Coursera - IBM - Unsupervised Learning
Spotify Audio Features

January 20, 2026

1 Overview

I have chosen the Kaggle Spotify Audio Features dataset for my final project.

This dataset contains lots of information about track in the form of "Audio Features". These are characteristics of music defined by Spotify that tell something about a specific property of the music. For example, "Acousticness" is a measure of how much the music contains acoustic instruments, as opposed to electronic instruments. "Danceability" and "Energy" are two other measures which seem related, but encode different types of energy in a track. More information about the meaning of the audio features can be found on the Spotify Developer Website

1.1 Project Goal

The dataset contains a column with the genre of the track. However, I am interested in looking at whether genre is actually a good measure of where a track belongs in the Audio Feature space. Supervised learning was used to group similar songs together based on audio features. Since advanced clustering methods can be slow, I will also reduce the dimensionality of the feature space as much as possible using PCA. As we will also see, there is some redundancy in the audio features due to correlations. It is unlikely that direct genres can be recovered from the audio features as different genres can often have similar audio feature parametrization, but it's expected that some clear groups of similarly classifiable tracks emerge.

1.2 Models

The following unsupervised learning methods were used:

- PCA is used for dimensionality reduction
- Gaussian Mixtures are used for grouping similar tracks together
- HDBSCAN for verifying the results of GMM.

Reducing the dimensionality helps with model fitting, the dataset is very large (¿1.000.000 rows) and as such will take a lot of computing power. Reducing the dimensionality by a few components will already help a lot for model selection iteration. As we will see, the data does not contain any natural clusters. Therefore I have chosen GMM to interpret latent features and soft clusters. Relating these back to the original data will give some insight into how tracks group together based on audio features.

Finally, for additional interpretability I will use HDBSCAN to verify the results. I attempted to use an agglomerative model, which would have been interesting since it builds clusters by combining similar subclusters, something that sounds like it would fit a dataset of music properties. But I ran into scaling issues where the amount of memory was too much for my system to handle. Another attempt was made with Factor Analysis, but it seems this dataset is not suited for this analysis, as I ran into numerical issues during the fitting. There is likely a way to make factor analysis work for this dataset, but since that is not aligned with the goal of this report that will have to be postponed.

1.3 Data Description

The full dataset contains 1.159.764 rows with 20 features

- 'id' → Int
- 'artist_name' → String
- 'track_name' → String
- 'track_id' → String
- 'popularity' → Int
- 'year' → Int
- 'genre' → String
- 'acousticness' → Float
- 'danceability' → Float
- 'duration_ms' → Int
- 'energy' → Float
- 'instrumentalness' → Float
- 'key' → Int
- 'liveness' → Float
- 'loudness' → Float
- 'mode' → Int
- 'speechiness' → Float
- 'tempo' → Float
- 'time_signature' → Int
- 'valence' → Float

In this analysis we'll restrict ourselves to the actual audio feature columns (acousticness, danceability, energy, instrumentalness, liveness, speechiness, valence), leaving 7 features to consider.

First we will perform EDA to see if there are any major issues with the data that need to be addressed.

Then we will investigate how much these features cover the variance of the data by performing PCA. It might be that some features are redundant or capture similar properties.

Then we use a gaussian mixture model to investigate which clusters are present in the data.

Finally hierarchical clustering is used to find out the process of clustering.

2 EDA and preprocessing

The data was investigated and no null or nan values were present. The audio features were already mostly within the 0.0-1.0 range, but to be sure they were all scaled to that exact range by using the MinMaxScaler.

The distribution of artists and genres was checked to identify if any were represented in a unbalanced manner. For the artists it seemed that there was a lot of traditional and classical music and a lot of electronic dance music, but the largest value ("Traditional" at 4058 occurrences) only consisted of 0.3% of the dataset.

artist_name	len
str	u32
"Traditional"	4058
"Grateful Dead"	2320
"Johann Sebastian Bach"	2125
"Giacomo Meyerbeer"	1345
"Elvis Presley"	1242
"Wolfgang Amadeus Mozart"	1084
"Armin van Buuren"	1061
"Astor Piazzolla"	932
"Hans Zimmer"	863
"Andrei Krylov"	841

Figure 1: The distribution of the 10 most represented artists.

The genres were plotted (see below) to discover that the most represented genre was "Black-Metal" at 1.7% of the dataset.

