



A Personalized and Context Aware Music Recommendation System

Champika H. P. D. Wishwanath, Supuni N. Weerasinghe,
Kanishka H. Illandara, A. S. T. M. R. D. S. Kadigamuwa,
and Supunmali Ahangama^(✉)

Faculty of Information Technology, University of Moratuwa,
Moratuwa, Sri Lanka
champikawishwanath@gmail.com,
supuni.l4@itfac.mrt.ac.lk,
kanishkaillandara@gmail.com, dskadiya@gmail.com,
supunmali@uom.lk

Abstract. People listen to music as it expresses and induce emotions. Considering millions of songs available in online streaming services, it is difficult to identify the most suitable song for an emotion of a user. Most of the music recommendation systems available today, are based on user ratings and acoustic features of the songs. Those systems are unable to address the cold start problem and rating diversity. Furthermore, song preferences will be altering based on the current mood of the users. If these problems were not addressed, then these online services will fail to achieve user satisfaction. To cope with those problems, this paper proposes a novel music recommendation approach that utilizes social media content such as posts, comments, interactions, etc. and recommend them with the most relevant songs to relax their mind considering the current mood (happy, sad, calm and angry). The current mood and the preferences are extracted from the social media profile of the users. Then the songs were classified to moods based on the lyrics and audio of the songs.

Keywords: Music recommendation · Social media · Personalized recommendation

1 Introduction

With the hectic schedule in the day to day life, everyone lives in different states of moods at different times of the day. These moods will alternate time to time. Based on the individual mood and preference, people tend to listen to different types of music [1] to change their mood, relax their minds and improve their existing pleasant feelings. Due these “emotional powers” of music, people have been attracted to music [2]. Music can have a great impact on both the emotions and the human body. For an instance, fast tempo music can make a person more alert while slow tempo music can calm and relax the mind and body. Moreover, music can be effectively used for relaxation and stress management. People will listen to music that suits their individual preference and situation. It is suggested that different individuals perceive different emotions from the

same piece of music [2]. Therefore, it is important to determine individual listening patterns and preferences.

As Social Network Sites (SNS) and streaming services had become the distribution channel of digital music, millions of music tracks are accessible to people. For example, more than 50 million songs are available in Apple Music stores [3]. Thus, it is merely impossible for a user to select the most suitable songs and it would take many hours to decide the most appropriate song for the current mood. To select the user preferred songs with less time, automated music recommendation systems (MRS) are required and MRS are extensively used by online music streaming services like Spotify. To suggest songs based on the user's emotion/mood, it is necessary to classify songs suitable for different moods. However, most of the existing playlists categories the songs based on the public opinion or meta data of the songs. In view of music recommendation, these states of moods should be significant for building proper user preferences to choose suitable music. Music recommendations are extensively used by online music streaming services like Spotify. Most of the recommendation systems consider content-based information [4] and context-based information [5, 6]. However, they do not provide personalized recommendations based on user mood. As such, in this study, the authors have developed a personalized recommendation system based on current mood of the user.

This paper proposes a MRS, that uses song features (lyrics and audio) to classify songs to four types of moods (as happy, sad, calm and angry) and recommends songs based on the current mood of the person. In the proposed MRS, there will be a music player and a playlist of recommended songs. Moreover, the user listening patterns will be captured to improve the recommendations further. The current mood of a person is captured from the individual profiles in SNS. SNS has become an essential part of life of individuals in different age levels and geographic areas. Among them Facebook has unique features which created virtual communication platform open to anyone to express their opinions and feelings. Thus, this study is based on the Facebook user data. Most of the SNS users freely share their own opinions through his/her social media account. From user's profile in Facebook, it is possible to extract attributes such as person's mood, activities, thinking style, and interactions. Moreover, music listening preferences of users can be identified by analyzing the user profile. Thus, it is possible to get a complete idea of the user's natural behavior.

The primary motivation for researching this area was to provide an easier way for music listeners to queue up songs based on their current moods. Thus, various alternative approaches to recommend mood fixing songs that enrich initial music preference of user based on artist and songs will be evaluated considering various factors. In the proposed recommendation system, there will be a music player and a playlist of recommended songs. Moreover, the user listening patterns will be captured to improve the recommendations further (learning process).

The remainder of the paper is arranged as follows. In the next section, a review of the related literature will be given. Subsequently, the approach followed will be explained along with the results of the evaluation. Finally, the discussion and conclusion are available.

