Rock or Not?

Defne Tuncer [1] **Kutay Barcin** [1]

Abstract

In the era of technology, millions of songs are brought to people everyday. The dramatic increase in the size of music collections has made the music genre recognition (MGR) an important task on machine learning. The goal of this paper is to give machines a chance to predict music genres given input features from music tracks. To do that, we applied various techniques based on machine learning on the dataset called Free Music Archive (FMA), and we have reached an accuracy score of 67.80% as our highest.

1. Introduction

When there is people, there is music. As people, living in today's world, music is always at our reach through technology. The ease of it has brought the demand of automatically generated playlists and customized music recommendations. The task in both those challenges is to be able to group songs in semantic categories. In this work, we aim to model and classify music genres with the assumption of different music genres are also different at the bit level.

In this paper, we will put forward the efforts we made concerning the classification models that allow us to recognize the genre of a given song from its audio features. As for the beginning, we introduced studies on the subject music genre recognition. Then we made a brief introduction to the dataset we bring into use, and explained how we handled our data. Thereafter, we implemented various baseline classification models, and discussed towards advancing the models to solve the problem of music genre recognition. These methods include: 4.1.1 Nearest Neighbor Classifier with/without dimensionality reduction through Principal Component Analysis (PCA) and weighting hyperparameter. 4.1.2 Logistic Regression through one-vs-one scheme, multinomial approach and one-vs-rest scheme with variety of solvers and regularization. 4.1.3 Support Vector Machines with linear and radial basis function (RBF) kernels. 4.1.4 Deep Learning method Neural Network also known as Multi-Layer Perceptron through various optimizers. To represent the audio tracks in building our baseline models we planned to use the combination of all the features, which have been shown to be effective in the task of predicting genres. We improved our methods with model and feature selection by using k-fold cross validation afterwards. Based on the results obtained from the algorithms, we performed experimental analysis. Finally, ended our work with a detailed conclusion, and proposed our feature work.

[1]Department of Computer Engineering, Hacettepe University, Ankara, Turkey. Correspondence to: Defne Tuncer <defnetuncer@hacettepe.edu.tr>, Kutay Barcin <kutaybarcin@hacettepe.edu.tr>.

BBM406 Fundamentals of Machine Learning, Fall 2018.

2. Related Work

For the music genre recognition task, the most common datasets are GTZAN (Tzanetakis & Cook, 2002), Million Song Dataset (MSD) (Bertin-mahieux et al., 2011) and FMA: A Dataset For Music Analysis (Defferrard et al., 2017). While FMA, which consists of 161 sub-genres among 106,574 tracks and published in 2017, is the most up-to-date dataset, and is especially suited for MGR as it features fine genre information. A challenge took place as one of challenges of Web Conference (WWW2018) by the publishers of FMA Dataset on the subject predicting genres of the music (Defferrard et al., 2018). The winner succeeded by examining through artist-related information and scored an accuracy of 66.29% on predicting 16 genres (Kim et al., 2018).

In Music Information Retrieval (MIR), there have been various number of studies on building effective models to predict genre of music using audio features. Mel-Frequency Cepstral Coefficients (MFCCs), one of the audio features, are generally used in music genre classification as the perceptual scale of pitches of a human hearing are represented by the Mel-scale. A Hidden Markov model with MFCCs is used to classify pop, country, jazz and classical genres in (Shao et al., 2004). On the other hand, another study focuses on a new feature called Renyi Entropy Cepstral Coefficients (RECCs) (Tsai & Bao, 2010). The highest achieved accuracy scores reported on the datasets ISMIR2004 which is from the contest (Cano et al., 2006) and GTZAN are accomplished by representing the auditory human perception with a proposed spectrogram (Panagakis et al., 2009). Most of their studies are done through researching the timbre texture,

the pitch and rhythmic content as well as their combinations. In addition to works focused on the features, an appealing study (Sanden & Zhang, 2011) is done by training various classifiers on the same data, and then combining the results into a single classification.

As for the learning models, one of the first successful implementation of Support Vector Machines on music genre classification was done by applying multiple layers of SVMs which achieved over 90% accuracy using a dataset with only four genres (Xu et al., 2003). Furthermore, in recent years, both deep learning and neural network models have been shown to be accurate in a great degree for music genre recognition. A study analyzes into potent learning algorithms for genre classification based on audio waveforms (Haggblade et al., 2011). In their work, they achieved higher accuracy scores by using neural network than other models when all the models are trained on the audio feature MFCC. In addition, Convolutional Neural Networks as an example have reached far beyond human capacity in this task. A study (Li et al., 2010) used CNN as a feature extractor of songs, and took MFCC audio feature where they trained a pattern extraction in order to recognize the genre of a song.

