Multi-class Music Genre Classification:
Spotify Data

Hamza Alshamy

May 2024

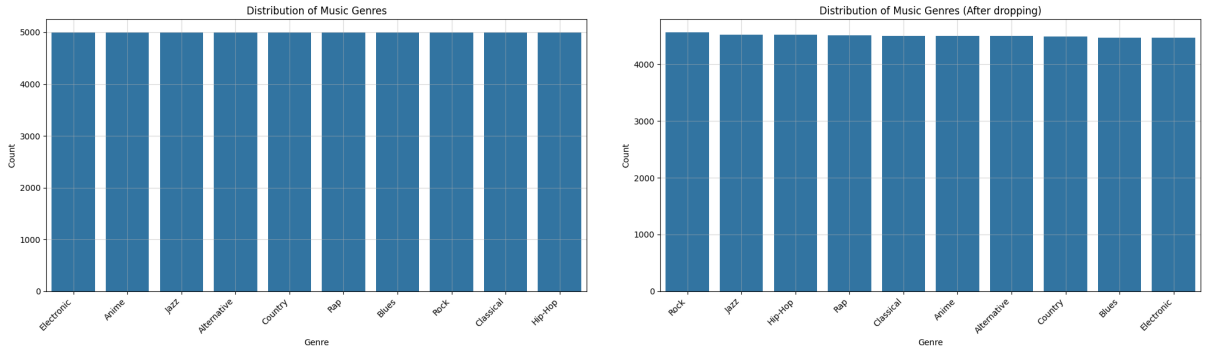
1 Pre-processing and Exploratory Data Analysis (EDA)

This section outlines the pre-processing steps taken to prepare the dataset for multi-class music genre classification.

Initialization and reproducibility: First, the seed for random number generation was set to my unique N-number to ensure reproducibility. This setting was applied globally, including any functions accepting a *random state* parameter.

Data cleaning:

- *Unnecessary columns* (*instance_id*, *artist_name*, *track_name*, and *obtained_date*) were removed as they do not contribute to genre classification.
- Five rows containing NaN values across all features were eliminated.
- The *tempo* feature included 4980 entries marked with "?". The rows with the value were dropped as the distribution of genres remained uniform as can be seen in Figure 1



(a) Distribution of Music Genres

(b) Distribution of Music Genres (After dropping)

Figure 1: Genre Distribution Before and After Dropping "?" in *tempo* feature

- *Duration* featured 4460 negative values, which are illogical. Therefore, in order not to disturb the uniform distribution of genres, the mean and median of each genre were calculated and then imputed the median to Exclude outliers. Figure 2 shows how the mean and median of each genre.

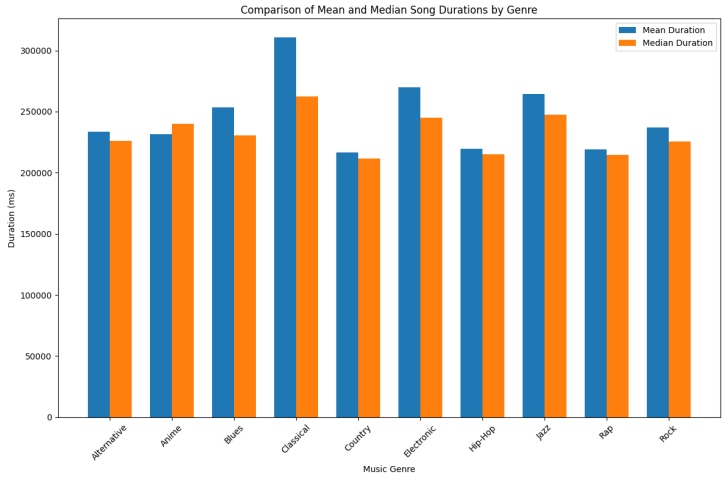


Figure 2: Comparison of Mean and Median Song Durations by Genre

Categorical feature handling:

- The *mode* column was transformed into a binary format and renamed to *major*.
- The *key* column was numerically encoded with values ranging from 1 (A) to 12 (G#).
- Genres were encoded numerically from 1 (Alternative) to 10 (Rock) in the *music_genre* column.

Standardization: All features were standardized except for the binary *major*, ordinal *key*, and the ordinal outcome variable, *music_genre*.

Train/test split: The dataset was manually split to ensure uniform representation across genres, preventing any overrepresentation or underrepresentation in the training or testing sets. The `train_test_split` function was utilized to allocate 500 songs per genre to the testing set, ensuring each genre was equally represented. The remainder was allocated to the training set.

