Music Genre Classification Using Machine Learning

Lorenzo Vittori

2023/2024

Data Mining Machine Learning Project

1 Introduction

The objective of this project is to develop a machine learning model capable of classifying the genre of a song based on various musical and audio features. Leveraging data mining techniques and machine learning models, the project aims to automatically predict the genre of a song using features derived from a music dataset. This automatic genre classification has a wide range of applications, such as music recommendation systems, content-based search engines, smart shuffle for music app, and music recognition apps.

1.1 Project Goal

Using machine learning techniques, we aim to train predictive models that can accurately classify the genre of a song based on features such as key, energy, rhythm, popularity, and other audio metrics.

1.2 Problem Statement

The problem tackled in this project falls under the category of supervised classification, where the goal is to assign each song to a predefined genre class (e.g., Pop, Rock, Jazz, Classical, etc.). The primary challenge lies in the complexity of the relationships between musical features and genres. Different genres may share similar characteristics, making it difficult to clearly distinguish them.

1.3 Methodology

The project involves the following key steps:

- **Dataset Analysis**: An in-depth analysis of the dataset is performed to understand the distribution of variables and to identify any missing data or anomalies.
- **Data Preprocessing**: Data cleaning and transformation are applied to prepare the data for modeling. This includes normalization, handling missing values, and encoding categorical variables.
- **Model Generation & Evaluation**: SVM, Random Forest, AdaBoost, and XGBoost, are employed to create predictive models. The performance of each model will be evaluated using standard metrics such as accuracy, precision, recall, and F1-score.

- **Interface Development**: A user interface is developed to allow users to input song attributes and receive genre predictions in real-time.

2 Dataset Description

2.1 Introduction to the Dataset

The dataset used in this project is sourced from Kaggle and is titled **Prediction of Music Genre**. It contains a variety of features related to songs, which are used for classifying their genre. The dataset consists of 50000 records and **14 attributes**, each representing a specific characteristic of a song. These features capture various audio properties and metadata that are relevant to the genre classification task.

2.2 Attributes of the Dataset

The main attributes in the dataset are:

- **Popularity**: A numerical value representing the popularity of the song.
- **Acousticness**: A measure of how acoustic the song is, with values ranging from 0 to 1.
- **Danceability**: Describes how suitable a track is for dancing, based on tempo, rhythm stability, and beat strength.
- **Duration (ms)**: The length of the song in milliseconds.
- **Energy**: A measure of intensity and activity, where higher values represent more energetic songs.
- **Instrumentalness**: Predicts whether a song has vocals; higher values indicate more instrumental content.
- **Key**: The key in which the song is composed
- **Liveness**: A measure of how "live" or natural the song sounds, with higher values representing live performances.
- **Loudness**: The overall loudness of the track in decibels.
- **Mode**: Represents the scale of the song (major or minor)

- **Speechiness**: Estimates the presence of spoken words in the track. Higher values indicate more speech-like qualities.
- **Tempo**: The speed of the song, measured in beats per minute (BPM).
- **Valence**: A measure of the musical positiveness conveyed by the track, where higher values indicate more positive moods.
- **Genre**: The target attribute, which represents the genre of the song (e.g., Pop, Rock, Jazz, Classical, etc.).

2.3 Dataset Overview

The dataset is well-structured and provides a rich set of features that capture various musical characteristics, making it a suitable candidate for machine learning tasks. The target variable, Genre, consists of multiple classes that represent different music genres. By analyzing these features, we aim to identify patterns and relationships that can help in building a robust genre classification model.

In the following sections, we will perform an exploratory data analysis (EDA) to understand the distribution of the features, their relationships, and their potential impact on the classification task.

3 Data Analysis and Pre-Processing

In this section, we conducted a thorough analysis and cleaning of the dataset to ensure that it was ready for model training. We explored the dataset structure to understand its characteristics. This included checking for missing values, handling anomalies, and performing basic statistical analysis.

3.1 Handling Missing Values and Anomalies

After a first analysis of the dataset through info, we found out that the dataset contained some missing and anomalous values that needed to be addressed, considering every case in a different manner depending on the information gathered and the type of anomaly. For instance, the `tempo` column had certain entries represented by a question mark ("?"), which were converted to `NaN` and subsequently replaced with the median value of the column. This imputation ensured that we preserved as much data as possible without introducing bias from anomalous values. We then verified that no missing values remained in the `tempo` column, confirming the dataset's completeness.

3.2 Distribution of Numerical Features

To better understand the dataset, we analyzed the distribution of the numerical features, including `popularity`, `acousticness`, `danceability`, `duration_ms`, and others. The histograms, along with the Kernel Density Estimations (KDEs), provide insights into how these features are spread across the dataset.

The distribution plots reveal several key insights. For example, `popularity` exhibits a fairly uniform distribution, with some concentration around specific ranges, while features such as `acousticness` and `instrumentalness` show a clear skew, with the majority of songs having low values. Similarly, `duration_ms` shows a highly skewed distribution due to the presence of long-duration songs, which might indicate potential outliers. On the other hand, features like `danceability` and `energy` appear to follow more normal distributions, with most values centered around specific ranges.

These observations provide important information for the following stages of preprocessing and model training, particularly when considering normalization or further handling of outliers.

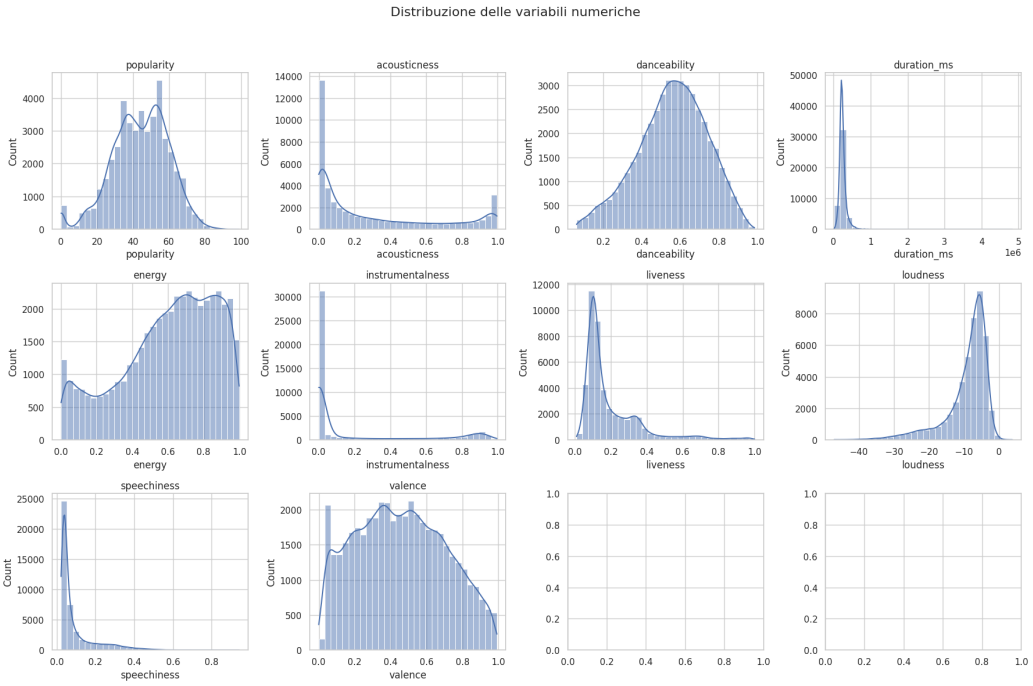

Figure 1: Distribution of Numerical Features in the Dataset

3.3 Relationship Between Features

To explore relationships between different features, we created various scatterplots that reveal patterns and correlations among key variables. These visualizations provide valuable insights that can inform feature engineering and model training, helping to uncover how different musical characteristics interact across various genres.