3. Dataset Exploration

The FMA dataset, a dump of the Free Music Archieve, includes 106,574 tracks with 161 sub-genres. In this task, we use 38,990 of the tracks with 15 top-genres sampled considering their metadata and popularity for computational efficiency. Figure 1 visualizes our data, which includes clips of 30s and an unbalanced distribution among genres that differ from 24 to 14,182 clips per top-genre.

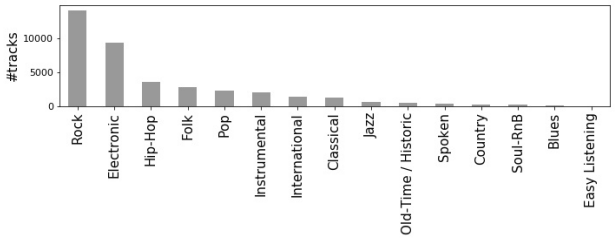

Figure 1. Top-Genre Distribution

While extracting features, each clip is processed through an audio analysis library librosa (McFee et al., 2018). Thus, each track contains 518 attributes categorized in 11 audio features; Mel Frequency Cepstral Coefficients (mfcc), Chroma Features (chroma cens, chroma cqt, chroma stft), Spectral Features (spectral bandwidth, spectral centroid, spectral contrast, spectral rolloff), RMS Energy (rmse), Tonal Centroids (tonnetz), Zero Crossing Rate (zcr). Each of these features are stored as statics, including kurtosis, max, mean, median, min, skew and std.

We split our data preserving the percentage of tracks per genre as a reflection of population (stratified sampling) into training, validation and test by 80/10/10%. Thus, our training data turned into a matrix of 31,386 rows and 519 columns consists of 518 audio features and a genre label.

4. The Approach

Starting with the assumption that examples from the same genres are similar, they will cluster closer to each other in the n-dimensional space, where n is the number of features. We discussed and compared several linear and non-linear classification methods in order to figure out which approaches best suited for our problem.[1]

4.1. Classification Methods

4.1.1. Nearest Neighbors Classifier

k-Nearest Neighbors Classifier (kNN) is a non-parametric method used for classification and regression. Although kNN is an easy implemented algorithm and performs well on a large number of classification problems, it suffers from the curse of dimensionality. Our model has a dimension space of 518 features which makes kNN vulnerable. In order to overcome this, we planned to apply Principal Component Analysis (PCA) to our matrix which reduces the input to a lower desired dimension. Figure 2. visualize scatter plot of two genres Rock and Classical after applying PCA to reduce the feature dimensions to three dimensions.

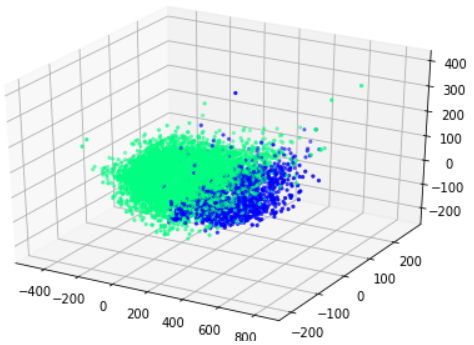

Figure 2. Scatter plot of Classical and Rock Genres

[1]Each of the approaches we implemented can be found in https://github.com/HUbbm409/rock-or-not/

4.1.2. Logistic Regression

Logistic Regression is a technique from the statistics field and provides a probability score for observations. It is a go-to method for binary classification problems, however for our multiclass problem, we implemented our Logistic Regression model in variety of approaches. These are one-vs-one scheme, multinomial approach and one-vs-rest (OvR) scheme. One-vs-one scheme uses sigmoid function while computing the probabilities of each labels, and normalizes these probabilities for all classes. Multinomial approach on the other hand minimizes the loss so that it fits across the entire probability distribution while using cross entropy loss. OvR scheme as for the last approach, takes a shape to fit one classifier per label that is a class is properly fitted towards the other classes for each classifier.

