Automated Genre Classification on the Free Music Archive

Erik Duus
MATH637 2021

Abstract

The FMA dataset is an extensive collection of audio tracks designed as a reference dataset for Music Information Retrieval (MIR) tasks. It comprises over 100,000 full audio tracks spanning a variety of genres. It also contains extensive track-level metadata such as album and artist information, and each audio track is assigned to genres by the authors themselves. Additionally, the dataset is enriched with a set of summary statistics for spectral audio features extracted using the Librosa package. In total, there are 518 such precomputed audio features available for MIR tasks. Notably, all audio tracks are freely downloadable, thus permitting further audio feature analysis.

This work applies various machine learning classification techniques to the FMA dataset to determine the feasibility of automated genre classification. A variety of classification approaches are used to build and test single-genre classifiers. Training data augmentation and hyper-parameter tuning are performed to improve the baseline performance of the classifiers. Multi-genre classification is investigated using a fully connected neural network, and a simple training data augmentation is incorporated to assess its impact on classifier performance.

Potential Pitfalls

Exploration of genre structure via dimensionality reduction showed some pairs of genres were more easily discerned than others. For example, Isomap discerned Rock and Folk more easily than Rock and Pop.

Additionally, it seems that the FMA genre hierarchy is somewhat arbitrary and unevenly applied. For example, International is rather vague; is French pop music of the genre International or Pop? Some tracks are tagged with more than 10 genres, while most are tagged with 3 or fewer.

Finally, the dataset is quite unbalanced, leaning heavily towards Rock, Experimental, and Electronic music. Despite its size of over 100,000 tracks, it may not be a good representation of the different genres it contains.

Classic Machine Learning Classifiers

Most of the FMA dataset tracks are labeled with multiple genres, and many tracks have genre combinations that roll up to different parent genres (even improbable combinations such as Experimental-Classical and Folk-HipHop). In order to simplify the classification problem, the ‘medium’ subset of the full dataset is used. The medium subset comprises approximately 25,000 tracks with a single root genre, although the genres are unbalanced. The larger dataset provides more training exemplars and a more realistic subset of the FMA dataset.

The Spoken, Blues, and Easy Listening genres are dropped from the dataset due to minimal support, resulting in the following genre breakdown:

Root Genre	# Tracks	Root Genre	# Tracks	Root Genre	# Tracks

Rock	7103	Instrumental	1350	Jazz	384
Electronic	6314	Pop	1186	Country	178
Experimental	2251	International	1018	Soul-RnB	154
Hip-Hop	2201	Classical	619		
Folk	1519	Old-Time / Historic	510		

The imbalance between genres is noteworthy, as is the genre representation. For example, Folk is more prevalent than Pop or Country, which does not reflect these genres' respective popularity. Again, this indicates that the full FMA dataset is perhaps not wholly representative of the music universe.

SVM, Logistic Regression, and KNN Classifiers

4 classic machine learning classifiers from the scikit-learn library are built against all features of the medium dataset with default settings. Features are scaled using min-max scaling due to the non-normal distribution of many of the input features. Training and test datasets are built using an 80/20 stratified split. For simplicity, cross-fold validation is not used. The resulting accuracy at first glance seems quite good:

	SVM-Linear	SVM-RBF	Logistic	KNN
Accuracy	0.68	0.66	0.64	0.59
F1-macro	0.52	0.44	0.43	0.46
F1-weighted	0.66	0.62	0.61	0.57

However, the macro-weighted F1 scores show that the genre imbalance skews the results. Examining the genre-specific F1 scores, it's clear that the poorly-supported genres generally have lower performance. Additionally, the genres that were highlighted by the visualization exercise as difficult to discriminate also have expectedly poor performance, as shown by the F1 scores for Pop and Experimental.

