LICENTA

Enhancing Our Lives Through Music:

Genre Classification and Generation

Abstract:

In this paper, I investigate a comprehensive approach to music genre classification and generation, utilizing a combination of deep neural networks, machine learning algorithms, variational autoencoders (VAEs), long short-term memory (LSTM) networks, and Transformers. My research aims to advance the understanding of music's impact on our lives and develop methodologies to create diverse and engaging musical experiences tailored to individual preferences.

I begin by extracting relevant features from a large and diverse collection of music samples from different genres. These features, encompassing spectral properties, rhythmic patterns, and tonal characteristics, serve as the foundation for my genre classification and generation models. I employ deep neural networks and machine learning algorithms to effectively classify music genres by capturing the distinct characteristics of each genre.

In order to generate music, I explore the potential of VAEs, LSTMs, and Transformers, each offering unique capabilities for handling different aspects of the task. VAEs are employed to learn a continuous latent space representation of the music samples, enabling the generation of novel compositions within a specified genre. LSTMs and Transformers, on the other hand, are used to model the temporal dependencies and intricate patterns inherent in music.

My methodology is evaluated through a series of experiments, with the results compared against conventional machine learning algorithms and other deep learning architectures. While not claiming state-of-the-art performance, my approach demonstrates promising outcomes in both classification and generation tasks, showcasing its potential to enhance music-related applications such as recommendation systems and creative tools for composers.

In conclusion, the suggested framework provides a flexible and thorough answer, opening the door for more research and advancement in music technology.

Contents

[1. Introduction 4](#_Toc136424810)

[1.1. Motivation 5](#_Toc136424811)

[1.2. Outline of the thesis 5](#_Toc136424812)

[1.3. Application fields 6](#_Toc136424813)

[2. Related work 7](#_Toc136424814)

[2.1. Similar research 7](#_Toc136424815)

[2.1.1. Music Genre Classification 7](#_Toc136424816)

[2.1.2. JukeBox 7](#_Toc136424817)

[2.1.3. Groove2Groove 7](#_Toc136424818)

[2.2. Applications in the field 8](#_Toc136424819)

[2.2.1. Spotify 8](#_Toc136424820)

[2.2.2. SoundHound 8](#_Toc136424821)

[2.2.3. Soundraw 9](#_Toc136424822)

[3. Technical background 10](#_Toc136424823)

[3.1. EDA 11](#_Toc136424824)

[3.1.1. Correlation Heatmap 12](#_Toc136424825)

[3.1.2. Principal Component Analysis 13](#_Toc136424826)

[3.1.3. Feature Engineering 15](#_Toc136424827)

[3.1.4. Feature selection 16](#_Toc136424828)

[3.2. Music Genre Recognition 16](#_Toc136424829)

[3.2.1. Audio signals 18](#_Toc136424830)

[3.2.1.1. Chroma STFT 20](#_Toc136424831)

[3.2.1.2. Root-Mean-Square (RMS) 21](#_Toc136424832)

[3.2.1.3. Spectral centroid 22](#_Toc136424833)

[3.2.1.4. Spectral bandwidth 23](#_Toc136424834)

[3.2.1.5. Spectral Rolloff 24](#_Toc136424835)

[3.2.1.6. Flatness 25](#_Toc136424836)

[3.2.1.7. Contrast 26](#_Toc136424837)

[3.2.1.8. Flux 27](#_Toc136424838)

[3.2.1.9. Zero crossing rate (ZCR) 28](#_Toc136424839)

[3.2.1.10. Harmonics and Perceptrual/Percussive 29](#_Toc136424840)

[3.2.1.11. Tempo 29](#_Toc136424841)

[3.2.1.12. Mel Frequency Cepstral Coefficients (MFCC) 30](#_Toc136424842)

[3.2.1.13. Mel Spectrogram 32](#_Toc136424843)

[3.2.2. Machine Learning 32](#_Toc136424844)

[3.2.2.1. KNN 32](#_Toc136424845)

[3.2.2.2. SVM 32](#_Toc136424846)

[3.2.2.3. Random Forest 33](#_Toc136424847)

[3.2.2.4. Logistic Regression 33](#_Toc136424848)

[3.2.2.5. Naïve Bayes 33](#_Toc136424849)

[3.2.2.6. XGBoost 33](#_Toc136424850)

[3.2.3. Deep Learning 33](#_Toc136424851)

[3.2.3.1. CNN 33](#_Toc136424852)

[3.2.3.2. RNN 33](#_Toc136424853)

[3.2.3.3. Transfer Learning 33](#_Toc136424854)

[3.2.3.4. Vision Transformers 33](#_Toc136424855)

[3.2.3.5. Reinforcement Learning 33](#_Toc136424856)

[3.3. Music Generation 33](#_Toc136424857)

[3.3.1. Music notation 33](#_Toc136424858)

[3.3.2. LSTM 33](#_Toc136424859)

[3.3.3. Transformer 33](#_Toc136424860)

[3.3.4. VAE 33](#_Toc136424861)

[3.3.5. GAN 34](#_Toc136424862)

[4. Proposed framework 34](#_Toc136424863)

[4.1. Music Genre Recognition 34](#_Toc136424864)

[4.2. Music Generation 34](#_Toc136424865)

[5. Evaluation 34](#_Toc136424866)

[5.1. Music Genre Recognition 34](#_Toc136424867)

[5.2. Music Generation 34](#_Toc136424868)

[6. Conclusion and future work 34](#_Toc136424869)

[7. Bibliography 34](#_Toc136424870)

# 1. Introduction

## 1.1. Motivation

The transformative power of music has been a cornerstone of human culture and expression throughout history. As diverse as our societies, the vast array of music genres reflects our rich cultural heritage and individual creativity. With the rapid advancements in technology, new paradigms in music analysis and synthesis are emerging, further enhancing our understanding of this universal language.

The motivation for this research paper stems from a desire to harness the potential of cutting-edge learning techniques, such as Machine Learning, Deep Neural Networks (DNNs), Long Short-Term Memory (LSTM) networks, Variational Autoencoders (VAEs), and Transformers, to address two primary challenges in the field of music informatics: genre classification and music generation.

The accurate classification of music genres is a crucial task that aids in music discovery, recommendation, and tagging. However, the subjective nature of genre definitions and the complexity of musical elements make this a challenging problem. By employing deep neural networks and machine learning techniques, this research aims to develop robust and sophisticated models that can effectively capture the intricate patterns and features of various music genres, leading to improved classification performance.

The creative process of composing music has long been considered the exclusive domain of human intellect. However, recent advancements in artificial intelligence and machine learning have demonstrated the potential for machines to generate music with remarkable coherence and originality. This research explores the application of LSTM, VAE, and Transformer models to the task of music generation, with the goal of developing systems capable of producing high-quality compositions across different genres, while respecting the unique characteristics and structures inherent to each.

By combining expertise in music theory, machine learning and artificial intelligence, this research paper aims to contribute to the ongoing efforts to revolutionize the field of music informatics. It is hoped that the findings will not only serve as a foundation for future research endeavors but also pave the way for innovative applications that enrich our understanding and appreciation of the world of music.

“Music gives a soul to the universe, wings to the mind, flight to the imagination, and life to everything.” – Plato

## 1.2. Outline of the thesis

This thesis is organized as follows:

After providing a brief introduction to the goals and motivation of this research paper, [Chapter 2](#_2._Related_work) will present various related experiments that demonstrate diverse approaches to music classification and generation. These studies include unique datasets, models, metrics, and results, giving a well-rounded view of the different methods used in this field.

Moving forward, [Chapter 3](#_3._Technical_background) offers a comprehensive exploration of music theory, delving into the various theoretical elements, structures and algorithms employed throughout this study. [Chapter 4](#_4._Proposed_framework) thoroughly outlines the data preparation process for information extraction, the selected models, and the overall framework. [Chapter 5](#_5._Evaluation) then explores the interpretation of outcomes and showcases a selection of pertinent metrics that effectively represent the issues under discussion.

Music tagging, like identifying genres or emotions, is a common practice in the music industry. However, there hasn't been a ton of work done in generating music in a specific style. Despite this, the work that has been done is pretty varied, because it can involve different ways of representing music and address different concepts. Pursuing the directions offered in this study, numerous intriguing concepts merit further exploration, including transferring styles between songs while preserving the original content or making remixes of melodies while controlling specific attributes like chromaticism, groove, or instrumentation, aspects discussed in [Chapter 6](#_6._Conclusion_and).

## 1.3. Application fields

The swift advancement of machine learning and artificial intelligence methodologies has accelerated the growth of music information retrieval and music creation. These technologies offer numerous applications, which possess the potential to radically transform our interaction, consumption and production of music.

Content-based music retrieval and recommendation systems constitute a principal application domain of music genre identification. By classifying music tracks into well-defined genres, these systems can deliver customized music suggestions based on users' predilections and listening behavior. Additionally, refined genre identification can aid in enhancing playlist creation, empowering users to uncover new music within their favored genres or based on their listening mood.

Music genre identification is integral to Music Information Retrieval (MIR), an interdisciplinary area concerned with extracting and organizing information from music. Advanced genre identification methods streamline the examination and cataloging of extensive music databases, facilitating the detection of patterns, trends, and connections between various musical pieces and enriching our comprehension of music as a cultural phenomenon.

On the other hand, music creation techniques have given rise to interactive music composition tools, allowing users to produce original music with minimal expertise. Employing AI-generated music, these tools can support users in composing melodies, harmonies, and rhythms, enabling them to concentrate on their artistic vision. Furthermore, such tools can serve educational purposes, assisting budding musicians in honing their skills and grasping music theory.

Automatic music transcription systems can leverage improvements in music genre identification and music creation. By incorporating genre-specific knowledge, these systems can enhance their precision in transposing audio recordings into symbolic notation, enabling musicians to analyze, modify, and perform the transcribed music more effectively.

Music creation techniques have been applied to video game and film scoring. AI-generated music can adapt to the dynamic nature of video games, offering immersive and responsive auditory experiences for players. Similarly, in the film industry, AI-generated music can function as an economical alternative to conventional scoring methods, addressing the unique demands of individual scenes and narratives.

Music therapy has long been acknowledged for its potential advantages in fostering mental and emotional well-being. AI-generated music can be customized to address the specific requirements of individuals, delivering personalized therapeutic experiences. Additionally, music creation techniques can be employed in the development of assistive technologies for people with disabilities, enabling them to interact with music in innovative and accessible ways.

# 2. Related work

## 2.1. Similar research

### 2.1.1. Music Genre Classification

The paper at [[derekahuang/Music-Classification (github.com)](https://github.com/derekahuang/Music-Classification)]

### 2.1.2. JukeBox

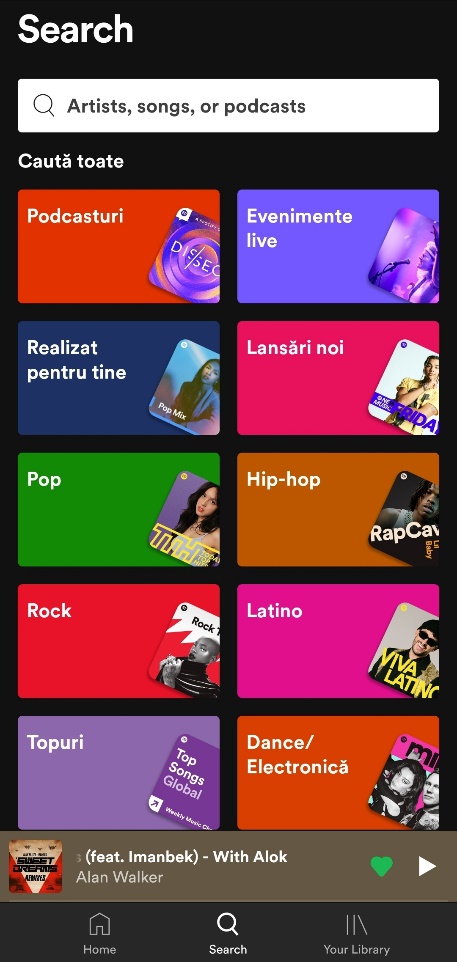
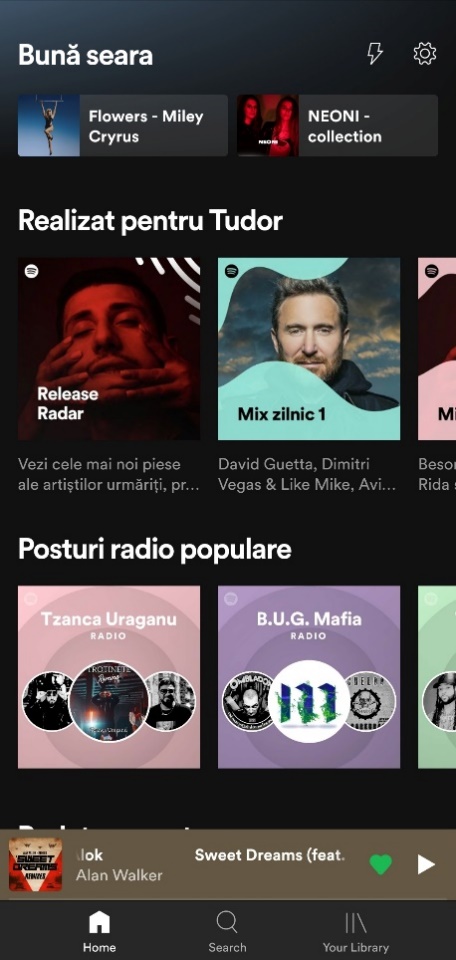
The paper at [[Jukebox: A Generative Model for Music (arxiv.org)](https://arxiv.org/pdf/2005.00341.pdf)]

