POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering

Master's Degree Thesis

EMOTION-BASED MULTIMODAL MUSIC CLASSIFIER FOR RECOMMENDER SYSTEMS

Advisors

Prof. Alessandro Aliberti
Prof. Natalie Parde, UIC

Candidate

Eleonora Quaranta
S316198

APRIL 2025

This thesis is dedicated to my niece, Pénélope. Your only limits are those you set for yourself, and they can be pushed too.

ACKNOWLEDGMENTS

While this work represents the closing chapter of my Master's Degree, it also stands as the culmination of an academic journey that began 19 years ago, shaped by curiosity, perseverance, and ambition. This achievement would not have been possible without the unwavering support of the people around me, to whom I owe my deepest gratitude.

First, I want to thank my advisors, Professor Natalie Parde from the University of Illinois at Chicago and Professor Alessandro Aliberti from the Polytechnic of Turin, for their paramount support and patience throughout this year-long research. Their guidance not only shaped this work, but also granted me the freedom to explore a field I am deeply passionate about. Additionally, I would like to extend my gratitude to the defense committee members, Professor Nikita Soni and Professor Paolo Garza, for dedicating their time and expertise to evaluate my work. Finally, thank you to Jenna Stephens for her invaluable support before, during, and after my permanence in Chicago.

To my parents, whose support and sacrifices have paved the way for every opportunity I ever had, thank you for letting me pursue every challenge I set my mind to and for being a great source of strength and encouragement, even from the other side of the world. Thank you to my brother, Alessandro, for always being by my side in every step of my life, big or small, constantly believing in me, and inspiring me every day with your kindness, strength, and support.

ACKNOWLEDGMENTS (continued)

To all my friends, who have been walking this path alongside me: thank you for filling all these years with laughter, understanding, and just the right amount of complaints. Knowing I could always count on you, whether for interminable study sessions, a shared moment of frustration, or to celebrate our victories, has made this journey even more meaningful. Thank you for believing in me even when I couldn't, for lifting me up in moments of doubt, for pushing me to chase my dreams, and for being the greatest gift these years have given me. To Veronica, Virginia, and Alice, thank you for these past eleven years; through highs and lows, you have always been by my side with patience, love, and impeccable humor. To Sara, thank you for having been my constant companion throughout these five years and for standing by me through every challenge and opportunity. I am sure 19-year-old us would be proud to see how far we've come and all that we've achieved - I know I am.

A special mention goes out to the people who have been my home-away-from-home in Chicago, my favorite *pseudo-sconosciuti*: thank you for being an incredible source of inspiration. I am beyond proud to have shared this journey with you, each and every single one of you left me with invaluable memories and lessons I will cherish forever.

Finally, a heartfelt thank you to Samuele, whose patience, support, and unwavering belief in me have been a guiding force throughout this process.

As this chapter closes, I know every ending simply is the beginning of a new adventure. I can now look forward to what comes next, knowing that the support of those around me and the lessons I have learned will continue to guide me.

EQ

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

FIGURE **PAGE**

LIST OF FIGURES (continued)

FIGURE **PAGE**

LIST OF ABBREVIATIONS

AI	artificial intelligence
BLF	block-level features
BOW	bag of words
CNN	convolutional neural network
CRNN	convolutional recurrent neural network
DL	deep learning
FFT	fast Fourier transform
LLM	large language model
LoRA	low-rank adaptation
LSTM	long short-term memory
MER	music emotion recognition
MEVD	music emotion variation detection
MIR	music information retrieval
ML	machine learning
MLP	multi-layer perceptron
MSE	mean square error
NLP	natural language processing

LIST OF ABBREVIATIONS (continued)

PCA	principal component analysis
PEFT	parameter-efficient fine-tuning
QLoRA	quantized low-rank adaptation
STFT	short time Fourier transform
t-SNE	t-distributed stochastic neighbor embedding
TF-IDF	term frequency-inverse document frequency
TL	transfer learning
UMAP	uniform manifold approximation and projection

SUMMARY

The emotional role of music has been widely studied over the years, highlighting its crucial role as an effective emotional regulator. The rising popularity of streaming platforms, along with the recent advancements in the artificial intelligence field, led to an increasing necessity for personalized content consumption experiences. Despite the strong performances of current recommendation algorithms, capable of modeling users' preferences from different perspectives ranging from content preferences and listening habits, the strong reliance on collaborative filtering methods makes them subject to issues such as the cold start problem caused by the lack of historical data for new users or new content.

This work aims to create a multimodal music classifier capable of inferring emotional qualities from both lyrics and audio of songs. In order to achieve this, it is necessary to tackle the data scarcity issue characterizing the music emotion recognition field: because of copyright limitations on song data, a standardized song dataset, including integral lyrics and acoustic content, is not currently publicly available. The solution to this issue adopted in this thesis consists of the use of the Music4All-Onion dataset, a vast collection of sub-datasets containing pre-processed lyrics and already-extracted audio features describing both the overall properties of each track and section-specific characteristics, allowing an acoustic analysis at different granularity levels.

After an unsuccessful attempt to infer emotional qualities from the anonymized users' tags available with the Music4All-Onion dataset, emotional labels for the lyrics modality are ob-

tained with transfer learning: the Edmonds Dance music dataset, a small dataset of song lyrics and corresponding emotions, is used as a source dataset to train a learning model to predict the emotional labels on the lyrics layer of the Music4All-Onion collection. Plutchik's wheel of emotions inspires the emotional model chosen for this task, consisting of eight emotional classes: joy, trust, surprise, anticipation, anger, fear, disgust, and sadness. The choice of this emotional model stems from the need for a more accurate feeling representation than those consisting of four classes usually found in literature.

As for what concerns the audio modality, the most relevant features in terms of emotional information are selected leveraging existing literature on the topic, and different dimensionality reduction methods are tested and compared to tackle the curse of dimensionality issues arising from the use of large datasets. Once a two-dimensional representation of acoustic data is obtained through the UMAP technique, unsupervised machine learning methods are used to obtain a meaningful clustering structure, separating the samples into seven final clusters - excluding the eighth group of outliers - based on their acoustic properties. Subsequently, interpretability studies are conducted to find a correlation between the original audio features and the obtained cluster indexes to identify meaningful acoustic properties of each class that can potentially be associated with emotional states. Additionally, supervised classifiers are trained to learn a mapping between the original audio features and the final class indexes to ensure the reproducibility of the clustering results independently of the dimensionality reduction technique applied.

The final multimodal classifier combines a multi-layer perceptron with convolutional and recurrent elements to achieve the classification pipeline: the multi-layer perceptron is used on audio data to predict the cluster index of each sample, whereas three parallel convolutional and recurrent block process the textual data to extract meaningful patterns and context; finally, the cluster indexes are combined with the output of the convolutional-recurrent layers to serve as an auxiliary feature for the definitive emotional classification.

Although the final classifier does not outperform the baseline techniques identified and tested on textual data, it provides valuable insights into the role of lyrics and acoustic information in the emotional classification task. Furthermore, this work contributes to exploring an underresearched field with promising applications in numerous domains.

CHAPTER 1

INTRODUCTION

Music plays a crucial role in many people's lives, complementing everyday experiences, providing comfort, or shaping the tone of both social and personal occasions. Researchers in music psychology have been exploring the social and emotional function of music for years, underlining the valence of music not only as a facilitator of social interactions or identity expression, but also as an effective mood regulator [1][2][3].

Driven by the rise of streaming platforms and digital distribution, the technological advancements of the past years have made music even more accessible, allowing users to explore an unprecedented number and variety of tracks, with suggestions tailored to their preferences. According to the 2024 year-end report by Luminate[1], the global number of audio streams recorded in 2024 increased by 14% compared to the previous year, reaching a total of 4.8 trillion streams and highlighting the expanding influence of music on daily life.

The rise of music streaming services shaped a new way of consuming such content, acting as a catalyzer for the transition from traditional music listening, in which a limited set of tracks is passively consumed, to a new interactive paradigm in which users can engage with a vast collection of tracks, usually guided by curated playlists and suggestions from dedicated algorithms.

---

[1]https://luminatedata.com/

Intrinsically, this culminates in the need for sophisticated recommendation algorithms capable of capturing the nuances of songs from different perspectives, ranging from audio features such as melody and rhythm to contextual information such as listening habits, social influences, and, naturally, emotional states. Achieving this level of personalization requires complex models able to process and model both song data and metadata and user preferences; machine learning and deep learning approaches, combined with data integration from various sources, are proven to be effective and efficient tools capable of providing tailored recommendations and enhancing user engagement.

Machine learning models use different types of data as predictive features, including songs' intrinsic characteristics and user-based data, including the listening and search history, created playlists, and other behavioral data, such as skipped songs, replay frequency, and session duration [4]. Other contextual factors, such as time of the day and device, are leveraged to model listening habits and dynamically adapt the recommendations to user behavior changes.

Music recommendation has relied for years on the collaborative filtering paradigm, which uses the aforementioned data to model each user's taste profile, which is, in turn, used to suggest songs liked by other users with a matching profile. Despite the undeniable success of this model, recommendations based exclusively on it suffer from several limitations. One major issue that has been widely explored is the cold-start problem, which can be essentially described as the challenge faced when user data is not yet available, either because a new user joins the platform or because a new artist, album, or song is added to the music catalog, resulting in the lack of interaction data usable to perform reliable recommendations. While the first

scenario is destined to self-resolve once the user starts interacting with the platform, the second one presents some additional criticalities, especially for lesser-known artists, whose music may struggle to gain visibility with insufficient user interaction, but user interaction is destined to remain low if content is not recommended. This self-reinforcing loop increases the risk that the collaborative filtering recommender acts as an echo chamber prioritizing mainstream content over emerging content that is, therefore, destined to remain underrepresented.

The recent years' migration toward a more content-oriented recommendation strategy addresses this issue by incorporating more data relative to the intrinsic properties of songs into the model, since they can be retrieved regardless of the presence of interaction data. While an exclusively content-based paradigm would fail to capture relevant information provided by cross-referencing different users' preferences, a hybrid approach can benefit from the pros of both techniques and better handle their drawbacks, further improving recommendation quality and pertinence.

The motivation behind this thesis stems from the need to further enhance recommendation strategies by incorporating emotional attributes embedded in songs along with the historical data already in use. Leveraging music's multimodal nature, both audio and textual data can be analyzed to extract the emotional content that human listeners may be responsive to, ensuring that content suggestions align not only with their past preferences and behavior but also with their inferred emotional state at a given time.

The primary context in which this work is positioned is, thus, the Music Emotion Recognition (MER) one, a research area of remarkable interest aimed at extracting emotional in-

formation from musical data. Because of the crucial role emotions play in music and in the way users engage with it, developing techniques capable of inferring emotional properties based exclusively on the content of songs, both audio and textual, is essential for enhancing user experience. Despite the importance of this task, its execution is inherently complicated because of the deep subjectivity of emotional perception, which can vary across different individuals due to their personal experiences and backgrounds. For this reason, it is crucial to outline unbiased components of music that can serve as reliable predictors of emotional content.

This work ultimately focuses on finding and leveraging those components, training machine learning models to infer emotional properties in a systematic and consistent way, hoping to contribute to narrowing the gap between machine learning and human music perception. While the ideal use for this model is its employment in generating content suggestions, integrating the two frameworks to achieve an emotionally aware recommendation system is left to future works.

The remaining content of the thesis is structured as follows: Section 2 provides an overview of the existing literature on music emotion recognition, multimodal learning and some hints on recommender systems. Section 3 is dedicated to exploring the available datasets in the context of music classification and providing an overview of emotional models used for similar tasks. Section 4 describes the labeling experiments performed on lyrics data to obtain the final emotional labels used for text classification. Section 5 presents the baselines models identified as benchmarks for this work, whereas Section 6 highlights the approaches that will be implemented to tackle the music emotion recognition task, with the corresponding results being presented

and discussed in Section 7. Finally, Section 8 summarizes the findings, draws conclusions from the collected data, and outlines future research areas.

CHAPTER 2

RELATED WORK

This section provides an overview of past and current literature on the primary topics discussed in this thesis, along with minor references to the broader domain of recommendation to highlight additional key concepts and provide the necessary context.

2.1 Music Emotion Recognition

The core of this research is music emotion recognition (MER), a multidisciplinary domain of growing interest in recent years due to its many potential applications spanning from recommendation systems to automatic music generation. The increasing attention given to the MER task is closely linked to the remarkable developments in the artificial intelligence realm observed over the last few years, with deep learning models offering novel methods to analyze and extract information from different complex data formats.

The importance of the emotional content encompassed by songs had already been theorized before the 20^{th} century, but contemporary studies confirmed and reinforced the idea of music as a conveyor of emotions [5][6].

The need for automatic music emotion recognition originates from the necessity to organize music into categories to improve users' experience in related applications, which is the same principle at the base of music genre recognition. Investigating the exact connections between musical features and emotional values is a challenging component of MER because of both the

subjectivity of feelings and the remarkable requirements in terms of labor and time necessary to manually explore the co-dependencies between them [7]. Automatic music emotion recognition has the potential to address or mitigate these issues by introducing models that are able to extract such information directly from the musical source.

Two main approaches can be outlined based on the level of granularity of the analysis: song-level MER aims to identify the overall emotional content of music, expressible through either a single label or multiple labels, whereas music emotion variation detection (MEVD) focuses on detecting the variations in the emotional tone of songs, highlighting the dynamism of feelings encompassed over different sections. Despite the undeniable value of section-specific emotional information, only traditional MER approaches are discussed in this thesis; this decision is dictated by the requirements of variation-based strategies, for which multiple sections of each song, each encompassing a single stable emotion, are essential [8]. As will be further discussed in Section 3, the data scarcity problem already poses significant challenges in the research of music datasets with song-level emotional labels; the additional requirement of segmented song datasets containing section-specific labels would further increase the already numerous limitations of this research.

The following section contains a brief overview of datasets tailored for the MER task, followed by an outline of the principal methodologies currently employed to extract emotional information from music audio and lyrics, and of how multimodality is achieved.

2.1.1 MER-Specific Datasets

The authors of [7] and [9] provide a comprehensive overview of datasets typically used for MER or MEVD tasks, highlighting the differences in emotional models used for the labels, data format, dimensions, years, and contained genres.

Although the initial focus of MER research was in the audio domain, the possibility of leveraging different data sources led to significant advancements in the research field, specifically when joined with traditional audio features to achieve multimodality. Among these, lyrics and biological features are those more often used [7]; the distinctive trait of biological features in this field is the fact that they are collected from listeners, rather than from the songs themselves. Specifically, some of the most used biological features include information on electroencephalograms (EEG), body temperature, heart rate, and functional magnetic resonance imaging (fMRI), thus embedding the response generated in listeners by music instead of intrinsic musical features. Other notable modalities that have become increasingly popular in multimodal music emotion recognition are visual images extracted from music videos or album covers.

As explained in [10], an issue many researchers have to face concerns the availability of data; this reason leads to many private datasets and labels being used for the task, which causes additional complications when trying to establish baselines or common practices. More details on this crucial topic are given in Section 3, which presents an in-depth discussion of the specific dataset requirements for this research, an overview of various emotional models

documented in the relevant literature, as well as an outline of the design choices made to achieve the objectives of this thesis.

2.1.2 Lyrics Classification

The task of lyrics classification is, in essence, superimposable to the broader domain of text classification, an element of remarkable interest in NLP research. Applications cover multiple fields, including news classification, information retrieval, topic modeling, and sentiment analysis, proving the paramount importance of methods able to provide reliable, objective, and, above all, automated solutions.

The increasing availability of digital textual data has driven the development of numerous techniques aimed at investigating the semantic meaning of texts, capturing the language nuances and layered meanings characterizing human language. Key issues in this field concern the necessity for models able to capture not only the semantic meaning of single words in a text but also the global context, considering that human language evolved to embed hidden secondary meanings in words, depending on their position in a sentence and on the tone used, which is particularly challenging to discern from textual cues only.