All of the approaches are additionally examined through different solvers and regularization. As for the solvers our alternatives are: (i) A Library for Large Linear Classification (liblinear) which takes the linear combination of the input features. (ii) Newton (newton-cg) uses a well performed quadratic function minimization as quadratic approximation is used. (iii) Stochastic Average Gradient (sag) and (saga) where sag applies a Stochastic Gradient descent while saga applies its advanced, unbiased version. (iv) Limited-Memory Broyden-Fletcher-Goldfarb-Shanno Algorithm (lbfgs) applies an approximation to the inverse Hessian Matrix. However, as it stores a small amount of vectors to represent approximation, we expect it to perform poorly on our large dataset.

4.1.3. Support Vector Machine

Support Vector Machine is a supervised learning method that can be used for classification. SVM works efficiently with high dimensional features even if the number of dimensions is greater than the number of samples, and it is also robust to noise. Various Kernel functions (SVC with linear kernel, SVC with radial basis function kernel and SVC with polynomial kernel) can be implemented for the decision function. SVM with linear kernel works efficient when the dataset is linearly separable as the decision region can be linearly constructed. For SVM with RBF kernel, the same assumption also applies. However, SVM with RBF kernel generates combinations of the features non-linearly in order to uplift the samples onto a more advanced dimensional feature space, and then the labels are separated by a linear decision boundary.

We preprocessed the data before applying both SVM with Linear and RBF kernels using standard scaler to represent standard normally distributed data. As for multi-class approach, SVM uses one-against-one scheme for classification. This method is consistent, which is not true for one-vs-rest classification.

4.1.4. Neural Network

A Neural Network or also known as Multi-Layer Perceptron (MLP) can learn a non-linear function estimator for both classification and regression. For our last approach we built a Supervised Deep Learning method MLP Classifier and discussed experimental results after applying feature and model selection. The model we have worked with optimizes the cross entropy loss function by using Softmax as activation function for the output layer. Before training our network, data is preprocessed as MLP is sensitive to feature scaling. Model is tuned by using k-fold cross validation on the following parameters: (i) Activation functions for the hidden layers which are tanh, rectified linear units (relu) and logistic, (ii) the size of hidden layers, (iii) the number of neurons in hidden layers, (iv) solvers which are lbfgs, sgd and adam.

5. Experimental Results

For our baseline models, we approached with three classification methods: Nearest Neighbors (kNN) / Weighted Nearest Neighbors (Weighted kNN), Logistic Regression and Support Vector Machine (SVM) with linear kernel. Each of the models was evaluated using the same training data of 31,386 clips, tuned on the validation data of 3,738 clips and tested on the 3,866 clips. The following Table 1. shows the accuracy performance for baseline models obtained with all the 11 audio features of 518 dimensions without further selection.

Table 1. Baseline Classification Methods for All Features

Models	Train Acc.(%)	Test Acc.(%)
kNN	67.94	60.14
Weighted kNN	99.89	59.65
Logistic Reg.	73.20	65.13
SVM Linear	71.32	63.35

All the methods used in the baseline models appear to have difficulties in capturing the non-linearities of the data, thus they achieve less accuracy than expected. As for the solution, we decided to combine our training and validation sets in order to improve accuracy rates on minor genres as less data will be wasted. We further apply model and feature selection with k-fold cross validation where k = 5. In addition to our models, we implemented Neural Network as a Deep Learning approach. Therefore, each of the models was evaluated using the same training data of 35,124 clips, and tested on the 3,866 clips. The following Table 2. shows the accuracy performance after the implementation of model and feature selection through regularization with cross validation.

Table 2. Improved Classification Methods

MODELS	TRAIN ACC.(%)	TEST ACC.(%)
KNN	69.24	63.24
WEIGHTED KNN	99.88	63.17
LOGISTIC REG.	71.76	66.19
SVM RBF	82.81	67.80
NEURAL NETWORK	75.27	67.62

Considering the Table 2. above, we have reached our highest accuracy score by using SVM with RBF kernel. However, compared to the baseline models, our final models still overfit the training data on a certain level. While Logistic Regression handles overfitting better due to its regularization through L1 and L2, SVM is affected the most despite the regularization with C.

Both our kNN and Weighted kNN models are outperformed by Logistic Regression, SVM and Neural Network models regardless of the chosen features as expected. The reason behind of the issue is that Nearest Neighbors algorithm treats vectors as inputs which makes the method work poorly in high dimensions.

In order to improve kNN performances, we implemented two approaches. The first approach was to apply Principal Component Analysis, short for PCA. For KNN with PCA, we found that first 3 principal component can only explain 26.17% variance, which is too low for PCA to have a good performance, and such that kNN-PCA obtains only 46.60% test accuracy. Due to the low variance on smaller dimensions, the test accuracy rates of PCA algorithms weren't sufficient enough to perform better than the actual kNN baseline results. As for the second approach, through model and feature selection with cross validation, we found that our kNN model offers the best results with 189 selected features from the set of MFCC and Spectral Contrast statistics when k = 25. Further reduction in the feature dimension decreases the test accuracy as additional necessary features are removed.