### 2.1.3. Groove2Groove

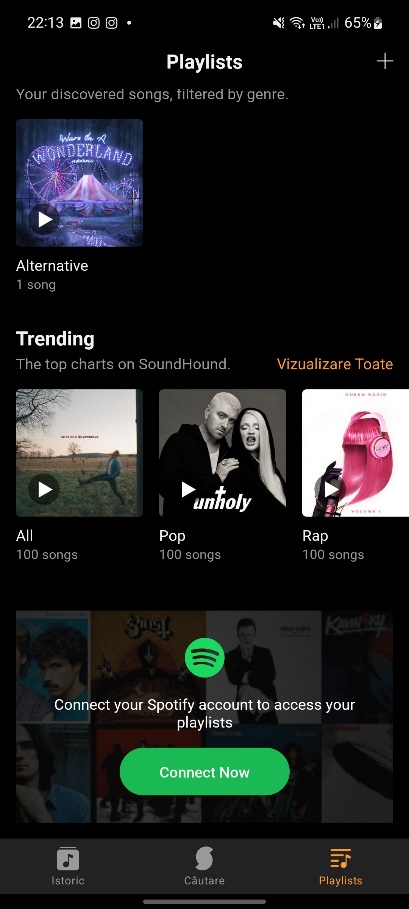
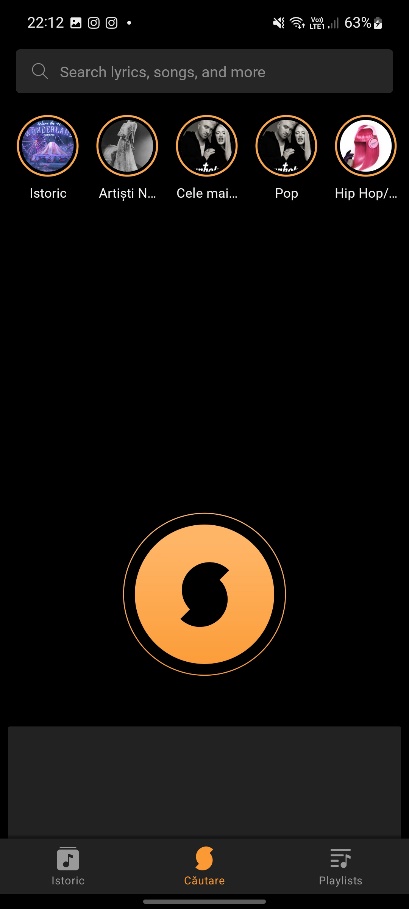
The paper at [[taslp-3019642-pp.pdf (hal.science)](https://hal.science/hal-02923548v2/document)]

## 2.2. Applications in the field

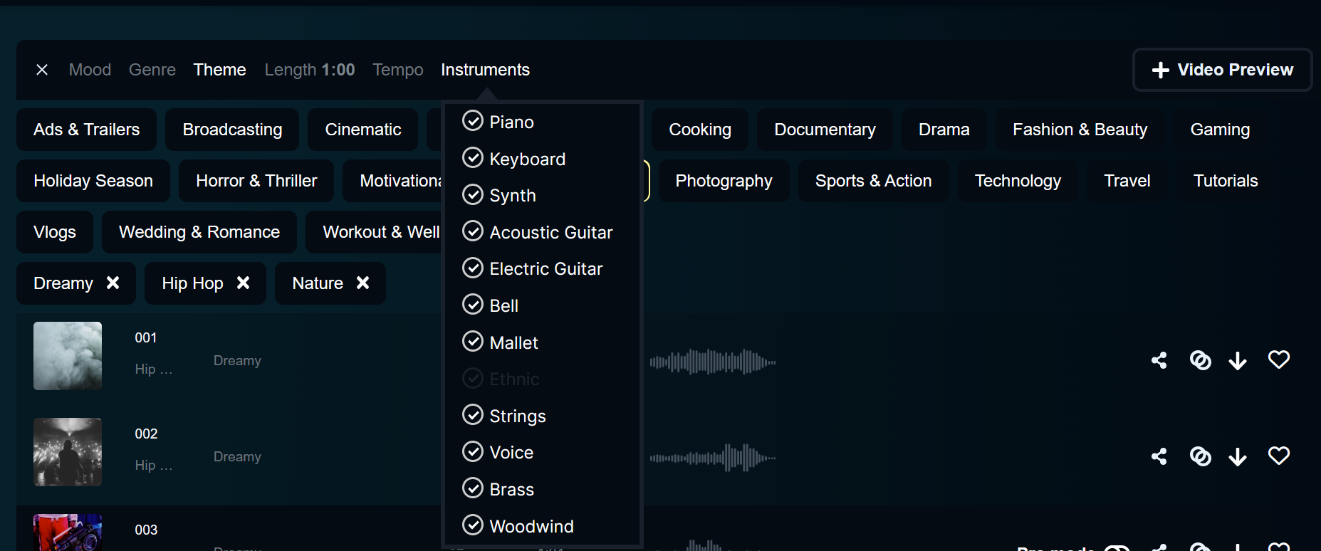
### 2.2.1. Spotify

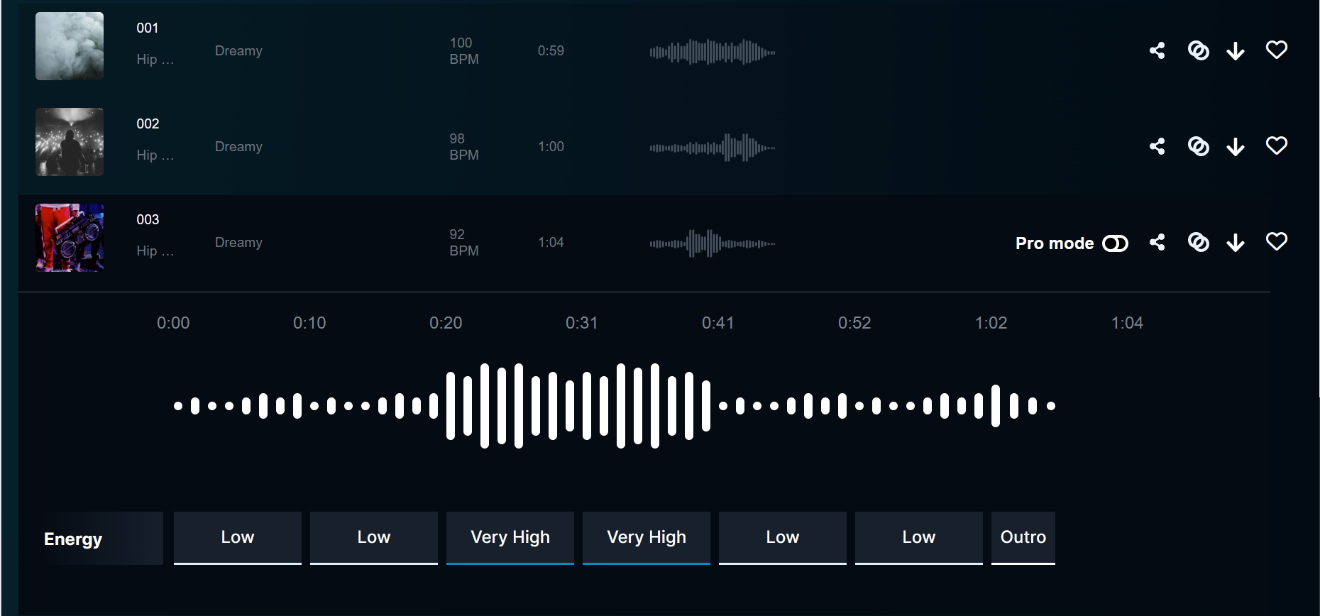


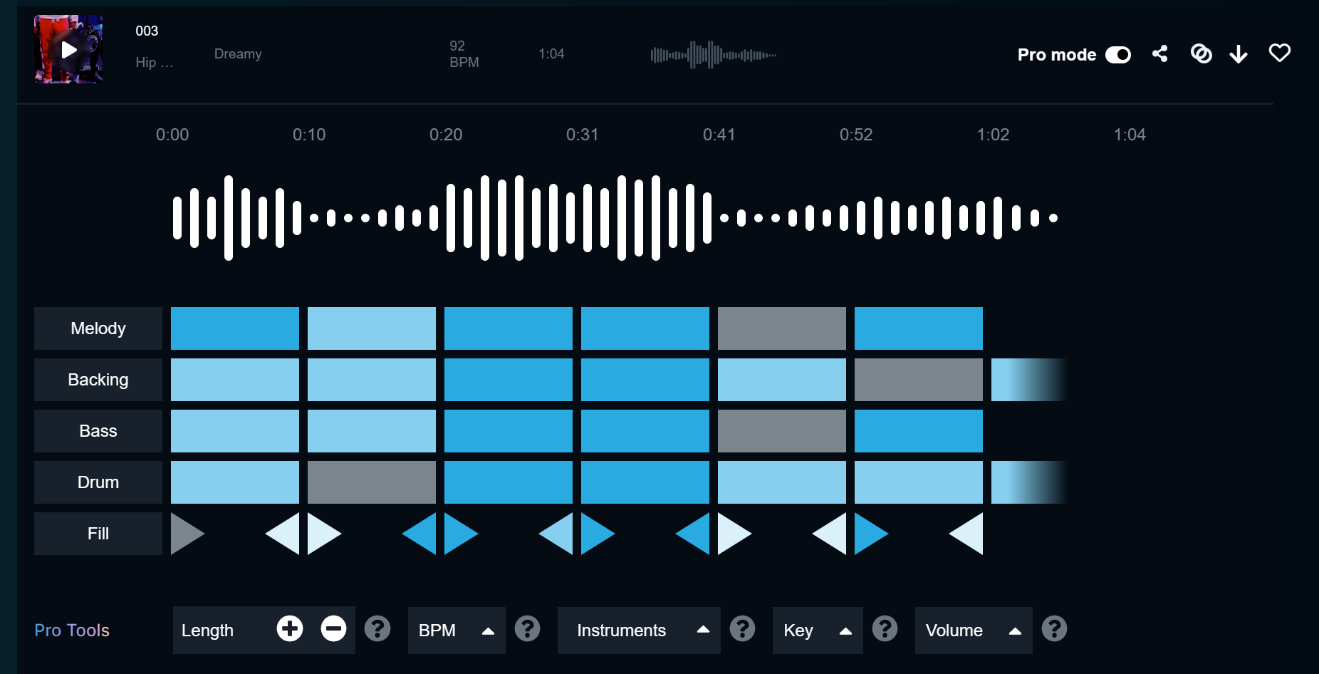
### 2.2.2. SoundHound



### 2.2.3. Soundraw







# 3. Technical background

Music Information Retrieval (MIR) is a field that involves the study of processing audio signals in order to access relevant information from music. It combines elements of music theory, physics, psychology, signal processing, linguistics, mathematics and computer science. A few applications to this field include:

1. Understanding Music Data: Music can be represented in various formats, such as audio signals, symbolic notations, or lyrics. MIR aims to extract meaningful information from these different data representations.

2. Feature Extraction: This process involves identifying key aspects of music data, such as pitch, tempo, rhythm, melody, harmony, and timbre. These features can help in the analysis and categorization of music.

3. Music Analysis: MIR can be used to analyze music on various levels. This includes low-level analysis (determining the beat or pitch), mid-level analysis (identifying beat tracking or chord recognition) and high-level analysis (classifying the genre or mood of a piece).

4. Search and Recommendation: techniques in the field are often used in music recommendation systems, such as those found in Spotify or Pandora. These systems analyze the features of music to provide personalized recommendations to users.

5. Music Transcription: it also encompasses automatic music transcription, which is the process of converting a music audio signal into a symbolic representation such as sheet music or MIDI.

6. Music Classification and Clustering: these techniques can also be used to classify music into different genres, moods, or even identifying similar songs which is extremely useful in music libraries and streaming platforms to organize music and create playlists.

7. Music Generation: MIR is also used in music generation, where algorithms learn the patterns in music data to create new compositions.

Research in Music Information Retrieval can lead to advancements in various applications such as music search engines, digital libraries, music composition software, interactive music systems, and even music education and therapy. However, it also poses great challenges due to the complexity and subjectivity of music perception.

## 3.1. EDA

[[What is Exploratory Data Analysis? | IBM](https://www.ibm.com/topics/exploratory-data-analysis)]

Exploratory Data Analysis (EDA) is a strategic approach employed by data scientists to analyze and investigate datasets, thereby summarizing their predominant characteristics, frequently via visualization techniques. It helps in identifying the most efficient way to manipulate data sources to derive the required information, discovery of patterns, detection of anomalies, hypothesis testing and verification of assumptions.

Primarily, EDA is utilized to glean insights that data can offer beyond the conventional modeling or hypothesis testing activities. It bestows a more profound understanding of the dataset's variables and their interrelationships. Moreover, it can assist in verifying whether the statistical techniques considered for data analysis are appropriate.

The core objective is to enable an in-depth examination of data prior to making any assumptions. It facilitates the identification of apparent errors, deeper comprehension of data patterns, detection of outliers or anomalous occurrences and discovery of interesting relationships among variables.

In the realm of data science, exploratory analysis serves to validate the results, ensuring their applicability to the intended outcomes and objectives. It further confirms the relevance of the questions being asked. It also provides answers regarding standard deviations, categorical variables, and confidence intervals. Upon completion of EDA and drawing of insights, its attributes can be employed for more advanced data analysis or modeling, including machine learning.

Specific statistical functions and techniques that can be performed with EDA tools include:

* Clustering and dimension reduction techniques, which help create graphical displays of high-dimensional data containing many variables.
* Univariate visualization of each field in the raw dataset, with summary statistics.
* Bivariate visualizations and summary statistics that allow you to assess the relationship between each variable in the dataset and the target variable you’re looking at.
* Multivariate visualizations, for mapping and understanding interactions between different fields in the data.
* K-means Clustering is a clustering method in [unsupervised learning](https://developer.ibm.com/articles/cc-unsupervised-learning-data-classification) where data points are assigned into K groups, i.e. the number of clusters, based on the distance from each group’s centroid. The data points closest to a particular centroid will be clustered under the same category. K-means Clustering is commonly used in market segmentation, pattern recognition, and image compression.
* Predictive models, such as linear regression, use statistics and data to predict outcomes.

