

# Music Recommendation System using Content and Collaborative Filtering Methods

Sheela Kathavate

Department of Information Science and Engineering  
BMS Institute of Technology and Management  
Bangalore, India

**Abstract-** Rapid development of mobile devices and internet has made possible for us to access different music resources freely. While the Music industry may favor certain types of music more than others, it is important to understand that there isn't a single human culture on earth that has existed without music. In this paper, we have designed, implemented and analyzed a song recommendation system. We have used Song Dataset provided to find correlations between users and songs and to learn from the previous listening history of users to provide recommendations for songs which users would prefer to listen most. The dataset contains over ten thousand songs and listeners are recommended the best available songs based on the mood, genre, artist and top charts of that year. With an interactive UI we show the listener the top songs that were played the most and top charts of the year. Listener also have the option to select his/her favorite artist and genres on which songs are recommended to them using the dataset.

**Keywords-** Collaborative model, Content based model, Recommendation System, Popularity model

## I.INTRODUCTION

Everyone's taste in music is unique which means that no matter what music you make, someone is bound to enjoy listening to it. While the Music industry may favor certain types of music more than others, it is important to understand that there isn't a single human culture on earth that has existed without music. Music is of great benefit to us, regardless of whether we are renowned recording artists, karaoke singers or merely fans of music. The number of songs available exceeds the listening capacity of single individual. According to the MarsBands.com there are at least 97 million songs. These are only the songs officially released. If we included songs everyone knows or the incredibly old Celtic songs with no names, we would reach 200 million songs since the website most likely does not include Happy Birthday or a nameless song from 1400 AC. This is when we take only the artists who had their name officially on Music charts. Starting there, let's say that there are currently around 1 million songwriters alive that we know about. If we use the same percentage as above, we can guess that there have been about 15.3 million songwriters ever. To get an idea, there are 4 million songs on Spotify that have never been played. In total, there must be billions just there and Spotify itself is by no means the limit of music. What about all the CDs and records made over the past century which have not been digitized? What, indeed, about song passed down the generations in small African communities? There are trillions and trillions of songs in the world, so many that an estimate is impossible, and the potential

more an infinitely greater number which have not yet been made, a world of music for us to enjoy.

Keeping this general idea, one can get that the number of songs are too high for a person, even if listening to music is his or her best hobby. People sometimes feel difficult to choose from millions of songs. Moreover, music service providers need an efficient way to manage songs and help their customers to discover music by giving quality recommendation. This means it not only gives user freedom of selecting the songs he or she wants to listen but also recommends songs according to their previous listening history. Thus, there is a strong need of a good recommendation system. In order to efficiently access, discover, and present music content to the final user, techniques for searching, retrieving, and recommending need to be appropriate for music content. There has been some work done in both academia and the industry to provide music recommendation services. Understanding patterns of music listening and consumption can help to perform accurate and satisfying music recommendations. This paper aims to present the development of music recommendation and discovery methods so far, and identify the issues in evaluation that still require careful consideration and research. Music recommender system is a system which learns from the users past listening history and recommends them songs which they would probably like to hear in future. Currently, there are many music streaming services like Pandora, Spotify, etc. which are working on building high-precision commercial music recommendation systems. These companies generate revenue by helping their customers discover relevant music and charging them for the quality of their recommendation service. Thus, there is a strong thriving market for good music recommendation systems.

In this proposed system, the motive is to build an efficient recommendation system for music and add a friendly user interface for the benefit for the listeners. The goal of a music recommendation system is to help consumers and the music industry with the discovery and delivery of music. In order to realize the personalized distribution of music, it may be beneficial for recommender designers to understand the music listening behaviors and know about the state of music consumption in the industry. Understanding user preference and behavior can help to propose a reasonable recommendation to a specific user. For example, some users show a clear bias towards style when choosing music, while some emphasize timbral similarity. In order to make recommendations respectively to these two types of listeners, the recommender needs to focus on different attributes. Moreover, users' feelings and expressions can be different towards the same music, such

that a personalized user profile is needed for each user before the system can make meaningful recommendations. Generally, a user's preference shifts with time, in terms of years, seasons, days, and even hours. For instance, a user who liked calm and soft music before, may like noisy music now. So, a user's profile needs update and maintenance to describe the music preference of the user at a time. Unlike the consumption of movie, books, and games, people listen to music repeatedly and continuously. This adds more complexity to capture a user's preference accurately, which is important for a music recommendation system. The block diagram of a music recommendation system is as shown in fig. 1.

