Resonance: Exploring Emotion Classification and Recommendation of Music using Lyric and Audio Embeddings

Capstone Project

CS-4998-1

Monsoon 2023

Chanchal Bajoria and Medini Chopra

Under the supervision of

Professor Debayan Gupta and Professor Raghavendra Singh

TABLE OF CONTENTS

I. ABSTRACT

Music can evoke emotions, and the advent of popular streaming sites has contributed to the evolution of music listening habits of people, where people prefer to listen to music in a context-based manner, by creating hyperspecific playlists that are not governed by a single factor like audio, lyrics or overall themes. However, music recommendation systems and emotion recognition systems are still subpar at best, which can be attributed to a lack of labeled quality data that is sizable and is accompanied by other relevant meta-data. This study creates a custom dataset of 20,000 songs using Spotify IDs with lyrics, and audio features and manually labeled emotion classes under anger, happiness, sadness, and fear. Further, it employs machine learning and deep learning-based techniques to experiment with combinations of word embeddings including TF-IDF, Word2Vec, and Doc2Vec, and concatenating them with audio features such as danceability, valence, energy, etc. Our final model achieves an accuracy of 78% in the emotion classification task for songs. Along with the analysis of the importance of lyric and audio features in emotion classification tasks, a recommendation system is also built and deployed using Streamlit which recommends songs based on contextual prompts input by the user.

Keywords: Music information retrieval, natural language processing, word embeddings, audio features, contextual recommendations, emotion recognition

II. INTRODUCTION

Music has a unique ability to evoke strong emotions, bringing people together through common human experiences, regardless of language, culture, or nationality. Research in music psychology has long explored the connection between music and emotion, with neuroimaging studies confirming that music activates brain regions typically associated with emotional responses (Schaefer, 2017, Hans-Eckhardt Schaefer. 2017. Music-evoked emotions—current studies. Frontiers in Neuroscience, 11:600).

The popularization of music streaming services has revolutionized our music listening habits, making recommendation systems an essential feature for navigating the vast song collections available on these platforms. Traditional methods like collaborative filtering and content-based recommendation systems have been widely used, but content-based systems often focus on metadata such as artist names and genres, neglecting the rich emotional context provided by lyrics and audio features.

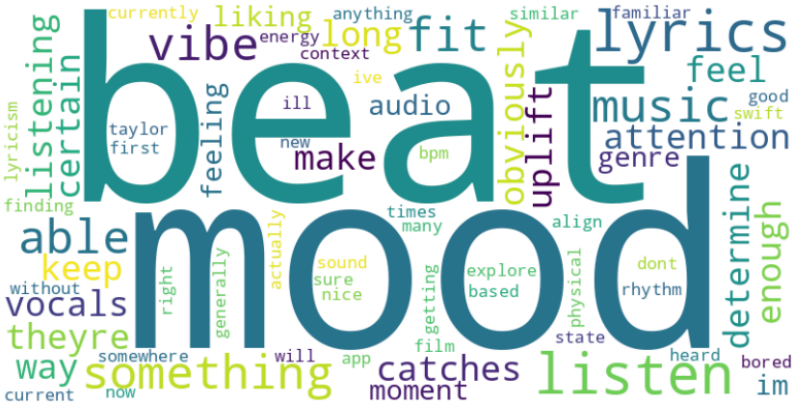
Collaborative filtering employs user behavior for recommendations. Additionally, if many users like a song, then it will recommend that song heavily which leads to popularity bias as well as the cold-start problem for smaller artists. Therefore, in collaborative filtering, popular artists are preferred. Content-based filtering uses the spectrogram, where a Convolutional Neural Network will take parameters such as time signature, key, mode, tempo, and loudness for each song, and compare it with other songs' features. This similarity measure aims to recommend songs that sound similar in terms of beats per minute, acoustics, etc. The challenge with this method is that songs of the same genre would be recommended to the user now. The content-based system also crawls the web for user sentiment on social media and the news (what adjectives are being used to

describe a song), as well as cultural context by analyzing lyrics. Combining the two - which is what most music retrieval systems do - leads to the algorithm preferring popular artists and genres. While there is an analysis of the 'cultural vectors' of songs, it is still below average.

Along with collaborative and content-based filtering, the trend is moving towards context-based song recommendation. The aim is to encapsulate the human experience of listening to songs, which is much more rooted in how they are feeling and what they are doing at the moment. Furthermore, human's response to music is a result of innate qualities of music such as its sonic characteristics like beats, tempo, etc., or resonance with the themes and emotions the lyrics are trying to convey.

Therefore, our project aims to detect emotions in songs through the use of audio features and lyrics. By focusing on the elements that humans process to evoke emotions, we aim to provide a deeper context about a song's themes and emotional conveyance. This framework supports a recommendation system tailored to a user's mood or activity, suggesting songs that align with the same emotional landscape.

User Survey



We conducted a user survey to validate these findings, whose demographic was mainly ages 19-30. The questions included: how often they listen to music, how they would rate Spotify's current algorithm, what they believe Spotify prioritizes when recommending songs, what the most important factor is to them for song recommendation, whether they would accept context-based song suggestions etc. There was an overwhelming majority agreeing that Spotify prioritizes popular artists and genres over the actual contents of the song, and the majority of users preferred recommendations that were determined by the beats and moods of the songs, as well as the lyrics.

