Emotion-Based Music Recommendation System

Balusu Bhanu Prakash
Department of Data Science and Artificial Intelligence.
Indian Institute of Information Technology Dharwad.
Dharwad, Karnataka
20bds012@iiitdwd.ac.in

Bhashapangu Nikhil
Department of Data Science and Artificial Intelligence.
Indian Institute of Information Technology Dharwad.
Dharwad, Karnataka
20bds013@iiitdwd.ac.in

Bommaragoni Karthik
Department of Data Science and Artificial Intelligence.
Indian Institute of Information Technology Dharwad.
Dharwad, Karnataka
20bds014@iiitdwd.ac.in

Gagan Vadlamudi
Department of Data Science and Artificial Intelligence.
Indian Institute of Information Technology Dharwad.
Dharwad, Karnataka
20bds019@iiitdwd.ac.in

***Abstract*—This paper introduces an Emotion-Based Music Recommendation System implemented in Apache Spark using Scala. Leveraging the power of unsupervised clustering and machine learning. The key objective is to categorize music into distinct mood clusters based on important audio features like Energy, Valence, Danceability, and Loudness.**

Our approach combines Spark's distributed processing capabilities with K-Means Clustering to group songs efficiently. It then employs the Random Forest classification algorithm, implemented in Scala, to associate users' emotional states with these song clusters. The resulting system provides personalized and emotionally resonant music recommendations, thereby enhancing the user's music listening experience.

This paper presents a novel approach to music recommendation, focusing on emotions rather than traditional genre or artist preferences, and showcases the scalability and efficiency of Apache Spark when handling large music datasets.

I. Introduction

Music has the ability to evoke strong emotions and can also bring people together and provide a common language and experience for people from different backgrounds and cultures. Many people find great joy and fulfillment in creating and performing music, and it can be a powerful tool for personal and social transformation. We wanted to integrate AI algorithms in the field of Music.The emergence of digital music platforms and streaming services has opened up new avenues for crafting music recommendations that align with the listener's emotional disposition. In this paper, we present an Emotion-Based Music Recommendation System that leverages the power of Apache Spark, making efficient use of Scala for processing and analysis. [1]

The core premise of this system revolves around the idea that music can be categorized based on the emotions it elicits. Rather than relying solely on traditional factors like genre or artist, we research deeper into the emotional resonance of songs. To achieve this, we employ a dataset encompassing a substantial collection of 36,000 Spotify songs, spanning nearly two decades from 2002 to 2020. Each song in this dataset is thoughtfully labeled with one of four distinct emotions: Happy, Sad, Neutral, and Disgust.A person's emotions can be detected by the suggested system, and if the individual is feeling down, a playlist of the most upbeat, musically-related songs will be played. Additionally, if the emotion is favorable, a particular playlist containing various musical genres that will amplify the pleasant emotions will be displayed.Implementation of Emotion-Based Music Recommendation System is performed using Random Forest which gives approximately 91.85% of accuracy. [2]

II. Datasets Description

A. Spotify Songs Dataset

The Spotify Songs dataset is a comprehensive collection of songs and their associated features obtained from the Spotify music streaming platform. The dataset consists of 36000 Spotify songs from 2002 to 2020. It provides valuable information about songs, such as audio features, metadata, and user interactions. The dataset includes a wide range of features for each song, including acousticness, danceability, energy, instrumentalness, loudness, speechiness, tempo, valence, and more. These features capture various aspects of the songs, such as their musical properties, mood, and genre.

- track id: The Spotify ID for the track.
- artists: the names of the musicians who sang the song. if there are many artists, there is a ; between them.
- album name: Album name of the song.
- track name: Track name.
- danceability: A song's danceability is determined by a number of musical factors, including pace, rhythm stability, beat intensity, and overall regularity. The least danceable value is 0.0, while the most danceable value is 1.0. 5
- energy: A perceptual gauge of intensity and activity, energy ranges from 0.0 to 1.0. In general, frenetic music seems quick, loud, and boisterous. For instance, a Bach prelude rates poorly on the energy scale compared to death metal.