### 3.1.1. Correlation Heatmap

A correlation heatmap is a type of data visualization tool that represents the correlations between multiple variables in a dataset. It's a grid of cells, where each cell represents a correlation coefficient between two variables. It's especially useful for feature selection in machine learning, identifying multicollinearity in regression analysis and for exploratory data analysis.

**Correlation** is a statistical measure that describes the degree of relationship between two variables. The correlation coefficient shows the strength of the correlation and its magnitude can range from -1 to 1. A correlation of -1 indicates a perfect negative correlation (as one variable increases, the other decreases), a correlation of 1 indicates a perfect positive correlation (as one variable increases, the other increases as well) and a correlation of 0 indicates no linear relationship between the variables. However, a correlation between two variables does not imply causation, it only indicates a statistical relationship between the variables. Further investigation is needed to determine if one variable is causing the other.

A **heatmap** is a graphical representation of data where values are depicted by color. It gives a visual representation of data using colors, where higher intensity ("hotter" / brighter) colors represent higher values, and lower intensity ("cooler" / darker) colors represent lower values. Heatmaps are used when there are too many data points to plot individually. The color of the cell is indicative of the correlation coefficient: positive correlations are displayed in one color gradient (increasingly intense shades of red) and negative correlations in another color gradient (increasingly intense shades of blue).

A correlation heatmap is useful for:

- Visual representation: easier to understand and interpret the relationships between variables, especially when dealing with a large number of them.

- Identifying relationships: highlights the variables that are highly correlated with each other and those that have little or no correlation.

- Feature selection: by identifying the variables that are highly correlated with the target variable, one can choose the most important variables for the model, reducing the risk of overfitting and improving model performance.

- Data cleaning: identify and remove redundant or unnecessary variables, which can improve the accuracy of the analysis and the interpretability of the results.

- Hypothesis testing: generate hypotheses about the relationships between variables, hypotheses that can then be tested further through statistical analysis.

### 3.1.2. Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction method that is often used **to reduce the dimensionality** of large data sets, by transforming a large set of variables into a smaller one that **still contains most of the information** in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data points much easier and faster for machine learning algorithms without extraneous variables to process.

PCA can be broken down into five steps:

1. **Standardization**: The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis. It is critical to perform standardization prior to PCA, because the latter is quite sensitive regarding the variances of the initial variables. That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges, which will lead to biased results. Mathematically, for each value of each variable:
2. **Covariance Matrix Computation**: Sometimes variables are highly correlated in such a way that they contain redundant information. It is the sign of the covariance that really matters: when positive, the two variables increase/decrease together (correlated); when negative, as one increases, the other decreases (inversely correlated).
3. **Eigenvectors and eigenvalues**: Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data. The relationship between variance and information is that the larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more information it has. To put all this simply, just think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible. Without further ado, it is eigenvectors and eigenvalues who are behind the PCA, because the eigenvectors of the Covariance matrix are actually the directions of the axes where there is the most variance (most information) and that we call Principal Components. And eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component. By ranking the eigenvectors in order of their eigenvalues, highest to lowest, it results the principal components in order of significance.
4. **Feature vector**: The feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep. This makes it the first step towards dimensionality reduction, because if we choose to keep only p eigenvectors (components) out of n, the final data set will have only p dimensions.
5. **Recast data along PCA axes**: Until the current step, the input data set remains the same in terms of the original axes (in terms of the initial variables). In this last step, the aim is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components. This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

PCA is useful for:

- Dimensionality reduction: reduce the dimensionality of a data set, which can be especially useful when dealing with high-dimensional data. This can improve the performance of machine learning models, increase the interpretability of the results, and reduce the risk of overfitting.

- Data visualization: visualize complex data sets by projecting the data onto the principal components. This can help to identify patterns and relationships in the data that may not have been apparent in the original variables.

- Data compression: compress the data by retaining only the most important components and discarding the others. This can reduce the storage and computation requirements for large data sets.

- Noise reduction: remove noise from the data by retaining only the components that explain the most variance. This can improve the accuracy and stability of the results.

- Feature extraction: extract new features from the data, which can be used as input variables in machine learning models. This can improve the performance of the models by reducing the dimensionality of the data and eliminating redundant or irrelevant variables.

### 3.1.3. Feature Engineering

Feature engineering is a critical step in building accurate and effective models. One key aspect of feature engineering is scaling, normalization and standardization, which involves transforming the data to make it more suitable for modeling. These techniques can help to improve performance, prevent the outliers from having a disproportionate effect on the results, improve the convergence of optimization algorithms and ensure that the data is on the same scale.

**Feature scaling** is a technique that involves transforming the values of features or variables in a dataset to a similar scale. This is done to ensure that all features contribute equally to the model and to prevent features with larger values from dominating the model. Feature scaling is essential when working with datasets where the features have different ranges, units of measurement or orders of magnitudes. It is critical to scale data for some machine learning algorithms, especially those that use distance measures (KNN) or those that use gradient descent to optimize their cost functions (logistic regression or deep learning algorithms).

**Normalization** is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1; it is also known as Min-Max scaling. It is sensitive to outliers, as an extreme value can shift the entire range. The mathematical formula used for scaling uses the minimum and maximum value of each feature column:

**Standardization** is also a scaling technique where the values are centered around the mean with a unit standard deviation. In this case, the values are not restricted to a particular range, which can be useful for algorithms that do not require a specific scale. Furthermore, it is not as sensitive to outliers as normalization.

s

### 3.1.4. Feature selection

From the machine learning point of view, features are not necessarily of equal importance or quality, and irrelevant or redundant features may lead to inaccurate conclusion. Experiments have shown that, although the performance can thus be improved to a certain extent, using too many features leads to performance degradation [J. L. Zhang, X. L. Huang, L. F. Yang, Y. Xu, and S. T. Sun, “Feature selection and feature learning in arousal dimension of music emotion by using shrinkage methods,” Multimedia Systems, vol. 23, no. 2, pp. 251–264, 2017].

With a highly discriminant sets of features, is not true that their combination produces a better discriminant power, for example if the set of features is *n*, the number of possible combinations is:

This can easily become impossible to compute, that is why some feature selection algorithm must be applied.

TODO: -maybe add more info from master thesis

## 3.2. Music Genre Recognition

Music holds a significant place in human existence, a prominence that has only increased in the era of digital technology. There's never been such an extensive array of music created and engaged with daily as there is today. Initially, compact audio formats like MP3 provided near CD quality music, but now numerous streaming platforms have further spurred the significant expansion of digital music collections.

Traditionally, organizing music collections has relied on cataloging metadata like the artist's name, album title, and song name. However, with the exponential surge of content, this traditional method might not be adequate any longer. The way we categorize and access musical information needs to evolve in response to the ever-growing demand for efficient and effortless information access.

Music, with its intricate acoustic and temporal structure, is abundant in content and expressiveness. When someone interacts with music, whether as a composer, performer, or listener, a broad array of cognitive processes come into play. These include representational processes like understanding rhythm, meter, melody, harmony, style, and form, and evaluative processes involving preferences, aesthetic experience, mood, and emotions. The term "evaluative" is used because these processes often involve subjective and differing responses. Both the representational and evaluative aspects of music listening could be exploited to improve music retrieval.

According to a study by Last.fm [https://www.last.fm/home], tags related to emotions are the third most commonly applied to music pieces by online users, with genre and geographical area being the first and second most frequently assigned, respectively.

A survey conducted in 2004 showed that a significant amount of the participants, about 62.7%, identified the style/genre of a musical piece as an important criterion in music seeking and organization.

Survey from [J. H. Lee and J. S. Downie, “Survey of music information needs, uses, and seeking behaviours: preliminary findings,” in ISMIR, vol. 2004, p. 5th, Citeseer, 2004.]

|  |  |
| --- | --- |
| SEARCH / BROWSE BY | POSITIVE RATE |
| Singer / performer | 96.2% |
| Title of work(s) | 91.6% |
| Some words of the lyrics | 74.0% |
| Music style/genre | **62.7%** |
| Recommendations | 62.2% |
| Similar artist(s) | 59.3% |
| Similar music | 54.2% |
| Associated usage | 41.9% |
| Singing | 34.8% |
| Theme (main subject) | 33.4% |
| Popularity | 31.0% |
| Mood / emotional state | 28.2% |
| Time period | 23.8% |
| Occasions to use | 23.6% |
| Instrument(s) | 20.8% |
| place / event where heard | 20.7% |
| storyline of music | 17.9% |
| Tempo | 14.2% |
| Record label | 11.7% |
| Publisher | 6.0% |

Table 3.2.1: Responses of 427 subjects to the question “When you search for music or music information, how likely are you to use the following search/browse options?”

In recent years, there has been a lot of activity in the field of audio research, especially with the use of advanced computer techniques like Machine Learning and Deep Learning. With the rise of "Big Data," there is now a lot of digital music available online, which has led to the creation of many online music databases. To make it easier for people to find the music they want, it is important to group music by genre. This is where genre classification comes in, and it has many practical applications, like helping to organize music collections and making it easier to find songs in search engines and music databases. Automated genre classification can be especially useful for companies like Spotify and iTunes, who add thousands of new songs every month.

Music is something that everyone around the world can understand and enjoy. It is a way for us to share our feelings and experiences, and it is also different everywhere you go. This is because there are so many types of music, called 'genres’ and they all sound unique. Think about how different blues music sounds compared to hip-hop, or how classical music is so different from electronic music. These genres have been created over many years and each one tells its own story about people's lives and history. This section will show you a map of all these different music genres. It is like a big adventure through the world of music, showing you just how many types there are and how beautiful each one is in its own way.



Figure 3.2.1. Every music genre from <https://everynoise.com/engenremap.html>

### 3.2.1. Audio signals

Sound is a type of energy vibrating through a medium (such as air), if there is no medium (in space) there is no sound; this energy, within a specific range of frequencies (20Hz – 20kHz), is interpreted by the human ear as sound. It is made up of three basic elements: **frequency**, how fast the vibrations are occurring, **intensity**, how loud the sound is, and **timbre**, the sound’s quality. Thus, in a physical sense, an audio signal is a representation of sound as a function of time, typically as a variation of air pressure. It can be converted to an electrical signal by a microphone and then digitized using an analog-to-digital converter.

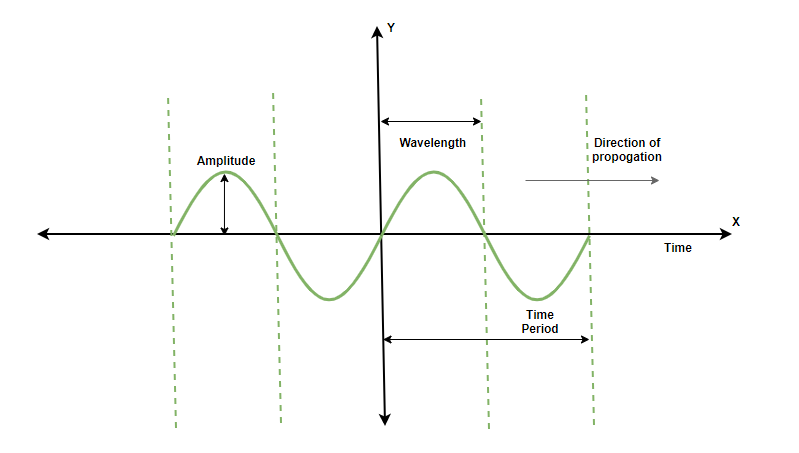


Figure 3.2.1.1. Sound waves in 2D axes

In a digital sense, the audio signal is a sequence of numbers that represents the amplitude of the signal at each point in time. For stereo audio, there are typically two such sequences (left / right channel), while mono audio consists of only a single sequence. Digital audio signals can be characterized by their sample rate (the number of samples per second) and bit depth (the number of bits used to represent each sample). For example, an audio of CD-quality has a sample rate of 44.1kHz and a bit depth of 16 bits.

There are several audio formats in which the sound can be stored and manipulated: .wav (lossless, large files), .mp3(lossy, compressed files), .flac(lossless, compressed files). In order to manipulate an audio file in Python one can use numerous libraries: Librosa, PyAudio or built-in modules.

In order to play audio files inside notebooks, IPython module offers a pretty and simple configuration:

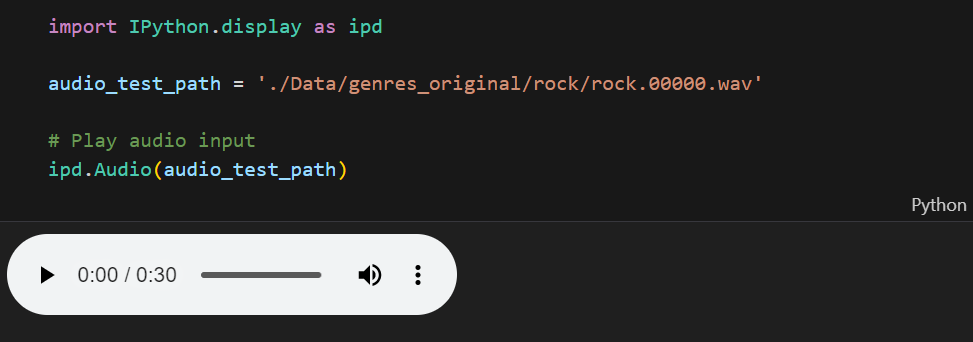


Figure 3.2.1.2. Playable in-notebook audio file

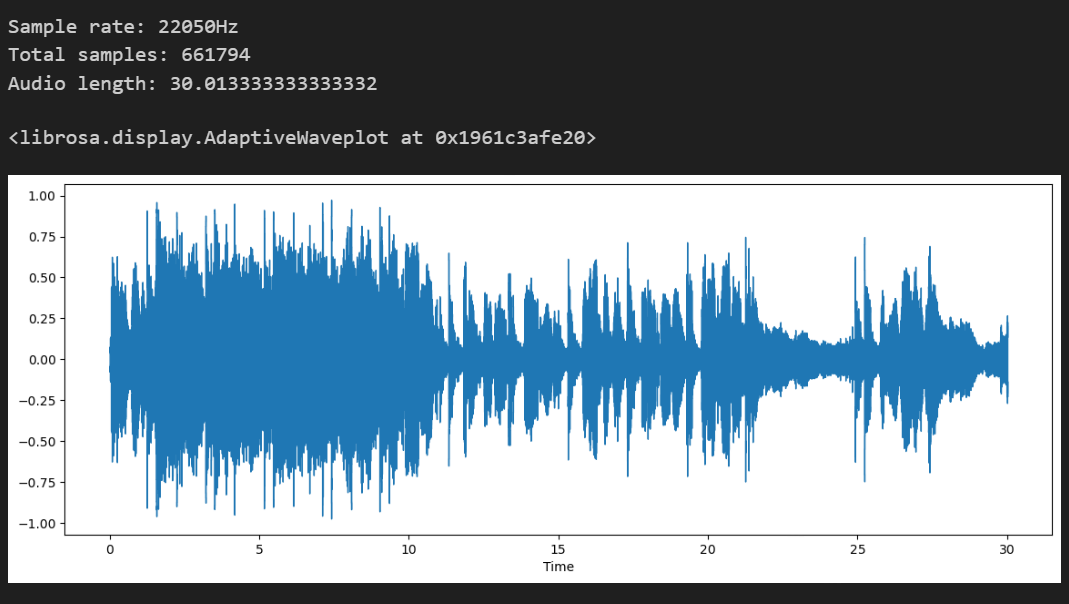


Figure 3.2.1.3. Waveform and audio specifications

In order to use audio signals / music for the task of genre classification it is necessary to extract relevant features from these signals. The features were extracted both taking into account the whole excerpt and by dividing the musical file into smaller windows. Audio features extracted can be grouped into: temporal features, chroma features, spectral features and cepstrum features.

