Capstone Project Report: Music Genre Classification
by Haojie Cai

Objective:

The goal of this capstone project is to develop a multi-class classification model that predicts the genre of a song using the audio features provided by Spotify, such as 'popularity', 'key', and 'tempo'.

Data Cleaning and Preprocessing:

The dataset includes 50005 samples/songs, and it contains multiple inconsistencies:

- **Missing values**: There are 5 rows that are missing (NaN in each column), which means these 5 rows are useless. After dropping them, we have 50000 songs.
- **Invalid numeric values**: There are 4980 songs with missing tempo (filled with '?'), 4939 songs with missing duration (filled with '-1'), and 47 songs with invalid loudness (value greater than 0). Since this is a large number of missing songs, about 10% of the total songs, I imputed these invalid values with genre-wise median values to help preserve the structure of each genre.
- **Unnecessary columns**: Linguistic features like 'instance_id', 'artist_name', 'track_name' and 'obtained_date' are unnecessary for predicting the genres, so I directly dropped these columns.
- **Categorical predictors**: For features like 'mode' and 'key' are categorical. For these two features, I dummy encoded them into numerical data: for 'mode', 1 represents 'Major' and 0 represents 'Minor'; for 'key', I dropped first key 'A' and added columns for all other keys, so that 1 in a specific key column represents the song is in that key, and 0 in all key columns represents the song is in key 'A'.
- **Category outcome**: For the outcome music genre, I converted it into integer labels (0-9 for the 10 genres).

After fixing these inconsistencies, I checked that all data is in numeric format. Then, I standardized each non-binary predictors with Z-score normalization. For the 50000 songs, there are 5000 songs in each genre. Thus, I randomly split 500 songs in each genre as the test set, and 4500 songs in each genre as the training set. As a result, there are 5000 songs for testing, and 45000 songs for training.

Dimensionality Reduction and Visualization before Modeling:

Principal Component Analysis (PCA) was applied to reduce dimensionality and help visualize genre separability. I first applied PCA on the training data, and visualized the 2D PCA (figure 1) and 3D PCA (figure 2) plots, colored by their true genres labels. I also applied k-means clustering and used the silhouette score to find the optimal number of clusters. The results were visualized in figure 3.

From the figures, we can clearly see that there are strong overlaps between each class. From the silhouette scores, k-means can only recognize 2 clusters. Both of them suggest that genres share acoustic similarities under the raw PCA data.

Modeling:

Since this is a multi-class classification task, and the data is relatively large and noisy, I chose not to use SVC. Instead, I explored Random Forest Classifier, AdaBoost Classifier, and Neural Network.

- **Random Forest**: For n_estimators in [50, 100, 200, 300], for criterion in ['gini', 'entropy'], for max_features in [None, 'sqrt', 'log2'], for max_depth in [None, 10, 15, 30, 50], for min_samples_split in [5, 10, 20], and for min_samples_leaf in [2, 4, 10, 20], I tested each

combination to build the Random Forest Classifier with processed data, and used the best parameters to fit the Random Forest Classifier with data reduced to varying cumulative numbers of PCA components and found the best AUC among them.

- **AdaBoost**: For n_estimators in [50, 100, 200], and for learning_rate in [0.01, 0.1 , 1], I tested each combination to build the Random Forest Classifier with processed data, and used the best parameters to fit the AdaBoost Classifier with data reduced to varying cumulative numbers of PCA components and found the best AUC among them.
- **Neural Network**: I trained a four layer multilayer perceptron, with first hidden dimension in [128, 256, 512], tried learning rate in [1e-3, 5e-4], dropout in [0.1, 0.2, 0.3], and weight decay in [1e-3, 5e-4, 1e-4]. I also trained a six layer neural network with residual connection, and tried different combinations of learning rate, dropout, and weight decay. Moreover, I trained the models with data reduced to varying cumulative numbers of PCA components, and found the best AUC among them.

Evaluation and Visualization of Latent Space:

Among all the models, the model with the highest AUC value of 0.9413 is the 4-layer MLP with first hidden dimension of 256, dropout of 0.3, learning rate of 1e-3, weight decay of 5e-4, and training epochs of 200, trained with data reduced to 19 components. Its ROC curve is visualized in figure 4.

To further visualize how the model separates the data for classification, I took the latent representation of data at layer 3 in my best model, which contains 64 features, and applied PCA on it to visualize the 2D PCA (figure 5) and 3D PCA (figure 6) plots, colored by their true genres labels. Then, I used k-means clustering and silhouette score to find the optimal number of clusters, and visualize the optimal 2D k-means clustering (figure 7) and 3D k-means clustering (figure 8).

In the latent space, we can clearly see that the data after MLP is much more separable compared to the original data. Some classes that are covered by other classes in the PCA before modeling have been separated out. For example, for class 1 with blue color, it can now be clustered by k-means clustering. From the silhouette scores, the optimal k-means has 7 clusters, which suggests that it's much easier to recognize each cluster now with the data in latent space.

Conclusion:

This project demonstrates that Spotify audio features contain sufficient information for effective multi-class music genre classification. By combining preprocessing, dimensionality reduction, and model tuning, I achieved an AUC of 0.9413. The most important factor that underlies my classification success is the use of the neural network to separate the representations of each class in the latent space. While the raw PCA projection shows overlapping genre distributions, the MLP can transform the input data into a higher-dimensional and separable latent space. This makes a good AUC performance.

