Investigating the Predictive Power of Spotify Audio Features on Music Genre Classification

Eric Tay

1 Introduction

Many companies have a use for music genre classification. Traditionally, this is done using audio signals from songs. However, Spotify has recently released an API [1] that allows users to easily obtain audio features of songs (e.g. energy) on Spotify. Classification based on audio features, instead of audio signals, has multiple advantages, but often suffers from poorer accuracy, which motivates the question of when this method is suitable. This paper hypothesizes that the loss of accuracy comes from classifying tracks into "closely" related genres, which may exist in "genre-clusters", where similar genres exist in the same cluster. This paper will hence first identify which genres are more similar to each other, and then investigate which subsets of genres would yield predictions with high accuracy. In doing so, we will investigate if the genre selection done prior to classification has a significant effect on classification accuracy, due to the existence of these "genre-clusters". Our goal is not to ascertain these clusters definitively in this work, but to motivate future study of genre similarity, and incentivize prudent pre-classification genre selection. The primary target audience of this paper is firms who wish to accurately classify music into pre-selected genres. Depending on the granularity of genre selections, this paper hopes to inform firms when it would be suitable to do classification based on Spotify Audio Features, or if more data and more sophisticated models should be used instead. A second target audience is personal programmers, whose traditional focus in genre classification works has been model selection, which this author believes should be assessed only after careful genre selection.

We shall use a dataset with over 230,000 tracks covering 27 genres (of which we will use 25). This project will verify the hypothesis that genres exist in clusters in two steps, a "Clustering" step and a "Classification" step. The first part will use various clustering methods to eventually cluster 25 selected genres into 8 clusters. After this clustering, we will use a Random Forest to predict genres from audio features, with multiple different categories of genre selections (inter-cluster, intra-cluster, mix), and draw conclusions from the classification results.

2 Background and Literature Review

Music service providers like Spotify and Soundcloud use genre classification to aid search, create playlists, and provide recommendations to consumers. Other companies, like Shazam, provide this as a feature in their products. Due to the available format of the data, traditional approaches for genre classification use audio signals as an input [2] [3] [3] [4]. These methods often involve complex feature extraction and deep neural networks, achieving good accuracy, but generally being computationally expensive and lacking interpretability.

Recently, Spotify provided a new format to the data that comes in the form of audio features. Using these features for classification [5] [6] [7] [8] allows access to more data, provides interpretable conclusions about the difference between genres [7], is easier to implement, and is computationally inexpensive. Generally, across various implementations, a Random Forest model provides the best classification accuracy. However, accuracy is still relatively low, possibly because there is less information encoded in audio features than raw audio signals.

One challenge with genre classification is that genres are not clearly defined. Hence, genres influence each other, giving rise to clusters [9]. Therefore, methods that classify music into fewer genres [3] [4] [5] [7] report better accuracy than those that use more genres [2] [3] [6] [8], with [5] showing a decrease in accuracy when classifying genres that are intuitively similar. Companies wishing to harness the benefits of using audio features for genre classification will therefore need to understand which genres these features can distinguish between.

For this problem, we need to use audio features to cluster tracks in a way that is most similar to the given genre labels. While there is work with regard to clustering using audio features [10] [11], these are usually done without a "ground truth" clustering to reference. We will therefore have new evaluation metrics and employ a variety of both unsupervised and semi-supervised clustering methods not used in other works.

Next, we will assess the accuracy of predictions with various genre selections. This is also new, as works on genre classification usually pre-select the genres arbitrarily, with the focus on the performance of different models on the same selection of genres. The results of this paper would therefore allow for a more systematic approach to genre selection, that would yield a more suitable dataset for model comparison.

3 Data

The data chosen is a list of 232,725 tracks obtained from Kaggle [12], with 27 distinct genre labels (listed in Table 2). Each track contains a genre label (response variable), and 14 audio features (predictive variables - Table 1) obtained using Spotify's API [1] in July 2019. It is important to note that the API does not label songs with genres. To obtain these labels, one would first select a genre from Spotify, and scrape songs from pre-made playlists corresponding to the genre. This is the method employed for [5], [7] and [8]. The limitation of this dataset, however, is that the code for attaching genre labels is not provided. Description of the data collection method suggests that a genre was first selected, before tracks were scraped from it. Hence, we assume that the procedure follows that of [5] [7] [8]. An additional limitation of this form of data in general, is that the accuracy of genre labels are contingent on the accuracy of the associated playlists. Such labelling might differ across different music service providers, as genre labelling is inherently subjective, and a direction for future work would be to cross-check this dataset against the genre labels of other music service providers, to ensure accurate labelling. However, visual inspection of the data indicates that the genre labels seem accurate and we speculate that subsequent removal of songs with multiple genre labels could reduce inaccuracies.

