# Song Mood Detection - Final Report

Yi Chen (yc2455), Yaxuan Huang (yh798), Guangwei Jiang (gj94),
Anqi Dong (ad829), Zhilin Li (zl635)

## Abstract

This project analyzes song lyrics to determine the mood of a song, known as music emotion detection. We believe that music mood detection will be needed for all music platforms. In development, to obtain accurate results, we processed the data by reading, analyzing, and cleaning it. We also improved the model and developed a dataset better suited to its requirements. Here's the link to our project.

## Background

The project is to analyze song lyrics to determine the mood, aiming to develop methods for accurately attributing music emotions. The insights gained from the project can improve existing recommendation systems, enhance user experience, answer questions related to accurate mood classification, predict new song moods, and identify important features for mood classification.

Our team explored related research and datasets, such as FMA, Moodlyrics, and muSe. We also referenced the following articles: Mood Classification using listening data[1], Music Mood and human emotion recognition based on physiological signals: a systematic review[2], and Mood Classification with Lyrics and ConvNets[3]. These articles explain how music mood classification systems work technically, including the taxonomy and identification of music moods, as well as the use of machine learning.

## Dataset

Our dataset is MoodyLyrics, a dataset retrieved from the paper MoodyLyrics: A Sentiment Annotated Lyrics Dataset[4]. MoodyLyrics is a widely used dataset that serves as a benchmark for music emotion recognition models. It was constructed using song lyrics from MillionSongSubset. Based on Russell's model, the songs were classified into one of four quadrants based on their normalized Valence and Arousal values. The resulting dataset contains 2,595 songs from various genres, with 4 emotion categories, including Happy, Angry, Sad, and Relaxed.

However, due to the copyright regulations, all song lyrics are removed. So we used the LyricsGenius, a Python library that allows developers to retrieve lyrics from the Genius.com database to collect the lyrics. It uses the Genius API to search for and download song lyrics by artist and title. After that, our dataset has four columns: Title, Artist, Mood, and Lyrics.

# Analysis

## 1. Data Cleaning

The process of data cleaning includes deleting duplicate data, solving the data imbalance issue, and transformation for further modeling. Initially, we examined the dataset for duplicate entries and successfully eliminated 86 rows with duplicate data. Then we visualized the distribution of four Mood categories in Fig 1 and found a slight data imbalance issue. To mitigate this, we applied an oversampling method to achieve a balanced dataset. In addition, we removed all the null values of the lyrics since there are only 75 of them.

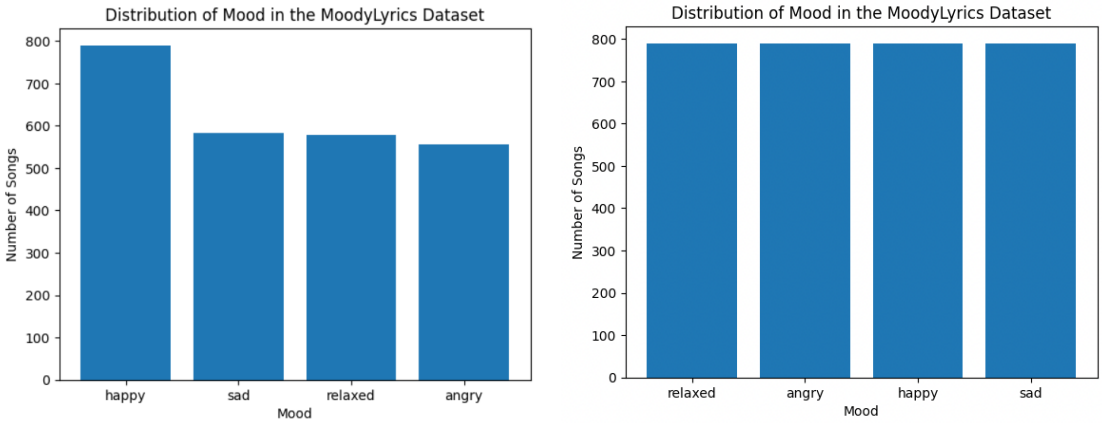


Fig1. Distribution of Mood Before & After oversampling

Through the describe function, we found out that there are only 2405 unique lyrics out of 3085 lyrics available. After we dived deeper into it, we found out that the duplicates majorly occurred because the same song from the same artist has been mentioned in a different playlist, with the same mood. For example, “Silent Night” from the Beatles has been mentioned 5 times. We therefore dropped all the duplicate lyrics accordingly.