TODO: for each category write the features

#### 3.2.1.1. Chroma STFT

[[LabCourse\_STFT.pdf (audiolabs-erlangen.de)](https://www.audiolabs-erlangen.de/content/05-fau/professor/00-mueller/02-teaching/2016s_apl/LabCourse_STFT.pdf)]

The human perception of pitch is periodic in the sense that two pitches are perceived as similar in “color” (playing a similar harmonic role) if they differ by one or several octaves (an octave = the distance of 12 pitches). A pitch can be separated into two components, which are referred to as *tone height* and *chroma*. The tone height refers to the octave number and the chroma to the respective pitch spelling attribute. In Western music notation, the twelve chroma values are given by the set:

*C, C#, D, D#, E, F, F#, G, G#, A, A#, B*

A pitch class is defined as the set of all pitches that share the same chroma; for example, the pitch class corresponding to the chroma *c = 0* (*C*) consists of the set {0, 12, 24, 36, …} == {C0, C1, C2, C3, …}. The main idea of chroma features is to aggregate all spectral information that relates to a given pitch class into a single coefficient. Chroma features can be significantly changed by introducing pre/post-processing steps that modify spectral, temporal and dynamical aspects. The following chromagram is obtained by performing Short Time Fourier Transform (SFTF):

**

Figure 3.2.1.1.1. Chromagram on time window

#### 3.2.1.2. Root-Mean-Square (RMS)

The RMS value of an audio signal is a measure of its overall energy or volume level. It is a measure of the amplitude of the audio signal over time. The RMS is a broad measure of an audio signal’s volume and it does not take into account human perception of loudness. Human ears perceive some frequencies as being louder than others, even if they have the same RMS level. Mathematically, since the audio signals are often represented as continuous-time signals over a period , the RMS value is computed as follows:

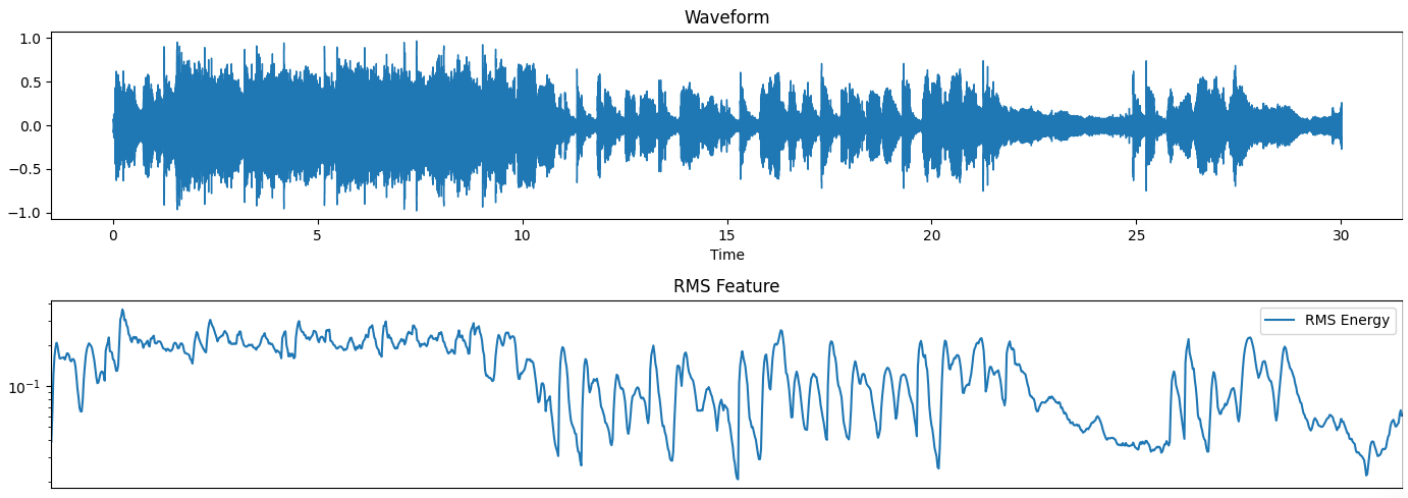


Figure 3.2.1.2.1. RMS and waveform

#### 3.2.1.3. Spectral centroid

Spectral centroid is a measure to characterize a spectrum and indicates where the ‘center of mass’ for a sound is located. In other words, it gives the frequency band where the most of the energy is concentrated. It maps into a very prominent timbral feature called “brightness of sound” (energetic, open, dull). Mathematically, it is computed as the weighted mean of the frequency bins present in the audio signal, determined using a Fourier transform, with their magnitudes as the weights:

: the weighted frequency value (magnitude) of bin number

: the center frequency of that bin

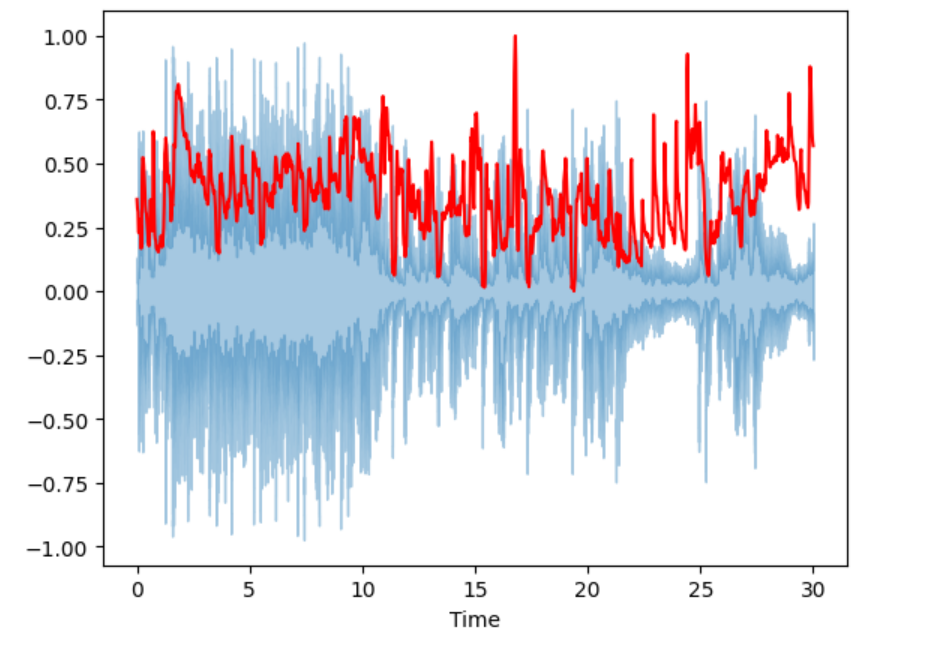


Figure 3.2.1.3.1. Spectral centroid and waveform

#### 3.2.1.4. Spectral bandwidth

The spectral bandwidth or spectral spread is derived from the spectral centroid. It is the spectral range of interest around the centroid, that is, the variance from the spectral centroid. It has a direct correlation with the perceived timbre. The bandwidth is directly proportional to the energy spread across frequency bands. It provides a measure of the spectral complexity of the sound: narrow-band signals (pure tones) have a low spectral bandwidth, whereas wide-band signals (like noise) have high spectral bandwidth. Mathematically, it is the weighted mean of the distances of frequency bands from the Spectral Centroid:

: the weighted frequency value (magnitude) of bin number

: the center frequency of that bin

: the spectral centroid

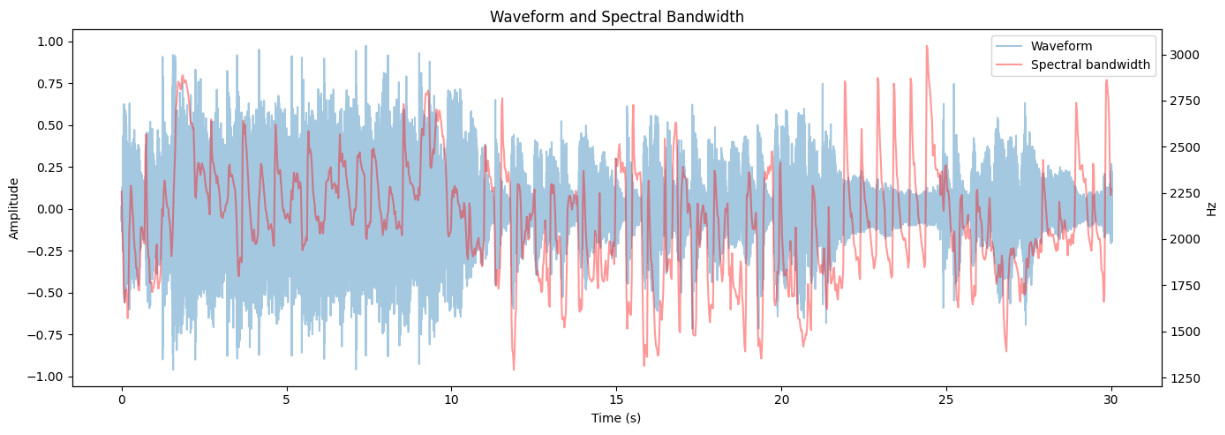


Figure 3.2.1.4.1. Spectral bandwidth and waveform

#### 3.2.1.5. Spectral Rolloff

Spectral rolloff is defined as the percentile of the power spectral distribution, where is usually 85% or 95%. The rolloff point is the frequency below which the N of the magnitude distribution is concentrated. For example, this can be used to approximate the maximum / minimum frequency by setting roll percent to a value close to 1/0. It is a measure of the shape of the signal and represents the frequency below which the specified percentage of the total spectral energy lies.

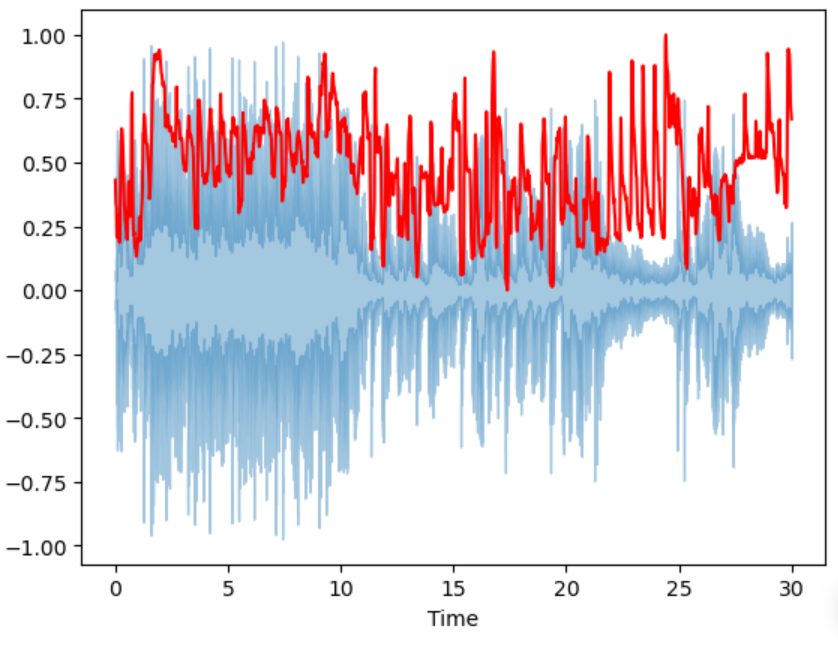


Figure 3.2.1.5.1. Spectral rolloff and waveform

#### 3.2.1.6. Flatness

Spectral flatness (or tonality coefficient, or Wiener entropy) is a measure used to characterize the audio spectrum. It provides a way to quantify how noise-like versus how tone-like a sound is. It is defined as the ratio of the geometric mean to the arithmetic mean of a power spectrum. If the spectrum is equally distributed (the same amount of power in all frequency bands), it is considered ‘flat’ and the spectral flatness is close to 1. Otherwise, if the spectrum is peaky (more power in some frequency bands than in others), it is considered ‘sharp’ and the value of the feature is close to 0. The value of the spectral flatness is computed as follows, where represents the power of the *i*-th frequency bin:

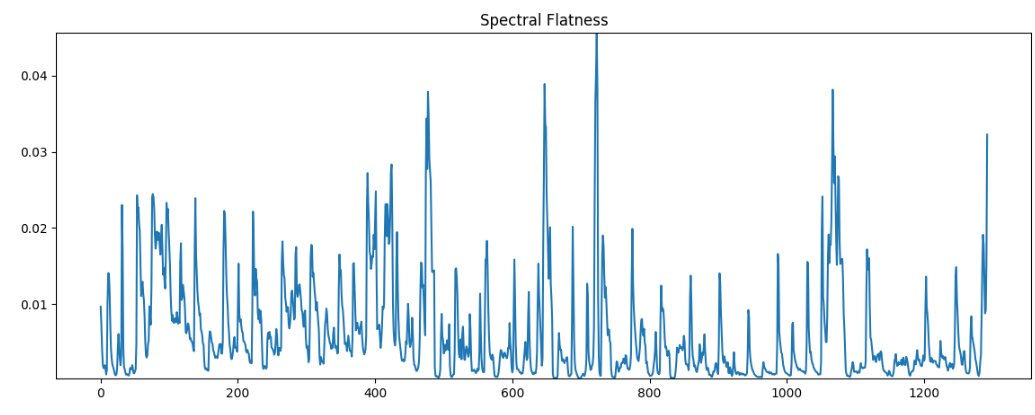


Figure 3.2.1.6.1. Spectral flatness

#### 3.2.1.7. Contrast

Spectral contrast is a feature used to classify music genres. Spectral contrast is expressed as the difference in decibels between peaks and valleys in the frequency spectrum, which can represent the relative spectral characteristics of music. It can be seen from the experimental results [D. N. Jiang, L. Lu, H. J. Zhang, J. H. Tao, and L. H. Cai, “Music type classification by spectral contrast feature,” in *Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 113–116, IEEE, Lausanne, Switzerland, August 2002] that the spectral contrast has a good ability to discriminate music genres.

