Research Inventy: International Journal of Engineering And Science
Vol.14, Issue 1 (January 2024), PP 55-59
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Personalized Music Recommendation based on mood detection using content based approach

Laksha Wadhwani[1] ,Meghna Chhanwal[2] ,Dr. Makarand Velankar[3]
[1,2,3]*MKSSS Cummins College of Engineering for Women, Pune*

Abstract:
This study explores the fusion of emotion, music, and technology to refine personalized music recommendations based on user moods. Leveraging machine learning algorithms, the research consists of how to accurately determine the top recommendations, by identifying similarity and patterns. The goal is to create an adaptive recommendation system that accurately aligns with users' current moods. S In addition, this research addresses the limitations of conventional static approaches in accommodating mood transition, proposing a dynamic content-based music recommendation system capable of real-time adjustment to users' evolving preferences.
The research also investigates the psychological aspects of music, considering how genres, tempos, and more audio features are classified to distinct emotional states. By integrating emotion into recommendation systems, this research aims to enhance user satisfaction, listening experience and mood.

Keywords: *Music recommendation system, content-based filtering, Spotify.*

Date of Submission: 29-12-2023 Date of acceptance: 08-01-2024

I. INTRODUCTION

Music serves as a facilitator, a source of joy, and a mood enhancer, permeating various aspects of life, from cinematic experiences to live events and gatherings. Individuals engage with music in diverse scenarios, be it celebratory occasions like birthdays, social events, or even in contemplative places like temples. The myriad of songs, spanning various genres and crafted for different contexts, ensures a rich tapestry of musical expression. In the course of daily life, people instinctively tune in to songs aligned with their emotional states, categorizing each composition based on its distinct characteristics. When grappling with anxiety or stress, there's a preference for mellower tunes with subdued beats, providing solace and gradually uplifting one's spirits. The routine involves seeking out these musical preferences on popular streaming platforms such as Spotify, Gaana, or Youtube. However, the repetitive act of searching for each song can become time-consuming and monotonous.

This paper delves into a method to enhance the user's music listening experience. Major streaming platforms adopt a hybrid approach, combining collaborative filtering and content-based filtering algorithms to offer precise suggestions. These recommendation systems play a pivotal role in retaining users' interest and engagement, as evidenced by the success of platforms like Spotify, which incorporates features like Discover Weekly. In our initiative, we designed a system prompting users to select a mood from five given options, followed by tailored music recommendations. While our project currently operates on a smaller scale, limited to our devices, we conducted analyses based on our Spotify account details. Due to privacy constraints imposed by Spotify, we opted for a content-based filtering algorithm. This decision was also influenced by the impracticality of requiring users to register on the Spotify Developer Dashboard, restricting access for everyone visiting our website. Our methodology involves retrieving listening history and saved playlists using Spotify credentials, creating a user profile based on preferences and interactions. For each track, essential features and metadata are extracted. The algorithm then employs mathematical equations to compare the user profile with the extracted features, identifying songs that align with the user's preferences. The system subsequently recommends the top N matching scores to enhance the user's musical journey.

II. EXPERIMENTAL PROCEDURE

Beginning from the scratch, this first crucial step that lets us perform the recommendation task is to implement the Spotify API OAuth procedure. It is an authorization step, where users must have an account already registered with Spotify and accordingly login with their credentials (Email and password). The below diagram summarizes the working of the OAuth procedure, additionally it defines four grant types to request and get an access token:

1.Authorization code.
2.Authorization Code with PKCE extension.

3.Client Credentials.
4.Implicit grant.

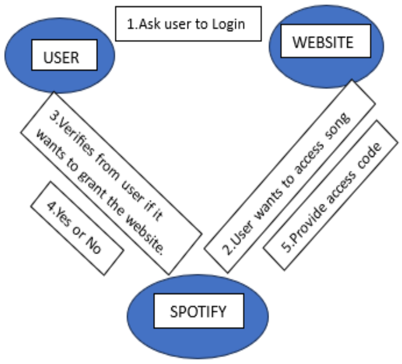

Figure 1: OAuth working.

