Automatic classification of instruments from supervised methods of machine learning

Rômulo Vieira[1, 2] , **João Teixeira Araújo**[1, 2] , **Edimilson Batista**[2] , **Flávio Schiavoni**[1, 2]

[1]Arts Lab in Interfaces, Computers, and Everything Else - ALICE

[2]Department of Computer Science - UFSJ

romulo_vieira96@yahoo.com.br, teixeira.araujo@gmail.com, edimilson.batista@gmail.com, fls@ufsj.edu.br

Abstract. *Sorting instruments is not an easy task for humans or computers, especially when it comes to elements with the same acoustic properties, such as wind, percussion, or strings. Nevertheless, the use of audio descriptors and artificial intelligence techniques can make this duty more accessible. In this paper, three supervised methods, Naive Bayes, decision tree and Support Vector Classifier (SVC) are used to categorize acoustic guitar and bass sounds in a database, using as a parameter the information extracted from audio descriptors. The research resulted in a performance comparison of these three algorithms, considering their hit rates and processing time when classifying samples in different parts of the dataset. After all, some relevant considerations about the feasibility of automatically classifying instruments are presented.*

1. Introduction

The constant technological advancement has enabled several changes in the area of computer music, which facilitates the creation of various works in digital format. Given the vast amount of generated files, there is a need to classify them based on their main characteristics. One way to do this is through metadata, where tags defined by experts and consumers inform the author, album, style, year of release, or record label of a sample.

However, this classification method presents some problems, such as high cost to hire professionals capable of performing this categorization properly; slow process, as it is manual; divergences in classifications used by different groups and users; preference for famous artists, which overshadows those more unknown; inability to fully capture the musical and cultural content of the tracks; and finally, the difficulty for the lay public to associate the true meaning of each tag [1].

Consequently, the automation of this process becomes an area of interest to provide better organization and recommendation of this content, especially when dealing with a very large database. In this context, the present paper uses supervised methods, audio descriptors, and a multi-classifier algorithm to separate the samples of a database between acoustic guitar and bass, based on the timbre of each one of them. Then, comparisons and analyzes are performed on the processing times of each method and the feasibility of automatically classifying instruments.

The remainder of the paper is structured as follows: Section 2 indicates the related works and the state of the art in classifying instruments by digital means; Section 3 presents a brief analysis of digital audio and the descriptors used, while Section 4 shows the methodology and applied tools in the classification. Section 5 displays the obtained results and a discussion about them. Finally, Section 6 brings summarized conclusions of this work.

2. Related Works

The classification of musical properties is a popular task among artificial intelligence researchers so that there is a multitude of papers dealing with this topic, using the most diverse techniques. A work that comes very close to what is proposed throughout this text is that of Lara Haidar-Ahmad [2], who applies a multi-classifier in an audio stream to classify the timbres into drums, piano, flute, or "other" when none of the above are identified. The audio is preprocessed to extract a Mel-spectrogram and the outputs determine the dominance or non-dominance of a given instrument in that analyzed audio track.

Other works use Support Vector Machines (SVMs) and Gaussian models to solve this problem, as can be seen in [3] and [4]. The first article classifies 8 instruments using two Gaussian methods and SVM, reaching an accuracy of 70%, while the second combined the Gaussian technique with a k-nn classifier to group instruments based on their features.

More recently, researchers have turned their attention to deep learning in performing this task, which removes the manual extraction of features from audio samples. From this, models emerged that can learn from the frequency spectrum of these same samples [5].

In the end, our literature research showed that most studies use a dataset with isolated sounds, consisting of the presence of a single instrument per sample. The same pattern was repeated in this work, with the difference that here, a much larger amount of samples was used to infer results. Another difference is that our work focus was on supervised methods, while the state of the art prioritizes unsupervised learning techniques.

3. Sound Analysis

Unlike analog audio, digital sound is not continuously represented, requiring conversion from one format to another.

This is possible thanks to a subfield of Computer Science, called signal processing, which performs the electronic manipulation of acoustic data received as input [6].

Another technology that helps in this process is the audio descriptor, an analytical tool that represents the characteristics of the musical signal in a dimensional curve. These descriptors reduce the complexity of information by focusing on specific aspects of the signal. They are still useful for creating a particular taxonomy of the content of the musical signal spectrum because they have a reductionist character. These attributes can be correlated with subjective perceptual properties related to timbre, such as “brightness”, “opacity” or even “smoothness” of the sound, whereas timbre is defined as a result of the combination of two components: the vibrations of the sound and the frequencies produced by these vibrations. It is important to say that each source produces sounds in different ways, which can be by fingering, percussion, breathing or electronic inference [7, 8].

