MoodTunes: An NLP-Driven Emotion-Based Music Recommendation System

Dylan Senarath, Anshul Panda, Arman Shah, Harish Dukkipati, Yang Kelty
{senarath, anshulp, armanrsh, hdukkipa, ykelty}@usc.edu

1 Abstract

This project aims to develop an NLP-based music recommendation system that analyzes user input text to identify the most prevalent emotions and generate a personalized playlist based on those emotions. By leveraging sentiment analysis and emotion classification techniques, the system will map user sentiment to specific emotional categories and correlate those with songs from the Emotions4MIDI dataset (1).

2 Introduction

Music has a profound impact on human emotions, serving as a powerful medium for expressing and processing feelings. With the rise of music streaming services, personalized recommendations have become essential for enhancing user experiences. Traditional recommendation systems rely on user preferences, listening history, and collaborative filtering techniques. However, these approaches often overlook the real-time emotional state of the listener, which can play a crucial role in music selection.

3 Related work

3.1 Paper 1: Emotion-Based Music Recommendation for Podcasts

This paper (2) describes a pipeline that detects emotional tone in podcast transcripts and uses it to recommend songs. The authors fine-tune several emotion classifiers—BERT, ALBERT, CNNs, LSTMs, and RNNs—on a labeled Twitter corpus, then apply domain adaptation so the models transfer to podcast text. Predicted emotions are mapped to songs via static emotion–song associations. They address misclassification and limited data through transfer learning and model fine-tuning. Our MoodTunes approach shares their use of transformer fine-tuning for multi-label emotion detection but differs in three ways: we accept arbitrary user-generated text rather than podcast-only scripts; we build playlists on-the-fly from a curated Emotions4MIDI dataset instead of relying on fixed song mappings; and we employ threshold optimization and class-balancing to yield more adaptive, nuanced emotion predictions.

3.2 Paper 2: Evaluating Transformer-Based Models for Emotion Classification

This study (3) evaluates several transformer-based models—specifically BERT, DistilBERT, and RoBERTa—for emotion classification tasks involving text. The models were tested on datasets such as GoEmotions (4) and ISEAR, which include labels for emotions like joy, sadness, anger, and fear. Emotion classification was performed by fine-tuning the transformer models on these labeled datasets, with the final hidden layer passed to a softmax classifier to predict the most probable emotion. The study explored the effects of various fine-tuning techniques, including freezing transformer layers, adjusting learning rates, and modifying hyperparameters. An important finding was that certain text preprocessing steps, such as removing stop words and punctuation, actually degraded performance, since transformers rely on full sentence structure to capture semantic meaning effectively. The study highlights key training considerations, including the negative impact of excessive text preprocessing and the importance of preserving sentence structure. These findings directly influenced our preprocessing strategy, where we avoided over-sanitization and emphasized structural integrity. While their work provides valuable performance benchmarks and insights into fine-tuning practices, our contribution expands on this by applying the emotion classification task to a real-time, user-facing music recommendation system. Additionally, we incorporate emotion-specific threshold tuning to improve multi-label prediction quality—an aspect not addressed in the compara-

tive study.

4 Problem Description

Natural Language Processing (NLP) has made significant strides in understanding human emotions from text. Sentiment analysis techniques, such as those based on the GoEmotions dataset, enable the classification of text into multiple emotion categories. The GoEmotions dataset provides fine-grained emotion labels across 28 different categories, allowing for a nuanced interpretation of user sentiment. By extracting the dominant emotions from a given text, we can match these emotions to songs in the Emotions4MIDI dataset, where each song is tagged with emotion scores based on its lyrics. This allows us to retrieve songs that align with the user's emotional state.

5 Methods

5.1 Datasets and Materials

The MoodTunes system integrates emotion-aware natural language processing with music recommendation by leveraging two primary datasets. The first is the **GoEmotions Dataset**, a corpus of over 58,000 English Reddit comments annotated with 28 emotion categories. This dataset was used to train our multi-label emotion classification model. The second is the **Emotions4MIDI Dataset**, a collection of over 12,000 songs, each labeled with a 28-dimensional emotion score (ranging from 0 to 1) reflecting alignment with GoEmotions categories. Metadata—including song title, artist, and album—was enriched using an `md5.json` lookup for MIDI hashes. This process combined automated matching techniques using Python tools such as Pandas and JSON parsing, along with manual annotation for entries sourced from Reddit.