AUC: 0.9413

(figures next page)

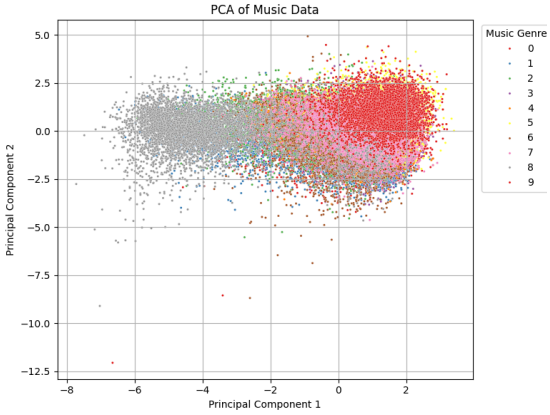

Figure 1

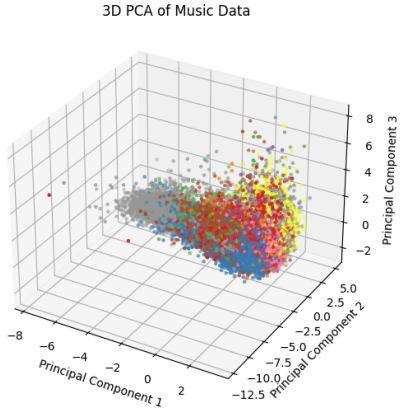

Figure 2

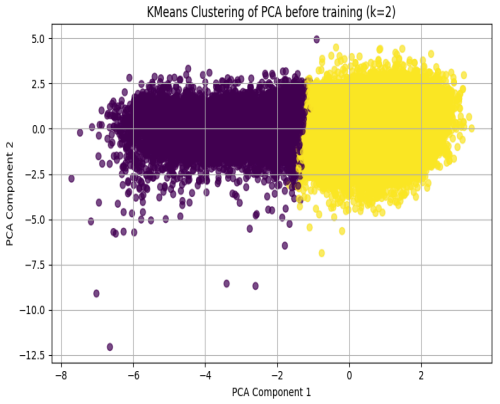

Figure 3

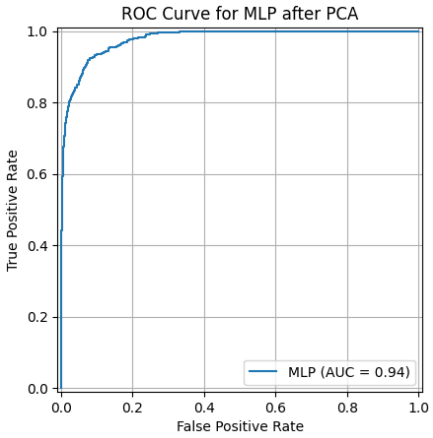

Figure 4

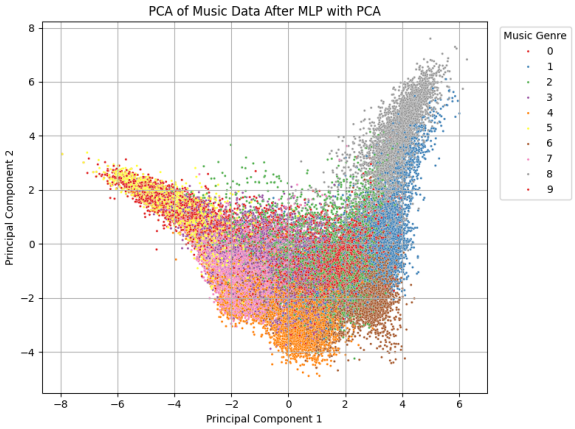

Figure 5

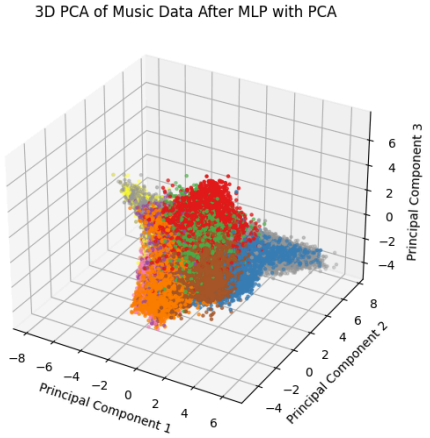

Figure 6

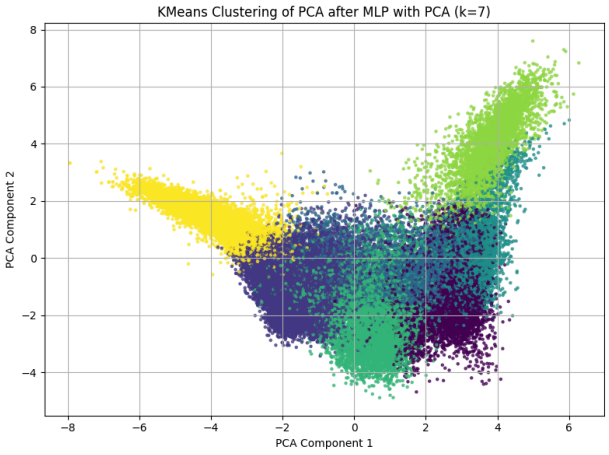

Figure 7

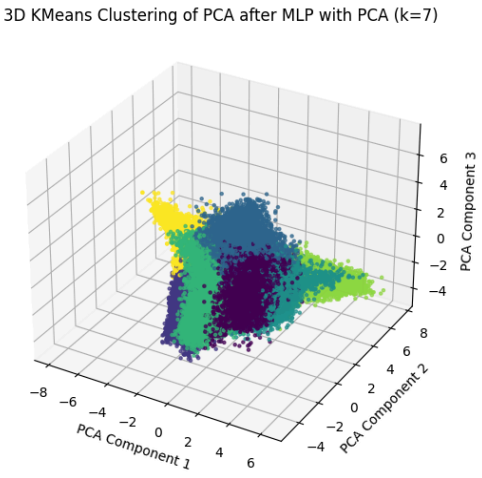

Figure 8