Variable	Description	Data Type	Min	Max
Key	Estimated overall key (1 of 12 values – C, C#, D, …). Music is usually composed in a scale, and the key denotes the tonal center of this scale.	Categorical, Nominal	NA	NA
Mode	Major or Minor. Within a key, a track can be major (happy) or minor (sad).	Categorical, Nominal	NA	NA
Time Signature	Notation to specify how many beats are in each bar (0/4, 1/4, 2/4, 3/4, 4/4). 4/4 indicates 4 beats in a bar, so the meter would be 1, 2, 3, 4, 1, 2 …	Categorical, Nominal	NA	NA
Popularity	Measure of popularity, calculated mainly based on the number of plays of a track and how recent those plays are.	Numerical, Discrete	0 (Unpopular)	100
Duration	Duration (ms).	Numerical, Discrete	15387	5552917
Acousticness	Likelihood that the track is acoustic (contains natural sounds, non-electric).	Numerical, Continuous	0 (low likelihood)	1
Danceability	How suitable a track is for dancing.	Numerical, Continuous	0 (least danceable)	1
Energy	Measure of intensity and activity. Energetic tracks feel fast, loud, and noisy.	Numerical, Continuous	0 (low energy)	1
Instrumentalness	Likelihood that the track contains vocal content.	Numerical, Continuous	0 (high likelihood)	1
Liveness	Likelihood of presence of an audience in the recording.	Numerical, Continuous	0 (low likelihood)	1
Valence	Measure of musical positiveness.	Numerical, Continuous	0 (negative emotions)	1
Speechiness	Presence of spoken words.	Numerical, Continuous	0 (low speechiness)	1
Loudness	Overall loudness in decibels (dB).	Numerical, Continuous	-52.457 (soft)	3.744
Tempo	Estimated overall beats per minute (BPM).	Numerical, Continuous	30.38	242.90

Table 1: Description of audio features in the dataset, used for clustering and subsequent prediction.

Despite its limitations, this dataset has unique advantages. Firstly, there are nearly 10,000 tracks in most of its genres. The huge number of tracks allows for better training and for further sampling to test our conclusions. The size also allows us to select a subset of songs, each with only one genre label, for the "Classification" step. Secondly, the number of genres is also ideal. It is large enough such that clustering is viable and useful (as seen in Sections 3.1 and 5.1), but the number of tracks per genre is still substantial.

3.1 EDA and Data Treatment

Our dataset contains 232,725 tracks, covering 27 genre labels. We first notice that *Children's Music* was repeated as a genre label, due to inconsistent apostrophe parsing. We merge these two genres under one label, and note that there were no duplicate entries. We then obtain a tally of tracks by genre, which shows that *A Capella* only has 119 tracks. We hence remove *A Capella* due to a lack of samples, as it could affect the consistency of our clustering and classification results, leaving 232,606 samples in our dataset. We also note that there are 35,124 tracks with multiple genre labels, making up 91,075 samples. If we remove these samples, *Rap* has the least remaining tracks with a count of 982. Figure 1 shows the count of tracks by genre, before and after these samples are removed.

Next, we obtain the density of tracks in each genre across the 11 numerical variables. The distributions for each genre show considerable variability over audio features, especially for danceability, energy, and tempo. Figure 3 shows the variability amongst different genres for energy, suggesting the existence of "low-energy" cluster consisting

of *Classical*, *Opera*, and *Soundtrack*. Our EDA also suggests that *Comedy* has very distinct audio feature values, with a much higher likelihood of having high speechiness (Figure 3) and liveness. From these distributions, we shortlist possible clusters of genres. One such cluster contains *Hip-Hop* and *Rap*, and we compare the distribution of tracks in each genre across the numerical audio features (Figure 2), noting their similarity. Plots similar to Figure 2 suggest a second cluster with *Classical*, *Opera*, and *Soundtrack*, and a third with *Reggae* and *Reggaeton*. The results of our EDA conform to our intuition about these genres, and are hence reasonable.

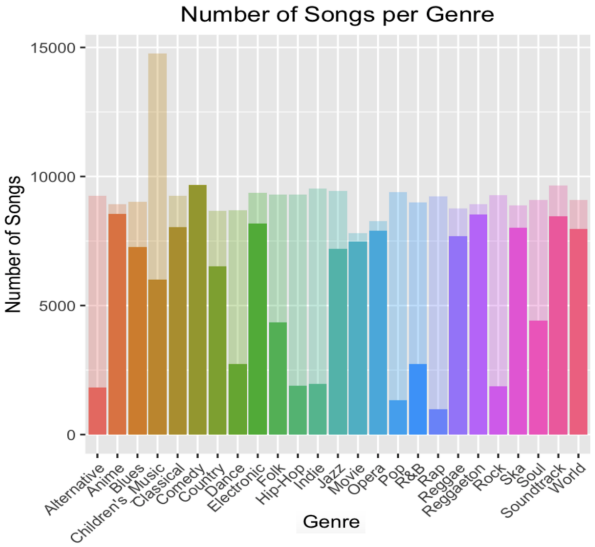

Figure 1: Count of Tracks by Genre, before (translucent) and After (opaque) Duplicates are Removed.