	SVM-Linear	SVM-RBF	Logistic	KNN	Support
Classical	0.79	0.77	0.72	0.68	124
Country	0.15	0.00	0.00	0.22	35
Electronic	0.72	0.72	0.71	0.64	1263
Experimental	0.44	0.42	0.39	0.32	450
Folk	0.63	0.60	0.58	0.50	304
Hip-Hop	0.68	0.65	0.63	0.49	440
Instrumental	0.50	0.49	0.39	0.34	270
International	0.51	0.35	0.31	0.48	204
Jazz	0.41	0.00	0.16	0.39	77
Old-Time/Historic	0.95	0.97	0.91	0.95	102
Pop	0.12	0.00	0.06	0.15	237
Rock	0.81	0.78	0.77	0.74	1421
Soul-RnB	0.06	0.00	0.00	0.12	31

Feature Selection using Correlation

The earlier analysis noted that a number of the input features are highly correlated. Correlation analysis indicated that a .90 correlation coefficient threshold permits ~120 features to be dropped, shrinking the input feature set to 401. The same 4 classifiers are fitted as before on the reduced dataset, without a notable improvement in performance:

	SVM-Linear	SVM-RBF	Logistic	KNN
Accuracy	0.68	0.66	0.64	0.60
F1-macro	0.52	0.43	0.44	0.48
F1-weighted	0.65	0.61	0.61	0.58

The results are unsurprising since the input feature space is only reduced by 20% and is still retains 400+ dimensions. This is a fairly significant reduction in input features and is also tested on the same 4 classifiers with their default settings.

Feature Selection / Dimension Reduction using PCA

It was previously noted that PCA showed ~150 principal components explained approximately 95% of the variance of the input feature set. Since this is a more significant reduction in the number of features, PCA is fitted on the training set to determine the number of components for the 95% explained variance ratio. Then PCA with the selected components is used to transform the training and test sets, and the resulting datasets fitted to the 4 classifiers. Again the results show no notable improvements:

	SVM-Linear	SVM-RBF	Logistic	KNN
Accuracy	0.67	0.69	0.66	0.59
F1-macro	0.50	0.52	0.49	0.47
F1-weighted	0.64	0.67	0.63	0.58

The results are also not that surprising. PCA did not return a small number of components that explain the bulk of the variance, so the resulting feature space is still rather larger. In addition, PCA transforms the features (not exactly feature selection), so any improvement due to the reduced number of features might be offset by the transformation of the data.

The subsequent analysis proceeds with the original medium dataset with no feature pruning or PCA transformation.

Addressing Training Data Imbalance with Class-weighting

Given the poor performance of the poorly-supported classes, addressing the training set imbalance could improve classification results. One way this can be done is by acquiring more data, which is not an option here. Another approach is to synthesize new data based on characteristics of the existing data, or based on a model representation of the data. Alternatively, the training data can be balanced by randomly oversampling underrepresented classes, and undersampling the overrepresented class. Finally, the classification algorithms themselves can be modified to pay more attention to minority classes during training.

As examples of the last approach, SVM and LogisticRegression provide the ability to weight classes according to their representation, therefore paying more attention to smaller genre classes. SVM-Linear, SVM-RBF, and LogisticRegression are fitted to the medium dataset with the usual 80/20 stratified test/training split. A modest improvement in F1 scores over the baseline classifiers is observed, though at the expense of some overall accuracy.

	SVM-Linear	SVM-RBF	Logistic
Accuracy	0.61	0.58	0.55
F1-macro	0.54	0.50	0.48
F1-weighted	0.63	0.60	0.58

Examination of the genre-specific F1 scores reveals significant improvement in the worst performers from the baseline classifiers. The better-performing genres from the baseline classifiers show slightly diminished results.

	SVM-Linear	SVM-RBF	Logistic	Support
Classical	0.74	0.73	0.70	124
Country	0.26	0.21	0.20	35
Electronic	0.66	0.62	0.58	1263
Experimental	0.44	0.41	0.38	450
Folk	0.62	0.61	0.59	304
Hip-Hop	0.64	0.62	0.60	440
Instrumental	0.47	0.45	0.44	270
International	0.48	0.42	0.42	204
Jazz	0.46	0.39	0.31	77
Old-Time/Historic	0.94	0.94	0.92	102
Pop	0.24	0.21	0.22	237
Rock	0.78	0.75	0.75	1421
Soul-RnB	0.26	0.18	0.13	31

Hyper-parameter Tuning

A round of elementary hyper-parameter tuning was performed on the 4 class-weight-balanced classifiers. Sci-kit's HalvingRandomSearch is used with a small subset of commonly-tuned parameters for each classifier. The intent of this exploration is to determine if parameter tuning can be beneficial, not to identify an optimal classifier.