Another expected result was the 99.88% training accuracy in wkNN. Since we took k = 25 as for the number of neighbors, which controls the model flexibility, the distance parameter used as weight allowed a better learning in the training set. However, this situation led to overfit.

Support vector machine with linear kernel performs a classification with a linear decision boundary. The reason we chose SVM with linear kernel as our baseline model was to observe whether the data we use is linearly separable or not. However, we had troubles in seizing the non-linearities of the data. When facing an input set of high dimensions, we came to a conclusion that we can improve our decision boundary by using a radial basis kernel which resulted in outperforming all the other models.

During the model selection of SVM with RBF kernel, C and gamma were the parameters to tune with. C is used for regularization that trades off between training and test accuracies. The parameter gamma on the other hand determines the amount of significance a training example has. Thus, when we used a high value of C, our model aimed to correctly classify all training examples which resulted in improved training accuracy score, consequently led us to overfitting. Choosing a small value of C to the contrary, created a smoother decision surface, however, misclassified examples occurred more often which resulted in lower accuracy ratings in both training and test scores. Therefore, as a result of our selections using cross validation, we chose the best C = 1.2 and gamma = 1e-1 (0.1) working on a feature set of 196 dimensions that is composed of MFCC, Spectral Contrast and Spectral Centroid.

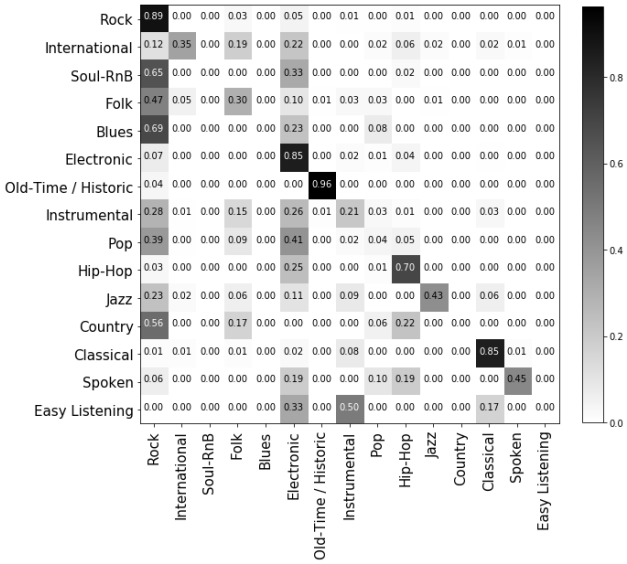

Figure 3. Confusion Matrix of SVM with RBF kernel

As we have an unbalanced data, we were expecting our normalized confusion matrix as appeared in Figure 3. As we observed the matrix, we figured out that since we were trying to preserve the percentage of the population we also miss-predicted most of the minority genres. The same problem has also occurred on the other studies that are done using FMA dataset. As a future work we consider both using a balanced dataset and building genre-specific models for minority genres.

Logistic Regression on the other hand outperformed all the methods we tested with respect to baseline model scores. Using Logistic Regression with one-vs-rest scheme is one of the reasons behind as it is a powerful choice compared to kNN and SVM with linear kernel in high dimensional

inputs considering the fact that we used all the 518 features as inputs. However, after our tests on SVM with one-versus-one scheme we came to a conclusion that one-versus-one approach achieves better modelling on our dataset, even though it is relatively slower than OvR due to complexity of $O(n)^2$ where n is the number of classes.

Table 3. Accuracy scores of Logistic Reg. approaches

APPROACHES	TRAIN ACC.(%)	TEST ACC.(%)
ONE-VS-ONE	71.68	66.19
MULTINOMIAL	70.36	65.70
ONE-VS-REST	70.84	65.96

The results on the Table 3. above shows the highest test accuracy scores along with their training accuracies obtained by using different Logistic Regression approaches. When we delve further into tuning with respect to parameters and features by applying cross validation, we found that a preselected feature set of 329 dimensions concluded the highest test accuracy scores for multinomial approach and OvR scheme. The combination of all features yields the highest test accuracy scores for one-vs-one scheme when L1 regularization is used. Thus, we continued our model selection with respect to the chosen features for each approach. The highest score we obtained by using multinomial approach was 65.70% through the solver sag which uses Stochastic Average Gradient descent and L2 for regularization. On the other hand, the test accuracy score we obtained by using one-vs-one scheme, 66.19%, through the solver liblinear which uses coordinate descent with L1 regularization performed better compared to both OvR and multinomial approach on a small scale.