5.2 Data Preprocessing

5.2.1 Text Preprocessing for Emotion Classification

Before training our DistilBERT-based emotion classifier, we applied several preprocessing steps to the GoEmotions dataset. Text was first cleaned by converting it to lowercase, removing URLs and user tags, and normalizing punctuation and whitespace. To increase model robustness, we applied data augmentation to 20% of the training data using techniques such as synonym replacement via WordNet, random word swaps, and deletions. To address the class imbalance in the dataset, particularly the over-representation of the "neutral" label, we performed targeted oversampling so that each minority class reached approximately 50% of the majority class size. Finally, the text was tokenized using the `distilbert-base-uncased` tokenizer, with a maximum sequence length of 128 tokens, and appropriate padding or truncation was applied.

5.2.2 Emotions4MIDI Dataset Processing

To enable efficient playlist generation, the Emotions4MIDI dataset was processed and optimized for fast lookups. Each song entry was enriched with human-readable metadata including title, artist, and album name. Songs were then grouped according to the emotion with the highest associated score and sorted in descending order of that score. For each of the 28 emotion labels, we retained only the top 10 tracks to ensure fast access and high relevance. These curated lists were serialized into a JSON lookup file, enabling real-time retrieval during playlist construction based on predicted emotion labels from user input.

5.3 Model Design and Training

Our core emotion detection model is a fine-tuned version of DistilBERT configured for multi-label classification. After tokenizing the input sequences, the token embeddings generated by DistilBERT were passed through an attention pooling layer, which computed weighted averages of token representations to form a context-aware embedding. This embedding was then processed by a final classification layer, which outputted 28 probabilities, each corresponding to an emotion label. To enhance classification performance, we implemented threshold optimization by calculating the optimal decision threshold for each emotion individually. These thresholds were determined by maximizing the F1 score on the training set, replacing the default fixed threshold of 0.5. This approach allowed the model to better handle class imbalance and improve prediction quality for underrepresented emotion classes.

5.4 Inference and Recommendation Procedure

At runtime, the system operates as follows:

1. User input text is cleaned, tokenized, and passed through the trained classifier.

2. The model outputs a set of emotions exceeding their respective thresholds.

3. For each detected emotion, the system retrieves songs from the precomputed JSON lists.

4. A 10-song playlist is generated by evenly distributing selections across the detected emotions. For example, if two emotions are detected, five songs per emotion are selected based on the highest scores.

Unlike traditional recommendation systems relying on collaborative filtering or user history, MoodTunes dynamically recommends music aligned with the user's real-time emotional state, offering a personalized and emotionally resonant listening experience.

6 Experimental Results

Hyper Parameters	Epochs	Batch Size	Max Length
#	3	32	128

Scores	Precision	Recall	F1-score
Average	0.54	0.62	0.58

Table 1: Hyperparameters used for training and resulting average performance scores.

6.1 Experimental Setup

For training our emotion classification model, we utilized the **GoEmotions** dataset, which contains over 58,000 English Reddit comments annotated across 28 emotion categories. We adopted a pretrained `distilbert-base-uncased` model fine-tuned for multi-label emotion detection.

The model was trained using the following hyperparameters: 3 epochs, a batch size of 32, and a maximum input sequence length of 128 tokens. These settings were chosen to balance performance with computational efficiency, given the dataset size and available resources.

To preprocess the data, we applied text cleaning, data augmentation, class balancing, and tokenization as described in the Methods section. The training was conducted on a GPU-enabled environment to accelerate computation.

6.2 Baseline Methods

As a baseline, we compared our results against a simple majority-class predictor, which always predicts the most frequent emotion labels (e.g., "neutral", "approval"). This baseline serves to highlight the challenge of achieving balanced performance across all 28 emotions, especially for less frequent categories.

6.3 Evaluation Protocols

We evaluated model performance using standard multi-label classification metrics: **Precision**, **Recall**, and **F1-score**. Evaluation was conducted on a held-out test set.

Our evaluation protocol enforced a strict criterion where predicted emotion sets had to exactly match the true label sets. Any deviation, including partial matches (e.g., predicting both "joy" and "amusement" when only "joy" is correct), resulted in both a false positive and a false negative. While this ensures rigorous assessment, it inherently penalizes near-correct predictions—common in multi-label emotion tasks where emotional overlap is frequent.

