Music Recommendation System Based on Emotion

Pranesh Ulleri [1], Shilpa Hari Prakash [1], Kiran B Zenith [1], Gouri S Nair[1], and Jinesh M. Kannimoola [2]

[1]Department of Computer Science and Engineering
[2]Department of Computer Science and Applications
Amrita Vishwa Vidyapeetham, Amritapuri, India

2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT) | 978-1-7281-8595-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCCNT51525.2021.9579689

***Abstract*—With the onset of lockdown in the COVID-19 scenario, people were forced to confine themselves within the four walls of their rooms which in the meantime invited mood disorders like depression, anxiety etc. Music has proven to be a potential empathetic companion in this tough time for all. The proposed emotion-based music recommendation system uses user emotion as an input to recommend songs that are ascertained using facial expression or using direct inputs from the user. The model uses a Random Forest classifier and XGBoost algorithm to identify the song's emotion considering various features like instrumentalness, energy, acoustics, liveness, etc. and lyrical similarity among songs with the help of Term-Frequency times Inverse Document-Frequency (TF-IDF). The results of comprehensive experiments on real data confirm the accuracy of the proposed emotion classification system that can be integrated into any recommendation engine.**

***Index Terms*—Recommendation System, Emotion classification, Machine Learning, Lyrics classification, Random Forest, XGBoost**

I. Introduction

As soon as the COVID-19 lockdown came into effect in March 2020, many discussions popped up on mental health. It is well-known that natural disasters, wars and other mass traumas can lead to significant worldwide psychological distress. Many experts feared that the world might have to face a mental health epidemic along with the global pandemic as weeks or months of anxiety, fear or social isolation can take their toll. The Hindu newspaper reported that in India, doctors have admitted that the number of patients coming with depression, anxiety, insomnia related to the COVID-19 pandemic has increased. According to a survey conducted by the World Health Organization (WHO), it was reported that the pandemic has also halted critical mental health services in 93% of the countries worldwide [1]. That has created a tremendous impact as people started not to receive the mental health care services as they had to confine themselves to the four walls of their room.

Depression is a topic of high priority conditions covered by the WHO's mental health Gap Action Programme (mhGAP) [2]. This program aims to scale up the mental, neurological and substance use disorder services for countries especially with low and lower middle incomes. Some conditions addressed by the mhGAP are: depression, schizophrenia and other psychotic disorders, suicide, dementia, epilepsy, disorders due to use of illicit drugs, disorders due to use of alcohol, and mental disorders in children. This package comprises interventions for prevention and management for each of these priority conditions. Music can reduce stress and pain and improve mood is a fact that been recognized a considerable time ago. According to the Medical Resonance Therapy Music by the classical composer and musicologist Peter Hübner, music offers unique advantages over depression [3]. Also it was identified that listening to music that congruent with your mood will bring a feel of presence of an empathetic friend [4]. This concept of using music to cure depression inspires us to develop an emotion based music recommendation system using deep learning.

Broadly recommendation systems are divided into two types which are content based filtering and collaborative filtering [5]. Since content based filtering deals with what the user liked in the past, it may not match with the genre of song that the user wants to listen presently and in addition there would be a cold start recommending a fresh user with all the music that they might like. Similarly, collaborative filtering checks for similarity among users and suggests songs that the user may like rather than what the user really wants to listen to at a point of time. Hence it is important to consider user's environment, activity and emotion to recommend the right genre of songs to them. Because of the above mentioned drawbacks of content based and collaborative filtering methods [6], today's streaming services like Spotify, iTunes, etc. use hybrid recommendation approach [5].

Our research focuses on emotion-based music recommendation system that recommends songs that suits the user's current emotion assuming that the user is an unbiased listener [7]. Listening can basically of many types: Biased, Discriminative, Critical, Appreciative, Therapeutic, Relationship, Empathetic, Sympathetic, Dialogic, Comprehension and Evaluative [8]. A listener will only be able to enjoy the music irrespective of artist and charts only if he/she is an unbiased listener. A study showed that simply being said that the performer is a professional changes way our brain responds to it [9].The system takes music files including its metadata and classifies it into sentimental, happy, energetic and calm. Then it identifies the user emotion and checks for a corresponding genre of songs and plays them until they are active or want to switch the mood. The main goal of this paper is to demonstrate that manually collecting sentiments from user provides the real time emotion of the user. Collecting real time emotion of a user can improve the performance of a music recommendation system. Collecting sentiments of the user by scraping social media or from real time face and sound detection may not match the right mood of the user. Manually collecting real time emotion of a user uses a low complexity solution and is handy to use.

