Fundamentals of Machine Learning Capstone Project

Classification Flavor

Udgam Bhattarai (ub2028)

May 6th, 2025

Introduction

Building an efficient classification model for this project using around 50k samples of Spotify songs, that have been broken down into features such as "tempo" and "energy" and classified into their respective genres, involves (1) preprocessing the data, (2) reducing dimensionality to study and identify trends in the data, and (3) using the preprocessed data to train and test a classification model and get it to a reasonable accuracy and AUC.

Data Preprocessing

The dataset has a number of issues, inconsistencies, and incompatibilities that need to be addressed in order to prepare it for further analysis. The primary issues were identified as: (1) Unique information in `instance_id, artist_name, track_name,` and `obtained_date`, (2) 5 rows of NaN values, (3) -1.0 values for `duration_ms` and '?' values for `tempo`, (4) Categorical features `key` and `mode`, and `string` labels in `music_genre`.

Dropping Features

Intuitively, as `instance_id, artist_name, track_name,` and `obtained_date` are features unique to each song, they are expected to have little to no correlation to the genre and with so many unique values prove increasingly difficult to process and prepare for classification models and dimensionality reduction. To that end, they were dropped for the purpose of the remainder of this project.

Missing Values

Given that the 5 rows of all NaNs in the dataset contained no real information for the purpose of this classification, they were immediately dropped from the dataset. For the 5000 data points with '-1.0' for `duration_ms` and '?' for `tempo` respectively, dropping the data points would result in the loss of approximately 20% of the data. To that end, the missing data is imputed by using the genre-wise median for both `duration_ms` and `tempo` because: (1) the distribution for both `duration_ms` and `tempo` is right-skewed so the median provides a more accurate representation of a central value, (2) using global mean or median heavily skews the distribution, and distorts local relationships, and (3) linear correlation between `tempo` and `duration_ms` values and the `music_genre` is preserved.

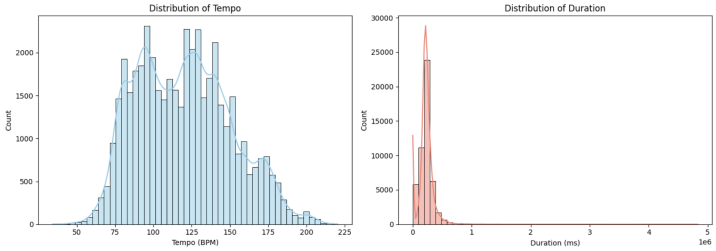

Figure 1: Distribution before imputing

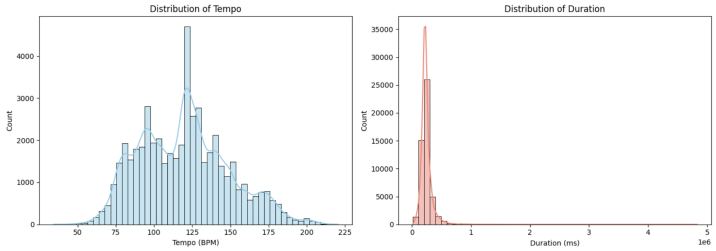

Figure 2: Distribution after imputing

Categorical Data

The dataset has categorical data for the features `key, mode,` and the labels `music_genre`. Since most classification and dimensionality reduction models don't work with categorical features, especially ones that are `string`, `key` and `mode` were one-hot encoded since they have a limited number of unique values, that have no shared ordinal relationship. However, the label `music_genre` was mapped to a set of unique integers using `LabelEncoder()`. Although this introduces an artificial ordinal relationship, it is acceptable for classification tasks using models like neural networks or tree-based methods that can handle integer class labels without assuming order.

Standardizing Data

Finally, since features such as `duration_ms, popularity` and `instrumentalness` are all on different scales, the data is z-scored using `StandardScaler()` to ensure that the scale of different features doesn't dictate their weights or the loss of the classification model.

