Spotify Music Genre Classification with Machine Learning

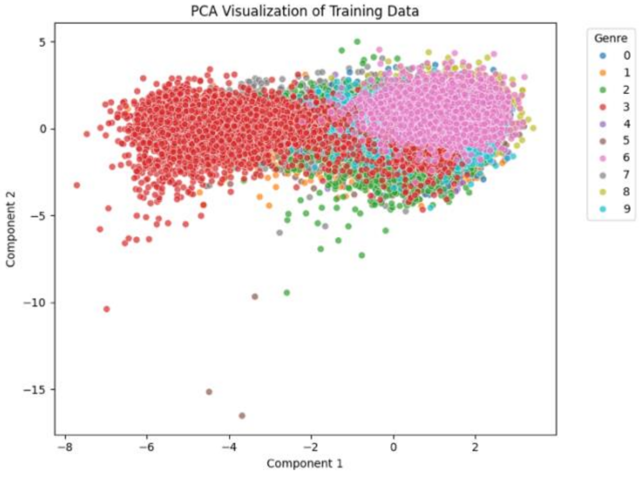
The project aims to classify music genres based on a set of numerical and categorical audio features using machine learning techniques. Given the challenges of missing values, mixed data types, and the high dimensionality of the feature space, the pipeline integrates preprocessing, dimensionality reduction, clustering, and classification. The goal is to build an interpretable, accurate multi-class model and evaluate its performance through AUC and visual diagnostics.

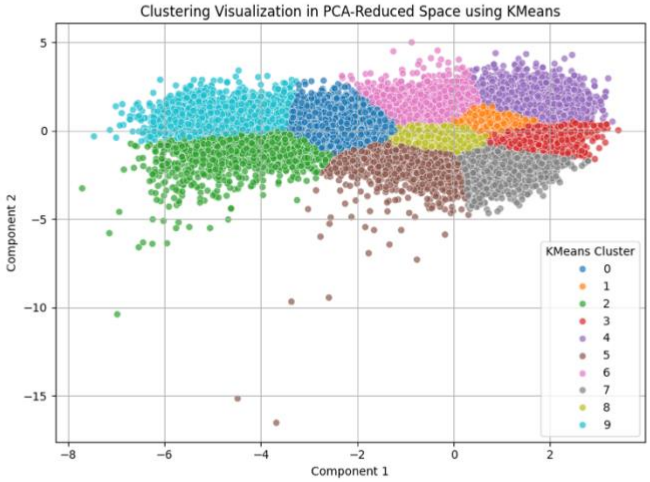
Data Preprocessing

The dataset includes features like loudness, danceability, tempo, key, mode, and so on, with randomly missing values such as "-1" and "?". These were first converted to NaN and handled through mean imputation, preserving useful samples while ensuring consistency across the input features. And since machine learning models cannot handle strings directly, the "key" feature was converted from string values to numeric codes using .astype('category').cat.codes. Similarly, the "mode" column, a categorical variable, was one-hot encoded to make it usable in machine learning algorithms. For the target variable, "music_genre", label encoding was applied to transform the textual genre labels into numeric class labels using LabelEncoder(), enabling multi-class classification. In addition, I also removed columns like artist name and track name because they are high-cardinality text fields that are not required and may introduce noise rather than helpful patterns into the model. Then, to handle class imbalance, I stratified the dataset by genre and reserved 500 samples per genre randomly for the test set and the other 4500 samples for training set. This ensured balanced evaluation across all classes, preventing the classifier from being biased toward dominant genres. And I normalized only the numerical features and excluded categorical variables like "mode" from scaling to avoid distorting their structure during dimensionality reduction.

Dimensionality Reduction and Clustering

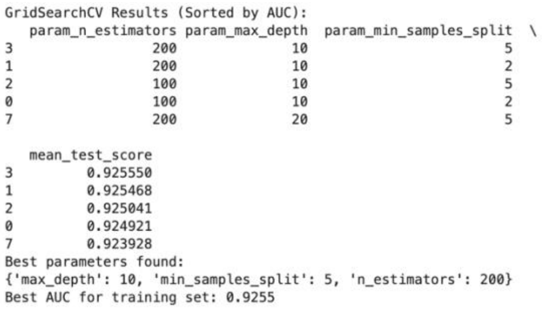
Before training, I applied Principal Component Analysis (PCA) to reduce dimensionality and visualize structure in the data. PCA was performed on the scaled numerical features only, preserving relationships among them while ensuring comparability. I reduced the data to 2 principal components, enabling clear 2D visualization of genre distribution.





The PCA scatterplot revealed partial genre separation. To complement this, I applied KMeans clustering with 10 clusters. The KMeans clustering in PCA space revealed meaningful separability, suggesting that even unsupervised methods can identify genre-like groupings.