To tackle such a challenging task, multiple techniques have been developed to cover the whole classification pipeline, from early preprocessing stages to the final classification phase, of which this section provides an overview.

2.1.2.1 Text Preprocessing and Representation

The first step addresses the need to transform raw data into a format suitable to be analyzed by computational models.

First, the text needs to be discretized to the desired level of granularity using tokenization[1] [11]. A common practice then consists of cleaning the obtained tokens to eliminate superfluous words, symbols, and other unnecessary elements that may negatively affect the final performance or introduce an excessive computational overhead. Stopwords[2] and noise are often removed in this step, while tokens are also lowercased and, in the case of abbreviations or contractions, standardized to reduce the total number of unique tokens to account for. Despite the widespread use of these practices, it is important to evaluate if the specific task tackled can benefit from operations such as stopword removal, which can profoundly change the overall meaning of a text, and standardization of tokens, which could lead similarly-spelled words having different meanings to be considered as equal [12].

After the necessary preprocessing steps are performed, it is possible to select a suitable representation format to convert the obtained tokens into vectors which can be projected into the feature space. The ultimate goal of this procedure is to transform the literal tokens into numerical representations that preserve the semantic and syntactic properties of the text and make them processable by learning models.

Different techniques of varying complexity have been introduced over the years; among these, the bag-of-words representation (BOW) is the most straightforward and intuitive method,

---

[1]A token is the atomic part of the text; usually, a token corresponds to a word or to a character n-gram.

[2]Words that often appear in texts without carrying significant semantic information, such as conjunctions, articles, or prepositions.

mapping the text using word frequency vectors [13]. Essentially, a vocabulary in which each word is associated with its number of occurrences in the text is created, and this is the only information stored; as a consequence, the overall context of the text may be lost[1]. Another criticality of this method is the potential size issue, as the cardinality of the vocabulary may reach remarkable dimensions, especially with longer texts. To address this, it is thus necessary to introduce a metric to measure the importance of words in documents; TF-IDF (Term Frequency-Inverse Document Frequency) is commonly used for this purpose as it allows to compute the significance of a word in a document in the context of a collection of documents. TF-IDF is computed as the product between two distinct metrics, TF and IDF, defined respectively in Equation 2.1 and Equation 2.2, where $f_{t,d}$ represents the number of occurrences of the token t in the document d, and N is the cardinality of the total collection of documents D.

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2.1}$$

$$IDF(t,D) = log \frac{N}{1 + |d : \{d \in D \ and \ t \in d\}|} \tag{2.2}$$

TF simply represents the frequency of a term t in a document d, whereas IDF can be seen as a measure of how rare a term t is in the collection, and therefore of how informative it is[2]. TF-IDF

---

[1]Sentences like *People like dogs* and *Dogs like people* have an identical BOW representation.

[2]Words with a high relative frequency in a document but present in many documents are not indicative of the document's content.

is, therefore, higher for words representative of a few documents, highlighting their importance, and lower for terms present in most of the documents, such as articles, conjunctions, or other less meaningful words.

An alternative to the bag-of-words approach addressing its order-independency issue is language modeling [14], implemented, for instance, through the use of *n-grams*[1] [15][16]. Language models enable feeding sequential information to models by considering both the order and dependencies of elements of a sequence. The base idea behind n-grams spans from the need to compute the probability of a specific item appearing considering the pre-existing history so far. The history preceding the item can, however, be significantly large and pose computational problems, and therefore language modeling through n-grams stems from the assumption that the most recent $n-1$ items are sufficient to approximate history, leveraging the Markov assumption. In practice, the base assumption in 4-gram language models is that the probability of an item occurring depends on the previous three.

Despite the increased contextual awareness provided by language modeling, the mentioned strategies still lack the ability to encode semantic information; this can be challenging when trying to project words having similar meanings but different spellings, such as synonyms: if no semantic information is available, the vectors obtained from the two distinct words are not going to reflect their similarities, causing a decrease in model's performances. Word embeddings have been introduced to tackle the semantic challenge just discussed [12]. One of the first word

---

[1]Sequences of items of size n; they can be constituted of words, characters, or any other atomic item in a language.

embedding models introduced, Word2Vec [17], aims at creating semantically accurate mappings of words of a corpus through their projection into highly dimensional vectors, leveraging shallow neural networks with different variations: a continuous bag-of-words (CBOW) and a continuous skip-gram. These variations consists respectively in the prediction of a target word given the context around it and vice-versa, in the prediction of the surrounding context given a word.

An alternative to the Word2Vec predictive model is GloVe [18], which gained increasing popularity in recent text classification pipelines [19]. Unlike Word2Vec, GloVe can be trained on remarkably large corpora of data, leveraging word-to-word co-occurrence matrix factorization to condense global information about the corpus instead of only focusing on the context of single words. Thanks to the computational efficiency of GloVe, achieved through dimensionality reduction of the co-occurrence matrix, the authors were able to train the model on five large corpora of texts taken from Wikipedia, Gigaword, and Common Crawl [18], providing embeddings able to capture a wider range of relationships between words.

Despite the notable improvements introduced by Word2Vec and GloVe embeddings, there is still an open issue: neither model can successfully be used to project terms that were not seen during training. The authors of [20] introduced a novel word embedding able to tackle this challenge: fastText. While Word2Vec and GloVe only considered whole words, fastText represents each term both as the term itself and as a bag of character n-grams representing the different *subwords* contained in it. In practice, fastText is trained similarly to Word2Vec, but

the whole process incorporates information on subwords as well[1]; this strategy allows to better handle unseen words at test time by splitting them into subwords as well and leveraging the relevant information acquired during training.

2.1.2.2 Unsupervised Methods

Some techniques have been leveraged to identify structures or patterns in an unsupervised scenario where the labels of the textual data are not provided. One task related to this domain is topic modeling, which is the identification of underlying topics in corpora of texts; Latent Dirichlet Allocation (LDA) was used in [21] for this purpose:the authors leverage a Bayesian Network to extract a Bayesian topic model able to analyze the content of a document collection and define the different topics which can be extracted from it. Once the topics are determined, it is possible to assign a relevance score for each document and for each topic, indicating how closely related the document is to the topic. LDA is based on a probabilistic approach in which each document is modeled as a random mixture over latent topics, and each topic as a distribution over terms [22].

Other unsupervised methods belong to the clustering family and can also be leveraged to identify underlying relationships and patterns in song collections. [23] proposes a clustering procedure as a complementary step to the classification one, performing similarity clustering on a whole collection of documents before training a classifier to improve its performance.

---

[1]e.g. the word *where* is represented by [wh, whe, her, ere, er] and [where].

2.1.2.3 Supervised Methods

In the domain of supervised learning, many methods have been used in literature with the purpose of achieving good performance metrics in the text classification task and, subsequently, in song lyrics classification.

Traditional machine learning methods have been used in [21], [24], [25], and [26], including techniques such as Support Vector Machines (SVM), random forest classifiers, naive Bayes classifiers, and logistic regression for continuous emotion models[1]. Despite the advancements in the field introduced by such techniques, their adoption has decreased over the last years in favor of more modern deep learning methodologies, which have proven superior capabilities to handle pattern extraction in complex data.

The use of deep learning in lyrics classification involves numerous different architectures and methodologies. The authors of [19] present three different models: a Convolutional Neural Network (CNN), a bi-directional Long Short-Term Memory (bi-LSTM), and a Convolutional Recurrent Neural Network (CRNN) combining convolutional layers with an LSTM. The input text fed to the model is transformed using GloVe embedding, and the authors report comparable performance for the three architectures in terms of accuracy, finally favoring CNNs for the shorter training time required. This paper highlights an interesting note: it appears that the sequentiality of lyrics is a secondary element in classification, since the use of models able

[1]The difference between continuous and discrete emotional models will be discussed in Section 3.1.2

to better leverage this property, namely the LSTM and CRNN, did not lead to performance improvements.

A language model often used in this context is BERT, Bi-directional Encoder Representation from Transformers [27]: the authors of [28] use a BERT model fine-tuned on social media data, but report a scarce generalization ability on song lyrics. On the other hand, the authors of [24] achieve remarkable classification results when using BERT as a transfer learning model to label a song lyrics dataset. More details on this will be given in Section 4.2.3.

Large Language Models (LLMs) introduced significant changes in the NLP domain by remarkably advancing contextual awareness and semantic representations. Despite their widespread diffusion, most of the literature on song lyrics classification does not mention them at the moment, probably due to how quickly the LLM field is evolving. Further discussion on this topic will be provided in Section 4.2.4.

2.1.3 Audio Classification

The task of audio classification in the music domain has been approached in different ways, ranging from genre classification to instrument classification, including the music emotion recognition problem discussed in this work. Music audio classification, as a sub-domain of the broader audio classification field, has gained increasing interest over the past years because of the numerous and diverse possible applications across a range of scenarios, including speech recognition, sound event detection, and environmental sound classification.

2.1.3.1 Audio Preprocessing and Feature Extraction

The first challenge in any audio classification pipeline is the preprocessing of tracks, which is necessary to convert the data into meaningful formats that learning models can employ. The elementary representation of audio data is the audio wave in the time domain, consisting of a two-dimensional plot having time on the x-axis and amplitude on the y-axis, with $y = 0$ usually representing silence. A common practice is to translate this representation into a more suitable one, representing audio through the frequency domain instead of the time one [29]. The Fast Fourier Transform (FFT) allows the conversion of a signal from the elementary amplitude-time representation to its individual spectral components, thus obtaining the desired amplitude-frequency representation of the track. A variation of the FFT is the Short-Time Fourier Transform (STFT), mitigating the natural loss of information on the time domain caused by the FFT. The STFT works by computing multiple Fourier transforms over smaller segments of the signal at different time stamps to capture both frequency and time information. Applying the STFT leads to a visual representation of the frequency content of the audio track called spectrogram.

Spectrograms have been employed in numerous works as input data for learning models performing the audio classification task [30][31][32][33], thanks to their compact representation of information and compatibility with deep learning models employing convolutional layers.

Mel-Frequency Cepstral Coefficients (MFCCs) are another popular representation format for audio data [34]; they are used to represent the short-term energy spectrum of signals, adjusting

it to a frequency scale closer to the one naturally characterizing human hearing[1]. Despite its popularity in speech recognition tasks [35][10], some studies employed them in the context of MER with other complementary features [31][32].

In general, a large number of features can be extracted from audio signals, and it is possible to group them into three distinct groups for simplifying purposes: low-level features, rhythmic features, and tonal features [36], each representing different properties. Further details on this topic are provided in Section 3.2.1.

Different novel features for audio classification have been introduced over the last years, proposing feature extraction methods leveraging different properties of audio tracks; [35] introduces a set of novel features explicitly tailored for music emotion recognition to capture high-level characteristics of the signal. [37] extends this work by providing an overview of emotionally relevant musical features leveraging the eight fundamental musical dimensions: melody, harmony, rhythm, dynamics, tone color, expressivity, texture, and form. As will be discussed in detail in Section 3.2.1.1, the authors of [38] introduced a novel system of *block-level features*, obtained by processing audio signals on a block level rather than a frame one to capture more temporal information.

---

[1]Humans do not perceive all frequencies equally, the sensitivity of the human ear is different across the frequency spectrum. The Mel scale is used to approximate human perception on various representations, including spectrograms.

2.1.4 Unsupervised Methods

The most popular unsupervised method applied to audio data for music emotion recognition is clustering, as performed in [10][30][31][32][39]. The higher popularity of clustering techniques in MER tasks on audio data when compared to lyrics data may be due to a more significant lack of emotional labels in music, underlining the challenges of extracting emotionally loaded information from audio tracks. This explanation is supported by the fact that clustering is often leveraged as an auxiliary technique paired with supervised methods like those described in Section 2.1.5, as performed in [30] and [31]; in this way, it is possible to leverage clustering as a self-supervised method to obtain pseudo-labels that can then be used by supervised algorithms, as done in [30], or as a way to identify emotionally representative key frames from spectrograms, as done in [31].

2.1.5 Supervised Methods

The supervised methods used in literature for audio classification do not differ significantly from those presented in Section 2.1.2.3 for the textual data case. In this general domain, CNNs have been proven efficient in the identification of patterns in spectrograms [40] thanks to their ability to extract meaningful information from images.

In the context of music emotion recognition, supervised methods reported in the literature include base machine learning algorithms such as support vector machines or random forest classifiers [19][41], but these methodologies have been replaced by more complex deep learning architectures. As mentioned in the previous section, [31] leverages convolutional neural networks in addition to clustering to recognize emotions from the key spectrogram frames identified. [7]

reports the use of convolutional and recurrent deep learning architectures, respectively for the MER and MEVD task[1]. VGGNet [42], a simple but deep convolutional neural network, has also been proven to improve the classification performance on music emotion recognition tasks [43].

2.1.6 Achieving Multimodality

Joining multiple sources of data together in music classification has been proved to better the quality of the performance, especially when multimodality is achieved through the use of audio and lyrics [7].

In the context of multimodal classification, one of the design choices that need to be made concerns the way in which different information modalities are combined. Two main tendencies can be outlined: using separate classifiers on the different modalities and then using a voting scheme to output the final prediction or using a single classifier on data extracted from the separate modalities. In [44], the chosen classification algorithm is used on both lyrics and audio features according to the different fusion methods: feature-level fusion, consisting of concatenating text and audio features of a sample in a single composite feature, and model-level fusion, using a different classifier for each modality and then employing a weighted voting scheme to achieve the final prediction. The results indicate that feature-level fusion yields slightly

[1]The reason why recurrent networks are preferred for the music emotion variation detection task lies in the necessity for sequential information to be processed in order to identify segments in which the emotional content is varying.

better results. The authors of [41] compared the same two approaches on their multimodal music classifier and report similar results.

2.2 Recommendation Systems Fundamentals

Because of the motivations of this work, some final concepts that are worth mentioning concern recommendation systems. This domain is remarkably large and it is outside the scope of this thesis to delve into the details of it, therefore only some essential ideas are presented, and, to further simplify things, the discussion is limited to recommendation systems in the music domain.

It is possible to identify two main types of recommendation systems based on the information leveraged to obtain recommendations: content-based systems, which recommend songs to a user based on how similar they are to other songs the same user has enjoyed, and systems based on collaborative filtering, in which the user's preferences are modeled and compared to those of other users to recommend songs that people with similar taste profiles have enjoyed. Most recommendation systems that are currently used by content platforms are mainly based on collaborative filtering, leveraging the large number of users of those platforms, but, as already mentioned in Section 1, this is not always the optimal solution: the cold start problem faced by new artists or new songs, for which no user data is available, deeply disadvantages them. Though current recommendation algorithms consist of a hybrid solution taking into account content information as well, this is usually limited to information on the pure sound of music, representing some key characteristics that are not always sufficient to describe the song as a

whole from the perspective of listeners[1], and cannot thus be used as foundational descriptors for recommendation.

Furthermore, the benefits of an emotion-based recommendation model based on song content are presented in [45], introducing a Personality and Emotion Integrated Attentive model (PEIA) to tackle the issue of music recommendation on social media. In this work, information on the users' music-listening tendencies is joined with emotion-related features[2] to model their short-term preference factors, and with personality-related features[3] to achieve long-term taste profiles. This model reportedly outperforms state-of-the-art methods by achieving a normalized discounted cumulative gain (NDCG)[4] of 0.53, proving the crucial role of emotion-related features in the recommendation domain. It is however important to highlight how this work and its results were made possible by the use of a very large collection of diverse user data, the WeChat dataset, containing all of the information necessary to perform the above-mentioned operations, which is not been made publicly available.

This research builds upon the existing literature on Music Emotion Recognition by integrating information concerning both modalities of songs: lyrics and audio. A key feature of this study is the strategy employed to address the data scarcity problem: instead of relying on exist-

---

[1]Some examples of popular features are danceability, tempo, key, energy.

[2]Time of the day, day of the week, emotions extracted from users posts.

[3]Demographics, social behavior, interests.