While the main aim is to achieve a better context-based song recommendation, the root of the problem begins with how we represent a song numerically in the first place - whether it takes into consideration audio features, lyrics, or social and genre-based external tags, or all at once. After an extensive literature review, we found that choosing the best textual encodings was a widely researched problem, and we started with that.

Our research aims to answer the following questions:

RQ1: Which textual encoding of lyrics best represents a song in latent space?
RQ2: Does the addition of audio features to the lyric features to increase accuracy in the task of classifying based on emotion labels?

III. RELATED WORK

Emotion tagging

Prior work was being done in the emotion classification of songs using a lexicon-based approach. [7] used lexicon mixing, as well as WordNet, WordNet-Affect, and ANEW, to translate each word in the lexicon to a sentiment value, and compounded that to map to the Valence-Arousal dimensions to find the quadrant each song belongs to. Building on the compound sentiment score, [2] used the VADER sentiment score by compounding scores given to each word. [1] uses the NRC Emotion Lexicon and classifies every word based on the Paul Ekman theory to also give a compound emotional score to each song, and makes use of KNN clustering for recommendation. [9] also uses WordNet Affect and ANEW. Further, the models they chose for the task were Random Forest, SVM, and Naive Bayes. However, such lexicon-based labels are highly rudimentary and insufficient for truly representing a song's evoked emotions.

Lyrics classification into emotion classes

[4] made use of TF-IDF and count vector embeddings to represent the lyrics, and tested it with Naive Bayes, Logistic Regression, SVM, and Random Forest. Further, they made use of transfer learning and re-predicted the emotion labels before using sentence transformers for recommendations. They also used the Russell emotion model. [10] experimented with Naive Bayes, Random Forest, and BERT to classify songs with emotion tags, with their highest accuracy averaging 60%. [3] used different embedding methods such as n-grams, TF-IDF, and weighted parts-of-speech, and implemented a 5-fold cross-validation with Logistic Regression, SVM, Random Forest, and Decision trees, with the highest F1 score of 65%. [6] used WordNet-Affect for emotion lexicons and classified them with an SVM. With TF-IDF, the highest F-score was 68%. [13] started moving towards a continuous vector space embedding model, away from sparse vectors. They used Google's Universal Sentence Encoder and clustered using KMeans, over which they used Cosine similarity to find similar songs. Their F1 score was 77%. Finally, [8] achieved a final accuracy of 71% after having tested with SVM, Random forest, KNN, Logistic Regression, and Naive Bayes. All of these papers used Russell and Thayer models of emotion for tags, and the Million Song Dataset, and the standard size of their manually annotated lyric datasets is between 1000 and 3000 songs.

Lyrics with audio features for music information retrieval

[11] used a Siamese Neural Network, where one model was used to train audio features, and the other to train lyrics using state-of-the-art Doc2Vec embeddings after some dimensionality reduction using PCA. [5] used TF-IDF, Doc2Vec, and Word2Vec embeddings, and then used cosine to find similarities in embeddings between two songs. While lyrics-based methods do not outperform meta-data-based approaches, when both of these types of features are concatenated, they provide much better recommendations. Lastly, [12] used Doc2Vec embeddings and a baseline Fuzzy C-means for recommendations.

IV. DATASET

Goals

One of the main aims of our project was to develop a sizeable dataset enriched with emotion labels, audio features, and lyrics to facilitate the accurate classification of songs based on their emotional context.

Dataset Components

Sizeable Dataset: Our dataset contains a substantial number of entries, providing a robust foundation for training and evaluating our recommendation system.

Emotion Labels: Emotion labels derived from the MuSe dataset using 279 mood tags from AllMusic, classified into superclasses such as happiness, sadness, anger, and fear.

Audio Features: Detailed musical characteristics, including attributes like acousticness, danceability, energy, and more.

Lyrics: Comprehensive lyrics data to analyze the textual content of the songs.

Challenges

Lack of Publicly Available Datasets: Most available datasets are constrained by copyright issues, lack of detailed emotion tags, and insufficient mapping parameters between audio features and lyrics.

Literature Review Datasets: Many existing datasets are manually annotated and small in size, limiting their utility.

Relevant datasets

Million Song Dataset (MSD): Although sizeable with audio features and lyrics, the lyrics are encoded in a Bag of Words format and lack emotion tags.

150k Songs Dataset: Contains unencoded lyrics and is sizeable, but lacks both audio features and emotion tags.

1. Collection

   Scraping Overview

   a) Lyrics Scraping

      Source: Spotify API using spotify_ids.

      Challenges: We utilized the Genius API via Spotify IDs and websites like LyricsFreak and lyrics.com. To overcome limitations with the Genius API, we developed a scraper using Selenium and multiprocessing. This involved creating Genius URLs from artist names and track titles. Although mixing web crawling with APIs like genius-lyrics-api and lyricsgenius produced inconsistent results, we successfully implemented a multiprocessing pipeline using azlyrics.com Selenium scraping and the Genius API with VPNs, ensuring efficient data extraction.

   b) Audio Features Scraping

      Sources: Various APIs and web scraping.

