Emotion Based Music System

Soham Sheth†
Computer Science
Michigan Technological University
Michigan USA
ssheth1@mtu.edu

Vrushali Shinde[2]
Computer Science
Michigan Technological University
Michigan USA
vshinde2@mtu.edu

Tejaswi Chintapalli[3]
Data Science
Michigan Technological University
Michigan USA
tchintap@mtu.edu

*All authors contributed equally to this research.

Abstract

Sentiment analysis or opinion mining is to identify the nature of text based on Natural Language Processing (NLP). Emotion detection is a part of sentiment analysis to detect the state of human emotion through analyzing the text or opinions. On the other hand, listening to music reduces anxiety, eases pain, and improves mental alertness. In this project, we explored human emotions by applying various machine learning techniques, performed sentiment analysis, predicted the emotion from the text, and suggested a playlist to the user which stimulates oxytocin to have positive and happy feelings. This paper presents different approaches used for emotion detection and recommended music systems. Numerous machine learning techniques like Multinomial Naive Bayes (MNB), Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree Classifier, XGB Classifier, and Random Forest Classifier are often used to predict the emotion once the text is processed with the help of Natural Language Toolkit (NLTK). Ensemble machine learning approach Random Forest Classifier outperforms all other techniques in the case of predicting human emotion.

Keywords— SVM(support vector machine), K-Nearest neighbors, Random Forest, Multinomial naive Bayes, XGB Classifier, Logistic Regression, Decision tree.

1 Introduction

People use music to achieve different goals such as recharging their batteries, focusing on their tasks, and reducing boredom. For example, sad music allows the listener to move away from a stressful situation (breakup, death, etc.) and focus instead on the beauty of the music. Additionally, lyrics that reflect the listener's personal experience can express feelings and experiences that they cannot express themselves. The proposed system helps provide an interaction between the user's emotional perception and the music it recommends for that emotion, which is a good reason for motivation. Music also can evoke strong emotional responses such as chills and thrills in the listener. Music evokes positive emotions. Soothing music can lead to the release of reward-related neurotransmitters such as:

B. Dopamine. Listening to music is an easy way to change your mood and relieve stress. People use music in their daily lives to regulate, amplify, and reduce unwanted emotional states (stress, fatigue, etc.). How does listening to music create emotions and joy in the listener? Listening to music also has an intellectual component. The dopamine system does not work in isolation and its effects are highly dependent on interactions with other areas of the brain. Thus, our ability to enjoy music can be seen as a result of the human emotional brain and the recently evolved neocortex. There is evidence that people who consistently emotionally respond to aesthetic musical stimuli have greater white matter connectivity between the auditory cortex and areas associated with emotional processing. Music is so closely related to our brain that this project attempts to study the same by creating and analyzing small models. The project takes a dataset stored in the form of a CSV file containing text associated with IDs, processed by various machine learning algorithms to read the text and extract sentiment from it. The same sentiment is compared to another data set and the corresponding songs are suggested. We also discuss analyzing the accuracy of machine learning algorithms.

2 Motivation

Artificial intelligence and machine learning plays a key role in any recommendation based applications. Music recommendation system is one of the few applications. Music plays an important role in everyone's life. Changing music continuously can most of the time break the momentum of one individual or a group of people so here we tried to build a music recommendation system based on emotions.

The motive behind the project was to build a fun game where the system tries to read the emotions of the people from the text, message or tweets which they write. The model will first read the message then try to predict the emotion based on the text, further it tries to find the genre related to that text and lastly when the genre is selected the machine will recommend some music, songs based on that genre.

