A Novel Music Recommendation System Using Filtering Techniques

Srishti Vashishtha[1], Deepika Varshney[2], Eva Sarin[3], Simran Kaur[4]
[1,3,4]Computer science and engineering dept., Bharati Vidyapeeth's College of Engineering, GGSIPU New Delhi, India
[2]Computer science and engineering dept., Jaypee Institute of information Technology, Noida, India
E-mail: deepikavarshney06@gmail.com

Keywords: music recommendation system, content-based filtering, collaborative filtering, TFIDF, KNN, cosine similarity

Received: November 17, 2023

With the growth of the World Wide Web, a large amount of music data is available on the Internet. A large number of people listen to music online rather than downloading and listening offline. But only some sites provide personalized and accurately recommended songs while they listen to an auto-playing playlist. Hence the need for recommendation systems arises. Two approaches can be applied to the recommendation system: content based filtering and collaborative filtering. While in content-based filtering approach, analysis on the songs' content which has been preferred by the user in history is done and the songs with relative similarity are recommended. While latter suggests songs that certain users of similar listening pattern have preferred. But collaborative filtering-based recommendation systems not only requires time to attain stability but also might recommend unsuitable music because it is not personalized to each user's preference. Also, the latter requires the songs to be listened by a few users already to recommend it any other user. Hence for overall analysis, answers to few questions need to be implemented in recommendation systems: the very first one is how the properties should be analyzed, next one is how the analysis should be done and last is how the songs related to the user's preference should be chosen. So, for suggesting the better system which solves these three questions as well as provides better and personalized recommendations, In this paper we build a collaborative filtering as well as content based filtering recommendation system, where various factors determine the essence of the songs, i.e. liveliness, keys used, loudness, pitch, valence, etc., are analyzed for comparison. From the experimental analysis in has been identified that content-based filtering technique performed best on KNN machine learning classifier with accuracy of 85%.

Povzetek: Članek predstavi nov sistem za priporočanje glasbe, ki uporablja tehnike filtriranja na osnovi vsebine in sodelovalno filtriranje. Z uporabo metod, kot sta TF-IDF in KNN, raziskuje izboljšanje personaliziranih priporočil za uporabnike.

1 Introduction

Music is an important vehicle for communicating to other people something relevant about our personality, history, etc. [9]. Music files are easily available on websites for free in this era. To this date, every single online music site holds over 2,000,000 songs [1]. More than 1,000 songs are available on storages of various personal computers (PCs) or hard disks equipped in MP3 players [5]. But only a few provide relevant music recommendations from this huge amount of music data available. Hence it becomes necessary to generate personalized recommendations using music recommendation systems [2]. For listeners, when choosing a new song to listen to, the song's contained affective content plays a huge role in its selection which should be as per their preference and music taste. However, when users have to pick a song from their current playlist, they often tend to rely on their current emotions to select the song they feel they want to listen to the most. Thus, a need for a music recommendation system prevails that not only understands the current feelings of the user but also understands what user might want to listen in similar moods. [8]. Music recommendation systems can be based on a lot of factors, on a user's facial expressions [7], past music listening behavior [11], current emotions [15], or other user's preferences based on sentiments [44] as well. The job of such system is filtering out every piece of content that is available and provides with only that part of information which provides interested content. [6]. The two approaches to such recommendation systems: 1. Content-based filtering and collaborative filtering. Collaborative filtering is based on the idea that similar users tend to behave in similar manner in similar situations [6]. Hence, it considers the actions, past behavior, user history, and preferences of other users for similar songs to the user of interest. While the idea behind the content-based filtering method offers the user of interest similar subjects as listened to by him previously. The difference between the two approaches is that in the latter the similarities between two songs is not evaluated on user actions, which is the case in the former, but on the features of the object, i.e. song itself.

Music recommendation systems can be called information retrieval tools, as well. [25] It means that there is a certain information demand in the audience which needs to be dealt with. While making a recommendation system, dealing with this demand becomes difficult since most of the time, the user in interest doesn't always know what type of music, in particular, he or she wants to listen to. In such cases, both the approaches mentioned above provide quality recommendations. But music is not objective and accepted universally [26]. Music conveys emotion and at the same moment plays a major role in changing the mood of the listener. Therefore, one cannot always rely on only user history for generating recommendations, but consider a comparison with different songs for similarities as well.

A large number of people listen to music online rather than downloading and listening offline. But only some sites provide personalized and accurately recommended songs while they listen to an auto-playing playlist. The purpose of this research work is to provide music recommendations to listeners and answer to few questions. The questions are: how the properties should be analyzed, next one is how the analysis should be done and last is how the songs related to the user's preference should be chosen.

Hence, this paper surveys both collaborative, as well as content-based filtering, approaches for comparison, which generates better and quality recommendations in terms of variety, genre, themes as well as emotions. For content-based filtering, various mathematical tools like TF IDF and cosine similarity will help in the generation of recommendations, while in the case of collaborative filtering, recommendation systems based on machine learning algorithms like KNN will be experimented on for generating recommendations.

The paper is organized in the mentioned sequence: Section 1 contains introduction, Section 2 has the literature review of recent works, Section 3 describes the proposed architecture of recommendation systems, the experimental setups and dataset description, Section 4 discusses the results, Section 5 is about the conclusion drawn along with future work followed by references.

2 Related work

The authors in [1] proposed a mechanism for providing tailored services. Authors employed STFT (Shortest Time Fourier Form) to examine properties of music. They presented a dynamic K-means clustering technique to examine users' preferences, which clustered the songs in the music list dynamically changing number of clusters. They created a music database with 100 songs from ballad, rock, jazz, and classical genres. Each piece's feature was divided by the average, which set the maximum radius (Rmax) to 0.5 and the cell size to 0.2. 80% of recommended pieces were near to the users' desire. Rock and classical music are the most frequently recommended genres. The user's preferred genres account for 74% of the total (34 percent rock + 40% classical).

The Music Suggestion System (MRS) was created to provide a personalized music recommendation service [2]. MIDI format musical instruments were first tested. The feature track of each polyphonic musical element was first determined, followed by six elements in the track. Use music object parameters that include tone, duration, and sound. Of the 100 MIDI files, the results showed an accuracy of 83%. The K-means algorithm has been used to classify the same database based on six recovered elements.

The DJ-Running initiative is a research endeavor aimed at tracking both the mental and physical exertion of athletes during training sessions to comprehend their emotional state. Its goal is to promptly curate the most fitting music selections to enhance motivation and performance. This article [3] outlines a renowned music support system devised within this project, which tailors the next song choice based on factors like the user's location, mood, and activity type. Leveraging Spotify's search technology, the system interprets user cues and transforms them into recommendation algorithms. The underlying database encompasses three key services: user data, geographic data, and music distribution, all constructed using the Spring framework. In a separate study by the authors of [4], they introduced a unified music recognition system coupled with automated genre classification and emotion-based categorization. This system employs frequency coefficients adjusted for logarithmic scale modulation, a novel and expedited characteristic. All relevant features are derived from the MP3 cutting tool, effectively halving the extraction time. The primary challenge proposed is to amalgamate and enhance the efficacy of music classification and classification theory through the utilization of the AdaBoost algorithm. The study utilized a dataset comprising 1000 songs (for uniform classification) and 800 songs (for emotional data). Results indicate a notable improvement in classification accuracy for five genres (classical, pop, hip-hop, soul, and punk), with accuracy climbing from 86.8% to 92.2% when compared to traditional methods. Accuracy of four-emotion classification (sad, calm, pleasant, and excited) was increased from 86.0% to 90.5%.

A content-based recommender system was developed by researchers in [5]. The system employs two strategies: 1. Acoustic feature analysis and 2. Application of deep learning and computer vision approaches to improve the recommender system's outcomes. The tasks were divided in 2 subtasks: A. In order to create an acceptable vector space representation of music composition of particular dimensions; B. In order to evaluate vector representations of songs. Distance metrics like Euclidean, Manhattan and Cosine Distances were used to determine the similarities between vector representations of the acoustic characteristics. The similarity between vector representations of auditory properties was determined using distances. Random recommendation, genre-specific random recommendations, auditory characteristics analysis, and artificial neural network all had precision of

0.006, 0.066, 0.112, and 0.148, respectively. Apparently, the quality of the direct integration of the vector expression type was high when using the neural network. The author's hypotheses that the vector representation created by ANN included more content about music were also validated by the results.

