Music Genre Classification using Lyrics and Audio Features

Group 19

Chen Wei, Liao Hung Chieh, Low Xu Yan Jessica, Wang Chuxin, Zhu Xueying

National University of Singapore

April 27, 2025

1 Introduction

1.1 Project Overview

In this project, we explored genre classification by leveraging both lyrics and audio features obtained from Spotify. Our objective was to assess how effectively different types of data contribute to genre prediction, especially lyrics, as they contain rich semantic information that can be utilized for classification.

Multiple models were implemented and evaluated to compare their accuracy and robustness. This report outlines the methodology, implementation details, and comparative analysis of our approaches. Through this, we gained insights into the strengths and limitations of each model type.

1.2 Motivation for Our Project

Music genre classification is inherently a multimodal task. While audio features capture the sonic qualities of a track—such as rhythm and energy—lyrics provide complementary semantic information about themes, emotions, and cultural contexts. Relying solely on a single modality, such as audio or lyrics alone, often results in limited performance, especially for genres that are stylistically diverse or lyrically driven. Thus, integrating multiple modalities provides a richer and more holistic representation of each song, enabling models to better disambiguate between genres that are close in sonic space but distant in thematic or cultural space.

Our motivation in this work is to systematically explore how combining lyrics and audio features can enhance genre classification performance compared to single-modality approaches. Through this multimodal integration, we aim to build models that more closely approximate human-level genre understanding, which naturally synthesizes multiple sources of information.

1.3 Overview of Our Approach

To tackle the multimodal music genre classification task, we adopted a progressive and systematic modeling strategy. We began by building baseline models that use only single modalities — a simple MLP based on audio features and a CNN based on lyrics — to establish reference performance levels for each modality independently. Next, we explored feature fusion models, combining audio and lyrics representations through concatenation and joint learning, aiming to leverage complementary information between modalities.

To better capture sequential and contextual dependencies within lyrics, we implemented sequence models, including BiLSTM, GRU and Transformer encoder architectures. Recognizing the potential of pre-trained language models, we further fine-tuned BERT representations of lyrics, and combined them with ensemble methods such as Random Forest and MLP classifiers.

Through this stepwise exploration, we aim to systematically assess the impact of different modalities, fusion strategies, and architectural choices on the music genre classification task.

2 Data Introduction

2.1 Dataset Description

2.1.1 Initial Data

Before settling on our current dataset, we initially worked with a dataset from a Kaggle competition that contained 10 numerical genre labels. However, the exact mapping of these numerical labels to actual genre names was unknown.

To enhance this dataset, we developed a web scraper to collect lyrics for each song. As expected, we were unable to retrieve lyrics for all entries, leading us to discard songs without lyrics. This filtering resulted in a highly imbalanced dataset, and due to the lack of clarity regarding the genre label mappings, we were unable to augment the dataset meaningfully to address the imbalance. Therefore, we decided to switch to the current dataset instead.

2.1.2 Current Data

The current dataset is retrieved from Kaggle [1], comprising 18,454 song entries with 25 variables. A detailed description of all variables is provided in Appendix 1.

The target variable for our classification task is `playlist_genre`, representing the genre of each song. The six genres are: Rap, EDM, Pop, Rock, Latin, R&B

Our models utilize two components of the dataset: the lyrics, which provide textual information, and a set of audio features sourced from Spotify. These features include: Danceability, Energy, Key, Loudness, Mode, Speechiness, Acousticness, Instrumentalness, Liveness, Valence, Tempo, Duration (ms), Language

By combining both lyrical and audio data, we aim to predict a song's genre without the need for actual audio playback.

2.2 Data Preprocessing

To ensure the effectiveness of subsequent modeling, we conducted a comprehensive data preprocessing on the original Spotify song dataset, as outlined below:

2.2.1 Data Cleaning

Dropped irrelevant columns (e.g., track IDs, album details, playlist info), removed rows with missing values, and eliminated duplicate entries to ensure data consistency and integrity.

2.2.2 Feature Encoding

Applied one-hot encoding to categorical features (*mode*, *key*, *language*) and label encoding to the target variable *playlist_genre*, making the data suitable for machine learning models.

2.2.3 Feature Correlation Analysis and Selection

Computed the feature correlation matrix and visualized it using a heatmap to examine relationships with the target variable. Redundant features with high inter-correlation (above 0.8)

and features weakly correlated with the target genre (below 0.05) were removed, retaining only meaningful predictors.

2.2.4 Lyrics Field Cleaning

Removed entries with placeholder lyrics such as "Lyrics for this song have yet to be released" to avoid misleading the model with incomplete data.

2.2.5 Data Splitting

Split the cleaned dataset into training (80%) and testing (20%) sets, saving them as CSV files for subsequent model development. The final training set contains 13,348 samples, and the test set contains 3,627 samples.

2.2.6 Feature Standardization and Outlier Removal Based on training set

Standardized audio-related features (e.g., danceability, energy, loudness) using *StandardScaler* and removed outliers with Z-scores greater than 3 in training set to improve data quality and model robustness.

2.3 Data Analysis

2.3.1 Genre Distribution

The genre distribution still shows some class imbalance. For example, the number of Pop songs is almost twice that of Latin and EDM. One reason why EDM could be the least represented might be the lack of lyrics in this genre.

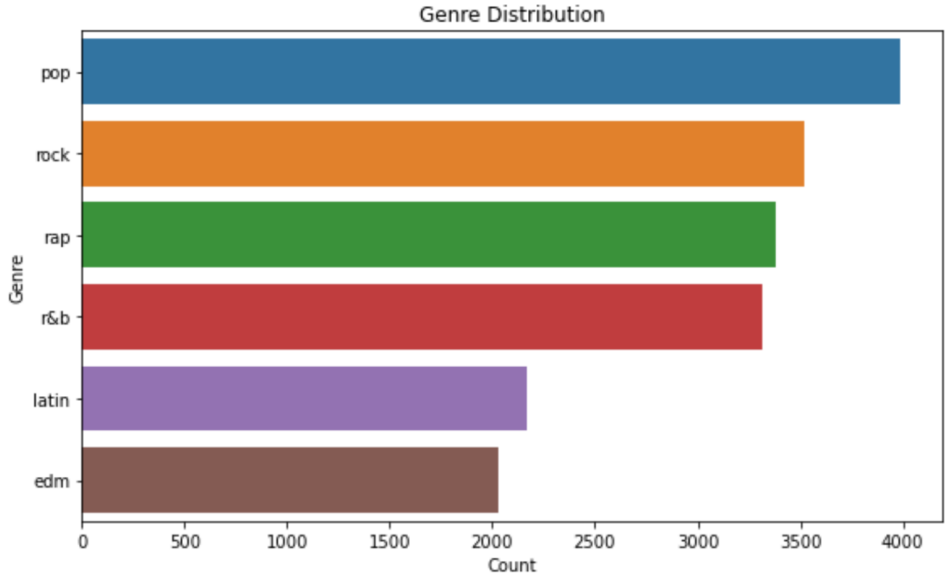

Figure 1: Dataset Genre Distribution

2.3.2 Feature Density by Genre

Based on the feature density analysis, we observed:

- **Rap**: High speechiness, moderate energy, and low danceability
- **EDM**: High energy, high tempo, and high danceability
- **Pop**: High danceability and valence
- **Rock**: High loudness and energy but less danceable
- **Latin**: High valence and danceability but low speechiness
- **R&B**: Low energy and loudness with an average valence

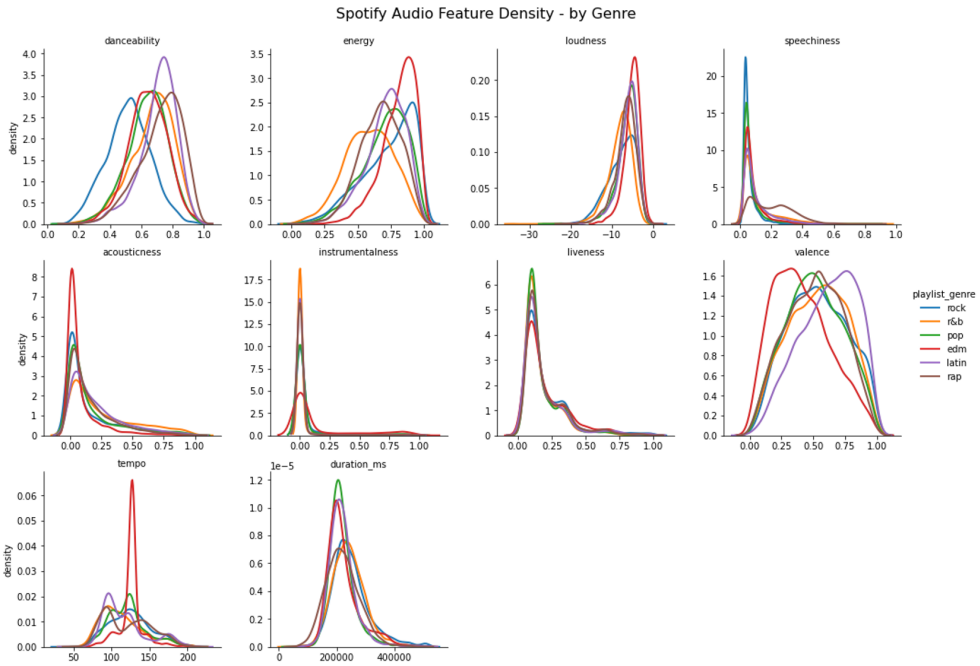

Figure 2: Feature Density

2.3.3 Feature Correlation

In terms of feature correlation:

- **Highest positive correlation**: Energy and loudness (0.67), indicating that louder songs are generally more energetic.
- **Second highest positive correlation**: Danceability and valence (0.34), suggesting that happier songs tend to be more danceable.

- **Lowest negative correlation**: Energy and acousticness (-0.55), meaning acoustic songs are generally less energetic.
- **Second lowest negative correlation**: Loudness and acousticness (-0.37), showing that louder songs tend to be less acoustic.

