Deep Learning Approach on Music Recommendation System

Christine Sangphet, Ann Liang, Namika Takada

May 2, 2025

Abstract

In this project, we develop a deep learning-based music recommendation system that leverages song lyrics to enhance personalization. While traditional systems rely on audio features or collaborative filtering, our approach focuses on the thematic content of lyrics to classify songs into topics such as romantic, sadness, or violence. Using a Kaggle dataset that contains song lyrics from 1950 to 2019, we preprocess the text, extract features, and train several neural network architectures, including feedforward networks and a Transformer-based model. The best-performing model–a fine-tuned feedforward neural network–achieved strong classification accuracy and forms the foundation of a recommendation system that suggests songs based on the user's current mood or song topic preferences. This system delivers a listening experience that takes into account both emotional and semantic context, with a user-friendly interface built using Gradio.

1 Introduction

With the increasing volume of digital music, recommendation systems have become crucial for helping users explore a wide variety of music available on platforms like Spotify, Apple Music, and YouTube Music. While many of these platforms use machine learning to personalize recommendations, some often process artist information (e.g., metadata, cultural context) and audio features (e.g., tempo) as separate data streams [4]. This project takes a different approach by emphasizing the lyrics themselves as a key feature for song classification and recommendation. Specifically, we:

- Preprocess lyrics to prepare them for input into machine learning models by normalizing text, tokenizing, and removing noise, ensuring that the focus remains on the meaningful content of the lyrics.
- Use deep learning models to classify songs into topic categories based solely on their lyrics, providing a new way to recommend songs based on their topic rather than just genre or artist.

The ultimate goal is to improve the recommendation system's ability to suggest songs that resonate with users based on the emotional and topical content of the lyrics, offering a more personalized listening experience. Through this project, we aim to bridge the gap between user preferences and the deeper meanings embedded in song lyrics, facilitating music discovery in a new and innovative way. The complete source code and implementation for this project are available on GitHub: https://github.com/christinesangphet/deep-learning-project [2]

2 Related Work

A closely related paper to our project is "Music Recommendation System using Machine Learning Algorithms" by Deshpande and Sharma. This study explores various machine learning algorithms for music recommendation, including content-based filtering, collaborative filtering and hybrid approaches [5]. They used a dataset from the Million Song Dataset and Last.fm, which has a similar scope to the Spotify dataset we are planning to use. This study did not release any code or models publicly which presents an opportunity for our project. By implementing our own methods, we can compare our results and validate our findings. We can also compare and provide insights into how different data sources affect recommendation quality.

In addition, a study mirrors the dual-input architecture with 9-song preference vectors feeding deep networks. By combining collaborative filtering and contenting-based filtering, they achieved 88% precision on the Spotify dataset - a direct performance benchmark to our future code [7]. The one-hot encoding for categorical features is similar to our artist label encoding and embeddings. Furthermore, this study implemented a triplet loss logic through positive/negative training examples derived from the playlist data. This is something we can implement within our code for better performance.

Another study titled "Creating a Reliable Music Discovery and Recommendation System" explores the development of a reliable music discovery system using the SYNAT database, which focuses on music and genre classification and parameterization for audio analysis. To overcome the challenges in music recommendation, the researchers look into the methods of identifying similar songs based on low-level features [1]. Therefore, this paper provides relevant insights into feature extraction and similarity-based recommendations for the project. Those techniques can refine the deep learning model's capability to map songs into meaningful latent space with higher accuracy on personalized music recommendations.

3 Datasets

The primary dataset for this project is the Music Dataset: 1950 to 2019, available on Kaggle [6]. This curated dataset contains a comprehensive collection of song lyrics and music metadata spanning nearly seven decades, with key infor-

mation such as artist name, track name, and genre. Additionally, it includes a range of audio-related features–such as danceability, energy, valence, and acousticness–which offer insights into the sonic and emotional characteristics of each track. These features make the dataset particularly suitable for developing a music recommendation system that leverages natural language processing (NLP) techniques on lyrics, potentially combined with audio features, to suggest songs aligned with a user's mood or musical preferences.

