Genre Classification in the Free Music Archive

Problem

According to Hypebot.com, in every 24 hours in 2018 roughly 20,000 musical tracks were uploaded to major streaming services. That amounts to approximately a million tracks every six weeks. This enormous amount of data is too much for any single human to sift through in a lifetime. Assuming three minutes a track, a million tracks would take a single individual almost six years to listen to.

In order for music to be made available to individuals it is necessary to perform some degree of automated classification. The task of genre classification (assigning similar music to groups) is one of the most fundamental of these tasks. Broad grouping of music along stylistic similarity allows increased ease of discovery of new artists for listeners, and for marketers it allows identification of potential consumers for music from new bands.

Data Set

The Free Music Archive (FMA) is a resource containing 20 second samples of more than 100,000 tracks across more than 15,000 artists. These tracks are tagged with a variety of information, including not only pre-assigned genre tags, but also geographic information, language of the track, and a number of computed sonic features. The data set is made freely available for analysis along with the associated audio files.

Data Wrangling

In order to begin our investigation of the Free Music Archive dataset, we must first get an idea of the data's organization and consistency. The FMA is divided into several flat files representing relational tables, four of which are of active interest for the task of classifying genre. An artists table, containing information relevant to the creators of each track; a genre table which consists of a hierarchical mapping of genre values; a features table containing audio features generated by the librosa package; and an echonest table, containing curated summary features (danceability, acousticness, etc) for a subset of 13,000 tracks in the archive.

An initial investigation of the amount of column-wise missingness by feature shows that many columns are missing more than half of the represented data. The majority of sparsely populated columns represent metadata features that are voluntarily populated by bands

submitting their tracks. For example, the ***associated label***, ***producer***, and ***related projects*** fields tend are missing more than 80% of values.

There are also likely biases in missing data. One instance is the ***active year end*** field, which is missing 95% of values. Since the FMA is composed of tracks submitted by artists themselves, it is likely that the data is biased toward bands that are currently active, or who were active when their tracks were submitted to the archive.

album	date_released	0.340421
	engineer	0.856485
	information	0.219800
	producer	0.830540
artist	active_year_begin	0.786899
	active_year_end	0.949566
	associated_labels	0.866093
	bio	0.332332
	latitude	0.582037
	location	0.341209
	longitude	0.582037
	members	0.560409
	related_projects	0.876593
	website	0.256329
	wikipedia_page	0.947633
track	composer	0.965564
	date_recorded	0.942209
	genre_top	0.534614
	information	0.977959
	language_code	0.859028
	lyricist	0.997082
	publisher	0.988149

Proportion of missing data in fields with more than 20% missing data. The most concerning are the genre_top and latitude/longitude data fields.

We also analyze the row-wise missingness in order to assess whether there are features where missingness is correlated. For example, if a band does not have a label they may not have a producer. In order to do this analysis we compute the Phi Coefficient, which is a measure of association between binary variables (in this case, missing or not missing).

The top correlated variables are logical. For example, if latitude is missing, longitude is also missing. Likewise, if location is missing, then latitude and longitude are also missing. If a band does not have a producer on their album, they are also likely not to have an engineer (likely an indicator that the track was self-produced). The correlations scale down relatively quickly after this, indicating that most missing fields are a relative coin flip. This should give us some faith that our features are largely independent, at least with regard to missingness.

Each track in the artists table contains three columns of high importance to the outcome of the genre classification task. The **genre_top** column contains the most highly associated genre label for the track, while the **genres** field contains a list of top level genre for the track with no indication which may be the predominant label. The **genres_all** feature contains both top level and subcategorized genre features. These labels are provided numerically and must be linked back to the genre table to get the string value of the genre identities.

These columns are especially of note because while all tracks have values for the genres and genres_all features, only 47% have a value for genre_top. In order to simplify the process of genre identity, we will focus only on tracks with a genre_top value. In future analyses it may be advisable to impute the top genre based on audio features, or to focus on tracks with

no overlap between the genres of interest being focused on. But on a first analysis, it is simpler and cleaner to focus on the remaining ~50,000 tracks with a described top genre in order to build a maximally efficient classifier.