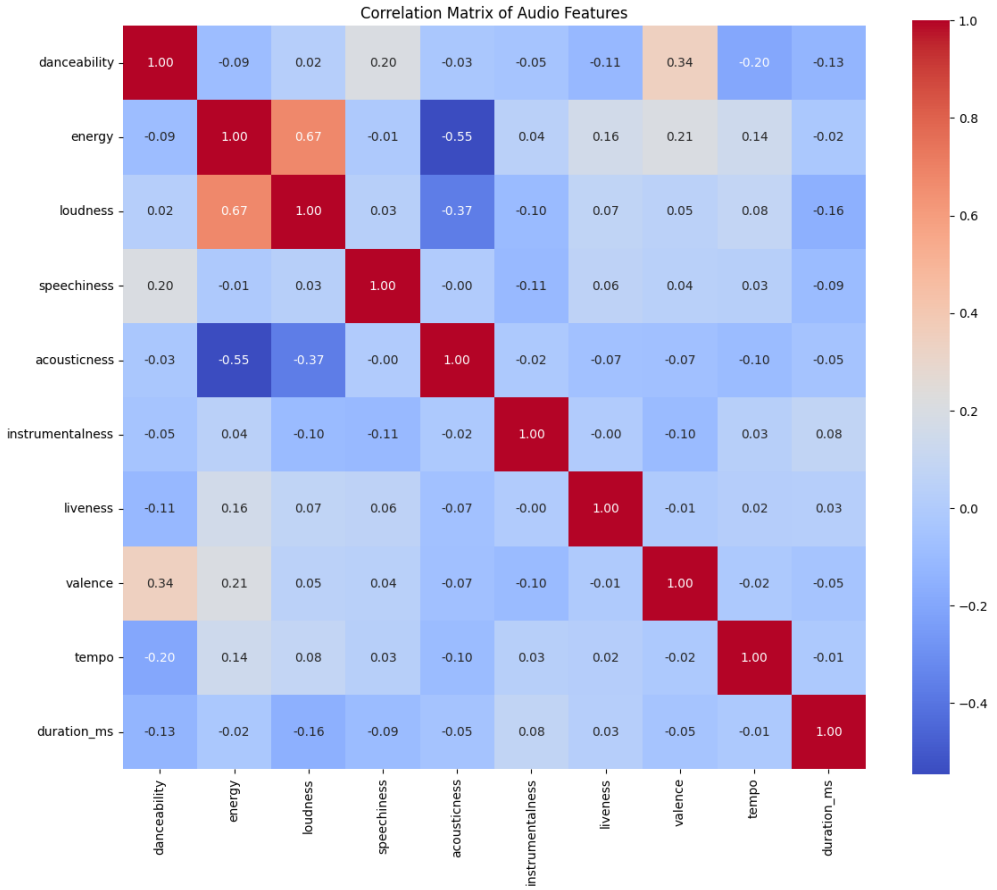

Figure 3: Feature Correlation

2.3.4 TF-IDF Analysis

The visualization of top TF-IDF words across genres highlights that:

- Frequent words like *oh*, *love*, and *don* are common across multiple genres.
- **Latin** and **Rap** stand out with more distinctive, genre-specific words, indicating stronger lyrical identities.
- **EDM**, **Pop**, **R&B**, and **Rock** exhibit similar top word patterns, suggesting that deeper analysis into sentence context or lyrical structure may be needed for better distinction.

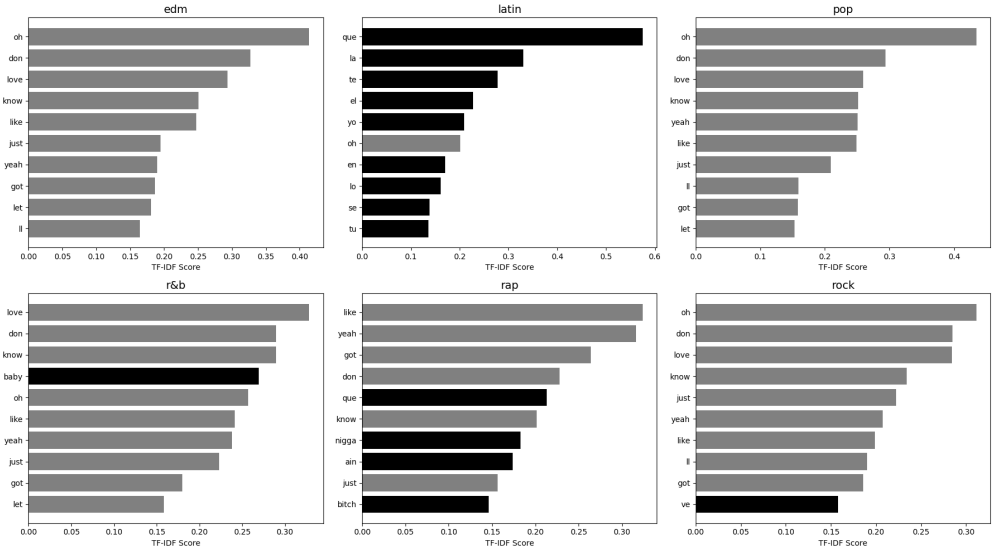

Figure 4: TF-IDF Words

3 Model Development

3.1 Models Overview

After completing data preprocessing and analysis, we proceeded to develop a series of machine learning and deep learning models for the music genre classification task. Our objective was to systematically explore how different architectures perform when leveraging lyrics, audio features, or both in a multimodal setting.

We started with baseline models that use only single modalities — such as audio-based MLPs and lyrics-based CNNs — to establish reference points. Building upon these, we explored increasingly sophisticated techniques, including feature fusion (CNN+MLP), sequence modeling (BiLSTM and GRU with attention), and Transformer-based architectures for enhanced contextual understanding.

Additionally, we integrated pre-trained models like BERT to capture deeper semantic representations from lyrics, and experimented with ensemble learning to combine the strengths of different models. In the following sections, we introduce each model individually, explain the intuition behind its design, describe its structure, and present the results obtained on the test dataset.

Lastly, we implemented an Ensemble model to mitigate individual bias for each model above. By doing a majority vote, we hope to make more reliable classifications overall.

3.2 Baseline: MLP Based on Audio Features

The genre of a song is deeply embedded in the physical properties of its audio. Different song styles usually have distinct combinations of audio features. This model uses structured audio features as input to capture these stylistic signals for genre prediction.

3.2.1 Model Structure

This model adopts a typical Multi-Layer Perceptron (MLP) structure, consisting of the following components:

- **Audio Feature Extraction Module:** This module alternates between multiple layers of linear transformations and activation functions to iteratively extract and compress key information from the audio features. The input is a set of structured audio features, and the output is a more expressive intermediate feature vector. Specifically, the hidden layer dimensions were determined through extensive hyperparameter tuning across multiple configurations.
- **Classification Module:** The extracted high-level audio representation vector is further passed into the classifier module, which applies a series of linear transformations and nonlinear activation functions to map the features to the final genre classification output space.

This model is designed for structured numerical features and efficiently models the differences between audio styles, making it particularly suitable for tasks where audio features have clear distribution differences and less reliance on lyrics. However, due to its lack of semantic processing for lyrics, it may misclassify genres that require textual understanding. Therefore, it serves as a basic model for audio feature modeling but is limited by the expressiveness of the structured features.

3.2.2 Results

The model achieved an accuracy of 53.07% (Test Loss: 1.2099) with a weighted average F1-score of **0.53** on the test set. These results indicate that the model demonstrates preliminary separability based on audio features for the six-class music genre classification task, but there remains significant room for optimization. Key observations include:

- **High-Performance Classes:** Rock (F1=0.63), Latin (F1=0.62), and Rap (F1=0.61) showed relatively strong separability, which may be attributed to the ability of audio features to capture rhythmic patterns characteristic of these genres.
- **Challenging Classes:** EDM (F1=0.34) exhibited the weakest performance, with a low recall of 28%, suggesting frequent misclassifications of EDM tracks into other genres. This likely reflects overlapping shallow feature representations between EDM and other genres.

Genre	Precision	Recall	F1-Score	Support
EDM	0.44	0.28	0.34	383
Latin	0.73	0.55	0.62	421
Pop	0.42	0.50	0.46	819
R&B	0.49	0.46	0.47	657
Rap	0.59	0.63	0.61	687
Rock	0.61	0.67	0.63	660

Table 1: Classification Report for MLP model Based on Audio Features

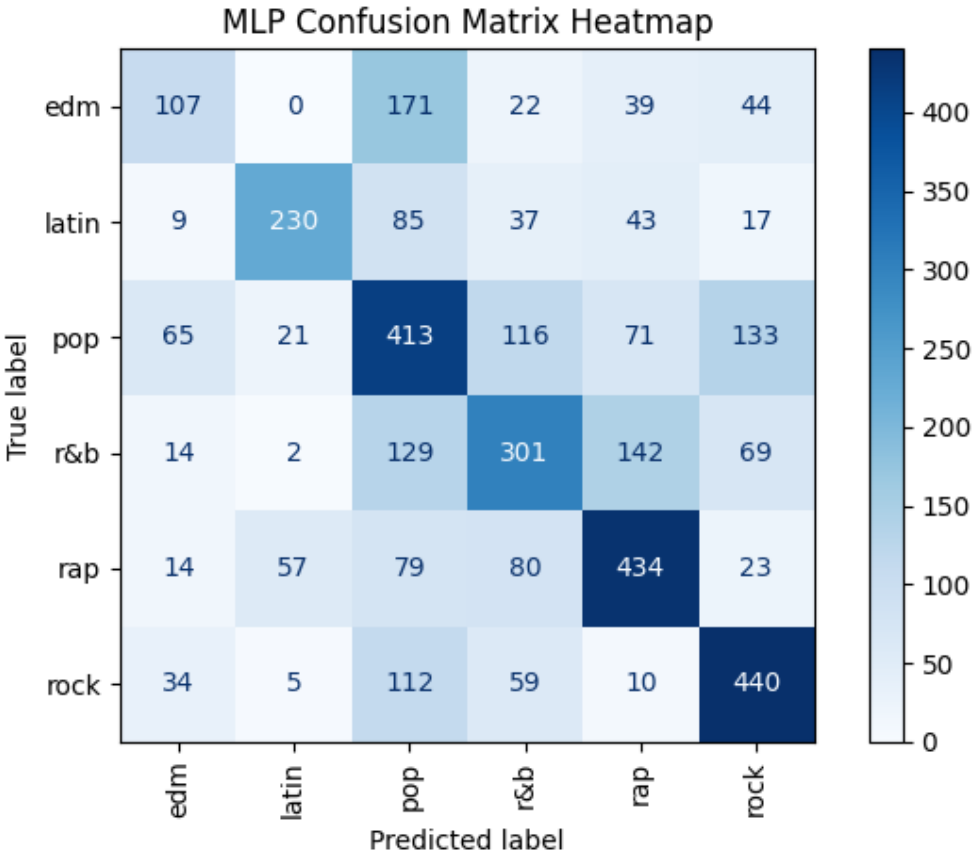

Figure 5: Confusion Matrix for MLP Model Based on Audio Features

3.3 Baseline: CNN Based on Lyrics

Lyrics are an essential part of a song, and different types of songs often use distinct language styles, rhythmic structures, and expressions. For example, Rap lyrics are dense, rhythmic, and often feature rhyme and repetition, whereas R&B lyrics tend to have more emotional expression. These differences can be reflected in forms like phrases and syntactic structures. By modeling the lyrics, we can extract information that represents the style, enabling the prediction of the song's genre.