In this paper, four descriptors will be used: Spectral Rolloff, Spectral Centroid, Spectral Flatness, and Mel-Frequency Cepstral Coefficient (MFCC). The first one extracts the sample’s rollover point, that is, the segments of the sound wave that are below a pre-defined percentage. The second model, as far as it is concerned, is responsible for indicating where the center of mass of the spectrum is, which represents the central tendency of the waveform. It has a connection with the impression of “brightness” of the sound, and its most common applications are aimed precisely at timbre classification.

The third descriptor is also known as Wiener Entropy. It is a measure used to quantify the appearance of noise as opposed to pitching. An example of how this tool works can be seen in Figure 1 [9, 10].

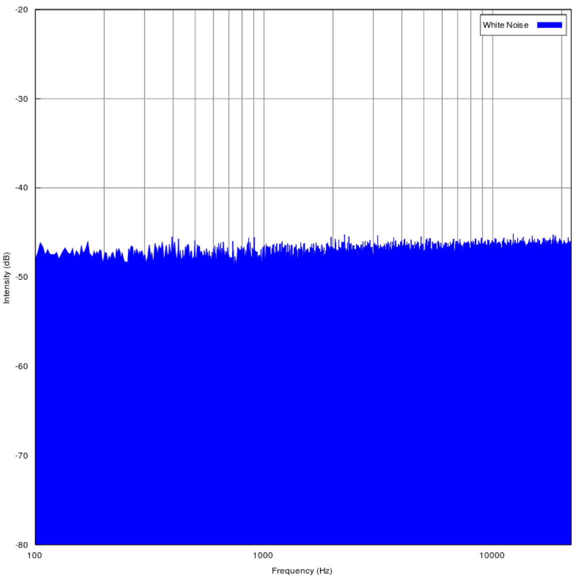

Figure 1: Flatness descriptor spectral curve [11].

The fourth and last descriptor is also the most important. It is a derivative of the representation of a non-linear audio spectrum, widely used in speech recognition systems, musical genre classification, and audio similarity measures. The fact that the MFCC is not robust to noise is also noteworthy, therefore, its values must be normalized. Figure 2 shows an audio sample after applying this descriptor [12].

4. Methodology

To carry out this study, we use some tools and libraries available for the Python programming language. One of these libraries was Librosa[1], developed for applications that involve some type of digital audio processing, to facilitate work involving the recovery of musical information. Among its advantages is familiarity with MATLAB, standardization of commands, and modular features, in addition to features that allow you to estimate the beats per minute in certain samples and the graphic representation of the sound in spectrograms. These fundamental properties can be observed in the Figure 3 [13].

Another library used to handle with audio was AudioLib[2]. Although less powerful than the Librosa, it provides a high quality system for reproducing sound data, its main feature being the ability to generate waveforms, which is essential for timbre differentiation.

Of the libraries for data classification, the chosen one was scikit-learn[3], as known as Sklearn, widely used to identify the category and continuous value of a given object, as well as automate the grouping of family values and extract and normalize features. It also allows integration with well-established technologies in the field of Artificial Intelligence, such as NumPy, Matplot, and Pandas [14].

The database that provided the information to be classified is Nsynth, considered a reference in the study of audio synthesis that presents large magnitudes, with about 20 GB and 305,979 musical notes, where each one has a unique timbre, pitch and envelope. These notes can be

[1] https://librosa.org/doc/latest/index.html
[2] https://pypi.org/project/AudioLib/
[3] https://scikit-learn.org/stable/index.html

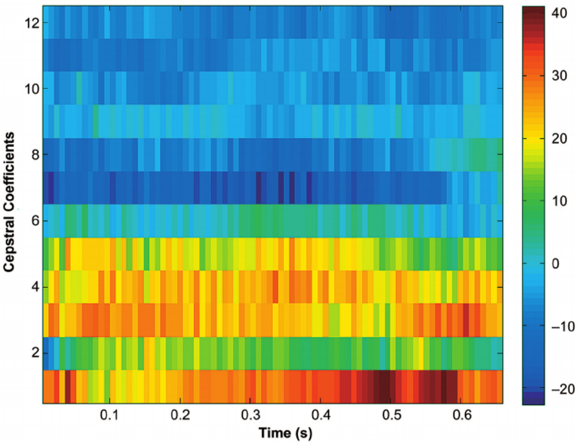

Figure 2: Mel-Frequency Cepstral Coefficient spectrogram [12].