1194	(artist, longitude)	(artist, latitude)	1.000000
1093	(artist, latitude)	(artist, longitude)	1.000000
411	(album, producer)	(album, engineer)	0.768470
160	(album, engineer)	(album, producer)	0.768470
613	(album, type)	(album, date_created)	0.725658
62	(album, date_created)	(album, type)	0.725658
1092	(artist, latitude)	(artist, location)	0.588272
1195	(artist, longitude)	(artist, location)	0.588272
1144	(artist, location)	(artist, longitude)	0.588272
1143	(artist, location)	(artist, latitude)	0.588272
843	(artist, bio)	(artist, website)	0.538926
1444	(artist, website)	(artist, bio)	0.538926
511	(album, title)	(album, date_created)	0.532504
60	(album, date_created)	(album, title)	0.532504
1450	(artist, website)	(artist, location)	0.485648

Most highly correlated missing values. As can be seen in this figure, most missing info is due to a lack of artist-provided metadata.

For this classification task, we will focus on the three most prevalent genres in the data set: Rock, Experimental, and Electronic. Tracks with one of three genres as the genres_top value make up roughly a third of the total data set. Conveniently, they should also be structurally different in tonality and instrumentation, making them a good test case for a classification task based on audio features.

For maximum overlap of the data set, we join the artist table with the librosa features table. This table is complete for all tracks, as it has been computed after addition to the archive. Each track has a matching set of 500 audio features related to measures like spectral bandwidth, spectral rolloff, and other high level audio summary features.

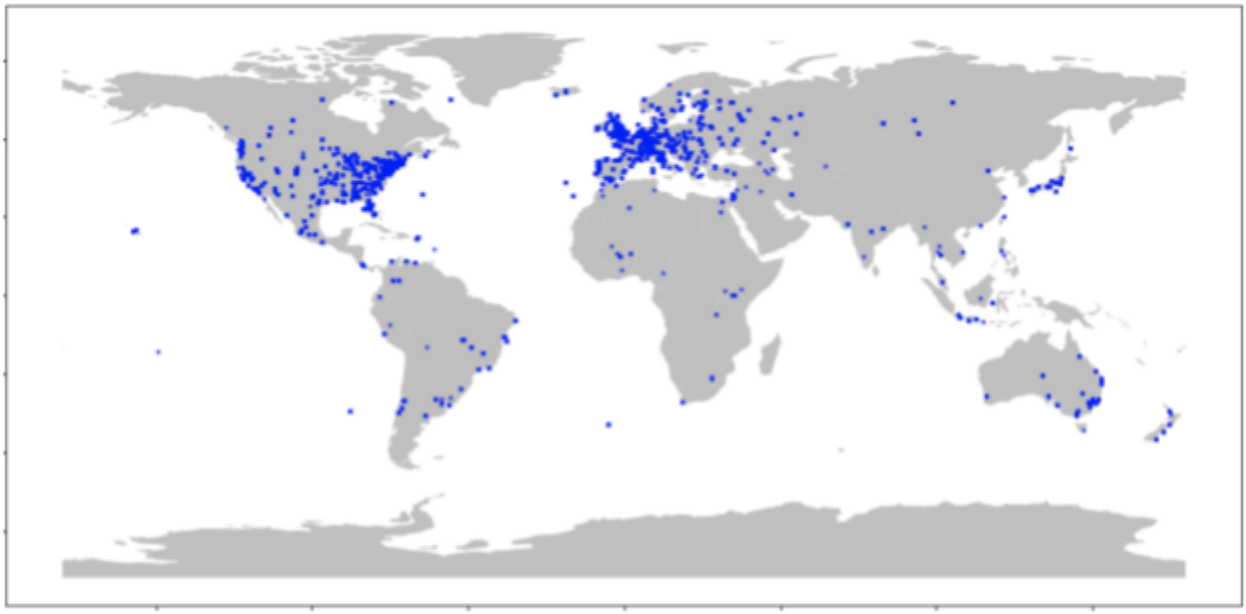
After joining of these frames we now have approximately 35,000 audio tracks labeled with a genre tag of Rock, Experimental, or Electronic, and with an associated set of audio features, ready for classification.