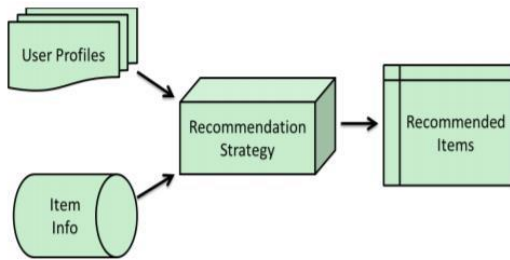


Fig. 1. An illustration of Music Recommendation System

The rest of the paper is organized as follows. In section 2, we explain related work. Section 3 highlights the proposed system. Section 4 discusses the music recommendation application. Section 5 does the result analysis and section 6 concludes the paper.

## II RELATED WORK

Bertin Mahieux T et al. [1] proposed a Million Song Dataset Challenge: a large scale, personalized music recommendation challenge, where the goal is to predict the songs that a user will listen to, given both the user's listening history and full information (including metadata and content analysis) for all songs. They describes the three algorithms used to produce baseline results: a global popularity based recommender with no personalization, a simple recommender which predicts songs by artists already present in the user's taste profile, and finally a latent factor model. All results are based on a train-test split that is similar, but may differ from the split to be used in the contest. The training set consists of the full taste profiles for approximately 1Million users, and partial taste profiles for the 10K test users.

An effective cross-platform music player, EMP, which recommends music based on the real-time mood of the user is proposed by Shlok Gilda et al. [2]. EMP provides smart mood-based music recommendation by incorporating the capabilities of emotion context reasoning within the adaptive music recommendation system. Music player contains three modules: Emotion Module, Music Classification Module and Recommendation Module. The Emotion Module takes an image of the user's face as an input and makes use of deep learning algorithms to identify their mood with an accuracy of 90.23%. The Music Classification Module makes use of audio features to achieve a remarkable result of 97.69% while classifying songs into 4 different mood classes. The Recommendation Module suggests songs to the user by

mapping their emotions to the mood type of the song, taking into consideration the preferences of the user.

In their paper, Miao Jiang et al. [3] propose an improved algorithm based on deep neural network on similarity between different songs. The proposed method makes it possible to make recommendations in a large system to make comparisons by "understanding" the content of songs. This paper proposes a model based on recurrent neural network to predict user's next most possible song by similarity. They conducted experiments and evaluations based on Million Song Dataset and demonstrate how it outperforms the traditional methods. It collected the lyrics for a total of 34412 songs, and audio samples for 4240 songs, resulting in a lyrics dataset consisting of 28,000 pairs, and an audio dataset consisting of 1000 pairs. Cross validations was conducted for both to split the datasets into training and testing sets. The proposed model in the paper is based on a Long-short term memory-based architecture. The motivation behind using an LSTM-based architecture stems from the fact that audio is inherently sequential in nature and the similarity between two songs (particularly between their audio signals) must in at least some way be determined by the similarities between their sequences over time.

Parmar Darsna [4] proposed a song recommendation system for user to get particular item of his/her interest based on 2 popular algorithms, Content Based Filtering and Collaborative Based Filtering. Content-Based method recommends music based on user data. Content based method music subjective features are Speechiness, Loudness and Acousticness etc. These features are stored in database using k-mean clustering algorithm. Collaborative Based method recommends on the user rating and content sharing between different users. In this method, rating given by user to particular music is considered and find cosine similarity between users. Cold-Start is solved by recommending most popular tracks to new user. The dataset is downloaded from MovieLens Website. It contains 100004 rating and 1296 tag application across 9125 movies. In this model, Spotify API is used to get the songs. In this, any of artist name, if information is available on Spotify should be given and then it will fetch the data related to it.

