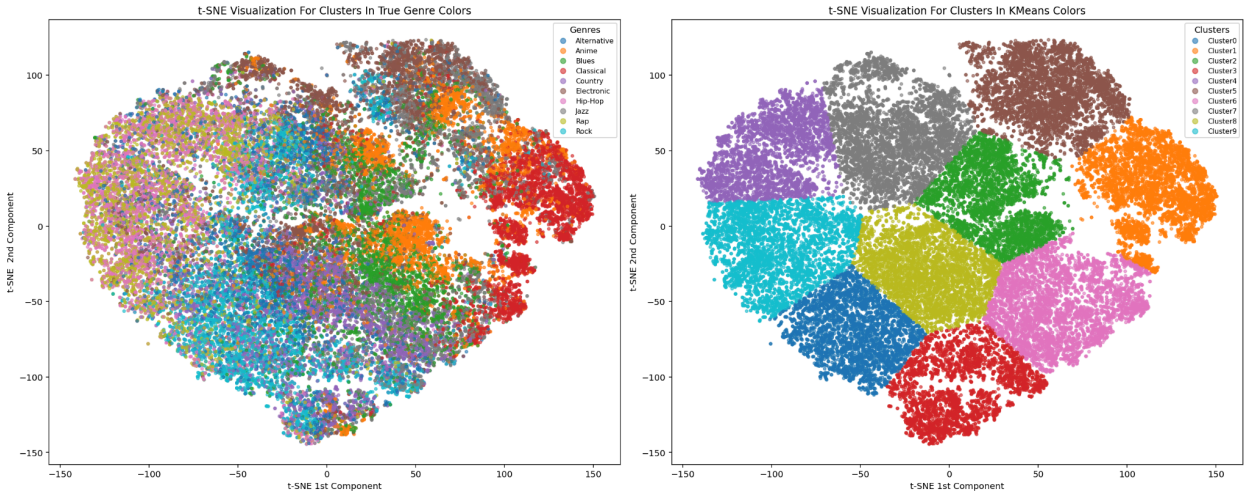
Eunjae (Jenny) Hong

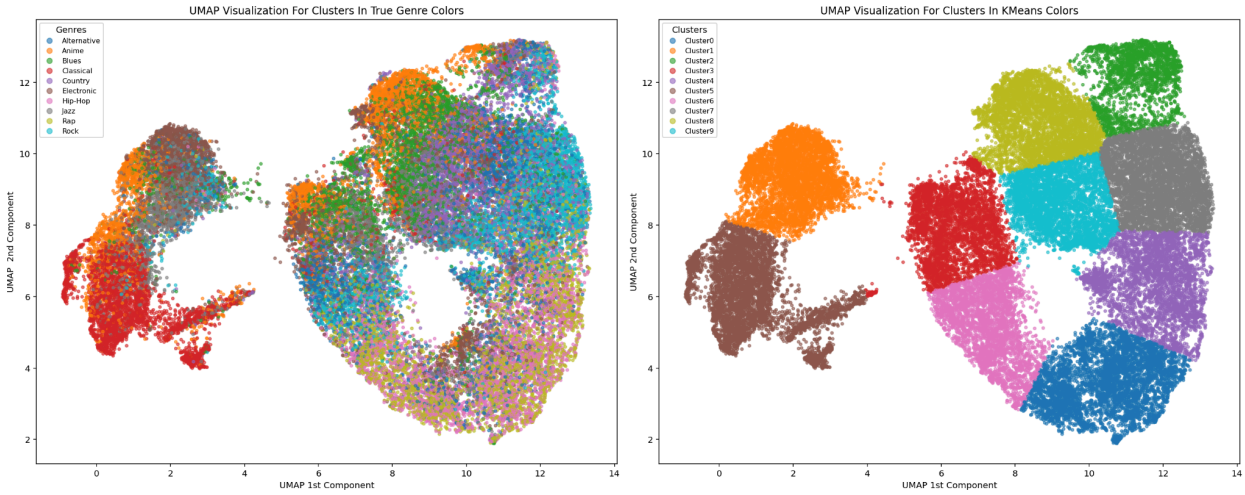
Fall 2025

Unsupervised ML Music Project Report

To prepare the dataset for dimensionality reduction, clustering, and for the classification model, I needed to first take care of the missing data. To address this, I first replaced the '?' values in the datasets into NaNs, and differentiated the features between categorical and numerical. Using the SimpleImputer from sklearn.impute, I imputed the NaNs in the numerical features using the median strategy and for the categorical features using the most frequent value. Then I moved on to do dummy variable coding for features 'key' and 'mode' to turn into numerical data to be useful, and