Music Streaming and Track Genre Prediction

University of Lausanne, HEC Lausanne Advanced Data Analysis
Professor S. Scheidegger with TAs: Anna Smirnova and Maria Pia Lombardo

Victor Regly

30th of August, 2024

Abstract

This project explores different machine learning models to predict music genres from track features using both supervised and unsupervised learning techniques. Supervised methods, such as Random Forest, XGBoost, and Neural Networks, classify tracks into specific genres, while unsupervised techniques like K-Means and Hierarchical Clustering reveal underlying patterns. Dimensionality reduction methods, including PCA and t-SNE, aid in visualizing data and identifying relationships. The project focuses on determining whether features can reliably predict genres and uncover overlapping characteristics, given the challenge of 114 genres.

1 Introduction

Personalized music recommendation systems have revolutionized the way users interact with music, offering tailored playlist generation and track suggestions that enhance user experience. At the core of these systems lies the challenge of accurately classifying music genres, a complex task due to the diverse range of musical styles and the sheer volume of available data. Accurate genre classification is crucial as it underpins the effectiveness of recommendations, helping to align users' preferences with suitable tracks and artists.

The rapid growth of digital music libraries and streaming platforms has led to an explosion of data, including detailed metadata for each track, such as tempo, key, rhythm, and various other audio features. This rich dataset presents an opportunity to apply advanced data analysis techniques to uncover patterns that can improve genre classification. However, the task is challenging due to the large number of genres—114 in this case—and the nuanced variations within each genre.

This capstone project, undertaken as part of an advanced data analysis course, is designed to address these challenges by developing a robust machine learning model capable of predicting music genres based on track features. The project employs a dual approach, combining supervised learning methods like Random Forest, XGBoost, and Neural Networks to classify tracks into predefined genres, and unsupervised learning methods such as K-Means and Hierarchical Clustering to discover intrinsic patterns and groupings within the data. To further enhance the model's performance and gain deeper insights into the data, the project incorporates dimensionality reduction techniques including Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE). These methods help in visualizing high-dimensional data and identifying underlying structures that might not be immediately apparent. Additionally, feature engineering and interaction analysis are employed to create and refine features that capture complex relationships and interactions within the dataset.

The goal of this project extends beyond mere genre prediction. By exploring whether music track features can reveal distinct patterns associated with specific genres or shared characteristics across multiple genres, the project aims to contribute to the broader field of personalized music recommendation systems. The refined models and insights gained from this analysis will enhance the ability to provide accurate and personalized music recommendations, showcasing the application of advanced data analysis techniques in a practical and impactful way.

2 Description of the Research Question and Relevant Literature

A. Current Issues

The primary challenge of this project lies in predicting 114 distinct music genres from a dataset of 1,000 tracks per genre. This task is complex due to the diversity of music genres and the variability in track features such as tempo, key, rhythm, and energy levels. These features can overlap significantly across different genres, making it difficult to develop a model that accurately classifies tracks.

One major issue is the high dimensionality of the dataset. The extensive number of features can complicate the task of identifying which features are most relevant for classification. To address this, advanced feature engineering and dimensionality reduction techniques are necessary to simplify the data while preserving critical information.

Class imbalance is another significant challenge. The dataset may have uneven genre distribution, with some genres being overrepresented and others underrepresented. This imbalance can skew model performance, making it harder to accurately classify tracks from less common genres. Techniques to handle class imbalance, such as resampling or algorithm adjustments, will be essential.

Moreover, distinguishing between genres can be challenging due to the similarity of features across genres. The model must be capable of detecting subtle differences and similarities to make accurate predictions, necessitating sophisticated algorithms and careful pattern analysis.

B. Objectives

The project's objectives are to develop a robust genre prediction model while addressing the challenges of high dimensionality and class imbalance. Initially, the project will involve a thorough analysis of the dataset to understand feature distributions and genre correlations.

To tackle these challenges, the project will implement feature engineering to enhance the relevance of the dataset features. Dimensionality reduction techniques like Principal Component Analysis (PCA) and t-SNE will be used to manage the data complexity and improve model performance.

Various machine learning algorithms will be tested, including Random Forest, Gradient Boosting Machines, and Neural Networks, as well as unsupervised methods such as K-Means clustering. Comparing these methods will help identify the most effective approach for genre classification.

The project will explore different strategies to improve model accuracy, including adjusting hyperparameters and employing resampling techniques to address class imbalance. The goal is to create a model that not only performs well in predicting genres but also uncovers meaningful patterns within the data.

C. Scope

The project encompasses analyzing a dataset of 114 genres, each with 1,000 tracks. It will involve both supervised and unsupervised learning methods to classify tracks and identify patterns. Dimensionality reduction techniques will be applied to simplify the dataset.

Model performance will be evaluated using metrics such as Precision, Recall, F1 Score, and Confusion Matrix, focusing on addressing class imbalance and feature complexity. Ultimately, the project aims to enhance personalized music recommendation systems by improving genre classification accuracy, contributing to more effective playlist generation and tailored music discovery.