Before the mid check point, we cleaned the lyrics including removing punctuations, Arabic numbers, newline characters, song titles, artist names, etc. However, the models we employed didn’t return a satisfying accuracy in general. Therefore, we dived into the dataset a bit deeper and found out that the lyrics text have a general pattern but varied across songs. Before the real lyrics, it has the index of song, contributor, the song’s title, and the artist's name. The main body of the lyrics is also divided by parts like “Intro”, “Chorus”, “verse1” etc. And the structure of each song differed. In the end, it has terms like “6Embed”, ”you might also like” etc. we then removed all patterns through regex and the null values.

```
13 ContributorsThere Goes My Baby Lyrics[Produced by Jim Jonsin and Rico Love]

[Intro]
Yeah... right...
Usher baby... okay, yeah man
Right... (turn the lights off)

[Chorus]
There goes my baby
(Oooh, girl, look at you)
You don't know how good it feels to call you my girl
There goes my baby
Loving everything you do
Oooh, girl, look at you

[Verse 1]
Bet you ain't know that I be checking you out when you be putting your heels on
I swear, your body's so perfect, baby
How you work it, baby yeah, wooo!
I love the way that you be poking it out, girl
Give me something to feel on
So please believe we gone be twerking it out
By the end of the night, baby

[Pre-Chorus]
I've been waiting all day to wrap my hands
Around your waist and kiss your face
Wouldn't trade this feeling for nothing
```

Fig 2. Lyrics of the first song

Furthermore, we removed all the stopwords and calculated the Percentage change in the number of words after stopword pruning. Before we add up the changes to the lyrics' detailed structure, we found that although there is a minimum impact on lyrics, which still require many words for analysis, only a few songs reduced their size by more than 50%. While this may pose some challenges for these particular songs, it is a small percentage of the overall dataset and does not negate the effectiveness of using stopword deletion as a pre-processing technique for our model. After the adjustment we've done through regex alternation, we found out that the word percentage change has been improved, but because of the structure alternation. Therefore, we can predict that the alternation of structure eliminated a lot of bias. It might lead to a better prediction in accuracy.

