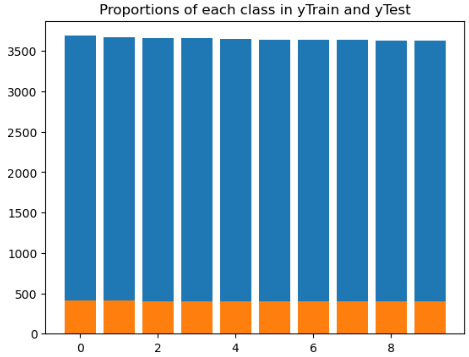
FML Classification Capstone - Erin Choi

Data Preprocessing

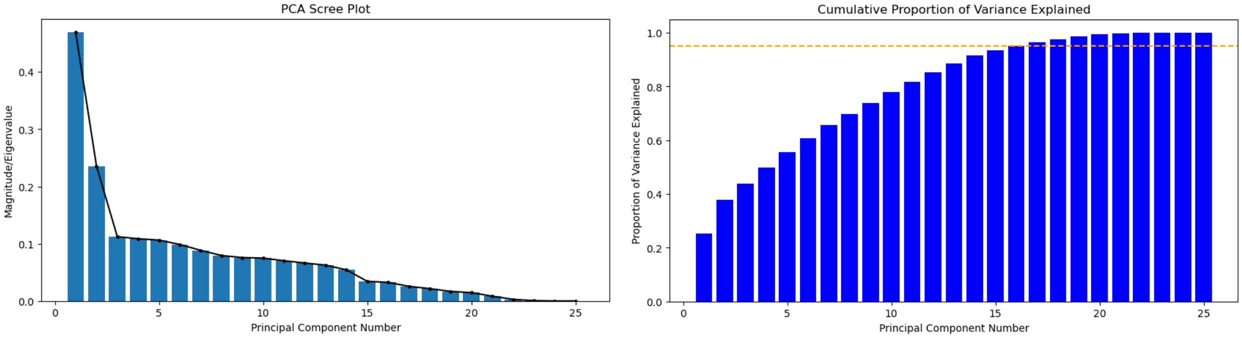
After importing the data, I dropped string variables and other uninformative features (instance_id, artist_name, track_name, obtained_date). I dropped 5 rows that consisted only of null values. The duration_ms and tempo columns contained missing values, but the proportion of songs with such missing values (almost 10,000 total) was large enough that imputing with means or medians could significantly affect classification results. I chose to drop these songs instead of imputing the missing values. I factorized the target (music_genre) to turn each genre into a number from 0 to 9. I also encoded key and mode into dummy variables since they are categorical predictors.

I performed a train/test split before scaling features to avoid data leakage. Since I no longer had 50,000 total songs, I split the dataset so that the test set consisted of 10% of the data while the proportions of genres in the training and test sets were equal. I accomplished this using StratifiedShuffleSplit; each genre still made up about 10% of each set. The bar chart on the right shows the numbers of songs from each genre in the training (blue) and test (orange) sets are very similar. I then scaled some of the features in each set. Upon inspection, most continuous features already had values ranging from 0 to 1, but popularity, duration_ms, loudness, and tempo did not. I normalized these columns so that their values lay between 0 and 1, fitting MixMaxScaler on the training set and transforming both the training and test set with the scaler.

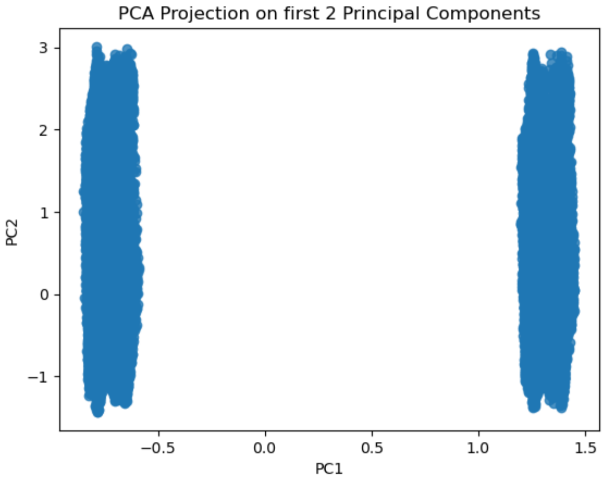


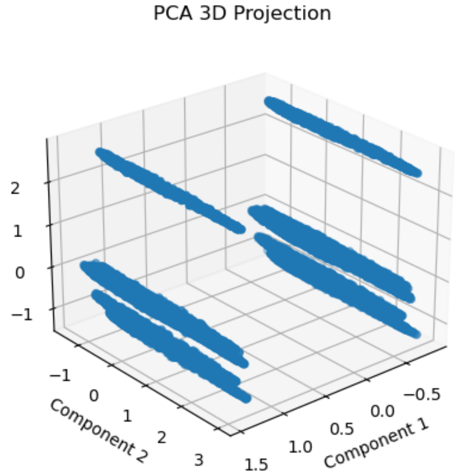
Dimensionality Reduction and Clustering

I chose to reduce dimensions with PCA since its components are the most interpretable, and there are more objective ways to determine how many PCA components to keep compared to other methods. The scree plot for the reduction shows that PC1 and PC2 have much greater eigenvalues than all other individual PCs, though none of the PCs have eigenvalues greater than 1. I wanted the reduced dimensions to explain about 95% of the original variance, so I plotted the cumulative proportion of explained variance for each PC. Using the first 16 PCs would explain slightly over 95% of the variance.

