Music Genre Classification

Bibit Bianchini and Thomas Froger Silva
{bibit, tfrogers} @stanford.edu

Abstract

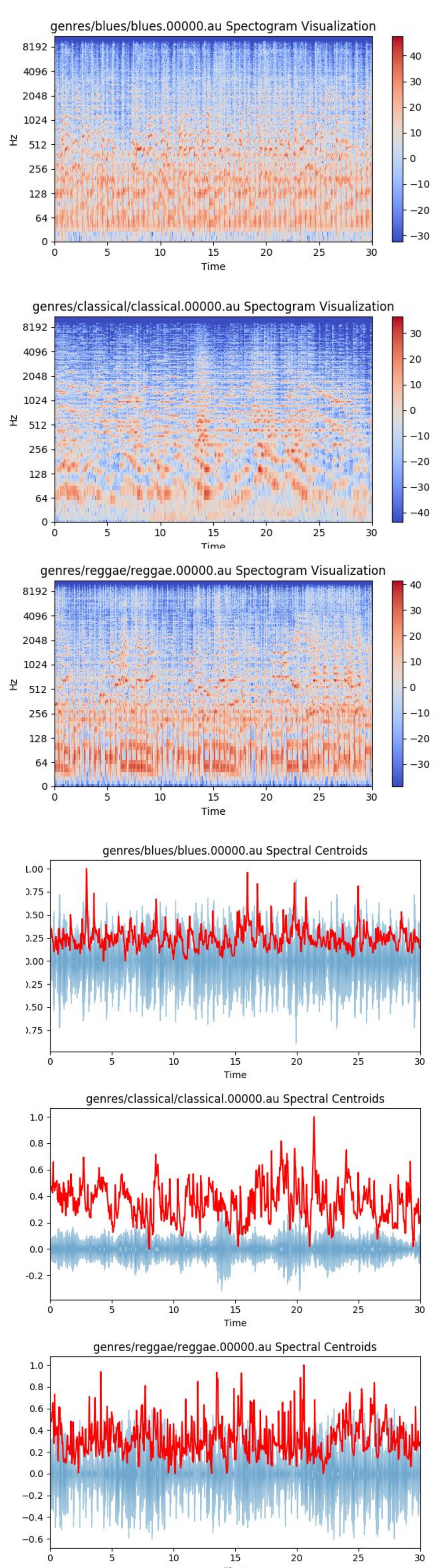
This paper presents different ways to extract features from song and different learning algorithms that allow to classify songs by there genre. The features we used were Zero Crossings, Spectral Centroid, Spectral Roll-off, Mel-Frequency Cepstral Coefficients and Chroma Frequencies. For the learning algorithms we use support vector machines (using linear approach with batch gradient descent and stochastic gradient descent). For our data set, we used 1000, 30 sec excerpts of songs classified in 10 different genres to train and test our algorithm. Despite the small data set, we managed to get a general 26% accuracy on our test set and up to 80% accuracy for recognising classical music.