[4]Items ranking quality metric.

ing labeled MER datasets, the use of a large unlabeled dataset is favored. The tools employed to address the lack of reliable tags vary across the different modalities: while transfer learning is used for the textual modality thanks to the availability of suitable source datasets, labeling of the audio modality leverages unsupervised learning techniques to model acoustic properties of songs and identify meaningful patterns. Because of the nature of the available data, this thesis proposes a new approach in which the textual modality plays the most significant role in the emotional classification task. The audio information available only serves as an auxiliary indicator of the general acoustic properties of songs. Additionally, the most significant literature baselines are re-implemented and tested on the newly obtained labeled textual dataset in order to obtain meaningful evaluation benchmarks that act as guiding resources for the design choices concerning the final deep classifier.

This chapter presented an overview of the state-of-the-art approaches that have been developed to tackle the MER task, highlighting the challenges emerging from the lack of unified datasets specifically tailored for it, an issue that will be further discussed in Section 3. The literature review has covered both lyrics-only and audio-only approaches that will serve as the foundation for the baseline definitions in Section 5, also underlining the importance of preprocessing operations and feature extraction; notions on the different ways in which multi-modality has been achieved have also been discussed. Moreover, a brief overview of current recommendation strategies has been delivered to provide additional context to the motivation behind this research. Finally, a brief overview of how this work is positioned in the current literature has been provided.

CHAPTER 3

DATASETS

3.1 The Data Problem

3.1.1 Data Availability

One of the main problems continuously encountered during the research and set the first limitations to this thesis is the data availability issue, as briefly discussed in Section 2.1.

Despite the broad investigation of MER tasks in recent years, a complete and satisfactory dataset is still to be made freely available to researchers. The lack of a dataset that allows to profoundly investigate the emotional side of music listening has multiple causes, embedded in both the commercial value of music itself and, in general, in the data commodification[1] phenomenon spreading throughout various fields.

The data availability problem translates practically into the inability to work with the complete music data because of copyright limitations [9]. In the context of this thesis, this is a twofold issue, as it concerns both the audio tracks and the lyrics of the songs. The only viable solution to this matter is settling with datasets of already preprocessed music tracks, where audio and lyrical features have been extracted from the original songs and partially elaborated in order to prevent copyright infringements.

---

[1]Transformation of data into a marketable good that can be sold, bought and exchanged for an assigned economic value.

It is of paramount importance to understand the complications that arose from this solution: a crucial aspect of any application leveraging artificial intelligence concerns processing operations on data. Being able to experiment with raw data not only allows a broader range of information to be incorporated into the model itself, but also enables experimenting with a more extensive set of models, architectures, and techniques. This consideration is specifically valid in the field of deep learning, where neural networks have been proven to be valid feature extractors for tasks that involve audio data [46].

Despite the above-mentioned disadvantages of preprocessed data, relaxing the requirements in terms of data format and completeness allows for a larger pool of datasets to choose from, focusing on other important requirements that should be satisfied.

For the multimodal MER task of interest, some form of emotional information on each track is also needed, in addition to audio and lyrics data. Such information is usually based on the emotional perception of listeners; this is true not only in the MER field but in any task involving affective information, due to both the subjectivity of emotions and the need for contextual understanding, as explained in [47]: different terms may evoke different emotions based on the context in which they are inserted. Despite recent developments in deep natural language processing and large language models, there are some nuances in languages that are still only perceivable by humans [48]. For the aforementioned reasons, the extraction of emotionally meaningful tags requires the availability of users' data or some other kind of information related to human perception, such as secondary metadata obtained through raw data processing. The

problem of emotion modeling is another crucial aspect of the preliminary phase of the research, and its details will be discussed in Section 3.1.2

Embedded in user data and the associated extracted metadata lies a potential issue: since organizations collecting, analyzing, and storing such data typically invest notable resources in them, they may have a tendency to keep data private to protect their competitive advantage or economic interests. This explanation may justify, even partially, the limited availability of well-maintained public databases and the existing restrictions on the applications for which said data can be used.

The goal is to work on a large enough dataset to train ML and DL models to acquire good generalization ability while maintaining decent data quality and reliability. Despite the crucial role of data quality in creating any AI model, the considerations made for this research on the trade-off between quality and quantity led to favor quantity over quality. The reasoning behind such a decision, which is in contrast with the usual practice, focuses on the need to create a model operating in a setting as close as possible to the real-world scenario. This decision optimistically enables the switch to a better-quality dataset whenever available.

3.1.2 Modeling Emotions

The task of modeling emotions for data science applications is a particularly critical one. We, as humans, are used to perceiving a range of emotions that are easily recognizable to us. Translating something as complex as feelings into something a computer can discern is a challenge that cannot be tackled without considering cognitive and sociological points of view.

It is possible to distinguish two different emotional paradigms: dimensional and categorical. Dimensional models leverage spaces defined by two or more axis, often mapping different emotions onto two-dimensional planes, or three-dimensional spaces. Among these, the most popular are Russell's model and Thayer's one. Russell's emotion model [49] is the most often referred to in MER literature [21][24][41]. This configuration is based on a circumplex interpretation of emotions, which are represented on a two-dimensional plane in which the two Cartesian axes represent valence[1] and arousal[2]. This model, often referred to as the AV model, is generally used as a base for a four-quadrant understanding of emotions: two high-valence emotions, happiness (arousal > 0) and relaxation (arousal < 0), and two low-valence ones, anger (arousal > 0) and sadness (arousal < 0).

The two-dimensional model proposed by Thayer [50] instead only considers the first quadrant of a Cartesian plane in which the x-axis represents stress and the y-axis energy, thus leveraging exclusively two different types of activation. This produces a four-emotion model consisting of contentment (low stress and low energy), depression (high stress and low energy), exuberance (low stress and high energy), and anxiety (high stress and high energy) [51].

Russell and Mehrabian also proposed an extension of the AV model in which a third axis representing dominance[3] is added to the two-dimensional representation, leading to the VAD three-

---

[1]Pleasantness (+) or unpleasantness (-) of an emotion

[2]High energy (+) or low energy (-) of an emotion, in terms of intensity of activation

[3]Degree of control perceived: dominance, feeling of confidence and control (+) or submissiveness, total loss of control on events (-).

dimensional model [52]. This model allows the crucial distinction between emotions mapped to the same valence and arousal values, like anxiety and anger: anger corresponds to a high dominance, whereas anxiety, which involve a feeling of loss of control, to low dominance. This three-dimensional representation leads to a model in which the six fundamental emotions are joy, surprise, anger, disgust, fear and sadness. It is important to note that the four, or six, emotions resulting from the models just presented have a merely explanatory meaning, since dimensional models start from the assumption that emotions exists in a continuous space.

Discrete models, on the other hand, assume the presence of base emotions that exist in non-continuous space, and thus that it is possible to identify some base emotions which act as foundational blocks to "build" more complex ones. The simpler discrete model is the one introduced by Ekman in [53] and based on four basic emotions: happiness, anger, fear, and sadness. Hevner additionally explored the specific correlation between music and emotion in her work [54], using eight clusters of adjective to describe the different nuances of emotions that music can evoke in people. The most used discrete model in MIR tasks is the one proposed by the Music Information Retrieval Evaluation eXchange (MIREX) in the dataset[1] developed for their mood classification task, consisting in five clusters of emotional adjectives. However, this model does not seem to be supported by psychological evidence.

Another objective of this research is to leverage an emotional model which is both complex enough to better reflect moods in which a user can be and that can influence their musical

[1]Not publicly available.

Figure 1: Plutchik's wheel of emotions.

tendencies, and psychologically accurate. A model respecting both of these requirements is Plutchik's wheel of emotions [55], which indeed uses the structure of a wheel, as shown in Figure 1, to represent the eight core emotions of anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. As will be discussed in Section 4.2 and mentioned in Appendix A, this model is specifically suitable for the purpose of this thesis.

3.2 Music4All-Onion Dataset

The dataset that better fits the requirements previously discussed is Music4All-Onion, a multimodal music dataset containing lyrics, audio, video, and metadata features for over a

hundred thousand songs [56]. Moreover, this dataset also incorporates a large-scale set of listening records of real-world users, allowing for further investigation of listening habits and patterns, which are outside of the scope of this thesis but relevant for future works.

This dataset is an extension of the original Music4All database [57], developed specifically to tackle MIR tasks and mainly contains songs' metadata. The choice of the *Onion* version was made in favor of all the additional data on different modalities of each song, symbolizing the layers of the onion, which are crucial for the purpose of this thesis.

The whole Music4All-Onion dataset actually consists of several more specific sub-datasets divided by modality, feature extraction method, and other parameters. This division not only allows the possibility to efficiently isolate the data to work with, but is also a convenient way to reduce the dimensionality of the data that must be processed at a time.

A brief description of the sub-datasets that have been considered in this thesis is shown in Table I, as explained in the dataset documentation.

3.2.1 Feature Extraction

As mentioned in Section 3.1.1, copyright issues prevent using full audio tracks and complete lyrics and, therefore, force to work with a dataset of extracted features. This section presents the different extraction frameworks used by the authors of the Music4All-Onion dataset, and highlights the preprocessing operations performed on raw song data to obtain the resulting usable datasets presented in Table I.

TABLE I: SELECTED DATASETS FROM THE MUSIC4ALL-ONION CORPUS

File Name	**Content Description**	**Onion Layer**	**Features**
id_blf_correlation	Correlation Pattern BLF	Audio	1325
id_blf_spectral	Spectral Pattern BLF	Audio	980
id_blf_deltaspectral	Delta Spectral Pattern BLF	Audio	1372
id_blf_vardeltaspectral	Variance Delta Spectral Pattern BLF	Audio	1342
id_blf_spectralcontrast	Spectral Contrast Pattern BLF	Audio	800
id_blf_logfluc	Logarithmic Fluctuation Pattern BLF	Audio	3626
id_emobase_f0_stats	Statistical aggregation of emotion-related features extracted with OpenSMILE	Audio	76
id_emobase_lsp_stats	Statistical aggregation of emotion-related features extracted with OpenSMILE	Audio	304
id_emobase_mfcc_stats	Statistical aggregation of emotion-related features extracted with OpenSMILE	Audio	456
id_emobase_pcm_stats	Statistical aggregation of emotion-related features extracted with OpenSMILE	Audio	114
id_emobase_voice_stats	Statistical aggregation of emotion-related features extracted with OpenSMILE	Audio	38
id_essentia	Spectral, time-domain, rhythm, and tonal frame descriptors aggregated via mean and standard deviation, extracted with Essentia	Audio	1034
processed_lyrics	Song lyrics after preprocessing	EMD[a]	1
id_tags_dict	Tags and corresponding weights extracted via the Last.fm API	UGC[b]	1

[a] Embedded Metadata

[b] User-Generated Content

3.2.1.1 Block-Level Features

Block-Level audio Features (BLF) have been proven to lead to successful results in the field of music information retrieval, tackling problems varying from music genre classification [58] to automatic music tagging [59]. These tasks are parallel to the MER one tackled in this thesis, especially the latter explicitly addressing the cold start problem explained in Section 1.

As clearly explained in [59], the idea behind BLFs is the processing of audio signals based on a *block* segmentation rather than a *frame* one. The upside of this approach is the ability to better describe temporal information, leveraging the fact that each block encompasses multiple frames.

BLF extraction is performed on a frequency-time representation of audio signals, specifically the cent-scaled magnitude spectrum[1]. The spectrum is divided into blocks of n frames, with n being the window size. Depending on the hop size[2] chosen, the extracted blocks can overlap or not.

After the BLF extraction, a generalization process is applied to obtain a global vector representation for the examined track. The generalization process uses a summarization function to each dimension of the extracted feature vectors.

The datasets of block-level features listed in Table I capture different aspects of the audio tracks:

---

[1]Representation obtained after applying a STFT to the audio signal, computing the magnitude spectrum and mapping it onto the logarithmic Cent scale to account for its musical nature.

[2]Distance in time (frames) between the beginning of two consecutive blocks.

- Correlation pattern: harmonic and rhythmic relations between song parts.
- Spectral pattern: timbral content of the song.
- Delta spectral pattern: timbral content of the song, emphasizing onset strength.
- Variance delta spectral pattern: timbral content of the song, emphasizing variations in onset strength over the blocks.
- Spectral contrast pattern: *tone-ness* of the blocks of the song, intended as the difference between tonal components (peaks) and noise (valleys).
- Logarithmic fluctuation pattern: rhythmic layout of the song.

3.2.1.2 Emobase Features

The authors of the Music4All-Onion dataset reference [60] as the inspiration for the emotion-related feature datasets in the corpus. This paper, proposing a music recommendation model based on data collected from social media modeling users' long and short-term taste, proposes the *emobase* features extracted with OpenSMILE [61] that are then included in the Music4All-Onion dataset. The OpenSMILE toolkit allows the extraction of various data from audio tracks, capturing a wide range of information.

The sub-datasets in the second part of Table I are obtained as follows: after extracting the low-level descriptors for each song, aggregation functions and delta coefficients are computed to present a mathematically-detailed summary of audio properties of the tracks.

The low-level descriptors extracted as emotion-related features are:

- F0: fundamental frequency of song and its envelope[1]; related to perceived pitch.
- Linear spectral pair (LSP) frequencies: encoding of the spectral envelope through linear predictive coding (LPC[2]).
- Mel-frequency cepstral coefficients (MFCC): timbre representation through the track's short-term power spectrum.
- Pulse code modulation (PCM): intensity and loudness of the song modeled through energy levels and zero-crossing rate[3].
- Voice probability: probability of vocality in the song.

These features then undergo summarization operations through mathematical and statistical functions extracting their minimum and maximum values and corresponding positions, their mean, standard deviation, skewness, kurtosis, quartiles, inter-quartile ranges, and linear regression coefficients with associated errors.

3.2.1.3 Essentia Features

The final audio features extractor employed is Essentia [36], an open-source library tailored for MIR. This toolkit allows information to be extracted on three different levels, which are outlined below

---

[1]How the trajectory of F0 changes over time; outlines the dominant pitch perceived in the sound.

[2]Prediction of the current audio sample as a linear combination of past samples.

[3]Rate at which the signal changes from positive to negative and vice versa.

- Low-level features
  - Loudness
  - Dynamic complexity
  - Silence rate
  - Spectral root mean square, flux[1], roll-off, strongpeak
  - Spectral entropy, complexity, contrast coefficients, valleys
  - Spectral energy (overall) and energy per frequency band[2] with corresponding crest, flatness, and skewness
  - Zero-crossing rate
  - High-frequency content
  - Sensory dissonance[3]
  - Pitch salience
- Rhythmic features
  - Beats positions and counts
  - Beats-per-minute and corresponding histogram
  - Beats loudness, loudness band ratio

---

[1]How quickly the power spectrum changes

[2]Barkbands, Melbands, Erbbands

[3]*Roughness* of song as perceived by listeners.

  - Onset rate
  - Danceability score obtained through Detrended Fluctuation Analysis (DFA)[1]

- Tonal features
  - Tuning frequency
  - Harmonic pitch class profile (hpcp[2]) and transposed-hpcp
  - Key estimation, scale and strength
  - Chords strength, histogram, changing rate, key and scale
  - Tuning diatonic strength
  - Tuning equal-tempered deviation and non-tempered energy ratio

3.2.1.4 Preprocessed Lyrics

The dataset's authors report having preprocessed the song lyrics following NLP standards; first, superfluous white spaces, newline sequences and annotations are removed. Afterward, the whole text is lowercase and translated into English, where other languages are detected; finally, numbers and contractions are replaced with their spelled-out form, special characters and stopwords are removed, and lemmatization and stemming are applied.

Here follows a sample from the processed lyrics dataset:

---

[1]Finding a pattern among noisy data and assigning a *predictability* score to the track, with a higher score corresponding to higher danceability.