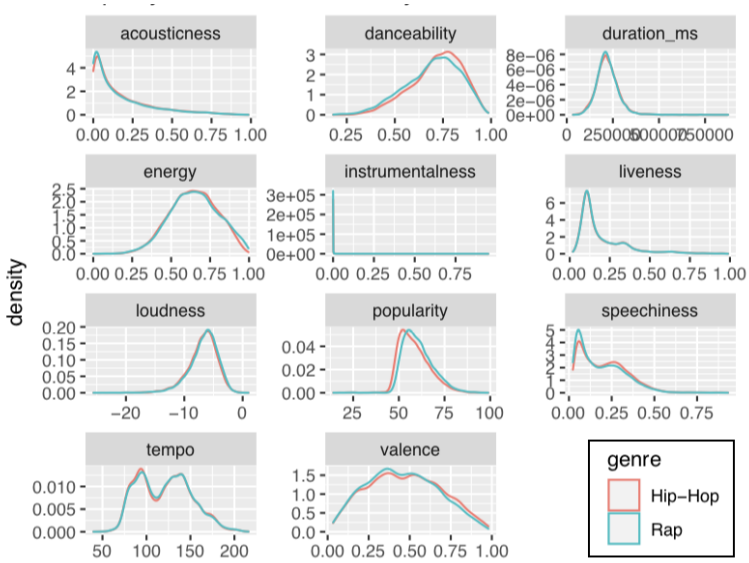

Figure 2: Distribution of Hip-Hop Rap Songs across Audio Features.

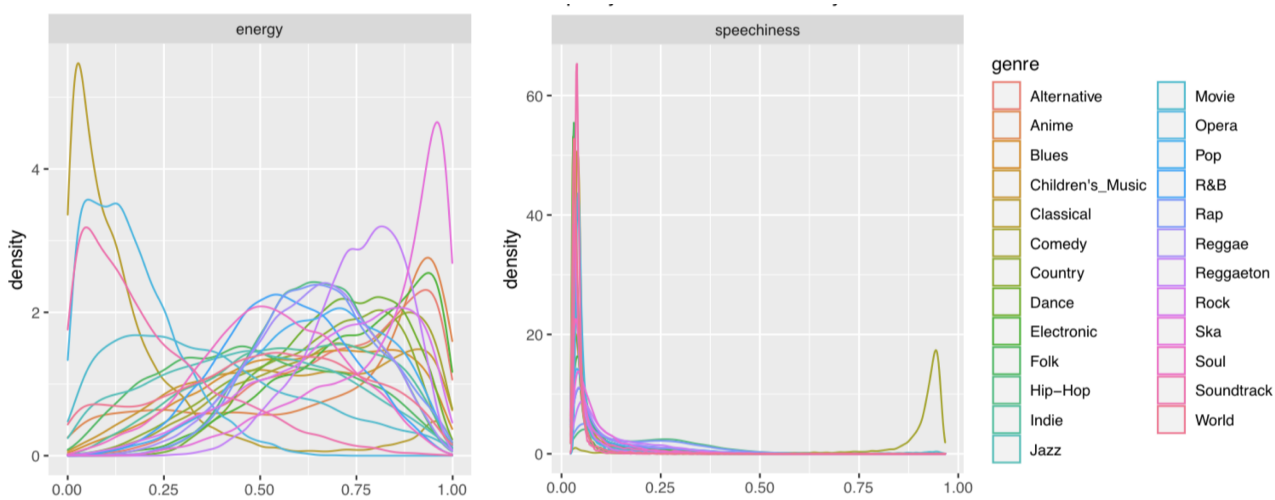

Figure 3: Distribution of Songs in Each Genre, across Energy (left) and Speechiness (right). Each curve plots the density across different values for all the tracks in a particular genre.

Following this EDA, we convert the categorical variables into indicator variables. For example, there were 12 new indicator variables corresponding to the 12 unique key values in the dataset. "C" was marked as 1 when the track had a key of C and 0 otherwise. We then centered and scaled all variables, for suitable clustering. We note that post-transformation, there is still significant variability across audio features.

4 Methods

To address the goals of this study, we first split our dataset, one to be used for clustering (clustering data) and the other for classification (classification data). We then use various clustering methods to cluster our 25 genres into K^* clusters. After this clustering, we will use a Random Forest to perform genre classification from audio features, with multiple different categories of genre selections (inter-cluster, intra-cluster, mix). As an extension, we shall also briefly explore a hierarchical approach for genre classification.

4.1 Clustering - Classification Split

The existence of songs with multiple labels complicates the classification problem. Multi-label classification is a relatively new problem, and there are few implementations of the algorithms we use in this paper (Random Forest, Gradient-Boosted Decision Trees) that are well adapted to this problem. Current adaptations require us to set prior probabilities over the multiple labels and penalize deviations from these priors when assessing predictions. We do not have knowledge about these probabilities, and while we could tweak the way these assessments are done, we would need to adjust this for different sets of genre selections (our classification step uses 77 sets). It is also relevant to note that existing work on using audio features for genre classification [5] [6] [7] [8] deals with the single-label classification problem. Therefore, to simplify our classification problem and limit the scope of this work, we shall leave the multi-label classification problem for future study.

To select our classification data, we hence first remove all tracks with multiple labels. *Rap* now has the least number of samples (982). As such, we randomly select 490 samples (approximately half of 982) from each genre, to create our classification data. If we took more samples from *Rap*, then our clustering data might not have sufficient *Rap* tracks that belonged to *Rap* alone, and our resultant clusters may not be representative. More samples for other samples was then not necessary because the train and test splits for the classification step always contained an equal proportion of each genre. The classification data hence consists of 12,250 tracks equally distributed across 25 genres. The clustering data consists of the remaining 220,356 $(232,606 - 12,250)$ samples.