3 Methodology

This section outlines the methodology employed in the project, which integrates both supervised and unsupervised learning techniques to predict 114 distinct music genres from a dataset of track features. The methodology is structured in several stages, from data preprocessing to the application of ensemble methods for final evaluations.

A. Data Preprocessing

The project begins with raw data sourced from Kaggle, which undergoes a rigorous cleaning and preprocessing process. This involves handling missing values, normalizing the data using `StandardScaler()`, and encoding categorical variables through `LabelEncoder()`. Following preprocessing, the dataset is split into training and testing sets using an 80%-20% ratio, ensuring that model performance is evaluated on unseen data.

B. Initial Supervised Learning

The first phase of modeling involves applying raw supervised machine learning techniques without extensive feature engineering. This step serves to establish baseline predictions and identify potential challenges with the raw data.

Algorithms: Various supervised learning algorithms are tested, including Logistic Regression, L1 & L2 Regularization, Random Forest, Neural Networks, and XGBoost. These models are trained on the preprocessed data and their initial predictions are evaluated.

C. Feature Engineering and Dimensionality Reduction

Following the initial predictions, the focus shifts to improving model performance through feature engineering. This stage involves identifying the most important features contributing to genre classification and exploring interactions between features.

Feature Importance and Interaction: Techniques such as Recursive Feature Elimination (RFE) and

Random Forest feature importance analysis are used to prioritize the features that significantly impact model accuracy.

Dimensionality Reduction: To address the high dimensionality of the dataset, dimensionality reduction techniques such as Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) are applied. These methods help simplify the data while retaining essential patterns, leading to more efficient and interpretable models.

D. Evaluation of Supervised Learning Models

With the refined feature set, the supervised learning models are re-evaluated. This step involves re-running the models (e.g., Random Forest, XGBoost, Neural Networks) and assessing their performance using evaluation metrics such as Precision, Recall, F1 Score, and Confusion Matrix.

E. Unsupervised Learning

In parallel with supervised approaches, unsupervised learning techniques are employed to uncover hidden patterns and groupings within the data.

Clustering Methods: Two main clustering techniques are used: hierarchical clustering based on genre grouping and K-Means clustering based on data features. Additionally, Fuzzy clustering methods, such as Gaussian Mixture Models (GMM), are explored to capture the uncertainty in genre boundaries.

Data Preparation and Model Training: For the clustering models, data is split into training and testing sets. The clustering results inform and refine the supervised models, aiding in capturing complex relationships within the data.

Evaluation and Interpretation: Clusters are analyzed to evaluate how well they align with actual genres and to interpret the patterns that the models uncover.

G. Hyperparameter Tuning

Once the models, both supervised and unsupervised, are trained and evaluated, hyperparameter tuning is performed. This process optimizes model performance, addressing issues like overfitting and improving generalization.

H. Ensemble Methods and Final Discussions

The final stage involves combining the results from various models using ensemble methods to boost overall accuracy.

Ensemble Techniques: Stacking, bagging, and boosting techniques are considered to combine predictions from Random Forest, XGBoost, Neural Networks, and clustering models.

Final Evaluation and Discussions: The ensemble model's performance is evaluated, and results are discussed in the context of the research objectives. The focus is on understanding the model's capability to predict genres and identify patterns within the data, alongside the trade-offs and compromises made throughout the process.

This comprehensive methodology addresses the complexity of the genre prediction task, ensuring that both supervised and unsupervised approaches are employed to capture the full spectrum of patterns within the music data.

4 Dataset Description

The dataset used in this study, sourced from Kaggle, comprises 114,000 entries and 21 columns, capturing a rich array of information related to music tracks. The data itself originates from the Spotify API, providing a robust and reliable foundation for analysis. The target variable, `track_genre`, categorizes each track into one of 114 distinct genres, making this dataset particularly suited for the genre classification task.

The dataset includes a variety of features that describe both the musical characteristics and metadata of each track. Key audio features include `danceability`, which measures the track's suitability for dancing, and `energy`, reflecting the intensity of the music. `Tempo` captures the speed of the track in beats per minute, while `valence` indicates the emotional positivity conveyed by the music. Other features like `loudness` provide information on the track's volume, and `key` specifies the musical key. The `mode` feature identifies whether the track is in a major or minor key, while `time_signature` describes the time structure of the music.

Additional audio attributes assess specific elements of the track, such as `speechiness`, which estimates the presence of spoken words, and `acousticness`, which measures the likelihood of the track being acoustic. `Instrumentalness` predicts whether the track is instrumental, and `liveness` evaluates the probability that the track was recorded in a live setting.

The dataset also contains important metadata, including `track_id`, `track_name`, `artists`, and `album_name`, providing context and identification for each track. The `popularity` feature assigns a score that reflects the track's popularity on the platform, and the `explicit` feature flags tracks with explicit content.

