Music Genre Classification: Comparative Analysis of Various Classification Techniques

1st Matteo Fare'
University of Milan
Department of Computer Science
Milan, Italy

***Abstract*—Due to the characteristics of genres itself, that do not imply a rigid specification of the various types of the operas, the automatic classification is not a trivial task.**
The following work aims to perform a classification based on musical genre and to do so adopts first an unsupervised learning technique, the k-means algorithm to clustering data, and subsequently different classification algorithms are compared in order to analyze their performance.

***Index Terms*—Classification, Clustering, Feature Extraction, MIR, Music Genre Classification, K-Means, Multilayer Perceptron, Random Forest, K-Nearest Neighbors, Support Vector Machine.**

I. Introduction

Classification is an intrinsic activity of the human being and its purpose is to harness meaning and arrange entities within a given domain of knowledge. It applies to many fields and music is one of them.

Since the beginning, the automatic classification of music has proven to be a complex task due to the fluid nature of its categories, that does not follow a rigorous definition. Indeed, musical genres are conventional categories used to identify musical works and are built on complex constructs based on public, marketing, historical and cultural factors, and therefore destined to change or adapt over time [1]. The automatic music classification process is of crucial importance today, thanks to the increase in systems and platforms that make use of digital content and services like Spotify for example. The usefulness of the automatic recognition of musical genres lies, for example, in the auto-tagging, in the automatic cataloging of enormous quantity of data and in the creation of personalized playlists upon user request through filters and parameters.

The project presented in this document is developed by working with the famous GTZAN database, which was built approximately between the years 2000-2001 by G.Tzanetakis and P.Cook in order to publish one of the first studies regarding music classification in 2002 [2]. The primary aim of the study is to perform the classification of 10 distinct musical genres, followed by an evaluation of the results and to do so involves the use of the clustering algorithms K-Means to effectively delineate the feature space, providing a graphical representations that visually depict the classification outcomes. The computed and extracted features were fed into multiple classifiers of different types, specifically a Multilayer perceptron network, an ensemble learning methods called random forest, the Support-Vector Machine or SVM algorithm, and finally another supervised learning method called K-nearest neighbors or K-NN, a well-established supervised learning approach known for its ability to identify similarities among data points.

II. Project Overview

The architecture of the project is structured basically in five blocks, in the first two we have the processes that can be sum up as the preprocessing functions. Then there is the computation and subsequent extraction of features. Finally, the last two parts corresponds to the clustering and classification blocks with their respective evaluation.

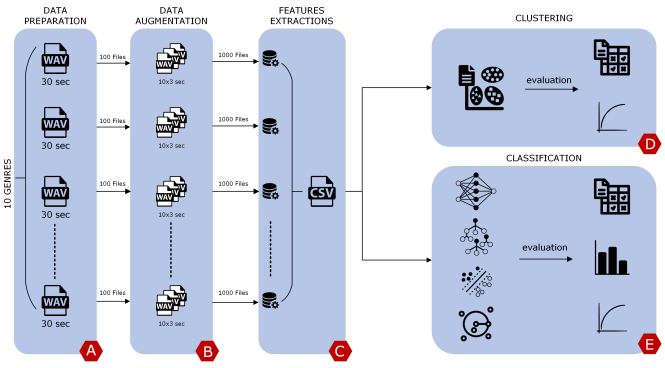

Fig. 1. High level architecture

A. Data Preparation

The first step is to retrieve the data set, which in this case is the GTZAN genre collection data-set, containing 1000 audio files each lasting 30 seconds. The data set contains 10 classes that represent 10 music genres, and each class contains 100 audio tracks. With the data set at our disposal, the preprocess phase begins, performing a check on the files (extension and duration).

B. Data Augmentation

In order to have more data to work with, data augmentation was performed, in particular in one of its simplest way, which is by splitting the data in multiple sub-samples. So the process starts with 100 samples that are 30 seconds long and ends with 1000 samples that are 3 seconds long.

C. Feature Extraction

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It is important to identify and describe that kind of features that are relevant for a given problem and implementing a way to extract those features.

1) **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:

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

2) **Entropy of Energy**: 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 "(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:

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

3) **Tempo**: 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.

4) **Root Mean Square Energy (RMSE)**: It is related to perceived sound intensity, so RMS Energy can be used for loudness estimation and as an indicator for new events in audio segmentation. As shown in "(3)", W_L represents frame size, i.e., the number of samples in each frame and $s_i(k)$ the signal.