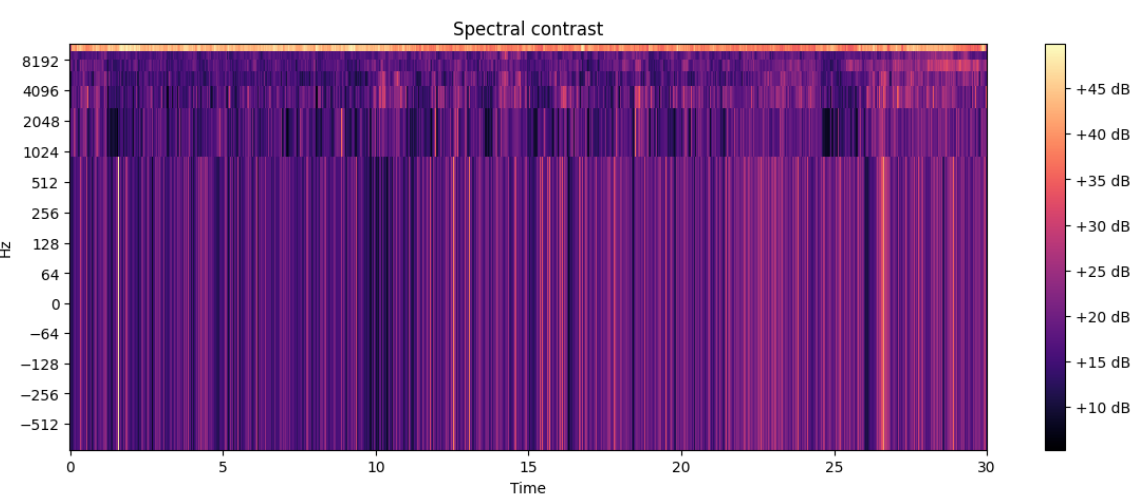


Figure 3.2.1.7.1. Spectral contrast log-scaled

#### 3.2.1.8. Flux

Spectral flux measures the spectral change between two successive frames and is computed as the squared difference between normalized magnitudes of the spectra of the two successive short-term windows. It is a measure of how quickly the power spectrum of an audio signal is changing. It can be used to detect the onset of musical notes or beats. A high spectral flux might indicate a percussive sound, while a low flux might indicate a sustained note. In order to compute the value, the time-domain signal must be transformed into the frequency domain, by using either a Discrete Fourier Transform (DFT) or a Fast Fourier Transform (FFT). Mathematically, the formula is as follows:

where is the spectral flux at time ‘t’ and is given by the sum over all frequencies ‘f’ in the analysis window. The power spectrum is given by , with being the Fourier transformed signal at time ‘t’ and frequency ‘f’.

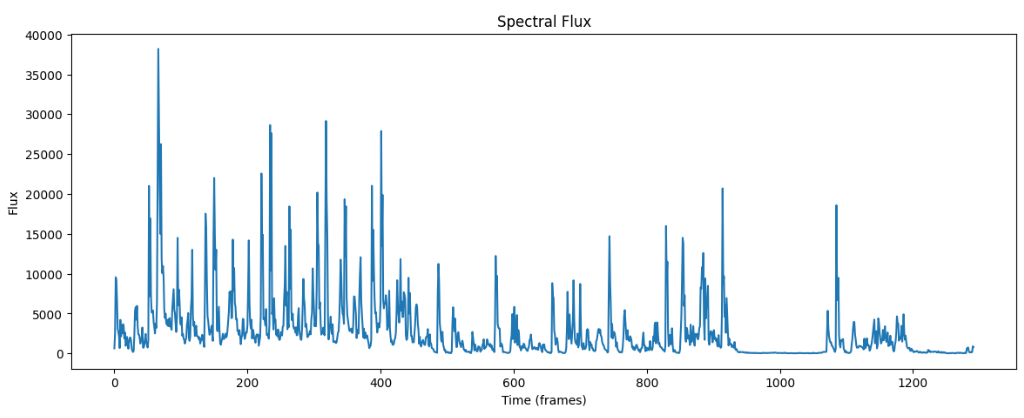


Figure 3.2.1.8.1. Spectral flux for each time frame

#### 3.2.1.9. Zero crossing rate (ZCR)

The ZCR is the rate of sign-changes along a signal (positive-zero-negative or negative-zero-positive). It has higher values for highly percussive sounds like those in metal and rock. The ZCR is computed as follows, where *T* is the length of the time window, is the magnitude of the time domain sample:

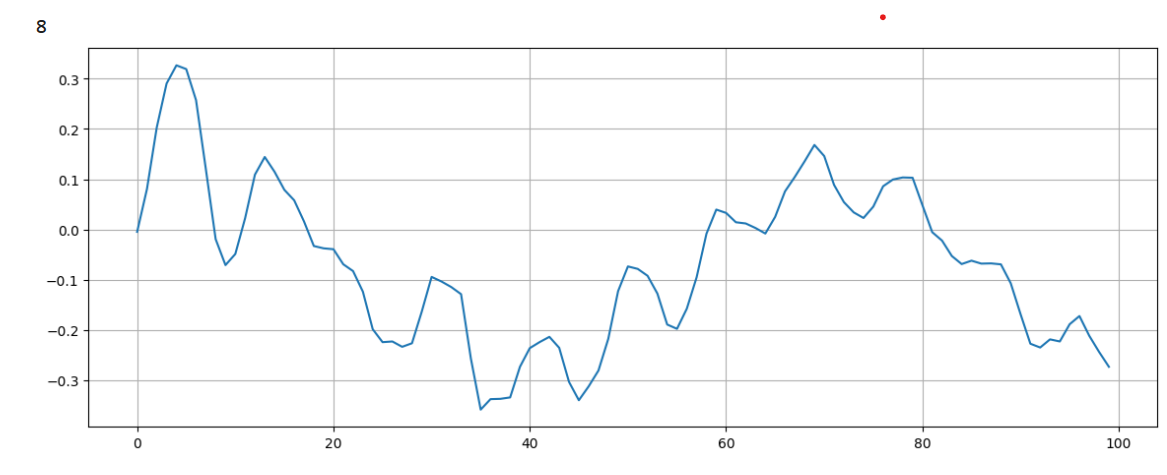


Figure 3.2.1.9.1. ZCR extracted from a time window

#### 3.2.1.10. Harmonics and Perceptrual/Percussive

Harmonics represent the multiple frequencies that are integer multiples of a fundamental frequency. When a musical instrument or a vocal chord produces a tone, it is not just a single frequency being produced, but a complex combination of tones known as the harmonic series. These overtones shape the timbre of the sound, contributing to the unique character that allows to distinguish between different types of sounds, even if they share the same pitch.

Perceptual audio features are attributes of audio that the human ear can perceive directly. Key perceptual features include loudness, pitch, timbre and spatial location. Timbre, often referred to as the 'color' or 'texture' of a sound, is largely determined by the balance of harmonics in the sound and it is what allows to distinguish between different instruments or voices even when they are playing or singing the same note.

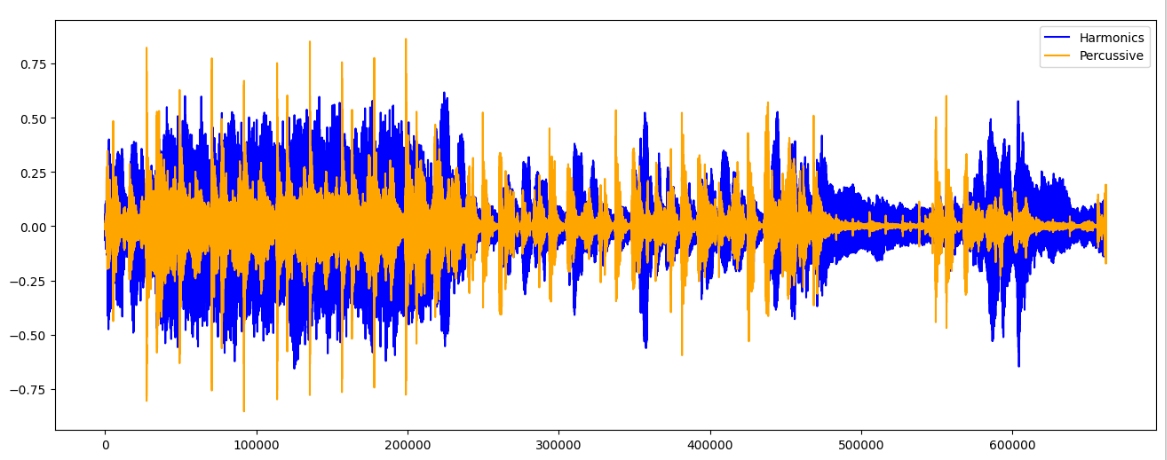


Figure 3.2.1.10.1. Harmonics and Perceptrual features

#### 3.2.1.11. Tempo

In musical lingo, tempo is a that is used to describe the speed at which a song is played. In classical music, it is typically indicated at the beginning of a piece and it is usually measured in Beats Per Minute (BPM). The tempogram is a feature that represents the rhythmic structure of a musical piece in time, showing how the tempo of the track changes over time.

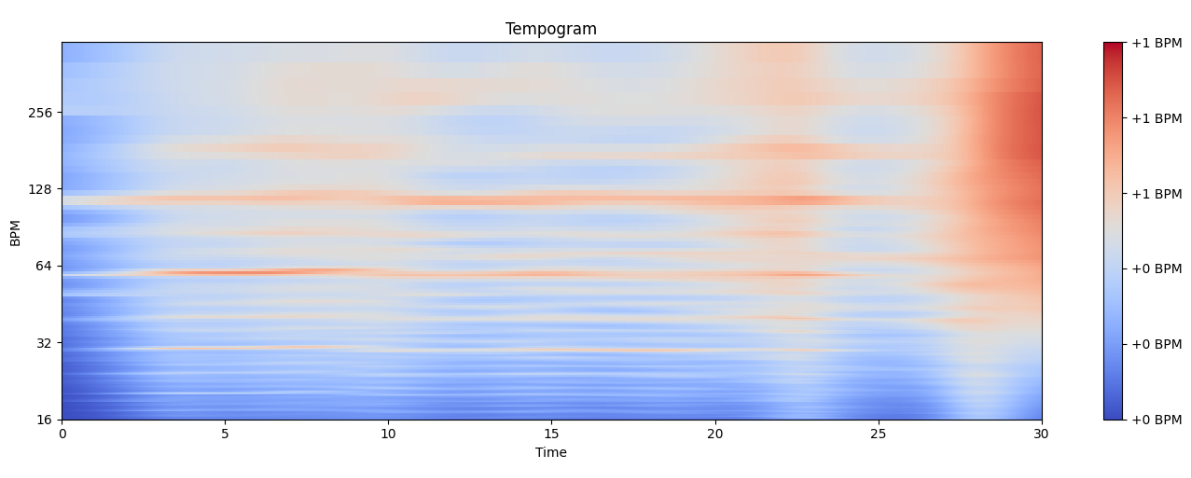


Figure 3.2.1.12.1. Tempogram of an audio file

#### 3.2.1.12. Mel Frequency Cepstral Coefficients (MFCC)

Mel Frequency Cepstral Coefficients are a set of features based on human auditory perception, particularly the frequency intervals at which human ears discern individual tones. They are computed in the Mel Scale, which is a perceptual scale of pitches that approximates the human ear’s response to different frequencies, equal distances in this unit of pitch sound equally distant to the listener. What makes MFCC very useful is that they provide a robust representation of the spectral shape, which not only reduces computational requirements, but also helps to suppress noise.