The first scatterplot compares `danceability` and `energy`, colored by music genre. This plot highlights distinct clusters for certain genres, such as Electronic and Hip-Hop, which tend to exhibit high levels of both danceability and energy, while genres like Anime and Classical are typically found in the lower ranges of these features. Unfortunately, given the very low range of values (both variables are normalized between 0 and 1) it is not easy to distinguish clusters quite distinct from each other (intra-class dissimilarities)

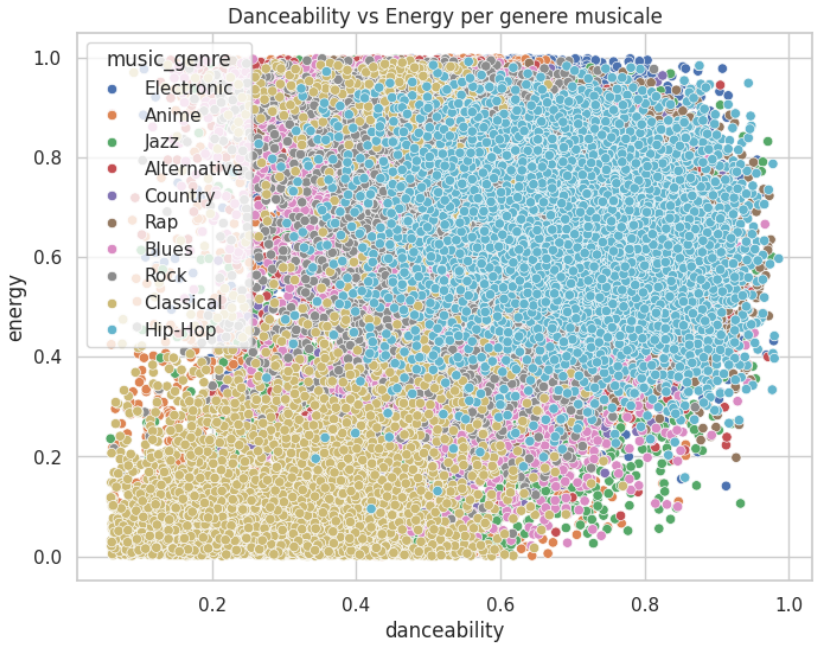

Figure 2: Danceability vs Energy by Music Genre

Next, we examined the relationship between `acousticness` and `valence`, again segmented by genre. Acousticness refers to how acoustic the song is, for example if recorded with a real instruments instead of a studio digital MIDI digitalized one, while valence measures the positivity conveyed by the music. The scatterplot shows that genres such as Classical and Jazz tend to have higher acousticness, in line with the time when most of the songs

of this genre are released, while genres like Electronic and Hip-Hop cluster in the lower acousticness (Hip-Hop is easily observable, while for Electronic we had to see some info about the distribution obtained in the boxplot later on. Valence is more varied across genres, though there is a noticeable trend where higher acousticness corresponds with lower valence.

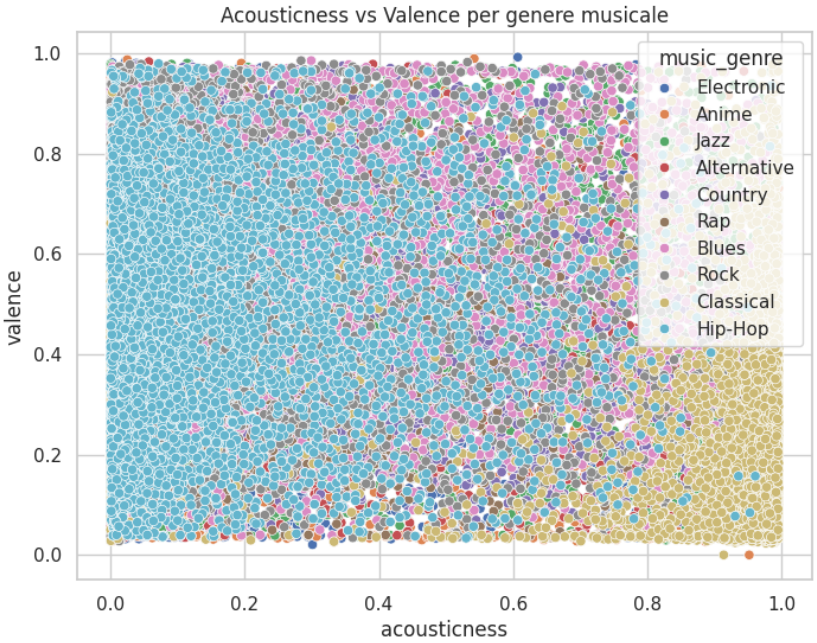

Figure 3: Acousticness vs Valence by Music Genre

Finally, we explored the relationship between `loudness` and `popularity`. This scatterplot with a regression line suggests a positive correlation between loudness and popularity, where louder songs tend to be more popular. However, there are numerous data points across all ranges of popularity, indicating that loudness alone may not be a definitive predictor of popularity but still plays a role in the general trend.

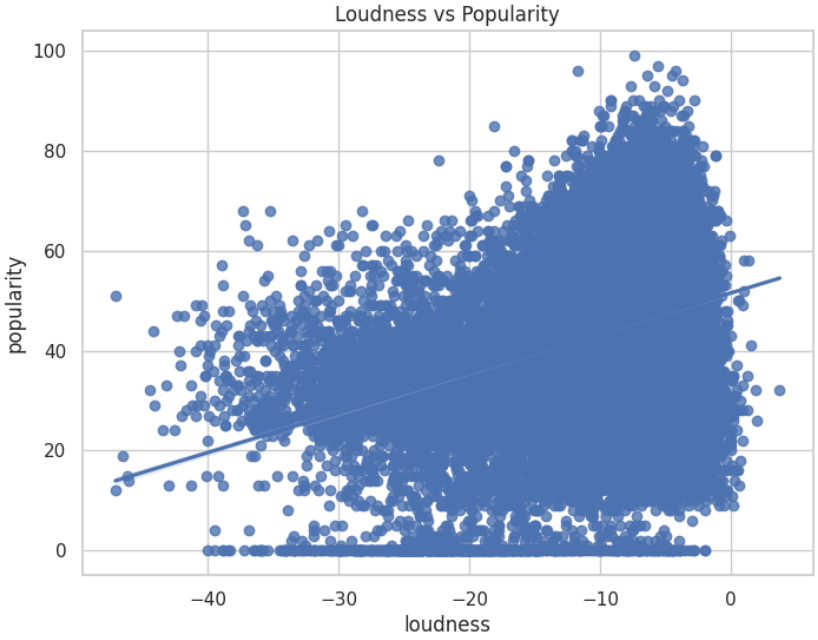

Figure 4: Loudness vs Popularity

3.4 Boxplots and Outlier Handling

Boxplots were generated to visualize the distribution of numerical features across various music genres. These visualizations highlight significant differences in attributes like energy, loudness, tempo, and duration_ms, which are crucial for classification models. Outliers were also identified in several features, such as in duration_ms, where some genres include tracks with extreme lengths, and in acousticness, which shows considerable variability within certain genres.

The boxplots reveal how numerical features differ across music genres. For example, "Anime" typically shows lower median popularity compared to "Country" and "Rap," which tend to have higher values. Extreme outliers with exceptionally high popularity are present in nearly all genres. Acousticness also varies widely, with "Classical" and "Jazz" showing higher values, while "Electronic" and "Hip-Hop" are characterized by much lower levels (like we said in the previous section).

Energy levels show notable differences among genres, with "Electronic" and "Hip-Hop" consistently exhibiting high energy, whereas "Classical" tracks display much lower energy, spread across a wider range of values. Instrumentalness follows predictable trends, with "Classical" being highly instrumental, while genres like "Rap" and "Country" are more vocal-heavy. These

patterns suggest that variables like energy, acousticness, and tempo could serve as strong predictors for genre classification. The presence of outliers, particularly in duration, underscores the importance of handling extreme values carefully during model development.

