Music Emotion Recognition Using Advanced Machine Learning to Analyze Lyrics and Audio Features

B.Sc. in Artificial Intelligence and Computer Science (40 credit)

School of Computer Science, University of Birmingham

Student Name: Yuchen Zhu

Student ID: 2335100

Supervisor Name: Dr. Jizheng Wan

Data: 8th April 2024

Word Count: 16520

Abstract

This research is dedicated to creating a comprehensive music emotion analysis model, which aims to mine the complex emotional content in musical works through the analysis of lyrics and audio data. A key aspect of the research was the replication and enhancement of methods from a key paper that utilized the MoodyLyrics dataset. This enhancement includes combining BoW, Tf-idf, and Word2Vec text embedding techniques that focus more on word frequency and local context with advanced preprocessing methods and machine learning models such as SVM, CNN, and NB. Among them, the combined SVM+Tf-idf achieves 94% accuracy and F1 score, which makes significant progress. Moreover, the research found that incorporating Spotify audio features can significantly improve the performance and generalization ability of the model. This is evidenced by the performance of CNN+Densenet+Word2Vec on the MoodyLyrics4Q dataset, which outperforms the baseline with an F1 score of 68%. Additionally, the research applied the model to predict the emotional content of Spotify's Top 100 songs from 2013 to 2023, validating these predictions with real-world societal data to provide an accurate dataset for future research. This research not only demonstrates that combining lyrics and audio data can enhance the understanding of musical emotion, but also provides an important resource for in-depth exploration of the emotional landscape in contemporary music.

Key Words: Music Emotion Analysis, Lyrics, Spotify Audio Features, Embedding, Preprocessing, Machine Learning.

Acknowledgements

I want to express my deepest gratitude to my supervisor, Dr. Jizheng Wan, for his invaluable insights and guidance throughout this research. I am also thankful to my inspector, Dr. Masoumeh Mansouri, for her valuable feedback during the project proposal and demonstration stages. Additionally, I want to thank the University of Birmingham as an institution, and to all lecturers in the B.Sc. program in Artificial Intelligence and Computer Science, who have provided us with quality education and the necessary resources to complete this research. This work would not have been possible without their imparted academic knowledge. Finally, I would like to thank my parents, family and friends for their unwavering encouragement and support.

Acronyms

MIR = Music Information Retrieval

MER = Music Emotion Recognition

CNN = Convolutional neural network

NB = Naive Bayes

LSTM = Long Short-Term Memory

BI-LSTM = Bidirectional Long Short-Term Memory

KNN = K-Nearest Neighbors

SVM = Support Vector Machine

RF = Random Forest

Lemma = Lemmatization

Stem = Stemming

LR = Lowercase Conversion

NR = Noise Removal

SR = Stopword Removal

PCA = Principal Component Analysis

t-SNE = t-distributed Stochastic Neighbor Embedding

RFE = Recursive Feature Elimination

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Background and Motivation

The aim of the project is to develop a computational model capable of analyzing and recognizing the emotional content of music, with a particular focus on lyrical and audio features. Music is an ancient and universal art form that has played an important role in human history. It is capable of containing and expressing emotions, and this relationship has been explored by various theories [1][2]. Contemporary empirical studies have further supported this connection by categorizing three distinct domains of listeners' cognitive and behavioral activities, each associated with a specific type of musical emotion: embodied emotions, cognitive emotions, as well as associative and contextual emotions[3]. Recent research has also found that the structure of music can trigger specific emotions, regardless of an individual's musical preference[4]. These findings collectively emphasize the powerful impact of music on our emotional experiences. Whether in moments of celebration or mourning, music can uniquely touches our souls, awakening deep feelings of joy, sadness, love, or nostalgia. Across different cultures and societies, music plays a central role, not only as a medium for emotional expression, but also as a bridge connecting people through emotions.Furthermore, from a neuroscience perspective, music has been shown to influence complex neurobiological processes. It can be used as an alternative therapy for various mental disorders[5], highlighting its potential as a tool for promoting mental health. In the past decade, with the exponential growth of easily accessible digital music libraries, the challenge of effectively organizing and searching music and its related data has become increasingly prominent. Music Information Retrieval (MIR)[6], as a scientific field

within this domain, is rapidly advancing towards automated systems for searching and organizing music and related data. While common search and retrieval categories like artists or genres can be quantified into a "correct" (or generally agreed upon) answer, the emotional expression inherent in the music itself can be highly subjective and difficult to quantify[7]. Therefore, the subjectivity and complexity of its emotional content mean that traditional methods may not be sufficient to meet growing needs[8]. Searching for music according to emotion is one of the main criteria used by users[9][10], hence real music databases from websites like Spotify and Last.fm are growing daily, requiring extensive manual work to stay updated. Lyrics and audio, as the two main components of music, are often used by artists to convey emotional dimensions, with different aspects potentially conveying different emotional dimensions and intensities. In today's data-driven environment, the rapid development of machine learning and deep learning has opened new possibilities for analyzing complex emotional content in music. These advanced technologies enable us to automatically process and analyze large music datasets and reveal hidden emotional layers in musical works, particularly when dealing with lyrics rich and audio features of complex emotional expression.

1.2 Existing Solutions and Problems

Despite the progress in music emotion analysis, existing approaches often have limitations in comprehensively capturing the emotional essence of music. Many studies have mainly focus on either lyrical content or audio features, ignoring the synergistic potential of integrating both. This oversight can lead to a partial understanding of music's emotional impact, limiting the effectiveness of applications in areas such as personalized music recommendation, therapeutic interventions, and emotional analysis in music education.

1.3 Objectives and Contributions

This project aims to apply machine learning, deep learning, and natural language processing technologies to develop a comprehensive model that analyzes lyrics and audio data to gain a deeper understanding of emotional expression in music. The goal is to surpass existing benchmarks for emotional classification and enhance the the generalization ability of the model. This model is not only expected to provide a new perspective for music sentiment analysis, but also to have a positive practical impact on fields such as music recommendation systems, emotional therapy, and music education, providing a more detailed and comprehensive approach to music emotion analysis.

In summary, the main contributions of the project include the following:

1. Replicating an existing model from academic literature, the project provides a solid foundation for future enhancements, effectively establishing a baseline for the research approach and confirming its validity.

2. Optimizing the lyrics-only model builds on this replication and makes significant improvements in embedding techniques, preprocessing methods, and adjustments to the model's architecture and parameters, resulting in surpassing the performance baseline of the original research paper.

3. Incorporating audio features, the development of an integrated model was achieved, exhibiting increased robustness and generalizability and surpassing the performance benchmarks of subsequent studies in the field.

4. Utilizing the integrated model to predict and label emotions in top songs over the past decade, the project not only compiled a valuable dataset but also used it for assessing the model's effectiveness against real-world events, showcasing its capability in accurately reflecting emotional trends for music recommendation and emotional analysis.

1.4 Report Organization

This report is organized into seven chapters.Chapter 1 starts with a broad overview, introducing the background of the project, its specific objectives and contributions. Chapter 2 presents the necessary background knowledge, including historical and technical contexts, as well as related work.Chapter 3 details the specific methodologies of the research and some visual results. Chapter 4 provides the quantified research results. Chapter 5 offers an in-depth analysis of the research findings and the proposed methods. Chapter 6 discusses future research directions and potential limitations of this research. Finally, the report concludes in Chapter 7.

CHAPTER 2

Literature Review

2.1 Music Emotion Recognition

In recent years, Music Emotion Recognition (MER) [7] has gradually become a hot topic of research and has emerged as one of the key areas in the field of MIR[6]. The annual conferences of the International Society for Music Information Retrieval (ISMIR)[11], which began in 2000, signify the maturation and development of the MIR field. MER aims to delve deeply into and accurately identify the emotions and moods expressed in musical works, and realize personalized music recommendation and accurate classification. The ability of music to express and evoke emotions remains a mystery. Different schools of psychology and musicology have their own interpretations and explanations for this phenomenon. The study by Patrik N. Juslin, Erik Lindström, and others reveal[12], on one hand, to study how individuals experience emotional responses evoked by music from a psychological perspective. On the other hand, focuses on how music, as an art form, is crafted and emotionally expressed.

When it comes to understanding the relationship between music and emotions, psychological models of emotions in music have been shown to be valuable tools. These models facilitate the reduction of the emotional spectrum into a set of practical categories. An early and groundbreaking study in this area was Hevner's research from 1936 [13], which outlined a categorical model consisting of 66 emotional adjectives, grouped into eight categories (Figure 2.1)[13]. Although the basic form of this model is not widely used in MER, it has been a reference point in some studies employing categorical models.

Figure 2.1: Hevner Emotional Adjective Model

The two-dimensional model of emotion by Russell[14], which classifies and understands emotions through dimensions of valence (the positive or negative aspect of emotions) and arousal (the intensity of emotions), is more valued in MER(Figure 2.2)[14].

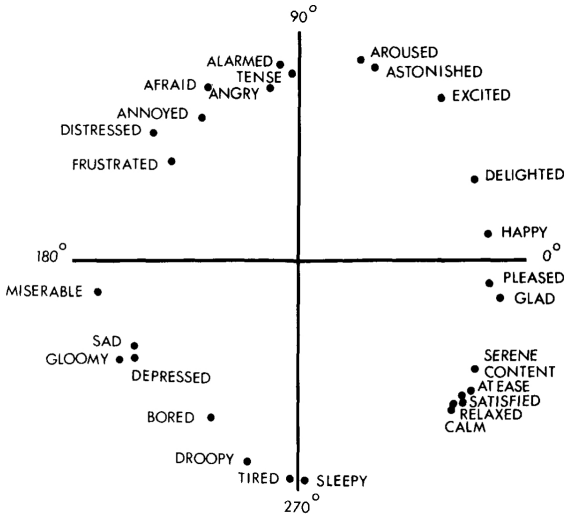

Figure 2.2: Russell Circumplex Model

However, the complexity of musical emotion is often beyond the scope of this model. Leading researchers like Schimmack and Grob [15] have developed a three-dimensional emotional model, which includes dimensions like pleasure–displeasure, awake–tiredness, and tension–relaxation.This multidimensional approach provides a more comprehensive and in-depth framework for understanding and interpreting the emotional experience evoked by music, reflecting the diversity and complexity of musical emotions.

MER involves analyzing various features of music (such as melody, rhythm, harmony, lyrics, and background) to identify and categorize the emotional content in music. This deep exploration not only drives the rapid development of personalized music systems but also reveals significant social value and potential for commercial applications [16]. Studies conducted

by Yang in 2008 [17] and 2014 [18] have demonstrated the application of automatic MER technology in developing emotion-based music therapy methods. Y Liu and O Sourina [19] have also proposed a real-time EEG-based emotion recognition algorithm for music therapy, highlighting the significance and prospects of MER.

2.2 Music Emotion Recognition Technology

With technological advancements, the information technology applied in the field of MER has become increasingly rich. Emotion analysis in music can be approached from many aspects. Studies have shown that the independence of lyrics and tune is consistent with the modular organization of the human cognitive system, as indicated by M. Besson, F. Faïta, I. Peretz, A. Bonnel, J. Requin in 1998[20]. Although music is a multimodal data form, as described by Rudolf Mayer and Andreas Rauber[21], and might have emotional tendencies influenced by the corresponding artist and cultural background, But study by Michael Fell and C. Sporleder [22] has shown that lyrics and audio are the backbone of music analysis. Consequently, most research in this field focuses on analyzing music's lyrics and audio.

Due to copyright issues and the subjectivity of music, the emotion labels corresponding to lyrics and audio and their acquisition are still a challenge in the field, resulting in a lack of rich datasets. In the lyrics aspect of MER, researchers Erion Çano and Maurizio Morisio[23] used ANEW and WordNet lexicons for word representation and clustering. They integrated the emotional values of each sentence and the entire song to create a four-quadrant MoodyLyrics dataset based on Russell's emotional model, which is available to the public. They employed dictionary and clustering methods and compared them with lyrics datasets annotated with user tags and human subjects, achieving an accuracy of 74.25%. This dataset focuses only on the emotional dimensions and labeling of lyric texts. In a study by Abdillah J et al. [24], using MoodyLyrics dataset, a combination of Bi-LSTM+GloVe with regularization layer and hyperparameter tuning was employed. The method achieves 91% accuracy only on lyrics. However, the audio dimension was not considered in this study.

Subsequently, Erion Çano created the MoodyLyrics4Q[25], a dataset based on the Russell emotional model for classification standards and collected from the music community platform Last.fm based on subjective user tags. This allows emotion analysis to have a wider dimension than just text. On this dataset, a study by Yinan Zhou[26], using XL-NET and Lemma preprocessing with early stopping, achieved an F1 score of 59.08%, but this study also only considered the lyric dimension and not the audio aspect.

In the field of word embedding, studies have shown different word embeddings have varied

performances for different tasks. Word2Vec and GloVe are often used by researchers in emotion analysis studies[27][28][29] and have shown good performance.

Regarding audio features in music, Spotify provides a rich set of audio features to help analyze and understand music. A study found that using audio features from Spotify and evaluating the performance of logistic regression in predicting song emotions achieved an F1 score of 86%[30]. Additionally, another study[31] used vector distance calculation combined with Spotify's Valance and Energy values, referencing Russell's emotional model, to define the emotion classification of music more accurately. However, these studies only considered the emotional dimension of audio features.

Therefore, this research project aims to combine lyrics analysis with audio features provided by Spotify to further analyze and understand the impact of these two modalities on MER. In addition, advanced machine learning techniques will be used to integrate these modalities to improve the accuracy and generalization ability of the model. In summary, the proposed method plans to integrate and optimize multiple data sources to achieve more effective on MER.

CHAPTER 3

Methodology

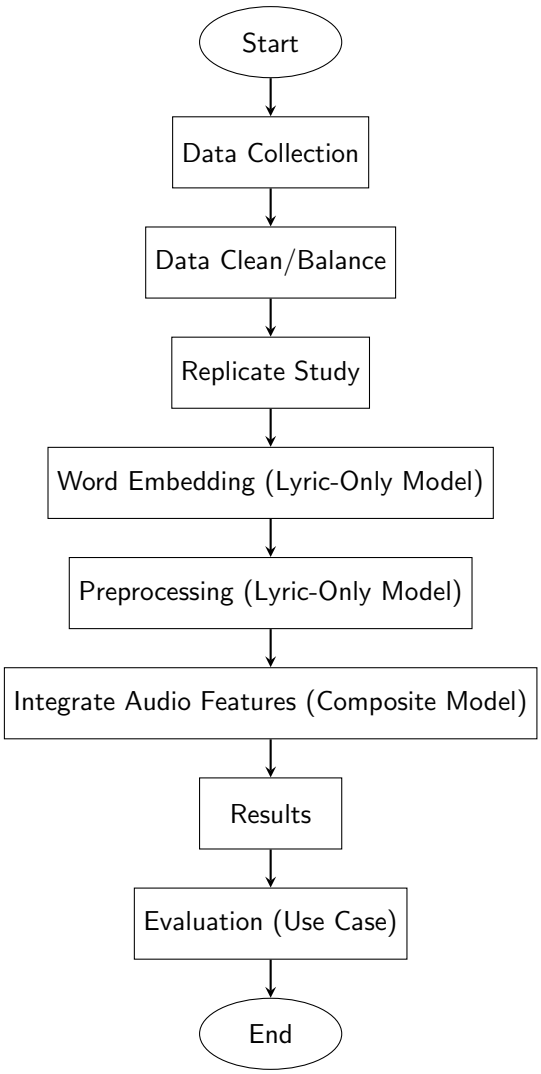

Figure 3.1: Experimental Process Flowchart

Detailed Experimental Process(Figure 3.1):

1. **Data Collection:** Use scripts to collect experimental data.
2. **Data Clean/Balance:** Clean and balance the data to improve quality and prevent bias.
3. **Replicate Study:** Replicate existing studies for baseline validation and issue identification.
4. **Word Embedding (Lyric-Only):** Experiment with different word embeddings.
5. **Preprocessing (Lyric-Only):** Optimize preprocessing experiments.
6. **Integrate Audio Features (Composite):** Analyze and combine audio features with lyrics.
7. **Results:** Experimental results of the methodology.
8. **Evaluation (Use Case):** Evaluation of the experimental results of the methodology.

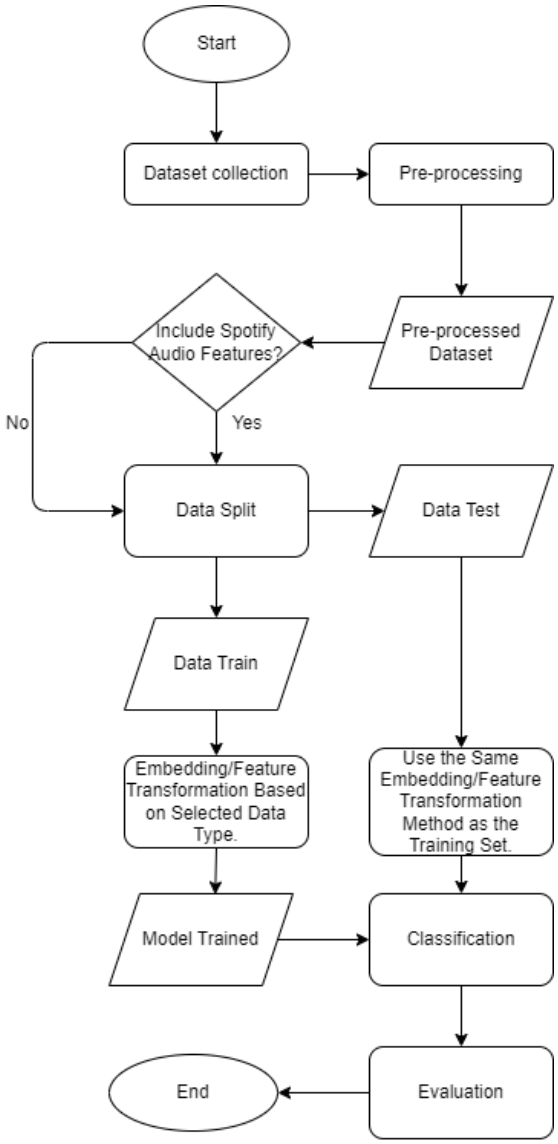

Figure 3.2: Flowchart of lyric analysis and audio feature integration

As shown in Figure 3.2, this flowchart shows the processing flow of lyric analysis and audio feature integration. Firstly, the data collection and preprocessing stage cleaned and prepared the original data. Then, the data is divided into training and testing sets according to whether audio features are included or not. The training set is used for embedding/feature transformation and model training, and the test set is used for model evaluation. Ultimately, based on the results of the performance evaluation of the model, it can be decided whether the process or model needs to be adjusted. This flowchart can help us understand the application of lyric analysis and audio feature integration in machine learning.

3.1 Dataset

3.1.1 Dataset 1 (MoodyLyrics)

MoodyLyrics[23] is a public dataset built specifically for music emotion analysis and contains the lyrics text of 2595 songs. The main feature of this dataset is that according to Russell's emotion model (Figure 3.3)[23], song lyrics are classified based on their content words and their valence and arousal norms in the emotion lexicon, and labeled in four different emotion quadrants. This approach is unique in that it identifies the emotional content of a song based only on textual analysis, without considering the melody, rhythm or sound features of the music. As such, MoodyLyrics offers a unique perspective that focuses on understanding how lyrics themselves convey emotions.