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

5) **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:

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

6) **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.

7) **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.

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

8) **Spectral Bandwith**: 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.

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

9) **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_{th} DFT coefficient corresponds to the spectral rolloff of the i_{th} frame, then it satisfies the following "(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.

$$Z(i) = \frac{1}{2W_L} \cdot \sum_{n=1}^{W_L} |sgn[s_i(n)] - sgn[s_i(n-1)]| \tag{7}$$

10) **Mel Frequecy Cepstral Coefficient (MFCCs)**: 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.

D. Clustering

Clustering is an unsupervised learning technique that try to group a set of objects in such a way that objects in the same cluster, i.e. in the same group, are more similar to each other than those in other clusters. The principal characteristic of unsupervised learning techniques is that of automatically extracting knowledge from the input data, without this

knowledge coming from a specific understanding of the analyzed data. In this project the K-Means algorithm was used, one of the main ones in this category. The algorithm works by iteratively assigning data points to the nearest centroid and then updating the centroids based on the new cluster assignments. The process continues until the centroids no longer move significantly or a maximum number of iterations is reached.

Algorithm 1: K-means

```
[1] Input: Set of data items, desired number of clusters K
[2] Output: Set of K clusters
[3] Select K points as the initial centroids;
[4] repeat
[5] |   Form K clusters by assigning all points to the closest
    |     centroids;
[6] |   compute the centroids of each cluster;
    until The centroids don't change;
[7] return
```

E. Classification

At this point of the project, the classification of the previously extracted characteristics takes place, with the aid of various classification methods. In our case we decided to use four of them described below.

1) **Multilayer Perceptron (MLP)**: The perceptron is inspired by the brain's most basic unit of thinking, the neurons. Like neurons in our brain a Multilayer perceptron reflects its organization, with every neuron connected with each others. consists of three types of layers—the input layer, output layer and hidden layer. An arbitrary number of hidden layers that are placed in between the input and output layer are the true computational engine of the MLP. Every neuron in the hidden layer is connected with the neurons of the next layer. The connecting wires between the neurons are known as weights whose values are updated with the help of the learning phase. The learning phase is continuously repeated until the value of the error will be less than the threshold level. The input layer is the combination of the values of the features. The output layer will predict the classification which is based on the information which is passed on by the input layer. The classified output compares with the observed one and calculates the error. According to the error, network weights are updated from the output layer toward the input layer through the intermediate layer. Transmitted information can be calculated by the combination of the connecting weights, node value, and activation function.

2) **K-Nearest Neighbors (KNN)**: The K-Nearest Neighbors algorithm, also known as KNN, is a supervised learning classifier, which uses proximity to make classifications about the grouping of an individual data point. The algorithm collects data from a training data set, and uses this data later to make predictions for new records and for each new record, the k-closest records of the training data set are determined. Based on the value of the target attribute of the closest records, a prediction is made for the new record. The basic nearest neighbor NN algorithm algorithm makes classification for an arbitrary instance and identifies a training instance that is closest to the it. Then, the NN algorithm returns the class label of the training instance as the predicted class label for the arbitrary instance. The KNN algorithm expands this process by using a specified number k1 of the closest training instances instead of using only one instance. Typical values range from 1 to several dozens. In KNN classification, the predicted class label is determined by the voting for the nearest neighbors, that is, the majority class label in the set of the selected k instances is returned.

3) **Random Forest**: random forest algorithm combines the output of multiple decision trees to reach a single result. It is made up of a collection of decision trees, and each tree in the ensemble is comprised of a data sample drawn from a training set with replacement. Of that training sample, one-third of it is set aside as test data, known as the out-of-bag (oob) sample. Another instance of randomness is then injected through feature bagging, adding more diversity to the data set and reducing the correlation among decision trees. The determination of the prediction for a classification task correspond to a majority vote, i.e. the most frequent categorical variable, will yield the predicted class. Finally, the oob sample is then used for cross-validation, finalizing that prediction.

4) **Support Vector Machine (SVM)**: Support vector machine is a supervised learning algorithm that have shown great performances at binary classification tasks and better result than other methods with large dimensional features. The objective of support vector is to find a hyperplane in a N-dimensional space, with N that represent the number of features, that distinctly classifies the data points. To separate the classes of data points in an appropriate way, the objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. By default the algorithms works only on binary classification, so it is necessary a way to extend it for multiclass classification. One of the possibility is to breaking down the problem into multiple binary classification problems per each pair of classes, i.e. the One-vs-One or ovo. Another approach is called One-vs-Rest or ovr, that splits a multiclass classification into one binary classification problem per class.