HalvingRandomSearch uses successive halving to search the parameter space. A large set of randomly selected parameter combinations are each evaluated with a small subsample of the training data. The winning combinations are then evaluated with successively larger subsets until a final winner is selected. The following parameters for each classifier are chosen for tuning:

- SVM-Linear: C (regularization)
- SVM-RBF: C and gamma (RBF kernel coefficient)
- LogisticRegression: C
- KNN: n_neighbors

The class-weighted classifiers from the prior step are used for the tuning exercise. A substantial improvement is noted in SVM-RBF. All other classifiers show marginal improvement at best.

	SVM-Linear	SVM-RBF	Logistic	KNN
Accuracy	0.61	0.69	0.59	0.59
F1-macro	0.53	0.63	0.51	0.47
F1-weighted	0.63	0.69	0.61	0.57

The genre-specific F1 scores shows that SVM-RBF shows improvement across all genres:

	SVM-Linear	SVM-RBF	Logistic	KNN	Support
Classical	0.72	0.83	0.75	0.68	124
Country	0.27	0.55	0.23	0.23	35
Electronic	0.66	0.73	0.62	0.64	1263
Experimental	0.42	0.53	0.42	0.31	450
Folk	0.62	0.67	0.59	0.49	304
Hip-Hop	0.63	0.67	0.64	0.53	440
Instrumental	0.47	0.54	0.43	0.34	270
International	0.46	0.61	0.46	0.45	204
Jazz	0.46	0.57	0.35	0.40	77
Old-Time/Historic	0.94	0.99	0.94	0.93	102
Pop	0.23	0.29	0.26	0.10	237
Rock	0.78	0.82	0.78	0.73	1421
Soul-RnB	0.24	0.40	0.21	0.21	31

All genres except Pop and Soul have an F1 score above 0.50. Country shows a dramatic improvement, although Pop still lags substantially.

Analysis

SVM-RBF performs extremely well on the single-genre classification task. This is expected as support vector machines perform well with large feature spaces. The set of support vectors only involve a small number of training instances, and consequently, SVM does not necessarily require information from all feature dimensions to fit a separating hyperplane. Using the RBF kernel adds non-linear separating surfaces.

KNN may suffer from both the curse of dimensionality and the class imbalance. A majority class can invade the ‘neighbor space’ of a minority class, thus degrading nearest-neighbor performance.

LogisticRegression is a linear model and therefore prone to overfitting in high dimensions due to the large number of parameters.

Ensembles - RandomForest and XGBoost

Ensemble techniques have some advantages over the classifiers used so far. They can fit complex separating surfaces between classes and yet are resistant to overfitting. They can also focus attention on hard-to-classify classes (XGBoost in particular). The same dataset with the usual 80/20 stratified training/test split is used, this time without min-max scaling. RandomForest and XGBoost are used to fit classifiers with default parameters. The baseline XGBoost performance is quite good, with an initial F1 score above the earlier baseline classifiers.

	RF	XGB
Accuracy	0.64	0.69
F1-macro	0.45	0.55
F1-weighted	0.60	0.67

Examining the genre-specific F1 scores reveals the expected poor performance of the poorly-supported classes together with Experimental and Pop.

	RF	XGBoost	Support
Classical	0.76	0.83	124
Country	0.11	0.20	35
Electronic	0.68	0.74	1263
Experimental	0.33	0.45	450
Folk	0.60	0.64	304
Hip-Hop	0.56	0.69	440
Instrumental	0.43	0.48	270
International	0.43	0.57	204
Jazz	0.17	0.96	77
Old-Time/Historic	0.96	0.18	102
Pop	0.07	0.18	237
Rock	0.77	0.81	1421
Soul-RnB	0.00	0.12	31

Addressing Training Set Imbalance with Class-weighting

In an attempt to improve the performance of the under-represented classes, another RandomForest is fit with class_weight='balanced'. XGBoost does not have an identical option but does have scale_pos_weight parameter, which tells XGBoost to overweight minority classes. No performance improvement is detected:

	RF-balance	XGB-balance
Accuracy	0.63	0.69
F1-macro	0.45	0.55

F1-weighted	0.59	0.67

After further investigation, it appears the XGBoost parameter is only relevant for binary classification. The lack of improvement in RandomForest is a bit surprising and requires more research to explain.

