Music Genre Classification Based on Audio Features

Ruoyun(Cynthia) Yang

Data Science Initiative | Brown University

GitHub Link:https://github.com/cynthiayry/1030ProjectCynthia.git

1. Introduction

Music has always been a powerful cultural artifact that connects people and shapes identity. It is a nexus of intercultural change, continuously shaped by the diverse cultural influences that intersect within it and often come with different labels. These labels could be potentially valuable in an age where virtual information and interactive music applications are in high demand: recommendation systems, playlist generation, audio analytics, and many more[6].

This project aims to develop a multi-class classification model and help address the challenges of predicting and classifying music genres based on audio features for real life applications. The dataset was found on Kaggle[1] and originally sourced from Machine Hack[2]. It includes 17,996 tracks with 16 features and one categorical label (class) representing 11 genres.

	Description	**Data Type**
Artist Name	The name of the artist.	Categorical(text)
Track Name	The title of the music track.	Categorical(text)
Popularity	A numerical score typically ranging from 0 to 100 that reflects the current popularity of the track.	Continuous
Danceability	A measure from 0.0 to 1.0 indicating how suitable a track is for dancing based on tempo, rhythm stability, beat strength, and overall regularity.	Continuous
Energy	A measure from 0.0 to 1.0 that represents a perceptual measure of intensity and activity.	Continuous
Key	The key the track is in, encoded as integers where C=0, C#=1, D=2, etc., up to B=11.	Categorical(numerical)
Loudness	The overall loudness of a track in decibels (dB).	Continuous
Mode	Indicates the modality (major or minor) of a track,	Categorical(numerical)

	derived from the scale used. (0 = minor, 1 = major).	
Speechiness	Measures the presence of spoken words in a track.	Continuous
Acousticness	A measure from 0.0 to 1.0 of whether the track is acoustic.	Continuous
Instrumentalness	Predicts whether a track contains no vocals. Values closer to 1.0 indicate the track contains no vocal content.	Continuous
Liveness	Detects the presence of an audience.	Continuous
Valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track.	Continuous
Tempo	The overall estimated tempo of a track in beats per minute (BPM).	Continuous
Duration_in min/ms	The length of the track either in minutes or milliseconds, a numeric variable that can influence the structure and style of a track.	Continuous
Time_signature	A notational convention to specify how many beats are in each bar (or measure).	Categorical(numerical)
Class	The target variable for classification tasks.Acoustic/Folk:0; Alt_Music:1; Blues: 2; Bollywood:3;Country:4;HipHop:5;Indie Alt:6;Instrumental: 7;Metal:8;Pop:9;Rock:10.	Categorical(numerical)

Table 1. Summary of dataset columns.

2. Exploratory Data Analysis

Two key challenges are identified during EDA: missing values and imbalance.

Three features with missing values are shown in Table2, with the total fraction of points with missing values being 34.46%.

	Fraction with Missing Values	**Data Type**
Popularity	0.023783	Continuous
Key	0.111914	Categorical(numerical)
Instrumentalness	0.243221	Continuous

Table 2.Summary of missing values.

As illustrated in Figure1, Certain genres, like Country and Bollywood, have significantly fewer samples, necessitating stratified splitting for balanced training and evaluation.

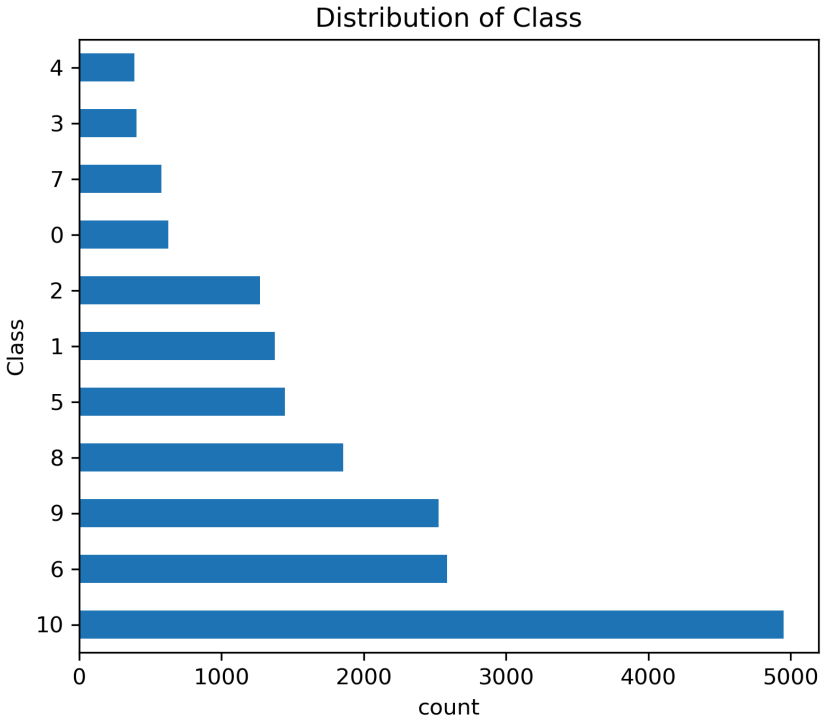


Figure 1. Distribution of class labels; with count on x-axis and class on y-axis.