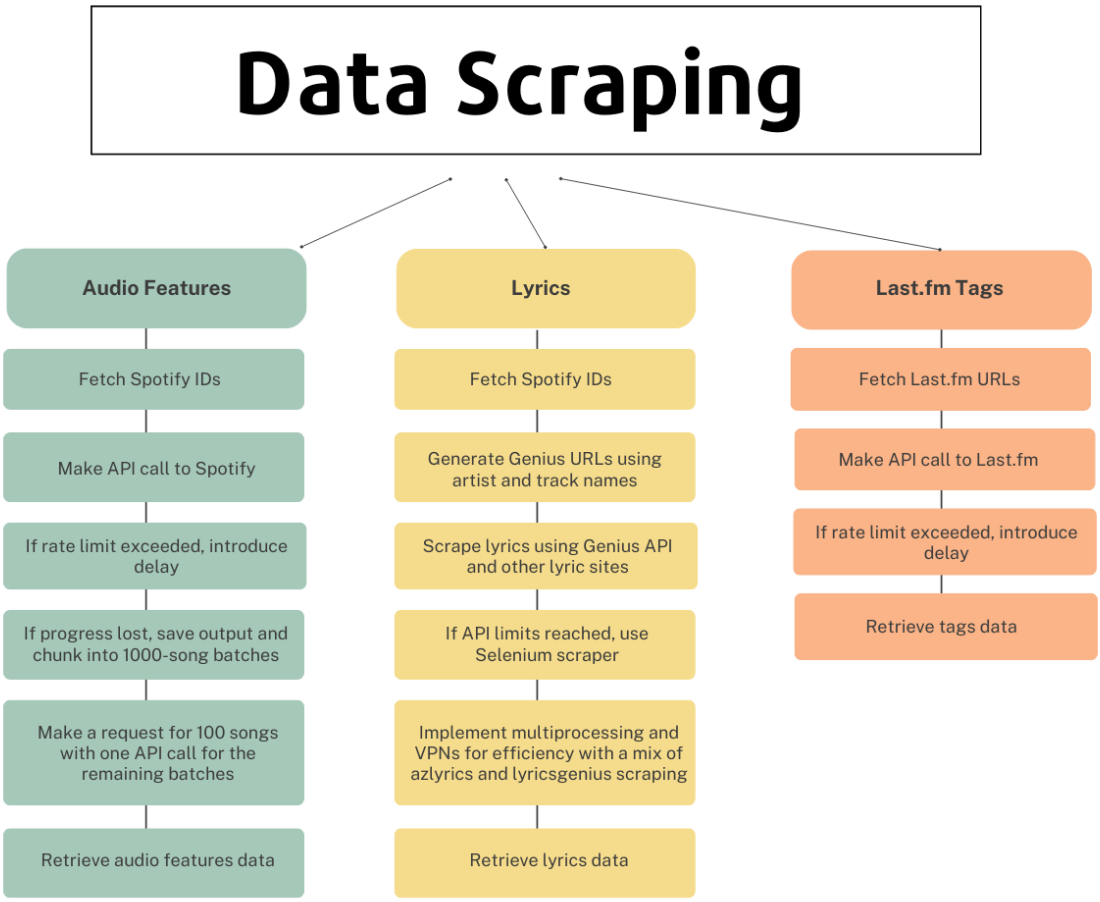
      Process: We encountered several challenges, including rate limits exceeded and a lack of documentation on the Spotify API. To address these issues, we experimented with call delays to manage the rate limits. However, we experienced progress loss every 2000 songs, which led us to chunk the data into 1000-song batches. Despite using multiprocessing and proxies, we couldn't fully alleviate the rate limiting. Ultimately, we achieved a successful implementation by processing 100 song IDs per request.

   c) Last.fm Tags Scraping

      Source: Last.fm API using URLs provided in the MuSe dataset.

      Challenges: Encountered rate limits and VPN blocks, but managed to extract data.

      Note: Last.fm tags were not incorporated in our analysis due to their focus on genre tags, as our emphasis was on audio features and lyrics.



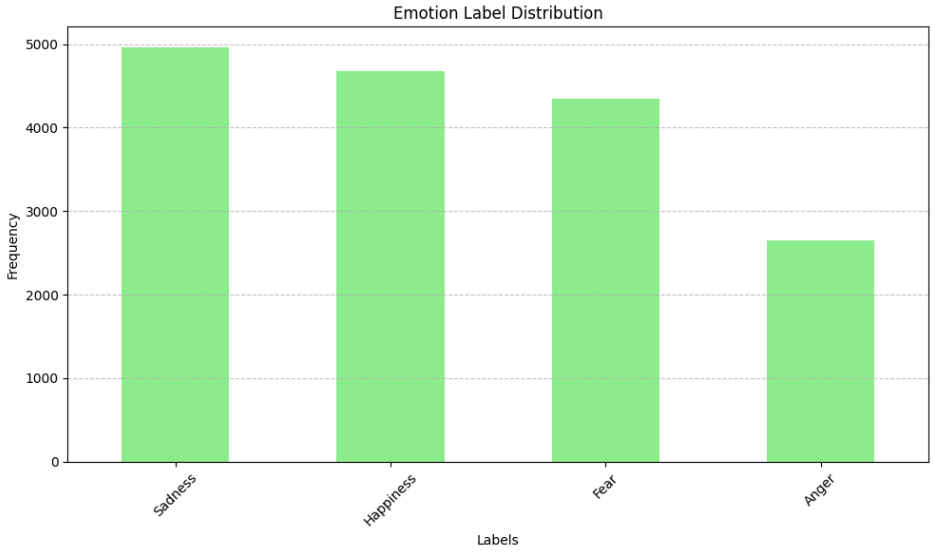
Background on Audio Features and Last.fm Tags

Audio Features:
These are musical characteristics of songs, such as acousticness, analysis_url, danceability, duration_ms, energy, id, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, time_signature, track_href, type, URI, and valence. These features are derived using methods like Mel-Frequency Cepstral Coefficients (MFCC) analysis.