First and second option require an access token, while the third and fourth option do not. After that, we must define a scope indicating what permissions we need from the user's metadata, in our case it is "user-read-recently-played", "user=library-read", "playlist-read-private". A list of parameters is declared, data we need to pass to the call we are making to Spotify. It contains the client id, client secret key and redirect Uri and the scope. We are making a get request to an Authorization URL which we declared and a request library in Python is used, then Spotify redirects to an URL which is equivalent to a get request and the parameters are passed. The code received is used to make an access token, it is unique to every user. The access expires every hour.
After the user logs in successfully, the user's listening history and saved playlists are retrieved.
The system analyzes the features, which are: genre, artist name, tempo, mood, instruments, key, liveness, speechiness, valence, energy, popularity, acousticness, loudness and danceability. Then all the extracted features are normalized, as it is essential for calculating the similarity between them. All the values are scaled between 0 and 1 or from -1 to 1, covering all the different types of measures of the features. Otherwise, one magnitude may dominate a smaller one, and may lead to incorrect similarity calculation.

One way to standardize these values are using the Min-Max Scaling with the formula given below (Eq. 1):

$$\frac{X - Xmin}{Xmax - Xmin} \qquad (1)$$

Where the min and max values are the minimum range and maximum range of the feature. Next step is to calculate the similarity between them, for that there are many mathematical measures, the most popular is the Cosine Similarity, for which we need to create corresponding vectors for each song, containing all the essential audio features.
Next step is crucial, which is measuring the similarity between the vectors suing the below equation (Eq. 2):

$$\frac{A \cdot B}{||A|| \cdot ||B||} \qquad (2)$$

This step, involves dot product between the vectors, for an instance let us take an example:
Track 1: Tempo: 150 Beats per minute, duration: 200 seconds.
Track 2: Tempo: 120 beats per minute, duration: 175 seconds.
User's choice: Tempo: 170 beats per minute, duration: 150 seconds.
To get the similarity, we apply the equation first with Track 1 and user's choice, secondly with Track 2 and user's choice. For simplicity, a small calculation is shown below to demonstrate it using the equation shown above:
Cosine Similarity (Track 1, User)
= (150x170) +(200x150) (1702+1502(1502 +2002)
=55,000/56629
=0.9809.
This similarity score conveys that there is a high possibility that user will like track 1.
Similarly, for Track 2 and User:
Cosine Similarity (Track 2, User)
= (120x170)+(175x150)(1202+1752(1702 +1502)
=46650/48615
=0.9676
Comparing both values, we can observe that Cosine similarity of track 1 is greater than that of track 2, so track 2 would come in top N recommendations. Similarly, calculation for all the user's saved tracks/playlist and un

interacted tracks of the user's, top 15 recommendations would be displayed based on the ranking of the cosine similarity scores.
As shown below:

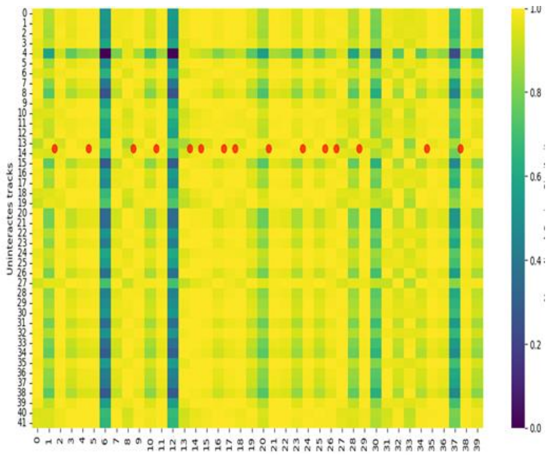

Figure II. Cosine similarities

On the x label it shows the user's preferred tracks, and on the y label it shows the tracks the user has not interacted with. And the 15 red dots represent the recommendation based on highest scores.
The numerical output is as below:
Top 15 Recommended Tracks:
Rank 1: Track 30 - Cosine Score: 0.9877
Rank 2: Track 16 -Cosine Score: 0.9871
Rank 3: Track 10 - Cosine Score: 0.9861
Rank 4: Track 19 - Cosine Score: 0.9860
Rank 5: Track 36 - Cosine Score: 0.9859
Rank 6: Track 39 - Cosine Score: 0.9856
Rank 7: Track 25 - Cosine Score: 0.9856
Rank 8: Track 3 - Cosine Score: 0.9853
Rank 9: Track 15 - Cosine Score: 0.9853
Rank 10: Track 18 - Cosine Score: 0.9843
Rank 11: Track 22 - Cosine Score: 0.9840
Rank 12: Track 28 - Cosine Score: 0.9815
Rank 13: Track 12 - Cosine Score: 0.9815
Rank 14: Track 6 - Cosine Score: 0.9812
Rank 15: Track 27 -Cosine Score: 0.9807
This was the fundamental approach to complete the work and give accurate recommendations, which is for the five moods taken: angry, sad, neutral, happy, and surprised. One way to know the user engagement is to keep asking for user's feedback, let us take the case where the user chooses a mood as sad/angry with tracks with lower valence, lower energy and slower tempo, our goal is to uplift the mood to neutral first, then gradually to happy. For that, sad songs would be recommended, users would be asked to give feedback. If the feedback is good for tracks with a bit higher valence and higher energy, then the user's profile would be adjusted again. And accordingly, tracks will be recommended. The parameter adjustment factor is used to adjust the number of times we want the user's profile to be influenced by their feedback. It determines the magnitude of the change in the user's profile based on their feedback rating. If this parameter is high, it indicates that the user's profile will change more significantly with each feedback, leading to faster adaptation to user preferences. Otherwise, the system will get less influenced by the individual feedback. The values of the audio features will keep updating each time receiving feedback. Let us observe the below detailed example:
Given the audio features for a track classified as sad/angry, and the adjustment parameter set as 0.1, the next step is to update these audio features with the below formula:

new_valence +=adjustment_factor * feedback_rating.

Same goes for all the audio features, by adding a portion of the feedback rating scaled by adjustment_factor. The multiplication by feedback_rating ensures that the change is proportional to the user's feedback, and the direction of the change (positive or negative) depends on the sign of the feedback_rating. When the user gives a low rating, the adjustments to their profile features will be relatively small. This design choice assumes that a low rating indicates a weaker preference or liking for the track, and therefore the impact on the user's profile should be limited.

The audio features used for the algorithm to give recommendations are as follows:
1.Acousticness: Measures the amount of natural and live instruments in the track.
2.Danceability: Measures the rhythm and beat strength of the track.
3.Energy: measures the intensity and activity of a track.
4.Instrumentalness: Measures the number of vocals used in the track, if less means more vocals are used than instruments.
5.Liveness: Amount of live audience present in the track.
6.Loudness: Volume of a track.
7.Speechiness: Measures the number of spoken words in the track.
8.Tempo: Measures the pace of a track, in beats per minute.
9.Valence: Measures the amount of positiveness in a track, lower values suggest sadness.
For different moods there are distinct values of the features extracted. If the user's mood is detected as happy, then along with the cosine similarity these parameters are set like this:
"target_popularity: 80,
"target_danceability": 0.7,
"target_energy": 0.65,
"target_valence": 0.6,
"target_acousticness": 0.15.
Another example, for neutral mood:
"target_danceability": 0.7,
"target_energy": 0.5,
"target_valence": 0.45,
"target_acousticness": 0.65.
"target_instrumentalness": 0

As observed, some values such as energy and valence are decreased for neutral mood, and some are increased such as acousticness. Same approach is applied for all the five moods. Along with that, there are some more filters, such as the different genres. For example, the genres used in our recommendations are Bollywood dance, Bollywood romantic, Bollywood romantic sad, etc...

III. RESULTS AND DISCUSSIONS

The computed cosine similarity matrix, which reflects the intrinsic features of liked and disliked songs, predominantly reveals elevated similarity scores, hovering mostly around 0.999 but with slight variations in decimals. Utilizing the cosine similarity matrix, the system identifies the top 15 recommended tracks. However, the heightened uniformity in similarity scores gives rise to concerns about the diversity and meaningfulness of these suggestions.