S Metilda Florence and M Uma in [6] presented users with suggestions that matched their interests. The authors hoped that by creating a recommendation system, they would be able to help a user decide which music to listen to in order to reduce stress levels. Emotion extraction module, Audio extraction module, and Emotion-Audio Integration Module were most commonly used methods. The HELEN dataset, which had over 2000 photos, was utilized to train the classifier which was used to detect facial landmarks from the user's face. The CK extensive dataset was utilized to determine user's expressed emotion. When this is detected, the music player chooses a song that best suits the user's mood. The system's accuracy was 80%. Disadvantage of this technology is that it must be tested in various lighting conditions to evaluate its robustness. If the classifier intends to determine the emotions of the user, the image quality must be at least 320p. The authors of [7] created a music recommendation system that tracks emotions of the user and offers songs by presenting a list of songs that is arranged appropriately based on the user's current emotion. The system ranks songs based on 2 criteria: relevance to the preferences of user; and influence on mentality of user’s feelings. Sensors record the user's bio-signal data, which is subsequently used as input for the emotion detection process. Using RF-ECG sensor, the system was taught to determine emotional state of user based on temperature of skin and heartbeat rate. The algorithm extracted the user's emotion with an estimated accuracy of 64.5% according to results. 6.66% of songs were disliked, 8.57% songs were categorized as “bad influence” by users from the list of songs recommended to them. 3.6/5 is the average scale for EmuPlayer’s work’s satisfation. EmuPlayer fails to give accurate results and doesn’t play the highest scored songs. In [8], researchers devised a recommendation mechanism leveraging Friend of a Friend (FOAF) and Rich Site Summary (RSS) vocabularies to suggest music to users based on various aspects of their musical taste, including psychological factors such as personality, demographic preferences, socio-economic status, and explicit musical inclinations. Web content was syndicated using XML format, with user preferences outlined through FOAF documents. User profiling, derived from the user's FOAF description, context-based information gleaned from music-related RSS feeds, and content-based descriptions extracted from the audio itself, facilitated music discovery. The system operated by: 1. Extracting interests from the user's FOAF profile; 2. Identifying artists and bands; 3. Selecting related artists based on those encountered in the user's FOAF profile; and 4. Ranking the results by relevance. The system achieved a 71.2 percent accuracy rate but fell short in accurately discerning the user's true emotions and musical preferences.

In [9], Yuri S. and Takayuki I. unveiled MusiCube, a music recommendation system that displayed a selection of songs as coloured icons in a 2-D cubic environment and provided a user interface for selecting suggested tunes intuitively. The authors demonstrated how MusiCube effectively depicted clouds of icons corresponding to sets of users' favourite tunes in a 2D cubic space. MusiCube initially calculates the feature values of tunes, then initialises the system, recommends numerous tunes by changing the colors of icons, receives user feedback, changes the colours of listening tunes' icons, and continues evolutionary computing to the next generation. The results are divided into three groups: positive (red), negative (orange), and not yet determined (yellow).

The authors' suggested system in [10] aimed to recommend music that the user like, are new to their ears, and fit the user's listening behaviour. For song recommendations, five factors were considered: genre, year, favour, freshness, and time pattern. The likelihood of playing a song at that time is computed using a Gaussian Mixture Model, which represents time pattern of listening. To forecast the genre, a web wrapper was created to collect genre information from AllMusic.com (a website that uses information to determine song genres). The ARIMA algorithm was used to forecast the next probable year and calculate the likelihood of a recording year. The Forgetting Curve was used to determine how fresh a song was to a user. The system was effective and accurate. (T-RECSYS) created an accurate recommendation system with real-time prediction by combining content-based and collaborative filtering as input to a deep learning classification model. The authors of [11] used data from Spotify Recsys Challenge to test their method. For content-based filtering, T-RECSYS considers six areas of metadata: genre, artist type, artist era, mood, tempo, and release year. T-RECSYS calculated the Sorenson index for each pair for collaborative filtering. Content-based and collaborative filtering information were combined to create input vector. The model was built, trained, and deployed using Google's Tensorflow and the Keras Python library. T-RECSYS can be used with Amazon, iTunes, Netflix, and other services. Real-time updates and many variable inputs were missing from the model.

To improve the Content Based technique, authors in [12] presented the TICI (Transaction-Interest-Count-Interest) method. They used two parameters: Count-Interest and Transaction-Interest, to allow users to choose which weight from the options offered they want to highlight. The CB approach could discover the most recently popular music group based on the user's access history; however, the result was unbalanced. Authors' proposed formula highlighted the importance of time as well as the number of musicians in a group. They place a high value on the passage of time. The TICI approach took into account the number of members in a music group and when they first appeared, allowing it to determine the group's weight rank more precisely than the CB method. This factor and system proved that TICI is a more accurate and efficient method than the CB method. Authors aimed to offer a better formula (COL-Collaborative technique) for successful music data grouping than the CB method [13]. The COL technique would result in the users being grouped together because the supports of the groups with varied densities are the same. As a result, the TICI approach was proposed to improve the CB method, while the DI (Density-Interest) method was proposed to improve the COL method. The DI approach calculated music group support and took into account the distributions of the music group's appearances. In terms of weight difference, the results confirmed that the TICI approach could outperform the CB method. In terms of Hamming distance, the DI technique may outperform the COL method because DI considers the density of the appearance of the music group, allowing DI to distinguish users with various access behaviors more clearly than the COL method.

In comparison to the brute-force method, the researchers presented a method of measuring the acoustic distance between pre-classified music files with same sort of emotion in [14]. This method considerably sped up the search process and increased precision. The AdaBoost method was used to classify the music signature derived from the entire music database into predetermined music emotion classes. A combination of numerous elements is the main feature of the music database used. Sad, peaceful, pleasant, and exciting were the four music emotions evaluated. The great precision of music classification led to a higher recall rate and search speed than the previous brute-force method.

The authors add time scheduling to the music playlist to the already existing approaches, and incorporated decision tree categorization technology to help people find music that suits them better. A music recommendation system with Hybrid time scheduling was built in a web environment using decision-tree classification learning as its fundamental architecture. In the initial step, the system collects personal data such as gender, age, occupation, and favoured music genres. The feedback from users was then captured, including the marked time, content elements, and evaluation information. In the second step, groups of people with similar tastes were joined together via collaborative filtering. Player platform was developed using C#, system interface using Flash to build platform of MuPa recommendation system. The overall precision of the system is 78.33%. The diverse set of approaches for music recommendation system is shown in Table 1.

Wang et al. (2023) proposed a Multi-view Enhanced Graph Attention Network (named MEGAN) for session-based music recommendation. MEGAN can learn informative representations (embeddings) of music pieces and users from heterogeneous information based on graph neural network and attention mechanism. MEGAN achieves better performance than baselines, including several state-of-the-art recommendation methods [45]. Liu et al. designed a novel emotion-based personalized music recommendation framework to meet users' emotional needs and help improve their mental status. In this framework, authors designed a LSTM-based model to select the most suitable music based on users' mood in previous period and current emotion stimulus. A care factor was used to adjust the results so that users' mental status could be improved by the recommendation [46]. A real-time system that can recognize human faces, assess human emotions, and even recommend music to users. This system deploys deep learning-based CNN model, it can predict six emotions: anger, fear, joy, neutral, sadness, and surprise. The proposed system can be utilized in different places where real-time facial recognition plays an important role. It has achieved accuracy of 73.02% [47].