3.1.2 Dataset 2 (MoodyLyrics4Q)

MoodyLyrics4Q[25] is another publicly available dataset designed for music emotion research and contains the lyrics of 2000 songs. The feature of this dataset is the use of last.fm user tags for emotion classification of songs, where each song is labeled with four different emotion categories in Russell's emotion model (Figure 3.3)[25]. Compared to MoodyLyrics, MoodyLyrics4Q relies more on user-generated tags, which reflect the emotional perception and reactions of a broad audience. Therefore, this dataset provides an emotion recognition method that is closer to the actual experience of listeners and enables researchers to understand music emotions from different perspectives.

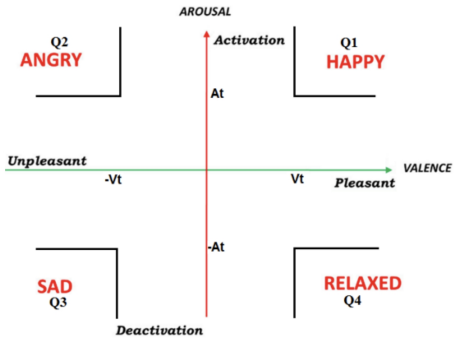

Figure 3.3: Russell Model for Erion Çano Dataset

3.1.3 Data Collection

Lyrics Collection

Due to copyright restrictions, the original datasets does not include lyrics, so a separate process for obtaining lyrics is required. The initial approach involved using the Genius API[32], which is a service providing access to a large collection of song lyrics.

The Genius API is accessible through the third-party community's `lyricsgenius`[33] package, which allows retrieving lyrics by song title and artist name. It provides a direct way to programmatically access accurate and legally available lyrics. However, it quickly became apparent that this method was prone to inaccuracies, especially in cases where song titles or artist names had multiple spellings.

To enhance the accuracy of lyric matching, improvements were made to our custom web scraping approach. This involved using the Google API[34] to perform targeted searches on the Genius website, and then parsing the HTML files of the returned results with the added capability of `BeautifulSoup`[35]. Our focus was on precisely locating and extracting the "div" element in the HTML that contained the correct lyrics, thereby ensuring a match with the artists and song titles from our dataset. This method combines the parsing and text extraction

capabilities of `BeautifulSoup` to locate and verify song and artist information more accurately. It significantly reduced the risk of retrieving incorrect lyrics, as well as the noise commonly found in lyrics obtained from the Genius API through the `lyricsgenius` library, such as "[Verse 1]" tags and "\d +Contributors". This refinement helped in minimizing the need for subsequent normalization and regularization steps. Although this custom approach, enhanced with `BeautifulSoup`, was more labor and time-consuming, it provided a higher degree of accuracy in the data collection phase.The detailed pseudocode implementation can be found in Appendix C: Pseudo code at Algorithm 1.

Audio Feature Collection

With the lyrics data, the next focus was on the acquisition of audio features for each track. This was done through the Spotify API[36], a comprehensive and powerful interface provided by the Spotify music streaming service.

The Spotify API provides a wide range of information about songs, albums, and artists. Of particular interest to this project were the audio features that Spotify provides, which include measurable aspects of a track like tempo, key, energy, danceability, loudness, and valence. These features are quantified based on Spotify's audio analysis algorithms.

To retrieve these audio features, we performed queries to the Spotify API using the song titles and artist names. This process involved sending a request to the API with the relevant track details and parsing the response to extract the desired audio features.

3.1.4 Data Cleaning and Standardization

After data collection is completed, the next crucial step was data cleaning and standardization, which was essential for ensuring the quality and reliability of the data for subsequent analysis.The primary tool employed for this process was custom regular expressions. Regular expressions are extremely powerful for text processing and pattern matching, enabling the identification and removal of specific text patterns from the data efficiently.

Removal of Non-Essential Information: This included the elimination of elements like "[Verse1]" tags, which are often found in lyrics but do not contribute to the overall analysis. Such tags are more about the structure of the song rather than its content and can introduce noise into the data.

Exclusion of Non-English Lyrics: Since the analysis was focused on English songs, lyrics in other languages were removed to maintain consistency and relevance in the dataset.

Correction of Erroneous Audio Features: Any incorrect or outlier values in audio features provided by the Spotify API were identified and corrected or removed. This step was

crucial to maintain the integrity and reliability of the dataset for analysis.

These cleaning and standardization steps were implemented to significantly improve the quality of the dataset, thereby laying a strong foundation for subsequent analysis and interpretation.

3.1.5 Data balance

Upon completion of the collection and standardization processes, the dataset comprised a total of 2123 songs. However, it was observed that the dataset was imbalanced, with Dataset 1 showing a skew towards certain categories. To address this issue:

Downsampling for Balance: In Dataset 1, downsampling techniques were applied to achieve balance. Specifically, 90 "Happy" songs were randomly removed. A specific random state was used to ensure that the process is reproducible, thereby maintaining the categories of the dataset balanced.

Random Shuffling for Unbiased Training: To prevent the model from learning any potential ordering in the data, we randomly shuffle the entire dataset prior to the training process, making it an unbiased step toward more reliable model training and evaluation by using specific random states to ensure consistency in the shuffling process.

Final Dataset Composition

After completing the data cleaning and standardization processes, the datasets were finalized with the following composition:

Dataset 1: The Final Dataset 1 contains 2033 records distributed across four different moods:

- Happy: 554 records (27.2%)
- Relaxed: 532 records (26.2%)
- Angry: 501 records (24.6%)
- Sad: 448 records (22.2%)

Figure 3.4: Dataset1 Word Cloud

Dataset 2: The Final Dataset 2 contains 1576 records distributed across four different moods:

- Happy: 394 records (25.2%)
- Relaxed: 396 records (25.4%)
- Angry: 396 records (25.4%)
- Sad: 375 records (24%)

Figure 3.5: Dataset2 Word Cloud

Dataset Structure

The dataset is structured with multiple fields, each capturing a specific aspect of the song. Below is a horizontal table of the datasets displaying the fields:

Artist	Title	Mood	Lyrics	Danceability	Audio Features...
Usher	There Goes My Baby	Relaxed	There goes my baby...	0.626	0.52, ...

Table 3.1: Dataset Structure

3.2 Reproducing the paper

After preparing the data, we conducted a comprehensive literature review, which led our to a significant study by Jiddy Abdillah et al.[24], titled "Emotion Classification of Song Lyrics using Bidirectional LSTM Method with GloVe Word Representation Weighting". A key part of our research was the replication of this study, which utilized Dataset 1.The main goal of this replication was to assess the reliability and validity of the paper's findings and to consider its potential as a benchmark for our own research. This replication was not solely about confirming the reproducibility of the results; it also provided deeper insights into the methodologies and logical frameworks of the original study. Additionally, this process allowed us to identify

potential areas for improvement and issues within the context of using Dataset 1.To maintain consistency with the replicated study, this research employed the `sklearn` library for machine learning tasks and utilized the `keras` library for deep learning tasks.

In line with our research, which also utilizes similar algorithms, and considering the replicated study's application of these methods, we decided to provide a more in-depth explanation of these algorithms and their embedding techniques, particularly focusing on their role in text analysis. This exploration is crucial as it helps to build a solid foundation to understand how these methods can contribute to sentiment analysis and sentiment classification of lyrics.

3.2.1 GloVe6B-100d

In their study, the authors employed the GloVe 6B-100d model to generate word embeddings. This model is part of the GloVe (Global Vectors for Word Representation)[37] project developed at Stanford University, which provides 100-dimensional vector representations for words, trained on a 6 billion word corpus. These embeddings are good at capturing complex semantic and syntactic patterns in text, such as equations like "king - man + woman = queen (Figure 3.6)[37]". The training objective of GloVe is to align the dot product of word vectors with the logarithm of their co-occurrence probabilities, effectively relating the logarithm of probability ratios to vector differences in the word vector space.

It combines matrix factorization and word context to generate word embeddings. GloVe constructs a co-occurrence matrix that counts how frequently words appear together in a given corpus, and then uses matrix factorization techniques to reduce the dimensionality of this matrix. This method effectively captures global statistical information about word relationships, but may not be as sensitive to the local context of word usage as Word2Vec.

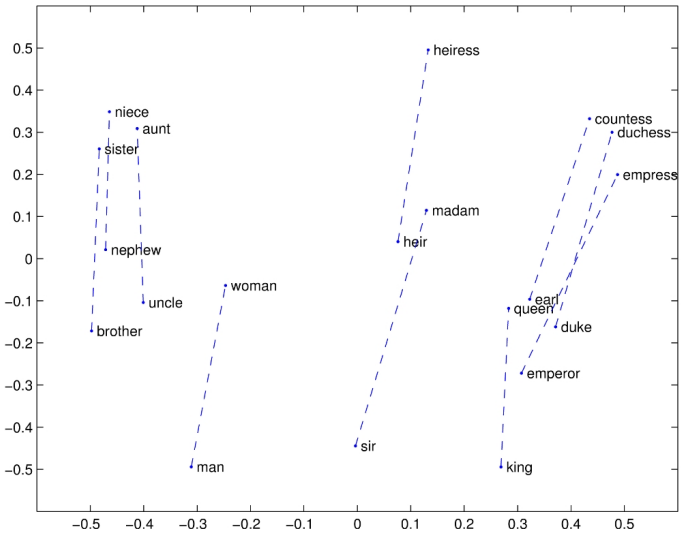

Figure 3.6: Glove Linear Substructures

3.2.2 Naive Bayes

Naive Bayes has its roots in the Bayes theorem developed by Thomas Bayes in the 18th century[38], and when it was introduced to the field of text information retrieval under different names in the 1960s, It has gained a prominent position as a basic classification technique. Renowned for its efficiency with large datasets, such as those in sentiment analysis of song lyrics, NB is effective for categorizing text into different emotional tones using categorical data. It calculates the product of the prior probability of an emotional tone $P(c)$ and the likelihood of lyrical features given that tone $P(f_i|c)$. The most likely emotional tone $\hat{c}$ is identified by maximizing this product, demonstrating the method's simplicity and effectiveness in classifying text.

$$\hat{c} = \arg\max_{c \in C} \left\{ P(c) \prod_{i=1}^{n} P(f_i|c) \right\}$$

.

In the replicated study, a particular variant of Naive Bayes, namely Multinomial Naive Bayes, is often used because of its suitability for classifying discrete features such as word frequency. The multinomial Naive Bayes formula applied to lyrics sentiment analysis is as follows:

$$P(C_k|\mathbf{x}) = \frac{P(C_k) \prod_{i=1}^{n} P(x_i|C_k)^{a_i}}{P(\mathbf{x})}$$

Here, $\mathbf{x}$ represents the frequency vector of words in lyrics, and a_i indicates the frequency of word i in the lyrics to be classified. The numerator $P(C_k) \prod_{i=1}^{n} P(x_i|C_k)^{a_i}$ calculates the likelihood of the lyrics belonging to a particular emotional class C_k, while the denominator $P(\mathbf{x})$ acts as a normalizing factor. In the lyric sentiment analysis, a smoothing parameter α was set to 0.05 to adjust for words not present in the training set and to prevent zero probabilities in subsequent computations.

3.2.3 K-Nearest Neighbors

K-Nearest Neighbors, initially introduced as a non-parametric method in the early 1950s and further developed in 1967 by Cover and Hart[39], is a supervised learning algorithm widely used in sentiment analysis of song lyrics. It is applicable to both classification and regression tasks, offering simplicity and intuitiveness in understanding and implementation. The non-parametric nature of KNN enables it to efficiently handle various types of data distributions, making it particularly suitable for various scenarios in natural language processing.

In the replicated study, KNN determines the emotional tone of a song lyric by considering its

nearest neighbors in the feature space. Each lyric is represented as a feature vector using GloVe embeddings to capture semantic information from the lyrics. KNN predicts the sentiment label based on the majority class or average value of the nearest neighbors in the training set. This proximity-based approach allows KNN to handle outliers and noise in the data, making it robust for sentiment analysis tasks.

Let $\mathbf{x}_q$ represent the query sample, $\mathbf{x}_i$ be a sample in the training set, and $d(\mathbf{x}_q, \mathbf{x}_i)$ denote the Euclidean distance between $\mathbf{x}_q$ and $\mathbf{x}_i$ in the feature space. The predicted sentiment label $\hat{y}_q$ for the query instance $\mathbf{x}_q$ is computed using a distance metric such as Euclidean distance (p=2).

$$\hat{y}_q = \text{argmax}_y \sum_{i=1}^{K} 1(y_i = y)$$

This formula calculates the predicted sentiment label $\hat{y}_q$ for $\mathbf{x}_q$ based on the sentiment labels of its K nearest neighbors. In sentiment analysis of song lyrics, $\mathbf{x}_i$ represents the feature vector of the ith lyric instance, and y_i is its sentiment label. In the replicated study, the parameter K was set to 29, which indicating that the algorithm considers the 29 nearest neighbors for sentiment classification using the Euclidean (p=2) distance metric, with lyric features represented through 100-dimensional GloVe embeddings.

3.2.4 Support Vector Machines

Support Vector Machines, developed by Boser, Guyon, and Vapnik in 1992[40], are a powerful and versatile supervised learning algorithm primarily used for classification and regression tasks. Known for their effectiveness in high-dimensional spaces, SVMs are particularly useful in situations where the number of dimensions exceeds the number of samples, such as in text classification using word embeddings like GloVe. In the replicated study, SVMs are employed to classify lyrics into emotional categories. Each lyric is represented by GloVe 100d embedded feature vectors that capture the semantic nature of the words. SVM aims to find a hyperplane in this 100d space that best separates different sentiment classes (Figure 3.7)[40].

The decision function for a linear SVM is given by:

$$f(x) = \text{sign}(w \cdot x + b)$$

Here, w is the weight vector determined by the SVM, x denotes the input feature vector represented by GloVe100d embeddings, b is the bias term, and the class label of x is determined by the sign of the decision function $f(x) = \text{sign}(w \cdot x + b)$.

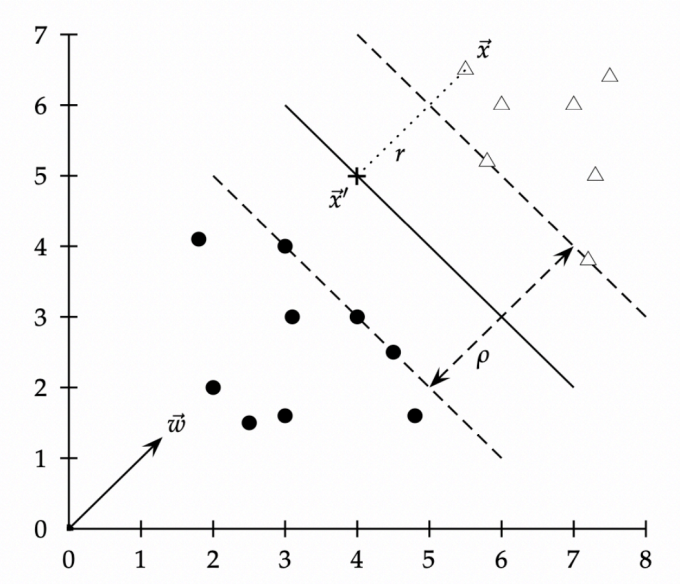

Figure 3.7: Visualization of the SVM Hyperplane

In the replicated study, the SVM algorithm was specifically configured with a linear kernel, reflecting the effectiveness of linear decision boundaries in high-dimensional spaces like those formed by GloVe embeddings. The penalty parameter C was set to its default value of 1, balancing the need for model complexity with the goal of regularizing the model. This setup helps to make a trade-off between the ability of the model to fit the data while controlling the complexity of the model, ultimately contributing to effective generalization.

3.2.5 Convolutional Neural Networks

Convolutional neural networks were primarily developed for computer vision applications and have been effectively adapted to NLP as well. A seminal work in this adaptation is Yoon Kim's 2014 paper on "Convolutional Neural Networks for Sentence Classification[41]", which demonstrated the efficacy of CNNs in text analysis tasks.

CNNs apply convolutional layers to text data, treated as sequential like pixels in images. Each layer uses a set of filters or kernels to scan through word embeddings(numerical representations capturing the semantics of words).The filter is characterized by its kernel size and learns to recognize specific patterns in the text. Smaller kernels capture local features (like n-grams), while larger ones understand broader contextual information. This enables CNNs to capture various levels of linguistic features, from simple word groupings to complex sentence structures, making them well-suited for a variety of NLP applications (Figure 3.8)[41].

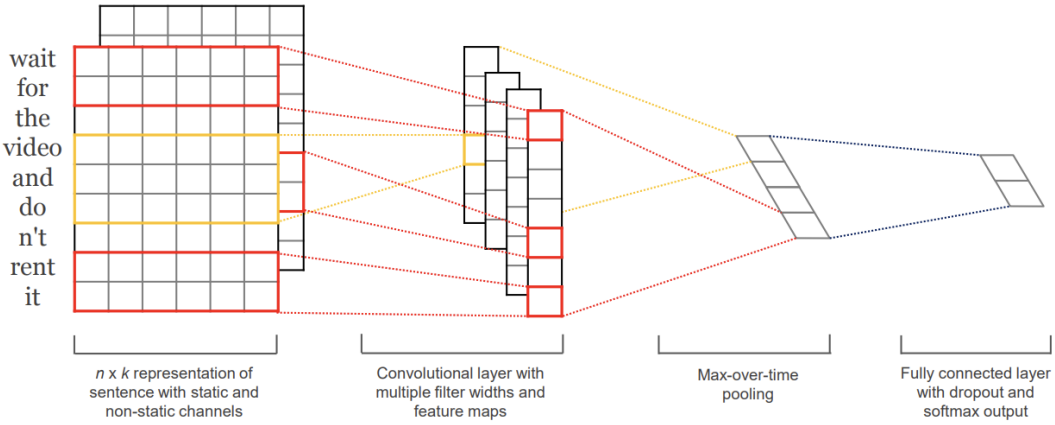

Figure 3.8: Text-CNN architecture as proposed by Yoon Kim (2014)

In the replicated study, the CNN model was specifically designed with three 1D convolutional layers, each equipped with 128 filters of kernel size 5 and the ReLU activation function. This architecture effectively processes the complex textual structures in song lyrics, with a maximum sequence length set to 1000. The model also includes three max-pooling layers with a pool size of 5, which help condense the feature maps and highlight salient features for classification. The final output layer utilizes the Softmax activation function to classify the lyrics into various emotional states. For the training of the CNN model, the ADAM optimizer with its default learning rate of 0.001 was used. The Training was conducted over 20 epochs with a batch size of 64, ensuring thorough learning while balancing computational efficiency.

3.2.6 Long Short-Term Memory and Bidirectional LSTM

Long Short-Term Memory networks, introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997[42], are a specialized form of RNN designed to capture long-term dependencies in sequential data. The distinctive feature of LSTM is its use of gating mechanisms that control the flow of information. These gates include:

- **Input Gate**: Determines how much of the new information to incorporate into the cell state.
- **Forget Gate**: Decides the amount of the previous cell state to retain.
- **Output Gate**: Controls the extent to which the value in the cell influences the output.