Data Storytelling

Now that we've decided on which genres are of interest to us, we can perform a more in-depth analysis trying to understand the organization of the data. As we have geocoordinate data for each track in the FMA, we can attempt to understand how geographic trends play a role in the way our data is constructed.

In order to perform this analysis, we make use of two major libraries: plotly for manipulation of polygons and points and geopandas for the connection of our data to mapping shapefiles that represent geographic areas.

We can first perform a broad mapping of the data across the globe. The dataset contains roughly 40,000 points which are tagged with both a genre of interest and geocoordinates.

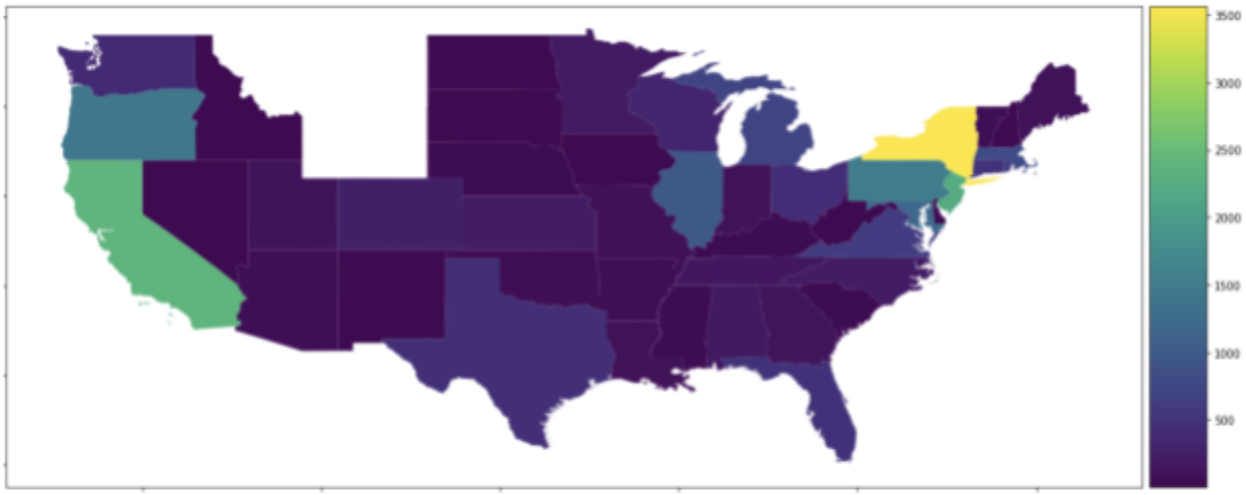


Mapping tracks with geocoordinate data globally gives a good impression of the distribution of data across different regions. Europe and the North America are especially well represented, while Africa, Asia, and South America fall behind by comparison.

Mapping these points gives us a rough idea of the distribution of our data around the world. Europe and North America are well represented, while the remaining five continents have only sparse coverage. Africa in particular seems to be underrepresented.

We can further investigate this trend by considering whether the type of music represented within these two areas is the same or different. We divide the data using a cross-table, separating continent and genre. A first glance at the data gives a first impression that North America has a greater proportion of Rock, while Europe has a larger proportion of Electronic music. However, this difference may simply be due to chance.

To assess whether the trends observed are due merely to chance, we perform a chi-squared table test of row-wise independence. The result of this test is highly significant, indicating that we are able to conclude that this difference in distribution of genres is due to something other than chance. It is important here to note that this could be due to other factors, including sampling error or some aspect of recruitment for the FMA, but for our purposes at this



Tracks in the United States are largely centered on the two coastal population centers of New York and California. Oregon is overrepresented in the FMA relative to its population size. Alaska and Hawaii are not pictured, but each area has less than ten submitted tracks in the FMA.

point it is sufficient to observe that the difference is highly statistically unlikely and these two populations can be considered different.