The dataset is based on the publicly cited work by Moura et al. (2020), published through Mendeley Data and assigned a DOI (10.17632/3t9vbwxgr5.3), which validates its credibility and relevance for academic and applied research [3].

4 Approach (or Methodology)

4.1 Preprocessing

To prepare the lyrics data for training, a comprehensive preprocessing pipeline was applied. First, only the relevant columns–artist_name, track_name, lyrics, and topic–were retained for analysis. The core text preprocessing steps included:

1. Text Normalization: All lyrics were converted to lowercase and stripped of punctuation to ensure consistency in tokenization.

2. Number Handling: Standalone numeric tokens (e.g., "2", "100") were replaced with their word equivalents (e.g., "two", "one hundred") using the inflect library to retain semantic value while standardizing format.

3. Tokenization and Cleaning: Lyrics were tokenized into individual words. Common English stopwords (e.g., "the", "and") were removed to reduce noise, and only alphabetic tokens were retained.

4. Lemmatization: Words were lemmatized using WordNet to reduce them to their base forms (e.g., "running" $\rightarrow$ "run"), helping to normalize the vocabulary without excessive dimensionality.

After preprocessing, the tokens were rejoined into space-separated strings for vectorization.

The labels (topics) were then encoded numerically using LabelEncoder to convert categorical classes into integers for model compatibility.

For feature extraction, a TF-IDF vectorizer was used to convert the cleaned lyrics into numerical vectors. Both unigrams and bigrams were considered (ngram_range=(1, 2)), and the feature space was limited to the top 5,000 most informative n-grams to reduce dimensionality while preserving relevant patterns in the text.

Following vectorization, the dataset was split into training and testing sets using an 80/20 split with stratification to preserve class distribution. Sparse matrices were converted to dense PyTorch tensors for model training.

Finally, DataLoaders were constructed to efficiently batch the data during training and evaluation phases.

4.2 EDA

To conduct some initial exploratory data analysis, we first looked at the basic numerical features of the dataset which were danceability, energy, loudness, acousticness, valence, and age. Initial inspection revealed no missing values in numerical features.

The distribution of acousticness is the most skewed with danceability having the most normal distribution.

Key correlations emerged:

- Energy and loudness (0.77)
- Acousticness and energy (-0.72)
- Valence and danceability (0.49)

Energy and loudness showing a high positive correlation indicates that high intensity tracks dominate modern playlists. Energy and acousticness showing a strong inverse relationship proves that songs that have high acousticness are low in energy. Finally, valence strongly correlates with danceability suggests that upbeat songs are perceived as happier.

Using K-means clustering (optimal 5 clusters determined by elbow method and silhouette scores), distinct listener preference groups emerged. PCA was strategically employed in this analysis to address two critical challenges in music recommendation systems. The dataset using the six audio features that exhibit complex correlations reduces the dimensionality while preserving 60% of total variance. It also reveals latent relationships through orthogonal components.

1. PC1 = energy (0.54) + loudness (0.51) - acousticness (0.50)
2. PC2 = danceability (0.67) + valence (0.69)

This confirmed the primary axes of variation in the data are intensity (high-energy vs. acoustic) and mood (upbeat vs. neutral). PC1 (38%): Dominated by energy (0.54), loudness (0.51), and negative acousticness (-0.50). This separates high-energy tracks (right on PC1) from acoustic tracks (left). PC2 (22% variance): Driven by danceability (0.67) and valence (0.69), contrasting upbeat songs (top) from older tracks (age has a mild positive loading, so older songs appear lower).

The EDA quantitatively links audio features to mood-like clusters, providing a bridge between lyrics (explicit mood signal via NLP) and audio traits (implicit mood signals via clusters). This dual approach addresses the "cold start" problem by using clusters when lyrics are unavailable.

