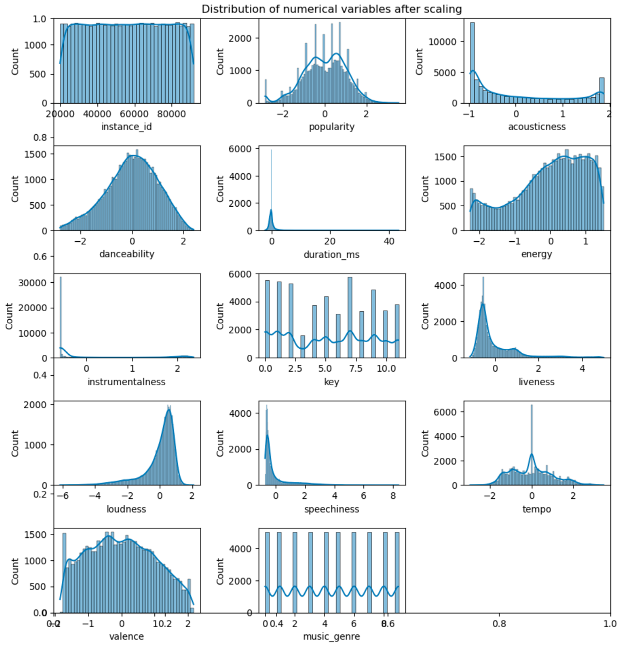
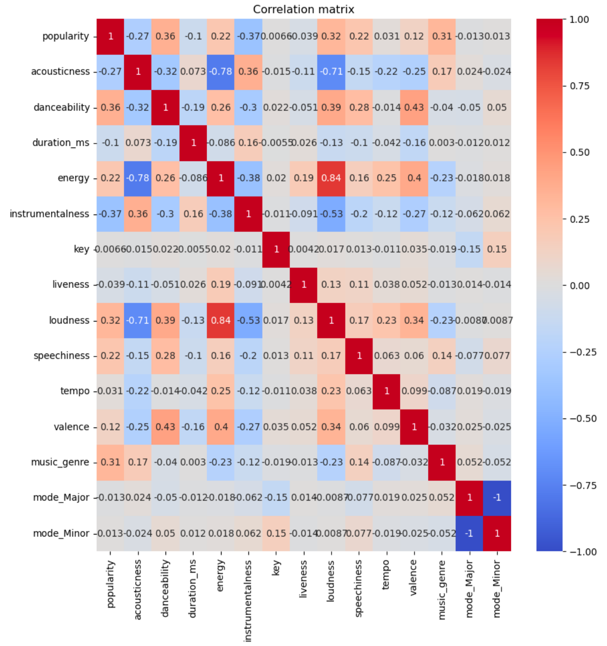
Capstone Project: Spotify Song Genre Classification
Krysten Nguyen

1. Preprocess data

After loading in the music data from Spotify, I first check for N/A rows and drop these. Next, I process non-numerical variables in the dataset. First, I convert 'key' of the songs currently in string into numerical value from 0 to 11, representing all notes raised by a semitone starting from C. Second, I one-hot encode 'mode', which indicates whether the the song is in minor or major scale, into 2 features 'mode_Major' and 'mode_Minor'. Lastly, I create a numerical mapping for the 10 genres so that it can be used as labels in classification models. Then, I impute the missing values in 'durations' (placeholder: '-1.0') and 'tempo' (placeholder: '?') with KNN imputer, using 2 neighbors and uniform distance.
Plotting the distribution of numerical predictors, I notice 'acousticness' and 'instrumentalness' both ranges from 0 to 1 and have strong skewed towards 0, so I log transform these variables to center the distribution at 0. I then scaled features except for 'mode', 'key' with StandardScaler. Lastly, I remove 'instance_id' and columns with linguistic variables such as ['artist_name', 'track_name', 'obtained_date'] as these don't contribute to the classification model.





The final preprocessed data has 5000 entries and 14 features.
From the correlation matrix between predictor features, energy and loudness is highly correlated at 0.84; acousticness is inversely correlated to energy and loudness. I will perform dimensionality reduction before building classification model to solve the problem of collinearity.

