MOOD CLASSIFICATION OF SONGS BASED ON LYRICS

A Project Report

Submitted by:

KALAGEE ANJARIA (AU1741052)
NIDHI GOLAKIYA (AU1741062)
RAJVEE KADCHHA (AU1741064)
RAJVI PATEL (AU1741078)

at

Ahmedabad University

School of Computer Studies (SCS)
Ahmedabad, Gujarat

Table of Contents

1. INTRODUCTION

1.1 Problem Definition :

- Increasing work pressure denies the necessary time to listen and evaluate music for a creation of a personal music library.
- Thus, if we can develop a system where the songs are classified into different moods, people can directly choose that mood of songs they want to listen to.
- Mood classification is the process wherein the emotions of a song are identified using various means which includes it's lyrics.
- Sentiment analysis refers to the use of natural language processing and text analysis.
- By analysing and training the lyrics of different songs, we can classify them into different classes of moods.

1.2 Project Overview/Specifications :

- The aim of this project is to classify the mood/sentiment of hindi songs from their lyrics.
- We are going to classify a song by training and testing the lyrics into different classes of moods.
- The moods we are going to classify the songs into are :
    - Romantic
    - Sad
    - Party
- For classification of songs, we have used different inbuilt Machine Learning classifiers available in the sklearn library of Python
- We are also considering the artists' names as the input feature for the classifiers along with the song lyrics.
- The flow of modules in which we are going to achieve this aim is :

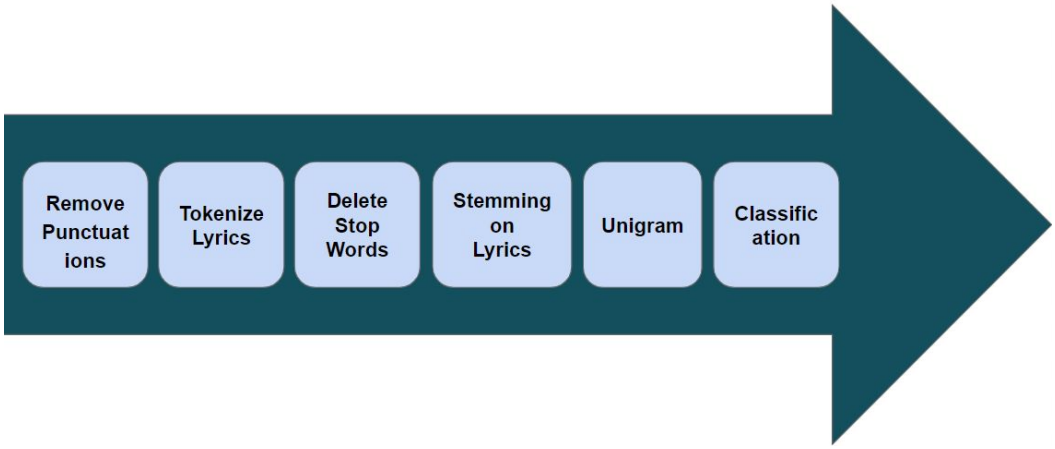

Figure 1 : Workflow of our project

2. LITERATURE SURVEY

2.1 Existing Systems :

- In the last few decades,many research work has been carried out to classify mood of the song. But, most of them are done using features such as rhythm, and intensity.
- And very few exist where classification is done based on lyrics, because it is seen that lyrics do not always correlate much with the mood of the song. Most of the work is for English songs.
- But for hindi songs, we have come across one article which is **Mood Classification of Hindi Songs based on Lyrics by-Braja Gopal Patra, Dipankar Das, and Sivaji Bandyopadhyay.** [1]
- They have first decided the polarity of songs based on positive and negative and then they have subclassified it into different moods which are angry, disgust, fear, happy, sad, surprise, excited, astonished, calm and relaxed.

- In this article, for classifying the songs based on their lyrics, the following steps are followed :
    - Transliteration of dataset into hindi by EILMT(English to Indian Language Machine Translation) tool.
    - They have trained and classified based on WEKA-which is ML tool to train and then further classify.