Since our dataset is assembled with stratified sampling, and multinomial approach covers the entire probability distribution, we observed slightly higher accuracy scores on small and average size genres compared to both one-vs-one and OvR scheme. However, for the same reason, multinomial approach misclassified more often on large size genres. As for the regularization, L1 shrinks the less significant input features' weights closer to zero which leaves the majority of the features insignificant. Due to the high dimensional input features we have, choosing L1 regularization with one-vs-one scheme over L2 regularization with OvR scheme resulted in better test accuracy as our learning was slightly higher with L1 regularization. Thus, we came to a conclusion that model selection tuned with one-vs-one scheme, liblinear optimizer, C = 0.072 and L1 regularization result the highest accuracy scores for Logistic Regression.

As for our last model, we built a Neural Network Classifier. Tests are made on three log-loss function optimizers that are lbfgs, sgd and adam. While sgd refers to Stochastic Gradient Descent, adam is a Stochastic Gradient Descent based optimizer which is proposed by (Kingma & Ba, 2014). As mentioned in 4.1.2 lbfgs performs inefficient on large datasets. On the contrary, Adam is designed to perform well on large datasets. From our observations, it converges to a solution rather faster compared to other optimizers and methods.

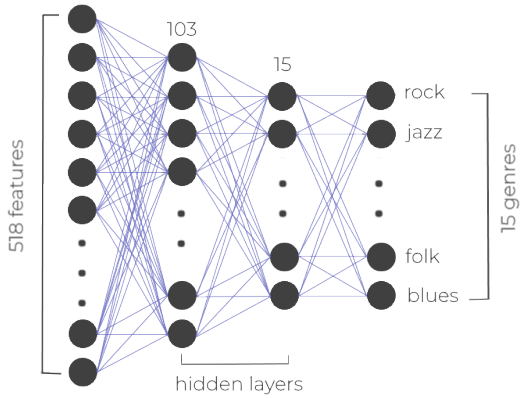

Figure 4. Neural Network

Training network with Adam as the optimizer and relu as the activation function for hidden layers, the highest accuracy 67.62% is reached for Neural Network approach. Other parameters such as hidden layer sizes and neuron counts (a hidden layer with 103 neurons and a fully connected layer), initial learning rate (0.0008996), alpha (3e-05) and beta (0.99) are tuned using 5 fold cross validation where convergence is obtained in 5 epochs. Figure 4. visualizes a representation of the Neural Network model.

While adam and relu together have shown to be successful, another approach is conducted using sgd as the optimizer and logistic as the activation function for hidden layers. Trained network performed an accuracy rate of 67.23% on 290 epochs. As for the hidden layers, first hidden layer is composed of 250 neurons and a fully connected layer forms the second hidden layer. Since sgd does not support an optimization as adam, converging takes more epochs.

Using all features as input for training results better compared to lesser feature sets since neural network can model large amounts of information. Although the network has 518 neurons as for input layer, having too many hidden layers causes model to overfit the training data. To avoid overfitting, alpha parameter, which is L2 regularization, is tuned. As we compare the results between Neural Network and our highest achieved by SVM with RBF kernel, even though slightly lower results are obtained, the difference between training accuracies indicates that regularization plays an important role in the performance of Neural Network.

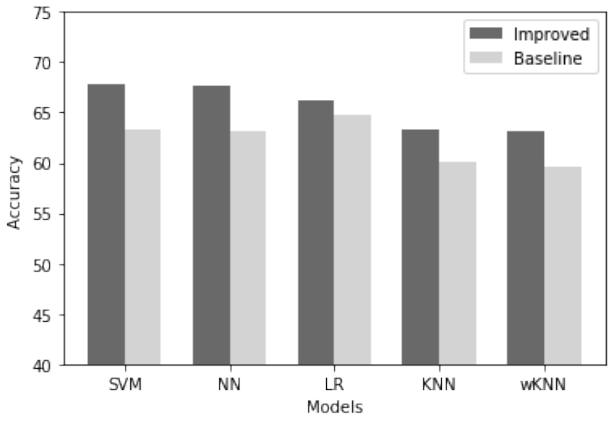

Figure 5. Comparison of Baseline and Improved Models

For our baseline models, having high dimensional input features consequently caused overfitting as collinearity between different features occurred. Thus, considering the Figure 5. above, our model and feature selection along with regularization have improved the test accuracy scores of each method.