2 Literature Review

Recommendation Systems' role is to provide users with the most relevant items based on their preferences or by their past evaluations and interactions [7, 8]. This information can be acquired explicitly or implicitly [8]. Explicit information is provided by the users themselves. For an instance, giving an opinion, a rating or a like on an item could be considered as explicit information. Implicit information is gathered from the user interaction with the service without users themselves giving the information. For an instance, information such as viewing and playing times are considered as implicit information.

The most used option in music recommendation is collaborative filtering (CF) [9]. Prominent services like Last.fm or iTunes Genius are using CF techniques [10, 11]. In CF, it is important to consider the interaction between users and song items. For an instance, recommendations in Last.fm are computed using user behavior and user generated content (feedbacks, tags) regarding a song. Similarly, Genius by iTunes uses user feedback. However, with the limited user preference data, music recommendations tend to fall short of human expectations. Furthermore, traditional CF approaches do not consider social media data [12].

Nowadays, music recommendation strategies are mostly based on content, that is available as low-level features (audio features like tempo, harmony, pitch, sound level) [13], or high-level features (metadata or social tags) [14] in order to provide recommendations. Rather than using CF, the content-based filtering recommendation (acoustic-based music recommendation) has been able to address the problems like cold start problem, that is, newly released or unpopular music when the music production continuously flowing in an uncontrolled manner. As content-based methods like matrix factorization, neighborhood-based methods, and Markov model are commonly used [15]. For example, Pandora is a popular music services that uses these approaches [11]. These recommender systems have considered audio features and the user ratings. To further improve the performance of the online music services, social media data has been used in addition to acoustic based content of the songs [12]. Moreover, there is a huge potential in using context-aware music recommendations [16]. Specifically, contextual situations like time, weather data, current mood of a person, users' physical location, age, gender and all the other impacting variables should be considered in music recommendation. However, some modern music providers like Spotify, Hungama and YouTube do not fulfil their systems to handle these situations.

In conventional music players, a listener had to manually search the playlist and select songs that would suit his/her mood. Music square can classify the songs manually according to four basic emotions; namely, passionate, calm, joyful and excitement. Today, there have been many advancements in the music players including features like streaming, local playback, fast forward, reverse, and grouping based on genre. Even though, these features might satisfy some of the requirements of the listeners when browsing through the playlists, still it is a manual and time-consuming process. Thus, developing a playlist that automatically recommends a song from his/her playlist, according to his mood and preferences will be useful [17].

Emo player was developed by [17, 18] uses facial emotions of the users and based on the current emotions, the songs are played. However, it is difficult to capture the facial emotions of the users. There are some of the playlists created using special hardware to measure the EEG or by analyzing the user speech. However, they are slow in processing and computationally expensive [18].

3 Approach

Even though there are many popular online streaming services with millions of subscribers, a truly personalized way of listening to music is missing in the music streaming industry. Many services have tried to handle this problem by adding features such as mood stations or a “radio” mode for discovering new music. However, none of these options are dynamically changing along with the user’s mood. This proposed approach aims to solve this problem. As shown in Fig. 1, there are three (3) main parts of the system. They are (1) user profile analysis, (2) song profile analysis and (3) music recommendation. There will be a music player and a playlist of recommended songs as the final output of the proposed MSR. Moreover, user’s listening behavior (number of times the user has listened, context and rating given) were captured to improve the recommendations.