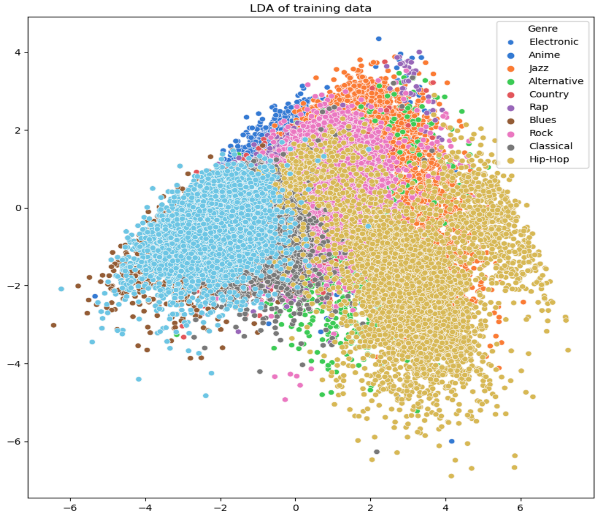
2. Split training and test set

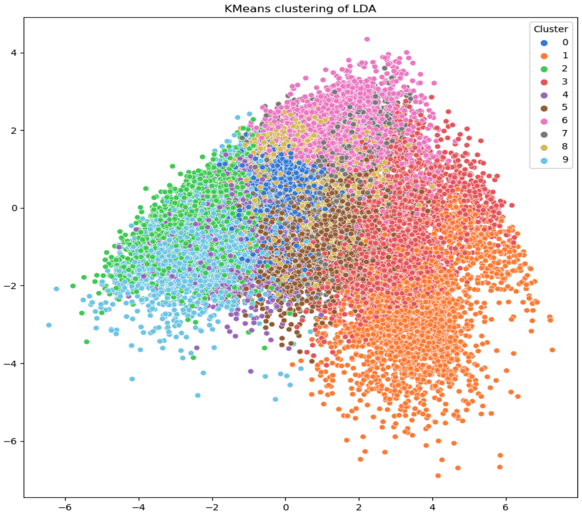
As per instruction, for each of the 10 genres, I sample 500 random songs (using my unique N-number SEED as random_state) for the test set and the other 4500 songs from that genre for the training set. The complete test set will be 5000 randomly picked genres (one per song, 500 from each genre).

3. Dimensionality Reduction and Clustering

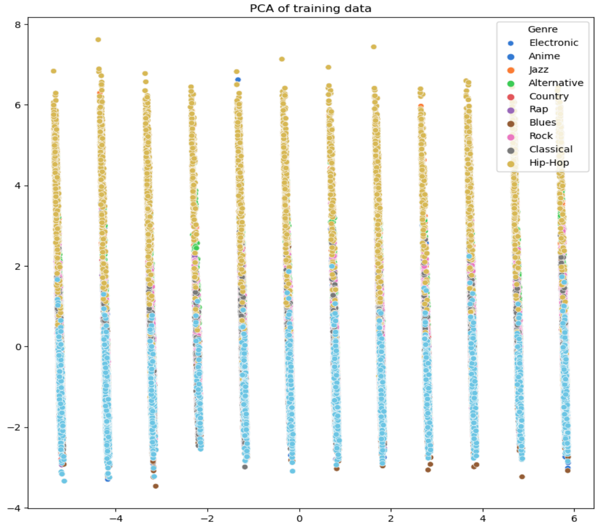
Because the dataset has labels, I first attempt to perform supervised dimensionality reduction with Linear Discriminant Analysis (LDA). I will also perform PCA for comparison. Manifold and embedding methods are not necessary because we wish to preserve the distance in the dataset for future classification models.

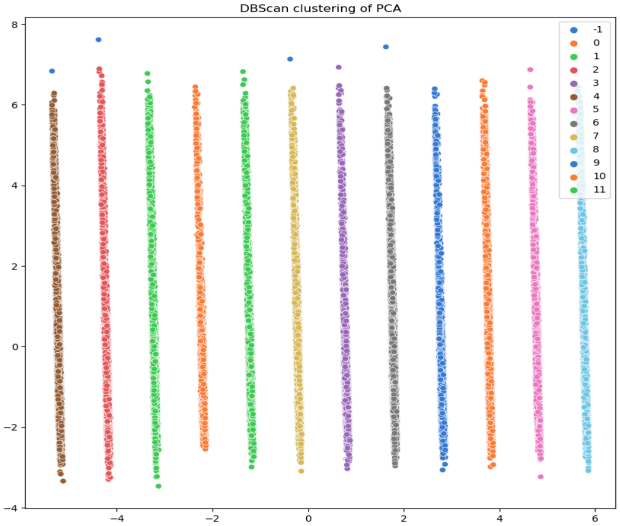
I plotted the first 2 dimensions of LDA and PCA transformed data. LDA managed to represent songs of the same genre closer to one another. The first 2 dimensions of LDA explain 81% of the data variance. After performing kMeans clustering on LDA data with k=10 (as there are 10 genres in the data), relative spaces of a genre are recovered at low precision as there is no distinct boundary between genres in 2D.