made sure to do drop_first=true to prevent collinearity problems. After that using the StandardScaler I scaled the dataset for only the numerical features, including acoustic features, for the purposes of dimensionality reduction later, and used the LabelEncoder to numerically encode the categorical genre labels for the music dataset. Despite the possibility of useful information in linguistic properties in the artist names and song name, I dropped those columns as well as the unique spotify ID of the song and the obtained date for the Random Forest classification model.

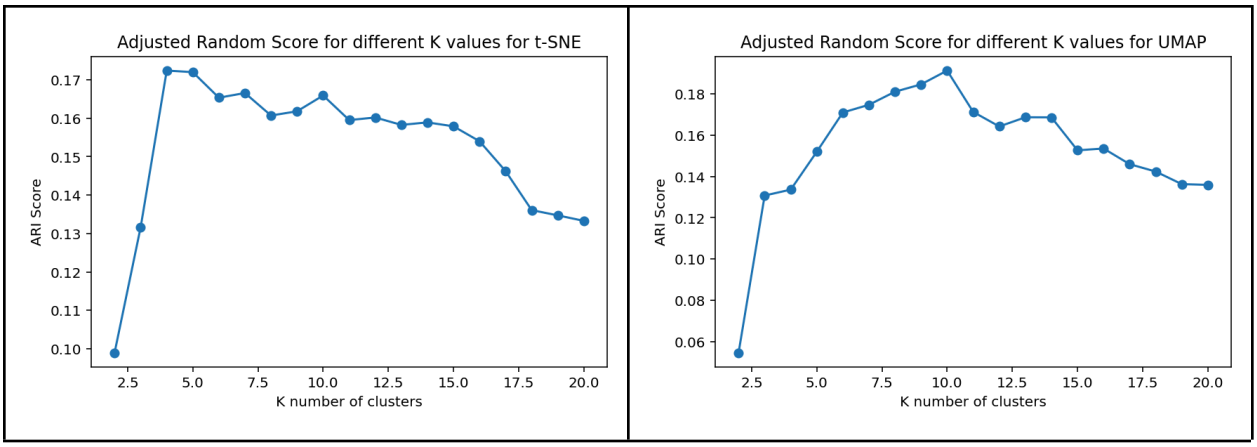


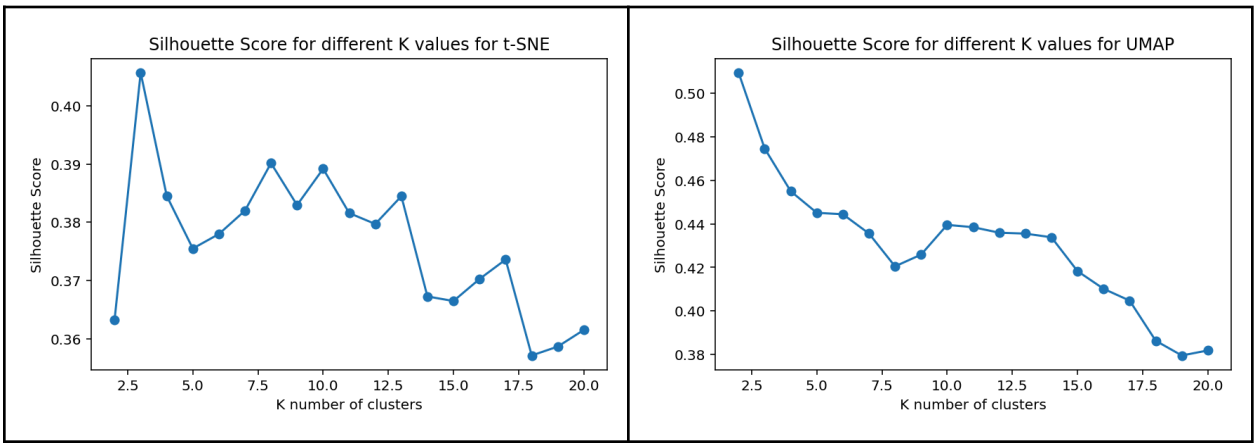
Genres as Clusters in Lower Dimensional Space : (LDA → t-SNE → kMeans)