In this project, we explore the use of Convolutional Neural Networks (CNNs) to extract

features from lyrics. Although CNNs are originally designed for image processing, they have been successfully applied to text data as well, particularly for capturing local patterns such as common phrases or repeated structures. Given the stylistic characteristics often present in lyrics, CNNs provide a reasonable approach for modeling such localized features.

3.3.1 Model Structure

Building on this idea, we designed a model based on a multi-scale 1D CNN architecture, which consists of the following components:

- **Embedding Layer:** Converts each word in the lyrics into a low-dimensional vector representation, forming a word embedding matrix.
- **Convolutional Module:** To capture linguistic patterns at different granularities, the model employs multiple parallel convolutional layers with kernel sizes of 3, 4, and 5, effectively extracting trigrams, four-grams, and five-grams from the lyrics. These patterns can be seen as meaningful local language structures within the lyrics. Each kernel "scans" the entire lyric sequence to capture the most discriminative local segments.
- **Max Pooling:** A one-dimensional max pooling operation is applied to each convolutional output to extract the strongest response from each feature map. This approach compresses the sequence information while highlighting the most important n-gram features, which typically represent key words or local sentence structures that determine the song's style.
- **Feature Concatenation and Classification Layer:** The pooled outputs from the three different convolutional kernels are concatenated into a high-dimensional feature vector, which is then passed through two fully connected layers for genre classification. The model's architectural parameters, such as the number of convolutional output channels and the size of the fully connected layers, were determined through hyperparameter tuning across multiple configurations.

This model focuses on capturing local semantic patterns from the lyrics, but since the convolutional output includes only the most prominent features of each kernel, it may lose some complex semantic information, especially for longer texts. Therefore, while the model is a highly efficient text classifier in a single-text modality, its capabilities are limited by the expressiveness of the lyrics themselves.

3.3.2 Results

The CNN model using lyrics achieved an accuracy of 56.63% (Test Loss: 1.2321) with a weighted average F1-score of 0.56, representing an improvement of 3.56 percentage points over the audio-based MLP model. This enhancement highlights the supplementary value of textual information in lyrics (e.g., semantic themes, lexical styles) for genre classification. Key findings include:

- **High-Performance Classes:** Rap (F1=0.72) and Rock (F1=0.61) performed exceptionally well.

  - Rap: Likely benefited from the CNN's ability to detect rhythmic patterns and slang-specific vocabulary.
  - Rock: Achieved a high recall of 84% but low precision of 48%, indicating a tendency to overgeneralize the "Rock" label to other genres.

- **Challenging Classes:** EDM (F1=0.37) remained the weakest class, with an extremely low recall of 27%, reflecting the limited ability of lyrics to characterize instrumental-driven genres like EDM.

Genre	Precision	Recall	F1-Score	Support
EDM	0.59	0.27	0.37	383
Latin	0.71	0.53	0.61	421
Pop	0.48	0.45	0.46	819
R&B	0.57	0.47	0.52	657
Rap	0.73	0.71	0.72	687
Rock	0.48	0.84	0.61	660

Table 2: Classification Report for CNN Model Based on Lyrics

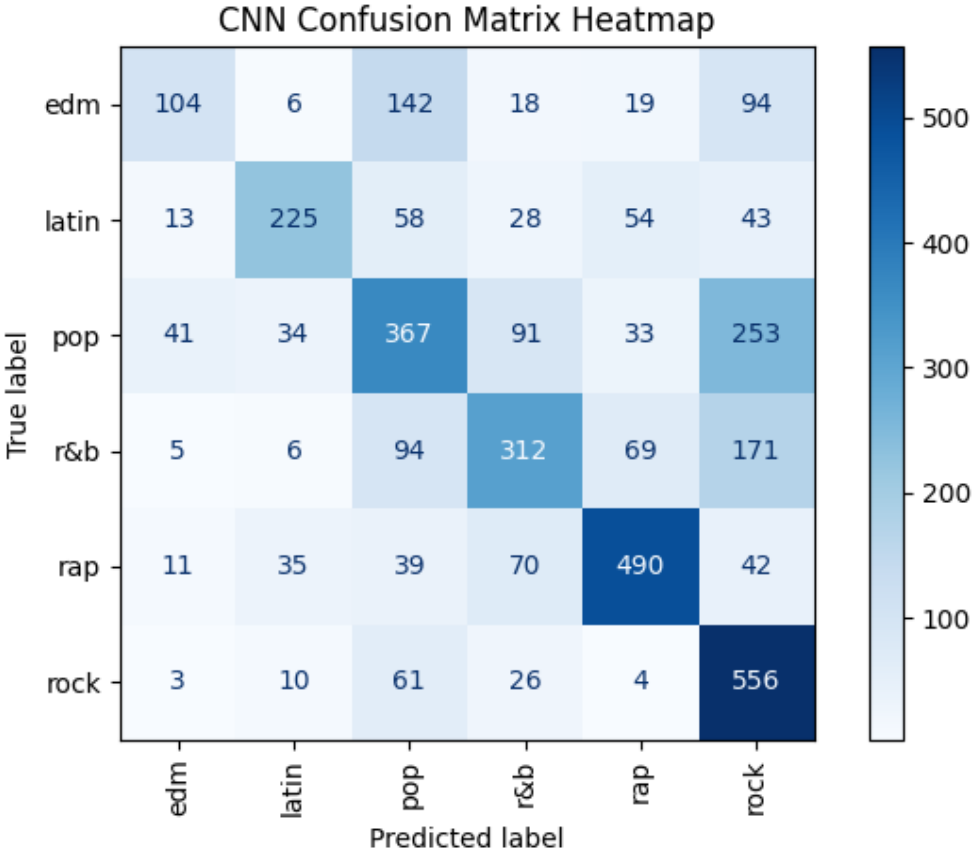

Figure 6: Confusion Matrix for CNN Model Based on Lyrics

3.4 CNN+MLP Fusion Model

3.4.1 Model Motivation

The true style of a song is often a combination of both its lyrical expression and audio characteristics. Using only a single modality typically doesn't capture the full picture. By combining lyrics and audio features, the model can make judgments on multiple dimensions, improving classification performance.

3.4.2 Model Structure

This model is a typical feature-level fusion multimodal model and consists of the following components:

- **Lyrics Feature Extraction Module:** The structure is identical to the CNN Based on Lyrics model.
- **Audio Feature Extraction Module:** The structure is identical to the Multilayer Perceptron Based on Audio Features model.
- **Feature Fusion and Classification Module:** After concatenating the lyrical and audio representations, the combined feature vector is processed by a two-layer fully connected network to generate the final genre classification.

This model fully leverages both the semantic content of the lyrics and the stylistic signals from the audio, offering the following advantages:

- **Stronger expressiveness:** Simultaneously modeling text and audio captures both the style and content dimensions.
- **Complementary information:** When one modality is weak, the other can provide complementary support.

As a multimodal fusion model, this structure achieves better classification performance than single-modality models without significantly increasing model complexity.

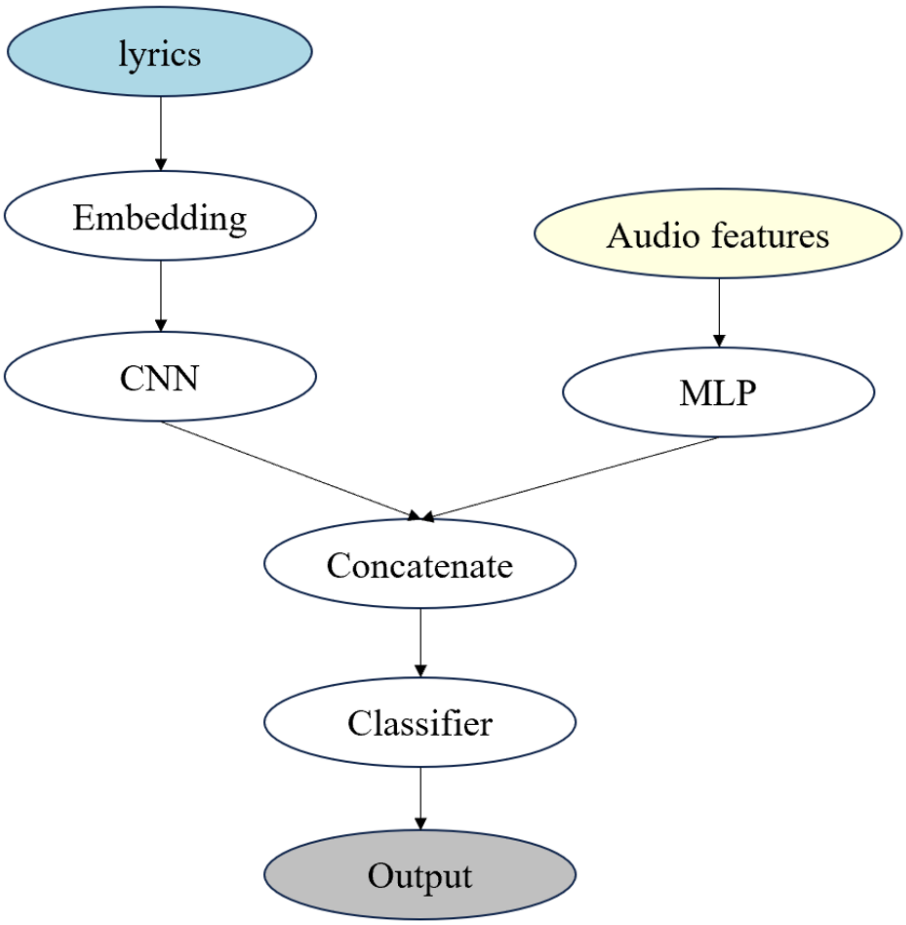

Figure 7: CNN+MLP Fusion Model Architecture

3.4.3 Results

By fusing audio (MLP) and lyric (CNN) features via concatenation, the model achieved an accuracy of 61.76% (Test Loss: 1.0425) and a weighted average F1-score of 0.62, demonstrating significant improvements over single-modality models (+8.69% vs. audio, +5.13% vs. lyrics). Key conclusions include:

- **Validation of Modality Complementarity:** The F1-score of EDM improved from a maximum of 0.37 (single modality) to 0.52, confirming the synergistic effect of audio-lyric features. The reduced test loss (decreased by 0.16 and 0.18 compared to audio and lyric models, respectively) suggests that multimodal fusion mitigates overfitting risks.
- **Classification Performance Divergence:**
  - **High-Performance Classes:** Rap (F1=0.71) and Rock (F1=0.71) maintained robust performance, indicating strong alignment between their audio and lyric features.

