Musical Genre Classification and Comparison Using Different Classifiers

1st Giuseppe Bonura
University of Milan
Department of Computer Science
Italy, Milan
giuseppe.bonura@studenti.unimi.it

***Abstract*—The automatic classification of musical genres is a complex operation and the features available are many. This study are divide in two parts, in the first, the k-means algorithm was used to determine the space of the features, subsequently were analyzed time and frequency features, compared, chosen and used with various classification algorithms in order to analyze their performance.**

***Index Terms*—Classification, Feature Extraction, Ensemble of classifiers, MLP, SVM, K-Means, Random Forest, Musical Genre Classification.**

I. Introduction

Since ancient times, humans has made use of classification, that is, the activity that consists in arranging the entities of a given domain of knowledge a) People of a given social group, b) Animal or plant species and c) Objects of a certain kind, in suitable containers of knowledge between which links are established regarding one or more relationships. This classification can also be applied in the musical field, specifically on one of the many properties that distinguish an audio sample, the **"genre"** [1]. A musical genre is represented by a label created and used by humans to categorize and describe the large universe of music. Taken together, musical genres don't have very precise and rigorous definitions but emerge as a result of a complex interaction between the public, marketing, historical and cultural factors. The automatic classification of musical genres has turned out to be a complex task from the earliest studies. The different characteristics that allow to identify a genre are different, in addition to those mentioned above, we can consider harmonic and rhythm as determining factors. Over the years, thanks to the evolution of the study of signals, it has been possible to implement functions capable of extracting these characteristics, which we will divide into two categories Time Domain Features and Frequency Domain Features. The importance of systems capable of automatically recognizing musical genres has had a certain relevance, especially in recent years, thanks to the increase in systems for using digital content online, and streaming/purchasing service such as spotify, deezer, soundcloud. We can place the recognition of musical genres in applications capable of performing music auto-tagging, automatically catalog music within huge databases and automatically create playlists according to some parameters provided by the user. In the literature there are many papers related to music classification, one of the first studies conducted by Tzanetakis and Cook in 2002 [2] involved using of both the mixture of Gaussian model and k-nearest neighbors along with three sets of carefully hand-extracted features representing timbral texture, rhythmic content and pitch content. Another study conducted by Tao Li and George Tzanetakis [3] who plans to take part in the study of Tzanetakis and Cook using different combinations of features and apply different classifiers obtaining better accuracy's. The study conducted in this document involves the use of a clustering algorithm known as K-Means to be able to determine the space of the features, also verifying through the Silhouette method the interpretation and validation of consistency within clusters of data [4]. The technique provides a succinct graphical representation of how well each object has been classified. Exploits the approaches used in the two papers cited, i.e. the extraction of multiple types of features, divided into two groups, Time Domain Features and Frequency Domain Features. These features were then combined and fed to multiple classifiers of different types, specifically a deep multi-layer perceptron network, an ensemble learning method known as random forest and in finally another supervised learning models called Support-Vector Machine or SVM. This approach was used to show with scientific evidence that not all features are necessary during multi-label classification, but that on the contrary, techniques can be exploited to reduce the number of features and therefore the computation complexity of the model. Furthermore, to provide objective feedback on the performance of some classifiers, and that not always the most complex can lead to the best result.

II. System Overview

The architecture of the implemented system is basically divided into 6 blocks, in the first two we have sample data preparation and data augmentation. The heart of the system is located in the third step or the extraction of the features and their choice. Next we have the clustering of the features and