Notably, other graphical analysis indicates potential correlations between different features.

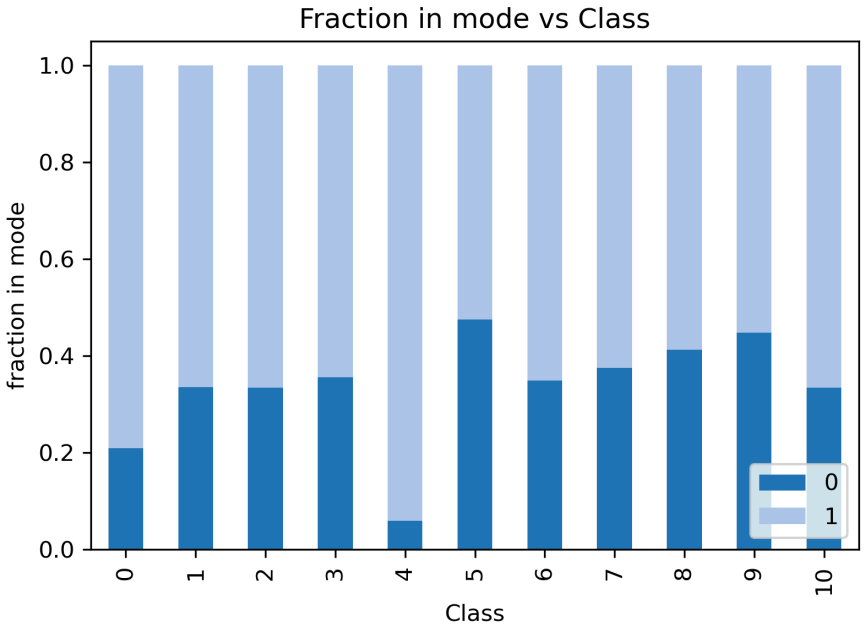
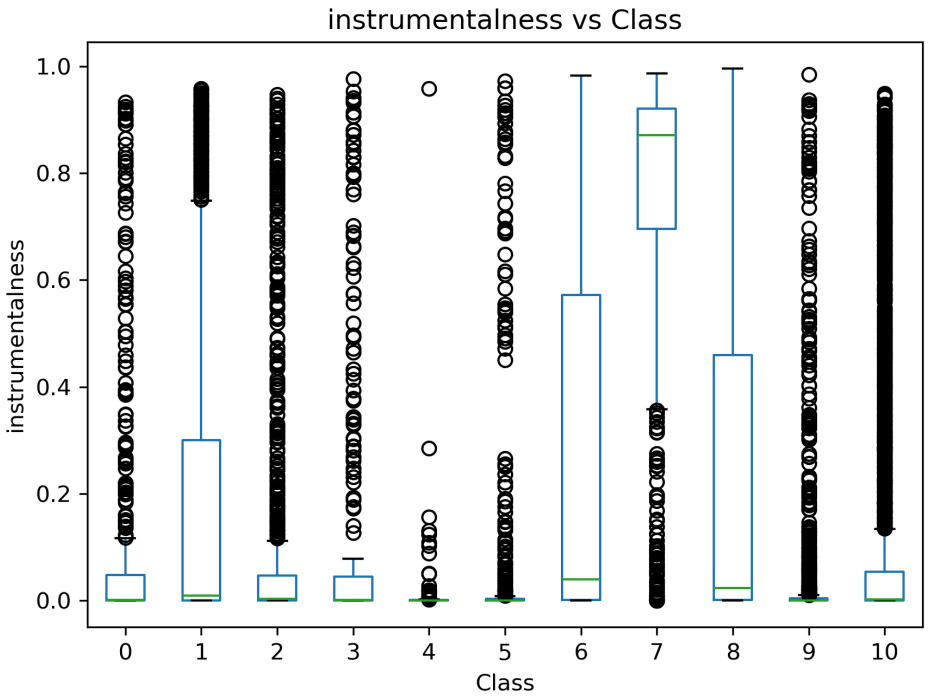


Figure 2. Stacked-bar plot of mode vs. class; with class on x-axis and fraction in mode on y-axis. Over 90% of country(4) music tracks use a major mode.

Figure 3. Boxplot of instrumentalness vs. class; with class on x-axis and instrumentalness on y-axis.



Country(4) and Hiphop(5) have the lowest instrumentalness.

