



Università degli Studi di Milano

FACOLTÀ DI SCIENZA POLITICHE, ECONOMICHE E SOCIALI
Corso di Laurea Magistrale in Data Science and Economics

Exploitation of Lyrics, Music and Vocals in Multimodal Song Classification Systems

Relatore:

Chiar.mo Prof. Alfio FERRARA

Correlatore:

Chiar.ma Prof.ssa Silvia SALINI

Laureando:

Sergio PICASCIA

Matricola 943865

Anno Accademico 2020-2021

Words make you think.
Music makes you feel.
A song makes you feel a thought.

—YIP HARBURG

Contents

1	Introduction	13
2	Related Work	15
2.1	Emotion Detection	15
2.1.1	Definition of Emotion and Related Terms	16
2.1.2	Representation of Emotions	16
2.1.3	Music Emotion Recognition	17
2.1.4	Music Mood Dataset	18
2.2	Music Information Retrieval	19
2.2.1	Music Related Features	20
2.2.2	Vocals Related Features	21
2.2.3	Lyrics Related Features	21
2.3	Multimodal Systems	23
2.3.1	Low-Level Fusion Approaches	23
2.3.2	High-Level Fusion Approaches	24
3	Methodologies	25
3.1	Feature Extraction	26
3.1.1	Audio Feature Extraction	26
3.1.1.1	Vocals Feature Extraction	27
3.1.1.2	Music Feature Extraction	28
3.1.2	Lyrics Feature Extraction	29
3.1.2.1	Lyrics Format Variations	29
3.1.2.2	Word Embeddings	30
3.2	Model Training	31
3.2.1	Single-Input Models	31

3.2.2	Multimodal Models	32
3.2.3	Emotion-Genre Relation	32
4	Feature Extraction	33
4.1	Data Retrieval	33
4.1.1	Lyrics Retrieval	34
4.1.2	Audio Retrieval	36
4.1.3	Genre Retrieval	36
4.2	Audio Preprocessing	37
4.2.1	Chorus Detection	38
4.2.2	Vocal Separation	40
4.2.3	Vocals Feature Extraction	40
4.2.4	Music Feature Extraction	40
4.3	Text Preprocessing	42
4.3.1	Extractive Text Summarization	43
4.3.2	Abstractive Text Summarization	43
4.3.3	Word Embeddings	45
5	Model Training	47
5.1	Single Input Models	47
5.1.1	Vocals Models	48
5.1.2	Music Models	50
5.1.3	Text Models	54
5.1.3.1	Models Using SSWE	54
5.1.3.2	Models Using sBERT	54
5.1.3.3	Models Using SSWE and sBERT	59
5.2	Multimodal Models	62
5.2.1	High-Level Fusion Models	62
5.2.2	Low-Level Fusion Models	63
5.3	Emotion-Genre Relation	67
5.3.1	Correlation Analysis	67
5.3.2	Conditional Probabilities	68

<i>CONTENTS</i>	7
-----------------	---

6 Conclusion	71
6.1 Research Findings	71
6.2 Future Developments	73
Bibliography	75

List of Figures

3.1	A diagram illustrating the workflow of the entire project.	25
3.2	The approach to audio processing.	27
3.3	The approach to text preprocessing.	29
3.4	The model training phase.	31
4.1	Thayer’s emotions scheme.	34
4.2	Example of a time-lag similarity matrix used to identify repeated sections.	39
4.3	The three waveforms of a song used for information retrieval: complete, 30 seconds and 10 seconds excerpts.	39
5.1	Confusion matrix for the best classifier of emotion using vocal features.	49
5.2	Confusion matrix for the best classifier of genre using vocal features.	51
5.3	Confusion matrix for the best classifier of emotion using music features.	52
5.4	Confusion matrix for the best classifier of genre using music features.	53
5.5	Confusion matrix for the best classifier of emotion using SSWE features.	55
5.6	Confusion matrix for the best classifier of genre using SSWE features.	56
5.7	Confusion matrix for the best classifier of emotion using sBERT features.	57
5.8	Confusion matrix for the best classifier of genre using sBERT features.	58
5.9	Confusion matrix for the best classifier of emotion using SSWE and sBERT features.	60

5.10	Confusion matrix for the best classifier of genre using SSWE and sBERT features.	60
5.11	Confusion matrix for the best classifier of emotion using multimodal high-level fusion approaches.	63
5.12	Confusion matrix for the best classifier of genre using multimodal high-level fusion approaches.	64
5.13	Confusion matrix for the best classifier of emotion using multimodal low-level fusion approaches.	65
5.14	Confusion matrix for the best classifier of genre using multimodal low-level fusion approaches.	66
5.15	Confusion matrix for the classifier of genre using emotion conditional probabilities.	69

List of Tables

4.1	Some observations from the Music Mood Dataset.	33
4.2	Songs without lyrics.	35
4.3	Tags retrieved for each song with the Spotify API.	36
4.4	Parent genres and their number of occurrences in the dataset. . .	37
4.5	Artists appearing the most in the dataset.	38
4.6	Full set of features extracted from vocals.	41
4.7	Full set of features extracted from vocals.	42
4.8	The four formats considered for each song.	44
5.1	Classifiers trained and their tuned parameters.	48
5.2	Results of emotion classification using vocal features.	49
5.3	Results of genre classification using vocal features.	50
5.4	Results of emotion classification using music features.	52
5.5	Results of genre classification using music features.	53
5.6	Results of emotion classification using SSWE features.	55
5.7	Results of genre classification using SSWE features.	56
5.8	Results of emotion classification using sBERT features.	57
5.9	Results of genre classification using sBERT features.	58
5.10	Results of emotion classification using SSWE and sBERT features.	59
5.11	Results of genre classification using SSWE and sBERT features. .	61
5.12	Results of emotion classification using multimodal high-level fusion approaches.	62
5.13	Results of genre classification using multimodal high-level fusion approaches.	63

5.14 Results of emotion classification using multimodal low-level fusion approaches.	65
5.15 Results of genre classification using multimodal low-level fusion approaches.	66
5.16 Contingency table between genres and emotions.	67
5.17 Results of genre classification using emotion conditional probabilities.	68

Chapter 1

Introduction

Songs play a central role in everyone's life nowadays, having become accessible from everywhere thanks to technological development which allows us to keep entire music libraries inside our pockets. For this reason, songs have also got the attention of numerous researchers from disparate areas, statistics and computer science above all, giving birth to new fields, like music information retrieval (MIR) and music emotion recognition (MER). Several methods for analyzing tracks and extracting valuable knowledge have been developed and the number of applications being built every day is growing exponentially, from recommendation systems to music generators.

This research will focus in particular on the classification of songs according to two target variables, emotion and genre, exploiting three different sources of input, vocals, music and lyrics. The aim of the project will be to compare the performances of models built on one input at a time to the ones of multimodal classifiers, which are instead based on multiple sources of data. For each of the two task, every combination of input will be considered, in order to evaluate the relevance of every data type with the respect to the variable being predicted. Another crucial point will concern the amount of original data to keep for the analysis: extracting portion or summaries could avoid the problem of involving lot of noise in the model training, but it could also result in leaving out relevant knowledge necessary to accomplish the task. Finally, it would be interesting to identify, if present, and eventually evaluate the relationship between emotions and genres, using one another in order to enhance the prediction performances.

In chapter 2, we will explore the literature concerning the process of emotion detection, the field of music information retrieval and the recent studies regarding multimodal systems; particular attention will be given to the notion and representation of emotions, to the dataset employed in this project and to all the existing procedures for extracting features from songs and the methods employed to analyze them. Chapter 3 will be devoted to the explanation of the methodologies adopted in this research, with a particular focus on two major phases: feature extraction, with the description of the approaches used for gathering information from audio and textual data, and model training, with comments about the classifiers built during the analysis. Chapter 4 and 5 will deepen the understanding about these two stages, showing step by step the results obtained. In the end, chapter 6 will present all the relevant findings and will discuss about potential future developments of the project.

Chapter 2

Related Work

The field of emotion detection is often confused with the ones of sentiment analysis and opinion mining: while they may share some techniques in common, the meaning of these terms is totally different. In the following paragraphs we will define precisely what an emotion is, how it differs from other related concepts and in which ways it is possible to represent it. We will discuss the role emotions have in music and which are the methods that can be employed to detect them.

Another field that plays a fundamental role in this research is the one of music information retrieval (MIR), an interdisciplinary area that allows to extract feature from different sources and employ them to solve several machine learning related tasks. In particular, we will focus on the methodologies for retrieving information from three types of input: music, vocals and lyrics.

Given this diversity in data sources, it is important to understand how and at which level of the model training process it is possible to combine them. It is, therefore, needed an overview of the literature about data fusion and multimodal models.

2.1 Emotion Detection

A first fundamental statement must be done about the core topic of this analysis: emotions. As claimed by Russell and Fehr [9], “Everyone knows what an emotion is, until asked to give a definition”, highlighting the difficulties in finding an exhaustive explanation to such a common phenomenon. In the following sections

we will go through the history of the field of emotion detection, underlining the subtle differences with other similar sensations, deepening, in particular, the applications of these researches to the music area. Finally, we will take a look at the dataset used in this work, the study behind it and how it was generated.

2.1.1 Definition of Emotion and Related Terms

There are several definitions of emotion, dependent on the field they subscribe to. Usually, these interpretations are used interchangeably with the ones of opinion, sentiment, mood and feeling, even though they are separated concepts. As summarized by Soleymani in [33], researchers in the past have been making clearer the distinction between these apparently overlapping terms. For example, Scherer differentiated between emotion and mood based on their duration: he defined the former as short-term phenomena, involving facial expressions, body reactions and feelings; the latter, instead, are long-term affect state that can last from hours to days. A similar discernment has been made by Munezero, which provides a discussion about the discrepancies between emotion and opinions: again, the main focus relies on the duration of the two, opinions being the long-term experience; however, a deeper analysis of the latter, defines them as judgments that do not need to be emotional triggered, being their polarity (positive, negative or neuter) expressed by the sentiment, which is in turn defined as a component of the opinion itself, together with the sentiment holder and the entity.

2.1.2 Representation of Emotions

Calvo in [2] identifies Darwin as the first to scientifically explore emotions as a result of the evolutionary process, spotting similar facial and body expressions in both humans and animals. However, that first emotion detection statement failed to explain the number of existent emotional behaviors. From then on, plenty of theories about the classification of emotions have been articulated. It is possible to distinguish them in two main strands of thought: categorical and dimensional representation of emotions.

The categorical representation puts emotions in a discrete space, treating each of them as a class. From a pragmatic point of view, this approach results in data having a binary nature (presence, non-presence) or presenting continuous

values indicating the intensity of the emotion itself. If, on one hand, this view is extremely useful since it allows to concretely assign observations (documents, expressions) to precise emotions, on the other hand, it is quite limited in terms of applications, because different tasks may require the identification of different emotions and, therefore, the same model could not be applied to two distinct problems. There are several classification models, but the most widely acknowledged is the one drafted by Paul Ekman [7], which identifies six basic emotions: anger, disgust, fear, happiness, sadness and surprise.

The dimensional representation, instead, places emotions in a continuous three-dimensional space, where the three axis are represented by the following aspects: valence (pleasant, unpleasant), arousal (energized, soporific) and dominance (controlling, controlled). Each dimension has an equal range of values which is usually identified as being between -1 and 1. This view allows to apply the same model to multiple tasks, even removing one of the dimensions (as often happens with dominance), and has been gaining much importance in this field during the last years. The model is referred to as VAD, while the first instance of this idea was the PAD model conceived by Mehrabian and Russell [21], with pleasure used in place of valence.

2.1.3 Music Emotion Recognition

Modern technologies like CDs, mp3 players and streaming services have rapidly made music accessible to everyone during the last decades. Songs have become omnipresent in our lives, being the soundtrack of our daily routines, but also evoking a wide range of emotions, letting us recall remarkable experiences. Therefore, it does not come as a surprise the arise of interest from researchers in building models able to classify music tracks according to the emotion they express. Yang and Chen [36] make a great effort in assessing the current state in the music emotion recognition field. They evaluate the possible approaches to tackle this problem, taking into consideration all the issues related to it, from the granularity of emotion representation to the subjectivity of its perception.

Depending on the emotion representation system employed, different machine learning approaches can be adopted in order to detect emotions in music. When dealing with the dimensional representation, regression models, like support vec-

tor regressors [12], are certainly the preferred choice in order to predict the values of valence and arousal. However, the most widely adopted system appears to be the categorical one, thus culminating in an abundance of studies based on classification models, from long-short term neural networks [14], to support vector classifiers [17] [18]. A more recent trend, instead, regards the exploitation of data mining techniques, like the Apriori algorithm, in order to evaluate the importance of features in predicting the target [20].

Another important distinction can be made according to the type of data used in the analysis: while the majority of studies focuses solely on the music component, lately new researches have been conducted also on lyrics and vocals [30]. While commonly text mining and sentiment analysis techniques can be easily applied to lyrics, the literature is not clear about the approach to use with vocals: in some studies they are processed using speech analysis techniques, while in others they are kept together with music, using the same information retrieval approach. A combination of these two inputs is used in [24], where features extracted from synchronized vocals and lyrics have been employed for detecting emotions.

2.1.4 Music Mood Dataset

The first main problem that is often encountered in music emotion recognition studies is the lack of available and reliable data. Even though nowadays music is widely accessible from everywhere, for both personal and commercial uses, it is still very difficult to find a properly labeled dataset about the emotions expressed in songs. One of the main issues, as we have seen before, can be found in the multitude of representation systems that have been built and, therefore, in the lack of consensus about a universal approach. This results in researches being conducted on different emotion systems which, in turn, require datasets assembled ad hoc. The solution which is mostly implemented regards the usage of services like Amazon Mechanical Turk and the final dataset being built on top of the feedback given by its workers. Unfortunately, this approach is often expensive, making the dataset not publicly published, and does not always satisfy the requirements of largeness and unbiasedness of data.

Çano and Morisio [4] try to address these problems with their Music Mood Dataset, based on tags from the last.fm website. First of all, they transformed

the dimensional representation of emotions given by the aspects of valence and arousal into a categorical one, dividing the Cartesian plane into four quadrants and assigning a completely distinct emotion to each of them. This approach leads to data being labeled according to four main categories: happy, relaxed, sad and angry. A large selection of songs were gathered from the Million Songs Dataset and then, in order to assign them to one of those labels, tags given by users on the last.fm website were collected and properly cleaned: terms non-related to the variable of interest were completely discarded, while others expressing moods or opinions were arranged in groups based on the similarity of their meaning. Finally, in order to avoid bias, only songs with a sufficient number of concordant tags were kept in the final dataset and assigned to the preponderant emotion expressed. The dataset was then used in [3], for music emotion recognition on songs lyrics.

In this study, a subset of 2000 songs has been used as available data for training and evaluating models, with each label being equally represented. In addition to the name of the songs and their emotion, lyrics and audio files have been used for feature extraction. An additional label, genre, was added to each song in order to assess the relationship between the emotion it transmits and the genre it belongs to.