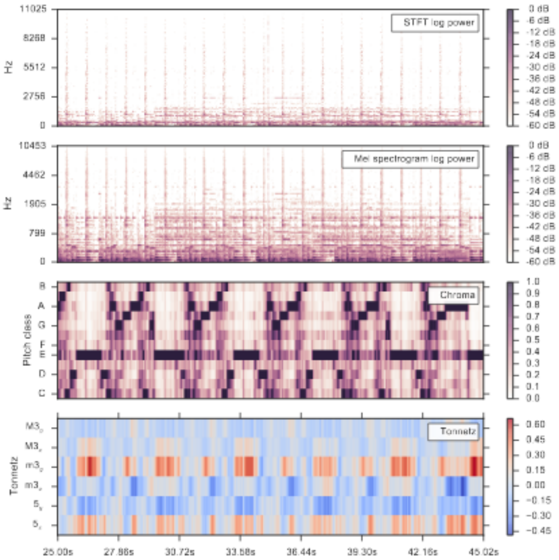

Figure 3: Spectrograms generated by the Librosa library

.

further divided according to their font, family and quality. The first category refers to the method of sound production (acoustic, electronic or synthesized), while the second category is responsible for classifying the instruments according to their main characteristics and the third separates the samples based on their sound resolution [15].

Despite being robust, NSynth has some particular characteristics that can interfere with the classification of instruments. Starting with the way the samples were created. Regardless of whether they represent acoustic guitar, bass, flute, or another instrument, their means of production was a standard MIDI piano. In consequence, tones range from 21 to 108 and detected speeds are 25, 50, 75, 100, 127. Therefore, not all instruments are capable of producing the 88 tones of a normal range, resulting in an average of 65.4 tones per instrument. There are also sample packs that contain sounds duplicated at different speeds, resulting in a unique 4.75 speed per pitch. These characteristics directly interfere in the investigation of the timbre, resulting in guitar samples with notes as low as a bass and bass notes as high as a guitar note [16].

4.1. Classification Algorithms

The process of classifying sound samples can be performed by applying machine learning techniques. That is, in a pre-processed database, the computer follows a set of steps to learn to differentiate between two different audios. One of the most famous methods for this function is the Naive Bayes algorithm, a probabilistic classifier, based on the theorem of the mathematician Thomas Bayes, which lends your name to the tool. This theorem consists of multiplying a probability collected before the system is executed (*a priori*) by a probability obtained after the occurrence of the event (*a posteriori*). This technique is based on conditional independence of values, where the presence of a certain event has no correlation with any other, hence the name "Naive Bayes". Its mathematical representation can be seen in the equation 1 [17].

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{1}$$

P(A) and P(B) are the *priori* probabilities of their respective events. P(A|B) is the *posteriori* probability of A, after the occurrence of B event, while P(B|A) is the occurrence of B after A. Its main advantage is that it only needs a few training data to estimate the values. It is considered one of the most effective methods for dealing with real-world problems, although its construction is relatively simple.

Another form of classification is using decision trees. This is a hierarchical structure characterized by supervised learning. As a consequence, the input data has its characteristics defined from the beginning and uses this knowledge to act on the new data that is received [18].

These trees are formed by nodes, which represent data attributes, by branches, which indicate the classification rules, and by leaves, which refer to the results. The objective of the algorithm is to progressively divide the database, partitioning the space into sub-regions based on a descriptive feature. This space reduction happens until they are so small that they can be classified using a single label. From there, the new input data are also categorized in this way.

For the problem discussed here, suppose a new audio sample has been inserted into the system and must be designated as a bass or acoustic guitar. The first question the algorithm must answer is how often this sample matches the root node of the decision tree. From there, other questions can be asked based on the previous answer. This sequence is organized to generate a tree, which will indicate the way forward to arrive at a ranking. For this, the algorithm simply leaves the root node of the tree and crosses it to a leaf node, responsible for indicating the class of that audio. Each choice for a different segment will result in different results.