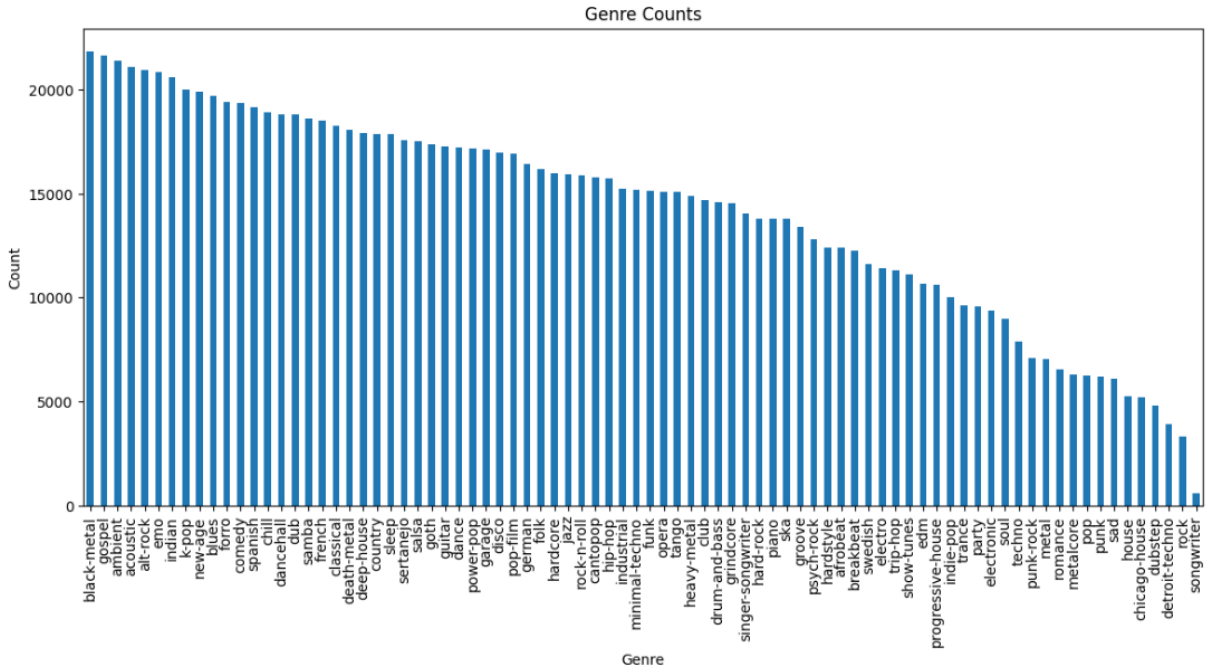

Figure 2: The distribution of genres in the dataset.

In both cases this does not represent a significant skew towards a specific genre or artist.

A pairplot was also made, which shows that no immediately identifiable clusters exist. There is some skew in the data with instrumentalness and acousticness tending towards low values. No steps were taken to remedy this.

