Music Genre Classification

John Dinh
University of California, Davis
jndinh@ucdavis.edu

Collin Kennedy
University of California, Davis
cjkennedy@ucdavis.edu

Abstract

The $GTZAN$ dataset is a popular dataset for machine learning research, signal processing, and music genre recognition. Popular machine learning classification algorithms such as neural networks, random forests, and support vector classifiers, to name a few, perform quite well on this classical dataset. Prior to fitting the models, audio processing in the time domain as well as the frequency domain were used to extract features from the $GTZAN$ using classical techniques such as windowing functions and Fourier transforms. After tuning our proposed models, it was found that XGBoost and a Sequential neural network with dropout regularization performed the best on the dataset, achieving high accuracy scores 90.05% and 90.04% respectively.

1 Introduction

Aside from the name of the band or artist, perhaps the most well-known attribute of a song is its *genre*, a simple yet powerful label that conveys an unbelievable amount of information, ranging from its tempo and rhythmic structure to its lyrical and pitch content.

In 2002, researchers George Tzanetakis and Perry Cook published "Musical Genre Classification of Audio Signals" in *IEEE Transactions On Speech and Audio*, where they described their work applying machine learning methods to automatically classify audio by genre. They achieved a modest accuracy rating of about 61% using a Gaussian Mixture Model to classify songs into 10 different genres, which the authors noted as being comparable to results obtained by human genre classification [14]

For the purposes of our project, we sought to improve upon the classification accuracy demonstrated by Tzanetakis and Cook by implementing a handful of different machine learning methods to solve this classification problem. More specifically, we focus on four different classification methods, including a Naive Bayes Classifier, Random Forest, Gradient Boosted Trees (XGBoost), and a Deep Neural Network.

Our motivation is two-fold. First, we hope to gain greater familiarity with ensemble learning and deep learning methods. Gradient Boosting and neural networks have become increasingly common in recent years, especially in classification settings, and it is important to be aware of both their effectiveness as well as any drawbacks, downsides, or caveats they pose or present to an analysis.

Music genre classification also felt like a unique yet still plenty-interesting sandbox to use to evaluate these machine learning models. And, music genre classification is very much a topic of concern to machine learning engineers, data scientists, statisticians, and the like. For example, music streaming platforms and services, like Apple Music and Spotify, are using machine learning to generate recommendations for listeners. In 2018, 31% of listening activity on Spotify was represented by machine-generated playlists [7]. Music genre classification clearly has role to play in the world of machine learning beyond the classroom.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

Our report is organized as follows: First, we introduce the *GTZAN* data. We then provide an overview of each of the machine learning models and algorithms we employ in the analysis, followed by a presentation of our results and a discussion of their implications and any potential caveats.

2 Methodology & Analysis

2.1 Data

Having appeared in well over 100 published works, the *GTZAN* dataset is one of the most frequently-used public datasets in machine learning research, especially for music genre recognition and classification. The data was originally analyzed in George Tzanetakis and Perry Cook's research.

The dataset consists of 1,000 songs, with 100 songs each belonging to one of 10 different genres: blues, classical, country, disco, hip-hop, jazz, metal, reggae, and rock. Table 1 provides an overview:

Table 1: **Summary of GTZAN Data**

Music Genre	Number of Songs
Blues	100
Classical	100
Country	100
Disco	100
Hip-Hop	100
Jazz	100
Metal	100
Pop	100
Reggae	100
Rock	100

While the data has been made publically available at MARSYAS (Music Analysis, Retrieval, and Synthesis for Audio Signals), an open-source software framework and database built by Tzanetakis himself, we retrieved a version of the data from Kaggle [1]. In total, there are 58 features, all of which are used for the purposes of our analysis. There are essentially 6 different types of features, a summary of which can be found in Table 2 below:

Table 2: **Feature Types**

Feature Type	Description
Root Mean Squared Energy	quantifies how loud audio is
Zero Crossing Rate	rate at which a signal changes from positive to zero to negative or vice versa
Mel-Frequency Cepstral Coefficient	describes the contour of an audio signal
Chroma Features	represent the melodic and harmonic attributes of a song
Spectral Centroid	describes the audio signal's 'center of mass'
Spectral Roll-off	approximately the maximum frequency

2.2 Feature Extraction

In audio processing, feature extraction can shortly be summarized as the process of highlighting the most impactful feature of a signal. Analysis of these signals used to be done either in the time domain, or the frequency domain. However, joint time-frequency domain techniques [12] have been developed and will also be one of the methods of extracting features to use in our models. The Python library, librosa, makes extraction of these features from our music data set very convenient.