2 Dimensionality Reduction: Linear Discriminant Analysis (LDA)

Given the supervised nature of the genre classification task, where each song is labeled with a specific genre, Linear Discriminant Analysis (LDA) was selected as the dimensionality reduction method over Principal Component Analysis (PCA) or t-SNE for two main reasons. First, unlike PCA which aims to maximize variance accounted for, LDA aims to maximize class separability. Second, LDA leverages the class labels, making it suitable for a supervised learning task. LDA was applied to the standardized data. Figure 3 shows the explained variance ratio by each discriminant.

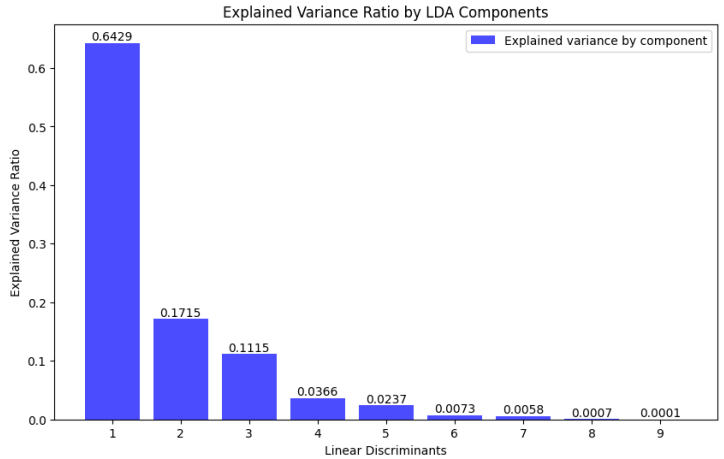
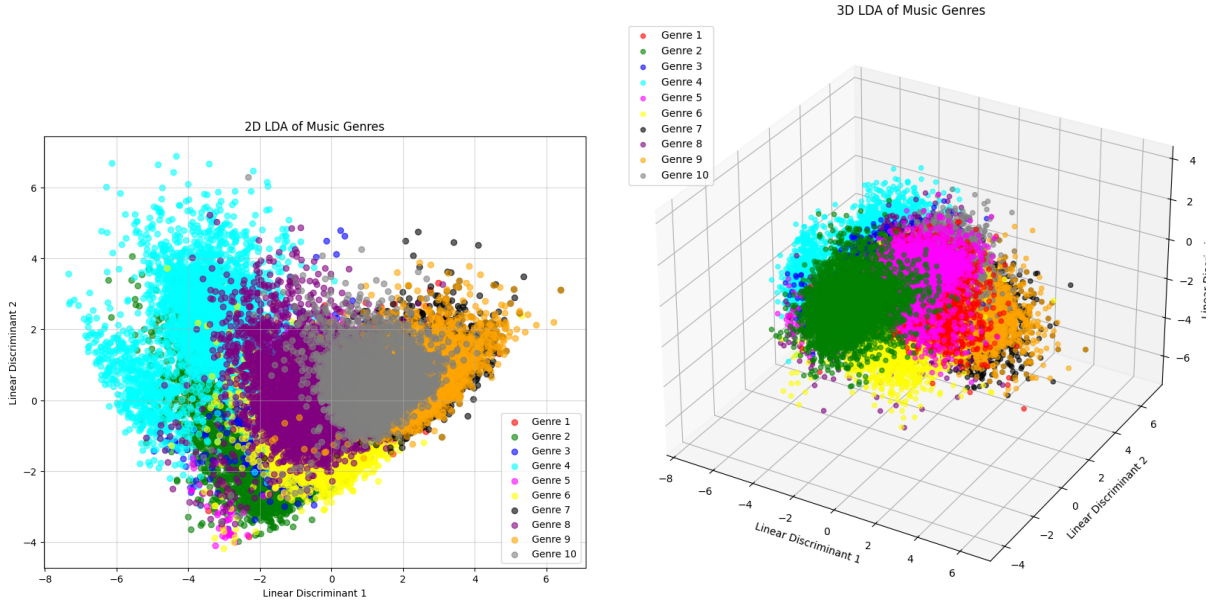


Figure 3: Explained Variance Ratio by LDA Components

Lastly, Figure 4 shows the projection of LDA results, visualized in both two and three dimensions. The visualizations in Figure 4 simplify the genre distinctions by projecting the data onto either the two or three most significant linear discriminants.



(a) 2D LDA of Music Genres

(b) 3D LDA of Music Genres

Figure 4: Projection of LDA Discriminants onto 2D and 3D

3 K-means Clustering

This section details the clustering process and discusses the insights obtained from the cluster visualization.

Since we know the number of genres, there was no need to conduct silhouette analysis to identify the number of clusters for K-Means. Therefore, the algorithm was initialized with 10 clusters, corresponding to the number of music genres. The algorithm was fitted to the LDA-transformed data, which provided a reduced-dimensional space for identifying distinct groupings based on genre characteristics. Figure 5 visualizes K-means clustering onto three LDA discriminants.