– **Bottleneck Classes:** EDM (F1=0.52) and Pop (F1=0.53) remained challenging. The confusion matrix revealed a primary misclassification triangle (Pop → EDM, Pop → Rock, R&B → Pop), likely due to overlapping audio energy features or lyric generality. Pop, as the largest class, became a major misclassification target for EDM, Rock, and R&B, reflecting its broad feature distribution that current models struggle to distinguish.

Genre	Precision	Recall	F1-Score	Support
EDM	0.51	0.52	0.52	383
Latin	0.74	0.57	0.65	421
Pop	0.50	0.56	0.53	819
R&B	0.61	0.55	0.58	657
Rap	0.76	0.68	0.71	687
Rock	0.65	0.78	0.71	660

Table 3: Classification Report for CNN+MLP Fusion Model

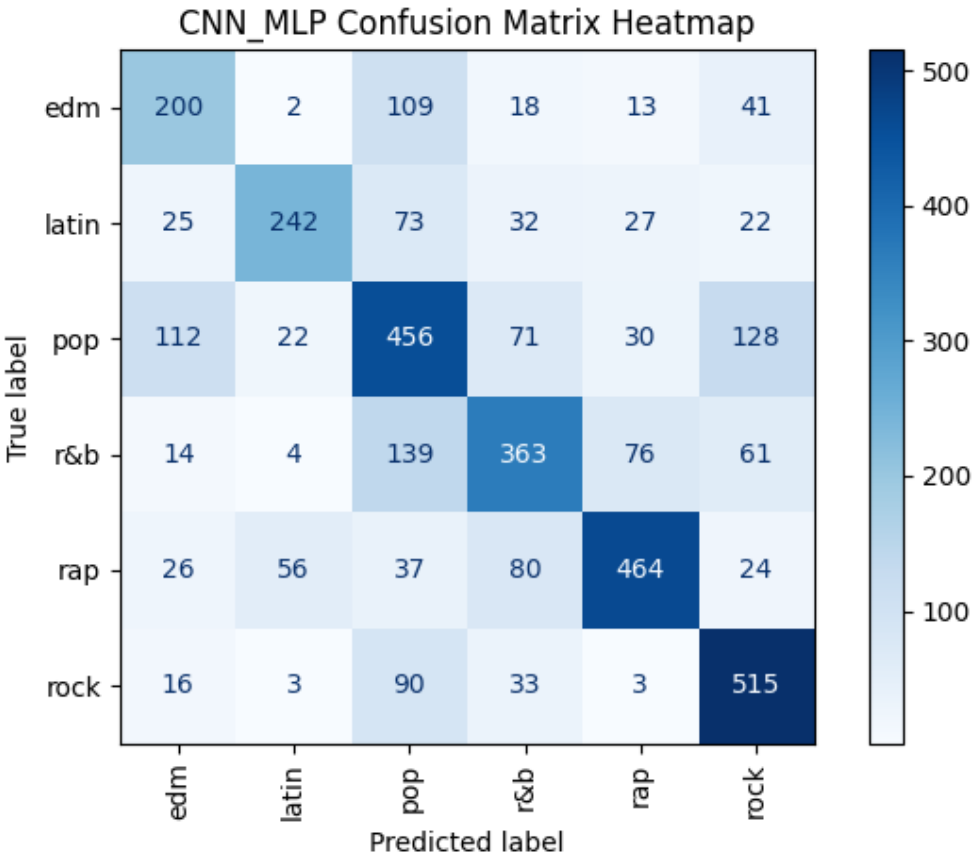

Figure 8: Confusion Matrix for CNN+MLP Fusion Model

3.5 AttentionBiLSTM

3.5.1 Model Motivation

Since lyrics are like sentences—they have order and meaning that depends on word position. An LSTM helps the model understand the flow and meaning of lyrics by learning which words come before and after. We use a **bidirectional LSTM**, which means it looks at the lyrics from both directions (forward and backward), capturing more context.

Since not every word is important for figuring out the genre, the **attention layer** learns to focus on the most helpful words. This makes the model smarter and more accurate.

Besides lyrics, things like **tempo**, **energy**, and **loudness** also give strong clues about the genre. For example, a fast tempo might suggest rock or EDM. We use a small neural network to process these features and turn them into a format the model can use.

After processing the lyrics and the numerical features separately, we **combine** them before making a prediction. This allows the model to learn from both words and numbers, making it stronger than using only one type of input.

3.5.2 Model Structure

In our final model architecture:

- We first process the lyrics using an **embedding layer**, followed by a **bidirectional LSTM layer** and an **attention mechanism**.
- For the numerical features, we use a **multilayer perceptron (MLP)** with **ReLU activation** and **dropout** to introduce non-linearity and mitigate overfitting.
- The outputs from both branches are **concatenated** and passed through a **final MLP layer**, which also incorporates **ReLU** and **dropout**.
- The output of this final layer is the **predicted genre**.

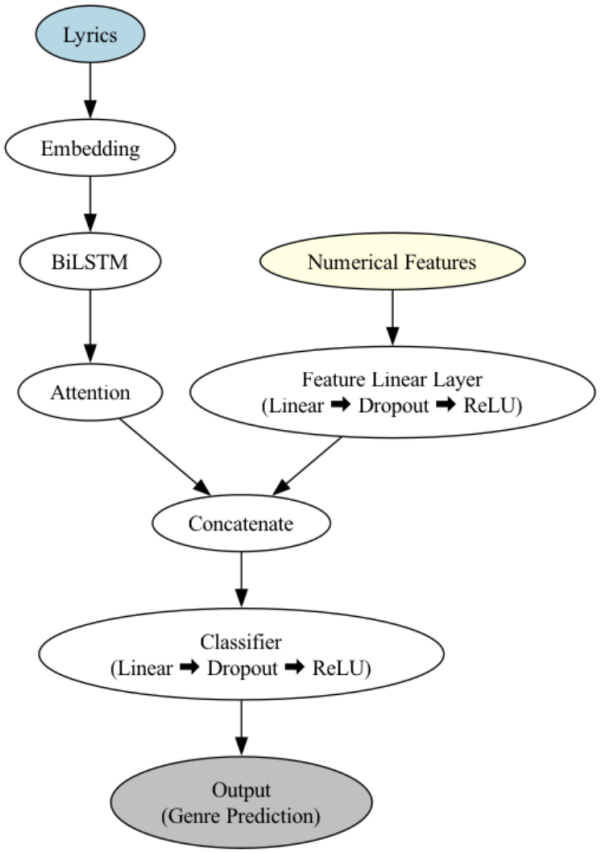

Figure 9: BiLSTM Model Structure

3.5.3 Results

When using both lyrics and audio features, the model had around **1.59M parameters** and achieved an **accuracy of about 0.61**. However, when relying only on lyrics, the parameter count was similar at **1.57M**, but the accuracy dropped noticeably to **0.55**. This shows that audio features provide valuable additional information that improves classification.

Among all genres, **pop** had the poorest classification performance, likely due to its broad and overlapping definition with other genres. On the other hand, **rock** and **R&B** showed the most improvement when audio features were added, suggesting that their musical characteristics are important cues for distinguishing them.

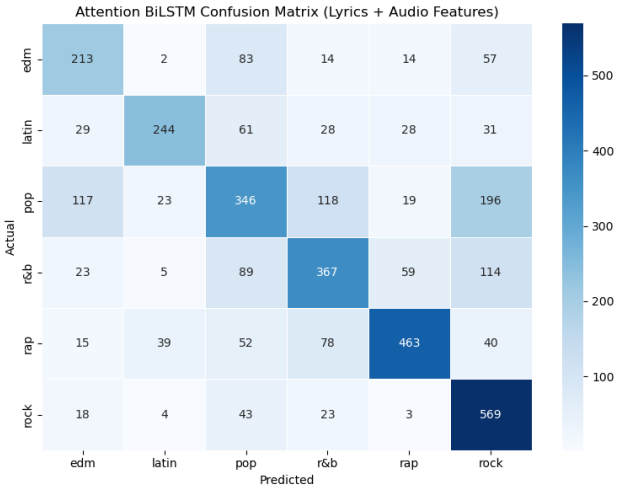

Figure 10: BiLSTM Model Results

3.6 GRU + Cross Attention

3.6.1 Model motivation

Lyrics are inherently sequential, with long-range dependencies that simple models like CNNs may fail to capture. We use a GRU to encode the lyrics efficiently, balancing memory capability and computational cost, especially given our median lyric length of around 300 tokens. To enable dynamic interaction between audio and lyrical modalities, we apply a cross-attention mechanism where audio features guide the focus over the encoded lyrics. Incorporating artist embeddings further provides stylistic priors, helping the model disambiguate genres when content features alone are insufficient.

3.6.2 Model Structure

- **Lyrics Encoding**: Lyrics are tokenized and passed through an embedding layer, mapping each word into a 128-dimensional space. These embeddings are then processed by a GRU (Gated Recurrent Unit), which captures the sequential structure and contextual dependencies within the lyrics.
- **Audio Feature Projection**: Audio features — including danceability, energy, loudness, and others — are projected from their original 7-dimensional space into a 128-dimensional hidden space via a linear layer.
- **Cross-Attention Fusion**:
  - The projected audio feature acts as the **query**.
  - The encoded lyrics (GRU outputs) act as the **keys** and **values**.