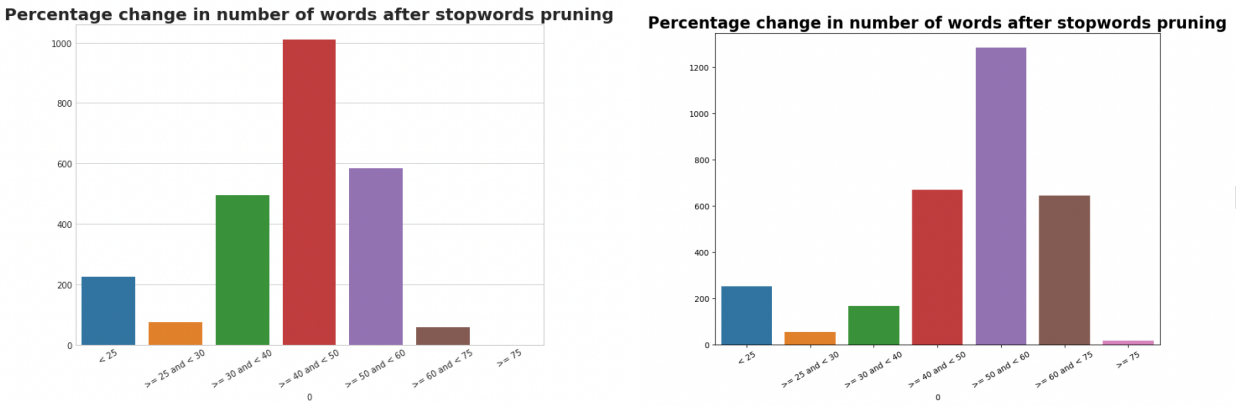


Fig 3. Percentage change in lyrics before & after adjustment

Lastly, we encoded Count_vector and TF-IDF according to the cleaned lyrics for further use in classification, especially for KNN and K-means clustering.

## 2. Feature Engineering

We first extracted some interesting features for further analysis. We processed the data using a bag of words analysis and converted the data using word embedding. To further extract other features, we moved on to sentiment analysis using TextBlob to extract polarity, lexical diversity, and average word length. We then used existing classifiers, here using k-means clustering to explore the emotion labels. We also plotted a bar graph for the top 20 words of all lyrics to visualize the most common words across the entire corpus. Besides, we also visualized sentiment polarity to better understand our data.

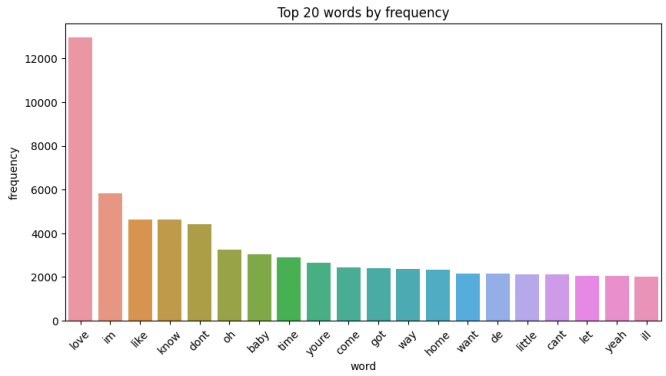


Fig 4. Top 20 words by frequency in lyrics

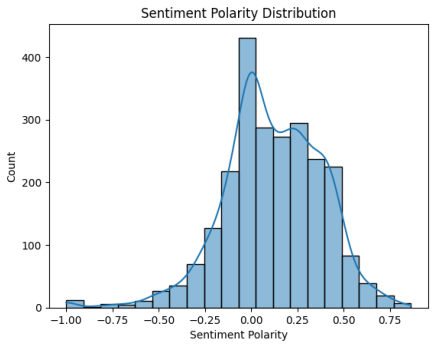


Fig 5. Sentiment Polarity Distribution

We utilized the "en_core_web_lg" pre-trained model, which is a language model trained on a large corpus of text data, to transform the lyrics into vector representations. This process involves mapping each word or phrase in the lyrics to a high-dimensional vector, capturing semantic and syntactic relationships between words. By leveraging this pre-trained model, our objective is to enhance the accuracy and quality of the data representation.

## 3. Model Training

Before the mid-check, we trained two different models: KNeighborsClassifier and Supervised K-Means.KNeighborsClassifier can capture patterns in lyrics based on proximity to neighboring data points. Supervised K-Means incorporates labeled information to assign class labels to clusters, making it useful for clustering songs based on mood.

Initially, we trained Supervised K-Means models using the lyrics data. However, it became evident that the obtained results were unsatisfactory, with an accuracy of only 23%. We then trained the KNeighborsClassifier model and achieved results with an accuracy of 67%.