In the time domain, the zero-crossing rate (ZCR) is the rate at which a signal changes sign, and is an effective way to detect whether a section of audio is voiced, unvoiced, or silent. Intuitively, ZCRs would be zero during silent segments. The ZCR is also an essential tool for estimating the lowest frequency of a periodic waveform, and thus, being a useful feature for classification systems [11].

Another feature in the time domain is the energy, or total magnitude of a signal. One thing to note is that sound signals are non-stationary time series, which simply put, means that the statistical properties of the signal change over time. One of the methods of analysis of energy is to look at the root mean square energy (RMSE). Since the signal is non-stationary, it is rather difficult characterize the overall energy of the signal, but the RMSE, similar to the short term energy (STE) of a signal, divides up the entire audio signal into smaller time chunks and gives a energy value based on the duration of the time series. Given some discrete time signal, $x(n)$, the RMSE is defined as

$$\sqrt{\frac{1}{N}\sum_{n}|x(n)|^2}$$

In the frequency domain, the time-domain signal is converted to the frequency-domain using techniques such as the Fourier transform. The Mel-frequency cepstral coefficient (MFCC) is a feature in this domain. To first calculate the coefficient, the signal is divided into smaller temporal fragments, usually 20-30ms, also known as frames. Next a technique called windowing is done, where the borders of the signals are smoothed and the change in signal statistics are minimized. [2]. To convert these temporal fragements into the frequency domain, the Fast Fourier transform (FFT) is performed. The FFT is a technique that is used to compress the speech signal without losing any relevant information. The magnitude of the frequency is then multiplied by a set band pass filters in order to obtain a smooth magnitude spectrum. This output is then fed into a discrete cosine transform (DCT) to obtain the Mel coefficients [2].

The chroma features extracted from the signals is a representation of spectral energy of a signal. In the librosa module, this is given by a 12-element representation, called the chroma vector, where each of the 12 elements represent a pitch class using equal western semitone spacing. Figure 1 illustrates how these features can vary over the duration of the signal.

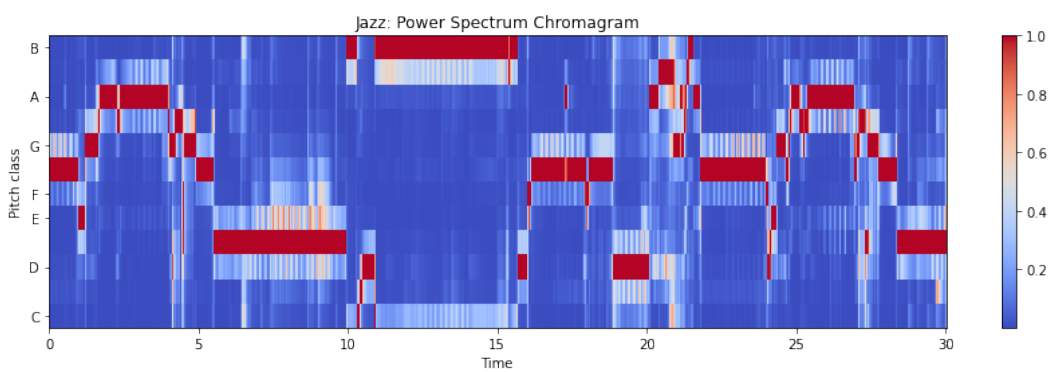

Figure 1: Chromagram using power spectrum of 30 second jazz audio file

Finally, to look at the joint domain features, the spectral centroid and roll-off of a signal are observed. The spectral centroid measures the 'center of mass' of a signal, and can be interpreted as the 'brightness' of a signal [10]. Mathematically, the centroid of a frame is the average frequency weighted by the amplitude divided by the sum of amplitudes. For a frequency $f(n)$ at frame n and amplitude $x(n)$, the spectral centroid is defined as:

$$\text{Centroid} = \frac{\sum\limits_{n=0}^{N-1} f(n)x(n)}{\sum\limits_{n=0}^{N-1} x(n)}$$