Some other information is important for building a tree. The first one to be mentioned is the gain information, which can be seen in the equation 2 and represents what was learned about the labels when a region is divided. In this context, H indicates the impurity of the region R, while Re is the left subregion and Rd the right subregion.

$$Gain(R, R_e, R_d) = H(R) - \frac{|R_e| * H(R_e) + |R_d| * H(R_d)}{|R|} \tag{2}$$

The second prominent factor is entropy, responsible for indicating the degree of purity and instability of the classifier. For this, it resorts to parents and child nodes, where the attribute that generates the greatest difference will be chosen. This condition can be seen in the equation 3 [19].

$$entropy(R) = -\sum p(c|R)log(p(c|R)) \quad (3)$$

P(c|R) indicates the probability that a point in region R belongs to class c. This probability is estimated by the ratio between the number of these points.

The last point to be considered when planning this method is the Gini factor, observed in the equation 4. For this condition, the variables c and R assume the same values as in the entropy calculation, with the highest index generated being the one chosen to integrate the system [19].

$$Gini(R) = \sum p(c|R)(1 - p(c|R)) \quad (4)$$

This paper also made use of two more classification methods: Support Vector Classifier (SVC) and multi-classifiers. The SVC is based on an adjustment of the data provided, applying penalties, loss functions, metric scores, and decision limits to determine which one best fits the analyzed problem. The best performing data is then transported to a hyperplane, where it will be fed some enhancement features to decide the classes of the input objects. Although it has some disadvantages, such as not punishing interception and converging at a slower time compared to other techniques, it was chosen because of its ability to be an individual scheme. In this way, each classifier is directly compared to another classifier, showing better scalability of penalty choices in large amounts of samples.

Multi-classifiers, on the other hand, consist of combining and applying a set of classification algorithms in the parts of the input space where they have better performance, in order to optimize the output values. This category of the system is characterized by allowing weak data to provide good results and not requiring such a precise adjustment of the base. In the classification of bass and acoustic guitar samples, the multi-classifier used was Bagging. This procedure consists of choosing a random sample from the dataset and generating new subsets from it. In each of these divisions, a different classification is applied, so that the algorithm obtains diversity in the response models. At the end of this step, it starts a simple voting process to determine which class to choose. The selected class is the one that got the best answers for the different classifiers. It is noteworthy that in this work the training and testing bases were applied equally to the 3 classifiers, not a subset for each one, as is the default in Bagging [20].

It is important to highlight that in a classification problem, data mining plays a role as important as the classification system itself. Good data selection prevents overfitting, improves performance, and reduces training time. A tool that helps in this task is the F-test, commonly used to better identify the model that best fits a certain population. For this, each test receives an important value for each resource, according to the improvement or deterioration caused in the system. For the problem proposed here, the F-test was used to find the 3, 5, 6, and 8 best samples [21].

4.2. Pre Processing Data

For this stage, 2500 bass and acoustic guitar samples were initially collected, at different frequencies. Through the Librosa library, the numerical values of each sample were extracted, as an audio/sound wave can be represented by a vector, which are used to reconstruct a sound wave from its sampling rate and the application of trigonometric equations.

Using this vector as a basis, the Short-Time Fourier Transform (STFT) was applied. This technique consists of dividing the spectrogram into smaller intervals, making each one of them become a constant. Subsequently, a Fourier transform is applied to each of these partitions. The method was used to collect the modulus of the vector result, as it has negative and positive values.

After this procedure, the STFT output is used in the audio descriptors to provide another numerical vector having specific audio characteristics. From this new vector, the mean and standard deviation between each one is applied, providing two features per descriptor.

In this way, each sample is treated and ready to be used in classification. Regarding the pre-processed data, there are 8 features for each sample. Furthermore, the class referring to the sample is returned, which can be 0 for bass and 1 for acoustic guitar. No problems were found regarding inconsistent or missing data.

5. Results and Discussions

We conducted our experiments separated in three steps, using different partitions of the dataset, namely: i) partial use of NSynth, containing 5000 guitar and bass samples; ii) full use of NSynth, containing all 24123 guitar and bass samples; and iii) full use of NSynth with all electronic guitar and bass samples, reaching a total of 99659 copies. In each of the three steps, the previously mentioned algorithms were used. It is noteworthy that all algorithms were executed in the f1 test and the number of resources used was gradually increased, generating for each stage, five tests based on the 2, 3, 5, 6 and 8 main features.