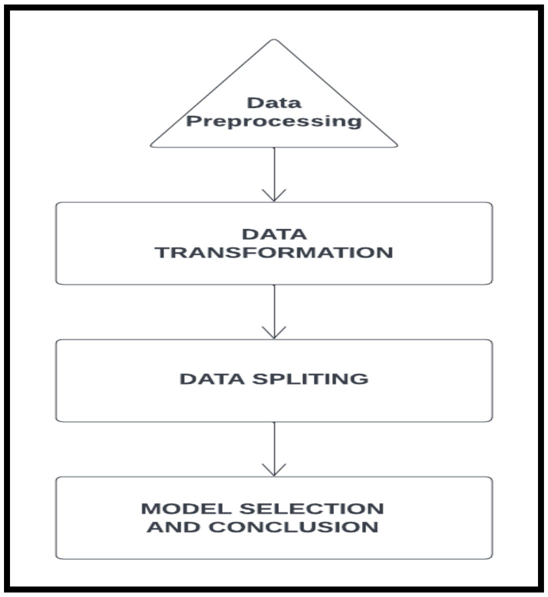

Figure 1: Methodology

Exploratory Data Analysis:

Data preprocessing is an essential part of Machine learning. It helps clean the historical data and transform it into a system-understandable format. A data set may consist of integers, text, images, videos, voices, etc. Missing information, duplicate values, outliers, skewness, and redundant information are present in this data. We get good predictions for our models when we process data well. Preprocessing also includes visualizations that help analyze the patterns, consistency, and trends.

Any AI and machine learning project begins with data collection. We get historical data from different sources. Several mini-projects get their data from online sources, including Kaggle, GitHub, and others. Also, we collect data from websites, sensors, APIs, social media monitoring, interviews and surveys, Transaction tracking, online tracking, and forms.

Data Set:

This project's data came from Kaggle. We need two datasets to work on this project. Emotion Detection Sentiment Analysis and music data set.

There are 40000 records and three columns in the Emotion Detection Sentiment Analysis dataset. The three columns are emotion, tweet_id, and content. Since it is a text dataset, we used natural language processing to analyze it. For the music data set, we have 12 records and three columns. The three columns are emotion, genre, and song.

Due to the lack of information online, we created music data.

Imbalance data set:

The bar plot below shows the distribution of the target variable. Knowing the distribution of the dependent variable is crucial because this might cause us biased performance on our model.

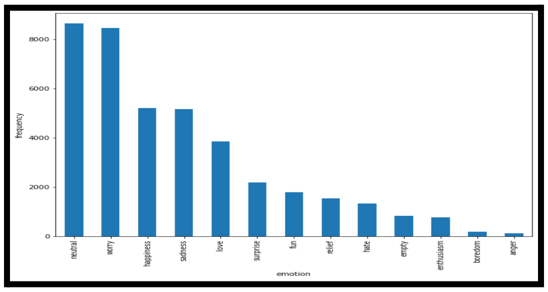

Figure 2 : Imbalance dataset

Missing data:

Blanks, null values, and errors are present in missing data. There are no blank values in our data. We can't forecast our feelings even when we have access to noisy information like usernames, punctuation, and stopwords like I, they, them, the, etc. So we took them out.

Duplicates:

When several identical data entries are present, there is data duplication. It may be the same row or column. There is a possibility that inaccurate performance estimates will result from having duplicate entries. Our dataset does not contain any duplicates.

Noisy data:

Remove any unnecessary information that doesn't correlate with our response variable. Tweet_id in our data set has unique values and doesn't help us predict our outcome. This result in the removal of the tweet_id column.

Punctuations:

The technique of removing punctuation will aid in treating each text equally. We eliminate punctuation such as !,.=":#$%^&*(){}[]|\ and other similar symbols when predicting emotions. With the help of regex, we can recognize these punctuation marks and eliminate them from the text. For instance, 'really' and 'really!' mean the same thing.

Stop word:

Stop words (I, they, they, and the) are common in all human languages. Our text becomes more concise by removing these terms, which helps the important information stand out. To eliminate stop words from our text, we use NLTK.Corpus.

Stemming and Lemmatization:

Cutting the final few letters from a word produces precise spelling and semantics by stemming. Talk, for instance, results from talking. Lemmatization is used to construct words' lemma or basic form. for example, eaten into eating.