Table 1: State of the art techniques for music recommendation system

Sr. No.	Theme	Year of Publish	Methodology	Results	Dataset
1	MSR with Dynamic K-means Clustering Algorithm	2007	• Shortest Time Fourier Form • K-means Clustering Algorithm	• Avg. Rmax: 0.5 • Avg. cell size: 0.2 • Accuracy: 80% • MSR recommended mainly rock and classical music	Database consisted of 100 pieces each from: Jazz, ballad, classical, rock
2	MSR using Music Data Grouping and User Interest	2010	• K-means Algorithm	• 83% accuracy	100 MIDI files
3	DJ-Running MSR	2019	• Nearest Neighbor Search Algorithm • Kubernetes Tech	Songs are selected considering the runner's profile, location and emotional state	Spotify
4	MSR using Classification	2019	• Classification on basis of Emotions and Genre • AdaBoost Algorithm	Accuracy of: • Five Genre Classification: 92.2% • Four-emotion Classification: 90.5%	• Genre Dataset (1000 songs) • Emotion Dataset (800 songs)
5	Content-based Music Recommendation System	2021	• Acoustic Feature Analysis • DL and Computer Vision • Euclidean, Manhattan, Cosine Distances	Precision: • Random recommendation: 0.006 • Genre-specific: 0.066 Acoustic analysis: 0.112	• MSD (Million Songs Dataset) • FMA (Free Music Archive): 106,000 tracks

6	MSR for Emotion Detection using Facial Expression	2020	• Music Extraction Module • Audio Extraction Module • Emotion-audio integration Module	Accuracy of 80%	• Cohn Kanade Extended • HELEN
7	EmuPlayer: Based on User Emotion	2010	• Bio-signal Data of User • RF-ECG sensor	Accuracy: • Liked Songs: 64.5% • Disliked songs: 6.66% • Bad-influence: 8.57% Avg. satisfaction scale: 3.6/5	• MY SQL database was used to study user's emotions and mental state
8	FOAF & RSS Music Recommendation System	2005	• XML/RDF files • TF-IDF • Collaborative Filtering • Content-based Filtering	System's accuracy: 71.2%	• MP3 Blogs • Podcast sessions
9	MusiCube: Visual MRS	2011	• iGA • PCA	Colour Representation in 2-D plane: • Positive: Red • Negative: Orange Not detected: Yellow	• RWC Music Database
10	Next One Player	2011	• ARIMA • Gaussian Mixture Model • Forgetting Curve	System was able to fit a user's taste and adjust recommendation strategy quickly whenever user skips a song	• AllMusic • ID3v1 or ID3v2
11	T-RECSYS: MRS using Deep Learning	2019	• Google Tensorflow • Python library Kreas • Content Based Filtering • Collaborative Filtering	• High accuracy Readily extensible to different market services like iTunes, Amazon	Modified Spotify Dataset
12	TICI: User-Interests Approach to Music Recommendation	2011	Content Based method + Rank of the group weight	More accurate that Content Based Method	User Behaviour and emotion
13	TICI: MSR based on Music Data Grouping	2011	• Collaborative method • Density-Interest	COL method is more accurate than CB with help of DI method	User's interest
14	MSR based on Music Emotion Classification	2017	• AdaBoost Algorithm • Modified Brute-Force method	Recall rate and search speed are better than brute force method	2000 songs (4emotions x 500 songs each)
15	MSR based on User behaviour in Time Slot	2009	• Decision Tree Classification • C# • Flash	Overall precision of system is 78.33%	Database developed using Microsoft Access 2003
16	Multi-view enhanced graph attention network for session-based music recommendation	2023	• Graph Neural Network • Heterogeneous Music Graph	MEAGAN achieves better Precision, Recall, F1, Mean Average Precision in session-based music recommendation tasks.	Real world datasets: Lastfm and 30Music
17	Emotion based personalized MRS	2023	• LSTM model • Care factor	Precision 0.83 Recall 4.17 MSE 0.032 RMSE 0.178 MAE 0.291	GTZAN dataset
18	Facial emotion recognition and MRS using CNN	2024	• CNN • Deep Learning	Accuracy 73.02%	OAHEGA and FER-2013 datasets

3 Proposed methodology

Content based MRS makes analysis of the content of every song the user has heard in history, revealing a common rule that essentially restores a user's listening behavior. Songs that match this rule are recommended. In this way, content based MRS can recommend songs which perfectly match the listening profile of the user. Unlike a collaborative filtering method, a content-based approach enables predicting by analyzing song tracks [28, 31]. It is based on retrieving information and filtering information [32] recommending a song similar to the one the user has listened to in the past than the user has rated it as 'like' [33, 34]. Many studies focus on extracting and comparing acoustic features in finding tracks of common understanding [35, 36]. Most represented so far are rhythm [37, 38, 39]. Basing on features which were extracted, the distance lying between tracks is calculated [40]. Some standard calculations of similar k-methods that combine the distance of the earth converter, the increase in expectation by the sample of Monte Carlo and the average vectors in the Euclidean range. [41]. As per the dataset parameters, we will be using expectation maximization technique for content-based filtering uses sampled vectors from parameters of the two comparable songs; the sampling is performed via generation of weights [42]. Here mathematical algorithms like cosine similarity, TF-IDF will be used. Eventually the songs will be recommended on the basis of their allotted personalized score according to the proposed methodology.

Figure 1: Content-Based filtering methodology

The steps to create recommendations are data wrangling, then creating workflows, connecting to the Spotify API, creating playlist vectors, and finally creating recommendations for users. The first step is the data preprocessing step that will be performed using a Python library called pandas. In this step, you can find comparison among the dance number, tone, life, voice, etc. of various opening songs has been done. Later we will be able to distinguish one of the names according to the type of music. The next step is to use feature engineering in conjunction with flexible variable customization to create hot code features and popular variants of the year. TF-IDF (Time Frequency and Inverse Document Frequency) properties for the actors are also created in this step. TF-IDF automatically displays metadata words according to their frequency (i.e., weight) in the entire list. The TFIDF diagram is shown in Figure 2 and Figure 3. The Figure 4 shows how to use TF-IDF in music application. The TF-IDF function creates a number vector of all songs by assigning a number value to each word in the word. The next step is to connect to Spotify API to download a specific playlist.

TF-IDF can be defined as a numerical statistic which indicates the significance of a word in given document. TF stands for term frequency while IDF stands inverse document frequency. The value of TF-IDF is directly proportional to the number of times a word occurs in the document and is offset by the frequency of the word in the corpus.

$$\boldsymbol{TF - IDF(t,r) = TF(t,r) * IDF(t)} \quad (1)$$

$$IDF(t) = log\frac{1+n_r}{1+df(r,t)} + 1 \quad (2)$$

When TF (t, r) indicates the number of times the word t appears in a review r, nr is the total number of revisions and the df (r, t) update number containing the word t. Reviews are considered the equivalent of a document. [27].

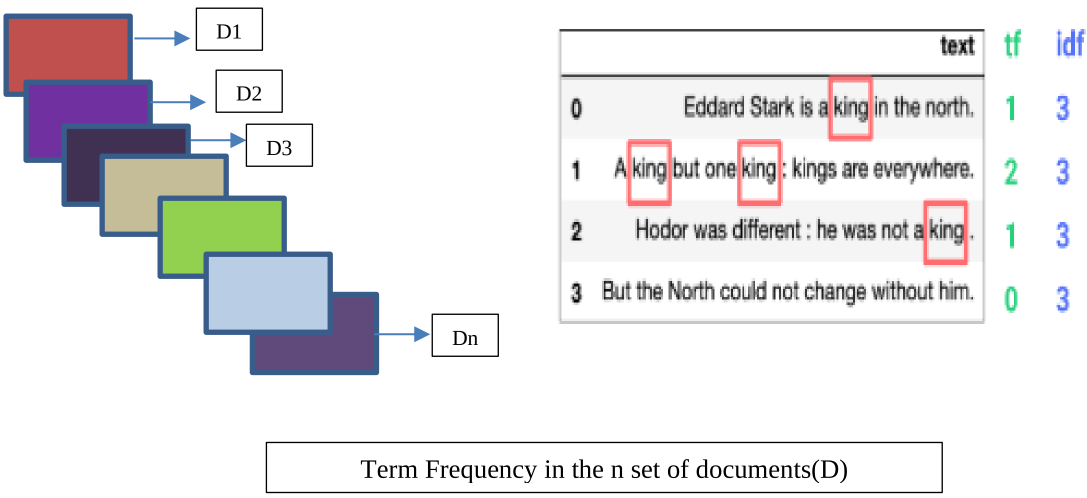

Figure 2: Pictorial representation to TF – IDF

$$cos\ (\theta)\ = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} \ A_i B_i}{\sqrt{\sum_{i=1}^{n}} \ A_i^2 \sqrt{\sum_{i=1}^{n}} \ B_i^2} \qquad \ldots\ldots. (3)$$

Eqn. (3) justifies the cosine similarity algorithm mathematically where A and B are vectors and

Where $\vec{a}.\ \vec{b}. = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + a_3 b_3 + \cdots \ldots a_n b_n$ is the dot product of the two vectors.