2.2 Music Information Retrieval

Music information retrieval is an interdisciplinary field that deals with the extraction of useful features from songs, to then use them as input for machine learning models. The most common applications are source separation, recommendation systems and music classification. The former, in particular, has been deeply explored in this study during the preprocessing phase of the data, in order to be able to perform a proper separation between the background music and vocals of each song. Among the large variety of services devoted to this particular task, one that certainly stands out is Spleeter [13], a project conducted by the Deezer research department. Its pre-trained models are among the most performing ones according to the musdb18 benchmark and, to ensure availability and speed, have been made publicly accessible and usable in just a single command line. The different models differ by the number of sources they are able to identify: for

2-stem models, only vocals are isolated from music, while 4 and 5-stem models are capable of detecting the track of single instruments.

Music information retrieval cannot be merely confined to the processing of audio files, especially when dealing with emotions. Therefore, it is necessary to perform an in-depth study about the state-of-art, not only of music and vocals processing, but also about the analysis of text, given the strong sentimental role played by song lyrics.

2.2.1 Music Related Features

Müller in his book “Fundamentals of Music Processing” [23] makes an overview of the literature in this field, starting from the way in which music can be represented, to the identification and extraction of its features. These can be categorized according to different aspects [6], above all: level of abstraction, temporal scope, musical aspect and signal domain. The level of abstraction refers to the degree of comprehension a human has of that variable, going from high-level features (chords, rhythm, genre), that are understood and enjoyed by people, to low-level ones (spectral centroid, zero crossing rate), statistical features that can only be captured by machines, passing through mid-level features (MFCCs, pitch), that are usually referred to as an aggregation of low-level ones. While the level of abstraction only refers to the field of music, the other aspects are valid for audio in general. The temporal scope identifies the time interval above which features are computed: instantaneous refers to a range of time in the order of milliseconds, segment-level extends to seconds, while global describes the whole sound. The signal domain refers to the area of origin of the features, with time and frequency being the possible sources; a combination of the two, referred to as time-frequency representation, is obtained by applying on the time domain waveform the Short-Time Fourier Transform (STFT).

There are several tools available today that simplify the process of feature extraction from audio. In particular, one of the first and most powerful toolbox was developed for MATLAB in 2008 [16], allowing to extract variables related to timbre, tonality, rhythm and form. Similar characteristics are present in python libraries like librosa, torchaudio and pyaudioanalysis. The latter, in particular, has been used for the purpose of this project, because of the availability of

functions for computing mid-term music features directly from short-term ones, choosing the desired audio interval.

2.2.2 Vocals Related Features

The main concern with these type of features is about the methodologies that should be employed for their extraction. In the literature, there are two main approaches adopted to accomplish this task: if on one side, researchers treat song vocals as they were music, on the other speech analysis techniques are used. Since both approaches deal with the inspection of audio frequencies, sometimes the extracted features may overlap, while in other cases they slightly differ. Deepali and Subbaraman [19] make a in-depth overview of the possible features that can be extracted from singing voice and the various applications for which they are better suited. According to the task that has to be performed, some classes of features may be usually preferred above the others. For classification problems related to the prediction of mood, genre or instruments, rhythm and timbre features are the most adopted ones, while pitch and harmonic suit better song similarity tasks.

One of the most adopted framework for feature extraction in the field of speech analysis is the Geneva Minimalistic Acoustic Parameter Set (GeMAPS) [8]. It allows to retrieve low-level descriptors related to different parameter groups, like frequency, energy and spectrum, resulting in a total of 62 features. An extended version, referred to as eGeMAPS, has been developed and it provides 26 more parameters, adding up to 88 in total. This variant, which is also the one adopted for this research, includes cepstral parameters and more dynamic ones. The tool was evaluated on different emotion datasets, from speech to singing voice, and it was able to reach the performances of other famous frameworks of this field, like ComParE and InterSp12, despite having a much lower number of parameters as input.

2.2.3 Lyrics Related Features

Which features to extract from texts and how to process them have always been important problems in the field of text mining. Different approaches can be adopted, depending on the source of the data, the length of the text and the

purpose of the analysis. The most common methods for text analysis are keyword or lexicon-based, while others rely on machine learning techniques. The former make use of a dictionary of words, each of them labeled according to its polarity, mood, part of speech; the data is then tokenized and each token is then matched to the terms in the vocabulary in order to classify observations. For example, the ANEW sentiment dataset is used in [15], where song lyrics are used together with audio features to predict emotions. Instead, for machine learning based approaches, the raw text data must be transformed into vectors in order to be utilized as input for the models; this results in adopting solutions like bag-of-words, tf-idf or word embeddings.

It has been demonstrated that word embeddings are usually able to capture better the semantic relationship among words and thus allowing to reach better performance levels in text classification. However, when dealing with emotion detection, we are more interested in the sentiment expressed by the words rather than in their intrinsic meaning; this translates in classical word embeddings evaluating ‘good’ and ‘bad’ as terms close to each other, while we would rather prefer putting them aside, given the opposite polarity they express. Therefore, a need for a word embedding model able to capture the emotion of words arose and the task was accomplished in [34] with the Sentiment Specific Word Embedding (SSWE). Three different models were built: the first basic model $SSWE_h$, a second basic model $SSWE_r$ with more relaxed constraints, and a final model $SSWE_u$, called unified, which tries also to capture the semantics of the words. In order to further improve the ability to capture the meaning of the terms, in [11] researchers combine a SSWE with another word embedding built using GloVe. They demonstrate the ability of the SSWE model to capture the discordance in polarity of words like ‘happy’ and ‘sad’, while GloVe interpret them as similar in meaning; ‘best’ and ‘great’ are instead identified as very close by the word embedding built on GloVe, as opposed to the SSWE. While GloVe is able to perform greatly on single terms or short ngrams, it does not ensure the same level of performances when dealing with longer texts. For this reason, it is suggested to employ models able to correctly process longer sentences, like sentence BERT built over siamese networks [29].

2.3 Multimodal Systems

When facing complex phenomenon, it is unusual that all the relevant information can be obtained using only one modality. For these reason, it has become a common practice to combine measurements from different sources in a single analysis; this process is referred to as data fusion or integration [32]. The techniques used to merge these data can be classified into different categories, according to the following criteria: relation between input data sources, input/output data type and nature, abstraction level of data, data fusion level and architecture type [5]. For the purpose of this research, we will focus on the aspect of abstraction level of data; in particular we identify three main degrees of fusion: measurements, characteristics and decisions. In the former, also referred to as low-level fusion, raw data is directly used as input for the analysis; in the characteristics level (mid-level fusion) features are combined, usually with linear functions, in order to obtain new variables to employ for solving the task; finally, in the decisions level, also called high-level fusion, the symbolic representation of features is combined to obtain a more accurate decision.

In the field of emotion detection, multimodal approaches are only a recent trend. This new tendency has arisen thanks to the massive availability of multimedia data on the Internet, coming from social networks in particular. It is now easy to retrieve a large number of video, thus being able to analyze facial expressions, body movements, voice and text from a multitude of different sources. Music is not an exception, allowing researchers to focus their studies on different inputs (music, voice, text) and combining them at different levels.

2.3.1 Low-Level Fusion Approaches

Low-level fusion refers to that set of multimodal approaches that combine directly raw data and utilize them as input for the models; the most common approach involves a simple concatenation of features from different sources, creating an individual stream of data. Different researches have demonstrated the benefits of adopting this kind of approach over simple models built on single inputs. The majority of the studies have been conducted on YouTube videos, extracting from them audio and visual components and then retrieving their text transcription

[22] [27] [25] [31] [1]. Classifiers are then built using each source of data individually as input, adopting classical machine learning algorithms, like support vector machines or hidden Markov models. Subsequently, their performances are compared with the one of multimodal models, built combining sources alternately in pairs or directly merging the whole set of data. In all of the previously cited papers, there has always been a significant improvement in prediction results using data fused at a low level.

2.3.2 High-Level Fusion Approaches

With high-level fusion we refer to that approach to data integration which is applied the latest in the model training process. In fact, if with low-level fusion the data is concatenated at the beginning before even starting the analysis, here instead the merging operation happens at a decision level, with the predictions being the object of the fusion rather than the raw data. The main idea behind this approach relies on the unification of the predicted probability for each class coming from a multitude of models. While in a multimodal context each model is built on a different source of data, this process can also be employed for combining results of several algorithms trained over the same dataset. Probabilities of each class are merged applying functions over them, like sum, multiplication or average; the final decision will then be the label associated with the highest probability after the computations. In some cases, one can also decide to assign a weight to each model, training in turn another algorithm in order to assign the properly estimated parameters.

Some papers, like [26] and [35], directly adopt this approach to multimodal analysis, while others, like the already cited [27] [31] [1], make also a comparison between the results of low-level and high-level fusion models. To date, the conclusions of these kind of tests are not unanimous and, therefore, it is impossible to define which is the best approach to employ. An interesting research made in the field of music emotion recognition is [28], where the audio and the lyrics of the songs are used as input; valence and arousal are the target variables and for each algorithm we have a different combination of source-target, resulting in a total of 7 models trained. Each model predicts labels using a rule-based method, mapping the estimated value of valence and arousal to Thayer's emotions scheme.

Chapter 3

Methodologies

Before going deep into the methods used to preprocess the data and evaluate the models, it is very important to understand the nature of the dataset and how I obtained it. The starting point, as previously observed, is the Music Mood Dataset, from Çanio and Morisio [4], consisting in a list of two thousand songs and the associated emotion label extracted from the last.fm tags. The other target label of the analysis, ‘genre’, was retrieved from the famous music streaming platform, Spotify: due to the excessive depth of sub-genres, I grouped all the collected labels into a final set of 15 ones, using an already existing mapping to major genres made by chosic.com.

As already explained, the aim of the project is to predict these two labels using different data sources, both individually and combined (Figure 3.1).

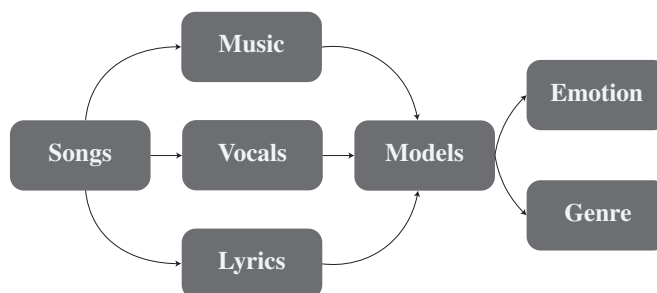


Figure 3.1: A diagram illustrating the workflow of the entire project.

Text data is represented by the lyrics of the songs that were retrieved from Genius.com, one of the most popular websites in the music sphere. The audio files, instead, were separated into two main tracks, identifying the background music

and the singing voice of the artist. This result was achieved by employing the 2-stem pre-trained model of Spleeter, a famous python library made by the research department of Deezer and capable of performing musical source separation tasks.

The following paragraphs will better explain all the approaches followed during this research: the process of feature extraction and the one of model training. These two phases need a separate in-depth inspection given the amount and the extent of the performed experiments.

3.1 Feature Extraction

For each of the three sources of information, we can identify two crucial steps needed to make the data ready to be used by the models. Firstly, I defined different ranges to be considered in order to section the initial files: both text and audio data were considered wholly and partially, for example extracting the chorus or excerpts of a specific length. Secondly, different feature extraction techniques were adopted to transform the initial raw data into vectors suitable to be used as input for the machine learning models.

3.1.1 Audio Feature Extraction

The most common approach in literature when dealing with music or, more in general, with longer audio files is to trim them down to a shorter excerpt, usually thirty seconds long, cut at second 30 up to second 60. This certainly allows to solve the problem of input files having different lengths and also helps in reducing the amount of computational time needed to process the data and extract the necessary features. Therefore, in this analysis I considered, along with the full-sized input, also two shorter versions of it: a thirty seconds long excerpt, retrieved adopting the approach previously explained, and a ten seconds long excerpt, derived by identifying the chorus of the songs, employing an algorithm capable of detecting repeated patterns in audio frequencies. At this point I applied the 2-stem Spleeter model for musical source separation, in order to obtain three versions of data, different in length, for both music and vocals (Figure 3.2).

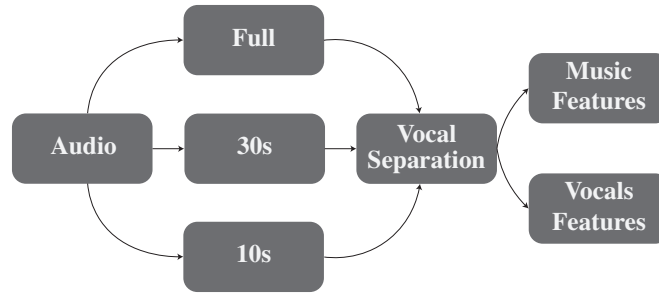


Figure 3.2: The approach to audio processing.

3.1.1.1 Vocals Feature Extraction

Starting from the vocal aspect, I employed the GeMAPS framework to retrieve voice features; in particular, I used the extended version called eGeMAPS, given the presence of cepstral description which, from literature, are consistently known to increase the accuracy of automatic affect recognition. The first and most important parameter extracted is the fundamental frequency F_0 , which refers to the approximate frequency of the periodic structure of voiced speech signals; it can be defined as the average number of oscillations per second and expressed in Hertz. The definition of fundamental frequency is often associated with the one of pitch, being the former a low-level feature while the latter a mid-level one. Another important feature is loudness, referred to as the estimate of perceived signal intensity from an auditory spectrum. These two, together with rest of the low level descriptors (18 in total) are smoothed over time with a symmetric moving average filter 3 frames long; in particular, while arithmetic mean and coefficient of variation are computed as functionals over all the descriptors, for fundamental frequency and loudness other 8 functionals were applied, like percentiles at different levels, range, and the mean and the standard deviation of the slope of signal parts. With the extended version of the framework, it is also possible to retrieve the first four mel frequency cepstral coefficients (MFCCs), values able to describe the overall shape of the spectral envelope and adopted in MIR to describe timbre. Finally, the harmonic aspect of voice is captured by formants which are the broad spectral maximum that results from an acoustic resonance of the human vocal tract; the formant with the lowest frequency is called F1, the second F2 and the third F3.

3.1.1.2 Music Feature Extraction

Features for the music component were extracted using the pyAudioAnalysis library, which allows to retrieve mid-term information starting from short-term ones; this is possible choosing the right interval step size and computing the mean and the standard deviation of each descriptor along the whole song or excerpt. An additional set of feature, identified by the prefix ‘delta’, were also computed: they represent the difference between values of subsequent time intervals and they are computed for each of the already retrieved variables. The most important gathered descriptors are:

- the zero-crossing rate which indicates the number of times a signal crosses the horizontal axis;
- the energy, corresponding to the total magnitude of the signal;
- the spectral centroid, indicating the frequency at which the energy of the spectrum is centered upon;
- the spectral spread which describes the average deviation of the rate-map around its centroid;
- the spectral entropy, measuring the complexity or spectral power distribution of a signal;
- the spectral flux, which measures the spectral change between two successive frames;
- the spectral rolloff, representing the frequency below which a specified percentage of the total spectral energy lies;
- 13 MFCCs, whose purpose has already been described above;
- the 12 chroma features, representing the tonal content of a signal in a condensed form and, for this reason, usually associated to the twelve pitch classes.