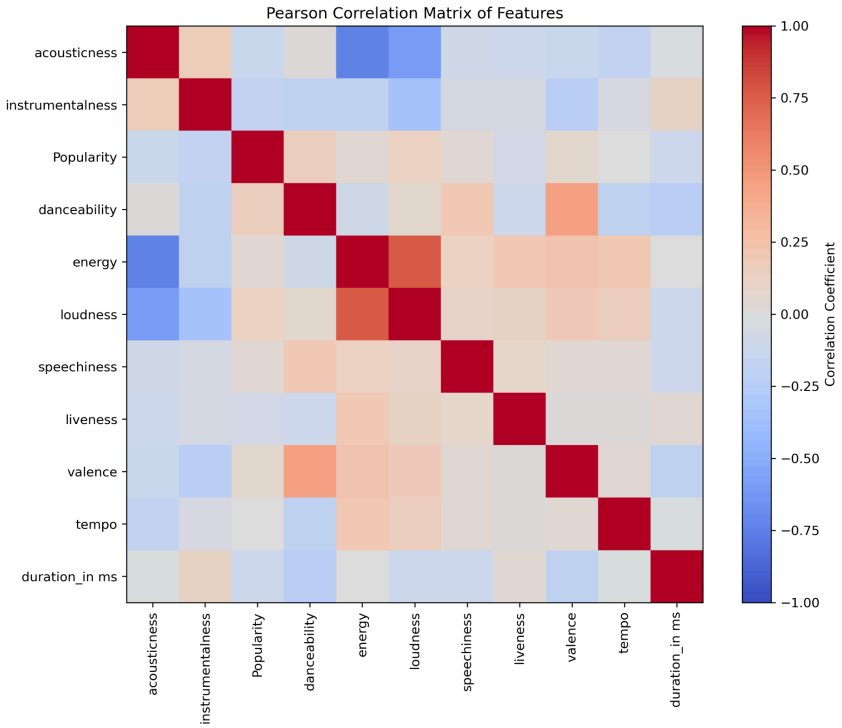


Figure 4. Correlation Matrix of continuous features(before preprocessing pipeline). Strong positive correlations between energy and loudness/valence and danceability; acousticness correlates negatively with energy/loudness.

3. Methodology

3.1 Splitting, Preprocessing and Missing Values

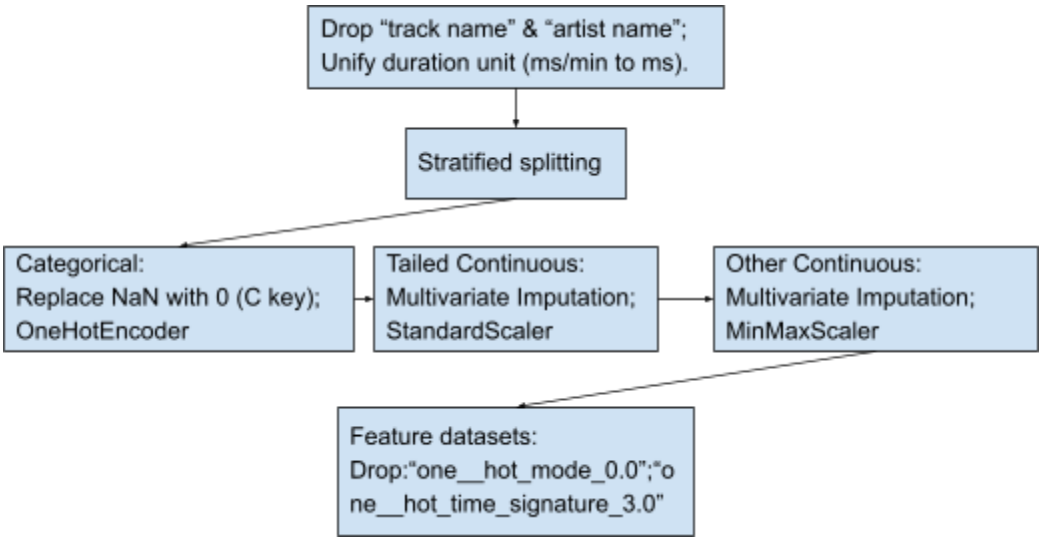


Figure 5. Splitting and preprocessing logic structure.

Stratified splitting is implemented to address class imbalance, which ensures proportional representation of all genres. The first split would be achieved using stratified simple train_test_split, and StratifiedKFold would be used during cross-validation.

For preprocessing, categorical features are one-hot encoded to generate dummy variables for each category. Continuous features with tailed distributions are scaled using the StandardScaler, while others are scaled with MinMaxScaler.

To handle Missing Values, missing “key” values are replaced with 0 (C key) using a simple imputer. This decision is supported by manual verification[4] of missing keys for 200 tracks through online research. For continuous features, multivariate imputation is applied. This technique estimates missing values using correlations among features, which ensures consistency(not needed for XGBoost[5]).

3.2 Evaluation Metrics

I choose log loss as the main evaluation metric for model selection since log loss evaluates the entire probability distribution. This is particularly advantageous for tasks like music recommendation, where ranking songs by their likelihood of belonging to a genre is important. Log loss penalizes confident but incorrect predictions, making it suitable for handling class imbalances by discouraging overfitting.