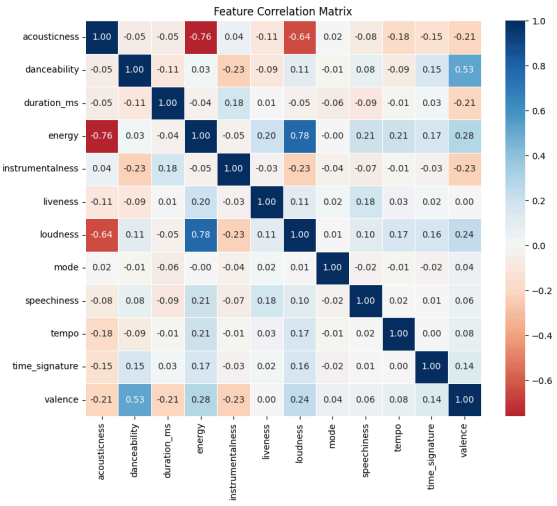
Emotion Tag Extraction:
We based our emotion tags on Paul Ekman's classification and other models such as Russell's Circumplex Model and Thayer's Model. The MuSe dataset provided us with 279 mood seeds, which we classified into four superclasses: happiness, sadness, anger, and fear, using manual annotation.

2. Cleaning

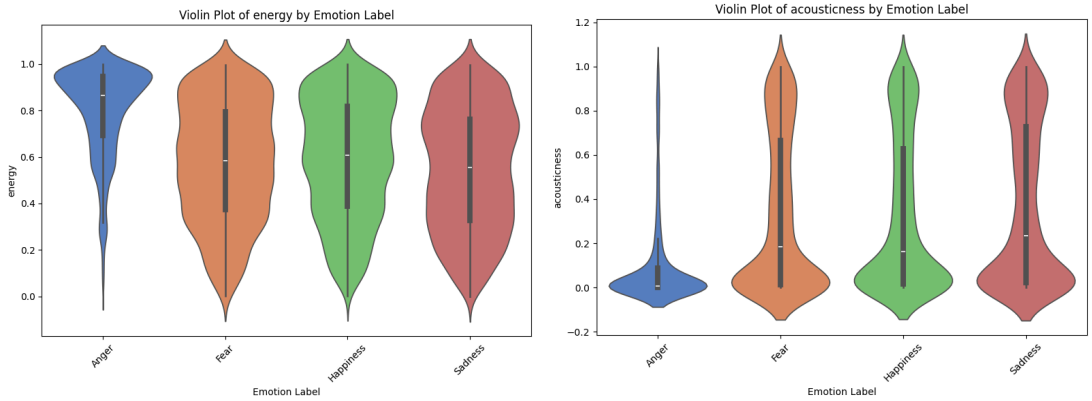
It is important to make the dataset as clean as possible to give the highest accuracy. Since the dataset was manually scraped, we had to remove erroneous values that came from the APIs including movie and play scripts. Using the sequence length of each song lyric, we extracted and removed those. Next, we removed custom stop words such as “Verse”, “Chorus”, and “Lyrics”. Since these lyrics were scraped from open-source platforms, there were also prefixes and suffixes such as “Contributors” and “Embed” which were included in the custom stop word list. Further, we made everything lowercase and removed HTML tags, URLs, numbers, punctuations, single characters, and extra spaces. We normalized to ASCII, fixed contractions, and followed the standard preprocessing pipeline of tokenization and lemmatization. We randomly sampled 2500 songs from each of the 4 emotion classes to have a balanced dataset to work with.



Beyond the actual cleaning process, we looked at an exploratory feature analysis of text data which included word frequencies across class labels. After removing stopwords, happiness had “love”, “life” and “come” as the top words; anger had “never”, “make” and “go” and interestingly “love” was common across fear and sadness, with possible indication of the negative effects of love. After extracting the 12 audio features from Spotify, we normalized them using a max-min scalar and found the correlation between features.



Trivially, the most positive correlations come from loudness & energy, and valence & danceability. For negative correlation, we see that acousticness and loudness, and acousticness and energy have high values. Building on this, when we took the violin plots of values across the four classes, we saw similar results. The energy and acousticness are negatively correlated, but also angry songs are majorly high energy and low acousticness, which comes from genres like “rock”.



VI. CLASSIFICATION

1. Lyrics analysis

Embeddings

For word embeddings, we used TF-IDF (term frequency-inverse document frequency), Word2Vec, and Doc2Vec representations. We wanted to analyze traditional techniques, reliable techniques, and

state-of-the-art techniques. Secondly, the majority of previous literature made use of TF-IDF and Word2Vec for the same tasks. It is new for Doc2Vec to be employed for this task.