3.1.2 Lyrics Feature Extraction

Similarly to what has been done with audio files, textual data were processed not only in its entirety, but also using parts of it; in particular case, I ended up with 4 different versions which will be subject to the feature extraction process (Figure 3.3). Before explaining which are and how I obtained each of the four lyrics variants, I have to specify that a significant portion of the songs were not in English and thus a translation was needed; this was achieved, where possible, employing the Opus-MT model, an open neural machine translation algorithm. Where the task was not accomplished, due to the lack of support of some languages, Google Translator was used in its place. Another important remark concerns the presence of metadata inside the lyrics indicating if a specific set of sentences represented a verse or the chorus of the song; unfortunately, this information was not available for all the texts and, as we will see, alternative methods were adopted to identify the refrain.

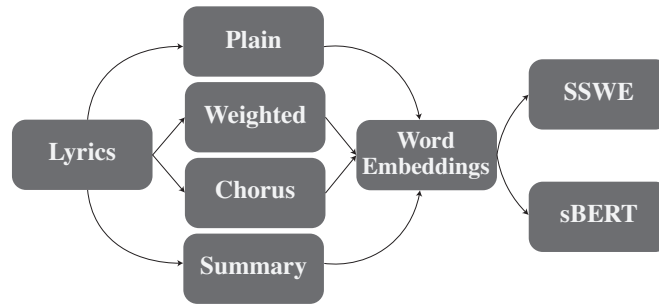


Figure 3.3: The approach to text preprocessing.

3.1.2.1 Lyrics Format Variations

The first set of data was represented by the plain lyrics, without any modification: in this case each song is represented by its whole text, deprived only of metadata in case they were present. Metadata were instead crucial in collecting choruses, which constitute the second version of the textual data used in this analysis. As I already explained, this retrieval was not possible for each song and, therefore, an extractive text summarization technique was employed: for each lyrics, a similarity matrix of sentences was computed using the cosine similarity and a page rank algorithm retrieved the one that were most representative of the song. At this

point I was able to identify refrains even when they were not explicitly declared. Combining the set of plain lyrics with the one of choruses, I was able to generate a third variant of textual data, referred to as weighted lyrics: for each song, I retrieved again the whole text, with the exception that this time the sentences were given an higher weight if belonging to the chorus and a lower weight otherwise. Finally, I employed an abstractive text summarization technique using the T5 model in order to generate summaries of the lyrics. This last one represented the fourth and final variation of raw textual datasets and concludes the discussion of their generation.

3.1.2.2 Word Embeddings

In order to make this type of data readable by machines and used as input for machine learning algorithms, it is necessary to extract features from it in a vectorial form. The most common approach concerns the adoption of word embeddings, which are capable of retrieving the semantics similarities and dissimilarities between words. Given the variability in lyrics lengths, I preferred to focus on entire sentences rather than single terms; for this reason, I employed the MiniLM-L6-v2 model from the sBERT family, which is intended to be used over a sentence or a short paragraph (like a chorus) and outputs a vector ready to be employed for clustering or sentence similarity tasks. While sBERT models focus more on retrieve the semantics of the sentence, I thought it was also necessary to capture the sentiment of the data, given the fact that the project is about emotion detection. For this reason, I also generated another word embedding adopting the implementation of the sentiment specific word embedding (SSWE) from the Microsoft's machine learning library, nimbusml. In the end, I concatenated the two set of embeddings, generating a new version of the set of textual features, which will hopefully grasp both the semantic and the sentiment aspect of the data. Hence, I ended up with four variations of raw data and, for each of them, I built the three different word embeddings, having a vector size of 150, 384 and 534 respectively, summing up to a total of 12 different combination of textual feature sets.

3.2 Model Training

This phase of the research allows us to train the models needed to gather all the information needed to evaluate the different experiment I decided to run, from the different forms of data input (whole and partial data) to the impact of combining different data sources together. The trained models belong to the family of classification algorithms, whose objective is the one of correctly associate emotions and genres to songs. The analysis will start with single-input models, that make use of one data source at time, and will continue with multimodal models, which instead combine all three of them alternatively; finally, we will take a look at the relationship between emotions and genres, studying their correlation and trying to improve the classification of the latter given the former (Figure 3.4).

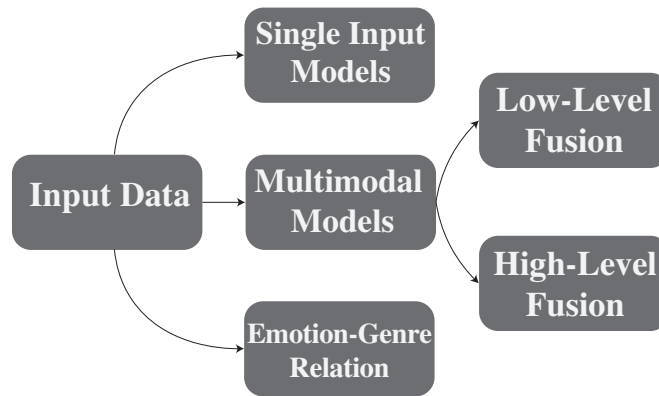


Figure 3.4: The model training phase.

3.2.1 Single-Input Models

The algorithms chosen for the analysis are among the most common ones when it comes to classification tasks: k-nearest neighbor, support vector machines, random forest, logistic regression and multilayer perceptron. Each of them was trained over the training part of the datasets, constituting 75% of the total data, using a 5-fold cross validation and tuning their parameters using a grid search. For every type of model, the best estimator in terms of F1 score was returned and tested over the remaining data, computing all the appropriate metrics: accuracy, precision, recall and, again, the F1 score. Also in this case, the latter was used

to determine the best classifier for each combination of data source and form of input.

3.2.2 Multimodal Models

Data inputs were combined at two levels: high and low. In the former case, as already seen, the fusion happens at the decision level, while in the latter it happens directly on measurements. Furthermore, sources were merged not only alternatively in pairs, but also all three together, resulting in a total of four possible combinations.

Starting from high-level fusion models, a soft voting approach was employed: each source was represented by the best single-input estimator trained in the previous section. Then, for each possible combination of inputs, the best classifiers were applied to predict the probabilities of belonging to a certain target class for each song in the test set. The label that was finally assigned to the observation was represented by the one reaching the highest probability after summing all the probabilities predicted by each model. The metrics already listed before, were computed also in this case in order to compare the performances of the different input combinations in predicting both emotions and genres of songs.

For low-level fusion models, features were simply concatenated according to the four possible combinations, forming new and wider sets that were provided as input for the already listed above classification algorithms. The process at this point follows the method already used, consisting of training the classifier over the test data, applying cross validation and grid search, and retrieving the best model according to the results on the test set.

3.2.3 Emotion-Genre Relation

The final experiment I wanted to conduct regards a possible connection between the emotion expressed in a song and the genre it belongs to. The first method that can be employed to accomplish this task consists in performing a correlation analysis: since we are dealing with two categorical variables, a contingency table is built and then the χ^2 test is run. The second approach resides instead in using the emotion conveyed by a song as a known fact and including this prior knowledge inside the model for predicting the genre and evaluating its performances compared with the ones of the model without this information.

Chapter 4

Feature Extraction

In this chapter, we will go deeper into the data preprocessing phase, understanding how the previously discussed methods were applied and which results have been obtained. Following the same scheme as before, the first step will be to explore the different stages that were gone through in order to obtain the final version of the dataset, and then discuss separately the application of those techniques to the audio files and the textual ones.

4.1 Data Retrieval

The Music Mood Dataset is a collection of 2000 songs, identified by the artist name and its title, and the associated label ‘mood’, retrieved from the analysis of the last.fm tags given by the users and representing the emotion transmitted (Table 4.1).

Table 4.1: Some observations from the Music Mood Dataset.

Artist	Title	Mood
A Day to Remember	If It Means a Lot to You	sad
Prince	Beautiful, Loved and Blessed	relaxed
Deine Lakaïen	The Game	sad
Fatboy Slim	Talking Bout My Baby	happy
Cooler Kids	All Around The World	happy

The classes chosen for the target variable are four, resembling the Thayer’s emo-

tion theory, a 2-dimensional system having as axis energy and stress, which can be identified as substitutes of valence and arousal; the intersection of these two characteristics generates four distinct quadrants, each representing a person's mood: happy, sad, relaxed and angry (Figure 4.1). The four emotions are equally represented in the dataset, with 500 observations belonging to each of the classes.

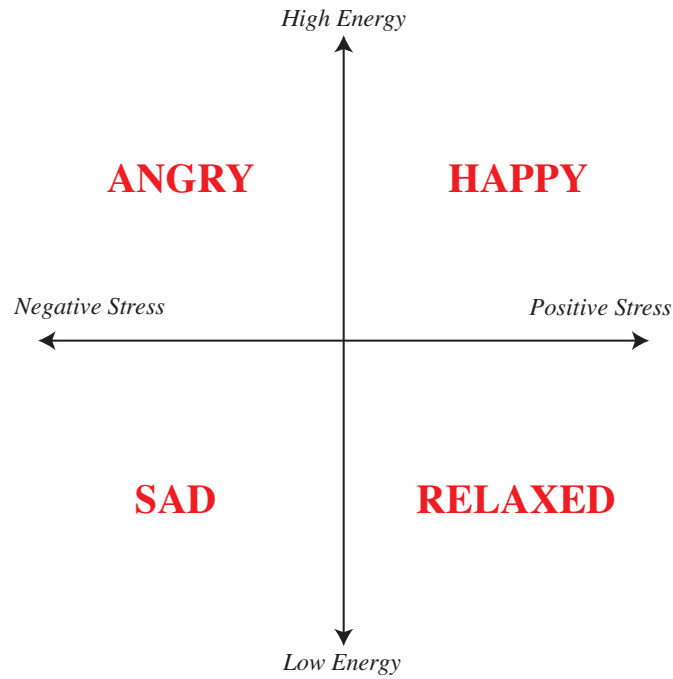


Figure 4.1: Thayer's emotions scheme.

4.1.1 Lyrics Retrieval

Even though the initial dataset does only contain a few details about each song, this knowledge is still enough in order to retrieve all the other necessary data. In particular, the two main source of information for the machine learning models will be audio and text, represented by the track and the lyrics. Starting from the latter, the words of each song were gathered from genius.com, a popular website in the musical field which, above all, offers an API that can be used to retrieve information about songs, artists, albums and so forth; given the combo artist plus title, it is possible to access the lyrics of the song concerned. One option that can

be very useful, as later explained, is the possibility to also collect section headers together with the text; this means that for each verse, there will be an associated meta tag indicating which part of the songs it represents (intro, chorus, bridge, etc):

[Verse 1]
 I'm not afraid of anything
 I just need to know that I can breathe
 I don't need much of anything
 But suddenly, suddenly
 I am small and the world is big
 All around me is fast moving
 Surrounded by so many things
 Suddenly, suddenly

[Chorus]
 How does it feel to be
 Different from me?
 Are we the same?
 How does it feel to be
 Different from me?
 Are we the same?
 How does it feel?

Having the lyrics being retrieved, it emerged that 5 songs were lacking of words, therefore being represented only by the instrumental component (Table 4.2).

Table 4.2: Songs without lyrics.

Artist	Title	Mood	Lyrics
Eric Clapton	Reptile	relaxed	instrumental
Yanni	Nostalgia	relaxed	instrumental
Nightmares On Wax	Flip Ya Lid	relaxed	instrumental
Kool & The Gang	Summer Madness	relaxed	instrumental
Miles Davis	Summertime	relaxed	instrumental


4.1.2 Audio Retrieval

The other important element to consider is the audio component of each song. There are two possible approaches in these kind of cases: finding datasets with already extracted features or gather the files and process them manually. The former method, while being faster, has multiple limitations, from the risk of not finding the extracted values for a particular song, to the impossibility of performing the analysis using two distinct streams of data, music and vocals. Therefore, audio files for the whole dataset were collected in mp3 format which was preferred to higher quality ones like flac for reasons related to their availability, but also to the disk space and processing time required.

4.1.3 Genre Retrieval

Using the Spotify API, it was possible to collect tags for each song and attach them to the corresponding audio file previously retrieved (Table 4.3).

Table 4.3: Tags retrieved for each song with the Spotify API.

	Artist:	Avril Lavigne
	Title:	How Does It Feel
	Album:	Under My Skin
	Genre:	Canadian Pop
	Duration:	03:45
	Sample Rate:	48 kHz

Among those tags, there is one of great importance for the purpose of this project, which is genre, the second target variable of interest. However, the number of genre identified by Spotify in its dataset is very high, more than five thousands labels; sticking with this approach, would have meant considering too many poorly represented categories and, thus, models unable to make correct predictions. For this reason, there was the need to assign all the sub-genres to a fixed set of main ones; this mapping has been already performed by the team behind chosic.com which, apart from being a tool for playlist generation and analysis, it has also a collection of all the sub-genres available on Spotify and the corresponding parent genres. After codifying the original labels into their parent classes, the final set

of genres is constituted by 15 different categories.

Unlike emotions, in this case there is not a uniform distribution of observations over each class, with some genres like rock, metal and pop being largely represented by hundreds of observations, while others having only a few dozens (Table 4.4).

Table 4.4: Parent genres and their number of occurrences in the dataset.

Genre	Occurrences
rock	651
metal	383
pop	255
folk/acoustic	126
hip hop	101
r&b	87
country	77
easy listening	77
dance/electronic	73
new age	50
world/traditional	43
latin	24
blues	23
jazz	17
classical	13

This could have also been expected looking at the number of occurrences of each artist in the dataset; the total number of distinct singers, indeed, sums up to 1291, with some famous metal and rock band from the 80s appearing several times (Table 4.5). This imbalance was addressed anyway before training the models, when the split between train and test set for the genre classifiers was made by stratifying over the target variable.

4.2 Audio Preprocessing

The source of audio data, as previously explained, are the mp3 files, which will be later processed in order to extract useful features. Before getting to that point, it is important to specify that the songs will be considered both entirely and

Table 4.5: Artists appearing the most in the dataset.

Artist	Occurrences
Slayer	26
Megadeth	26
Rage Against the Machine	22
Nine Inch Nails	12
Enya	10

partially: a thirty second excerpt was easily extracted using the AudioSegment tool, trimming from second thirty up to second sixty, as usually happens in literature. In the remaining part of this section, we will take a deeper look at the tools used to generate the final format of the files and how they were separated; finally, the complete set of variables for vocals and music will be shown.

4.2.1 Chorus Detection

The third and final format of audio files, employed in the process of feature extraction, is a 10 second excerpt, following the idea of testing if, with less noise data, model performances would improve or not. This shorter portion of the song was retrieved adopting an algorithm for chorus detection, which approximately follows the paper by Goto [10]. The method is implemented in the library *pychorus* and it is based on the idea of detecting notes using chroma parameters and finding similarities among sections of a certain length (Figure 4.2). Trying to output excerpts of 30 seconds, however, would sometimes fail, due to the inability of finding a recurrent pattern in certain songs by the algorithm; therefore, I decided to lower the time interval to 10 seconds. Below, they are displayed the waveforms of a song in the three different audio length used (Figure 4.3). Unfortunately, even with a shorter timeframe, it happened on rare occasions that the algorithm was still unable to detect a chorus. For the few songs where this occurred, I used the values of the features extracted from the 30 seconds excerpts also for the 10 seconds ones.

