
**Degree on Computer Science and Engineering – 2021/2022, 3rd year, 2nd semester**
Department of Informatics Engineering, University of Coimbra

# Multimedia

## Practical work no 2

### Multimedia Information Retrieval

## Report

Sancho Amaral Simões | 2019217590 Tiago

Filipe Santa Ventura |2019243695

Rui Bernardo Lopes Rodrigues | 2019217573

Coimbra, 7th April 2022

# 1. Index

# 2. Goal

The goal of the present report is to explain and summarize the work done in the practical project number two within the scope of the *Multimedia* subject of the Degree on Computer Science and Engineering at the University of Coimbra. The main goal of the project consists in the consolidation and application of fundamental concepts of *Multimedia*, such as information retrieval, initially acquired in theoretical/theoretical-practical *Multimedia* classes.

To put this knowledge into practice, the *Python* programming language and some of its available modules, such as *Numpy, Scipy and Scikit-learning,* were used in order to build a content-based music recommendation system.

This report, in addition to briefly explaining the concepts related with the work in question, will make known the results obtained and interpretations formed regarding the resolution of the exercises in practical sheet 1, carried out over six weeks of work.

# 3. Introduction

## 3.1. Multimedia Information Retrieval

Multimedia information retrieval (*MMIR* or *MIR*) is a research discipline of computer science that aims at extracting semantic information from multimedia data sources. Data sources include directly perceivable media such as audio, image and video, indirectly perceivable sources such as text, semantic descriptions, bio signals as well as not perceivable sources such as bioinformation, stock prices, etc. The methodology of *MMIR* can be organized in three groups:

1. Methods for the summarization of media content (feature extraction). The result of feature extraction is a description.
2. Methods for the filtering of media descriptions (for example, elimination of redundancy).
3. Methods for the categorization of media descriptions into classes.

## 3.2. Feature Extraction Methods

Feature extraction is motivated by the sheer size of multimedia objects as well as their redundancy and, possibly, noisiness. Generally, two possible goals can be achieved by feature extraction:

- Summarization of media content. Methods for summarization include in the audio domain, for example, mel-frequency cepstral coefficients, Zero Crossings Rate, Short-Time Energy.
- Detection of patterns by autocorrelation and/or cross-correlation. Patterns are recurring media chunks that can either be detected by comparing chunks over the media dimensions (time, space, etc.) or comparing media chunks to templates (e.g., face templates, phrases).

## 3.3. Categorization Methods

Generally, all forms of machine learning can be employed for the categorization of multimedia descriptions: though some methods are more frequently used in one area than another. The list of applicable classifiers includes the following:

- Metric approaches (Cluster analysis, vector space model, *Minkowski* distances, dynamic alignment);
- Nearest Neighbor methods (K-nearest neighbors' algorithm, K-means, self-organizing map);
- Risk Minimization (Support vector regression, support vector machine, linear discriminant analysis);
- Density-based Methods (*Bayes* nets, *Markov* processes, mixture models);
- Neural Networks (Perceptron, associative memories, spiking nets);
- Heuristics (Decision trees, random forests, etc.).

## 3.4. Features

Instead of using the above-mentioned methods, which involve a much deeper understanding of how machine learning works and how it can be applied to *MMIR*, the work group instead chose a much more traditional, simpler and well-studied methodology: feature extraction. This implies the compression of the large datasets (audio buffers) into arrays of numerical values that represent the way the data is characterized in the time, frequency and other domains. Some of these numerical values are formally known as:

- Spectral features: *MFCC*, spectral centroid, spectral bandwidth, spectral contrast, spectral flatness and spectral roll-off.

- Temporal Features: fundamental frequency, *RMS* and zero crossing rate.
- And a non-listed feature: Tempo.

As introduced above, there are two types of features that characterize an audio signal:

- The temporal features (time domain features), which are simple to extract and have easy physical interpretation, like: the energy of signal, zero crossing rate, maximum amplitude, minimum energy, etc.
- The spectral features (frequency-based features), which are obtained by converting the time-based signal into the frequency domain using the Fourier Transform, like: fundamental frequency, frequency components, spectral centroid, spectral flux, spectral density, spectral roll-off, etc. These features can be used to identify the notes, pitch, rhythm, and melody.

### 3.4.1. MFCC

The short-term power spectrum of any sound can be represented by the Mel frequency cepstral coefficients (*MFCCs*). It can be derived from a type of inverse Fourier transform (this is where the anagram cepstral comes from) representation. The *MFCCs* allow a better representation of sound because the frequency bands are equally distributed on the Mel scale, which approximates the human auditory system's response more closely.

The *MFCCs* can be derived by mapping the *Fourier* transformed signal onto the male scale using triangle or cosine overlapping windows. Where after taking the logs of the powers at each of the Mel frequencies and after discrete cosine transform of the Mel log powers give the amplitude of a spectrum. The amplitude list is MFCC.

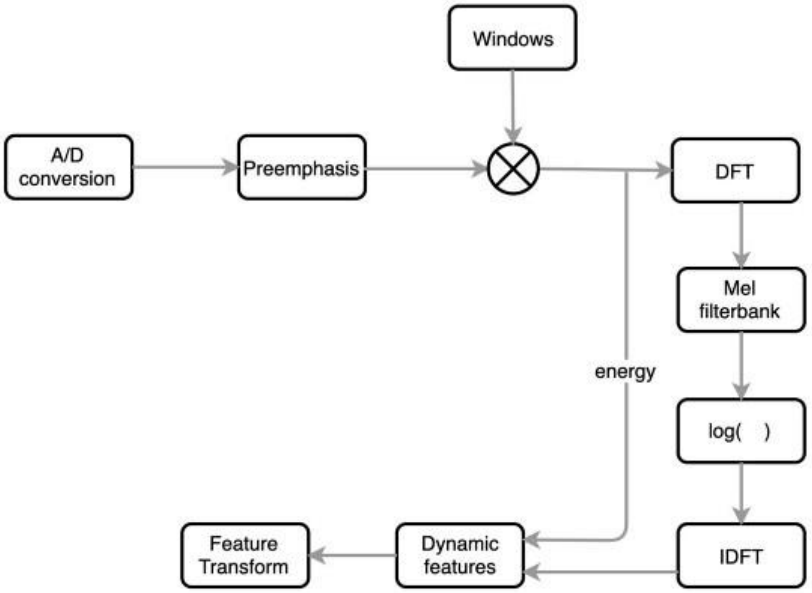



Fig.1 - MFCC Technique Road Map

### 3.4.2. Spectral Centroid

A spectral centroid is the location of the center of mass of the spectrum. Since the audio files are digital signals and the spectral centroid is a measure that can be useful in the characterization of the spectrum of the audio file signal.