Hyper-parameter Tuning

HalvingRandomSearch is again used to determine if hyper-parameter tuning can produce improvement. The following parameters are included in the tuning search:

- RandomForest: n_estimators (number of trees), max_features (number of features to consider at a node)
- XGBoost: n_estimators, learning_rate

RandomForest shows no improvement. XGBoost does show a modest improvement in both accuracy and macro-weighted F1 score:

	RF-tuned	XGB-tuned
Accuracy	0.63	0.71
F1-macro	0.45	0.58
F1-weighted	0.59	0.69

	RF-tuned	XGB-tuned	Support
Classical	0.79	0.83	124
Country	0.11	0.28	35
Electronic	0.67	0.76	1263
Experimental	0.29	0.49	450
Folk	0.63	0.66	304
Hip-Hop	0.53	0.71	440
Instrumental	0.46	0.51	270
International	0.38	0.60	204
Jazz	0.21	0.46	77
Old-Time/Historic	0.95	0.97	102
Pop	0.07	0.23	237
Rock	0.76	0.82	1421
Soul-RnB	0.00	0.23	31

Analysis

RandomForest significantly underperformed XGBoost. This may simply be due to its construction and the imbalanced training data; RandomForests grows decision trees randomly, with no attention paid to

minority classes. XGBoost, by contrast, focuses on misclassifications with each round of training, so even with an imbalanced training set, hard-to-classify classes are given extra focus.

Training data with better genre balance could improve the performance of both classifiers. Further research with downsampling and data augmentation should be pursued.

MLPClassifier

Given the high-dimensional input feature space and the lack of a clear genre structure visible through dimension reduction, neural networks seem like a good match for the genre classification problem. Since the input feature space is exclusively summary statistics, a straightforward, fully connected network probably the appropriate architecture. CNN or LSTM would be a better choice if the temporal structure of the audio were available.

Again, the medium subset is split into training and test data using the usual 80/20 stratified split. A scikit-learn MLPClassifier is fitted using the training data using default parameters and no cross-validation. The classification report reveals decent baseline performance, similar to XGBoost:

	precision	recall	F1 score	support
accuracy			0.69	4958
macro avg	0.67	0.52	0.54	4958
weighted avg	0.68	0.69	0.67	4958

Genre-specific F1 scores are similar to those from SVM-RBF and XGBoost, with the usual genres underperforming:

	MLP	Support
Classical	0.78	124
Country	0.23	35
Electronic	0.74	1263
Experimental	0.47	450
Folk	0.64	304
Hip-Hop	0.68	440
Instrumental	0.48	270
International	0.56	204
Jazz	0.38	77
Old-Time/Historic	0.17	102
Pop	0.17	237
Rock	0.81	1421
Soul-RnB	0.17	31

Balancing the Training Data via Over/Undersampling

To address the genre imbalance of the training data, over and undersampling are employed. For each genre with support < 500, the training set is randomly oversampled to bring the genre support to 500.

Additionally, for each genre with support > 3500, the training set is randomly undersampled to bring the genre support down to 3500. The test set is left unchanged, and an MLPClassifier trained on the rebalanced training data with default settings. A modest improvement in F1 score is noted, with little impact on overall accuracy:

	precision	recall	F1 score	support
accuracy			0.68	4,958
macro avg	0.58	0.58	0.58	4,958
weighted avg	0.67	0.68	0.67	4,958

The genre-specific F1 scores show improvement in the bottom performers:

	MLP	Support
Classical	0.78	124
Country	0.29	35
Electronic	0.72	1263
Experimental	0.45	450
Folk	0.63	304
Hip-Hop	0.67	440
Instrumental	0.49	270
International	0.57	204
Jazz	0.39	77
Old-Time/Historic	0.96	102
Pop	0.20	237
Rock	0.80	1421
Soul-RnB	0.27	31

Balancing the Training Data via Oversampling with SMOTE

An alternative to oversampling is the synthesis of new training exemplars. One such approach is the Synthetic Minority Oversampling Technique (SMOTE) [1], which creates a new minority class instance from a set of minority class neighbors. The algorithm randomly selects a minority class instance and then locates a number of its nearest minority class neighbors. A new instance is then created by interpolating between the first instance and a randomly selected neighbor.