  This mechanism allows the model to dynamically focus on the most relevant parts of the lyrics based on the audio characteristics.
- **Artist Embedding**: Each artist ID is mapped to a 32-dimensional embedding vector, capturing latent stylistic traits associated with the artist.
- **Final Classification**: The attention-fused lyrics-audio representation is concatenated with the artist embedding into a single feature vector, which is then passed through a two-layer MLP classifier (with ReLU activation and dropout) to predict the music genre.

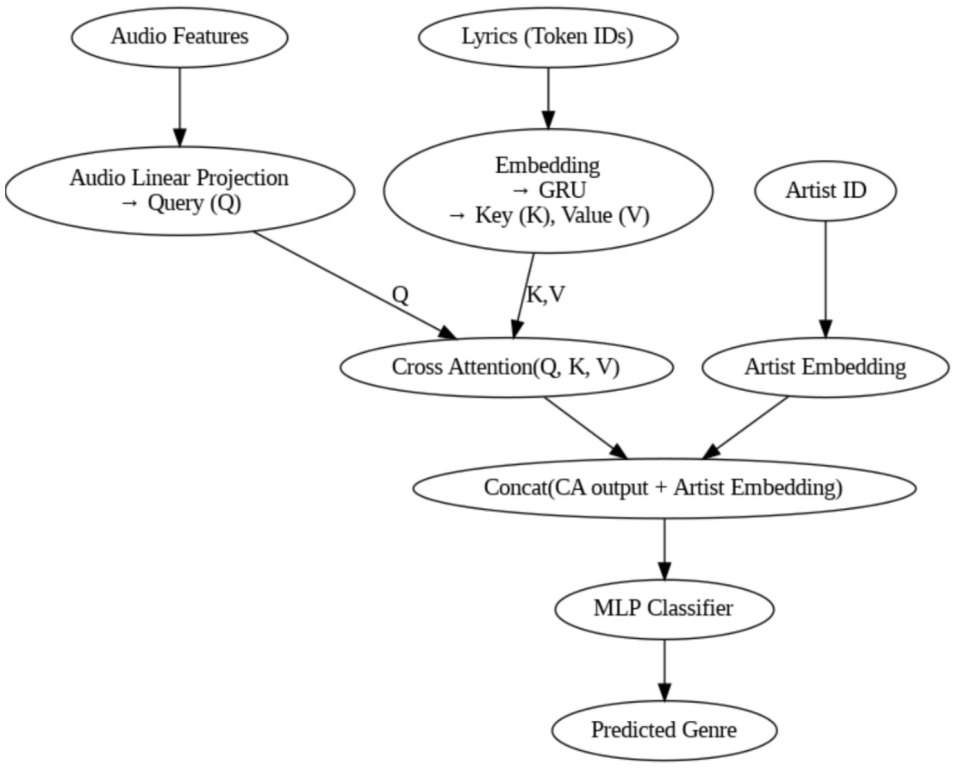

Figure 11: GRU + Cross Attention Model Architecture

3.6.3 Results

Based on the classification reports, the GRU + Cross Attention model achieved: Accuracy: 60%, Recall: 60%, F1-Score: 60% The results demonstrate that integrating lyrics, audio features, and artist information via GRU and cross-attention achieves solid performance on the genre classification task. We see the model predict Rap and Rock well with above 0.7 accuracy, however, we can also see edm and pop being the bottleneck classes.

Classification Report:

	precision	recall	f1-score	support
edm	0.4525	0.4230	0.4372	383
latin	0.6542	0.5796	0.6146	421
pop	0.4945	0.4908	0.4926	819
r&b	0.5985	0.6012	0.5998	657
rap	0.7278	0.7278	0.7278	687
rock	0.7011	0.7818	0.7393	660
accuracy			0.6118	3627
macro avg	0.6048	0.6007	0.6019	3627
weighted avg	0.6092	0.6118	0.6098	3627

Figure 12: Classification Report for GRU + Cross Attention

We also run a comparison of the same model structure without artist information: Accuracy: 57%, Recall: 56%, F1-Score: 56% . This intuitively makes sense as artist identity can play a critical role in music genre classification. Many artists are strongly associated with specific genres or stylistic patterns. Including artist information yields an additional 0.03 better accuracy, indicating it could provide additional contextual cues that enhance genre prediction.

📋 Classification Report:

	precision	recall	f1-score	support
edm	0.4058	0.3655	0.3846	383
latin	0.6813	0.5534	0.6107	421
pop	0.4716	0.4969	0.4839	819
r&b	0.5611	0.5099	0.5343	657
rap	0.6889	0.7060	0.6973	687
rock	0.6005	0.7061	0.6490	660
accuracy			0.5696	3627
macro avg	0.5682	0.5563	0.5600	3627
weighted avg	0.5698	0.5696	0.5678	3627

Figure 13: Classification Report for GRU + Cross Attention (without artist)

Looking at the confusion matrix, the GRU + Cross Attention model struggled particularly with distinguishing EDM from Pop, which is expected given the musical similarity between some tracks in these genres. However, the model performed well in predicting Rap and Rock genres overall.

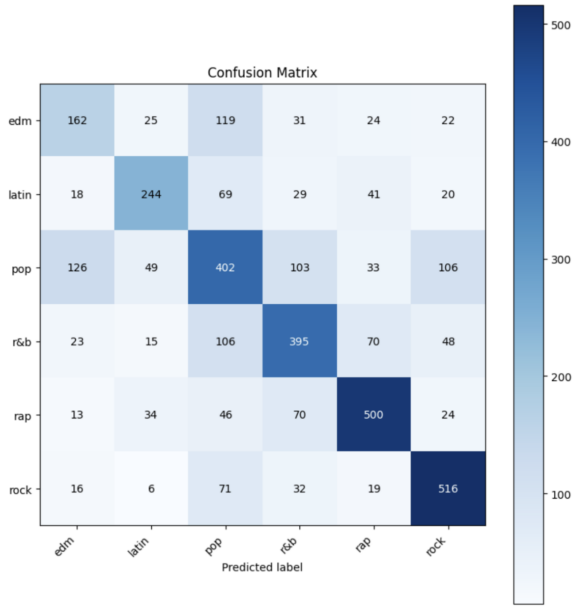

Figure 14: Confusion Matrix for GRU + Cross Attention

3.7 BERT - Chunked Lyrics

3.7.1 Model Motivation

We explored the application of Bidirectional Encoder Representations from Transformers (BERT) as both a standalone and a hybrid classifier. BERT, pre-trained on large-scale text corpora using masked language modeling and next sentence prediction, is capable of understanding context beyond individual keywords. By choosing BERT, we aim to leverage its ability to capture semantic meanings to improve genre prediction.

3.7.2 Model Structure

- **BERT-Only**:
  - Due to BERT's maximum input token limit of 512, lyrics were split into overlapping chunks using a sliding window mechanism.
  - Each chunk was independently tokenized and passed through BERT.
  - A classification layer was applied to each chunk during training.
  - For final prediction, genre labels from all chunks of a song were aggregated using majority voting.

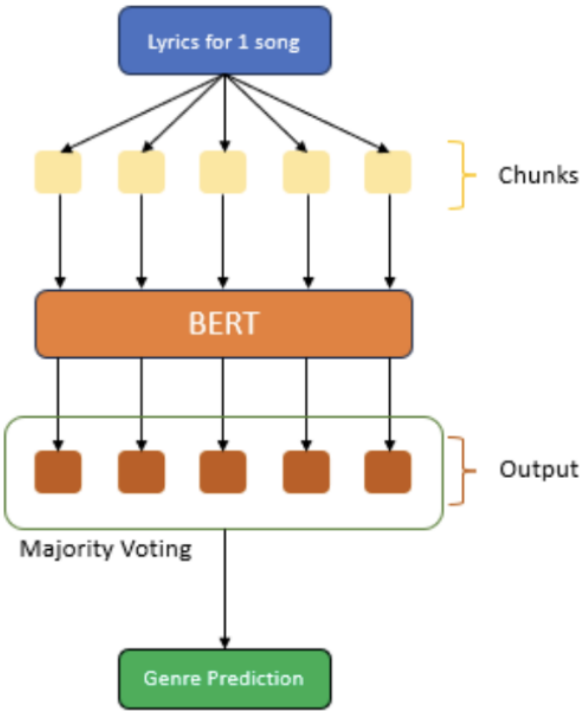

Figure 15: Model Structure- Bert Only

- **BERT + Audio Features (MLP)**:
  - BERT embeddings for all chunks were averaged to obtain a single fixed-size vector per song.
  - This averaged BERT embedding was concatenated with standardized audio features.
  - The concatenated vector was passed through a small MLP classifier for genre prediction.
  - Averaging embeddings treats BERT outputs as general features, rather than individual chunk predictions, making them suitable for MLP input.

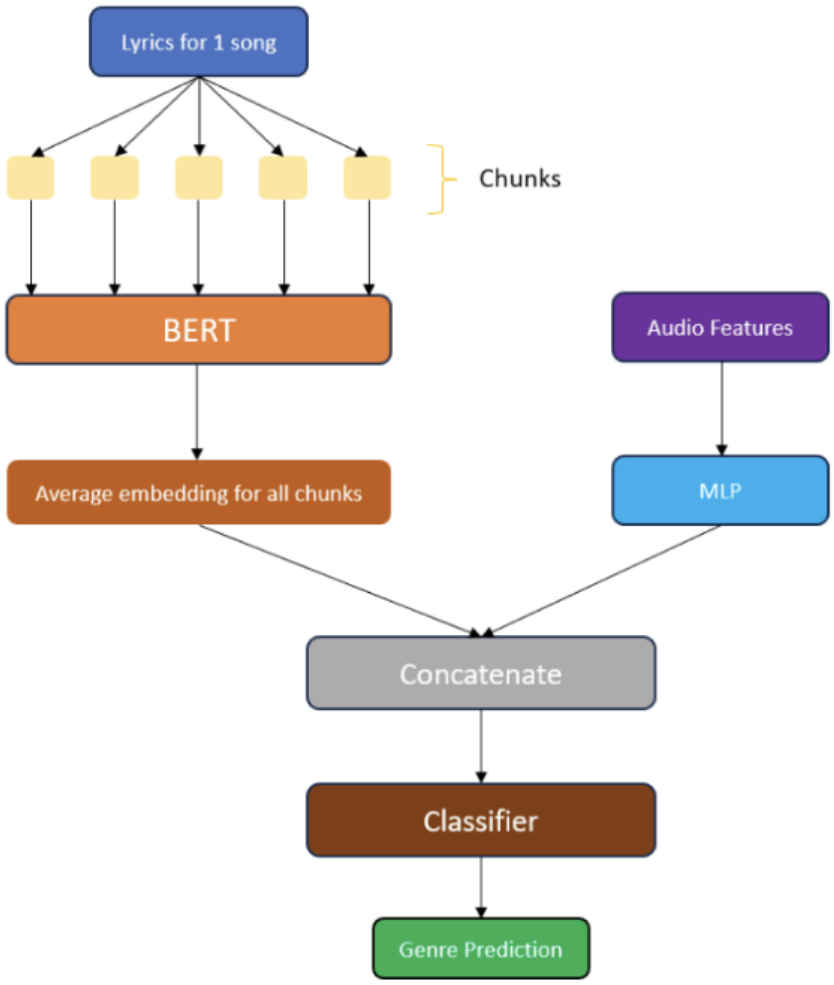

Figure 16: Model Structure- BERT + Audio Features (MLP)

- **BERT + Audio Features (Random Forest)**:
  - Similar to the BERT + MLP model, averaged BERT embeddings were concatenated with standardized audio features.
  - Instead of an MLP, a Random Forest classifier was used.
  - Random Forest captures nonlinear interactions between audio and lyrical semantics, contrasting with the parametric nature of MLP.