In some places, it can be considered as the median of the spectrum but there is a difference between the measurement of the spectral centroid and median of the spectrum. The spectral centroid is like a weighted median and the median of the spectrum is similar to the mean. Both of them measure the central tendency of the signal. In some cases, both of them show similar results.

Mathematically it can be determined by the Fourier transform of the signals with the weights.

$$C_i = \frac{\sum_{k=1}^{Wf_L} kX_i(k)}{\sum_{k=1}^{Wf_L} X_i(k)}$$

### 3.4.3. Spectral Bandwidth

Bandwidth is the difference between the upper and lower frequencies in a continuous band of frequencies. As we know the signals oscillate about a point so if the point is the centroid of the signal, then the sum of maximum deviation of the signal on both sides of the point can be considered as the bandwidth of the signal at that time frame.

### 3.4.4. Spectral Contrast

In an audio signal, the spectral contrast is the measure of the energy of frequency at each timestamp. Since most of the audio files contain the frequency, whose energy is changing with time. It becomes difficult to measure the level of energy. Spectral contrast is a way to measure that energy variation.

High contrast values generally correspond to clear, narrow-band signals, while low contrast values correspond to broad-band noise.

### 3.4.5. Spectral Flatness

White noise has a flat power spectrum. So, a reasonable way to measure how close a sound is to being pure noise is to measure how flat its spectrum is.

Spectral flatness is defined as the ratio of the geometric mean to the arithmetic mean of a power spectrum. The arithmetic mean of a sequence of n items is what you usually think of as a mean or average: add up all the items and divide by n. The geometric mean of a sequence of n items is the nth root of their product. You could calculate this by taking the arithmetic mean of the logarithms of the items, then taking the exponential of the result. What if some items are negative? Since the power spectrum is the squared absolute value of the FFT, it can’t be negative.

### 3.4.6. Spectral Roll-off

It can be defined as the action of a specific type of filter which is designed to roll off the frequencies outside to a specific range. The reason we call it roll-off is because it is a gradual procedure. There can be two kinds of filters: hi-pass and low pass and both can roll off the frequency from a signal going outside of their range.

More formally we can say the spectral roll-off point is the fraction of bins in the power spectrum at which 85% of the power is at lower frequencies

This can be used for calculating the maximum and minimum by setting up the roll percent to a value close to 1 and 0.

### 3.4.7. Zero Crossing Rate

As the name suggests zero-crossing rate is the measure of the rate at which the signal is going through the zeroth line more formally signal is changing positive to negative or vice versa.

Mathematically it can be measured as:

$$zcr = \frac{1}{T-1}\sum_{t=1}^{T-1} 1_{\mathbb{R}_{<0}}(s_t s_{t-1})$$

Where:

s = signal
T = length of signal

It can be used for pitch detection algorithms and also for voice activity detection.

### 3.4.8. RMS (Root Mean Square)

*RMS* is a meaningful way of calculating the average of values over a period of time. With audio, the signal value (amplitude) is squared, averaged over a period of time, then the square root of the result is calculated. The result is a value, that when squared, is related (proportional) to the effective power of the signal.

Unfortunately, calculating the RMS value of anything but a simple sine wave is very difficult. The further a signal gets in harmonic content from a sine wave, the less accurate *RMS* values will be. For a dynamic signal like most music, it is nearly impossible to get even close to a true *RMS* value.

### 3.4.9. F0 (Fundamental Frequency)

The fundamental frequency of a speech signal, often denoted by F0 or F0, refers to the approximate frequency of the (quasi-)periodic structure of voiced speech signals. The oscillation originates from the vocal folds, which oscillate in the airflow when appropriately tensed. The fundamental frequency is defined as the average number of oscillations per second and expressed in Hertz. Since the oscillation originates from an organic structure, it is not exactly periodic but contains significant fluctuations. In particular, amount of variation in period length and amplitude are known respectively as jitter and shimmer. Moreover, the F0 is typically not stationary, but changes constantly within a sentence. In fact, the F0 can be used for expressive purposes to signify, for example, emphasis and questions.

$$T = \frac{1}{F_0}$$

### 3.4.10. Tempo

Tempo refers to the speed of a musical piece. More precisely, tempo refers to the rate of the musical beat and is given by the reciprocal of the beat period. Tempo is often defined in units of beats per minute (BPM).

## 3.5. Similarity Measure

In statistics and related fields, a similarity measure or similarity function or similarity metric is a real-valued function that quantifies the similarity between two objects. Although no single definition of a similarity exists, usually such measures are in some sense the inverse of distance metrics: they take on large values for similar objects and either zero or a negative value for very dissimilar objects. Though, in more broad terms, a similarity function may also satisfy metric axioms.

Cosine similarity is a commonly used similarity measure for real-valued vectors, used in (among other fields) information retrieval to score the similarity of documents in the vector space model. In machine learning, common kernel functions such as the RBF kernel can be viewed as similarity functions.

### 3.5.1. Euclidian Distance

The basis of many measures of similarity and dissimilarity is Euclidean distance. The distance between vectors X and Y is defined as follows:

$$d(x,y) = \sqrt{\sum_{i}^{n} (x_i - y_i)^2}$$

In other words, Euclidean distance is the square root of the sum of squared differences between corresponding elements of the two vectors. Note that the formula treats the values of X and Y seriously: no adjustment is made for differences in scale. Euclidean distance is only appropriate for data measured on the same scale. As you will see in the section on correlation, the correlation coefficient is (inversely) related to the Euclidean distance between standardized versions of the data.

Euclidean distance is most often used to compare profiles of respondents across variables. For example, suppose our data consist of demographic information on a sample of individuals, arranged as a respondent-by-variable matrix. Each row of the matrix is a vector of m numbers, where m is the number of variables. We can evaluate the similarity (or, in this case, the distance) between any pair of rows. Notice that for this kind of data, the variables are the columns. A variable record the results of a measurement.

### 3.5.2. Manhattan Distance

Manhattan distance is a metric in which the distance between two points is calculated as the sum of the absolute differences of their Cartesian coordinates. In a simple way of saying it is the total sum of the difference between the x-coordinates and y-coordinates.

Suppose we have two points A and B. If we want to find the Manhattan distance between them, just we have, to sum up, the absolute x-axis and y-axis variation. This means we have to find how these two points A and B are varying in X-axis and Y-axis. In a more mathematical way of saying Manhattan distance between two points measured along axes at right angles.

