Music Genre Classification Using Spotify Audio Features

name: Mike

Data Loading and Reproducibility

I loaded the Spotify music dataset containing approximately 50,000 songs across ten genres. To ensure reproducibility, I initialized the random number generator using a fixed seed based on my N-number before performing any data processing or model training.

Data Cleaning

I first inspected the dataset for data quality issues. Duplicate rows were identified and removed. Songs with invalid duration values (negative duration_ms) were excluded. The tempo variable, which was originally stored as a string, was converted to a numeric format, resulting in missing values that needed to be handled in later preprocessing steps.

Feature Selection and Cleaning

I removed non-informative or identifier-based columns, including **artist name, track name, instance ID, and obtained date**, as these fields do not contribute directly to genre classification. The remaining features consisted of numerical audio attributes and categorical musical descriptors.

Handling Missing Values

Missing values introduced during preprocessing, primarily from the tempo variable, **were handled using median imputation**. Median imputation was chosen because many audio features exhibit skewed or non-normal distributions, making it more robust than mean imputation.

Encoding Categorical Variables

Categorical variables were converted into numerical form. The musical key was transformed using **one-hot encoding with the first category dropped to avoid multicollinearity**, and all resulting boolean columns were converted to integer format. The musical mode was encoded as a binary variable, with Major mapped to 1 and Minor mapped to 0. After these transformations, all features were numeric and suitable for modeling.

Train–Test Split

To avoid data leakage and ensure fair evaluation, I performed a genre-stratified train–test split. For each genre, exactly 500 songs were randomly selected for the test set, resulting in a balanced test set of 5,000 songs. All remaining songs were used for training.

Feature Scaling and Dimensionality Reduction

I first examined the relationships among features by constructing a correlation matrix, which revealed a strong correlation between *loudness* and *energy*. This indicated the presence of redundancy in the feature space. To address this, I constructed a preprocessing pipeline that included median imputation, feature standardization, and principal component analysis

(PCA). PCA was applied after scaling to reduce dimensionality while preserving approximately 90% of the total variance, resulting in a reduced feature space of 17 principal components.

Low-Dimensional Visualization and Clustering

I visualized the training data in the lower-dimensional PCA space using the first two principal components to examine genre structure. In addition, I applied KMeans clustering and used the elbow method to explore potential cluster structure and assess how genres overlap in the reduced feature space. The elbow analysis suggested that $k = 10$ is a reasonable choice, which aligns with the number of genres in the dataset.

Comment:

The visualization of genres in the lower-dimensional PCA space reveals that genre structure is only partially preserved after dimensionality reduction. While certain genres, such as Classical, form relatively compact and distinguishable clusters, many genres exhibit substantial overlap in the reduced feature space. In particular, closely related genres such as Hip-Hop and Rap, as well as Jazz and Blues, occupy highly overlapping regions, suggesting strong similarity in their underlying audio characteristics. Although the elbow method indicates that k=10-12 is a reasonable choice consistent with the number of genres in the dataset, the resulting clusters are not cleanly separable. This suggests that genre boundaries are inherently fuzzy rather than well-defined, even in a lower-dimensional representation. Overall, while dimensionality reduction helps reveal the overall structure, significant overlap between genres persists. This helps explain the persistent misclassification observed in downstream classification models

---

Model Training

Using the PCA-transformed features, I trained and compared multiple classification models to evaluate their ability to distinguish music genres. All models were trained on the same processed training data and evaluated on an identical test set to ensure a fair comparison. Model performance was assessed using the macro-averaged ROC curve and AUROC.

Decision Tree

A decision tree classifier was trained as a baseline model to establish a reference level of performance. To control model complexity and reduce overfitting, the maximum tree depth was limited to 10 and the minimum number of samples required at each leaf node was set to 50. The model achieved a macro-averaged AUROC of **0.82**.

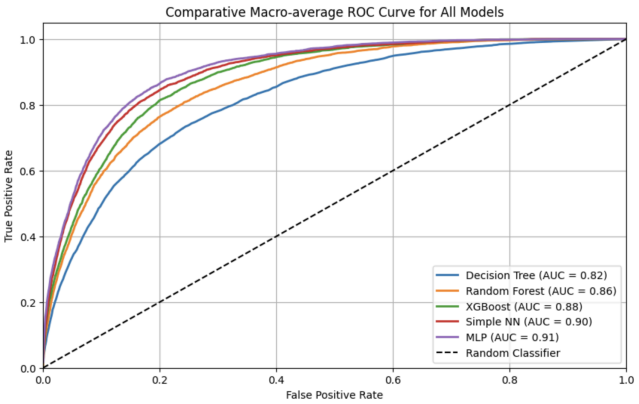
Random Forest
To improve upon the single decision tree, a random forest classifier was trained using an ensemble of 100 decision trees. The maximum tree depth was limited to 10, with a minimum of 50 samples per leaf to control overfitting. This ensemble approach reduced variance and improved generalization, resulting in a macro-averaged AUROC of **0.86**.