These gates allow LSTMs to selectively remember or forget patterns over time, which is crucial for tasks that require understanding data over long time intervals, such as in language processing (Figure 3.9)[42].

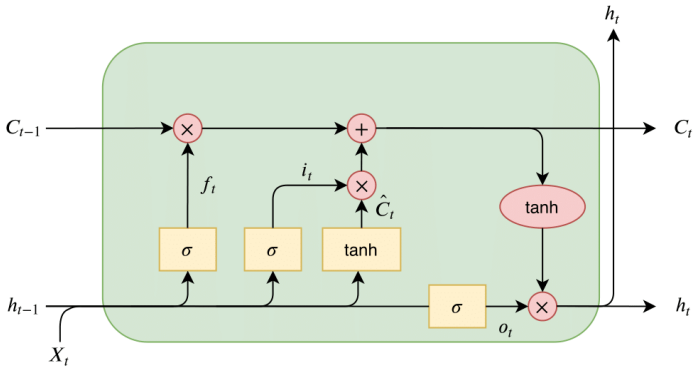

Figure 3.9: LSTM Structure

Expanding on the LSTM concept, the Bidirectional LSTM (Bi-LSTM) processes data in both forward and backward directions, allowing the model to have both past and future context. This feature is particularly useful in text analysis, where the context surrounding each word can significantly influence its meaning. The Bi-LSTM can be described by the following formulas, where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ represent the forward and backward hidden states, respectively:

$$\overrightarrow{h_t} = H(W^f_{xh}x_t + W^f_{hh}\overrightarrow{h_{t-1}} + b^f_h)$$

$$\overleftarrow{h_t} = H(W^b_{xh}x_t + W^b_{hh}\overleftarrow{h_{t-1}} + b^b_h)$$

Here, H is an activation function like tanh or ReLU, W and b are the weights and biases for each gate, and x_t is the input at time step t. The final output at each time step t in BiLSTM is then the concatenation of $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, providing a comprehensive view of the sequence from both directions.

In the replicated study, both LSTM and Bi-LSTM models were configured with identical parameters to achieve optimal performance. The models utilized Glove100d embeddings for text representation, and included layers with 100 hidden units, tailored to process sequence lengths of up to 1000.The Training was conducted with a batch size of 64, and the models employed the Softmax activation function in the output layer for effective sentiment categorization. The ADAM optimizer was chosen for its efficiency, with a learning rate set at 0.006, ensuring a balanced convergence speed. Additionally, the training was performed over 20 epochs, allowing the models sufficient time to learn and adapt to the complexities of the lyrical content.

Process of replicating

In the process of replicating the original study, we divided the dataset in accordance with the guidelines of the original research, allocating 80% for training and 20% for testing. The

text preprocessing included tokenization, lemmatization (Lemma), removal of stopwords (SR), conversion to lowercase (LC), and noise removal (NR), aimed at standardizing and clarifying the data. For the NB model, we chose Tf-idf embedding because NB classifiers typically assume that features are independently and non-negatively distributed, as shown in the multinomial model. GloVe embeddings, which can contain negative values, are not commonly used in probabilistic models like NB, as standardizing or normalizing these embeddings to fit NB could potentially lead to a loss of some original data information. Therefore, Tf-idf embedding was selected to transform the text data into a suitable format for NB. In contrast, for other models including CNN, LSTM, Bi-LSTM, KNN, and SVM, we employed GloVe embeddings as described in the original paper. This adherence to a specific embedding technique ensures a unified basis for model comparison and analysis that strictly conforms to the methodological design of the replicated study.

The purpose of this section is to detail the steps and decisions followed during the replication process to ensure a faithful reproduction of the original study, and to provide the necessary background for the subsequent results and analysis sections. If the results are close to those of the original study, the validity of the study and its authority as a benchmark will be verified. If there are significant discrepancies, we will explore the potential reasons for these differences, which may include questioning the methods of the original study or identifying issues in our implementation process. The results of this replication will be presented in Chapter 4.

Building upon the current replicated study, three key objectives have been identified for potential enhancements in sentiment analysis of song lyrics. These objectives are:

1. **Word embedding**: To explore alternative text embedding methods that may offer a more nuanced understanding and representation of lyrics.

2. **Text Preprocessing**: To refine and potentially develop more effective preprocessing strategies that could lead to better feature extraction and model accuracy.

3. **Audio Feature**: To investigate how incorporating audio data can enhance sentiment analysis, providing a more holistic view of the songs beyond just textual content.

Each objective is aimed at addressing specific limitations or areas of improvement identified in the replicated study, with the goal of advancing the robustness and depth of sentiment analysis approaches.

3.3 Word embedding

As part of the methodology for the first objective focused on embedding techniques, an initial analysis of the original data distribution was conducted. After applying the same preprocessing methods as the original study, including tokenization, Lemma, LC, SR and NR, it was observed that 97% of the lyrics in the dataset had a sequence length of 250 words or fewer post-processing. This insight led to the realization that the 1000 word sequence length used in the original paper might introduce excessive 0-padding, potentially adding noise to the dataset. Consequently, for deep learning models, the sequence length was adjusted to 250 to better match the actual distribution of the lyrics in the dataset (Figure 3.10).

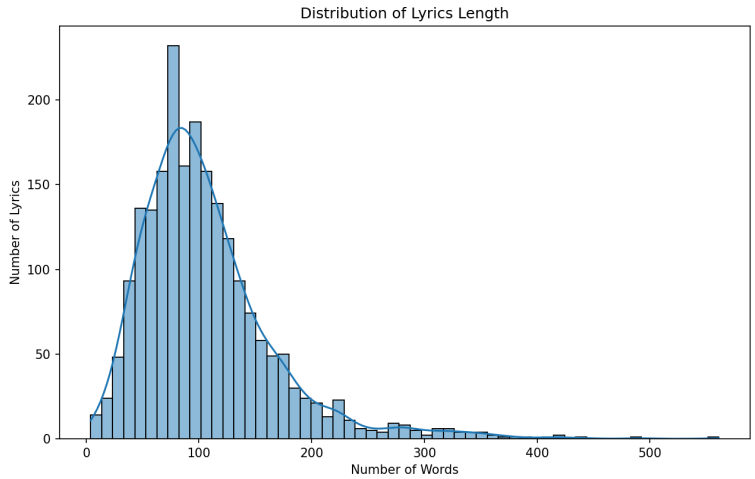

Figure 3.10: Distrilbution of lyrics in Dataset 1

In addressing the unique features of song lyrics, particularly their repetitive and rhythmic nature, the research focused on selecting suitable embedding techniques. For this purpose, Bag of Words, Tfidf, and Word2Vec were chosen. These methods are expected to be more effective in capturing the essence of lyrical content than GloVe embeddings, which are generally oriented toward global statistical analysis and might not sufficiently represent localized and repetitive lyric patterns. This selection was informed by research such as "Rhythm and Myth in the Lyrical Genre" by G. Oripova[43] and "Modeling Discourse Segments in Lyrics Using Repeated Patterns,"by K. Watanabe[44] which highlight the critical nature of these features in song lyrics. The detailed rationale behind the choice of these specific embeddings and their expected alignment with the characteristics of song lyrics will be explored in subsequent sections of the research.

3.3.1 BoW

The Bag of Words model[45] has direct text representation method, when applied to lyrics analysis may have certain advantages. Given that song lyrics often contain repetitive patterns and keywords that are crucial for understanding the overall sentiment and theme, BoW can be particularly effective in capturing these essential elements.

In the BoW model, each unique word in a set of lyrics is treated as a feature, and the document is represented as a vector indicating the frequency of each word. This representation fits well with the structure of the lyrics, as it emphasizes the presence and recurrence of significant words. For instance, in a song expressing happy or sad, words associated with these emotions will likely appear multiple times, and their frequency can be a strong indicator of the overall sentiment (Figure 3.11).

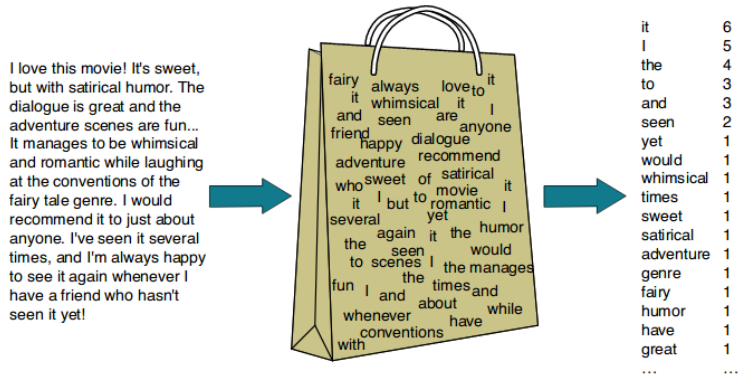

Figure 3.11: BoW Model

Mathematically, if we consider a song's lyrics as a document D and our vocabulary as V consisting of words $w_1, w_2, \ldots, w_n$, the BoW representation of D is a vector $[x_1, x_2, \ldots, x_n]$, where x_i is the frequency of word w_i in D. For example, if the vocabulary is ["love", "heart", "joy", "sad"], and the lyrics include "love" and "heart" multiple times but not "joy" or "sad", the BoW vector might look like $[2, 3, 0, 0]$, indicating the frequencies of these words in the song. This is computed as:

$$x_i = \text{freq}(w_i, D)$$

where $\text{freq}(w_i, D)$ is the number of times w_i appears in D.

While the BoW model offers simplicity and computational efficiency in analyzing song lyrics, it's important to recognize its limitations. A primary shortcoming of BoW is that it treats all words with equal importance, without considering their frequency across different documents. This can result in common words being given the same weight as more unique and potentially

more meaningful words in the context of lyrics. While certain words may overpower others when expressing the emotional tone of a song, Bow only focuses on the presence of words and fails to capture subtle meaning.

3.3.2 Tf-idf

Term Frequency-Inverse Document Frequency[46] is an advanced text representation method that improves upon the basic BoW approach. Tf-idf combines the raw frequency of a word (Term Frequency, TF) with its inverse frequency in the entire document corpus (Inverse Document Frequency, IDF), thereby balancing word frequency with its uniqueness.
Mathematically, the TF of a word is simply the number of times it appears in a document, normalized by the total number of words in that document. The formula for TF is given as:

$$\text{TF}(w, D) = \frac{\text{Number of times word } w \text{ appears in document } D}{\text{Total number of words in document } D}$$

IDF, on the other hand, is calculated as the logarithm of the number of documents divided by the number of documents containing the word. The formula for IDF is:

$$\text{IDF}(w) = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents with word } w}\right)$$

The Tf-idf score of a word in a document is then the product of its TF and IDF scores:

$$\text{Tf-idf}(w, D) = \text{TF}(w, D) \times \text{IDF}(w)$$

For instance,consider a set of song lyrics with a vocabulary consisting of the words ["love", "heart", "joy", "sad"]. If in a particular song the frequencies of these words are [2, 1, 3, 0] respectively, and their corresponding IDF values are [1.2, 1.5, 1.0, 1.7], the Tf-idf vector would be computed as [2×1.2, 1×1.5, 3×1.0, 0×1.7], resulting in a vector representation of [2.4, 1.5, 3.0, 0]. This vector effectively captures the importance and frequency of each word in the context of the lyrics, providing a more nuanced understanding of the emotional content of the lyrics.

This scoring method is particularly effective in the context of song lyrics analysis. Lyrics often contain repetitive phrases and common words that, while important, may not contribute significantly to the overall sentiment or theme of the song. Tf-idf helps to attenuate the weight of these common words (is, or, and) while highlighting unique or rare words that could be more indicative of the song's emotional and thematic content. By focusing on these significant words, Tf-idf provides a more nuanced understanding of the lyrics, capturing not

just the frequency of word usage but also its relative importance in the broader context of the song and the corpus.

3.3.3 Word2Vec

Word2Vec is a popular word embedding technique in NLP. Word2Vec, developed at Google by Mikolov et al[47], uses a neural network model to embed words from a text corpus into a high-dimensional space. It essentially learns word representations either by predicting the context of the word (in Skip-gram) or according to the context (in CBOW)(Figure 3.12)[47]. This approach captures the nuances of word usage in a particular context, making it particularly effective for understanding the meaning of words used in real-world text. When applied to lyrics, which tend to have repetitive and rhythmic patterns, Word2Vec's context-focused learning offers the potential for deeper insights. It can identify the subtle ways in which words are used within the unique structure of lyrics, potentially providing a more thorough understanding of their thematic and emotional layers.

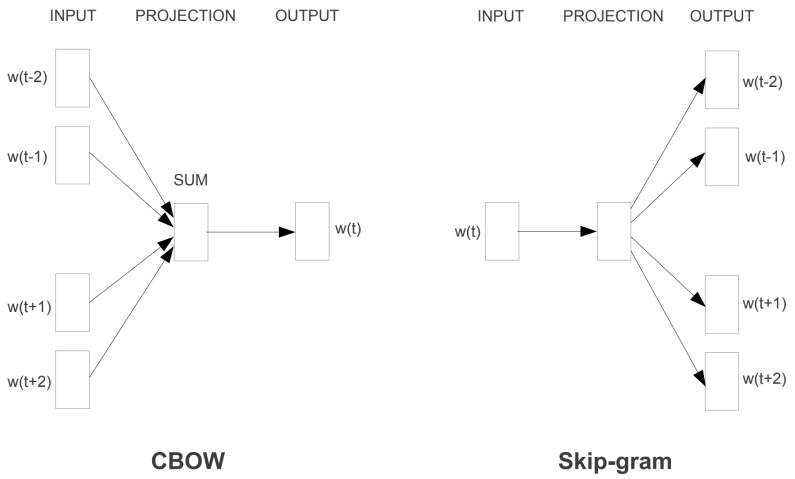

Figure 3.12: CBOW and Skip-gram

For instance, in emotion analysis of song lyrics using Word2Vec, a word like "melody" might be represented by a 300-dimensional vector such as $[0.08, -0.21, 0.09, ..., -0.05]$. Each dimension in this vector represents a unique feature that captures some aspect of how "melody" is related to various emotions based on the training data. For example, one dimension might capture its association with Happy contexts, another with Sad sentiment. This comprehensive representation allows for a deeper understanding of the emotional connotations and nuances associated with words in lyrics.

While GloVe excels in capturing broad word relationships, Word2Vec's emphasis on context may provide unique advantage, especially for analyzing lyrics. Lyrics have their own

inherent structure and often rely on repetition and rhythm to convey emotion and narrative. The context-aware nature of Word2Vec potentially allows for a deeper understanding of the emotional and thematic expressions in lyrics. To capitalize on this, we utilized a pre-trained 300-dimensional Word2Vec model for embedding. The model was developed by a team at Google and trained on a large corpus of data containing a wide range of lexical semantic and grammatical information. The depth and breadth of its training enable it to capture nuanced meanings and associations between words, which is particularly beneficial for the complex and often metaphorical language found in song lyrics. The 300-dimensional vectors provide a detailed and rich representation of words, thus providing a more refined understanding of the lyrical content.

Process of embedding

In the word embedding experiment, we applied a uniform preprocessing method Lemma+SR+NR +LC across all models. This consistency was crucial for accurately assessing the impact of different embedding techniques. For traditional models like NB, SVM, and KNN, we employed BoW and Tf-idf embeddings. In contrast, for neural network models, including CNN, LSTM, and Bi-LSTM, we used Word2Vec300d embeddings for deep semantic understanding. To optimize these models, we utilized GridSearch for the traditional models and learning and loss curve analysis for the neural networks. For CNN and Bi-Lstm, early stopping is used to find the optimal number of iterations, and for CNN, an exponentially decaying learning rate is also used to enhance training stability and prevent overfitting. This rate decreases over time according to the formula: Learning Rate $=$ Initial Learning Rate $\times\, e^{(-\text{decay rate}\times\text{epoch})}$, ensuring that adjustments to the model become more refined as training progresses. The goal was to evaluate how each embedding technique affects the performance of the model in handling lyrical content. Chapter 4 will present the detailed results and performance analysis of these embedding techniques across various models, demonstrating their effectiveness in sentiment analysis of song lyrics.

3.4 Text Preprocessing

Following the outcomes of the embedding experiments, we selected four models that demonstrated the best performance for further investigation: SVM+Tf-idf, NB+BoW, CNN+Word2Vec, and Bi-LSTM+Word2Vec. A fundamental and common step in the preprocessing stage for all these models was tokenization, a process where text is broken down into smaller units, typically words, which makes it possible for machines to understand and analyze the text effectively.

Preprocessing, especially tokenization, is essential in NLP as it directly impacts how text data is interpreted and analyzed. For example, consider the lyric line 'Echoes of heartbeats, rhythms in the dark.' When tokenized, it becomes: ['Echoes', 'of', 'heartbeats', 'rhythms', 'in', 'the', 'dark']. For song lyrics, which often contain rich and nuanced expressions, accurate tokenization ensures that each word's sentiment and meaning are appropriately captured and assessed. This step, along with other preprocessing methods like Stemming (Stem), Lemmatization (Lemma), Lowercasing (LC), Noise Removal (NR), and Stopword Removal (SR), helps refine the input data, allowing the models to focus on the most relevant aspects of the lyrics. These preprocessing techniques collectively contribute to the accuracy and efficiency of the sentiment analysis, providing a clearer insight into the emotional tone and nuances present in song lyrics.

3.4.1 Stemming

Stemming reduces words to their base or root form, which can be particularly useful in analyzing song lyrics where different forms of a word convey the same sentiment or theme. For example, stemming transforms variations like "loving", "loved", and "loves" to the root word "love", simplifying the analysis of recurring themes in lyrics.

3.4.2 Lemmatization

Lemmatization considers the part of speech and context of a word, thereby reducing it to its dictionary form. This is essential in understanding the nuanced meanings in song lyrics, where words like "better" might carry different emotional weights, and are more accurately represented when lemmatized to "good".

3.4.3 Lowercasing

Lowercasing is crucial in processing lyrics as it addresses the issue of inconsistent capitalization. It ensures words like “Moon” and “moon” in the lyrics are treated uniformly, facilitating consistent sentiment analysis.

3.4.4 Noise Removal

Noise Removal in lyric data is crucial, requiring the elimination of non-linguistic elements such as special characters and format symbols, while ensuring that meaningful linguistic aspects such as abbreviations and slang are preserved to convey emotional depth.

3.4.5 Stopword Removal

Stopword Removal in song lyrics filters out common but less significant words like 'the', 'is', and 'at'. This enables a more focused analysis on emotionally meaningful words, enhancing the understanding of the lyrics.

Process of preprocessing

To conduct a comprehensive evaluation of preprocessing methods for sentiment analysis of song lyrics, we established a systematic testing loop including four models: SVM+Tf-idf, NB+BoW, CNN+Word2Vec, and Bi-LSTM+Word2Vec. This loop was designed to focus on examining various preprocessing combinations, involving techniques such as Stem, Lemma, LC, NR, and SR. The primary objective was to assess the impact of these preprocessing methods on the ability of the model to accurately analyze and interpret the subtle complexities of lyrical content, identifying which preprocessing approach most effectively captures the sentiment of the lyrics. For deep learning models CNN and Bi-LSTM, we employed cross-validation and multiple iterations of testing to mitigate the effects of random weight initialization, ensuring the reliability and robustness of performance outcomes under different preprocessing scenarios. The following Tables 3.2 and 3.3 illustrate the preprocessing combinations tested in this research.