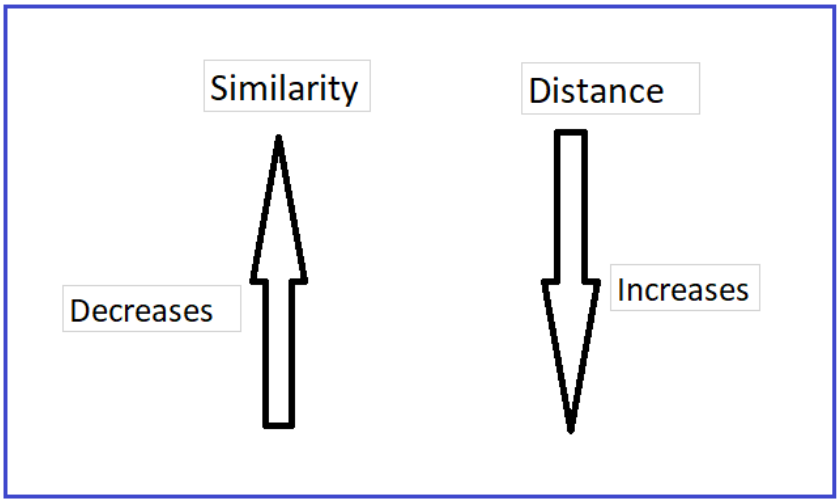
$$d(x,y)=\sum_{i=1}^{N}|x_i-y_i|$$

### 3.5.3. Cosine Distance

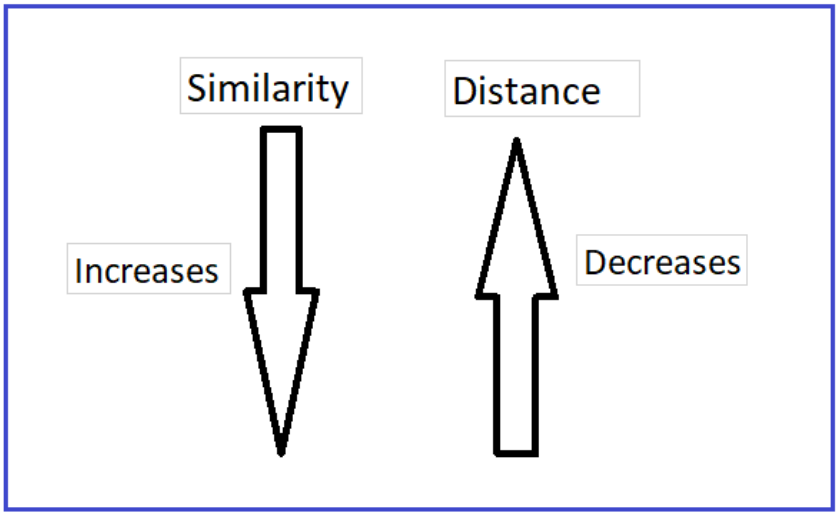
Cosine similarity is used to determine the similarity between documents or vectors. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. There are other similarity measuring techniques like Euclidean distance or Manhattan distance available but we will be focusing here on the Cosine Similarity and Cosine Distance.

The relation between cosine similarity and cosine distance can be define as below.

Similarity decreases when distance between two vectors increases:




Similarity increases when distance between two vectors decreases:



# 4. Procedure

In this practical work the main goal is to apply all the knowledge acquired during the *Multimedia* classes, especially information retrieval, to create a music recommendation program. This goal was achieved throughout six weeks of group-work, investigation, analysis and experimentation. Below is presented the timeline of all the work done in order to accomplish this goal:

**Week 1**

- Creation of a *Git* repository in order to enhance the workflow – https://github.com/smartlord7/MMIRMusicRecommender.

- Analyze other recommendation systems like *Spotify*, *Jango.com*, etc.
- Perceptual analysis of the dataset: audio files, metadata, features.
- Installation of the framework (*librosa* and *ffmpeg*).
- Analysis of the base code given.
- Study of the framework documentation.

**Week 2**

- Implementation of a function to process all the features in the file *top100_features.csv*.
- Implementation of several functions that will extract all the different features (Spectral, Temporal and others) of the *librosa* framework and stores the information in a 2D numpy array.

**Week 3**

- Implementation of a function that calculates the statistics of the *librosa* features.
- Implementation of a function that will create and save on a file the *numpy* array with the extracted features.

**Week 4**

- Research on the different similarity metrics.
- Implementation of functions that will calculate the different similarity metrics.

**Week 5**

- Optimization of all the code done until now.
- Implementation of a function responsible to create and save the similarity matrices.
- Implementation of a function that will get the ranking of the 20 recommended songs based on the metadata.
- Implementation of a function that creates and saves the similarity matrix based on the metadata.
- Implementation of a function that will calculate the metric precision.

**Week 6**

- Subjective evaluation was performed based on the context data.
- Calculation of the mean and standard deviation of all the members by query and globally for the 4 queries.
- Implementation of the features made from scratch (Bonus).
- Code refactors in order to improve its modularity and organization.
- Code documentation and formatting.
- Final adjustments and tuning.

It is relevant to say that the redaction of this report was continuous and carried out along with the development of the code developed in the scope of this project.

# 5. Discussion and results

## 5.1. *A priori* work

In order to gain a certain understanding of what was ahead, the group started this project by looking at others music recommendation systems like *Spotify*, *Jango.com*, *Last.fm*, *Torch*, etc. Next, came the environment preparation: download of the database of audios, all the metadata and the features, installation of the *librosa* and *ffmpeg* libraries and study and analysis of the base code as well as their documentation.

## 5.2. Feature Extraction

### 5.2.1. Existing Features Processing

In order to process all the features in the file *top100_features.csv*, a *numpy* matrix was created with the given features, normalized in the [0, 1] interval. Then, the procedure was to save that matrix into a *.csv* file.

### 5.2.2. *Librosa* Features

The group started by extracting all the 900 files from the database into a new folder*, database/all*, making it easier to access them all. After that, the feature extraction process follows: the library *librosa* was used in order to make this process easier, and also because the goal of this project was not to learn how to implement each different feature but rather how to use it in the *MMIR* domain.

### 5.2.3. Statistics

In order to compress even more the data outputted by the features computations, some statistics were applied to each feature, using some functions from the *spicy.stats.skew* and *numpy* libraries that were applied in order to get the mean, standard deviation, skewness, kurtosis, median, maximum and minimum. After that, the only thing left was to normalize all the features into the [0, 1] interval and save the resulting *numpy* matrix into a *.csv* file.

### 5.2.4. Features implementation from scratch (Bonus)

Even though the necessary features were already implemented in the *librosa* library, the group accepted the professor challenge to program those features from scratch, using only libraries like *numpy*, *scipy*, … This required some more profound knowledge about those features, specifically aspects concerning the area of discrete time series transforms such as *DFT* and *DCT* and the manipulation of the data in question in the time and frequency domains.