XGBoost
An XGBoost classifier was trained to further enhance performance through gradient boosting, which sequentially builds trees to correct errors from previous iterations. The model was configured for multi-class classification with probabilistic outputs, a maximum depth of 10, and a minimum child weight of 50 to control complexity. This model achieved a macro-averaged AUROC of **0.88**.

Simple Neural Network (One Hidden Layer MLP)
A simple multi-layer perceptron with one hidden layer was trained to introduce non-linear modeling capacity beyond tree-based methods. The network consisted of a single hidden layer with 100 neurons, using the ReLU activation function and the Adam optimizer. Early stopping was enabled to prevent overfitting. This model achieved a macro-averaged AUROC of **0.90**.

Multi-Layer Perceptron (MLP)
Finally, a deeper multi-layer perceptron was trained to capture more complex non-linear relationships in the data. The network consisted of three hidden layers with 100, 50, and 25 neurons, respectively, using ReLU activation and the Adam optimizer. Early stopping was applied to improve convergence and generalization. This model achieved the best overall performance, with a macro-averaged AUROC of **0.91**, and was selected as the final classifier.

Model Evaluation
Model performance was evaluated on the held-out test set using macro-averaged ROC curves and AUC, which are appropriate for balanced multi-class classification problems. ROC curves were computed by binarizing class labels and aggregating performance across all genres.



--- Comparative AUROC Scores ---
Decision Tree AUROC: 0.82
Random Forest AUROC: 0.86
XGBoost AUROC: 0.88
Simple Neural Network AUROC: 0.90
MLP AUROC: 0.91

Model Comparison and Final Selection

I compared the AUROC scores across all evaluated models to assess their relative performance. The decision tree achieved the lowest performance, followed by the random forest and XGBoost models, which benefited from ensemble and boosting strategies but remained limited in capturing complex non-linear relationships. Neural network models performed better overall, reflecting their greater modeling flexibility. In particular, the multi-layer perceptron (MLP) with three hidden layers achieved the highest macro-averaged AUROC of approximately **0.91**. The deeper architecture allowed the model to capture more complex interactions among audio features, while early stopping helped prevent overfitting and ensured good generalization. Based on its superior performance and stable training behavior, the MLP was selected as the final classification model.

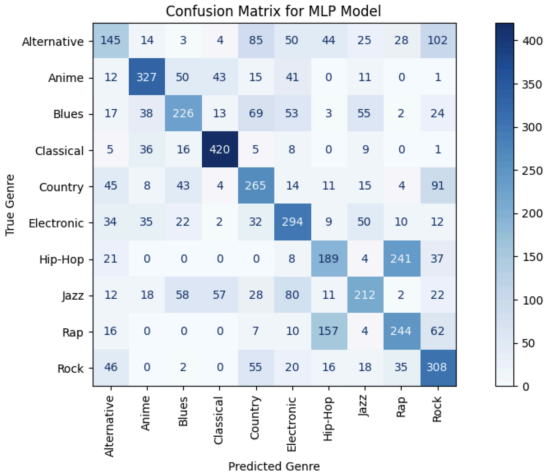
Most important factor

I believe data preprocessing is the key factor behind successful classification. Throughout the project, I experimented with multiple modeling approaches and discovered that model performance was significantly constrained by data quality. In early attempts, deficiencies in the preprocessing stage such as inadequate missing value handling, improper feature scaling, or classification encoding issues which led to subpar performance regardless of the model used. Only after thoroughly resolving these issues did model architecture optimization and hyperparameter tuning translate into substantial performance gains. This experience demonstrates that model quality is fundamentally determined by preprocessing quality, while hyperparameter tuning though a secondary factor still significantly impacts final performance.

Extra Credit: Confusion Matrix Insights

An informative non-trivial observation emerges from the confusion matrix of the final MLP model. While the model performs strongly for well-defined genres such as *Classical*, *Rock*, and *Anime*, systematic misclassifications appear between stylistically adjacent genres. In particular, a substantial number of *Hip-Hop* tracks are misclassified as *Rap*, and vice versa, indicating that these two genres occupy highly overlapping regions in the feature space. This confusion is asymmetric, with Rap tracks more frequently predicted as Hip-Hop than the reverse, suggesting that Hip-Hop may represent a broader acoustic category in the learned representation.

Similar patterns are observed between *Jazz* and *Blues*, as well as between *Alternative* and *Rock*, where misclassifications tend to occur along musically intuitive boundaries rather than randomly. These observations are consistent with the PCA visualization, which showed significant overlap among these genre pairs in the reduced-dimensional space. Together, the confusion matrix and low-dimensional visualization suggest that classification errors are driven more by intrinsic genre ambiguity than by model limitations, reinforcing the conclusion that feature representation and genre definitions fundamentally constrain performance.



Final multi-layer perceptron (MLP) with three hidden layers AUROC (MLP): 0.91, so it was selected as the final classification model.