Genres as Clusters in Lower Dimensional Space : (LDA → U-MAP → kMeans)

After the data was prepared for dimensionality reduction and clustering, I sent the X_train values through the LDA for linear dimensionality reduction and tried two different non linear dimensionality reduction methods, t-SNE and U-MAP before going through the kMeans step. To comment on these two representations of the clustering, I personally thought the first one that used t-SNE after initializing with LDA and then used kMeans looked better in terms of visualization. LDA initialization had worked well as the variance explained by it resulted in 0.9999999999999998. TSNE and UMAP both had outputs of shape (45000,2) after doing .fit_transform for X_lda_train. However I ultimately chose to re do it and also include the second version using UMAP after LDA to do kMeans as even if LDA had initialized the local and global structures of the data well when doing linear dimension reduction, t-SNE would have distorted the global structure when doing the non-linear dimension reduction which could hurt the kMeans step due to not being reliable in metrics. On the other hand, when introducing UMAP, the non-linear dimensionality reduction taught during recitation due to time constraints in lecture, the global structure would have been preserved better for the kMeans.



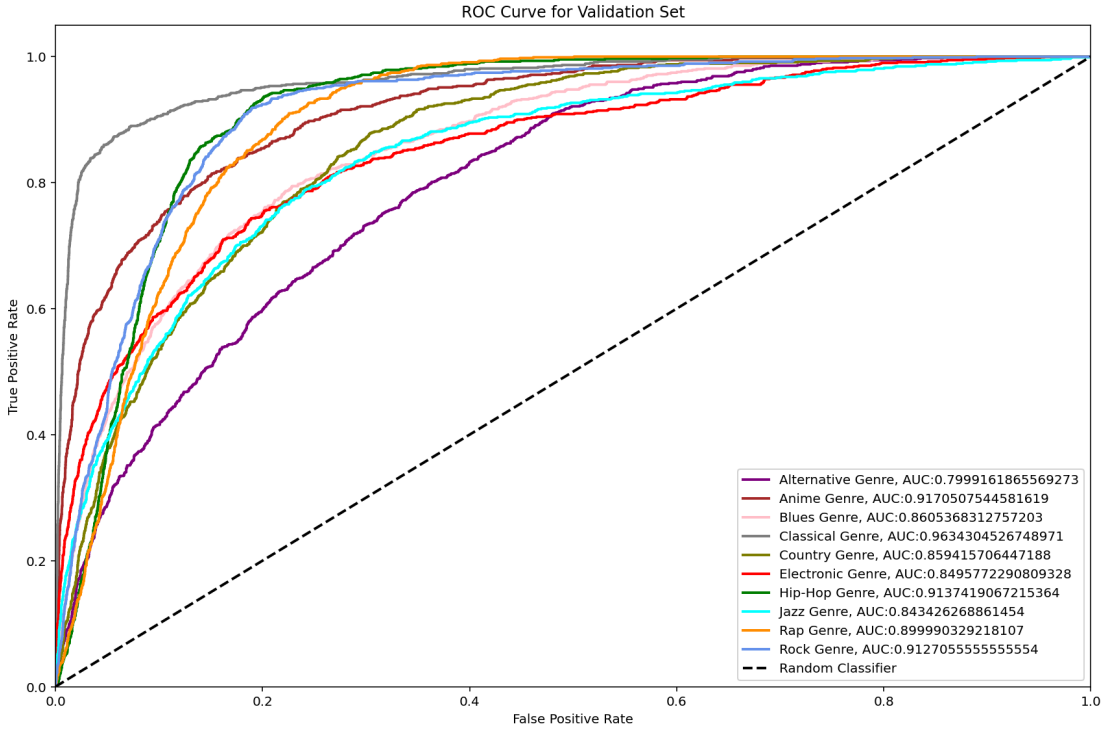


The UMAP method resulted in a larger maximum silhouette score of 0.439535086 compared to t-SNE's 0.38919222, and a larger adjusted random index of 0.1911517800592779 compared to t-SNE's 0.16592026263283727. The optimal K for UMAP being 10 for the maximum adjusted random index score and 2 for maximum silhouette score was more in line with the 10 labeled genres we had as well, compared to the optimal K for t-SNE being 4 for the maximum adjusted random index score and 3 for maximum silhouette score. The maximum silhouette scores and adjusted random indexes were found by looping through k values from 2 to 21.