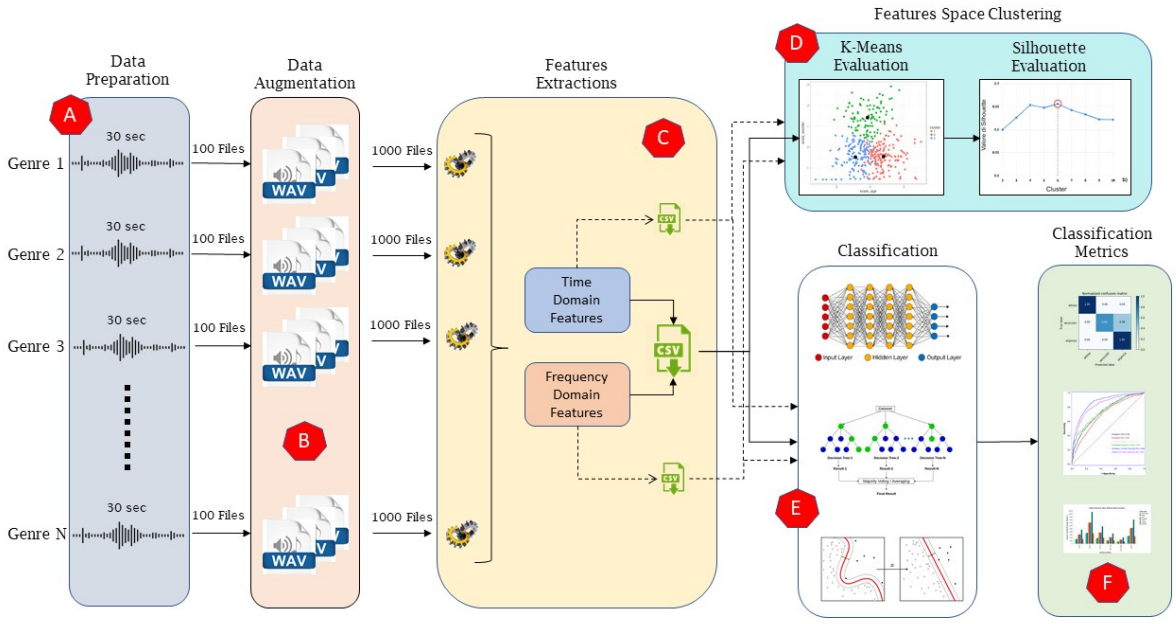

Fig. 1: High level architecture for music genres classifications system.

the classification through various approaches. As a last step we have the evaluation of the obtained results.

A. Data Preparation

As a first approach to the problem, after having identified the data set of interest GTZAN, based on this we have created our data-set choosing from various albums and compilations of various years for each musical genre that wee are chose to considered.

B. Data Augmentation

There are a lot techniques for audio data augmentations, in this project we used one of the most simple, or split data in more sub samples for increase the number of data that. Starting from 100 samples of 30 seconds and reaching 1000 samples of 3 seconds.

C. Features Extraction

Feature extraction is the process of computing a compact numerical representation that can be used to characterize a segment of audio. The design of descriptive features for a specific application is the main challenge in building pattern recognition systems. Once the features are extracted standard machine learning techniques which are independent of the specific application area can be used. [2]

1) Time Domain Features
   a) **Tempo**: Particularly useful features to extract from musical sources may be an approximation of tempo as well as the beat onset indices, an array of frame numbers corresponding to beat events.
   b) **Energy**: This feature provides the so-called power of the signal. Let $s_i(k)$, where $k = 1...W_L$ be the sequence of audio samples of the i_{th} frame, where W_L is the length of the frame. The short-term energy is computed according to the equation:

$$E(i) = \frac{1}{W_L} \cdot \sum_{n=1}^{W_L} s_i(n)^2 \cdot \tag{1}$$

   c) **Energy Entropy**: The entropy of energy can be interpreted as a measure of abrupt changes in the energy level of an audio signal. In order to compute it, we first divide each short-term frame in K sub-frames of fixed duration. Then, for each sub-frame, j,we compute it's energy as in Eq.1 and divide it by the total energy. At a final step, the entropy,H_i of the sub-frames sequence is computed according to the equation:

$$H(i) = -\sum_{j=1}^{W_L} e_j \cdot \log_2(e_j) \cdot \tag{2}$$

   d) **Root Mean Square Energy (RMSE)**: Root-Mean-Square Energy is typically denoted RMS energy, is another time domain feature, computed as shown in Eq.3 where W_L represents frame size, i.e., the number of samples in each frame and $s_i(k)$ the signal. As it relates to perceived sound intensity, RMS energy can be used for loudness estimation and as an indicator for new events in audio segmentation. [5]

$$RMSE = \sqrt{\frac{1}{W_L} \cdot \sum_{n=1}^{W_L} s_i(n)^2} \cdot \tag{3}$$

   e) **Zero-Crossing Rate (ZCR)**: Measures the number of times the amplitude value changes it's sign. In other words, the number of times the signal changes value,

from positive to negative and vice versa, divided by the length of the frame, according to the equation:

$$Z(i) = \frac{1}{2W_L} \cdot \sum_{n=1}^{W_L} |sgn[s_i(n)] - sgn[s_i(n-1)]| \cdot \quad (4)$$