We divide the approach into three steps: user emotion extraction, music emotion extraction, and music recommendation. The system extracts emotion of the songs in the datasets, later when the user selects an emotion, it recommends a song within the emotion with the help of content based recommendation system. The music emotion model gives out a pretty accuracy in the range 77% to 85%. The observational result is an important factor in such project, as it is nearly impossible to conclude the actually mathematical accuracy of the complete system considering each user interprets each song with a different emotion. The remainder of this paper is organized as follows: Section 2 briefly describes the related work in this field; Section 3 gives the detailed description of our methodology and implementation; Section 4 discusses our result; and, finally, Section 6 gives conclusions and future directions.

II. Related Works

Recommendation system is one of the most extensively studied research problem in literature. In this section, we summarize the studies related to emotion based music recommendation systems. User Emotion can be extracted in many approaches such as sentiment analysis from social media, wearable sensors, real-time video or audio inputs. In all these methods, user emotion is extracted from user's activities and produce mere assumptions about their mood. In case of sentiment analysis from social media, eSM with Sentimeter-Br2 metric associated with a novel correction factor based on the user's profile are used after scraping the required sentences from social media or micro-blogs [10]. However, the use of machine learning requires a large amount of data to get a reliable sentiment result, because an unusual sentence may cause noise in the sentiment's calculation. Also, if a social network extracted message doesn't match the user's emotion then recommendation may go wrong and it violated user's privacy.

Ayata et al. [11] proposed a method is to learn emotion of a user from signals obtained from wearable physiological sensors like bands, rings, glasses and headsets. Here Galvanic skin response is used to capture physiological reactions that generate excitement. When people get excited body sweats, amount of salt on skin and skin's electrical conductance changes. Also photo plethysmography signals are used to measure volume changes in different parts of the body. Performance of existing collaborative or content based recommendation system can be increased by using the data from the sensor as a supplement to it. But accuracy rate depends based on the quality of physiological wearable sensor.

Using feature extraction on face image or audio input and classification of the same with Support Vector Machine (SVM) is another real-time method to detect user emotion. Viola and Jones are used to detecting a face from image and classification using support vector machine [12]. The system is prone to give unpredictable results in difficult light conditions and also causes concerns about privacy in users. Another method suggested in [13] asks user about their mood and then recommends a music. We can accomplish this approach by preparing a user interface which has different emotions as choices and then, depending on user's selection, recommending songs of the same genre. In this way, there is no chance for ambiguity in the way how people express themselves with greater or lesser intensity of feeling.

Music recommendation system identifies the emotion of the songs by using feature extraction and SVM on a part of the song or by parsing through the lyrics of the song. Million Song dataset(http://millionsongdataset.com/) is one of the most commonly used data set for song emotion detection. Gilda et al. [14] use part of a song for emotion detection. In this approach, we need to convert the song into an audio clip for feature selection, which is a time-consuming process and requires huge storage as well. The noise added during conversion may also affect the classification. In another approach [15], Zhang et al. extracted emotion from the lyric of a song. It processes text data for feature selection, which comparatively a simple method. The extracted features processed with machine learning algorithms to classify the emotion. Further, addition of new songs in future becomes simpler by this approach.

The system recommends songs to the user by mapping both user mood and song's emotion. This requires easily approachable database for faster recommendation [16]. Table I and II summarize the related works that we have discussed so far.

TABLE I
Comparison of User Emotion Extraction Methods

Algorithm Name	Cost	Privacy	Accuracy	User Friendly
Social Media [7]	Moderate	High Risk	Less	Moderately
Wearable Sensors [17]	High	Moderate Risk	Better	Moderately
Image Input [18]	Moderate	Moderate Risk	Better	Moderately
Audio Input [5]	Moderate	Moderate Risk	High	No
Manual Input [11]	Less	No Risk	High	Yes

TABLE II
Comparison of Music Emotiofn Extraction Methods

Algorithm Name	Data-Availbility	Future Addition	Easiness
Million Song Dataset [7]	Available	Not Possible	Easiest Method
Audio [17]	Manual Preparation	Possible	Complex
Lyrics Parsing [18]	Manual Preparation	Possible	Moderate