MFCC is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. To explain this statement, several terms must be defined:

* a sound spectrum is a representation of a sound in terms of the amount of vibration at each individual frequency;
* sine and cosine transforms can be easily computed using fourier transform;
* the power spectrum is the result after the fourier transform, also known as periodgram;
* the frequency is scaled in mel scale in order to match what the human ear can hear: ;

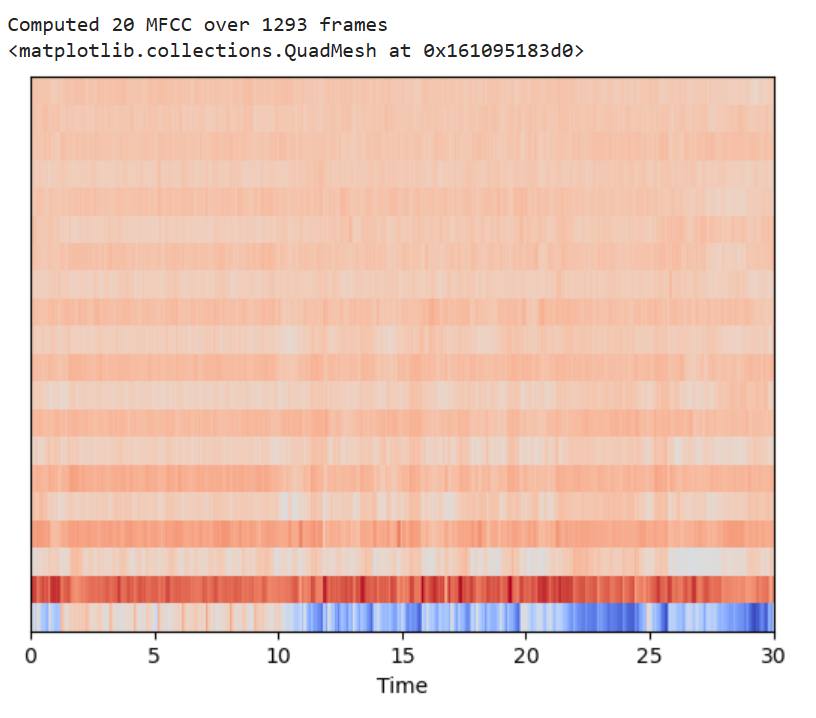


Figure 3.2.1.12.1. Mel Frequency Cepstral Coefficients

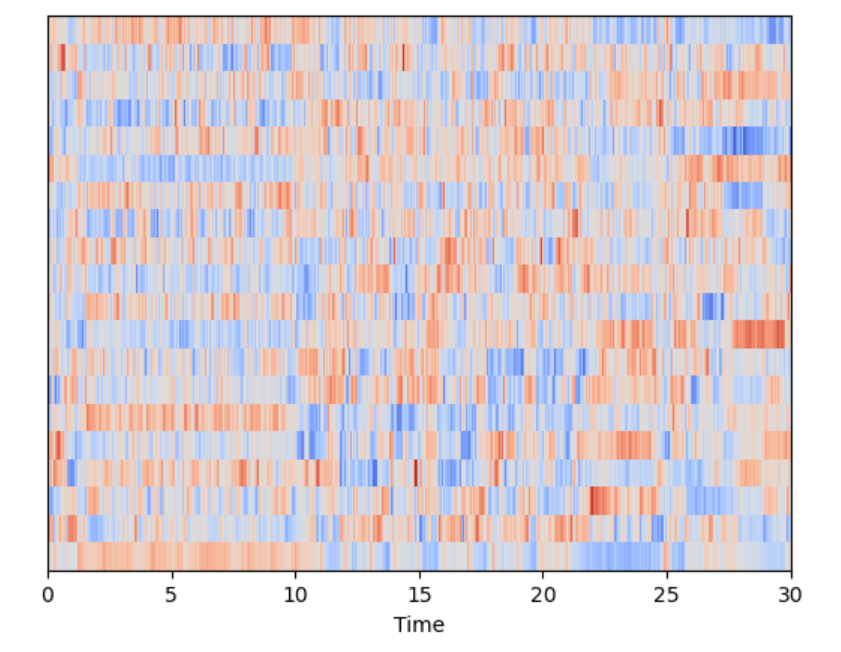


Figure 3.2.1.12.2. MFCC scaled to zero mean and unit variance

#### 3.2.1.13. Mel Spectrogram

[[Understanding the Mel Spectrogram | by Leland Roberts | Analytics Vidhya | Medium](https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53)]

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it changes over time. The plot is created by showing frequency on one axis, time on another and intensity (magnitude) of the frequencies at each point in time on a third axis, which is often represented as color. Standard spectrograms have a significant limitation for human-centered tasks as they display the frequency on a linear scale. Human auditory perception is more accurately modeled with logarithmic frequency scale, especially at lower frequencies. The Mel scale is a perceptual scale of pitches that approximates the human ear’s response to different frequencies. Thus, a Mel spectrogram is a spectrogram where the frequency is Mel-scaled in order to match human perception. To create a Mel spectrogram, the first step is to take the Fourier transform of the signal to obtain the spectrum of frequencies, then apply the Mel filterbank to this spectrum to obtain the result. The Mel filterbank is a set of triangular filters that are spaces according to the Mel scale and applying them amounts to calculating the sum of the energy in each filter.

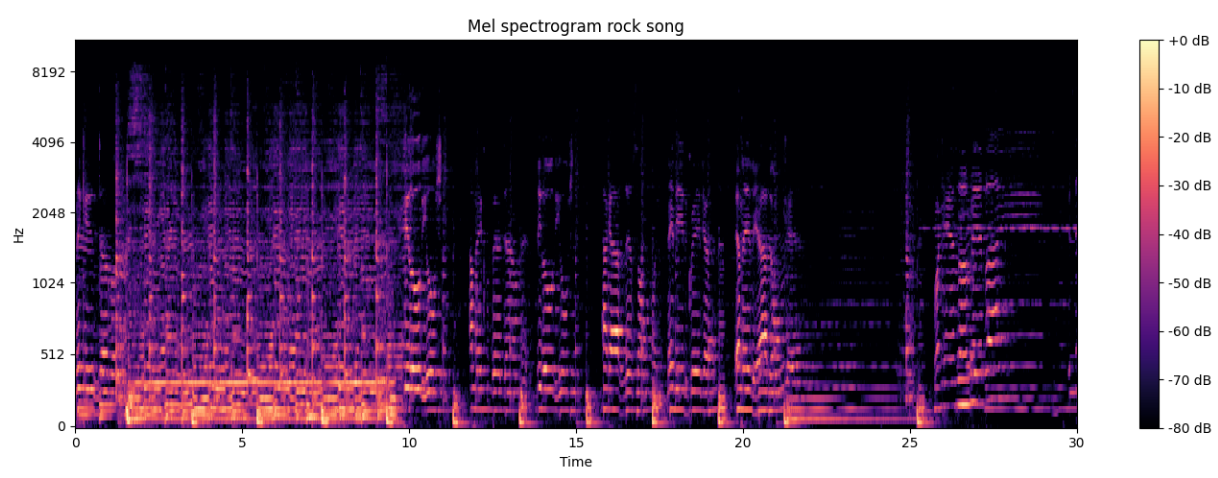


Figure 3.2.1.13.1. Mel Spectrogram

### 3.2.2. Machine Learning

#### 3.2.2.1. KNN

#### 3.2.2.2. SVM

#### 3.2.2.3. Random Forest

#### 3.2.2.4. Logistic Regression

#### 3.2.2.5. Naïve Bayes

#### 3.2.2.6. XGBoost

### 3.2.3. Deep Learning

#### 3.2.3.1. CNN

#### 3.2.3.2. RNN

#### 3.2.3.3. Transfer Learning

#### 3.2.3.4. Vision Transformers

#### 3.2.3.5. Reinforcement Learning

## 3.3. Music Generation

### 3.3.1. Music notation

### 3.3.2. LSTM

### 3.3.3. Transformer

### 3.3.4. VAE

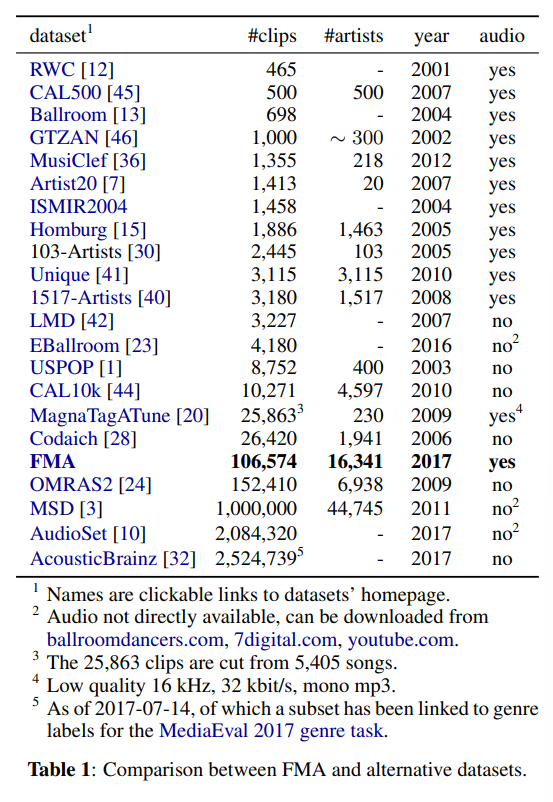
### 3.3.5. GAN

# 4. Proposed framework

## 4.1. Music Genre Recognition

### 4.1.1. Datasets

When seeking the perfect dataset for the music genre classification task, it is essential to consider the size of the dataset, its diversity, balance and the quality of the genre labels. A synthesis of many musical datasets is presented in [[1612.01840.pdf (arxiv.org)](https://arxiv.org/pdf/1612.01840.pdf)]:



but the experiments took into account only three of the most commonly utilized datasets for this purpose: GTZAN [[GTZAN Dataset - Music Genre Classification | Kaggle](https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification)], Free Music Archive (FMA) [[mdeff/fma: FMA: A Dataset For Music Analysis (github.com)](https://github.com/mdeff/fma)] and the Million Song Dataset (MSD) [[Getting the dataset | Million Song Dataset](http://millionsongdataset.com/pages/getting-dataset/)]. Each dataset has it unique strengths and weaknesses, that were carefully analyzed before choosing the perfect candidate for analysis and experimentation.

#### 4.1.1.1. GTZAN

Usually, GTZAN is the first choice for music genre recognition tasks, primarily due to its simplicity, reduced size and balanced structure. It provides a neat, evenly distributed collection of a thousand audio tracks across ten popular genres. Because of the balanced genres, the need for additional preprocessing is eliminated; otherwise, the class imbalance could have caused bias in the models. The collection contains a folder with the 30 seconds audio files, each subfolder is labeled with one of the 10 genres and contains the respective 100 files. The other folder contains the visual representation of each audio file (Mel Spectrograms), also distributed in 10 balanced subfolders. Furthermore, there are two CSV files containing features of the audio files. One has the mean and variance computed over multiple features extracted from the entire 30 seconds songs, while the other has the same structure, but the songs were split into 3 seconds, increasing 10 times the amount of data fed to the classification models. Even if it seems that 3 seconds windows cannot model properly an entire genre, the results tend to agree with the motto “the more data the better”. There are 29 features that can be used for the classification task (the columns in the tables). The filename, length and the label of the song will not be used in the models, so they are removed from the tables before any data processing.

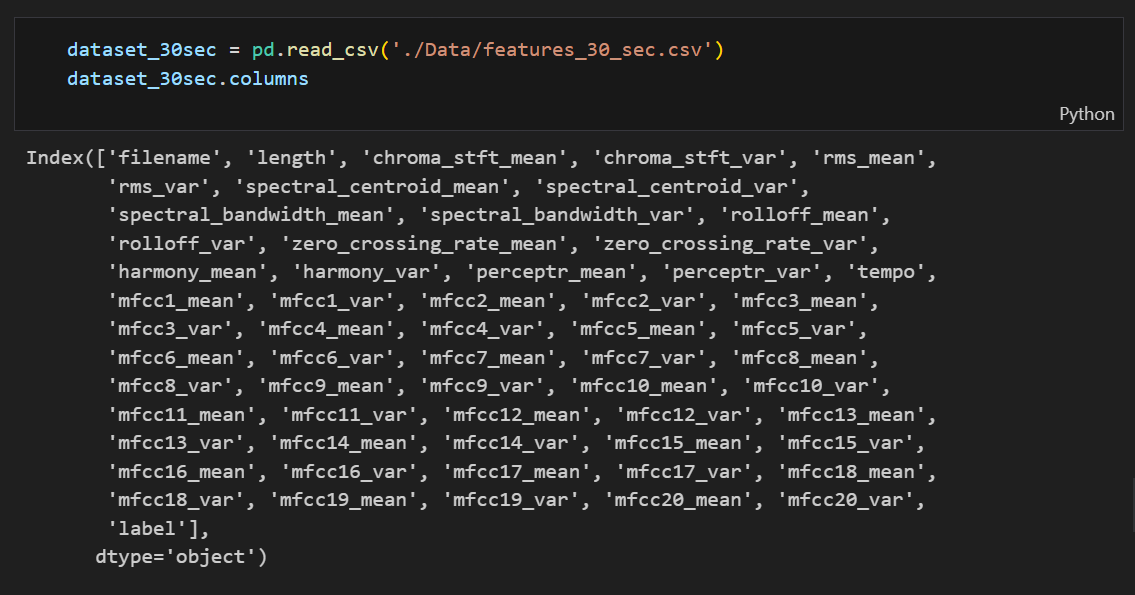


Figure 4.1.1.1.1. The columns of the CSV files

The relationship between the variables, only their mean being taken into consideration, can be seen through the correlation heatmap. From here, the process of data cleaning and feature selection can begin, looking at the “heat” patterns. It can be observed that there are several variables exhibiting a positive correlation and a few that display an inverse correlation.