Preprocessing Methods (Lemma)	**Preprocessing Methods (Stem)**
Lemma + LC + NR + SR	Stem + LC + NR + SR
Lemma + NR + SR	Stem + NR + SR
Lemma + NR + LC	Stem + NR + LC
Lemma + SR + LC	Stem + SR + LC
Lemma + NR	Stem + NR
Lemma + SR	Stem + SR
Lemma + LC	Stem + LC
Lemma	Stem

Table 3.2: Combinations of Preprocessing Methods (Lemma,Stem)

Preprocessing Methods (SR)	**Preprocessing Methods (NR)**	**Preprocessing Methods (LC)**
SR + LC + NR	NR + LC + SR	LC + NR + SR
SR + NR	NR + SR	LC + SR
SR + LC	NR + LC	LC + NR
SR	NR	LC

Table 3.3: Combinations of Preprocessing Methods (SR,NR,LC)

The results of the preprocessing steps will be presented in Chapter 4.

3.5 Audio feature

While exploring sentiment analysis for lyrics, we realized that relying on textual information could be limiting. The emotional content of a song is not only conveyed through its lyrics but also significantly through its musical elements such as melody, rhythm, and tonality. Therefore, to gain a more comprehensive understanding of emotional expressions in songs, we decided to integrate audio features into the analysis.

To achieve this, we utilized a suite of audio features provided by Spotify. These audio features from Spotify represent quantitative metrics that disclose various musical aspects of a track. For instance, "danceability" measures whether a track is danceable by examining the stability of the rhythm, the strength of the beat, and the regularity of the rhythm. "Energy" measures the intensity and activity of a track through its dynamics, loudness, and timbre. Additional features, such as "Key" and "Loudness", provide quantitative measurements of the tonal center and overall volume level of the track.

The following Table 3.4 lists the primary audio features provided by Spotify with their descriptions:

Table 3.4: Description of Spotify Audio Features

Spotify Audio Feature	**Description and Range**
Danceability	Describes a track's suitability for dancing, based on tempo stability and beat strength. Range: 0.0 to 1.0.
Energy	Measures a song's intensity and activity, considering dynamics, loudness, and timbre. Range: 0.0 to 1.0.
Key	Indicates the key of the track, with values representing different musical keys. Range: 0 to 11.
Loudness	Represents the overall loudness of a track in decibels (dB). Range: Typically between -60 and 0 dB.
Speechiness	Detects the presence of spoken words in a track, with higher values indicating more speech. Range: 0.0 to 1.0.
Acousticness	Measures the acoustic quality of a track, with higher values indicating more acoustic sounds. Range: 0.0 to 1.0.
Instrumentalness	Predicts whether a track contains no vocals, with higher values indicating less likelihood of vocals. Range: 0.0 to 1.0.

Continued on next page

Table 3.4 continued from previous page

Spotify Audio Feature	Description and Range
Liveness	Detects the presence of an audience in the recording, with higher values indicating a live performance. Range: 0.0 to 1.0.
Valence	Indicates the musical positiveness conveyed by a track, with higher values suggesting happier and more positive music. Range: 0.0 to 1.0.
Tempo	Measures the overall tempo of a track in beats per minute (BPM). Range: BPM values (118.211).
Duration_ms	Provides the length of the track in milliseconds. Range: Variable, in milliseconds (237040).
Time Signature	Specifies the type of meter a track is in, indicating the rhythmic pattern. Range: Common values include 3, 4, etc.
Mode	Indicates the modality of the track, where major is represented by 1 and minor is 0. Range: 0 (minor) or 1 (major).

By integrating these audio features, our analysis goes beyond the textual dimension to provides a richer interpretation of emotions from a musical perspective, thereby capturing and understanding the overall sentiment of songs more accurately.

3.5.1 Audio Feature Data Preprocessing and Standardization

In the field of audio feature analysis, preprocessing and standardization of data are crucial steps. The heterogeneity and scale variance inherent in audio data necessitate these steps to ensure consistent and meaningful model inputs. Standardization is particularly important in transforming various audio features into formats that facilitate efficient processing by machine learning models.

Standardization was performed on continuous audio features using the formula:

$$z = \frac{(x - \mu)}{\sigma}$$

where x is the original data value, μ is the mean, and σ is the standard deviation. This approach normalizes the range of continuous features, such as 'Duration_ms' and 'Loudness', thereby enabling each feature to contribute proportionately to the final prediction. It circumvents the risk of features with larger scales disproportionately influencing he learning process of

the model, facilitating more accurate and stable predictions. After normalization, these key audio features retain their interpretability, which is a key aspect for analyzing and understanding the mood of a song.

For categorical features like 'Key','Mode' and 'Time Signature', one-hot encoding was implemented. This process transforms these categories into a binary format that machine learning models can interpret and utilize effectively. One-hot encoding expands the feature space to represent each category distinctly, ensuring that the categorical nuances of audio data are preserved and accurately represented in the analysis process.

Through these preprocessing and standardization steps, the audio data is presented in an ideal state that facilitates subsequent analysis, contributing to a more precise and insightful exploration of the emotional tone conveyed through the audio features of the song.

3.5.2 Audio Feature Clustering Tendency Analysis

After standardizing Dataset1, a crucial step was to understand its clustering tendency, which is pivotal for guiding the analytical approach in sentiment analysis. Clustering tendency is the tendency of a data set to form different groups or patterns, which is an important consideration for models that focus on pattern recognition and classification. To assess this tendency, particularly in the high-dimensional space of Dataset1, we used the Hopkins statistic[48]. This method can effectively determine whether a dataset has significant groupings with non-random distribution, which can help to select appropriate modeling techniques and methods for analysis.

The process involved generating random points, rand_X, uniformly distributed across each dimension of the dataset. Distances u from these points to their nearest neighbors within the dataset were computed, and contrasted with distances w from randomly selected data points to their second nearest neighbors. The Hopkins statistic was calculated as:

$$H = \frac{\text{mean}(u)}{\text{mean}(u) + \text{mean}(w)}$$

where u represents the distances from high-dimensional random points to their nearest neighbors, and w the distances from actual data points to their second nearest neighbors. Detailed pseudocode can be found in Appendix C: Pseudo code at Algorithm 2.

Upon applying the Hopkins statistic to Dataset 1, the values consistently hovered around 0.75. This result suggested the presence of potential clusters within the dataset, indicating that the audio features might not be randomly distributed. Recognizing this trend, we proceeded to employ PCA and t-SNE for further exploration and visualization of these potential clusters within the dataset.

3.5.3 Audio Features High-Dimensional Data Visualization (PCA)(t-SNE)

Based on the clustering trend of dataset 1 as shown by Hopkins statistics, we proceed with the high-dimensional data visualization. This step was critical in further exploring and understanding the underlying structures within the audio features. For this purpose, we employed two distinct techniques: Principal Component Analysis (PCA)[49] and t-Distributed Stochastic Neighbor Embedding (t-SNE)[50].

Principal Component Analysis (PCA)

PCA is a linear dimensionality reduction technique that reduces the dimensionality of data by identifying principal components. It starts by calculating the covariance matrix of the data. The principal components are then extracted as eigenvectors of this covariance matrix and sorted by their eigenvalues. Mathematically, this process can be represented as:

$$\text{Covariance Matrix: } \Sigma = \frac{1}{n-1}(X - \bar{X})^T(X - \bar{X})$$

$$\text{Principal Components: Eigenvectors of } \Sigma$$

where X is the original data, and $\bar{X}$ is the mean-centered data.

T-Distributed Stochastic Neighbor Embedding (t-SNE)

T-SNE is a non-linear dimensionality reduction technique particularly well-suited for embedding high-dimensional data into a space of two or three dimensions. It works by minimizing the Kullback-Leibler divergence between two distributions: one representing pairwise similarities of the input data points in the high-dimensional space, and the other representing pairwise similarities of the corresponding points in the lower-dimensional space. The cost function in T-SNE is given by:

$$C = \text{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

where P represents the probability distribution in the high-dimensional space, and Q represents the probability distribution in the low-dimensional space. The probabilities p_{ij} and q_{ij} correspond to the similarities between data points.

These visualization techniques excel at depicting linear and nonlinear aspects, respectively, enabling a comprehensive exploration of the complex, high-dimensional nature of audio data. This approach provided valuable insights into underlying patterns and relationships, revealing dimensions of data that might not be readily apparent through traditional analysis methods.

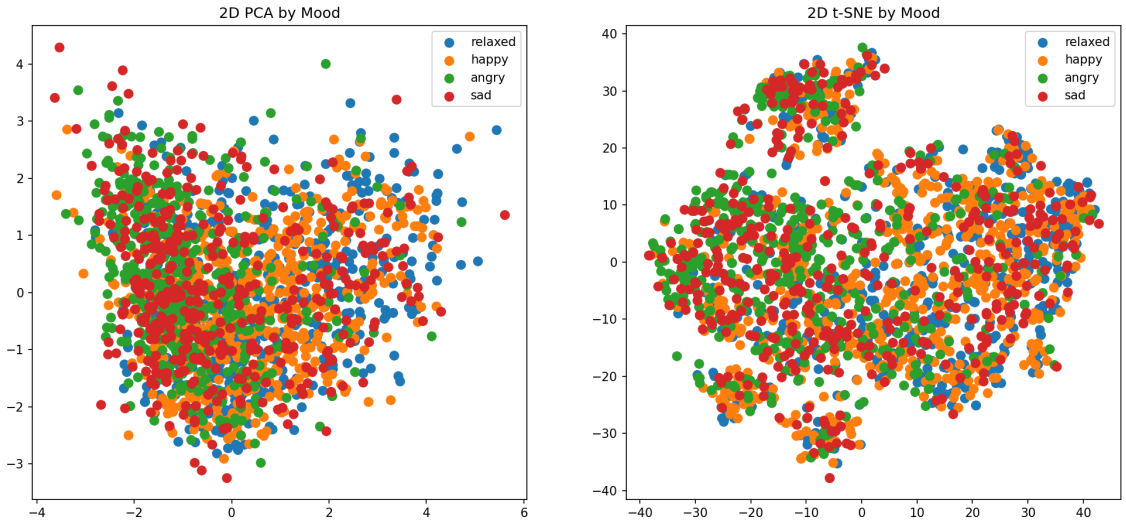

Figure 3.13: 2D PCA and t-SNE in Dataset 1

The visualization results from PCA and t-SNE in Dataset1 (Figure 3.13) indicated some degree of overlap within the entire dataset. However, a pattern emerged where most songs expressing "angry" tended to cluster towards the left side of the graph, while "happy" songs were mainly positioned on the right side. This distribution might reflect significant differences in certain audio features between music of different emotional categories. For instance, "angry" songs may have unique features in certain tonal attributes that help to group them on one side of the dimensionality reduction space after PCA and T-SNE processing. Similarly, "happy" songs could have unique features affecting their placement on the opposite side of the space.

Combining these observations with the Hopkins statistic value, which was close to 0.75, indicates that while the dataset exhibits a clustering tendency in high-dimensional space, there is some overlap in the lower-dimensional space post-PCA and T-SNE processing. However, this overlap doesn't completely negate the clustering properties of the dataset in high dimensions. The overlaps might partly result from information loss during the dimensionality reduction process. These findings provide a foundation for further analysis, guiding our to use various tools and methods for a deeper understanding of the relationship between audio features and emotional categories.

3.5.4 Audio Feature Heatmap Analysis

To deepen the understanding of how various audio features might correlate with specific emotions in songs, Heatmap[51] analysis was utilized. This method offers a visual and intuitive means to observe the complex interactions among different audio features. By representing the Pearson correlation coefficient for pairs of audio features, heatmaps facilitate the identification of significant relationships and patterns among the features, revealing their collective influence

on the emotional content of a song. The cornerstone of this analysis is the Pearson correlation coefficient, which quantifies the linear relationship between two variables and is crucial for elucidating the connections between different audio features. This coefficient is defined as:

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}}$$

where x and y represent the values of two different audio features, $\bar{x}$ and $\bar{y}$ are the mean values of these features.

The resulting coefficient r varies between -1 and 1, where 1 indicates a strong positive correlation, -1 indicates a strong negative correlation, and 0 indicates no correlation. The heatmap visualizes these correlation values, with color intensity and patterns revealing the strength and direction of the relationships between features.

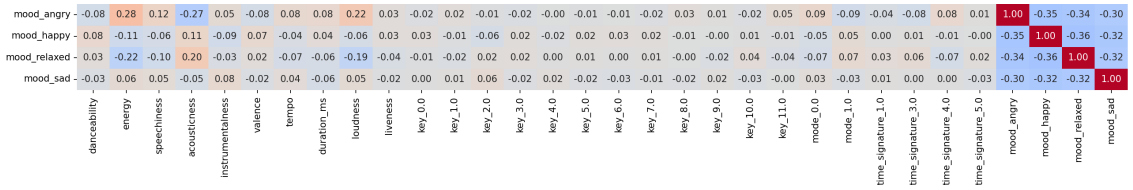

Figure 3.14: Heatmap in Dataset 1

This heatmap analysis in Dataset 1 yielded key insights into the relationships among various audio features, especially in the context of their influence on specific moods in songs (Figure 3.14). For instance, a notable positive correlation was observed between "Loudness" and "Energy" in relation to the mood classified as "Angry". This correlation suggests that tracks with higher loudness and energy levels are more likely to be perceived as embodying anger or aggression. On the other hand, these same features exhibited a negative correlation with the "Relaxed" mood, indicating that songs with lower "Loudness" and "Energy" levels tend to be associated with a more relaxed or calm emotional tone. These correlations highlight the nuanced ways in which different audio features can collectively shape the emotional character of a song, thereby enriching our understanding of the complex interplay between music and emotions.

3.5.5 Audio Feature selection

Following the Heatmap analysis, which highlighted certain audio features with weak correlations in mood prediction, we employed a combination of Random Forest (RF)[52] and Recursive Feature Elimination (RFE)[53] to accurately determine the importance of these features.

This process involves analyzing the improvement in prediction accuracy when each audio feature is incorporated into the decision trees of the Random Forest. Features that split higher

in the trees are given greater importance as they contribute more significantly to reducing the uncertainty or impurity of the dataset. The RF algorithm provides a quantifiable measure of importance for each audio feature, allowing us to rank and understand the relative significance of features like tempo, energy, and danceability in determining the emotional tone or mood of a song. When combined with RFE, this approach further enhances the efficiency of feature selection. RFE works by recursively removing the least contributing features in the RF model, enabling us to simplify the model and focus on those specific audio features that have the most substantial impact on the effective classification of song moods. This detailed method allows us to more effectively identify and utilize the most critical audio features for predicting song moods.

From the outcomes of both the RF+RFE feature importance (Figure 3.15) and heatmap analysis (Figure 3.14) that one-hot encoded features such as Key, Mode and Time Signature show very low correlation and importance. Additionally, these features significantly increased the dimensionality of the dataset, potentially leading to the curse of dimensionality. Therefore, we decided to exclude these three audio features from further analysis.

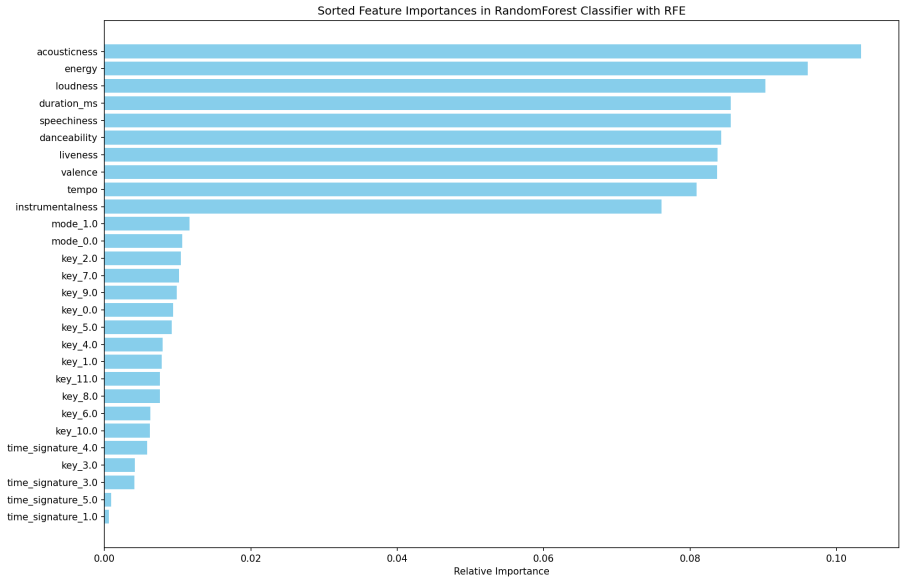

Figure 3.15: RF+RFE feature importance in Dataset 1

3.5.6 Integration of Spotify Audio Features

Initial Exploration with Audio Features only

After processing and analyzing the audio features, we employed SVM, Dense neural network layers, and RF in our experiments to explore the potential of audio features alone in sentiment analysis. The choice of these methods was based on their unique strengths. This phase was crucial for establishing a baseline understanding of the independent contribution of audio

elements to sentiment analysis. SVM was chosen for its effectiveness in high-dimensional spaces; DenseNet have the capability to recognize and process complex patterns through their interconnected structure; RF was included for its ensemble learning approach, which brings together multiple decision trees to enhance overall predictive accuracy and reliability while effectively reducing the risk of overfitting.

Integrated Approach with Textual and Audio Feature

In building upon the textual analysis, we continued to integrate audio features into the existing models, in line with our goal of enriching sentiment analysis by combining textual and audio insights. During this process, SVM was utilized for early feature fusion, combining audio and text data to take advantage of SVM's proficiency in high-dimensional spaces. This approach fused with SVM has the potential to improve overall accuracy by exploiting additional audio feature to correct misclassifications near the decision boundary. NB was excluded from this phase due to its limitation in handling negative values.

For the CNN and Bi-LSTM models, we incorporated additional Dense layers, where audio features are processed by the Dense layers, while the CNN and Bi-LSTM model handles the textual data. Finally, the outputs from both models are merged for further analysis. This bifurcated approach ensures that each data type is first understood independently, allowing the models to harness the distinctive characteristics of both lyrical content and audio feature. The dense layers assigned to handle audio data allow for a nuanced interpretation of musicality, which complements the lyrical analysis performed by the CNN and Bi-LSTM structures. In this way, the model is trained to detect complex patterns where lyric elements and audio elements are interwoven to enable a more sophisticated perception of the emotional expression in the song. The goal of this methodology is to exploit the synergistic potential of textual and audio information to capture a wider range of sentiment metrics present in music.

Advanced Ensemble Strategy

While advancing sentiment analysis model, we realized the need for a more comprehensive approach that could effectively incorporate insights from both text and audio data. To achieve this, we used stacking. Stacking[54][55] is a complex ensemble learning technique that combines multiple classification or regression models via a meta-classifier or a meta-regressor. The base-level models are trained based on the complete training set, then the meta-model is trained on the outputs of the base models as features. This approach is beneficial in sentiment analysis, particularly with complex data like song lyrics and audio features, because it exploits the predictive power of multiple learning algorithms, thereby improving overall accuracy.