5.1. Step 1: Partial NSynth dataset

Initially, 5000 samples were collected, 2500 referring to acoustic guitar and 2500 to bass. Despite providing acoustical and synthetical samples, it used only electrical ones, with a database randomly divided between 60% for training and 40% for test.

Even with a small base, the algorithms demonstrated different types of behavior. In relation to Naive Bayes and the algorithm provided by Sklearn, by using the same technique for classification, they obtained an almost identical result, starting in the range of 69% for 2 features, and increasing the precision significantly until reaching 6 features, where it stabilized at 80%.

The decision tree was the algorithm that achieved the best accuracy, reaching a peak of 92% with 5 features and declining slightly in the following analyses. SVC, else ways, presented a very peculiar behavior. Under the use of up to 3 features, he got a result of 75% hit, however, the increase in the number of features caused a considerable drop of almost 25%. This behavior highlights the idea that increasing features does not always improve the accuracy of an algorithm.

On the implemented multiclassifier, the technique used was the voting system, where the Naive Bayes, decision tree and SVC algorithms were used. In this way, each instance received classifications from the algorithms, with the most frequent class being the one used as a result. It can be seen that, based on the voting system, the multi-classifier ends up tending to behave similarly to the best classifier used. In this case, even with worse accuracy, the multi-classifier obtained results very similar to those of the decision tree. It is noteworthy that the decrease in SVC accuracy from the use of 5 features did not affect the results of this method. All these behaviors are summarized in Figure 4.

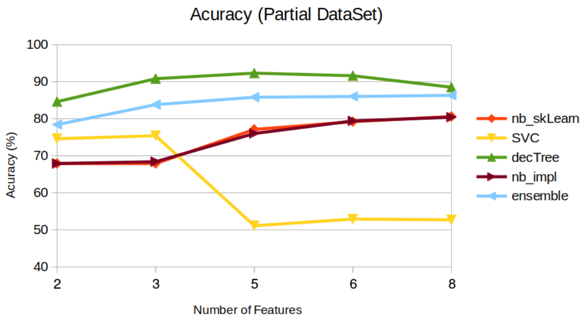

Figure 4: Partial Dataset Classification.

Aiming at a deeper analysis of the influence of the descriptors on the result of the classifiers, the correlation matrix between the invoices was created, which can be seen in Figure 5.

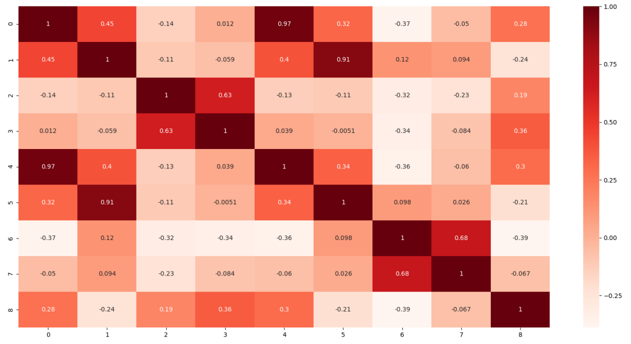

Figure 5: Correlation Matrix Between Features.

As much as the matrix demonstrates all the correlations between the resources, the task of defining which ones were really important is a little difficult. For this, taking into account the classes of each sample, a ranking was made of which characteristics were more correlated with each other. From the generated classification, it was noticed that the use of the mean and standard deviation under a descriptor does not always result in features that will have similar correlations in relation to the classes. In Figure 6, it is observed that the four most correlated features were collected from different descriptors. That is, when the mean for a descriptor resulted in a similar feature, the standard deviation resulted in an inverse behavior.

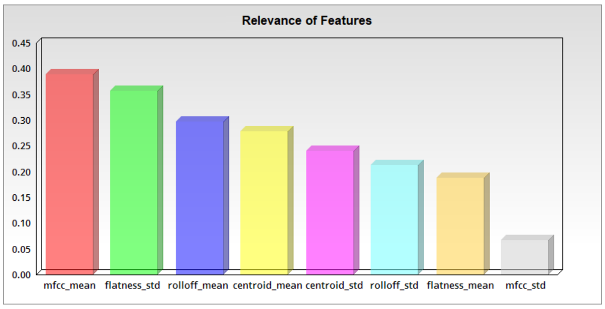

Figure 6: Feature Ranking.

The correlation between features is not always directly related to the accuracy of the classifier. Thus and so, removing a feature that has a lower relationship to the others will not necessarily cause an improvement in accuracy.