4.2 Clustering

We implement two forms of clustering, and define them as such: Track-level clustering involves clustering each track. This is often then compared to the "ground-truth" clustering, their genre labels. Genre-level clustering takes each genre as a point, and clusters these 25 points. There is no "ground-truth" for this.

4.2.1 Algorithms and Metrics

To motivate the choice of algorithms, we shall give a brief description of the clustering algorithms used in this paper, along with how they are implemented here and their unique advantages and assumptions. Henceforth, we assume the data contains M points in N dimensions, and we wish to cluster into K clusters.

K-means takes the $M \times N$ input matrix and first initializes K cluster centers. Then, it assigns points to the cluster centers based on Euclidean distance, before setting the centers to the mean of these points. These steps are repeated to convergence, and the algorithm's advantage is its speed [13].

Hierarchical Clustering takes an $M \times M$ distance matrix D, D_{ij} being the distance between points i and j, and defines the distance between two clusters to be the maximum distance between their elements. At each step, the two nearest clusters are merged into a new cluster [14], until there are K clusters. This might be appropriate due to the hierarchical structure of genres [9], but is also an assumption we make when using this algorithm.

Spectral Clustering takes an $M \times M$ affinity matrix A, where A_{ij} is a measure of similarity of points i and j, and clusters such that the size of each cluster is approximately equal [15]. This is useful to obtain meaningful clusters, and when we have a metric of similarity between points. However, we acknowledge that this might not be a valid assumption as the sizes of genre clusters can be different.

Constrained K-means (CK-means) performs similarly to K-means, but additionally takes in two sets of constraints, must-links and cant-links. Two points with a must-link should be clustered together, and two points with a cant-link should not be. The algorithm tries to maximize the number of constraints satisfied. This is useful for track-level clustering, such that tracks in the same genre end up in the same cluster.

When using these algorithms for track-level clustering, we generally make the assumption that tracks close to each other (as defined by the distance metric used) in euclidean space (as defined by their features) should be clustered together. This is appropriate if we assume that genres with songs that have similar audio features are similar to each other. Such an assumption is valid given that our purpose is to address classification accuracy when predictions are made based on these audio features, but it should therefore be noted that our derived genre clusters may not be intuitive, as they reflect similarity in terms of audio features.

We use the Adjusted Rand Index (ARI) to assess the degree of similarity between two cluster assignments X and Y. 0 indicates that one assignment is random, and 1 indicates that the two assignments are in complete agreement. Let X have r clusters and Y have s clusters, then we denote $X = \{X_1, \ldots, X_r\}$, $Y = \{Y_1, \ldots, Y_s\}$, where Y_1 contains the elements that Y assigns to cluster 1. Now define $n_{ij} = |X_i \cap Y_j|$, $n = \sum_{k=1}^{r} \sum_{l=1}^{s} n_{kl}$,

$a_i = \sum_{k=1}^{s} n_{ik},\ b_j = \sum_{l=1}^{r} n_{lj},\ \ 1 \le i \le r,\ 1 \le j \le s$. Then the ARI between X and Y is:

$$ARI_{XY} = \frac{\sum_{i=1}^{r}\sum_{j=1}^{s}\binom{n_{ij}}{2} - \left[\sum_{i=1}^{r}\binom{a_i}{2}\sum_{j=1}^{s}\binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_{i=1}^{r}\binom{a_i}{2} + \sum_{j=1}^{s}\binom{b_j}{2}\right] - \left[\sum_{i=1}^{r}\binom{a_i}{2}\sum_{j=1}^{s}\binom{b_j}{2}\right]/\binom{n}{2}}$$

4.2.2 Clustering Approach

We first determine which predictor variables to use, by performing track-level K-means clustering on the full dataset, setting $K = 25$, and assessing the accuracy of these assignments against the "ground-truth" with the ARI. We iteratively remove variables, starting with categorical variables, excluding variables that lead to significantly lower accuracy from further analysis. We also use this to assess the suitability of K-means to perform track-level clustering, such that tracks in the same genre are clustered together. Note that this variable selection assumes that variables important for K-means are important for the other clustering algorithms. We recognize this limitation, but make this assumption to simplify further implementation, and because the algorithms all use a distance-based metric to quantify similarity. We additionally conduct sensitivity analysis by performing hierarchical clustering on a subset of data, which randomly samples 100 tracks from each genre, to ensure that our variables are significant.

Then, we proceed to perform genre-level clustering. We first separate tracks by genre, and for each genre, obtain the mean for each selected variable. This gives us a $25 \times N$ matrix G, where N is the number of selected predictor variables. We now describe the method to obtain each clustering assignment in Table 2. *Spectral* is obtained by taking G, and computing a 25×25 distance matrix D using the radial basis function kernel, before using the k-nearest neighbors approach to obtain affinity matrix A, used for Spectral Clustering. $Hierarchical$ is obtained by taking G, and computing distance matrix D by calculating the Euclidean distance between points, before performing Hierarchical Clustering. $K\text{-}Means$ is obtained by performing K-means on G. Lastly, we also analyze tracks with multiple genre labels. We first initialize a 25×25 affinity matrix A with 0s. Next we sort the genres alphabetically, and store them in vector g, $g = (Alternative, \ldots, World)$. Now, for every track i with genre label g_k and g_l, we increment A_{kl} by 1. The choice to "weight" each assignment equally was arbitrary, as we had no strong reason to favor an assignment over another. We note that no track is labelled with the same genre label twice. We then perform Spectral Clustering on A, calling the resultant clustering assignment $Pairs$.