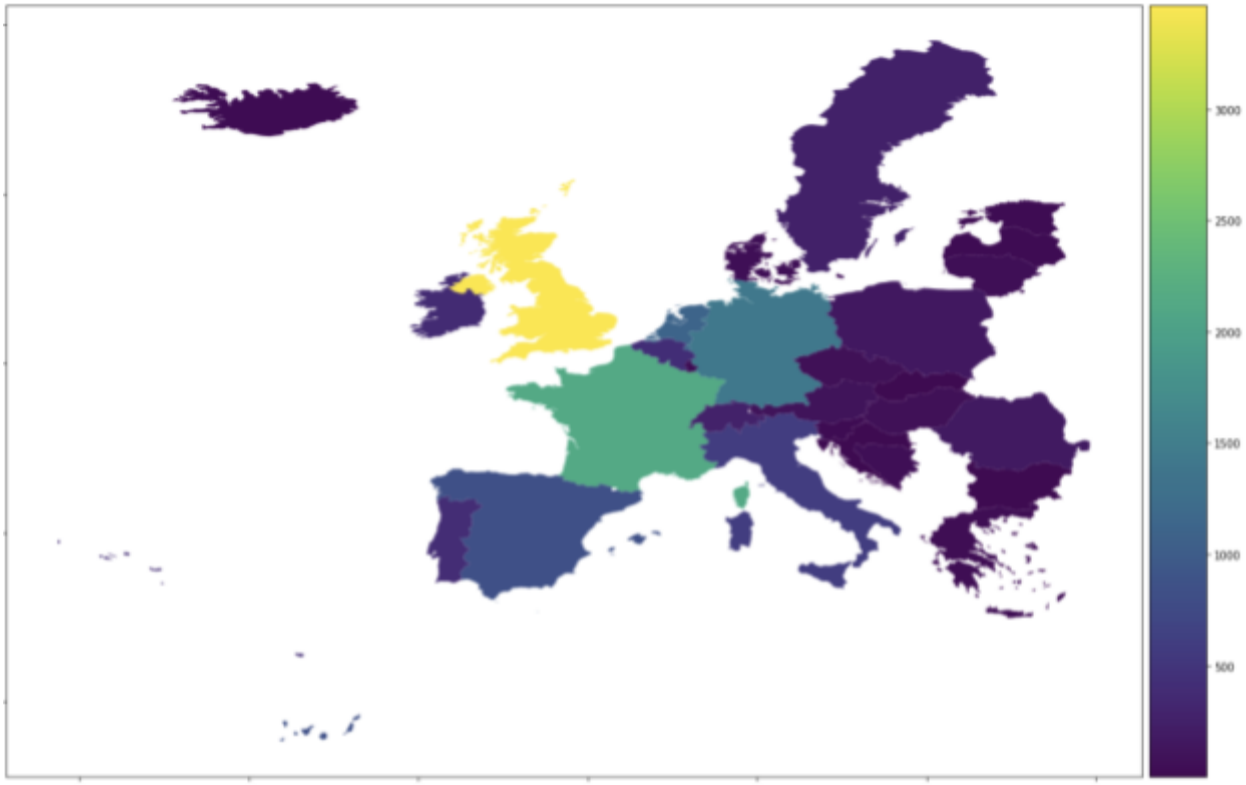
In service of some of these questions, we can then continue to drill down further into the organization of the data and consider how the information is distributed within each country. Using geopandas, we are able to assign geocoordinates to individual states by using publicly available shapefiles.

This process involves a small amount of geopandas magic, as we need to perform a join between data frames to assign coordinates to the polygons that contain them. After performing this manipulation, we are able to assess the differences in density of musicians in each country.

The North America data shows us that New York is the main hotspot for FMA data, with the state containing roughly 3500 submitted tracks. California and New Jersey fall a tier behind with roughly 2500 submitted tracks each.