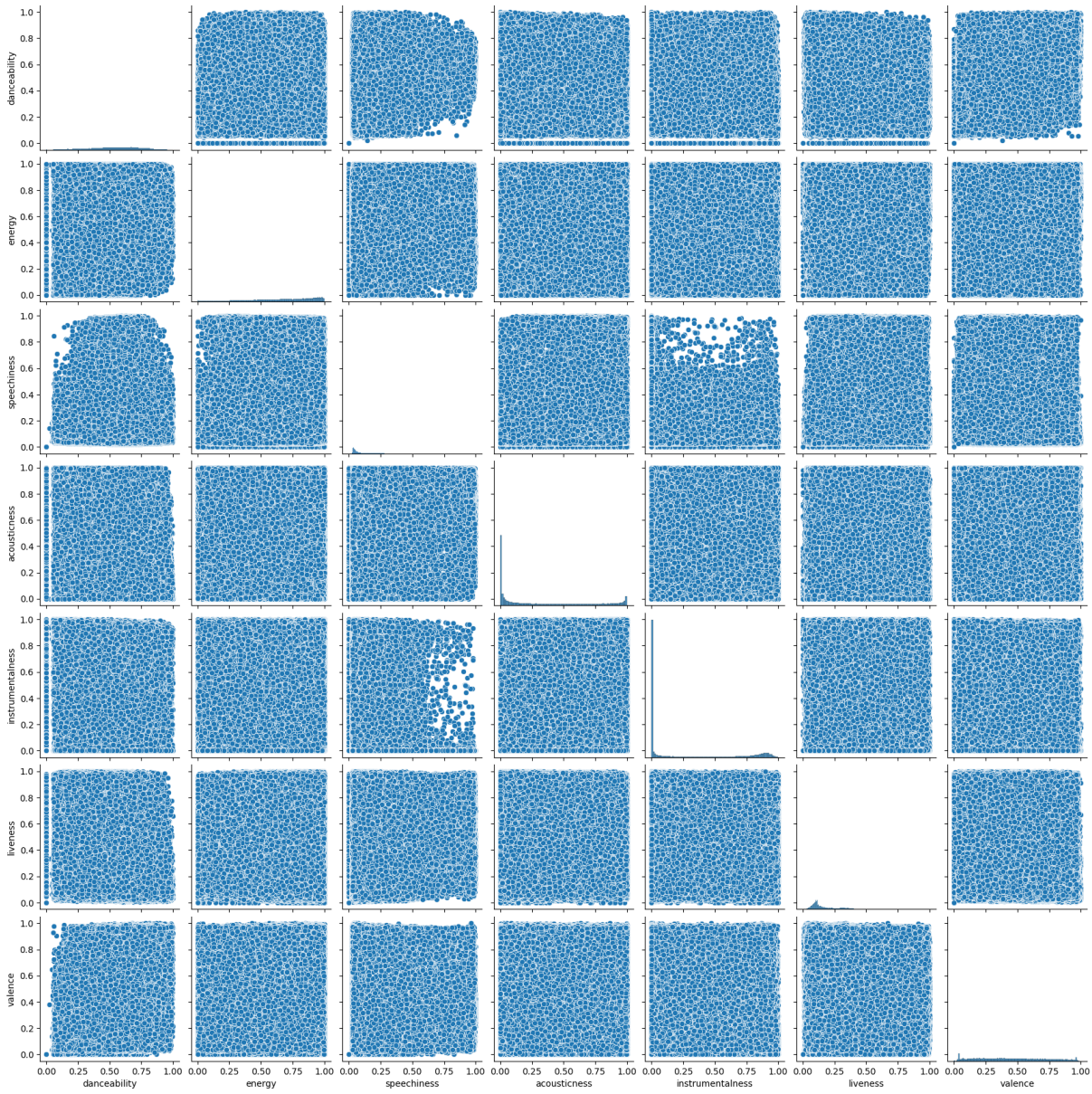

Figure 3: The pairplot for the scaled audio features.

The pairplot does not give any meaningful insight in what types of correlations we can expect. At least when plotting the data on a 2D projection to the feature axes, we can't see any clustering.

The only thing that can be glanced from the above plot is

- High speechiness is unlikely to go with middle levels of instrumentalness.
- High speechiness is unlikely to go with low danceability.

Which both seem to make intuitive sense from a music perspective. The rest of the data seems to be relatively uniformly distributed throughout the feature space.

From the correlation matrix we can see that some of the features have moderate to high correlations, specifically:

- (0.52) Valence/Danceability
- (-0.75) Acousticness/Energy

So a lot of high energy songs will likely have a positive sentiment (high valence) and many acoustic songs will likely be low energy.

From this it looks like there is an opportunity for dimensionality reduction. Since no clear grouped data points in the form of identifiable clusters were present in the data we can probably get away with taking a lower variance threshold if that means taking fewer components. Later we will be using Gaussian Mixtures to find similarity between groups of points which can be slow on large datasets. Since this is mostly a model that already provides more nuance by yielding probabilities instead of hard class labels, we can do away with some more variance if this means being able to iterate model selection quicker.

3 PCA for dimensionality reduction

By fitting a PCA model it was possible to remove 2 features at the cost of 6.2% variance. The resulting components are shown below:

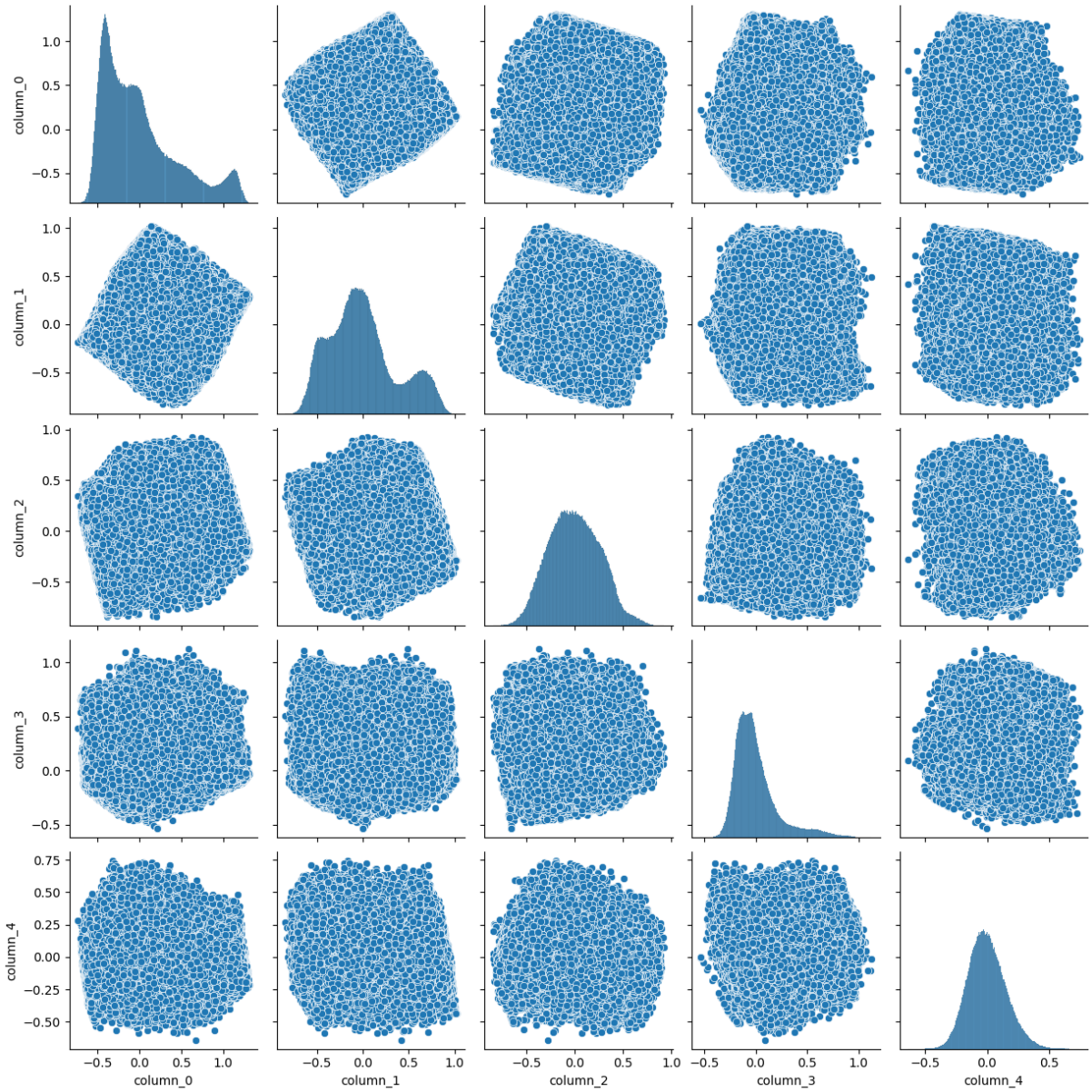

Figure 4: The reduced feature space for the dataset after removing 2 components to retain 93.8% variance.

From this we can see that there is still no clear clustering visible in this dataset. This is likely due to the fact that different types of music often span large ranges of values for music features and as such will overlap in the feature space. A soft clustering method like GMM might be able to reveal latent groupings (indirectly observable clusters) by allowing probabilistic cluster membership.

4 Gaussian Mixtures for audio feature grouping

We'll use the previously obtained dimensionality reduced features as input for the Gaussian Mixture model. This clustering method will fit a preselected amount of Gaussians to the data and attempt to assign cluster membership probabilities to each data point. Because there is no apparent best choice for the number of components, we'll use model selection for GMM provided by sklearn.

This works by using grid search with a GMM model and providing the following parameters:

- n_components
- covariance_type
- scoring = gmm_bic_score

Where the scoring method is the "Bayesian Information Criterion" (BIC), which is defined as BIC $= k\ln(n) - 2\ln(\hat{L})$, where k is the number of parameters in the model, n is the number of data points and $\hat{L}$ is the maximized value of the likelihood function of the model. The goal of this scoring criterion is to avoid overfitting by keeping the number of parameters low. This is similar to using regularization in regression models, but with the number of parameters instead of the size of the parameters.

However, before fitting the model, we need to standardize the dataset to improve the accuracy of the model. We'll do this by applying 'sklearn.preprocessing.StandardScaler'.

By applying gridsearch with to find the best parameters it became clear that there is no natural clustering in the data. As can be seen from the gridsearch results below, the 'full' covariance description yielded the best fits for the GMM model.

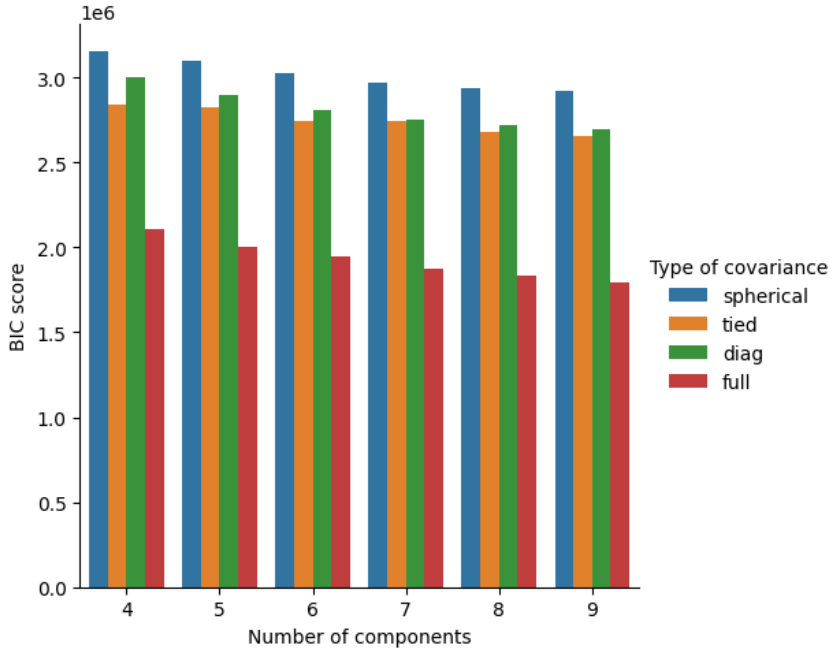

Figure 5: Gridsearch results for GMM models with different types of covariance representations.