LDA of training data


KMeans clustering of LDA

PCA shows 12 distinct, slightly slanted vertical patterns because PC1 picks up ‘key’, a categorical variable, as the dimensions of the largest variance to project onto. PC2 displays a split between ‘Electronic’ and ‘Hip-Hop’ genre, with ‘Electronic’ ranges in the lower half and ‘Hip-hop’ at the upper. These 2 dimensions explain 67.9% of the variance in the dataset. DBSCAN with epsilon 0.5 and min_samples 5 is able to recover the 12 slant shapes in 2D.


PCA of training data


DBscan clustering of PCA

In conclusion, linear dimensionality reduction methods like PCA and LDA are not able to recover the genre labels. However, they can be useful as training input for classification models.

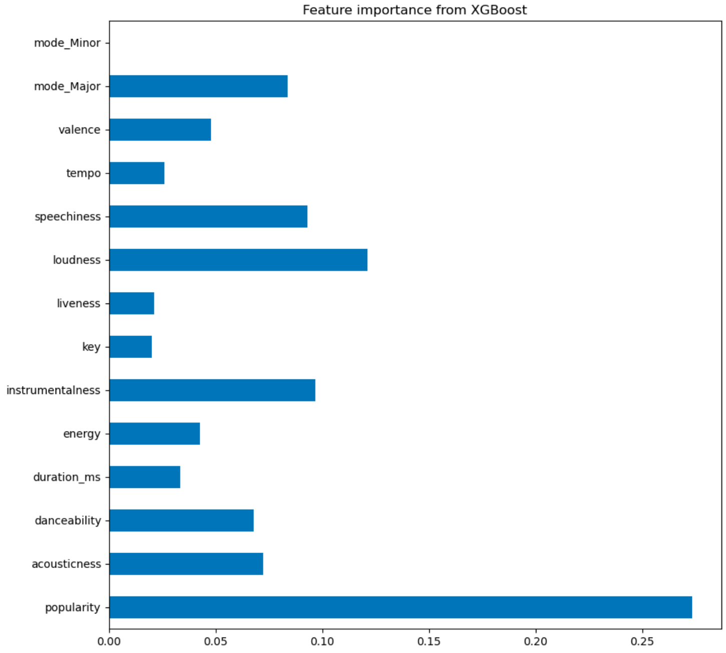
4. Classification models

I built classification models using training data and LDA and PCA-transformed training data including Logistic Regression, Random Forest, XGBoost and Neural Network for initial comparisons. These models are tested on the test set, transformed according to the training set they are built on.