- Another article that we referred to was : **Mood classification of songs based on lyrics. By Francesco Cucari, 732A92 Text mining - Linkoping University. (For English songs)** [2]
- In this article, for classifying the songs based on their lyrics, the following steps are followed :
    - Downloaded the dataset from a reliable source.
    - Filtered the dataset by removing non-English songs using the NLTK python library.
    - Tokenize the lyrics into words or sequence of words(n-grams).
    - Stemming : Porter stemmer, Lancaster stemmer, Snowball stemmer
    - Used TfidfVectorizer from NLTK library for feature extraction
    - Classify the text by analysing the different classifiers available in the library.

2.2 Proposed System :

- In our project, we are going to classify Hindi songs. And Hindi as in, the lyrics are in the Hindi language itself, and are not Hindi to English transliterated.
- So first of all, we have manually created our dataset consisting of 793 Hindi songs, along with their artists' names.
- We have tagged the songs as romantic/sad/party of our dataset according to our classes. It was very difficult to classify the song as lyrics of one song may contain

mixed emotions.

- After that, we have removed all the punctuation marks from the lyrics.
- We have manually identified the stopwords in the context of Hindi lyrics and created a list for the same.
- Then we have written a code for removing them from the dataset.
- Then, we have applied stemming on the lyrics.
- After that, we have collected all these stemmed words obtained from all the songs and added them as features in our dataset.
- The songs which contain the particular word, have the value '1' for the feature of that word, otherwise the value '0'.
- This is done for all the songs and all the word features obtained.
- Then, different inbuilt classifiers are applied on this dataset and their accuracies are compared. The classifiers applied are :
    - Decision tree classifier
    - Naive Bayes classifier
    - Random Forest classifier
    - Support Vector Machine
- We have tried to develop a frontend website for our project. The functionalities that it provides are :
    - You can play the songs, also go to previous and next songs.
    - You can see the youtube video of the corresponding song.
    - You can create a playlist of your favourite songs.
    - And, the songs in your playlist are classified into the three categories of moods : romantic, sad, party.
    - Also, we've inserted a button for taking audio as what you say. Like if you say the song name, it will fetch the lyrics from Google and classify it. This is not accomplished yet but can be considered as future work.
- The lyrics and the artists' names are sent to the Python file. Then the lyrics will be preprocessed, converted to unigrams, and then are classified.

2.3 Feasibility Study :

- As we know, most of the accomplished work in song mood classification is based on the features related to audio signals.
- But, according to some research, the text based and the tune based information are processed individually in our brains. Even though both are closely related.
- Thus, just exploring the text based (lyrics) information for the classification of songs is relevant.
- And as this has not been widely explored in the literature, this study has been important. [2]

3. SYSTEM ANALYSIS & DESIGN

3.1 Algorithms :

3.1.1 Removing Punctuations:

Step 1: Load the dataset of hindi songs.

Step 2: Take just lyrics from the dataframe.

Step 3: In the for loop take one lyrics at a time until all the lyrics are covered

Step 4: Call the function “Remove Punctuations” , pass one lyrics as an argument.

Step 5: Inside “Remove Punctuations” : Replace punctuations with empty String using “re” library.

Step 6: Return string without punctuations.

Step 7: Replace string without punctuation in dataframe.

Step 8: Go back to Step 3.

3.1.2 Identification of Stopwords:

Step 1: Take dataframe with removal of punctuations.

Step 2: For loop for taking one input at a time.

Step 3: Tokenize lyrics.

Step 4: Do frequency distribution on tokenized lyrics.

Step 5: Take most common 25 words into consideration.

Step 6: If the word can be said stopword then add it into the list. (Making sure of not removing ना , नहीं etc.)

Step 7: Go back to Step 2.

Step 8: Make this list a “set” for removing duplicates.

Step 9: Write this set into a text file “stopwords.txt”.

3.1.3 Removal of Stopwords

Step 1: Take dataframe with removal of punctuations.

Step 2: For loop for taking one input at a time.

Step 3: Tokenizing the lyrics. (one lyric at a time.)

Step 4: For loop for taking one word of tokenized lyric.

Step 5: Check if the current word exists in the stopword file.

Step 6: If yes then do not consider it.

Step 7: If no then append it in the string without stopword.

Step 8: Go back to Step 4.