Dmitry Bogdanov et al. [5] discussed a recommendation system in which the workflow of the implementation of the system can be divided into data gathering, audio analysis, music recommendation, and preference visualization. The user specifies his/her account name on Last.fm and/or SoundCloud services from which the preferred tracks should be retrieved. Canoris 8 API has been used to obtain semantic descriptions. Canoris is a web service developed by the UPF's Music Technology Group 9 for the analysis and synthesis of sound and music. To generate recommendations, an in-house music collection of 50,000 music excerpts, covering a wide range of musical genres was used. This collection was analyzed via the Canoris API to retrieve the same semantic descriptions as used for the preference set. Using this, a set of songs close to the user's preference set is created and presented to the user.

Ms. Nishigandha Karbhari et al. [6] presents a model to placement recommendations System based on marks of student. It uses 3 approaches; collaborative recommendation approach, content recommendation approach and hybrid recommendation approach. They present a method to consider the diverse needs with varying level of competence.

Categorizing students based on their credentials thereafter, it discovers best solutions to generate recommendations for placement based on the marks and various other factors included in the profile of the student. Using these soft computing techniques, the student can be referred to the job profile which is not used as reference for the placement otherwise.

Markus Schedl [7] focused on the group of classic music listeners and investigated a wide range of recommendation approaches and variants for the task of music artist recommendation. Analyzation is done in stand-alone and hybrid recommendation approaches. Each user  $u$  has a listening profile  $L_u$ , which contains all items (artists) listened to. The standalone models are popularity based, collaborative filtering, content based and random based while hybrid model is a fusion of one or more Single model. The listeners are divided into groups based on their age, country and time of the day they prefer listening to music. The full dataset covers almost 200 million listening events by about 16,500 Last.fm users, who listen to more than 1 million unique artists. Since the work at hand focuses on fans of classical music, it yielded a set of 362 listeners. After performing five experiments per user, it was found that random based was least precise, hybrid model was more precise than single based PB model was best in group of teenagers and overall hybrid of CF and IB gave best results.

Kunhui Lin et al. [8] proposed the use of improved user-based collaborative filtering algorithm to deal with the user's long-term preferences. Then, according to the user-tag-music relationships, getting the music that associated with the user via recommendation algorithm based on bipartite graph is done. For music personalized recommendation, the commonly used methods include content based recommendation technology, the collaborative filtering recommendation technology and hybrid recommendation technology, where hybrid is the combination of the two. The content based is on generating playlist based on the user's favorite music while the collaborative based in where each music comes with a tag, thereby recommending similar tag music to the user. There is use of k-means clustering algorithm to cluster users to fill user-music matrix, finding user of similar music taste. The improved recommendation algorithm based on bipartite graph mainly uses information of user-tag two-dimensional relationship and tag-music two dimensional relationship. After testing it was found that improved personalized music recommendation system was most accurate rather than user based collaborative model or recommendation based on bipartite graph.

Ms. M. Sunitha et al. [9] developed an Android application for music recommendations. The application allows the user to select and listen to the songs available in their device. Whenever a user listens to a particular song, a log is created, consisting of certain fields which identify the song. The fundamental style employed here is that of collaborative filtering. It is very popular and is being used widely by companies like Amazon, Google, Yahoo, etc. Collaborative filtering methodology tries to find similarity between two users or items. It is independent of the attributes of those entities. Thus, collaborative filtering is a content agnostic approach.

Kunal Shah et al. [10] proposed a wide and diverse variety of techniques for generating recommendations which include collaborative, content based, knowledge based and other

techniques. These methods are blended in hybrid recommenders to improve performance. Collaborative filtering and content-based filtering approaches are extensively used in information filtering application. The hybrid approach used here involves individual implementation of collaborative and content-based methods and aggregation of their predictions to generate recommendations. Integration of some pro characteristics from content based methods into a collaborative approach, integration of some pro characteristics from collaborative approach methods into a content-based approach is used. A generic consolidative model that is the assimilation of both content based and collaborative characteristics is proposed.