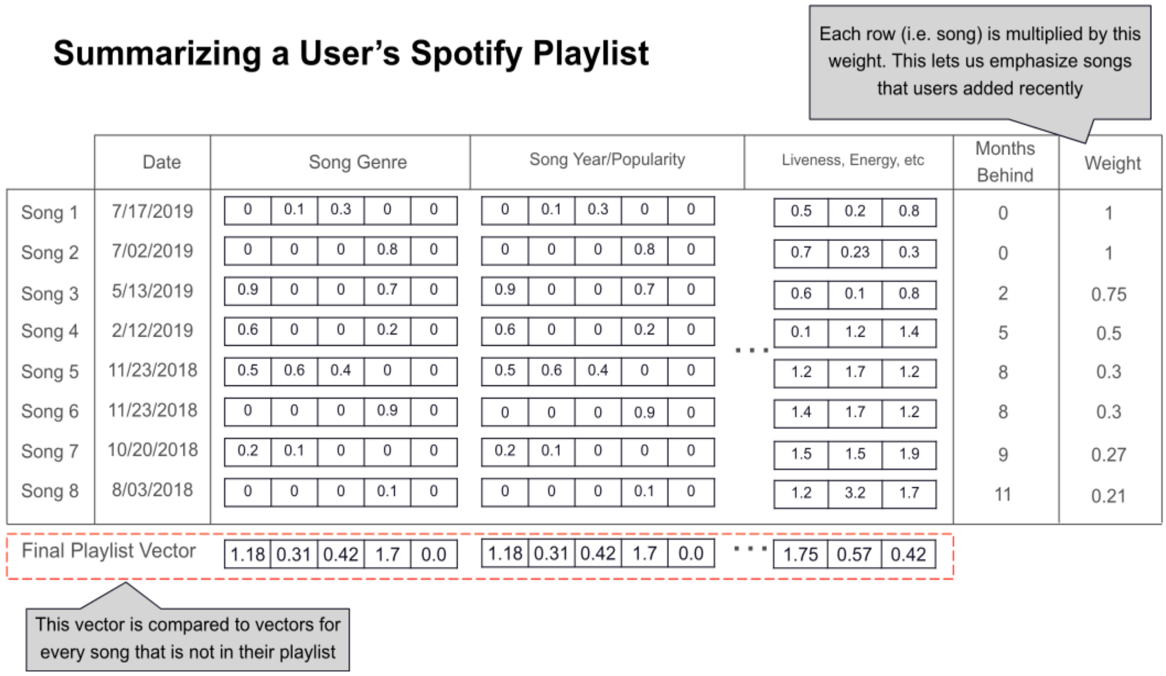

Figure 3: Summarizing a user's spotify playlist

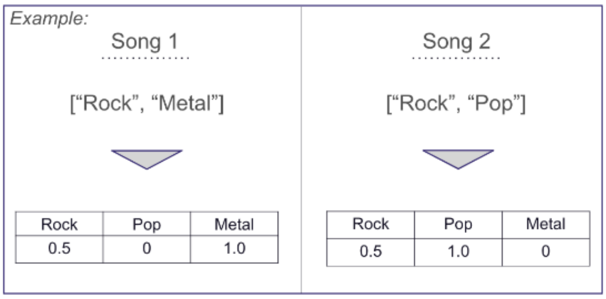

Figure 4: Example of TF – IDF

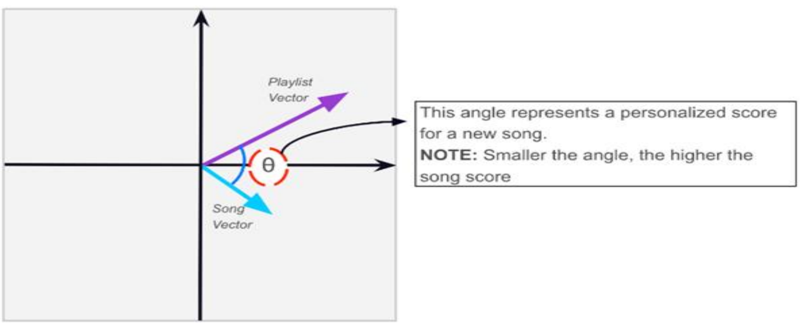

Figure 5: Cosine similarity

The playlist vector is produced by repeating each line namely, song, thunder year song and important things like life, dancing etc. by shared weight. Figure 5.3 represents a summary of the User's Playlist, which includes columns like Song name (with the ID), Date, Song Genre, Popularity, All Feature Variables, Months Behind, and Weight. 'Date' represents the release date of the song, the song genre specifies how much exact percentage of each genre a song comprises. Nowadays, many artists tend to merge a number of genres together, and only a few artists are true to just one music style/genre. This can be explained in detail in figure 5.4. If we consider 2 song choices, the first song, Song 1, say, is put under the genre Metal, but it does not mean that the song is entirely Metal, Metal itself is a subdivision of Rock. Rock itself comprises several songs, but when it comes to putting the songs under categories, it gets categorized into subcategories. Here, Song 1, will have both rock and metal properties, but no Pop elements. Hence, while calculating the Final Playlist Vector, we will take into account only Rock and Metal. Similarly, Song 2, has some Rock and Pop elements to it, but no Metal elements. Popularity is given in the form of a gradient where the value ranges from 1 to 100, where 1 signifies the least popularity score, and 100 is very popular and likely is on the billboard charts. However, these columns are all that were in the original dataset as well, and the only new columns are Months Behind and Weight. The 'Months Behind' column signifies "When was the song added in the playlist" if the song was added in the same month, it's given a weight 0, if it has been 2 months, the weight becomes 0.75. The weight simply means recency bias and has a float value. The larger the recency bias value, the more priority the song gets. This is due to the reason that we need to focus on the user's most recent taste. We need to consider what the user likes most nowadays and not something that is outdated for them. The recommendations need to bias toward the new songs that attract users' attention. We needed songs that reflected the user's current/recent taste, hence considered weight. In the end, we just add up all the columns and get a final playlist vector, and this is where the 'Weight' actually comes into play. Since the weight is multiplied across all the rows, the last song, Song 8, will not have much significance while generating recommendations, since it has the lowest weight and hence will be affecting the sum as well. Cosine similarity helps in generating usic recommendations in the last step. There are more than 2600 inputs that go into this model, but here, we need to imagine that all those inputs are one arrow, since these inputs are just vectors. Each row of our feature would be considered a single vector. In Fig 5, both the arrows are vectors and have the same attributes, one represents the playlist of the user and the other represents the song. What cosine similarity actually does is that it takes the angle between both of these arrows/vectors. The angle represents how good of a recommendation that song is. The smaller the angle is, the higher the song score is. If the song vector and the playlist vector are pointing in the same direction at a very small angle, it means that the song is a good fit for the playlist. And that's how the recommendations are made.

3.1 Collaborative-based filtering methodology & experimental setup

This approach involves clustering users based on their preferences, facilitating the sharing and refinement of music among users within the same group. The filtering process typically relies on analyzing the content of genres, artists, or albums extracted from users' listening and download histories. One significant advantage of this method is the high likelihood that users will discover unexpected yet familiar songs through the recommendations. [8]. While collaborative filtering encompasses three categories—model-based, memory-based, and hybrid collaborative—we focus on model-based collaborative filtering in this context. This approach utilizes data mining and machine learning algorithms to train the system according to other users' preferences. It essentially represents user preferences using a set of rating scales and constructs a unique prediction model for comparison with other users' preferences. [28]. The system generates test forecasts and real-world data based on a known model. One of the primary challenges is generating online recommendations within a reasonable timeframe while handling large datasets. KNN (K-Nearest Neighbors) stands out as one of the most effective neighboring algorithms for collaborative filtering with large datasets. This algorithm considers the central ratings of each user, often referred to as a pivot [29]. By comparing this pivot value, the algorithm initiates the search for the most similar music, ultimately recommending the one with the highest likelihood estimation to the user [30].

