Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 2, No. 2, February : 2024

MUSICONN-DISCOVER PEOPLE AND MUSIC USING SPOTIFY AND ML TECHNIQUES

N. Sabitha, K. Murali Krishna, Vinuthna Ambati, Reshwanth Vadala, Shreshta Amireddy,
Dept. of Computer Science & Engineering Maturi Venkata Subba Rao (MVSR) Engineering College, Osmania University Hyderabad, India sabitha_cse@mvsrec.edu.in

I. ABSTRACT

Humans have an inherent need to interact with others to gather information and ensure their survival. However, initiating conversations with new people can be challenging for many individuals. One intriguing approach to initiating a conversation is by using music as a common ground. Music holds a universal appeal, as people enjoy listening to songs. To facilitate this connection, we have created a machine learning-powered user recommendation and social networking application named "**MusiConn**." MusiConn's primary goal is to simplify the process of connecting individuals who share similar musical preferences, thereby fostering meaningful connections. The application utilizes Spotify's Web API to gather user-related data, including playlists and music listening history. It employs various machine learning techniques such as cosine similarity, collaborative filtering, and matrix factorization to categorize this data. The application's key functionality revolves around connecting users who have similar music tastes, achieved through the analysis of data obtained from Spotify's API.MusiConn offers several other features, including personalized song recommendations, chat rooms, and private chats. Song recommendations are tailored to the user's current song choice, ensuring that they receive pertinent and captivating song suggestions that align with their musical tastes.

II. INTRODUCTION

In the era of social media, people seek to expand their social connections globally. However, the conventional approach of friend suggestions based on shared connections often yields irrelevant recommendations, as users with common connections may differ significantly in interests and habits. This limitation persists across major platforms like Facebook and Twitter, where suggestions rely on factors such as mutual friends, age, and location. This research addresses the inadequacies of existing friend suggestion methods by proposing a novel approach. Our objective is to develop an application that leverages users' music preferences to facilitate more personalized and accurate friend recommendations. By focusing on music taste, we aim to create a social network system that offers nuanced matches, connecting users with others who share similar lifestyles and interests.

III. RELATED WORK

In various studies, different methods have been proposed for friend recommendation in social networking.

a. **Content-based methods:** "Friendbook: A Semantically Based Friend Referral System” [1] uses LDA algorithms and user-centred sensor data to extract user lifestyles and make recommendations based on the similarities between users.

b. **Network-based methods:** "Social Friend Recommendation Based on Multiple Network Correlation” uses an NC-based SFR algorithm to improve recommendation accuracy. [2]"Social recommendation across multiple relational domains" uses hybrid random walk and RDR algorithms to recommend friends.[6]

c. **Personality-based methods:** "Matchmaker" uses collaborative filtering and personality matching to suggest friends.[4]

d. **Context-aware methods:** "A friend recommendation system" uses physical and social contexts to make recommendations.

e. **Genetic algorithm-based methods:** "Recommendation systems to help improve user experience" [3] uses genetic algorithms and network topology to make recommendations.

f. **Trust-based methods:** "A lightweight referral scheme for privacy-conscious friends" uses a trust-based model for privacy.

Other methods have been proposed for friend recommendation using weblog-based analysis, examining network structures, layered friendship models, and preserving privacy in social networks.

In conclusion, various methods have been proposed for friend recommendation in social networks [7], using techniques such as the LDA algorithm, NC Based SFR, hybrid random walk method, personality matching, context-awareness, genetic algorithms and network topology, trust-based referral scheme, mode of transportation inference, and privacy preservation.

IV.PROPOSED SYSTEM

The proposed system utilizes the Spotify API and machine learning techniques to provide personalized music recommendations and artist discovery to users on the Spotify platform. The system accesses users, songs, and other relevant data from Spotify using the Spotify API. The goal of the system is to make it easier for users to discover new people with similar music taste and reduce the time and e ort required to search for new friends.

The system uses two main machine-learning algorithms: collaborative filtering and matrix factorization. Collaborative filtering uses the behaviour and preferences of a group of users to make recommendations to individual users. The algorithm considers the similarities between users and the items they have interacted with (in this case, songs they have listened to) to make predictions about what a user is likely to be interested in. Matrix factorization, on the other hand, factorizes the user-item interaction matrix into lower dimensional matrices, which can be used to make predictions about a user's preferences for items.