## 5.3. Implementation of the similarity metrics

In this project three types of similarity metrics were used and in order to implement them, first they

have to be calculated.

This process was considerably simple, thanks to the *scipy.spacial.distance* library. For each distance (*euclidian*, *manhattan* and *cosine*) there was just need a single line of code. In case of the *manhattan* and *cosine* distances they were calculated with the use of the *cityblock* and *cosine* metrics, respectively. As for the euclidian distance it's calculated using the sum of squared differences between two arrays.

After the distances are calculated, it's time to do the similarity matrices, one for each feature set and distance metric. And as the final touch the creation of the similarity rankings for each one of the four given queries, each with 20 songs.

### Ranking results based on "Euclidean" distance (librosa) features:

| | | | |
|---|---|---|---|
| 1 - MT0000202045.mp3 (0.0000) | 1 - MT0000379144.mp3 (0.0000) | 1 - MT0000956340.mp3 (0.0000) | 1 - MT0000414517.mp3 (0.0000) |
| 2 - MT0033841575.mp3 (1.0519) | 2 - MT0009010830.mp3 (1.1429) | 2 - MT0003106472.mp3 (1.0958) | 2 - MT0003900455.mp3 (1.0838) |
| 3 - MT0027002641.mp3 (1.1503) | 3 - MT0003778826.mp3 (1.1592) | 3 - MT0010736208.mp3 (1.1493) | 3 - MT0000203193.mp3 (1.2476) |
| 4 - MT0030487841.mp3 (1.1583) | 4 - MT0015005100.mp3 (1.1815) | 4 - MT0014703649.mp3 (1.1519) | 4 - MT0009897495.mp3 (1.2549) |
| 5 - MT0008575372.mp3 (1.1830) | 5 - MT0028627699.mp3 (1.1950) | 5 - MT0005409948.mp3 (1.1908) | 5 - MT0009521580.mp3 (1.2609) |
| 6 - MT0014576739.mp3 (1.1995) | 6 - MT0012124855.mp3 (1.2142) | 6 - MT0014615863.mp3 (1.1932) | 6 - MT0018651126.mp3 (1.3025) |
| 7 - MT0030422114.mp3 (1.1998) | 7 - MT0000044741.mp3 (1.2204) | 7 - MT0013612461.mp3 (1.3238) | 7 - MT0013955066.mp3 (1.3045) |
| 8 - MT0003390733.mp3 (1.2135) | 8 - MT0005478759.mp3 (1.2293) | 8 - MT0004032071.mp3 (1.3573) | 8 - MT0034186195.mp3 (1.3250) |
| 9 - MT0027835071.mp3 (1.2254) | 9 - MT0027035970.mp3 (1.2399) | 9 - MT0010615428.mp3 (1.3623) | 9 - MT0000901959.mp3 (1.3297) |
| 10 - MT0009188643.mp3 (1.2575) | 10 - MT0003794106.mp3 (1.2399) | 10 - MT0003724610.mp3 (1.3666) | 10 - MT0004428604.mp3 (1.3533) |
| 11 - MT0010617945.mp3 (1.2768) | 11 - MT0000992846.mp3 (1.2575) | 11 - MT0040033011.mp3 (1.3718) | 11 - MT0009346128.mp3 (1.3602) |
| 12 - MT0011145388.mp3 (1.2806) | 12 - MT0031996897.mp3 (1.2676) | 12 - MT0002372242.mp3 (1.3728) | 12 - MT0000888329.mp3 (1.3612) |
| 13 - MT0009213083.mp3 (1.2829) | 13 - MT0001376988.mp3 (1.2770) | 13 - MT0004882280.mp3 (1.3760) | 13 - MT0000218346.mp3 (1.3649) |
| 14 - MT0005331755.mp3 (1.3003) | 14 - MT0009208842.mp3 (1.2771) | 14 - MT0009800907.mp3 (1.3903) | 14 - MT0012331779.mp3 (1.3819) |
| 15 - MT0002233402.mp3 (1.3131) | 15 - MT0002262181.mp3 (1.2948) | 15 - MT0004293364.mp3 (1.3938) | 15 - MT0034005433.mp3 (1.3848) |
| 16 - MT0026727455.mp3 (1.3139) | 16 - MT0005737276.mp3 (1.2969) | 16 - MT0004028719.mp3 (1.4023) | 16 - MT0013633209.mp3 (1.3872) |
| 17 - MT0005265641.mp3 (1.3212) | 17 - MT0010465830.mp3 (1.2975) | 17 - MT0007535042.mp3 (1.4172) | 17 - MT0002379222.mp3 (1.3915) |
| 18 - MT0010344415.mp3 (1.3344) | 18 - MT0018031959.mp3 (1.3252) | 18 - MT0030036616.mp3 (1.4276) | 18 - MT0001703346.mp3 (1.3930) |
| 19 - MT0000711493.mp3 (1.3478) | 19 - MT0033958450.mp3 (1.3259) | 19 - MT0014794891.mp3 (1.4281) | 19 - MT0005752234.mp3 (1.3948) |
| 20 - MT0004428604.mp3 (1.3547) | 20 - MT0012041920.mp3 (1.3282) | 20 - MT0013822237.mp3 (1.4334) | 20 - MT0017667847.mp3 (1.3979) |
| 21 - MT0018029465.mp3 (1.3611) | 21 - MT0012396528.mp3 (1.3468) | 21 - MT0010624346.mp3 (1.4446) | 21 - MT0009991160.mp3 (1.4057) |

