Genre Visualization using the Free Music Archive

Erik Duus
MATH637 2021

Abstract

The FMA Dataset [1] is used to investigate the ability of audio features to discern between different genres. Multiple dimension reduction techniques are applied to visualize the feature space in 2 dimensions and determine if the genre structure is readily visible. Clustering approaches are applied to the high-dimensional feature space to see if they can extract genre structure. Lastly, the genre relationships embedded in the track labelings are visualized to explore the universe of hybrid genres.

FMA Dataset

The FMA Dataset comprises over 100,000 freely downloadable audio tracks and extensive per track metadata. It is enriched with summary statistics of a variety of audio spectral features computed by the Librosa package [2].

FMA also incorporates a genre hierarchy comprising 161 genres and 18 root genres. A track may be tagged with more than 1 genre, and the genres can belong to different root genres. The tracks are organized into 4 collections:

- full: all tracks as untrimmed audio files
- large: all tracks as 30-second audio clips
- medium: 25,000 tracks as 30-second audio clips
- small: 8,000 tracks as 30-second audio clips

The medium subset only includes tracks with all genres belonging to a single root genre and is unbalanced across those root genres. The small subset is similar but is restricted to 8 root genres and is balanced.

The track metadata is extensive, containing information about the artist, the album, artist biographies, and so forth. The track collections are available as compressed archives, and the track and genre data are distributed as CSV files.

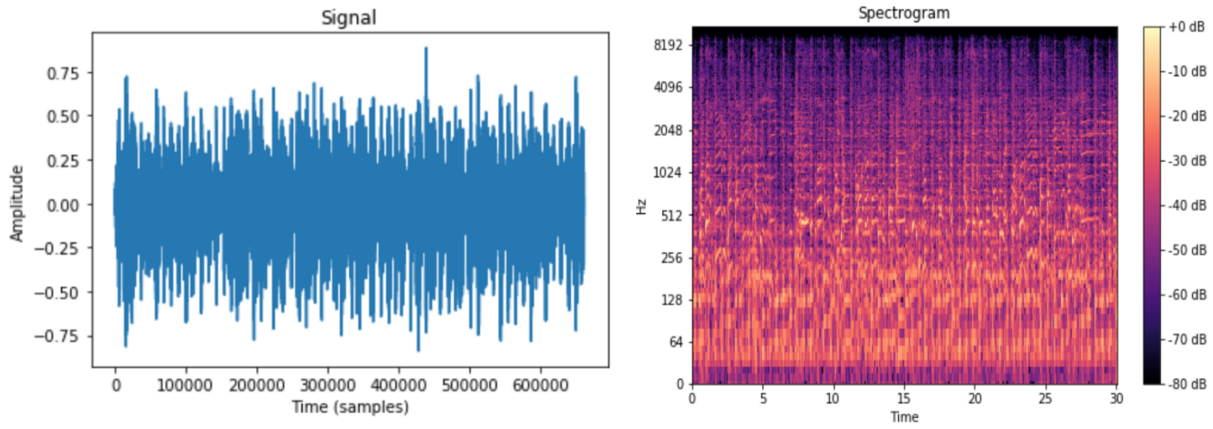
Audio Features

The dataset is enriched with a set of summary statistics for audio spectral features extracted using Librosa [2]:

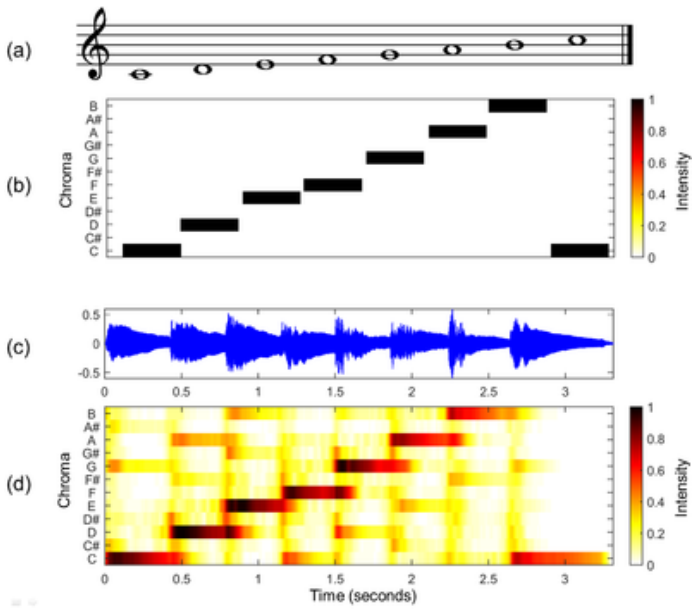
`chroma_stft`	Compute a chromagram from a waveform or power spectrogram.	12 bins
`chroma_cqt`	Constant-Q chromagram	12 bins
`chroma_cens`	Computes the chroma variant “Chroma Energy Normalized” (CENS)	12 bins

`mfcc`	Mel-frequency cepstral coefficients (MFCCs)	20 bins
`rms`	Compute root-mean-square (RMS) value for each frame	
`spectral_centroid`	Compute the spectral centroid.	
`spectral_bandwidth`	Compute p'th-order spectral bandwidth.	
`spectral_contrast`	Compute spectral contrast	7 bins
`spectral_rolloff`	Compute roll-off frequency.	
`tonnetz`	Computes the tonal centroid features (tonnetz)	6 bins
`zero_crossing_rate`	Compute the zero-crossing rate of an audio time series.	

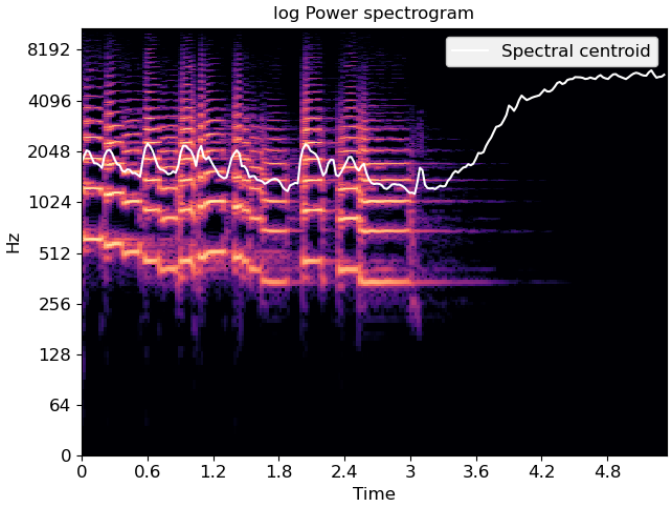
Essentially each of these audio features post-processes an audio track spectogram, which is a 3-dimensional representation of an audio signal, reflecting the intensity of frequencies over time:



For example, a chromagram relates the spectogram to the 12 pitch classes [3]:

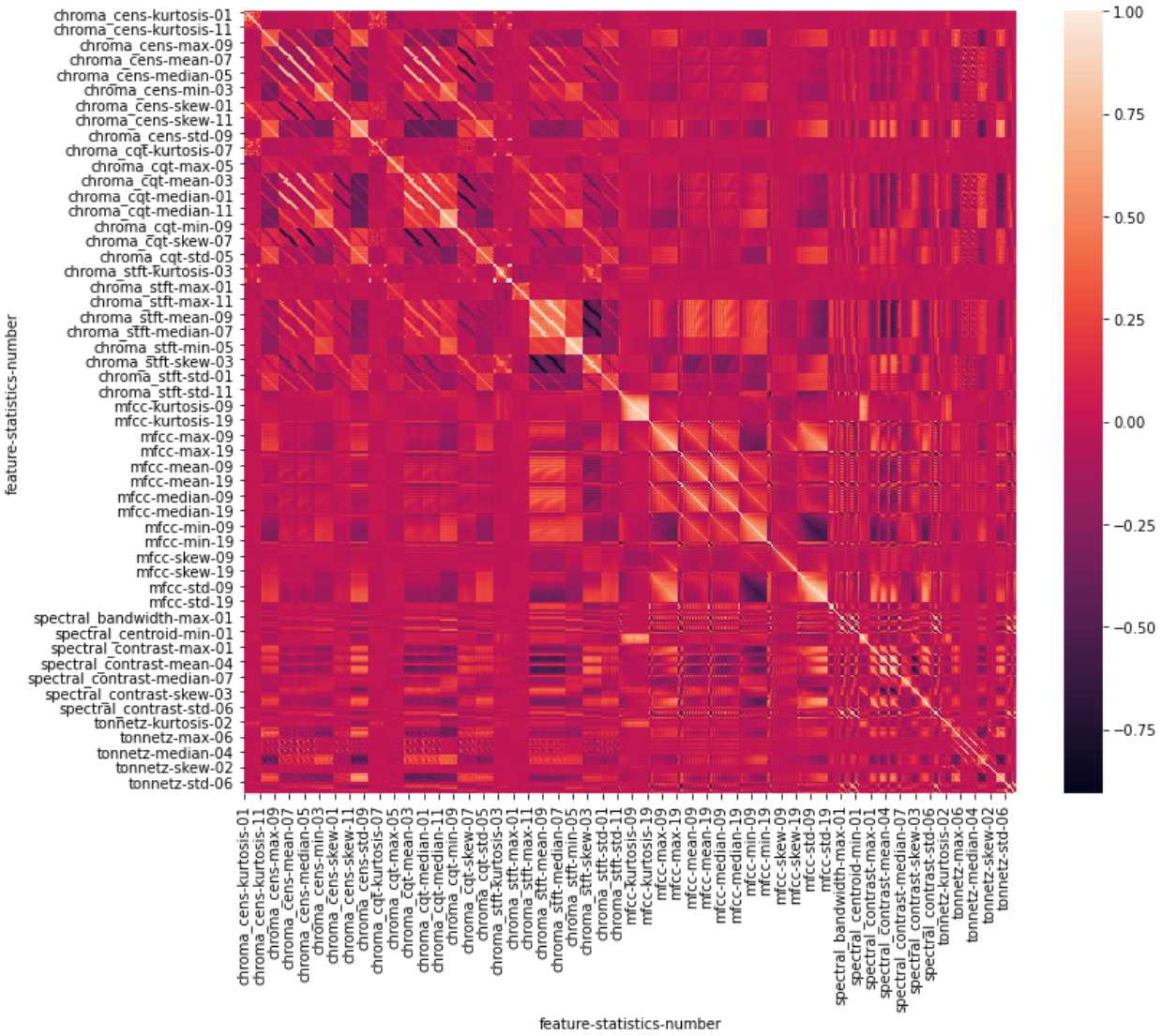


As another example, a spectral centroid is a measure of the location of the center of mass of the spectrum over time [2:



FMA computes a set of 7 summary statistics for each of the audio features over the available dimensions of each feature; mean, median, standard deviation, min, max, skew, and kurtosis. For the 2 examples above, chroma_cqt would result in 12 x 7 = 84 features, while spectral_centroid would result in 1 x 7 = 7 features. Across all features, bins, and statistics this results in 518 total spectral features for each track. This dataset is also available for download as a CSV.

While this is an extensive set of features, it is important to note that any temporal structure has been discarded through summarization. Furthermore, given that each of these features post-processes the audio spectogram, we might expect some features to be correlated. A heatmap indicates that there are some highly correlated features (feature names are sampled for brevity):



Looking at the count of correlations between features where abs(r) > .90, we see that the mean and median are often correlated within the same feature. We also that chroma_cens and chroma_cqt are highly correlated.

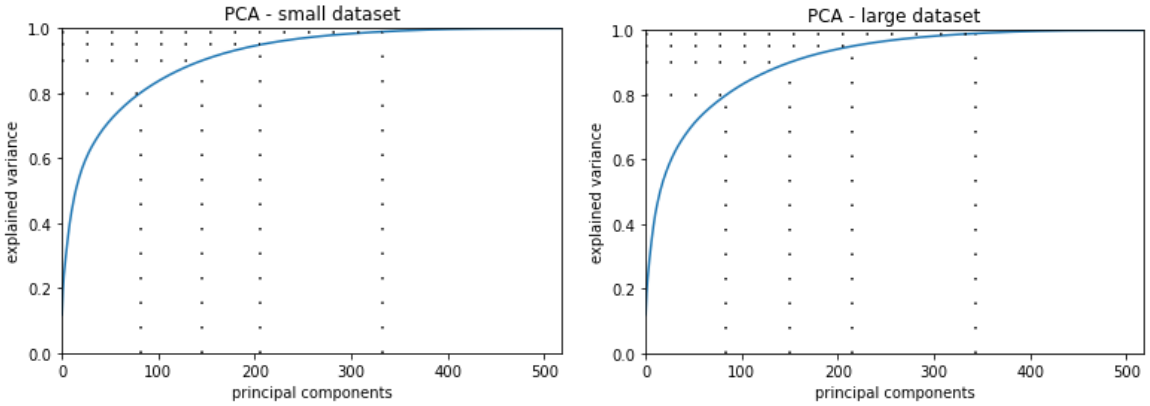
mfcc_mean-mfcc_median	20
chroma_cens_mean-chroma_cens_median	12
chroma_cqt_mean-chroma_cqt_median	12
chroma_stft_mean-chroma_stft_median	12
chroma_stft_kurtosis-chroma_stft_kurtosis	10
chroma_cens_mean-chroma_cqt_mean	9
chroma_cens_median-chroma_cqt_median	9
spectral_contrast_mean-spectral_contrast_median	7

chroma_cens_mean-chroma_cqt_median	6
tonnetz_mean-tonnetz_median	6

These features would be candidates for removal as part of a feature selection process.

PCA Analysis of Audio Features

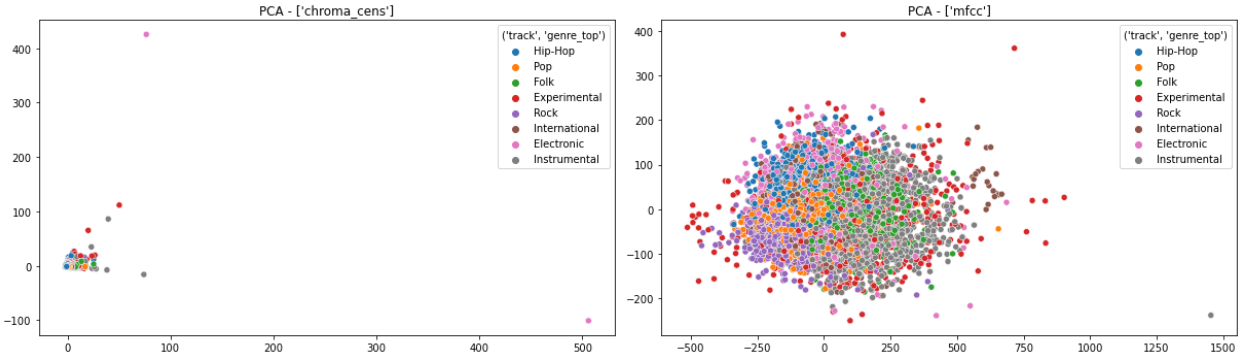
PCA from scikit-learn is applied to the entire set of audio features for both the small and large subsets. Examining the cumulative explained variance ratio reveals that close to 100 principal components are required to explain 80% of the variance of the datasets:



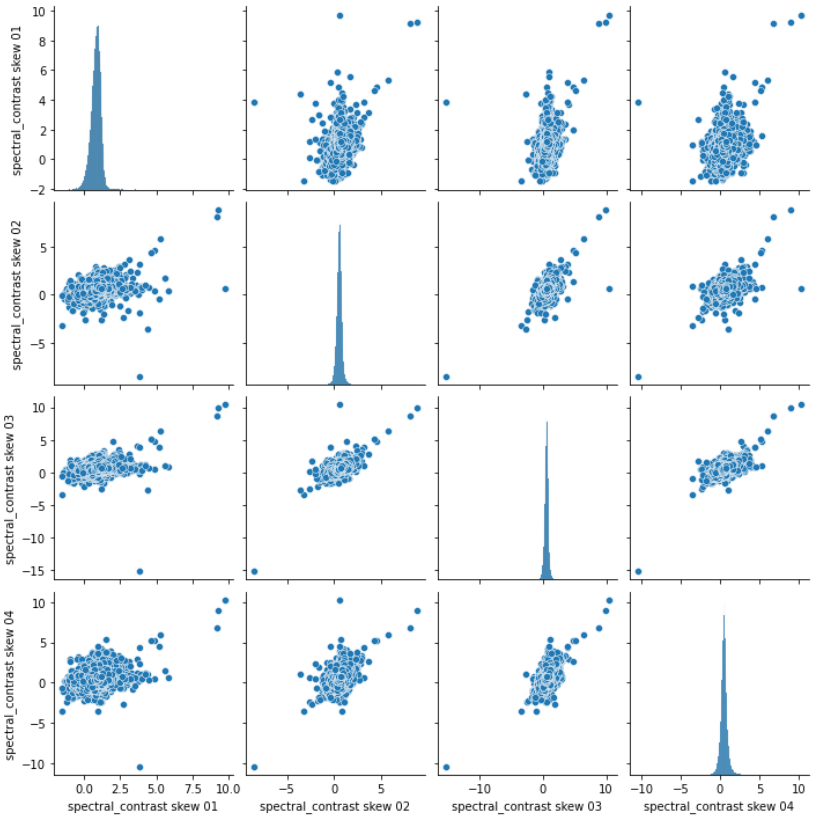
While 99% of the variance is captured by a bit more than 50% of the components, in order to retain most of the variance for subsequent analysis, between 150 to 200 components are required. This would produce some dimensionality reduction but still result in a fairly high dimensional space. It also indicates that projecting down to 2 or 3 dimensions for visualization may be challenging since a handful of components will only capture about 25% of the variance in the feature space.

Visualizing Genre Structure - PCA

Applying a 2-component PCA to each set of audio features using the small dataset with no feature scaling produces mixed results. Some features have large outlier values that dominate, causing the rest of the points to cluster in a small ball, while others produce plots with some degree of genre clusters, showing that the audio feature has some discriminatory power:

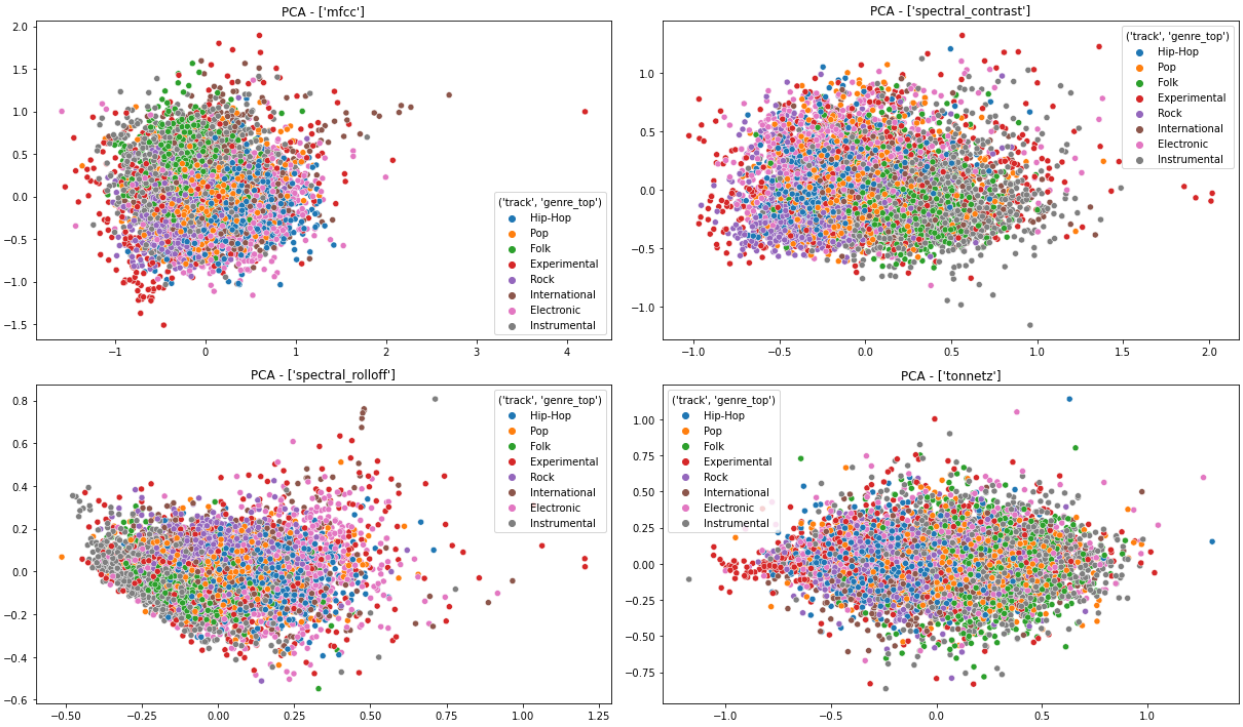


Examination of feature distributions reveals that they are not universally normal, as these examples show:



Consequently, min-max scaling may be the appropriate choice.

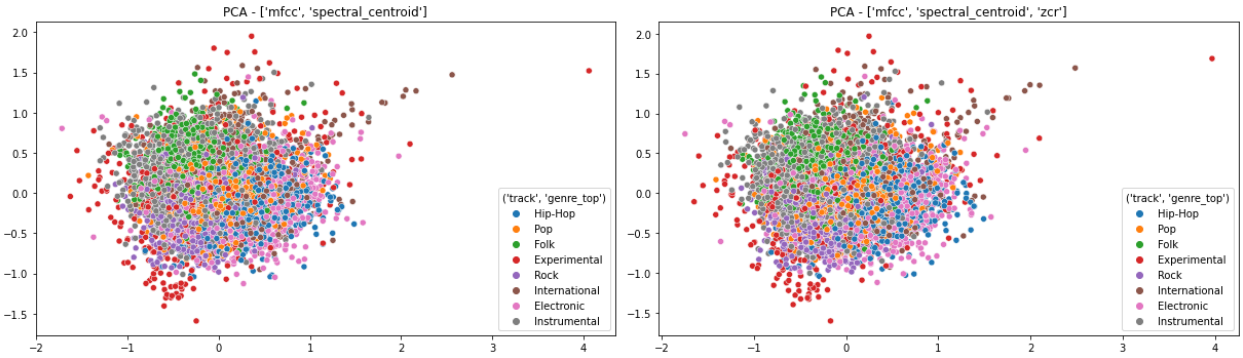
Repeating the PCA decomposition on scaled features produces slightly better results. Visual inspection indicates that features like mfcc and spectral_centroid show slightly more separation, while others such as zcr and rmse show slightly less:



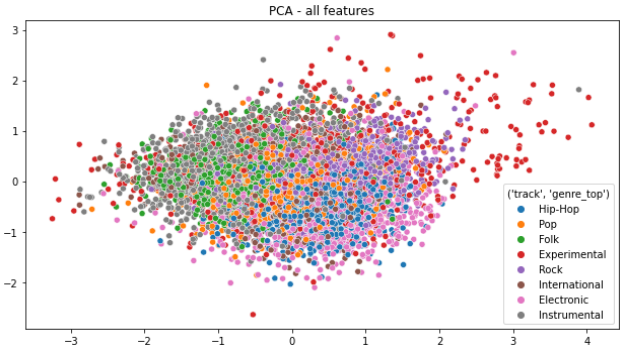
Looking at the mfcc feature, Folk appears better-clustered than Experimental. This could be due to a variety of reasons:

- The set of audio features does not capture enough information to discriminate all genres
- The genre structure exists in high dimensions, but PCA is not powerful enough to capture it and project it down to 2 dimensions.
- Some genres are ‘closer’ in nature to other genres and therefore more difficult to discern.
- The selected tracks in the small dataset are not accurate representations of their genres
- The genre tagging of the dataset is not accurate

Adding features in combination does not seem to produce improvement, as these representative examples show:



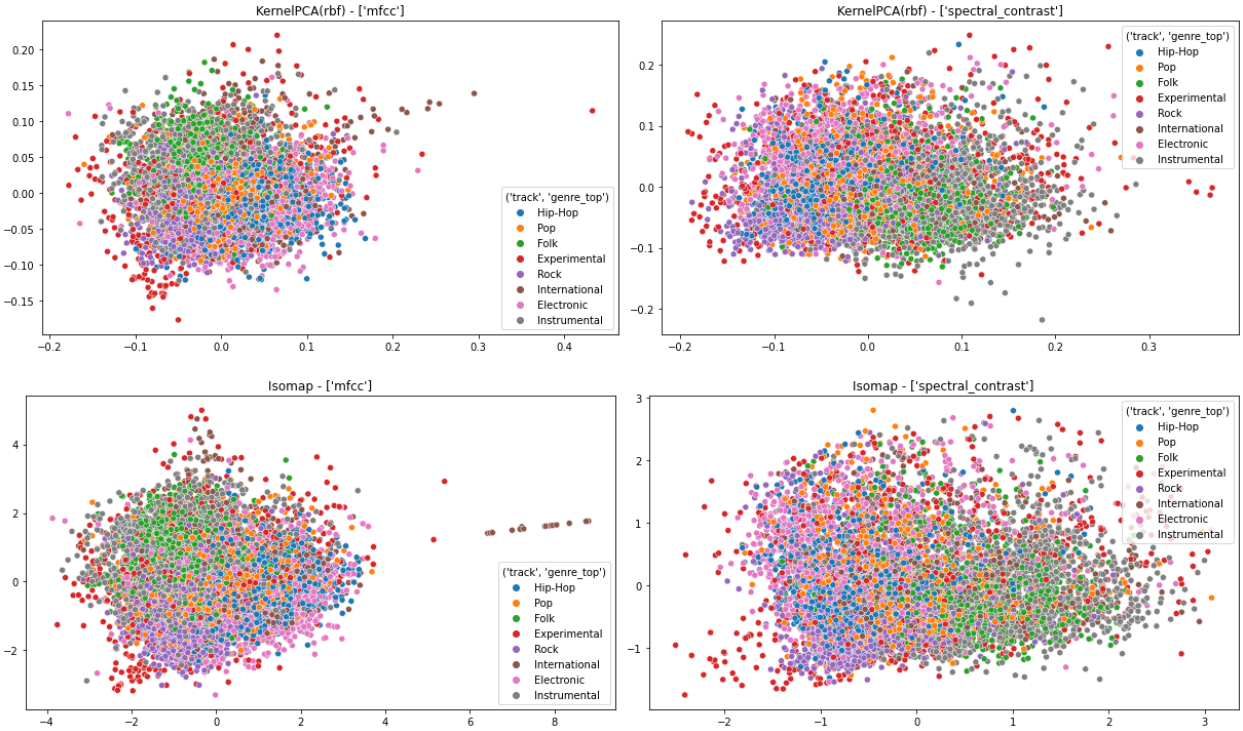
PCA on all features also does not produce a better result. This makes sense since all the features are derived from the same spectogram, so in some sense, they are related to each other. And any modest increase in discriminatory power is complicated by the increase in the dimensionality of the feature space: going from MFCC to all features increases dimensionality from 140 to 518.

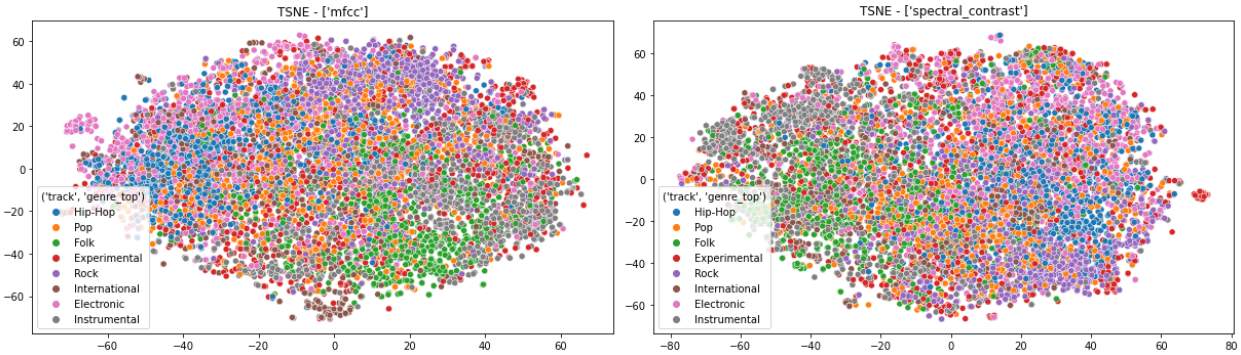


Any genre structure that exists might be nonlinear, or the audio feature set does not have enough power to discriminate genres at lower dimensions.

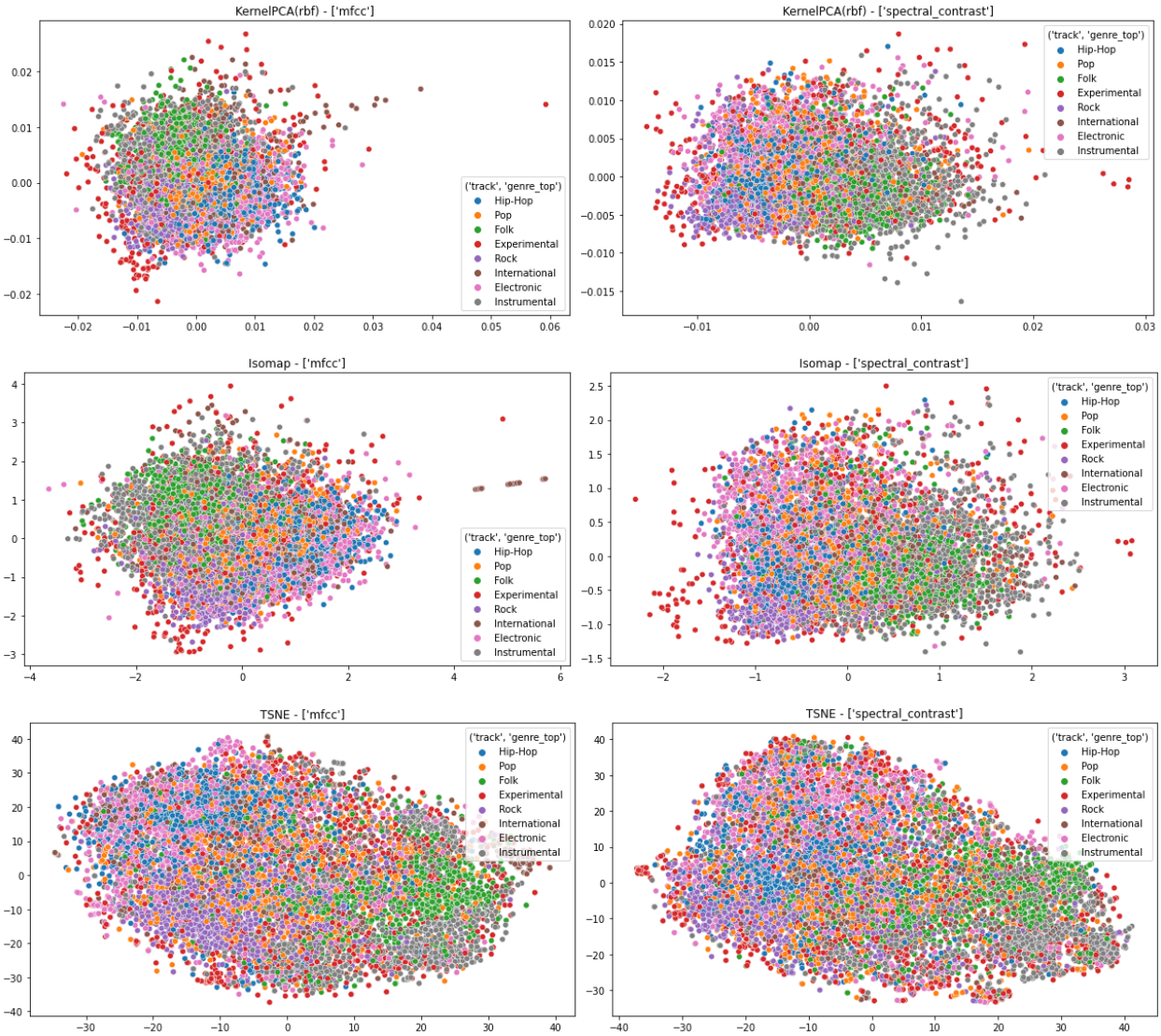
Non-linear techniques - Kernel PCA, Isomap, TSNE