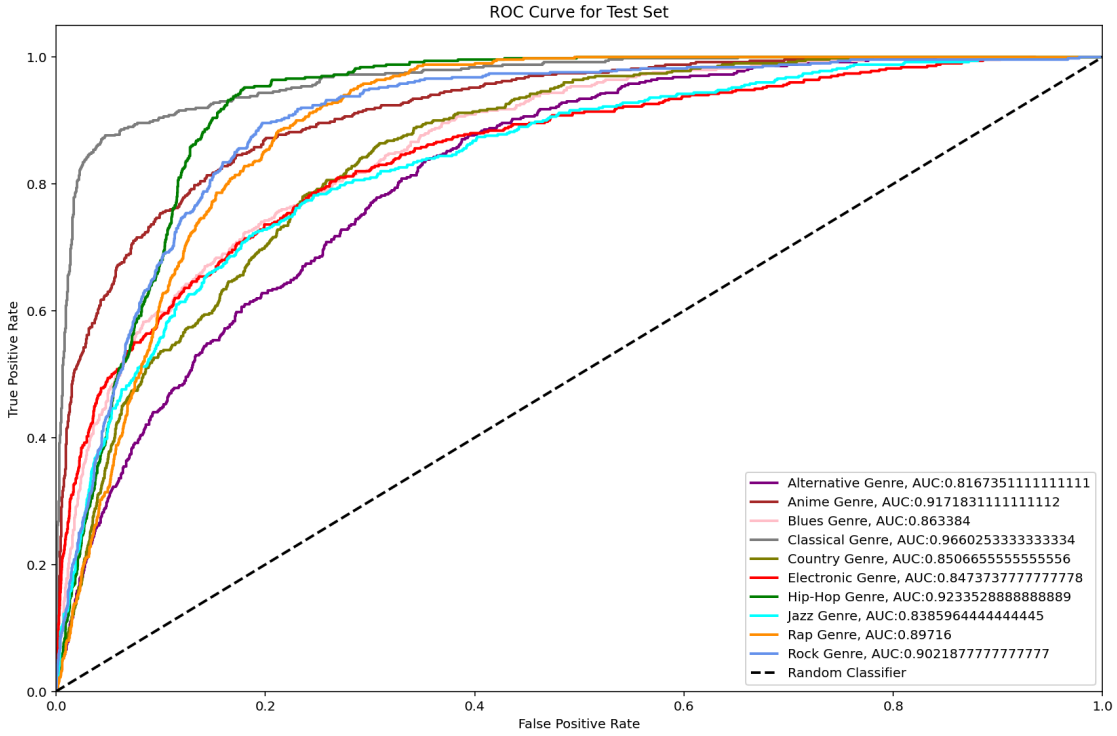
To build the classifier using Random Forests, I first split the training set I had after the UMAP implementation into a smaller training set and validation set in an 80/20 manner to later test the validation set ROC AUC score before doing the final ROC AUC score for the classifier. Using the RandomizedSearchCV, I set the estimator as a RandomForestClassifier I set with my N number and searched for the best hyperparameters including n_estimators, max_depth,

min_samples_split, min_samples_leaf. Using the best hyperparameters in the RandomForestClassifier, the code calculated the accuracy of the validation set and AUC ROC score, and plotted the ROC curve. After confirming that the AUC score for the validation set was above 0.88, I also ran the code for the accuracy, AUC ROC score, and plotted the ROC curve for the test set to get the final result.

I think the most important factor that underlied the classification success was using LDA instead of PCA to initialize the local and global structure of the higher dimension of the Spotify music data. Using LDA resulted in helping to do linear dimensionality reduction in a way that we projected the data into a dimension that had maximized the distance between the projected class means, so that the existing classes of our labeled music data could be more separable compared to PCA that only focuses on projecting into a dimension that maximizes variance of the dataset. Using UMAP instead of t-SNE afterwards also allowed us to preserve the global structure better that prepped the dataset well for the Random Forest classification model. These careful steps enabled a good AUC ROC score result above 0.88 even in a multi class classification problem.



AUC for Validation Set : 0.8819791220850481



Final AUC (Test Set) : 0.8822664000000001