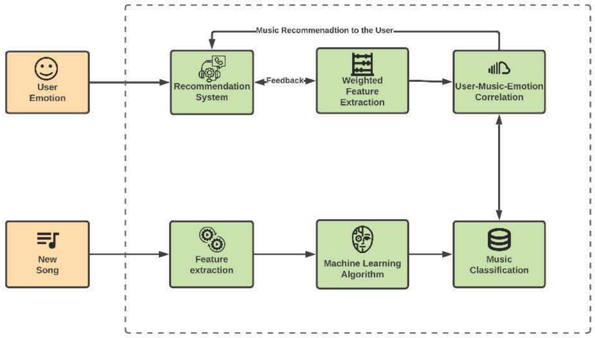

Fig. 1. High level Architecture for Emotion Based Music Recommendation System

III. Design and Implementation

When a new song is introduced into the system, initially it undergoes the process of feature extraction with the help of machine learning algorithms. The music has been classified into various types of moods with the help of the feature extraction technique. The recommendation system takes the user's emotion as input to form an association with the mood of the music. The system recommends songs that match the emotion of both the user and music.

We used the following data sets in our recommendation systems for training, testing, and predicting

- Emotion data set [19]: The training data set requires an additional column to describe the emotion of songs. We collected this data set from an open-source project in Github. The dataset was originally an older version of Spotify dataset with 686 non-null values
- Lyrics dataset [20]: It comprises lyrics of the song with some additional feature
- Spotify dataset [21]: It is a fresh dataset that is available in the Kaggle competition. The dataset contains audio features of songs that were released between 1922 and 2021
- Million Song Dataset's subset [22]: It is a freely available collection of audio features and metadata of songs that were legally available to The Echo Nest. This subset of the actual dataset contains the audio features of 10,000 songs.

A. *Data pre-processing*

The data set with emotion can be prepared first for training the model. The system should be able to identify the mood of a song with the features in the dataset. As it is noticed that certain features do not contribute to the training process but rather add noise, such features are removed while data processing [23]. For example, features like "ID" or "length", which is the duration of the song, would not contribute to the efficiency or effectively finding the emotion or mood of the song, hence it is better to drop those features before training. Other features dropped are "album" which is a unique name given to a song or a group of songs that could have mixed emotions, "artist" because an artist could sing songs of all emotion in his/her/their career, "popularity" here does not give the understanding of the emotion of the song, it is effective to understand the trend but not for the best interest to train the model with the feature. Similarly, "key" and "time_signature" are also removed. Then a data structure that contains columns and rows, popularly known as dataframe is generated. The dataframe contains the exact features necessary, considering the above mentioned criteria, other features are dropped. We used both the data sets from Kaggle in this process. The lyrics feature from the lyrics dataset is joined to the Spotify dataset with the common feature "name".

B. *Algorithms*

We train a model with the dataset which has the emotion features. Random Forest classifier is selected for this first training, the n_estimator which has a range of 10 to 100 is the number of trees that forest would contain, was fixed at 20 [24]. Criterion was kept as the default which is "Gini". The min_sample_split that gives the idea of the minimum number of samples required to split the internal node was kept at default integer value of 2. The model was trained with these parameters and the rest of parameters were kept as default as described in Fig. 2. After training the model, the importance