And though the BIC score went down with increasing number of components, there is little advantage over adding more components. Especially since there is no clear point of diminishing returns and adding components will harm the interpretability of the model.

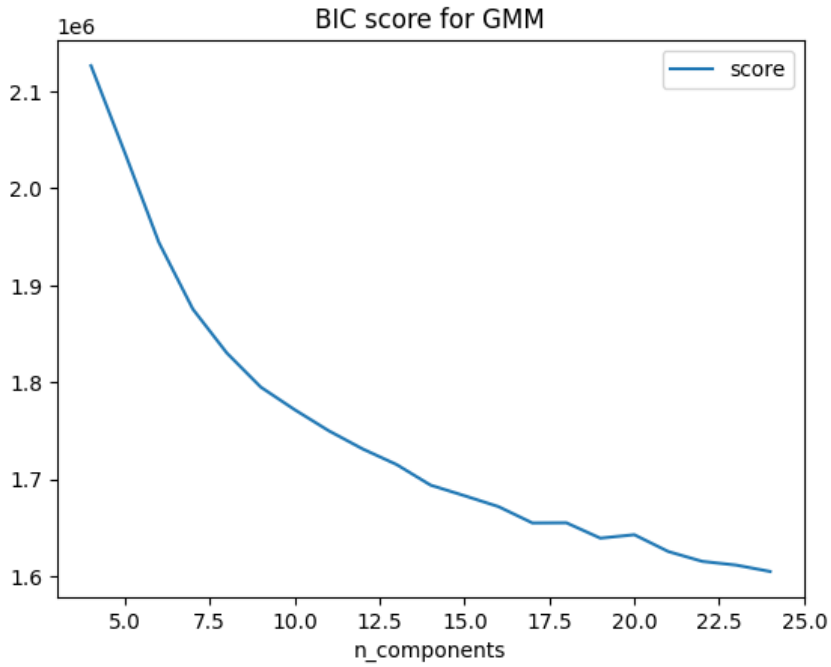

Figure 6: BIC curve for different amounts of components in GMM gridsearch fit.

From this we can conclude that the musical features as defined by spotify are not good predictors of musical genre as classically defined, which is something we already saw during EDA. However, we can try to see if another set of musical archetypes naturally appears from the analysis.

5 GMM as archetype descriptor

To keep the model interpretable and attempt to keep maximal separation between Gaussian clusters, 5 Gaussians were chosen as cluster basis. The BIC score, while decreasing, never hit a meaningful minimum, and interpretability and insight is more interesting in this case than exact clustering, which is something that this type of dataset does not allow for in any case.

By fitting a new GMM with 5 clusters the pairplot below can be generated. Which shows that there are clear correlations between clusters, but that there are no clearly defined boundaries in all dimensions.

