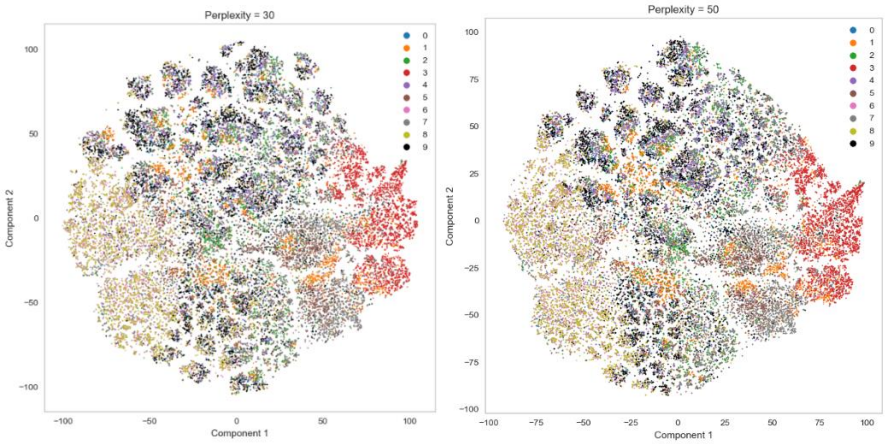
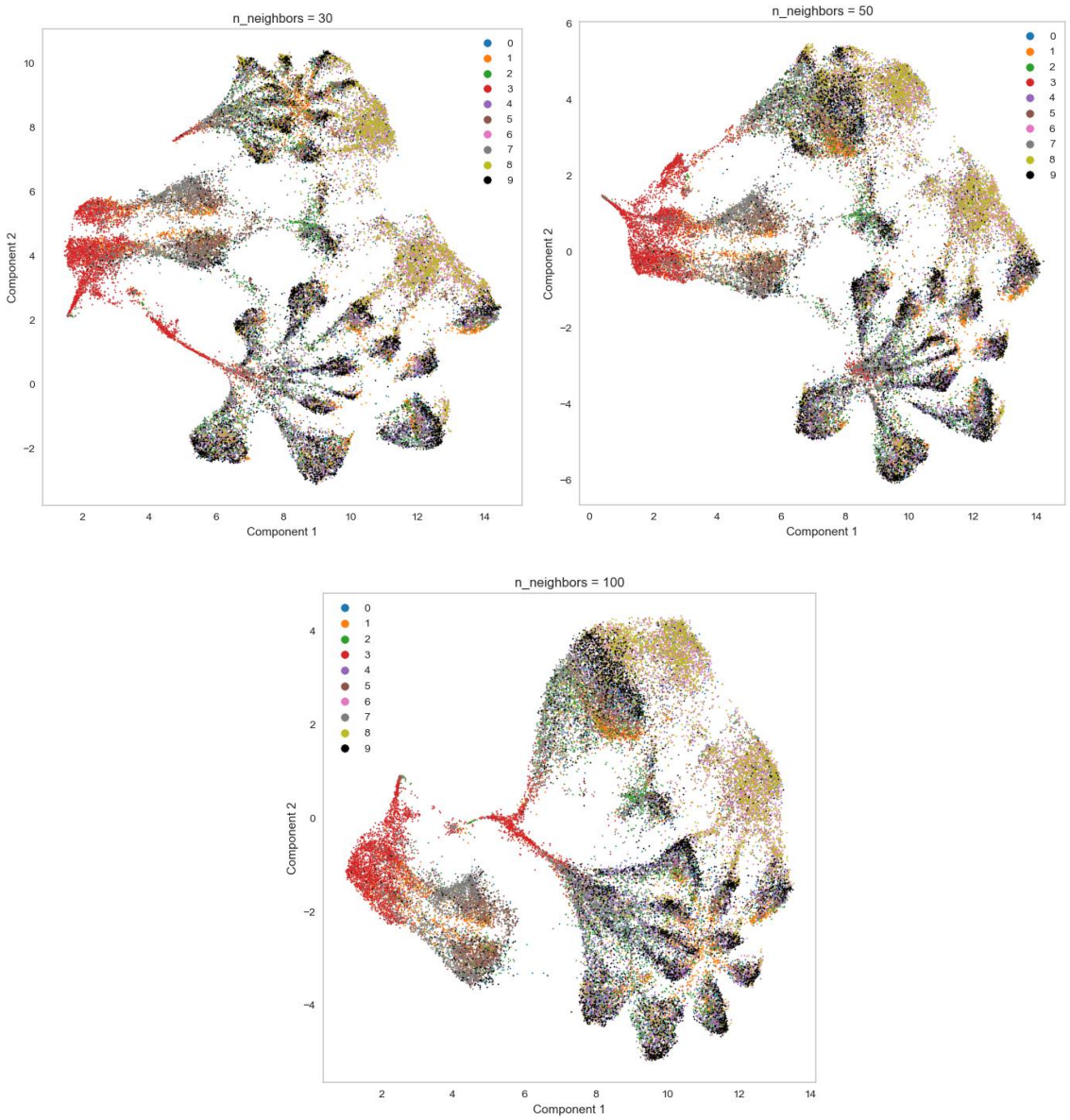
I first started with the preprocessing of the data. There are a few rows in the original csv file that are completely blank, so they are useless and need to be discarded. After doing that there are exactly 50,000 data points (songs) left. Some of the durations of the songs are missing (represented as "-1") and so are some of the tempo data (represented as "?"). In order to make best use of the data available, I replaced the missing data with the average values of their column. The "mode" and "key" columns contain categorical data, and dummy coding is required to turn them into numerical data. I chose to use one-hot encoding to transform both columns because modes (minor vs major) and keys (from A to G) are not in numerically increasing order (e.g. minor mode is not "smaller" in quantity than major mode). Finally, I used label encoding to assign an integer number to each music genre (from 0 to 9) because sklearn machine learning algorithms do not accept categorical labels.