After the mid-check, we aimed to improve the accuracy further. First we made further data cleaning to optimize data, including detailed analysis about the lyrics structure, the index and structure name, stop words etc. Also, by incorporating models of different kinds, we gain a better understanding of their strengths and weaknesses in the context of song mood detection, aiding us in selecting the most suitable model for our specific

requirements.As a result, we introduced four new models: Gradient Boost, Random Forest, ANN, and SVM. Gradient Boost and Random Forest are ensemble learning methods known for high predictive accuracy and handling complex relationships. SVM effectively handles high-dimensional data and offers robust performance. While ANN excels at capturing non-linear patterns and extracting high-level representations from the input. The ANN structure includes two hidden layers with 60 units each, utilizing the 'sigmoid' activation function. Dropout regularization with a rate of 0.5 is applied after each hidden layer to prevent overfitting. The output layer consists of 4 units, employing the 'softmax' activation function for multi-class classification. The ANN is compiled with the 'adam' optimizer, 'categorical_crossentropy' loss, and 'accuracy' metric. Fortunately, all of which achieved exceptional results. For example, even the least accurate model, Gradient Boost, achieved an accuracy of 81%.

To optimize the models, we conducted parameter tuning and feature selection using GridSearch. For reliable model evaluation, we implemented 10-fold cross-validation. Evaluation metrics such as accuracy, recall, precision, and F1-score were used to comprehensively assess model performance. Detailed information on these processes can be found in the subsequent subsections, providing a thorough analysis of model effectiveness.

# Results

The comparison between models are listed in Table 1 below. Prior to the interim progress report, our team implemented machine learning algorithms such as K-Nearest Neighbors (KNN), Supervised K-means, and Support Vector Machines (SVM) to classify the content of song lyrics. The KNN and SVM methods yielded satisfactory results with accuracy scores of approximately 0.76 and 0.84, respectively. The Supervised K-means method, specifically, calculated the centroids for each emotional class. The classification of a specific lyric involved the comparison of its word vector norm to the centroids of four emotional clusters. The label of the cluster closest to the lyric in question was then assigned to that lyric.

It's noteworthy to mention that we anticipated a less than optimal result with the purpose of using the outcome as a guide for future improvements in our algorithm design. Furthermore, the results hinted towards inadequacies in our initial lyric cleansing process. It was apparent that there were lingering biases that posed challenges in standardizing the lyrics to fit neatly within the four centroids.

Therefore, in response to this finding, we embarked on a second, more in-depth investigation and cleaning of the lyrics data, during which we identified and eliminated specific patterns. Following this process, our attention shifted to ensemble methods which promised superior performance.

We subsequently applied Gradient Boosting and Artificial Neural Networks (ANN) with two hidden layers. Comparatively, these approaches showed superior results to the initial machine learning models.

The results below revealed varying levels of performance for each method. KNN, with a k value of 7, achieved moderate accuracy, precision, recall, and F1-score. Supervised K-Means, despite assigning labels to clusters, exhibited lower performance across all metrics. Next, we turned our attention to ensemble methods, deploying both Gradient Boost and Random Forest models. The Gradient Boost model, with a learning rate of 0.7 and 200 estimators, achieved better performance with an accuracy, precision, recall, and F1-score of approximately 0.81. Similarly, the Random Forest model, characterized by a max_depth of 10, min_samples_leaf of 2, min_samples_split of 5, and 2 estimators, mirrored the performance of the Gradient Boost model across all metrics. The SVM model, employing an 'rbf' kernel with a C value of 100 and a gamma of 0.01, yielded a moderate and balanced performance, with all metrics at 0.84. However, the standout performer was the ANN model, which achieved the highest accuracy, precision, recall, and F1-score. With an accuracy of 0.87 across all metrics, the ANN model exhibited excellent overall performance, showcasing the power and versatility of neural networks in classification tasks.

Table 1.The comparison between models

| method | hyperparameter | accuracy | precision | recall | f1-score |
|---|---|---|---|---|---|
| KNN | k=7 | 0.76 | 0.77 | 0.76 | 0.75 |
| Supervised K-Means | n_cluster=4 | 0.23 | 0.14 | 0.23 | 0.17 |
| Gradient Boost | learning_rate=0.7, n_estimators=200 | 0.81 | 0.82 | 0.81 | 0.81 |
| Random Forest | max_depth=10, min_samples_leaf=2, min_samples_split=5, n_esitimators=2 | 0.81 | 0.81 | 0.81 | 0.81 |
| SVM | {'C': 100, 'gamma': 0.01, 'kernel': 'rbf'} | 0.84 | 0.84 | 0.84 | 0.84 |
| ANN | / | 0.87 | 0.87 | 0.87 | 0.87 |

As ANN performed best, we made further analysis of the confusion matrix to see the model's performance varies for each class. The confusion matrix of ANN is shown in Fig 7, indicating an overall good performance in each class.