[2]Vector representing the intensities of the different semitone pitch classes

take shirt run slowli away find fine float decid lone heart live room mine tri make sound oh escap envi pride oh realli dead sin take oh matter time keep chang mind keep chang mind carri plastic bag throw side show lone oh mayb need man loos reduc infant depend wine oh matter time keep chang mind keep chang mind keep chang mind keep chang mind see clear get stuck come luck take shirt run slowli away find fine keep chang mind keep chang mind keep chang mind

It is notable how different this extract looks from properly understandable song lyrics, highlighting the criticality of not being able to work with the full song data. However, some hints on the general sense of the full text can still be extracted by focusing on the remaining words.

3.2.1.5 Users' Tag

The users' tag dataset is one of the crucial elements of the final dataset that this thesis aims to create. The tags have been extracted using the Last.fm API, and consist of a dictionary for each song, in which each key corresponds to a tag assigned by at least one user to that track, and the corresponding value is the weight associated with the tag, representing how reliable a tag is on the base of how many times it appears in association to that specific song.

One issue that can already be highlighted is that this collection of tags is not ready to use. For the MER task tackled in the thesis, the goal would be a dataset of emotion-related tags only, whereas in this case the tags are not filtered nor cleaned, and therefore their content and relevancy widely vary. The first part of the experimental section of this research is thus

dedicated to the tentative extraction of emotional tags that can be assigned to each track in the dataset, as will be discussed in Section 4.

3.2.2 Data Dimensionality

A note worth mentioning concerns the dimensionality of the datasets. As shown in the fourth column of Table I, the dimensionality of some datasets in the audio layer is extremely high. This poses a problem on multiple fronts: on the one hand, working with such high dimensional data requires a big computational power, which can be difficult to reach using current public tools, and on the other hand it also leads to the curse of dimensionality. This term is used to identify the issues arising from such high-dimensional data, such as the data sparsity problem, which increases the difficulty in identifying patterns and clusters in the data or the loss of meaning of distance metrics when applied to highly dimensional spaces.

For these reasons, great effort is put into trying to efficiently reduce the dimensionality of the bigger datasets, investigating multiple dimensionality reduction techniques whose details will be better discussed in the following chapters.

This chapter highlighted the key challenges faced in the early phases of this research, including data scarcity and copyright limitations, and provided an initial overview of the possible solutions to these constraints. An overview of emotional models used in the context of machine learning has also been provided, underlining the key characteristics of each one and the motivation for the selection of the final model for this research. Furthermore, the chosen dataset, Music4All-Onion, has been described detailing its layers and sub-datasets, along with the spe-

cific issues resulting from this choice, such as the availability of already-preprocessed data only and the dimensionality problem that will be tackled in the following sections.

CHAPTER 4

DATASET LABELING

The first practical issue tackled in this thesis is the aforementioned absence of emotional tags in the dataset, which are a fundamental aspect of any classification task. Tackling this issue required a detailed review of the solutions presented in literature, and the subsequent implementation of multiple failed techniques.

4.1 Labeling through Users' Tags

The first approach tested consisted of the tentative extraction of emotional labels from the users' tags dataset. To achieve this, multiple problems need to be taken into account: aside from the necessity of separating emotional tags from irrelevant ones, the fact that not every song has been tagged, specifically with emotional labels, needs to be considered. A detailed explanation of the operations carried out in this stage is available in Appendix A.

The evident criticalities of this method are, first of all, the lack of a proper validation strategy to check the reliability of the final dataset and, secondly, the fact that it relies solely on general opinions on the songs given by users. The issue that arises from the latter regards the fact that users were not clearly instructed to assign tags following any specific criteria, but the tags are simply the results of a personal annotation process, which cannot be automatically assumed as reliable. A representative example of this is the clear class imbalance of tags referring to love and appreciation, which is also discussed in Appendix A and visible in Figure 77, Appendix

A. This issue is due to the fact that users tagging a song with the words *love* or *like* may be a reference to their preferences for the songs, using, for example, the tag *I love this song*, and not an indicator of the fact that a song evokes *love* in the minds of listeners. Despite the possible mitigation strategies applicable to this issue, it is almost impossible to eliminate the uncertainty that may lead to misinterpretation of the tags and, therefore, to an unreliable labeling of the data.

The final issue with this procedure regards the obsolescence of the library used to extract emotional information from the general tags: WordNet-Affect [62], however groundbreaking its introduction was, has been deployed over twenty years ago. This does not necessarily mean that it is outdated as a tool, but only that better options may be available after this many years, especially considering the development of AI tools over the past few years.

4.2 Labeling through Transfer Learning

The data scarcity problem characterizes the whole ML and AI field, posing the first limitation to many applications. For this reason, methods to efficiently address this issue have been widely studied and employed. Among these methods, one of the most often used is transfer learning, whose basis was first introduced in the machine learning community in [63] and has since been widely used in numerous ML and DL applications.

Transfer learning allows to successfully tackle the scarcity of labeled data, as stated in [64], by leveraging the information learned while performing a task on an initial set of data, called the *source data*, to improve its performance on a different set of *target data*.

Inspired by [24], transfer learning is leveraged in this thesis as a labeling method for the Music4All-Onion lyrics dataset. The choice of focusing exclusively on the lyrics part is dictated by the necessity of a labeled source dataset to train the predictive model, with data in a format that is comparable to the one in the target dataset. This is possible for the lyrics dataset thanks to the Edmonds Dance dataset [65], a small lyrics dataset with emotional labels based on Plutchik's wheel of emotions [55]. This dataset includes the full lyrics of 524 songs and is particularly suitable for this task because it is one of the few employing an emotional model consisting of eight fundamental emotions: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust.

The reason why this dataset was not initially selected as a possible working dataset lies both in the lack of multimodality, since no audio features are available, and in its small size: even though it may be a suitable source dataset for the transfer learning task, using it to train stand-alone models would be too limiting and lead to an insufficient generalization ability.

4.2.1 Data Preparation

In order to maximize the model's ability to correctly predict the missing labels on Music4All-Onion, it is important that the format of the fine-tuning data is the same as the one that the model will get as input in the classification step. For this reason, the same preprocessing operations that were performed on the lyrics layer of the Music4All-Onion dataset, mentioned in Section 3.2.1.4, need to be applied to the Edmonds Dance source dataset.

It is also worth investigating the source dataset's label distribution in order to highlight any possible imbalance that may cause inaccuracies in the labeling process and, therefore, needs to

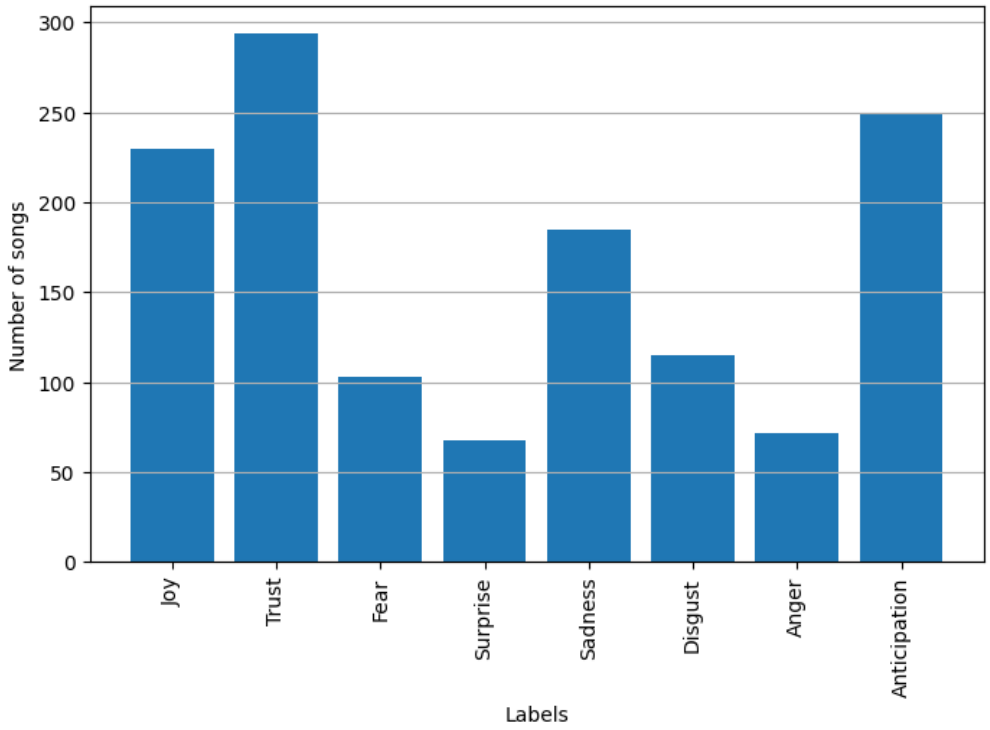

Figure 2: Distribution of emotional labels in the Edmonds Dance dataset.

be addressed. Figure 2 shows indeed an imbalance against the labels fear, surprise, disgust, and anger.

The reason why imbalanced datasets are an issue for learning models lies in the assumption made by some algorithms during their training: if the distribution of the labels is skewed, models will learn how to perform optimally in those scenarios; this may mean that less importance is given to the under-represented classes, therefore leading to a bias in the prediction process. As an example, considering the distribution of the labels of the dataset at hand, the model could

learn during its training that surprise is almost never predicted, and thus it could reflect this knowledge at test time by avoiding predicting that label in favor of more frequent ones.

To address this issue, different experiments are run using different versions of the dataset: the original version, an *upsampled* version and a *downsampled* version. Upsampling and downsampling are two popular techniques used to tackle the class imbalance problem in datasets [66]: the former consists of increasing the cardinality of the data by duplicating a number of samples belonging to the under-represented classes, whereas the latter in decreasing the cardinality by removing a number of samples from the over-represented classes.

Some additional preliminary considerations need to be discussed on the nature of the classification problem: music classification as a multi-class problem[1] can be considered multi-label or mono-label, depending on the number of labels that can be assigned to each sample. For the scope of this thesis, the choice of a multi-label dataset is made, since each song can encompass more than one single emotion, especially given the non-mutually exclusive nature of the emotional classes used; the multi-label nature of the Edmonds Dance dataset also supports this decision.

In order for the multi-label dataset to be successfully used in association with the classification model, it is possible to represent the labels leveraging one-hot encoding: the labels associated with a sample are represented using a vector of length n, with n being the number of possible classes, such that each i^{th} element of the vector $[x_1, x_2, ..., x_i, ..., x_n]$ is equal to 1

[1]More than two labels to choose from.

only if the i^{th} emotion in the labels set is assigned to the sample, 0 otherwise. The ordered set of emotions is joy, trust, fear, surprise, sadness, disgust, anger, and anticipation. An example of this encoding strategy is shown in Table II.

TABLE II: EXAMPLE OF ENCODED DATASET

idx	**Lyrics**	**Labels**	**Encoded labels**
0	one day life understand fight surviv taught lu...	Joy, Trust, Surprise	[1, 1, 0, 1, 0, 0, 0, 0]
1	hypnot love without air even breath lead way l...	Fear, Sadness	[0, 0, 1, 0, 1, 0, 0, 0]
2	stand littl close stare littl long danc everi ...	Joy, Trust, Anticipation	[1, 1, 0, 0, 0, 0, 0, 1]
3	fall piec need need fault weak turn cold cut b...	Surprise, Sadness, Disgust	[0, 0, 0, 1, 1, 1, 0, 0]
5	end alon done paid get want load gun made alon...	Surprise, Sadness, Disgust, Anger	[0, 0, 0, 1, 1, 1, 1, 0]

4.2.2 Evaluation Metrics

The transfer learning model's performance is evaluated on a dedicated test set using precision, recall, F1 score, and Hamming loss; precision, recall, and F1 score are computed on the whole data according to Equation 4.1, Equation 4.2, Equation 4.3 and following to the

micro-average approach to account for imbalance in the data: this strategy allows each sample to have the same weight, instead of assigning the same weight to each class[1].

$$precision := \frac{TP}{TP + FP} \tag{4.1}$$

$$recall := \frac{TP}{TP + FN} \tag{4.2}$$

$$F1 := 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.3}$$

Where TP, FP, and FN are the number of true positives, false positives, and false negatives, respectively, as defined by the standard confusion matrix:

		True label	
		Positive	Negative
Predicted label	Positive	TP	FP
	Negative	FN	TN

As for the Hamming loss, which represents the fraction of labels that are incorrectly labeled, it is computed for each sample as the Hamming distance between the vector true labels y and

[1]As it happens when using macro-average, where the metrics are independently computed for each class, and then the average between the classes is calculated to obtain the final result.

the vector of predicted labels $\hat{y}$, according to the format defined in Table II and as formalized in Equation 4.4.

$$d(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4.4}$$

The overall Hamming loss is simply computed as the arithmetic mean between the losses of all the samples.

The choice of these metrics is made according to the standard practice in multi-label classification problems [67]; the reason why recall, precision, and F1 score are preferred over accuracy lies in their ability to better capture the performance of models by leveraging the relevance of true positives and accounting separately for false negatives and false positives. The additional evaluation of the Hamming loss compensates for the lack of a measure of accuracy by providing a more general estimate of the overall performance.

4.2.3 BERT Pre-Trained Model

The authors of [24] set up the transfer learning part of their experiments leveraging a BERT model for multiclass classification, in which the possible emotional labels are those of Russell's four-emotion model.

BERT, which stands for Bidirectional Encoder Representation from Transformers, is a transformer-based model developed by Google. Its key characteristics include the bidirectionality of its text processing and the use of masked language modeling [27], with the former allowing to leverage preceding and following context by conditioning on both left and right context in texts. The latter is a strategy consisting in randomly masking some tokens in the input text to train the model to predict them in order to refine its contextual awareness.

As already mentioned in Section 3.1.2, this research aims to use a more complex emotional representation, namely Plutchik's wheel of emotions, consisting of eight different labels. Since the dataset intended to be used as a source of information for the TL model only consists of 524 songs, using a pre-trained model could lead to an increase in the quality of the prediction. Therefore, a pre-trained BERT model for emotion classification publicly available on Hugging Face[1] is selected for its compatibility with the task at hand. Specifically, the BERT model chosen is pre-trained for classification on the SemEval-2018 dataset, composed of tweets and corresponding emotions; the emotional labels that the model is able to predict consist of Plutchik's set of eight emotions in addition to three other emotional labels: love, optimism, and pessimism. Since no labeled lyrics dataset compatible with these 11 emotions was found, the model is adjusted to only predict the eight base emotions present in the Edmonds Dance source dataset.

To leverage transfer learning as a labeling method for the lyrics dataset of Music4All-Onion, the first step consists of fine-tuning the selected BERT model on the training split Edmonds Dance dataset while keeping a validation set to be able to quantitatively evaluate the performance of the model before using it on the unlabeled data.

To implement the multi-label classification problem in practice, it is also necessary to know how the classification model works: the chosen pre-trained BERT model outputs for each sample and for each possible class a score indicating the probability that that sample belongs to that

---

[1]https://huggingface.co/ayoubkirouane/BERT-Emotions-Classifier

class. In the mono-label case, it would be sufficient to select the class corresponding to the highest score, but the multi-label case requires some more reasoning: a way of achieving the desired outcome is to set a probability threshold above which a sample is assigned to the class. Selecting the right threshold for this task is complex, as it requires considering the trade-off between the higher precision given by a higher threshold[1] and the higher recall given by a lower threshold[2]. Given the complexity just mentioned, the threshold is not selected a priori; instead, different thresholds are tested with the different dataset versions to identify which combinations lead to the best results on the validation set.

After splitting the Edmonds Dance Dataset into a training and a validation set, respectively containing 80% and 20% of the original data, the BERT pre-trained model is trained on the dedicated split of the dataset for 10 epochs[3]. The low number of epochs chosen for these experiment reflects the computational limitations faced, however, given the small nature of the dataset and the experimental results obtained, it can be considered sufficient. A specific indicator of this that will be discussed later in this section is how the loss of the model changes during training.

---

[1]Reducing false positives.

[2]Reducing false negatives.

[3]30 and 60 epochs have also been tested for some configurations but did not lead to an increase in performance, probably causing the model to overfit.