In order to evaluate the various models described above, two main methods were used: the confusion matrix and the ROC curve.

1) **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: True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN).

2) **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%.

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

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

III. Project Steps and Results

The first step during the practical construction of the project was to check the data set at our disposal and perform pre-processing to prepare the data for further steps. In particular, it has been checked that the audio files were encoded with the desired extension, i.e. the Waveform Audio File Format or WAV, and that their duration corresponded to 30 seconds. Having resolved sporadic errors in the data set, we moved on to the data augmentation procedure, where the 30 second long files were divided into 3 second long chunks. Through the check on the files duration carried out previously, a numerical discrepancy between the different categories of musical genres was avoided, thus obtaining 10 classes of 1000 samples each. The next step in the preparation of our data set was the one about the definition of methods to compute and extract the characteristics from the audio samples and inset them into a CSV file. To obtain this result, most of the work was completed using the Librosa library. Once the file containing all the features was obtained, it was possible to proceed with the execution and subsequent evaluation of the two types of classification, the unsupervised learning system and the supervised learning system.

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 dimensionality reduction.
  5) Plot clusters and centroids on 2D matrix.
  6) Plot confusion matrix.
  7) Plot ROC curve.
  8) Silhouette score analysis.
- Supervised Learning:
  1) Load data.
  2) Split data for train and test.
  3) Calculate correlation matrix.
  4) Load models
  5) Plot confusion matrix.
  6) Plot ROC curve.
  7) Plot prediction comparison.
  8) Compute evaluation metrics.

To understand and compare the results obtained from the different classification models, several data graphs and tables have been defined. For example, as it is possible to see below in Fig. 2, with the large number of cluster to be analyzed, the K-means algorithm is not performing very well, and this is further verifiable by looking at the confusion matrix results in Fig. 3. It is possible to evaluate the most appropriate value of cluster thanks to the Silhouette Algorithm [3].

Unsupervised Learning

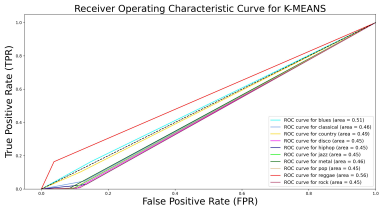

Fig. 2. K-Means ROC Curve

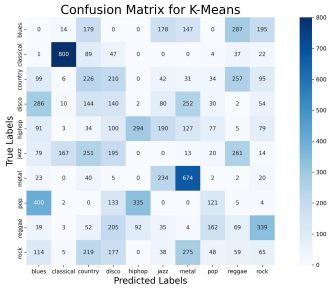

Fig. 3. K-Means Confusion Matrix

As it is possible to verify below in Figs. 4, 5, 6 and 7, compared to the performance of the K-means algorithm, better results were obtained by the supervised learning algorithms.

Supervised Learning - ROC Curve

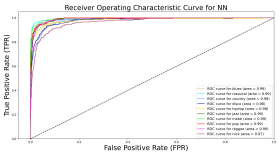

Fig. 4. NN

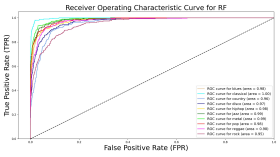

Fig. 5. RF

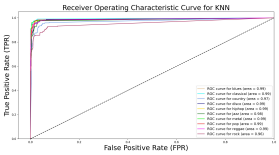

Fig. 6. KNN

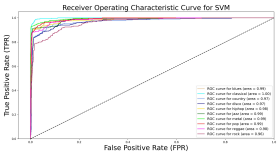

Fig. 7. SVM

Several metrics were calculated to better display the performance of each classifiers.

1) **Accuracy**: Accuracy is a metric that summarizes the performance of a classification model as the number of correct predictions divided by the total number of predictions.

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FP} \tag{10}$$

2) **Precision**: Precision (positive predictive value) should ideally be 1 (high) for a good classifier. Precision becomes 1 only when the numerator and denominator are equal i.e TP = TP + FP, this also means FP is zero. As FP increases the value of denominator becomes greater than the numerator and precision value decreases (which we don't want).