Regarding the execution time for this step, in addition to the SVC that obtained times between 400 and 600 milliseconds, the other algorithms obtained acceptable times, ranging between 2 and 30 milliseconds.

5.2. Step 2: Full NSynth dataset

The results of the first stage were obtained from a very small and specific database. Wherefore, step 2 used the complete database, totaling 99659 samples, being 33342 guitar and 66317 bass, available in acoustic, synthetic and electric formats. In respect of training and testing data, Nsynth already provides the partitioned data, totaling 98164 samples for training and 1495 for testing.

While the results of step 1 averaged between 75% and 95%, when using the entire database, a better result was expected, since the use of more input values improves the training of classifiers, and consequently, increases accuracy. However, the results ended up being between 50% and 65%, as shown in Figure 7.

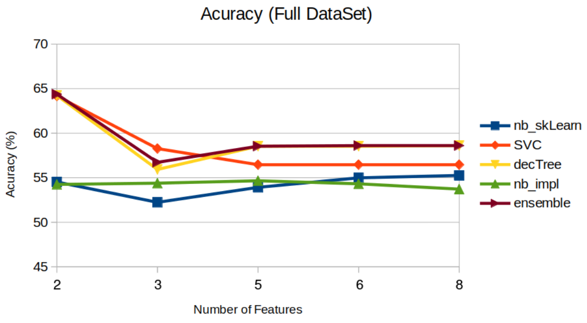

Figure 7: Full Dataset Classification.

In other words, even with the increase in the database, some other factor ended up causing a sharp drop in the accuracy of the algorithms. As the database used ended up encompassing samples from more sound sources and not just electric instruments, the timbre classification may have been compromised.

In relation to the execution time of the algorithms, the SVC had a much higher execution time when compared to the others. Despite its contribution to the design and its use being desirable to reduce calculation time, the basic composition of the SVC made it have worse performance. This is because the kernel acts as a space function, which can be highly complex. It is also necessary to consider the size of the database to which the SVC was submitted, as this can cause the kernel array to be stored directly in memory, thus affecting its performance as a whole. Still, on the performance of the algorithms in relation to the execution time, all the algorithms had their execution worsened by increasing the number of features desired. Figure 8 shows this decay.

time (ms)	nb_skLearn	SVC	decTree	nb_impl
2 features	17.12	551539.33	200.41	285.21
3 features	17.06	493275.20	289.28	344.80
5 features	19.30	589837.13	456.41	396.59
6 features	20.02	624353.81	550.10	507.95
8 features	21.76	638291.16	755.98	669.39

Figure 8: Full Dataset Time Classification.

5.3. Step 3: Full electronic NSynth dataset

To better investigate the reason for the accuracy decrease, a third and final analysis was applied, but now only with electronic instruments, totaling 16206 guitar samples and 7917 bass samples. Besides, it is possible to investigate whether the drop in accuracy is related to samples generated by different sound sources. Once the tests were carried out in this way, the algorithms returned to obtain good results, with an accuracy between 65% and 95%, without considering the SVC result, which was much lower than the other classifiers. Such values are graphically represented in Figure 9.

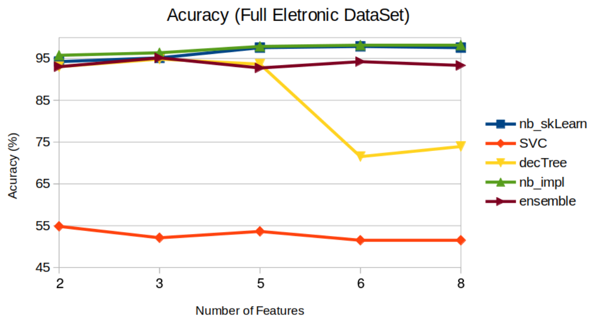

Figure 9: Full Electronic Dataset Classification.

Analyzing each algorithm individually, the decision tree obtained a decrease in precision when using 6 and 8 features. Naive Bayes achieved the best results, even being better than the multi-classifier. This fact can be explained by the behavior of the SVC, which obtained low accuracy, between 50% and 55%. That is, the SVC ended up casting wrong votes for the multi-classifier, considerably reducing its accuracy. Consequently, we conclude that implementing a voting system may sometimes not be a good choice when one of the classifiers used is getting poor results.