The unique strength of Stacking lies in its ability to fuse different types of models, capturing different patterns and relationships in the data(Figure 3.16).

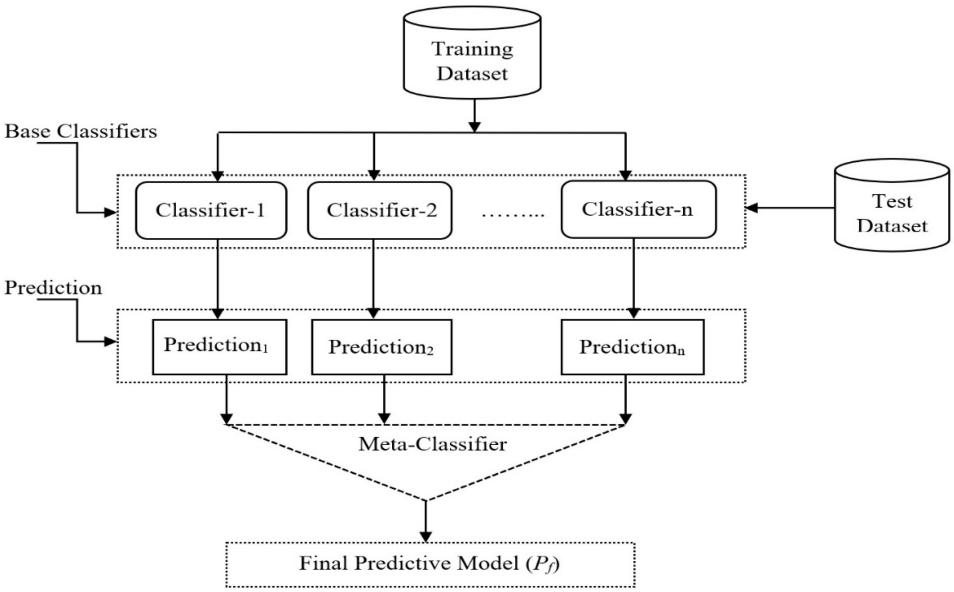

Figure 3.16: Architecture of the Stacking

XGBoost (eXtreme Gradient Boosting)[56] is a powerful machine learning algorithm that is widely used in winning solutions of various machine learning competitions. It is a scalable and accurate implementation of gradient boosting machines, known for its speed and performance. XGBoost works by building an ensemble of decision trees in a sequential manner, where each tree corrects the errors of its predecessor. This iterative optimization makes XGBoost highly effective in handling complex datasets with multiple features. In our research, XGBoost is used as a meta-classifier to interpret and weigh the predictions of the base models, SVM and RF, which perform best in text and audio analysis, respectively. Its gradient boosting framework allows it to evaluate the relative importance of different features, thus effectively combining textual and audio data insights for sentiment analysis. Because text and audio are two completely different types of data sources, XGBoost may have an advantage in integrating this kind of heterogeneous data, as it can capture complex data patterns and relationships.

Composite Model Architecture

In terms of model architecture, each model retains the core structure of previous experiments. The SVM model continues to use the same regularization technique for early fusion, ensuring that the combined data is processed in a consistent way. Bi-LSTM and CNN keep the same approach to process lyrics and keep the training parameters the same, but incorporate DenseNet to process audio features.

Process of Composite model

Upon integrating audio features into our sentiment analysis models, we noted that the performance improvement in Dataset 1 was not as significant as expected. This observation could be attributed to the specific nature of Dataset 1, which concentrates on content words in lyrics and their corresponding valence and arousal scores, according to emotion lexicons. The creation of the dataset, guided by Russell's model, focused on lyrics that exhibit strong correlations within its four quadrants. This approach may result in limited correlation between labels and audio features, leading to performance comparable to models using only lyrics.

To further validate our approach, we turned to Dataset 2, providing a different perspective. Unlike Dataset 1, Dataset 2 derives from Last.fm user tags, which integrate both audio and lyrical aspects for sentiment tagging. This variation in data origin provides a unique angle to assess the efficacy of combining audio features with textual analysis.

In our initial experiments testing models trained on Dataset 1 on Dataset 2, while overall model performance was not high, combine models consistently outperformed those relying on the lyrics modality.

To further validate the efficacy of our integrated models, we conducted a comparative analysis using Dataset 2 against an established benchmark: an XL-Net-based textual model. This approach involved replicating the structure of the XL-Net model for training and testing on Dataset 2. The results were quite significant. The integrated models, which combined both textual and audio data, not only outperformed their single-modality counterparts (text-only or audio-only models) but also surpassed the benchmark set by the XL-Net model. The findings validated the enhanced effectiveness and generalization capability of the integrated models, highlighting the value of a multimodal approach in music sentiment analysis compared to relying only on lyric or audio data types.

The results of all audio feature methodologies will be presented in Chapter 4.

Dataset 2 analyze

After training the models on Dataset 2, we proceeded to analyze its audio features using PCA, t-SNE, and heatmap visualization, similar to the approach taken with Dataset 1. This step was essential to understand the characteristics and patterns of audio features in Dataset 2, especially since it incorporates user-generated tags that reflect a combination of lyrical and musical elements in songs.

Based on the 2D and 3D visualizations, it appears that Dataset 2 demonstrates more pronounced clustering tendencies. Using PCA and t-SNE on Dataset 2 May help highlight

these unique patterns, providing a clear visual representation of how audio features relate to user-generated sentiment. This clustering trend in Dataset2 provides a notable contrast to the insights gained from Dataset1, as referenced in (Figure 3.17) and (Figure 3.18).

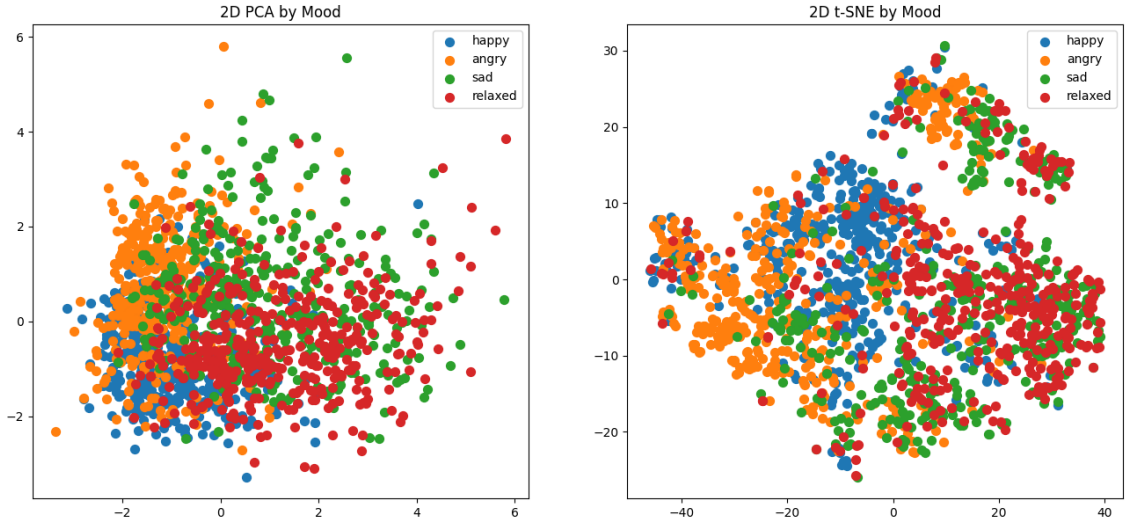

Figure 3.17: 2D PCA and t-SNE in Dataset 2

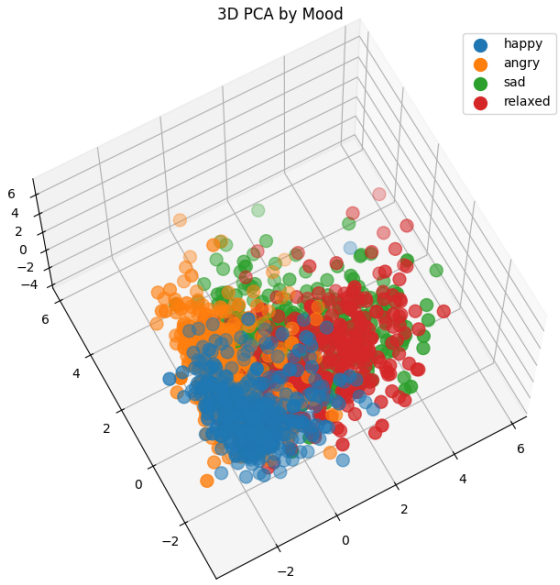

Figure 3.18: 3D PCA in Dataset 2

The heatmap analysis on Dataset 2 focused on revealing the correlations among various audio features. Analyzing these correlations after training the model on dataset 2 helped identify specific audio features that play an important role in shaping the emotional tone of the song in this unique dataset, which is different from Dataset 1. Moreover, this analysis shows that Dataset2 exhibits stronger correlations between its audio features than Dataset1 (Figure 3.19).

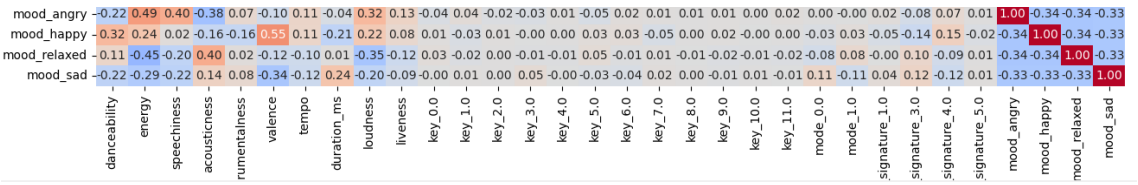

Figure 3.19: Heatmap in Dataset 2

After conducting heatmap analysis on Dataset 2, it became evident that "Energy" and "Valence" had the strongest correlation with emotions among the audio features. To delve deeper into this relationship, we created a scatter plot with "Energy" on one Y-axis and "Valence" on the X-axis. This visualization was instrumental in understanding the distribution and interaction of these two features in relation to the emotional content of the songs in Dataset 2 (Figure 3.20).

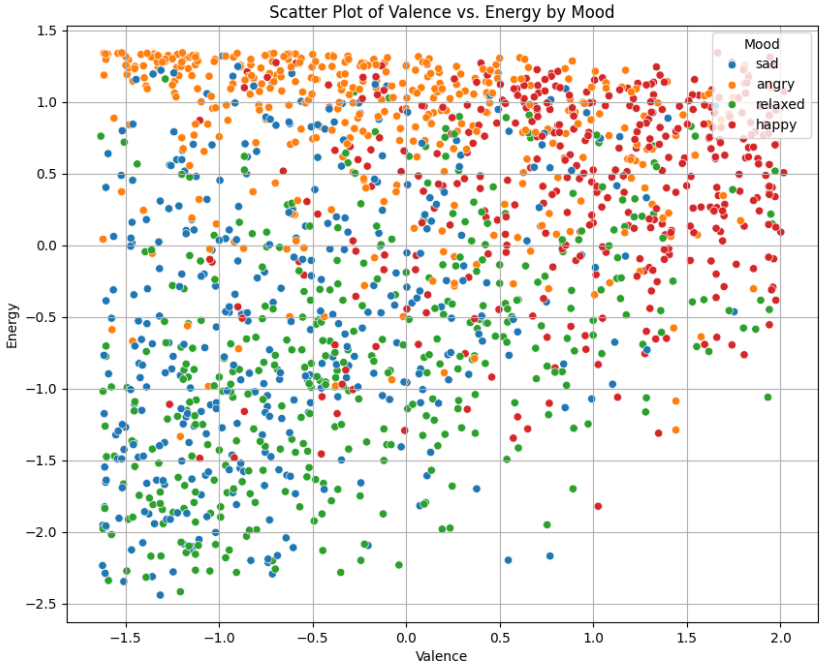

Figure 3.20: V-E in Dataset 2

Analyzing "Energy" versus "Valence" in Dataset 2, the patterns in emotional song classification became clear. Tracks in the (-V, +E) quadrant, characterized by low valence and high energy, were predominantly classified as "Angry". In contrast, songs in the (+V, +E) quadrant, with high valence and high energy, tended to be "Happy". This trend reflects how high-energy tracks usually evoke intense emotions, with valence indicating whether these emotions are positive or negative.

In the (-V, -E) quadrant, songs with low valence and low energy were primarily labeled as "Sad" or "Relaxed", although some overlap was observed between these categories. "Relaxed"

tracks often had marginally higher valence than "Sad" ones, but the distinction wasn't consistently clear. This indicates that while these audio features suggest emotional tones, they may not always definitively differentiate nuanced states like sadness and relaxation. Incorporating both lyrics and audio data might make this overlap clearer, providing a corrective measure when either modality alone leads to misclassification.

This pattern mirrors the labeling approach in both Dataset 1 and Dataset 2, where songs are classified according to valence and arousal within quadrants of the Russell model. Thus, songs in the (-V, +A) quadrant are "Angry", those in the (+V, +A) quadrant are "Happy", those in the (-V, -A) quadrant are "Sad", and those in the (+V, -A) quadrant are "Relaxed". The overlap in text and audio analysis highlights the value of considering textual and audio feature methods for music sentiment analysis, providing a more comprehensive understanding of the emotional nuances of songs.

CHAPTER 4

Result

4.1 Evaluation Metrics

This section presents the comprehensive results of our extensive experiments and analysis, combining textual and audio features in music sentiment analysis. This research investigates the results achieved by using different machine learning algorithms and deep learning architectures and the impact of combining audio features with text data on their performance. These results recapitulate the effectiveness of the various approaches explored throughout the research.

4.1.1 Confusion Matrix

In the evaluation of our sentiment analysis models, the Confusion Matrix was selected as a key metric due to its effectiveness in four-category emotion classification. This approach is particularly relevant considering that each dataset entry has an assigned emotion label that enables its classification into four different emotion categories. The Confusion Matrix provides an in-depth view of classification accuracy across these categories, not only highlighting the quantity of misclassifications but also the nature of these errors[57].A more in-depth study of misclassified cases can reveal potential biases or limitations in our feature set or training data. This detailed error analysis, guided by the Confusion Matrix, is crucial for iteratively improve the model's performance in emotion recognition. It also helps in understanding the subtleties and overlaps between different emotional states, thereby enhancing the model's ability to more effectively handle the complexity of emotions.

Confusion Matrix		Predicted Label			
		Angry	**Happy**	**Relaxed**	**Sad**
Ground Truth	**Angry**	TP	FP	FP	FP
	Happy	FN	TP	FP	FP
	Relaxed	FN	FN	TP	FP
	Sad	FN	FN	FN	TP

Table 4.1: Confusion Matrix for 4-Category Emotion Classification

In the context of sentiment analysis for song lyrics and audio features, these evaluation measures will be crucial for assessing the effectiveness of the predictive models, as shown in the Confusion Matrix (Table 4.1). For our purposes, the terms are defined in relation to the classification of song emotions:

- TP ← Correct identification of a song as belonging to its actual emotional category.
- FP ← Incorrectly labeling a song as belonging to an emotional category it does not.
- FN ← Missing a song that should have been identified as belonging to a specific emotional category.
- TN ← Correctly identifying a song as not belonging to a certain emotional category.

4.1.2 Accuracy Score

The accuracy score is a fundamental metric in sentiment analysis and measures the ratio of correctly predicted instances to the total number of instances. While it provides a quick overview of the overall performance of a model, its effectiveness may be reduced on an imbalanced dataset where one class is dominant [58]. In this case, the accuracy may not fully reflect the ability of the model to classify all classes equally. The formula for accuracy is:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

4.1.3 Precision Score

Precision Score is an essential metric in sentiment analysis, particularly when the costs of False Positives are high. It measures the proportion of correctly identified positive instances among all instances predicted as positive. This metric is crucial in scenarios where being accurate in the positive predictions is more important than the the overall accuracy of the model. However,

precision itself doesn't consider False Negatives and might not fully reflect the effectiveness of the model across all categories[59]. The formula for precision is:

$$\text{Precision} = \frac{TP}{TP + FP}$$

4.1.4 Recall Score

Recall Score is a crucial metric in sentiment analysis, particularly when it is vital to capture as many true positives as possible. This metric measures the proportion of actual positives that are correctly identified by the model. It's particularly important when the cost of missing a true positive (False Negative) is high. However, the recall does not account for False Positives, which can be a limitation if specificity is also a concern[59]. The formula for the recall is:

$$\text{Recall} = \frac{TP}{TP + FN}$$

4.1.5 F-Beta Score

The F-Beta Score is a valuable metric in sentiment analysis, as it provides a balance between Precision and Recall. It is particularly useful when we need to consider both False Positives and False Negatives in our evaluation. In this research Beta is set to 1, F-Beta score becomes F1 score, precision and recall have the same weight. This balanced approach is crucial in situations where both types of errors (False Positives and False Negatives) are equally important[59]. The formula for the F1 Score is (This metric is crucial for evaluating our model's performance):

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

4.1.6 Loss Function

In our sentiment analysis task, the dataset is categorized into four distinct sentiment classes: Angry, Happy, Relaxed, and Sad. This multi-classification lends itself well to the use of categorical cross-entropy as the loss function, which is adept at handling situations where each sample is definitively assigned to a single class[60]. Categorical cross-entropy quantifies the divergence between the predicted probability distribution outputted by the model for each class and the actual distribution, represented by one-hot encoded target labels.

$$H(y, \hat{y}) = -\left(y_{\text{Angry}} \log(\hat{y}_{\text{Angry}}) + y_{\text{Happy}} \log(\hat{y}_{\text{Happy}}) + y_{\text{Relaxed}} \log(\hat{y}_{\text{Relaxed}}) + y_{\text{Sad}} \log(\hat{y}_{\text{Sad}})\right)$$

In this equation, y is the true label vector represented using one-hot encoding, where

the vector has a length equal to the number of classes (four in this case), and the element corresponding to the correct sentiment class is set to 1, with all other elements set to 0. The predicted output $\hat{y}$ from the model is a vector of probabilities, with each element representing the model's predicted probability for each respective class. During training, the goal is to minimize this loss function across all samples in the training dataset, effectively adjusting the model's predictions to align closely with the true labels.

All of the above revealed indicators will be used for quantitative analysis of the results.

4.2 Reproduce the results of the paper

Model + Embedding Method	Accuracy (%)		F1 Score (%)	
	Original	Replication	Original	Replication
NB+Tf-idf	83%	**82%**	82%	**82%**
KNN+Glove	76%	**71%**	74%	**70%**
SVM+Glove	71%	**78%**	68%	**78%**
CNN+Glove	90%	**85%**	89%	**84%**
LSTM+Glove	90%	**88%**	90%	**88%**
BI-LSTM+Glove	91%	**88%**	91%	**88%**

Table 4.2: Comparative Performance of Original and Replicated Studies on Dataset 1

The results of the replicated model obtain comparable performance to the original study, successfully validating the earlier work. Importantly, the replication of the SVM+Glove model showed significant improvement over the performance reported in the original paper, as detailed in Table 4.2. This enhancement not only reinforces the validity and reliability of the original study but also offers insights for future research improvement.