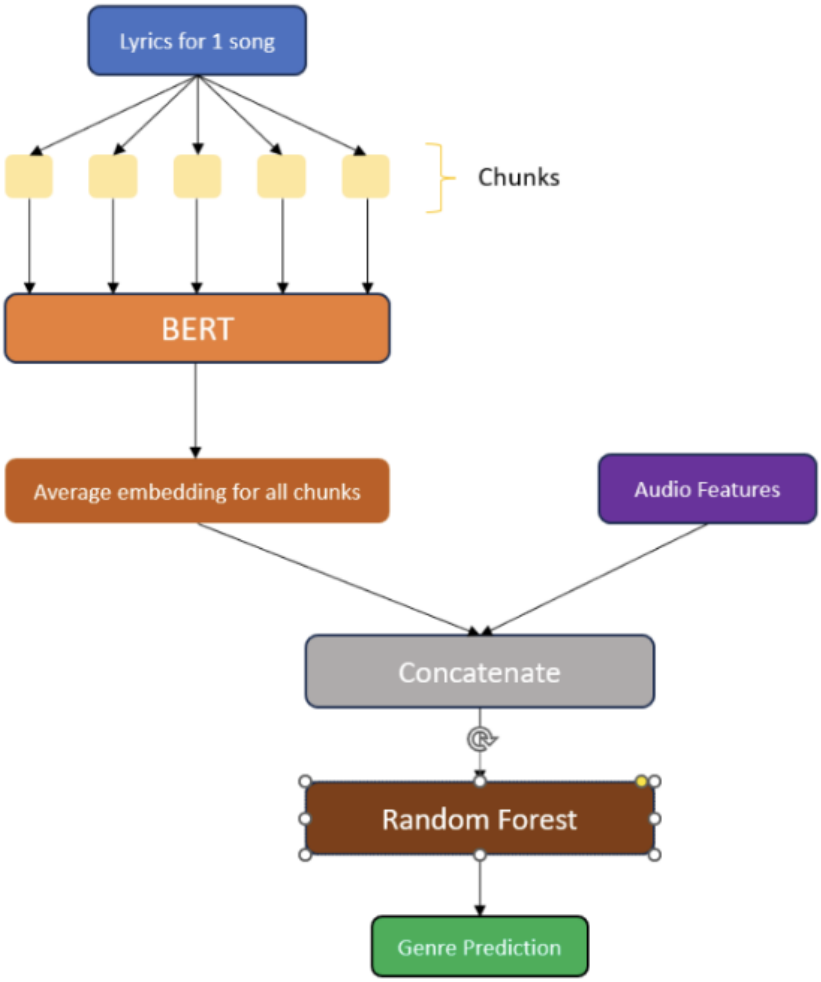

Figure 17: BERT + Audio Features (Random Forest)

3.7.3 Results

- **BERT-Only**: Achieved an accuracy of 59%, with a recall of 56% and F1-score of 57%.
- **BERT + Random Forest**: Also achieved an accuracy of 59%, but with slightly better recall and F1-score of 58% each.

- **BERT + MLP**: Achieved the lowest accuracy of 57%, but the best recall (59%) and F1-score (58%), indicating better handling of class imbalance and semantic generalization.

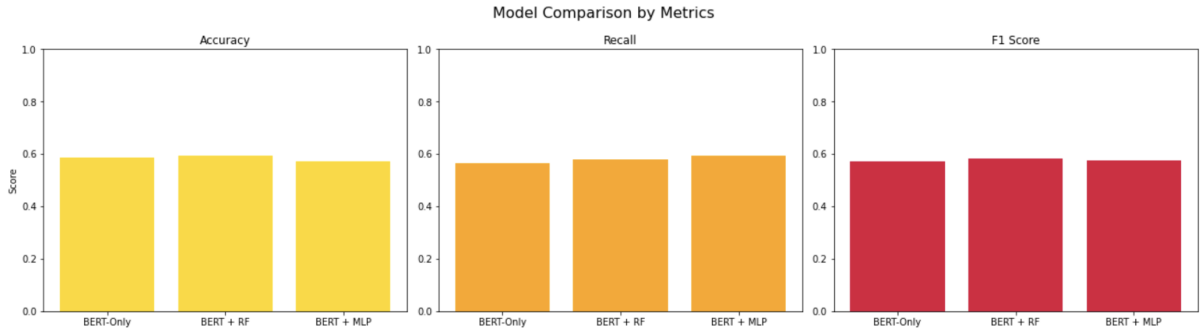

Figure 18: BERT-based model: Accuracy Comparison

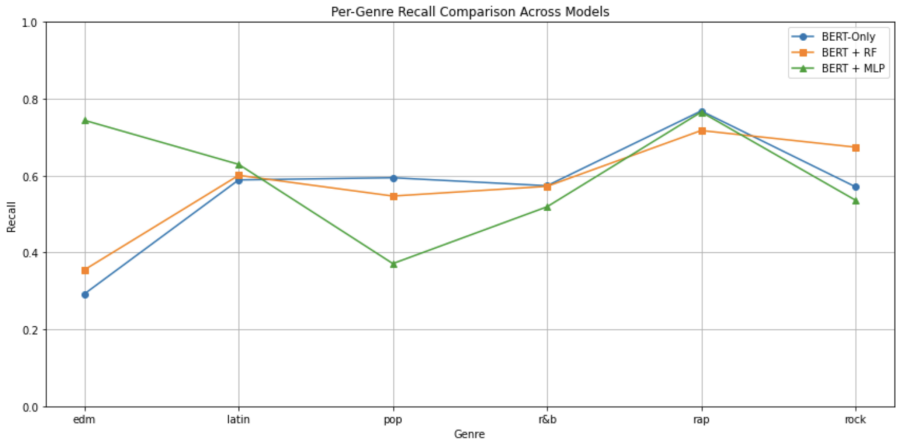

Figure 19: BERT-based Models: Accuracy across Genres

Analysis of Confusion Matrix

- **BERT-Only**: Struggled significantly with distinguishing EDM from Pop.
- **BERT + Random Forest**: Reduced misclassifications slightly, but still exhibited notable overlap among Pop, Rock, and EDM.
- **BERT + MLP**: Achieved the best genre-specific precision, though it continued to face challenges in distinguishing Pop and EDM.

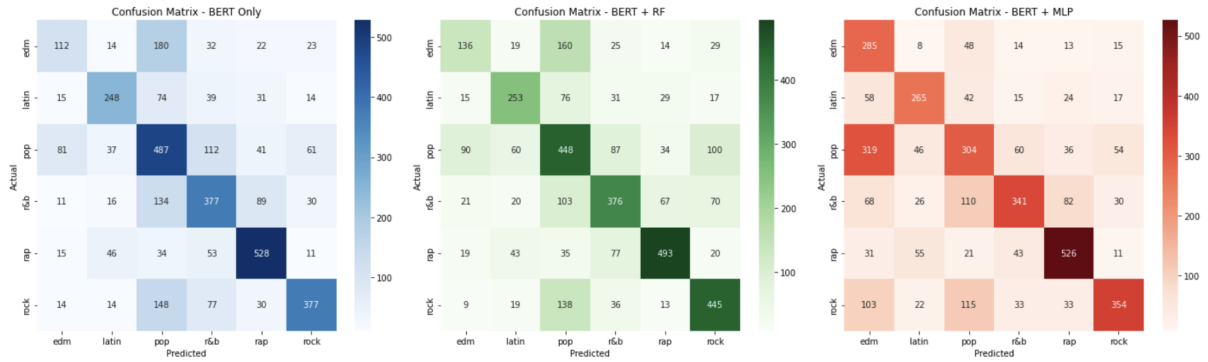

Figure 20: Confusion Matrix for BERT-based Models

3.8 Transformer

3.8.1 Model Motivation

The Transformer+MLP model was chosen to address the music genre classification problem because it effectively handles both sequential (lyrics) and tabular (numerical and dummy features) data. The Transformer excels at capturing long-range dependencies and contextual patterns in lyrics through its attention mechanism, which is crucial for understanding semantic relationships in text. Meanwhile, the MLP processes numerical features (e.g., danceability, energy) and dummy variables (e.g., language indicators) to extract relevant patterns from structured data.

Combining these two architectures allows the model to leverage both textual and numerical information, making it well-suited for a multi-modal classification task like predicting music genres.