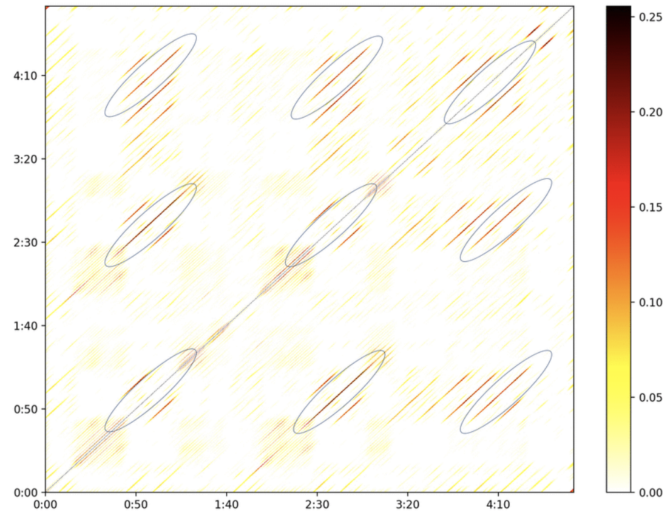


Figure 4.2: Example of a time-lag similarity matrix used to identify repeated sections.

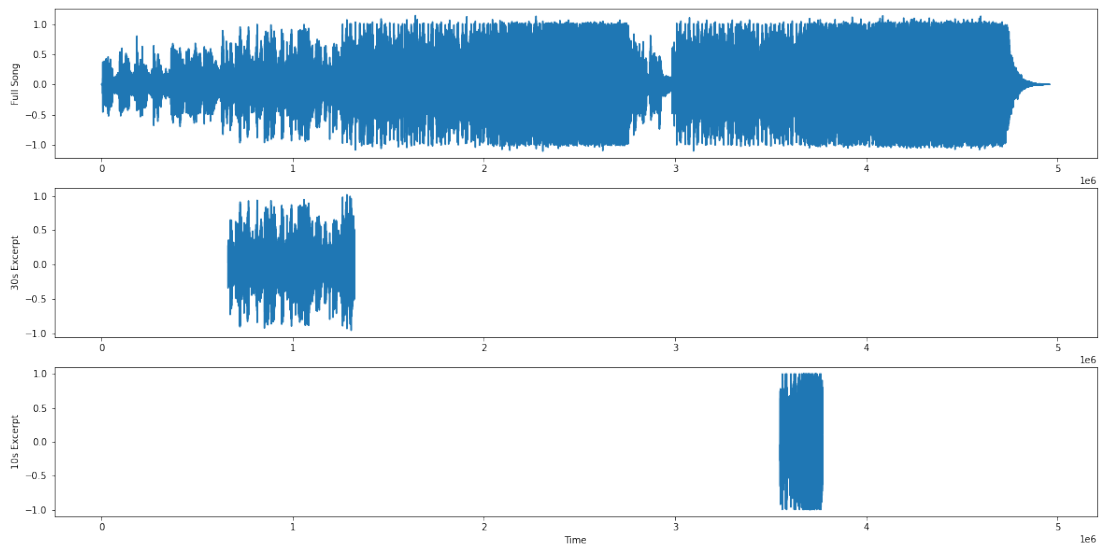


Figure 4.3: The three waveforms of a song used for information retrieval: complete, 30 seconds and 10 seconds excerpts.

4.2.2 Vocal Separation

Different AI-based tools for vocal separation available online, both payed or free, were tried before deciding to stick with Spleeter: it is faster, offers both an easy command line and python implementation, returns better results. In particular, its 2-stem model was used in order to generate the two necessary files, one containing the background music and the other the singing voice. From a rapid listening, the results can be considered as satisfactory, the two track being properly separated, except for some negligible mistakes. Furthermore, the algorithm seems to work better on older songs, like the one considered in this project; indeed, from some tests conducted on more recent ones, certain type of electronic sound were erroneously considered vocals, returning a noisy output.

4.2.3 Vocals Feature Extraction

Features regarding singing voice were extracting using the library openSMILE; among the available feature sets, eGeMAPS in its last version v02 was chosen, being an extended, thus more complete, version of the basic GeMAPS, but still faster with respect to ComParE_2016, which may have slightly better performances, but very slower computational times given the number of available features, 6373 against 88.

For each song, all three format were considered in order to be processed: the full audio, the 30 seconds excerpt and the 10 seconds one. The files were read using the audiofile library, for a faster import and conversion to the wav format, returning an array representing the signal channels and its sampling rate. These two components are then used as input for the function that process the signal and returns the predefined set of features, in this case the 88 from eGeMAPSv02. The full list of variables is available below (Table 4.6).

4.2.4 Music Feature Extraction

The library pyAudioAnalysis, employed for the extraction of music features, allows to both retrieve short-term and mid-term features, the former representing values computed over small intervals, in the order of milliseconds, and constituting the basis for the calculation of the latter, which are simple statistics, mean

Table 4.6: Full set of features extracted from vocals.

F0semitoneFrom27.5Hz_sma3nz_amean	F0semitoneFrom27.5Hz_sma3nz_stddevNorm
F0semitoneFrom27.5Hz_sma3nz_percentile20.0	F0semitoneFrom27.5Hz_sma3nz_percentile50.0
F0semitoneFrom27.5Hz_sma3nz_percentile80.0	F0semitoneFrom27.5Hz_sma3nz_pctlrange0-2
F0semitoneFrom27.5Hz_sma3nz_meanRisingSlope	F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope
F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope	F0semitoneFrom27.5Hz_sma3nz_stddevFallingSlope
loudness_sma3_amean	loudness_sma3_stddevNorm
loudness_sma3_percentile20.0	loudness_sma3_percentile50.0
loudness_sma3_percentile80.0	loudness_sma3_pctlrange0-2
loudness_sma3_meanRisingSlope	loudness_sma3_stddevRisingSlope
loudness_sma3_meanFallingSlope	loudness_sma3_stddevFallingSlope
spectralFlux_sma3_amean	spectralFlux_sma3_stddevNorm
mfcc1_sma3_amean	mfcc1_sma3_stddevNorm
mfcc2_sma3_amean	mfcc2_sma3_stddevNorm
mfcc3_sma3_amean	mfcc3_sma3_stddevNorm
mfcc4_sma3_amean	mfcc4_sma3_stddevNorm
jitterLocal_sma3nz_amean	jitterLocal_sma3nz_stddevNorm
shimmerLocaldB_sma3nz_amean	shimmerLocaldB_sma3nz_stddevNorm
HNRdBACF_sma3nz_amean	HNRdBACF_sma3nz_stddevNorm
logRelF0-H1-H2_sma3nz_amean	logRelF0-H1-H2_sma3nz_stddevNorm
logRelF0-H1-A3_sma3nz_amean	logRelF0-H1-A3_sma3nz_stddevNorm
F1frequency_sma3nz_amean	F1frequency_sma3nz_stddevNorm
F1bandwidth_sma3nz_amean	F1bandwidth_sma3nz_stddevNorm
F1amplitudeLogRelF0_sma3nz_amean	F1amplitudeLogRelF0_sma3nz_stddevNorm
F2frequency_sma3nz_amean	F2frequency_sma3nz_stddevNorm
F2bandwidth_sma3nz_amean	F2bandwidth_sma3nz_stddevNorm
F2amplitudeLogRelF0_sma3nz_amean	F2amplitudeLogRelF0_sma3nz_stddevNorm
F3frequency_sma3nz_amean	F3frequency_sma3nz_stddevNorm
F3bandwidth_sma3nz_amean	F3bandwidth_sma3nz_stddevNorm
F3amplitudeLogRelF0_sma3nz_amean	F3amplitudeLogRelF0_sma3nz_stddevNorm
alphaRatioV_sma3nz_amean	alphaRatioV_sma3nz_stddevNorm
hammarbergIndexV_sma3nz_amean	hammarbergIndexV_sma3nz_stddevNorm
slopeV0-500_sma3nz_amean	slopeV0-500_sma3nz_stddevNorm
slopeV500-1500_sma3nz_amean	slopeV500-1500_sma3nz_stddevNorm
spectralFluxV_sma3nz_amean	spectralFluxV_sma3nz_stddevNorm
mfcc1V_sma3nz_amean	mfcc1V_sma3nz_stddevNorm
mfcc2V_sma3nz_amean	mfcc2V_sma3nz_stddevNorm
mfcc3V_sma3nz_amean	mfcc3V_sma3nz_stddevNorm
mfcc4V_sma3nz_amean	mfcc4V_sma3nz_stddevNorm
alphaRatioUV_sma3nz_amean	hammarbergIndexUV_sma3nz_amean
slopeUV0-500_sma3nz_amean	slopeUV500-1500_sma3nz_amean
spectralFluxUV_sma3nz_amean	loudnessPeaksPerSec
VoicedSegmentsPerSec	MeanVoicedSegmentLengthSec
StddevVoicedSegmentLengthSec	MeanUnvoicedSegmentLength
StddevUnvoicedSegmentLength	equivalentSoundLevel_dBp

and standard deviation. The number of distinct variables is 34 and their role has been already discussed in the previous section; considering also delta values, the count rises to 68 and, in the mid-term case, it reaches 136 since we are considering the two mentioned statistics computed over values rather than features individually. The list of the base 34 features is displayed below (Table 4.7).

Table 4.7: Full set of features extracted from vocals.

zcr	energy	energy_entropy
spectral_centroid	spectral_spread	spectral_entropy
spectral_flux	spectral_rolloff	mfcc_1
mfcc_2	mfcc_3	mfcc_4
mfcc_5	mfcc_6	mfcc_7
mfcc_8	mfcc_9	mfcc_10
mfcc_11	mfcc_12	mfcc_13
chroma_1	chroma_2	chroma_3
chroma_4	chroma_5	chroma_6
chroma_7	chroma_8	chroma_9
chroma_10	chroma_11	chroma_12
	chroma_std	

In order to gather single values representing an entire song and, therefore, avoid the problem of having signals of different length, I used as window and step parameters of the function for retrieving features the length of the signal, computed by multiplying the duration of the song by its sampling rate. As in the case for vocals, features were retrieved for all the format of input previously generated.

4.3 Text Preprocessing

Before proceeding with the standardization and normalization of text, it was necessary to generate the fourth different formats of text used in this analysis. Firstly, I retrieved the chorus of each song using the appropriate meta tag, if present. In this phase, I split the texts in sentences and detected the language of the song, since many of them were different from English. In order to translate them, I used a neural machine translator, Opus-MT, available within the EasyNMT library; unfortunately, not all the needed languages were available,

thus Google Translate was employed for the remaining lyrics. Finally, all the exceeding meta tags were removed from the texts, making possible to generate the first dataset composed by complete plain lyrics.

4.3.1 Extractive Text Summarization

As already pointed out, some lyrics were lacking of meta tags, making impossible to retrieve its chorus this way. Therefore, I decided to employ an algorithm of extractive summarization capable of detecting a possible chorus, based on sentence similarity. To each possible pair of sentences, the bag of words model is applied and the cosine distance is computed between the two resulting vectors; repeating this process for the whole text leads to a similarity matrix, used to generate a similarity graph. The graph is used as input for a page rank algorithm, whose scores returned are used in the final decision process: the four sentences with the highest score will compose the summary, representing the chorus of that song.

This method allows to complete the previously created dataset of choruses, filling the space left by the missing ones. This collection of refrains is also used as starting point for the generation of another set, called weighted lyrics. Here, each sentence of the complete text is given a weight, based on the condition of belonging to the chorus: if the sentence is part of it, it receives a weight of 5, otherwise of 1; this method is used in order to give more emphasis on the most important part of the song, while still maintaining the entire lyrics.

4.3.2 Abstractive Text Summarization

The last dataset generated is composed by summaries of the lyrics. The idea resembles the one, previously explored and applied to audio for chorus detection. I employ an abstractive summarization technique, exploiting on the T5 model, where the entire text is used as input for the model, which returns as output a summary of it. This process should be even more effective than the extractive approach seen before since, in that case, it simply reutilizes sentences from the lyrics, while now the model tries to completely reprocess the text and generate a self-made summary, with a length between 50 and 200 characters. In Table 4.8 it is shown a song in all the four formats used in this research.

Table 4.8: The four formats considered for each song.

Plain	Weighted	Chorus	Summary
I'm not afraid of anything	I'm not afraid of anything, 1		
I just need to know that I can breathe	I just need to know that I can breathe, 1		
I don't need much of anything	I don't need much of anything, 1		
But suddenly, suddenly	But suddenly suddenly, 1		
I am small and the world is big	I am small and the world is big, 1		I am small and the world is big
All around me is fast moving	All around me is fast moving, 1		All around me is fast moving
Surrounded by so many things	Surrounded by so many things, 1		Surrounded by so many things
Suddenly, suddenly	Suddenly suddenly, 1		Suddenly, suddenly
How does it feel to be	How does it feel to be, 5	How does it feel to be	I am young and I am free
Different from me?	Different from me, 5	Different from me?	but I get tired and I get weak
Are we the same?	Are we the same, 5	Are we the same?	I get lost and I can't sleep
How does it feel to be	How does it feel to be, 5	How does it feel?	If you could comfort me
Different from me?	Different from me, 5		Would you cry with me?
Are we the same?	Are we the same, 5		
How does it feel?	How does it feel, 5		
(...)	(...)		

4.3.3 Word Embeddings

A crucial step in every text mining project lies in all the techniques adopted in order to standardize and normalize words to make them easier to understand for the machine. Each text considered in whichever of its formats was split into sentences, based on new line and punctuation characters, and went through the process of contraction fixing and lemmatization, applied used the spacy tokenizer. In order to make these sentences interpretable by a machine learning algorithm, it is needed to transform them in vectors capturing their semantic and sentiment similarities. For datasets containing longer texts, plain and weighted lyrics, embeddings were applied for each sentence and then averaged in order to obtain a single vector representing the entire document; choruses and summarized lyrics were instead considered as a whole given their shorter lengths.

The two word embeddings employed are the sentiment specific word embedding and the sentence BERT. The former is implemented via the Microsoft machine learning library nimbusml; it tries to detect the sentiment aspect of each sentence, encoding them in 150-long vectors. The latter is included in the library sentence_transformers which, as the name says, focuses on models designed to transform entire sentences rather than single words; the all-MiniLM-L6-v2 model returns 384-long vectors which should instead capture the semantic characteristics. A final combined version is achieved by concatenating the two embeddings, resulting in 534-long vectors.

Chapter 5

Model Training

Having prepared all the necessary input, it is finally possible to start training the classifiers. In this chapter, we will explore deeper this phase, examining all the different experiments conducted and measuring the performances of the several models created. A first comparison will be made between different formats of the same input, in order to understand the amount of data best suited for the tasks; then, there will also be a discussion about how the different data sources perform in predicting the two target labels. We will then move to multimodal models, combining the different inputs and training new models, analyzing the performances among them, but also juxtaposing their results with the one of the single input classifiers. Finally, there will be a discussion about the relation between emotion and genre, and the possibility of enhancing the predictability of the latter exploiting the former.