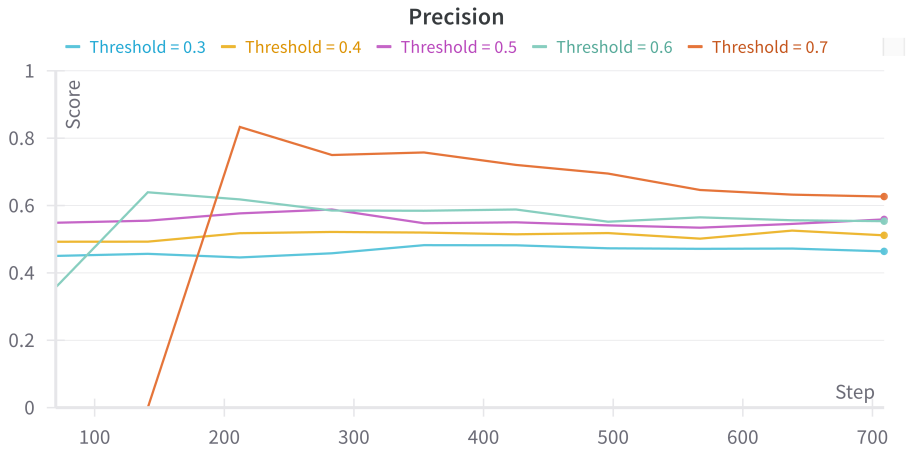

Figure 3: Precision of BERT pre-trained model over epochs' steps.

The results of the experiments on the base dataset where no upsampling or downsampling operations are performed, according to different threshold values, are shown in Figure 3, Figure 4, Figure 5, and Figure 6.

As anticipated, the value of the threshold has a direct influence on precision and a recall: a lower threshold leads to a lower number of false negatives, therefore improving recall, whereas a higher threshold causes fewer false positives and therefore improving precision. The F1 score, keeping into account both precision and recall values, is thus considered as the base metric for evaluation in this scenario.

In order to avoid extreme scenarios, two threshold values are initially selected for further experiments based on the results achieved: since 0.3 and 0.4 lead to similar results in terms

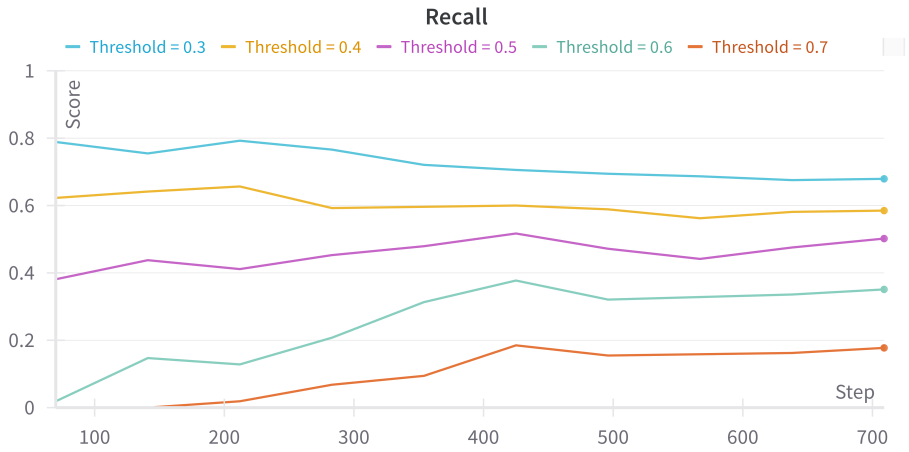

Figure 4: Recall of BERT pre-trained model over epochs' steps.

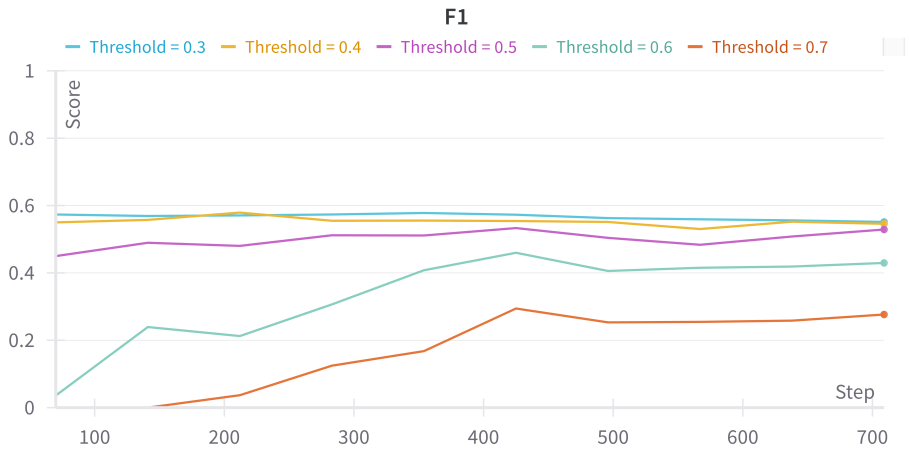

Figure 5: F1 score of BERT pre-trained model over epochs' steps.

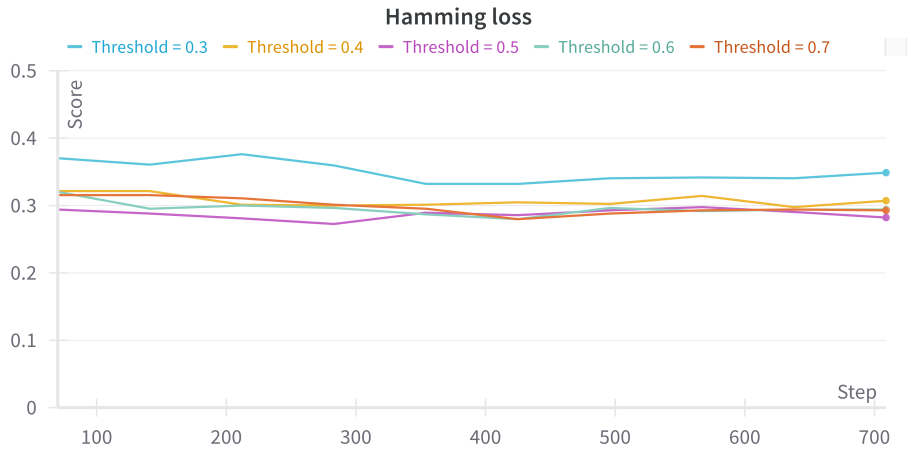

Figure 6: Hamming loss of BERT pre-trained model over epochs' steps.

of F1 score, their average 0.35 is considered; 0.5 leads to an overall good performance both in terms of F1 score and in terms of Hamming loss, and is thus chosen as the second viable threshold.

Other experiments are then run using the threshold values just selected and the different dataset versions including corrective measures for the class imbalance problem. The results on the upsampled dataset are shown in Figure 7, Figure 8, Figure 9, and Figure 10, whereas those on the downsampled data in Figure 11, Figure 12, Figure 13, and Figure 14.

One detail that is immediately noticeable from the performance reports is the lack of improvement over the epochs: the graphs show that the metrics oscillate around a stable value after the initial settling stage. To further investigate this, it is possible to consider the loss of

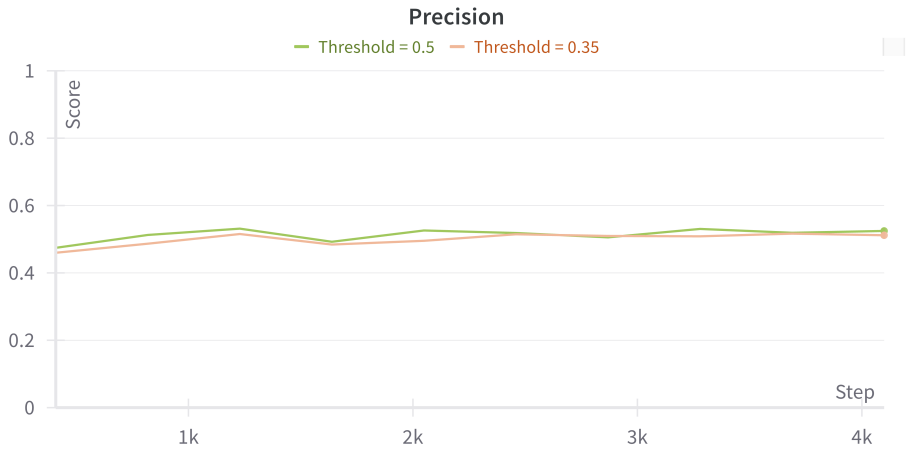

Figure 7: Precision of BERT pre-trained model over epochs' steps (upsampled data).

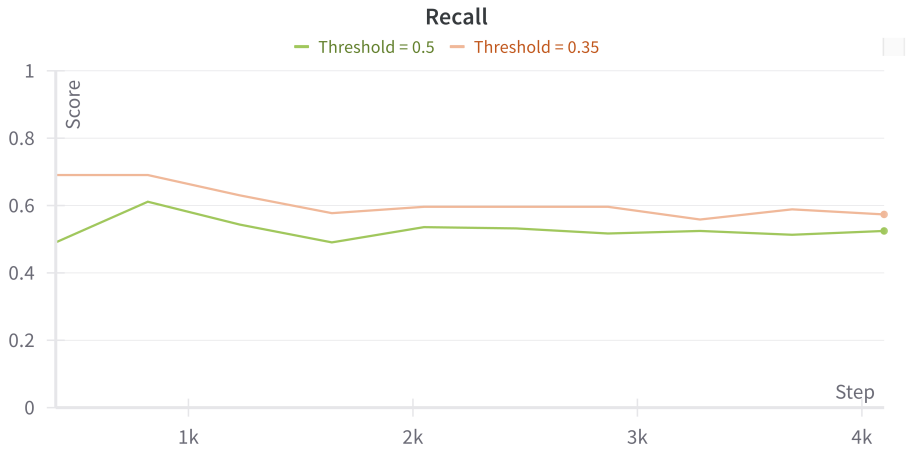

Figure 8: Recall of BERT pre-trained model over epochs' steps (upsampled data).

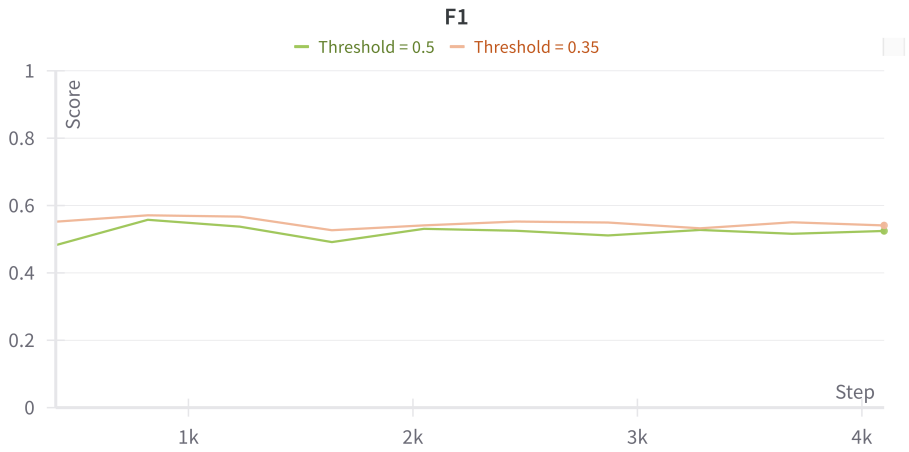

Figure 9: F1 score of BERT pre-trained model over epochs' steps (upsampled data).

Figure 10: Hamming loss of BERT pre-trained model over epochs' steps (upsampled data).

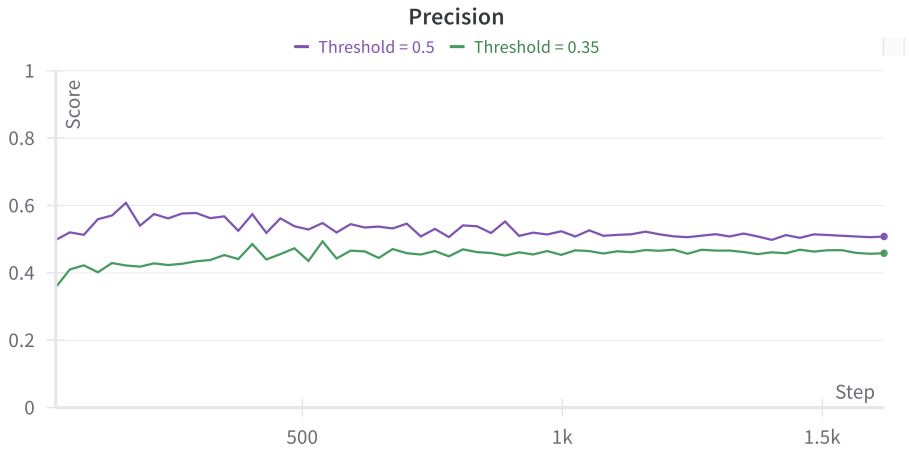

Figure 11: Precision of BERT pre-trained model over epochs' steps (downsampled data).

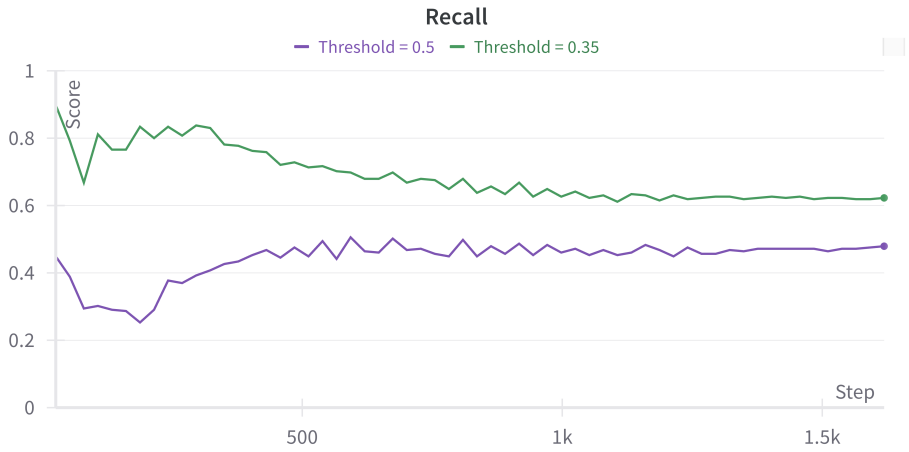

Figure 12: Recall of BERT pre-trained model over epochs' steps (downsampled data).

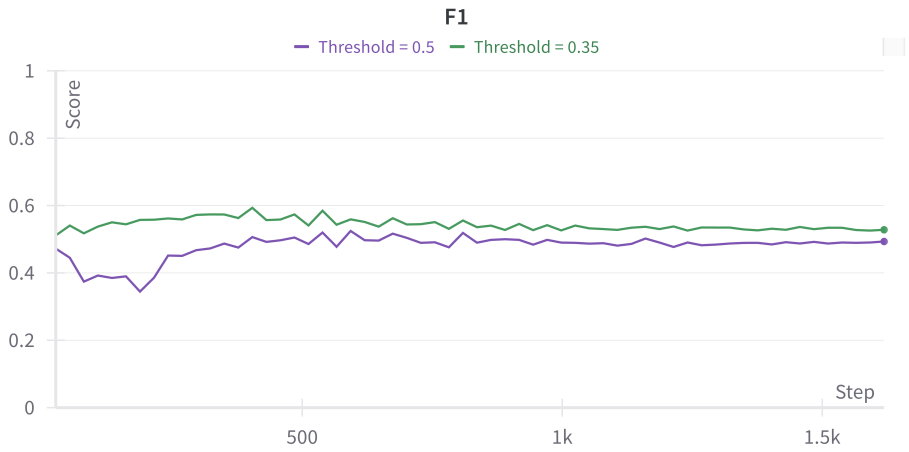

Figure 13: F1 score of BERT pre-trained model over epochs' steps (downsampled data).