After preprocessing the data, I normalized all numerical columns that are not dummy coded by subtracting the mean and dividing by std. This ensures that all columns have a similar scale and will have similar influences on the prediction results. I then separately tried TSNE (perplexity = 30, 50, 100) and UMAP (neighbors = 30, 50, 100) to reduce the data frame to 2D (n_components = 2) and look for clustering tendencies. Unlike PCA, they are unaffected by the fact that musical attributes are not normally distributed. Here are the TSNE results:



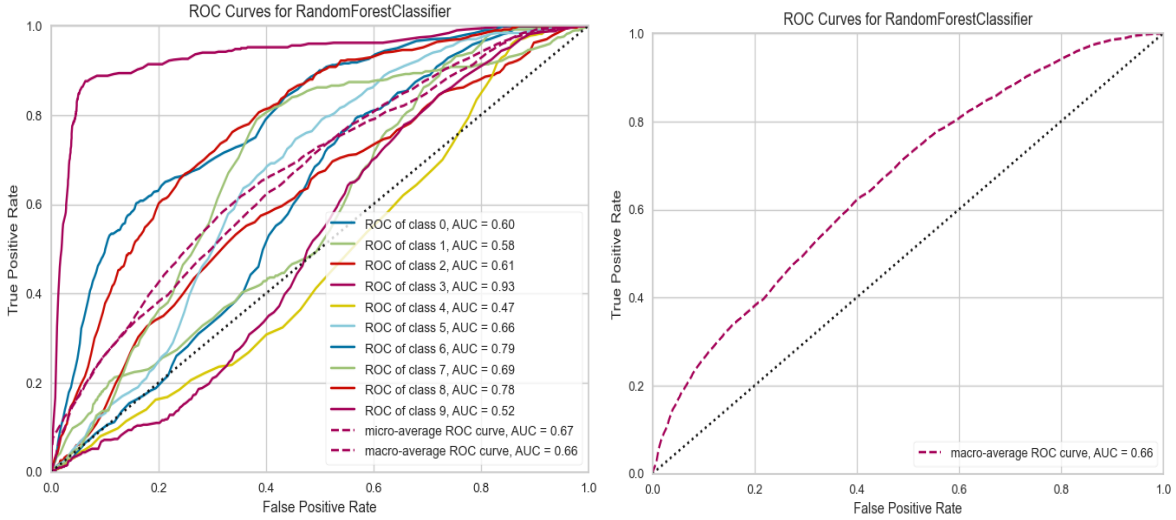
The UMAP results look a bit different (clustering seems more prominent than TSNE):



Dimensionality reduction reduces noises in the data and cuts down computing cost. Columns like artist name, track name, ID and date were not used. Upon closer inspections, there are some shared traits between the results of TSNE and UMAP: genres 3 (classical), 6 (Hip-Hop) and 8 (Rap) exhibit stronger clustering effect compared to other genres, in which 6 and 8 tend to mix with each other, suggesting shared musical qualities between Hip-Hop and Rap. This tendency may affect the outcomes of the classification step.

The common TSNE algorithm only supports n_components smaller than 4, otherwise it takes much longer to compute. This means that the reduced data can only have three features, which may not be enough for classification models like RandomForest. Thus I used UMAP dimension reduction results to train the classification models.

I mainly tried two classifiers: RandomForest and Deep Neural Network. RandomForest involves finetuning the number of trees in the forest, the number of datapoints each tree is trained on (bootstrapped), the number of features considered when making each split (to decorrelate the trees), and the criterion. Deep Neural Network involves adjusting the number of hidden layers and the number of neurons in each hidden layer, as well as activation functions used (ReLU, Sigmoid, …). The number of neurons in the input layer is the same as n_components in UMAP, and number of neurons in the output layer is equal to the number of genres – 10. Different n_components of data from UMAP are also tried, finally settling at n=10. RandomForest ultimately achieved an AUROC of about 0.66 (“one-versus-rest”), with n_estimators=1000, max_samples=0.6, max_features=0.3,bootstrap=True, criterion='gini'. Deep Neural Network achieved an AUROC of about 0.65, with two hidden layers having 30 and 10 neurons. ReLU was picked because it best addresses the problem of vanishing gradients.



The left diagram plots the ROC of each genre vs. all the other genres combined as one, so the multi-class classification problem can be broken down to 10 binary classification problem (“one-versus-rest”). It can be seen that class 3,6 and 8 have the highest AUROC, validating the suggestions by TSNE and UMAP diagrams that these 3 genres are “well clustered” (they don’t mix with others). Thus dimensionality reduction and clustering can be said to foretell the prediction performances for each class. Sklearn does not have this function, so another library called yellowbricks was imported. Finally, an overall ROC curve was plotted as the average of the individual ROCs. The RandomForest model worked quite well for certain genres but poorly for others, possibly due to their shifting and amorphous nature. If given more time, prediction results may be improved by further feature engineering or adjusting n_neighbors in UMAP.