5.1 Single Input Models

In this section, we will analyze the models trained using only one source of data at a time. Before looking at the results for each type of input, some general remarks must be done. For audio features, both music and vocals, a scaler was used in order to standardize them, given the difference in the range of values of each domain; this was not the case for text features since standardizing the values of a word embedding would result in losing the distances between words and, thus, their (dis)similarities. The data was split between train and test, with

ratios of .75 and .25; the process happened taking into consideration the values of the target labels: if for emotions the stratification could have also been avoided, given the uniform distribution among classes, this was not the case for the genre label that, as seen before, presents an unbalanced placement of observations in the 15 categories.

The models were then fitted on the training set, using a grid search with a 5-fold cross validation, meaning that each possible combination of parameters for each classifier was trained on the 80% of the set and tested on the remaining part, used as validation set. The algorithms taken into consideration were chosen among the most common for classification tasks and were tuned according to the following parameters:

Table 5.1: Classifiers trained and their tuned parameters.

Classifier	Parameters
K-NN	number of neighbors, point weight
SVM	regularization parameter C, kernel type, kernel coefficient gamma, class weight
Random Forest	number of trees, split criteria
Logistic Regression	norm of the penalty, regularization strenght
MLP	size of hidden layers, optimizer, activation function, regularization term

The best combination of parameters for each classifier was chosen according to the average F1 score obtained on the validation test. The best estimator was then applied to the test set, computing all the necessary classification metrics: precision, recall, accuracy and F1 score; the confusion matrix was also retrieved. In the end, in order to define the best classifier for each data source, the F1 score computed on the test set was again chosen as the discriminating factor.

5.1.1 Vocals Models

The first input exploited are the features extracted from vocals, recalling that they assumed three formats: full song, 30 seconds excerpt and 10 seconds excerpt. Starting from the classification of emotions, results are shown in Table 5.2. The best performances are achieved by a multilayer perceptron applied on vocals_full, the set of features extracted from the whole song; in particular, the classifier was

Table 5.2: Results of emotion classification using vocal features.

Input	Classifier	F1	Accuracy	Precision	Recall
vocals_full	KNN	0.592797	0.594000	0.598015	0.594000
vocals_full	SVM	0.608976	0.610000	0.609587	0.610000
vocals_full	RF	0.593146	0.594000	0.592692	0.594000
vocals_full	LR	0.612045	0.612000	0.613853	0.612000
vocals_full	MLP	0.626106	0.626000	0.627601	0.626000
vocals_30s	KNN	0.508781	0.514000	0.514539	0.514000
vocals_30s	SVM	0.559578	0.562000	0.558137	0.562000
vocals_30s	RF	0.531433	0.532000	0.531717	0.532000
vocals_30s	LR	0.568747	0.570000	0.568378	0.570000
vocals_30s	MLP	0.580009	0.582000	0.580920	0.582000
vocals_10s	KNN	0.437102	0.442000	0.447295	0.442000
vocals_10s	SVM	0.457050	0.458000	0.457671	0.458000
vocals_10s	RF	0.443605	0.446000	0.445160	0.446000
vocals_10s	LR	0.469925	0.470000	0.472747	0.470000
vocals_10s	MLP	0.466972	0.466000	0.469020	0.466000



Figure 5.1: Confusion matrix for the best classifier of emotion using vocal features.

trained with a single hidden layer of size 32, adam used as optimizer, relu as activation function and an alpha of 1. Comparing the other results, it is clear that reducing the length of the original file, the performances lower drastically.

Observing the confusion matrix (Figure 5.1), it can be noticed the ability of the algorithm in discerning between the labels ‘angry’ and ‘happy’, while it encounters more difficulties when differentiating between ‘relaxed’ and ‘sad’.

The same process was applied to the genre classification, with results available in Table 5.3. Also in this case, the best model is a multilayer perceptron trained on the `vocals_full` set, this time using as activation the tanh function, and models trained on features extracted from shorter audio files perform always worse than the longer ones. Compared with the previous results, the values of the metrics are clearly lower, due to the higher complexity of this task and the greater number of categories in which observation are split into. From the confusion matrix (Figure 5.2) it is evident that genres which are more represented are better classified with respect to the others which, in some cases, are not even detected once.

Table 5.3: Results of genre classification using vocal features.

Input	Classifier	F1	Accuracy	Precision	Recall
<code>vocals_full</code>	KNN	0.407330	0.470000	0.406218	0.470000
<code>vocals_full</code>	SVM	0.435455	0.486000	0.409276	0.486000
<code>vocals_full</code>	RF	0.439444	0.504000	0.446480	0.504000
<code>vocals_full</code>	LR	0.445798	0.486000	0.433587	0.486000
<code>vocals_full</code>	MLP	0.453876	0.490000	0.437683	0.490000
<code>vocals_30s</code>	KNN	0.381651	0.424000	0.359246	0.424000
<code>vocals_30s</code>	SVM	0.381732	0.450000	0.352644	0.450000
<code>vocals_30s</code>	RF	0.399342	0.478000	0.408784	0.478000
<code>vocals_30s</code>	LR	0.403637	0.452000	0.385209	0.452000
<code>vocals_30s</code>	MLP	0.413482	0.450000	0.389348	0.450000
<code>vocals_10s</code>	KNN	0.302127	0.356000	0.316493	0.356000
<code>vocals_10s</code>	SVM	0.350867	0.414000	0.364858	0.414000
<code>vocals_10s</code>	RF	0.326300	0.398000	0.324929	0.398000
<code>vocals_10s</code>	LR	0.387403	0.436000	0.370635	0.436000
<code>vocals_10s</code>	MLP	0.355237	0.392000	0.333205	0.392000

5.1.2 Music Models

The same approach used for vocals features set were also used for music ones; the variables extracted from the three formats were parsed as input to the different

blues	0	0	1	0	0	0	0	0	3	0	0	0	2	0
classical	0	0	0	0	0	0	0	0	0	0	1	0	2	0
country	0	0	4	1	0	1	1	0	0	0	3	1	8	0
dance/electronic	0	0	0	0	0	0	1	0	0	2	1	2	0	11
easy listening	0	0	0	1	4	4	0	0	0	1	0	3	0	6
folk/acoustic	0	0	2	1	3	5	0	0	0	1	0	6	0	13
hip hop	0	0	0	0	1	0	21	0	0	1	0	1	0	1
jazz	0	0	0	0	0	0	0	0	0	0	0	0	4	0
latin	0	0	0	0	0	1	1	0	0	0	0	1	0	2
metal	0	0	0	1	0	0	3	0	0	64	2	3	0	23
new age	0	0	0	2	0	0	0	0	0	4	2	1	0	4
pop	0	0	4	1	1	3	3	0	0	0	30	3	19	0
r&b	0	0	0	0	0	0	0	0	0	3	0	7	2	9
rock	0	0	4	1	4	5	0	0	0	21	0	15	1	12
world/traditional	0	0	0	0	1	1	0	0	0	0	0	2	1	5

Predicted label

Figure 5.2: Confusion matrix for the best classifier of genre using vocal features.

classifiers. The results for emotion classification, displayed in Table 5.4, show that the best performances are achieved by a logistic regression, using l1 as penalty and with parameter C equal to 0.1, with a F1 score of about 0.6, slightly less than the one got with vocals; the same conclusions made before are also valid here, with feature extracted from longer files performing better than the shorter ones and the majority of errors coming from the inaccuracy in correctly distinguish ‘relaxed’ and ‘sad’, as shown in Figure 5.3. The results for genre classification (Table 5.5) are also in this case slightly lower than the one obtained with vocal features and, for the first time, there is not a clear superiority of data coming from longer files, even though the music_full is still the set achieving the highest F1 score, but with a lower margin compared to the other cases. This time the highest score is reached by the random forest classifier, trained with 50 trees and Gini as criterion. ‘rock’ and ‘metal’ are the labels better classified, with the former being also the cause of lots of false positives (Figure 5.4).

Table 5.4: Results of emotion classification using music features.

Input	Classifier	F1	Accuracy	Precision	Recall
music_full	KNN	0.565195	0.564000	0.572268	0.564000
music_full	SVM	0.592777	0.596000	0.595062	0.596000
music_full	RF	0.572236	0.574000	0.574009	0.574000
music_full	LR	0.603836	0.604000	0.603956	0.604000
music_full	MLP	0.586844	0.588000	0.589300	0.588000
music_30s	KNN	0.513297	0.522000	0.518780	0.522000
music_30s	SVM	0.548017	0.548000	0.549104	0.548000
music_30s	RF	0.542404	0.544000	0.547400	0.544000
music_30s	LR	0.548834	0.550000	0.549780	0.550000
music_30s	MLP	0.545215	0.546000	0.545729	0.546000
music_10s	KNN	0.502922	0.512000	0.508400	0.512000
music_10s	SVM	0.554560	0.552000	0.559575	0.552000
music_10s	RF	0.542091	0.544000	0.541740	0.544000
music_10s	LR	0.520227	0.522000	0.519164	0.522000
music_10s	MLP	0.537085	0.536000	0.538646	0.536000

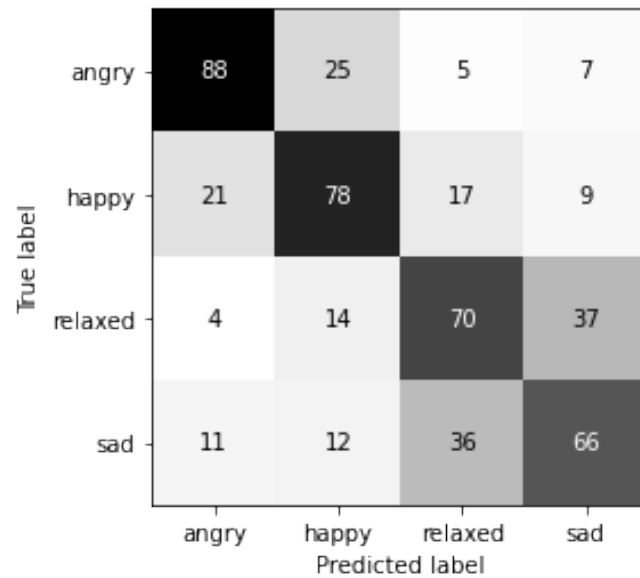


Figure 5.3: Confusion matrix for the best classifier of emotion using music features.

Table 5.5: Results of genre classification using music features.

Input	Classifier	F1	Accuracy	Precision	Recall
music_full	KNN	0.342443	0.390000	0.328321	0.390000
music_full	SVM	0.379689	0.408000	0.358756	0.408000
music_full	RF	0.388450	0.432000	0.423811	0.432000
music_full	LR	0.380216	0.408000	0.368085	0.408000
music_full	MLP	0.335876	0.340000	0.333230	0.340000
music_30s	KNN	0.343895	0.404000	0.322984	0.404000
music_30s	SVM	0.346543	0.394000	0.324131	0.394000
music_30s	RF	0.358931	0.424000	0.374650	0.424000
music_30s	LR	0.368945	0.404000	0.351340	0.404000
music_30s	MLP	0.355290	0.414000	0.339865	0.414000
music_10s	KNN	0.320323	0.398000	0.303620	0.398000
music_10s	SVM	0.382462	0.424000	0.394435	0.424000
music_10s	RF	0.363623	0.428000	0.409965	0.428000
music_10s	LR	0.369431	0.420000	0.361816	0.420000
music_10s	MLP	0.343183	0.410000	0.313304	0.410000

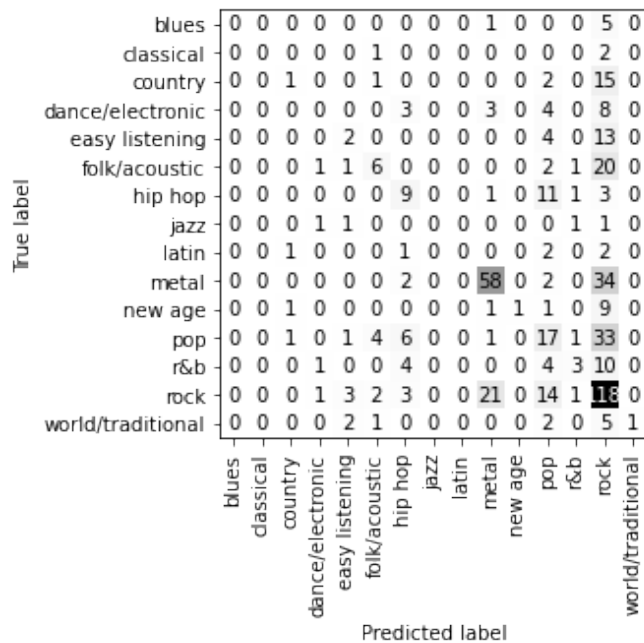


Figure 5.4: Confusion matrix for the best classifier of genre using music features.

5.1.3 Text Models

For text models, there is an additional level of comparison given to the availability of three word embeddings formats: SSWE, sBERT and the combination of the two. Given the amount of models generated, each case will be discussed individually.

5.1.3.1 Models Using SSWE

The first case taken into consideration is the one using the sentiment specific word embedding that, as we will see, will return the worst results, even for emotion classification, despite its ability of capturing the sentiment aspect. The results shown in Table 5.6 indicate as best estimator the random forest trained on features extracted from the full text, `lyrics_plain`, in compliance with the trend of audio inputs; the model, build on 200 trees and with the Gini criterion, encounters the majority of difficulties in detecting the ‘relaxed’ label, often confused with ‘happy’ and ‘sad’ (Figure 5.5). The genre classification results are not satisfactory at all and it does come not as a surprise; songs belonging to different genres may talk about the same topic, thus making lyrics not a discriminant factor for its classification. The results, however, are still presented for completeness, with the best model being the SVM trained on `lyrics_plain` (Table 5.7, Figure 5.6).

5.1.3.2 Models Using sBERT

The following models will exploit only the features retrieved using sBERT, therefore capturing only the semantic aspect of the text; it will be shown that they are the best in terms of performances for textual data. For emotion classification, the SVM trained on `lyrics_plain` is the best model, with a F1 score of 0.54; this time the differences with the other inputs are not so evident, with the classifiers trained on `lyrics_summarized` being very close in terms of performances (Table 5.8). Again, the mistake arise when the model has to predict ‘relaxed’ and ‘sad’, which are often confused (Figure 5.7). The results for genre classification are slightly better than before, but still far away from the performances obtained with audio features (Table 5.9, Figure 5.8).

Table 5.6: Results of emotion classification using SSWE features.

