a track, the system searches for the trajectory's neighbors based on the genre proximity and offers tracks closest to the user's musical preferences. The approach allows

a user to look for songs within a particular genre or find sub genres they were not aware of before.

*3) Artist-Based Recommendation:* The recommendation system based on an artist selects tracks using the k-Nearest Neighbors (k-NN) algorithm of artists similar to the user selected artist. Following data pre-processing and selection of the 'artist' feature, the dimensionality of the data is normalized to bring all features to a level with each other. Then, the k-NN model is followed by identifying tracks of other artists that are similar in either style or other features. After training, the model predicts tracks and suggests the input artist from the calculated the nearest neighbors with tracks associated with similar artists. It comes in handy to allows the users to find new music by artists they would probably not have come across on their own.

## IV. Results

The music recommendation system offered the functionality of recommending music to the user based on an artist or genre, as suggested by the user. The implemented strategies applied the k-Nearest Neighbors (k-NN) for track recommendation based on the track selected by the user.

### A. Track Based Recommendations

It also provide the users search for a particular track, and it suggested other tracks that had close similar features to the entered track. Depending on what song a user liked, the system suggested other songs with the same speed, style or having other features of the liked songs and so was able to introduce songs similar to what the user would like.

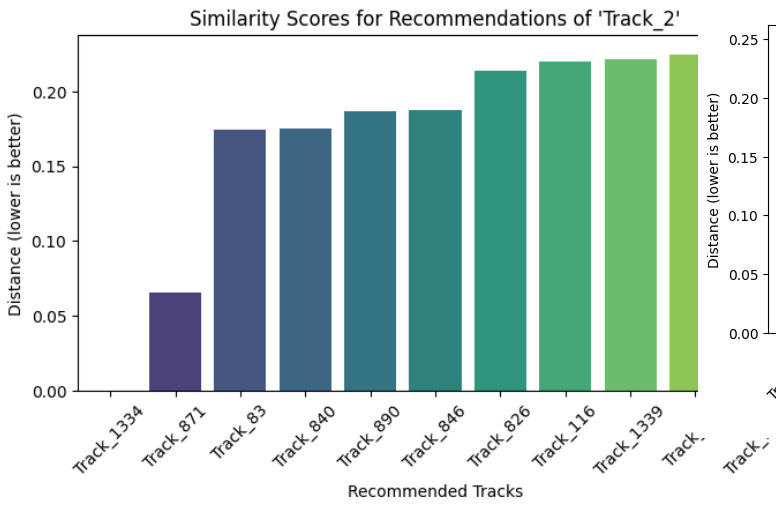


Fig. 1. Track Based Recommendations

### B. Genre Based Recommendations

Another feature that could be implemented was a Genre taste; that allowed users to set a particular genre, say 'Pop', the system will recommend songs in the same generic or in generic closely related to it. For instance, if a user was a fan of pop music, he/she would receive suggestions for tracks from pop artists or from the feature pop or other categories, and based on this, use the system for finding new music that suits his/her preference.

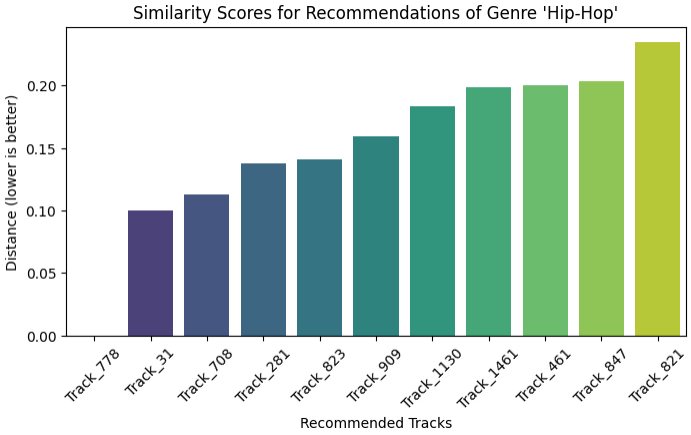


Fig. 2. Genre Based Recommendations

### C. Artist Based Recommendations

And when the user has chosen the specific artist, the system suggested other tracks of artists with related music genres. For instance, if a user selected "Artist A", the system provided other tracks from other Artist A and Artist B who has a similar kind of music or genre as of the selected one. This let a user suddenly find that he might potentially like an artist whom he has never listened to before.

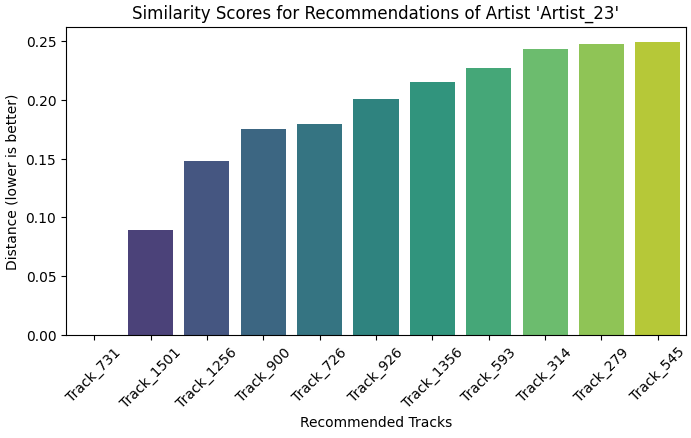


Fig. 3. Artist Based Recommendations

### D. Interactive Streamlit Music Recommendation App

The music recommendation system was developed in the form of an Streamlit app that accepts as input the artist, genre or track. A slider is a gadget that enables user configuring how many recommendations they want, ranging from 1 to 20. In this case, the k-Nearest Neighbors (k-NN) algorithm has been used to make the recommendation as close to real-time as possible depending on the inputs from the user. It also

includes relevance ratings indicating how closely each track most recommended to the users fits their choices. This app also helps the users to explore new music that they love easily with less efforts.

## V. Conclusion

The discovered k-Nearest Neighbors (k-NN) based music recommendation system has been found particularly useful for niche music recommendations. Since users can enter an artist, a genre, or track, the system provides users with relevant and personalized recommendations for listening to more music that various users might like. The inclusion of an ”application developed using Streamlit” improves navigation as the app features a slider that allows the number of recommendations to be set at users' choice ranging from 1-20. Similarly, real time recommendations accompanied with real-time similarity scores enhance user experience and satisfaction. Thus, even with the problems associated with similarity thresholds, the system can be considered tuned between precision and variety, providing useful recommendations. In compelete, this system enhances the experience of finding music that one can listen to, thus can be referred as a must instrument for music lovers.

## References


[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] B. Badrul Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[3] C. A. Gómez-Uribe and N. Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):1–19, 2016.

[4] F. Ricci, L. Rokach, and B. Shapira. Recommender systems: Introduction and challenges. In *Recommender Systems Handbook*, pages 1–34. Springer, 2015.

[5] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1-2):115–153, 2001.

[6] M. Sullivan. The spotify playbook: Building a music recommendation system, 2013. TechCrunch.

[7] D. Vázquez and S. Bravo. A collaborative filtering model for music recommendation using a user-based approach. In *Proceedings of the 18th International Conference on Enterprise Information Systems*, 2016.