4.3 Model Development

To classify lyrics into topic categories based on their dense vector representations, we developed and evaluated three neural network architectures in PyTorch: a baseline feedforward neural network (FFNN), a fine-tuned FFNN with regularization and staged training, and a Transformer-based model adapted for tabular data.

1. Feedforward Neural Network (FFNN): The initial model was a basic feedforward neural network comprising three fully connected (Linear) layers with ReLU activations in between:

   - Input $\rightarrow$ 256 $\rightarrow$ ReLU $\rightarrow$ 128 $\rightarrow$ ReLU $\rightarrow$ Output.
   - The output layer directly maps to the number of classes.

   The model was trained using the Adam optimizer and the cross-entropy loss function, which combines a softmax activation over the output logits with negative log-likelihood loss. This allows the model to assign a probability distribution over the possible classes, with softmax ensuring that the outputs sum to 1. Training was conducted over 20 epochs, tracking loss and accuracy on both training and test datasets.

2. Fine-Tuned FFNN: Building on the baseline, the second model incorporated regularization (Dropout) and transfer learning techniques:

   - Dropout layers (with 0.3 probability) were added after each ReLU activation to reduce overfitting.
   - Training was conducted in two phases:
     - Phase 1: Only the final classification layer (fc3) was trained while earlier layers were frozen, allowing for adaptation without destabilizing earlier representations.
     - Phase 2: All layers were unfrozen and fine-tuned at a reduced learning rate, with a learning rate scheduler (ReduceLROnPlateau) to adaptively lower the rate when validation performance plateaued.

   This approach improved generalization, while the remaining training settings are the same as the initial feedforward neural network model.

3. Transformer-Based Model: For the third architecture, we implemented a Transformer encoder adapted for tabular (non-sequential) data:

   - Each input vector was first passed through a Linear projection to a d_model dimensional embedding space.
   - A learnable positional encoding was added to each vector (though no time sequence exists, this encourages the model to learn feature interactions).
   - The embedded inputs were passed through two stacked TransformerEncoderLayers. A final Linear layer was used for classification.

Even though the data didn't have a time-based structure, the Transformer model was tested to see if its self-attention mechanism could capture relationships between features better than a regular neural network. Like previous models, this architecture used cross-entropy loss and the Adam optimizer.

4.4 Model Evaluation

To evaluate the performance of the models, we followed a structured approach:

1. Overall performance metrics
   We computed the following aggregate metrics:
   - Accuracy: the proportion of correct predictions
   - Weighted precision and recall: these account for the class imbalance by averaging metric values proportionally to class support
2. Detailed classification report / Class-wise Analysis: A full classification report was generated, which includes:
   - Accuracy
   - Precision
   - Reall
   - F1-score
   - Support

   This report allows for performance inspection across all classes individually.
3. Confusion Matrix: To better understand the misclassifications, we visualized the confusion matrix as a heatmap. This clearly shows how often each class was confused with others, offering insights into common misclassification patterns.
4. Key Findings Summary
   We identified:
   - The best performing class (highest class-wise accuracy)
   - The most challenging class (lowest class-wise accuracy)
   - The most frequent confusion pair

5 Evaluation Results

5.1 Baseline Feedforward Neural network

The baseline feedforward neural network achieved an overall accuracy of 90.10%, with a weighted precision of 90.17% and a weighted recall of 90.10%. Class-wise, it performed best in the world/life category (accuracy: 93.27%), while

feelings was the most challenging class (accuracy: 77.87%). The most frequent misclassification occurred between violence and world/life, suggesting semantic overlap or model confusion between these categories.

5.2 Fine-Tuned Feedforward Neural Network

After fine-tuning, the feedforward neural network demonstrated a noticeable improvement across all metrics, reaching an overall accuracy of 92.74%, with a weighted precision and recall of 92.75% and 92.74%, respectively. The most accurate class prediction was for obscene (accuracy: 94.37%), while feelings remained the most difficult to classify correctly (accuracy: 82.79%). The most frequent confusion in this model occurred between sadness and violence, potentially due to emotional or contextual similarities in the input data.