Certain genres show unique distributions in specific features. For example, "Electronic" and "Hip-Hop" consistently display high energy, with most tracks in these genres being energetic, as indicated by a narrow interquartile range (IQR). In contrast, "Classical" music has much lower energy with a more spread-out distribution. In terms of valence, which measures the positivity of a song, Classical music tends to evoke fewer positive emotions compared to genres like Jazz and Rap.

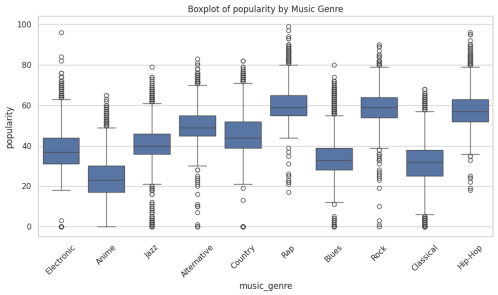

Figure 5: Boxplot of Popularity by Music Genre

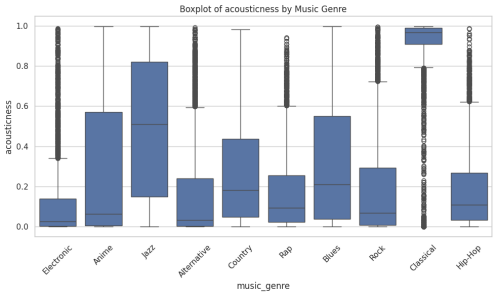

Figure 6: Boxplot of Acousticness by Music Genre

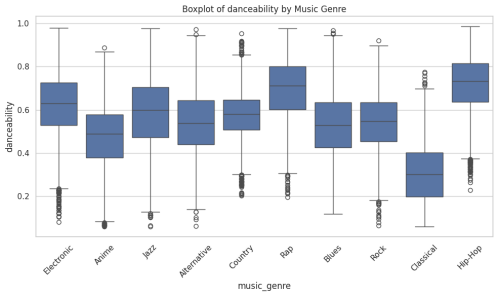

Figure 7: Boxplot of Danceability by Music Genre

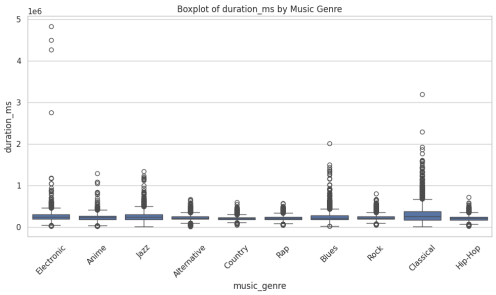

Figure 8: Boxplot of Duration by Music Genre

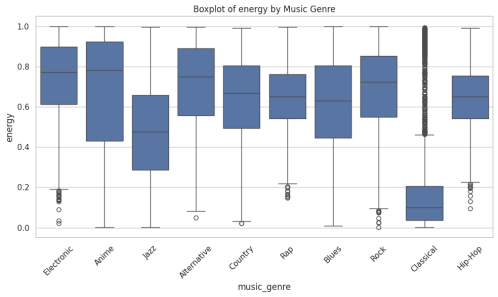

Figure 9: Boxplot of Energy by Music Genre

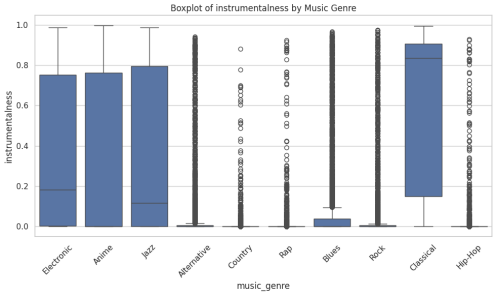

Figure 10: Boxplot of Instrumentalness by Music Genre

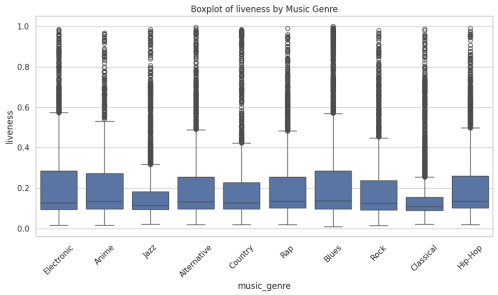

Figure 11: Boxplot of Liveness by Music Genre

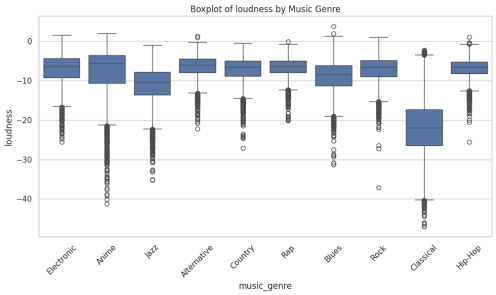

Figure 12: Boxplot of Loudness by Music Genre

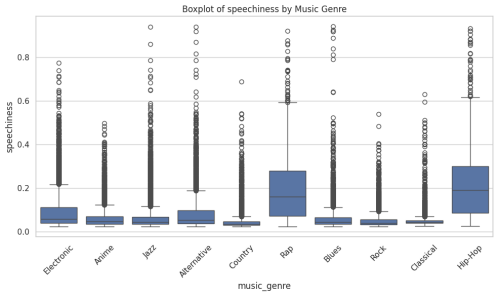

Figure 13: Boxplot of Speechiness by Music Genre

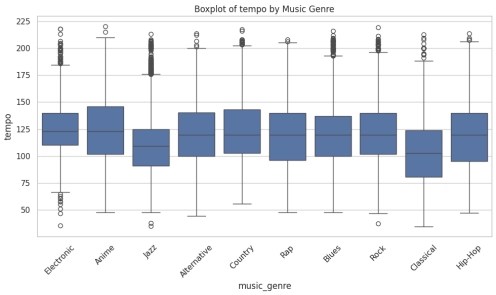

Figure 14: Boxplot of Tempo by Music Genre

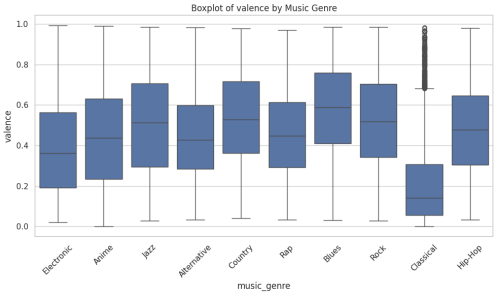

Figure 15: Boxplot of Valence by Music Genre

From these visualizations, we can derive important insights for the model. The distinct differences in features like energy, tempo, and acousticness between genres suggest that these variables could serve as strong predictors for genre classification. Moreover, the presence of outliers in duration and tempo highlights the need for careful handling of extreme values to enhance

model performance.

3.5 Handling Outliers in Duration_ms

The duration_ms feature, which represents the length of songs in milliseconds, presented a significant number of outliers in the dataset. These outliers consisted of extremely long songs, particularly in genres like Classical, that could skew the analysis and affect the performance of the model.

To better manage these outliers, we applied a capping technique based on the interquartile range (IQR). This technique helped limit the extreme values without entirely discarding the data, which is crucial for genres where longer compositions are common. The IQR method involves calculating the difference between the first quartile (Q1) and the third quartile (Q3) and then setting an upper bound for acceptable values. Specifically, any duration above Q3 + 1.5 * IQR was capped to the upper bound.