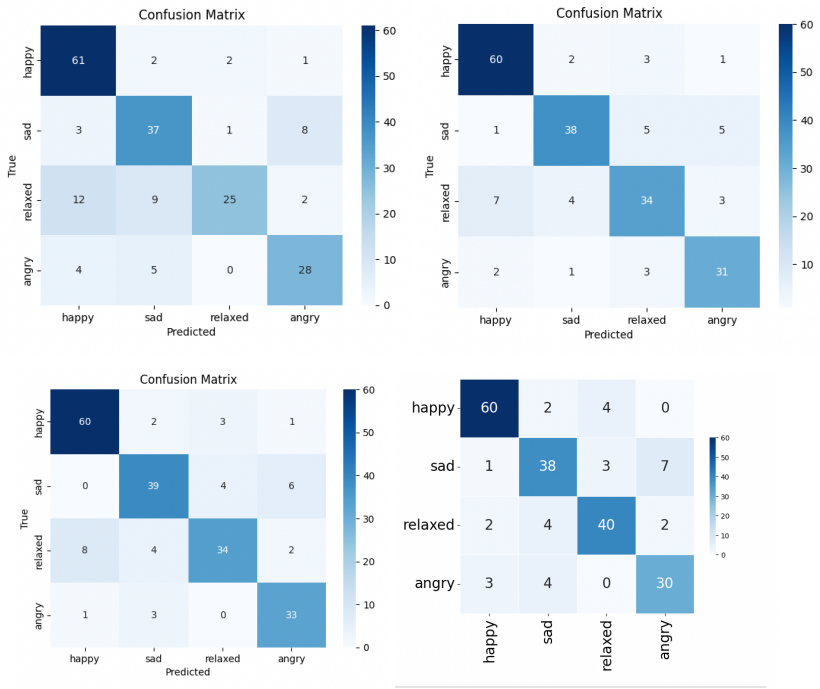


Fig 7. Confusion matrix for KNN & Random Forest & Gradient Boosting & SVM

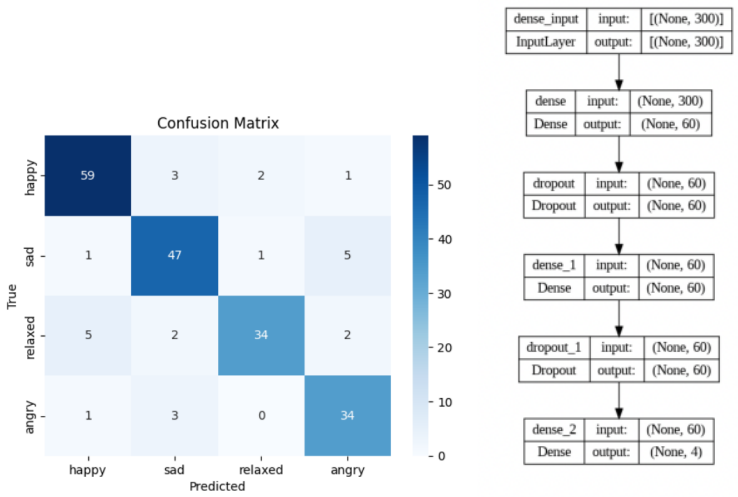


Fig 8. Confusion matrix & Structure for ANN model

# Conclusion

In summary, this project aimed to analyze song lyrics and determine the mood of the songs. The project has made several important discoveries and achievements that we find valuable.