4.3 Embedding results

Model + Embedding Method	Preprocessing Combination	Accuracy(%)	F1 Score(%)
BI-LSTM + Glove(Benchmark)	Lemma + LC + NR + SR	91%	91%
NB + BoW	Lemma + LC + NR + SR	**92%**	**92%**
NB + Tf-idf	Lemma + LC + NR + SR	90%	90%
SVM + Tf-idf	Lemma + LC + NR + SR	**93%**	**93%**
SVM + BoW	Lemma + LC + NR + SR	81%	82%
KNN + Tf-idf	Lemma + LC + NR + SR	85%	84%
KNN + BoW	Lemma + LC + NR + SR	69%	67%
CNN + Word2Vec	Lemma + LC + NR + SR	**91%**	**91%**
LSTM + Word2Vec	Lemma + LC + NR + SR	89%	89%
BI-LSTM + Word2Vec	Lemma + LC + NR + SR	89%	89%

Table 4.3: Results of Embedding Experiments on Dataset 1

The result of the embedding experiments, in which Tf-idf, BoW, and Word2Vec embeddings were applied and further fine-tuned through Gridsearch, learning, and loss curve analysis, showed that the performance of most models significantly improved compared to using GloVe embeddings in the original study. Specifically, the CNN+Word2Vec implementation achieved the benchmark performance established in the original paper, while models such as SVM+Tf-idf and NB+BoW not only met but exceeded this benchmark, as detailed in Table 4.3.

4.4 Preprocessing results

Model + Embedding Method	Best Combination	Accuracy (%)	F1 Score (%)
BI-LSTM + Glove(Benchmark)	Lemma + LC + NR + SR	91%	91%
NB + BoW	Lemma + LC + NR + SR	**92%**	**92%**
SVM + Tf-idf	LC + NR + SR	**94%**	**94%**
CNN + Word2Vec	LC + NR + SR	**92%**	**92%**
BI-LSTM + Word2Vec	Lemma + NR + SR	90%	90%

Table 4.4: Results of Preprocessing Experiments on Dataset 1

The results of the preprocessing experiments, as displayed in Table 4.4, show a significant improvements in model performance due to customized preprocessing strategies. Techniques such as GridSearch,learning, and loss curve analysis have been instrumental in fine-tuning the preprocessing for each model. Notably, each model has shown enhancement, as depicted in Figures 4.1 and 4.2, which show the learning curve of the SVM before and after preprocessing.Detailed data on the performance of each preprocessing group are presented in Appendix B, as referenced in Appendix B: Preprocessing test results.

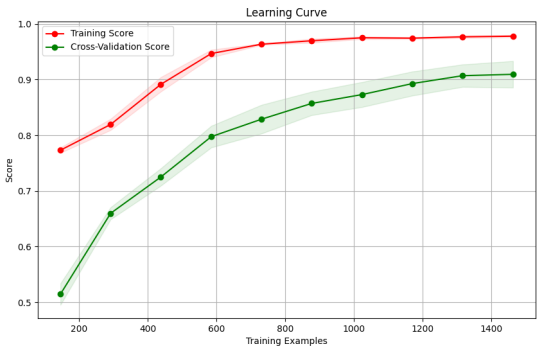

Figure 4.1: Learning curve of SVM+Tf-idf before the best preprocessing

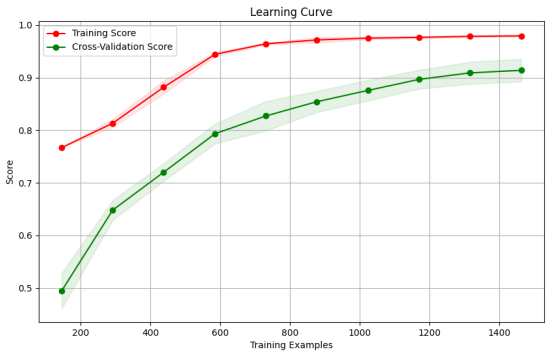

Figure 4.2: Learning curve of SVM+Tf-idf the best preprocessing

4.5 Lrycis-only Model architecture and parameters

The parameter settings for each model after text lyric processing are as follows:

NB+BoW

The model utilizes a MultinomialNB classifier with an alpha parameter of 0.5 to handle frequency data and smooth probability calculations. Text is transformed into a BoW format using the CountVectorizer, with parameter settings of a maximum document frequency of 0.5, a minimum document frequency of 3, and an n-gram range of (1, 2).

SVM+Tf-idf

This model employs a linear kernel SVM for text classification. The regularization parameter C is set to 0.36, balancing model complexity and accuracy. Text data is processed through Tf-idfvectorization, where the TfidfVectorizer's parameters include a maximum document frequency of 0.5, a minimum document frequency of 0.001, and an n-gram range of (1, 1).

CNN+Word2Vec

The CNN model is structured to process input sequences of length 250. It consists of three convolutional layers, each followed by a max-pooling layer and a dropout layer, which are included to mitigate the risk of overfitting. The initial layer of the model, an embedding layer, is responsible for mapping word indices into a 300-dimensional space using pre-trained word vectors. The convolutional layers are designed with kernel counts of 128, 64, and 32, respectively, and each has a kernel size of 5. Following each convolutional layer, max-pooling with a pool size of 5 and dropout layers with a rate of 0.2 are employed. The network also includes a global max-pooling layer that connects to a fully connected layer, followed by a softmax activation layer for multi-class classification, which is used for the classification task. The training parameters of the model include epochs, with early stopping set at 20 epochs, a batch size of 16 samples, an initial learning rate of 0.0005, and an exponential decay rate of 0.1.

Bi-LSTM+Word2Vec

The input layer processes sequences of length 250. An embedding layer generates word embedding vectors, followed by a dropout layer with a 0.2 dropout ratio. The bidirectional LSTM layer consists of 100 hidden units, reading the input from both directions and merging their outputs. Finally, a fully connected layer with a Softmax activation function is used for multi-class classification. Trained for 30 epochs, with early stopping criteria set based on the performance of the model, using batches of 16 samples. Employed an Adam optimizer with a learning rate of 0.0001.

4.6 Audio Feature Results

Dataset 1 Audio-Only

Model	Accuracy (%)	F1 Score(%)
SVM	39%	36%
DenseNet	38%	37%
RF	40%	40%

Table 4.5: Performance of Audio Only Model in Dataset 1

The results in 4.5 show that models relying only on audio features perform poorly on Dataset 1, indicating a weak correlation of audio features in this dataset.

Dataset 1 Train and Test with Lyrics and Audio Feature

Model + Embedding Method	Accuracy (%)	F1 Score (%)
SVM + Tf-idf (Early Fusion)	89%	89%
CNN + DenseNet+Word2Vec	**92%**	**92%**
Bi-LSTM + DenseNet+Word2Vec	90%	90%
Stacking	91%	91%

Table 4.6: Performance comparison of combine models on Dataset 1

Based on the results in Table 4.6, the performance comparison of various models on Dataset 1, which involved training and testing with both lyrics and audio features. In our analysis of these results, we observed that integrating audio features did not substantially improve model performance, which may be due to the fact that the Dataset 1 only focuses on the lyric content with strong emotional sentiment. Notably, among the models tested, CNN+DenseNet+Word2Vec demonstrated best performance, suggesting that some models may be better at exploiting audio features. Here is the learning curve (Figure 4.3), loss curve (Figure 4.4), and architecture (Figure 4.5) of CNN + DenseNet+Word2Vec. Consequently, it becomes crucial to evaluate the generalization of audio features in different contexts. Dataset 2, with its user-centric perspective, provides a contrasting setting to Dataset 1.

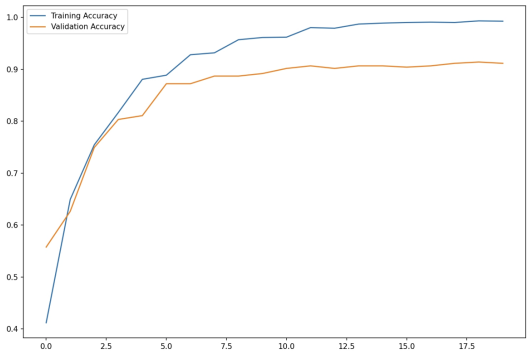

Figure 4.3: Learning curve of CNN + DenseNet+Word2Vec

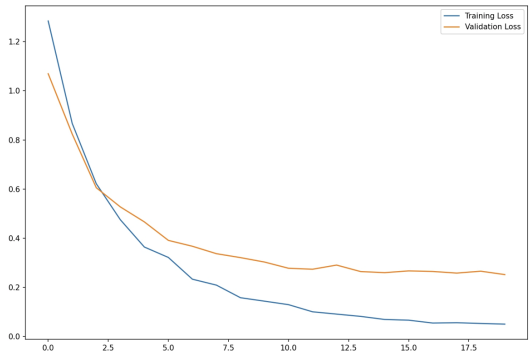

Figure 4.4: Loss curve of CNN + DenseNet+Word2Vec

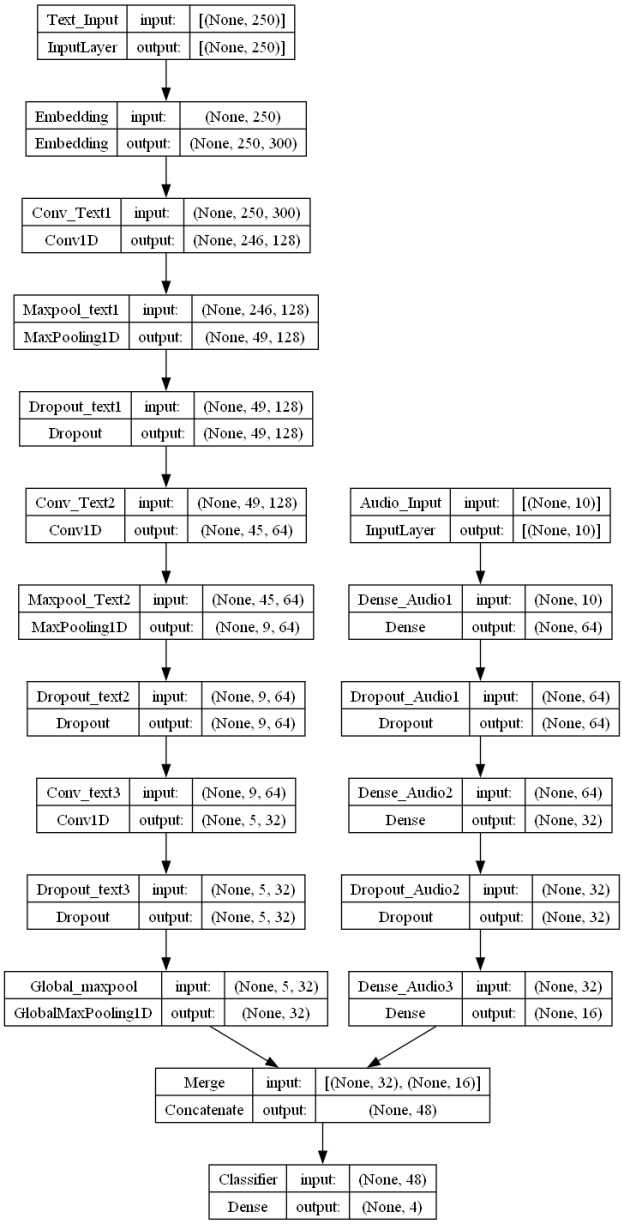

Figure 4.5: Architecture of CNN+DenseNet+Word2Vec

Dataset 2 Test with Dataset 1 model

Model + Embedding Method	Configuration	F1 Score(%)
SVM+Tf-idf	Lyrics Only	32%
SVM+Tf-idf(Early Fusion)	Combined	**34%**
CNN+Word2Vec	Lyrics Only	36% (Angry F1: 49)
CNN+DenseNet+Word2Vec	Combined	**38% (Angry F1: 54)**
Bi-LSTM+Word2Vec	Lyrics Only	35%
BI-LSTM+DenseNet+Word2Vec	Combined	**37%**
NB+BoW	Lyrics Only	35%
Stacking	Combined	**37%**

Table 4.7: Comparison of F1 Scores across Different Models and Configurations (Trained on Dataset 1 and Tested on Dataset 2)

Based on the results in Table 4.7 , we observed that despite generally low F1 scores, the models exploiting the combination of lyrics and audio features outperform those relying only on lyrics. This trend was particularly notable in the CNN model, which displayed significant improvements. To further affirm the value of audio features, we trained and tested using Dataset 2 with the same model parameters and architecture. In addition, studies that focus only on the lyrics in dataset 2 using the XL-Net model are used as a benchmark for comparison. This methodical approach aimed to provide a comprehensive assessment of the models' performance, reinforcing the relevance of audio features in music sentiment classification.

Dataset 2 Train and Test

Model + Embedding Method	Configuration	F1 Score(%)
XL-Net + Lemma(Benchmark)	Lyrics Only	59%
SVM+Tf-idf	Lyrics Only	54%
SVM+Tf-idf(Early Fusion)	Combined	**62%**
CNN+Word2Vec	Lyrics Only	57%
CNN+DesenNet+Word2Vec	Combined	**68%**
Stacking	Combined	**64%**
NB+BoW	Lyrics Only	52%
BI-LSTM+Word2Vec	Lyrics Only	53%
BI-LSTM+DesenNet+Word2Vec	Combined	**64%**
SVM	Audio Only	61%

Table 4.8: Comparison of F1 Scores across Different Models and Configurations (Trained and Tested on Dataset 2)

Based on the results in Table 4.8, it was found that the composite models exhibited better performance in Dataset 2 compared to models using either lyrics or audio features in isolation. Notably, the CNN+DenseNet+Word2Vec achieved an F1 score of 68%, which is significantly higher than the benchmark of 59% reported in the reference paper and surpasses the perfor-

mance of both other individual and composite models.

4.7 Combine model architecture and parameters

SVM + Tf-idf (Early Fusion)

For the SVM + Tf-idf (early fusion), the same word vectors and parameters as the previous single-text Tf-idf were used, with a regularization parameter C set to 0.36.

CNN and Bi-LSTM models

For both CNN and Bi-LSTM models, the text processing components remained unchanged from their previous configurations. In parallel, DenseNet was utilized to process audio features, with a shared architecture across both models. This architecture consists of an audio input layer followed by three Dense layers with 64, 32, and 16 units, respectively, each with ReLU activation for hierarchical feature extraction. Dropout layers with a rate of 0.2 are interleaved between Dense layers to reduce overfitting. The processed audio features are then merged with the text feature layers. The combined features feed into a Dense classifier layer with a softmax activation function corresponding to the number of output labels. For the CNN model, early stopping is set to 20 epochs, and an exponential decay rate of 0.1 is applied. The models are compiled using the categorical_crossentropy loss function, optimized with Adam. Additionally, the training parameters for both CNN and Bi-LSTM models, including epochs, batch size, and learning rate, remain consistent with their previous configurations for text processing.

Stacking

For stacking, following optimization with Grindsreach, our model employs a layered approach for comprehensive data analysis.The SVM component, designated for text data processing, maintains consistent parameters and architecture as established in previous text handling iterations.In parallel, for the audio data, a RF Classifier with n_estimators set to 100 has been optimized for maximum efficiency in interpreting audio features.Central to the architecture of our model is the XGBoost Classifier. Configured with 100 estimators, a learning rate of 0.4, a maximum depth of 3, and softmax was used for multi-class classification.

CHAPTER 5

Evaluation and Discussion

5.1 Lyrcis Evaluation

In evaluating the repeatability results of our study, we validated the authority of the baseline paper and found the limitations of GloVe embeddings related to max_sequence and lyric text preprocessing for deep learning. A key limitation found in this paper is GloVe's global focus, which is often less aligned with the complex requirements of processing lyric text. Subsequent experiments indicated that Word2Vec, Tf-idf, and BoW were significantly more effective than GloVe in handling lyric texts. This superior performance is largely due to their emphasis on context and the importance of individual words. This approach more effectively matches the unique properties of music lyrics, which are characterized by rhythmic and repetitive patterns[43][44]. These findings highlight the effectiveness of embeddings that focus on contextual understanding and word-level significance, in contrast to GloVe's broader, more global approach to text representation.

In preprocessing, particularly for lyrics, it was observed that after preprocessing, 97% of the lyrics in both Dataset 1 and Dataset 2 fall within a 0-250 sequence length range. Establishing this specific length was crucial in minimizing the noise from max_sequence 0-padding, thereby enhancing the model's performance [61]. Our research also found that various models perform differently to distinct preprocessing methods. The widespread use of abbreviations and slang in lyrics necessitated steps like NR and SR are required for each model, which significantly improved both the accuracy and generalizability of the models. However, the stem method, which is designed to reduce words to their root form, was less effective. Much of this reduced

effectiveness can be attributed to the potential of stem to change the original meaning or context of words in lyrics [62], an important aspect of precise sentiment analysis.

After comparing the performance of different models in the task of sentiment classification of song lyrics, we observed that SVM and CNN models perform better without Lemma preprocessing, while NB and Bi-LSTM models show improved performance with Lemma preprocessing. This finding is closely related to the processing mechanisms inherent in these models and their sensitivity to lexical changes.

Particularly, the combination of NB+BoW model and Bi-LSTM+Word2Vec both exhibited high sensitivity to word form changes. NB, which relies on statistical analysis of word frequencies, benefits from lemma preprocessing as it consolidates different forms of the same word and thus improves the accuracy of frequency estimation, which is crucial for capturing subtle emotional nuances in lyrics. Concurrently, Bi-LSTM uses its long-term dependency capture ability and the rich semantic information of Word2Vec to more accurately interpret the sentiment in the lyrics. Lemma preprocessing provides a more unified form of thesaurus and helps the model better understand emotional coherence[63].

Conversely, models combining SVM+Tf-idf and CNN+Word2Vec demonstrate strong robustness to lexical form variations. SVM effectively differentiates between emotional categories by constructing optimal boundaries, not overly relying on individual word forms[64]. Simultaneously, CNN identifies key patterns and features from the embedding [41] of Word2Vec, which usually goes beyond the word surface form, so that lemming preprocessing has less impact on the performance of CNN.

Notably, while both CNN and Bi-LSTM utilize Word2Vec embeddings, their approaches to processing these embeddings are distinct. CNN focus on capturing immediate, local features in embeddings and are good at identifying specific sentiment indicators in text [41]. This ability is particularly effective in processing lyrics, because the rhythmic and repetitive nature of lyrics coincides with the ability of CNN to identify and interpret local patterns. On the other hand, Bi-LSTM, with its ability to understand context over longer text spans, benefits more from the consistent word forms offered by lemma preprocessing, utilizing the depth of Word2Vec to grasp the overall emotional narrative in lyrics[42]. However, according to the results, CNN are better at handling lyrical content.

Finally, In the lyric-only experiment,the combination of optimal preprocessing improves the accuracy and verification set by 1%, the SVM+Tf-idf achieved a top accuracy of 94%, surpassing the baseline set in the research. This result also shows that SVM and NB using BoW and Tf-idf embeddings perform better on this dataset compared to other deep learning models, which may be due to the higher sentiment weight of specific words in the lyrics. In

certain emotional categories, the frequency of some keywords is notably higher than in others, making these words significant indicators of emotion[65]. For instance, as shown in Table 5.1, in the "Happy" category, the word "love" carries substantial weight. For the BoW method, which is based on word frequency, this means these words play a decisive role in classification.