Interestingly, Oregon seems to be somewhat overrepresented in the FMA data. Though it has a population of only 2% of the US total population, it is one of the highest states by submitted tracks, with nearly 2000 in the dataset, a number roughly equivalent to Pennsylvania, which has a population three times as large.

Additional investigation into these trends could consider whether this is due to a larger number of musicians submitting tracks to the FMA in these areas, or if there are simply a



European submissions are also centered on major population centers, with Britain the main focal point of submitted tracks. France and Germany are the next most active, with The Netherlands submitting a large number of tracks relative to its smaller population.

smaller number of musicians more involved in the FMA dataset, each of which has submitted a larger number of tracks.

Considering the distribution of tracks across Europe, we are able to see that Great Britain is Europe's equivalent to New York. The territory also has roughly 3500 tracks assigned. France is the next most active submitter at 2500 tracks, with Germany just behind at 2000. The remaining European countries trail far behind these three, though the Netherlands seem to be overrepresented when compared with their population.

We can see if this distribution matches language tags in the data. Despite the diverse locations providing data, we see that English language is very prevalent, comprising 94% of the total submitted tracks. Spanish and French are the next most common languages in the dataset, and these make up only 1.4% and 1.3% respectively.

From this analysis we can see that the FMA is not so much an international dataset as it is a dataset representing music for predominantly English-speaking audiences, centered mostly on the United States and Europe, with a strong bias to music produced in the major urban centers of London and New York. This gives us useful context when evaluating any applications

of our statistical findings as we direct marketing responses or attempt to find supplemental data to increase the power of this dataset.

Statistical Anaysis

In the process of investigating the data, two broad questions were addressed using frequentist statistical analysis. The first question is whether different regions differ in their representation of different musical genres. The second question is which values in the audio feature set can be considered outliers that may unduly bias the data.

In order to address the first question, a 2x3 table of values was first created using count data of the number of tracks with genre tags for experimental, rock, and electronic music based in North America and Europe. A chi-squared test was applied to the data, with the null hypothesis that there would be no difference in preference between the genre groups for the two continents (ie genre assignment is not dependent on continent). The alternative hypothesis in this case would be that genre representation is dependent on geographical reason.

In order to perform this test, we use the chisquare object available in SciPy. We first create a table dataframe in pandas, subsetting for the variables of interest, and then simply run the chisquare method. The default operation of the method tests for columnar differences, giving a significance score for each genre of music. The chi-squared test is robust to unbalanced data, as long as a minimum baseline of around five values per cell is met.

Performing the test returns significant result for the distribution of all genres, with the p-values far beyond the minimum 0.05 threshold considered to be classically significant. The difference in distribution of rock music counts is the most extreme, approaching the limits of python's significant digit representation, but broadly we can conclude that our null hypothesis is false and that when considering North America and Europe, the distribution of genre tags is dependent to some degree on continent.

The second question we address is whether we are able to identify outlier values in the audio feature data provided through FMA. Each track has an associated 518 features calculated using the Librosa library. Since these features are calculated as tracks are added to the FMA, they are present for all tracks.

For each feature, we can use the zscore function from scipy stats to calculate the number of standard deviations that each feature falls from the mean. For an initial attempt, we use the second columnar feature, a librosa measure of kurtosis. This feature has a minimum

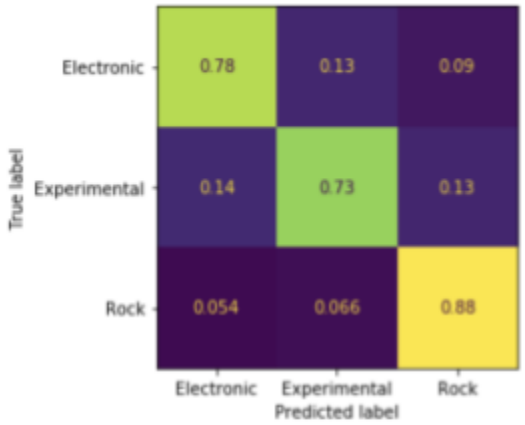
value of -1.97, a median of -0.15, and a max of 1777. This seems to point to a small number of potentially very influential values away from the mean.