Data Transformation:
Data transformation is changing the cleaned data into an NLP understandable format. It uses methods like vectorization in which it considers each word as a token and represents it as a sparse matrix. The project uses two methods for transforming

Countvectorizer
The project uses count vectorizer because we have observed here that since dataset does not contain much attributes, we thought using this technique in a way that each word in the text it will consider as index pair and store in matrix and then this is how we can have multiple variables.CountVectorizer is a amazing device supplied via way of means of the scikit-analyze library in Python. It is used to convert a given textual content right into a vector on the idea of the frequency (count) of every phrase that takes place withinside the complete textual content. This is beneficial while we've got more than one such text, and we want to transform every phrase in every textual content into vectors (for the usage of in addition textual content analysis).
eg : a content text from the dataset says **"want hang friend soon**" after cleaning the actual sentence "wants to hang out with friends SOON!". If we apply count vectorizer to this we get ::

	friend	hang	soon	want
tweets['new_content'][0]	1	1	1	1

The tweets['new_content'][0]contains 4 unique words displayed as columns in a table.
The cell contains a number representing the number of words in that particular text.
All words converted to lowercase. The words in the columns are ordered alphabetically.
As said, the words are not stored as string but given index value.

0	1	2	3
1	1	1	1

The process is carried out for the entire content array in tweets.

TFIDF transformer
It is at this factor that we used TfidfTransformer to normalize the matrix, weighting it with diminishing significance of tokens that arise withinside the majority of documents.

Data Splitting:
Data splitting is commonly used in machine learning to avoid overfitting. In this case, the machine learning model is too well-fitted to the training data and certainly not the additional data.
The original data for a machine learning model is typically taken and split into three or four sets. Three commonly used sets are the training set, development set, and test set.
A training set is a portion of the data used to train a model. The model should be learned by optimizing each parameter and observing the training set.
Also called cross-validation set or model-validation set. This dataset is intended to rank the accuracy of models and is useful for model selection.
The test set is the portion of the data that will be tested with the final model and compared to the previous dataset. The test set serves as the final mode and algorithm evaluation. The project here does the following

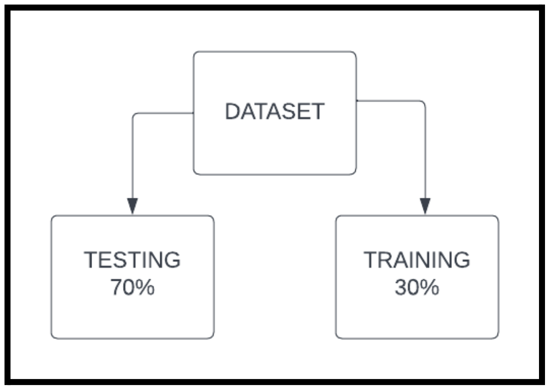

Figure 3 : Data Splitting

After data splitting, the project is made to test with different machine learning algorithms to determine the best accuracy in a way that we can say that yes, this is the best accuracy giving algorithm.

3 Model Selection

The overview of how is the flow of the project's trial basis of various ML algorithms. Note that the dataset contains categorical data ie. algorithms like linear regression cannot be helpful here hence being basic not used.
Using a CountVectorizer, we split our tweets into words and create an index of each word in order to predict sentiment based on that word.
Next it creates a pipeline object which will be used throughout this program to process our data.
It then splits the training data into two parts: one part for training the model, one part for testing
Multinomial Naive Bayes::
The project uses fit() from scikit-learn's Pipeline class to fit MultinomialNB(), which predicts sentiment based on words in text documents given their frequency of occurrence within those documents. Finally it prints out classification report(y_test) with predicted sentiment values along with confusion matrix(y_test)

showing how accurate predictions are at different levels of confidence.The vectorizer to transform each row in the train set into a vector of values that are then fed into a MultinomialNB classifier which predicts what emotion each tweet contains.