Input	Classifier	F1	Accuracy	Precision	Recall
lyrics_plain_sswe	KNN	0.359714	0.366000	0.371140	0.366000
lyrics_plain_sswe	SVM	0.379368	0.380000	0.383223	0.380000
lyrics_plain_sswe	RF	0.406558	0.410000	0.412490	0.410000
lyrics_plain_sswe	LR	0.118648	0.242000	0.105707	0.242000
lyrics_plain_sswe	MLP	0.217705	0.282000	0.207526	0.282000
lyrics_weighted_sswe	KNN	0.348888	0.356000	0.356789	0.356000
lyrics_weighted_sswe	SVM	0.371646	0.374000	0.374062	0.374000
lyrics_weighted_sswe	RF	0.361780	0.366000	0.361573	0.366000
lyrics_weighted_sswe	LR	0.163674	0.284000	0.426896	0.284000
lyrics_weighted_sswe	MLP	0.375041	0.380000	0.389365	0.380000
lyrics_choruses_sswe	KNN	0.304880	0.314000	0.309690	0.314000
lyrics_choruses_sswe	SVM	0.348272	0.350000	0.350622	0.350000
lyrics_choruses_sswe	RF	0.354788	0.360000	0.356803	0.360000
lyrics_choruses_sswe	LR	0.108295	0.254000	0.229544	0.254000
lyrics_choruses_sswe	MLP	0.318876	0.320000	0.320951	0.320000
lyrics_summarized_sswe	KNN	0.332131	0.332000	0.335413	0.332000
lyrics_summarized_sswe	SVM	0.375322	0.376000	0.376362	0.376000
lyrics_summarized_sswe	RF	0.378686	0.380000	0.379525	0.380000
lyrics_summarized_sswe	LR	0.099518	0.248000	0.062249	0.248000
lyrics_summarized_sswe	MLP	0.380956	0.384000	0.399321	0.384000

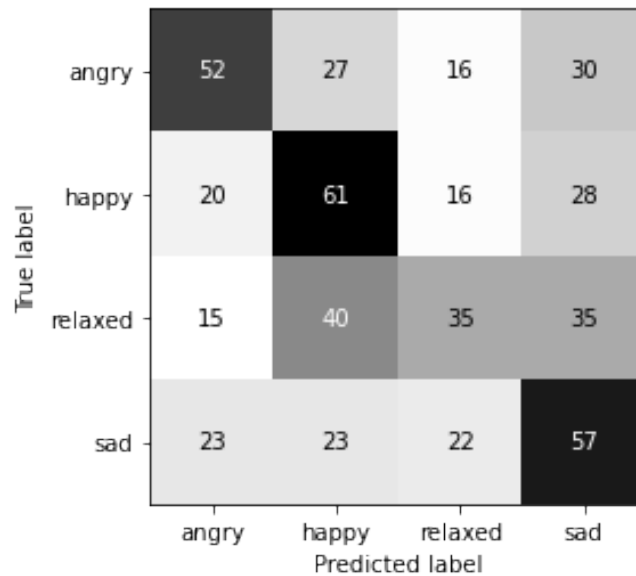


Figure 5.5: Confusion matrix for the best classifier of emotion using SSWE features.

Table 5.7: Results of genre classification using SSWE features.

Input	Classifier	F1	Accuracy	Precision	Recall
lyrics_plain_sswe	KNN	0.234143	0.304000	0.218421	0.304000
lyrics_plain_sswe	SVM	0.278204	0.318000	0.253491	0.318000
lyrics_plain_sswe	RF	0.247167	0.334000	0.243756	0.334000
lyrics_plain_sswe	LR	0.032170	0.024000	0.081526	0.024000
lyrics_plain_sswe	MLP	0.273362	0.322000	0.254902	0.322000
lyrics_weighted_sswe	KNN	0.209843	0.292000	0.184354	0.292000
lyrics_weighted_sswe	SVM	0.263732	0.310000	0.240761	0.310000
lyrics_weighted_sswe	RF	0.234169	0.332000	0.248707	0.332000
lyrics_weighted_sswe	LR	0.035295	0.030000	0.087533	0.030000
lyrics_weighted_sswe	MLP	0.261704	0.302000	0.269838	0.302000
lyrics_choruses_sswe	KNN	0.232361	0.298000	0.239408	0.298000
lyrics_choruses_sswe	SVM	0.266429	0.300000	0.253548	0.300000
lyrics_choruses_sswe	RF	0.249669	0.350000	0.315781	0.350000
lyrics_choruses_sswe	LR	0.004215	0.014000	0.108812	0.014000
lyrics_choruses_sswe	MLP	0.249535	0.318000	0.218807	0.318000
lyrics_summarized_sswe	KNN	0.240547	0.304000	0.241050	0.304000
lyrics_summarized_sswe	SVM	0.263003	0.282000	0.253930	0.282000
lyrics_summarized_sswe	RF	0.233945	0.318000	0.288879	0.318000
lyrics_summarized_sswe	LR	0.008096	0.016000	0.163145	0.016000
lyrics_summarized_sswe	MLP	0.245610	0.316000	0.219733	0.316000

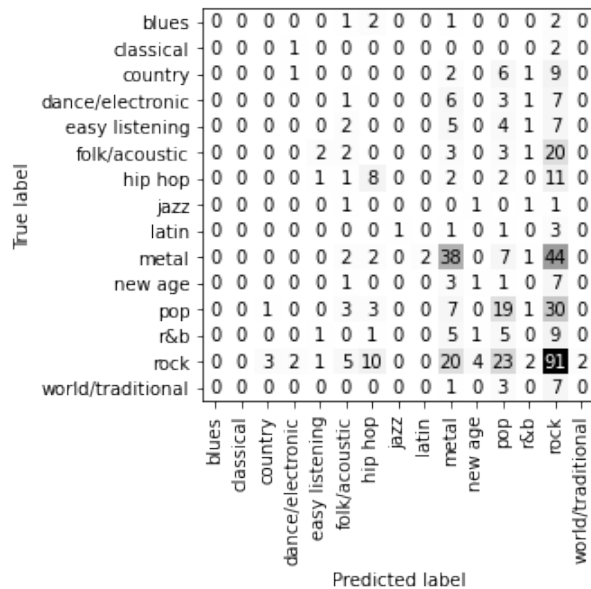


Figure 5.6: Confusion matrix for the best classifier of genre using SSWE features.

Table 5.8: Results of emotion classification using sBERT features.

Input	Classifier	F1	Accuracy	Precision	Recall
lyrics_plain_sbert	KNN	0.411777	0.426000	0.432376	0.426000
lyrics_plain_sbert	SVM	0.540439	0.540000	0.541351	0.540000
lyrics_plain_sbert	RF	0.493456	0.500000	0.492029	0.500000
lyrics_plain_sbert	LR	0.524187	0.528000	0.523146	0.528000
lyrics_plain_sbert	MLP	0.509852	0.510000	0.509822	0.510000
lyrics_weighted_sbert	KNN	0.417989	0.424000	0.436554	0.424000
lyrics_weighted_sbert	SVM	0.488210	0.494000	0.485139	0.494000
lyrics_weighted_sbert	RF	0.443387	0.448000	0.441064	0.448000
lyrics_weighted_sbert	LR	0.485930	0.494000	0.485444	0.494000
lyrics_weighted_sbert	MLP	0.513944	0.522000	0.513349	0.522000
lyrics_choruses_sbert	KNN	0.451726	0.454000	0.466845	0.454000
lyrics_choruses_sbert	SVM	0.512292	0.516000	0.514531	0.516000
lyrics_choruses_sbert	RF	0.449916	0.456000	0.450478	0.456000
lyrics_choruses_sbert	LR	0.498779	0.502000	0.499091	0.502000
lyrics_choruses_sbert	MLP	0.493811	0.498000	0.494274	0.498000
lyrics_summarized_sbert	KNN	0.454119	0.452000	0.467382	0.452000
lyrics_summarized_sbert	SVM	0.532397	0.534000	0.532662	0.534000
lyrics_summarized_sbert	RF	0.486821	0.492000	0.489610	0.492000
lyrics_summarized_sbert	LR	0.531824	0.534000	0.533698	0.534000
lyrics_summarized_sbert	MLP	0.528857	0.532000	0.530665	0.532000

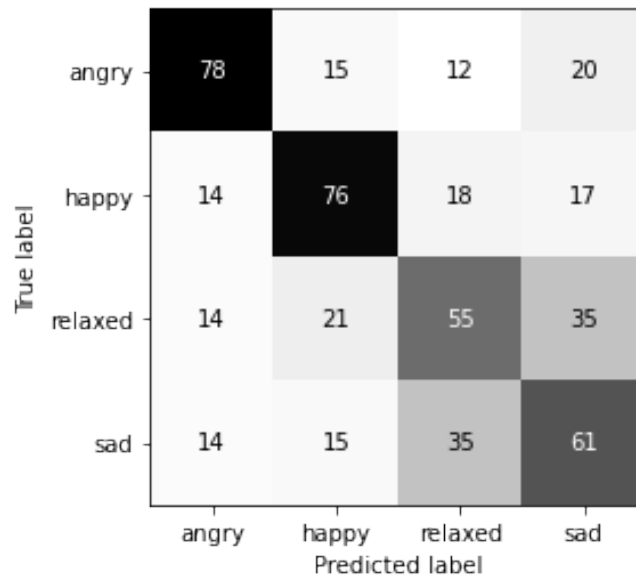


Figure 5.7: Confusion matrix for the best classifier of emotion using sBERT features.

Table 5.9: Results of genre classification using sBERT features.

Input	Classifier	F1	Accuracy	Precision	Recall
lyrics_plain_sbert	KNN	0.273006	0.360000	0.251492	0.360000
lyrics_plain_sbert	SVM	0.296337	0.372000	0.319548	0.372000
lyrics_plain_sbert	RF	0.281481	0.378000	0.297377	0.378000
lyrics_plain_sbert	LR	0.291284	0.352000	0.264782	0.352000
lyrics_plain_sbert	MLP	0.313117	0.372000	0.286331	0.372000
lyrics_weighted_sbert	KNN	0.255710	0.338000	0.244271	0.338000
lyrics_weighted_sbert	SVM	0.289394	0.342000	0.266949	0.342000
lyrics_weighted_sbert	RF	0.262229	0.356000	0.269474	0.356000
lyrics_weighted_sbert	LR	0.275399	0.318000	0.255519	0.318000
lyrics_weighted_sbert	MLP	0.277630	0.342000	0.240522	0.342000
lyrics_choruses_sbert	KNN	0.282530	0.342000	0.274229	0.342000
lyrics_choruses_sbert	SVM	0.302253	0.358000	0.321502	0.358000
lyrics_choruses_sbert	RF	0.235792	0.334000	0.293823	0.334000
lyrics_choruses_sbert	LR	0.290754	0.336000	0.272797	0.336000
lyrics_choruses_sbert	MLP	0.266827	0.312000	0.241419	0.312000
lyrics_summarized_sbert	KNN	0.304260	0.370000	0.304788	0.370000
lyrics_summarized_sbert	SVM	0.299529	0.352000	0.315696	0.352000
lyrics_summarized_sbert	RF	0.242709	0.330000	0.272931	0.330000
lyrics_summarized_sbert	LR	0.292445	0.318000	0.278046	0.318000
lyrics_summarized_sbert	MLP	0.307835	0.338000	0.291810	0.338000

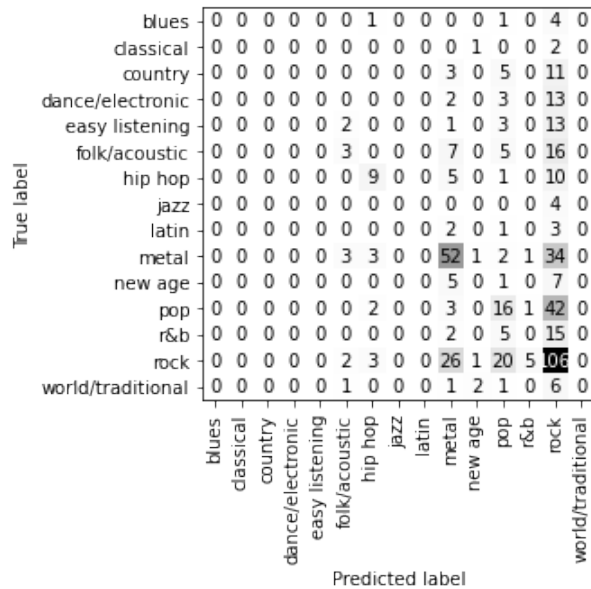


Figure 5.8: Confusion matrix for the best classifier of genre using sBERT features.

5.1.3.3 Models Using SSWE and sBERT

In this section, there will be displayed the results for models trained on SSWE and sBERT features together; the classifiers are not an improvement if compared with the one built only upon sBERT features, which are therefore the one that will be considered for multimodal high-level fusion models. Performances are still better than the one of models using only SSWE, with the best classifier for emotion detection being the MLP applied on lyrics_plain (Table 5.10, Figure 5.9); for genre classification, we have the only case in which features extracted from a shorter input are the ones on which the best classifier is built: indeed, the MLP trained on lyrics_choruses reaches the highest score (Table 5.11, Figure 5.10).

Table 5.10: Results of emotion classification using SSWE and sBERT features.

Input	Classifier	F1	Accuracy	Precision	Recall
lyrics_plain	KNN	0.367216	0.374000	0.379229	0.374000
lyrics_plain	SVM	0.396324	0.398000	0.395186	0.398000
lyrics_plain	RF	0.501516	0.504000	0.502388	0.504000
lyrics_plain	LR	0.118648	0.242000	0.105707	0.242000
lyrics_plain	MLP	0.502174	0.500000	0.514701	0.500000
lyrics_weighted	KNN	0.355494	0.362000	0.366822	0.362000
lyrics_weighted	SVM	0.425290	0.428000	0.429358	0.428000
lyrics_weighted	RF	0.442277	0.446000	0.443182	0.446000
lyrics_weighted	LR	0.159381	0.282000	0.174283	0.282000
lyrics_weighted	MLP	0.440409	0.446000	0.445520	0.446000
lyrics_choruses	KNN	0.308919	0.320000	0.313276	0.320000
lyrics_choruses	SVM	0.417635	0.424000	0.417433	0.424000
lyrics_choruses	RF	0.441647	0.450000	0.441555	0.450000
lyrics_choruses	LR	0.108295	0.254000	0.229544	0.254000
lyrics_choruses	MLP	0.443177	0.448000	0.442902	0.448000
lyrics_summarized	KNN	0.337663	0.338000	0.341683	0.338000
lyrics_summarized	SVM	0.459642	0.462000	0.460875	0.462000
lyrics_summarized	RF	0.493796	0.496000	0.495176	0.496000
lyrics_summarized	LR	0.099518	0.248000	0.062249	0.248000
lyrics_summarized	MLP	0.468769	0.472000	0.471035	0.472000

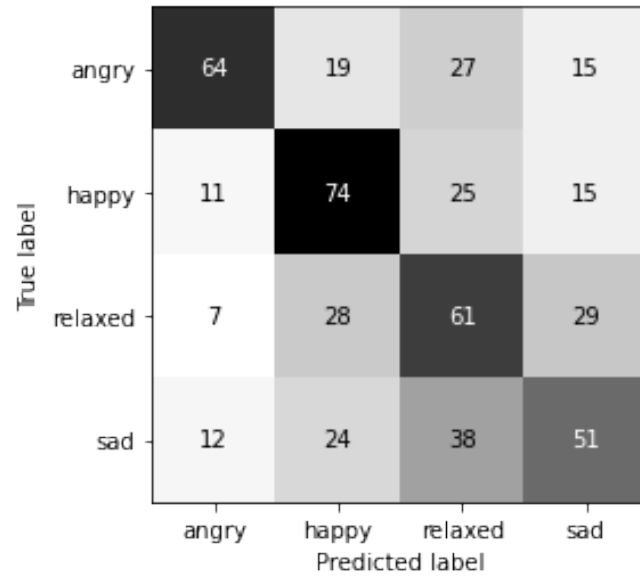


Figure 5.9: Confusion matrix for the best classifier of emotion using SSWE and sBERT features.