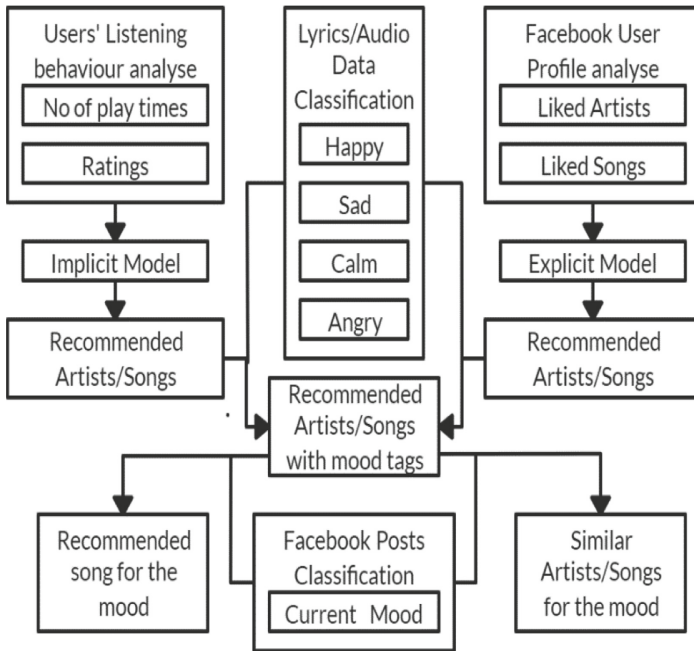


Fig. 1. Overall design of the system

3.1 User Profile Analysis

Current mood of a person is detected using recent (e.g. within a 24-h period) posts, comments, text in images and emoticons shared in Facebook by a user. The current mood will be determined only if there are recent posts in the Facebook profile of the user. After pre-processing the text extracted from text posts and image posts, a mood classification will be performed using 1D CNN Model through Tensorflow (with 85% accuracy). Overall current mood level for a post (if there are text and emoticons) is computed as in (1) using both text content and emoticons extracted from a post of a user [19].

$$\text{Score for a mood in a post} = 0.6 \times \text{emoticon score} + 0.4 \times \text{text content score} \quad (1)$$

If only emoticons or textual content is available, then the (1) would not be used. The individual score obtained would be used as the overall mood score for a specific post.

If there are multiple posts within the time period, a time-based approach is used to calculate the score for the current mood. A weight is added to each post based on the posted time so that the most recent post will get a higher weight compared to other posts.

$$\text{Final Current Mood Score for } M = \sum_{i=1}^n \left((n+1-i) / \sum n^{S_i} \right) \quad (2)$$

In (2) S represents a value obtained for a mood M (i.e. happy, sad, calm or angry) in a specific post i and n represents total number of posts within a 24 h period for a mood M . Thus, a lower weight is given for the oldest post while recent post is given a higher weight.

3.2 Music Profile Analysis

In many datasets, songs are classified by genre and audio features. It is difficult to find a proper dataset which has been classified by moods. For example, the Million Song Dataset contains audio features and meta data (e.g. name of the artist, album, genre, created year, duration, energy level and loudness) for a million of music tracks. However, they do not provide complete track due to copyright issues [20] and do not provide a proper tone to distinguish songs by its mood. The final dataset has been created by using websites Lyrics Wiki [21], Last.fm [22] and NJU-MusicMood-v1.0 (a freely available dataset). To extract features from the lyrics, TF-IDF with N-Gram model was used as it gave a better accuracy compared to Word2Vec model. Then SVM model was used to classify the songs (lyrics) into four moods.

When people listen to music, lyrics is not the only thing that they are concerned. Most of the time when someone listen to a song, the rhythm is the first thing that they will grasp. In a song there are many audio features that affect each mood differently. The features considered for this study are Acousticness (sense to the hearing), Danceability (suitability to dance), Energy (intensity and activity of the music track), Loudness (quality of the sound), Tempo (beats per minute) and Valence (positiveness

of a track, where high valance sound are more positive) [22]. For example, a happy mood can be represented by fast and small tempo, medium-high sound level, small sound level variability, and small timing variability [2]. These features for music tracks were extracted from the developer API for Spotify. The list of songs was obtained from the Million Song Dataset. Random forest was used to predict the mood for each song based on the dataset developed using the low-level features extracted from the music tracks.

Finally, the overall mood for each song is identified based on the moods predicted for lyrics data and audio data. If both lyrics and audio features predict a certain mood that will be considered as the mood of the music track. However, human intervention is used if different moods were predicted as per the lyrics and audio features.

As illustrated in Fig. 1, the recommendation system will map the music tracks to users based on the current mood of the users.

3.3 Music Recommendation

Recommendation systems work by collecting data on the preferences of its users for a set of items. This data can be acquired explicitly or implicitly. Explicit data is provided by the users themselves such as giving an opinion, a rating or a like on an item. For example, a song can be given a rating from 1 to 5; where closer to 1 indicates that the user dislikes the song and closer to 5 indicate that the user like the song very much. However, most of the time users do not rate the songs they listened or purchased. As it is difficult to capture explicit data, implicit data like user behavior or things that the user has consumed would be considered (e.g. number of times a song is played). Implicit data is gathered from the user interaction with the service without users themselves giving the information. In the proposed system we have considered the number of times a user listened a song.