Applying the zscore function to the data shows us that the distribution is strongly right skewed. We can consider that the small number of values falling more than three standard deviations away represent outliers that may bias our statistical analysis. Using numpy's array functions we can select all values with a z-score greater than three standard deviations from the mean (roughly speaking the most extreme 1% of values). Of the 106,574 tracks, 102 have values in this range for the feature in question.

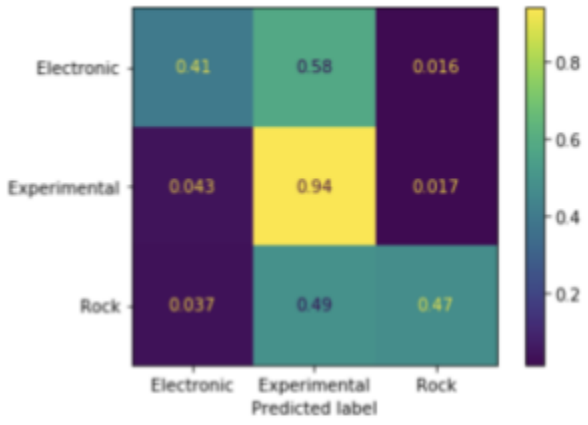
We can wrap this procedure into a function which can be applied across columns, allowing us to deal with these outlier values as we see fit. Either by substituting a specific value like nan, or by substituting a measure of centrality like the median or mean.

Machine Learning and Modeling

In order to address the primary problem of genre classification using audio features, the data must first be cleaned and organized to allow compatibility with scikit-learn's model/fit/evaluate workflow. We first organize the data to have a single column of our target outcome. This is our reference genre classification, consisting of experimental, rock, and electronic labels. We organize the Librosa audio features as a matrix of values that matches the

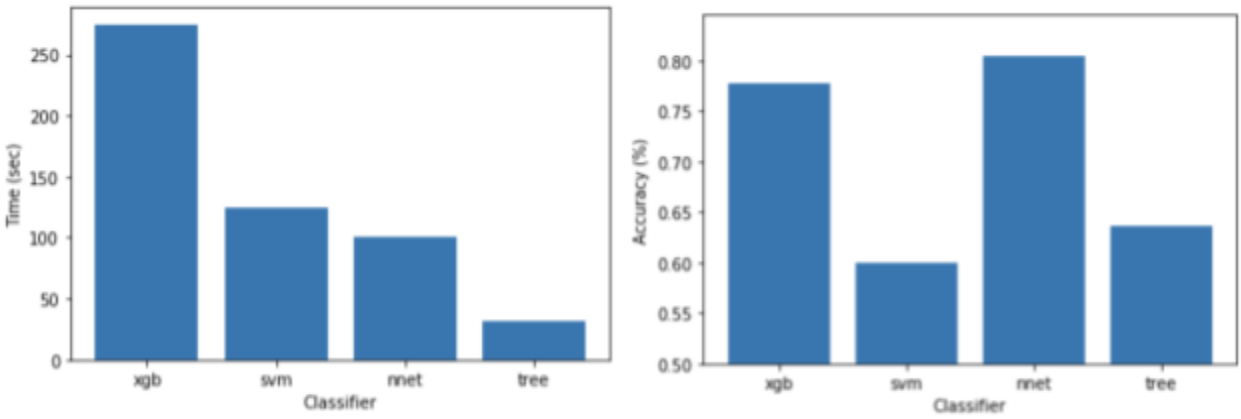


Neural Net



Support Vector Machine

Comparing the confusion matrices of the neural network classifier with the support vector machine classifier, the difference in discriminative ability is clear. We are able to see that the SVM faces significant challenges separating Electronic and Rock music from Experimental. It is also clear that despite its lower overall accuracy, the SVM does a better job of classifying Experimental music, though this may be due to a tendency to simply classify everything as experimental regardless of feature data.



Training time of the tested classifiers varies wildly, as does the resulting classification accuracy. However from this figure we can see that accuracy does not necessarily track with training time. The neural network classifier was selected for further evaluation due to its fast training time as well as its high achieved accuracy.

row position with the index of the target features. We then split into test and training data with a 25%/75% cut of the data.