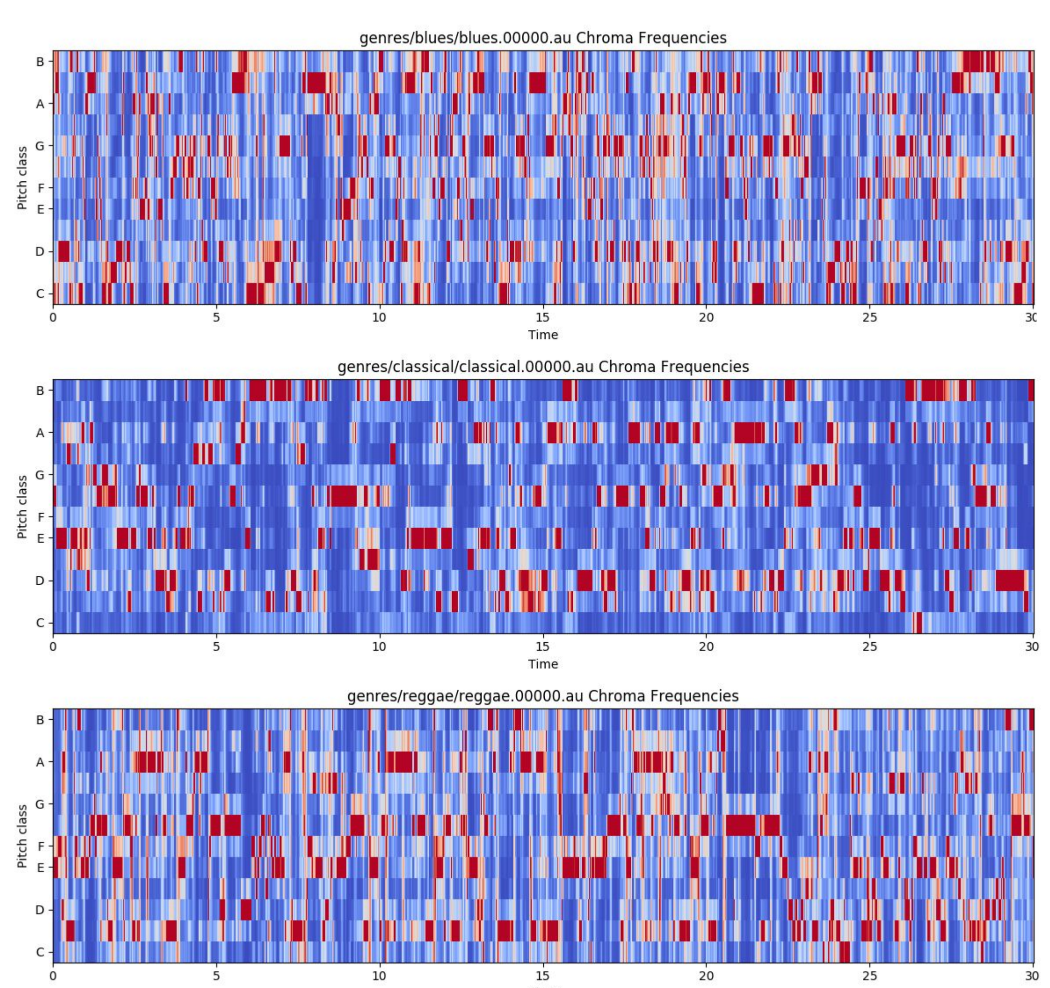
Dataset + Feature Extraction

The GTZAN dataset has 1000 30-second audio snippets across 10 different genres: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock.