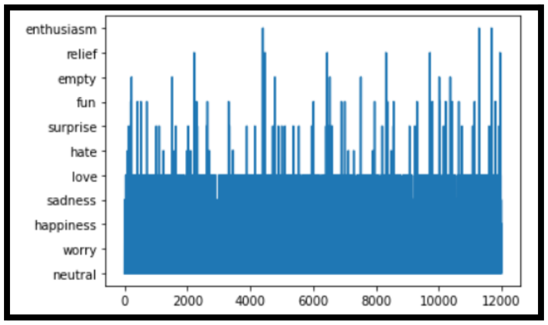

Figure 4 : Multinomial Naive Bayes

Logistic regression::
The TfidfVectorizer object with an analyzer set to word and max_features set to 1000. Next, it fits a LogisticRegression model on the training data using X_train as input and y_train as output.
Finally, it predicts the y_test given x_val_tf which is created from fitting X on test data using tf.fit().

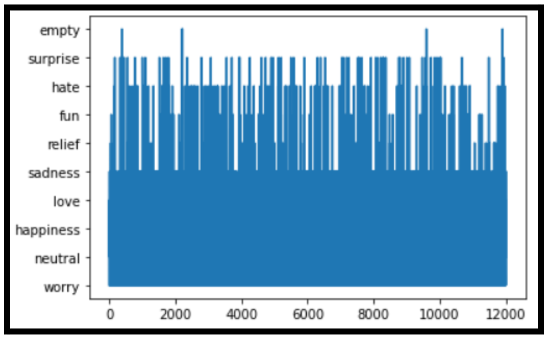

Figure 5 : Logistic Regression

For SVC and decision tree classifier the project did:
Creating a CountVectorizer object with C=1 (the default value).Then it transforms all of its inputs into vectors that are 1-grams long (meaning they only have one element) so that they can be easily classified later on.Next, it trains an MLP classifier on this transformed vectorized dataset with train/test split at 0.3 ratio each time through while shuffling them randomly every time through so that there's no bias towards any particular outcome or classification scheme when making predictions about future outcomes based on past outcomes.

Decision Tree Classifier::

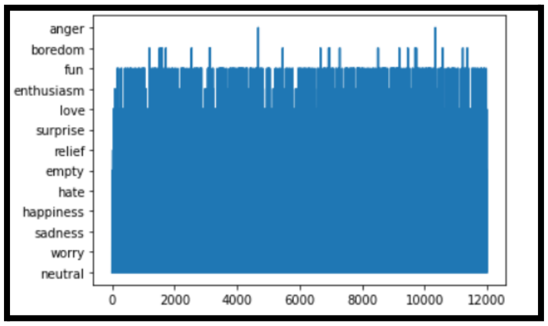

Figure 6 : Decision Tree Classifier

SVC - Support Vector Machine Algorithm based::

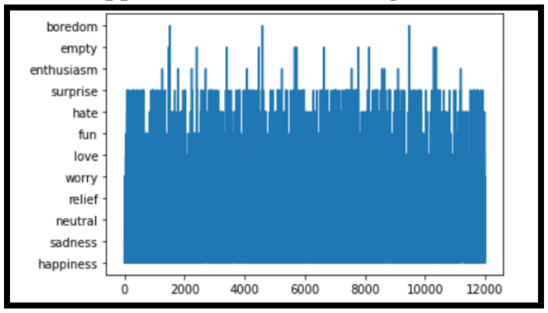

Figure 7 : Support Vector Machine

KNN::
Creates a list of all possible values for k_range, which is between 1 and 15.Next, it creates an instance of KNeighborsClassifier with n_neighbors=11. The fit method will use this model to predict y given x in order to get a prediction value that can be used as input into the next step: classification report.