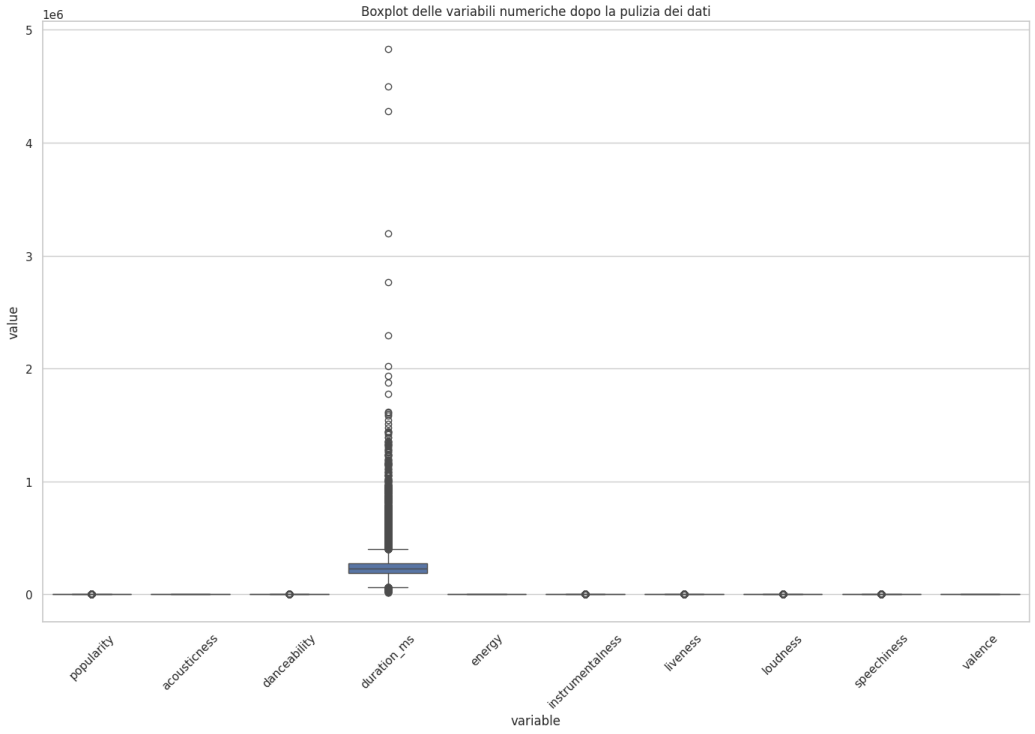

Figure 16: Boxplot of Numerical Variables After Data Cleaning

As the boxplot in Figure 16 demonstrates, after cleaning the data, the presence of outliers in duration_ms was significantly high.

To visualize the impact of the capping process, the distribution of duration_ms was analyzed both before and after the outlier management. The original distribution (left side of Figure 17) exhibited a long tail, with several tracks having durations far exceeding the typical range. After applying the IQR-based capping, the extreme values were capped, resulting in a more normalized distribution, as shown on the right side of Figure 18.

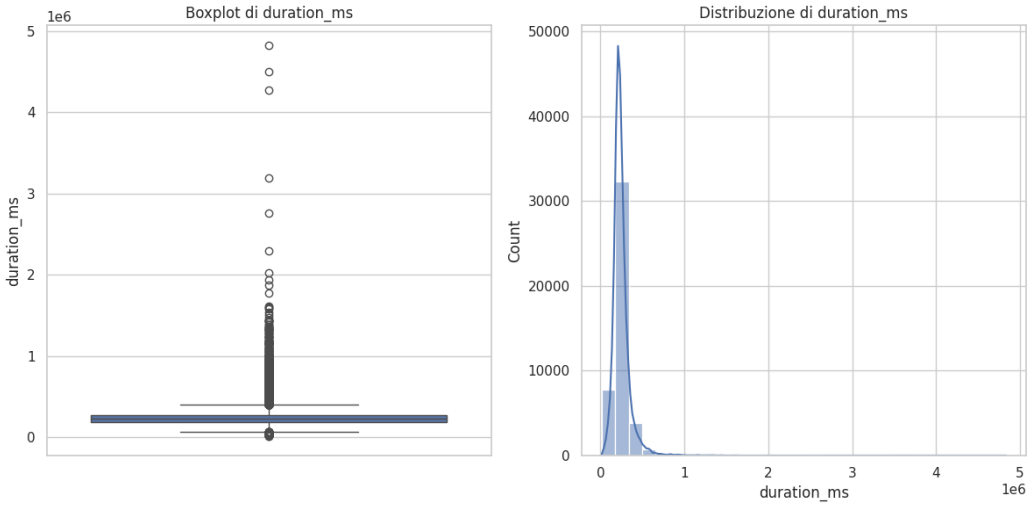

Figure 17: Boxplot and Distribution of Duration_ms before Capping

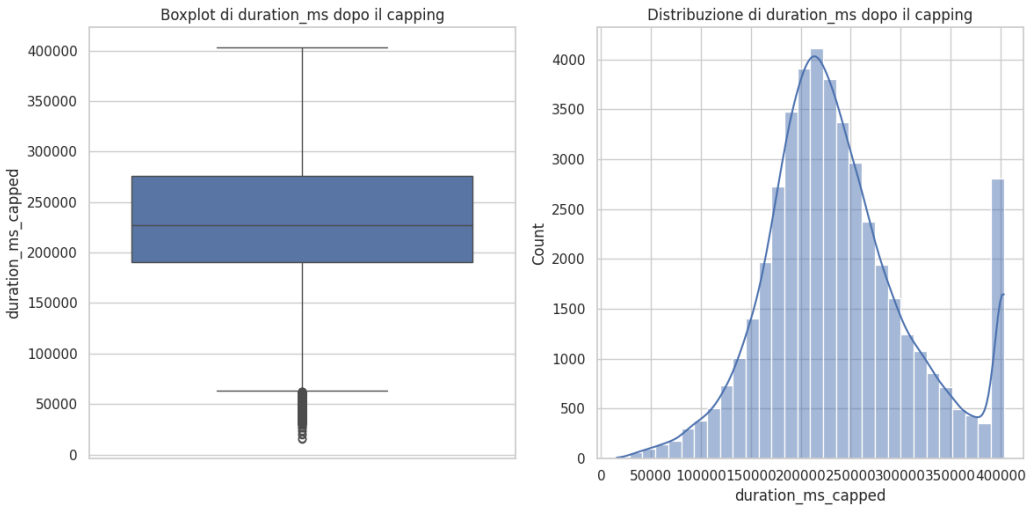

Figure 18: Boxplot and Distribution of Duration_ms after Capping

This approach helped maintain a more consistent range for the duration feature while ensuring that important long-duration tracks, especially in genres like Classical, were still represented, but without distorting the overall analysis.

3.6 Data Encoding and Feature Scaling

To prepare the dataset for modeling, we removed irrelevant columns such as instance_id and artist_name, and applied encoding to categorical variables. The music_genre column was transformed using one-hot encoding, while key and mode were label encoded. Finally, numerical features were normalized using the MinMaxScaler, ensuring that all features were on the same scale and ready for use in machine learning algorithms.

popularity	acousticness	danceability	duration_ms	energy	...
0.45	0.12	0.65	0.015	0.89	...
0.56	0.35	0.45	0.021	0.78	...

Table 1: Sample of the preprocessed dataset.

3.7 Conclusion

After these pre-processing steps, we generated a clean and well-structured dataset, which is now ready for the next phase: model generation and evaluation.