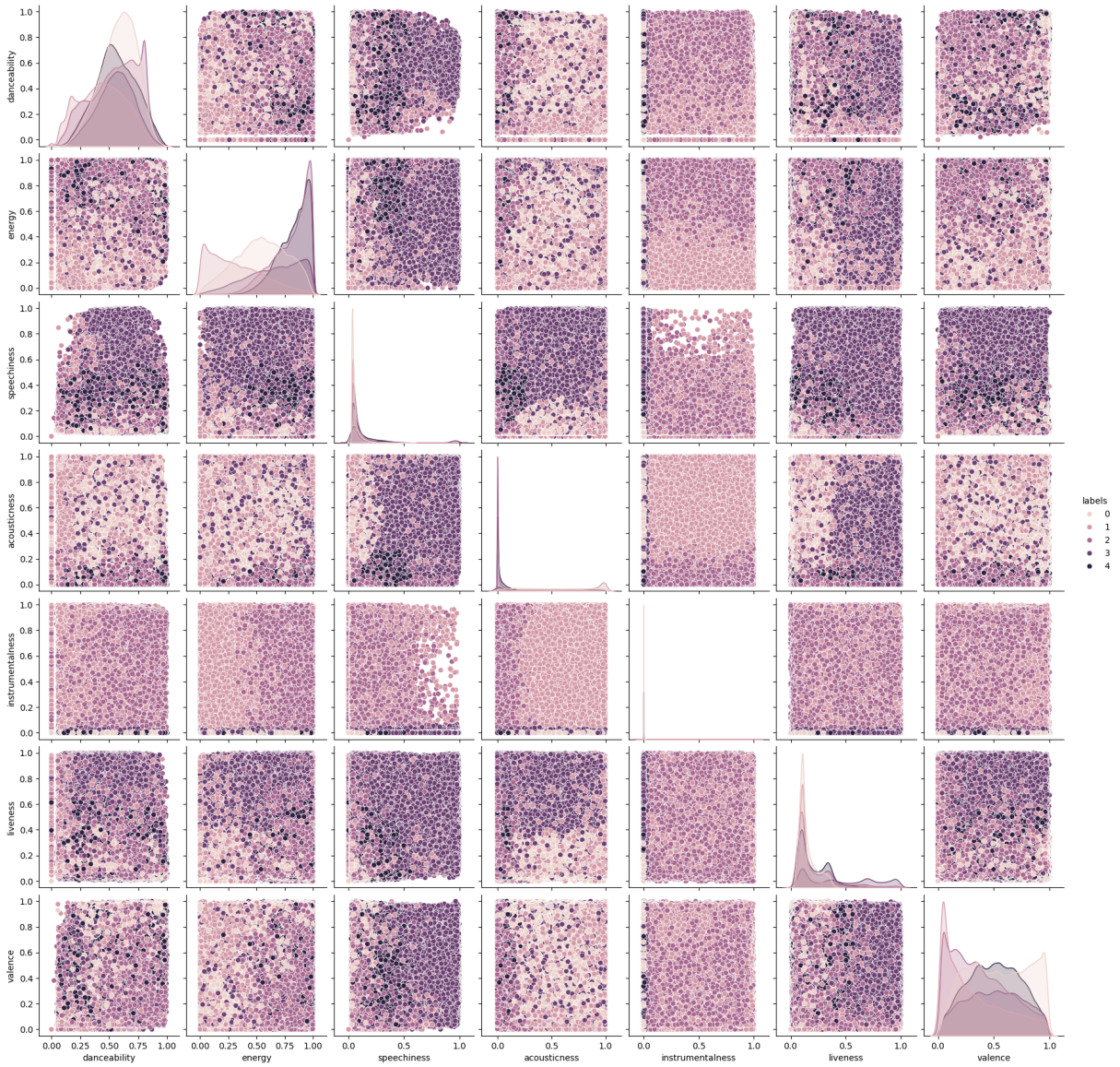

Figure 7: Pairplot of the audio features dataset, colored by cluster index.

5.1 Interpretation

We can plot the cluster means from each Gaussian to interpret the meaning by inspecting which audio features are important for each cluster.

From this we can see that each cluster captures different proportions of each feature. Probably the most pronounced and easily interpretable is cluster 2, with high energy, instrumentalness and danceability and low acousticness and will likely contain lot of electronic dance music. For each cluster the defining features are:

- **Cluster 0** = Low liveness, instrumentalness and speechiness. Average on all others
- **Cluster 1** = High acousticness and instrumentalness. Low speechiness.
- **Cluster 2** = High energy, instrumentalness and danceability. Low speechiness and acousticness.
- **Cluster 3** = High energy danceability. Low instrumentalness.

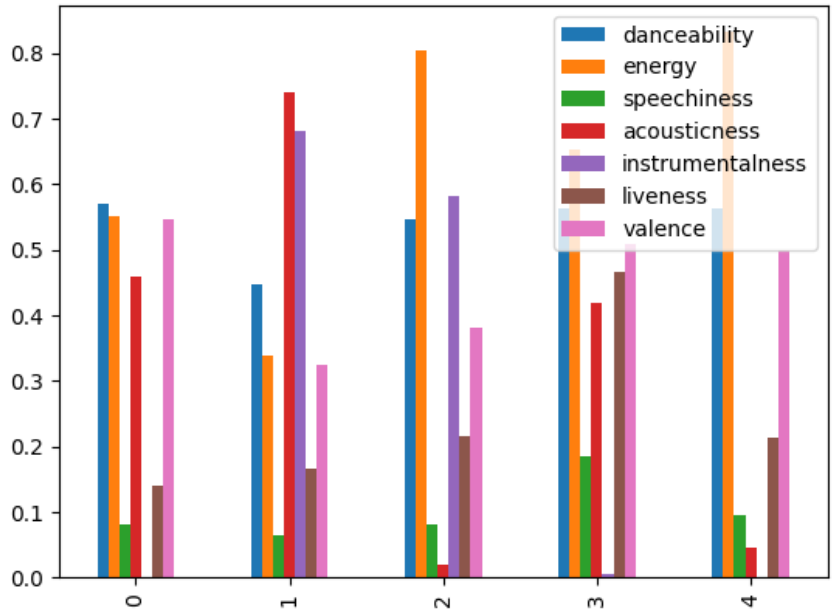

Figure 8: Cluster means for each Gaussian

- **Cluster 4** = High energy, danceability. Low speechiness, acousticness, instrumentalness.