The proposed recommendation system was created considering both CF and content-based filtering approaches and the songs were predicted based on the real time current-mood updates of the users. The proposed system is accessing Last.fm service through its public API. For each artist, name, titles of their most popular tracks, play count of those tracks and a set of tags that describe the artist were accessed from Last.fm. Subsequently, these artist related data were mapped with the user preferred artist list identified through the recommendation system. To identify the user preferences, user's music playing histories or sequences were examined.

We have considered implicit model and explicit model. Implicit model finds similar items and make recommendations to users while explicit model suggests the most appropriate songs and artists for the current mood by analysing the Facebook user profile.

Implicit Model. As the initial step, a matrix was created with number of times a user played a song. Each user will have listened only to a subset of songs. Therefore, for each user, number of times a song is played will be available only for a subset of the songs. Since all the users have not listened to all the songs available, it is not possible to know the entries in this matrix. In a sparse matrix for implicit data, the missing values may represent a user's dislike for a song or song is unknown to the user (but

they might have liked the song if they had known it). Thus, a missing value has a value or a meaning.

With CF, the idea is to approximate the matrix by factorizing it as a product of two matrices: one that describes properties of each user, and one that describes properties of each song. Thus, even if two users have not listened to the same song; still, they can be mapped to each other based on the common properties of the songs. When selecting the two matrices, it is important to reduce the error for the values for user/song pairs where the correct number of times a song is played is known. The Alternating Least Squares (ALS) algorithm was used to accomplish matrix factorization. Initially, user's matrix was filled with values randomly and then iteratively optimizing the value of the songs such that the error is minimized. It holds the song's matrix constant and optimizes the value of the user's matrix. Given a fixed set of user factors (i.e., values in the user's matrix), a known number of plays will be used to find the best values for the song's factors using the least squares optimization. Then "alternate" and can pick the best user factors for given fixed songs factors.

Here, as suggested by Hu et al. [8], the aim is to merge the preference (p) for an item with the confidence (c) for that preference. In this model, the missing values as a negative preference with a low confidence value and existing values as a positive preference with a high confidence value will be considered [8]. The confidence can be calculated using interactions like play count or time spent listening to a song. The preference is taken by considering whether the user listens to a song or not (r). That is, preference (p) would be a binary value of 1 or 0 based on the feedback data (e.g. count of playing a song), r .

$$p_{ui} = \begin{cases} 1, & r_{ui} > 0 \\ 0, & r_{ui} = 0 \end{cases} \quad (3)$$

If the user listens to a song (feedback r is greater than 0), then p is set as 1 or else it is set as 0 as given in (3).

Then the confidence can be calculated as follows:

$$C_{ui} = 1 + \alpha \cdot r_{ui} \quad (4)$$

Where in (4), C is the confidence, r is the feedback (e.g. play count) and α is the linear scaling factor. As per Hu et al. [8], $\alpha = 40$, gives the best results.

If a user has played, viewed or clicked a song more times, then this confidence score will increase. For example, if user listens to a song 100 times there will be a higher confidence value compared to listening only once. If the user has not listened to the song, still the confidence value would be 1.

To find similarities between two items, the equation given below was used.

$$\text{Similarity score} = V \times V_i^T \quad (5)$$

Where in (5), V_i is song item vector, V_i^T is the transpose of song vector. In this study, we have used (5) to find the similarity between artists.

To make recommendations, the equation given below was used.

$$\text{Score} = U_i \times V^T \tag{6}$$

Where in (6), U_i is the User vector and V^T is the transpose of item vector. Thus, recommendation score will be calculated for a specific user for each song item.

To improve the performance, three different implicit ALS functions were considered.

Explicit Model. To make recommendation without the impact of “cold start problem” [23], the liked artists of users available through Facebook user profiles were considered. If a new user joins the proposed recommendation system, the most appropriate songs and artists for them should be suggested. To handle that, the nearest neighbor search method was used to identify artists and songs for these new users. Various algorithms were proposed to improve efficiency and accuracy of nearest neighbor search. Most popular methods that can be used are metric-tree, ball-tree, cover tree, Brute-force, and KD tree. After comparing accuracies for using these popular methods, KD tree was selected as the explicit model algorithm for recommendation.

4 Analysis

Performance is evaluated for the implicit model and the explicit model. The performance was evaluated considering various approaches to identify the most suitable approach to be used in the recommendation system.