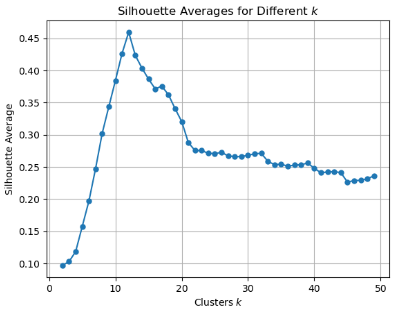


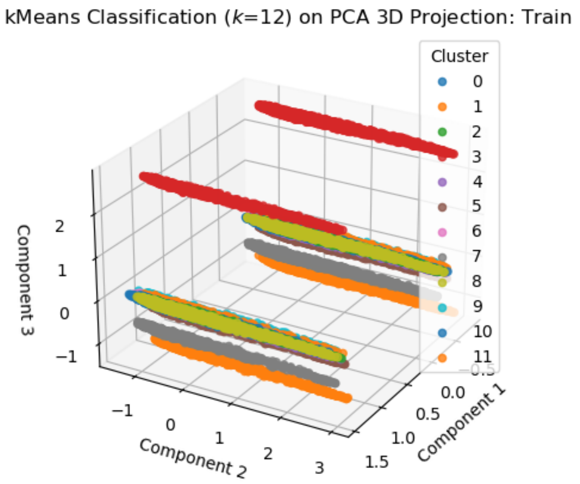
Two distinct clusters formed when the projection onto the first two PCs was plotted. However, in the 3D projection, there are several long clusters stacked on top of each other in the PC3 dimension.

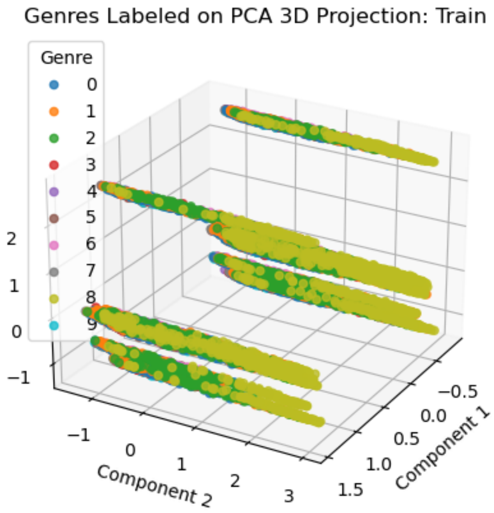




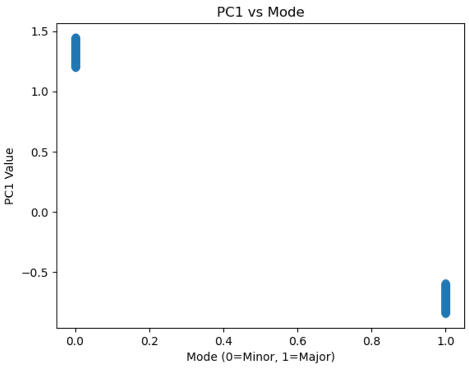
I used the Silhouette method to find the optimal number of clusters for K-means on the PCA dimension reduction; the plot shows that *k*=12 yielded the greatest Silhouette score average, which is close to the number of genres present in the dataset (10). I performed K-means with 12 clusters on the PCA components, and it seemed to separate clusters along PC3, as shown in the labeled 3D projection (PC1-3) and 2D projection (PC1 and PC3). However, these clusters do not appear to correspond to genres; coloring songs by genre on the 3D projection does not look like the K-means clustering result at all.

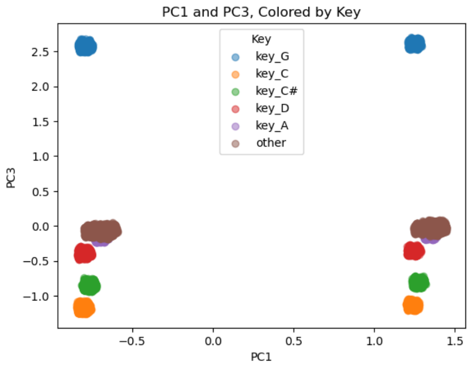






I inspected which features were most important to PC1, PC2, and PC3, and the clustering made a lot of sense even though it did not separate songs by genre. PC1 essentially split songs based on their mode (major/minor) - plotting PC1 values against mode results in a clean split of points. PC2 was mainly a combination of acoustic and subjective sonic features, with its most important features being acousticness, energy, and instrumentalness, which explains the even spread of songs across its axis. PC3 seems to be related to musical key, with its top 5 features being dummies for the keys G, C, C#, D, and A. Most of these keys are very commonly used (C# is a slight outlier in this sense), and key_G has a much higher importance than all other features. Plotting PC1 and PC3 against each other with these keys colored differently, there are different clusters for songs in G, C, C#, and D, with songs in A mixed in