We then vary K from 1 to 24, computing the 4 clustering assignments for each K, and obtaining the ARI between each pair of assignments. The 6 ARI values are summed together to get T_K, which determines the general agreement of each clustering method, for each K. We then select $K = K^*$ that maximizes T_K, with the consideration that K^* cannot be too small (many genres in the same cluster), nor too large (many clusters with just one genre). This ensures that our clustering assignment is meaningful.

We then compute the 4 clustering assignments for $K = K^*$, taking note of their similarities. To check that these similarities are robust, we conduct track-level CK-means. Due to the high computational cost of this algorithm, we randomly select 50 tracks from each genre, to form 2 batches of 1250 tracks each. For each batch, we randomly specify n must-links between tracks in a genre. n must be small enough such that the tracks are not all clustered in the same genre, but large enough such that tracks of the same genre are clustered together. n will be selected by varying n slowly, taking note of cluster assignments. CK-means will then be used to cluster each batch into K^* clusters. A genre will then be assigned to a cluster, depending on which cluster contains the most tracks from that genre. Assignments for batch 1 and 2 will be labelled $CK\text{-}Means1$ and $CK\text{-}Means2$ respectively. We then check if these 2 assignments have similar characteristics as the previous 4.

We then aggregate these 6 clustering assignments into $Final$, the final clustering assignment. To do so, we again initialize a 25×25 affinity matrix A with 0s. For each clustering assignment, every time genre g_k and g_l were clustered together, we increment A_{kl} by 1. Spectral clustering is performed on A, with $K = K^*$ to obtain $Final$. We shall note similarities and differences between $Final$, and the other 6 clustering assignments. For sensitivity analysis, we also perform CK-means once more on another batch of 1250 tracks to form $CK\text{-}Means$ 3 (Appendix).

4.3 Classification

Having now obtained clusters of genres, we proceed to classify our data. We first select the number of genres, p, and then select p genres from three possible categories of selections. Category 1 includes selections such that each genre belongs to a different cluster. Category 2 includes selections such that each genre belongs to the same cluster. Category 3 includes selections such that there are genres belonging to the same cluster, yet not all genres

are in the same cluster. For each category, we shall attempt to randomly obtain 5 selections when possible, to assess accuracy robustly. For each selection, we then do 5-fold cross validation, each fold containing an equal proportion of each genre, and storing the mean of the classification accuracy results (our error being the misclassification error) to ensure that our results are robust. As a benchmark, we define good accuracy as one with classification accuracy over 70% (human-level accuracy [19]).

For our data, the predictor variables identified in Section 4.1.2 shall also be the predictor variables for classification, and the response variable will be the genre label. For each selection of genres, we use a Random Forest Classifier [17], setting the number of trees to 500, and training on the corresponding train data, before obtaining predictions on the test data, classifying each observation according to the class that receives the majority vote amongst the trees. From our classification results, we shall make some initial conjectures, which shall be verified by varying p, performing 5-fold cross-validation, and using a second prediction algorithm, XGBoost [18], which implements Gradient Boosted Decision Trees. These two algorithms were selected because they gave the highest accuracy for works on genre classification using audio features [5] [6] [7] [8].

4.3.1 Hierarchical Approach

To provide additional sensitivity analysis and to draw additional conclusions, we briefly explore a hierarchical approach where we first select l clusters, and then classify tracks into clusters, before classifying tracks further into their genres. We first select $l = K^*$ (all), and then 5 random selections for $l = 2$. For each cluster selection, we split the tracks belonging to the corresponding genres into an $80\%-20\%$ train-test split, with both splits having an equal distribution of tracks across genres. We then record the classification accuracy values on the test set when using the Random Forest model, first performing genre classification normally (as per Section 4.3), and then applying this hierarchical approach. A more systematic procedure to evaluate this approach can be explored in future work.

Genre	Final	Pairs	Spectral	Hierarchical	K-Means	CK-Means 1	CK-Means 2
Dance	1	1	1	1	1	1	1
Pop	1	2	1	1	1	2	1
Hip-Hop	1	2	1	1	1	2	1
Rap	1	2	1	1	1	2	1
Classical	2	3	2	2	2	3	2
Opera	2	3	2	3	3	3	2
Soundtrack	2	4	2	2	2	3	2
Reggaeton	3	2	3	4	4	4	1
Reggae	3	6	3	4	4	4	3
Ska	3	6	3	4	4	4	3
Children's Music	4	5	4	5	5	2	4
R&B	4	1	1	6	5	2	1
Soul	4	1	5	6	5	1	4
Folk	4	7	5	6	5	5	4
Indie	5	7	6	6	5	1	1
Alternative	5	8	6	6	5	6	3
Country	5	7	6	6	5	5	3
Rock	5	7	6	6	5	2	1
World	6	3	7	5	6	5	5
Electronic	6	3	7	5	6	6	6
Jazz	6	3	5	5	6	6	3
Movie	7	4	4	7	7	3	3
Anime	7	4	7	5	6	4	3
Blues	7	7	4	5	6	1	3
Comedy	8	7	8	8	8	7	7

Table 2: Clustering Assignments for Various Clustering Methods. Within each column, genres with the same number are clustered together.