Dimensionality Reduction

For dimensionality reduction, the one-hot encoded variables were converted to `float`, given that dimensionality reduction methods don't usually work with `boolean` data.

PCA

Doing a PCA in the data set revealed 3 Principal Components with eigenvalues greater than 1, wherein the first 2 PCs explained about 42% and the first 3 PCs explained roughly 50.8% of the variance in the data respectively. However, when the dataset was projected onto these PCs, most `music_genre` were barely separable, except for "Classic," which easily stood out in both 2D and 3D.

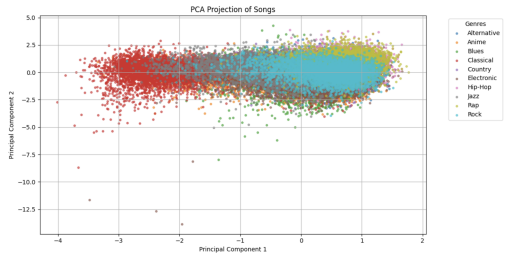

Figure 3: 2D PCA Projection

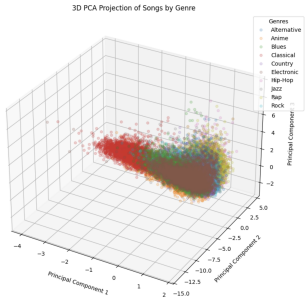

Figure 4: 3D PCA Projection

tSNE

While 2D and 3D t-SNE projections, with KL divergences of 3.171 and slightly lower respectively, show improved local clustering compared to PCA, these clusters do not align strongly with the `music_genre` labels, thus suggesting that `music_genre` is not the primary factor shaping local similarities in the feature space.

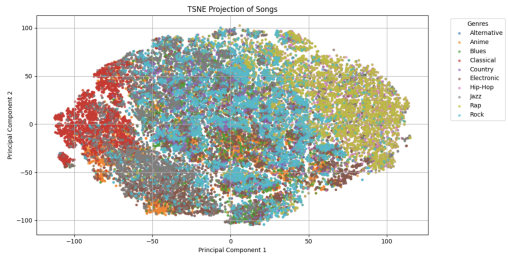

Figure 5: 2D tSNE Projection

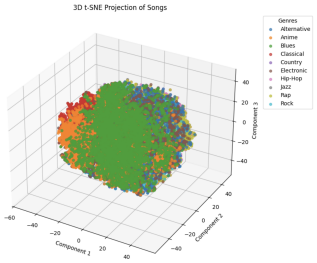

Figure 6: 3D tSNE Projection

LDA

As LDA is supervised and already works with labels, the 2D and 3D projections on LDA components provide the cleanest genre separation. However, since LDA assumes a gaussian distribution of features, this visually interpretable separation doesn't translate accurately for classification.

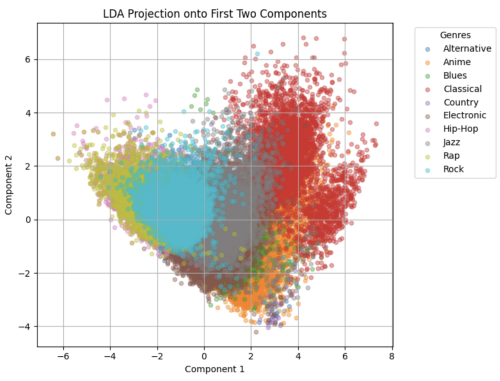

Figure 7: 2D LDA Projection

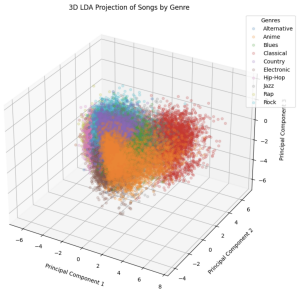

Figure 8: 3D LDA Projection

Clustering

When trying to cluster the 2D PCA projection of the dataset, neither KMeans nor DBScan produced results that even mildly represented the distribution of `music_genre`. KMeans determined 3 to be the optimal number of clusters, while DBScan almost always identifying one big central cluster and other very small clusters for any given hyperparameters. This suggests that `music_genre` is not a dominant factor shaping the global structure of the feature space, indicating that genres likely overlap significantly in the dimensions captured by PCA.