Macro F1 is also calculated as a side reference. It balances precision and recall, which are ideal for imbalanced datasets as they focus on the positive classes and exclude TNs. The harmonic mean of precision and recall ensures the metric is sensitive to both identifying all genres and minimizing misclassifications.

3.3 Cross Validation Pipeline

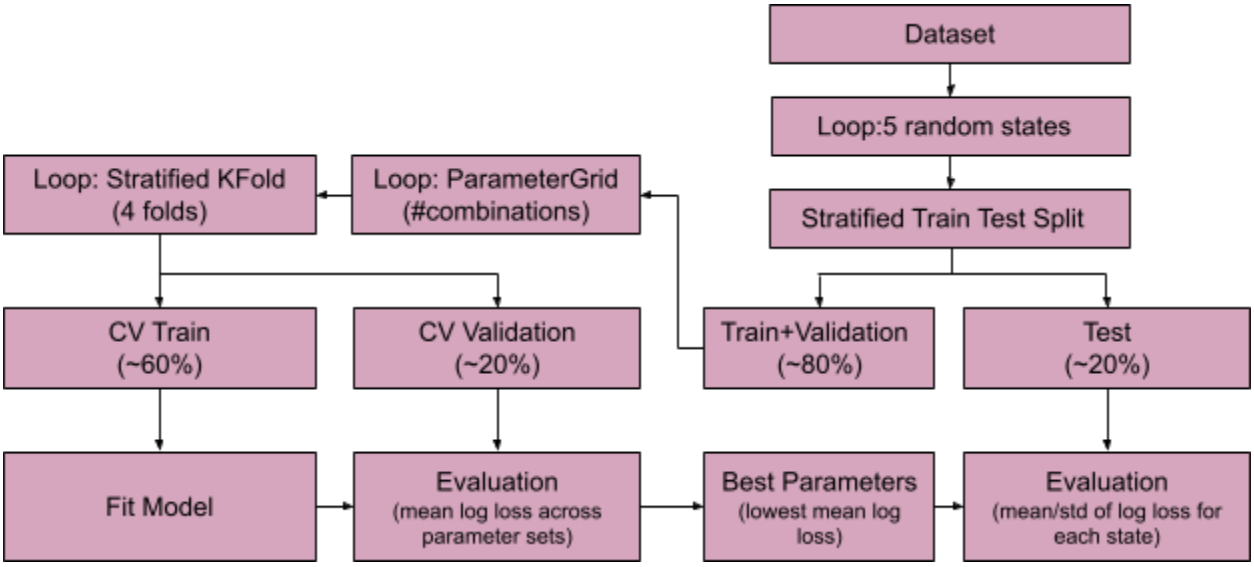


Figure 6. Cross validation pipeline logic structure.

As illustrated in Figure6, the cross-validation pipeline follows a multi-level loop structure. The process starts with entering the outer loop of 5 random states, where the dataset is

split into 80% training/validation (other) and 20% test sets using stratified train-test split. In the middle loop, the pipeline iterates over a grid of parameters. In the inner loop of StratifiedKFold, for each parameter combination, K = 4 is applied to the training/validation split so that 4 models are trained and 4 validation scores are calculated. The parameter set with the lowest mean log loss across the folds is chosen as the best for that random state, and is applied on the test set.

3.4 Machine Learning Models and Hypertuning

	Logistic Regression(l2)	**Logistic Regression (elastic net)**	**Random Forest Classifier**	**Support Vector Classifier**	**XGBoost Classifier**	**K Nearest Neighbors**
Classification Strategy	One-vs-Rest	One-vs-Rest	Multi-class	One-vs-One	Multi-class	Multi-class
Parameters Tuned	C-l2reg strength (log scale)	C-reg strength (log scale); l1_ratio (linear scale)	max_depth; (linear scale) max_features; n_estimators (both log scale)	C-reg strength; gamma-kernel coef (both log scale)	reg_alpha-l1reg; reg_lambda-l2reg; (both linear scale) max_depth (log scale)	n_neighbors (log scale); weights (["uniform", "distance"]); p-distance metric ([1, 2])
Values Tuned	C: [0.01, 0.1, 1, 10, 100]	C: [0.01, 0.1, 1, 10, 100]; l1_ratio: [0.1, 0.5, 0.9]	max_depth: [1, 3, 10, 30, 100]; max_features: [0.25, 0.5, 0.75, 1.0]; n_estimators: [50, 100, 500]	C: [0.1, 1, 10, 100], gamma: [0.01, 0.1, 1]	reg_alpha: [1e0, 1e1, 1e2]; reg_lambda: [1e1, 1e2, 1e3]; max_depth: [3, 10, 15]	n_neighbors: [3, 9, 20, 30, 50, 100, 200]; weights: ["uniform", "distance"]; p: [1, 2]
Optimum Parameters	C=1	C=1; l1_retio = 0.9	max_depth=30; max_features=0.25; n_estimators=500	C=1; gamma = 0.1	reg_alpha=10; reg_lambda=100; max_depth=10	n_neighbors=200; weights=uniform; p=1
Evaluation Metric	Log loss (+macro f1)	Log loss (+macro f1)	Log loss (+macro f1)	Log loss (+macro f1)	Log loss (+macro f1)	Log loss (+macro f1)

Table 2.Summary of ML models and hyperparameter tuning.