2) Frequency Domain Features

a) **Spectral Centroid**: The spectral centroid is a measure to characterize the "center of mass" of a given spectrum. Perceptually, it has a robust connection with the impression of sound brightness, or rather as an indication of the amount of high-frequency content in a sound, obtained according to the following equation, where $m_t(n)$ represents number of frequency bins, i.e., the number of the highest frequency band.

$$SC_i = \frac{\sum_{k=1}^{N} m_t(n) \cdot n}{\sum_{n=1}^{N} m_t(n)} \quad (5)$$

b) **Spectral Bandwidth**: Spectral bandwidth is derived from the spectral centroid and indicates the spectral range of the interesting parts in the signal, i.e., the parts around the centroid. It can be interpreted as variance from the mean frequency in the signal. The definition is given in Eq.6.The average bandwidth of a music piece may serve to describe its perceived timbre. [6]

$$BW_i = \frac{\sum_{n=1}^{N} |n - SC_i| \cdot m_t(n)}{\sum_{n=1}^{N} m_t(n)} \quad (6)$$

c) **Spectral Contrast**: Four each frame of a spectrogram s is divided into sub-bands. For each sub-band, the energy contrast is estimated by comparing the mean energy in the top quantile (peak energy) to that of the bottom quantile (valley energy). High contrast values generally correspond to clear, narrow-band signals, while low contrast values correspond to broad-band noise. [7]

d) **Spectral Rolloff**: This feature is defined as the frequency below which a certain percentage (usually around 85-90%) of the magnitude distribution of the spectrum is concentrated. Therefore, if the m_{t_h} DFT coefficient corresponds to the spectral rolloff of the i_{t_h} frame, then it satisfies the following Eq. 7, where C is the adopted percentage (user parameter). The spectral rolloff frequency is usually normalized by dividing it with N, so that it takes values between 0 and 1.

$$\sum_{n=1}^{m} m_t(n) = C \cdot \sum_{n=1}^{N} m_t(n) \quad (7)$$

e) **Mel Frequency Cepstral Coefficient**: Mel frequency cepstral coefficients (MFCCs) have their origin in speech processing but were also found to be suited to model timbre in music. The MFCC feature is calculated in the frequency domain, derived from the signal's spectrogram and for each frame, cepstral coefficients are computed using Mel-filter bank with a variable numbers of Mel filters. In music signal processing, between 13 and 25 MFCCs are typically computed for each frame. The MFCC feature extraction for this project can be summarized in five points [8]:

- **Pre-emphasis.**
- **Frame blocking and windowing.**
- **DFT spectrum.**
- **Mel spectrum.**
- **Discrete cosine transform (DCT).**

f) **Chromogram**: In music, the term chromagram is attentive to the twelve different pitch classes. This vector of features is computed by grouping the DFT coefficients of a short-term window into 12 bins. Each bin represents one of the 12 equal tempered pitch classes of Western-type music. Each bin produces the mean of log-magnitudes of the respective DFT coefficients. One main characteristic features of colour is that they capture the harmonic and melodic features of music, at the same time robust to changes in timbre and instrumentation.

g) **Constant-Q chromagram**: The frequencies that have been chosen to make up the scale of Western music are geometrically spaced. Thus the DFT, although extremely efficient in the FFT implementation, yields components which do not map efficiently to musical frequencies. This is because the frequency components calculated with the DFT are separated by a constant frequency difference and with a constant resolution. A calculation similar to a DFT but with a constant ratio of center frequency to resolution has been made; this is a constant Q transform and is equivalent a 1/24-oct filter bank. In addition to advantages for resolution, representation with a constant pattern has the advantage that note identification, instrument recognition, and signal separation. [9]

h) **Chroma Energy Normalized**: That there are different ways of computing chroma features and that the properties of chroma features can be adjusted by applying suitable post processing steps such as logarithmic compression, normalization, or smoothing. The two steps, quantization and smoothing, amount to computing weighted statistics of the energy distribution over a window of l consecutive vectors. Therefore, we call the resulting features CENS. The main idea of CENS features is that taking statistics over relatively large windows smooths out local deviations in tempo, articulation, and execution of note groups such as trills or arpeggios. [10]