It is also observed that the execution time of the algorithms remained acceptable, below 200 milliseconds. The SVC, oppositely, because it uses complex functions, ended up obtaining an execution time between 29 and 35 milliseconds.

6. Conclusions

This paper presented an automatic instrument classification experiment using a well-known dataset, audio descriptors commonly used to identify timbres, and three different classification algorithms: Naive Bayes, Decision Tree, and SVC.

From the analysis of the results, it is clear the importance and reason for choosing each of these algorithms. Naive Bayes stands out for being very efficient in estimating probabilities, having a simple and easy-to-implement structure. The decision tree has the advantages of easy implementation and interpretation of data compared to more traditional models, while SVC appears as an alternative for the part of the database that was not as robust or had some noise.

Finally, it is noteworthy that the complexity of the problem of dividing two timbres is strongly influenced by the categories of instruments used. The task of distinguishing a drum from a flute, for example, can be easier than differentiating string instruments with similar timbres. Hence, the importance and applicability of automatic classification systems for music are explicit. Another strong point to be highlighted is the interdisciplinary character of this field of action, assisting in research, projects, and teaching of the most diverse types of science.

Acknowledgments

Authors would like to thanks to all ALICE members that made this research and development possible. The authors would like also to thank the support of the funding agencies CNPq (Grant Number 151975/2019-1), CAPES (Grant Number 88887.486097/2020-00) and FAPEMIG.

References

[1] Roberto Piassi Passos Bodo and Marcelo Gomes de Queiroz. Três abordagens para similaridade musical utilizando melodia, ritmo e timbre. Master's thesis, Universidade Federal de São Paulo, São Paulo, 2018.

[2] Lara Haidar-Ahmad. Music and instrument classification using deep learning technics. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 1–6. Nips.cc, 2017.

[3] Janet Marques and Pedro Moreno. A study of musical instrument classification using gaussian mixture models and support vector machines. *Cambridge Research Laboratory - Technical Report Series*, 09 1999.

[4] A. Eronen and A. Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 2, pages II753–II756 vol.2, 2000.

[5] Babak Toghiani-Rizi and Marcus Windmark. Musical instrument recognition using their distinctive characteristics in artificial neural networks. *ArXiv*, abs/1705.04971, 2017.

[6] Martin Vetterli Paolo Prandoni. *Signal Processing for Communications (Communication and Information Sciences)*. EPFL Press, 2008.

[7] Ivan Eiji Simurra. A utilização descritores de áudio à análise e composição musical assistidas por computador: um estudo de caso na obra labori ruinae, 2016.

[8] Ivan Eiji Simurra. Análise musical assistida por descritores de Áudio: um estudo de caso da obra reflexões de jônatas manzolli, 2015.

[9] Sound Analysis Pro Editorial. Wiener entropy, Outubro 2014. [Online; chapter 4].

[10] Luscinia. Spectrogram window, Outubro 2017. [Online; Parameters].

[11] Geoffroy Peeters. A large set of audio features for sound description. *Ircam - Analysis/Synthesis Team*, 2004.

[12] N. Fakotakis e G. Kokkinakis T. Ganchev. Comparative evaluation of various mfcc implementations on the speaker verification task. *10th International Conference on Speech and Computer (SPECOM 2005)*, 2005.

[13] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. *14th Python in Science COnference*, s/n(s/n), 2015.

[14] Eli Bressert. *SciPy and NumPy*. O'Reilly Media, 2012.

[15] Yotam Mann. The nsynth dataset, Abril 2017. [Online; NSynth Dataset].

[16] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1068–1077. JMLR. org, 2017.

[17] Raphael Campos. Machine Learning Beyond Deep Learning Árvores de decisão. `https://medium.com/machine-learning-beyond-deep-learning/\%C3\%A1rvores-de-decis\%C3\%A3o-3f52f6420b69`. Accessed: 2019-12-04.

[18] Diego Lopez Yse. Towards data science: The complete guide to decision trees. Accessed: 2019-12-04.

[19] Márcio Porto Basgalupp. *Árvores de Decisão*. PhD thesis, Universidade Estadual de Campinas - UNICAMP, 2010.

[20] GIlbert Tanner. Towards data science: A guide to ensemble learning: Increase your accuracy by combining model outputs, 2019. Accessed: 2019-12-04.

[21] G. E. P. BOX. NON-NORMALITY AND TESTS ON VARIANCES. *Biometrika*, 40(3-4):318–335, 12 1953.