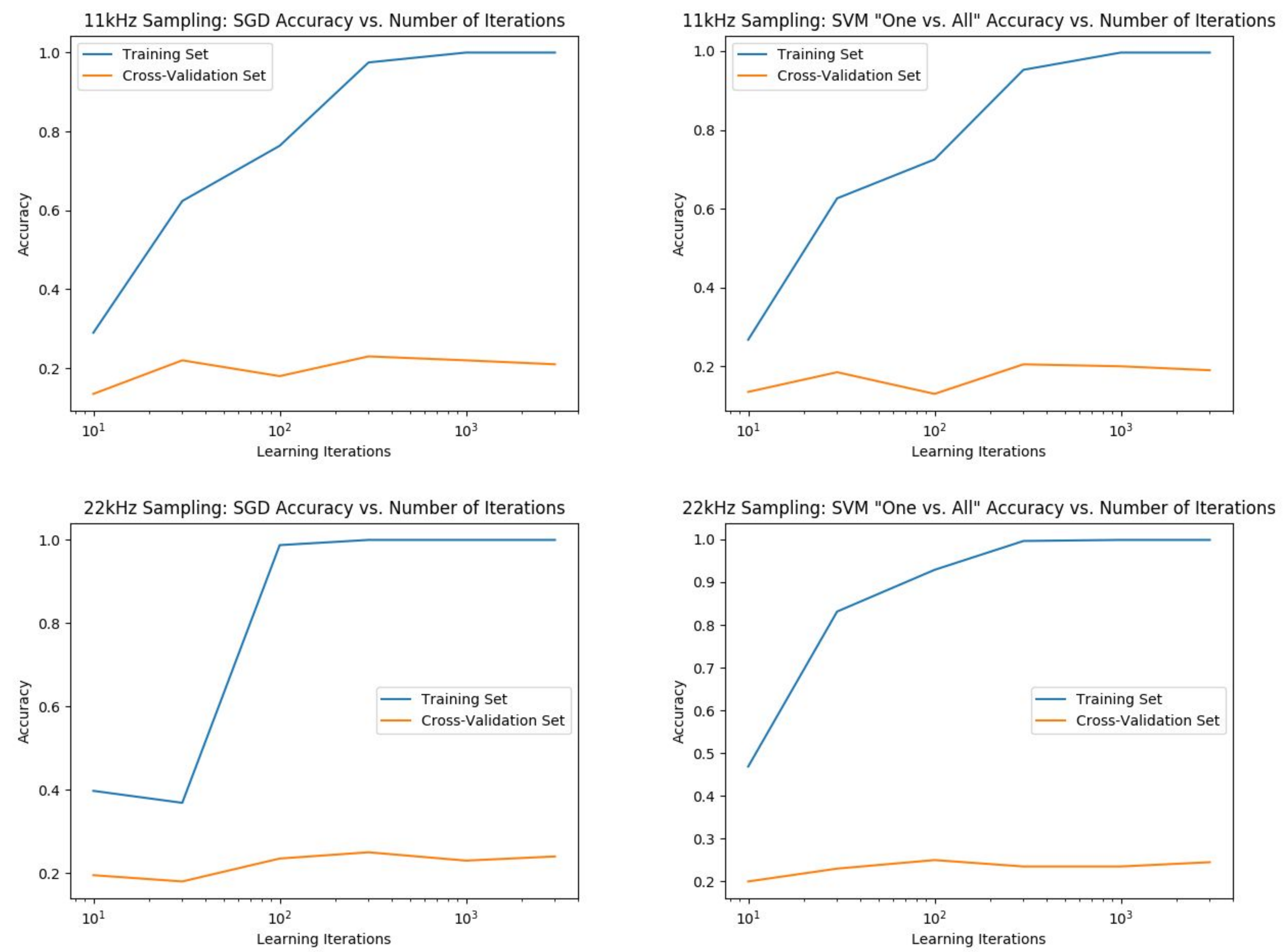
Feature extraction:

- Zero Crossings
- Spectral Centroids
- Spectral Roll-off
- Mel-Frequency Cepstral Coefficients (MFCC)
- Chroma Frequencies