4 Model Generation and Evaluation

Various machine learning models were trained and evaluated, including:

- Support Vector Machine (SVM)
- Random Forest
- AdaBoost
- XGBoost

4.1 Support Vector Machine (SVM)

For the classification of music genres, a Support Vector Machine (SVM) model was trained using the preprocessed dataset. The features used for

training included various audio characteristics such as tempo, loudness, and danceability, while the target variable was the corresponding music genre for each track. The dataset contained 10 different genres, which were originally one-hot encoded. Once the features and labels were extracted, the dataset was split into training and testing sets using an 80/20 ratio. This ensured that 80% of the data was used to train the model, while 20% was reserved for testing its performance. To optimize the performance of the SVM, hyperparameter tuning was conducted using a grid search strategy. The parameters that were optimized included:

- `C`: The regularization parameter, controlling the trade-off between maximizing the margin and minimizing classification errors. The values tested were 1 and 10.
- `Gamma`: This parameter controls the kernel coefficient, with values `scale` and `auto` being evaluated.
- `Kernel`: The function used to transform the data. Three kernel types were tested: radial basis function (RBF), polynomial, and linear.

Grid search with 5-fold cross-validation was applied to determine the best combination of hyperparameters. Cross-validation divides the training data into five subsets, using four for training and one for validation, ensuring the model generalizes well to unseen data.

The best-performing model, based on the grid search, was used to make predictions on the test set. The results obtained were 'C': 10, 'gamma': 'auto', 'kernel': 'rbf'. Although the results were not ideal in terms of accuracy and recall in the first run, so we decided to run an analysis and find a solution to improve this score. Every run on the model generation were followed by a report and a confusion matrix to see the main problematic genres.

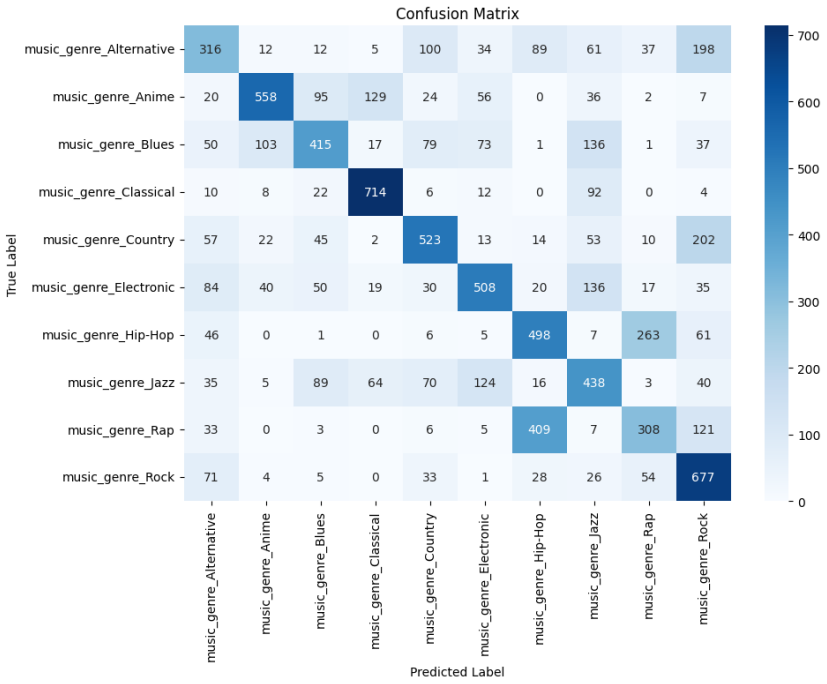

Figure 19: Example of a confusion matrix.

4.1.1 Feature Importance Analysis

After training the Support Vector Machine (SVM) model with a linear kernel, we analyzed the importance of the features used for genre classification. The SVM with a linear kernel allows for the interpretation of feature importance by examining the coefficients associated with each feature. The absolute values of these coefficients were extracted and visualized to identify the most influential features.

The first analysis displays the feature importance across all genres combined. It highlights that certain features, such as popularity and loudness, stand out as the most significant factors for genre classification. Other features like acousticness and instrumentalness play a smaller, but still notable role. This general importance analysis provides insight into which audio characteristics are essential for predicting the genre across multiple categories.

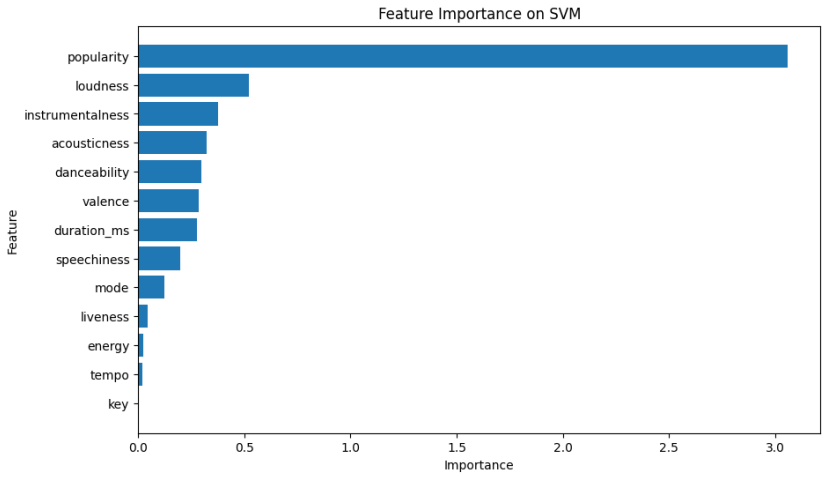

Figure 20: Feature Importance on SVM for All Genres

A second analysis has been done in order to show the feature importance for each specific genre. The results have exposed that, for example,in the Anime genre, loudness is the most significant feature, followed by instrumentalness and energy. These features are crucial in distinguishing Anime tracks from other genres. Interestingly, popularity shows a negative importance, indicating that it plays a lesser role in classifying Anime music compared to other variables.

This analysis was repeated for each genre, revealing unique feature importance patterns for every category. While loudness and energy were common significant features across several genres, other variables, such as tempo or acousticness, played larger roles depending on the genre.

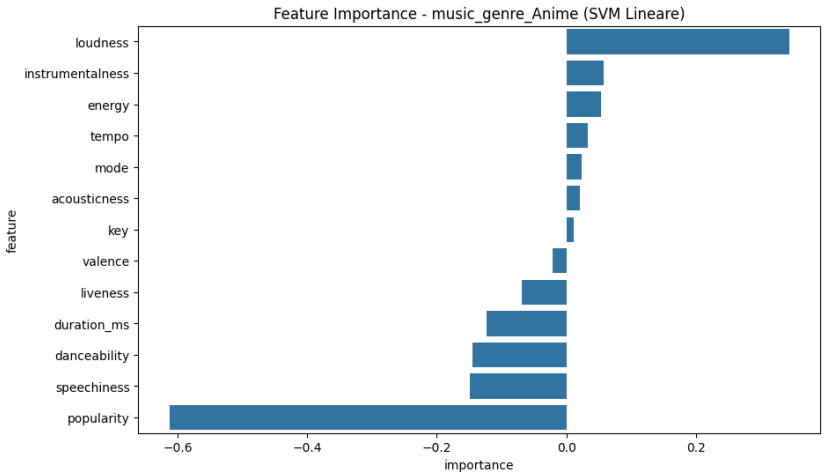

Figure 21: Feature Importance for the `Anime` Genre (SVM Linear Kernel)

Overall, the feature importance analysis highlights the key variables influencing genre classification, with **popularity** and **loudness** consistently emerging as prominent factors. However, the specific importance of features can vary significantly depending on the genre being considered, as seen with the `Anime` genre.

4.1.2 First tentative of imporving

In this phase of the SVM model development, we aimed to improve the classification performance by addressing two particular genres that posed challenges: Alternative and Hip-Hop, founded with an algorithm that searched for the combination of the 2 most problematic genres. These genres were removed from the dataset to evaluate whether their exclusion could improve model accuracy and generalization, despite reducing the model's completeness.

the results showed a significant increase in the accuracy and recall of the model.