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=20,
                       n_jobs=None, oob_score=False, random_state=0, verbose=0,
                       warm_start=False)
```

Fig. 2. Random classifier

of each feature was obtained and was plotted as shown in Fig. 3. This would give the viewer a better analytic idea about the features and the relevance of the feature with respect to the corresponding mood [25]. By performing the following above mentioned process, the research also get a clear idea of features that are not contributing or makes least difference in predicting the mood. These features could be dropped as their existence in the data set could only add more noise while training it in future. In order to remove the least important features, a value of 0.04 was kept as the threshold and any feature that hold importance value, less than mentioned value, would be dropped. This refined data frame is saved and split into train data and test data in the 80:20 ratio, ie. 80% of the dataset is split for training the data and the remaining 20% of the data is for testing and validation of the model.

The test data is subjected to the eXtremeGradientBoost (XGBoost) classifier with the following parameters grid: "learning_rates" with values [0.1, 0.2, 0.5], "max_depth" with values [5, 10, 15], "n_estimators" with values [150, 250, 300], "min_child_weight" with values [3, 5, 10] In RandomizedSearchCV, cross validation is Stratified Shuffle split of 4 n_splits, 0.2 test_size and 0 random_state. The best estimator for the cross validation that was obtained was as follows The

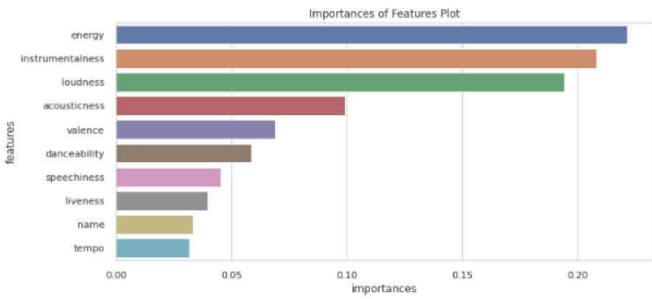

Fig. 3. Feature Importance

model was trained for these parameters and the accuracy was obtained in the range of 77 to 85 percent. This model is saved and can be used to predict the mood of any song which has a set of features.

Recommendation systems can be prepared using both supervised or unsupervised learning models. The model that was described above was trained and validated using an open source dataset which was prepared from Kaggle's Spotify dataset with emotions tagged as labels. Since actual Spotify dataset is unlabeled and cannot be used to train a supervised learning model, it can only prepare a model that has no notion of the output during the learning process. Such algorithms learn the underlying features from data which are available and then group them according to similar characteristics. K-means cluster is one of the most simple and popular unsupervised clustering algorithms [26]. It deduces classifiers or classification models by using a statistical vector as its input. Hence, to compare both supervised and unsupervised model, a K-Mean Clustering model was prepared and it clustered the songs into various tags. Fig. 4 shows the workflow of the unsupervised learning classification model. After finding an optimal k value

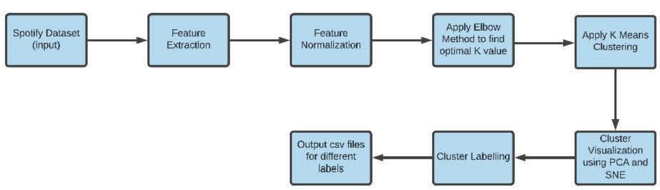

Fig. 4. High level Architecture of Unsupervised Model for classifying Spotify Dataset to various Emotions.

by using elbow method and then clustering the song data based on various emotions from the Spotify dataset. We observed that there was no clarity on the reason for grouping various songs into a particular emotion tag or the specific commonalities that the members of each emotion group shared. Also the number of song data for the third cluster seemed to be less compared to that in the first cluster. To clarify, we manually checked few random songs and noticed that the emotions of some songs did not match to its tag. K- mean cluster cannot give accurate results when there are outliers and is sensitive to initialization. Due to these reasons and after manually checking both tagged dataset, it was understood that using the Spotify dataset with label predefined was better.

C. *Music Emotion Extraction*

The XGboost model is used for the extraction of emotion from the dataset which has lyrics. This is done after making an identical dataset and then trimming the excess features by dropping out. The features that remain in the identical dataset are the same features as the model was trained. The features are 'valance', 'acousticness', 'danceability', 'energy', 'instrumentalness', 'loudness' and 'speechiness'. After prediction, it produces an array of moods and converted to a list and concatenated with the dataset which has the lyrics. This dataframe is divided into four dataframes for each emotion. We are keeping in mind that at a later point it would be more accurate and effortless for the system to select songs according to the emotion that they select.

D. *Lyrical Similarity*

The lyrics of any song hold the true meaning of that song. The algorithm uses this feature to improve the recommendations. In python environment we import 'TfidfVectorizer' library from 'sklearn.feature_extraction.text'. We convert convert raw text format into matrix with Term-Frequency times Inverse Document-Frequency (TF-IDF) [27] using TfidfVectorizer function in the library mentioned above. The lyrics feature in the dataframe is passed as a parameter and a document term matrix is returned. A similarity matrix (as showin in Fig. 5) is generated with the linear_kernel function with early generated document term matrix as both the parameters. A square gram matrix is returned after the linear kernel. A user defined function is created which takes name of a song and the similarity matrix as the parameter, and sorts the values that are close to the value of the song that was passed and returns a list of ten lyrically similar song.