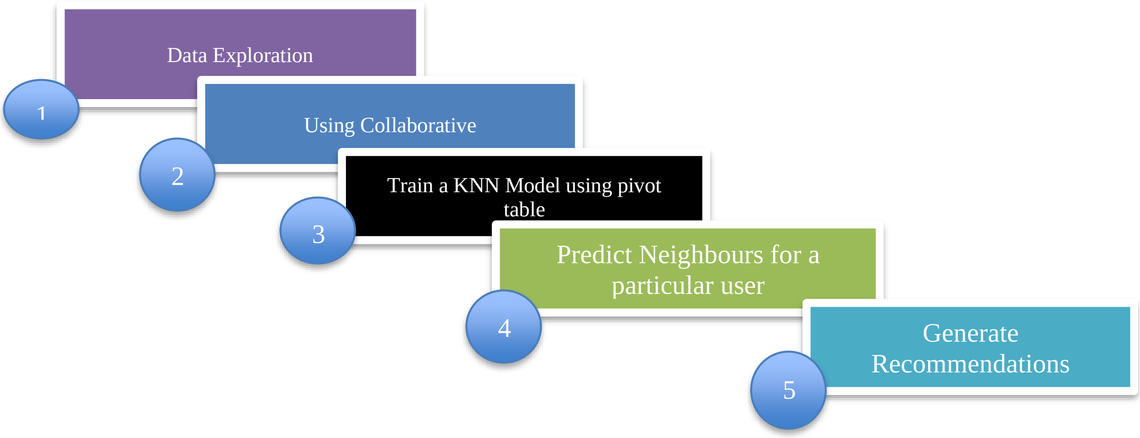

Figure 6: Collaborative-Based filtering methodology

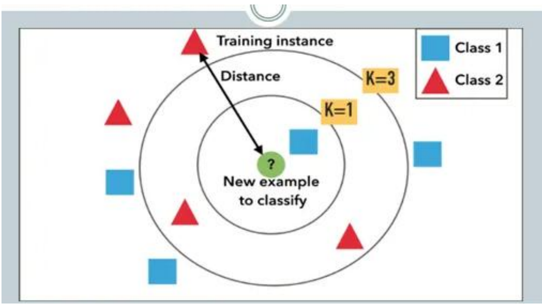

Fig 7: KNN algorithm decision process[1]

The prediction r_{ui} is set as:

$$r_{ui} = \frac{\sum_{v \in N_i^k(u)} sim(u,v) * r_{vi}}{\sum_{v \in N_i^k(u)} sim(u,v)} \quad (4)$$

Table 2: Parameter's measure

Parameters	Meaning
r_{ui}	The estimated rating of user u for item i.
$N_u^k(u)$	The *k* nearest neighbors of user *u* that have rated item *i*.
r_{vi}	The true rating of user *v* for item *i*.
Sim_options(dict)	A dictionary of options for the similarity measure.
k(int)	The number of neighbors to take into account for aggregation.

The meaning of each parameter in the formula (4) is shown in Table 2.
It can be seen from the formula that KNN is influenced by the ratings of the song given by other users as shown in Figure 6 and 7.

3.2 Dataset description

The user's Spotify Playlist is extracted using API calls, where the playlist includes a set of songs added by the user over a period, ranging from 0 to 12 months, i.e., a course of a year. In order to extract information about these songs, another dataset was used, which was available on Spotify's research website. The dataset itself consists of a set of a million playlists, including titles of those playlists, as well as the songs. It includes information about all the songs, including Song ID, Track, Artist Name, rhythm, vitality, pitch, volume, manner, vocal presence, acoustic quality, instrumental presence, liveliness, emotional tone, pace, song's time signature, date of release, and so forth.
We have considered a personalized playlist where the songs are not only the mainstream sellouts but also the less popular, yet appreciable music pieces. The playlists are customizable meaning, the songs can be added, deleted, or rearranged at any point in time. All the songs have the (approximate) same variable scores (example, valence, loudness, danceability, etc.). For instance, the songs "Body" and "Working for It", possess identical danceability scores of 0.752 and 0.776 respectively, and comparable energies as well. However, the songs do seem to be of fairly different keys when compared. However, the majority of songs are in key 11, popularly called key Bb (B flat). There are 12 keys in music: C, Db, D, Eb, E, F, Gb, G, Ab, A, Bb, B. As for the time signature, every song has a time signature of 4. Apart from 2 outliers, the entire dataset seems to be very consistent.

[1]https://medium.com/machine-learning-researcher/k-nearest-neighbors-in-machine-learning-e794014abd2a

The personalized playlist, however, includes the name of the song, artist name, and date added. There are a total of 16 songs in the playlist, making the total duration 54:06 minutes. The songs added are by different artists to avoid a biased generation of recommendations. However, the time signature and mode of the songs are taken the same, 4 and 1, respectively.
There were minimal differences between the values of the variables, making it a consistent playlist, with minimal outliers.
We visualized the cover arts of the respective songs that are present in the inputted dataframe using matplotlib. The column numbers were specified (here, 5) in order to present the cover arts along with the song names in a structured manner. Spaces between the covers were

adjusted as well to avoid cluttering and provide enough space for the longer title names. The visual representation of playlist and set of playlists is shown Figure 8 and Table 3 respectively.

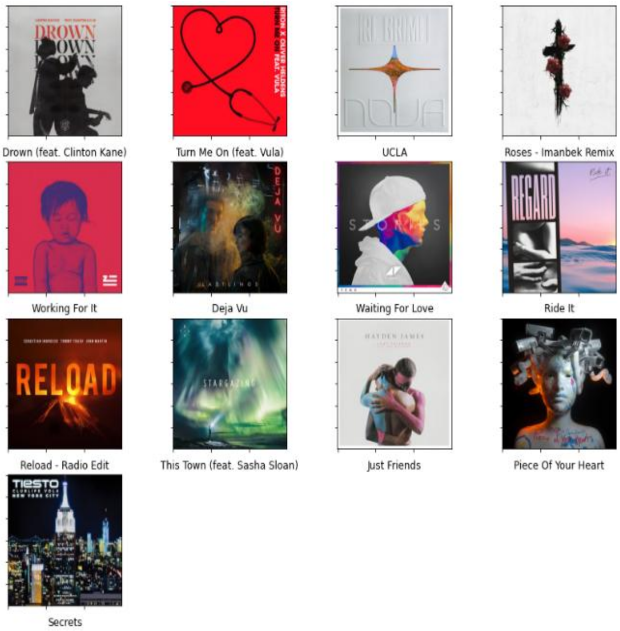

Figure 8: Visual representation of songs in our playlis

4 Experimental analysis and result

Upon analyzing the results of content-based recommendations, it was observed that the variables showed strikingly closer value to the songs in the original playlist. Taking an example of the recommendation, the song “Roots”, as generated, has a very close value of danceability, valence, tempo, energy, and key when compared to an average score of the original playlist table. The observed average variables were: Danceability: 0.67093, Energy: 0.75593, Key: 8, Loudness: -5.29927, Mode: 0, Speechiness: 0.06437, Acousticness: 0.11175, Instrumentalness: 0.07404, Liveness: 0.20707, valence: 0.51013, and Tempo: 124.56. Majority of the songs maintained danceability, energy, loudness, speechiness, and tempo, which can be seen as a major contributor and in the recommendations. Similarly, if we consider the song “Hold On (feat. Cheat Codes) - 2020 Edit”, we can see that the tempo is 124.859, which is very close to 124.5601, a liveness of 0.2630, and a loudness of -4.969, all of which are comparable to the ideal dataset values. However, there were major variations in mode (some songs were in major key some were in a minor key) like the song “No Money” and “Runaway (U & I)”, both being in major key, while all the songs in the dataset were in a minor key, and valence. These are Spotify’s Original Recommendations.

Initially, Spotify provides only 10 recommendations. Upon pressing the “Refresh” button provided, more songs can be recommended. But the limitation is that the songs will get repeated in the recommendations, and this step is highly influenced by the label the artists are under, and the popular songs in the country or worldwide at the moment. Hence, these are not true to the users’ taste. Along with this, when compared to the songs in the playlist, major deviances were observed, especially in Acousticness, Key, Mode, and Valence. However, there was a match in the time signature, except for 2 to 3 outliers. Upon analyzing the results of collaborative filtering, it was observed that the majority of the song recommendations were similar to Spotify's original recommendations, mainly the songs by Galantis and Steve Aoki. It was also observed that the songs were heavily influenced by the playlists of other users and the user’s overall preference. In this case, the user's top artists for the past 2 years were Tove Lo, Troye Sivan, and Panic! At the Disco. Hence, the recommendations were biased towards them, however, the recommendations were the remixes of the popular songs by these artists, since they fit the overall genre better as compared to original songs which have a significantly divergent tempo, valence, energy, and danceability. All the remixes that are recommended have done extremely well on the billboards in the past. Examples being One Kiss by Dua Lipa, Copycat by Billie Eilish, Attention by Charlie Puth, and Talking body by Tove Lo, Without Me by Halsey, to name a few. The details are shown in Figure 9, 10 and 11. The result analysis has been performed on 2 category of filtering techniques (Content based and collaborative filtering technique) as shown in Table 4. The Table 4 clearly shows that content-based filtering approach outperform earlier work.