V. SYSTEM DESIGN

System Architecture:

The system architecture diagram provided above depicts the overall structure and communication flow of a music-focused Progressive Web App (PWA) that utilizes machine learning (ML) to generate user recommendations based on their music interests. The diagram shows several components that interact with each other to provide the app's functionality. The User, represented as an actor, interacts with the ReactApp, which serves as the user interface for the app. The ReactApp communicates with the Backend component through a REST API, and the Backend retrieves and processes data from several interfaces including Firebase, FlaskAPI, and SpotifyAPI. Firebase is used to store user data, and Flask API is responsible for processing the ML model that generates user recommendations. Spotify API is used to fetch user song data for ML model processing. The diagram also shows that the ReactApp interacts with Firebase to fetch and display user data and chat messages. Additionally, the app includes chat features that allow users to send chat messages to each other, and the ReactApp sends these chat messages to the Backend. Finally, the Backend communicates with the FlaskAPI through webhooks to update the ML model.

The system architecture diagram provides an overview of the different components and interfaces of the system, and how they interact with each other to provide the app's functionality. The diagram can serve as a useful reference for developers and stakeholders to understand the overall structure and communication flow of the system and can aid in the design, implementation, and testing of the app.

The system architecture diagram provided above depicts the overall structure and communication flow of a music-focused Progressive Web App (PWA) that utilizes machine learning (ML) to generate user recommendations based on their music interests. The diagram shows several components that interact with each other to provide the app's functionality. The User, represented as an actor, interacts with the ReactApp, which serves as the user interface for the app. The ReactApp

communicates with the Backend component through a REST API, and the Backend retrieves and processes data from several interfaces including Firebase, FlaskAPI, and SpotifyAPI. Firebase is used to store user data, and FlaskAPI is responsible for processing the ML model that generates user recommendations. Spotify API is used to fetch user song data for ML model processing.

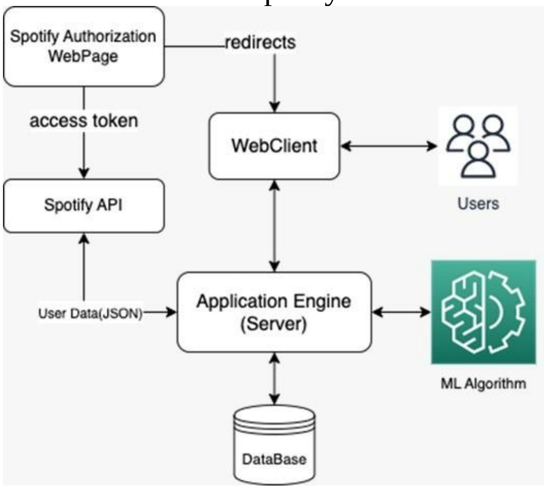

Fig 1. System Architecture

Figure 2. Module Overview

VI.IMPLEMENTATION

Our application is divided into 5 modules:

1. **React Web App:** We built a progressive web application (PWA) using React JS[l11]. It can be viewed as a normal website and can also be downloaded as a native app on our mobiles and desktops. Our website is divided into various components where each representing a feature.
2. **Spotify API:**Spotify API is one of the key requirements of our work. API calls are made to Spotify to get data. To set up Spotify API calls, we require client secret and client id which are available once we register in the Spotify Dashboard. To make API calls, we require an access token and a refresh token. Only with the help of an access token, we can make API calls.
3. **Firebase Databases:** To deploy a web application using Firebase, we first created a Firebase and configured it to work with our React Application.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 2, No. 2, February : 2024

4. **ML User Recommendation:** We used Machine Learning for recommendations. Users ’data such as their playlist, followed artists, saved tracks, genre etc., are fed to our machine learning model which in return recommends users with similar music interests. We initially tested the recommendations using cosine similarity and then improvised the recommendations by using matrix factorization.

5. **Chat and Security:** Chat is divided into 2 categories: Chat Room and Personal Chat. People listening to the same song can join the chat room and chat with other people inside the chat.
Users can have personal chat with others by going to the Chat List page.