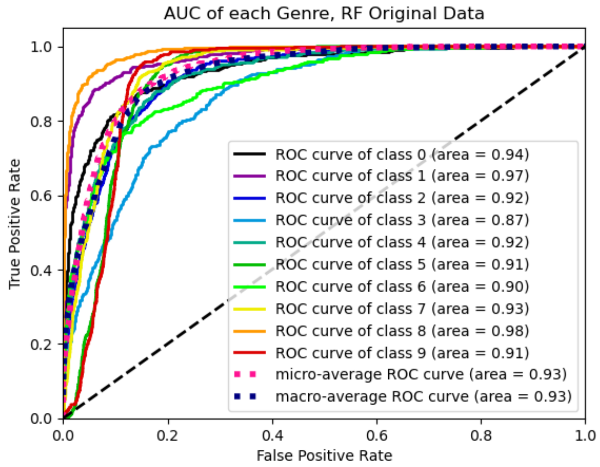
with songs in other keys. Thus this particular dimension reduction by PCA could be more useful for visualizing the different modes or keys of songs.

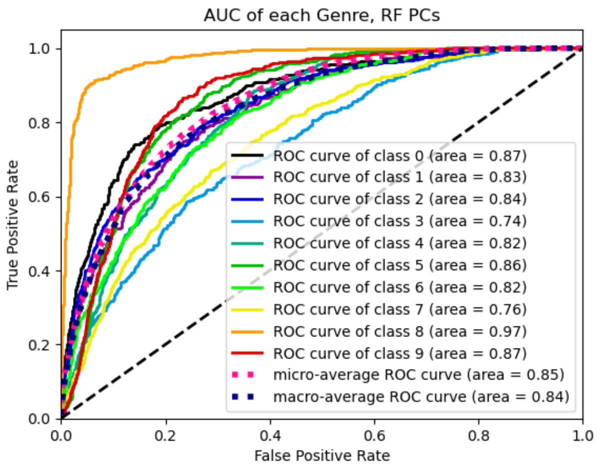
Building and Evaluating the Classification Model

As many of the continuous predictors are not normally distributed and there are many categorical features, I used tree-based models that can still perform well using these kinds of features. I compared tuned Random Forest and AdaBoost models that used (a) the original (encoded and scaled) dataset, (b) the original dataset with PCA/K-means cluster numbers added as a feature, and (c) the first 16 PCA components with the cluster number feature. The Random Forest models performed better than AdaBoost as a whole. Random Forest using the original dataset had the highest one-vs-rest AUC, and unexpectedly, adding the cluster numbers did not improve the AUC, though the accuracy increased slightly. Using the reduced dimensions with cluster numbers led to poorer performance in both AUC and accuracy.

Feature importance was analyzed using the feature_importances_ attribute of the Random Forest models. For the Random Forest model using the original data, popularity was the most important feature. This makes sense as popular music usually comes from specific genres like Pop (not included in this data) and Rap/Hip-hop; knowing whether a song is more or less popular can provide insight as to whether or not it belongs to such genres. The Random Forest model using PCA components had PC2 as its most important component, which, as stated above, mainly combines some acoustic and subjective sonic features, such as acousticness, energy, and instrumentalness. These features are also among the predictors of higher importance in the model using the original data. Cluster number features do not appear among the most important features of this model.

The final macro-average AUC using only original data is 0.92561 (accuracy = 56.02%). **The final macro-average AUC using 16 PCs and cluster numbers is 0.83753** (accuracy = 38.04%). The ROC curve plots for each model's average predictions and predictions for each class are displayed below.


AUC of each Genre, RF Original Data
True Positive Rate
False Positive Rate
ROC curve of class 0 (area = 0.94)
ROC curve of class 1 (area = 0.97)
ROC curve of class 2 (area = 0.92)
ROC curve of class 3 (area = 0.87)
ROC curve of class 4 (area = 0.92)
ROC curve of class 5 (area = 0.91)
ROC curve of class 6 (area = 0.90)
ROC curve of class 7 (area = 0.93)
ROC curve of class 8 (area = 0.98)
ROC curve of class 9 (area = 0.91)
micro-average ROC curve (area = 0.93)
macro-average ROC curve (area = 0.93)


AUC of each Genre, RF PCs
True Positive Rate
False Positive Rate
ROC curve of class 0 (area = 0.87)
ROC curve of class 1 (area = 0.83)
ROC curve of class 2 (area = 0.84)
ROC curve of class 3 (area = 0.74)
ROC curve of class 4 (area = 0.82)
ROC curve of class 5 (area = 0.86)
ROC curve of class 6 (area = 0.82)
ROC curve of class 7 (area = 0.76)
ROC curve of class 8 (area = 0.97)
ROC curve of class 9 (area = 0.87)
micro-average ROC curve (area = 0.85)
macro-average ROC curve (area = 0.84)