```
[[1.         0.02303102 0.0513553  ... 0.01567518 0.03734159 0.        ]
 [0.02303102 1.         0.03742349 ... 0.01351745 0.00615388 0.        ]
 [0.0513553  0.03742349 1.         ... 0.0836582  0.03749053 0.        ]
 ...
 [0.01567518 0.01351745 0.0836582  ... 1.         0.02802071 0.00776975]
 [0.03734159 0.00615388 0.03749053 ... 0.02802071 1.         0.        ]
 [0.         0.         0.         ... 0.00776975 0.         1.        ]]
```

Fig. 5. Similarity matrix

The result of recommendation has improved phenomenally and these results can not be exactly placed under the analytical result or in other words the conclusions of such results can not be used as the accuracy of the system but rather used as an observational result.

E. *User Emotion Extraction*

The user can select from two options: either they can input the emotion by typing or the system would take the picture of the user and get emotion from the image. One of the major parts of any music recommendation system is to extract the user's emotion. In our system, the user's facial emotion can be considered to be an input to the recommendation system. We have extracted the real-time facial with the help of OpenCV and DeepFace facial feature recognition framework by Facebook. OpenCV [28] is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture, and analysis, including face detection and object detection. DeepFace [29] is a lightweight face recognition and facial attribute analysis (age, gender, emotion and race) framework for python.

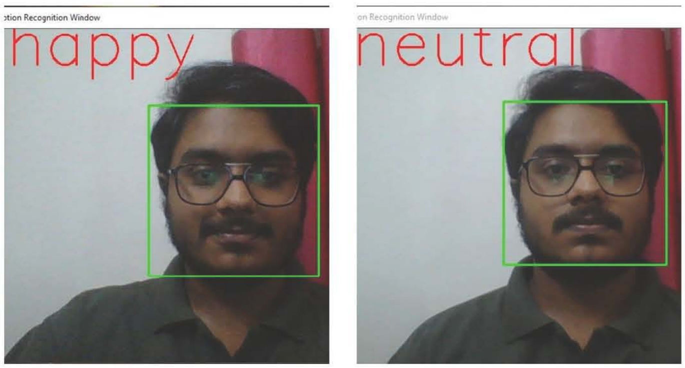

Fig. 6. Emotion extraction from Facial expression

We use the system's webcam to capture frames from the live feed as described in Fig. 6. These frames are taken as input for the user emotion recognition. Using DeepFace facial feature recognition framework the dominant emotion is extracted from the given frames. Therefore the extracted dominant emotion can be considered as the user's emotion. This emotion is used for recommending songs accordingly.

F. *Emotion Based Music recommendation system*

The implemented music recommendation system extracts the user emotion as mentioned above. The most prominent emotion is either directly obtained from the user input or by extracting the emotion from the image captured.

The implemented music recommendation system extracts the user emotion as mentioned above. The most prominent emotion is directly obtained from the user input or by extracting the emotion from the image captured. The collected emotion is passed and a song from the dataframe of the corresponding emotion is selected with the help of the sample function.

The user could simply just input the emotion as text or the the user could get recommendation by just clicking a picture which is demonstrated in Fig.7

The lyrically similarity function is called and a list of ten song is generated for the user and is saved as a comma separated values (csv) file, this file is updated each time the user tries to get a recommendation.

G. *Music Player GUI*

The main purpose of recommending songs is for listening to a particular group of songs that are having similar features or lyrical similarities. Hence to play music online or offline, a user needs a user interface. A simple music player user interface using Tkinter and Pygame module is made for the users to play songs offline. Dataframe with id, name and artists of the recommended songs were used to access the links of the songs in spotify. These links were obtained by adding the id at the end of the common track link of open spotify. With the help of the links and spotdl package these songs were made locally available within a folder which further was specified as the directory in the user interface. The described user interface is shown in Fig.8. The links that were prepared from the recommended songs can also be used to listen the songs online.