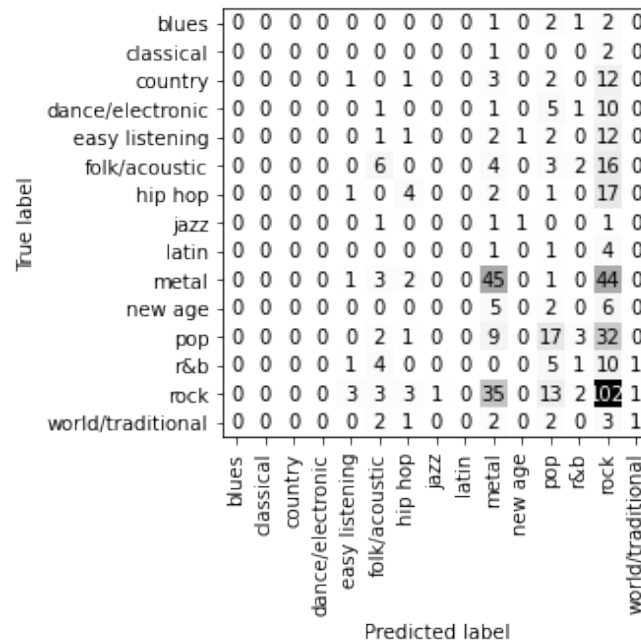


Figure 5.10: Confusion matrix for the best classifier of genre using SSWE and sBERT features.

Table 5.11: Results of genre classification using SSWE and sBERT features.

Input	Classifier	F1	Accuracy	Precision	Recall
lyrics_plain	KNN	0.238720	0.310000	0.225026	0.310000
lyrics_plain	SVM	0.284745	0.304000	0.276233	0.304000
lyrics_plain	RF	0.286369	0.376000	0.292883	0.376000
lyrics_plain	LR	0.032170	0.024000	0.081526	0.024000
lyrics_plain	MLP	0.289050	0.310000	0.278502	0.310000
lyrics_weighted	KNN	0.214908	0.296000	0.192294	0.296000
lyrics_weighted	SVM	0.286777	0.330000	0.265385	0.330000
lyrics_weighted	RF	0.267756	0.370000	0.287898	0.370000
lyrics_weighted	LR	0.035295	0.030000	0.087533	0.030000
lyrics_weighted	MLP	0.268476	0.320000	0.242798	0.320000
lyrics_choruses	KNN	0.233726	0.292000	0.235069	0.292000
lyrics_choruses	SVM	0.270547	0.306000	0.257430	0.306000
lyrics_choruses	RF	0.270628	0.368000	0.329081	0.368000
lyrics_choruses	LR	0.004215	0.014000	0.108812	0.014000
lyrics_choruses	MLP	0.302073	0.352000	0.281094	0.352000
lyrics_summarized	KNN	0.239478	0.300000	0.244542	0.300000
lyrics_summarized	SVM	0.297849	0.322000	0.295196	0.322000
lyrics_summarized	RF	0.263954	0.358000	0.332549	0.358000
lyrics_summarized	LR	0.008096	0.016000	0.163145	0.016000
lyrics_summarized	MLP	0.298039	0.344000	0.279474	0.344000

5.2 Multimodal Models

Having analyzed all the single input models, the project will continue exploring the two multimodal approaches proposed: high-level and low-level fusion. The classifiers previously trained will be reused in the process of building multimodal high-level fusion models, in order to predict the probabilities of the labels used in the soft voting process; while in multimodal low-level fusion models, the method of combining grid search and cross validation will be applied also here.

5.2.1 High-Level Fusion Models

In order to build these models, I retrieved the best estimators for each of the three inputs (vocals, music and lyrics) and for both of the dependent variables (emotion and genre), resulting in a total of six classifiers, three for each target. These three models were then combined in pairs and also all three together, in order to improve the prediction performances. Each classifier is applied on the test set of the previous section and for each observation the probabilities of the labels are estimated; then, based on the combination considered, these probabilities are summed and the label chosen as prediction is the one having the highest value. The classification metrics are computed for each model to make possible their comparison.

The full list of results for the classification task is displayed in Table 5.12. The best performance is achieved when combining all three inputs, resulting in an F1 score of 0.68, an increase of almost 0.06 with respect with the best previous estimator, the one built only on vocals. Also the confusion matrix (Figure 5.11) shows an improvement in predictions, despite the still presence of the evident confusion between ‘relaxed’ and ‘sad’.

Table 5.12: Results of emotion classification using multimodal high-level fusion approaches.

Input	F1	Accuracy	Precision	Recall
vocals_music	0.6503994004800938	0.65	0.6528932754671266	0.65
vocals_lyrics	0.6524193286075226	0.652	0.6532953169862276	0.652
music_lyrics	0.6446075005823434	0.646	0.6447449122526163	0.646
vocals_music_lyrics	0.6801279196777354	0.68	0.6824330465714766	0.68

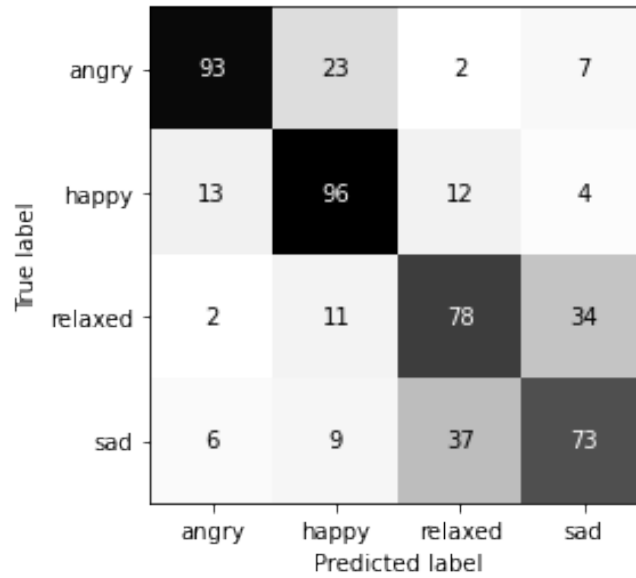


Figure 5.11: Confusion matrix for the best classifier of emotion using multimodal high-level fusion approaches.

For the genre classification task there is only a minor improvement with respect to the previously trained models and it is achieved by combining vocals and music classifiers; involving the lyrics one would only lower dramatically the performances (Table 5.13, Figure 5.12).

Table 5.13: Results of genre classification using multimodal high-level fusion approaches.

Input	F1	Accuracy	Precision	Recall
vocals_music	0.46438374908777097	0.516	0.4818013546018045	0.516
vocals_lyrics	0.43337613216664633	0.502	0.4385994229136536	0.502
music_lyrics	0.35878506701996327	0.428	0.3684660852713178	0.428
vocals_music_lyrics	0.43814012020909243	0.512	0.4978306621199205	0.512

5.2.2 Low-Level Fusion Models

With low-level fusion, the merge happens at measurements level, meaning that the concatenation is done directly on raw data. For each source of input, the dataset chosen is the one which performed better on single input classifiers and

True label	blues	0	0	0	0	0	0	0	0	3	0	0	0	3	0
	classical	0	0	0	0	0	0	0	0	0	0	0	0	3	0
	country	0	0	1	0	0	0	1	0	0	0	2	1	14	0
	dance/electronic	0	0	0	0	0	0	1	0	0	0	3	0	13	1
	easy listening	0	0	0	0	4	2	0	0	0	0	2	0	11	0
	folk/acoustic	0	0	1	0	2	5	0	0	0	0	3	0	20	0
	hip hop	0	0	0	0	0	0	20	0	0	1	0	2	0	2
	jazz	0	0	0	0	0	0	0	0	0	0	0	0	4	0
	latin	0	0	0	0	0	1	1	0	0	0	2	0	2	0
	metal	0	0	0	0	0	2	0	0	70	0	0	0	24	0
	new age	0	0	0	0	0	0	0	0	2	2	1	0	8	0
	pop	0	0	2	1	1	2	3	0	0	0	25	1	29	0
	r&b	0	0	0	0	0	0	0	0	1	0	8	2	11	0
	rock	0	0	2	0	0	5	0	0	0	17	0	10	1	28
	world/traditional	0	0	0	0	1	1	0	0	0	0	1	0	7	1
		blues													
	classical														
	country														
	dance/electronic														
	easy listening														
	folk/acoustic														
	hip hop														
	jazz														
	latin														
	metal														
	new age														
	pop														
	r&b														
	rock														
	world/traditional														

Figure 5.12: Confusion matrix for the best classifier of genre using multimodal high-level fusion approaches.

thus equivalent to the set of features extracted from the longest format in all the three cases: `vocals_full` and `music_full` for audio features and `lyrics_plain` for textual ones. The set of variables are then concatenated according to the combinations seen before for building high-level fusion models. The resulting data streams are used as input for training the previously listed classifiers and tuning their parameters, using grid search and cross validation.

The results for emotion classification are displayed in Table 5.14, with the best classifier being the multilayer perceptron built on all the three inputs; the performances, however, are marginally lower with the respect to the best high-level fusion model. In this case, the label which is more misclassified is ‘sad’, with ‘relaxed’ being almost on the same level of ‘happy’ and ‘angry’ (Figure 5.13).

For genre classification, as in the case of high-level fusion models, the improvements from single input models are slight, reaching in this case the highest F1 score until now of 0.479 with a logistic regression built on vocals and music features; again, it is demonstrated the poor ability of textual data in predicting genres (Table 5.15). For the first time, genres like ‘dance’ and ‘jazz’ are predicted correctly in at least one case (Figure 5.14).

Table 5.14: Results of emotion classification using multimodal low-level fusion approaches.

Input	Classifier	F1	Accuracy	Precision	Recall
vocals_music	KNN	0.633813	0.632000	0.643029	0.632000
vocals_music	SVM	0.637851	0.638000	0.638926	0.638000
vocals_music	RF	0.650899	0.652000	0.651902	0.652000
vocals_music	LR	0.636399	0.636000	0.637529	0.636000
vocals_music	MLP	0.634035	0.634000	0.636725	0.634000
vocals_lyrics	KNN	0.592934	0.594000	0.597533	0.594000
vocals_lyrics	SVM	0.634726	0.638000	0.635212	0.638000
vocals_lyrics	RF	0.632473	0.634000	0.632078	0.634000
vocals_lyrics	LR	0.608002	0.608000	0.609257	0.608000
vocals_lyrics	MLP	0.670521	0.670000	0.671295	0.670000
music_lyrics	KNN	0.571349	0.570000	0.578527	0.570000
music_lyrics	SVM	0.590425	0.594000	0.592687	0.594000
music_lyrics	RF	0.588219	0.592000	0.589265	0.592000
music_lyrics	LR	0.603836	0.604000	0.603956	0.604000
music_lyrics	MLP	0.634619	0.634000	0.638182	0.634000
vocals_music_lyrics	KNN	0.631900	0.630000	0.641077	0.630000
vocals_music_lyrics	SVM	0.636588	0.638000	0.637213	0.638000
vocals_music_lyrics	RF	0.645222	0.646000	0.646301	0.646000
vocals_music_lyrics	LR	0.636399	0.636000	0.637529	0.636000
vocals_music_lyrics	MLP	0.676129	0.678000	0.676091	0.678000

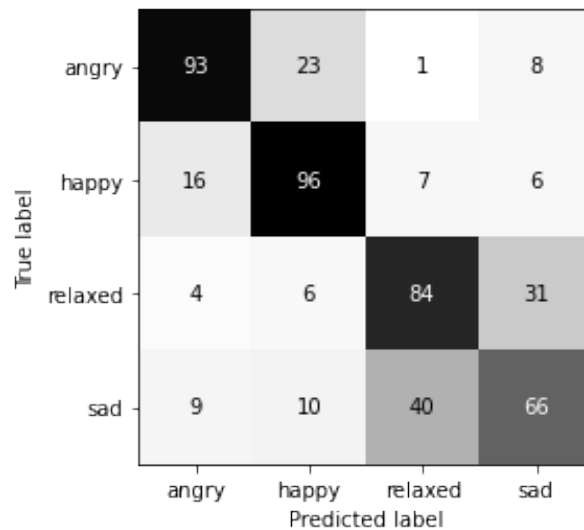


Figure 5.13: Confusion matrix for the best classifier of emotion using multimodal low-level fusion approaches.

Table 5.15: Results of genre classification using multimodal low-level fusion approaches.

Input	Classifier	F1	Accuracy	Precision	Recall
vocals_music	KNN	0.435219	0.476000	0.468840	0.476000
vocals_music	SVM	0.418293	0.432000	0.445172	0.432000
vocals_music	RF	0.443670	0.512000	0.453467	0.512000
vocals_music	LR	0.479855	0.500000	0.468847	0.500000
vocals_music	MLP	0.465979	0.520000	0.464969	0.520000
vocals_lyrics	KNN	0.412156	0.474000	0.410629	0.474000
vocals_lyrics	SVM	0.443637	0.462000	0.433293	0.462000
vocals_lyrics	RF	0.392157	0.470000	0.389388	0.470000
vocals_lyrics	LR	0.450802	0.488000	0.440381	0.488000
vocals_lyrics	MLP	0.463675	0.498000	0.455341	0.498000
music_lyrics	KNN	0.345785	0.394000	0.331391	0.394000
music_lyrics	SVM	0.381668	0.410000	0.361160	0.410000
music_lyrics	RF	0.368318	0.434000	0.387969	0.434000
music_lyrics	LR	0.386874	0.416000	0.373717	0.416000
music_lyrics	MLP	0.364827	0.378000	0.357393	0.378000
vocals_music_lyrics	KNN	0.435219	0.476000	0.468840	0.476000
vocals_music_lyrics	SVM	0.419615	0.434000	0.447196	0.434000
vocals_music_lyrics	RF	0.413641	0.490000	0.462013	0.490000
vocals_music_lyrics	LR	0.478328	0.498000	0.467876	0.498000
vocals_music_lyrics	MLP	0.453048	0.516000	0.445447	0.516000

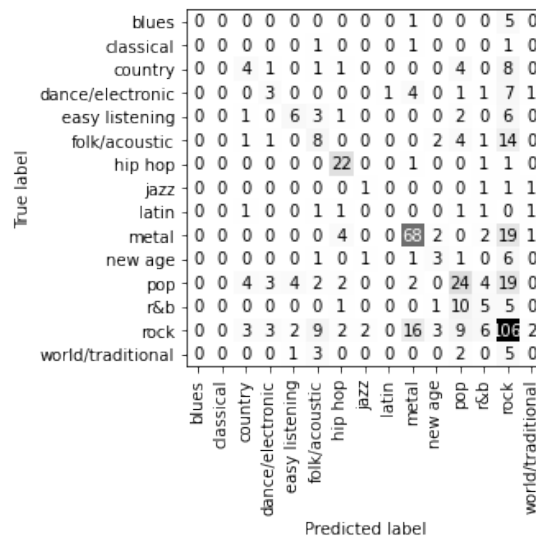


Figure 5.14: Confusion matrix for the best classifier of genre using multimodal low-level fusion approaches.