The data was split into training and testing sets using again a 80/20 ratio. A pipeline was built to standardize the input features using `StandardScaler` before applying the SVM model with the RBF kernel. The parameters for the SVM model included a regularization parameter $C = 10$ and the `gamma` parameter set to `scale`, which automatically adjusts the kernel coefficient based on the input feature space.

The model was trained using the filtered dataset, and predictions were made on the test set. The accuracy score and classification report were generated to evaluate the performance of the model.

The accuracy achieved using the RBF kernel was reasonably high, indicating that the removal of the problematic genres may have positively impacted the model's performance. The classification report provided detailed insights into precision, recall, and F1 scores for each remaining genre, allowing us to evaluate how well the SVM model performed across different music genres.

Genre	**Precision**	**Recall**	**F1-Score**	**Support**
music_genre_Anime	0.78	0.76	0.77	948
music_genre_Blues	0.59	0.54	0.56	875
music_genre_Classical	0.83	0.85	0.84	903
music_genre_Country	0.60	0.61	0.61	853
music_genre_Electronic	0.69	0.63	0.66	903
music_genre_Jazz	0.57	0.54	0.56	908
music_genre_Rap	0.83	0.80	0.81	917
music_genre_Rock	0.61	0.77	0.68	902
Accuracy			0.69	7209
Macro Avg	0.69	0.69	0.69	7209
Weighted Avg	0.69	0.69	0.69	7209

Table 2: Classification report for SVM with RBF kernel after removing Alternative and Hip-Hop genres.

This approach demonstrated that selectively filtering genres can be a viable strategy for improving classification results, especially when certain genres cause confusion or overlap in the feature space. Further experimentation could explore grouping similar genres or refining the feature selection process to boost accuracy further.

4.1.3 Second tentative of imporving

In this second approach, we attempted to address the performance limitations observed in the first model by grouping genres. Specifically, the genres "Alternative," "Rap," and "Hip-Hop" were merged into a single category labeled as "Grouped Genre". The decision on which genre group has been purely logical. Alternative and Rap genres are basically a sub-genre of Hip-Hop culture. This step aimed to reduce ambiguity and overlap between these genres, which exhibited similar patterns in the feature space.

The remaining genres were maintained as distinct categories: Anime, Blues, Classical, Country, Electronic, Jazz, and Rock. We used the SVM model with the RBF kernel and the grid search specific to train and test the model, using an 80/20 split for the dataset. After training the model on the modified dataset, the predictions were evaluated on the test set.

Below are the accuracy and classification report for this run:

Genre	Precision	Recall	F1-Score	Support
Grouped_Genre	0.74	0.82	0.78	2643
music_genre_Anime	0.76	0.75	0.75	927
music_genre_Blues	0.60	0.54	0.57	912
music_genre_Classical	0.84	0.84	0.84	868
music_genre_Country	0.59	0.57	0.58	941
music_genre_Electronic	0.68	0.59	0.63	939
music_genre_Jazz	0.54	0.47	0.50	884
music_genre_Rock	0.53	0.58	0.56	899
Accuracy			0.68	9013
Macro Avg	0.66	0.64	0.65	9013
Weighted Avg	0.68	0.68	0.68	9013

Table 3: Classification report for SVM with RBF kernel after grouping genres.

The overall accuracy of the model is 67.87%, with moderate precision and recall values across different genres. Grouping the genres "Alternative," "Rap," and "Hip-Hop" seems to have stabilized the performance, particularly in the Grouped Genre category, which achieved an F1-score of 0.78. However, some genres, such as Blues and Jazz, still show lower performance, indicating potential areas for further optimization.

4.2 Random Forest

For the Random Forest model, we conducted similar tests to evaluate its performance in predicting the music genre. The best result was achieved by removing the two problematic genres ("Alternative" and "Hip-Hop"), which helped improve the model's accuracy.

After tuning the model and optimizing parameters, we achieved an accuracy of 68.65% on the test set. Below is the classification report that summarizes the precision, recall, and F1-scores for each genre:

Genre	Precision	Recall	F1-Score	Support
Anime	0.80	0.74	0.77	948
Blues	0.61	0.50	0.55	875
Classical	0.83	0.85	0.84	903
Country	0.59	0.59	0.59	853
Electronic	0.64	0.60	0.62	903
Jazz	0.58	0.52	0.55	908
Rap	0.83	0.82	0.82	917
Rock	0.60	0.86	0.71	902
Accuracy			0.69	7209
Macro Avg	0.69	0.68	0.68	7209
Weighted Avg	0.69	0.69	0.68	7209

Table 4: Classification report for Random Forest after removing two genres.

The Random Forest model performed particularly well on the "Rap" and "Classical" genres, achieving F1-scores of 0.82 and 0.84, respectively. However, "Blues" and "Jazz," again displayed lower precision and recall values, indicating potential challenges in distinguishing these categories. Overall, the model's performance with a 68.65% accuracy demonstrates that removing the two problematic genres was effective in improving classification performance across the remaining genres in this model too.

4.2.1 Learning Curve for Random Forest

To better understand the performance and behavior of the Random Forest model, we generated a learning curve to observe how the model's accuracy evolves as the training set size increases. This plot highlights two important metrics: the training accuracy and the cross-validation accuracy, as shown in Figure 22.

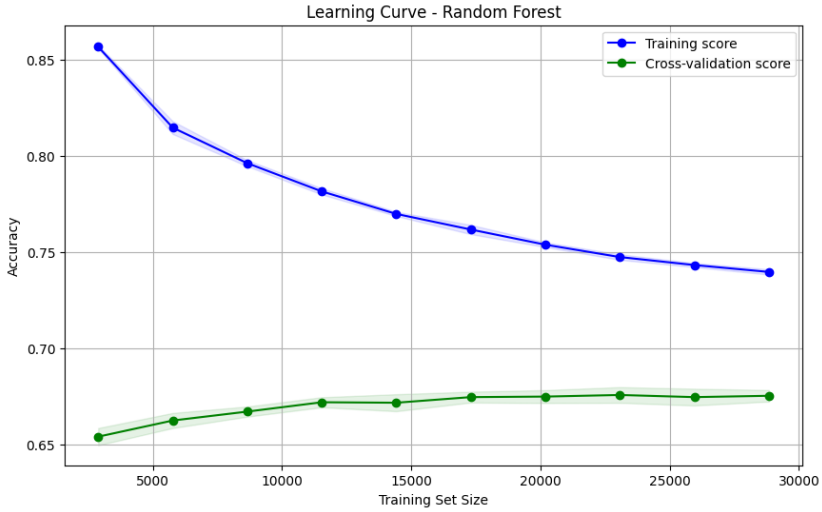

Figure 22: Learning curve of the Random Forest model.

The training score starts relatively high, above 85%, and gradually decreases as the training set size increases. This decrease is expected since smaller training sets tend to overfit, resulting in higher accuracy on the training data. As the dataset grows, the training accuracy stabilizes around 75%, indicating a more generalized model.

In contrast, the cross-validation score starts lower, at around 65%, and slowly improves as more training data is added, eventually stabilizing around 68%. The gap between the training and cross-validation accuracy suggests a mild overfitting, which could potentially be addressed by further tuning hyperparameters or increasing the training set size.

Overall, this learning curve demonstrates that the Random Forest model benefits from more data, as both training and validation accuracy stabilize as the dataset grows.

4.2.2 AdaBoost

The same tests conducted with Random Forest and SVM have been conducted in the generation of the Adaboost model. Although even in his best case scenario (the one obtained by combining grid search for finding the best variable and removing the 2 most problematic genres) this model didn't reach an higher accuracy or recall than the Random forest and SVM model. Further testing should focus on finding better parameters, considering a wider range of variables or conducting a more specific analysis on the challenging genres.