3.8.2 Model Structure

The model learns patterns through two parallel paths, as shown in the following workflow:

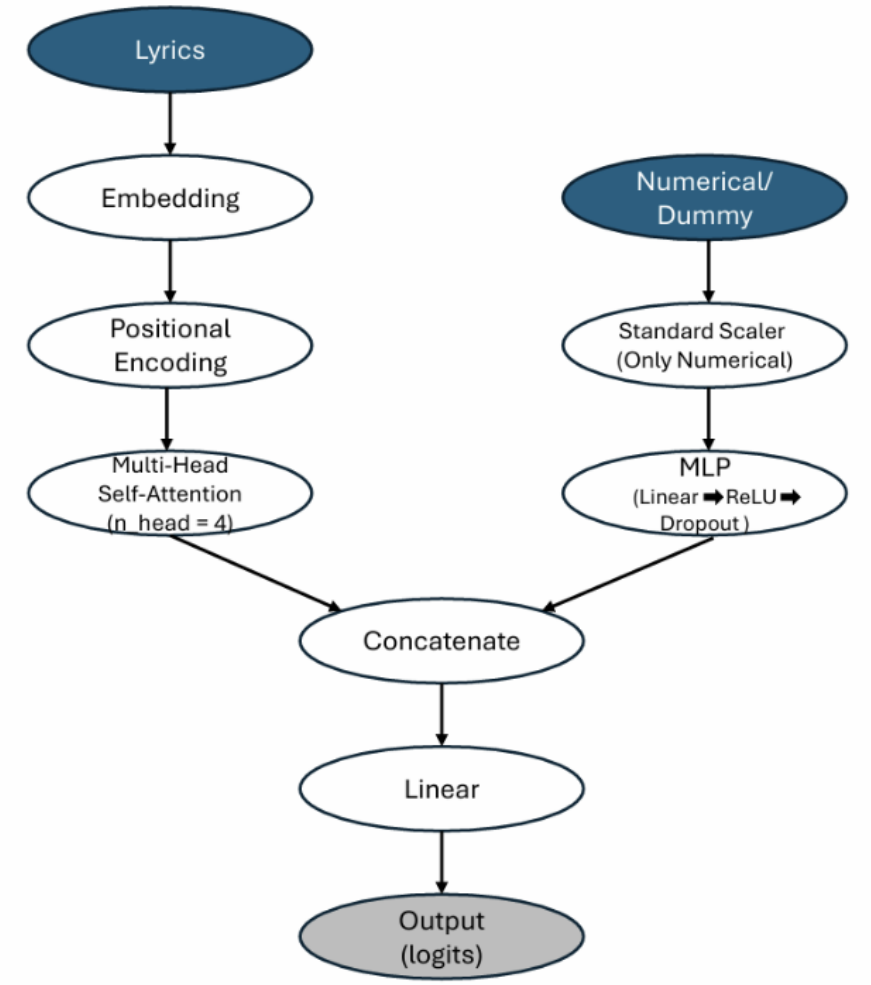

Figure 21: Model Structure for Transformer

- **Lyrics Path (Transformer)** Lyrics are tokenized into sequences and mapped to embeddings using an `nn.Embedding` layer, forming an embedding bank. Positional encodings are added to preserve sequence order. The Transformer encoder, consisting of 2 layers, uses a multi-head self-attention mechanism (4 heads) to weigh the importance of each token relative to others, capturing contextual relationships (e.g., how "singing" relates to "wind" in a lyric). This is followed by a feed-forward network (FFN) for further feature transformation. The output is mean-pooled into a fixed-length vector.
- **Numerical/Dummy Path (MLP)** Numerical features (e.g., danceability) are standardized, concatenated with dummy variables, and passed through an MLP with structure ($11 \to 128 \to 64 \to 32$) to extract patterns.
- **Combination** The outputs from the Transformer and the MLP are concatenated and fed into a final linear layer for 6-class classification.

- **Attention Mechanism** The multi-head self-attention computes attention scores using:

$$\text{scores} = \frac{QK^\top}{\sqrt{d_k}},$$

applies softmax to obtain weights, and computes a weighted sum of values:

$$\text{attention\_output} = \text{attention\_weights} \times V,$$

allowing the model to focus on relevant tokens.

- **Learning and Saving** The model is trained using cross-entropy loss and the Adam optimizer. During training, gradients update the embedding bank, Transformer parameters, MLP weights, and the final classifier. The best model (based on validation accuracy) is saved as `best_model.pt`.

3.8.3 Results

The model achieved a best validation accuracy of **0.5793** at epoch 6. The confusion matrix provides the following insights:

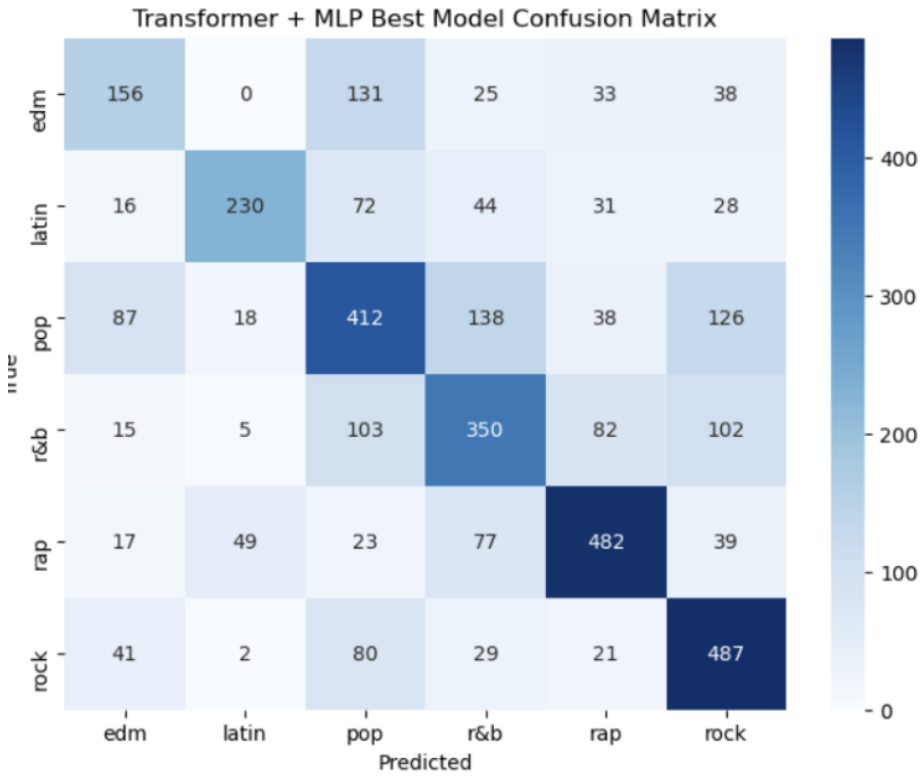

Figure 22: Confusion Matrix for Transformer Model

- **Misclassifications:**

  - **Pop** is often misclassified as **r&b** (103 instances) and **edm** (131 instances), suggesting overlap in features.
  - **Edm** is frequently confused with **pop** (87 instances).
  - **R&b** has notable misclassifications with **pop** (138 instances).

- **Overall Accuracy:** The best validation accuracy of **0.5837** indicates moderate performance, with room for improvement in distinguishing certain genres.

- **Key Observations:**

  - **Moderate Performance**: The model achieves a validation accuracy of 0.5837, reflecting reasonable but improvable generalization.
  - **Overfitting Risk**: The training accuracy (0.6160) is slightly higher than validation accuracy (0.5837), and validation loss increases steadily from epoch 4 to 10 (1.1491 to 1.4475), suggesting potential overfitting.
  - **Genre Confusion**: Genres like pop, r&b, and rock show significant misclassification, possibly due to overlapping lyrical or numerical features (e.g., similar energy or danceability values).

- **Conclusion:** The Transformer+MLP model demonstrates moderate success in music genre classification, with a best validation accuracy of 0.5837. It performs well for **rock** and **rap** but struggles with distinguishing **pop**, **r&b**, and **edm**. To improve performance, a larger and more balanced dataset, better feature engineering (e.g., using pre-trained embeddings for lyrics), or advanced regularization techniques could be explored.

3.9 Ensemble (Majority Vote)

The ensemble model works by combining the predictions from the previous different models using **majority voting**. Instead of relying on a single model, the final prediction is determined by which label the majority of models agree on.

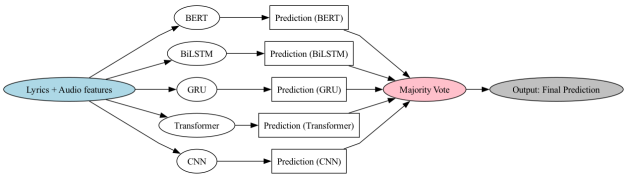

Figure 23: Ensemble Model Structure

This simple but effective strategy helped us achieve an accuracy of **0.66**, outperforming all individual models, which had accuracies around 0.62 at best. This result shows that each model captures different patterns and makes different types of errors, and by aggregating their predictions, the ensemble model can correct for individual weaknesses and make more reliable classifications overall. However, the model also costs the most as it requires to run through all models.

Genre	Precision	Recall	F1-Score	Support
EDM	0.57	0.50	0.53	383
Latin	0.76	0.58	0.66	421
Pop	0.56	0.54	0.55	819
R&B	0.65	0.65	0.65	657
Rap	0.79	0.72	0.75	687
Rock	0.65	0.89	0.75	660

Table 4: Classification report for Ensemble model

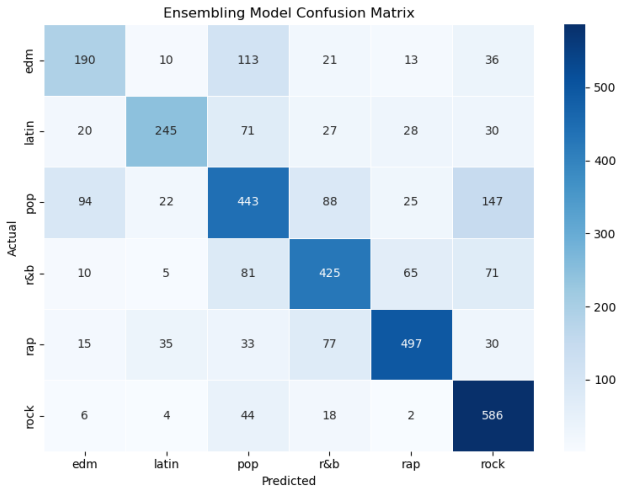

Figure 24: Ensemble Model Results

Genre	Base1	Base2	CNN	BiLSTM	Transformer	BERT1	BERT2	BERT3	GRU
edm	0.44	0.59	**0.51**	**0.51**	0.47	0.45	0.47	0.33	0.45
latin	0.73	0.71	0.74	0.71	**0.76**	0.66	0.61	0.63	0.65
pop	0.42	0.48	**0.50**	0.42	**0.50**	0.46	0.47	0.47	0.49
r&b	0.49	0.57	**0.61**	0.48	0.53	0.55	0.59	0.67	0.60
rap	0.59	0.73	**0.76**	0.70	0.70	0.71	**0.76**	0.74	0.63
rock	0.61	0.48	0.65	0.55	0.59	0.73	0.65	**0.74**	0.70
Overall	0.53	0.56	0.62	0.61	0.58	0.59	0.59	0.57	0.61
Num	2.4K	1.9M	1.9M	1.6M	10.3M	109.5M	109.5M	109.6M	1.6M

Table 5: Comparison of Accuracy Across Different Models

3.10 Results Comparison

In our experiments, we observed that using either **audio features** (via MLP) or **lyrics alone** (via multiple models) provided modest classification performance, with accuracies of **0.53** and about **0.56**, respectively. These results suggest that both modalities contain relevant but incomplete information for genre prediction. However, when we combined the two inputs in our models, accuracy significantly **increased to 0.66 for the Ensemble model**. This improvement confirms that **lyrics and audio features contribute different, complementary signals**. Leveraging both allows the model to make more informed predictions, highlighting the effectiveness of a multi-modal approach in music genre classification. However, this also comes with a significantly increasing cost, a higher number of parameters.