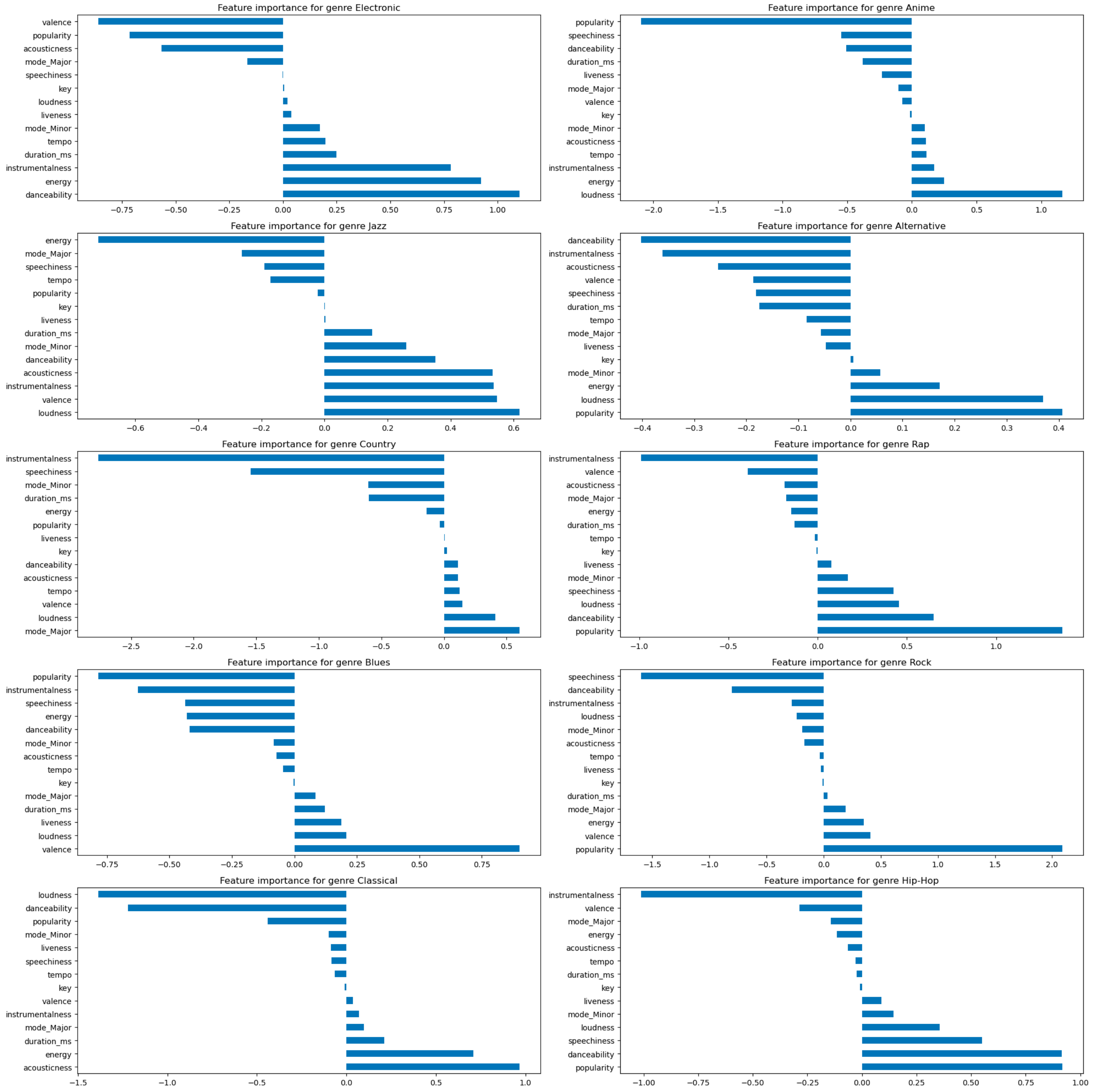
	Training set		*LDA-tranformed training set*		*PCA-tranformed training set*	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
Logistic Regression (max_iter=1000, multi_class='ovr')	0.515	0.891	0.515	0.890	0.515	0.891
Random Forest (n_estimators=100, max_depth=10)	0.572	0.927	0.527	0.907	0.550	0.918
XGBoost (n_estimators=100, objective='multi:softmax')	0.569	0.932	0.543	0.919	0.535	0.913
Neural Network (hidden_layer_sizes=(100, 100), max_iter=1000, random_state=SEED)	0.559	0.923	0.550	0.919	0.558	0.920

- **Feature Importance**

From XGBoost model, I plot feature importance in this dataset to classify music genres. Popularity surpasses other features as the one contributing the most to the classification model.



I compare this feature ranking with that by Logistic Regression models. As coefficients are available for each genre, I compile the feature importance for the classification of each genre. 'Popularity' is the common most important feature, having the largest (absolute) coefficients in the classification of 'Anime', 'Alternative', 'Rap', 'Rock'. 'Instrumentalness', 'loudness' and 'danceability' follow as key features in classifying many genres. For example, clear distinctions between 'Classical' and 'Rock' are evident in the negative coefficients in 'loudness' and 'danceability' and 'popularity' - while in 'Hip-Hop' these features are all positive.