To determine if non-linear techniques show more discriminatory power, Kernel PCA (using radial basis function), Isomap, and t-SNE are applied using default parameters against the audio features and fare little better, as some representative examples show:

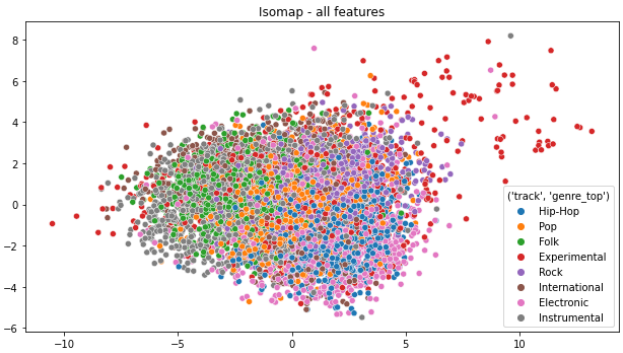




TSNE and Isomap perhaps show slightly better clustering than PCA, but really don't discern more structure. The Folk genre still shows some visible clustering in contrast to Experimental. Parameter tuning perhaps shows very slight improvements:



As before with PCA, applying these techniques to the full feature set does not produce improvement, as shown by Isomap for example:

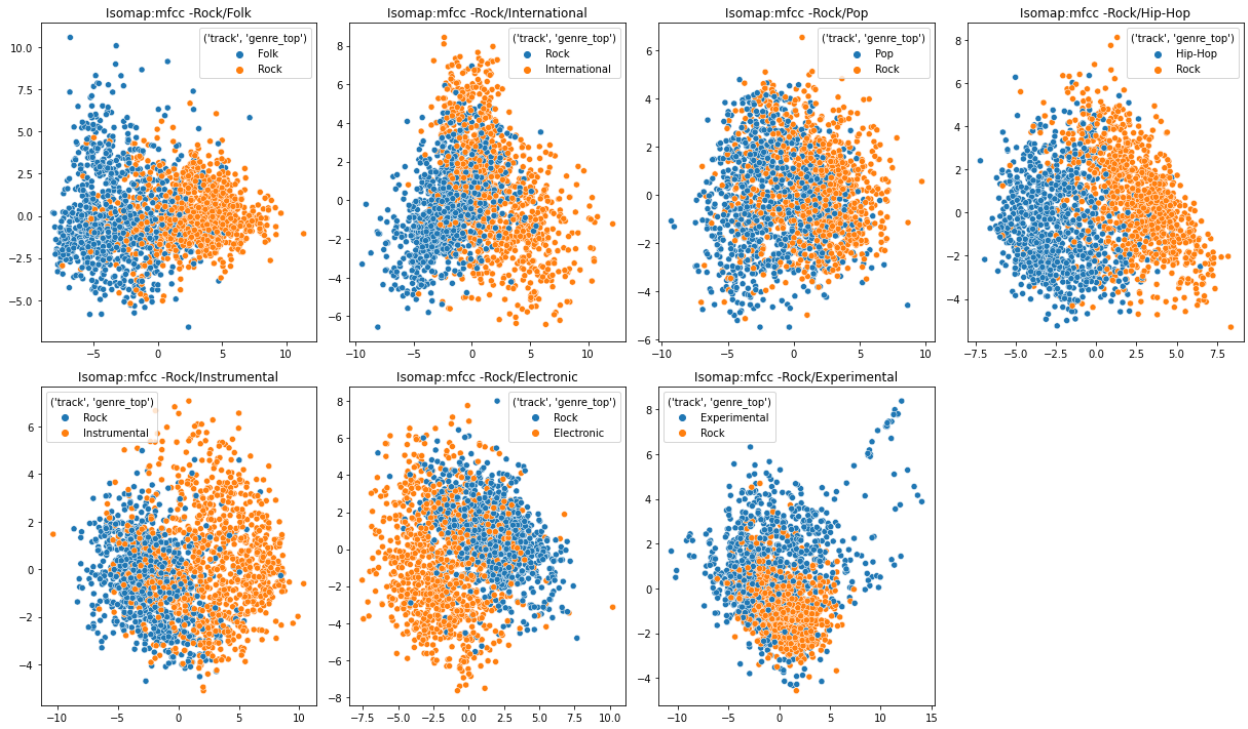


As with PCA, these techniques do not extract a clear genre structure in 2 dimensions. As mentioned earlier another possible reason is that some genres are ‘closer’ together than others. Visualization using dimension reduction could provide some insight into whether this is true.

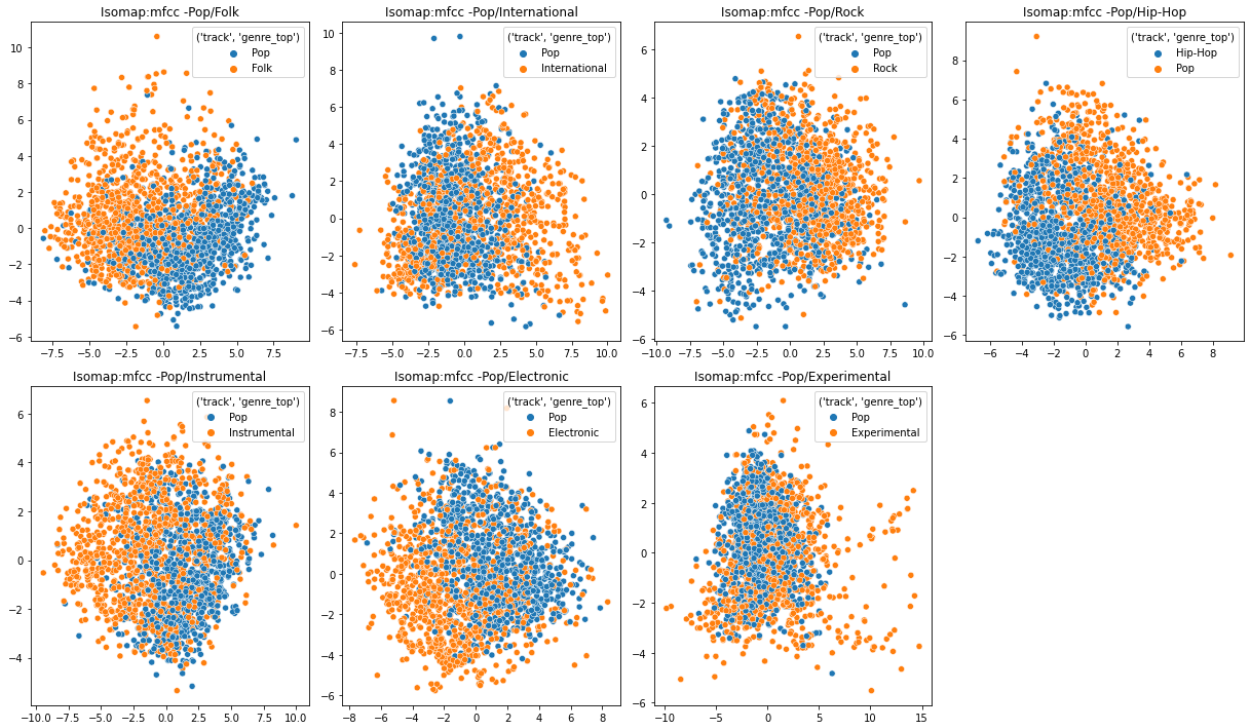
Isomap on Genre Pairs

Applying Isomap on the mfcc feature set and genre-pair subsets of the small dataset produces some interesting results. Clearly, some genres are more easily discriminated in low dimensions than others. Additionally, some particular pairs are also difficult to discriminate.