D. Features Space Clustering

In this architecture block, after extracting the characteristics from the audio samples we applied an unsupervised learning technique. It too, as in any classification problem, tries to learn in order to associate a label y of the set Y with a given point x of the input space X, in our case the correct musical genre. The substantial difference between unsupervised and supervised learning techniques is that of automatically extracting

knowledge from the input data. This knowledge is extracted without specific knowledge of the content to be analyzed. One of the main algorithms of this category is the clustering algorithm called K-Means. The aim of the K-means algorithm is to minimize the total inter-cluster variance and to maximize the intra-cluster variance. We first choose K initial centroids, where K is a user specified parameter, namely, the number of clusters desired. Each point is then assigned to the closest centroid basis of a predetermined measure of similarity or distance, and each collection of points assigned to a centroid is a cluster. The centroid of each cluster is then updated for each iteration based on the points assigned to the cluster, until the algorithm converges.

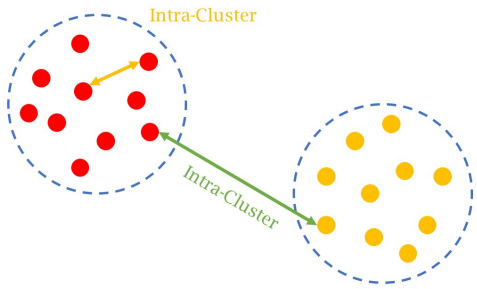

Fig. 2: Schematic example of intra-cluster and inter-cluster distance.

Now we see depicted below a pseudo code version of the K-Means algorithm:

Algorithm 1: K-means

Input: Set N patterns x_i, desired number of clusters K
Output: Set of K clusters
Choose K patterns as the c_j centers of clusters;
repeat
  Assign each pattern x_i to the cluster P_i that has the center c_j closest to x_i (that is, the one that minimizes $d(x_i, c_j)$);
  Recalculate the c_j centers of the clusters using the patterns that belong to each cluster (mean or centroid);
until *Convergence criterion are satisfied*;
return

E. Classification

The fifth block is dedicated to the classification of the characteristics extracted with the aid of a different supervised classification methods. The ones we have decided to use to be able to compare them are three, we'll see the main peculiarities of each one.

1) **Support Vector Machine**: Support vector machines have shown superb performance at binary classification tasks and handle large dimensional feature vectors better than other classification methods. Basically, a Support Vector Machine aims at searching for a hyper-plane that separates the positive data points and the negative data points with maximum margin. To extend SVM for multi classification there are two approaches called:
   - One-Vs-Rest (ovr): Is a heuristic method for using binary classification algorithms for multi-class classification. It involves splitting the multi-class data set into multiple binary classification problems. This approach requires that each model predicts a class membership probability or a probability-like score. The max of these scores is then used to predict a class. [11]
   - One-Vs-One (ovo): Like ovr, ovo splits a multi-class classification data set into binary classification problems. Unlike ovr that splits it into one binary data set for each class, the ovo approach splits the data set into one data set for each class versus every other class. This is significantly more data sets, and in turn, models than the ovr strategy described in the previous section. The formula for calculating the number of binary data sets, is as follows: $K \cdot (K-1)/2$. [11]
2) **Random Forest**: The random forest algorithm is an extension of the bagging method as it utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees. Feature randomness, also known as feature bagging generates a random subset of features, which ensures low correlation among decision trees. After several data samples are generated, these models are then trained independently, producing a more accurate estimate. This is a key difference between decision trees and random forests. While decision trees consider all the possible feature splits, random forests only select a subset of those features.
3) **Deep Neural Network**: An artificial neural network (ANN)is a mathematical model that tries to simulate the functioning of neurons present in biological organisms. These models are directly inspired by the functioning of the brain. Neural networks (NN) are made up of a layer of input and output neurons, and possibly one or more intermediate layers called hidden, and if a number of hidden layer is greater than 1 the suffix deep will be prefixed to the network. The interconnections go from one layer to the next and the signal values can be both discrete and continuous. The values of the weights associated with the inputs of each node can be static or dynamic in such a way as to adapt the behavior of the network based on variations in the input signals. The functioning of a NN can be schematized in two phases: the learning phase and the recognition phase. In the learning phase the network is instructed on a sample of data taken from the set of those that will then have to be processed; in the recognition phase, which is then that of normal operation, the network is used to process the incoming data based on the configuration achieved in the previous phase.

F. Classification Metrics

In this section we analyze how to evaluate the goodness of a model or several classification models. The methods used are, the ROC curves and the confusion matrix.