Finally, the spectral roll-off is N^{th} percentile frequency of the spectral distribution. Usually, N is taken to be 85% or 95% and can be interpreted as the skewness of the spectral shape and is used distinguish voiced from unvoiced signals[8]. Mathematically, for $N = 85$, where $M_t[n]$ is the magnitude of the Fourier transform for a frame t at window n, it is the frequency R_t such that:

$$\sum_{n=1}^{R_t} M_t[n] = .85 \sum_{n=1}^{N} M_t[n]$$

2.3 Algorithms

2.3.1 Naive Bayes

While this simple supervised classification technique has gone by many different names over the years, including but not limited to *idiot Bayes*, *simple Bayes*, and *independence Bayes*, they all capture perhaps the most important aspect of the Naive Bayes classifier, that it is incredibly straightforward.

Given some feature vector $\mathbf{X} = (x_1, x_2, ..., x_n)$ and some class variable c, Bayes theorem can be expressed as

$$P(c|\mathbf{X}) = \frac{P(\mathbf{X}|c)P(c)}{P(\mathbf{X})} \tag{1}$$

Noting the critical assumption of independence, this can be re-written as

$$P(c|\mathbf{X}) = \frac{P(x_1|c)P(x_2|c)...P(x_n|c)P(c)}{P(x_1)P(x_2)...P(x_n)} \tag{2}$$

Making use of proportionality and simplifying further, the Naive Bayes model classifies an observation by assigning it to the class that maximizes the posterior probability, s.t.

$$c = \arg\max_{c}\ P(c) \prod_{i=1}^{n} P(x_i|c) \tag{3}$$

Now, just because it is 'naive' doesn't mean it is necessarily a poor classifier, and there is plenty of empirical evidence that suggests it can perform quite well, even when the independence assumption is blatantly violated, which is often the case in applied, high-dimensional settings [6]

This comes down to the fact that the Naive Bayes model requires that fewer parameters be estimated compared to more complex models that attempt to capture the effect(s) of interactions between distributions. So in the case of large sample sizes, the Naive model will generally have lower variance for the estimates of the posterior probability, $P(c|\mathbf{X})$ [3]

Because of both its simplicity and its robustness in spite of such simplicity, we figured it would serve as a good baseline in our comparison of different machine learning methods for music genre classification.

2.3.2 Random Forest

The Random Forest is an ensemble method that overcomes the weaknesses of decision trees by making two noteworthy modifications. While decision trees generally have very low bias and high variance as they tend to overfit the training data, the Random Forest trains numerous decision trees in parallel on *bootstrapped* data using a random subset of features, and then *aggregates* the predictions of each of the individual trees to produce its own prediction. In the context of classification, this is done by taking the mode of the individual trees' predictions [9].

Figure 2 helps illustrate the process:

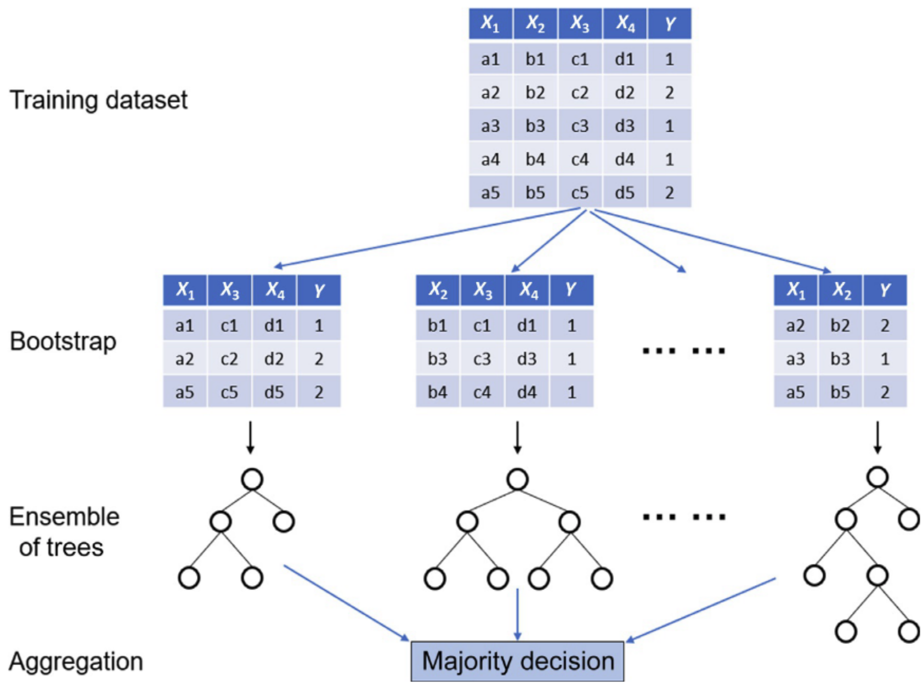

Figure 2: Random Forest Example

2.3.3 XGBoost

Extreme gradient boosting (XGBoost), is a regularized, sampling based implementation of gradient boosted trees [5]. Usually, gradient boosting is done for regression and classification and is an ensemble learning technique that produces a collection of weak learners. The model is built in a stage-wise fashion similar to other boosting methods, and is generalized to allow any differentiable loss function to optimize the model. More concretely, function estimation is done with gradient descent using some loss function, $\mathcal{L}$ in the space of functions, that is:

$$\hat{f} = \text{argmin}_f \mathcal{L}(f)$$

To transition to gradient boosted *trees*, it is asserted that the collection of weak learners is a decision tree. For some region in the predictor space R_{jm} and predicted output w_{jm}, an observation x can be modeled as:

$$F(x) = \sum_{j=1}^{J_m} w_{jm} \mathcal{I}(x \in R_{jm})$$

To incorporate this model into gradient boosting, good regions of R_{jm} are found using regular decision tree learning on the residuals, and then the weights of each leaf node are iteratively updated.

That is, for some loss function ℓ, the optimal weight at node jm, w_{jm}, is updated using the previous estimated function f_{m-1} as:

$$\hat{w}_{jm} = \text{argmin} \sum_{x_i \in R_{jm}} \ell(y_i, f_{m-1}(x_i) + w)$$

Finally, XGBoost improves this framework by adding some more restrictions. One improvement is that tree complexity is regularized in order to prevent extremely deep trees and over-fitting. Similarly to random forests, XGBoost samples predictors at internal nodes in the trees but has an advantage over RF due to additional regularization. Mathematically, XGBoost optimizes the following objective function:

$$\sum_{i=1}^{N} \ell(y_i, f(x_i)) + \gamma J + \frac{1}{2}\lambda \sum_{j=1}^{J} w_j^2$$

Above, $\lambda, \gamma \geq 0$ are regularization coefficients, and J is the number of leaves. Above all, XGBoost also provides some advantages in terms of computing speed and efficiency to ensure scalability. [4]

2.3.4 Neural Network

The architecture of the our Sequential neural network is an input layer, 4 hidden layers with dropout regularization, and a soft-max dense layer output. The sequential model is the basic building blocks of more complicated deep learning models and is a fundamental architecture to be aware of. The trainable number of parameters at each layer is $60416, 524800, 131328, 32896, 8256$, and 650. Each hidden layer has a drop out ratio of $.3 - .5$ to ensure the model does not over-fit. The model was compiled with the Adam optimizer and since genre classification is a multi-class problem, the loss function used was the sparse categorical cross entropy loss. Sparse categorical cross entropy loss implies that the labels are coded as integers instead of one-hot vectors, which also reduces time to train the model. Each layer used the ReLU activation function, and the final layer uses the soft-max activation function to predict class membership.

3 Results

3.1 Popular Algorithms

Fitting the algorithms described in Section 2.3 reveal that the testing accuracy on these algorithms are the highest among classical machine learning and statistical models (Table 3). Naive Bayes, which we will consider as the baseline model, performed as well as a random guess considering overall accuracy across all classes. The sequential neural network and XGBoost performed quite well with a

Table 3: Testing accuracy of popular classification methods

Algorithm Name	**Testing Accuracy**
Naive Bayes	51.00%
Stochastic Gradient Descent Classifier	65.30%
Decision Trees	57.42%
Random Forest	89.42%
Support Vector Classifier	88.49%
Elastic Net Logistic Regression	71.71%
Multi-Layer Perceptron	81.82%
XGBoost	**90.05%**
Sequential Neural Network	90.04%

testing accuracy of more than 90% for both models. An interesting and expected result is how much better XGBoost performs than decision trees. In addition, with additional tuning, the classic support vector classifier also performs similarly to the random forest, as their testing accuracies are within a percent.