### III. PROPOSED SYSTEM

The main objective of this work is to develop an application for music recommendations. The application allows users to select and listen to the songs available in the device. Whenever a user listens to a particular song, a log is created. In order to suggest songs to the users, we use various strategies to implement recommendation engine. The main motive of this Proposed System is extending the capabilities of the traditional recommendation System. Traditional music recommendation systems depend on collaborative filtering or content-based filtering to generate recommendations. Hybrid approaches combine the collaborative filtering and content-based filtering together to leverage the strengths and weaknesses of each approach. User modeling aims to develop a better user profile. Context awareness associates users and items in a specific circumstance such as working or dancing. Tag-based recommendation labels items with users' opinions. Recommendation in the long tail tries to minimize the popularity bias and mostly accompanies collaborative filtering and for content-based filtering ignores item popularity. Recommendation networks introduces some new properties to the recommendation strategies. Playlist generation can be deemed as a variation of top-N recommendations, satisfying the needs specified by users. Group recommendation involves some pre or post processing by either aggregating multiple user preferences into a unit user profile or uniting separate recommendation results into one recommendation list. A detailed model for music recommendation is as shown in fig. 2.

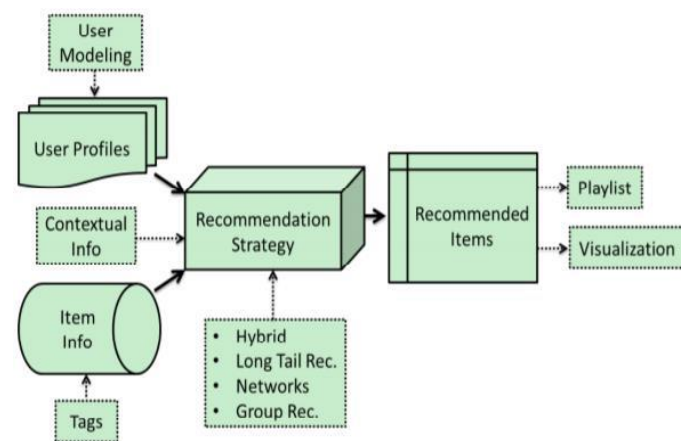


Fig. 2. A Detailed Model of Music Recommendation System

The system is divided into three modules:

#### A. Recommendation Module:

Recommendation module generates recommendation based on the user profile. It analyzes the previous listening history and preferences of a user and provides a list of songs that user might prefer to listen. We have used a global popularity model, Content based model and collaborative filtering model.

#### B. File Server Module:

We have implemented the file server using MongoDB, GridFS and NodeJS modules for efficient upload and retrieval of items. Grids for MongoDB provide many advantages over traditional file systems such as: if the file system limits the number of files in a directory, we can use GridFS to store as many files as needed. Information can be accessed from portions of large files without having to load whole files into memory. GridFS can be used to recall sections of files without reading the entire file into memory. We can keep the files and metadata automatically synced and deployed across a number of systems and facilities using GridFS. When using geographically distributed replica sets, MongoDB can distribute files and their metadata automatically to a number of MongoDB instances and facilities.

#### C. Web Application Module:

Web application provides an intuitive user interface to the user and interacts with file server and recommendation module.

### IV. MUSIC RECOMMENDATION APPLICATION

#### A. Popularity Model

It is a basic model which sorts the songs in the training set according to popularity in descending order and recommends most popular songs. This method doesn't take user's preference into account.

#### B. Content Based Model

Content-based filtering methods are based on a description of the item and a profile of the user's preferences. These methods are best suited to situations where there is known data on an item (name, location, description, etc.), but not on the user. Content-based recommenders treat recommendation as a user-specific classification problem and learn a classifier for the user's likes and dislikes based on product features. To create a user profile, the system mostly focuses on two types of information: model of the user's preference and history of the user's interaction with the recommender system.

We implemented a K-Nearest Neighbor model to recommend songs based on song metadata. First, we created space of songs based on different features in the metadata (artist, genre, etc.) and then to recommend similar song. We select k nearest neighbors of the songs present in the user's profile. Ball Tree based nearest neighbor algorithm is used to address the computational inefficiencies of the brute-force approach. Ball tree algorithm partitions data in a series of nesting hyper-spheres that results in a data structure which can be very efficient on highly structured data, even in very high dimensions.

A ball tree recursively divides the data into nodes defined by a centroid C and radius r, such that each point in the node

lies within the hyper-sphere defined by r and C. The number of candidate points for a neighbor search is reduced through use of the triangle inequality. With this setup, a single distance calculation between a test point and the centroid is sufficient to determine a lower and upper bound on the distance to all points within the node.