Six models are implemented with ParameterGrid for hyperparameter tuning with StratifiedKFold cross-validation, and their corresponding information is summarized in Table3. What's more, to better deal with imbalance, I use sample_weights for XGBoost and class_weight = balanced for other models (except KNN).

Uncertainties from splitting and non-deterministic models are addressed by aggregating the standard deviations of test scores for each ML model across 5 random states, and will be visually showcased as error bars below.

4. Results

4.1 Best Model Selection

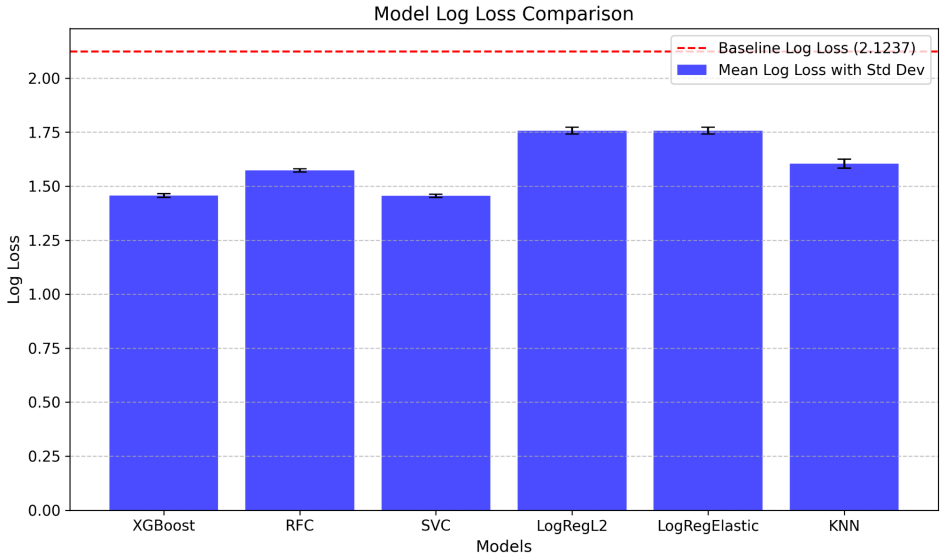


Figure 7. Comparison of mean log loss scores across 6 ML models; with models on x-axis, mean log loss on y-axis, standard deviations of the log loss scores as the error bars. (*Baseline log loss calculated as the negative sum of each class's probability multiplied by the log of that probability.)

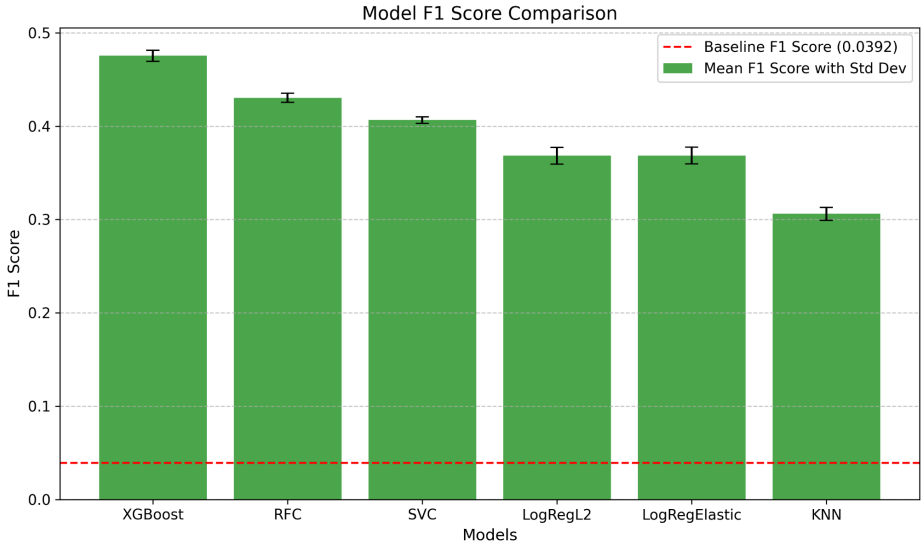


Figure 8. Comparison of mean macro F1 scores across 6 ML models; with models on x-axis, mean macro F1 scores on y-axis, standard deviations of the F1 scores as the error bars. (*Baseline F1 score calculated by predicting the most frequent class for all samples and measuring the average (macro) F1 score across all classes.)