VII.RESULTS

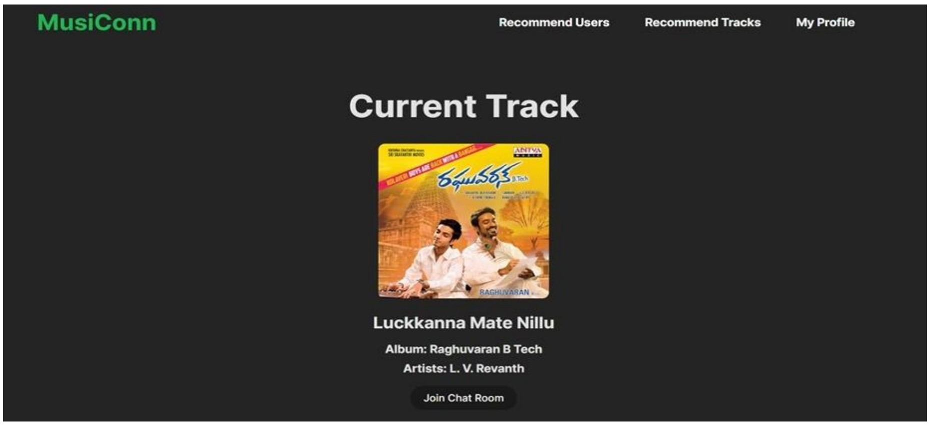

Figure 3: Page Showing Current Track and all its Features

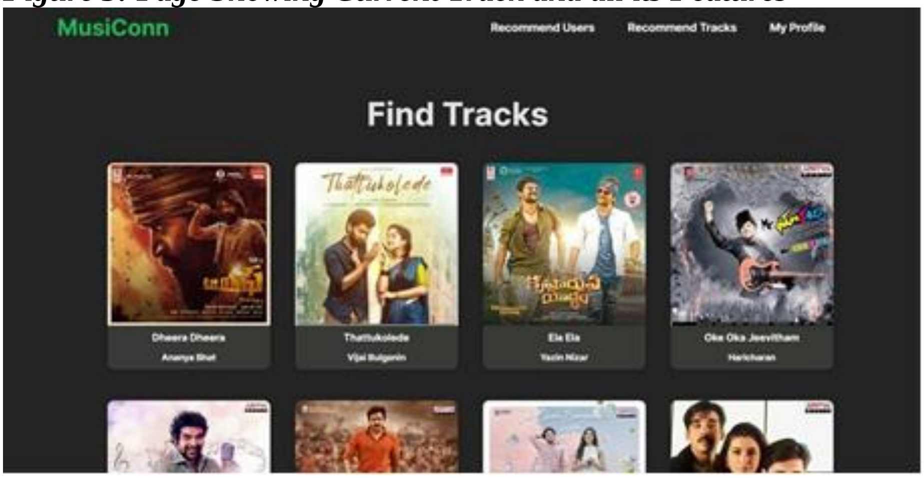

Figure 4: Page showing Song recommendations for current playing track

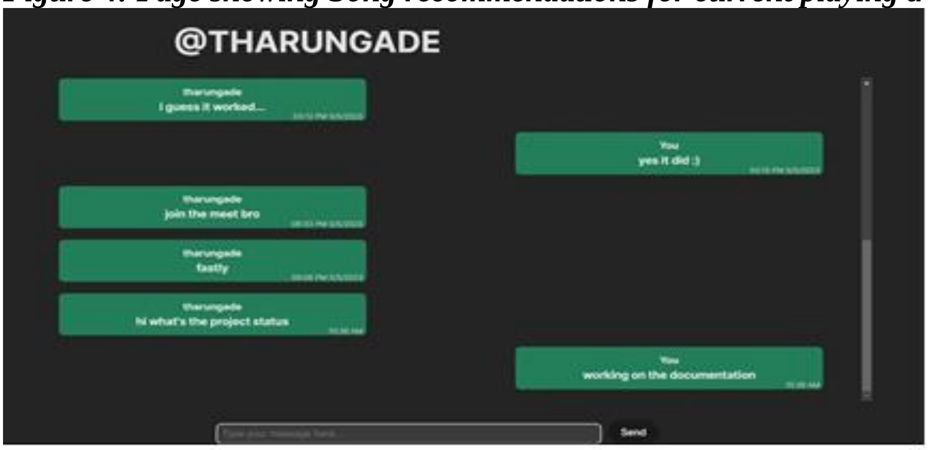

Figure 5: Page showing individual user’s chat

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 2, No. 2, February : 2024

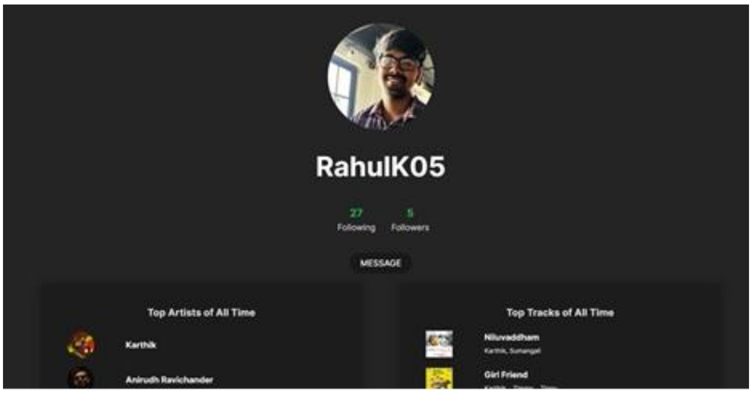

Figure 6: Page showing user's profile

VIII. APPLICATIONS

a. **Personalized Music Recommendations:**Provides personalized music recommendations to users based on their listening habits and those of similar users. This helps users discover new music they may be interested in and enhances their overall listening experience.

b. **Artist Discovery:** Users can explore the music and people on Spotify in more depth, discovering new artists, and learning more about their favorite artists.

c. **Music Industry Insights:** offers insightful information about the music business, helping artists, music labels, and streaming platforms understand user preferences and trends. This can be used to inform marketing strategies, music production, and distribution.

d. **Music Education and Research:** Used in music education and research to examine and comprehend patterns and trends in music. This can help researchers understand the music preferences of different generations and demographic groups.

e. **Social Recommendations:** Allow users to share their music preferences and recommendations with friends and followers on the platform. This can help users discover new music through their social network and create a sense of community.

f. **Music-Focused Community:** Makes a music-centered local area where clients can interface with other people who share comparable melodic preferences and interests. This can lead to the discovery of new music and artists and enhance the overall music-listening experience.