For example, Rock discriminates rather well versus other genres besides Pop, and to a lesser degree, International:



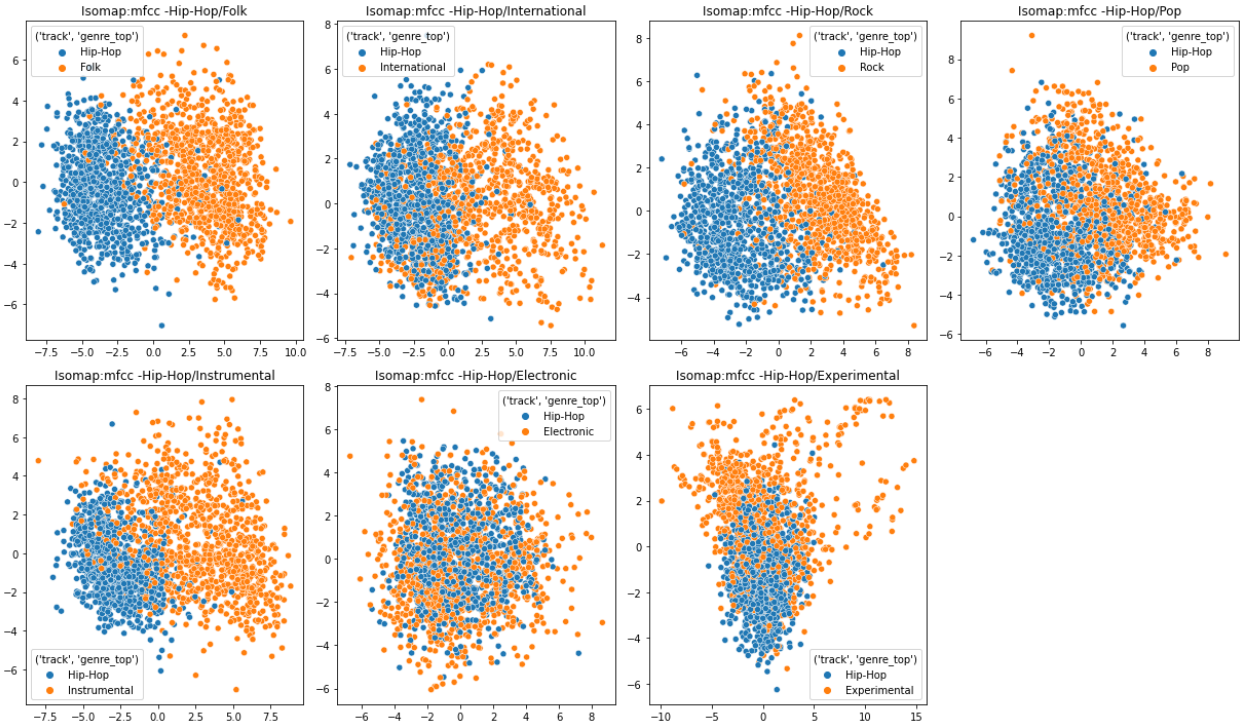
Pop, on the other hand, struggles against all genres besides Hip-Hop and Folk:



Considering the FMA genre hierarchy, intuitively this seems reasonable. Pop refers to ‘popular music’ and therefore is perhaps a more ambiguous genre. International-Pop, Pop-Rock,

Experimental-Pop, Electronic-Pop all seem like viable hybrid genres. The question is whether these tracks are incorrectly tagged with genres (recalling that this dataset is comprised of tracks with a single root genre), or if these genres are very close together and require extra features for discrimination.

Some specific genre pairs are also difficult to distinguish (understandably so in retrospect). Hip-Hop, for example, discriminates quite well against most genres except for Electronic:

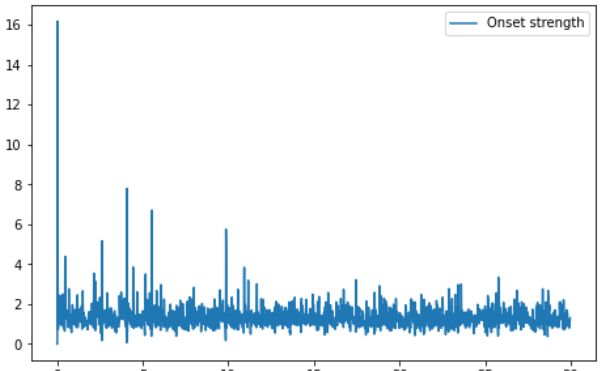


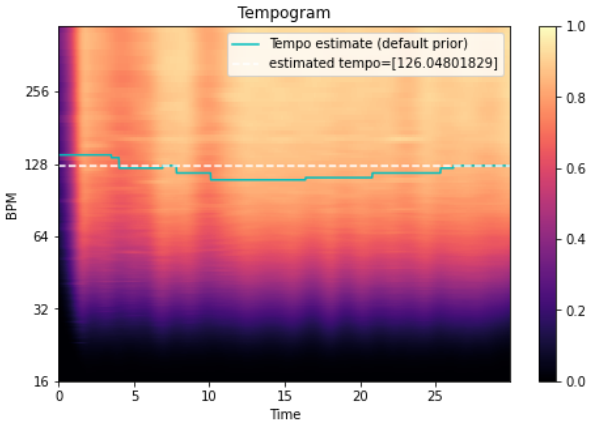
Intuitively, this result seems reasonable and perhaps points to another strategy for building a genre classifier. In addition to features/techniques that discriminate well generally, a classifier may require additional features aimed at discriminating purely between 2 hard-to-distinguish genres. In this case, perhaps vocal features could be extracted and identified as ‘speaking’ vs. ‘singing.’

Tempo Features

While the FMA dataset contains an extensive set of audio features, they are all spectral features. Given that the current feature set does not discriminate well between all genres, it is natural to investigate if other features could be utilized.

