

GAP ANALYSIS OF MUSIC GENRE CLASSIFICATION ML MODULE

1. Limited Dataset Diversity

As the music industry comprises of many different genres and sub-genres each of them clashing and overlapping on each other, If the dataset provided isn't diverse enough it can lead to potential a lot of un-reliable predictions due to the following problems.

Bias in Model Predictions

- **Overrepresentation of Certain Genres:** If certain genres are overrepresented in the dataset, the model may become biased towards them, leading to higher accuracy for those genres and lower accuracy for underrepresented genres. This results in skewed predictions, where the model is less likely to correctly classify rare genres.
- **Cultural or Regional Bias:** A dataset lacking geographic or cultural diversity can cause the model to favor genres popular in one specific region or culture, neglecting others.

2. Poor Generalization

- A model trained on a non-diverse dataset may not generalize well to new, unseen data. For example, if the dataset is heavily dominated by Western music genres but tested on global music, the model may struggle to accurately classify non-Western genres.

3. Overfitting

- If the dataset is homogenous, the model may memorize patterns specific to the limited types of music present, leading to overfitting. This means the model will perform well on the training data but poorly on real-world or diverse datasets.

4. Reduction in Creativity and Innovation

- A narrow dataset might lead to the exclusion of experimental or fusion genres, causing the model to be unable to classify innovative music that blends different styles. This limits the model's usefulness in identifying emerging trends in music.

5. Misrepresentation of Genre Complexity

- Music genres are often fluid and overlap with each other. A diverse dataset that includes sub-genres, fusion genres, and evolving styles ensures that the model can capture these complexities. Without diversity, the model may learn overly simplistic distinctions between genres, reducing its ability to understand nuanced musical differences.

6. Inaccurate Evaluation Metrics

- If the dataset is imbalanced, common evaluation metrics like accuracy might be misleading. For instance, if 80% of the dataset consists of rock music and only 5% jazz, a model could achieve high accuracy simply by predicting rock most of the time, without actually learning to recognize jazz properly.

7. Exclusion of Minority Audiences

- Lack of dataset diversity can result in the exclusion of niche or minority genres, which may be significant for specific cultural or social groups. This can lead to marginalizing certain musical traditions and reinforcing a narrow view of what constitutes popular or important music.

8. Unreliable Recommendations

- In music recommendation systems that rely on genre classification, a lack of diversity in the training dataset could lead to biased or repetitive recommendations. Listeners may receive suggestions for mainstream genres, while missing out on more diverse or unique options.

9. Underperformance in Multilingual or Instrumental Genres

- Genres that include non-English lyrics or unconventional instrumental compositions could be underrepresented, causing the model to struggle in correctly classifying non-mainstream or experimental works.

10. Difficulty with Edge Cases

- Lack of diverse edge cases in the dataset (e.g., songs that blend multiple genres or songs from subgenres) can make it difficult for the model to handle cases that don't fit neatly into well-defined genre categories.



Addressing the issues caused by lack of dataset diversity in a music genre classification problem requires a combination of strategies to improve data representation, model fairness, and overall performance. Here are some effective ways to tackle these problems:

1. Collect a Diverse Dataset

- **Broaden Data Sources:** Include music from various platforms, cultures, languages, and regions to ensure diversity in genres. Make sure the dataset contains songs from both mainstream and niche genres.
- **Incorporate Sub-genres and Fusion Genres:** Collect data that includes sub-genres and genre blends, as these are common in modern music. This will help the model understand genre fluidity and hybridization.
- **Use Multilingual Data:** Include music with lyrics in different languages or instrumental music to avoid language bias.

2. Balance the Dataset

- **Class Balancing:** Use data augmentation techniques, such as oversampling underrepresented genres or undersampling overrepresented ones, to create a more balanced dataset. This will prevent the model from being biased toward certain genres.
- **Synthetic Data Generation:** In cases where certain genres are severely underrepresented, use techniques like data synthesis or style transfer to create new samples in those genres, enhancing the diversity of the dataset.
- **Stratified Sampling:** When collecting data, ensure that samples are drawn in a way that maintains the proportional representation of different genres. This ensures better generalization.

3. Use Domain Knowledge for Genre Representation

- **Leverage Expert Input:** Use musicologists, musicians, or domain experts to define clear genre boundaries or relationships between genres. They can also help with classifying ambiguous tracks or suggesting better ways to categorize fusion genres.
- **Hierarchical Genre Labels:** Instead of flat classification, implement a hierarchical structure of genres (e.g., super-genres like “rock” with sub-genres like “indie rock” or “psychedelic rock”). This helps the model understand the nuances between related genres.