g. **Music-Centered Social Networking:** Serves as a social network where users can connect, share music and playlists, and engage in discussions about music. This can help users build connections with others who share their music preferences and interests.

IX.CONCLUSION

In conclusion, the proposed system represents a significant improvement over the existing system because it provides users with a more personalized and efficient way to discover new users(friends), also music, and artists on the Spotify platform. This increases user engagement and satisfaction with the platform.

REFERENCES

[1] Zhibo Wang, Jilong Liao, Qing Cao, Hairong Qi and Zhi Wang "Friendbook: A SemanticBased Friend Recommendation System for Social Networks," IEEE Transactions on mobile computing, vol. 14, no. 3, (2015), pp. 214-249.

[2] S. Huang, J. Zhang, L. Wang and X. -S. Hua, "Social Friend Recommendation Based on Multiple Network Correlation," in IEEE Transactions on Multimedia, vol. 18, no. 2, pp. 287-299, Feb. 2016, doi: 10.1109/TMM.2015.2510333.

[3] F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh, Recommendation systems: Principles, methods and evaluation, Egyptian Informatics Journal, Volume 16, Issue 3,2015, Pages 261-273, ISSN 11108665

[4] Bian, Li & Holtzman, Henry. (2011). Online Friend Recommendation through Personality Matching and Collaborative Filtering.
[5] “Suggesting Friends Using the Implicit Social Graph" by L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. (2008)
[6] “Friend Recommendation System Using a Hybrid Method of Collaborative Filtering and Content-Based Filtering" by S. Kim and H. Park. (2018)
[7] “Friend Recommendation for Social Networking Services Using Genetic Algorithm and Social Network Analysis" by S. S. R. T. M. De Silva and R. K. N. D. Rajapakse. (2018)