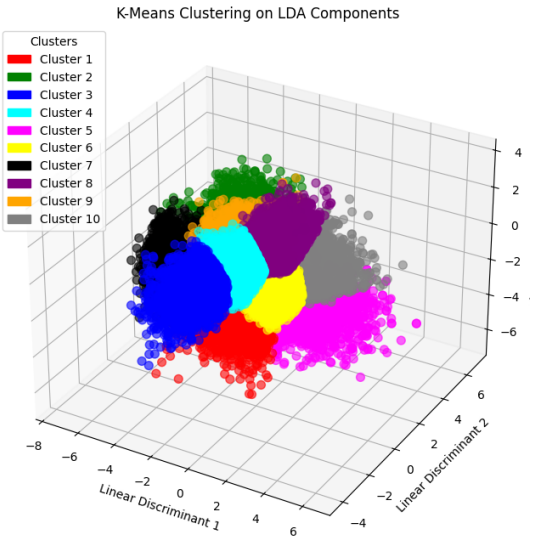


Figure 5: K-Means Clustering on LDA Components

Generally, the spatial arrangement of the clusters could be better in the sense that more space separating the classes would be desirable. This indicates some overlapping musical features across genres.

4 Classification 1: Support Vector Machines (SVM)

The first classification model applied was SVM using the LDA data. Using the (SVM) model with the `LinearSVC` class from `sklearn`, hyperparameter tuning was conducted to find the optimal value of the regularization parameter C which controls the model's tolerance for misclassification. The optimal C value, determined to be 10, was based on maximizing accuracy through cross-validation.

Table 1 summarizes the main metrics for the model's performance. The macro-average ROC curve of 0.83 is moderately high. Further, given that the baseline of random guessing is $\frac{1}{10}$ (10 genres) and accuracy and test accuracy is 0.446, we can say the model is better than random guessing and has discriminatory power.

Model	Training Accuracy	Test Accuracy	Avg. ROC	Avg. Precision	Avg. Recall
SVM	0.445	0.446	0.83	0.43	0.43

Table 1: SVM Model Performance Metrics

Lastly, Figure 6 shows the ROC for each genre as well as the macro-average ROC curve. The model seems inconsistent in its discriminatory power. That is, the model classifies some genres better than others.

