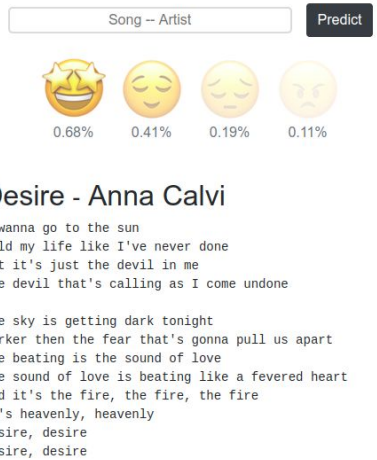
Emotion Classification of Songs

Goal: To create a model that predicts the emotions associated with a song and displays emoticons that best represent it. The model analyses the emotions associated with the lyrics and classifies accordingly.



Datasets Used

- MoodyLyrics
- Lyricwiki
- Spotify
- Million Song Dataset Subset

For extraction of lyrics

For extraction of audio features

Final Dataset used for lyrics

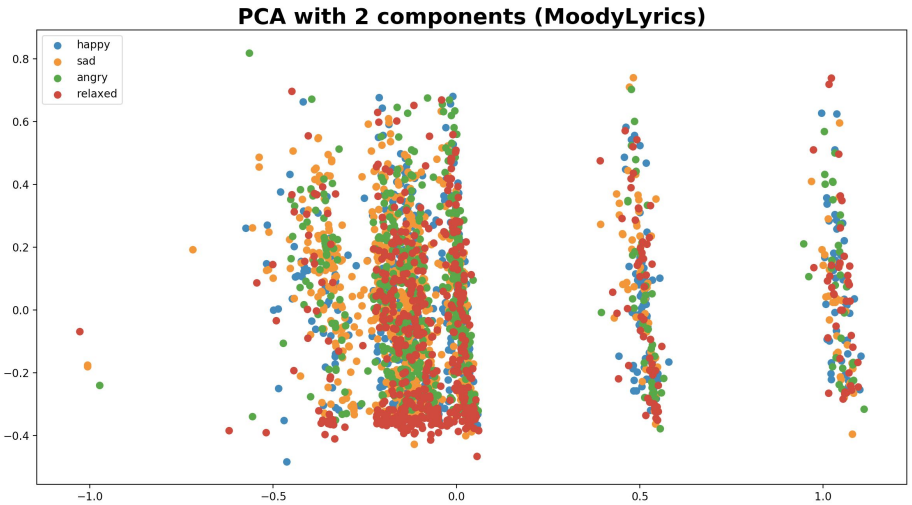
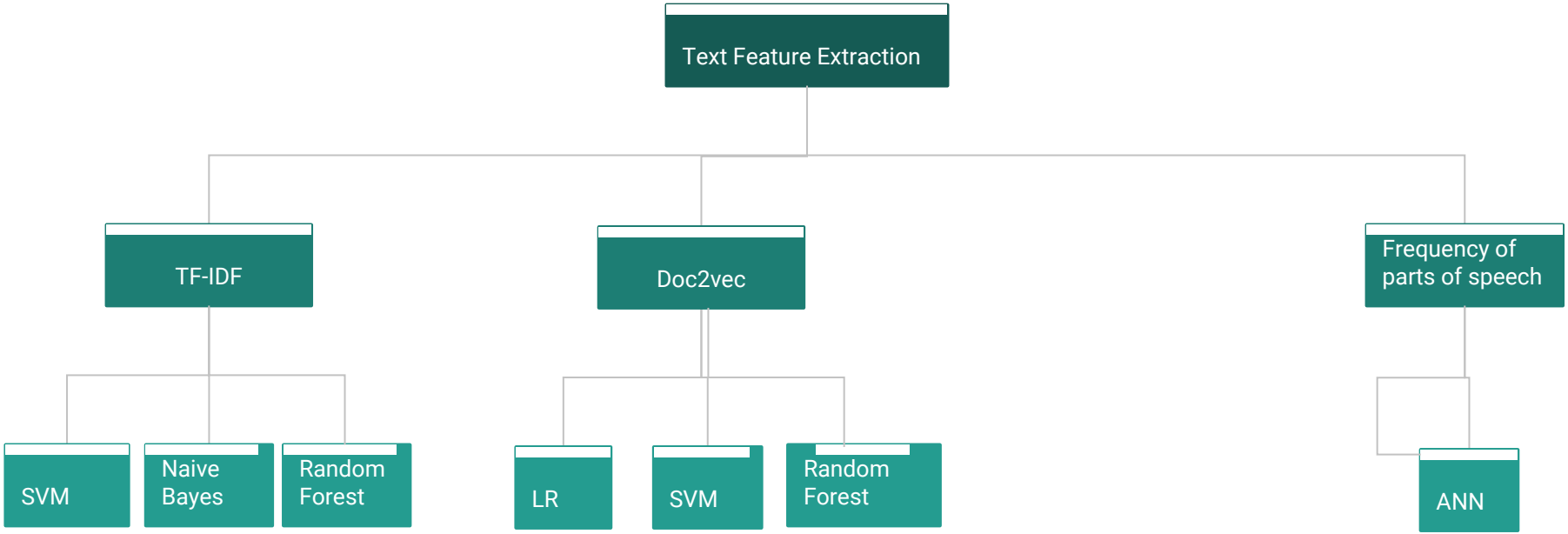


Table 1


Text Feature Extraction
TF-IDF
Doc2vec
Frequency of parts of speech
SVM
Naive Bayes
Random Forest
LR
SVM
Random Forest
ANN

Approaches Used

1. Term frequency–inverse document frequency(TF-IDF): It is a weight(statistical measure) that reflects how important a word is to a document in a collection.