Despite using different architectures and methods to process lyrics—ranging from CNNs, LSTM, Transformer, GRU, to BERT—the resulting **confusion matrices** across these models are **strikingly similar.** This consistency suggests that the models, regardless of structure, are **capturing similar patterns and limitations** within the lyric data. In particular, it implies that certain genres may **inherently have overlapping lyrical styles or language use**, making them consistently harder to distinguish. However, the ensemble model shows that these models **capture different patterns** and have their own bias.

3.10.1 Accuracy

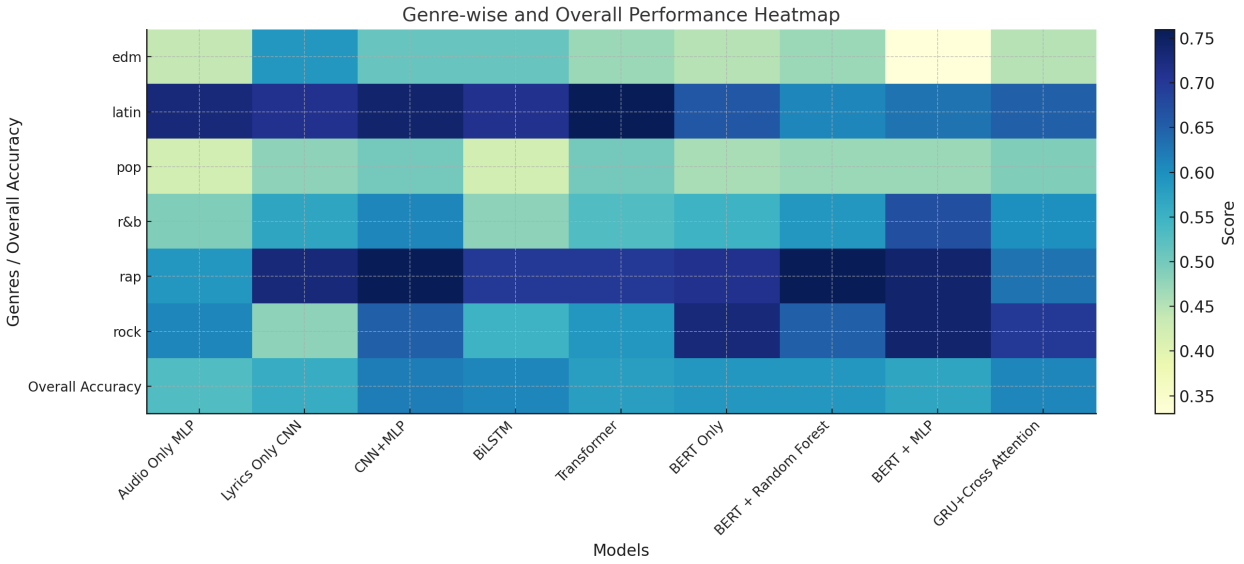

Figure 25: Accuracy Across Genres for All Models

4 Experiments and Other Findings

4.1 Common Classification Errors

Regardless of model,*pop* is the genre most frequently confused with others. We believe this is because *pop* music, by definition, refers to any music that gains widespread popularity, regardless of its underlying style. As a result, pop songs often blend elements from various genres—such as rock, EDM, or R&B—which can lead to overlapping lyrical or musical characteristics. This genre overlapping likely causes models to struggle in drawing clear boundaries, making *pop* a common source of misclassification. We found that for lyrics-only models, *Latin*, *Rap*, and *Rock* achieved the highest F1-scores, which aligns well with our earlier TF-IDF analysis in the EDA—these genres tend to have more distinct and consistent lyrical patterns. When we added audio features into the model, *R&B* showed the most significant improvement. This suggests that while its lyrics may be less distinctive on their own, the audio characteristics of R&B—such as rhythm, instrumentation, and vocal style—provide strong signals for genre classification.

4.2 Unsupervised clustering Analysis

To better understand the underlying structure of the music dataset, we applied unsupervised clustering using K-Means based on audio features extracted from the raw data.

First, we selected key audio attributes — including danceability, energy, loudness, speechiness, instrumentalness, and duration — and standardized them to ensure balanced scaling

across dimensions. We then performed K-Means clustering, automatically determining the optimal number of clusters by selecting the K that maximized the silhouette score, a metric that measures how well-separated the clusters are. From the plot below we can see K=4 is the best K number according to this analysis, whereas we have 6 categories, so the overlapping is bit inevitable. By plotting the clustered songs in the PCA space, we were able to visually assess how well the tracks grouped based on their musical characteristics.

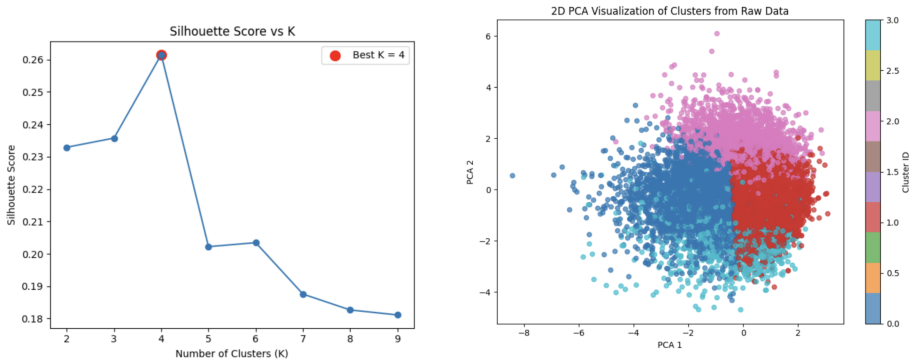

Figure 26: Unsupervised clustering Analysis Result

5 Conclusion and Future Work

5.1 Summary of Findings

In this project, we systematically evaluated a variety of models for music genre classification using lyrics and audio features.

- Our experiments showed that **lyrics-only** and **audio-only** models achieved moderate performance, with accuracies around 0.53 to 0.56. Notably, combining **both modalities** significantly boosted performance, reaching an accuracy of **0.66** in the **Ensemble model**, confirming that lyrics and audio carry complementary signals essential for accurate genre prediction.

- Despite exploring a range of lyric models—from traditional CNNs and RNNs (LSTM, GRU) to advanced Transformers and pre-trained BERT—the confusion matrices remained remarkably consistent, highlighting inherent limitations in lyric-based genre classification. Certain genres, especially Pop, proved harder to distinguish due to their genre-blending nature.

- Analysis of F1 scores revealed that **Latin, Rap, and Rock** consistently performed well in lyric-only models, while **R&B** saw notable improvements once audio features were incorporated, underscoring the unique contributions of musical characteristics beyond lyrics.

5.2 Potential Future Directions

While multi-modal models proved highly effective, several areas for future work are clear:

- **Parameter Efficiency:** Current models come at a cost of significantly increased parameters (e.g., Transformer+MLP with 10M+ and BERT-based models with 109M+). Future work could explore model compression techniques (like pruning, distillation) or lightweight architectures to reduce computational demands.
- **Advanced Fusion Strategies**: Instead of simple feature concatenation, more sophisticated fusion techniques—such as attention-based cross-modal interactions—could better capture the relationship between lyrics and audio.
- **Explainability**: Incorporating explainable AI techniques would allow better understanding of what features the model uses to make decisions, which is particularly important in subjective areas like music.
- **Multi-task learning**: In future work, we propose exploring a multi-task learning approach to address potential feature overlap between music genres. Specifically, we observe that the learned feature space might not naturally align with the six predefined genre categories, leading to confusion during classification. To better structure the latent representations, we suggest applying unsupervised clustering (e.g., K-Means) on the fused audio-lyrics features to discover the underlying natural groupings within the data. The model would then be trained to jointly predict both the unsupervised cluster assignment and the true genre label, encouraging it to organize its internal feature space more meaningfully. By adding this auxiliary clustering prediction task, we hope to regularize the model, improve its generalization, and gain additional insights into how the songs are structured beyond the manual genre labels. This multi-task setup could be especially beneficial in cases where genres are broad or stylistically diverse, allowing the model to capture more fine-grained distinctions between tracks.

References

[1] Nakhaee, M. (2020). *Audio features and lyrics of Spotify songs*. Kaggle. `https://www.kaggle.com/datasets/imuhammad/audio-features-and-lyrics-of-spotify-songs`
https://huggingface.co/google-bert/bert-base-uncased

Appendix

Full Definition of Variables in Raw Data

Variable	Class	Description
track_id	character	Song unique ID
track_name	character	Song Name
track_artist	character	Song Artist
lyrics	character	Lyrics for the song
track_popularity	double	Song Popularity (0-100) where higher is better
track_album_id	character	Album unique ID
track_album_name	character	Song album name
track_album_release_date	character	Date when album released
playlist_name	character	Name of playlist
playlist_id	character	Playlist ID
playlist_genre	character	Playlist genre
playlist_subgenre	character	Playlist subgenre
danceability	double	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
energy	double	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
key	double	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation. E.g. $0 = \mathrm{C}, 1 = \mathrm{C}\sharp/D\flat, 2 = D, and so on. If no key was detected, the value is -1.$
loudness	double	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db.
mode	double	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.

Variable	Class	Description
speechiness	double	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
acousticness	double	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
instrumentalness	double	Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
liveness	double	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
valence	double	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
tempo	double	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
duration_ms	double	Duration of song in milliseconds
language	character	Language of the lyrics