Simultaneously, when using the Tf-idf method, these words are not only significant due to their high frequency in specific categories but also gain higher weight(IDF) due to their uniqueness across the entire dataset. This approach helps emphasize words that are significant to particular emotional categories but less common in other parts of the dataset. Therefore, the specific distribution of these emotionally charged words provides a powerful distinguishing feature for models employing these embedding methods, thereby enhancing the accuracy of emotion classification.

Relaxed	**Happy**	**Angry**	**Sad**
home (1266)	love (9535)	fire (1153)	lonely (736)
baby (1019)	baby (1244)	war (653)	time (560)
girl (1010)	know (1077)	like (602)	know (511)
love (724)	oh (1053)	know (557)	like (452)
oh (717)	like (754)	oh (520)	get (422)

Table 5.1: Top words per mood category with their counts in Dataset1

5.2 Audio and Combine model Evaluation

Regarding audio features, the results during the feature selection stage revealed important considerations regarding the impact of each audio feature on emotions. This can reduce the curse of dimensionality[66], allowing the model to focus more on audio features relevant to the emotional dimension, thereby enhancing model performance and generalization. When analyzing the emotional dimensions of individual audio features, their impact in Dataset 1 was relatively minor. This is primarily attributed to Dataset 1 considering only the strong emotional dimensions of lyrics based on the Russell model when labeling the dataset. However, Dataset 1's visualizations and heatmaps did reveal some correlations with audio features, but including these features in the composite model did not significantly enhance accuracy in Dataset 1 tests. In terms of performance, CNN+Densenet+Word2Vec is comparable to CNN+Word2Vec with lyrics-only modality. Due to observations that the validation set loss and performance did not converge well without exponential decay, tending to overfit quite early, early stopping and exponential decay methods were employed. Comparatively, models using early stopping and exponential decay(Figure 5.2) showed smoother and better-fitting training curves than without exponential decay(Figure 5.1). By focusing on the learning process of the model

and preventing overfitting, this approach contributes to more efficient training and potentially better generalization ability.

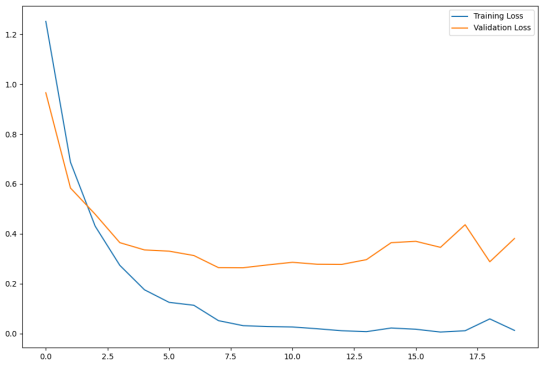

Figure 5.1: CNN Model without Decay

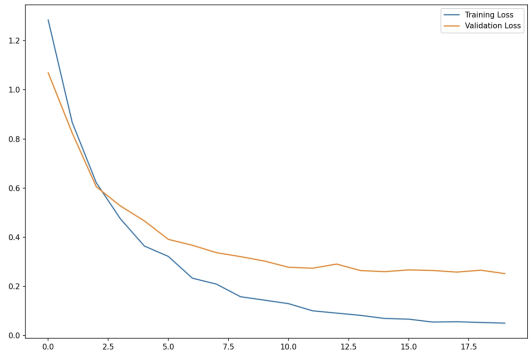

Figure 5.2: CNN Model with Decay

Therefore, when further considering the generalizability of the model, introducing a second more diverse music dataset provided additional insights. Despite lower overall results when testing the model trained on Dataset 1 with Dataset 2, primarily due to 7,775 unseen tokens in Dataset 2 causing out-of-vocabulary issues, the composite model exhibited better generalization capabilities compared to single-modality models. The higher performance of the composite model observed in testing may be attributed to the findings of the heatmap for Dataset 1. These heatmaps indicated a strong positive correlation between "Angry" emotions and audio features such as Energy and Loudness, and a negative correlation with the "Relaxed" emotion. This pattern suggests that songs with higher Energy and Loudness are most likely indicative of "Angry", while those with lower Energy and Loudness are most likely representative of "Relaxed". This was validated when the composite models and lyrics-only models trained on Dataset 1 were tested on Dataset 2. For the 'angry' class, the composite model CNN+Densenet+Word2Vec showed a 5% improvement in F1 score compared to the single-modality CNN+Word2Vec. This indicates that the models trained on Dataset 1 had effectively learned this correlation, although it is not evident in Dataset 1. The overall audio feature visualizations and heatmaps in Dataset 2 also exhibit stronger patterns and linear correlations compared to Dataset 1, suggesting that such user-subjective perspective emotion labels are more sensitive and salient to audio features in Dataset 2.

To further validate the proposed method, based on the literature, the same model architecture and parameters are trained and tested on dataset 2 as a benchmark. The results showed that the proposed composite model significantly outperformed the single-modality models and exceeds the baseline established in reference [26]. The CNN+DenseNet+Word2Vec model achieved a 68% F1 score. This success underscores a key discovery: The same emotional dimensions of lyrics and audio features can map onto the same quadrants of Russell's model.

In lyric classification, it was observed that the quadrant (-V, +A) corresponds to "Angry", (+V, +A) to "Happy", (-V, -A) to "Sad", and (+V, -A) to "Relaxed". Similarly, in our audio analysis, (-V, +E) corresponds to "Angry", (+V, +E) to "Happy", (-V, -E) to "Sad", and (+V, -E) to "Relaxed". These results validate the effectiveness of incorporating Energy (E) alongside Arousal (A) in Russell's model, providing a subtle understanding of musical emotion.

The experimental results also show that SVM outperforms CNN when trained and tested only on the pure lyrics modality of Dataset 1, but the situation is different for the other test configurations. When trained on the single lyric modality of Dataset 1 and then tested on Dataset 2, as well as in the composite model trained on Dataset 1 and tested on Dataset 2, and when both trained and tested on Dataset 2, CNN consistently outperform other models including SVM in terms of performance and generalization ability. The superior generalization of CNNs compared to SVMs can be attributed to the deep learning architecture of CNNs, which excels at capturing complex textual patterns specific to lyrics. This includes nuanced contextual and semantic variations, critical for accurately interpreting emotions in lyrics, which linear approaches of SVMs might not recognize as effectively. The convolutional layers of CNNs are particularly effective in this context, enhancing the model's ability to adapt and accurately assess sentiments across a diverse range of lyrical scenarios, leading to stronger generalization in lyric sentiment analysis.

This alignment between the emotion classification methods for music and the V-A and V-E model quadrants demonstrates a significant parallel in how both textual and audio features capture the emotional essence of a song. This finding confirms that latent patterns in textual and audio features described by Russell's model can more accurately define the emotional context of a song. This relationship highlights the advantages of integrating textual and audio features to enhance the model's generalizability and accuracy. By leveraging the strengths of both modalities, the approach provides a comprehensive and nuanced understanding of the emotional landscape of songs, which is conceptualized by the V-A and V-E models. This integrated approach has proven to be particularly effective in capturing the complex emotional expressions inherent in music.

5.3 Use case Evaluation

After obtaining these results, we further assessed the effectiveness of the model by applying it to analyze the emotions of the top 100 Spotify songs for each year from 2013 to 2023, relating the findings with historical events to demonstrate the model's wide applicability. This analysis provides a reliable dataset for music emotion analysis, a field currently hampered by the lack

of publicly available datasets that simultaneously consider both lyrics and audio, primarily due to copyright issues. Through cross-validation and accuracy analysis, the CNN model exhibited the best average generalization capability and precision across both Dataset 1 and Dataset 2, which led to the selection of this task.

The results in Figure 5.3 revealed two parallel trends from 2020 to 2022 in the emotional content of top Spotify songs. First, there was an increase in the prevalence of "sad" songs, peaking in 2022. This rise may reflect the collective appeal of more introspective and contemplative themes in music, possibly reflecting the public mood over the years. At the same time, we observe a significant reduction in the number of "happy" songs, dropping to the lowest level in the last five years. This trend may indicate a reduction or popularity of the production of more optimistic, upbeat musical themes, which may be a reaction to the global atmosphere of the time, influenced by significant social and economic challenges.

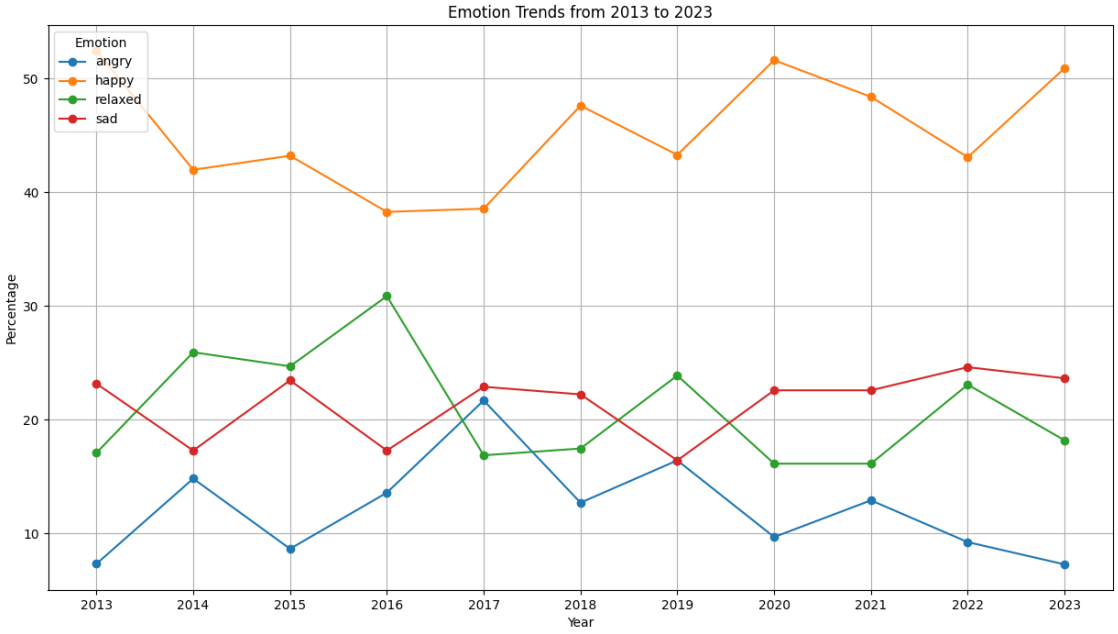

Figure 5.3: Predictive Trends of Spotify Top 100 Tracks from 2013 to 2023

A survey of the authoritative literature from 2020 to 2022 validates the validity and reliability of the proposed model and dataset, especially with respect to social events during this period, including the COVID-19 pandemic from 2020 to 2022 and the Russia-Ukraine conflict from 2022. These external events indirectly confirm the trends observed in the predictions of our model.

A large-scale survey during the COVID-19 pandemic reported by Neuroscience News[67] revealed that individuals experiencing increased negative emotions used music for emotional regulation, while those with positive feelings turned to music as a social interaction substitute. Our model is consistent with these findings, showing an increase in sad songs and a decrease

in happy songs during the pandemic to meet changing socio-emotional needs. Moreover, the study "Music and mood regulation during the early stages of the COVID-19 pandemic"[68] found that during the quarantine period of the pandemic, individuals feeling stressed or sad could improve their mood by listening to negatively-valanced music. This indicates that there is a strong connection between emotional states and music preferences during periods of heightened stress and isolation, particularly in times of heightened stress and isolation.

Another study highlighted by Denk et al. [69] emphasizes the significant impact of the COVID-19 pandemic on overall music market consumption and consumer spending, which may have influenced the emotional content of music produced during this period.

The Russia-Ukraine conflict in 2022, which coincided with the coronavirus disease pandemic, had a significant impact on mental health globally. Numerous studies and reports, including those from the United Nations, have focused on the serious global mental health implications of the simultaneous Russia-Ukraine conflict and the COVID-19 pandemic. These studies highlight that the dual burden of the conflict and the pandemic has exacerbated mental health conditions such as depression, anxiety, and stress [70]. Notably, an increase in symptoms like war anxiety among large populations indicates that these concurrent crises have had a widespread and severe impact on emotions [71]. The conflict has not only disrupted Ukraine's healthcare system, exacerbating psychological stress in the country already burdened with COVID-19 pandemic and limited medical supplies, but its effects have also extended globally, intensifying the stress caused by the ongoing pandemic [72].

Together, these insights highlight how societal events like the COVID-19 pandemic and the Russia-Ukraine conflict from 2020 to 2022 profoundly affected individual emotional states and coping strategies, subsequently influencing their music preferences. The trends observed in our model are to some extent confirmed by these studies. This correlation emphasizes the ability of the model to capture subtle changes in music preferences during times of global stress and uncertainty, further evaluating and validating the effectiveness of the model.

CHAPTER 6

Limitation and Future Work

6.1 Selection of Audio Features

In our research, we utilized Heatmaps and RF+RFE to assess the importance and relevance of certain audio features. However, this approach may not have fully considered the impact of features like key, mode, and time signature on emotional analysis. Reflecting on this, future work should deeper into analyzing these audio features to determine whether they indeed negatively impact model performance or if they could be significant in certain emotional expressions or musical genres. Expanding this investigation, applying various statistical methods or machine learning models promises to more accurately determine the value and role of these features in different contexts, thus making the selection of audio features more scientific and comprehensive.

6.2 Dataset Enhancement

Building upon the current approach, the size of Datasets 1 and 2 and the richness of the vocabulary may affect the performance of the model. Specifically addressing this, future work could involve expanding the vocabulary(Lexicon), including words less prevalent in the existing dataset, such as slang. It enhances the model's ability to understand and process multiple lyrics types. This enhancement is expected to improve the model's ability to handle complex textual content, especially lyrics containing uncommon or culturally specific vocabulary, thereby enhancing the model's generalization ability and accuracy.

6.3 Confidence Analysis in Lyrics and Audio Modalities

In our current research, when integrating text and audio modalities into a composite model, we did not consider the confidence scores between these modalities. To address this issue, future work could involve a more precise approach to independently analyze the confidence level of each modality across different emotional categories. For example, "Angry" might be more prominently represented in audio features, while "Sad" might be more easily detected in text features. To enhance the modal fusion strategy, it might be beneficial to use the Transformer's QKV (Query, Key, Value)[73] mechanism with an attention model before the fusion of modalities. This attention-based QKV mechanism would allow for a more effective evaluation of each modality's contribution to emotional prediction. The Query in this mechanism represents the current focus or information being processed; the Key is what the model uses to match the Query; and the Value provides the relevant content or data. For instance, in music sentiment analysis, identifying a specific emotional category could be seen as a Query, with text and audio features serving as Key and Value, thus analyzing which modality more effectively represents a given emotion. In addition, using the weighted average method to combine the confidence of the two modalities is another method worth exploring. The proposed method trades off each modality according to its reliability in predicting a specific emotion category, resulting in a more accurate fusion of text and audio data. These approaches are expected to enhance the ability of music sentiment analysis models to recognize and understand complex, multidimensional emotional states in future research by combining more detailed confidence analysis in text and audio modalities.

6.4 Causal Relationship between Lyrics and Audio Data

Lastly, in our research, the model still processes Lyrics and audio data independently to some extent, and does not explicitly address their dynamic relationship or interaction before combining these data. Therefore, it does not consider the dependency relationship between text and audio. Future work in this area could consider undertaking combined feature engineering to analyze how specific types of lyrics (such as Happy or Sad) usually combine with certain audio features (like Danceability, Loudness). Moreover, further work may include treating pure audio signals from music as time-series data, and using temporal models with attention mechanisms to process these audio data and lyrics, capturing the evolving emotional dynamics in music. This approach is expected to enhance the understanding of the causal relationship between lyrics and audio.

CHAPTER 7

Conclusion

In the conclusion of this research, we comprehensively review all aspects from the theoretical foundation to the concrete practice of MER. Through in-depth chapter analysis, this study not only reveals the complexity of MER, but also proposes innovative solutions to current challenges. Each chapter shows the whole process from data processing to model evaluation in detail, highlighting the important contribution of this research in advancing the understanding and application of MER field.

In Chapter1, we provide an in-depth review of the background and motivation for MER, highlighting the important role music has played in human history, in therapeutic domains, and in everyday song recommendation systems. This section not only details the challenges and limitations in the MER field, especially the difficulties encountered when dealing with lyrics and audio data, but also clearly states the main goals and contributions of this research.

In Chapter 2, we explore the historical background of the MER field in detail, with a special focus on the development of the Music Emotion quadrant and its impact on the MER field. In addition, it examines in detail various techniques applied in the MER domain, including audio processing, lyrics analysis, and advanced machine learning and artificial intelligence applications. The evolution of these techniques shows how to gain a deeper understanding of emotional expression in music and provides an important rationale for the methodology and experimental design of this study.

In Chapter 3, we began with the collection, cleaning, and balancing of datasets, elaborating on our methodological framework in detail. By replicating the MER study based on Bi-LSTM+Glove, we revealed potential problems in this research and proposed correspond-

ing solution strategies. Addressing these challenges, we set three experimental objectives: embedding, preprocessing, and audio features. In the embedding stage, we found that embeddings focusing on word frequency and context performed better than Glove in analyzing lyrics. During the preprocessing stage, we demonstrated the impact of different preprocessing combinations on various model performances, underscoring the importance of choosing the most suitable preprocessing methods. Regarding audio features, we explored how to effectively integrate audio information to enhance the performance of emotion analysis models. By employing various statistical and visualization methods to analyze the structure of audio data and selecting audio features highly correlated with emotional intensity, we avoided the curse of dimensionality and enhanced the model's understanding of complex data. We found that integrating audio features significantly improved the accuracy and generalizability of emotion analysis.

In Chapter 4, we showcased the performance of various experimental setups and models on the MER task. We found that most models surpassed the benchmarks of previous studies during the embedding and preprocessing experiment phases. Notably, the SVM+Tf-idf model achieved an accuracy and F1 score of 94%, proving the effectiveness of our approach. We also analyzed the relationship between audio features and lyric data and discovered that integrating audio features did not significantly improve performance on Dataset 1. Consequently, we further utilized Dataset 2 for testing and training to verify the generalizability of our models. We observed that the composite models performed better than single-modality models on Dataset 2, both as a test and training set. Particularly, when using Dataset 2 as a training set, we benchmarked against a study that utilized Xl-Net with the same dataset. Our CNN+DenseNet+Word2Vec composite model exceeded the single-modality models and the baseline, achieving an F1 score of 68%.

In Chapter 5, we provide an in-depth analysis and evaluation of the interaction between the method and the results, focusing on the effectiveness of the method and its potential problems. Notably, we found significant correlations between lyrical valence-arousal (V-A) and audio features, specifically valence-energy (V-E). We also use our model for real-world evaluation, including predictive analytics on Spotify's top 100 songs from the last decade. This involves using real-world data to evaluate and validate the validity and reliability of our predictions and models. Such comprehensive evaluation not only affirms the robustness of the method, but also provides a reliable data set for future research in the field of MER.

In Chapter 6, we view the limitations of our study as potential directions for future work, focusing on four key areas: First, audio feature selection, focusing on developing more accurate methods to identify and exploit key audio features; Secondly, dataset enhancement is to expand

the size and diversity of the dataset to improve the generalization ability of the model. The third is the confidence analysis of lyrics and audio modalities, which explores the certainty and reliability of the model in these two modalities. Finally, we study the causal relationship between lyrics and audio data, and deeply analyze how the two interact and jointly affect the emotional expression of music. These research directions not only extend the current research, but also provide clear guidelines for future development in the field of MER.