```
User Input mode :
1 - type in
2 - Facial emotion recognition
2
Saved to photo.jpg
```

```
happy
you selected  happy songs

Picking up  happy  song for you :  Things Change
recommendation of similar songs :
```

	id	name	artists
2031	3uqxwLp7Hs9sJhm4uVF8li	The Changeling	['The Doors']
4629	1iQ9wn7X6X9MRJVpVqLPC1	Modern World	['The Modern Lovers']
5715	1pXXriSh4lpOKFqsAWVdHe	The Modern World	['The Jam']
5714	1YK29h3DE8929r8r5uzECx	Modern Love	['Peter Gabriel']
5103	1vR0FJWxmazYPfU4lvkyAb	Don't Ever Change	['The Kinks']
5523	2bgW1SJqUjIkWcbxTPTIdy	Right By My Side	['Nicki Minaj', 'Chris Brown']
2405	6K8oEYCYwvEQWvq8hEDuDi	So It Goes	['Nick Lowe']
2448	7pU7xdjTwZyDW1XoZpV5Dl	Good Things	['Bodeans']
406	5EWPGh7jbTNO2wakv8LjUI	Free Bird	['Lynyrd Skynyrd']
2657	5DBmXF7QO43Cuy9yqva116	Family Business	['Kanye West']

Fig. 7. Sample recommendation with Facial Recommendation

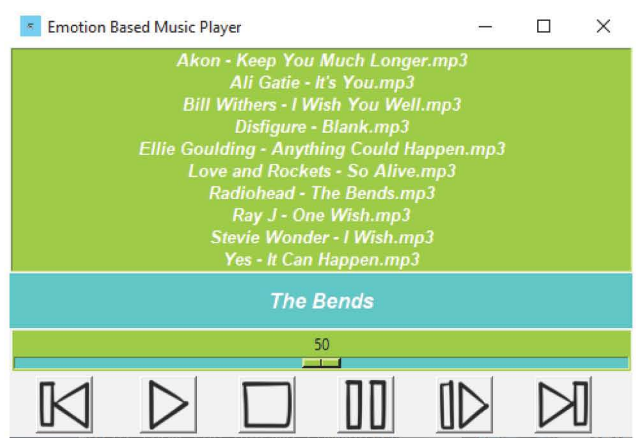

Fig. 8. Displaying the graphical user interface that runs locally to play recommended songs.

H. Further Addition of Songs

According to Spotify, the audio and podcast streaming app had more than 50 million songs during the beginning of the year 2019 and now has nearly 20 million more songs which adds up to 70 million songs in the year 2021. With this increase in the number of songs in a span of two years, the dataset that we have used to prepare the emotion based music recommendation system will become outdated. Identification of the emotion of the new songs can be easily done by parsing through its lyrics. By this way, the music dataframe will remain updated as new songs get released.

Subset of Million song dataset is used for predicting lyric's emotion. Training and test dataset were created and a dataframe was prepared. The main features required were File name, Artist Name, Song Title and Lyrics which was created empty at first. Those songs without having lyrics were dropped. Using the Artist Name and Song Title features, lyrics were fetched for all the songs using the PyLyrics package, which uses LyricWikia.com API to get the same as per the song name. The created model uses english lyrics as of now. Hence songs of any other language were removed from the Dataset. Consequently, tags were extracted from Last.FM API for remaining songs after data cleaning. A few groups were taken for Mood Categories namely happy, sad, angry and relaxed. Tags which were found from Last.FM and the tag groups generated before are correlated to create a class label for moods in the dataset. Dataframe was created with lyrics and mood after necessary additions and appendings. For testing purpose, an open source datset was used with the lyrics of 250 hindi songs prepared by a group of data enthusiasts. The songs were translated from other languages to english. Here, lyrics were taken from hindi bollywood songs and after translating them into english using Google Translate API, it was annotated into happy, sad, angry and relaxed categories, to check accuracy on the hindi songs. Words in the lyrics were further tokenized, stemmed, stopwords were removed and lemmatized on the dataset. The dataset was further divided into training label and data, testing label and data respectively. Since it is impossible to give input as a list of words in machine learning models, feature engineering was done on the lyrics column. Three types of NLP models were used for the same. At first, label encoding was done, later CountVectorizer model was used, then TfidfVectorizer model and finally Tfidf-NGram model was used. After this step there were three Training Dataset, three test dataset produced and ready to fit into the models. The final step was to create various mood prediction models using predefined ones such as Naive Bayes, Random Forest, Logistic Regression and XGBoost classifiers. Confusion matrices were prepared for each model and the model with highest accuracy was used to create two functions that test the model by predicting the emotion of the song by its lyrics. The workflow of the lyrics emotion extraction model is shown in Fig.9.

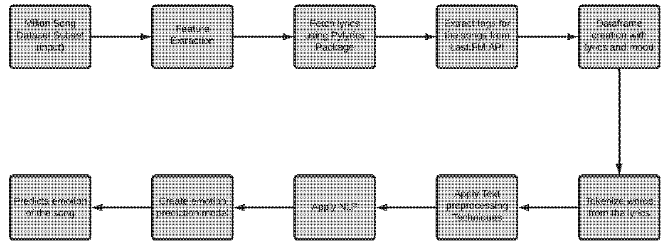

Fig. 9. High level Architecture of Lyrics Based Emotion Recognition Model

IV. Results

The recommendation system contains two conclusions as follows

A. Model Accuracy