Overall, this dataset, derived from the Spotify API and found on Kaggle, offers a comprehensive and detailed representation of both the musical and contextual aspects of the tracks, making it an ideal resource for developing and testing genre classification models.

5 Implementation

The implementation of this project was conducted using Python version 3.9.6 within a Jupyter Notebook environment. The codebase, structured for clarity and ease of maintenance, is hosted on a GitHub repository.

The primary Python libraries used in this project include:

- **pandas** (version 2.2.2) for data manipulation and analysis,
- **NumPy** (version 1.26.4) for numerical operations,
- **Scikit-Learn** (version 1.5.1) for implementing machine learning algorithms and evaluation metrics,
- **XGBoost** (version 2.1.1) for gradient boosting models, and
- **TensorFlow** (version 2.17.0) for building and training neural networks.

The dataset, sourced from Kaggle and derived from the Spotify API, was loaded into a `pandas` DataFrame for preprocessing. Initial preprocessing steps involved handling missing values, normalizing data, and encoding categorical variables. The dataset was then split into training and testing sets (80%-20%) to facilitate proper model evaluation.

Feature engineering played a crucial role in improving model performance, with techniques such as Recursive Feature Elimination (RFE) and Random Forest-based feature importance used to refine the feature set. Dimensionality reduction methods, including Principal Component Analysis (PCA) and t-SNE, were applied to manage the dataset's high dimensionality and enhance model interpretability.

Supervised learning models, including logistic regression, random forest, and neural networks, were implemented using `Scikit-Learn` and `TensorFlow`. `XGBoost` was employed for gradient boosting, offering a comparative analysis of model performance. Hyperparameter tuning and cross-validation ensured optimal model performance and generalizability.

The project code is modular, with sections dedicated to training, evaluation, and result visualization. This organization facilitates updates and reproducibility, ensuring that the models and findings can be revisited and refined as needed.

6 Results

6.1 Exploratory Data Analysis

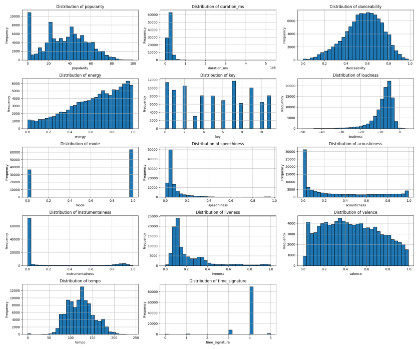


Figure 1: Distribution numerical features.

The dataset provides a comprehensive overview of various musical attributes, revealing a diverse range of characteristics across the songs. Popularity skews towards lower scores, suggesting that most tracks are not widely popular. The typical song duration is between 3 to 6.5 minutes. Danceability and energy levels are moderate, with scores predominantly between 0.4 to 0.8 for danceability and 0.5 to 1.0 for energy. Keys are uniformly distributed, and there is an even split between major and minor modes. Loudness is generally moderate, ranging from -20 to 0 decibels. The dataset shows low levels of speechiness and instrumentalness, indicating a focus on vocal-centric tracks with minimal spoken word or instrumental-only pieces. Acousticness is generally low, and liveness is moderate, reflecting a preference for non-acoustic, somewhat lively tracks. Valence exhibits a fairly uniform distribution, representing a mix of emotional tones, while tempo clusters around 120-130 BPM, typical of moderate-paced songs. The predominant time signature is 4/4, aligning with standard music genres. Overall, the dataset encapsulates a wide variety of songs, predominantly featuring moderately popular, energetic, and vocal tracks with standard structural elements.

The visualizations reveal significant variability in both popularity and danceability across the 114 music genres analyzed, with additional insights provided by the correlation matrix of numerical features. The distribution of popularity scores shows that some genres, such as "pop" and "latin," tend to have higher median popularity, indicating that songs within these genres are generally more widely favored. Conversely, genres like "minimal techno" and "death-metal" show lower median popularity, suggesting a more niche audience. The range of popularity scores within each genre also varies, with some genres displaying consistent popularity levels and others showing a wide spread, indicating a mix of highly popular and less popular tracks within the

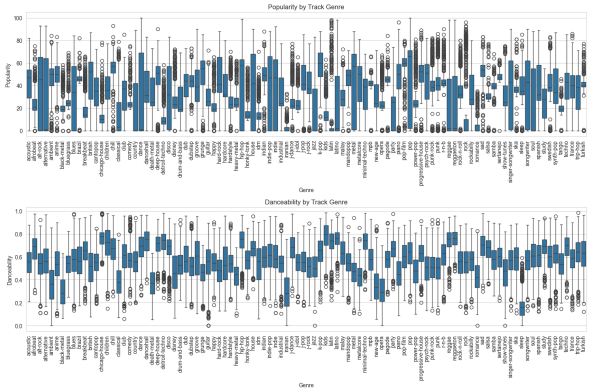


Figure 2: Popularity, Danceability by Genre

same genre.