6. Conclusions and Future Work

In this work, we implemented and discussed various machine learning models including Nearest Neighbors Classifier, Support Vector Machines, Logistic Regression and Neural Network to recognize music genres using FMA dataset. A specific model selection along with a feature selection for each method was involved in order to minimize the overfitting and the complexities of models, and we observed increase in test accuracy scores for each model. Considering the fact that the test accuracy score of the baseline kNN is 60.14%, we have concluded an improvement of 12.74% with respect to classification accuracy on the music genre recognition task. To advance the accuracy scores further, more balanced training data is definitely needed, and as an addition, a better detailed amount of sub-genres can be effective as well.

There is a number of extensions we are interested to pursue along the following:

(i) Representing audio tracks with Mel-scale spectrogram and modeling with Convolutional Neural Networks can be considered as a method for predicting music genres.

(ii) Implementing L1 and L2 regularization options in order to overcome the overfitting problem of SVM with RBF kernel.

(iii) Having a more balanced data by including new and up-to-date datasets can help achieving better generalized, real-world models and advanced solutions.

(iv) Ensemble models that is predicting each genre with the model which predicts the selected genre with the highest probability.

(v) A real life application: Machine learning algorithms can be applied the way we covered after the features of a music track with an unknown genre is extracted so that its genre can be predicted.

References

Bertin-mahieux, T., Ellis, D. P. W., Whitman, B., and Lamere, P. The million song dataset. In *In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR*, 2011.

Cano, P., Gómez Gutiérrez, E., Gouyon, F., Herrera Boyer, P., Koppenberger, M., Ong, B. S., Serra, X., Streich, S., and Wack, N. Ismir 2004 audio description contest. 2006.

Defferrard, M., Benzi, K., Vandergheynst, P., and Bresson, X. Fma: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference*, 2017. URL https://arxiv.org/abs/1612.01840.

Defferrard, M., Mohanty, S. P., Carroll, S. F., and Salathé, M. Learning to recognize musical genre from audio. In *WWW '18 Companion: The 2018 Web Conference Companion*, 2018. URL https://arxiv.org/abs/1803.05337.

Haggblade, M., Hong, Y., and Kao, K. Music genre classification. 2011.

Kim, J., Won, M., Serra, X., and Liem, C. Transfer learning of artist group factors to musical genre classification. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pp. 1929–1934. International World Wide Web Conferences Steering Committee, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Li, T. L., Chan, A. B., and Chun, A. Automatic musical pattern feature extraction using convolutional neural network. In *Proc. Int. Conf. Data Mining and Applications*. sn, 2010.

McFee, B., McVicar, M., Balke, S., Thom, C., Lostanlen, V., Raffel, C., Lee, D., Nieto, O., Battenberg, E., Ellis, D., Yamamoto, R., Moore, J., WZY, Bittner, R., Choi, K., Friesch, P., Stter, F.-R., Vollrath, M., Kumar, S., nehz, Waloschek, S., Seth, Naktinis, R., Repetto, D., Hawthorne, C. F., Carr, C., Santos, J. F., JackieWu, Erik, and Holovaty, A. librosa/librosa: 0.6.2,

August 2018. URL https://doi.org/10.5281/zenodo.1342708.

Panagakis, Y., Kotropoulos, C., and Arce, G. R. Music genre classification via sparse representations of auditory temporal modulations. *2009 17th European Signal Processing Conference*, pp. 1–5, 2009.

Sanden, C. and Zhang, J. Z. Enhancing multi-label music genre classification through ensemble techniques. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 705–714. ACM, 2011.

Shao, X., Xu, C., and Kankanhalli, M. S. Unsupervised classification of music genre using hidden markov model. In *ICME*, volume 4, pp. 2023–2026. Citeseer, 2004.

Tsai, W.-H. and Bao, D.-F. Clustering music recordings based on genres. In *Information Science and Applications (ICISA), 2010 International Conference on*, pp. 1–5. IEEE, 2010.

Tzanetakis, G. and Cook, P. Musical genre classification of audio signals. volume 10, pp. 293–302, July 2002. doi: 10.1109/TSA.2002.800560.

Xu, C., Maddage, N. C., Shao, X., Cao, F., and Tian, Q. Musical genre classification using support vector machines. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 5, pp. V–429. IEEE, 2003.