

Report

**Homework 1**

*April 26, 2021*

**Group 8**

Personal codes of the group components: 10805780 – 10582571 – 10782704 – 10531049

**Presentation of the work**

In our homework, we implemented a classifier able to predict the musical genreof a music piece according to some specific metrics.In order to do this, we used and downloaded the dataset GTZAN Genre Collectionat the following link ([www.kaggle.com/carlthome/gtzan-genre-collection](http://www.kaggle.com/carlthome/gtzan-genre-collection)). It is simply a collection of songs, divided in subfolders depending on the genre. It includes 1000 songs for 10 genres at 22050 Hz sampling rate.

As assigned, we classified only 4 genres among the 10 available in the dataset. The genres classified are: Classical, Disco, Jazz, Country.

The results of the classification have been reported as a confusion matrixand also other metrics of your choice.

This report illustrates and explains every step of your classification system and the results are shown and commented.

The link to our GitHub repository containing the commented code is available at the following link: [www.github.com/enricovalente23/CMLS-Homework-1-Group-8.git](https://github.com/enricovalente23/CMLS-Homework-1-Group-8.git).

The general pipeline we followed during our work was composed of these steps:

* **Collect data** from the folders of the original dataset;
* **Retrieve the annotations**:sometimes annotations are implicit in the organization of the folders of the dataset, sometimes are expressed in the name of the file, sometimes are in a separated CSV file;
* If necessary, **apply some preprocessing** to audio files;
* Split your dataset in **training set** and **testing set**: what proportion would you use in this split? Can you add some constraints such that you are sure that training set and testing set are as different as possible? For classification, check also if the dataset is balanced, i.e. it has a similar number of samples for each class. If it is not, you can decide either to balance it by sub- sampling or, in the evaluation part, to use metrics suitable for un-balanced datasets. You can also decide to apply cross-validation (k-fold for example) in your project;
* **Extract the features**: think which set of features can be useful for your task, test different parameters for computing them and think how they should be aggregated over time. Depending on the task it might be better to aggregate all the features in time (e.g. taking statistical moments) and estimate a single label/value per audio track **or** predict a label/value for each time window and then aggregate all of them in a single result per audio track;
* Apply **feature selection methods**;
* Choose one **classification/regression method**: fit the model on your training set and use the testing set for the evaluation of your method. If you want, repeat the fit with different parameters for your classifier/regressor for fine tuning your model. In the evaluation part, choose which metrics are meaningful for your task (maybe more than one!).

A first option to **extract features** can be to implement from scratch the features you need. Alternatively, the libraries we used were **Librosa (**<https://librosa.github.io>) and **Essentia** (<https://essentia.upf.edu/documentation.html>) using the Python bindings.

In order to **define and train** our models, we found several classifiers, regressors, feature selection algorithms implemented in the **scikit-learn** library (<https://scikit-learn.org/stable/>).

We also plotted the distribution of our data. This helped us in figuring out which were the best design choices for your homework.

**Description of the work**