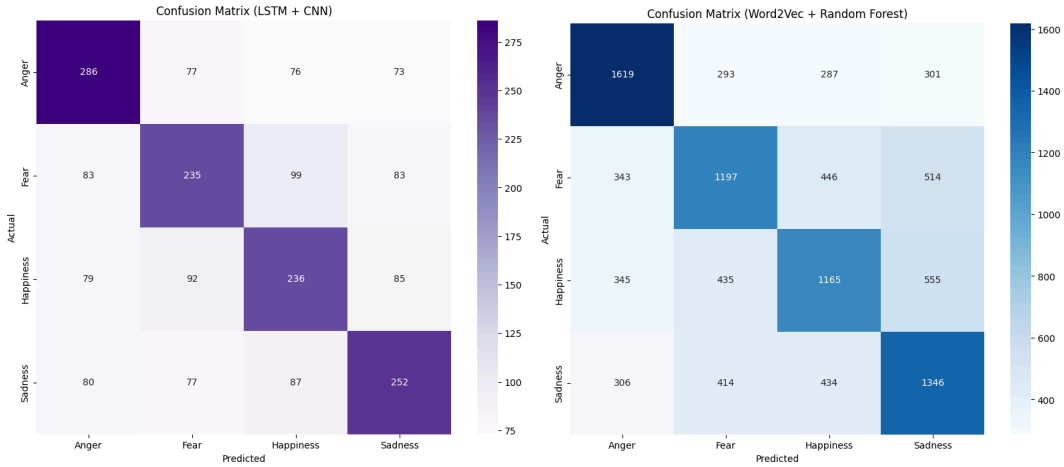
a) TF-IDF The way it works is that it calculates the importance of a word relative to the corpus, and it represents a document as a high-dimensional, sparse vector. It treats every word independently from each other. For TF-IDF we used max_features of 1000.
b) Word2Vec operates by mapping every word into a continuous vector space, and compared to TF-IDF it is a low-dimensional, dense vector representation. This captures the context of words and updates the numeric representation based on semantic meaning. For Word2Vec google-news-300 pre-trained embeddings we used 10,000 vocabulary size.
c) Building on Word2Vec, a new model called Doc2Vec has been developed, which is a neural network-based approach as well. Instead of learning the distributed representation of words, it learns for documents. For Doc2Vec we trained the model with a vector size 100 with 40 epochs.

Algorithms

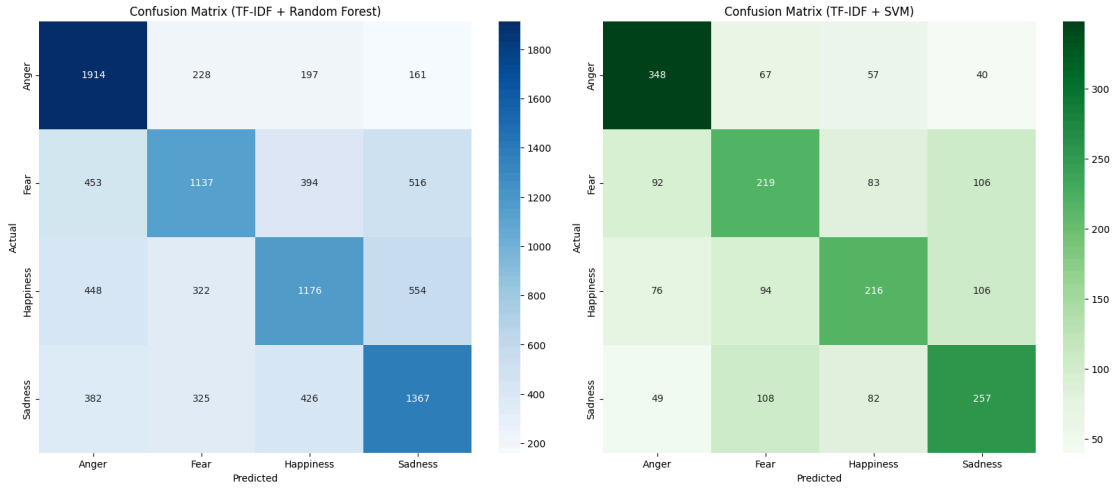
The algorithms we picked were SVM, Random Forest, and LSTM. The reasons were twofold again: firstly, these were the best-performing algorithms in the above-mentioned literature review provided and/or are standard for text classification, and secondly, we wanted to do a comparative analysis between traditional, ensemble, and neural network methods for text classification. Our SVM employed an RBF kernel due to the non-linear nature of our data, and for both SVM and Random Forest we used a 5-fold cross-validation to have more robust results. The LSTM consisted of an embeddings layer, a convolutional layer for local feature and pattern extraction (64 filters with kernel size 3 and 'relu' activation), a pooling layer, an LSTM layer with 128 dimensions, and a final dense layer with 'softmax' activation to classify the lyrics based on anger, happiness, sadness, and fear. We used categorical cross entropy for the loss due to the multilabel classification task and used 40 epochs for training. The convolutional layer was removed when experimenting with TF-IDF since Word2Vec and Doc2Vec make use of local semantics to update their values, and there are patterns to be found in those contextual areas, but TF-IDF would not merit from that architecture also because it is a sparse, high-dimensional vector.