- **Confusion Matrix**: A classifier can be described as a function that maps the elements of a set into certain

classes or groups. In the case of supervised classification, the set of data to be classified contains a subdivision into classes, with respect to which it's possible to evaluate the quality of the result produced. In a binary classification problem, the set of data to be classified is divided into two classes that we can conventionally indicate as positive **(p)** or negative **(n)**. The results of applying a binary classifier fall into one of the following four categories.

1) True Positive (TP).
2) False Positive (FP).
3) True Negative (TN).
4) False Negative (FN).

- **ROC Curve**: The classification model would be optimal if it maximized both sensitivity and specificity at the same time. However, this isn't possible. Given the definitions of specificity and sensitivity, we have that, raising the value of specificity, the false positives decreases, but false negatives increase, which leads to a decrease in sensitivity. It can be observed that therefore there is a trade-off between these two parameters, which leads to more sensitive but less specific and, vice versa. Generally the optimal classification corresponds to the point closer to the upper left corner, representing a sensitivity and specificity of 100%. The curve below is called the ROC curve.

$$sensitivity = \frac{TP}{TP + FN} \tag{8}$$

$$specificity = \frac{TN}{TN + FP} \tag{9}$$

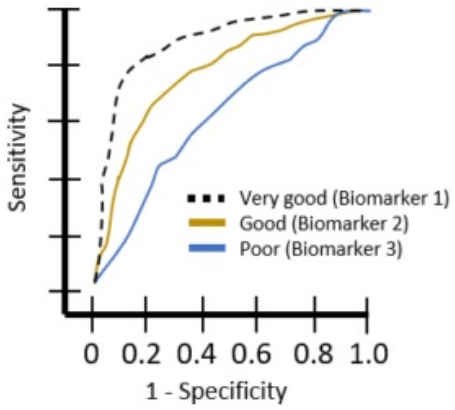

Fig. 3: Example of ROC curve.

III. Experimental Results

The first thing during the practical construction of the project was to find various musical albums and collections of 10 genres from various vintages. This operation was necessary in order to populate the data set with sufficient variability. The first script used he's able to divide each audio track into many sub samples of 30 seconds each, then through a second script 100 samples were chosen randomly. In order to increase the amount of data available to us while keeping the extraction of features low at the computational level, the 30-second samples were divided into 10 3-second samples, thus obtaining 1000 3-second samples. The next step in the creation of the data set was the one related to the construction of the script used to extract the characteristics from the audio samples to insert them into a CVS file. The extraction took place using the Librosa library as a support while the two functions for energy extraction and energy entropy were written from scratch. Once the file containing the characteristics of each sample was obtained, the approach we chose to use was to compare the two types of classifications, unsupervised and supervised considering combinations of features and variable number of MFCCs coefficients (13, 20, 40):

- Time Domain Features
- Frequency Domain Features

For the extraction of the features, were used the following parameters:

- Sample Length: 3 sec
- Sample Rate: 22050 Hz
- Length of the FFT window: 2048
- Hop Length: 512
- Window Function = Hann

For the evaluation of the two classification methodologies (unsupervised and supervised), two scripts in python have been created respectively. Through the aid of third-party libraries and specially created functions were able to conduct a careful evaluation of various models. Let's see the sequential steps performed for the two approaches:

- Unsupervised Learning:
  1) Load data.
  2) Calculate correlation matrix.
  3) Run K-Means algorithm.
  4) Use PCA algorithm for reduce dimensionality.
  5) Plot clusters and centroids on 2D matrix.
  6) Plot confusion matrix.
  7) Plot ROC curve.
  8) Calculate consistency using Silhouette.
- Supervised Learning:
  1) Load data.
  2) Split data for train and test.
  3) Calculate correlation matrix.
  4) Use PCA for plot features on 2D matrix.
  5) Plot BPM graph.
  6) Load models.
  7) Evaluate model and get results.