- loudness: The overall loudness in decibels (dB).
- valence: A scale from 0.0 to 1.0 used to describe the overall musical positivity of a tune. High-valence music sound happier, cheerier, and more euphoric, whereas low-valence tracks sound more depressing, angry, and sad.
- track genre: The genre of the song. 6

III. Methodology

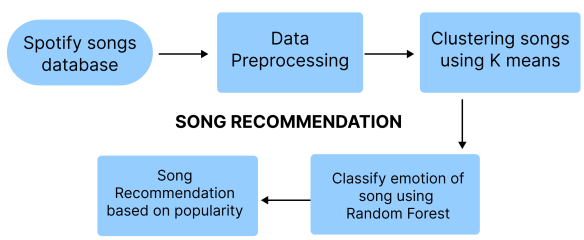

Fig. 1. *Proposed Solution Architecture.*

A. Data Preprocessing

To create an effective song recommendation system, we preprocess the Spotify songs dataset by filtering the dataset to include only songs released after the year 2002. This ensures that the data is contemporary and aligns with the preferences of modern music listeners.To ensure data quality, we employ a process to identify and eliminate duplicate entries based on song names. This step ensures that each song in the dataset is unique and avoids overrepresenting certain songs.We extract relevant features, including danceability, energy, valence, and loudness, which are used for clustering and emotion prediction. Data cleansing involves handling missing values, outliers, and inconsistencies in the dataset. While Spotify's dataset is generally reliable, some data cleaning may be necessary to ensure the accuracy of the features extracted.Since the dataset includes emotions such as Happy, Sad, Neutral, and Disgust, we encode these emotions into numerical values for machine learning purposes. [3]

B. Clustering

While applying the K-means algorithm, we carefully select the key audio features that best capture the emotional characteristics of songs. These features include: 'danceability', 'energy', 'valence', and 'loudness'. In our case, we set K to four to create four distinct clusters—one for each of the emotions: Happy, Sad, Neutral, and Disgust. This clustering process forms the foundation for accurate emotion prediction. When a user interacts with the system and expresses their emotional state, the system can now map that emotional input to the nearest cluster, determining which emotional category (Happy, Sad, Neutral, or Disgust) best matches the user's mood. This step enables the system to categorize songs based on their shared attributes, providing a foundation for accurate emotion prediction. [5]

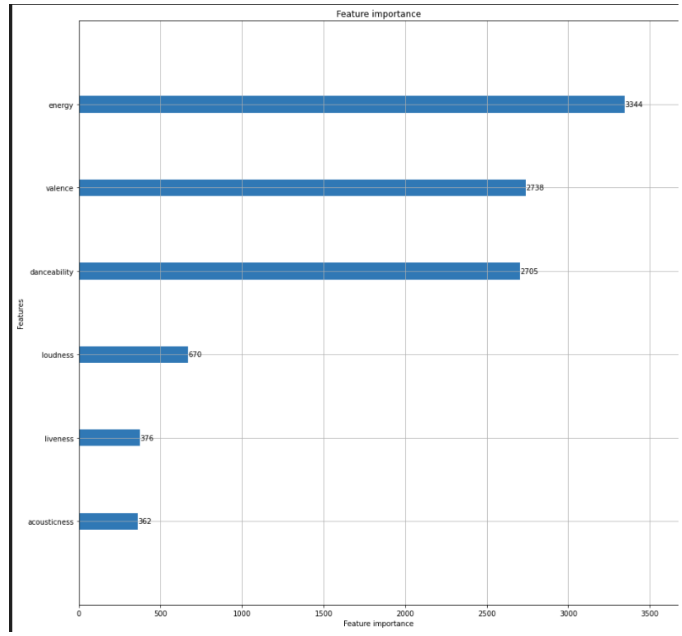

Fig. 2. *Important Features*

C. Model Random Forest Classifier

The Random Forest model is trained using a portion of the preprocessed dataset, where each song's emotional label (Happy, Sad, Neutral, or Disgust) serves as the target variable. The model learns the relationships between the selected audio features and the emotional labels through iterative decision tree construction. Once the Random Forest model is trained, it can predict the emotional state of a user based on their input. For instance, when a user expresses feeling "happy," the model can map this input to one of the four clusters (Happy, Sad, Neutral, or Disgust) created during the K-means clustering process. The strength of Random Forest lies in its ability to generalize and make accurate predictions even for data points it has never encountered before.It learns the complex relationships between audio features and emotional states, allowing us to predict user emotions accurately. This, in turn, enables us to recommend music that resonates with the user's mood, providing a highly personalized and emotionally satisfying music listening experience. The combination of clustering and Random Forest classification forms a robust foundation for our innovative music recommendation system. [4]

D. Song Recommendation

Our system operates by suggesting songs that belong to specific clusters, each of which corresponds to a particular emotional state. This approach allows us to cater to the user's current emotional disposition. The songs recommended within these clusters are carefully chosen based on two key criteria: popularity and completeness of information.

1) Emotional Relevance: Our system categorizes songs into clusters that encapsulate various emotional tones or themes. When a user interacts with our system

and expresses their emotional state or preference, our algorithm identifies the appropriate cluster that aligns with the user's feelings at that moment.