The system contains an ensemble model which is XG-Boost and the accuracy obtained after selecting the optimum parameters ranges from 77% to 85%. The model that uses both music lyrics and emotion as the main features brings up more accuracy. This can recommend songs of the same emotion. Whereas the model which only uses music lyrics as the main feature may or may not recommend songs of the same emotion.

B. Observational result

All results that cannot be scaled nor be determined under any specific criteria but just can be concluded to understand are observed in this section. This section also emphasizes the true necessity of emotion in the recommendation system. As we observed that when a user has mentioned the emotion either in text or extracted from the picture, the person gets a song followed by 10 lyrically similar songs which also have the same emotion. The same song when run on the whole dataframe without considering emotion would also give 10 lyrically similar songs, this time the songs are of a different emotion. For a person to get mood swings when subjected to such variation in song emotion when the user consumes it more often and in a longer period.

V. Conclusion and Future Scope

Emotion based music recommendation system is an innovative model prepared for recommending songs to users based on their moods. Right music at the right time helps listeners to enlist their soul or to empty bad thoughts. Also, it identifies that music that congruent with your mood will bring a feeling of the presence of an empathetic friend. Hence, this concept of using music to cure depression inspired us to develop this emotion based music recommendation system. The system takes the user's emotions either by directly asking them or by capturing their facial image and analyzing the mood with the user's approval. According to the emotion of the user at that specific moment, it recommends 11 songs, which include a random selection of one song with similar emotion and 10 songs with lyrical similarity. The music dataframe with emotions tag was prepared with the help of a supervised learning model such as Random Forest and XGBoost that used an open source dataset for its training. Also, since a vast amount of songs gets added every day, a NLP model was prepared to analyze the lyrics of such new songs and to predict their emotion so that we can add them later to the existing song collection. Though recommendations of songs end with providing a list of songs to the user, the ultimate goal for them is to listen to them. A user can either open the songs one after the other in Spotify using the links generated or can use spotdl to download these songs into a folder and then listen to them locally using the python GUI prepared. This approach is a fully-fledged solution that can be incorporated in any music player to recommend songs following the user's emotion. But like we saw chances for innovations on existing systems, there is always a scope to improve the existing model in the future. It includes the addition of collaborative filtering and its respective features to prepare the model, preparation of user database and emotion charts according to a user history [30], use of mel spectrogram, including time and place as factors to recommend songs, and finally preparation of the whole music player website or android app that uses our recommendation system.

REFERENCES

[1] B. S. Pereppadan, "In india, doctors see a surge in depression, insomnia cases," *The Hindu*, 2020. [Online]. Available: https://www.thehindu.com/news/national/in-india-doctors-see-a-surge-in-depression-insomnia-cases/article32774060.ece

[2] R. C. Keynejad, T. Dua, C. Barbui, and G. Thornicroft, "Who mental health gap action programme (mhgap) intervention guide: a systematic review of evidence from low and middle-income countries," *Evidence-based mental health*, vol. 21, no. 1, pp. 30–34, 2018.

[3] G. Gerasimovich and V. Sidorenko, "Stress reducing effects of medical resonance therapy music–studies presented at the world health organisation conference on society," *Stress and Health, Moscow*, vol. 1995, pp. 96–7, 1995.

[4] C. Lee, E. Andrade, and S. Palmer, "Interpersonal relationships and preferences for mood-congruency in aesthetic experiences," *Journal of Consumer Research*, vol. 40, 04 2013.

[5] D. Paul and S. Kundu, *A Survey of Music Recommendation Systems with a Proposed Music Recommendation System*, 01 2020, pp. 279–285.