After having launched the scripts for the two classification methodologies, we have created appropriate graphs and tables to be able to compare the results obtained.
Using this type of features and the large number of clusters, it's possible to notice from Fig. 7 that the k-means algorithm isn't very performing given the high data density and the non-globular shapes, the supervised algorithms behave much better. With reference to the number of clusters, it's possible to verify and evaluate the most appropriate value using the Silhouette algorithm. The second series of graphs involves the second

ROC results curves for time features

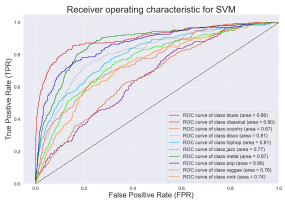

Fig. 4: SVM Time Features

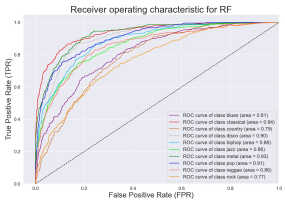

Fig. 5: RF Time Features

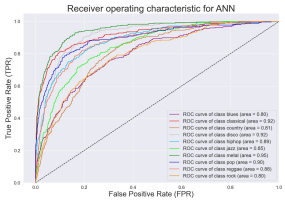

Fig. 6: ANN Time Features

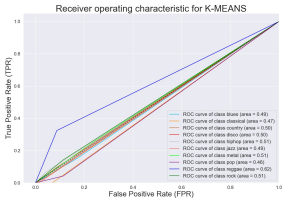

Fig. 7: K-Means Time Features

set of features, that is the frequency domain features. In this evaluation we were able to evaluate the impact of the number of mel-frequency cepstrum coefficients.

13 MFFCs - ROC results curves for frequency features extractions

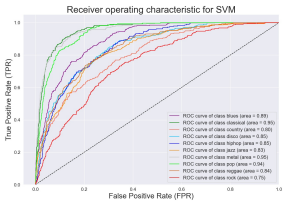

Fig. 8: 13MFCCs SVM

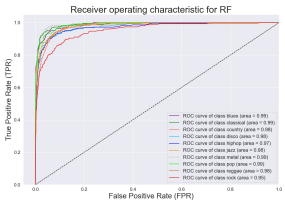

Fig. 9: 13MFCCs RF

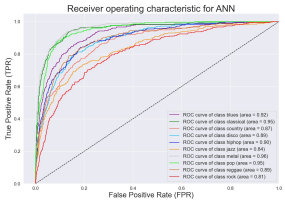

Fig. 10: 13MFCCs ANN

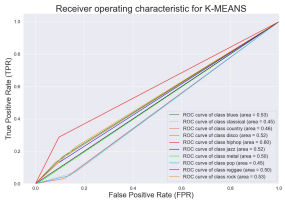

Fig. 11: 13MFCCs K-Means

TABLE I: 13MFCCs supervised approach frequency features

	13MFCCs			
Classifier	Acc(%)	RMSE	F1	Ex.Time(s)
SVM	49,60	2,94	0,48	43,62
RF	**81,56**	**1,85**	**0,81**	**52,98**
ANN	56,69	2,77	0,56	519,50

From data reported in Tables I, II, III it's possible to verify how the increase of the spectral coefficients has led to a certain improvement. While as regards the classification of the K-Means algorithm by viewing the Figs. 11, 15, 19, it's possible to note that, even with a set of different and more explanatory features, it's performances are worse. It's also possible to note how for the different classifiers of the supervisioned approach,

20 MFFCs - ROC results curves for frequency features

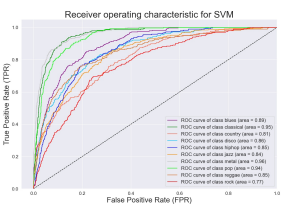

Fig. 12: 20MFCCs SVM

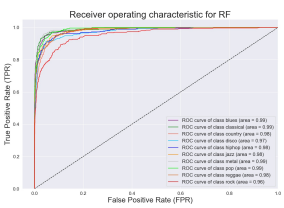

Fig. 13: 20MFCCs RF

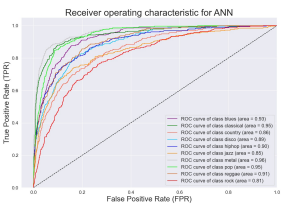

Fig. 14: 20MFCCs ANN

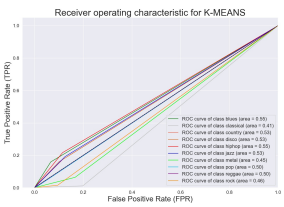

Fig. 15: 20MFCCs K-Means

TABLE II: 20MFCCs supervised approach frequency features

	20MFCCs			
Classifier	Acc(%)	RMSE	F1	Ex.Time(s)
SVM	50,96	2,90	0,49	45,84
RF	**82,39**	**1,76**	**0,82**	**55,34**
ANN	56,39	2,74	0,56	499,91