Step 9: Write this string without a stopword into the dataframe.

Step 10: Go back to Step 2.

3.1.4 Stemming :

Step 1: Make a dictionary with suffixes in hindi.

Step 2: Take dataframe with removal of stopwords.

Step 3: For loop for taking one input at a time.

Step 4: Tokenizing the lyrics. (one lyric at a time.)

Step 5: For loop for taking one word of tokenized lyric.

Step 6: Check the length of the current word. If it is greater than suffixes.

Step 7: If yes then check every suffix whether the word ends with given Suffix. And return the word after removing the suffix.

Step 8: Go back to Step 5.

Step 9: Store it in a dataframe.

Step 10: Go back to Step 3.

4. RESULTS/OUTPUTS

- **Classification :**

  We are considering all the lyrics , artist name and after tokenizing it we are making them features. If the particular word exists in the first example then the value is 1 otherwise 0.

  Same goes for artists, if a particular artist name is the artist name of the song then value is 1 otherwise 0. Here our data “X” will be all the features(words) , artist name , lyrics and prediction will be “y” that is the type of the song (sad, romantic , party).

index	Song_name	type	artist	lyrics	नीले	अम्बर	चाँद	प्यार	बरस	तरस	...	बइया	तोड़ो	ठहेर	जारे	गाओं	नखरेवाली	रामू	गीले	तोड़ासा	बेसरम
1	नीले नीले अम्बर पर	romantic	किशोर कुमार	नीले नीले अम्बर चाँद प्यार बरस तरस नीले नीले अ...	1	1	1	1	1	1	...	0	0	0	0	0	0	0	0	0	0
2	अक्कड़ बक्कड़	party	बादशाह	अक्कड़ बक्कड़ बॉम्बे बो पुरे रात बज पौ अक्कड़ ...	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	अखियाँ	sad	पोपोन	थक गेया अँख जग्ग दिय अख माह ना लभद अँखियाँहाये...	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	0
4	अंग से अंग लगाना	romantic	अलका याग्निक, बिनोद राठौड़, सुदेश भोसले	आए आए चाहो बाँहो भर अंग अंग लग सजन रंग लग अंग ...	0	0	0	0	1	0	...	0	0	0	0	0	0	0	0	0	0
5	अगर ज़िन्दगी हो	romantic	आशा भोसले	ज़िन्दगी संग ज़िन्दगी संग मौत मौत ज़िन्दगी संग...	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 9284 columns

Figure 2 : Dataset with new binary features

Here we have applied different classification methods for example: Naive Bayes , SVM , Decision Tree , Random Forest. Accuracy table for it is shown below.

Classifier	Naive Bayes	SVM - Linear	SVM - Polynomial	SVM - RBF	Decision Tree	Random Forest
Only lyrics	68.75	65.0	58.75	66.25	62.264	58.75
With lyrics and artist	71.25	70.0	61.25	72.5	63.749	76.25