For each of the clusters we can check the most common artist and genre within the cluster.

rank	0	1	2	3	4
i64	str	str	str	str	str
1	"Traditional (1653)"	"Traditional (2025)"	"Armin van Buuren (524)"	"Grateful Dead (734)"	"Vybz Kartel (340)"
2	"Madhu Balakrishnan (562)"	"Johann Sebastian Bach (1741)"	"Orlando Voorn (413)"	"Elvis Presley (464)"	"Jack Hartmann (277)"
3	"Giacomo Meyerbeer (516)"	"Grateful Dead (1206)"	"OnDaMiKe (396)"	"Giacomo Meyerbeer (413)"	"Glee Cast (242)"
4	"Sonu Nigam (467)"	"Andrei Krylov (826)"	"Huda Hudia (387)"	"Traditional (374)"	"Armin van Buuren (238)"
5	"Unnikrishnan (406)"	"Astor Piazzolla (805)"	"Louie Vega (382)"	"Vybz Kartel (318)"	"SICK LEGEND (220)"

Figure 9: Artists per cluster, ranked by count

rank	0	1	2	3	4
i64	str	str	str	str	str
1	"salsa (13035)"	"ambient (17749)"	"black-metal (15752)"	"comedy (16477)"	"emo (11072)"
2	"gospel (12512)"	"new-age (17478)"	"minimal-techno (13634)"	"sertanejo (6625)"	"hardcore (9550)"
3	"forro (11965)"	"sleep (14004)"	"deep-house (12812)"	"samba (5436)"	"alt-rock (9450)"
4	"acoustic (11072)"	"guitar (13977)"	"death-metal (11917)"	"forro (5141)"	"hip-hop (8633)"
5	"cantopop (10935)"	"classical (13811)"	"drum-and-bass (11849)"	"gospel (4771)"	"dancehall (8577)"

Figure 10: Genre per cluster, ranked by count

With this we can interpret the clusters more informatively as:

- **Cluster 0**: Rhythmic & Upbeat Popular. Includes salsa, gospel, Brazilian and other popular world music genres.
- **Cluster 1**: Acoustic and Ambient Instrumental. Classical music, instrumental guitar and new-age.
- **Cluster 2**: High-Energy Electronic and Metal. EDM, death-metal and other high energy tracks with few vocals.

- **Cluster 3**: Expressive Vocal & Performance. Comedy, Samba,
- **Cluster 4**: High-Energy Alternative & Hip-Hop. Alternative rock type music and hip-hop with high energy.

Interestingly, cluster 0 and 3 seem to capture much of the same genres. The main distinction here seems to lie in speechiness and liveness. This is probably because much of this music is meant to be enjoyed live and likely distributed as live albums. This also makes it more likely to contain spoken sections since the artist may be adressing the crowd.

6 Model Validation with HDBSCAN

DBSCAN is a density based clustering method that finds sections of high density and clusters them together. Since this dataset is quite high density and covers the feature space widely, a density based method is useful here since it is sensitive to local variations in the number of points close together. By using HDBSCAN (Hierarchical DBSCAN), it is possible to eliminate a section of hyperparameter selection by having the model find a good value for ϵ (the neighborhood parameter). An exploratory HDBSCAN clustering was performed with min_cluster_size=5000 and min_samples=20. This resulted in only 3 clusters being found with the following distribution:

- **-1**: 577914
- **0**: 50492
- **1**: 481789
- **2**: 49569

Where "-1" means that the point could not reliably be clustered and was designated as noise.

The AMI score was calculated to find the overlap between the HDBSCAN and GMM results, which came out to 0.535. This shows good agreement in how the distribution of these clusters is between methods. A second HDBSCAN fit was performed with smaller cluster sizes to probe a slightly finer structure and try to recover the 5 clusters from the original data. This was mostly done because approximately half the dataset was dropped as noise.

Because this seemed to miss some of the subtleties from the dataset, another attempt was done by tweaking the minimum cluster size to 2000. By doing this we were able to recover 4 clusters with good correspondence to the results from the Gaussian Model with an AMI score of 0.56. The mean values of each of the audio features of the clusters found by HDBSCAN are shown below.