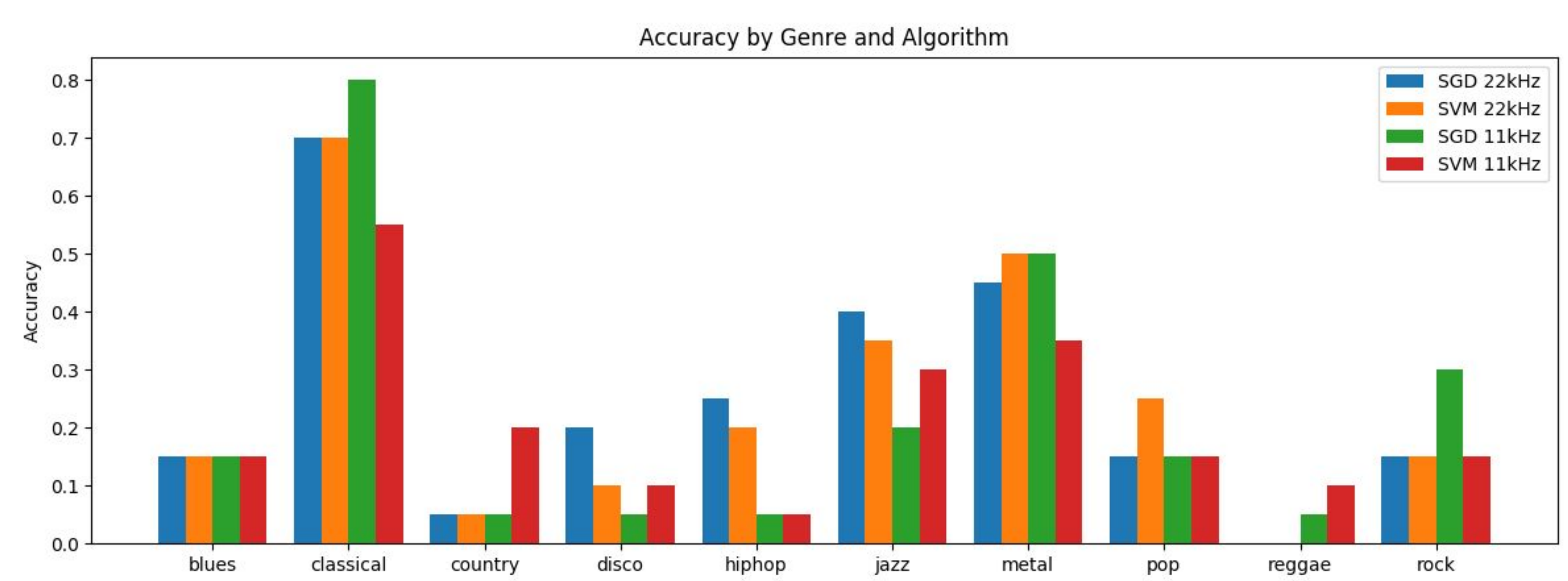


Learning Curves



Cross-validation sets were used to determine the number of iterations to train our models as well as the data sampling rate for feature generation. We received similar results across all combinations of sampling rates and machine learning models in that 300 iterations provided the best cross-validation accuracy.

Test Set Accuracies



Classical was the clear outlier in that it was easiest to identify. Jazz and metal were also more distinct from the other genres. Reggae was the most difficult genre for our models to identify, frequently misclassified as blues or jazz.

Model / Accuracy	Training Set	Cross-Validation Set	Test Set
SVM @ 22 kHz sampling rate	100%	22.5%	24.8%
SGD @ 22 kHz sampling rate	100%	24.5%	26.3%

Learning Algorithms

We used support vector machines (SVM) with two different gradient descent methods. For the first algorithm we used a linear support vector machine and one-vs-the-rest scheme using batch gradient descent and for the other we used SMV with stochastic gradient descent (SGD).

For predictions, took the genre with the highest probability for each sample of the data set:

$$h_\theta^{(n)} = P(y = n \mid x;\ \theta)$$

$$prediction = max_i(h_\theta^{(i)}(x))$$

The linear hypothesis equation is:

$$h_\theta = \frac{1}{1+e^{-\theta^T x}}$$

The cost function used for linear SMV is:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))$$

For the Batch gradient descent update function we used:

$$\theta_j := \alpha * \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})\, x_j^{(i)}$$

For the stochastic gradient descent update function we used:

$$\theta := \theta - \alpha * (h_\theta(x^{(i)}) - y^{(i)})\, x_j^{(i)}$$

References

1. Tzanetakis, G., and P. Cook. “Musical Genre Classification of Audio Signals.” IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, 2002, pp. 293–302., doi:10.1109/tsa.2002.800560.
2. Pandey, Parul. “Music Genre Classification with Python.” Towards Data Science, 12 Dec. 2018, towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8.
3. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
4. Leben, Jakob. “Data Sets.” Music Analysis, Retrieval, and Synthesis for Audio Signals, 2015, marsyas.info/downloads/datasets.html
5. C. N. Silla Jr., A. L. Koerich and C. A. A. Kaestner, "Feature Selection in Automatic Music Genre Classification," 2008 Tenth IEEE International Symposium on Multimedia, Berkeley, CA, 2008, pp. 39-44. doi: 10.1109/ISM.2008.54