Table 1 : Comparison of accuracies for different classifiers

- **Accuracy Graph:**

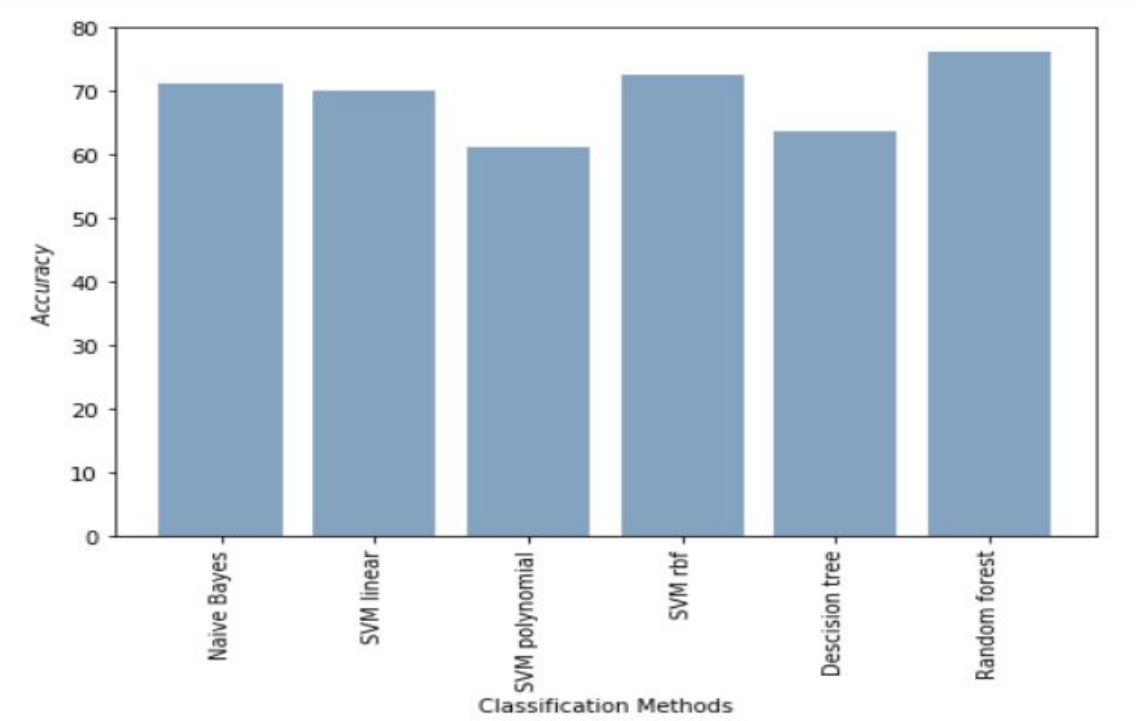

Figure 3 : Accuracy graph

For making better predictions we have tried different methods. Such as one is as mentioned above using lyrics and artist name both. Others are just by lyrics and just by artist name and this is done by using pipeline.It automatically vectorise the lyrics and then classify it,in pipeline it is classified by random forest.