5.3 Emotion-Genre Relation

The final experiment performed in this analysis concerns the relation between the two target variables, emotion and genre, and the possibility of enhancing the predictions of the latter knowing the former. Up until now, the performances for the genre classifiers have not been at the level of the ones for the classification of emotion; therefore, in the following sections, the information about emotions will be put at disposal of the models predicting genres. Firstly we will explore if and which kind of relation exists between these two variables and then we will try to improve the classification of genres.

5.3.1 Correlation Analysis

In order to discover the presence of some sort of relationship between the two variables, I performed a correlation analysis which, in case of two categorical variables, is represented by the χ^2 test. Firstly, a contingency table was built, in order to observe the distribution of songs among the two variables combined (Table 5.16). Then the test was run, resulting in a score of 933.89, with a p-value

Table 5.16: Contingency table between genres and emotions.

	angry	happy	relaxed	sad
blues	11	7	2	3
classical	1	2	4	6
country	1	25	19	32
dance/electronic	3	20	40	10
easy listening	1	22	43	11
folk/acoustic	4	14	63	45
hip hop	49	30	11	11
jazz	0	1	16	0
latin	2	11	9	2
metal	242	22	11	108
new age	3	1	26	20
pop	9	125	55	66
r&b	2	25	55	5
rock	171	186	119	175
world/traditional	1	9	27	6

of $1.684e-168$, allowing the refuse the null hypothesis of independence. The same test was also run on group of emotions, rather than single ones. In the first case, they were group according to the valence value, with ‘happy’ and ‘relaxed’ in the positive group while ‘angry’ and ‘sad’ in the negative one; then, the grouping happened according to the arousal aspect, arranging ‘happy’ and ‘angry’ in the active group, while ‘sad’ and ‘relaxed’ in the passive one. The test confirmed in both cases the presence of a strongly significant dependence between the two variables.

5.3.2 Conditional Probabilities

In order to improve the performances of the genre classifiers, the prior knowledge about the emotion expressed by the song is employed. The methods assessed for embedding this information in the predictions are basically two: the first including the emotion as a dependent variable in the training process, the second adopting conditional probabilities and adjusting the final predictions according to them. The former approach does not bring in any improvement in the classification results, while the latter enhances the scores, returning the best genre classifier with a significant boost in F1 measure (Table 5.17, Figure 5.15).

Table 5.17: Results of genre classification using emotion conditional probabilities.

Metric	Score
F1	0.5088043848554519
Accuracy	0.52
Precision	0.505866473548899
Recall	0.52

True label	blues	0	0	0	0	0	0	0	0	1	0	0	0	5	0
	classical	0	0	0	0	0	1	0	0	0	1	0	0	0	1
	country	0	0	5	0	0	2	1	0	0	0	0	4	0	7
	dance/electronic	0	0	0	2	0	1	0	1	1	1	2	2	1	7
	easy listening	0	0	1	0	7	3	0	0	0	0	0	2	0	6
	folk/acoustic	0	0	1	2	0	9	0	0	0	1	2	4	1	11
	hip hop	0	0	0	0	0	0	22	0	0	1	0	0	1	1
	jazz	0	0	0	0	0	0	0	1	0	0	0	0	1	1
	latin	0	0	1	0	0	1	1	0	1	0	0	1	0	0
	metal	0	0	0	0	0	0	3	0	0	72	3	0	1	17
	new age	0	0	0	0	0	1	0	1	0	2	3	1	1	4
	pop	0	0	4	3	1	6	0	0	0	1	1	28	3	16
	r&b	0	0	0	0	1	1	1	0	0	0	1	5	10	3
	rock	1	0	4	4	2	9	2	2	0	21	2	8	8	98
	world/traditional	0	0	0	0	1	3	0	0	0	0	0	1	1	3
		blues	classical	country	dance/electronic	easy listening	folk/acoustic	hip hop	jazz	latin	metal	new age	pop	r&b	rock
	Predicted label														

Figure 5.15: Confusion matrix for the classifier of genre using emotion conditional probabilities.

Chapter 6

Conclusion

In the final chapter of this work, there will be discussed the most important discoveries and conclusions that can be made about the experiments run, from the performances of multimodal models over single input ones, to the comparison between the algorithms involved in the classification processes. There will also be a short discussion about possible developments and continuations of this research.

6.1 Research Findings

Multimodal models are without doubt the path to follow not only in music emotion recognition, but in any field that involves multiple data sources. The increase gained in performances is significant enough to justify the extra step of merging the inputs and generate new and more powerful classifiers. For emotion detection, the improvement brought by multimodal models is significant with the respect to the results obtained in single input classifiers: about 0.06 points more in F1 score when adding music and lyrics features to the ones extracted by vocals. Although, the metrics are better in the case of high-level fusion than in the low-level fusion one, the differences are negligible and may be attributed to the nature of the data or to the necessity of a further parameter tuning. All the three sources of input seems to have a significant role in predicting emotions, with audio features slightly outperforming textual ones.

The same can not be affirmed for genre classification where, as expected, lyrics appear not to be capable of discerning between the different categories as audio

inputs can; however, this task turned out to be challenging, with scores much lower compared to the one of emotion classification, probably due to the higher amount of target labels and to the complexity of detecting them. Furthermore, the improvement brought by multimodal approaches is not sufficiently meaningful, with the only decent combination being the one of vocals and music features. The performances become marginally better when introducing the prior knowledge of the emotion expressed by the song, combining the predicted probabilities with the conditional ones. Any conclusion can be made about high-level fusion and low-level fusion models: they certainly boost the performances of single input models, but neither of the two can be considered as more efficient than the other, the former performing better on emotion detection while the latter on genre classification.

Another important conclusion can be made about the format of the inputs employed to train the model: in every case, classifiers built upon features extracted from the complete source file had better results than the others. This meaning that vocals and music variables are more useful when extracted from the entire song, as well as word embeddings built on the full lyrics rather than smaller parts. While these differences are relevant for audio features, they are minor for textual ones; indeed, the performances of models built on the set of summaries, the ones generated with the T5 model, were close to the results of classifiers trained on the set lyrics_plain.

For what concerns the algorithms used, the most valid one seems to be the multilayer perceptron, which is able to achieve high scores in almost any experiments regardless of the type of the input; it is the best classifier when using vocals features, but also in the low-level fusion model for emotion classification. Surprisingly, a simple model like logistic regression reveals to be one of the most powerful ones, especially when dealing with audio features; indeed, it returns the best model for genre classification, where are only considered variables from vocals and music inputs. Support vector machines behave better when analyzing only textual features, as it can be seen from their performances on lyrics data; random forests, while returning still acceptable results, they do not excel in any particular task. Finally, classifiers based on the k-nearest neighbor algorithm performed usually worse than any other model.

An interesting deduction can be drawn looking at the confusion matrix of the

emotion classification algorithms: there seems to be a recurrent pattern, showing a better ability of discerning emotions according to their arousal value rather than their valence. Indeed, labels are often confused in pair of two, with more mistakes happening between the couple ‘happy’ and ‘angry’, and between ‘relaxed’ and ‘sad’, with this behavior being more clear especially in the last case. For genre classification, the majority of errors concerns the label ‘rock’, being the one most represented in the dataset and, thus, causing the majority of false positives and negatives; in particular, it is often confused with the label ‘metal’, maybe given the similarities in some cases of the two genres.

6.2 Future Developments

The current project can be certainly developed further, starting from the dataset employed for training the models; more observation would allow to better capture the peculiarities of each category, in particular for genre classification, where a more uniform distribution of observations would have benefited for sure the performances of the algorithms. A refined version of the method for retrieving emotion labels could be employed in order to augment the size of the dataset considered. Regarding the analysis of the data, audio features could be extracted over shorter time frames, allowing to gain more information with respect to values averaged over the whole sound wave; for textual data, a better method for retrieving the sentiment aspect of the data could be explored. Finally, in the phase of model training, given the number of independent variables considered, options involving neural networks could be taken into account, maybe involving also the temporal aspect, like in LSTM architectures.

The results of the analysis can be applied to emotion detection tasks in other related fields, like movies, advertisements or other media formats. In these cases, sources of input could be identified in the voices of the interpreters, the transcript of the dialogs and the background music; an additional layer could be represented by the visual aspect, extracting facial and body expressions. This last characteristic, while not being proper related to songs, could be also implemented in further works related to music emotion recognition, taking into consideration official music videos or live performances. The final purpose of these kind of researches could be developing recommendation systems for streaming platforms

or e-commerce websites, to enhance the ability of proposing relevant items to the users according to their preferences or current mood.

Bibliography

- [1] Carlos Busso, Zhigang Deng, Serdar Yildirim, Murtaza Bulut, Chul Min Lee, Abe Kazemzadeh, Sungbok Lee, Ulrich Neumann, and Shrikanth Narayanan. Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, pages 205–211, New York, NY, USA, 2004. Association for Computing Machinery.
- [2] Rafael A Calvo and Sidney D'Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. 1(1):18–37, January 2010.
- [3] Erion Çano. Text-based sentiment analysis and music emotion recognition. 10 2018.
- [4] Erion Cano and Maurizio Morisio. Music mood dataset creation based on last FM tags. Academy & Industry Research Collaboration Center (AIRCC), May 2017.
- [5] Federico Castanedo. A review of data fusion techniques. *TheScientificWorld-Journal*, 2013, 10 2013.
- [6] Devopedia. Audio feature extraction, May 2021.
- [7] Paul Ekman. An argument for basic emotions. *Cognition & Emotion*, 6:169–200, 1992.
- [8] Florian Eyben, Klaus Scherer, Björn Schuller, Johan Sundberg, Elisabeth Andre, Carlos Busso, Laurence Devillers, Julien Epps, Petri Laukka, Shrikanth Narayanan, and Khiet Truong. The geneva minimalistic acoustic

- parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7:1–1, 01 2015.
- [9] Beverley Fehr and James A. Russell. Concept of emotion viewed from a prototype perspective. 113(3):464–486, 1984.
- [10] Masataka Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech, and Language Processing*, 14:1783–1794, 2006.
- [11] Umang Gupta, Ankush Chatterjee, Radhakrishnan Srikanth, and Puneet Agrawal. A sentiment-and-semantics-based approach for emotion detection in textual conversations. *CoRR*, abs/1707.06996, 2017.
- [12] Byeong-jun Han, Seungmin Rho, Roger Dannenberg, and Eenjun Hwang. Smers: Music emotion recognition using support vector regression. pages 651–656, 01 2009.
- [13] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. Spleeter: a fast and efficient music source separation tool with pre-trained models. 5(50):2154, June 2020.
- [14] Serhat Hızlısoy, Serdar Yildirim, and Zekeriya Tüfekci. Music emotion recognition using convolutional long short term memory deep neural networks. *Engineering Science and Technology an International Journal*, 24:760–767, 11 2020.
- [15] Adit Jamdar, Jessica Abraham, Karishma Khanna, and Rahul Dubey. Emotion analysis of songs based on lyrical and audio features. *CoRR*, abs/1506.05012, 2015.
- [16] Olivier Lartillot, Petri Toivianen, and Tuomas Eerola. A matlab toolbox for music information retrieval. pages 261–268. Springer Berlin Heidelberg, 2008.
- [17] C. Laurier and P. Herrera. Automatic detection of emotion in music: Interaction with emotionally sensitive machines. 2009.

-
- [18] Cyril Laurier and Perfecto Herrera. Audio music mood classification using support vector machine. *MIREX Task on Audio Mood Classification*, 01 2007.
 - [19] Deepali. Y. Loni and Dr. Shaila Subbaraman. Extracting acoustic features of singing voice for various applications related to mir: A review. 2013.
 - [20] Lucía Martín-Gómez and María Cáceres. Applying data mining for sentiment analysis in music. pages 198–205, 06 2018.
 - [21] Albert Mehrabian and James A Russell. An approach to environmental psychology. 1974.
 - [22] Louis-Philippe Morency, Rada Mihalcea, and Payal Doshi. Towards multimodal sentiment analysis: Harvesting opinions from the web. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, pages 169–176, New York, NY, USA, 2011. Association for Computing Machinery.
 - [23] Meinard Müller. *Fundamentals of Music Processing*. Springer International Publishing, 2015.
 - [24] Loreto Parisi, Simone Francia, Silvio Olivastri, and Maria Tavella. Exploiting synchronized lyrics and vocal features for music emotion detection. 01 2019.
 - [25] Verónica Pérez-Rosas, Rada Mihalcea, and Louis-Philippe Morency. Multimodal sentiment analysis of spanish online videos. *IEEE Intelligent Systems*, 28:38–45, 2013.
 - [26] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2539–2544, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
 - [27] Soujanya Poria, Erik Cambria, Newton Howard, Guang-Bin Huang, and Amir Hussain. Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing*, 174, 08 2015.

-
- [28] Fika Hastarita Rachman, Riyanarto Samo, and Chastine Fatichah. Song emotion detection based on arousal-valence from audio and lyrics using rule based method. *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, pages 1–5, 2019.
 - [29] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019.
 - [30] Kashfia Sailunaz, Manmeet Dhaliwal, Jon Rokne, and Reda Alhajj. Emotion detection from text and speech - a survey. 03 2018.
 - [31] Behjat Siddiquie, Dave Chisholm, and Ajay Divakaran. Exploiting multimodal affect and semantics to identify politically persuasive web videos. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 203–210, New York, NY, USA, 2015. Association for Computing Machinery.
 - [32] Agnieszka Smolinska, Jasper Engel, Ewa Szymanska, Lutgarde Buydens, and Lionel Blanchet. *General Framing of Low-, Mid-, and High-Level Data Fusion With Examples in the Life Sciences*, pages 51–79. Data Handling in Science and Technology. Elsevier Ltd, Academic Press, 2019.
 - [33] M. Soleymani, David García, Brendan Jou, Björn Schuller, Shih-Fu Chang, and Maja Pantic. A survey of multimodal sentiment analysis. *Image Vis. Comput.*, 65:3–14, 2017.
 - [34] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for Twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565, Baltimore, Maryland, June 2014. Association for Computational Linguistics.
 - [35] Martin Wollmer, Felix Weninger, Tobias Knaup, Björn Schuller, Congkai Sun, Kenji Sagae, and Louis-Philippe Morency. Youtube movie reviews: Sentiment analysis in an audio-visual context. *Intelligent Systems, IEEE*, 28:46–53, 05 2013.

-
- [36] Yi-Hsuan Yang and Homer H. Chen. *Music Emotion Recognition*. CRC Press, Inc., USA, 1st edition, 2011.