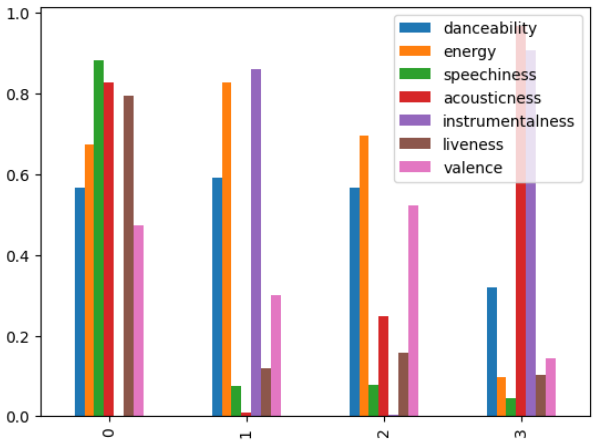

Figure 11: Audio Feature means of the clusters found by HDBSCAN.

Where we can again get the most represented artists and genres for each cluster.

rank	0	1	2	3
i64	str	str	str	str
1	"Jim Norton (65)"	"Boris Brejcha (243)"	"Traditional (1273)"	"Steven Halpern (582)"
2	"Jim Gaffigan (52)"	"Orlando Voorn (190)"	"Glee Cast (526)"	"Traditional (523)"
3	"Alonzo Bodden (42)"	"Frankyeffe (172)"	"Vybz Kartel (475)"	"Johann Sebastian Bach (501)"
4	"Kathleen Madigan (36)"	"DJ 3000 (145)"	"Giacomo Meyerbeer (448)"	"Frédéric Chopin (452)"
5	"Glenn Wool (32)"	"Niereich (135)"	"Sonu Nigam (435)"	"Ludwig van Beethoven (418)"

Figure 12: Most represented artist per HDBSCAN cluster.

rank	0	1	2	3
i64	str	str	str	str
1	"comedy (3126)"	"minimal-techno (7054)"	"k-pop (14144)"	"new-age (9152)"
2	"show-tunes (23)"	"black-metal (4021)"	"alt-rock (13676)"	"ambient (7368)"
3	"sertanejo (23)"	"deep-house (3847)"	"country (13212)"	"classical (6728)"
4	"samba (23)"	"drum-and-bass (2997)"	"emo (13059)"	"sleep (5582)"
5	"singer-songwriter (12)"	"grindcore (2830)"	"spanish (12035)"	"piano (5354)"

Figure 13: Most represented artist per HDBSCAN cluster.

From this we can see that the combination of features largely correlates with the clusters found earlier with GMM. The following clusters can be identified:

- **0** - Expressive Vocal & Performance. Comedy, Samba. (Corresponding to cluster 3 from the GMM model).
- **1** - High-Energy Electronic and Metal. EDM, death-metal and other high energy tracks with few vocals. (Corresponding to Cluster 2 from the GMM model)
- **2** - Popular music, easy to listen to. (Has fewer direct correspondence with GMM, but contains features from both GMM cluster 0 and cluster 4)
- **3** - Acoustic and Ambient Instrumental. Classical music, instrumental guitar and new-age. (Corresponding to cluster 1 from the GMM model.)

There is good overall agreement with the two methods, as also indicated by the AMI score of 0.56. One thing to note is that HDBS cluster 0 is very small. So the distinction here is not as clear as desired. In the GMM clusters this was also the smallest cluster (byt a much smaller margin) but this might be instructive for further research. It would be interesting to see which tracks were not able to be classified by the HDBS method (approximately half the dataset was not assigned to a cluster) and to see if performance increases with more detailed feature selection or hyperparameter tuning.

7 Summary and Reflection

Since this analysis was only applied to a small subset of tracks available on Spotify, there are large groups of music types and representation missing. However, the form of the analysis is still insightful. By using PCA to remove two of the features we can significantly reduce the dimensionality of the audio features space while keeping approximately 93% of the variance. This suggests that there is some redundancy in the way these are defined and that audio features are correlated. This also follows from a correlation matrix which indicates that danceability/valence and acousticness/energy are correlated.

Next we clustered the data using Gaussian Mixtures and found that there is no natural clustering of this dataset. However, using 5 clusters (chosen to keep interpretability as high as possible), we see

that the data splits up into meaningfully distinct archetypes which also enlighten why some of the audio features had moderate to high correlation.

Finally a separate analysis was done with HDBSCAN to validate the clusters found with GMM. This analysis largely agreed with the GMM results and showed that a few archetypes clearly emerge. Though much of the data was not able to be classified by the HDBS method and more research would be required to make a definitive comparison.

7.1 Next Steps

The result that audio features collapse into clear archetypes of music is clear. However, how this distinction is best made is not fully clear yet. Different clustering methods give slightly different results and would require more investigation to determine if a definitive clustering method is possible based on Spotify Audio Features. For this I would recommend finding hyperparameters that would allow HDBSCAN to cluster more of the available data (together with investigating what the reason could be that a large subset of the data could not be classified).

Another option is to add more of the available features that are not Spotify Audio Features. For example, I suspect that adding the tempo of the track would already split up some of the clusters. The same goes for the time signature since western pop music often has a 4/4 time signature and music from other cultures tends to have more variability there.