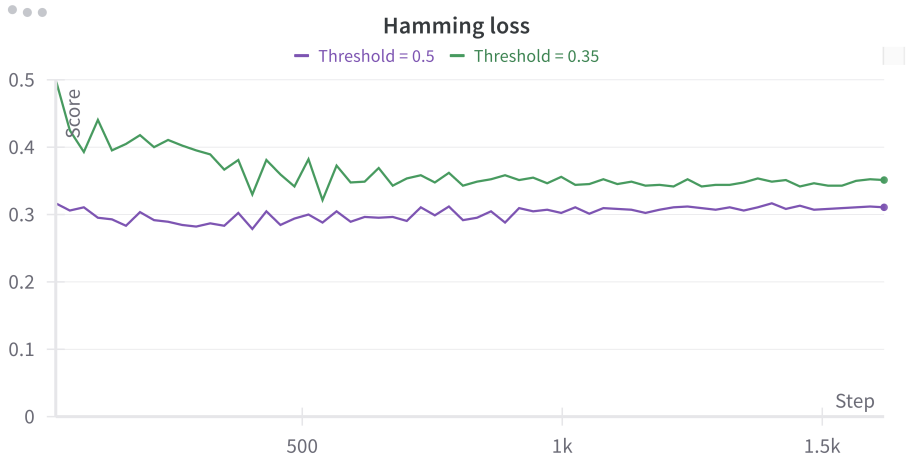

Figure 14: Hamming loss of BERT pre-trained model over epochs' steps (downsampled data).

Figure 15: Evaluation loss of BERT pre-trained model over epochs' steps (upsampled data).

the model on the evaluation set. As an example, Figure 15 shows the loss during training computed on the evaluation set: the increasing trend of the graph reflects the missing improvement in performance, indicating that the model does not benefit from additional training but instead overfits the training data and performs thus poorly on the unseen evaluation set.

The same considerations can be made for the loss of the model trained on the downsampled data, as shown in Figure 16.

The results of the training of the pre-trained BERT model on the Edmonds Dance dataset are not completely satisfactory: even in the best scenario, the highest F1 score reached on the evaluation set barely reaches 0.6, and the Hamming loss never falls below 0.3, indicating that if this model was used to label the Music4All-Onion lyrics dataset, only 70% of the assigned

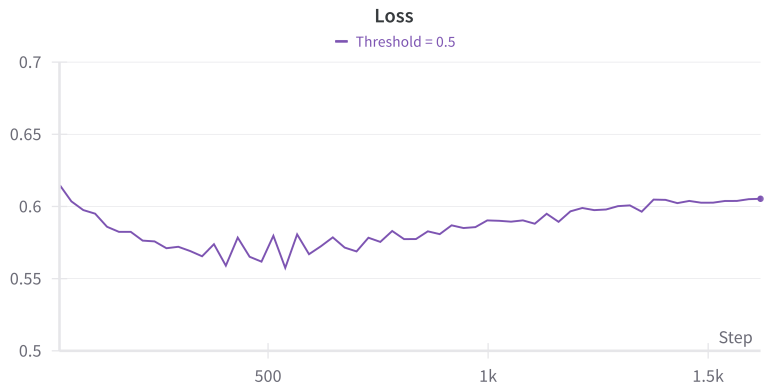

Figure 16: Evaluation loss of BERT pre-trained model over epochs' steps (upsampled data).

labels could be considered correct. For this reason, it is necessary to consider different models for this task.

4.2.4 Large Language Models

Large language models, or LLMs, have become increasingly popular over the past years due to their remarkable performances on NLP tasks and the ease of use provided by their integration into user-friendly tools like chatbots. In short, LLMs are large neural networks trained on very extensive amounts of data to be able to process and generate text similar to natural human language.

Like BERT, LLMs are based on the transformer architecture, first introduced in [68], which revolutionized the deep learning field by presenting a model exclusively based on a self-attention mechanism. This allows models to assign different weights to inter-connected words of a sentence

and thus incorporate larger context without the need for recurrent or convolutional elements. While BERT can be considered the first large language model, modern LLMs, such as those discussed and used in this section, propose an alternative approach to its encoder-only architecture by implementing a decoder-only one. As a consequence, BERT's primary focus is the bidirectional analysis of context for classification tasks, whereas architectures such as general pre-trained transformers (GPT) yield generative auto-regressive models leveraging a sequential learning strategy in a single direction (from left to right). This leads to different applications for the architectures, with modern LLMs such as GPT being preferred for natural language generation or conversational tasks, whereas BERT excels in the interpretation of existing texts.

In practice, large language models consist of tens or hundreds of billions of parameters that are trained on remarkably large corpora of text data, which allows them to capture a more extensive range of language patterns. This is why LLMs present a promising alternative to BERT for the labeling task at hand, although such tasks are typically closer to BERT's domain. Despite the large dimensions of these models, their popularity led to the development of a great number of techniques that make their fine-tuning and usage accessible even under the computational constraints imposed by the limited hardware resources available to the general public.

To complete the labeling task at hand, multiple LLMs are tested and compared using a similar strategy to the one described in Section 4.2.3. Specifically, the same train and evaluation splits are used to fine-tune and then evaluate the performance of the models, with modifications only to the format of the dataset. The creation of an instruction dataset that can be fed to the

model consists of using a fixed prompt to specify the task that the model needs to perform, in addition to some examples, to the sample to label, and to the corresponding true labels for those in the training split of the dataset.

The instruction given to the models is the following, with additional appropriate separators according to the specific model used.

```
Associate the preprocessed song lyrics to at least one of the following
emotions: Joy, Trust, Fear, Surprise, Sadness, Disgust, Anger, Anticipation.
Only use the listed emotions. Each song can be associated with multiple
emotions. No song is associated with no emotion.
Here are some examples:
Input: high drunk moonlight fli touch sky eye danc star shine light night bodi
blind blow mind got ta know cuz gon ride night come aliv lit like fire lit like
fire tonight come aliv lit like fire lit like fire tonight hot got burn feel
good swear could die alright want touch cross line bodi blind blow mind got ta
know cuz gon ride night come aliv lit like fire lit like fire tonight come aliv
lit like fire lit like fire tonight
Response: Joy, Trust, Anticipation

Input: turn somebodi save soul caus sin citi know mani troubl lover got lose
control like drug luxuri sugar gold want good life everi good night hard one
hold caus even know make hand clap said make hand clap somebodi save soul caus
```

sin citi know mani troubl lover got lose control like drug luxuri sugar gold want affect hold close ha ha ha caus even know make hand clap said make hand clap everi night star come live soul around need believ could hold caus need someth good right could scream til sun come wake sound get knee say prayer jame brown make hand clap make hand clap turn make hand clap flesh search worst best ever deni like stranger gim danger wrong right secret broadway freeway keeper crime fear convict grape wrath sweeten wine even know make hand clap said make hand clap everi night star come live soul around need believ could hold caus need someth good right could scream til sun come wake sound get knee say prayer jame brown make hand clap make hand clap make hand clap make hand clap make hand clap get handclap
Response: Anticipation, Sadness, Surprise

Please also keep in mind that texts containing insults or derogative words should influence the text in terms of negative emotions perceived (e.g. disgust, anger). Similarly, texts containing references to love and affection should point to positive emotions such (e.g. joy, trust).
A text with many verbs in the future form could suggest anticipation.
Please give the same importance to all the eight classes.

The format of the prompt has been subject to some refinements before reaching its final form; at first, the LLMs were fed a prompt in which no examples nor additional suggestions were provided, corresponding to the first four lines of the prompt displayed above. The selection of this prompt originated from the attempt of using instructions that were as simple and clear as possible as a starting point. Since the models' answers were not consistent with what the prompt instructed at first, two examples have been added to assess whether the prediction capabilities could benefit from their additions. After noticing that the models' performance had improved due to the examples, the final suggestions about how to recognize specific classes and the requirement to balance the importance of the classes have also been added for further refinement and to prevent the models from focusing on the most recurrent classes only.

The first models tested are Meta's LLaMA 2 [69] with 7B parameters and LLaMA 3.1 [70] with 8B parameters, and Google's Gemma 2 [71] with 2B parameters, all united by a decoder-only architecture.

These models have been made freely available by their creators on Hugging Face, and can therefore be fine-tuned locally. To reduce the computational need of such operations, a Parameter-Efficient Fine-Tuning (PEFT) strategy is used: this technique enables training on a limited set of parameters, while the majority of them remains fixed. The technique chosen for this purpose is a QLoRA [72], a quantized version of the Low-Rank Adaptation (LoRA) [73], which adds low-rank updates to the weights of some layers of the network.

To better understand the need for a Parameter-Efficient Fine-Tuning strategy, it may be useful to highlight the key stages of how fine-tuning is performed on neural networks and, thus,

large language models. It is possible to simplify the structure of a neural network to a set of interconnected subsequent layers of units, called *neurons*, which are organized into one input layer, one or more hidden layers, and one output layer. The neurons of a layer are connected to those of the following layer through weighted edges, whose weight represents the importance of the connection. The weights are the elements that are modified during fine-tuning through backpropagation[1], and are thus the core element of the process. The matrix containing the weights of all the connections in the network is often very large, and operations on it in its base form can become memory-intensive and computationally expensive.

When applying LoRA, the weights of the pre-trained network are frozen and instead two low-rank matrices A and B are used to store the updates resulting from the fine-tuning process. In this way, it is only necessary to store A and B instead of the original weights matrix, and predictions will be made using weights obtained by applying the modifications stored in A and B to the model's weights matrix. Eventually, QLoRA introduces a further level of efficiency by quantizing[2] the low-rank matrices A and B to further reduce the memory usage.

The final model tested is OpenAI's GPT-4o-mini updated to July 18^{th} 2024, belonging to the GPT-4 family [74], whose exact number of parameters has not been disclosed. This model can be fine-tuned using OpenAI's API, therefore eliminating the need for any PEFT technique and

---

[1]Algorithm allowing the adjustment of weights from the output layer to the input layer to minimize a selected loss function, measuring the distance between the model's predictions and the correct ones.

[2]Conversion of the data format from a higher to a lower precision (e.g. from 32-bit floating-point to 8-bit integers).

allowing every parameter to be corrected according to the new information acquired through the Edmonds Dance training data.

Table III shows the performance achieved by the four models tested on the unseen evaluation set after 100 fine-tuning epochs.

TABLE III: COMPARISON OF PERFORMANCE OF LLMS TESTED

Model	**Parameters**	**Hamming loss**	**Precision**	**Recall**	**F1-score**
Gemma	2B	0.31	0.48	0.60	0.53
LLaMA2	7B	0.32	0.48	0.60	0.53
LLaMA3	8B	0.29	0.54	0.62	0.58
GPT-4o-mini	NA	0.05	0.93	0.90	0.91

The model that performed better, as is immediately visible from Table III, is GPT-4o-mini; such a large difference in performance may be attributed to the possibility of fine-tuning the whole model without the need for any PEFT techniques, which enables greater customization of the model on the task at hand. Another factor that could have affected the performance is the total number of parameters of the model, but a real comparison cannot be performed due to the lack of information about GPT-4o-mini.

Since the experiments on the unseen portion of the Edmonds Dance dataset resulted in such a good performance, GPT-4o-mini is the final model chosen for the labeling of the Music4All-Onion lyrics dataset. Due to the lack of true labels, it is impossible to estimate the performance of the model on this specific dataset, but assuming that the samples of the two datasets are not remarkably different allows the final dataset to be considered reliable overall.

GPT-4o-mini is the only model among those tested that is not freely available to the public but is accessible on a pay-per-use basis, with limitations on the number of usable tokens per day; this led to the necessity of reducing the cardinality of the Music4All-Onion working dataset from the original 109.269 tracks to approximately 33.000.

The experiments described in this section played a key role in this research, providing emotional labels to the lyrics component of the Music4All-Onion dataset. While the approach leveraging users' tags has been deemed insufficiently reliable, the transfer learning one provided more robust results, especially when using the GPT-4o-mini model, which outperformed all other alternatives.

CHAPTER 5

BASELINES DEFINITION

In order to ease the evaluation of the methods proposed in this research, it is helpful to preliminarily define some baselines as benchmarks for comparison. As already stated in Section 2.1.1, this introduces challenges because of the lack of public resources for researchers in this field: not having, for instance, a standardized dataset on which to evaluate novel strategies complicates the comparative evaluation of techniques, forcing each research to rely on a different, often private, dataset, limiting the reproducibility of results. To address this issue, the proposed baselines are evaluated using the emotional extension of the Music4All-Onion dataset developed for this thesis.

Further reproducibility issues arise from the lack of public ready-to-use code related to many publications, imposing the creation from scratch of the models described; however, some necessary technical details are often missing, making it difficult to precisely re-implement promising architectures and evaluate them fairly.

One final consideration on reproducibility concerns the audio modality: due to the nature of data used in this research, most audio classification techniques presented in the literature are not applicable. This is primarily due to the lack of data representations used as inputs to the models reported in Section 2.1.5, namely spectrograms and MFCCs. These limitations further aggravate the challenges posed by the already discussed lack of labels, restricting the approaches that can be explored in this research.

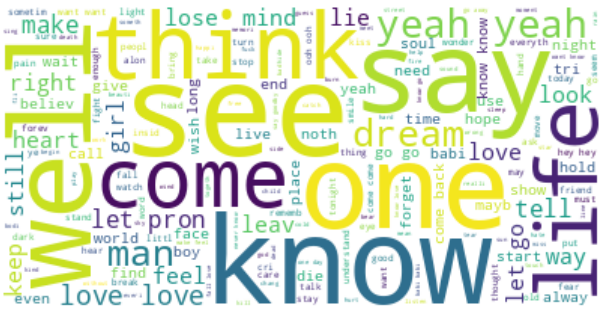

Figure 17: Word cloud representation of frequent words in the lyrics layer of the dataset.

5.1 Lyrics Modality

These experiments are run on the Music4All-Onion lyrics layer, containing preprocessed texts on which standard NLP standardization techniques have been applied, as explained in Section 3.2.1.4.

The first baseline on which lyrics data is tested is the Latent Dirichlet Allocation (LDA) proposed in [21]. This aims to model the different topics in the lyrics dataset and identify the most significant ones to highlight underlying patterns. Even though LDA is an unsupervised method per se, the dataset labels are used in a second moment to evaluate the extracted topics, inspired precisely by [21].

Before running LDA on the data, a word cloud is created to easily visualize the most frequent words in song lyrics, shown in Figure 17.

According to the authors of the referenced paper, applying LDA on data after TF-IDF has been applied yields better results since it eliminates the risk of terms that are not sample-specific being given too much importance. The number of topics to be extracted is a hyperparameter

that can be chosen according to the dataset's properties. Since the authors of [21] used an emotional model consisting of 4 base emotions, they set the parameter to 4. The parameter in reproduction is, therefore, set to 8 at first, according to the emotional model used for labeling.

The topics extracted by the LDA algorithm can be described by the top 10 most relevant terms for each of them, presented in Table IV.