[6] S. .S, K. Ramar, A. Omanakuttan, and A. Sasidharan, *Automated Analytical Model for Content Based Selection of Web Services: 4th International Symposium SIRS 2018, Bangalore, India, September 19–22, 2018, Revised Selected Papers*, 01 2019, pp. 309–321.

[7] C. Dhahri, K. Matsumoto, and K. Hoashi, "Mood-aware music recommendation via adaptive song embedding," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2018, pp. 135–138.

[8] "Types of listening," *Changing Minds*. [Online]. Available: http://changingminds.org/techniques/listening/types_listening.html

[9] "Overcoming bias about music takes work - uconn today," *UConn Today*. [Online]. Available: https://today.uconn.edu/2018/04/overcoming-bias-music-takes-work

[10] R. Rosa, D. Z. Rodriguez, and G. Bressan, "Music recommendation system based on user's sentiments extracted from social networks," *IEEE Transactions on Consumer Electronics*, vol. 61, pp. 359–367, 08 2015.

[11] D. Ayata, Y. Yaslan, and M. Kamasak, "Emotion based music recommendation system using wearable physiological sensors," *IEEE Transactions on Consumer Electronics*, vol. 64, pp. 196–203, 2018.

[12] A. Dureha, "Article: An accurate algorithm for generating a music playlist based on facial expressions," *International Journal of Computer Applications*, vol. 100, no. 9, pp. 33–39, August 2014, full text available.

[13] A. Abdul, J. Chen, H.-Y. Liao, and S.-H. Chang, "An emotion-aware personalized music recommendation system using a convolutional neural networks approach," *Applied Sciences*, vol. 8, p. 1103, 07 2018.

[14] S. Gilda, H. Zafar, C. Soni, and K. Waghurdekar, "Smart music player integrating facial emotion recognition and music mood recommendation," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 154–158.

[15] F. Zhang, H. Meng, and M. Li, "Emotion extraction and recognition from music," 08 2016, pp. 1728–1733.

[16] W.-P. Lee, C.-T. Chen, J.-Y. Huang, and J.-Y. Liang, "A smartphone-based activity-aware system for music streaming recommendation," *Knowledge-Based Systems*, vol. 131, 06 2017.

[17] A. V. Iyer, V. Pasad, S. R. Sankhe, and K. Prajapati, "Emotion based mood enhancing music recommendation," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, 2017, pp. 1573–1577.

[18] S. Chandran and S. Elango, *An Efficient Context-Aware Music Recommendation Based on Emotion and Time Context: ACM-WIR 2018*, 01 2019, pp. 215–228.

[19] Cristobalvch, "cristobalvch/spotify-machine-learning." [Online]. Available: https://github.com/cristobalvch/Spotify-Machine-Learning/blob/master/data/data_moods.csv

[20] A. Neisse, "Song lyrics from 6 musical genres," Nov 2019. [Online]. Available: https://www.kaggle.com/neisse/scrapped-lyrics-from-6-genres?select=lyrics-data.csv

[21] Y. E. Ay, "Spotify dataset 1922-2021, 600k tracks," Apr 2021. [Online]. Available: https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks

[22] "The musixmatch dataset." [Online]. Available: http://millionsongdataset.com/musixmatch/#getting

[23] P. Devika, R. C. Jisha, and G. P. Sajeev, "A novel approach for book recommendation systems," in *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2016, pp. 1–6.

[24] S. K. Thangavel, P. D. Bkaratki, and A. Sankar, "Student placement analyzer: A recommendation system using machine learning," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2017, pp. 1–5.

[25] A. Ajesh, J. Nair, and P. S. Jijin, "A random forest approach for rating-based recommender system," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016, pp. 1293–1297.

[26] M. Khanam, T. Mahboob, W. Imtiaz, H. Ghafoor, and R. Sehar, "A survey on unsupervised machine learning algorithms for automation, classification and maintenance," *International Journal of Computer Applications*, vol. 119, pp. 34–39, 06 2015.

[27] K. S. Nikitha and G. P. Sajeev, "An improved rocciho algorithm for music mood classification," in *2018 International Conference on Data Science and Engineering (ICDSE)*, 2018, pp. 1–6.

[28] G. Bradski and A. Kaehler, "Opencv," *Dr. Dobb's journal of software tools*, vol. 3, 2000.

[29] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," 2015.

[30] R. C. Jisha, R. Krishnan, and V. Vikraman, "Mobile applications recommendation based on user ratings and permissions," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018, pp. 1000–1005.