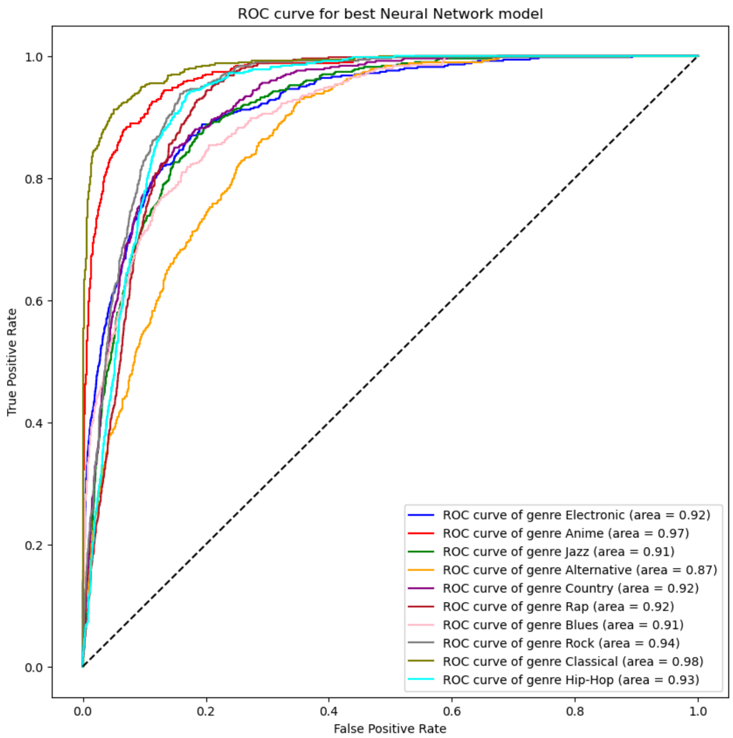


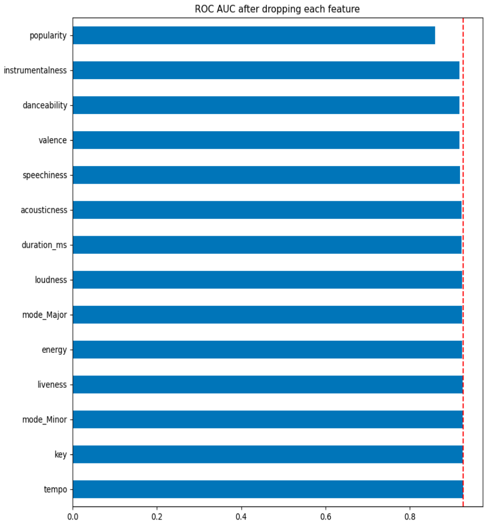
- **Hyperparameter tuning**

I tune hyperparameters of the Neural Network since the accuracy and AUROC indicate good and stable performance compared to Logistic Regression and tree-based methods. I use LDA training data because genre labels are preserved while the data dimensions is reduced.

Utilizing RandomizedSearchCV with a custom multiclass roc_auc_scorer, n_iter=10 and cv=3, I define a parameter grids with hidden layer number ranging from 1 to 5, alpha from 0.001 to 100, and learning rate 'constant', 'invscaling', 'adaptive'.

The configuration yielding the best AUROC is {'learning_rate': 'constant', 'hidden_layer_sizes': (100,), 'alpha': 0.01}. Tested on LDA-transformed test set, this model yields an accuracy of 0.5769 and AUROC of 0.929, suggesting excellent classification ability between the 10 genres. However the accuracy is average because there are subtle similarities between some genres that hinder the model's accurate classification.





From the graph of ROC curves for each genre, 'Classical' and 'Anime' have the highest AUROC, indicating that these genres are the most accurately classified – which makes logical sense because these 2 genres are quite sonically distinct from the other genres in the dataset. 'Alternative' has the lowest AUROC at 0.87, implying difficulty to classify, as this genre has a very loose definition and encompass a wide range of sound.

To determine the most important feature in this model, I drop each feature from the training set, transform the remaining train data with LDA and test on the same LDA test set. The most important feature is one that feature when dropped, results in a model with the lowest AUROC - because this feature contribute the most to the model overall performance. Similar to previous results, 'popularity' is the key feature of this model as well.

Overall, LDA dimensionality reduction and a Feed Forward Network with 1 hidden layer trained on Spotify song dataset result in good performance classifying music genres.