Method	With Only Lyrics	With Only Artist
Accuracy (in %)	66.0377	61.006289

Table 2 : Comparison of accuracies using different input features by pipeline

- **Frontend website screenshots**

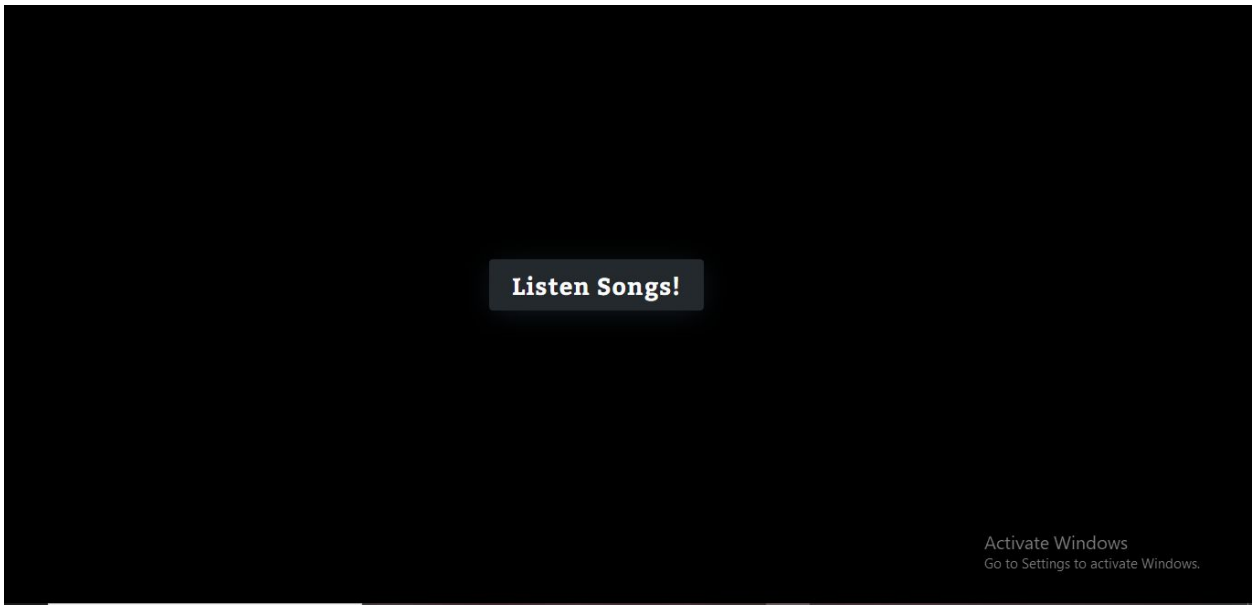

Figure 4 : Home page

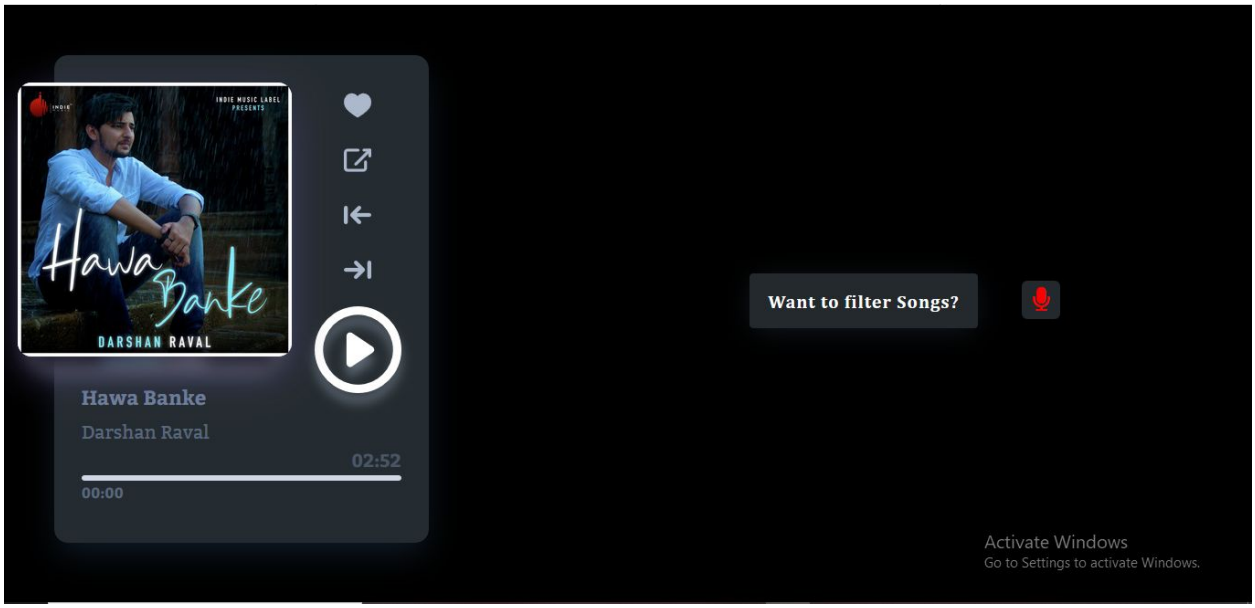

Figure 5 : Songs player

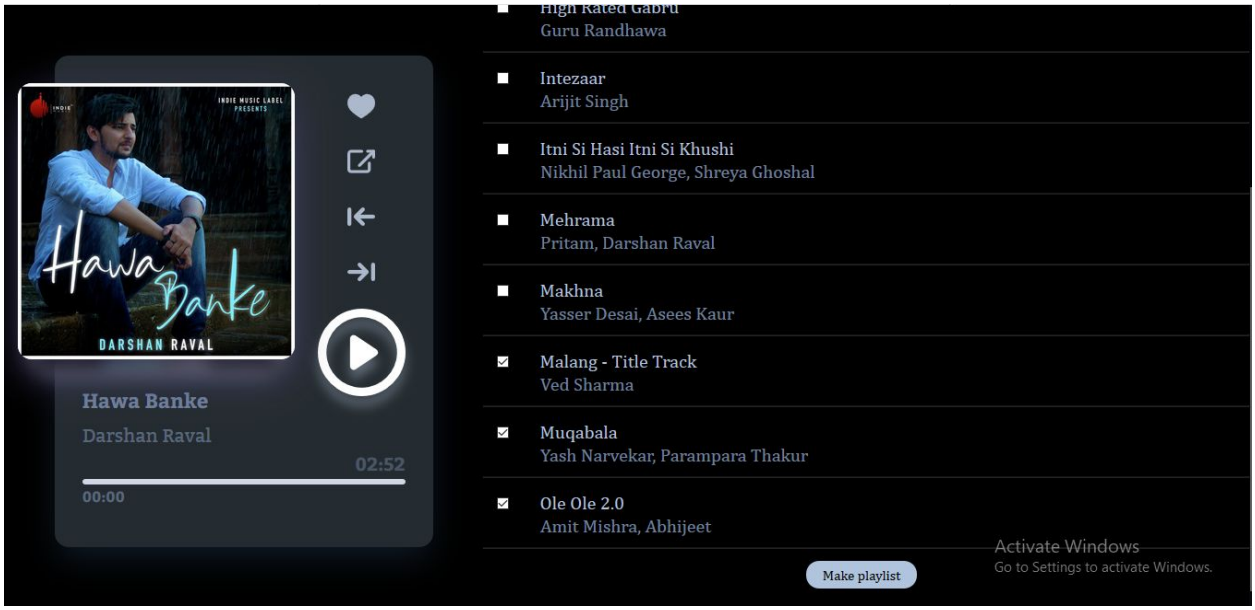

Figure 6 : Selecting make playlist button to create one

Figure 7 : The playlist created

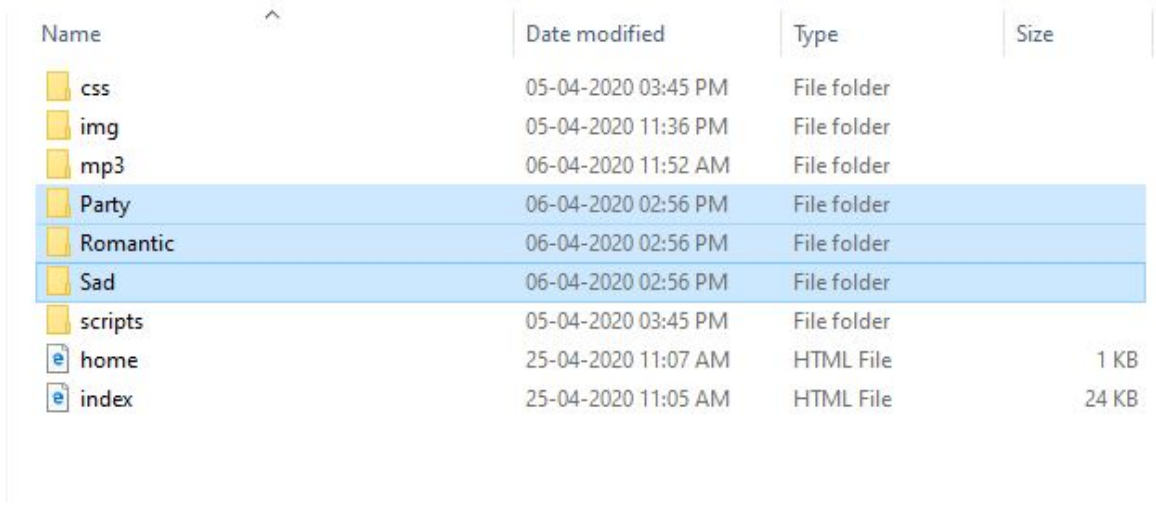

Figure 8 : The sub-playlists created after classification

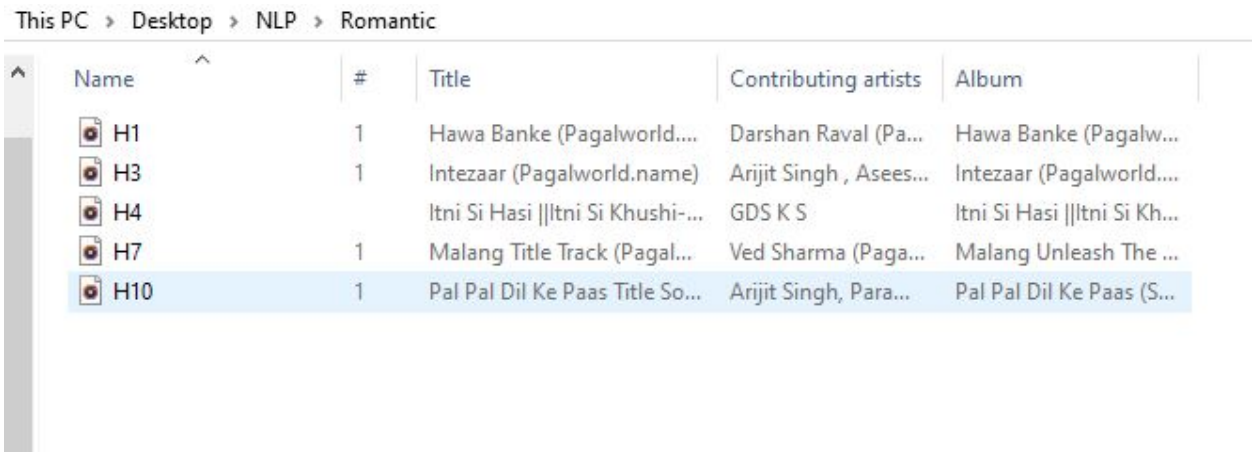

Figure 9 : Playlist of romantic songs