### Ranking results based on "Manhattan" distance (librosa) features:

| | | | |
|---|---|---|---|
| 1 - MT0000202045.mp3 (0.0000) | 1 - MT0000379144.mp3 (0.0000) | 1 - MT0000414517.mp3 (0.0000) | 1 - MT0000956340.mp3 (0.0000) |
| 2 - MT0030422114.mp3 (7.6174) | 2 - MT0015005100.mp3 (8.4042) | 2 - MT0003900455.mp3 (7.9424) | 2 - MT0003106472.mp3 (6.7650) |
| 3 – MT0033841575.mp3 (7.8210) | 3 - MT0009010830.mp3 (8.4861) | 3 - MT0003949060.mp3 (8.8439) | 3 - MT0010736208.mp3 (7.6884) |
| 4 - MT0027835071.mp3 (7.9161) | 4 - MT0031996897.mp3 (8.7633) | 4 - MT0000203193.mp3 (8.9973) | 4 - MT0014703649.mp3 (7.7953) |
| 5 - MT0008575372.mp3 (8.0173) | 5 - MT0012124855.mp3 (8.9617) | 5 - MT0009521580.mp3 (9.0996) | 5 - MT0014615863.mp3 (7.8693) |
| 6 - MT0002297016.mp3 (8.3299) | 6 - MT0028627699.mp3 (9.0742) | 6 - MT0013955066.mp3 (9.1786) | 6 - MT0005409948.mp3 (7.9337) |
| 7 - MT0027002641.mp3 (8.4368) | 7 - MT0003778826.mp3 (9.1294) | 7 - MT0000218346.mp3 (9.2222) | 7 - MT0004028719.mp3 (8.6280) |
| 8 - MT0030487841.mp3 (8.5132) | 8 - MT0000992846.mp3 (9.1483) | 8 - MT0001333258.mp3 (9.4122) | 8 - MT0002372242.mp3 (8.6717) |
| 9 - MT0040033011.mp3 (8.5312) | 9 - MT0001376988.mp3 (9.1857) | 9 - MT0009897495.mp3 (9.4335) | 9 - MT0007535042.mp3 (8.7277) |
| 10 - MT0005265641.mp3 (8.6978) | 10 - MT0000044741.mp3 (9.2249) | 10 - MT0034186195.mp3 (9.4593) | 10 - MT0004032071.mp3 (8.8422) |
| 11 - MT0010617945.mp3 (8.7648) | 11 - MT0005478759.mp3 (9.2767) | 11 - MT0001703346.mp3 (9.6418) | 11 - MT0004085907.mp3 (8.8791) |
| 12 - MT0009213083.mp3 (8.7963) | 12 - MT0027035970.mp3 (9.2891) | 12 - MT0018651126.mp3 (9.6592) | 12 - MT0003724610.mp3 (8.9736) |
| 13 - MT0006096934.mp3 (8.9070) | 13 - MT0003794106.mp3 (9.4269) | 13 - MT0034005433.mp3 (9.6594) | 13 - MT0004942017.mp3 (9.2089) |
| 14 - MT0005897799.mp3 (8.9427) | 14 - MT0005737276.mp3 (9.6017) | 14 - MT0013633209.mp3 (9.7162) | 14 - MT0013613461.mp3 (9.3242) |
| 15 - MT0004428604.mp3 (9.0028) | 15 - MT0003903675.mp3 (9.6689) | 15 - MT0005469880.mp3 (9.7927) | 15 - MT0010615428.mp3 (9.3842) |
| 16 - MT0026727455.mp3 (9.0445) | 16 - MT0002262181.mp3 (9.6747) | 16 - MT0017797643.mp3 (9.8590) | 16 - MT0040033011.mp3 (9.3893) |
| 17 - MT0004867185.mp3 (9.0653) | 17 - MT0009208842.mp3 (9.7082) | 17 - MT0034125967.mp3 (9.9112) | 17 - MT0035334027.mp3 (9.4036) |
| 18 - MT0003787478.mp3 (9.1341) | 18 - MT0008222676.mp3 (9.8274) | 18 - MT0009346128.mp3 (9.9399) | 18 - MT0011697297.mp3 (9.4086) |
| 19 - MT0001340713.mp3 (9.1672) | 19 - MT0006367176.mp3 (10.1682) | 19 - MT0013161246.mp3 (9.9644) | 19 - MT0004882280.mp3 (9.4510) |
| 20 - MT0009188643.mp3 (9.1802) | 20 - MT0006540794.mp3 (10.2066) | 20 - MT0005752234.mp3 (9.9787) | 20 - MT0009800907.mp3 (9.5403) |
| 21 - MT0011145388.mp3 (9.1938) | 21 - MT0015742096.mp3 (10.2568) | 21 - MT0000040632.mp3 (9.9899) | 21 - MT0010624346.mp3 (9.6686) |

Ranking results based on “Cosine” distance (librosa) features:

| | | | |
|---|---|---|---|
| 1 - MT0000202045.mp3 (0.0000) | 1 - MT0000379144.mp3 (0.0000) | 1 - MT0000414517.mp3 (0.0000) | 1 - MT0000956340.mp3 (0.0000) |
| 2 - MT0033841575.mp3 (0.0271) | 2 - MT0009010830.mp3 (8.4042) | 2 - MT0003900455.mp3 (7.9424) | 2 - MT0003106472.mp3 (6.7650) |
| 3 - MT0027002641.mp3 (0.0320) | 3 - MT0003778826.mp3 (8.4861) | 3 - MT0003949060.mp3 (8.8439) | 3 - MT0010736208.mp3 (7.6884) |
| 4 - MT0030487841.mp3 (0.0331) | 4 - MT0015005100.mp3 (8.7633) | 4 - MT0000203193.mp3 (8.9973) | 4 - MT0014703649.mp3 (7.7953) |
| 5 - MT0008575372.mp3 (0.0338) | 5 - MT0028627699.mp3 (8.9617) | 5 - MT0009521580.mp3 (9.0996) | 5 - MT0014615863.mp3 (7.8693) |
| 6 - MT0014576739.mp3 (0.0346) | 6 - MT0012124855.mp3 (9.0742) | 6 - MT0013955066.mp3 (9.1786) | 6 - MT0005409948.mp3 (7.9337) |
| 7 - MT0030422114.mp3 (0.0356) | 7 - MT0027035970.mp3 (9.1294) | 7 - MT0000218346.mp3 (9.2222) | 7 - MT0004028719.mp3 (8.6280) |
| 8 - MT0003390733.mp3 (0.0364) | 8 - MT0000044741.mp3 (9.1483) | 8 - MT0001333258.mp3 (9.4122) | 8 - MT0002372242.mp3 (8.6717) |
| 9 - MT0027835071.mp3 (0.0366) | 9 - MT0001376988.mp3 (9.1857) | 9 - MT0009897495.mp3 (9.4335) | 9 - MT0007535042.mp3 (8.7277) |
| 10 - MT0011145388.mp3 (0.0385) | 10 - MT0000044741.mp3 (9.2249) | 10 - MT0034186195.mp3 (9.4593) | 10 - MT0004032071.mp3 (8.8422) |
| 11 - MT0009188643.mp3 (0.0392) | 11 - MT0005478759.mp3 (9.2767) | 11 - MT0001703346.mp3 (9.6418) | 11 - MT0004085907.mp3 (8.8791) |
| 12 - MT0010617945.mp3 (0.0403) | 12 - MT0027035970.mp3 (9.2891) | 12 - MT0018651126.mp3 (9.6592) | 12 - MT0003724610.mp3 (8.9736) |
| 13 - MT0009213083.mp3 (0.0407) | 13 - MT0003794106.mp3 (9.4269) | 13 - MT0034005433.mp3 (9.6594) | 13 - MT0004942017.mp3 (9.2089) |
| 14 - MT0005331755.mp3 (0.0419) | 14 - MT0005737276.mp3 (9.6017) | 14 - MT0013633209.mp3 (9.7162) | 14 - MT0013613461.mp3 (9.3242) |
| 15 - MT0002233402.mp3 (0.0419) | 15 - MT0003903675.mp3 (9.6689) | 15 - MT0005469880.mp3 (9.7927) | 15 - MT0010615428.mp3 (9.3842) |
| 16 - MT0026727455.mp3 (0.0429) | 16 - MT0002262181.mp3 (9.6747) | 16 - MT0017797643.mp3 (9.8590) | 16 - MT0040033011.mp3 (9.3893) |
| 17 - MT0005265641.mp3 (0.0430) | 17 - MT0009208842.mp3 (9.7082) | 17 - MT0034125967.mp3 (9.9112) | 17 - MT0035334027.mp3 (9.4036) |
| 18 - MT0010344415.mp3 (0.0436) | 18 - MT0008222676.mp3 (9.8274) | 18 - MT0009346128.mp3 (9.9399) | 18 - MT0011697297.mp3 (9.4086) |
| 19 - MT0000711493.mp3 (0.0447) | 19 - MT0006367176.mp3 (10.1682) | 19 - MT0013161246.mp3 (9.9644) | 19 - MT0004882280.mp3 (9.4510) |
| 20 - MT0018029465.mp3 (0.0447) | 20 - MT0006540794.mp3 (10.2066) | 20 - MT0005752234.mp3 (9.9787) | 20 - MT0009800907.mp3 (9.5403) |
| 21 - MT0005469880.mp3 (0.0455) | 21 - MT0015742096.mp3 (10.2568) | 21 - MT0000040632.mp3 (9.9899) | 21 - MT0010624346.mp3 (9.6686) |