4.1 Performance Evaluation of Implicit Model

The evaluation of the model was performed using three different functions for implicit data (Fig. 2).

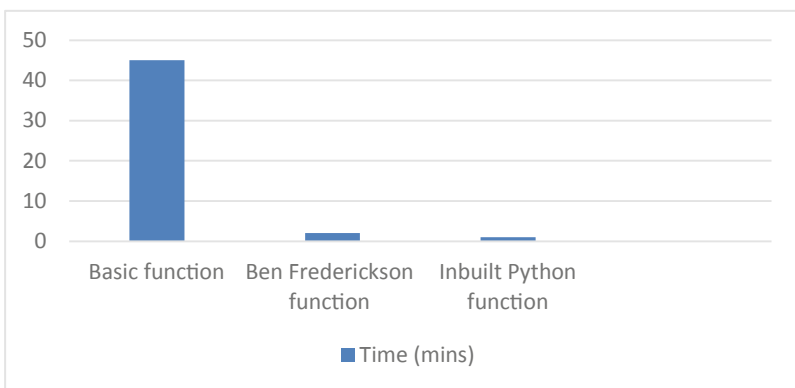


Fig. 2. A comparison of execution time for basic ALS function, Ben Frederickson ALS function and inbuilt python ALS function.

Those implicit ALS functions used are basic implicit ALS function, implicit ALS Ben Frederickson function and inbuilt Python ALS function. As illustrated in Fig. 2, inbuilt Python ALS function performs better than the other two functions based on the execution time. Thus, it was used in the proposed MRS. Therefore, inbuilt python implicit library was used to get the similar artists according to a score. As shown below, higher value of the score hold for the most suitable artists.

Similarity scores are given below.

	artist	score
0	brian wilson	1.141458
1	the beta band	1.130788
2	harry nilsson	1.120744
3	buck 65	1.112741
4	sam cooke	1.111302
5	sly & the family stone	1.110645
6	the onion	1.108948
7	elvis costello & the attractions	1.102296
8	donovan	1.100646
9	npr	1.097487

4.2 Performance Evaluation of Explicit Model

To check the explicit model accuracy, the authors of this paper used various methodologies as indicated in Table 1. Accuracy is compared by changing the neighbor size against the three models. Based on experimental observations, it can be concluded that the KD tree performs faster than the ball-tree when constructing the tree and solving the problem. Moreover, accuracy of the KD tree is comparable to other models. Since KD tree performs well in low dimensional data when compared to other two algorithms and as this system's recommendation module deal with low dimensional data of music, KD tree was selected.

Table 1. Accuracy of the explicit model

No of neighbors	Brute-force	KD tree	Ball tree
45	0.589	0.624	0.610
60	0.688	0.695	0.674
90	0.597	0.577	0.567

5 Discussion

The proposed recommendation system is created considering both CF and content-based filtering approaches. Based on the real time current mood updates of the users, music tracks were predicted. To predict music tracks for a given user, the implicit model with ALS algorithm was used. It was noted that there are many songs that have

only been listened once, as a result, had to compare models that include and exclude those songs that have been listened only once. The model including those songs returns a slightly lower root-mean-square deviation on the test dataset compared to the model excluding them. Since the training set includes more songs, it may be more reflective of their listening profiles.

One of the main challenges in the recommendation model was how to address the songs that have been never heard by a user. To handle this, implicit function with matrix factorization was used, where, unseen items are treated as negative with a low confidence. As the implicit model, the inbuilt python ALS function was used based on the execution time as shown in the Fig. 2.

For any recommendation system, another problem is providing a recommendation when there is a new user. When a new user enters the music recommendation system, it should recommend those users with the most appropriate songs according to their current mood. In order to face this problem, the proposed system used Facebook user profile data including liked artists based on their preferences. With this option, the system can generate a personalized playlist based on the mood of a user. Based on the evaluation results, KD tree with nearest neighbour search algorithm was used.

Since a music player needs to perform as a real time application, the proposed model should actively run on every state of the system. As such, algorithms that execute instantly were selected.

As future work to improve the results of the recommendation system, it is important to develop a context aware recommendation system. For example, need to consider the time of the day as user preferences changes with the time of the day and based on the weather user moods will change. Through social media it is important to gather more details about the user personalities, thus, it will improve the recommendations further. In addition, through other online streaming services like YouTube history and online playlist logs.