Figure 10 : Playlist of sad songs

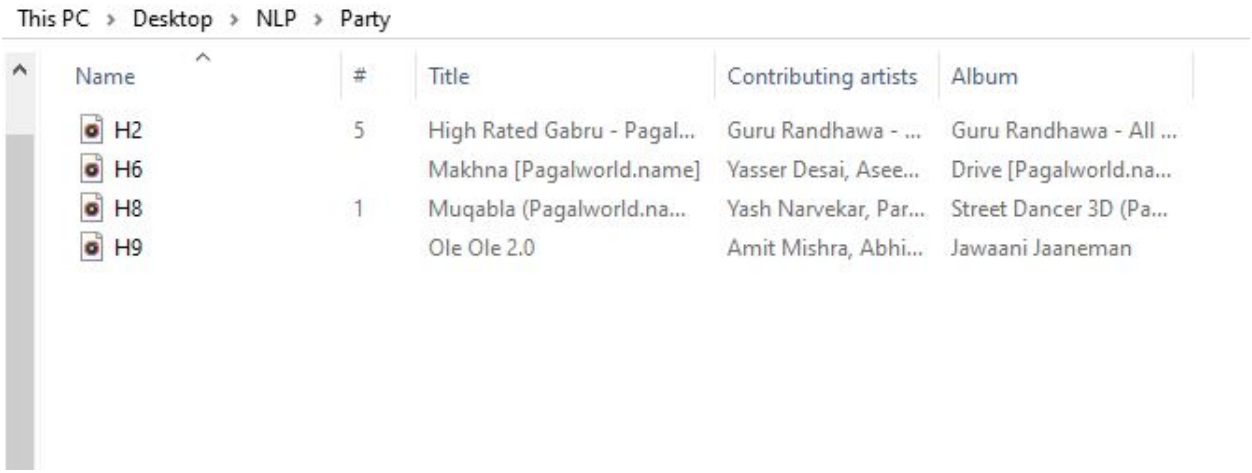

Figure 11 : Playlist of party songs

5. CONCLUSION

We have used many different classification techniques, from that we can say that Random forest gives better accuracy when both lyrics as well as name of the artist is taken into consideration and accuracy in this case is 76.25%. And if only lyrics of the song are considered then, Naive Bayes performs better with 68.75% accuracy.And another important observation is that,even after using inbuilt vectoriser,which tokenize lyrics and then classify,our manually created vectorised data gives better accuracy. It is not possible to further increase the accuracy as we have concluded that for many songs, mood of the song does not depend entirely on the lyrics, music also plays an important role. If audio features are also added along with lyrics and artist then, this accuracy can be further increased, but our main aim was to check correlation of mood with the lyrics.

6. REFERENCES

[1] Patra, Braja & Das, Dipankar & Bandyopadhyay, Sivaji. (2015). Mood Classification of Hindi Songs based on Lyrics.

[2] Mood classification of songs based on lyrics. By Francesco Cucari, 732A92 Text mining - Linkoping University.