As before, the majority classes in the training data are undersampled down to 3500 instances. Then SMOTE is used to oversample the minority classes up to 500 instances. An MLPClassifier with default parameters is then trained on the rebalanced dataset with the following results:

	precision	recall	F1 score	support
accuracy			0.66	4,958
macro avg	0.55	0.58	0.55	4,958

weighted avg	0.55	0.66	0.66	4,958

The resulting performance is slightly worse than the randomly oversampled MLPClassifier. Examining the genre-specific scores provides some insight:

	MLP	Support
Classical	0.78	124
Country	0.33	35
Electronic	0.70	1263
Experimental	0.43	450
Folk	0.64	304
Hip-Hop	0.65	440
Instrumental	0.47	270
International	0.55	204
Jazz	0.46	77
Old-Time/Historic	0.97	102
Pop	0.13	237
Rock	0.80	1421
Soul-RnB	0.29	31

The SMOTE-trained classifier performs similarly to the randomly oversampled classifier across most genres. However, Pop is a notable exception, with a significant decrease in performance. Two (related) hypotheses come to mind:

- SMOTE was unlucky and picked difficult-to-classify instances as the basis for synthesis.
- The Pop genre does not have an under-representation problem; it has a difficult-to-classify problem. Synthetic instance generation using challenging instances as a basis may produce even more challenging instances.

The first hypothesis can be ruled out through more synthesize / train / validate / test cycles. Given the poor performance of the Pop genre across all classifiers so far, the second hypothesis seems more likely.

Hyper-parameter Tuning the MLPClassifier

Scikit-learn HalvingGridSearch is used to search the hyper-parameter space for a more optimal classifier. The search explores several hidden layer arrangements and different values of the learning rate. Again the intent is to determine if tuning can improve performance, not to identify an optimal classifier. F1 performance is slightly improved and on par with SVM-RBF and XGBoost. F1 scores are shown here against the baseline and rebalanced classifiers for comparison:

	MLP	MLP rebalanced	MLP tuned
Accuracy	0.69	0.68	0.67
F1-macro	0.54	0.58	0.60

F1-weighted	0.67	0.67	0.67

This classifier improves F1 scores to above 0.4 for all genres except Pop:

	F1	support
Classical	0.79	124
Country	0.53	35
Electronic	0.74	1263
Experimental	0.48	450
Folk	0.64	304
Hip-Hop	0.43	440
Instrumental	0.59	270
International	0.52	204
Jazz	0.52	77
Old-Time / Historic	0.97	102
Pop	0.23	237
Rock	0.80	1421
Soul-RnB	0.48	31

Analysis

MLPClassifier performs quite well after data augmentation and tuning, rivaling the SVM-RBF and XGBoost classifiers. This is not unexpected since neural networks can fit arbitrarily complex functions and perform well with enough training data. The unbalanced training data again causes certain genres to perform poorly. Additionally, the Pop genre continues to be hard to classify, even though it is moderately well-represented in the training data.

Data augmentation using random oversampling and SMOTE both show promise as a means to introduce more balance to the training data. While SMOTE underperformed random oversampling in this limited investigation, both approaches merit further investigation.

Multi-Genre Classification using MLPClassifier

The preceding analysis was restricted to tracks with genres that share a single root genre, thereby eliminating more than half the dataset. While this choice simplified the classification problem, it may be suboptimal for several reasons:

- The selected tracks may not be representative of the full corpus.
- Eliminating more than half the dataset from consideration may remove important exemplars.
- Some genres may be more prevalent in combination.

Therefore it may be more appropriate to reframe genre classification as a multi-label problem. Music genres might be viewed as an ‘influence’ or a ‘flavor’ as opposed to a single identity. Intuitively this seems natural; genre combinations like Country-Rock and Electronic-Pop are real and popular.

Given the size of the FMA genre hierarchy and the inconsistency in the number of genres tagged per track, some crude processing is performed to streamline the multi-genre problem. The genre list for each track is rolled up into a root genre list. In addition, the number of contributions to each root genre is noted, and the 2 most contributed root genres are used for the classification analysis.

The MLPClassifier from scikit-learn is used to build the multi-label classifier. It naturally supports multi-label classification since it generates probabilities for each class, so a multi-label output is produced by selecting those classes above a probability threshold.