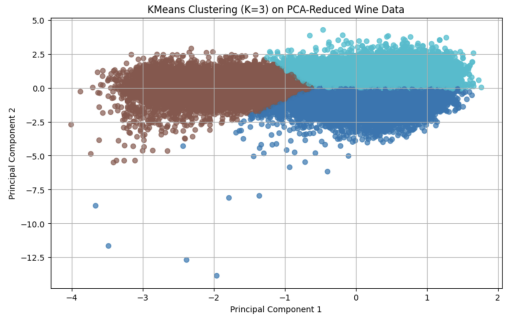

Figure 9: 2D KMean

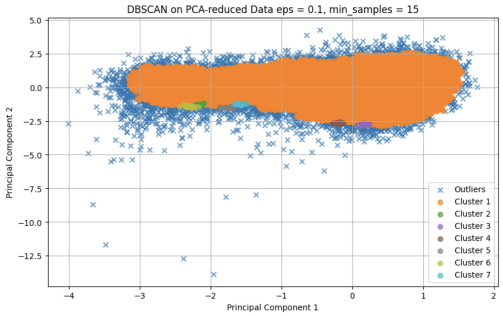

Figure 10: 2D DBScan

Classification

When building a classification model for this case, SVMs and Decision Trees were immediately ruled out of consideration given their inability to work well with multi-class classification of high-dimensional data, and inherent improvement in performance promised by ensembles of decision trees in Random Forests and ADABoost respectively.

Random Forest

A random forest classification with `n_estimators = 200, max_samples = 0.75, max_features = 'log2', min_samples_leaf = 2, min_samples_split = 10, max_depth = 10`, and `bootstrap = False` yielded an **AUC of 0.9289** with an **accuracy of 59.92%**. Meanwhile, running it on the dataset with PCs that retain 95% of variance, the AUC and accuracy both declined to 0.86 and 42.05% respectively.

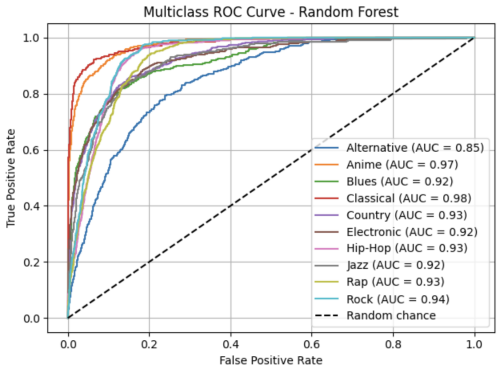

Figure 11: Random Forest ROC

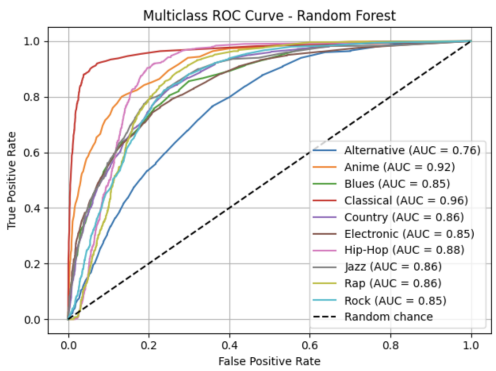

Figure 12: Random Forest PCA ROC

ADABoost

An ADABoosted classification model with `max_depth = 3, min_samples_split = 10, n_estimators = 200`, and `learning_rate = 0.1` resulted in an **AUC of 0.911** and an **accuracy of 56.54%**. Meanwhile, running it on the dataset with PCs that retain 95% of variance, the AUC and accuracy both declined to 0.51 and 41.76% respectively.