4. Data Preprocessing and Feature Engineering

- **Extract Diverse Features:** Use advanced feature extraction methods to capture more complex and diverse musical elements, such as rhythm, tempo, key signatures, timbre, and harmonic structures, which vary across genres. Avoid relying solely on basic features like tempo or beat.

- **Multi-modal Data:** Use a combination of audio features (spectrograms, chroma features), metadata (artist, region, language), and even user interaction data to better represent different genres.

5. Use Advanced Machine Learning Techniques

- **Transfer Learning:** Apply pre-trained models (e.g., trained on a large, diverse corpus) to leverage knowledge about a wide variety of genres, then fine-tune the model on your dataset. This helps if your dataset is small or lacks diversity.
- **Ensemble Methods:** Use ensemble techniques (combining multiple classifiers) to improve the model's robustness across a variety of genres.
- **Regularization Techniques:** Use regularization methods like dropout and data augmentation to reduce overfitting, especially when working with a diverse dataset that includes many underrepresented genres.

6. Improve Evaluation Metrics

- **Use Class-balanced Metrics:** Evaluate model performance with metrics that account for class imbalance, such as:
 - **F1 Score:** Focuses on both precision and recall, which is especially useful for underrepresented genres.
 - **Confusion Matrix:** Analyze how often the model confuses certain genres and look for patterns in misclassification.
 - **Weighted Accuracy:** Calculate the weighted average of accuracy for each genre to ensure that the overall performance reflects success across all genres, not just the majority ones.
- **Per-class Evaluation:** Break down the evaluation by genre to monitor how the model performs on underrepresented genres and adjust accordingly.

7. Handle Genre Ambiguity and Overlap

- **Multi-label Classification:** Instead of forcing each song into a single genre, use multi-label classification where songs can belong to multiple genres. This is useful for songs that mix styles or fit into multiple categories.
- **Soft Classification:** Assign probabilities or confidence scores to multiple genres rather than making hard classifications. This approach better reflects the fluid boundaries between genres.

8. Improve Representation of Minority Genres

Cultural Partnerships: Collaborate with music platforms or cultural institutions that focus on niche genres or lesser-known musical traditions. This can give access to datasets from underrepresented genres.

- **Crowdsourced Annotations:** Use crowdsourcing or community-driven efforts to label or categorize genres in underrepresented or evolving music styles.

9. Use Fairness-aware Algorithms

- **Fairness Metrics:** Use fairness-aware algorithms to ensure that the model's predictions are not biased toward specific genres. Implement fairness constraints that force the model to treat all genres more equally.
- **Bias Mitigation:** Regularly audit the model's predictions to check for biases. Implement corrective algorithms if certain genres are under-predicted or ignored.

10. Regularly Update the Dataset

- **Dynamic Dataset Updates:** Continuously update the dataset to reflect new musical trends, emerging genres, and shifts in the popularity of existing genres. This prevents the model from becoming outdated.
- **Incorporate Feedback Loops:** Use user feedback and model performance data to iteratively improve the dataset and retrain the model, ensuring better genre representation and classification over time.

2. Over-reliance on Default Hyperparameters

Over-relying on hyperparameters can be especially relevant in the context of a **music genre classification algorithm** in several ways, leading to suboptimal results if not addressed properly. Here's how it manifests:

Ignoring Audio Data Quality and Feature Engineering

- **Limited Focus on Audio Features:** In music genre classification, much of the success of the model relies on extracting relevant and meaningful audio features (e.g., MFCCs, chroma features, tempo, timbre). Over-reliance on hyperparameter tuning might shift focus away from effective feature engineering, which can have a much greater impact on classification performance.
- **Ignoring Spectrogram Quality:** If the quality of audio input data (e.g., spectrogram resolution or sampling rates) is suboptimal, hyperparameter tuning alone won't compensate for this. Feature extraction techniques or data preprocessing steps (such as noise reduction or normalization) might be more impactful.

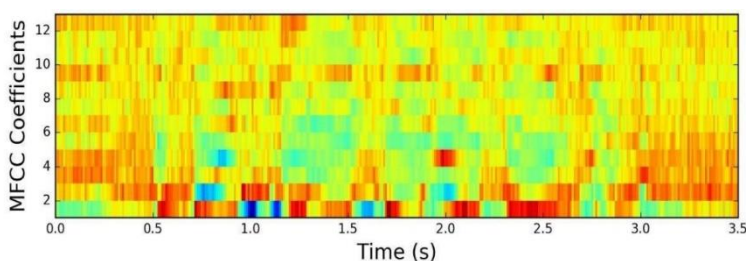
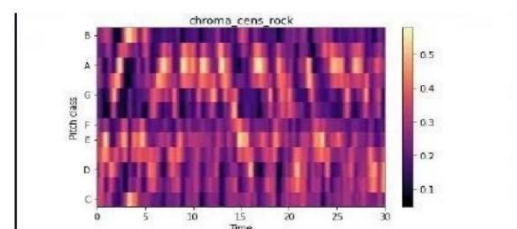