Class	Precision	Recall	F1-Score	Support
music_genre_Anime	0.81	0.63	0.71	948
music_genre_Blues	0.53	0.50	0.51	875
music_genre_Classical	0.79	0.75	0.77	903
music_genre_Country	0.61	0.57	0.59	853
music_genre_Electronic	0.62	0.58	0.60	903
music_genre_Jazz	0.47	0.54	0.50	908
music_genre_Rap	0.82	0.80	0.81	917
music_genre_Rock	0.60	0.81	0.69	902
Accuracy			0.65	7209
Macro Avg	0.66	0.65	0.65	7209
Weighted Avg	0.66	0.65	0.65	7209

Table 5: Classification report with accuracy of 0.65 on the test set

4.2.3 ROC Curve Analysis

To further evaluate the performance of the AdaBoost model, we analyzed the ROC (Receiver Operating Characteristic) curves for each genre. Figure 23 displays the ROC curves, which provide a measure of the model's ability to differentiate between the genres.

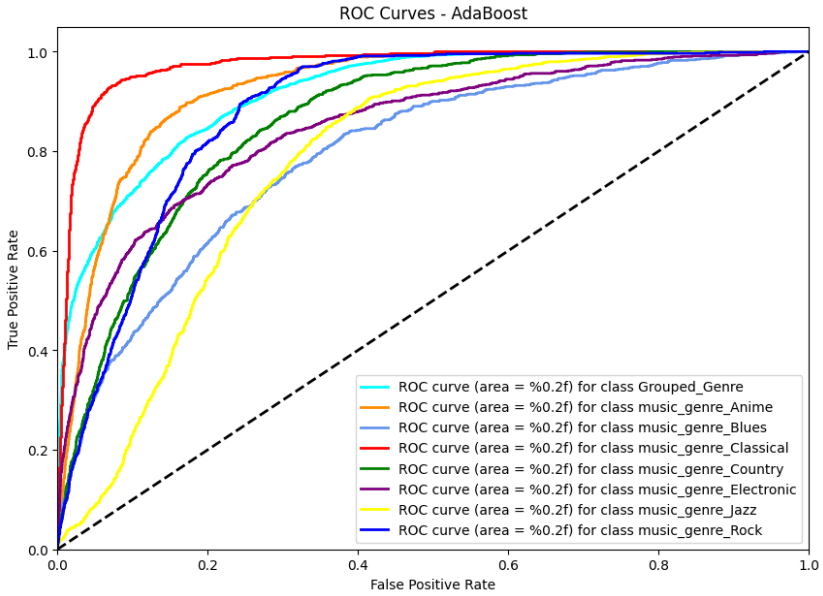

Figure 23: ROC Curves of AdaBoost Model for Each Music Genre

The ROC curves give insights into the trade-off between true positive

rate and false positive rate for each genre. From the curves, it is clear that genres such as "Grouped Genre" and "Classical" show higher AUC values, indicating that the model classifies them more effectively. On the other hand, some other genres exhibit lower AUC values, suggesting that the model has more difficulty in distinguishing these genres, as we have seen in the previous tests.

Overall, while the accuracy of AdaBoost was similar, but mildly worse than the one on Random Forest, the ROC curve analysis highlights the varying levels of difficulty in classifying different genres, providing a more granular view of the model's strengths and weaknesses.

4.3 XGBoost

XGBoost was the final model we tested, and it delivered the best overall performance among all the models evaluated. Initially, without applying any of the improvements such as genre removal or grouping, XGBoost achieved an accuracy of 60%, already surpassing previous models like AdaBoost and Random Forest.

To further optimize the model, we applied the same strategies used in the other experiments. The removal of the two "problematic" genres, "Alternative" and "Hip-Hop," led to the highest accuracy of 71%.

Genre	Precision	Recall	F1-Score	Support
Anime	0.83	0.78	0.80	948
Blues	0.62	0.55	0.58	875
Classical	0.84	0.86	0.85	903
Country	0.64	0.65	0.64	853
Electronic	0.70	0.63	0.66	903
Jazz	0.58	0.57	0.57	908
Rap	0.84	0.83	0.84	917
Rock	0.63	0.80	0.70	902
Accuracy		0.71		
Macro Avg	0.71	0.71	0.71	7209
Weighted Avg	0.71	0.71	0.71	7209

Table 6: Classification Report for XGBoost After Genre Removal

4.4 Analysis with Shap and feature importance

In the additional tests conducted with XGBoost, we aimed to further analyze feature importance and interaction between different audio characteristics

using SHAP (SHapley Additive exPlanations) values. These analysis help us understand the model's decision-making process more transparently.

The first plot (Feature Importance - Gain) shows the relative importance of each feature based on the gain metric, which measures the improvement brought by a feature to the branches it is used in. As we can see, popularity and loudness are the most important features, significantly contributing to the model's decisions. These two features dominate over others like instrumentalness, speechiness, and mode, which have moderate importance. The lower-ranked features, such as key, tempo, and liveness, have a smaller impact on the model's predictions.

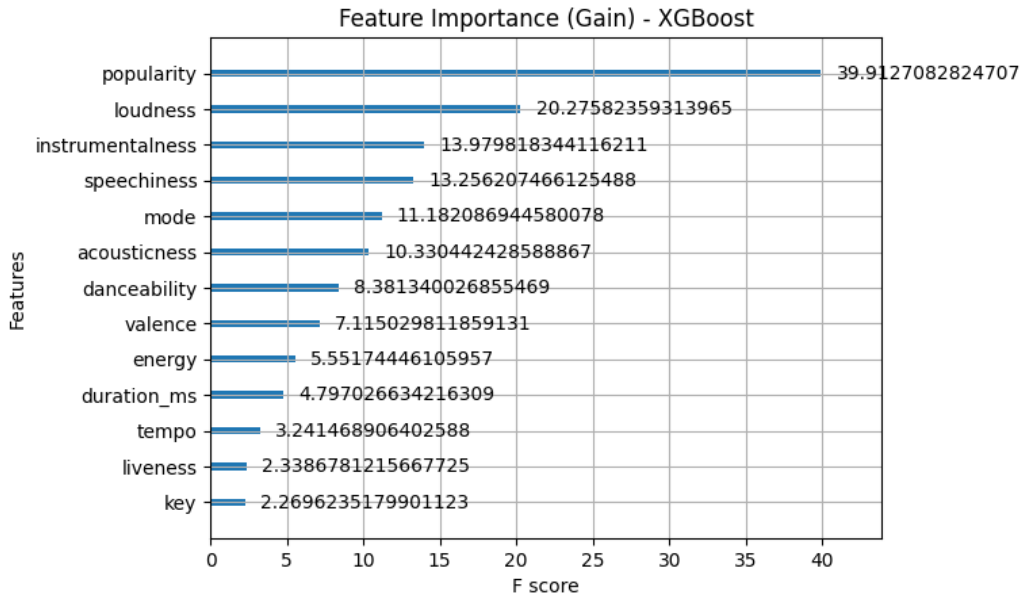

Figure 24: Feature Importance (Gain) - XGBoost

The second plot (SHAP Interaction Values) offers a more nuanced look at how different features interact with each other in determining the model's output. SHAP values quantify the contribution of each feature to a prediction, while the interaction values specifically highlight the interaction between two features.The violin shapes in the plot show how each feature and its combinations with others affect the model's output. For example, popularity has a significant influence on predictions, with colors indicating the direction of the contribution: higher values of popularity tend to increase the likelihood of certain classes, while lower values decrease it. Some features, such as liveness and duration, exhibit less pronounced interactions, indicat-

ing that their effects are less influential compared to features like popularity or energy. In other words, a song's popularity and energy levels play key roles in determining its genre, whereas other variables have a secondary role or contribute primarily in combination with other features.