As shown in Figure 7 and 8, the best models based on log loss are XGBoost(mean= 1.4568) and SVC(mean = 1.4553) with absolute decrease by 0.67 and percentage decrease by 31% compared to baseline. However, since XGBoost also outperforms all the others with the highest mean f1 = 0.4753 (absolute increase by 0.4361 and percentage increase by 1113%), XGBoost is the best model.

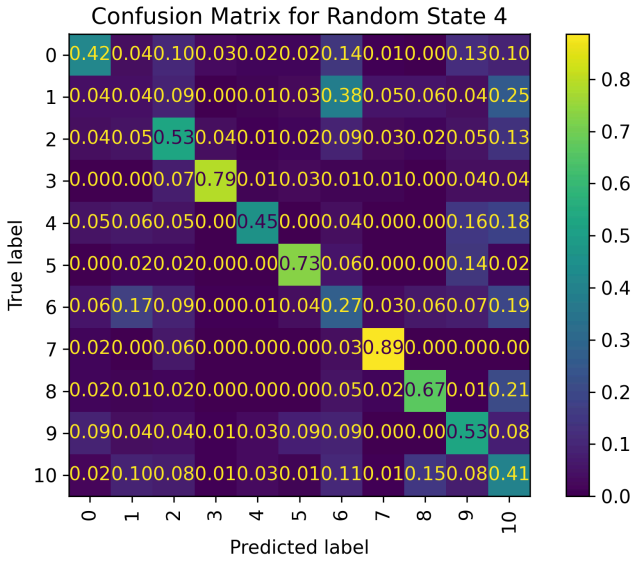


Figure 9. Confusion matrix for XGBoost random state = 4 (normalized row-wise).

Class 3(Bollywood), 5(Hiphop), and 7(instrumental) are classes with the highest proportion of correct predictions, while class 1(Alternative Music) is the lowest. The reason behind may lie in the distinctiveness of features associated with classes like Hiphop and Instrumental. In contrast, Alternative Music likely overlaps in feature space with other genres with a diversity of sub genres, leading to more misclassifications.

4.2 Best Model and Global Importance

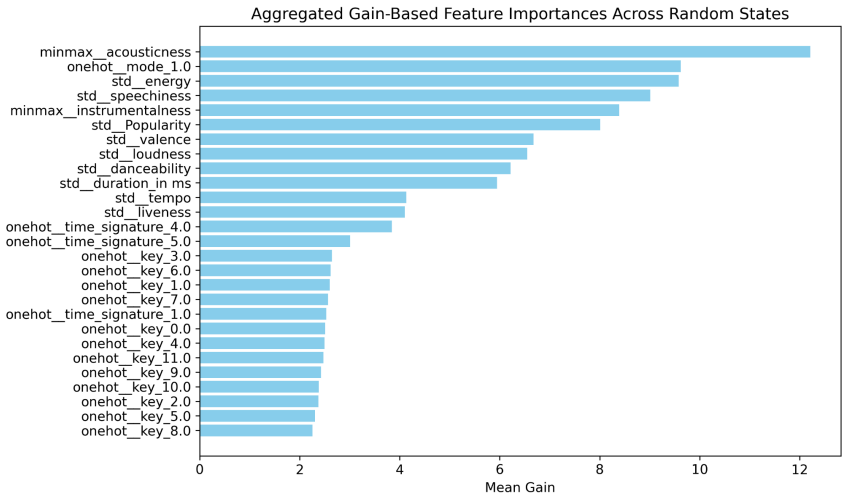


Figure 10. Aggregated gain-based global feature importances for XGBoost.
(Top 5: acousticness, mode, energy, mode, speechiness, instrumentalness)

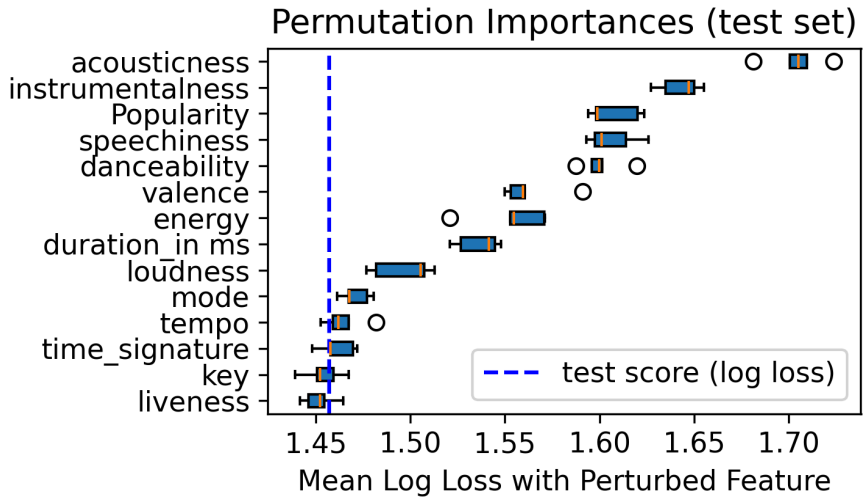


Figure 11. Permutation based global feature importances for XGBoost.
(Top 5: acousticness, instrumentalness, popularity, speechiness, danceability)