3.2 Music Genre Classification Assessment

Considering that XGBoost performed the best among all algorithms, beating the SNN by a single basis point, further assessment about the predictive ability of the model is done using a confusion matrix in Figure 3. The precision for each genre, except country, jazz, and rock are above 90% and the recall for every genre except rock is above 90%. This implies that rock may be more difficult to classify and future experiements can be done with fewer classes to understand why rock has the most poor classification rate, with an accuracy of only 81.52%

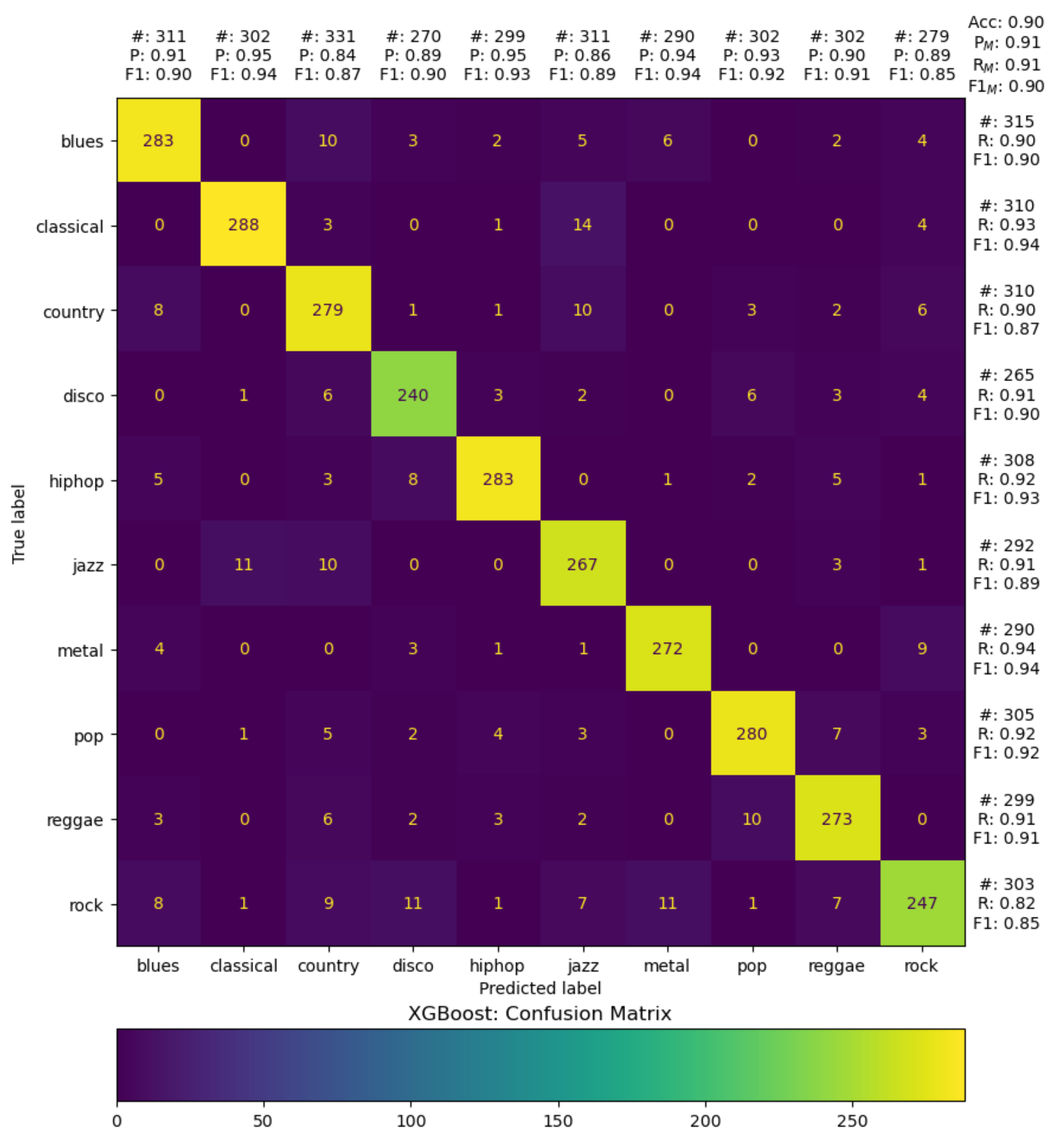

Figure 3: Confusion Matrix: XGBoost

4 Conclusion

Of the nine machine learning algorithms we considered for this music genre classification problem, the random forest, gradient-boosted trees, and the sequential neural network all performed exceptionally well, with the gradient-boosted trees beating the SNN by a single basis point ($90.05\% > 90.04\%$).