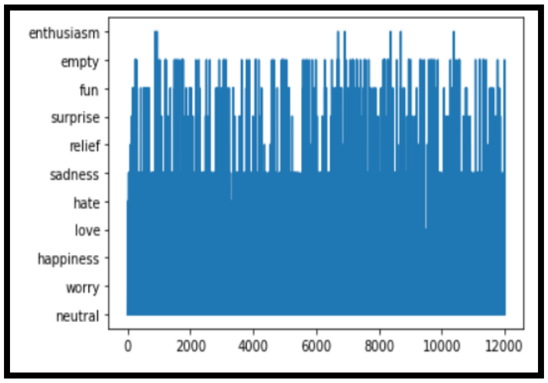

Figure 8 : KNN

XGB Classifier::
We create an XGBClassifier object and call its fit method on our training data set (X_train) using our test data set (y_test).

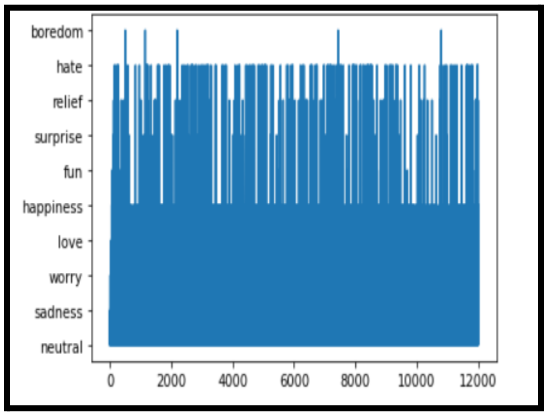

Figure 9 : XGB Classifier

Random Forest Classifier::
Random forest uses n estimators value which is nothing but how many decision trees we need to consider before we predict our best results. Also cross validation is applied to select and choose the best model.

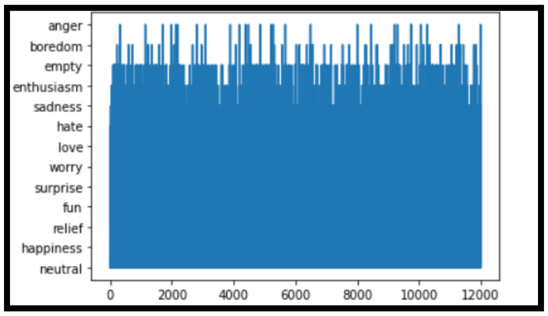

Figure 10 : Random Forest Classifier

FACTORS ON WHICH THE ANALYSIS OF ALGORITHMS IS DONE ::
There are four factors on which we try to analyze our algorithm they are as follows :
1) **F1-score** :It is employed to assess how well binary categorization performs. It is typically employed to evaluate the effectiveness of two classifiers. By calculating their harmonic means, it integrates the precision and recall into a single statistic..
F1 score = (2*precision*recall)/(precision + recall)

2) **Precision** : The precision is the ratio tp / (tp + fp) where tp is true positives and fp is false positives. Intuitively, the classifier's precision is its capacity not to classify a negative sample as positive. 1 is the best value, while 0 is the worst.
Precision : Precision = TruePositives / (TruePositives + FalsePositives)

3) **Accuracy** : One way to gauge how frequently a machine learning classification algorithm classifies a data point correctly is to look at the algorithm's accuracy. The proportion of accurately predicted data points among all the data points is known as accuracy.
Accuracy = TP + TN / TN + TP + FN + FP

4) **Recall** : The recall is determined as the proportion of Positive samples that were correctly identified as Positive to all Positive samples. The recall gauges how well the model can identify positive samples. The more positive samples that are identified, the larger the recall
Recall = TruePositives / (TruePositives + FalseNegatives)

4 Analysis Of The Applied Algorithms

The reason why we went for the below given options is because we have categorical data and are familiar with the work of each and how one algorithm outperforms the other algorithm.

Multinomial Naive Bayes::
Naive Bayes performs well when it works with text classification and several classes. It is very straightforward but as it holds one basic assumption to be true i.e predicted features can be independent of the classification variables and that is the reason it is called "Naive" to give high accuracy. However, the project wants to relate the content text and then depict the emotion based on it assuming that content text and the predicted emotion is dependent on each other. Hence, we infer this might be the reason for the accuracy not being up-to the mark.