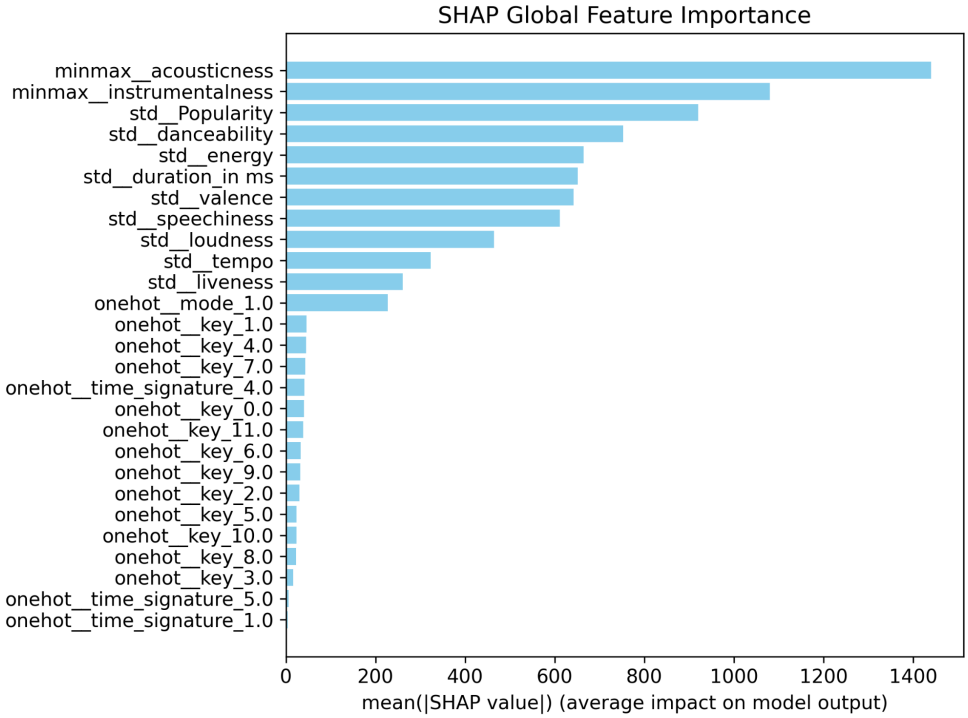


Figure 12. Mean absolute SHAP value based global feature importances for XGBoost.
(Top 5: acousticness, instrumentalness, popularity, danceability, energy)

Three types of global importances are calculated: permutation feature importance based on the drop in model performance due to random perturbation of a feature; XGBoost gain based on the average improvement in cost brought on by a feature; Shap values based on Shapley values from game theory. After aggregating the top 5 most important features from each method, acouticness and instrumentalness stand out to be always in the top 5 and therefore generally the most important features.

4.3 Best Model and Local Importance

Local Feature Importance metrics and visualizations help gauge the contribution of a feature to the predictability of a certain point. In this section, I will take 2 points for class 5 (hiphop) as examples for interpretability. *SHAP values for multi-class XGBoost models report log odds as opposed to predicted probabilities[3].

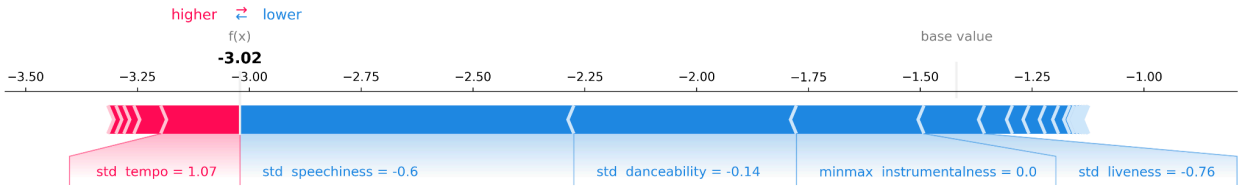


Figure 13. A Local feature importance SHAP values for point 10, class Hiphop. Each feature's importance + expected value sums to the log(odds) of -3.02 for probability of Hiphop.

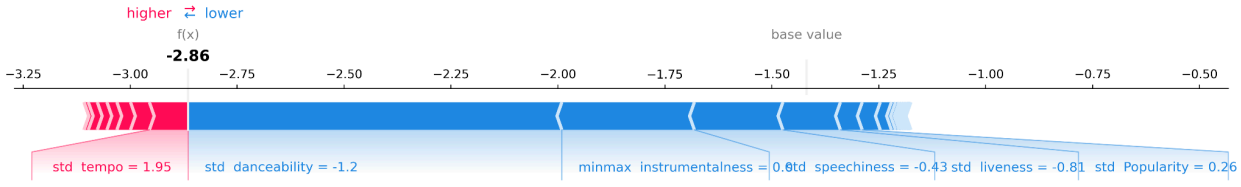


Figure 14. A Local feature importance SHAP values for point 100, class Hiphop. Each feature's importance + expected value sums to the log(odds) of -2.86 for probability of Hiphop.