Tracks with <= 2 root genres from the full dataset are selected, providing ~87,000 tracks for analysis. The genre breakdown is still extremely imbalanced but quite different from the single root genre dataset.

genre	# tracks	genre	# tracks	genre	# tracks
Experimental	28,001	Folk	8,272	Country	1,191
Electronic	26,084	Hip-Hop	6,810	Soul-RnB	926
Rock	25,554	International	3,623	Old-Time / Historic	792
Instrumental	9,079	Classical	2,695	Blues	689
Pop	9,076	Jazz	2,359		

To prepare the data for training, the root genre list is binarized, which is similar to one-hot encoding but allowing multiple classes to be represented per instance. An MLPClassifier with default settings is trained on an 80/20 split of the dataset, with no attempt at stratification:

accuracy	hamming loss	hamming loss - zeros
0.3208	0.0757	0.1025

Accuracy measures if the predicted labels match the true labels exactly; this is a stringent measure that gives no credit for a partial match. Hamming Loss is often used to score multi-label classification, but here it is not a good fit. It measures the ‘distance’ from the predicted labels to the true labels. Since genre classification is dealing with 15 classes, with only 1 or 2 predicted, the predicted and true labels are sparse. As seen in the table above, Hamming Loss on prediction of all zeros scores similarly to the trained classifier’s predictions.

The F1 scores from the classification report seem more appropriate. The scores are computed for each label, and several different averages are produced:

	precision	recall	F1 score	support
micro avg	0.72	0.42	0.53	25009
macro avg	0.63	0.28	0.35	25009
weighted avg	0.7	0.42	0.5	25009
samples avg	0.55	0.47	0.49	25009

Precision scores are much higher than recall. This is due to the 50% probability threshold used by MLPClassifier, leading the classifier to ‘underpredict.’ While MLPClassifier does not support changing the

threshold for training, the predictions of the test set can be modified by using a lower probability threshold, in this case 30%:

	precision	recall	F1 score	support
micro avg	0.57	0.61	0.59	25009
macro avg	0.53	0.41	0.43	25009
weighted avg	0.56	0.61	0.57	25009
samples avg	0.61	0.65	0.6	25009

Overall F1 scores increase at the expense of some precision. Examining the genre-specific F1 scores shows extremely poor performance from the thinly-supported genres.

	F1	support
Blues	0.07	139
Classical	0.55	537
Country	0.07	231
Electronic	0.67	5267
Experimental	0.65	5573
Folk	0.52	1613
Hip-Hop	0.51	1350
Instrumental	0.37	1797
International	0.42	732
Jazz	0.33	478
Old-Time / Historic	0.84	161
Pop	0.30	1819
Rock	0.69	5113
Soul-RnB	0.00	199

It is also notable that Pop shows better performance versus the single genre classifiers. This could be due to the greater genre representation. It could also be due to the classifier making use of hybrid genre structures. For example, if Pop occurs most often as part of a hybrid genre, then the presence of the paired genre becomes a hint that Pop is more likely.

Addressing Training Set Imbalance

The training data is severely unbalanced, with a nearly 50-fold spread between the smallest minority class and the largest majority class. Undersampling and data augmentation would typically be applied to a single label dataset. For a multilabel dataset, however, these techniques are not supported by standardized implementations.

A simple approach to augmenting multilabel data is to randomly oversample the label powerset. Each combination of labels is essentially treated as a class, and thus the combinations are brought into balance.

A more interesting approach is LP-ROS [2], which proposes an oversampling method that works cloning random samples of minority label sets until the size of the multi-label dataset is 25% larger than the original. The authors propose several measures of class imbalance that they use to assess the performance of LP-ROS, and its undersampling corollary, LP-RUS.

A synthetic data generation approach MLSMOTE [3] extends SMOTE to the multi-label case. Minority instances are identified on a per-label basis, and synthetic instance generation then proceeds in an analogous fashion to the single-label case.

While the preceding approaches are interesting, broadly available implementations of either are not available. To simply test whether data augmentation produces beneficial results, a crude oversampling is performed where all tracks are duplicated that are not tagged with Experimental, Electronic, or Rock. Modest improvement is noted:

	precision	**recall**	**F1 score**	**support**
micro avg	0.73	0.43	0.54	25009
macro avg	0.64	0.31	0.37	25009
weighted avg	0.7	0.43	0.51	25009
samples avg	0.56	0.48	0.5	25009

Reclassifying predictions with a 35% threshold, we again see modest improvement over the baseline classifier:

	precision	**recall**	**F1 score**	**support**
micro avg	0.56	0.64	0.60	25009
macro avg	0.48	0.49	0.47	25009
weighted avg	0.56	0.64	0.59	25009
samples avg	0.6	0.68	0.61	25009

The genre-specific F1 scores show a definite improvement in the less-represented genres:

	F1	**support**
Blues	0.20	139
Classical	0.58	537
Country	0.24	231
Electronic	0.68	5267
Experimental	0.67	5573
Folk	0.51	1613
Hip-Hop	0.54	1350
Instrumental	0.40	1797
International	0.38	732
Jazz	0.38	478

Old-Time / Historic	0.84	161
Pop	0.34	1819
Rock	0.69	5113
Soul-RnB	0.09	199

Hyper-parameter Tuning

Using a larger network of 400 hidden nodes and a larger regularization parameter, a configuration which showed some success in the previous analysis, modest improvement is again noted:

	precision	**recall**	**F1 score**	**support**
micro avg	0.69	0.49	0.57	25009
macro avg	0.6	0.37	0.44	25009
weighted avg	0.67	0.49	0.56	25009
samples avg	0.59	0.54	0.54	25009

Reclassifying predictions with a 35% threshold, we again see modest improvement, but no better than the smaller network:

	precision	**recall**	**F1 score**	**support**
micro avg	0.59	0.6	0.6	25009
macro avg	0.51	0.45	0.47	25009
weighted avg	0.59	0.6	0.59	25009
samples avg	0.61	0.64	0.6	25009

The lack of improvement could be due to the fact that the larger network is able to learn information about the relationships between genres; common pairs of genres could be important indicators.

Analysis

While multi-genre classification is more complex than the single-genre problem, it merits attention for several reasons:

- A large portion of recorded music is multi-genre; single genre recognition is not a representative solution.
- The restriction of the training universe to audio tracks labeled with a single genre vastly reduces the quantity of available training data.
- There might be important structure present in the genre combinations of the training data that could contribute to genre classification.

Summary

A subset of the FMA dataset is used to train 4 classic ML classifiers for single genre recognition. They all display some ability to discern genres, with SVM and a RBF kernel performing the best. Ensemble methods are also applied, with XGBoost performing similarly to SVM. Finally, an MLPClassifier is built

that also performs close to SVM-RBF and XGBoost. Some genres, such as Pop, are identified as being harder to classify despite having reasonable representation in the training data.

Several data augmentation techniques are applied to address the genre imbalance of the training data, and all produce some improvement in the classification performance of minority classes. The results highlight the need for a broader set of training data with better representation of all major genres.

Finally, genre classification is reframed as a multi-label classification problem. A much larger subset of the FMA dataset is enhanced with a simplified set of labels. An MLPClassifier is trained on the new dataset and used for multi-genre prediction with some success. The results also demonstrate that multi-genre classification exacerbates the genre imbalance problems of the dataset.

Several multi-label data augmentation techniques are discussed. A rudimentary oversampling is performed, showing some modest improvement in performance. As with the single-genre case, the genre imbalance of the dataset is a drag on classifier performance.

While the FMA dataset is a good step towards a benchmark dataset for MIR research, and genre classification specifically, this work has identified areas for improvement:

- The genre hierarchy seems arbitrary; a reference hierarchy is needed.
- The accuracy of genre assignments is questionable.
- The dataset is imbalanced, with poor representation of some major music genres.
- The music collection might not be representative since it is from less well-known artists.

Although most of this work experiments with single-genre classification, multi-genre classification is also investigated. The multi-genre problem is probably the more fruitful problem to address for the reasons discussed earlier. However, multi-label classification also introduces problems with data augmentation:

- Proper stratification of test and training sets.
- Under and over-sampling.
- Synthetic data generation.

The audio features available with the FMA dataset show some ability to discriminate genres, but none of the classifiers work well across all genres. While the dataset imbalance is partly to blame, the audio features are probably incapable of discerning all genres. In addition to the points discussed above, improved classifier performance probably requires:

- Full spectral features (not summary statistics) that retain temporal structure.
- Additional features such as rhythm, lyric content, etc.
- Classifier architectures that can work with feature structure: CNN and LSTM come to mind.

References

[1] Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* 16 (2002): 321-357.

[2] Charte, Francisco, et al. "A first approach to deal with imbalance in multi-label datasets." *International Conference on Hybrid Artificial Intelligence Systems*. Springer, Berlin, Heidelberg, 2013.

[3] Charte, Francisco, et al. "MLSMOTE: approaching imbalanced multilabel learning through synthetic instance generation." *Knowledge-Based Systems* 89 (2015): 385-397.