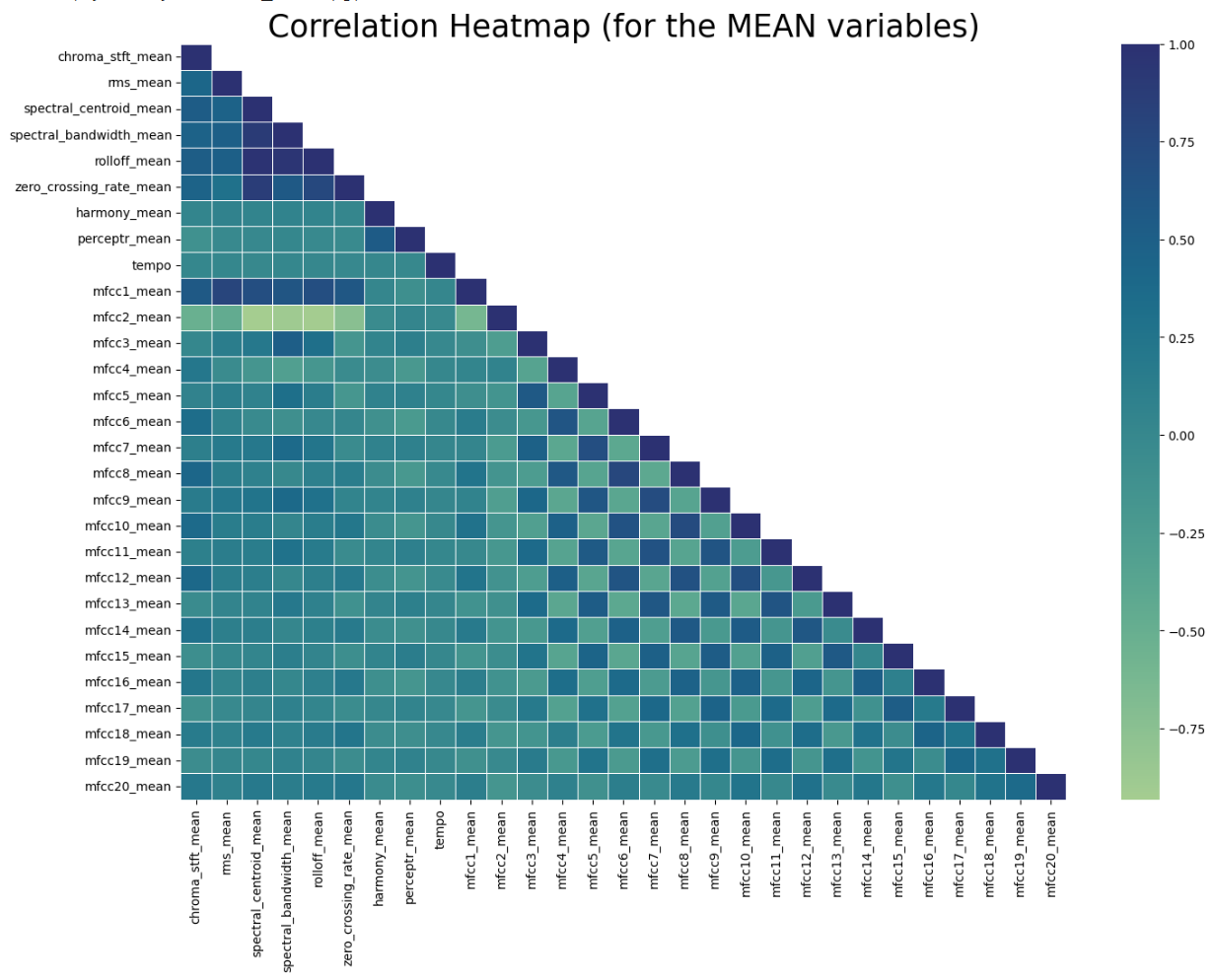


Figure 4.1.1.1.2. Correlation Heatmap for audio features

In order to visualize possible clusters of genres, Principal Component Analysis has been made. PCA is sensitive to scaling, so it is important that the variables are filtered (filename removed) and properly normalized before applying the analysis. The first two principal components often explain a significant proportion of the data’s variation and can indicate whether the data can be represented in this reduced two-dimensional space without losing much information. The colored PCA points may naturally form clusters, which could correspond to different categories and could indicate underlying patterns or structures.

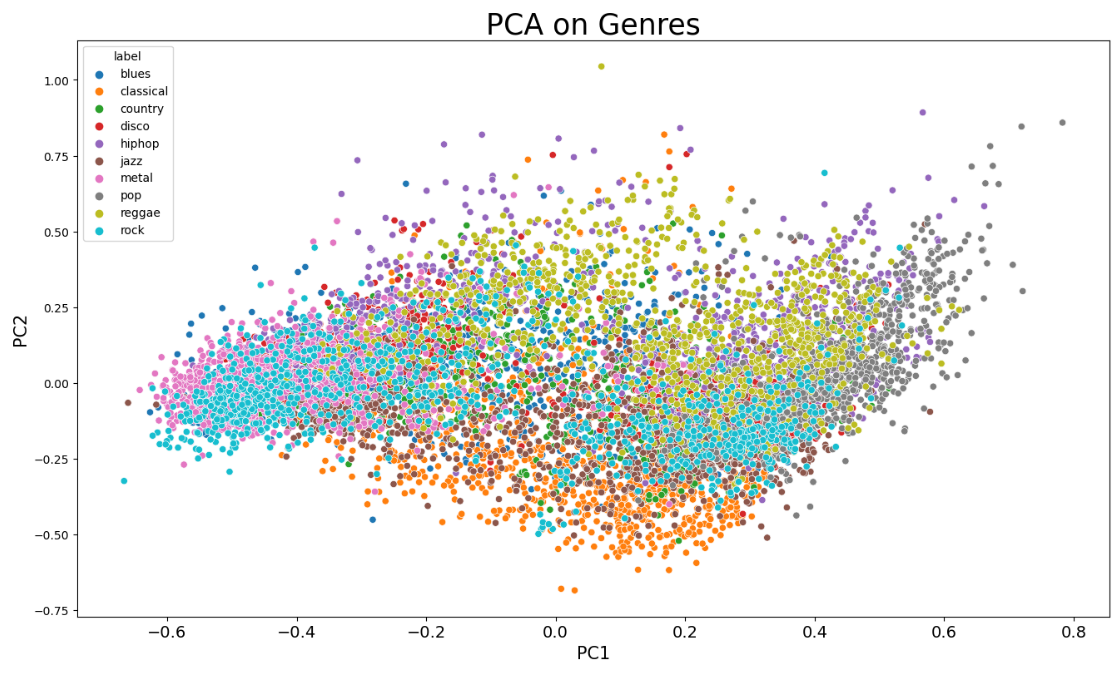


Figure 4.1.1.1.3. PCA clusters for each genre

In what concerns the folder that contain images, there are clear differences in the corresponding Mel Spectrograms, which can lead to an efficient use of Deep Learning techniques for classifying these visual representations into musical genres. The original files are 288x432 which may influence the final results when using Transfer Learning methods, so the images were re-generated in an even format of 256x256.

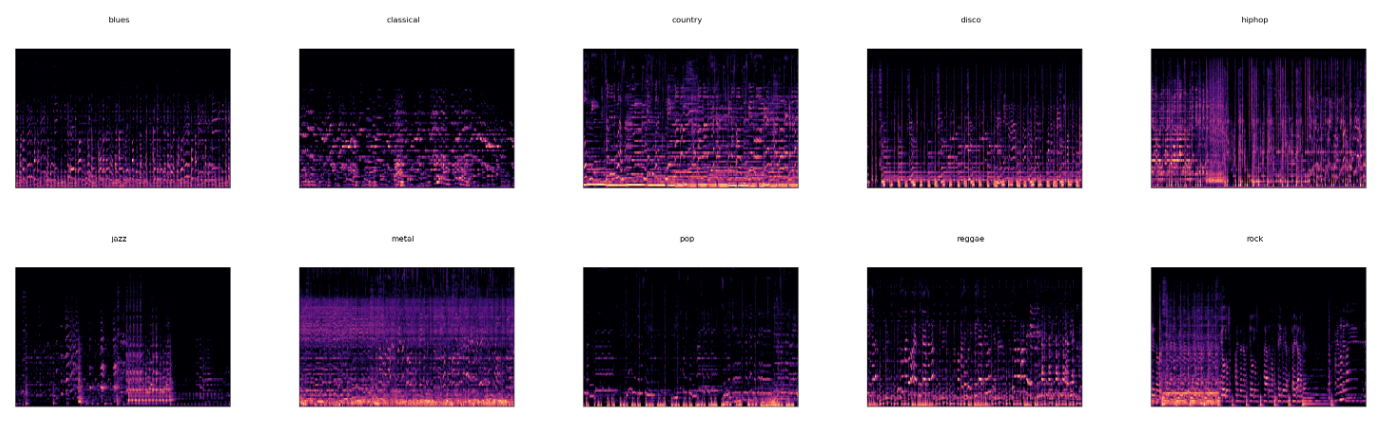


Figure 4.1.1.1.4. Mel Spectrograms for each of the 10 genres

While working with the spectrograms, it has been observed that the ‘jazz’ folder contains only 99 images, instead of 100 for each audio file. Digging a bit into the problem, it seems that the file ‘jazz00054.png’ is missing from the dataset and the audio file ‘jazz.00054.wav’ is corrupted. In order to maintain the balance between the 10 genres, a spectrogram from the same genre has been duplicated, in hopes that it does not affect the performance of the model too much. Despite its widespread use in the field of MIR, [[1306.1461.pdf (arxiv.org)](https://arxiv.org/pdf/1306.1461.pdf)] it is indicated that the GTZAN is faulty and has an unexpected bias on the performance of the models and potentially skews the results. The most significant issue with the dataset is the presence of duplicate and mislabeled audio tracks. Around 6% of the songs are duplicates and several others are mislabeled with incorrect genres, which can lead to confusion in the training and even inflate the accuracy of the models. Furthermore, the distribution of the dataset across the 10 defined genres is balanced, which helps the process of data preprocessing but does not reflect the actual distribution of genres in the real world. Music is highly diverse and does not neatly fall into only 10 categories. Real-world systems would need to handle a far greater variety of music of varying quality and in a range of different environments. However, for the sake of computational efficiency it is enough to limit the musical spectrum to the most popular genres.

#### 4.1.1.2. FMA

[[1612.01840.pdf (arxiv.org)](https://arxiv.org/pdf/1612.01840.pdf)]

The Free Music Archive (FMA) dataset is a richly annotated, high-quality and diverse collection of music that can be utilized for various research tasks related to music analysis, including genre tagging. It provides several levels of annotation, for each track there is a unique ID, title, album and release year. Additionally, there are track genres and sub-genres, providing a robust groundwork for a broad spectrum of music analysis tasks. Tracks are categorized according to a hierarchical taxonomy that spans 16 top-level genres (like Pop, Rock, Electronic), which are further broken down into 161 sub-genres (such as Psych-Rock, Lo-Fi, Drone).

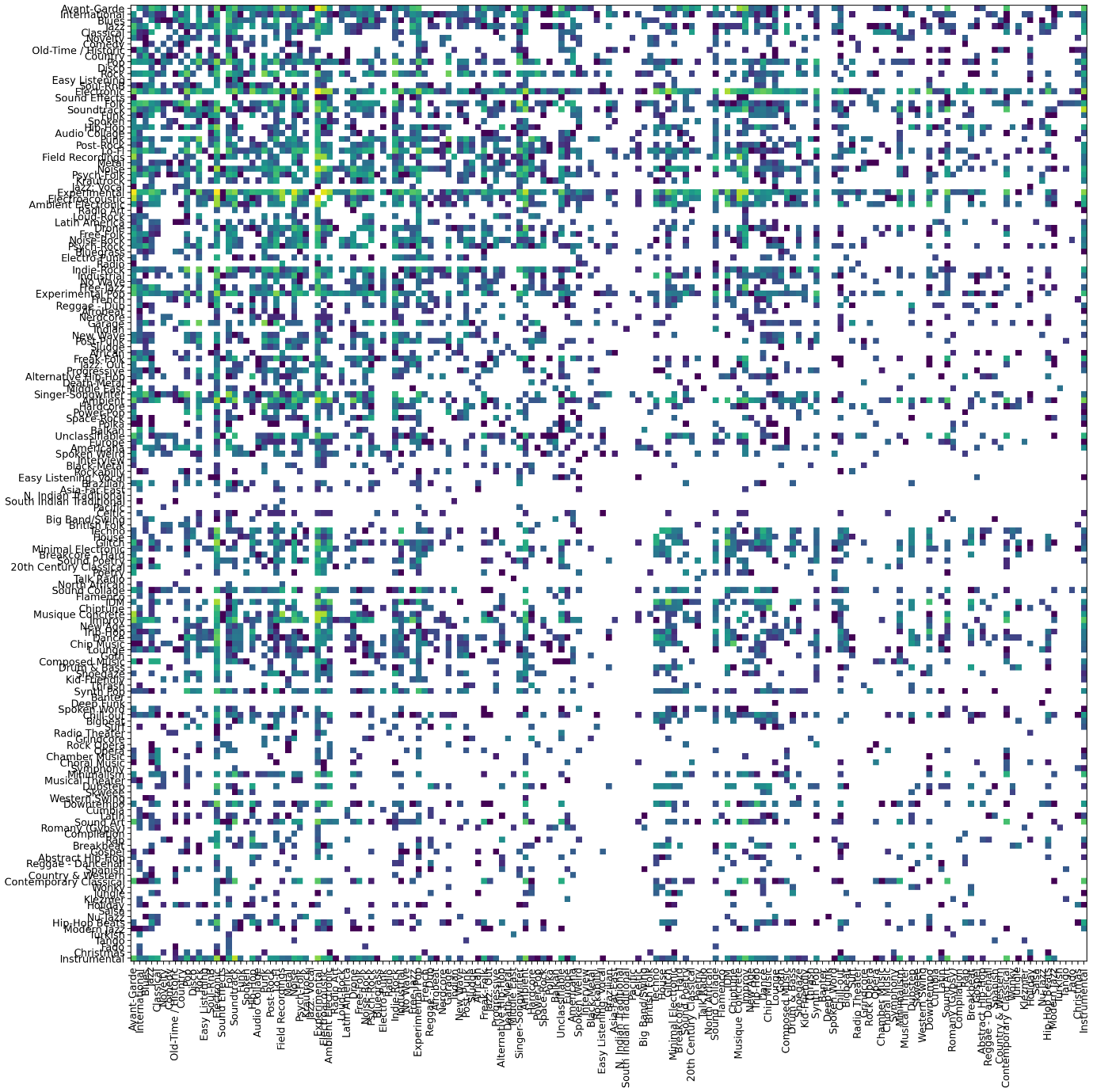


Figure 4.1.1.2.1. Related genres by cross-appearance

One important feature of this dataset is its availability in subsets of different sizes: small, medium, large and full version. Starting from the small subset which is featuring 8000 clips it stretches to the full-length version with more than 106000 tracks, making this dataset suitable for a variety of projects, from personal small scale studies to large scale experiments.