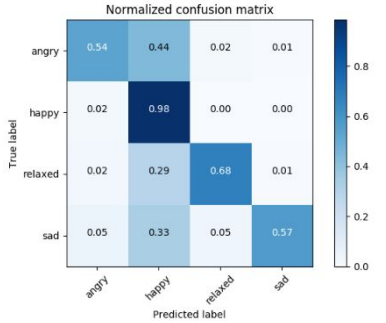
   TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)
   IDF(t) = log_e(Total number of documents / Number of documents with term t in it)
2. Doc2Vec: This object contains the paragraph vectors. The goal of doc2vec is to create a numeric representation of a document, regardless of its length.
3. Frequency of parts of speech + Word2Vec: This approach segments each sentence and classifies each word as noun, verb, adjective etc. which is further converted into a vector for for the training of the model (with the help of SpaCy library).

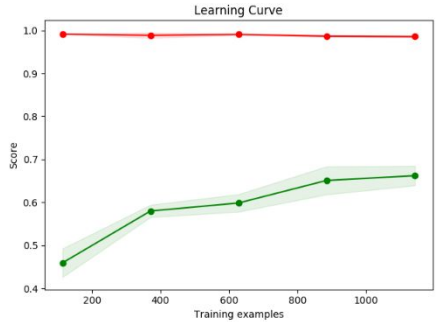
   We applied SVM, Naive Bayes , Logistic Regression, Random Forest and ANN algorithms to get the best models as shown in the table 1.

Analysis

TF-IDF technique

Algorithm	F1 score	Accuracy
Naive Bayes	0.72131148	72.9%
SVM	0.51463169	34.6%
Random Forest	0.6809816	71.2%



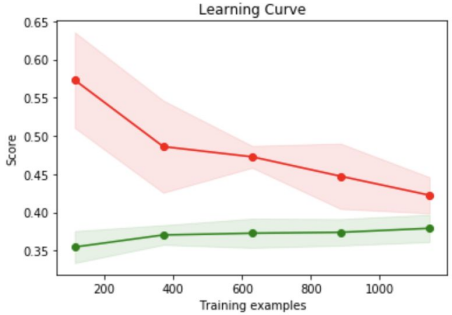


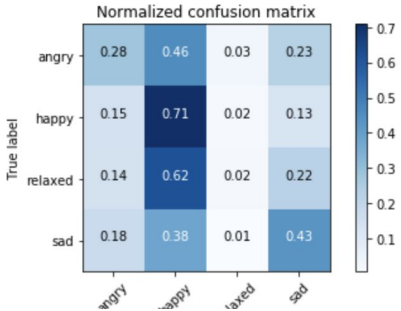
Confusion Matrix and Learning Curve of Random Forest Algorithm

Analysis

Doc2Vec technique

Algorithm	F1 score	Accuracy
Logistic Regression	0.354	36.1%
SVM	0.503	30.2%
Random Forest	0.413	39.5%





Confusion Matrix and LOC of Random Forest Algorithm

Ablative Analysis

We performed Ablative Analysis on the Frequency of parts of speech + Word2Vec method.

Features Initially chosen : Line count, Word Count, Past tense Verbs, Present tense verbs, is Title In Lyrics, Selfish Degree,

Verbs, Punctuations, Adjectives, Rhymes, interjections, Adverbs, Nouns, Numerics, Pronouns, Proper nouns, Word2Vec(Lyrics) mean, Word2Vec(title)

After running Analysis(NN)

Three useless features(Accuracy remains same): Line count, Word Count, Numerics

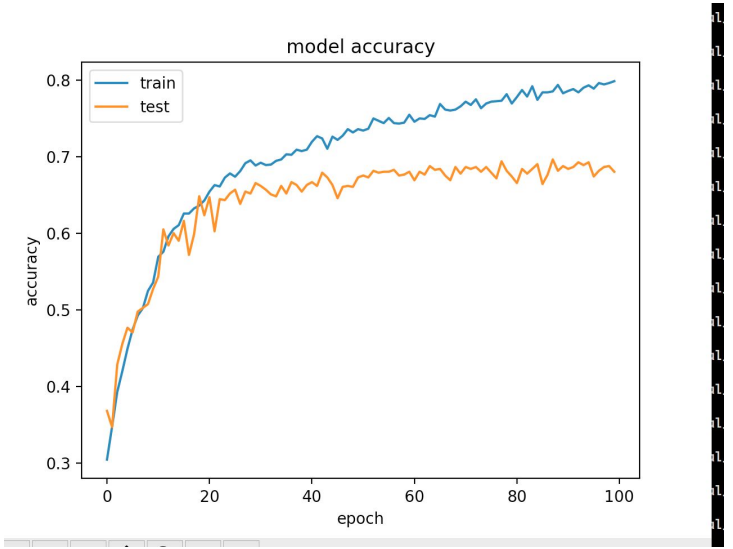
Three most import features: Word2Vec(Lyrics), Word2Vec(title) In both the cases (300 dimension each) the accuracy drops by around 7%

Results

Frequency of parts of speech + Word2Vec technique

Model	Accuracy	Loss
ANN	Test: 70% Train: 77%	0.8772

It gave better results than other techniques so this is a better feature set compared to the previous one.



Other Results and Challenges

Classification using audio features: We used the Spotipy library to extract the audio features: danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence and tempo. Upon training these features with KNN, SVM and Naive Bayes gave poor results of about 40% accuracy.

We initially extracted data (social tags) from last.fm which gave noisy data to train the models for emotion classification. So, later had to amend our approach accordingly.

Individual Contribution

- Yajur - Feature Extraction, Data Processing
- Nirmita - Model Analysis, Feature Extraction
- Meet - Data Extraction, Model Tuning