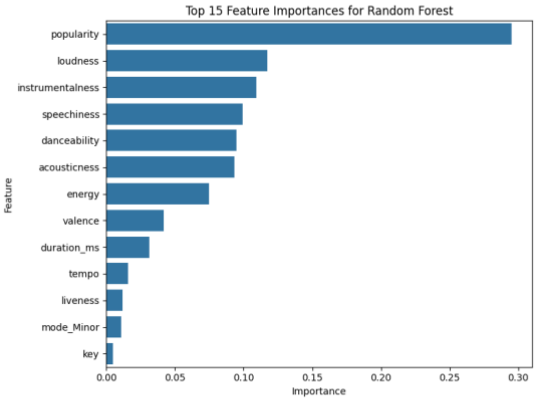
Model Training and Hyperparameter Fine-Tuning



```
GridSearchCV Results (Sorted by AUC):
   param_n_estimators param_max_depth  param_min_samples_split  \
3                 200              10                        5
1                 200              10                        2
2                 100              10                        5
0                 100              10                        2
7                 200              20                        5

   mean_test_score
3         0.925550
1         0.925468
2         0.925041
0         0.924921
7         0.923928
Best parameters found:
{'max_depth': 10, 'min_samples_split': 5, 'n_estimators': 200}
Best AUC for training set: 0.9255
```

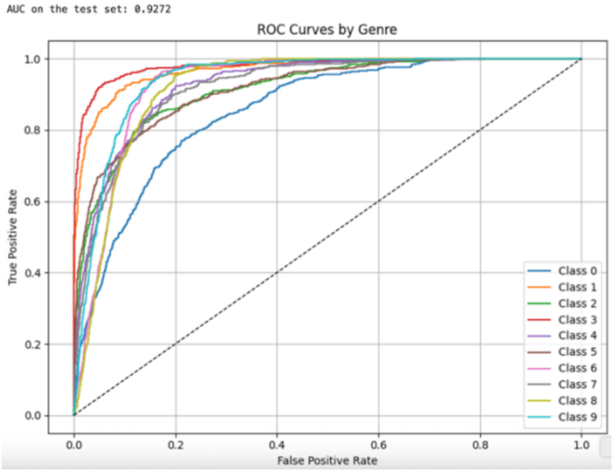
For classification, I selected the Random Forest because it is well-suited for high-dimensional, mixed type data and can model non-linear relationships without requiring extensive preprocessing. Its ensemble structure reduces overfitting, and its built-in feature importance makes the model both powerful and interpretable, making it ideal for understanding which audio features drive genre classification. And a grid search was conducted over 3 hyperparameters: n_estimators, max_depth, and min_samples_split. I used 3-fold cross-validation and optimized the model using the roc_auc_ovr metric, appropriate for multi-class settings. The best model used 200 trees, a maximum depth of 10, and a minimum samples split of 5. This model achieved a cross-validated AUC of 0.9255, indicating strong internal generalization.



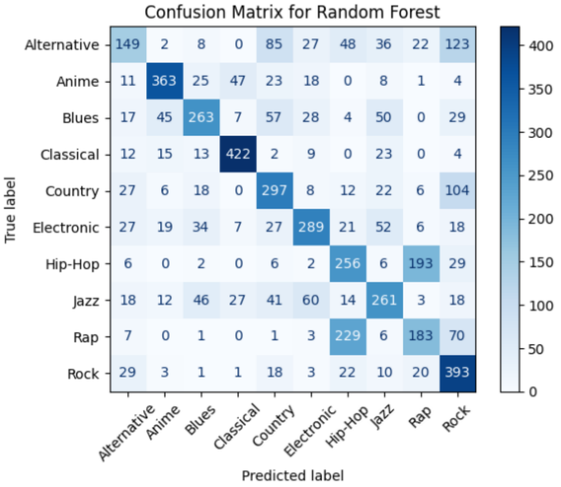
To understand which features contributed most to classification, I also plotted the top 15 features based on the Random Forest's importance scores. The top features were popularity, loudness, and instrumentalness. This implies that popularity capturing mainstream and energy-related characteristics played crucial roles in genre prediction.

Model Evaluation and Results

On the test set, the final best model achieved an AUC of 0.9272, confirming that it generalized well. ROC curves were plotted for each class, and results showed strong separation for genres like Class 3 and Class 1, while genres like Class 0 were harder to distinguish.



```
AUC on the test set: 0.9272
```



To further analyze performance, I plotted a confusion matrix. Misclassifications often occurred between similar genres. For example, Rap and Hip-Hop were frequently confused, suggesting either overlapping feature distributions or genuine genre similarity.

Therefore, the most important factor underlying my classification success was the strong predictive signal in the audio features themselves, particularly variables like popularity, loudness, and instrumentalness, which captured genre-specific characteristics effectively. Combined with careful preprocessing and the robustness of the Random Forest model, this allowed the classifier to achieve high accuracy and generalize well across all music genres.