While the *GTZAN* dataset has become the *MNIST* dataset of sound, so to speak, that isn't to say that it is not without its faults. For example, in 2012, Bob Sturm published "An Analysis of the *GTZAN* Music Genre Dataset"[13]. He describes how, despite the fact that the *GTZAN* dataset has become a "benchmark" for music genre recognition, there is little evidence that many of the authors who cite the data in their research have even listened to any of the audio files, and that he is able to catalog a multitude of incorrect labelings, replicas, and distortions in the dataset. More specifically, he notes that about 10.6% of the dataset is mislabelled [13]. With that said, since all music genre classification research that has used *GTZAN* to train machine learning models have also had to face the same issues with the data, the results are comparable, and for our purposes, sufficient.

Looking forward, there is still plenty of room for improvement. While all of the features in the dataset were produced using signal processing techniques, an analysis that also considers certain types of metadata, like the song title, album, or lyrics could enable or further improve classification results. After two decades, perhaps its time for researchers to move on from the *GTZAN* dataset.

References

[1] 2022. URL: https://www.kaggle.com/code/andradaolteanu/work-w-audio-data-visualise-classify-recommend/data) (visited on 05/21/2022).

[2] Shalbbya Ali et al. "Mel Frequency Cepstral coefficient: A Review". In: *Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development, ICIDSSD 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India* (2021). DOI: 10.4108/eai.27-2-2020.2303173.

[3] Patricia Altham. "Improving the Precision of Estimation by Fitting a Model". In: *Journal of the Royal Statistical Society Series B (Statistical Methodology)* 46 (Sept. 1984), pp. 118–119. DOI: 10.1111/j.2517-6161.1984.tb01283.x.

[4] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *CoRR* abs/1603.02754 (2016). arXiv: 1603.02754. URL: http://arxiv.org/abs/1603.02754.

[5] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." In: *The Annals of Statistics* 29.5 (2001). DOI: 10.1214/aos/1013203451.

[6] David Hand and Keming Yu. "Idiot's Bayes: Not So Stupid after All?" In: *International Statistical Review* 69 (May 2007), pp. 385 –398. DOI: 10.1111/j.1751-5823.2001.tb00465.x.

[7] *How Spotify Beat Apple, Amazon, and Google Using Machine Learning*. 2018. URL: https://digital.hbs.edu/platform-rctom/submission/how-spotify-beat-apple-amazon-and-google-using-machine-learning/#_edn5 (visited on 05/28/2022).

[8] Jinliang Liu, Changhui Wang, and Lijuan Zha. "A middle-level learning feature interaction method with deep learning for multi-feature music genre classification". In: *Electronics* 10.18 (2021), p. 2206. DOI: 10.3390/electronics10182206.

[9] Siddharth Misra and Hao Li. "Noninvasive fracture characterization based on the classification of sonic wave travel times". In: *Statistical Shape and Deformation Analysis* (2017), 243–287. DOI: https://doi.org/10.1016/B978-0-12-817736-5.00009-0.

[10] Unjung Nam. *Special area exam part II, April 28, 2001 Unjung Nam - CCRMA*. URL: https://ccrma.stanford.edu/~unjung/AIR/areaExam.pdf.

[11] C. Panagiotakis and G. Tziritas. "A speech/music discriminator based on RMS and zero-crossings". In: *IEEE Transactions on Multimedia* 7.1 (2005), pp. 155–166. DOI: 10.1109/TMM.2004.840604.

[12] Shie Qian and Dapang Chen. "Joint time-frequency analysis". In: *IEEE Signal Processing Magazine* 16.2 (1999), pp. 52–67. ISSN: 1558-0792. DOI: 10.1109/79.752051.

[13] Bob L. Sturm. “The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use”. In: *CoRR* abs/1306.1461 (2013). arXiv: 1306.1461. URL: http://arxiv.org/abs/1306.1461.

[14] George Tzanetakis and Perry Cook. “"Musical Genre Classification of Audio Signals"”. In: *IEEE Transactions On Speech and Audio* 10.5 (2002), pp. 379–420.