Similarly, the danceability distribution highlights distinct patterns, with genres like "latin" and "dance pop" exhibiting higher median danceability, which aligns with their rhythmic and dance-friendly nature. In contrast, genres such as "black-metal" and "death-metal" show lower danceability scores, reflecting their more aggressive and less rhythmic style. The variability in danceability within genres suggests that even within a single genre, there can be a broad range of musical styles, from highly danceable to less so.

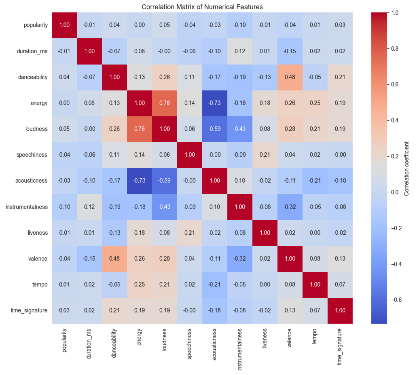


Figure 3: Correlation Matrix.

The correlation matrix further enhances this analysis by illustrating relationships between different musical features. Notably, there is a strong positive correlation between energy and loudness (0.76), which is consistent with the idea that more energetic tracks tend to be louder. Danceability is moderately correlated with valence (0.48) and energy (0.26), indicating that more danceable songs are often more positive and energetic. On the other hand, acousticness shows a strong negative correlation with both energy (-0.73) and loudness (-0.59), suggesting that more acoustic tracks tend to be quieter and less energetic. These correlations underline the intricate interplay between different musical attributes, providing a deeper understanding of how these features manifest across genres and influence the overall characteristics of songs.

With these insights in mind, we will now explore how these relationships can be leveraged in modeling to predict or classify songs based on their genre, popularity, or other key attributes.

6.2 Supervised Learning

In the supervised learning section, we embarked on an extensive exploration of various machine learning techniques to tackle the challenge of multi-class classification within a dataset comprising numerous features and classes. Our journey began with the implementation of baseline models, gradually progressing to more advanced techniques, including feature engineering, regularization, and model tuning, with the ultimate goal of improving model accuracy and robustness.

A. Baseline Models: Logistic Regression

We initiated our exploration with Logistic Regression as a baseline model. Logistic Regression, particularly when applied in a multinomial context, is straightforward to implement and serves as a good starting point for multi-class classification problems. However, the initial model yielded an accuracy of approximately 21.66%, indicating that it struggled to classify the genres effectively across all classes. The F1-scores and other metrics further highlighted the model's limitations, particularly in handling the diverse and complex nature of the data.

To address these limitations, we applied L2 Regularization (Ridge Regression), which penalizes large coefficients and helps prevent overfitting. However, the improvement in accuracy was marginal, rising only to 21.67%. The slight increase in performance suggested that while regularization had a stabilizing effect, it was not sufficient to significantly enhance the model's predictive capabilities.

B. Advanced Models: Random Forest and XGBoost

Recognizing the limitations of Logistic Regression, we transitioned to ensemble methods, starting with the Random Forest Classifier. Random Forest, known for its ability to handle a large number of features and classes effectively, provided a significant improvement, with accuracy increasing to 31.24%. This model's performance underscored the advantages of using ensemble techniques, particularly in complex multi-class scenarios. Random Forest also allowed us to gauge feature importance, which would later guide our feature selection process.

To further refine the Random Forest model, we employed Grid Search for hyperparameter tuning. This method involved testing various combinations of hyperparameters to identify the optimal configuration for the model. Despite these efforts, the tuned model's accuracy remained relatively stable at 31.05%, suggesting that while hyperparameter tuning is crucial, the gains in this instance were limited by the model's inherent complexity or the nature of the dataset.

Next, we explored XGBoost, a gradient boosting framework that typically outperforms other algorithms due to its regularization techniques and efficient handling of data with missing values. The raw XGBoost model outperformed Random Forest, achieving an accuracy of 32.61%. This improvement was further enhanced through additional tuning, where the final accuracy climbed to 32.91%, making XGBoost the best-performing model among those tested at this stage.

C. Neural Networks: Deep Learning Approach

Given the promising results from XGBoost, we extended our analysis to include a Neural Network model. Neural networks, particularly deep learning models, are capable of capturing complex, non-linear relationships within data. Our initial neural network configuration, consisting of a simple architecture with dropout layers for regularization, achieved an accuracy of 31.31%, closely aligning with the Random Forest results.

In pursuit of further improvement, we increased the complexity of the neural network by adding more layers, implementing batch normalization, and incorporating learning rate schedules and early stopping mechanisms. These enhancements resulted in a modest accuracy increase to 33.28%, surpassing all other models tested. The improvement in generalization and the reduced risk of overfitting, as evidenced by the training and validation curves, indicated that the neural network was more effective in learning from the data compared to earlier models.