40 MFFCs - ROC results curves for frequency features

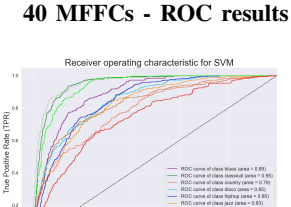

Fig. 16: 40MFCCs SVM

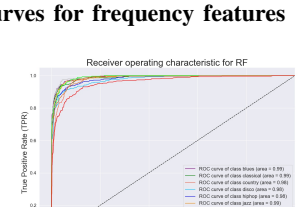

Fig. 17: 40MFCCs RF

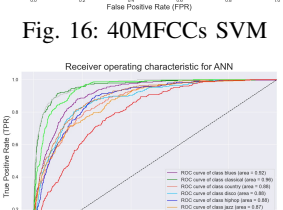

Fig. 18: 40MFCCs ANN

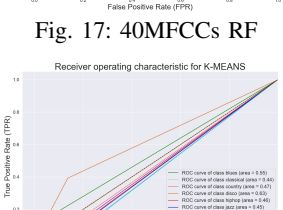

Fig. 19: 40MFCCs K-Means

TABLE III: 40MFCCs supervised approach frequency features

	40MFCCs			
Classifier	Acc(%)	RMSE	F1	Ex.Time(s)
SVM	49,33	2,97	0,47	49,79
RF	**83,39**	**1,69**	**0,83**	**65,04**
ANN	52,76	2,86	0,52	270,98

this features and the increase of the coefficients as a function of the execution time allows us to make detailed evaluations on which is the most suitable classifier for musical genres classification. You can see the best result among the three classifiers in bold, and the best among the three in green.

The last approach depicted is related to the fusion of the temporal and frequency features to verify the improvements obtained.

13 MFFCs - ROC results curves for Time+frequency features

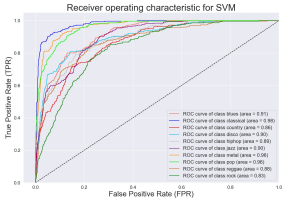

Fig. 20: 13MFCCs SVM

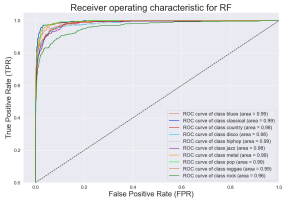

Fig. 21: 13MFCCs RF

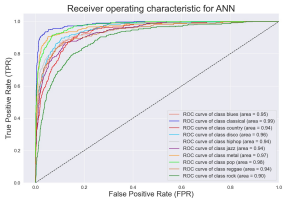

Fig. 22: 13MFCCs ANN

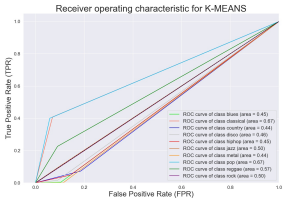

Fig. 23: 13MFCCs K-Means

TABLE IV: 13MFCCs supervised approach Time+frequency features

	13MFCCs			
Classifier	Acc(%)	RMSE	F1	Ex.Time(s)
SVM	60,09	2,77	0,59	39,50
RF	**84,93**	**1,64**	**0,84**	**52,93**
ANN	70,06	2,42	0,70	428,97

20 MFFCs - ROC results curves for Time+frequency features

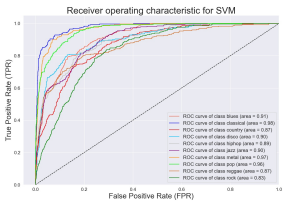

Fig. 24: 20MFCCs SVM

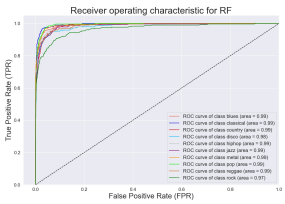

Fig. 25: 20MFCCs RF

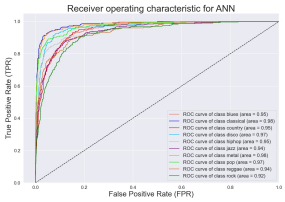

Fig. 26: 20MFCCs ANN

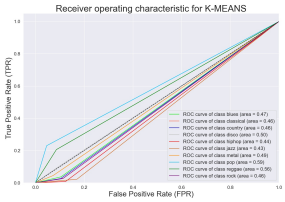

Fig. 27: 20MFCCs K-Means

TABLE V: 20MFCCs supervised approach Time+frequency features

	20MFCCs			
Classifier	Acc(%)	RMSE	F1	Ex.Time(s)
SVM	60,23	2,75	0,59	37,38
RF	**85,96**	**1,59**	**0,85**	**58,22**
ANN	71,89	2,33	0,71	310,88