We emphasized the critical role of data cleanup on model performance. During the cleaning process, we dealt with duplicates, imbalances and null values in the lyrics, and

even eliminated punctuation, Arabic numerals, line breaks, song titles and artist names. We also removed stop words and observed small changes in word count that did not significantly affect our dataset. After the cleaned lyrics were added to our data and concatenated with emotion, the training accuracy of the model was greatly improved. This process emphasizes the necessity of cleaned and optimized data for improving the model performance.

Next, feature engineering techniques allowed us to extract interesting features for analysis. Through the application of bag-of-words analysis and word embedding, we obtained information such as polarity, lexical diversity, and average word length. This facilitated sentiment analysis using TextBlob and further exploration of emotion labels through the K-Means clustering algorithm. These additional features enriched our understanding of the music sentiment classification process.

Our project utilized the pre-trained model "en_core_web_lg" to convert lyrics into vectors, thereby enhancing the accuracy of the data. Initially, our KNeighborsClassifier model achieved an accuracy of 67%. After discussion, we introduced four new models: Random Forest, Gradient Boost, ANN, and SVM, all of which demonstrated exceptional performance. Even the least accurate model, Gradient Boost, achieved an accuracy of 81%. Notably, the ANN model outperformed the others, attaining an impressive accuracy of 87.5%.

Our progress showed valuable discoveries and how to create effective music sentiment detection systems. A notable highlight in our project was the discovery of the critical role that data cleaning plays in the creation of these systems. First, As we dived into the raw data, we realized that the efficiency and accuracy of our models were greatly influenced by how well the initial data was cleansed. The process of data cleaning has proven to be pivotal in removing noise and irrelevant data points that could potentially skew the results of our sentiment detection models. It's through this meticulous approach to data preparation that we've been able to fine-tune our algorithms and yield higher levels of accuracy. Secondly, we discovered that the data needs to align with the requirements of the model. During the course of the task, we encountered an issue where the dataset did not match the required format of the model due to inadequate clarification of the requirements. As a result, adjustments had to be made. For instance, we desired the dataset to have the song title, preprocessed lyrics, and corresponding emotion tags placed in a single line for easy retrieval. This design enhances logical coherence and comprehensibility.

In the future, we hope to move away from relying on MoodyLyrics and instead rely on data obtained directly from Spotify. This will allow us to have access to more accurate and diverse information. By leveraging Spotify's extensive user base and sophisticated algorithms, we can collect data on music preferences, trends and patterns. This new method of accessing data will give us a qualitative leap forward, of which we think the most important is the ability to fully align the data with real-world data.

# Contributions by each member of the team

The idea discussion, report writing and final video presentation are distributed equally. Here's the contribution of each team member in the technical part:
Yi Chen (yc2455) : Gather and Clean Data, Train initial models and evaluate performance, Train better models. (Gradient Boost)
Guangwei Jiang :Gather and Clean Data, Train initial models and evaluate performance, Train better models. (ANN)
Anqi Dong : Conduct EDA analysis for feature engineering, Train initial models and evaluate performance, Base on mid check: Further data optimization
Zhilin Li : Conduct EDA analysis for feature engineering, Train initial models and evaluate performance, Train better models. (SVM)
Yaxuan Huang : Gather Data. Conduct EDA analysis for feature engineering, Train initial models and evaluate performance, Train better models. (Random forest)

# Reference

[1]Korzeniowski, Filip, et al. "Mood classification using listening data." arXiv preprint arXiv:2010.11512 (2020).

[2]Chaturvedi, Vybhav, et al. "Music mood and human emotion recognition based on physiological signals: a systematic review." Multimedia Systems 28.1 (2022): 21-44.

[3]Akella, Revanth, and Teng-Sheng Moh. "Mood classification with lyrics and ConvNets." 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). IEEE, 2019.

[4]Çano, Erion, and Maurizio Morisio. "Moodylyrics: A sentiment annotated lyrics dataset." Proceedings of the 2017 international conference on intelligent systems, metaheuristics & swarm intelligence. 2017.