Table 3: Our playlist

Artist	Name	Id	url	Date_added
Martin Garrix	Drown (feat. Clinton Kane)	4RVtBIHFKj51lpvpfv5ER4	https://i.scdn.co/image/ab67616d00001e02b154bc...	2020-08-01 01:27:34+00:00
Riton	Turn Me On (feat. Vula)	OqaWEvPkts34WF68r8Dzx9	https://i.scdn.co/image/ab67616d00001e02216a27.	2020-07-09 :34:03+00:00
RL Grime	UCLA	3OaunNUIXXs5e2PXtNAzzG	https://i.scdn.co/image/ab67616d00001e02eded2e...	2020-06-20 00:34:44-00:00
SAINt JHN	Roses - Imanbek Remix	7fPuWripwDcHm5aHCH5D9t	https://i.scdn.co/image/ab67616d00001e022b6e2f.	2020-04-19 06:26:21+00:00
Loud Luxury	Body	21RzyxY3EFaxVy6K4RqaU9	https://i.scdn.co/image/ab67616d00001e02af5e18...	2020-03-26 22:28:23+00:00
ZHU	Working For It	2HJQcyUpmUuvzS5vBAICIc	https://i.scdn.co/image/ab67616d00001e02bfaac9...	2019-12-19 15:53:47+00:00
Lastlings	Deja Vu	649HM5IOHHqsoG5nldMo6L	https://i.scdn.co/image/ab67616d00001e02129817.	2019-11-19 16:04:48+00:00

Avicii	Waiting For Love	2P4OICZRVAQcYAV2JReRfj	https://i.scdn.co/image/ab67616d00001e025393c5...	019-11-17 03:38:47+00:00
Regard	Ride It	2tnVG71enUj33lc2nFN6kZ	https://i.scdn.co/image/ab67616d00001e025c2781...	2019-11-13 04:13:21+00:00
Dimitri Vegas & Like Mike	Mammoth	76fqWMe0buqQoaNTIbLWmr	https://i.scdn.co/image/ab67616d00001e0216bf35...	2019-10-26 19:11:43+00:00
Sebastian Ingrosso	Reload Radio Edit	5jyUBKpmaH670zrXrEOwmO	https://i.scdn.co/image/ab67616d00001e0270e2e5...	2019-10-04 15:50:31+00:00
Kygo	This Town (feat. Sasha Sloan)	4aSfgWmRa9KsISD4Jmx7QB	https://i.scdn.co/image/ab67616d00001e02a33355.	2019-09-30 20:05:19+00:00
Hayden James	Just Friends	6tB4XVKceo2307SSWXaO0y	https://i.scdn.co/image/ab67616d00001e024b6940.	2019-09-30 20:04:53+00:00
MEDUZA	Piece Of Your Heart	1DFD5Fotzgn6yYXkYsKiGs	https://i.scdn.co/image/ab67616d00001e02ead130...	2019-09-30 20:04:47+00:00
Dimitri Vegas & Like Mike	Tremor - Sensation 2014 Anthem	6AE0G24YXnDyEgE4L0efpB	https://i.scdn.co/image/ab67616d00001e023d4c4f.	2019-09-30 20:04:43+00:00
Tiesto	Secrets	ONIC4unbe5KZOp1d9T7OaF	https://i.scdn.co/image/ab67616d00001e02de5f51...	2019-09-30 20:04:40+00:00

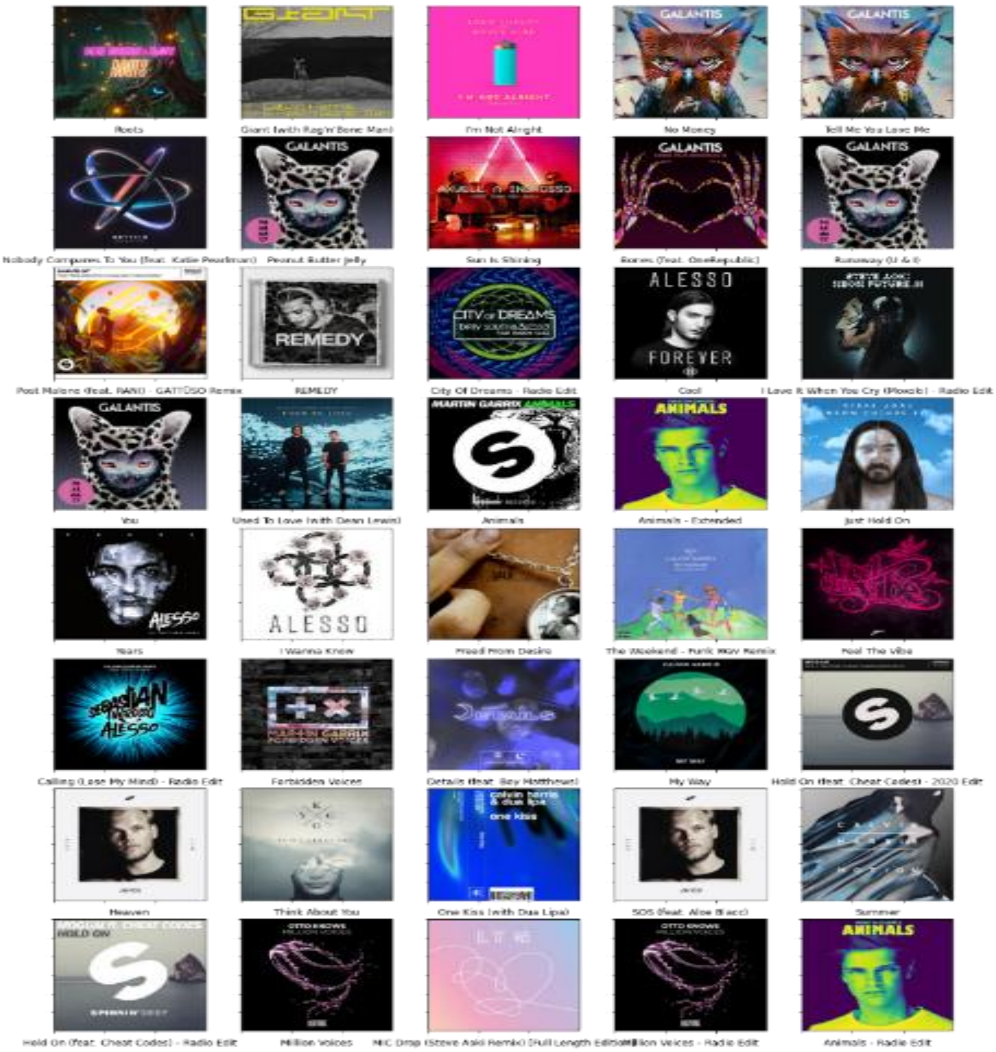

Figure 9: Content-Based recommendations

Table 4: Comparative result study of proposed work

Filtering Technique	Accuracy	Precision	Recall
Content based (Proposed work)	0.85	0.84	0.81
Collaborative based (Proposed work)	0.81	0.80	0.79
Earlier work	0.71	0.71	0.70