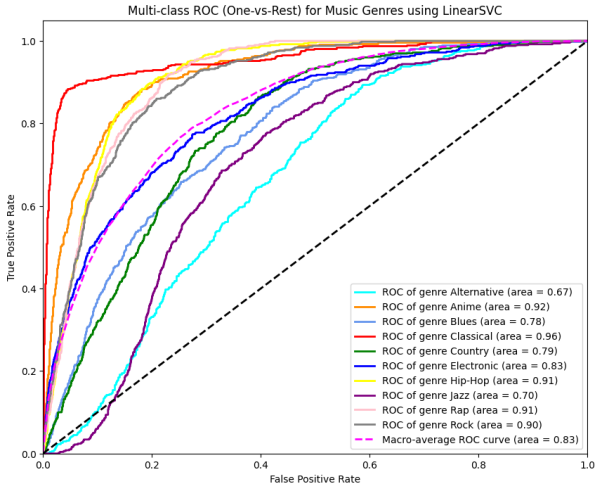


Figure 6: Multi-class ROC for Music Genres using LinearSVC

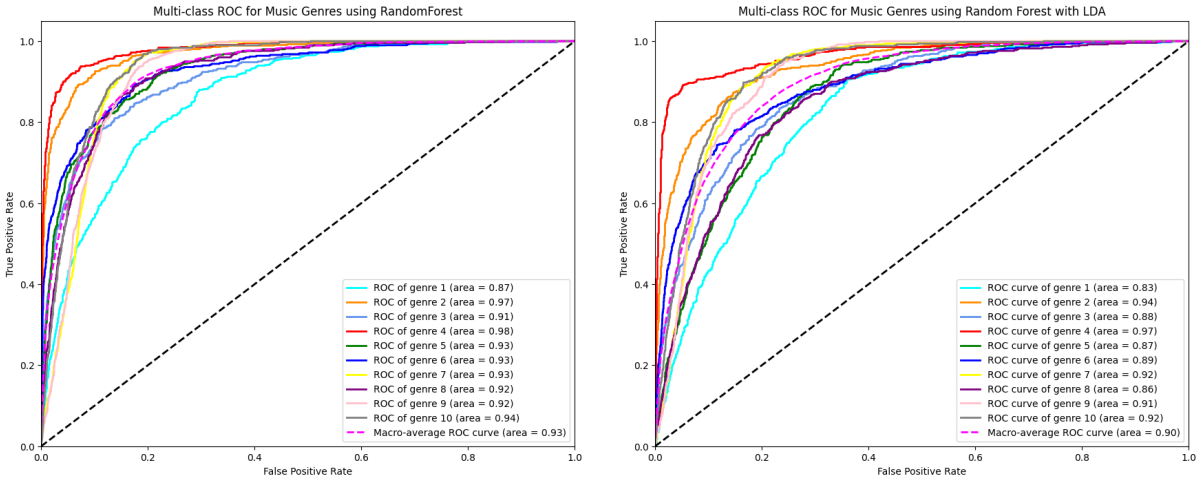
5 Classification 2: Random Forests

Random Forests were employed for the second classification method. The initial instinct was to use the standardized data rather than LDA-transformed data. The primary reason for this choice is the intrinsic nature of Random Forests to handle feature interactions and non-linear relationships. Unlike LDA, which reduces dimensionality by projecting data onto a direction that maximizes class separation, Random Forests can make use of the complete feature set to improve accuracy. In any case, both models were built and the models' ability to classify diabetes was optimized by tuning the `max_depth` parameter, which controls the complexity of the decision tree. The criterion for selecting the best depth was based on the accuracy score. Table 2 summarizes the metrics for both models.

Model	Training Accuracy	Test Accuracy	Avg. ROC	Avg. Precision	Avg. Recall
Random Forest	0.705	0.586	0.93	0.59	0.59
Random Forest + LDA	0.535	0.502	0.9	0.5	0.5

Table 2: Performance Metrics for Random Forest Models

As can be seen from Table 2, the Random Forest Model using only standardized data and not LDA-transformed data performs better in all metrics. Further, Figure 7 shows the ROC curves for both models and once again, the Random Forest model without LDA-transformed data performed better across all genres and macro-average ROC curve. This Random Forset model without LDA also outperforms SVM. This confirms the intuition that using non-transformed data on Random Forest performs better.



(a) Multi-class ROC for Music Genres using Random Forest

(b) Multi-class ROC for Music Genres using Random Forest with LDA

Figure 7: Projection of LDA Discriminants onto 2D and 3D

6 Classification 3: AdaBoost

The last classification model, AdaBoost model was implemented using the `AdaBoostClassifier` from `sklearn`. A `GridSearchCV` approach was employed to identify the optimal depth within a range of 1 to 10.`max_depth` of 10 was selected for the base estimator. Furthermore, the number of estimators was set to a modest 100. Table 3 shows the metrics for the model. The model preforms similar to the Random Forest model, which shows great discriminatory abilities.

Model	Training Accuracy	Test Accuracy	Avg. ROC	Avg. Precision	Avg. Recall
AdaBoost	0.632	0.525	0.91	0.53	0.52

Table 3: Performance Metrics for the AdaBoost Model

Lastly, the ROC curve in Figure 8 shows that the model has great discriminatory abilities across all genres, better than SVM and similar to the Random Forest models.

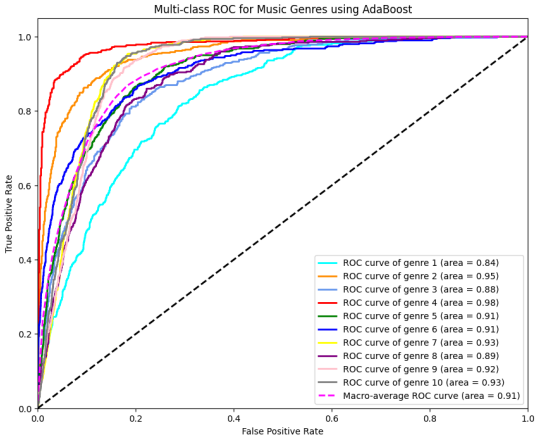


Figure 8: Multi-class ROC for Music Genres using AdaBoost

7 Summary

Table 4 summarizes the metrics for all four models in order to compare and choose the best model. As can be seen, using Random Forest without LDA dimensionality reduction resulted in the best discriminatory power across all genres and metrics. Therefore, this is the classification method of choice. Observe that AdaBoost follows, then Random Forst + LDA, and lastly SVM.

Model	**Training Accuracy**	**Test Accuracy**	**Avg. ROC**	**Avg. Precision**	**Avg. Recall**
SVM	0.445	0.446	0.83	0.43	0.43
Random Forest	0.705	0.586	0.93	0.59	0.59
Random Forest + LDA	0.535	0.502	0.9	0.5	0.5
AdaBoost	0.632	0.525	0.91	0.53	0.52

Table 4: Performance Metrics for Classification Models