5 Results

5.1 Clustering

Track-level K-means clustering yields a clustering assignment with ARI value of 0.038 when compared to the "ground-truth" genre labels. This increases to 0.094 when indicator variables corresponding to time signature and key were removed, and increases to 0.102 when the indicator variable corresponding to mode (Mode) was removed. Further removal of numerical variables do not change this value significantly. For sensitivity analysis, we take a random subset of 100 samples from each genre and perform Hierarchical Clustering. Similarly, removing indicator variables corresponding to time signature and key increase the ARI, and further removal of numerical variables do not change this value significantly. However, removing Mode decreased the ARI slightly. We hence selected all variables excluding time signature and key, as the remaining variables might be useful for different clustering methods. We also note that the low ARI values in general suggest that unsupervised track-level clustering will be unable to cluster songs from the same genre together, providing justification for our use of CK-means later.

The 4 clustering assignments, *Pairs*, *Spectral*, *Hierarchical*, and *K-means* have the most agreement when

$K = 3$, followed by $K = 8$. As explained, we do not want K to be too small. We hence obtain clustering assignments for $K = 8$, summarizing our results in Table 2, with $n = 25$ for our implementation of CK-Means. With these 6 clustering assignments, we compute our final clustering assignment, *Final* (Table 2). *Final* largely clusters genres in a way that matches intuition, our EDA, and the other 6 cluster assignments. To conduct sensitivity analysis, we noted that *Pairs* did not use any audio feature data. To ensure that *Final* was still relevant for our classification step, we aggregated the remaining 5 clustering assignments (excluding *Pairs*), creating a new clustering assignment *Final*2 following the same procedure used to create *Final*. *Final*2 had an ARI of 0.784 when compared with *Final*, providing support for *Final*, and ensuring that it was reflective of audio features.

5.2 Classification

We obtain accuracy values for different genre selections, and plot the results (Figure 4). Each box displays the 25^{th}, 75^{th} percentile, and mean, of 5 values. Each value is the mean of the results from cross-validation, for a particular genre selection. The same genre selections were used for both the Random Forest and XGBoost.

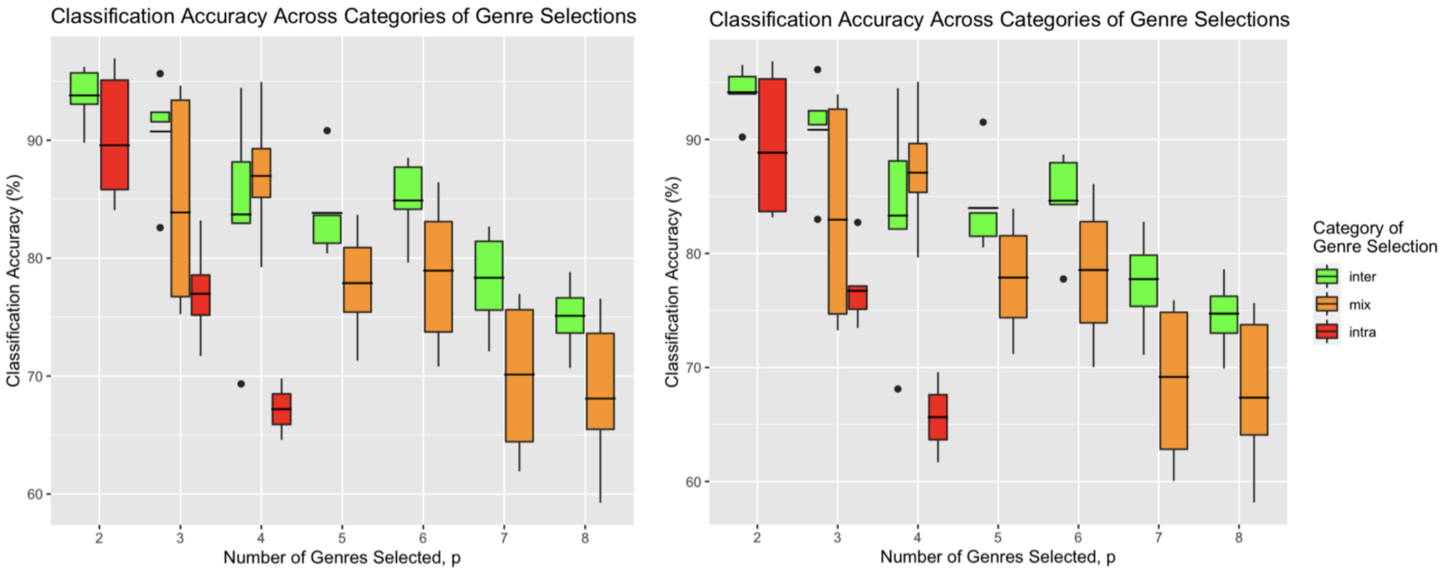

Figure 4: Classification Results for the Random Forest (left) and XGBoost (right).