The system endeavors to address mood transitions by dynamically adapting the user profile. When detecting a user's melancholic or irate mood, it strives to shift towards more euphoric content by flexibly adjusting the user profile's valence, energy, and tempo based on user feedback.

In the realm of music recommendation, precision can be seen as the ratio of correctly recommended tracks to the total number of recommended tracks, while recall is the ratio of correctly recommended tracks to the total number of relevant tracks. Precision is the ratio of true positives to the total number of predicted positives, and recall is the ratio of true positives to the total number of actual positives. Precision and recall are calculated for the implemented content-based recommendation algorithm:

True Positive (TP) is defined as songs that are liked and correctly predicted as liked, and False Positive (FP) is defined as songs that are not liked but incorrectly predicted as liked. True Negative (TN) comprises songs that are not liked and correctly predicted as not liked, while False Negative (FN) is defined as songs that are liked but incorrectly predicted as not liked.

True Positives(TP) = 37
False Positives(FP) = 35
False Negatives(FN) = 0

Precision = TP/(TP + FP)
= 37/ (37 + 35)
= 0.514

Recall = T{/(TP + FN)
= 37 / (37 + 0)

=1.0

The precision value of approximately 0.514 indicates that around 51.4% of the predicted liked songs were actually liked. The recall value of 1.0 indicates that all liked songs were correctly identified. Conducting a literature survey is crucial for understanding existing research on dynamic content-based music recommendation systems. This survey should explore subjects such as the limitations of static approaches, the significance of mood adaptation, and the role of content-based filtering in addressing evolving user preferences. Future consideration may involve refining the visualization of similarity scores, exploring alternative recommendation algorithms, and conducting user studies to validate the effectiveness of the mood transition and profile adjustment mechanism.

IV. CONCLUSION

The problem statement highlighted the limitations of conventional static approaches and emphasized the need for a dynamic system that could seamlessly adapt to users' changing preferences, specially during mood transitions. The significance of real-time adjustments to provide a more personalized and engaging user experience was a driving force behind the system's development. Looking ahead, prospective considerations involve refining the visualization of similarity scores, exploring alternative recommendation algorithms, and conducting user studies to validate the effectiveness of the mood transition and profile adjustment mechanisms. Overcoming challenges, such as the considerable uniformity in cosine similarity scores, remain a priority to enhance the system's recommendation capabilities.

REFERENCES

[1]. Makarand, V., & Sahasrabuddhe, H. V. (2014, January). Novel approach for music search using music contents and human perception. In 2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies (pp. 1-6). IEEE.

[2]. Velankar, M., & Kulkarni, P. (2022). Music recommendation systems: overview and challenges. Advances in Speech and Music Technology: Computational Aspects and Applications, 51-69.

[3]. Velankar, M., & Kulkarni, P. (2023). Employing cumulative rewards based reinforcement machine learning for personalized music recommendation. Multimedia Tools and Applications, 1-14.

[4]. Shetty, R., Kasbe, S., Jorwekar, K., Kamble, D., & Velankar, M. (2015). Study of Emotion Detection in Tunes Using Machine Learning. International Journal of Scientific and Research Publications, 5(11).

[5]. Velankar, M., Kotian, R., & Kulkarni, P. (2021). Contextual mood analysis with knowledge graph representation for Hindi song lyrics in Devanagari script. arXiv preprint arXiv:2108.06947.

[6]. Shivananda, S., Dutt, R., Perumal, B., H, A., & Latha, A. P. (2022, July). Mood-Based Music Recommendation System: VIBY. International Research Journal of Modernization in Engineering Technology and Science, 4(7), 1158.

[7]. Teckchandani, H., Patel, K., & Enbibel, M. (2022). Smart Automatic Song Playlist Generation using Facial Expression. *ResearchGate*.

[8]. Mahadik, A., & Milgir, S. (2021, June). Mood-based music recommendation system. Don Bosco Institute of Technology.

[9]. singh, A.,shoaib,M.,singh,s.,&sharma,S.(2023).Analysis of Real-Time Emotion Detection Techniques.Available at SSRN 4623338