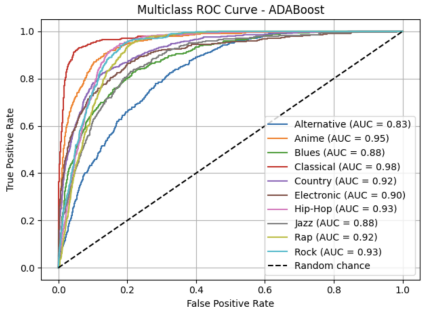

Figure 13: ADABoost ROC

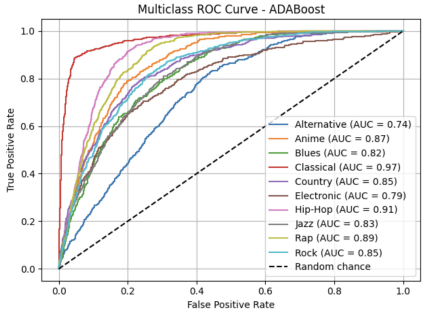

Figure 14: ADABoost PCA ROC

Neural Network

Neural Networks proved to be, by far, the best approach towards classifying the songs into their respective genres. A fully connected Neural Network with ReLU activation function yielded an **AUC of 0.884** and **accuracy of 45.62%** for 3 hidden layers, **0.910 AUC, 53.78% accuracy** with 2 hidden layers, and **0.856 AUC, 35.30% accuracy** with 1 hidden layer.

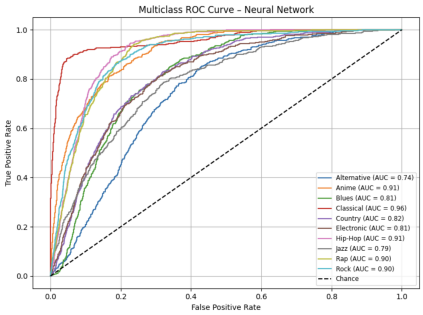

Figure 15: 1 hidden layer

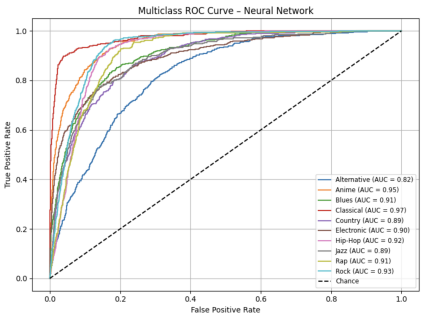

Figure 16: 2 hidden layers

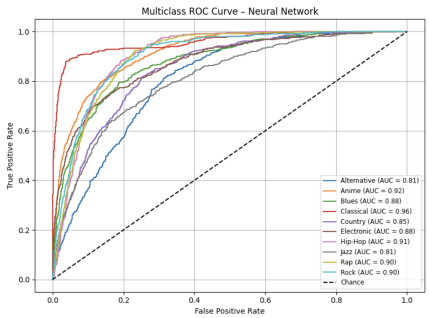

Figure 17: 3 hidden layers

Best Classification Model

Since the AUC begins to plateau between 2 and 3 hidden layers, and the change in accuracy also graduates, instead of adding more hidden layers to the Feedforward Neural Network, the existing layers were modified and complicated to better handle the non-linearity of the data. To that end, the Neural Network with 2 hidden layers, and the architecture as shown in Figure (12) yielded the best **AUC of 0.9313** and **accuracy of 59.0%** over 100 epochs.