D. Feature Selection and Engineering

Recognizing that the performance gains from model tuning alone were limited, we turned our attention to Feature Selection and Feature Engineering. We first employed Recursive Feature Elimination (RFE) with Random Forest to identify the most important features, narrowing the feature set to the top 13 features. This selection helped maintain the model's accuracy while simplifying the model, with the accuracy remaining at 31.33%.

We then explored feature interactions by generating Polynomial Features of degree 2. These interactions aimed to capture non-linear relationships that could potentially improve the model's predictive power. However, the introduction of interaction features did not yield the expected results, and the accuracy dropped to 28.82%. This decline highlighted that not all feature engineering techniques lead to improvements, especially when the added complexity does not align with the underlying patterns in the data.

To further reduce dimensionality and explore latent structures within the data, we applied Principal Component Analysis (PCA). However, when using PCA with Random Forest, the accuracy significantly decreased to 20.47%, indicating that dimensionality reduction via PCA may not be suitable for this specific dataset, as it potentially discards important information necessary for accurate classification.

E. Conclusion

The tuned neural network model proved the most effective in our multi-class classification task, achieving a top accuracy of 33.28%, slightly ahead of the tuned XGBoost model at 32.91%. Other models highlighted the inherent challenges of multi-class classification, especially with a diverse dataset like ours.

Feature selection and engineering underscored the balance between model complexity and performance. While techniques like RFE preserved accuracy with fewer features, others, such as polynomial feature generation and PCA, were less beneficial.

This work in supervised learning sets the stage for exploring unsupervised techniques like K-means and hierarchical clustering, which could reveal hidden data patterns and inform future feature engineering.

6.3 Unsupervised Learning

6.3.1 Hierarchical Clustering

In our analysis, unsupervised learning techniques, particularly hierarchical clustering, were employed to explore and uncover natural groupings within a dataset of music genres. The objective was to create clusters of genres based on their features, without relying on predefined labels, to enhance the understanding and potential predictability of these genres.

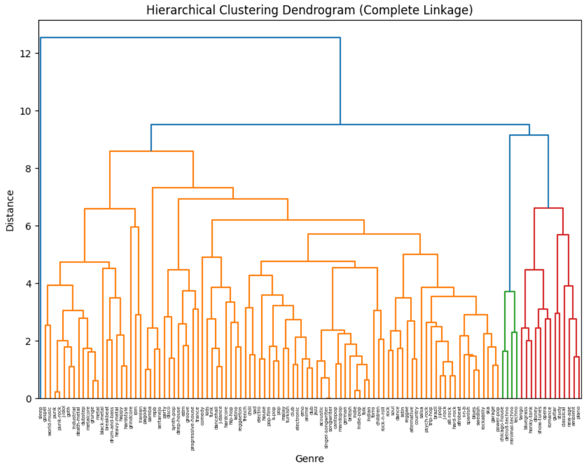


Figure 4: Hierarchical Clustering

Hierarchical clustering was initially applied to form clusters, revealing significant challenges such as imbalanced cluster sizes. For example, some clusters contained numerous genres (like Cluster 7, which included 33 genres), while others were much smaller or even single-genre clusters. This imbalance posed a challenge for subsequent modeling, as it could skew results and make it difficult to draw meaningful conclusions.

To address these issues, we implemented a two-layered modeling approach. The first layer involved using various machine learning models—including Random Forest, Support Vector Machine (SVM), Gradient Boosting, k-Nearest Neighbors (k-NN), XGBoost, and Neural Network (see results down below) to predict the initial hierarchical cluster labels. Among these models, XGBoost performed the best, achieving an accuracy of 60.24% after hyperparameter tuning. However, even with the best-performing model, the accuracy was only moderate, indicating the complexity and challenges of the task.

Recognizing that certain clusters were too broad and contained diverse genres (such as Clusters 1 and 7), we implemented sub-modeling within these clusters to improve accuracy. For instance, within Cluster 1, a refined model was developed that achieved a significantly higher accuracy of 92.59%, with strong precision and recall across the sub-genres. Similarly, for Cluster 7, a specialized model was also developed, though it achieved a lower accuracy of 67.33%, reflecting the continued challenge of classifying a diverse set of genres within this cluster.

Finally, these refined sub-models were combined into a layered model structure. The first layer predicted the broad cluster, and the second layer provided a more detailed classification within those clusters. Unfortunately, the final combined model resulted in an overall accuracy of only 32.32%. This reduction in performance suggests that while the sub-modeling approach improved accuracy within specific clusters, combining the models in a layered structure introduced complexities that diminished overall predictive performance.