#### C. Collaborative Filtering Model

Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past. The system generates recommendations using only information about rating profiles for different users or items. By locating peer users/items with a rating history similar to the current user or item, it generates recommendations using this neighborhood. We have implemented item based collaborative filtering model. Listen count parameter is used as implicit feedback for training. To calculate similarity between two items, we look into the set of items the target user has rated and compute how similar they are to the target item i and then select K most similar items. Similarity between two items is calculated by taking the ratings of the users who have rated both the items and thereafter using the cosine similarity function as in (1).

$$s(i; u) = \mu_i + \frac{\sum_{j \in I_u} (r_{uj} - \mu_j) w_{ij}}{\sum_{j \in I_u} |w_{ij}|} \quad (1)$$

Once we have the similarity between the items, the prediction is then computed by taking a weighted average of the target user's ratings on these similar items. The formula to calculate rating is very similar to the user based collaborative filtering except the weights are between items instead of between users. We use the current users rating for the item or for other items, instead of other users rating for the current items.

### V. RESULTS

In this experiment, we were able to make a music recommendation system using a hybrid approach of collaborative and content filtering. We were able to play and recommend songs in four languages covering more than forty artists. To improvise the system, we asked the users about their preferences and we also provided them with playlists of popular and latest songs. We have tested the system with at least twenty users and the results shown were quite promising. We received an accuracy of 96% on the music recommendation system.

### VI. CONCLUSION AND FUTURE SCOPE

The experimentation is done using twenty artists. In the future, we will try to add a greater number of artists and languages which will make the recommendation stronger giving even better playlists for the users. We can try the system with other machine learning models as well to compare the results and look for better results. When there are millions of songs out there, our motive was to give the users their preference of songs which they want to listen to and we are satisfied after getting one step closer to it. For future applications, an emotional detector system that will recommend the songs by recognizing our facial emotion can be developed.

## REFERENCES

- [1] McFee, B., Bertin Mahieux, T., Ellis, D. P., Lanckriet, G. R. "The million song dataset challenge", In Proceedings of the 21st international conference companion on World Wide Web (pp. 909916) ACM. April 2012.
- [2] Shlok Gilda, Husain Zafar, Chintan Soni, Kshitija Waghurdekar "Smart music player integrating facial emotion recognition and music mood recommendation", International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) 2017.
- [3] Miao Jiang, Ziyi Yang, Chen Zhao, "What to play next? A RNN-based music recommendation system", 51st Asilomar Conference on Signals, Systems, and Computers, 2017.
- [4] Parmar Darsna "Music Recommendation Based on Content and Collaborative Approach & Reducing Cold Start Problem", IEEE 2nd International Conference on Intensive Systems and Control, 2018.
- [5] D. Bogdanov; M. Haro; F. Fuhrmann ; A. Xambo ; E. Gomez ; P. Herrera, "A content-based system for music recommendation and visualization of user preferences working on semantic notions, 9th International Workshop on Content-Based Multimedia Indexing (CBMI), 2011.
- [6] Ms. Nishigandha Karbhari, Prof. Asmita Deshmukh, Dr. Vinayak D. Shinde, "A case study for College Campus Placement" International Conference on Energy, Communication and Data Analytics and Soft Computing (ICECDS-2017).
- [7] Markus Schedl, "Towards Personalizing Classical Music Recommendations", IEEE International Conference on Data Mining Workshop (ICDMW), 2015.
- [8] Kunhui Lin ; Zhentuan Xu ; Jie Liu ; Qingfeng Wu; Yating Chen, "Personalized music recommendation algorithm based on tag information", 7th IEEE International Conference on Software Engineering and Service Science, 2016.
- [9] Ms. M. Sunitha, Dr. T. Adilakshmi, "Mobile Based Music Recommendation System", Dept. of CSE, Vasavi College of Engineering, Hyderabad.
- [10] Kunal Shah, Akshay Kumar Salunke, Saurabh Dongare, Kisandas Antala, "Recommender Systems: An overview of different approaches to recommendations", Dept. Computer Science, Sinhgad Institute of Technology, Lonavala, India International Conference on Innovations in Information Embedded and Communication Systems (ICIIECS), 2017.