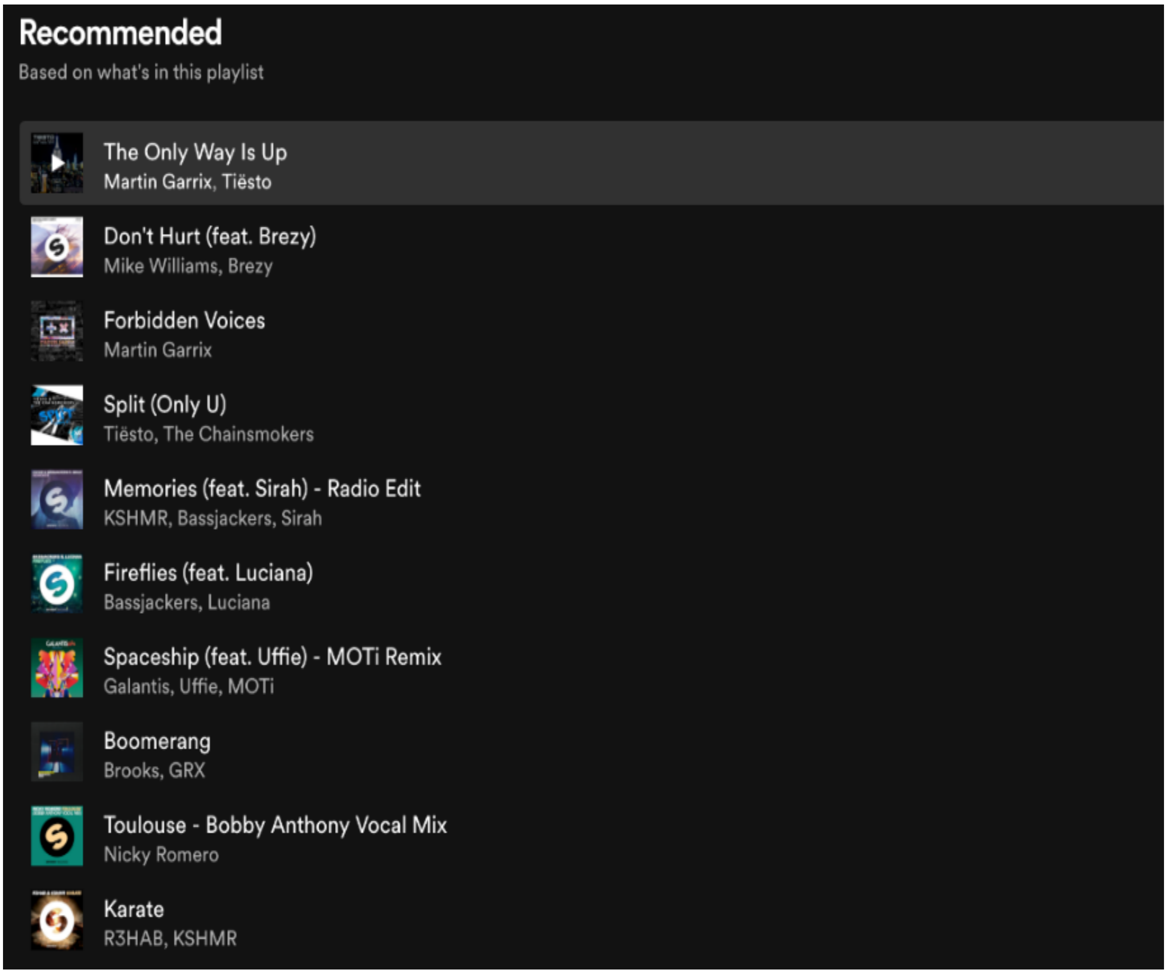

Figure 10: Spotify recommendations

4 Conclusion and future work

In this paper, we have proposed a novel approach to building a content based filtering and collaborative based music recommendation system using mathematical algorithms and KNN. The result of the Content-Based Recommendations was much closer to the user's initial taste than collaborative filtering, upon comparing the attributes, it was found that all variables were remarkably closer to the ones in the source playlist, except the time signature. . From the experimental analysis in has been identified that content-based filtering technique performed best on KNN machine learning classifier with accuracy of 85%.

Unlike Spotify's recommendation system which emphasizes mostly on valence, the proposed recommendation systems recommended songs with parameters like tempo, liveliness, danceability and loudness being in comparable range with those of the user's original playlist songs while keys used and mode observed high variations. Hence, recommendation systems should focus on more than just valence in order to recommend better and personalized music to listeners. The future work of this project includes implementation of a recommendation system based on deep learning in order to generate more accurate and user specific music recommendations for both content and collaborative based filtering.

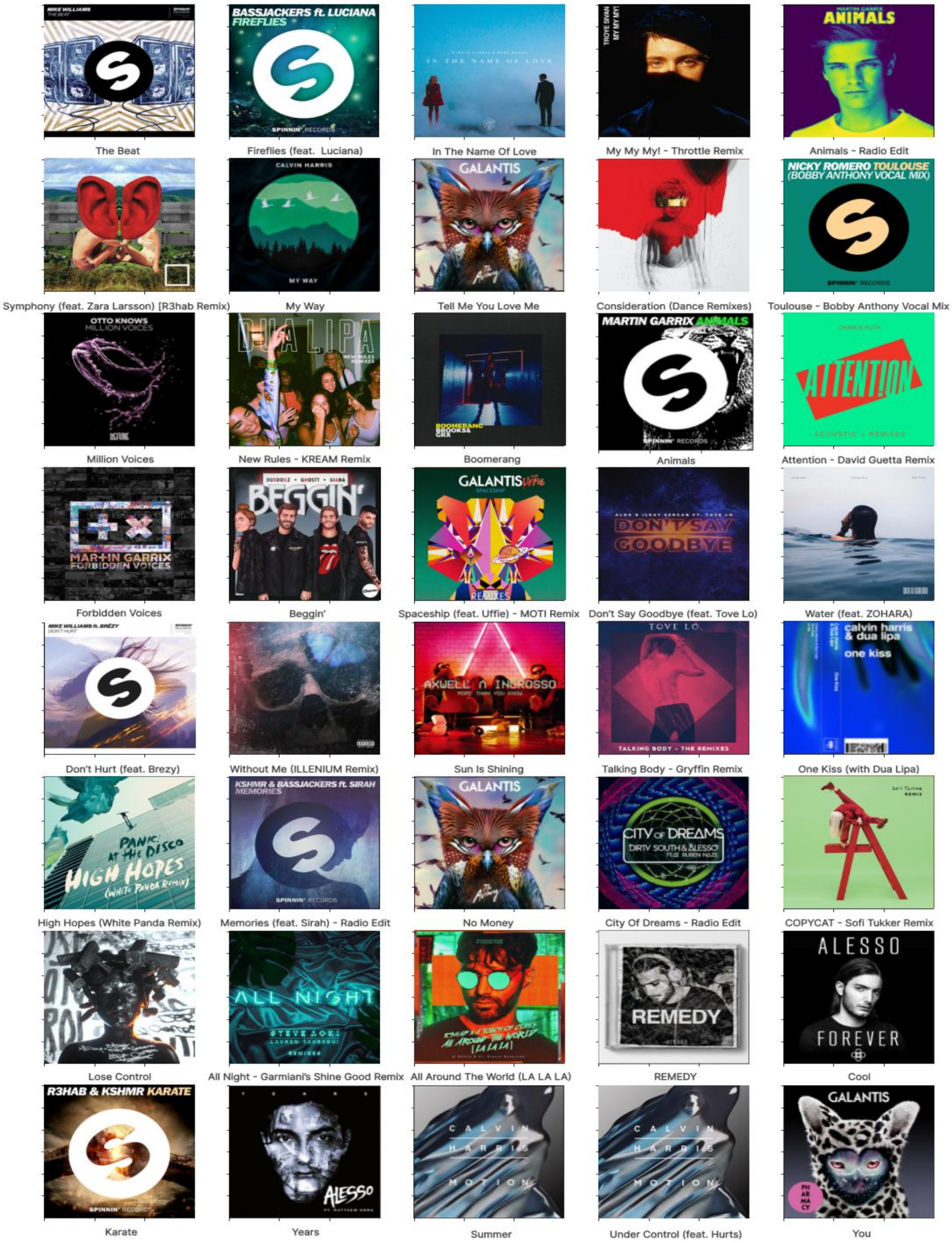

Figure 11: Collaborative based recommendations

References

[1] D. Kim, K. Kim, K. Park, J. Lee and K. M. Lee, "A music recommendation system with a dynamic k-means clustering algorithm," Sixth International Conference on Machine Learning and Applications (ICMLA 2007), 2007, pp. 399-403, doi: 10.1109/ICMLA.2007.97.

[2] Chen HC, Chen AL. A music recommendation system based on music data grouping and user interests. InProceedings of the tenth international conference on Information and knowledge management 2001 Oct 5 (pp. 231-238).