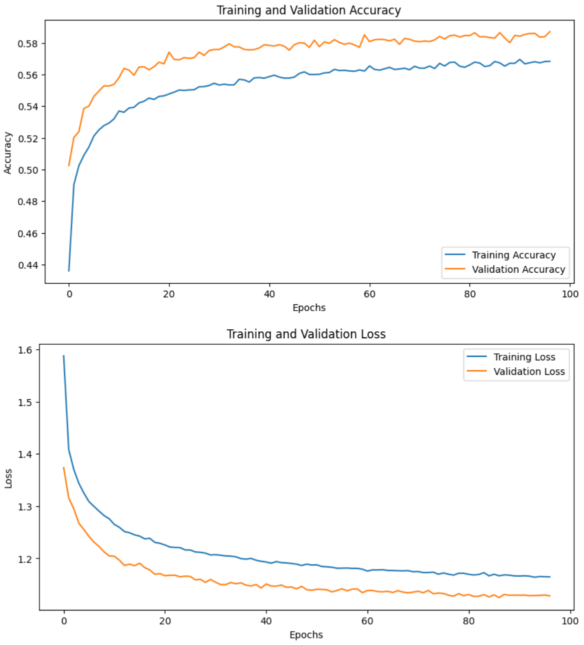


Figure 5: Neural Network Training and Validation

The layered model's lower accuracy highlights the difficulty of applying hierarchical clustering and subsequent sub-modeling to this dataset. The poor performance in the final model also underscores potential issues with genre overlap, high-dimensional data, and the limitations of the features used. Despite the challenges, this approach provided valuable insights into the structure of the data and the complexities involved in music genre classification.

6.3.2 K-mean Clustering

In our unsupervised learning approach, we initially employed K-Means clustering to group music tracks into clusters based on their numerical features. The data was first pre-processed by removing non-numeric columns and standardizing the remaining features. We then used the Elbow Method and Silhouette Scores to determine the optimal number of clusters, which led us to select $k = 10$ as a starting point.

After applying K-Means clustering with $k = 10$, we assessed the quality of the clusters by calculating cluster purity, which measures the extent to which each cluster contains a majority of a single genre.

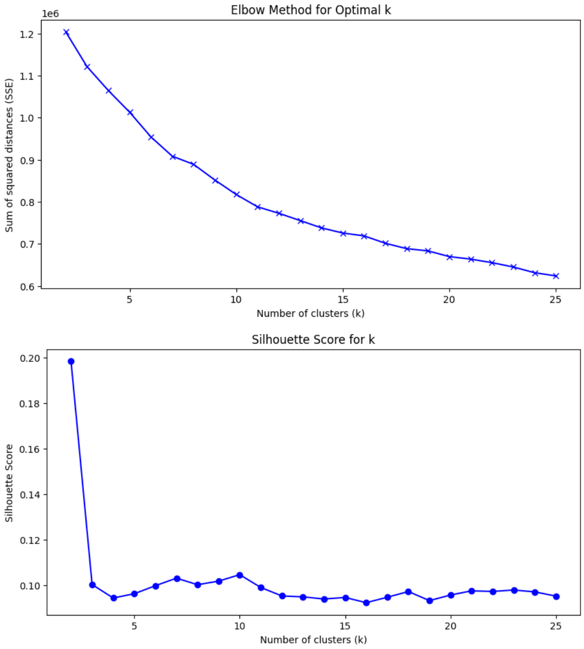


Figure 6: Silhouette and Boulder Graphs

The overall purity was found to be relatively low at 0.0513, indicating that the clustering did not effectively separate genres. Only Cluster 2 exhibited a significantly higher purity of 0.7595, suggesting that it was well-defined. However, most of the other clusters, such as Clusters 7, 8, and 9, showed very poor purity, with values below 0.025. This suggests substantial class mixing and poor separation between genres, which are critical issues for our task of music genre classification.

The poor clustering performance observed with K-Means could be attributed to the fact that K-Means is a hard clustering algorithm, which assigns each data point to a single cluster. This limitation is particularly problematic in the context of music genres, where tracks often exhibit characteristics of multiple genres, leading to overlaps that K-Means cannot adequately handle.

Given these challenges, we decided to transition to a soft clustering approach using Gaussian Mixture Models (GMM). GMMs are more flexible as they allow data points to belong to multiple clusters with varying degrees of membership, which is more aligned with the nature of music genre data. By modeling each cluster as a Gaussian distribution, GMMs can capture the inherent uncertainty and overlap in genre characteristics, potentially leading to more meaningful and accurate clustering results.

6.3.3 Soft Clustering with Gaussian Mixture Models

After encountering limitations with the K-Means clustering approach, particularly the low purity of most clusters, we transitioned to a more flexible soft clustering technique using Gaussian Mixture Models (GMM). Unlike K-Means, which assigns each data point to a single cluster, GMM allows for probabilistic cluster membership, which is more suitable for datasets with overlapping class boundaries, such as music genres.