6.4 Results and Discussion

Under these conditions, our model achieved an average Precision of 0.54, Recall of 0.62, and an F1-score of 0.58 (Table 1). While modest, these results show that the model effectively captures emotional signals in user input, even with a highly imbalanced label distribution.

The relatively higher recall suggests the model is proficient at identifying relevant emotions, though precision is impacted by the strict exact-match evaluation metric, which penalizes partially correct predictions—particularly in multi-label emotional contexts.

Training was early stopped after 3 epochs based on stagnating validation loss, indicating no significant improvement with further training.

In practice, the model performs well on real-world examples, as illustrated in our demo. For instance, user input like "my girlfriend broke up with me" was accurately classified with "sadness" as the top emotion, reflecting the model's ability to identify clear emotional tone in everyday language.

Overall, the model provides a strong foundation for real-time, emotion-aware music recommendation within the MoodTunes system.

6.5 Demo

To demonstrate the functionality of MoodTunes, we present an example where user input is transformed into a personalized playlist based on detected emotions.

In this scenario, the user expresses emotional distress with the input: *"my girlfriend broke up with me"* (Figure 1). The system analyzes this text using a fine-tuned emotion classification model, which outputs probability scores for each of the 28 emotion categories. Emotions exceeding optimized thresholds are selected, with **sadness** detected as the dominant emotion.

Following classification, the system queries a curated database of songs labeled with emotion scores. It retrieves and ranks songs most aligned with the detected emotions. In this case, *"True Love"* by Joan Armatrading is recommended as the top result, having the highest sadness score of 0.9298 in the dataset (Figure 2).

This example highlights how MoodTunes integrates natural language processing and emotion-driven recommendation to deliver music that resonates with a user's emotional state.

```
Text to classify: my girlfriend broke up with me

Emotion Classification Results:
==============================
Emotion          | Probability  | Threshold  | Prediction
-------------------------------------------------------
sadness          |    0.9920    |   0.7969   |     ✓
disappointment   |    0.7331    |   0.7520   |     X
neutral          |    0.3748    |   0.4656   |     X
love             |    0.1894    |   0.7578   |     X
grief            |    0.1275    |   0.8369   |     X
approval         |    0.1267    |   0.7324   |     X
remorse          |    0.1253    |   0.6489   |     X
disgust          |    0.0821    |   0.9561   |     X
annoyance        |    0.0772    |   0.7988   |     X
embarrassment    |    0.0716    |   0.8984   |     X
optimism         |    0.0635    |   0.8438   |     X
nervousness      |    0.0589    |   0.9336   |     X
joy              |    0.0586    |   0.7993   |     X
```

Figure 1: User input and emotion classification results, detecting sadness as the primary emotion.

```
Detected emotions: sadness
{'title': 'True Love', 'artist': 'Joan Armatrading', 'score': 0.9298}
{'title': "I'm gonna miss you forever", 'artist': 'Aaron Carter', 'score': 0.9189}
{'title': 'You lost me', 'artist': 'Christina Aguilera', 'score': 0.9132}
{'title': 'Chere Amie (Toutes Mes Excuses)', 'artist': 'Marc Lavoine', 'score': 0.9013}
{'title': 'Sorry seems to be the hardest word', 'artist': 'Charles John', 'score': 0.9013}
{'title': 'Reflections of my life', 'artist': 'The Marmalade', 'score': 0.8941}
{'title': 'Help Me Make It Through The Night', 'artist': 'Sammi Smith', 'score': 0.8941}
{'title': 'Little child', 'artist': 'The Beatles', 'score': 0.8876}
{'title': "I can't be with you", 'artist': 'The Cranberries', 'score': 0.8765}
{'title': 'Jamaica farewell', 'artist': 'Belafonte', 'score': 0.857}
```

Figure 2: Top recommended song based on detected emotion: *"True Love"* with a sadness score of 0.9298.

7 Conclusion and Future Works

MoodTunes presents a novel approach to music recommendation by leveraging natural language processing to analyze user input and generate playlists based on emotional intent. By fine-tuning a DistilBERT model on the GoEmotions dataset, we developed a multi-label emotion classifier capable of identifying nuanced emotional signals from text. These predicted emotions are then matched to a curated set of songs labeled with emotion scores, enabling the system to deliver real-time, personalized playlists aligned with the user's mood. Our implementation includes enhancements such as attention pooling, threshold optimization, and targeted oversampling to address class imbalance and improve model robustness. Experimental results demonstrated solid performance across metrics like precision, recall, and F1 score, and real-world examples showcased the system's ability to produce emotionally resonant music recommendations.