Results

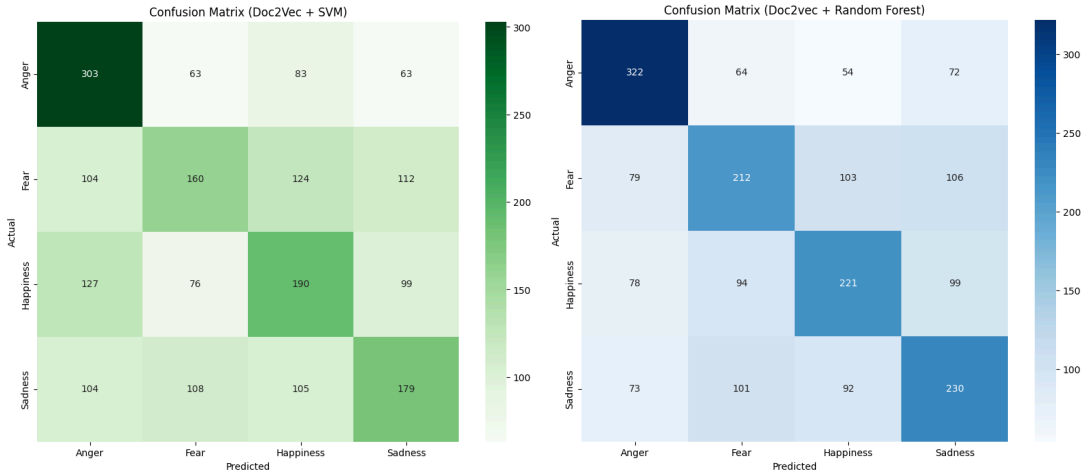
Since the dataset was balanced, we employed accuracy as our metric to compare the baseline models' performance with each other. The highest-performing model was the LSTM+CNN architecture with the Word2Vec embedding at 65%, where the final Word2Vec embeddings used a 10k vocabulary size and 150 sequence length. Random forest with Word2Vec and TF-IDF gave 54% and 56% respectively.



Following that, TF-IDF with SVM gave 52% and with LSTM yielded 50%.



The remaining few were the Doc2Vec performances with our chosen three models, along with the SVM + Word2Vec combination.



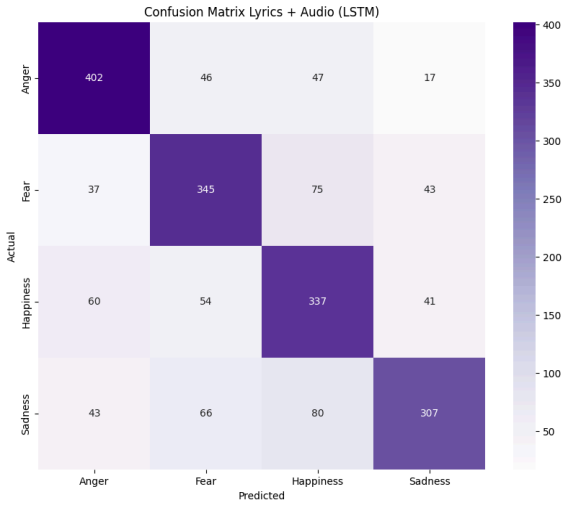
The resulting accuracies are provided in the table below:

	SVM	**Random Forest**	**LSTM**
TF-IDF	52%	56%	50%
Word2Vec	38%	54%	65%
Doc2Vec	45%	49%	35%

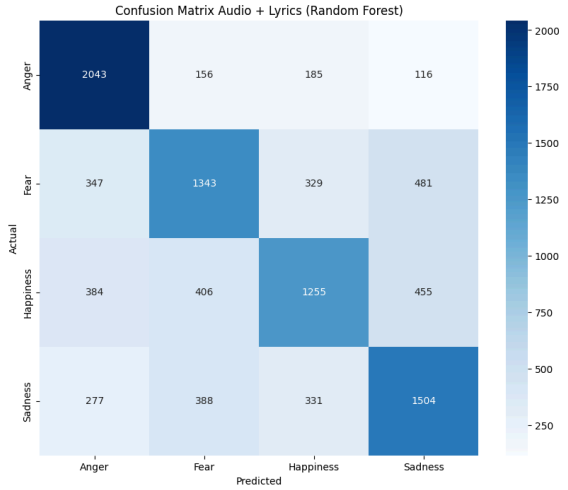
2. Lyrics + audio analysis

Since our two best-performing models were Random Forest and LSTM, we chose those for our second phase of analysis to see if adding audio features to text embeddings improves the accuracy of our baseline models. The options also allowed for a comparison between ensemble and neural network architectures. Our best-performing embedding overall was Word2Vec, especially with those two models. We achieved the addition of audio features to lyrics in two ways, where each model used a different approach.

a. For **LSTM+CNN**, we let the word embeddings flow through the layers as is and merged the 12 audio features as input right before the classification layer. This meant that the textual embeddings were processed via the CNN+LSTM layers, and right before calculating final probabilities, we merged the numeric, normalized audio feature. This yielded an accuracy of 69%, which is a 5% improvement from the previous iteration of only using lyrics.



b. For **Random Forest**, we concatenated the 12 audio features with the 300-dimensional word embeddings of the sequences, which resulted in a singular 1D vector representation for every song. These were fed to the classifier which yielded an accuracy of 78%, which was the highest out of the lot. This representation was then chosen for the next phase of our study: the recommendation system.