5.3 Hierarchical Approach

We obtain accuracy values for different cluster selections, and summarize the results in Table 3.

6 Discussion

6.1 Clustering Results

In this paper, we have carried out 7 clusterings of the 25 genres, aggregating 6 of them for our proposed clustering assignment used in the classification step. Some decisions were admittedly pretty arbitrary, because there was a lack of prior work done in the field of genre clustering based on audio features, and there was often no motivation to pick one choice over another. To address this, we implement a variety of clustering algorithms, at different levels (genre-level, track-level), and using different kinds of information (audio features, or tracks that belong to multiple genres). Each of these approaches have their unique assumptions that might be suitable for the problem. As such, the variety of methods employed can be seen as a form of sensitivity analysis, first to see if genres exist in clusters, and if they do, the varying levels of confidence we have regarding these clusters.

Clusters	Genre Accuracy	Cluster Accuracy	Hierarchical Accuracy
5, 8	74.1%	99.8%	73.7%
3, 6	65.6%	86.1%	65.1%
5, 7	68.7%	93.4%	68.4%
2, 3	76.0%	98.0%	76.4%
2, 5	72.7%	97.0%	71.2%
All	50.5%	65.1%	49.3%

Table 3: Classification Results for the Normal and Hierarchical Approach. Cluster numbers correspond to *Final* in Table 2. Genre Accuracy is the accuracy obtained when we do genre classification directly. Cluster Accuracy is the accuracy when first classifying tracks into their clusters. Hierarchical Accuracy is the final accuracy of genre classification, after having first classified songs into their clusters.

The 4 genre-level cluster assignments generally agree on the following cluster assignments: $Dance^1$, Pop^1, $Hip\text{-}Hop^1$, Rap^1; $Classical^2, Soundtrack^2$; $Reggaeton^3$, $Reggae^3$, Ska^3; $R\&B^4$, $Soul^4$; $Indie^5$, $Alternative^5$ $Country^5$, $Rock^5$; $World^6$, $Electronic^6$, $Jazz^6, Comedy^8$, with genres with the same superscript often clustered together. Other genres did not have consistent clustering patterns. Clusters 1, 2, 3, 8, as denoted above, are supported by $CK\text{-}Means1$ and $CK\text{-}Means2$, but there is a non-negligible level of uncertainty involved with the other clusters (Table 2). $CK\text{-}Means$ 3 reflects similar uncertainty regarding other clusters, with support for clusters 1, 2 and 8. These results signal that genres could indeed lie in clusters. We are most confident about clusters 1, 2 and 8, moderately confident about cluster 3, and slightly confident about clusters 5 and 6. The clusters that we are confident about are also generally supported by our intuition and EDA. As a general note, however, caution should be taken when making inferences from $CK\text{-}Means$ 1, 2, and 3, because each batch of data only contains 1,250 tracks, and may hence be unrepresentative.

We also note that the results of this work may point to varying "levels" of clusters. We previously noted high agreement between cluster assignments when $K = 3$. We also note that $R\&B$ is sometimes clustered with cluster 1, or that while there is only slight confidence about cluster 5, that $Country$ and $Rock$ are often clustered together. There could therefore be "super-clusters" or "sub-clusters" that would be interesting to investigate as future work.

6.2 Classification Results

As expected, selections from Category 1 generally yield the highest accuracy, followed by those in Category 3, and then Category 2. We perform sensitivity analysis by using XGBoost, and varying p, which generally supports this assertion. This provides support for the accuracy of $Final$. This also indicates that there is sufficient variability in audio features between clusters, but not so much within each cluster. Using 70% as a benchmark for "good" accuracy, we can hence see that companies may therefore be able to capitalize on the advantages of genre classification when their genre selections span different clusters, but not when they belong to the same cluster(s).

We also note that there exists "outliers" in our plots - some inter-cluster genre selections having lower accuracy and some mix-cluster genre selections having higher accuracy values. This indicates that our clusters may not be entirely accurate, as previously noted in Section 5.1. Further investigation with respect to these "outliers" show that they generally belong to genres that we previously expressed uncertainty about.

6.3 Hierarchical Approach Results

We note that there is generally no difference in accuracy when doing genre classification directly, or applying the hierarchical approach. It might therefore not be worthwhile to further explore this approach. However, the results do support findings in earlier sections, and can be used for sensitivity analysis.

We note that classification into clusters is generally very accurate ($> 90\%$), which therefore supports $Final$, our assertion that inter-cluster variability is significant, and our claim that genre classification using audio features could be suitable for inter-cluster genre selections. We also note a significant decrease in accuracy after having classified tracks into their corresponding genres. This, once again, shows that the predictive power of audio features for intra-cluster genre selections is limited. Non-negligible variability in cluster classification accuracy ($86.1\% - 99.8\%$) could also indicate uncertainty in our clustering result for clusters 3 and 6, the possible existence of "super-clusters", or additional relationships between genres that we have yet to consider with hard clustering assignments.