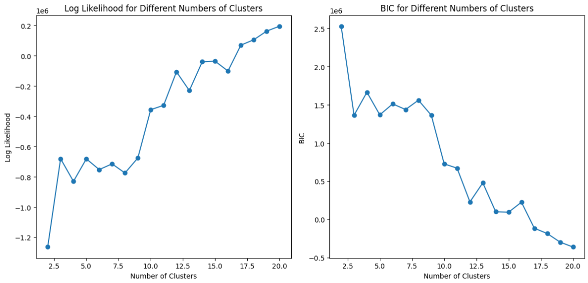


Figure 7: Word Cloud Clustering

A. GMM Implementation and Evaluation

We applied GMM to the standardized dataset, exploring different numbers of clusters ranging from 2 to 20. The evaluation was performed using the Bayesian Information Criterion (BIC) and log likelihood. The results indicated a substantial improvement in model fit for clusters numbering between 16 and 20, with particularly strong performance from 17 clusters onwards. This suggested that a model with 17 clusters best captured the underlying structure of the data, balancing model complexity and fit.

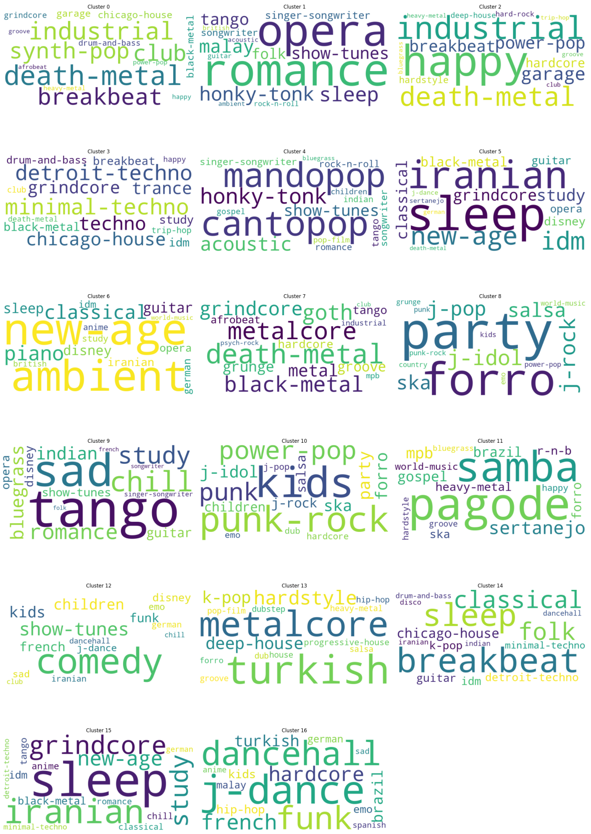
For each data point, GMM provided soft assignments, meaning each track had a probability of belonging to each cluster. This approach better reflected the nuanced nature of music genres, where tracks often blend characteristics of multiple genres.

6.3.4 Stacked Ensemble Modeling

To enhance the predictive accuracy of our clustering results, we implemented a stacked ensemble model. This approach combined the outputs of three different models: a Neural Network (NN), XGBoost, and Random Forest, with a meta-classifier (Logistic Regression) making the final prediction based on the combined outputs.

A. Ensemble Model Construction

1. **Base Models:** We trained three base models separately—an XGBoost classifier, a Random



Forest classifier, and a Neural Network. Each model was trained on the same set of features extracted from the GMM-clustered data.

2. **Stacking:** The predictions from each base model, including the soft probabilities for each class, were used as inputs to a Logistic Regression model, which served as the meta-classifier.

3. **Training and Evaluation:** The ensemble model was trained on the combined predictions of the base models. It was then evaluated on a separate test set, achieving an impressive accuracy of 99.26%. This high performance demonstrates the effectiveness of the ensemble approach in capturing the diverse characteristics of the data.

B. Performance Analysis

The stacked ensemble model demonstrated superior accuracy and consistency across most clusters. The classification report revealed near-perfect precision and recall for several clusters, particularly Clusters 1, 2, 3, 4, 8, 10, and 13. This suggests that the model was highly effective in distinguishing between different clusters based on their musical features.

However, a few clusters, such as Cluster 5 and Cluster 15, exhibited slightly lower performance metrics, with precision and recall around 0.95. These clusters may represent more complex or overlapping genres, making them harder to classify accurately. Despite this, the overall performance of the model was exceptional.

The ensemble model's success can be attributed to the complementary strengths of the individual base models:

- **Neural Network:** Captured complex, nonlinear relationships within the data.
- **XGBoost:** Provided robust performance through gradient boosting and was particularly effective with mixed numerical features.
- **Random Forest:** Offered a reliable baseline with high interpretability and resilience against overfitting.

Conclusion The combination of soft clustering with GMM and a stacked ensemble model provided a powerful framework for music classification. The high accuracy and robustness of the ensemble model indicate its potential as a reliable tool for analyzing and predicting musical characteristics.