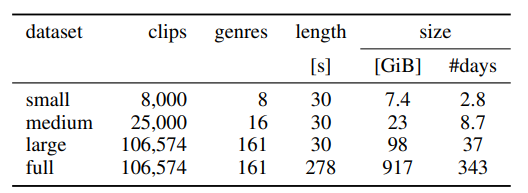


Figure 4.1.1.2.1. Subsets of the FMA

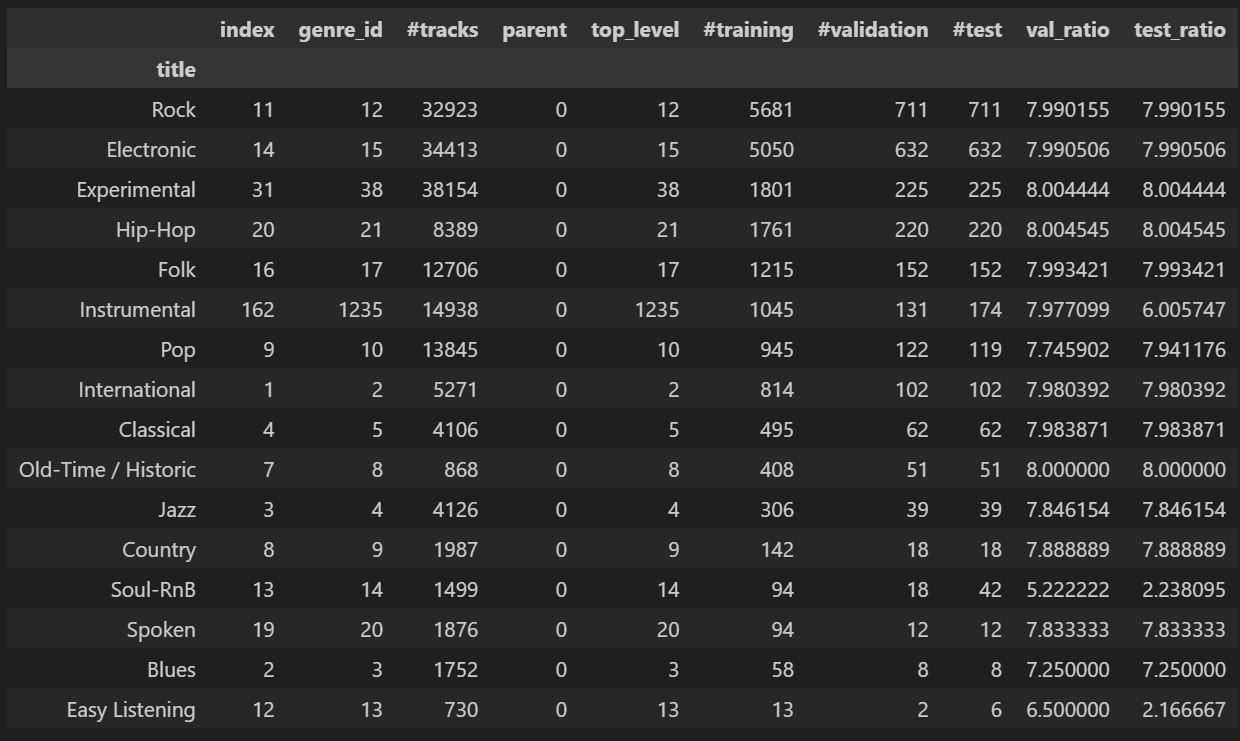


Figure 4.1.1.2.2. FMA medium partition details

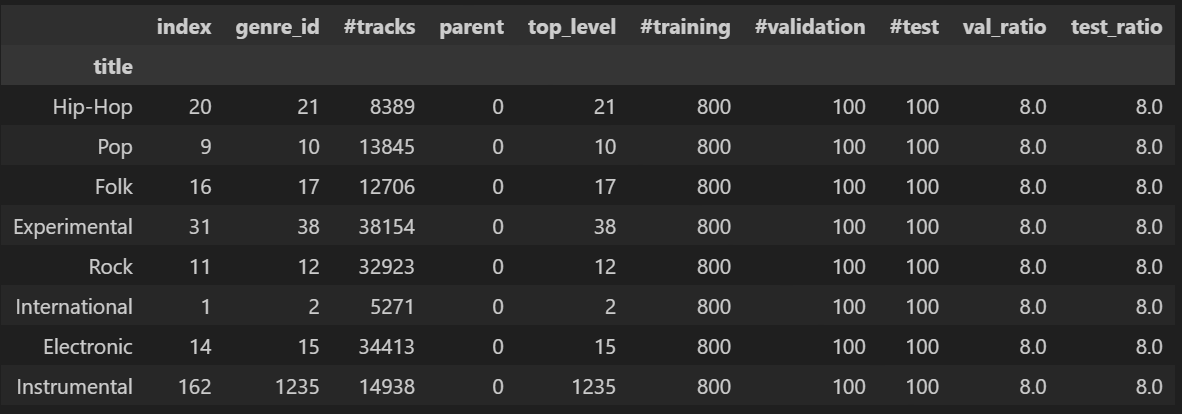


Figure 4.1.1.2.3. FMA small partition details

In terms of audio features, the dataset provides a rich selection of precomputed features using librosa library, including chroma, spectral features, tonnetz, mfcc and others. Each feature set (except zero-crossing rate) is computed on windows of 2048 samples spaced by hops of 512 samples, resulting in seven statistics over all windows: mean, standard deviation, skew, kurtosis, median, minimum and maximum.

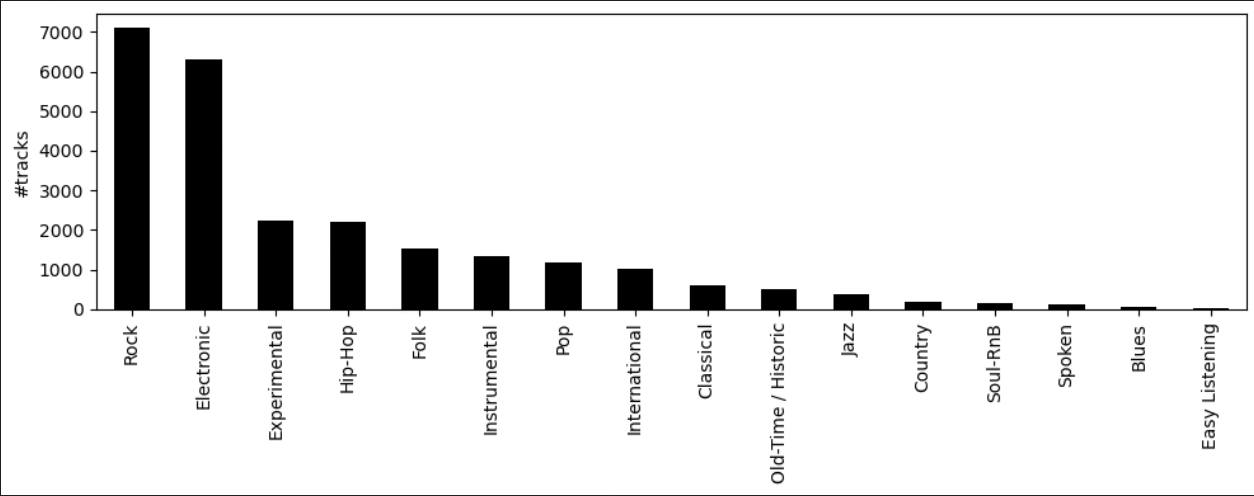


Figure 4.1.1.2.4. FMA medium dataset genres distribution

While experimenting with spectrograms for the medium dataset, several files were found to be erroneous, mostly because of the audio file’s length, which did not match with the 30 seconds that each of the other files had. In addition to this, there is also a ticket on Github regarding this issue: [https://github.com/mdeff/fma/issues/8].

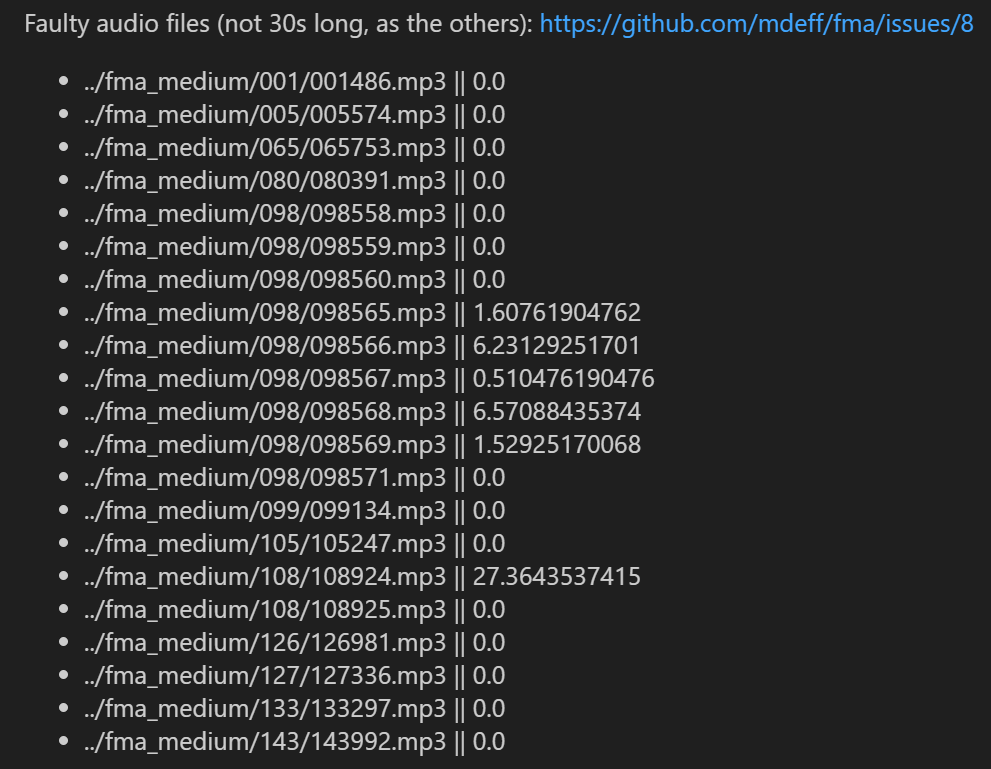


Figure 4.1.1.2.5. FMA dataset faulty audio files

After an in-depth exploratory data analysis, the conclusion is that the small partition of the FMA dataset is too small, containing only 8 genres that are not inclusive enough (classical, blues and other popular genres missing), while the medium subset is way too unbalanced and too large to be computational efficient.

#### 4.1.1.3. MSD

The Million Song Dataset (MSD) is a publicly accessible collection of audio features and metadata for a million contemporary music tracks. With a million songs included, its sheer size offers an extensive range of data for all kind of tasks in music research, including music genre classification. Each song includes data points like the release year, the artist, the popularity, the key, tempo or duration. In addition to this, there are segments, bars and beats indicating the rhythm and the timbre of audio segments, which give an idea of the sound color and melody. The data for each track has been computed using an API called The Echo Nest’s, which gives objective information about each song, unaffected by subjective factors like personal opinion or cultural context.

However, there are also a few drawbacks to this dataset. Firstly, it includes only metadata and precomputed features, it does not contain the actual audio files for the songs, due to copyrights reason. For researching with deep learning techniques using melspectrograms, this dataset is limited. Secondly, the MSD is heavily weighted towards popular Western music. Even if it offers an extensive range of data for this segment, it does not provide a diverse or representative sample of all genres within the world music.



Figure 4.1.1.3.1. MSD all genre labels

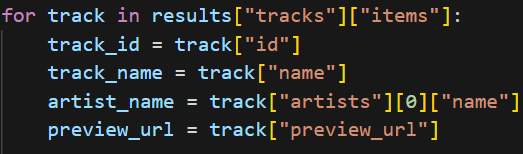
#### 4.1.1.4. Spotify

While GTZAN, FMA and MSD are well-known datasets used in MIR, they do not always meet specific requirements for certain projects. For instance, GTZAN is relatively small with only 1000 audio tracks and is criticized for its inaccuracies and duplications. The FMA dataset is larger and provides a broad range of metadata, but its imbalance regarding genres distribution does not align with this project’s needs. MSD is even more expansive, containing a wealth of audio features and metadata for a million songs, but accessing the audio for these tracks is a challenging task. Thus, there may be a need to build a more up-to-date, easily accessible dataset from scratch.

The Spotify API is a rich source of music data, providing access to a wide variety of information such as track details, artist information, album details, playlists and even audio analysis and features, but the most important thing is that it contains genre labels for each track, making it a suitable source for building up a dataset for music genre recognition. Downloading from the Spotify API requires client credentials obtained after making an account on the Spotify Developer platform [[Home | Spotify for Developers](https://developer.spotify.com/)]. The query used to extract audio tracks is the following:



, where the search is done according to a musical genre and the result shows the Spotify page starting with the **offset** song and following with other 50 (**limit**) songs. As shown:



only the track name, its artist and an audio preview of 30 seconds are saved from the query’s result. The genres used to quiz the API are the following: blues, classical, country, disco, reggae, metal, hip-hop, jazz, pop, rock. It can be seen that the list is very similar to the one used in GTZAN, in order to make pertinent comparisons between the 2 datasets. The final version of the dataset contains two folders, **Audio\_Spotify** with 500 audio tracks per genre and **Audio\_Spotify\_Test** with 300 tracks per genre and can be found here [[Spotify Dataset | Kaggle](https://www.kaggle.com/datasets/pricoptudor/spotify-dataset)].

## 4.2. Music Generation

# 5. Evaluation

## 5.1. Music Genre Recognition

## 5.2. Music Generation

# 6. Conclusion and future work

# 7. Bibliography