6 Conclusion

The proposed recommendation system will be considering various factors in developing a personalized MSR. Music tracks were categorized to types of user moods based on the lyrics and acoustic features of the music tracks and current mood of the users were captured through the image and textual posts in Facebook. Subsequently, songs were recommended to users based on the user preferences and their current mood. Based on the accuracies and execution time, inbuilt Python ALS function and KD tree were used in the proposed MSR.

Acknowledgements. This research was supported by the Faculty of Information Technology, University of Moratuwa and the Senate Research Committee, University of Moratuwa.

References

1. Kim, J.-Y., Belkin, N.J.: Categories of music description and search terms and phrases used by non-music experts. In: Proceedings of the 3rd International Society for Music Information Retrieval Conference (ISMIR 2002), pp. 209–214 (2002)
2. Juslin, P.N., Laukka, P.: Expression, perception, and induction of musical emotions: a review and a questionnaire study of everyday listening. *J. New Music Res.* **33**(3), 217–238 (2004). <https://doi.org/10.1080/0929821042000317813>
3. Apple Music - Apple. <https://www.apple.com/lae/apple-music/>. Accessed 29 May 2019
4. Wang, X., Wang, Y.: Improving content-based and hybrid music recommendation using deep learning. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 627–636 (2014)
5. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 811–820 (2010)
6. Forbes, P., Zhu, M.: Content-boosted matrix factorization for recommender systems In: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys 2011, p. 261 (2011)
7. Park, M.-H., Hong, J.-H., Cho, S.-B.: Location-based recommendation system using bayesian user's preference model in mobile devices. In: Indulska, J., Ma, J., Yang, Laurence T., Ungerer, T., Cao, J. (eds.) UIC 2007. LNCS, vol. 4611, pp. 1130–1139. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73549-6_110
8. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: 2008 Eighth IEEE International Conference on Data Mining, Pisa, pp. 263–272 (2008)
9. Li, Q., Myaeng, S.H., Kim, B.M.: A probabilistic music recommender considering user opinions and audio features. *Inf. Process. Manag.* **43**(2), 473–487 (2007)
10. Chen, Y.X., Boring, S., Butz, A.: How Last.fm illustrates the musical world: user behavior and relevant user-generated content. In: CEUR Workshop Proceedings, vol. 565 (2010)
11. Sunitha, M., Adilakshmi, T.: Music recommendation system based on unsupervised discretization. *Int. J. Comput. Appl.* **145**(7), 22–25 (2016)
12. Bu, J., et al.: Music recommendation by unified hypergraph: combining social media information and music content. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 391–400 (2010)
13. Logan, B.: Music recommendation from song sets. In: Proceedings of the 5th ISMIR Conference, pp. 425–428 (2004)
14. Nanopoulos, A., Rafailidis, D., Symeonidis, P., Manolopoulos, Y.: Musicbox: personalized music recommendation based on cubic analysis of social tags. *IEEE Trans. Audio Speech Lang. Process.* **18**(2), 407–412 (2009)
15. Wu, X., et al.: Personalized next-song recommendation in online karaokes. In: RecSys 2013, pp. 137–140 (2013)
16. Schedl, M., Knees, P., Gouyon, F.: New paths in music recommender systems research. In: RecSys 2017, pp. 392–393 (2017)
17. Hemanth, P., Adarsh, Aswani, C.B., Ajith, P., Kumar, V.A.: EMO PLAYER: emotion based music player. *Int. Res. J. Eng. Technol.* **5**(4), 4822–4827 (2018)
18. Nathan, K.S., Arun, M., Kannan, M.S.: EMOSIC—an emotion based music player for Android. In: 2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Bilbao, pp. 371–376 (2017)

19. Weerasinghe, S., Ahangama, S.: A method to identify the current mood of social media users. In: 2019 IEEE International Conference on Industrial and Information Systems, ICIIS 2019, pp. 1–6 (2019)
20. Million Song Dataset: Code | Million Song Dataset. <https://labrosa.ee.columbia.edu/millionsong/pages/code#datacreation>. Accessed 19 Sept 2019
21. Welcome to Lyrics Wiki - Lyrics Wiki. <http://lyricswiki.com/>. Accessed 19 Sept 2019
22. Last.fm: Last.fm | Play music, find songs, and discover artists
23. Lam, X.N., Vu, T., Le, T.D., Duong, A.D.: Addressing cold-start problem in recommendation systems. In: Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, ICUIMC 2008, p. 208 (2008)