EMOSENSE

Group 55

Ashwin Sheoren
Ashish Kamathi
Shreya Bhatia
Srishti Singh
Vedant Patil
Tarushi Gandhi

Problem Statement

- Develop a mood detection system to predict mood of a user based on the physiological data collected from Fitbit devices (heart beats, SPO2, etc.) & recommend music accordingly.
- Create a personalised music experience for user that enhances their mood & overall well being.
- Challenges-
  - Processing large amounts of Fitbit data
  - Identifying user's mood based on data
  - Recommending suitable music
  - Classifying mood of music based on lyrical & audio data

Motivation

- Smartwatches, capable of monitoring fitness data, can be used in music categorization
- Use of physiological data for better recommendation
- Investigate the potential of AI & ML, to categorise music based on audio and lyrical content and further enhance the recommendation through user feedback.

Literature Review

1. Induced Emotion-Based Music Recommendation through Reinforcement Learning.
2. Artificial Neural Network (ANN) Enabled Internet of Things Architecture for Music Therapy
3. Music Emotion Classification based on Lyrics-Audio using Corpus based Emotion
4. Multi-modal Music Emotion Classification based on audio and lyrics
5. Based on Improved Convolutional Neural Network
6. Deep learning-based late fusion of multimodal information for emotion classification of music video

Novelty

- We aim to leverage the capabilities of smartphones to collect physiological data & use it for music recommendation based on mood detection.
- Our approach combines emotion detection from human body parameters with music recommendation, which is largely unexplored area for recommendation.
- By utilizing smartwatches for this purpose, we can potentially enhance the overall health and fitness of individuals.
- Widespread availability & affordability of smartwatches makes our project promising in terms of its potential impact on improving quality of life.

Methodology

Our system takes in the user's physiological data like heart beat rate and SPO2 levels (blood oxygen levels) to categorise their mood into three categories: Relaxed, Normal, and Stressed. The thresholds for heart rate and blood oxygen can be defined based on the literature reviewed so far. For example:

1. Relaxed: Heart rate below 60 bpm (beats per minute) and blood oxygen level above 95%.
2. Normal: Heart rate between 60 bpm and 100 bpm and blood oxygen level above 95%.
3. Stressed: Heart rate above 100 bpm or blood oxygen level below 95%.

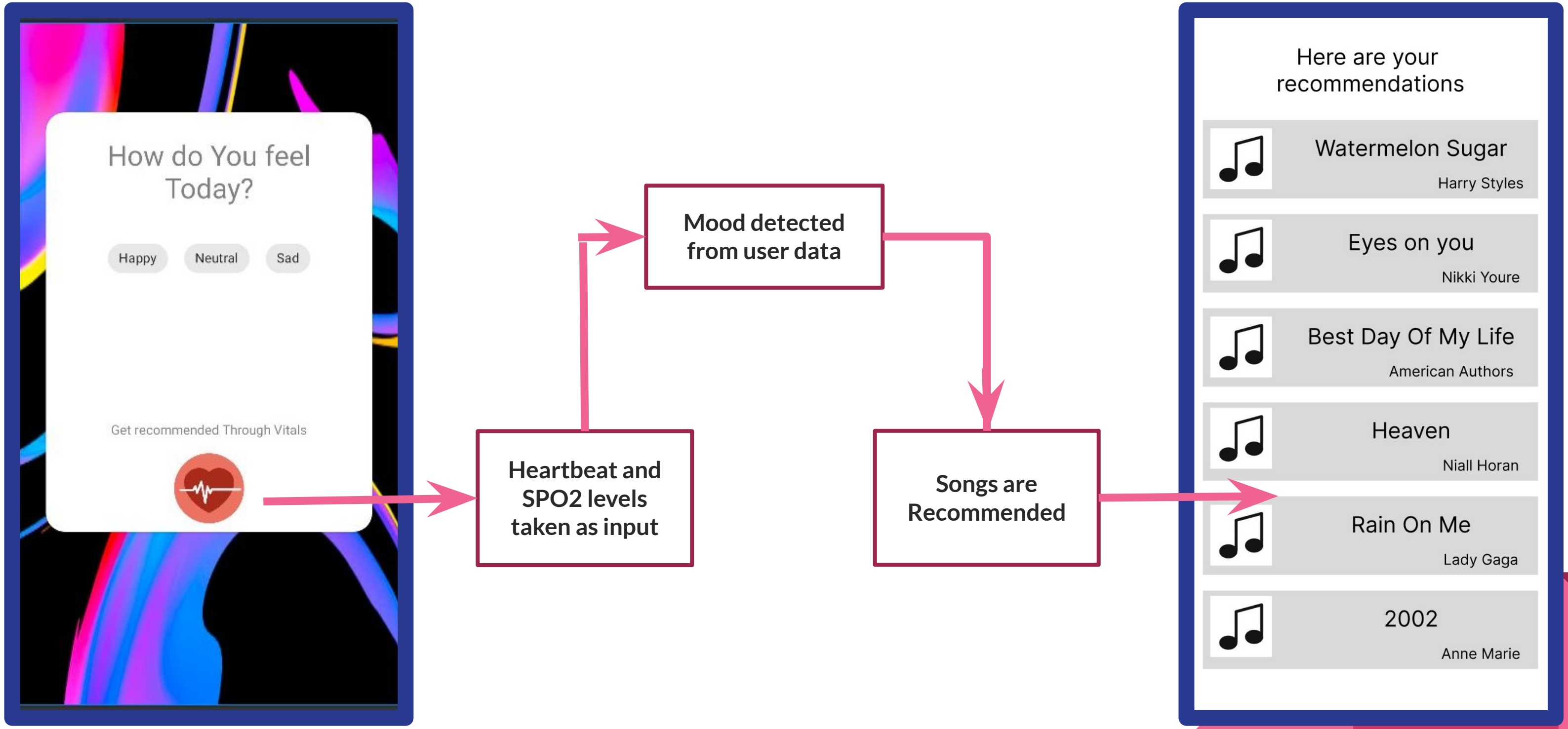
- Recommend top 10 songs to the user based on their mood
- Use both lyrics and audio features for sentiment analysis of a song
- Obtain songs from SpotifyAPI, Analyze lyrics sentiment using VaderSentiment library
- Perform natural language preprocessing using TextBlob
- Get sentiment scores of lyrics using Lyricsgenius to access the Genius API
- Assign sentiment scores such that a score close to 1 corresponds to happy songs, close to 0 is for neutral, and -1 for sad songs.