Logistic regression::
Logistic regression works in a way that it predicts the discrete feature/observation. This can be more explained in a way that it works the best with outputs like true/false, yes/no i.e binary outcome. The content text in our dataset does contain some text which might be dicey in order to decide between two emotions and hence, we infer that this might be the reason for the accuracy to not be that much comfortable. For eg consider the text "happy mother day love" this can be love or happy emotion.

Decision tree classifier::
This one was tricky as we wanted to use a decision tree, but then in a single one we might have issues like instability, sensitivity and sometimes overfitting as well. Thus to keep the root the same the project also went for random forest as that being more strong modeling technique and robust.

Support Vector Machine::
The size of the dataset greatly affects the accuracy of this algorithm as the data becomes complex and the training complexity of SVM is very high. We do have around content text in multiple thousands and so it can be the possible reason why we think that SVM is not behaving that accurately.

KNN::

The KNN was among the one which we planned to consider in the race but we decided to keep on hold for the following reasons:

- Every time reading was very time consuming as we needed to calculate the best K value and hence it might lead to computational cost as well when used in real time.
- No training period is mandatory and any new data can be added at any time and that leads to sensitivity of the model as the second thought.
- When it comes to large dataset, in order to calculate the distance between all points and try to cover all the distance, if we keep on increasing the k value i.e number of clusters then that might potentially lead to overfitting which any model should avoid.

For these various reasons, despite giving second close accuracy or even the best if the value of K is manipulated, we decided to not go with this algorithm.

XGB Classifier::

XGB classifiers tend to avoid the scenarios like features which are generated using BOW (Bag of Words) model and tf-idf transformer which we exactly used in the data transformation for creating a sparse matrix. The reason we used it is because we do have less attributes to perform the experiments; so we thought of using a count vectorizer which will consider each word as a separate variable and then represent itself to perform ML algorithms on it.

Random Forest Classifier::

To start with, this algorithm we believe does not have any of the above concerns. Secondly, it gave us the best accuracy amongst all the tried ones. Yes, it does have disadvantages like requiring more time, not suitable for continuous data, very sensitive to small changes in data but above all it gives best accuracy. Random forest is something in an informal way like it applies multiple decision tree algorithms to the model before choosing the best one. Hence, we believe it's better than a single decision tree and can be more robust and one will overcome other's issues and ultimately result in the chosen one.

Now that we have analyzed all the algorithms, here's the summary of what the project have:

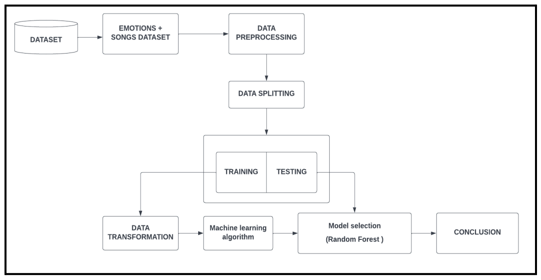

Figure 11: Emotion Based music recommendation System

Table 1 : Results and analysis of implemented model

Models	**Accuracy**
MultiNomial Naive Bayes(NB)	32 %
Logistic Regression	35 %
Support Vector Machine (SVM)	34 %
K-nearest neighbors	33 %
Decision Tree classifier	27 %
XGB Classifier	33 %
Random Forest	98%

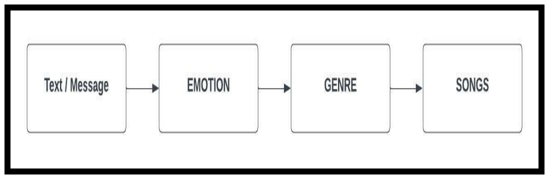

Figure 12 : Emotion based music system

5 Acknowledgement

We would like to thank our professor Prof. Wenbin Zhang and Teaching Assistant, Zhenlin Wang for the valuable support and guidance they gave us on every step of the project execution and we tried our best to adhere to what they gave as input. We would also like to thank the authors of the references mentioned and the websites we referred to which helped us accomplish this work.