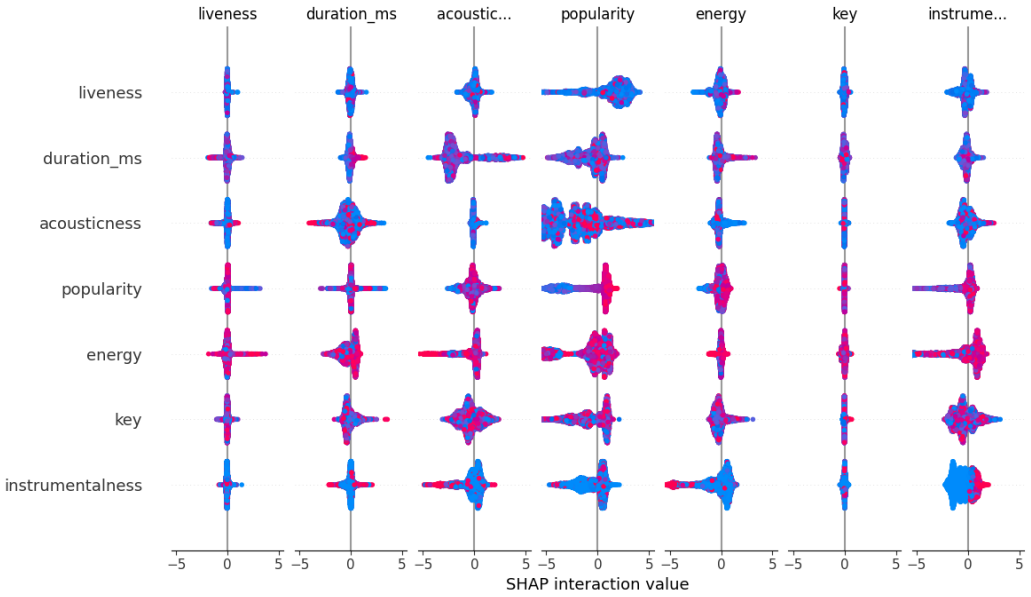

Figure 25: SHAP Interaction Values - XGBoost

4.4.1 Jazz and Blues

As we have seen, Jazz and Blues consistently showed lower performance compared to the other genres. After thoroughly analyzing the results, we can conclude that this is likely due to the fact that both Jazz and Blues serve as foundational genres upon which others, such as Country, Rock, Rap, and Pop, have been built. These genres often share overlapping characteristics, making it more difficult for the model to clearly distinguish between them. The similarities in musical features like rhythm, acousticness, and instrumentation between Jazz, Blues, and these more modern genres may contribute to this reduced predictive accuracy. Additionally, the evolving nature of these genres means that they have influenced each other over time, further blurring the lines between them in terms of audio features.

4.4.2 Model selection for the interface

Although XGBoost performed exceptionally well with the genre removal strategy, for the final interface, we chose to implement the model that grouped "Alternative," "Rap," and "Hip-Hop" into a single category. This decision was made because the grouped model achieved an overall accuracy of 71%, very close to the genre removal strategy, while maintaining a broader range of classification categories. The grouped model offers a more generalized approach and simplifies the classification process without compromising too much on performance, making it a better fit for the user interface.

5 Interface

5.1 Introduction

An interactive interface was developed within the Colab environment to allow users to predict the genre of a song based on its audio characteristics. This interface, named the **Music Genre Predictor**, enables users to input song attributes such as tempo, loudness, valence, and other features using input widgets like sliders and dropdown menus provided by Colab. After entering the song's attributes, the model predicts the genre in real-time. The interface is accessible within Colab, making it convenient for users working in the notebook environment, and is easy to use for testing different song attributes.

5.2 Features of the Interface

The interface allows users to input song attributes through intuitive input widgets. Sliders are provided for numerical values such as energy level, popularity, and danceability, while dropdown menus are used for categorical variables like the song's key and mode. Once all the data is entered, the user can click the Predict Genre button to initiate the prediction. The predicted genre is displayed immediately below the input section, making the process highly interactive. Built-in validations ensure that inputs remain within valid ranges, thus avoiding errors due to incorrect data entries.

5.3 User Guide

To use the interface, users simply need to run the Colab notebook, which displays input widgets for entering the song's attributes and load the XGB-model inside it. For numerical values, such as popularity and duration, users can adjust sliders, while for attributes like key and mode, users can select

from dropdown menus. For example, the key can be chosen from options like C, C#, D, etc., and the mode can be set to major or minor. After entering the required attributes, the user can click the predict button, which triggers the prediction model to display the genre. The prediction is then displayed in a clear format just below the input widgets.

5.4 Future Enhancements

Several potential enhancements could improve the Colab-based interface. One key addition could be the ability to upload a dataset of multiple songs, enabling the system to predict genres for a batch of songs simultaneously. Furthermore, visualizations such as dynamic charts could be integrated to display the relationships between input attributes and predicted genres, offering a more comprehensive analysis experience.

5.5 Screenshots

Below is a screenshot of the developed interface in Colab, showing the input widgets for entering song attributes and the genre prediction result:

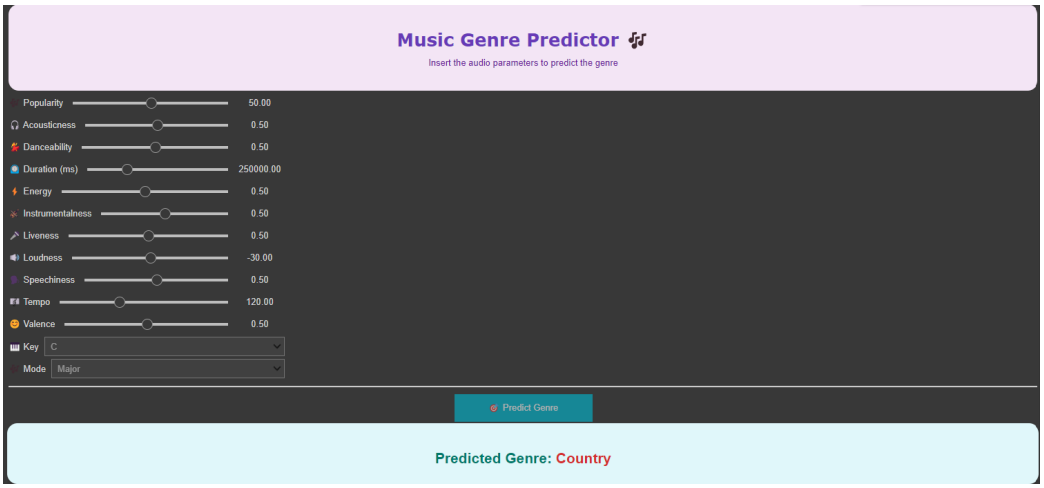

Figure 26: Colab interface for inputting song attributes and predicting the genre.

6 Conclusion

Throughout this project, we explored various machine learning models to classify music genres based on audio features. Each model presented unique strengths and weaknesses, with XGBoost standing out as the most effective, achieving 71% accuracy after refining the dataset by removing or grouping certain genres. However, performance across models highlighted challenges in distinguishing genres like Jazz and Blues, which overlap significantly with other styles like Rock and Country.

The results emphasize that while machine learning can be powerful for genre classification, it faces limitations when genres share foundational traits. This was especially evident with the models' difficulty in distinguishing closely related genres, as seen in the lower precision for Jazz and Blues. It became clear that these genres, fundamental to many modern styles, often blur boundaries, complicating classification efforts.

The interface developed for this project further demonstrated the practical applications of these models, allowing real-time predictions based on user input. Although the system performs well in most cases, expanding its capabilities to handle more complex genres or adding support for batch predictions could further improve its utility.

In conclusion, while our approach successfully classifies most genres, more sophisticated techniques or additional features may be needed to address the complexities of overlapping musical styles. This project offers a solid foundation but also highlights areas for further research and development in genre classification.