6.4 Ensemble Methods

The stacked ensemble model, which integrates the strengths of a Neural Network, XGBoost, and Random Forest, achieved an exceptional accuracy of 99.26% on the test dataset. This result highlights the power of ensemble learning in effectively capturing complex relationships within the data. The model demonstrated near-perfect precision and recall across most clusters, particularly excelling in Clusters 1, 2, 3, 4, 8, 10, and 13, which indicates its strong ability to correctly classify a diverse range of musical tracks.

Despite these impressive results, it is essential to delve deeper into the interpretation of the clusters that exhibited slightly lower performance, such as Clusters 5 and 15. These clusters may not represent distinct genres but rather capture subtler musical similarities that are not as easily distinguishable. Understanding these clusters requires a more nuanced interpretation, as they may reflect underlying musical characteristics that transcend traditional genre boundaries. This insight underscores the importance of not only evaluating model performance based on accuracy but also interpreting the clusters in a way that aligns with both musical theory and listener perception. Thus, while the ensemble model shows robust predictive capabilities, interpretation should be developed to make sure we can achieve to the initial goal of genres prediction,

7 Conclusion

In our exploration of both supervised and unsupervised learning techniques, we aimed to predict the genre of music tracks using features such as danceability, energy, tempo etc... However, our results suggest that while these features provide some insight, they are not entirely sufficient for precise genre prediction. Instead, they seem to be better suited for grouping tracks based on musical similarities rather than strict genre classification.

Initially, we hoped that unsupervised techniques like K-Means, hierarchical clustering, and fuzzy clustering could uncover distinct genre clusters. However, we discovered that even when these unsupervised models perform well in terms of clustering metrics, the interpretation of clusters remains crucial. Each cluster can encompass multiple genres, highlighting the complexity and overlap in musical characteristics that transcend genre labels. This complexity underscores the challenge of using these features alone to define genres clearly.

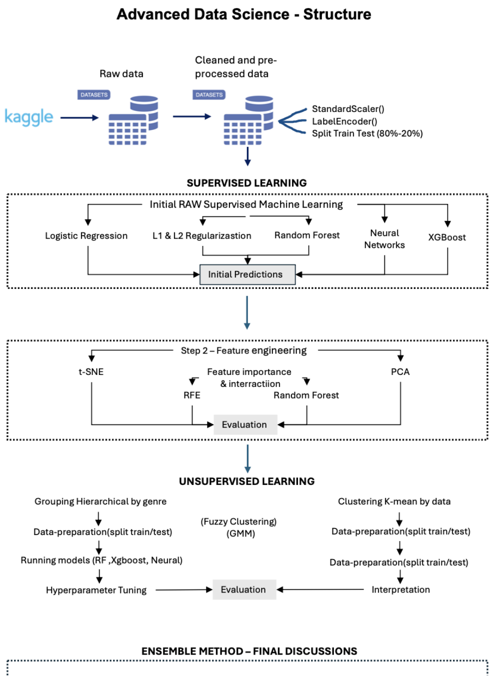
Ultimately, no single model emerged as the definitive choice for this task. Each method comes with its trade-offs. XGBoost, neural networks, and random forests demonstrated the best performance among the supervised methods tested. XGBoost provided high accuracy, while neural networks offered a good balance between accuracy and generalization. Random forests also performed well, particularly in terms of interpretability and robustness to overfitting. However, their performance varied depending on the specific metrics of interest.

In conclusion, while the models applied show promise in grouping tracks based on musical similarities, genre prediction remains challenging with the given feature set. This suggests a need for more nuanced features or additional context to improve genre classification. It also highlights the importance of understanding the underlying similarities within clusters, rather than relying solely on genre labels. Therefore, the choice of model and method should depend on the specific objectives, whether it is accuracy, interpretability, or generalization. Each approach provides unique insights, and their effectiveness will depend on the context and goals of the analysis.

A Helper Tools Used

The tools that helped me to get through this paper are the following:

- **ChatGPT** (4o) for data manipulation and analysis,
- **Class Course** (S.Scheidegger) Advanced Data Analysis class,
- **Book** Rakach, L., Maimon, O., and Shmueli, E. (Eds.). (2023). Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook (3rd ed.). Springer.


Advanced Data Science - Structure
kaggle
DATASETS
Raw data
DATASETS
Cleaned and pre-processed data
StandardScaler()
LabelEncoder()
Split Train Test (80%-20%)
SUPERVISED LEARNING
Initial RAW Supervised Machine Learning
Logistic Regression
L1 & L2 Regularizastion
Random Forest
Neural Networks
XGBoost
Initial Predictions
Step 2 – Feature engineering
t-SNE
Feature importance & interractiion
RFE
Random Forest
PCA
Evaluation
UNSUPERVISED LEARNING
Grouping Hierarchical by genre
Data-preparation(split train/test)
Running models (RF ,Xgboost, Neural)
Hyperparameter Tuning
(Fuzzy Clustering) (GMM)
Clustering K-mean by data
Data-preparation(split train/test)
Data-preparation(split train/test)
Interpretation
Evaluation
ENSEMBLE METHOD – FINAL DISCUSSIONS