The results have been summarized in the following table:

Using Word2Vec	**Lyrics**	**Audio + Lyrics**
Random Forest	54%	78%
LSTM+CNN	65%	69%

VII. EVALUATION

For phase 1 - lyrics analysis, we observe that TF-IDF has performed well with SVM, and this is because SVM is good at handling high dimensional data. For that same reason, as well as the robustness to noise, Random Forest performed even better. For TF-IDF, neural networks are known not to perform well due to the sparse nature of the embeddings. Doc2Vec functions in a manner that it takes an entire song to be a single document, and calculates its semantic similarity based on neighboring songs. As it can be observed, such a representation of songs may not yield accurate results compared to word-level representations that capture more information about the words themselves, especially when they contribute to the class labeling (for example mentions of certain words in respective classes). We see that Word2Vec gave a fairly decent accuracy across the board for Random Forest and LSTM. For SVM, the low measure can be attributed to its inability to learn decision boundaries for the word representations. Word2Vec offers a good mix of reliability on a word level but also captures semantic similarity, falling in the middle of TF-IDF and Doc2Vec in terms of functioning.

Furthermore, in phase 2 - lyrics and audio analysis, we observe that concatenating the audio features to the text embeddings increased accuracy significantly, proving that it is useful to use a multimodal (audio and lyric analysis based) approach to song representation. While Random Forest accuracy jumped a decent amount, the LSTM can be further fine-tuned by adding intermediate layers before the final classification layer to ensure a progressive dimensionality reduction and prevent a bottleneck (which was outside the scope of this project).

VIII. RECOMMENDATION SYSTEM

Objective

Our goal was to develop a deployable working interface that users could interact with to test our hypothesis for contextual song recommendation. We aimed to generate a list of songs that capture specific moods or moods linked to particular activities by using our basis for mapping emotion context in music—audio features and lyrics.

Methodology

Emotion Superclasses and Mood Tags: As mentioned earlier, we generated our emotion superclasses based on 279 mood tags derived from the MuSe dataset. The recommendation system utilizes these mood tags along with the concatenated audio features and lyric embedding vectors, generated in phase 2, for classification to suggest songs to the user.

User Interface: The user interface was built using Streamlit, enabling quick deployment and user-friendly interaction. The interface prompts the user to enter a text prompt to "Describe your current mood or

activity." Then, the user is provided with a list of 5 moods and asked to select all that apply. Based on the selected moods, the user is given a list of 5 songs and asked to pick the one that best fits their mood or activity. Upon selecting a song, the user receives a playlist of 10 songs recommended by the algorithm.

Recommendation algorithm

a) Input Handling
   The algorithm prompts the user to input text describing their current mood or activity.
   The user input text is passed to a function called map_to_seeds.
   This function uses Hugging Face's BART LLM model to generate a likelihood score between each seed and the text prompt.
   Seeds and their respective likelihood scores are sorted in descending order, according to the likelihood scores, and the top 5 seeds of the sorted seeds are returned as the "relevant seeds".
b) Mood Selection
   The relevant seeds are displayed to the user, who selects all moods that apply.
   The user-selected moods update the list of relevant seeds.
c) Filtering Songs:
   Songs from the dataset are filtered to include those with at least one of the updated relevant seeds.
   These filtered songs are sorted based on the degree of overlap with the updated relevant seeds list.
   The top 5 songs with the greatest overlap are displayed to the user
d) Playlist Generation
   The user selects the most fitting song from this list, which is then returned to the recommendation algorithm.
   A cosine similarity function calculates the similarity of each song in the filtered list (excluding the selected song) based on the concatenated audio features and lyrics embedding vector.
   The filtered songs list is sorted in descending order according to the similarity score, and the top 10 most similar songs are displayed to the user as their playlist.

Rationale for Method Choices

Cosine Similarity: We chose cosine similarity over methods like Fuzzy C-means or K-means due to the high-dimensional nature of our data and the variety of input types.

Exclusion of Last.fm Tags: We focused on inherent musical qualities (audio features and lyrics) for contextual cues, excluding Last.fm tags which are primarily social and genre tags.

Highlights

1) Extensive Dataset: The dataset includes lyrics, audio features, genre, social tags, and emotion classes, providing a rich resource for diverse recommendations.

2) Enhanced Emotion Classification: Incorporating audio features into lyric embeddings significantly improves emotion classification, as demonstrated in phase 2.
3) Deployable Tool: The tool is ready for deployment, allowing for real-time user testing and further improvements in contextual recommendations.
4) User-Centric Experience: The system enhances music discovery by aligning recommendations with users' emotional states and activities, creating personalized listening experiences.