TABLE IV: TOP 10 WORDS PER TOPIC

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
mystifi	hallelujah	shoop	jane	hello	nah	love	ba
unchang	carolin	knockin	thou	doctor	dem	get	blah
indigo	dum	twentyfour	lala	suicid	alic	want	groovin
dyou	bop	yonder	jupit	sugar	betti	go	bluebird
messiah	bom	anni	choo	sacrific	ey	know	wonderland
union	strawberri	georgia	arc	ash	banana	come	eagl
woooo	te	utopia	mississippi	sacrific	mash	say	breakdown
reckon	cuckoo	gogo	roman	pill	euphoria	never	dee
lifelin	venus	potion	undead	flag	disco	let	passag
ella	foggi	amsterdam	joan	earli	wasp	like	you

The top 10 terms shown in Table IV already demonstrate that LDA did not extract any evident underlying emotional topic, but another valuable qualitative evaluation tool can be

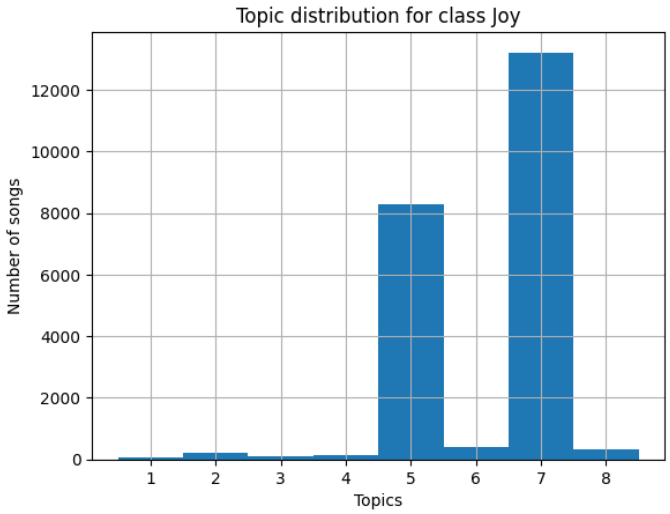

Figure 18: Topic distribution for class Joy.

leveraged to further investigate this: the distribution of topics assigned by the LDA algorithm across the different emotional classes within the dataset can assist in determining if there is any meaningful correlation between LDA topics and emotional labels.

Unfortunately, as seen from Figure 18 to Figure 25, the topic distribution does not vary significantly over the different classes of the dataset, indicating that LDA does not yield good results on this dataset. The reason for this may lie in the dataset's imbalance, as suggested in [75], or it may originate from the peculiarities of song lyrics: unlike documents such as news articles or essays, this type of data is typically shorter and does not present large quantities of text to model; furthermore, it is less common for song lyrics to address a single coherent topic, and often emotions are not explicitly mentioned as *topics* of songs, but instead conveyed through the underlying subtext of the words chosen. An additional note that should be considered lies

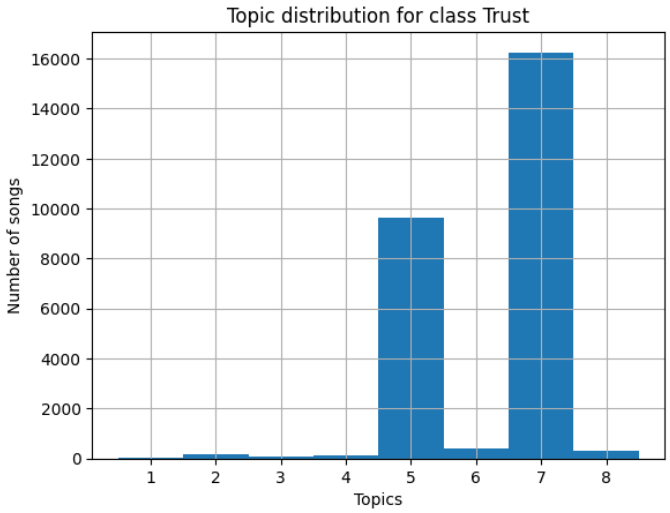

Figure 19: Topic distribution for class Trust.

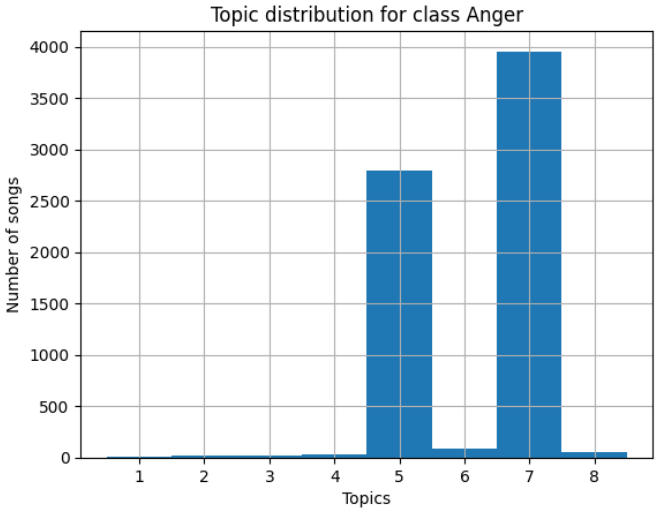

Figure 20: Topic distribution for class Anger.

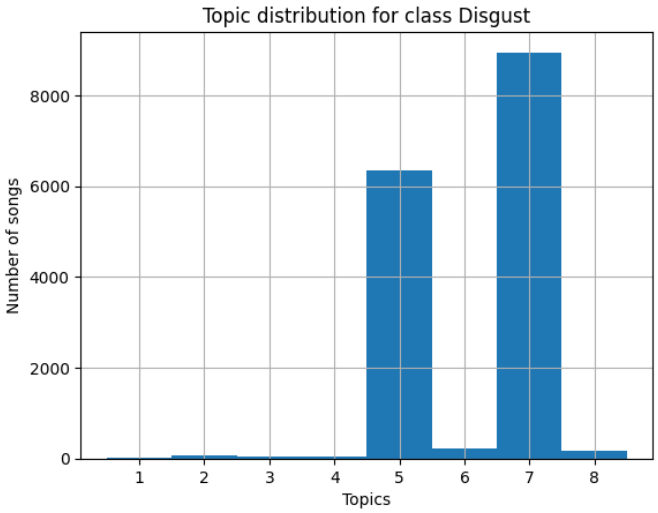

Figure 21: Topic distribution for class Disgust.

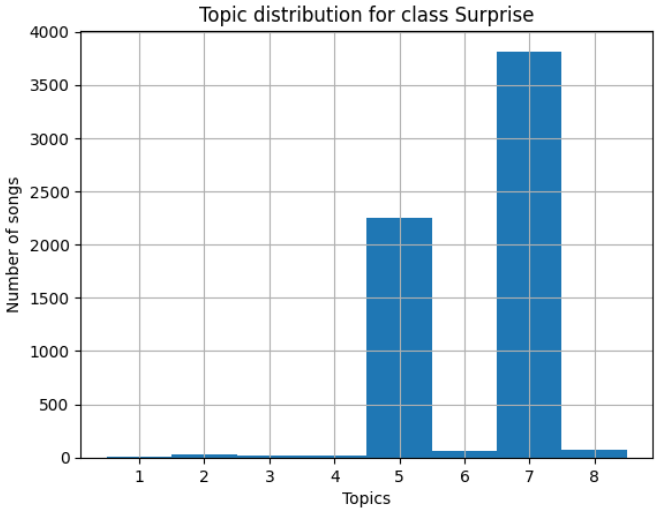

Figure 22: Topic distribution for class Surprise.

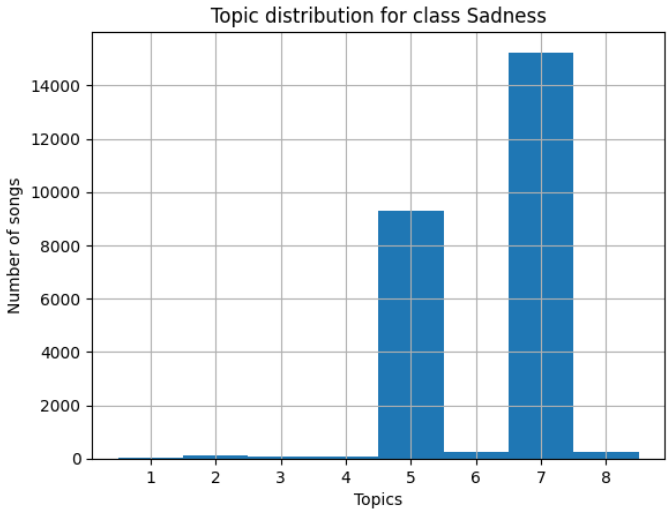

Figure 23: Topic distribution for class Sadness.

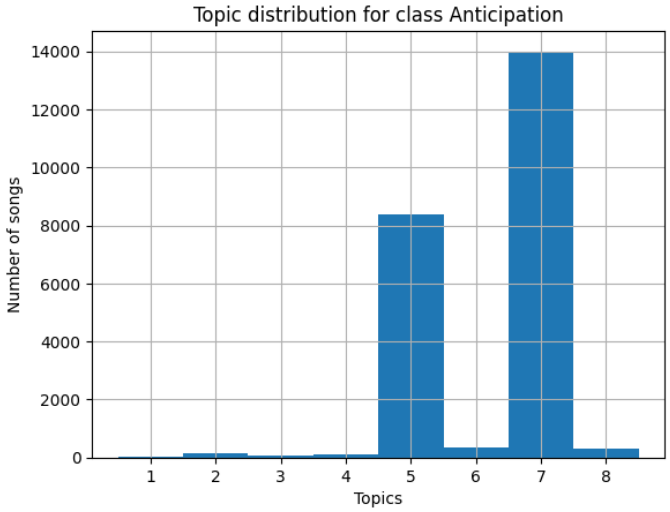

Figure 24: Topic distribution for class Anticipation.

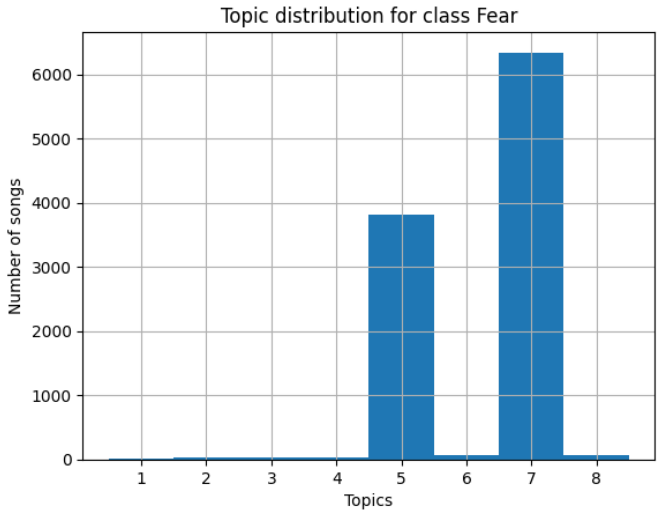

Figure 25: Topic distribution for class Fear.

in the multi-label nature of the problem tackled in this thesis, which is not shared over all the methodologies presented in the literature.

A second paper is chosen as a baseline for lyrics classification: [19], in which a CNN, a bi-directional LSTM, and a CRNN are tested as classifiers. This choice is motivated both by the strong results reported by the authors and by the uniqueness of this work in applying neural networks to lyrics classification. Other works in literature, such as those discussed in Section 2.1.2.3, rely on traditional machine learning models that achieve lower performances or on the BERT model, which has already been tested in Section 4.2.3 with unsatisfactory results.

First, it is crucial to recreate the architectures as closely as possible, using the information presented by the authors. The CNN consists of three concatenated convolutional layers

of varying kernel sizes[1] to allow the identification of patterns of different dimensions, each followed by max pooling; these are followed by two dense layers with a Rectified Linear Unit (ReLU) activation function, a dropout layer to address the risk of overfitting and, finally, a softmax layer. The chosen loss function is categorical cross-entropy, suitable for multi-class classification problems. No other details are provided on the training parameters, and it is thus impossible to reproduce the experiments presented precisely. However, for research purposes, the experiments are carried out anyway, and the missing parameters are tuned by leveraging Bayesian Optimization [76].

Bayesian Optimization is a strategy developed to address the hyperparameter tuning issue of machine learning algorithms. The fundamental idea behind it is modeling a learning algorithm's generalization ability as a sample from a Gaussian process[2], which allows the development of a probabilistically guided search strategy which is remarkably more efficient than the brute-force approach in which numerous configurations of candidate parameter values are tested indiscriminately, such as random search and grid search.

As mentioned above, it is crucial to account for differences in problem definitions: the design choices made by the authors reflect the needs of a multi-class single-label classification problem, whereas the task of interest is a multi-label classification problem. To make the architecture compatible with the type of problem tackled in this thesis, some slight modifications must be

---

[1] [2, 5, 10]

[2] Stochastic process in which every finite subset of random variables follows a multivariate normal distribution.

made to a few crucial elements of the CNN, such as the selected loss function. An overview of the categorical cross-entropy loss function is provided below to better understand the reasons for this.

The categorical cross-entropy loss function, also called *Softmax* loss, combines a Softmax activation function with a cross-entropy loss, whose computation is shown in Equation 5.1, where C is the number of possible classes, and t and p are respectively the true and predicted probability distribution over all the possible classes.

$$CE(t,p) = -\sum_{i=1}^{C} t_i log(p_i) \tag{5.1}$$

The Softmax activation function, whose formula is presented in Equation 5.2 and whose trend is shown in Figure 26, is used on logits[1] to convert them into probability values representing the likelihood of a sample belonging to each possible class.

$$\sigma(\overrightarrow{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{5.2}$$

Since the goal is to transform logits into a probability distribution, the function ensures that all output values are in the range $[0, 1]$ and that they sum up to 1. One key feature of the Softmax activation function is that it cannot be applied independently to each logit value; instead, the probabilities of a sample belonging to different classes are interdependent. As

---

[1]Raw and unnormalized outputs of a neural network.

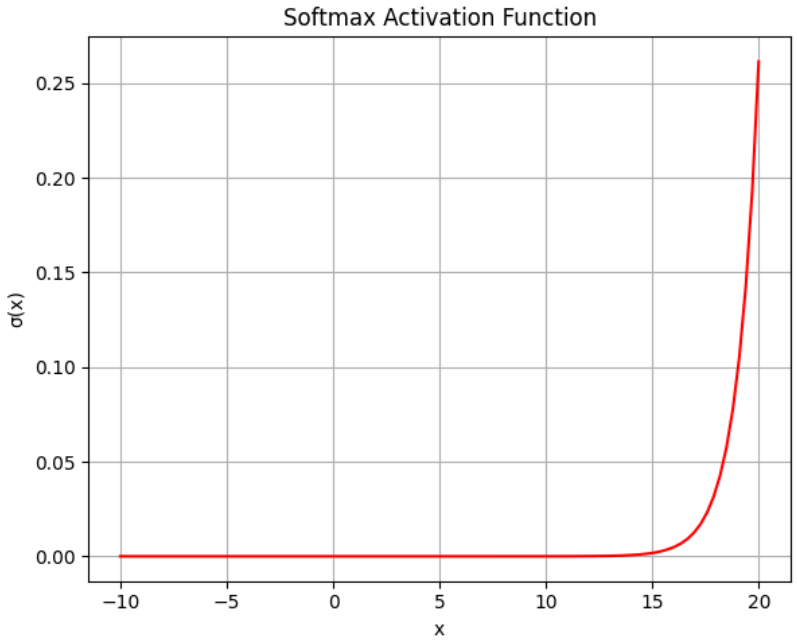

Figure 26: Softmax function trend.[a]

[a] Because the Softmax function is multivariate and converts vectors of any length and values into a vector of sum 1, its shape may vary from case to case.

a result, an increase in the likelihood of a sample belonging to a certain class is necessarily balanced by a decrease in other probabilities, reflecting the mutually exclusive relationship of labels in single-label classification.

To adapt the proposed architecture to the multi-label classification problem, it is necessary to revise the loss function to make class probabilities output by the model independent from each other: in this case, the likelihood of a sample belonging to a specific class should be independent of the likelihoods of it belonging to different classes. One way of achieving this is by leveraging the binary cross-entropy loss, also called *Sigmoid* loss. Similarly to the Softmax loss, this combines a Sigmoid activation function with the cross-entropy loss defined in Equation 5.1. The sigmoid activation function is formally defined in Equation 5.3, and its trend is shown in Figure 27.

$$S(x) = \frac{1}{1 + e^{-x}} \tag{5.3}$$

The Sigmoid function also returns, for each class, probability values in the range $[0, 1]$; since the likelihood values are independent, they do not sum to 1, reflecting the non-mutual exclusivity of labels. In other words, using binary cross-entropy as a loss function leads to considering the multi-label classification problem as the union of multiple independent binary classification tasks, each involving one of the C classes.