5.3 Transformer-Based Model

The Transformer-based model achieved an overall accuracy of 91.08%, with a weighted precision of 91.16% and weighted recall of 91.08%. The model performed best on the romantic class, achieving an accuracy of 94.75%, suggesting it was particularly effective at identifying content related to this topic. In contrast, the night/time class posed the greatest challenge, with a lower accuracy of 82.74%, showing a potential area for further model refinement. The most common misclassification was between sadness and violence, echoing patterns observed in the fine-tuned feedforward neural network.

6 Song Recommendation System

To demonstrate the practical utility of our model, we developed a song recommendation system. Since the fine-tuned feedforward neural network achieved the highest overall accuracy compared to the other models, it was chosen as the foundation for this task.

System Design:

- User Input: The user selects a mood or topic (e.g., "sadness").
- Topic Conversion: The selected topic is encoded into a numeric label using label encoding.
- Prediction: The model processes the lyrics of all songs and outputs a probability distribution for each topic.
- Confidence Threshold: Songs with a probability greater than 0.5 for the selected topic are recommended.
- Fallback: In cases where there are insufficient high-confidence song recommendations (especially for less common topics), the system defaults to recommending songs from the selected topic, even without applying the confidence threshold.

The model's output is interpreted using the softmax function, which converts the logits into normalized probability scores, representing the model's confidence in each topic assignment. The recommendations are presented through a Gradio-based interface (public URL: https://5f450278c027cde988.gradio.live/, expires one week after launch), which allows users to select a topic and specify the number of songs (up to 10) they wish to receive. If the link is no longer active, the interface can be regenerated by running the notebook song_recommendation_system_1.ipynb located in the "song recommendation system" folder on GitHub.

7 Conclusion

This music recommendation aims to leverage deep learning techniques to create a personalized and engaging music discovery experience. By utilizing the Music Dataset: 1950-2019 and implementing neural network models, we aim to develop a system that accurately recommends songs based on the user's mood and preference using lyrics. Through this project, we aim to contribute to the field of music recommendation systems by exploring the potential of deep learning in understanding. Ultimately, we aspire to create a tool that enhances the music listening experience, helps users discover new artists and songs, and showcases the power of machine learning in personalized content recommendations.

References

[1] B. Kostek, P. Hoffmann, A. Kaczmarek, and P. Spaleniak. Creating a reliable music discovery and recommendation system. In *Studies in Computational Intelligence*, pages 107–130. Springer, 2014. Accessed: 2025-04-05.

[2] A. Liang, C. Sangphet, and N. Takada. Song recommendation system based on mood and preferences. `https://github.com/christinesangphet/deep-learning-project`, 2025. Accessed: 2025-04-30.

[3] L. Moura, E. Fontelles, V. Sampaio, and M. França. Music dataset: Lyrics and metadata from 1950 to 2019. `https://data.mendeley.com/datasets/3t9vbwxgr5/3`, 2020. Accessed: 2025-04-30.

[4] D. Pastukhov. Inside spotify's recommender system: A complete guide to spotify recommendation algorithms. `https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022`, 2022. Accessed: 2025-04-06.

[5] A. Rahul, R.S. Sabeenian, D. Gurang, R. Kirthika, and S. Rubeena. Ai based music recommendation system using deep learning algorithms. In *IOP conference series: earth and environmental science*, volume 785, page 012013. IOP Publishing, 2021. Accessed: 2025-04-05.

[6] S. Shahane. Music dataset: 1950 to 2019. https://www.kaggle.com/datasets/saurabhshahane/music-dataset-1950-to-2019, 2020. Accessed: 2025-04-30.

[7] P. Sharma and R. Singh. Music recommendation system using deep learning. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 9(5):1157–1161, 2021. Accessed: 2025-04-05.