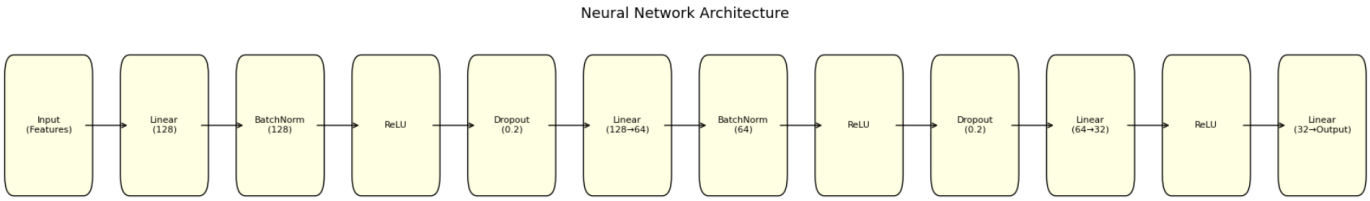

Figure 18: Architecture of NN

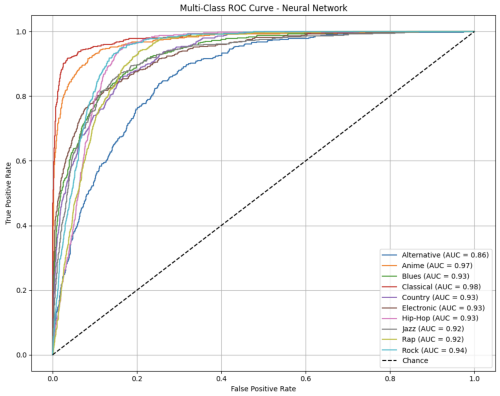

Figure 19: Best classification Model

Additional Observations

Genres

When implementing classification models, the most common misclassifications were Hip-Hop and Rap that would often be classified as other. This shows that "Hip-Hop" and "Rap" are closely related genres that share very similar features. Likewise, "Alternative" was the most misclassified genre with an even distribution of misclassifications across all other genres, showing that songs from this genre aren't usually consistent and often incorporate a lot of characteristics and features from other genres.

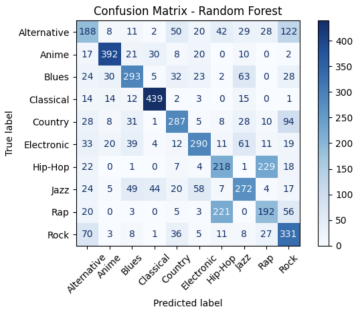

Figure 20: Random Forest Confusion Matrix

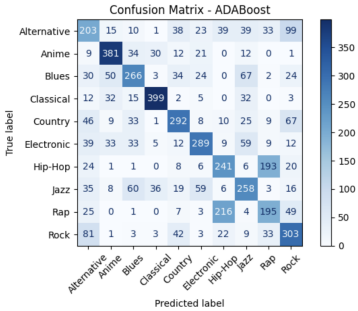

Figure 21: ADABoost Confusion Matrix

PCA Clusters

Cluster	Popularity	Acousticness	Danceability	Duration (ms)	Energy	Instrumentalness	Liveness	Loudness	Speechiness	Tempo	Valence
0	40.0	0.027	0.511	239238.0	0.810	0.000319	0.180	-5.624	0.0484	129.9850	0.456
1	54.0	0.175	0.686	214447.0	0.609	0.000004	0.116	-7.327	0.0665	113.3120	0.540
2	32.0	0.945	0.355	247850.0	0.141	0.811000	0.110	-18.944	0.0409	98.2875	0.165

Table 1: Cluster-wise Median Values of Audio Features

The clusters reveal distinct musical traits: Cluster 0 is energetic and loud, Cluster 1 is danceable and mainstream, and Cluster 2 is soft, acoustic, and instrumental. These differences suggest that the first principal component may capture energy or loudness, while the second may reflect acousticness or instrumentality, helping differentiate tracks by both intensity and musical texture.

Conclusion

The most important feature to the relative success of this classification metric was (1) imputing the missing data by using genre-wise median thus retaining local structure while not losing a large chunk of the data, and (2) introducing `nn.Dropout()` and `nn.BatchNorm()` which increased the AUC of a 2-hidden-layer feedforward Neural Network from 0.915 to 0.928 by enhancing generalization and stability.