By observing the above presented results, it is plausible to say that, even though the distance metric varies, the results are pretty much the same, with small variations. The group thinks that this could not be verified if the number of utilized features was to be larger, because each *MMIR* object would have a more specific signature (the feature array) and, therefore, the distance between other objects would be sharper.

## 5.4. Objective Evaluation

For each one of the given queries there were 20 recommended songs based on the following metadata: artist, genre, quadrant and emotion. In order to do this, some text processing was performed using the data contained in the file *panda_dataset_taffc_metadata.csv*. For each in one of the metadata constituents, one point was added to the overall punctuation of the hypothetical recommendation. After this process, the 20 songs with the highest score were recommended for the given query.

For simplicity purposes, the obtained results were suppressed, but are still present in the file *ouput.txt*. The metadata-based results are the ones considered more objective and real. So, taking into account this fact, the precision of the results obtained using feature extraction was computed in respect to the ones obtained using metadata similarity. These precision values vary between 0.00 and 10.00 which are values that were considered to be surprisingly low.

## 5.5. Subjective Evaluation

For a more accurate evaluation the group scored the recommended songs using on the *Likert* scale (from 1(Awful) to 5(Very Good)) taking into account the query song. For example, if someone thought the third

recommended song wasn't a very good match to the query song, they would score it a one. Furthermore, it was also needed to calculate the mean and standard deviation of the group scores which was done using *Excel*, in order to provide a better perspective of the distribution of the recommendations scores.

There were two subjective evaluations: one taking into account the metadata recommendations and another taking into account the 100-feature set and cosine distances.