One of the primary areas for enhancement lies in improving the emotion classification performance. Despite achieving reasonable F1 scores, there is room to increase accuracy, particularly for underrepresented emotions. Future work could explore more advanced transformer architectures, such as RoBERTa or DeBERTa, as well as techniques like label-wise threshold optimization, focal loss tuning, and data augmentation strategies specifically tailored for emotional nuance in short texts. Additionally, leveraging multimodal sentiment cues—such as voice or facial expression inputs—could further improve the system's ability to detect complex and blended emotional states.

In terms of playlist generation, the current approach retrieves songs based on individual emotion scores, which can lead to fragmented playlists when multiple emotions are detected. To address this, future iterations could implement a cumulative emotion scoring system that ranks songs not just by their association with single emotions, but by how well they align with the overall emotional profile of the input. This would ensure smoother, more cohesive playlists that reflect blended or transitional emotional states, improving user experience and emotional continuity throughout the listening session.

8 Division of Labor

The project was a collaborative effort, with each team member contributing to distinct components of the system. Below is a summary of individual responsibilities:

Dylan

- Resolved core issues in the emotion classification pipeline by implementing proper sigmoid thresholding, handling class imbalance using `pos_weight`, and improving training stability—enabling meaningful evaluation metrics and influencing key aspects of the final model.
- Led development of the song recommendation system by sourcing the Emotions4MIDI dataset, matching hashed filenames to metadata, handling inconsistent file formats, and generating the emotion-to-song JSON used for playlist generation.
- Conducted extensive experimentation with alternative transformer architectures, focal loss, data augmentation, and class balancing strategies—several of which informed enhancements adopted in the final implementation.

Anshul

- Designed and implemented the DistilBERT-based training pipeline, incorporating attention pooling and a custom emotion classification layer.
- Developed an efficient `Dataset` class and `DataLoader` setup to streamline data handling during training and validation.
- Created evaluation metric functions (precision, recall, F1-score) to assess model performance and guide iterative improvements.

Arman

- Finalized the emotion classification model using attention pooling, focal loss with dynamic weighting, and targeted oversampling, achieving a micro F1 score of 0.5854.
- Implemented data cleaning and text augmentation techniques (e.g., synonym replacement, word swaps, deletions) to enhance dataset diversity and address class imbalance.
- Designed and built the user interface for real-time text input, integrating it with the emotion detection pipeline for seamless end-to-end functionality.

Yang

- Expanded the dataset with 8,682 augmented samples and applied class-specific oversampling strategies to address imbalance while preventing overfitting.
- Performed hyperparameter tuning and explored approaches such as emotion grouping—ultimately refining methods to improve model precision and recall.
- Integrated the song recommendation system into the user interface, enabling users to receive emotion-based playlists from their text inputs.

Harish

- Matched 8,000 MIDI hashes from the Emotions4MIDI dataset to LMD-matched JSON entries to extract song metadata (title, artist, album) and manually annotated metadata for 4,000 Reddit-sourced tracks.
- Experimented with NER tagging models to automate metadata extraction and normalization from non-standard filenames.
- Designed and implemented the playlist generation system, leveraging Dylan's emotion-to-song JSON to select 10-track playlists by evenly distributing top-ranked songs across detected emotions.

References

[1] Serkan Sulun, Pedro Oliveira, and Paula Viana. Emotion4MIDI: A Lyrics-Based Emotion-Labeled Symbolic Music Dataset. *arXiv preprint*, July 27, 2023. https://arxiv.org/abs/2307.14783.

[2] A. Lee, S. Ravi, and A. Tsun. Music Recommendation for Podcast Strips: Detecting Emotion from Text. *Stanford Research*, 2021. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1204/reports/custom/report07.pdf.

[3] Mahdi Rezapour. Emotion Detection with Transformers: A Comparative Study. *arXiv preprint*, July 27, 2024. https://arxiv.org/pdf/2403.15454v4.

[4] Dana Alon and Jeongwoo Ko. GoEmotions: A Dataset for Fine-Grained Emotion Classification. *Google Research Blog*, October 28, 2021. https://ai.googleblog.com/2021/10/goemotions-dataset-for-fine-grained.html.