2) Popularity: Within the selected emotional cluster, our system prioritizes songs that have a high degree of popularity which should be greater than 85%. This ensures that users are presented with songs that are not only emotionally resonant but also well-received by a broader audience. Popularity metrics can include factors like the number of listens, downloads, or positive user reviews.
3) Data Completeness: To provide a seamless and enjoyable listening experience, we take into consideration the completeness of song information. We avoid recommending songs that have missing or incomplete metadata, such as artist names, album details, or cover art. This ensures that users receive a well-curated list of songs with all the necessary information to enhance their listening experience.

E. Spark Architecture

Our project's architecture is a well-orchestrated blend of Apache Spark with Scala, Amazon S3 storage, and a carefully configured virtual cluster (V-cluster) with worker nodes. This combination leverages the strengths of each component to create an efficient and scalable Emotion-Based Music Recommendation System.

- In our project the backbone of data storage and retrieval lies in Amazon S3 buckets. S3 provides a highly reliable, scalable, and cost-effective solution for housing the extensive Spotify songs dataset and other relevant data. By storing the dataset in S3, you ensure that your project can seamlessly access and process the music data as needed.
- The Databricks runtime version which we used ensures that our Spark application runs smoothly on this configuration, with Scala as the chosen language to develop and execute data processing and machine learning tasks.We have configured a Spark cluster with a minimum of 2 worker nodes and a maximum of 8 worker nodes. This dynamic scaling allows our cluster to adapt to varying workloads.The worker nodes in our Spark cluster are i3.xlarger instances, which offer high CPU and memory resources. These instances are well-suited for data-intensive and memory-intensive workloads.

F. Benefits of using Spark

Without Spark, handling a dataset of 36,000 Spotify songs would be a formidable challenge, as it excels in parallel data processing, significantly reducing processing times for tasks like clustering and model training. Scala and Spark simplify code development and maintenance. Without Spark, coding complex data pipelines and machine learning models could be more challenging.Spark clusters are optimized for performance and resource utilization. Without Spark, resource management might require manual intervention and expertise.

Emotion-Based Music Recommendation System architecture, built on Apache Spark with Scala, Amazon S3, and a carefully configured V-cluster, delivers a robust and scalable solution for personalized music recommendations. This architecture leverages Spark's distributed computing power, allowing efficient processing of a vast music dataset and enhancing user engagement by providing music recommendations that align with their emotional states.

IV. Results

We experimented with models to recommend the songs using emotion, specifically using RandomForest,The model learns the relationships between the selected audio features and the emotional labels through iterative decision tree construction. Our results are summarized below:

Model	Accuracy
Random Forest	91.8%

We linked the labels to the folders of the music database and used a random number generator for popularity filters more than 75, so the user will receive a random popular playlist recommendation based on the emotions that were detected. Below table shows sample results of recommended playlist.

Emotion	Songs
Happy	The Nights (Avicii), Sour Candy (Lady Gaga, Blackpink), Say So (Nicki Minaj)
Sad	Lovely (Billie Eilish, DJ Khalid), Bubblegum (Clairo), No time to die (Billie Eilish)
Neutral	Getting good (Lauren Alaina), Give it to you (Julia Michaels), What if I never get over you (Lady A)
Disgust	Take a Bow (Rihanna), Fall Apart (Post Malone), Scared to be Lonely (Dua Lipa, Martin)

V. CONCLUSION

Our Emotion-Based Music Recommendation System not only provides personalized and emotionally aligned music recommendations but also showcases the significance of adopting modern big data and machine learning technologies in redefining the music streaming landscape. Our project leverages the power of Apache Spark with Scala, a dynamic and efficient combination that enables us to efficiently process a massive dataset of 36,000 Spotify songs spanning nearly two decades. Spark's distributed processing capabilities are very useful for handling large music datasets, conducting K-Means Clustering, and training Random Forest models. It is a testament to the potential of technology to create more immersive and tailored user experiences, enriching the way we interact with music in the digital age.

References

[1] Youngmoo E Kim, Erik M Schmidt, Raymond Migneco, Brandon G Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A Speck, and Douglas Turnbull. Music emotion recognition: A state of the art review. In *Proc. ismir*, volume 86, pages 937–952, 2010.

[2] Pranesh Ulleri, Shilpa Hari Prakash, Kiran B Zenith, Gouri S Nair, and Jinesh M Kannimoola. Music recommendation system based on emotion. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2021.

[3] Xinyu Yang, Yizhuo Dong, and Juan Li. Review of data features-based music emotion recognition methods. *Multimedia systems*, 24:365–389, 2018.

[4] Yi-Hsuan Yang, Yu-Ching Lin, Ya-Fan Su, and Homer H. Chen. A regression approach to music emotion recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):448–457, 2008.

[5] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. In *2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 10)*, 2010.