[3] Song Y, Dixon S, Pearce M. A survey of music recommendation systems and future perspectives. In9th international symposium on computer music

modeling and retrieval 2012 Jun 19 (Vol. 4, pp. 395-410).
[4] Schedl M. Deep learning in music recommendation systems. Frontiers in Applied Mathematics and Statistics. 2019:44.
[5] Zhu X, Shi YY, Kim HG, Eom KW. An integrated music recommendation system. IEEE Transactions on Consumer Electronics. 2006 Oct 9;52(3):917-25.
[6] Niyazov A, Mikhailova E, Egorova O. Content-based music recommendation system. In2021 29th Conference of Open Innovations Association (FRUCT) 2021 May 12 (pp. 274-279). IEEE.
[7] Florence SM, Uma M. Emotional Detection and Music Recommendation System based on User Facial Expression. InIOP Conference Series: Materials Science and Engineering 2020 Aug 1 (Vol. 912, No. 6, p. 062007). IOP Publishing.
[8] Le NT. EmuPlayer: Music Recommendation System Based on User Emotion Using Vital-sensor. Bachelor Dissertation. Keio University. 2010.
[9] Celma Ò, Ramírez M, Herrera P. Foafing the Music: A Music Recommendation System based on RSS Feeds and User Preferences. InISMIR 2005 Sep 11 (pp. 464-467).
[10] Saito Y, Itoh T. MusiCube: a visual music recommendation system featuring interactive evolutionary computing. InProceedings of the 2011 Visual Information Communication-International Symposium 2011 Aug 4 (pp. 1-6).
[11] Hu Y, Ogihara M. NextOne Player: A Music Recommendation System Based on User Behavior. InISMIR 2011 Oct (Vol. 11, pp. 103-108).
[12] Fessahaye F, Perez L, Zhan T, Zhang R, Fossier C, Markarian R, Chiu C, Zhan J, Gewali L, Oh P. T-recsys: A novel music recommendation system using deep learning. In2019 IEEE international conference on consumer electronics (ICCE) 2019 Jan 11 (pp. 1-6). IEEE.
[13] Chang YI, Wu CC, Tsai MC. A user-interests approach to music recommendation. InProceedings of the World Congress on Engineering 2011 (Vol. 3).
[14] Chang YI, Wu CC, Tsai MC. A Fair Approach to Music Recommendation Systems Based on Music Data Grouping. IAENG International Journal of Computer Science. 2011 Oct;38(4):418-27.
[15] Kim HG, Kim JH. Music Similarity Search Based on Music Emotion Classification. The Journal of the Acoustical Society of Korea. 2007;26(3E):69-73.
[16] Liu NH, Lai SW, Chen CY, Hsieh SJ. Adaptive music recommendation based on user behavior in time slot. IJCSNS International Journal of Computer Science and Network Security. 2009 Feb 28;9(2):219-27.
[17] Xing Z, Wang X, Wang Y. Enhancing Collaborative Filtering Music Recommendation by Balancing Exploration and Exploitation. InIsmir 2014 Oct 27 (pp. 445-450).
[18] Shakirova E. Collaborative filtering for music recommender system. In2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) 2017 Feb 1 (pp. 548-550). IEEE.
[19] Su JH, Chang WY, Tseng VS. Effective social content-based collaborative filtering for music recommendation. Intelligent Data Analysis. 2017 Jan 1;21(S1): S195-216.
[20] Hu R, Pu P. Enhancing collaborative filtering systems with personality information. InProceedings of the fifth ACM conference on Recommender systems 2011 Oct 23 (pp. 197-204).
[21] Van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. Advances in neural information processing systems. 2013;26.
[22] Shao B, Wang D, Li T, Ogihara M. Music recommendation based on acoustic features and user access patterns. IEEE Transactions on Audio, Speech, and Language Processing. 2009 Sep 1;17(8):1602-11.
[23] Crisan A, Fisher SE, Gardy JL, Munzner T. GEViTRec: Data Reconnaissance Through Recommendation Using a Domain-Specific Visualization Prevalence Design Space. IEEE Transactions on Visualization and Computer Graphics. 2021 Aug 27.
[24] Sen A, Larson MA. From Sensors to Songs: A Learning-Free Novel Music Recommendation System using Contextual Sensor Data. InLocalRec@ RecSys 2015 Sep 19 (pp. 40-43).
[25] P. Knees and M. Schedl, Music similarity and retrieval: an introduction to audio- and web-based strategies. Berlin: Springer, 2016.
[26] Yazhong Feng and Y Zhuang. Popular Music Retrieval by Detecting Mood. In International Society for Music Information Retrieval 2003, volume 2, pages 375– 376, 2003
[27] Vashishtha S, Susan S. Inferring sentiments from supervised classification of text and speech cues using fuzzy rules. Procedia Computer Science. 2020 Jan 1; 167:1370-9.
[28] 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, June 2005.
[29] G. Li and J. Zhang, "Music personalized recommendation system based on improved KNN algorithm," 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2018, pp. 777-781, doi: 10.1109/IAEAC.2018.8577483.
[30] Liu NH. Comparison of content-based music recommendation using different distance estimation methods. Applied intelligence. 2013 Mar;38(2):160-74.

[31] Qing Li, Byeong Man Kim, Dong Hai Guan, and Duk Oh. A Music Recommender Based on Audio Features. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pages 532–533, Sheffield, United Kingdom, 2004. ACM.

[32] M.A. Casey, Remco Veltkamp, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. Content-based Music Information Retrieval: Current Directions and Future Challenges. Proceedings of the IEEE, 96(4):668–696, 2008.

[33] Jean-julien Aucouturier and Francois Pachet. Music Similarity Measures: What is the Use. In Proceedings of the ISMIR, pages 157–163, 2002.

[34] Beth Logan. Music Recommendation from Song Sets. In International Conference on Music Information Retrieval 2004, number October, pages 10–14, Barcelona, Spain, 2004.

[35] Dmitry Bogdanov, J. Serra, Nicolas Wack, Perfecto Herrera, and Xavier Serra. Unifying Low-level and High-level Music Similarity Measures. IEEE Transactions on Multimedia, 13(99):1–1, 2011.

[36] Chun-man Mak, Tan Lee, Suman Senapati, Yu-ting Yeung, and Wang-kong Lam. Similarity Measures for Chinese Pop Music Based on Low-level Audio Signal Attributes. In 11th International Society for Music Information Retrieval Conference, number ISMIR, pages 513–518, 2010.

[37] Dmitry Bogdanov and Perfecto Herrera. How Much Metadata Do We Need in Music Recommendation? A Subjective Evaluation Using Preference Sets. In 12th International Society for Music Information Retrieval Conference, number ISMIR 2011, pages 97–102, 2011.

[38] Pedro Cano, Markus Koppenberger, and Nicolas Wack. An Industrial-strength Content-based Music Recommendation System. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’05, page 673, New York, New York, USA, 2005. ACM Press.

[39] Pedro Cano, Markus Koppenberger, and Nicolas Wack. Content-based Music Audio Recommendation. In Proceedings of the 13th annual ACM international conference on Multimedia, number ACM, pages 211–212, 2005.

[40] Beth Logan. Music Recommendation from Song Sets. In International Conference on Music Information Retrieval 2004, number October, pages 10–14, Barcelona, Spain, 2004.

[41] Terence Magno and Carl Sable. A Comparison of Signal of Signal-Based Music Recommendation to Genre Labels, Collaborative Filtering, Musicological Analysis, Human Recommendation and Random Baseline. In ISMIR 2008: proceedings of the 9th International Conference of Music Information Retrieval, pages 161–166, 2008.

[42] F. Pachet and J.J. Aucouturier. Improving Timbre Similarity: How High is the Sky? Journal of negative results in speech and audio sciences, 1(1):1–13, 2004.

[43] FlexerA, Stevens J. Mutual proximity graphs for improved reachability in music recommendation. Journal of new music research. 2018 Jan 1;47(1):17-28.

[44] Sarin, E., Vashishtha, S., & Kaur, S. (2022, February). SentiSpotMusic: a music recommendation system based on sentiment analysis. In 2021 4th International Conference on Recent Trends in Computer Science and Technology (ICRTCST) (pp. 373-378). IEEE.

[45] Wang, D., Zhang, X., Yin, Y., Yu, D., Xu, G., & Deng, S. (2023). Multi-view enhanced graph attention network for session-based music recommendation. *ACM Transactions on Information Systems*, *42*(1), 1-30.

[46] Liu, Z., Xu, W., Zhang, W., & Jiang, Q. (2023). An emotion-based personalized music recommendation framework for emotion improvement. *Information Processing & Management*, *60*(3), 103256.

[47] Bakariya, B., Singh, A., Singh, H., Raju, P., Rajpoot, R., & Mohbey, K. K. (2024). Facial emotion recognition and music recommendation system using CNN-based deep learning techniques. *Evolving Systems*, *15*(2), 641-658.