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

3) **Recall**: Recall, also known as sensitivity or true positive rate, should ideally be 1 (high) for a good classifier. Recall becomes 1 only when the numerator and denominator are equal i.e TP = TP +FN, this also means FN is zero. As FN increases the value of denominator becomes greater than the numerator and recall value decreases (which we don't want).

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

4) **F1 Score**: F1-score is a metric which takes into account both precision and recall, that becomes high only when both precision and recall are high. It is the harmonic mean of Precision and Recall and is a better measure than accuracy.

$$Accuracy = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{13}$$

TABLE I
NN - EVALUATION METRICS

	Precision	**Recall**	**F1-Score**
Blues	0.91756	0.80503	0.85762
Classical	0.94649	0.91883	0.93245
Country	0.80524	0.74653	0.77477
Disco	0.81271	0.80198	0.80731
Hip hop	0.91608	0.81875	0.86469
Jazz	0.81288	0.91379	0.86039
Metal	0.90785	0.91096	0.90940
Pop	0.91513	0.84932	0.88099
Reggae	0.79088	0.91049	0.84648
Rock	0.68730	0.79623	0.73776
Accuracy			**0.84800**

TABLE II
RF - EVALUATION METRICS

	Precision	**Recall**	**F1-Score**
Blues	0.87266	0.73270	0.79658
Classical	0.90432	0.95130	0.92722
Country	0.60947	0.71528	0.65815
Disco	0.71739	0.76238	0.73920
Hip hop	0.88806	0.74375	0.80952
Jazz	0.79751	0.88276	0.83797
Metal	0.80119	0.92466	0.85851
Pop	0.84228	0.85959	0.85085
Reggae	0.78354	0.79321	0.78834
Rock	0.73604	0.54717	0.62771
Accuracy			**0.79333**

TABLE III
KNN - EVALUATION METRICS

	Precision	**Recall**	**F1-Score**
Blues	0.93667	0.88365	0.90939
Classical	0.90214	0.95779	0.92913
Country	0.85866	0.84375	0.85114
Disco	0.85671	0.92739	0.89065
Hip hop	0.94915	0.87500	0.91057
Jazz	0.85953	0.88621	0.87267
Metal	0.96167	0.94521	0.95337
Pop	0.93706	0.91781	0.92734
Reggae	0.89086	0.93210	0.91101
Rock	0.82422	0.79623	0.80998
Accuracy			**0.89800**

TABLE IV
SVM - EVALUATION METRICS

	Precision	**Recall**	**F1-Score**
Blues	0.89908	0.92453	0.91163
Classical	0.93610	0.95130	0.94364
Country	0.82828	0.85417	0.84103
Disco	0.83758	0.86799	0.85251
Hip hop	0.90123	0.91250	0.90683
Jazz	0.89632	0.92414	0.91002
Metal	0.95819	0.94178	0.94991
Pop	0.94485	0.88014	0.91135
Reggae	0.92810	0.87654	0.90159
Rock	0.80460	0.79245	0.79848
Accuracy			**0.89400**

From the data reported in Tables I, II, III and IV and from the data summary presented in Table V, the algorithm that presents the best performance appears to be KNN algorithm, which not only manages to classify the different genres very efficiently but also also quite quickly.

TABLE V
CLASSIFIERS PERFORMANCES SUMMARY

	NN	RF	KNN	SVM
Accuracy	84.80	79.33	**89.80**	89.40
RMSE	1.80	1.99	**1.44**	1.54
F1 Score	0.85	0.79	**0.90**	0.89
Execution Time	63.72	52.76	**7.73**	21.40

IV. CONCLUSION

In this project it was possible to analyze the behavior of unsupervised and supervised learning techniques, with the aim of classifying the content of the famous GTZAN dataset of music genres. To carry out this task, numerous characteristics have been extracted from the available data and the results obtained show that there is a clear performance gap between the unsupervised model adopted, i.e. the K-Means algorithm, and the four supervised classification algorithms used, among which the KNN turns out to be the best. Some future developments to broaden this work could concern the efficiency of some algorithms, for example as regards the neural network, it is certainly possible to add neurons, modify the topology and perform a very deep tuning, increasing the quality of the result, but probably also the execution time and computational complexity. Another point of reflection and development concerns the review and improvement of the unsupervised learning technique, whose performance should be improved, for example through the implementation of an algorithm such as CNN.

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] 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