6 Conclusion

In this project, the emotion was predicted based on the content text and the song was suggested based on the emotion. We used the trained machine learning model to test the incoming test and the split percentage was 70-30% Experimental methods included the study of the used machine learning algorithms and how categorical data behaves with each. The project made us learn how practically machine learning algorithms work and how important are artificial intelligence concepts. One of the roadblocks we faced is the less number of attributes and while running the model, overfitting, time consuming and that we used one custom made dataset. There is no proven theory that always in such a case of dataset random forest

will work the best and that we cannot conclude a strong statement like so. Moreover we do understand that if the
attributes were huge in number the output would have turned out differently. However, the initial intention was to pick a sample dataset from the kaggle dataset and then even if the number of attributes are less, we did use functions like count vectorizer to increase the number of considerable variables and then furthered the project for predicting the result. The project also analyzed as to why every single assumption or formula of the algorithm and metrics can affect the overall accuracy of the model. The project is unique in a way that the text used is just a go to simple sentences and that we try to study the behavior mood of the individual sentence and then suggest songs to uplift it. The later implementation we think can be done on social media with more use of real time automation. When every person types something on a story on instagram, facebook or twitter may it be birthday wishes or congratulations messages, our model can detect the message and suggest a song according to the caption as per the story or the post one makes. It can be time saving in some sense and you can get all the songs related to that text without searching or finding as the recommendation list will help you with the songs. The future more could be how depending on the heartbeat i.e ECG data of a human, i.e. if there is a rush of nervousness or excitement we can count the heartbeats to predict emotion based on the pattern and suggest songs. With the further improvement we can also try to change the dataset and have more attributes and try more different algorithms.

7 References

[1] Y. Kodama et al., "A music recommendation system," 2005 Digest of Technical Papers. International Conference on Consumer Electronics, 2005. ICCE., 2005, pp. 219-220, doi: 10.1109/ICCE.2005.1429796.
[2] Ankita Mahadik , Shambhavi Milgir , Janvi Patel , Vijaya Bharathi Jagan, Vaishali Kavathekar, 2021, Mood based Music Recommendation System, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 06 (June 2021).
[3] C. -F. Huang and C. -Y. Huang, "Emotion-based AI Music Generation System with CVAE-GAN," 2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), 2020, pp. 220-222, doi: 10.1109/ECICE50847.2020.9301934.
[4] M. -J. Yoo and I. -K. Lee, "Affecticon: Emotion-Based Icons for Music Retrieval," in IEEE Computer Graphics and Applications, vol. 31, no. 3, pp. 89-95, May-June 2011, doi: 10.1109/MCG.2011.36.
[5] Erion Çano, "Text-based Sentiment Analysis and Music Emotion Recognition"
[6] Science Direct http://www.sciencedirect.com
[7] IEEExplore http://ieeexplore.ieee.org
[8] Ankita Mahadik , Shambhavi Milgir , Janvi Patel , Vijaya Bharathi Jagan, Vaishali Kavathekar, "Mood based Music Recommendation System "
[9]https://www.kaggle.com/code/rishabh2007/emotion-detection-sentiment-analysis-2-4
[10] AI based Music Recommendation system using Deep Learning Algorithms R Anand1, R.S Sabeenian1, Deepika Gurang1, R Kirthika1 and Shaik Rubeena1
[11] Hemanth P,Adarsh ,Aswani C.B, Ajith P, Veena A Kumar, "EMO PLAYER: Emotion Based Music Player", International Research Journal of Engineering and Technology (IRJET), vol. 5, no. 4, April 2018, pp. 4822-87
[12] Renata L. Rosa, Demóstenes Z. Rodríguez, and Graça Bressan, "Music Recommendation System Based on User's Sentiments Extracted from Social Networks"