In summary, this research not only deepened our understanding of MER, but also points to clear directions for future research in the field, demonstrating great potential to address existing challenges by integrating methodologies and innovative techniques.

Project Management

Planning Stage: The project began with detailed planning, where I set clear objectives and a timeline. Regular meetings with my supervisor ensured that each phase progressed smoothly. And get feedback from the inspector on the suggestions in the Proposal.

Execution Stage: I adhered closely to the plan, addressing challenges promptly. Weekly meetings with my supervisor were pivotal for maintaining progress and resolving issues.

Monitoring and Control: Continuous monitoring was crucial, particularly during model training and testing phases. This is evidenced in the concentrated Git commit history during these stages, as shown in Figure 7.1.

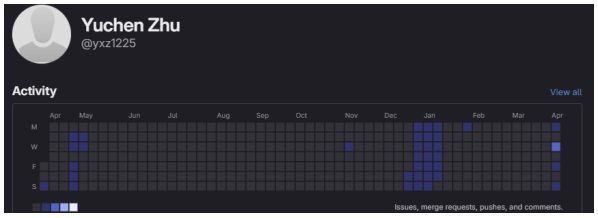

Figure 7.1: Git commit history

Risk Management: I identified potential risks and formulated strategies to mitigate them, including technical challenges and resource constraints. Regular discussions with my supervisor were key to managing these risks effectively.

Demo Feedback: In the final stage of the project, I presented the research progress to the inspectors and supervisors. Their feedback was crucial to guide improvements and refine the research direction, helping me identify areas in the project that needed strengthening.

Conclusion: Through effective management and collaboration with my supervisor and inspector, I successfully completed the research, reached the set goals, and accumulated valuable experience for future projects.

Bibliography

[1] Patrik N Juslin, John A Sloboda, et al. Music and emotion. *D. DEUTSCH (Org.)*, 2001.

[2] Malcolm Budd. *Music and the emotions: The philosophical theories*. Routledge, 2002.

[3] Piotr Przybysz. Music and emotions. *AVANT. Pismo Awangardy Filozoficzno-Naukowej*, (3):174–196, 2013.

[4] Marcel Zentner, Didier Grandjean, and Klaus R. Scherer. Emotions evoked by the sound of music: characterization, classification, and measurement. *Emotion*, 8 4:494–521, 2008.

[5] Shuai-Ting Lin, Pinchen Yang, Chien-Yu Lai, Yu-Yun Su, Yi-Chun Yeh, Mei-Feng Huang, and Cheng-Chung Chen. Mental health implications of music: Insight from neuroscientific and clinical studies. *Harvard review of psychiatry*, 19(1):34–46, 2011.

[6] J Stephen Downie. Music information retrieval. *Annual review of information science and technology*, 37(1):295–340, 2003.

[7] Youngmoo E Kim, Erik M Schmidt, Raymond Migneco, Brandon G Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A Speck, and Douglas Turnbull. Music emotion recognition: A state of the art review. In *Proc. ismir*, volume 86, pages 937–952, 2010.

[8] Tao Li and Mitsunori Ogihara. Detecting emotion in music. In *International Society for Music Information Retrieval Conference*, 2003.

[9] Audrey Laplante. Users' relevance criteria in music retrieval in everyday life: An exploratory study. In *International Society for Music Information Retrieval Conference*, 2010.

[10] Jinhyeok Yang, Woo-Joon Chae, SunYeob Kim, and Hyebong Choi. Emotion-aware music recommendation. In *Interacción*, 2016.

[11] International society for music information retrieval. https://ismir.net/. Accessed: 2024.

[12] Alf Gabrielsson & Lindström and Erik. The role of structure in the musical expression of emotions. In Patrik N. Juslin and John Sloboda, editors, *Handbook of Music and Emotion: Theory, Research, Applications*. Oxford University Press, 2011.

[13] Kate Hevner. Experimental studies of the elements of expression in music. *The American journal of psychology*, 48(2):246–268, 1936.

[14] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[15] Ulrich Schimmack and Alexander Grob. Dimensional models of core affect: a quantitative comparison by means of structural equation modeling. *European Journal of Personality*, 14:325 – 345, 2000.

[16] Juan Sebastián Gómez Cañón, Estefanía Cano, Tuomas Eerola, Perfecto Herrera, Xiao Hu, Yi-Hsuan Yang, and Emilia Gómez. Music emotion recognition: Toward new, robust standards in personalized and context-sensitive applications. *IEEE Signal Processing Magazine*, 38:106–114, 2021.

[17] Yi-Hsuan Yang, Yu-Ching Lin, Ya-Fan Su, and Homer H. Chen. A regression approach to music emotion recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):448–457, 2008.

[18] Yuan-Pin Lin, Yi-Hsuan Yang, and Tzyy-Ping Jung. Fusion of electroencephalographic dynamics and musical contents for estimating emotional responses in music listening. *Frontiers in neuroscience*, 8:83280, 2014.

[19] Yisi Liu, Olga Sourina, and Minh Khoa Nguyen. Real-time eeg-based emotion recognition and its applications. *Transactions on Computational Science XII: Special Issue on Cyberworlds*, pages 256–277, 2011.

[20] Mireille Besson, Frederique Faita, Isabelle Peretz, A-M Bonnel, and Jean Requin. Singing in the brain: Independence of lyrics and tunes. *Psychological Science*, 9(6):494–498, 1998.

[21] Rudolf Mayer and Andreas Rauber. Musical genre classification by ensembles of audio and lyrics features. In *Proceedings of international conference on music information retrieval*, pages 675–680, 2011.

[22] Michael Fell and Caroline Sporleder. Lyrics-based analysis and classification of music. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, pages 620–631, 2014.

[23] Erion Çano and Maurizio Morisio. Moodylyrics: A sentiment annotated lyrics dataset. In *Proceedings of the 2017 international conference on intelligent systems, metaheuristics & swarm intelligence*, pages 118–124, 2017.

[24] Jiddy Abdillah, Ibnu Asror, Yanuar Firdaus Arie Wibowo, et al. Emotion classification of song lyrics using bidirectional lstm method with glove word representation weighting. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(4):723–729, 2020.

[25] Erion Çano, Maurizio Morisio, et al. Music mood dataset creation based on last. fm tags. In *2017 International Conference on Artificial Intelligence and Applications, Vienna, Austria*, pages 15–26, 2017.

[26] Yinan Zhou. *Music Emotion Recognition on Lyrics Using Natural Language Processing*. McGill University (Canada), 2022.

[27] Hande Aka Uymaz and Senem Kumova Metin. Vector based sentiment and emotion analysis from text: A survey. *Engineering Applications of Artificial Intelligence*, 113:104922, 2022.

[28] Samar Al-Saqqa and Arafat A. Awajan. The use of word2vec model in sentiment analysis: A survey. *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control*, 2019.

[29] Yash Sharma, Gaurav Agrawal, Pooja Jain, and Tapan Kumar. Vector representation of words for sentiment analysis using glove. *2017 International Conference on Intelligent Communication and Computational Techniques (ICCT)*, pages 279–284, 2017.

[30] Marvin Ray Dalida, Lyah Bianca Aquino, William Cris Hod, Rachelle Ann Agapor, Shekinah Lor Huyo-a, and Gabriel Avelino Sampedro. Music mood prediction based on spotify's audio features using logistic regression. In *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, pages 1–5. IEEE, 2022.

[31] RUSSELL'S CIRCUMPLEX MODEL and VECTOR DISTANCE CALCULATION TO. International journal of modern pharmaceutical research. *Psychology*, 11:14.

[32] Genius api documentation, Accessed in 2024. Accessed through the `lyricsgenius` Python package for retrieving song lyrics.

[33] John W. Miller. lyricsgenius: A python client for the genius.com api. https://github.com/johnwmillr/LyricsGenius, 2018. Accessed: 2024.

[34] Google custom search json api documentation, Accessed in 2024. Used for performing targeted searches on the Genius website and parsing HTML files for lyrics matching.

[35] Leonard Richardson. Beautiful soup documentation. https://www.crummy.com/software/BeautifulSoup/bs4/doc/, 2023. Accessed 2024.

[36] Spotify web api documentation, Accessed in 2024. Used for acquiring audio features of tracks in the research project.

[37] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[38] Thomas Bayes. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. *Philosophical transactions of the Royal Society of London*, (53):370–418, 1763.

[39] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[40] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[41] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

[42] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[43] Gulnoza Murodilovna Oripova. Rhythm and mything in lyrical genre. , 2020.

[44] Kento Watanabe, Yuichiroh Matsubayashi, Naho Orita, Naoaki Okazaki, Kentaro Inui, Satoru Fukayama, Tomoyasu Nakano, Jordan Smith, and Masataka Goto. Modeling discourse segments in lyrics using repeated patterns. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1959–1969, 2016.

[45] Zellig S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954.

[46] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[47] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[48] BRIAN HOPKINS and J. G. SKELLAM. A New Method for determining the Type of Distribution of Plant Individuals. *Annals of Botany*, 18(2):213–227, 04 1954.

[49] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[50] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[51] Leland Wilkinson and Michael Friendly. The history of the cluster heat map. *The American Statistician*, 63(2):179–184, 2009.

[52] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[53] Xue-wen Chen and Jong Cheol Jeong. Enhanced recursive feature elimination. In *Sixth international conference on machine learning and applications (ICMLA 2007)*, pages 429–435. IEEE, 2007.

[54] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[55] Mukesh Kumar, Saurabh Singhal, Shashi Shekhar, Bhisham Sharma, and Gautam Srivastava. Optimized stacking ensemble learning model for breast cancer detection and classification using machine learning. *Sustainability*, 14(21), 2022.

[56] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[57] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[58] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[59] CJ van Rijsbergen. *Information retrieval*. Butterworth-Heinemann, 1979.

[60] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[61] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[62] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge university press, 2008.

[63] Constituency Parsing. Speech and language processing. *Power Point Slides*, 2009.

[64] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[65] Bing Liu. *Sentiment analysis and opinion mining*. Springer Nature, 2022.

[66] Richard Bellman. Dynamic programming. *science*, 153(3731):34–37, 1966.

[67] Neuroscience News. Music can help reduce anxiety and stress during covid, 2021. Accessed: insert-date-of-access.

[68] Sarah Hennessy, Matthew Sachs, Jonas Kaplan, and Assal Habibi. Music and mood regulation during the early stages of the covid-19 pandemic. *PLOS ONE*, 16(10):e0258027, 2021.

[69] Janis Denk, Alexa Burmester, Michael Kandziora, and Michel Clement. The impact of covid-19 on music consumption and music spending. *PLOS ONE*, 17(5):e0267640, 2022.

[70] United Nations. The human toll and humanitarian crisis of the russia-ukraine war: the first 162 days. *BMJ Global Health*, 2022.

[71] BMJ Global Health. Potential impacts of russo-ukraine conflict and its psychological consequences among ukrainian adults: the post-covid-19 era. *Frontiers*, 2022.

[72] United Nations. Combined effects of war in ukraine, pandemic driving millions more into extreme poverty, senior united nations official tells second committee. *Press Release*, 2022.

[73] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Appendices

Appendix A: GitLab Repository

All the code, implementation and commit history of the project can be found on the GitLab server at the University of Birmingham at the following URL:

https://git.cs.bham.ac.uk/projects-2023-24/yxz1225

Data Folder

Contains the 'Moodylyrcis', 'Moodylyrcis4Q', and 'Top100' datasets, along with scripts for data analysis and visualization.

Model Folder

Includes scripts for replicating paper models, enhancing models with lyric-only feature scripts, single audio feature analysis model scripts, and comprehensive model scripts. Also contains saved models and testing scripts.

Preprocessing Folder

Contains scripts for crawling and processing lyrics and audio features, as well as code for preprocessing experiments.

Research Papers Folder

Contains the papers associated with the datasets and two baseline studies.

Instructions on setting up the environment, configuring the dataset, and running the code are explained in the readme.md file.

Repository File Structure and Explanation

```
yxz1225/
├── Data/
│   ├── data_analysis
│   ├── data_moody
│   │   └── data_pre
│   ├── data_top100
│   │   ├── data_top100_raw
│   │   └── data_predicted
├── Model/
│   ├── model_audio
│   ├── model_combine
│   ├── model_improve
│   ├── model_replication
│   ├── model_save
│   │   ├── combine
│   │   ├── lyrics_only
│   │   ├── tokenizer
│   │   └── top100
│   └── model_test
├── Preprocessing/
│   ├── pre_audio_features
│   ├── pre_combination_test
│   └── pre_lyrics
├── Research Papers/
├── Readme.md/
└── requirements.txt/
```

All the code was independently written by me, specifically for replicating studies and models from academic papers where no code was provided. The parameters and structures used are based on the descriptions in the original papers.

Appendix B: Preprocessing test results

Table 1: SVM+Tf-idf preprocessing Results

Preprocessing Combination	Accuracy	F1 Score
NR, SR, LC, Lemma	0.936	0.936
NR, SR, LC, Stem	0.934	0.934
NR, SR, LC	**0.943**	**0.944**
NR, SR, Lemma	0.934	0.934
NR, SR, Stem	0.934	0.934
NR, SR	0.943	0.944
NR, LC, Lemma	0.921	0.921
NR, LC, Stem	0.929	0.929
NR, LC	0.926	0.927
NR, Lemma	0.924	0.924
NR, Stem	0.929	0.929
NR	0.926	0.927
SR, LC, Lemma	0.934	0.934
SR, LC, Stem	0.929	0.929
SR, LC	0.931	0.931
SR, Lemma	0.929	0.929
SR, Stem	0.929	0.929
SR	0.931	0.931
LC, Lemma	0.916	0.917
LC, Stem	0.921	0.922
LC	0.916	0.917
Lemma	0.919	0.919
Stem	0.921	0.922
None	0.916	0.917

Table 2: NB+Bow preprocessing Results

Preprocessing Combination	Accuracy	F1 Score
NR, SR, LC, Lemma	**0.919**	**0.919**
NR, SR, LC, Stem	0.897	0.896
NR, SR, LC	0.919	0.919
NR, SR, Lemma	0.916	0.916
NR, SR, Stem	0.897	0.896
NR, SR	0.919	0.919
NR, LC, Lemma	0.889	0.890
NR, LC, Stem	0.870	0.870
NR, LC	0.899	0.899
NR, Lemma	0.892	0.892
NR, Stem	0.870	0.870
NR	0.899	0.899
SR, LC, Lemma	0.904	0.904
SR, LC, Stem	0.902	0.901
SR, LC	0.916	0.916
SR, Lemma	0.902	0.902
SR, Stem	0.902	0.901
SR	0.916	0.916
LC, Lemma	0.892	0.892
LC, Stem	0.885	0.885
LC	0.902	0.902
Lemma	0.897	0.897
Stem	0.885	0.885
None	0.902	0.902

Table 3: CNN+Word2Vec preprocessing Results

Preprocessing Combination	Accuracy	F1 Score
NR, SR, LC, Lemma	0.902	0.902
NR, SR, LC, Stem	0.885	0.885
NR, SR, LC	**0.924**	**0.924**
NR, SR, Lemma	0.897	0.897
NR, SR, Stem	0.882	0.882
NR, SR	0.907	0.907
NR, LC, Lemma	0.889	0.889
NR, LC, Stem	0.850	0.850
NR, LC	0.897	0.897
NR, Lemma	0.885	0.885
NR, Stem	0.857	0.857
NR	0.897	0.897
SR, LC, Lemma	0.904	0.904
SR, LC, Stem	0.877	0.878
SR, LC	0.921	0.922
SR, Lemma	0.909	0.909
SR, Stem	0.885	0.885
SR	0.913	0.913
LC, Lemma	0.899	0.899
LC, Stem	0.870	0.870
LC	0.904	0.904
Lemma	0.889	0.890
Stem	0.865	0.865
None	0.880	0.880

Table 4: Bi-LSTM+Word2Vec preprocessing Results

Preprocessing Combination	Accuracy	F1 Score
NR, SR, LC, Lemma	0.8673	0.8667
NR, SR, LC, Stem	0.8698	0.8697
NR, SR, LC	0.8845	0.8839
NR, SR, Lemma	**0.8993**	**0.9001**
NR, SR, Stem	0.8649	0.8660
NR, SR	0.8919	0.8919
NR, LC, Lemma	0.7273	0.7222
NR, LC, Stem	0.7936	0.7913
NR, LC	0.7592	0.7576
NR, Lemma	0.7690	0.7695
NR, Stem	0.8108	0.8107
NR	0.8378	0.8383
SR, LC, Lemma	0.8722	0.8721
SR, LC, Stem	0.8157	0.8132
SR, LC	0.8845	0.8846
SR, Lemma	0.8943	0.8944
SR, Stem	0.8010	0.8040
SR	0.8624	0.8622
LC, Lemma	0.8084	0.8083
LC, Stem	0.7715	0.7714
LC	0.7518	0.7527
Lemma	0.7666	0.7527
Stem	0.7912	0.7894
None	0.7666	0.7626

Appendix C: Pseudo code

Algorithm 1 Scrape Lyrics from Webpage

function SCRAPE_LYRICS(url, $title$, $artist\_name$)
  $webpage \leftarrow$ request webpage from url with user-agent
  $soup \leftarrow$ parse $webpage$ with BeautifulSoup using 'html.parser'
  if not VALIDATE_SONG_INFO($soup$, $title$, $artist\_name$) **then**
    return "Incorrect Song Information"
  end if
  $lyrics\_div \leftarrow soup$.FIND('div', {'data-lyrics-container': 'true'})
  $lyrics \leftarrow$ extract text from $lyrics\_div$ with separator "\n" if found
  return $lyrics$ **or** "Lyrics Not Found"
end function
function VALIDATE_SONG_INFO($soup$, $title$, $artist\_name$)
  Extract $song\_title$ and $artist\_name\_on\_page$ from $soup$ with BeautifulSoup
  Normalize and compare extracted and input values
  return comparison result
end function
function EXTRACT($title$, $artist\_name$, $timeout$)
  $query \leftarrow$ "genius lyrics" $+$ $title$ $+$ $artist\_name$
  Perform search with $query$, extract url of first result
  return SCRAPE_LYRICS(url, $title$, $artist\_name$) **or** "Lyrics Not Found or Timeout"
end function

Algorithm 2 Calculate Hopkins Statistic

function HOPKINS_STATISTIC(X, $subsample = 0.1$, $seed = 42$)
  $n \leftarrow X$.SHAPE(0)
  $d \leftarrow X$.SHAPE(1)
  $m \leftarrow$ INT$(subsample \times n)$
  Set random seed to $seed$
  Initialize NearestNeighbors with Minkowski metric, $p = 2$
  $rand\_X \leftarrow$ generate uniform points within X's range
  $u \leftarrow$ distances to nearest neighbors from $rand\_X$
  $idx \leftarrow$ random subset indices from X
  $w \leftarrow$ distances to second nearest neighbors from $X[idx]$
  $U \leftarrow$ mean of u
  $W \leftarrow$ mean of w
  $H \leftarrow U/(U + W)$
  return H
end function