| | MT0000202045 | | | Mean | Std. | MT0000379144 | | | Mean | Std. | MT0000414517 | | | Mean | Std. | MT0000956340 | | | Mean | Std. | Global mean | | | Global std. | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RB | SS | TV | | | RB | SS | TV | | | RB | SS | TV | | | RB | SS | TV | | | RB | SS | TV | RB | SS | TV |
| 1 | 2 | 1 | 1 | 1.33 | 0.58 | 3 | 3 | 3 | 3.00 | 0.00 | 4 | 5 | 4 | 4.33 | 0.58 | 5 | 5 | 5 | 5.00 | 0.00 | 3.50 | 3.50 | 3.25 | 1.29 | 1.91 | 1.71 |
| 2 | 3 | 2 | 3 | 2.67 | 0.58 | 3 | 3 | 3 | 3.00 | 0.00 | 2 | 3 | 2 | 2.33 | 0.58 | 5 | 5 | 5 | 5.00 | 0.00 | 3.25 | 3.25 | 3.25 | 1.26 | 1.26 | 1.26 |
| 3 | 3 | 3 | 4 | 3.33 | 0.58 | 4 | 5 | 5 | 4.67 | 0.58 | 2 | 2 | 2 | 2.00 | 0.00 | 5 | 5 | 5 | 5.00 | 0.00 | 3.50 | 3.75 | 4.00 | 1.29 | 1.50 | 1.41 |
| 4 | 3 | 3 | 2 | 2.67 | 0.58 | 5 | 4 | 4 | 4.33 | 0.58 | 2 | 2 | 1 | 1.67 | 0.58 | 5 | 5 | 5 | 5.00 | 0.00 | 3.75 | 3.50 | 3.00 | 1.50 | 1.29 | 1.83 |
| 5 | 2 | 2 | 1 | 1.67 | 0.58 | 2 | 3 | 2 | 2.33 | 0.58 | 3 | 3 | 3 | 3.00 | 0.00 | 5 | 5 | 5 | 5.00 | 0.00 | 3.00 | 3.25 | 2.75 | 1.41 | 1.26 | 1.71 |
| 6 | 3 | 3 | 2 | 2.67 | 0.58 | 5 | 5 | 5 | 5.00 | 0.00 | 3 | 3 | 3 | 3.00 | 0.00 | 2 | 2 | 2 | 2.00 | 0.00 | 3.25 | 3.25 | 3.00 | 1.26 | 1.26 | 1.41 |
| 7 | 2 | 2 | 2 | 2.00 | 0.00 | 5 | 5 | 5 | 5.00 | 0.00 | 3 | 2 | 3 | 2.67 | 0.58 | 5 | 4 | 5 | 4.67 | 0.58 | 3.75 | 3.25 | 3.75 | 1.50 | 1.50 | 1.50 |
| 8 | 5 | 5 | 5 | 5.00 | 0.00 | 3 | 2 | 2 | 2.33 | 0.58 | 2 | 2 | 1 | 1.67 | 0.58 | 3 | 4 | 4 | 3.67 | 0.58 | 3.25 | 3.25 | 3.00 | 1.26 | 1.50 | 1.83 |
| 9 | 3 | 4 | 2 | 3.00 | 1.00 | 1 | 1 | 1 | 1.00 | 0.00 | 2 | 3 | 2 | 2.33 | 0.58 | 5 | 5 | 5 | 5.00 | 0.00 | 2.75 | 3.25 | 2.50 | 1.71 | 1.71 | 1.73 |
| 10 | 4 | 5 | 4 | 4.33 | 0.58 | 3 | 2 | 4 | 3.00 | 1.00 | 3 | 4 | 2 | 3.00 | 1.00 | 1 | 4 | 1 | 2.00 | 1.73 | 2.75 | 3.75 | 2.75 | 1.26 | 1.26 | 1.50 |
| 11 | 1 | 2 | 2 | 1.67 | 0.58 | 1 | 2 | 2 | 1.67 | 0.58 | 2 | 3 | 4 | 3.00 | 1.00 | 5 | 5 | 5 | 5.00 | 0.00 | 2.25 | 3.00 | 3.25 | 1.89 | 1.41 | 1.50 |
| 12 | 4 | 4 | 2 | 3.33 | 1.15 | 3 | 3 | 2 | 2.67 | 0.58 | 3 | 2 | 3 | 2.67 | 0.58 | 2 | 3 | 4 | 3.00 | 1.00 | 3.00 | 3.00 | 2.75 | 0.82 | 0.82 | 0.96 |
| 13 | 3 | 3 | 3 | 3.00 | 0.00 | 3 | 4 | 3 | 3.33 | 0.58 | 1 | 3 | 1 | 1.67 | 1.15 | 2 | 5 | 3 | 3.33 | 1.53 | 2.25 | 3.75 | 2.50 | 0.96 | 0.96 | 1.00 |
| 14 | 3 | 3 | 2 | 2.67 | 0.58 | 2 | 2 | 2 | 2.00 | 0.00 | 2 | 1 | 1 | 1.33 | 0.58 | 5 | 2 | 5 | 4.00 | 1.73 | 3.00 | 2.00 | 2.50 | 1.41 | 0.82 | 1.73 |
| 15 | 4 | 4 | 4 | 4.00 | 0.00 | 4 | 4 | 4 | 4.00 | 0.00 | 3 | 2 | 4 | 3.00 | 1.00 | 1 | 2 | 1 | 1.33 | 0.58 | 3.00 | 3.00 | 3.25 | 1.41 | 1.15 | 1.50 |
| 16 | 4 | 3 | 2 | 3.00 | 1.00 | 4 | 4 | 5 | 4.33 | 0.58 | 4 | 5 | 3 | 4.00 | 1.00 | 2 | 5 | 2 | 3.00 | 1.73 | 3.50 | 4.25 | 3.00 | 1.00 | 0.96 | 1.41 |
| 17 | 3 | 3 | 2 | 2.67 | 0.58 | 1 | 1 | 1 | 1.00 | 0.00 | 5 | 3 | 5 | 4.33 | 1.15 | 4 | 4 | 5 | 4.33 | 0.58 | 3.25 | 2.75 | 3.25 | 1.71 | 1.26 | 2.06 |
| 18 | 3 | 2 | 4 | 3.00 | 1.00 | 4 | 5 | 4 | 4.33 | 0.58 | 4 | 5 | 5 | 4.67 | 0.58 | 5 | 5 | 5 | 5.00 | 0.00 | 4.00 | 4.25 | 4.50 | 0.82 | 1.50 | 0.58 |
| 19 | 2 | 1 | 2 | 1.67 | 0.58 | 2 | 3 | 2 | 2.33 | 0.58 | 4 | 3 | 3 | 3.33 | 0.58 | 5 | 4 | 5 | 4.67 | 0.58 | 3.25 | 2.75 | 3.00 | 1.50 | 1.26 | 1.41 |
| 20 | 1 | 2 | 3 | 2.00 | 1.00 | 4 | 4 | 4 | 4.00 | 0.00 | 1 | 1 | 1 | 1.00 | 0.00 | 4 | 4 | 4 | 4.00 | 0.00 | 2.50 | 2.75 | 3.00 | 1.73 | 1.50 | 1.41 |
| | Precision | | | 70.00 | | Precision | | | 65.00 | | Precision | | | 60.00 | | Precision | | | 85.00 | | | | | | | |

Fig.18 – Subjective Evaluation (Feature Extraction)