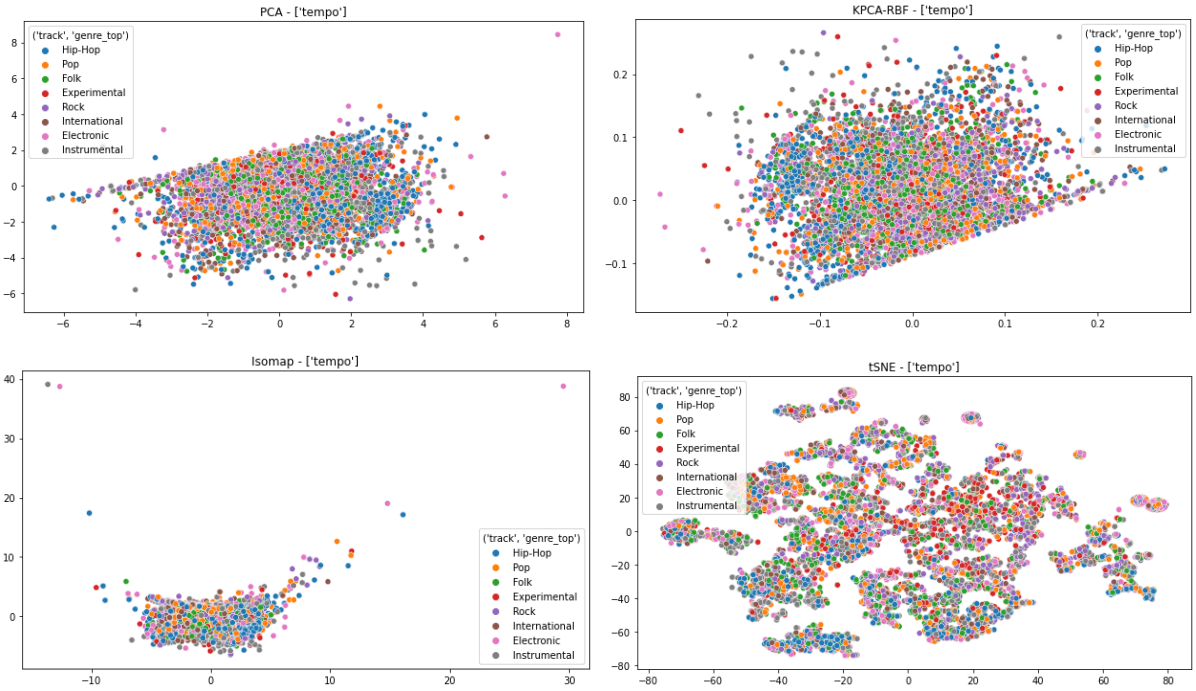
Librosa can also extract beat/tempo features, producing a tempogram (vs. a spectogram). Onset event detection (this can be energy-based or spectral-based) is used to capture beats, from which a tempogram can be derived. The tempogram shows the intensity of tempo over time:



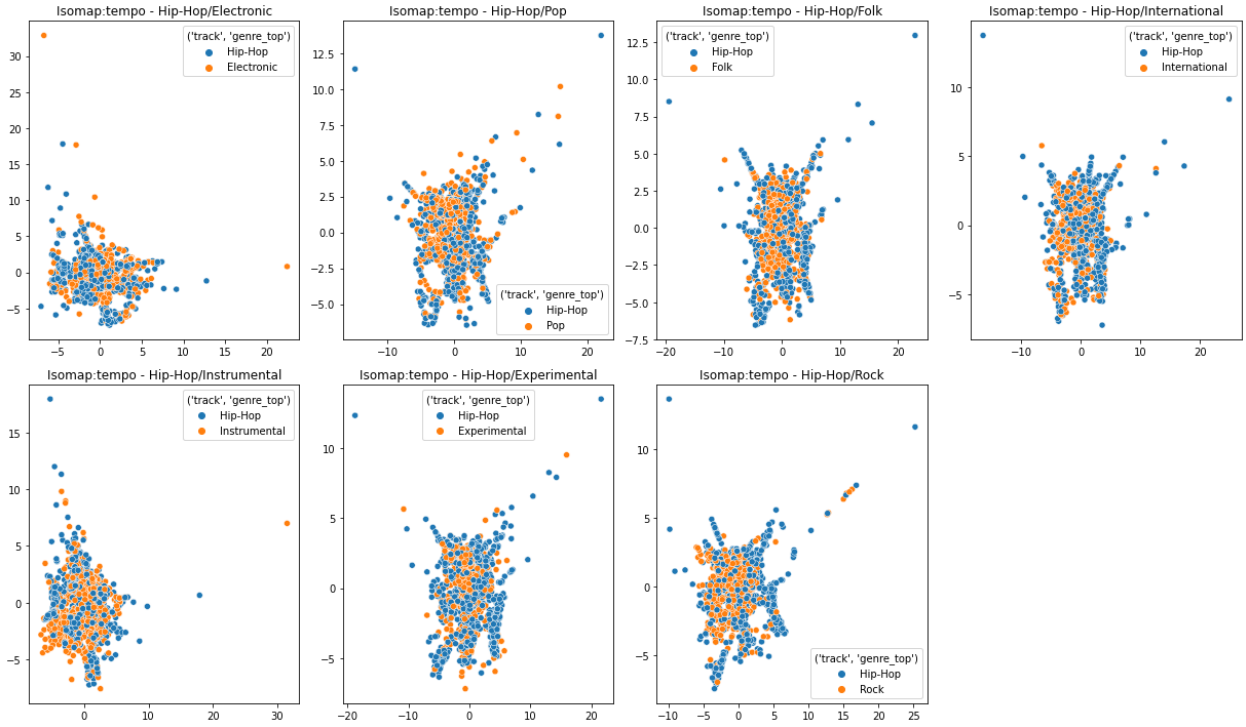


Since FMA works with summary features, Librosa is used to estimate the dynamic global tempo for each track in the small dataset, and then the usual 7 summary statistics are computed. The small dataset is enriched with these new features, and both PCA and non-linear dimensionality reduction performed.

The results are not encouraging. Very little genre structure is discernible at lower dimensions using any of the techniques:



Isomap on genre pairs fares no better, as shown by Hip-Hop:

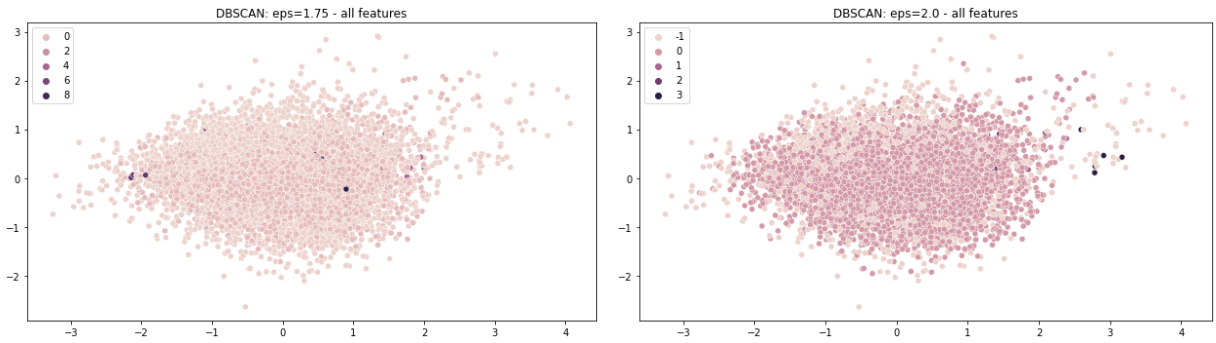


Summary statistics of the dynamic tempo are probably not useful features for classification. The temporal structure of the audio may still have the potential to discriminate between genres, but the use of summary statistics discards that structure.

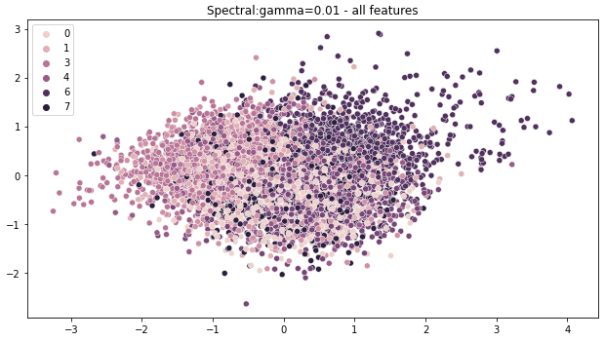
Clustering

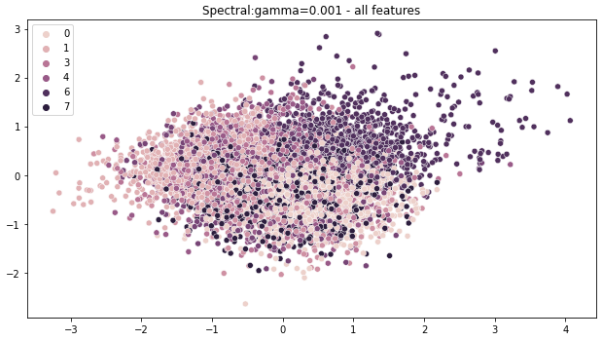
Instead of projecting the audio features to lower dimensions to discern genre structure, clustering techniques can be applied directly to the high dimensional feature space. DBSCAN and Spectral Clustering from scikit-learn are used, and the embedded clustering visualized on a 2-dimensional PCA projection.