Our first task is to decide which classifiers are correctly aligned with the data. Since we have primarily continuous values, we must pick a classifier that can easily deal with these. We consider three major classes of supervised classification algorithms: support vector machines (SVM), neural networks (NN), and tree-based methods.

For our purposes here, the majority of our classifiers are implemented using scikit-learn, both for simplicity and for consistency of workflow. We consider the Linear Support Vector Classification SVM implementation, the Multi-Layer Perceptron NN implementation, and the Decision Tree Classifier tree-based classifier. In order to add another more powerful reference classifier, we also import the XGBoost classifier, which has been demonstrated to perform well on continuous data sets.

In this first round of evaluation, we find that MLPClassifier achieves the highest accuracy at 81% followed by the XGBoost method which achieves 78% accuracy. The SVM (60% accuracy) and Decision Tree (64% accuracy) perform substantially worse.

We can also consider the time to train for each model. In this area, the tree method shows its strength in dealing with complex data, training in under 30 seconds, while the XGBoost method takes nearly four and a half minutes to complete. Though its accuracy is the second highest of the considered methods, the neural network trains very quickly, taking only 75 seconds to train.

From this information, we can focus on the neural network classifier as our best option for continued development. It has best-in-class performance for classification on our music data, and it also is a strong performer in training time. We can now consider hyperparameter tuning of the model to see what additional gains we might be able to achieve by further specializing this classifier for our dataset.

Using the tools available in scikit-learn, we can evaluate a variety of different hyperparameter options. The GridCV method provides a method by which to perform cross validated hyperparameter tuning. There are several fields we may wish to change to better approach our data. The alpha parameter controls the learning rate of the network, with larger values leading to larger adjustments of the weights on each iteration. Hidden-layer size dictates how many layers of neurons will be used and how many nodes each layer will contain. The learning rate and solver parameters represent different methods by which the back propagation adjustments will be made. And finally, the maximum iteration parameter determines at what point convergence of the classifier will be detected.

Once we have specified a collection of possible values for each of these parameters, we can run GridCV with five-fold cross validation. The classifier is then run across a variety of scenarios, and the best performing hyperparameters are reported.

The final recommendations of the grid search lead to a classifier that actually drops back a step to 80% classification accuracy. This likely shows that our initial neural network model was somewhat overfit to the data set. The cross validation and parameter tuning give the final mode more weight, as it has been evaluated against a variety of different circumstances.

Conclusions

The inclusion of a wide variety of data types in the FMA give a great deal of flexibility when training a classifier to infer genre from short audio tracks. However, the dataset provides several important limitations to classification tasks.

While computational audio features are well represented with very little missing data, the high degree of missing information in the track metadata causes issues with constructing a robust classifier, especially in critical fields like genre tag and geolocation data.

A classification accuracy of 80% was achieved using a simple neural network classifier. This indicates that audio feature data is enough to construct a relatively accurate classifier across three dissimilar genres. This is a problem that is likely to become much more difficult as additional genres are considered. Seventeen top-level genre classifications are included in the

FMA, though only a small subset of them contains a sufficient number of tracks for a strong comparison, and only three of which are considered in this analysis.

In order to provide a more accurate classifier, it is necessary to consider additional data sources. The FMA also contains audio data for all included tracks, meaning that with sufficient computing power it is possible to create classifiers that operate directly on the raw audio data. It is likely that employing deep learning techniques that incorporate a model of time-related awareness could allow better results. LSTM and autoencoder models may be more successful than simple multilayer perceptron models, and would also allow incorporation of additional data to form ensemble classification strategies. In particular, we show in this analysis that geographic data encodes some degree of useful information related to genre tagging.

In any event, as time goes on there will be ongoing need for genre classification strategies, as the amount of audio information entering digital marketplaces continues to grow over time along with internet access across international markets. Machine learning solutions to these problems are likely to continue to be an important part of the music ecosystem.