Methodology

- Obtain audio features for each song in the dataset using the Spotify API
- Combine audio features with sentiment scores to create a comprehensive dataset
- Train K Means clustering model using dataset to group songs based on features & sentiment scores
- Incorporate user listening history through collaborative filtering to improve recommendations
- Recommend top 10 songs from the cluster that best matches the user's mood
- Use a newer dataset to better cater to present generations
- Allow user to provide feedback on songs using GUI buttons
- Songs recommended based on thorough sentimental analysis of both lyrics and audio features
- Integrated GUI using tkinter library.

The following audio
features are used-[’acousticness’, ’danceability’, ’energy’, ’instrumentalness’, ’liveness’,
’loudness’, ’speechiness’, ’valence’, ’sentiment’]
After recommending songs, they are also played using the pygame library.

App GUI Workflow



DataBase and Code

We have used songs extracted from SpotifyAPI. Here are some examples of what the Spotify API can return when called:

- Search API: When you search for a keyword, the Spotify API returns a list of tracks, albums, artists, and playlists that match that keyword.
- Top Tracks API: This API returns the most popular tracks for a specific artist based on their Spotify user data.
- Recommendations API: This API returns a list of recommended tracks based on a variety of parameters, such as user listening history, mood, genre, and tempo.
- Featured Playlists API: This API returns a list of featured playlists that are curated by Spotify based on different themes and genres.

- Physiological data -

The code and README files can be found at our github repository

https://github.com/Vedant0925/EmoSense

Evaluation

Personalization - assesses if a model recommends many of the same items to different users. A high personalization score indicates user’s recommendations are different, meaning the model is offering a personalized experience to each user.

Intra List Similarity - high intra list similarity indicates that the recommendation are of similar type. It is calculated as the average cosine similarity of all items in a list of recommendations.

Evaluation

Precision@k is a fraction of top k recommended items that are relevant to the user

P = (# of top k recommendations that are relevant)/(# of items that are recommended)

Recall@k or HitRatio@k is a fraction of top k recommended items that are in a set of items relevant to the user. The larger the k, the higher the hit ratio since there is a higher chance that the correct answer is covered in recommendations.

R = (# of top k recommendations that are relevant)/(# of all relevant items)

Personalization Score for mid:
0.6666666666666667

intra list similarity for mid_happy = 0.25
intra list similarity for mid_sad = 0.25
intra list similarity for mid_nuetral = 0.16

Precision@K for mid_happy (k=5) 0.2
Precision@K for mid_sad (k=5) 0.2
Precision@K for mid_nuetral (k=5) 0.4

Personalization Score for final:
0.686721169655858

intra list similarity for fin_happy = 0.64
intra list similarity for fin_sad = 0.79
intra list similarity for fin_nuetral = 0.91

Precision@K for fin_happy (k=5) 0.4
Precision@K for fin_sad (k=5) 0.6
Precision@K for fin_nuetral (k=5) 0.4

Recall@K for mid_happy (k=8) 0.6

Recall@K for mid_sad (k=8) 0.75

Recall@K for mid_nuetral (k=8) 1.0

Recall@K for fin_happy (k=8) 0.6667

Recall@K for fin_sad (k=8) 0.8

Recall@K for fin_nuetral (k=8) 1.0

Contributions

Ashish - GUI and Android App, Data collection

Ashwin - Creating Flask app and deployment on online server, Data Preprocessing

Shreya - Mood Classification from physiological data, Literature review

Srishti - Evaluation of the Final model, Data collection

Vedant - User History Extraction and Final modelling, Data Preprocessing

Tarushi - Evaluation of the final model, Literature review

Thank You