DBSCAN shows some clusters after some experimentation with the eps parameter:



Spectral Clustering also produces some clusters after some experimentation with the gamma parameter:





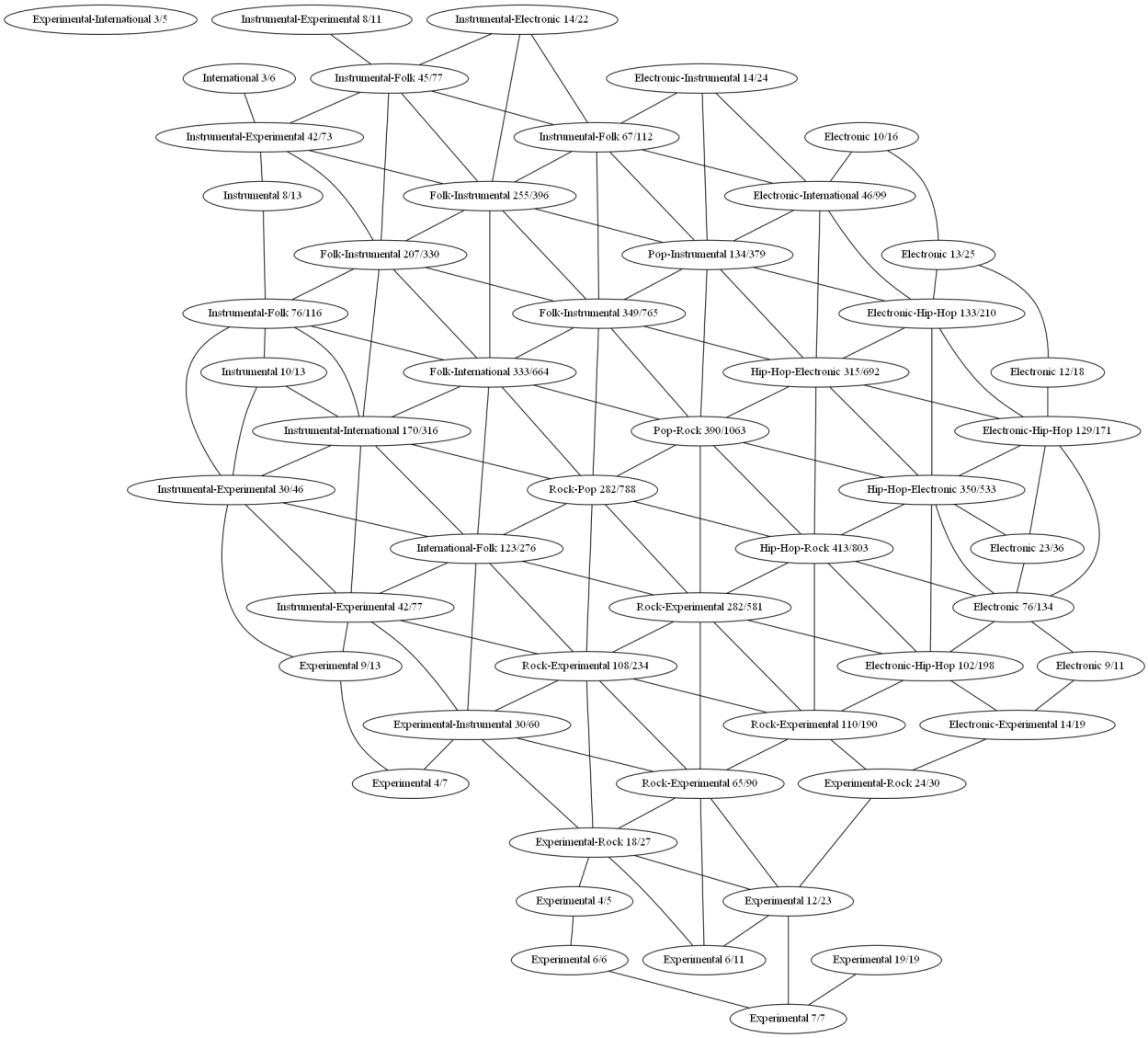
However, examination of the cluster-genre assignments reveals that the clusters don't correlate well with genres. For example, spectral clustering at gamma=0.01 produces the following clusters:

	cluster_1	cluster_5	cluster_6	cluster_4	cluster_3	cluster_2	cluster_7	cluster_0
Hip-Hop	62	434	74	37	16	90	124	163
Folk	101	28	38	154	239	160	111	169
Experimental	103	201	241	61	140	85	91	78
Pop	71	196	126	74	131	113	141	148
Rock	63	75	320	95	29	73	165	180
International	53	239	82	89	297	73	70	97
Electronic	95	375	59	61	36	85	120	169
Instrumental	183	32	75	152	201	139	118	100

The preceding experiments have highlighted the challenges of genre discrimination, including an ambiguous and inconsistently applied genre hierarchy and underpowered audio features. In addition, attempting to cluster in high dimensions is hindered by the use of the small dataset. The 8,000 exemplars are distributed through 500+ dimensions, introducing the curse of dimensionality.

KeplerMapper

In contrast to using dimensionality reduction or clustering to visualize the assigned genre structure of the dataset, KeplerMapper [4] is used to visualize the relationships between tracks to see if the existing genre structure is apparent or if novel combinations of genres can be discerned. Several combinations of cover parameters are applied, and the resulting nodes are labeled by the 2 most prevalent genres, producing the following illustrative output:



The resulting graph is both interesting and inconclusive. There is some grouping of genres in connected nodes, and traversing the graph perhaps reveals which genres are more or less related to others. For example, at the bottom of the graph, a vertical traversal shows the Experimental cluster gradually changes to Experimental-Rock and then Rock-Pop. At the same time, according to the genre counts, most of the nodes are not definitive members of a single genre, so perhaps care should be taken in over-interpreting the graph.

Further experimentation may reveal KeplerMapper could be useful in revealing genre structure in the sense of which genre is near to or influenced by other genres. However, greater confidence in the audio feature set, the balance of the tracks, and the genre assignments is probably required in order for KeplerMapper to be successfully utilized.

Conclusion

A variety of dimension reduction and clustering techniques are applied to the audio features of the FMA dataset. While they showed some discriminatory capability, they were unable to visualize clear genre structure from the feature space. Possible reasons were discussed, including:

- Spectral features alone are insufficient.
- Summary statistics discard temporal structure.
- The techniques used are not sufficiently powerful.
- The dataset is not a fully representative sample.
- The genre hierarchy is ambiguous and inconsistently applied.

References

[1] Defferrard, Michaël, et al. "Fma: A dataset for music analysis." *arXiv preprint arXiv:1612.01840* (2016), [PDF] arxiv.org, [GitHub].

[2] Brian McFee, Vincent Lostanlen, Alexandros Metsai, Matt McVicar, Stefan Balke, Carl Thomé, … Taewoon Kim. (2020, July 22). librosa/librosa: 0.8.0 (Version 0.8.0). Zenodo, http://doi.org/10.5281/zenodo.3955228, https://librosa.org/

[3] Jiang, Nanzhu; Grosche, Peter; Konz, Verena; Müller, Meinard (2011). "Analyzing Chroma Feature Types for Automated Chord Recognition" (PDF). *Proceedings of the AES Conference on Semantic Audio*.

[4] Hendrik Jacob van Veen, Nathaniel Saul, David Eargle, & Sam W. Mangham. (2019, October 14). Kepler Mapper: A flexible Python implementation of the Mapper algorithm (Version 1.4.1). Zenodo. http://doi.org/10.5281/zenodo.4077395