| | MT0000202045 | | | Mean | Std. | MT0000379144 | | | Mean | Std. | MT0000414517 | | | Mean | Std. | MT0000956340 | | | Mean | Std. | Global mean | | | Global std. | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RB | SS | TV | | | RB | SS | TV | | | RB | SS | TV | | | RB | SS | TV | | | RB | SS | TV | RB | SS | TV |
| 1 | 3 | 3 | 3 | 3.00 | 0.00 | 3 | 3 | 3 | 3.00 | 0.00 | 3 | 3 | 3 | 3.00 | 0.00 | 5 | 5 | 5 | 5.00 | 0.00 | 3.50 | 3.50 | 3.50 | 1.00 | 1.00 | 1.00 |
| 2 | 2 | 2 | 2 | 2.00 | 0.00 | 3 | 3 | 3 | 3.00 | 0.00 | 2 | 2 | 3 | 2.33 | 0.58 | 5 | 5 | 5 | 5.00 | 0.00 | 3.00 | 3.00 | 3.25 | 1.41 | 1.41 | 1.26 |
| 3 | 5 | 5 | 5 | 5.00 | 0.00 | 4 | 5 | 5 | 4.67 | 0.58 | 4 | 4 | 4 | 4.00 | 0.00 | 5 | 5 | 4 | 4.67 | 0.58 | 4.50 | 4.75 | 4.50 | 0.58 | 0.50 | 0.58 |
| 4 | 3 | 4 | 4 | 3.67 | 0.58 | 2 | 2 | 2 | 2.00 | 0.00 | 5 | 5 | 5 | 5.00 | 0.00 | 5 | 5 | 5 | 5.00 | 0.00 | 3.75 | 4.00 | 4.00 | 1.50 | 1.41 | 1.41 |
| 5 | 3 | 3 | 2 | 2.67 | 0.58 | 2 | 3 | 2 | 2.33 | 0.58 | 5 | 4 | 4 | 4.33 | 0.58 | 5 | 5 | 5 | 5.00 | 0.00 | 3.75 | 3.75 | 3.25 | 1.50 | 0.96 | 1.50 |
| 6 | 1 | 1 | 1 | 1.00 | 0.00 | 1 | 1 | 1 | 1.00 | 0.00 | 4 | 4 | 4 | 4.00 | 0.00 | 1 | 1 | 1 | 1.00 | 0.00 | 1.75 | 1.75 | 1.75 | 1.50 | 1.50 | 1.50 |
| 7 | 3 | 2 | 2 | 2.33 | 0.58 | 3 | 3 | 3 | 3.00 | 0.00 | 4 | 5 | 4 | 4.33 | 0.58 | 5 | 5 | 5 | 5.00 | 0.00 | 3.75 | 3.75 | 3.50 | 0.96 | 1.50 | 1.29 |
| 8 | 2 | 2 | 3 | 2.33 | 0.58 | 1 | 2 | 3 | 2.00 | 1.00 | 3 | 3 | 2 | 2.67 | 0.58 | 1 | 1 | 1 | 1.00 | 0.00 | 1.75 | 2.00 | 2.25 | 0.96 | 0.82 | 0.96 |
| 9 | 1 | 1 | 1 | 1.00 | 0.00 | 5 | 3 | 3 | 3.67 | 1.15 | 1 | 1 | 1 | 1.00 | 0.00 | 1 | 1 | 1 | 1.00 | 0.00 | 2.00 | 1.50 | 1.50 | 2.00 | 1.00 | 1.00 |
| 10 | 2 | 2 | 1 | 1.67 | 0.58 | 2 | 2 | 1 | 1.67 | 0.58 | 4 | 4 | 3 | 3.67 | 0.58 | 1 | 1 | 1 | 1.00 | 0.00 | 2.25 | 2.25 | 1.50 | 1.26 | 1.26 | 1.00 |
| 11 | 1 | 1 | 2 | 1.33 | 0.58 | 2 | 2 | 2 | 2.00 | 0.00 | 4 | 3 | 3 | 3.33 | 0.58 | 1 | 2 | 2 | 1.67 | 0.58 | 2.00 | 2.00 | 2.25 | 1.41 | 0.82 | 0.50 |
| 12 | 1 | 2 | 2 | 1.67 | 0.58 | 4 | 4 | 5 | 4.33 | 0.58 | 1 | 1 | 1 | 1.00 | 0.00 | 1 | 1 | 1 | 1.00 | 0.00 | 1.75 | 2.00 | 2.25 | 1.50 | 1.41 | 1.89 |
| 13 | 2 | 2 | 2 | 2.00 | 0.00 | 4 | 4 | 4 | 4.00 | 0.00 | 3 | 3 | 4 | 3.33 | 0.58 | 1 | 1 | 1 | 1.00 | 0.00 | 2.50 | 2.50 | 2.75 | 1.29 | 1.29 | 1.50 |
| 14 | 1 | 1 | 1 | 1.00 | 0.00 | 4 | 4 | 4 | 4.00 | 0.00 | 2 | 1 | 1 | 1.33 | 0.58 | 4 | 5 | 5 | 4.67 | 0.58 | 2.75 | 2.75 | 2.75 | 1.50 | 2.06 | 2.06 |
| 15 | 3 | 2 | 3 | 2.67 | 0.58 | 3 | 3 | 2 | 2.67 | 0.58 | 2 | 3 | 3 | 2.67 | 0.58 | 5 | 5 | 5 | 5.00 | 0.00 | 3.25 | 3.25 | 3.25 | 1.26 | 1.26 | 1.26 |
| 16 | 2 | 1 | 1 | 1.33 | 0.58 | 3 | 3 | 3 | 3.00 | 0.00 | 5 | 4 | 4 | 4.33 | 0.58 | 5 | 5 | 5 | 5.00 | 0.00 | 3.75 | 3.25 | 3.25 | 1.50 | 1.71 | 1.71 |
| 17 | 1 | 1 | 1 | 1.00 | 0.00 | 3 | 4 | 4 | 3.67 | 0.58 | 1 | 2 | 2 | 1.67 | 0.58 | 2 | 1 | 1 | 1.33 | 0.58 | 1.75 | 2.00 | 2.00 | 0.96 | 1.41 | 1.41 |
| 18 | 1 | 1 | 1 | 1.00 | 0.00 | 2 | 3 | 3 | 2.67 | 0.58 | 2 | 2 | 2 | 2.00 | 0.00 | 2 | 2 | 1 | 1.67 | 0.58 | 1.75 | 2.00 | 1.75 | 0.50 | 0.82 | 0.96 |
| 19 | 2 | 3 | 4 | 3.00 | 1.00 | 5 | 5 | 5 | 5.00 | 0.00 | 4 | 4 | 4 | 4.00 | 0.00 | 4 | 4 | 5 | 4.33 | 0.58 | 3.75 | 4.00 | 4.50 | 1.26 | 0.82 | 0.58 |
| 20 | 3 | 3 | 3 | 3.00 | 0.00 | 4 | 4 | 4 | 4.00 | 0.00 | 4 | 4 | 3 | 3.67 | 0.58 | 4 | 5 | 5 | 4.67 | 0.58 | 3.75 | 4.00 | 3.75 | 0.50 | 0.82 | 0.96 |
| | Precision | | | 35.00 | | Precision | | | 70.00 | | Precision | | | 70.00 | | Precision | | | 55.00 | | | | | | | |

Fig.19 – Subjective Evaluation (Metadata)

After analyzing both figures above, the group came to a conclusion that is found to be a bit contradictory: the feature extraction process provided more precise than the one based in metadata. This demonstrates one of the most common and studied problems in *MMIR*: how to formalize sets of concepts that represent the human perspective of the music, since it may be influenced by memories, emotions, current state of spirit, etc. Also, it is consensual that the last query, which is a rap song, allowed to obtain the best results. This may be because rap is a type of music which may be not as complex in the lyrical and melodical domains, when compared to other genres, such as pop, country or even classic.

# 6. Conclusion

Upon finishing this practical work, the following familiarities were acquired:

- Feature extraction, both spectral and temporal.

- The differences of *librosa* features implementation to root features implementation.

- The perceptual knowledge and analysis behind the Multimedia Information Retrieval.

- The music similarity analysis of a computer compared to a human (subjective).

- Some aspects of machine learning in music emotion recognition.

# 7. Bibliography

[1] Multimedia Information Retrieval, Morgan & Claypool, Stefan Rüger

[2] Multimedia Information Retrieval, Elsevier, Roberto Raieli (2013)

[3] Wikipedia, a enciclopédia livre, Music Information Retrieval, https://en.wikipedia.org/wiki/Music_information_retrieval. Acedido em 07/05/2022.

[4] A Tutorial on Spectral Feature Extraction for Audio Analytics, Yugesh Verma, https://analyticsindiamag.com/a-tutorial-on-spectral-feature-extraction-for-audio-analytics/, Acedido em 09/05/2022.

[5] Music Feature Extraction in Python, Sanket Doshi, https://towardsdatascience.com/extract-features-of-music-75a3f9bc265d, Acedido em 10/05/2022.

[6] librosa: Audio and Music Signal Analysis in Python, Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, Oriol Nieto, PROC. OF THE14th PYTHON IN SCIENCE CONF. (SCIPY 2013)