Low danceability and low speechiness contribute negatively to the probability of class 5, while higher tempo contributes positively. These observations are reasonable as hiphop is often known as verbally intense and highly danceable with rhythmically regular patterns.

Other graphs(class 5):

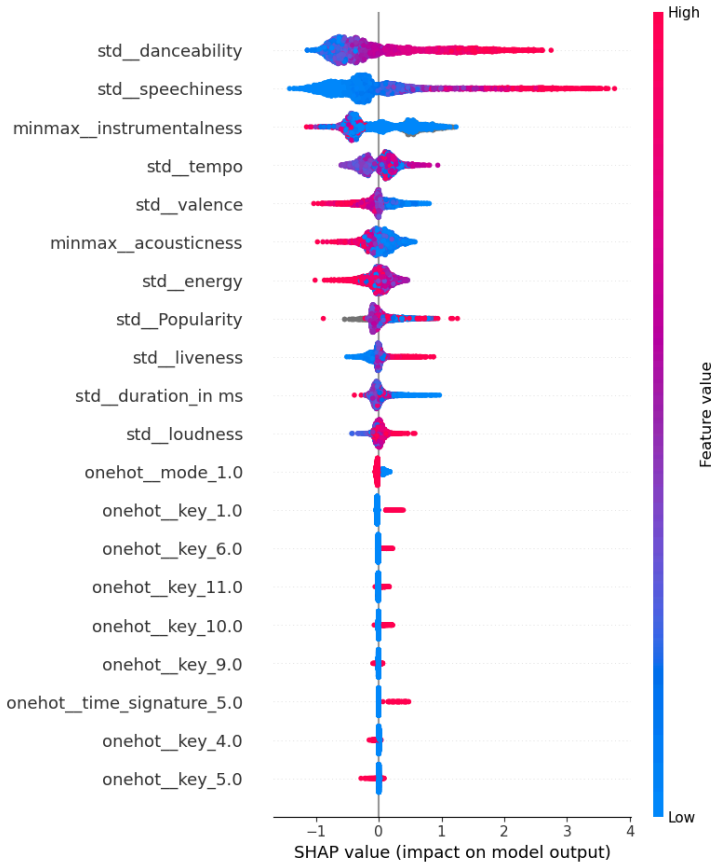


Figure 15. Summary plot for SHAP values with class 5(XGBoost).

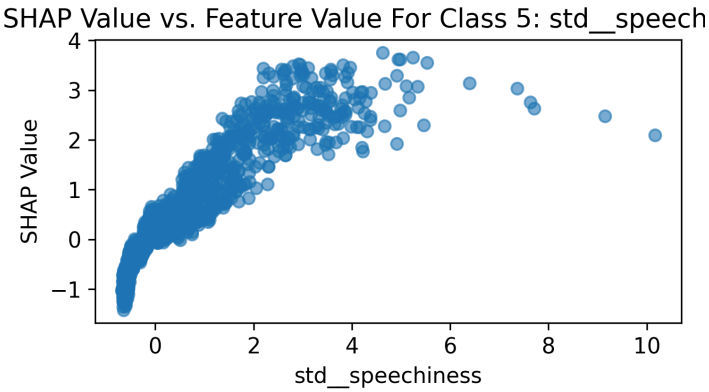


Figure 16. SHAP Value vs. Speechiness with class 5(XGBoost).

5. Outlook

To improve predictive power and interpretability, I have a few ideas for future implementation: Fine-tune hyperparameters by including more possible values; try reduced-features models and compare with XGBoost or multivariate imputation; create interaction terms to capture relationships between features; to improve prediction on genres like Alternative and Indie Alternative, try to collect more data on the production/company side since some are hard to define solely by audio features; experiment with oversampling techniques like SMOTE; expand SHAP analysis by generating more local plots; include additional evaluation metrics as other references or customize and balance an evaluation metric.

6. References

[1]https://www.kaggle.com/datasets/purumalgi/music-genre-classification

[2]https://machinehack.com/hackathons/music_genre_classification_weekend_hackathon_edition_2_the_last_hacker_standing/overview

[3]https://evgenypogorelov.com/multiclass-xgb-shap.html

[4]https://songbpm.com/@nick-cave-the-bad-seeds/deanna--tVDbfraa0

[5]https://xgboost.readthedocs.io/en/latest/faq.html

[6]https://scholarworks.calstate.edu/downloads/73666b68n#:~:text=Music%20genre%20classification%20is%20the,and%20content%2Dbased%20music%20retrieval.