Fig.1 MFCC Coefficients for Rock music



These problems can be addressed by:

- **Improve Data Quality:** Focus on feature engineering and preprocessing (e.g., spectrogram quality, noise reduction, and relevant audio features like MFCCs or chroma features).
- **Handle Genre Imbalance:** Balance the dataset by oversampling underrepresented genres or using data augmentation techniques like pitch shifting or time-stretching.
- **Choose the Right Model Architecture:** Experiment with models better suited to audio data, like CNNs for spectrograms or RNNs for temporal patterns, instead of over-tuning hyperparameters.
- **Use Ensemble Models:** Combine multiple models (e.g., CNNs and RNNs) to capture different aspects of music data instead of fine-tuning one model excessively.
- **Limit Hyperparameter Tuning Cycles:** Avoid overfitting by limiting the number of tuning iterations and using cross-validation with different splits.
- **Focus on Multi-label Classification:** Use multi-label models to handle genre ambiguity and overlapping genres, rather than fine-tuning hyperparameters for rigid classification.
- **Regularly Evaluate Model Performance:** Use class-balanced evaluation metrics (e.g., F1 score, precision/recall) to monitor the model's ability to generalize across different genres.
- **Increase Dataset Diversity:** Ensure the dataset covers various cultures, languages, and genre types to help the model generalize, rather than relying on tuning to fix dataset limitations.
- **Explore Simpler Models:** Try simpler models with fewer hyperparameters to reduce over-complication and achieve comparable results faster.

3. Pre-existing applications/software related to the project:

There are several pre-existing music genre identification applications that utilize audio signal processing and machine learning techniques to automatically classify and identify genres of music. These applications use features such as tempo, rhythm, timbre, harmony, and more to determine the genre of a track. Here are some notable applications:

1. Shazam

Primary Use: Shazam is mainly used for music recognition, identifying specific songs, but it also provides genre-related information about the track.

2. ACRCLOUD

Primary Use: ACRCLOUD is an audio recognition service used by developers for various applications, including genre classification.

3. Spotify (via EchoNest API)

Primary Use: Spotify uses music metadata, including genre classification, for its music catalog and recommendation engine.

4. Deezer (Spleeter)

Primary Use: Deezer, a streaming platform, offers audio separation through Spleeter, but its core platform includes genre-based classification for tracks.

5. Last.fm

Primary Use: Last.fm is a music discovery platform that uses user-generated data to tag songs with genres.

6. AudD Music Recognition

Primary Use: AudD is an API for recognizing songs and their metadata, including genre information.

7. GTZAN Dataset-Based Models:

Essentially an open-source library for audio analysis and music information retrieval, which can be used to build genre classification systems.

3. How the project stands out from the pre-existing application/software:

1. Multi-label Classification with Genre Overlap

- **Handle Genre Fluidity:** Unlike most models that classify songs into a single genre, the project could allow songs to be labeled with multiple genres, reflecting how modern songs often blend styles (e.g., a song being both "pop" and "indie rock").
- **Dynamic Genre Detection:** The project could identify genre transitions within a song, labeling different parts with varying genres (e.g., rock verses and jazz-like choruses).

2. Real-time Genre Detection

- **Real-time Classification:** The project could classify music genres in real-time as the song plays, offering streaming services or users live genre tagging as they listen.
- **Interactive Genre Prediction:** It could allow users to interact with the model while the song is playing, offering feedback that helps refine its predictions for personalized genre detection.

3. Focus on Sub-genres and Niche Genres

- **Support for Sub-genres:** Many existing models focus on broad genre categories. The project could incorporate niche sub-genres and emerging styles (e.g., "lo-fi hip-hop," "vaporwave") to cater to more specialized tastes.
- **Cultural and Regional Genre Inclusion:** The project could integrate lesser-known global genres (e.g., "Afrobeat," "Qawwali," "K-pop") to capture the full diversity of world music, providing a richer experience for a broader audience.

4. Personalized Genre Classification

- **User-Centric Model:** The project could personalize genre classification based on a user's listening habits and preferences. For example, genre boundaries could be adapted, or the model could learn custom genres based on user feedback.
- **Context-Aware Classification:** The model could classify genres not just based on audio features but also on context, such as listening time of day, user mood, or activity, providing more meaningful recommendations.

Conclusion:

By understanding all of the different gaps and the literature related to the project, it allows to perform better and have a deep understanding of what is required and how the project should be proceed. By keeping the different pre-existing softwares/applications related to our project while also understanding where they are lacking gives us a clear look of what to focus and address on to achieve this project.