7 Conclusion

In conclusion, this work provides support for the hypothesis that similar genres exist in "clusters", that this similarity is reflected in Spotify audio features, and consequently, genre classification based on audio features is significantly affected by pre-classification genre selection. This work also suggests a possible clustering assignment, $Final$, for the 25 genres selected in this dataset, with varying levels of confidence over the clusters suggested. Using this assignment, we note that using audio features for genre classification is most suitable when genres exist in different clusters. We hope that this work can inform companies if using audio features to perform genre classification will be suitable for their classification purposes, thereby allowing them to capitalize on the method's relatively low computational cost and interpretability. We also hope that this paper can raise awareness about the need for prudent genre selection, prior to classification, for works that focus on model selection. Ultimately, we believe that more research about the relationship between genres is integral to the problem of genre classification, and we wish that this work can lead to further investigation of this topic.

SUPPLEMENTAL MATERIALS

Submission III.Rmd: Single file that contains all the code used to reproduce results in this paper (R Markdown). Package versions: `dplyr == 0.8.5`, `tidyr == 1.0.2`, `ggplot2==3.3.0`, `tidyverse == 1.3.0`, `repr == 1.1.0`, `Dict == 0.1.0`, `fcd == 0.1`, `mclust == 5.4.6`, `caret == 6.0-86`, `fossil == 0.4.0`, `kernlab == 0.9-29`, `conclust == 1.1`, `randomForest == 4.6-14`, `xgboost == 1.0.0.2`

References

[1] Spotify (2020). Get Audio Features for a Track. Retrieved from https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/.

[2] D. A. Huang, A. A. Serafini, E. J. Pugh (2018). Music Genre Classification. Retrieved from http://cs229.stanford.edu/proj2018/report/21.pdf.

[3] H. Bahuleyan (2018). Music Genre Classification using Machine Learning Techniques. *arXiv preprint arXiv:1804.01149.*

[3] T. Dammann and K. Haugh (2017). Genre Classification of Spotify Songs using Lyrics, Audio Previews, and Album Artwork. Retrieved from http://cs229.stanford.edu/proj2017/final-reports/5242682.pdf.

[4] A. J. H. Goulart, R. C. Guido, C. D. Maciel (2012). Exploring different approaches for music genre classification. *Egyptian Informatics Journal*, 13(2), 574-579.

[5] A. Gittleman (2019). Genre Classification with Spotify API. Retrieved from https://github.com/aggittle/Genre-Classification-with-Spotify-API.

[6] I. Basyar (2019). Spotify Genre Classification. Retrieved from https://www.kaggle.com/iqbalbasyar/spotify-genre-classification.

[7] K. Pavlik (2019). Understanding + Classifying Genres Using Spotify Audio Features. Retrieved from https://www.kaylinpavlik.com/classifying-songs-genres/.

[8] Y. Dua (2019). Name That Genre. Retrieved from https://towardsdatascience.com/music-genre-prediction-with-spotifys-audio-features-8a2c81f1a22e.

[9] Musicmap (2020). The Genealogy and History of Popular Music Genres. Retrieved from https://www.musicmap.info.

[10] J. D. D. Santos (2017). Discovering similarities across my Spotify music using data, clustering and visualization. Retrieved from https://towardsdatascience.com/discovering-similarities-across-my-spotify-music-using-data-clustering-and-visualization-52b58e6f547b.

[11] S. Malpani (2019). Clustering Methods in R - Spotify Songs. Retrieved from https://www.kaggle.com/surajpm/clustering-spotify-songs-using-r/data.

[12] Z. Hamidani (2019). Spotify Tracks DB. Retrieved from https://www.kaggle.com/zaheenhamidani/ultimate-spotify-tracks-db.

[13] J. A. Hartigan and M. A. Wong (1979). Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics*, 28, 333-344.

[14] C. Yau (2020). R: Hierarchical Clustering. Retrieved from http://www.r-tutor.com/gpu-computing/clustering/hierarchical-cluster-analysis#:~:text=The%20hclust%20function%20in%20R,distance%20between%20their%20individual%20components.

[15] A. Y. Ng, M. I. Jordan, Y. Weiss (2001). On Spectral Clustering: Analysis and an algorithm. *Neural Information Processing Symposium*, 849 - 856.

[16] K. Wagstaff et al. (2001). Constrained K-means Clustering with Background Knowledge. *Proceedings of the Eighteenth International Conference on Machine Learning*, 577-584.

[17] L. Breiman (2001). Random Forests. *Machine Learning*, 45(1), 5-32.

[18] T. Chen and C. Guestrin (2016). XGBoost: A Scalable Tree Boosting System. *22nd SIGKDD Conference on Knowledge Discovery and Data Mining*, 785–794.

[19] M. Dong (2018). Convolutional Neural Network Achieves Human-level Accuracy in Music Genre Classification. *arXiv preprint arXiv:1802.09697.*

Appendices

A CK-Means 3

Genre	CK-Means 3
Dance	1
Pop	2
Hip-Hop	2
Rap	2
Classical	3
Opera	4
Soundtrack	3
Children's Music	5
Reggaeton	6
Reggae	1
Ska	1
R&B	2
Soul	2
Folk	1
Indie	2
Alternative	6
Country	1
Rock	1
World	5
Electronic	5
Jazz	1
Movie	4
Anime	6
Blues	5
Comedy	7

Table 4: CK-Means 3 Cluster Assignment.