IX. DISCUSSION

One of the main contributions of this project is the extensive dataset that has been created with lyrics, audio features, genre and social tags, and emotion classes, as well as the existing fields including Spotify IDs, artist name, track name, etc. From our phase 1, we conclude that Word2Vec embeddings perform better in classifying songs concerning their underlying emotions conveyed. From our phase 2, we conclude that emotion classification is enhanced by incorporating audio features into lyric embeddings. Through phase 3, we contribute a deployed tool that can be utilized for further testing and improvement of contextual recommendations. This study has aimed at enhancing music discovery and listening experiences of humans, by incorporating a nuanced understanding of how music evokes and expresses emotions. We hope to establish a relationship between a creator's intentions with a song and how it translates to resonating with a listener's mood as a result of shared experiences among humans.

X. LIMITATIONS

There are a few limitations to this project. Firstly, we utilized a subsample of the full dataset which ended up being 20k songs out of the total 60k dataset. Secondly, the dependency on a zero-shot classification LLM for condensing text input into relevant seeds means compromising on wholly accurate results. Thirdly, since the recommendation system is user-facing, there is currently no direct way of validating the results quantitatively, and there are limited "mood seeds" to filter for as well. Lastly, there is potential for enhancing the overall accuracy of the classification task, as well as integrating with the Spotify Web API to recommend songs that are outside of the training dataset.

XI. FUTURE SCOPE

For our given emotion classifier, by analyzing the confusion matrices more closely, we can find interesting patterns of misclassification. Furthermore, the lyrics of each song can be summarized into fewer tokens using an LLM. In the current list of songs being recommended to the user, there may be some songs that do not completely match the prompt, since we had an 80% success rate. In that case, as an initial filter, we can remove songs that have mismatched titles from what the theme and mood would anticipate. Additionally, the Last.fm tags that we have made accessible with the dataset can be used for more music

information retrieval studies in the subarea of contextual recommendations. It can be further expanded to low-resource languages as well.

XII. ACKNOWLEDGEMENTS

The paper thanks Prof. Debayan Gupta and Prof. Raghavendra Singh for constant support and direction, as well as the MuSe dataset for contributing to the basis of this paper.

XIII. REFERENCES

[1] J. Choi, J. -H. Song and Y. Kim, "An Analysis of Music Lyrics by Measuring the Distance of Emotion and Sentiment," 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Busan, Korea (South), 2018, pp. 176-181, doi: 10.1109/SNPD.2018.8441085.
[2] U. Kryva and M. Dilai, "Automatic Detection of Sentiment and Theme of English and Ukrainian Song Lyrics," 2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 2019, pp. 20-23, doi: 10.1109/STC-CSIT.2019.8929732.
[3] Detecting Emotions in Lyrics: https://www.diptanu.com/data/lyrics_emotion_detection.pdf
[4] V. R Revathy, Anitha S. Pillai, Fatemah Daneshfar, LyEmoBERT: Classification of lyrics' emotion and recommendation using a pre-trained model, Procedia Computer Science, Volume 218, 2023, Pages 1196-1208, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2023.01.098 (https://www.sciencedirect.com/science/article/pii/S1877050923000984)
[5] Michaela Vystrčilová and Ladislav Peška. 2020. Lyrics or Audio for Music Recommendation? In Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics (WIMS 2020). Association for Computing Machinery, New York, NY, USA, 190–194. https://doi.org/10.1145/3405962.3405963
[6] Hu, Xiao & Downie, J. & Ehmann, Andreas. (2009). Lyric Text Mining in Music Mood Classification.. Proc. Int. Soc. Music Information Retrieval Conf.. 411-416.
[7] Erion Çano and Maurizio Morisio. 2017. MoodyLyrics: A Sentiment Annotated Lyrics Dataset. In Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence (ISMSI '17). Association for Computing Machinery, New York, NY, USA, 118–124. https://doi.org/10.1145/3059336.3059340
[8] R. Akella and T. -S. Moh, "Mood Classification with Lyrics and ConvNets," 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 2019, pp. 511-514, doi: 10.1109/ICMLA.2019.00095
[9] Rachman, F. H., Sarno, R., & Fatichah, C. (n.d.). Music Emotion Classification based on lyrics-audio using corpus-based emotion. International Journal of Electrical and Computer Engineering (IJECE). http://doi.org/10.11591/ijece.v8i3.pp1720-1730

[10] Edmonds, D., & Sedoc, J. (n.d.). Multi-emotion classification for Song Lyrics. https://aclanthology.org/2021.wassa-1.24.pdf
[11] S. F. R. Ali, T. P. S. D. Thanuj, D. S. S. Kiran and M. Ashwin, "Recommender System using Audio and Lyrics," 2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2023, pp. 772-781, doi: 10.1109/ICESC57686.2023.10193474.
[12] Braja Gopal Patra, Dipankar Das, and Sivaji Bandyopadhyay. 2017. Retrieving Similar Lyrics for Music Recommendation System. In Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017), pages 290–297, Kolkata, India. NLP Association of India.
[13] V. Gupta, J. S. and S. Kumar, "Songs Recommendation using Context-Based Semantic Similarity between Lyrics," 2021 IEEE India Council International Subsections Conference (INDISCON), NAGPUR, India, 2021, pp. 1-6, doi: 10.1109/INDISCON53343.2021.9582158.
[14] Christopher Akiki, and Manuel Burghardt. (2021). MuSe: The Musical Sentiment Dataset [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/2250730