40 MFFCs - ROC results curves for Time+frequency features

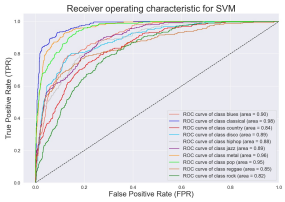

Fig. 28: 40MFCCs SVM

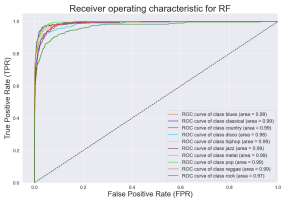

Fig. 29: 40MFCCs RF

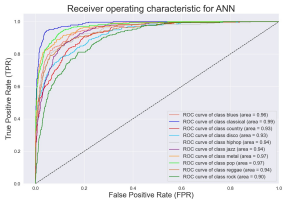

Fig. 30: 40MFCCs ANN

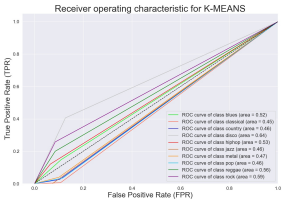

Fig. 31: 40MFCCs K-Means

TABLE VI: 40MFCCs supervised approach Time+frequency features

	40MFCCs			
Classifier	Acc(%)	RMSE	F1	Ex.Time(s)
SVM	57,83	2,77	0,56	40,68
RF	**86,06**	**1,52**	**0,86**	**72,40**
ANN	67,13	2,40	0,66	335,67

IV. Conclusions And Future Work

In this article it was possible to prove how the increase of features isn't always the best solution. It's possible to see how the optimal result considered, taking into account costs and benefits is that with 20 MFCCs (Tab. V). Although the table relating to 40 MFCCs (Tab. VI contains slightly higher values, the execution time and above all the time used for the realization of the data sets isn't worth the obtained benefit. Also as regards the classifiers, the most used ones aren't always the best, or in any case they can achieve the same performance but with a greater increase in execution time. In fact, with a lot of probability it's possible to reach close values obtained in Tab. V by adding neurons, modifying the topology of the neural network and and performing a very thorough tuning, going however to meet a greater computational complexity and a higher execution time. Another point of attention is related to the unsupervised approach, we have been able to see how the high number of features, with values very close to each other, do not allow the algorithm to be able to cluster optimally while keeping high the inter cluster distance. Future developments are related to the search for efficient methods of selecting the most relevant features for the unsupervised approach, the use of new classification algorithms or networks already known such as CNN that can be used by directly passing the extracted images of Mel-Spectrogram or Log-Mel Spectrogram.

References

[1] J. Samson, "In grove music online. oxford music online," 2012, online; accessed 4-March-2012. [Online]. Available: https://doi.org/10.1093/gmo/9781561592630.article.40599

[2] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[3] Tao Li and G. Tzanetakis, "Factors in automatic musical genre classification of audio signals," in *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684)*, 2003, pp. 143–146.

[4] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377042787901257

[5] P. Knees and M. Schedl, *Music similarity and retrieval: an introduction to audio-and web-based strategies*. Springer, 2016, vol. 36.

[6] A. Lerch, *An introduction to audio content analysis: applications in signal processing and music informatics*. Hoboken NJ: Wiley-IEEE Press, 2012. [Online]. Available: https://cds.cern.ch/record/1540227

[7] Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai, "Music type classification by spectral contrast feature," in *Proceedings. IEEE International Conference on Multimedia and Expo*, vol. 1, 2002, pp. 113–116 vol.1.

[8] K. E. M. Rao, K.Sreenivasa, *Speech Recognition Using Articulatory and Excitation Source Features*. Springer International Publishing, 2017.

[9] J. Brown, "Calculation of a constant q spectral transform," *Journal of the Acoustical Society of America*, vol. 89, pp. 425–, 01 1991.

[10] M. Müller, *Fundamentals of Music Processing*. Springer International Publishing, 2015.

[11] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, 1st ed., ser. Adaptive Computation and Machine Learning. The MIT Press, 2012.