Limited information is provided on both the bi-directional LSTM and the CRNN, further complicating the reproduction of the architectures. However, following what is done for the CNN architecture, the models are built according to the information provided, and all other

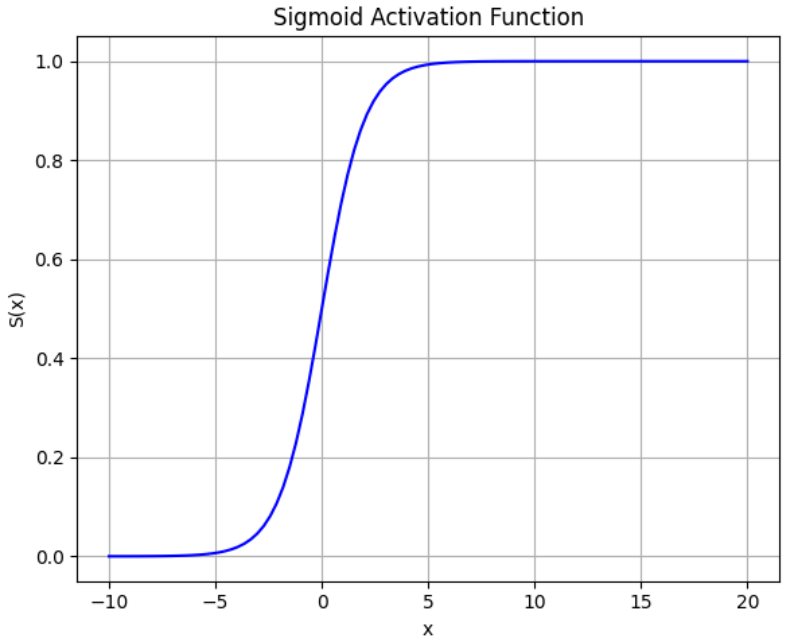

Figure 27: Sigmoid function trend.

design decisions are performed leveraging Bayesian Optimization; the same considerations on loss functions made for the convolutional network are valid for both the bi-directional LSTM and the CRNN.

The length of the training stages for the architectures is provided only in temporal terms[1], but no information allows inferring how many training epochs the models underwent; the experiments are thus conducted for a number of epochs constrained by the hardware and computational resources available. It is evident that all of these adjustments significantly affect the performance of models, and, therefore, the comparative evaluation of models should only be considered as a general qualitative measure and not be used to infer definitive conclusions.

Before being input into the deep learning models, data is transformed using GloVe embeddings, which provide for a better semantically aware translation of textual tokens into numerical vectors, as explained in Section 2.1.2.1 and implemented by the authors of the paper under consideration.

After recreating the architectures, performing the necessary adjustments, and tuning the missing parameters with Bayesian Optimization, each model is trained on a subset of the Music4All-Onion lyrics layer dataset containing 70% of the samples, using the official labels obtained in Section 4.2.4. 15% of the samples are then used for validation by the model after each training epoch, and the final 15% constitutes the separate unseen test set used after training is complete to evaluate the final performances. The training procedure is performed

---

[1] 20 minutes for the CNN, 4 hours for bi-LSTM, 45 minutes for CRNN.

for 15 epochs on each model, and the results look promising. The evaluation is performed using precision, recall, and F1 score, for the same reasons stated in Section 4.2.2; again, the output of the model is a vector of probabilities for each sample, and to identify the final assigned labels it is necessary to set a threshold, as done in Section 4.2.3, which is fixed in this case to 0.4 for the CNN and bi-LSTM models, and to 0.2 for the CRNN[1].

The final results on the test data are shown in Table V, whereas the progression of the loss and the metrics over the training epochs are shown in Figure 28 and Figure 29 for the CNN, in Figure 30 and Figure 31 for the bi-LSTM, and in Figure 32 and Figure 33 for the CRNN.

TABLE V: BASELINE PERFORMANCES ON MUSIC4ALL-ONION LYRICS LAYER

	CNN	bi-LSTM	CRNN
Precision	0.68	0.65	0.55
Recall	0.52	0.53	0.10
F1 Score	0.59	0.59	0.17
Hamming Loss	0.22	0.23	0.30
Training Time	60 min	140 min	300 min

[1]Empirically, a threshold of 0.4 is too high and leads to very few labels being assigned.

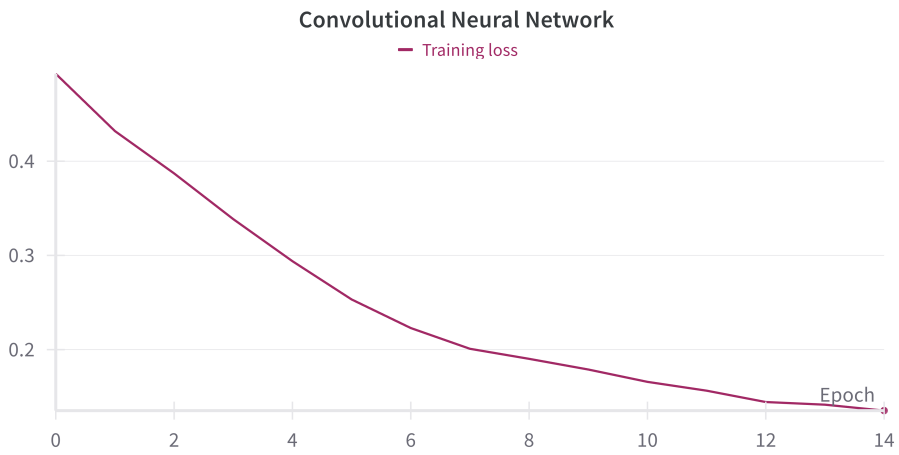

Figure 28: Convolutional Neural Network training loss.

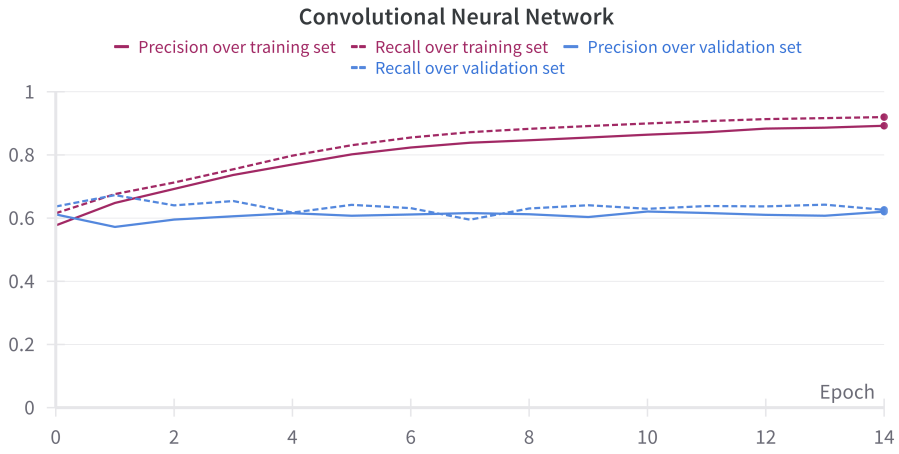

Figure 29: Convolutional Neural Network training metrics.

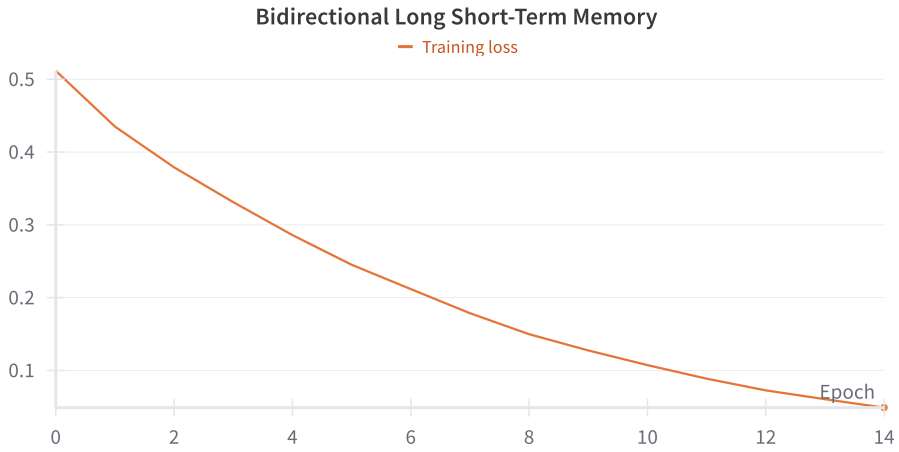

Figure 30: Bidirectional Long Short-Term Memory training loss.

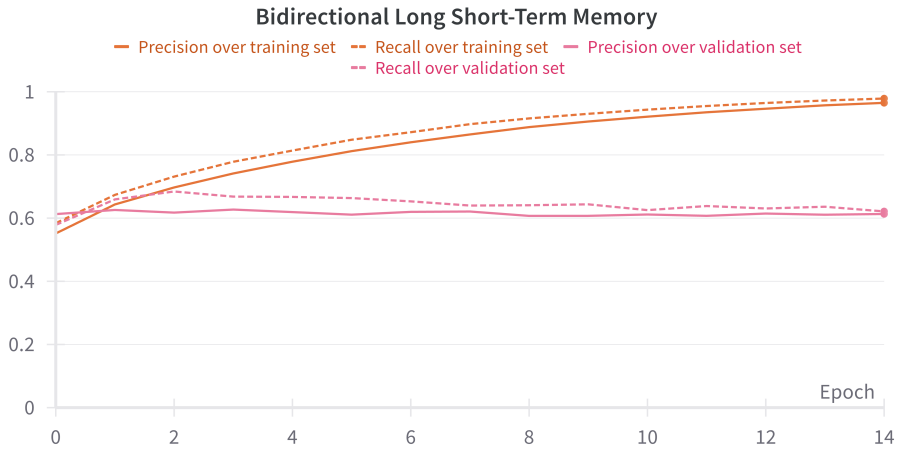

Figure 31: Bidirectional Long Short-Term Memory training metrics.

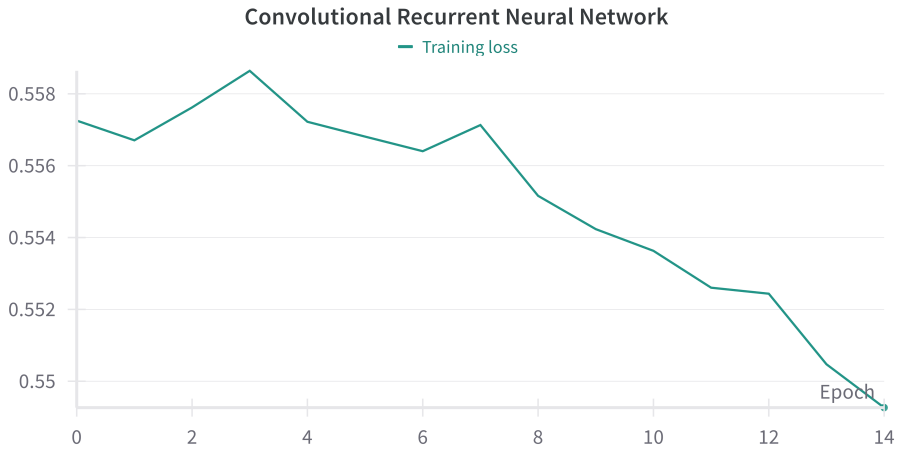

Figure 32: Convolutional Recurrent Neural Network training loss.

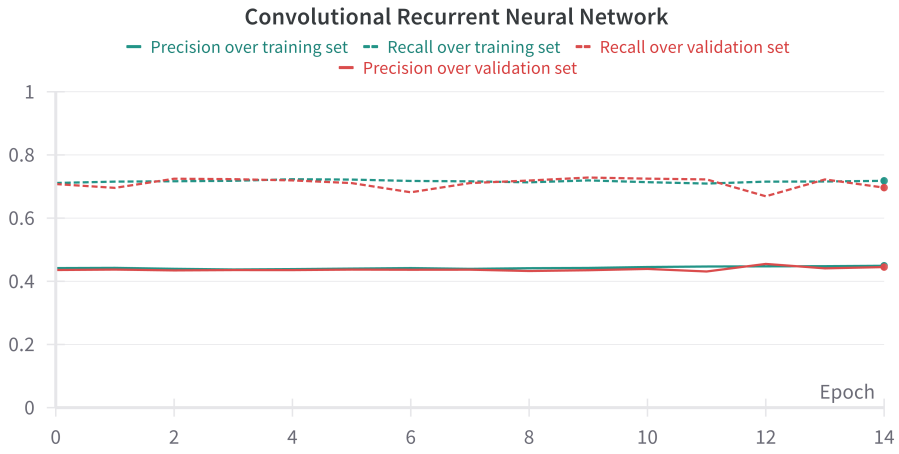

Figure 33: Convolutional Recurrent Neural Network training metrics.

The authors provide an evaluation of the performance of the models only in terms of accuracy, which, for the reasons already stated in Section 4.2.2, is not a suitable metric for the multi-label classification problem. However, the best accuracy reported in [19] is 0.71 obtained with the CNN architecture, followed by 0.69 and 0.67 obtained with the bi-LSTM and the CRNN, respectively. Comparing their results with the results obtained on the dataset used in this thesis from a merely qualitative point of view, it is possible to notice good performances with both the CNN and the bi-LSTM, along with good training times; the same cannot be said for the CRNN, which, despite achieving good precision, has a remarkably low recall, indicating the model's inability to detect positive labels. The good performance of the CNN and the bi-LSTM is reflected by the loss progression over the training epochs, shown in Figure 28 and Figure 30. Although the loss progression for the CRNN is not as smooth as those of the other architectures, indicating unstable learning, the overall trend is still promising and indicates successful learning by the model.

5.2 Audio Modality

Because of the unsupervised nature of the audio modality datasets, the baselines found in the literature are discussed at a very high level due to the challenges in objectively evaluating different methodologies with no ground truth labels available.

As mentioned in Section 3.2.2, one of the first problems to tackle when approaching the audio layer of the Music4All-Onion dataset is the data dimensionality problem: the audio sub-datasets made available are all characterized by a remarkably high dimensionality in terms

of features per dataset, which may lead to issues such as the curse of dimensionality, and to challenges in extracting meaningful information from data.

Dimensionality reduction is a crucial cross-sectional problem in the artificial intelligence field, as the dimensionality problems mentioned above are common to all machine learning domains. For these reasons, numerous techniques have been proposed and are currently in use to mitigate this issue.

The first baseline considered for this phase is the one provided by the authors of the block-level feature extraction approach [38]: as explained in the publication, the dataset on which the automatic music tag classification task is performed is extremely high-dimensional, thus requiring the use of a compression method. The technique chosen by the authors is the Principal Component Analysis (PCA) [77], which works by identifying a new set of axes, namely the *principal components*, to provide an alternative representation of the data. The new axes are chosen to maximize the variance of data over said axes, and are obtained as linear combinations[1] of the original features. The principal components found, which are all orthogonal to one another, are ordered based on the portion of variance explained, with the first PC being in the direction explaining the most variance.

From a mathematical perspective, the computation of the principal components is performed by computing the covariance matrix of the data, which highlights potential correlations between features, and extracting its eigenvectors and eigenvalues, corresponding respectively to the

---

[1]The original dimensions $[x_1, x_2, ..., x_n]$ are weighed and summed, resulting in $w_1x_1+w_2x_2+...+w_nx_n$.

principal components and the relative explained variance. The first n principal components in order of decreasing explained variance are selected based on the desired balance between dimensionality reduction and information preservation, aiming at the optimal trade-off between computational efficiency and data quality. As an example, the authors of the referenced paper performed several experiments to identify how much explained variance to preserve to achieve the optimal balance, and they finally set the threshold to 65%, reducing the number of features from 9448 to 37.

The mathematical discussion highlights the necessity of operating on standardized data to prevent features having different magnitudes from having a disproportionate impact on the results; accordingly, the first operation applied on each audio sub-dataset before performing Principal Components Analysis is normalization, which ensures that every feature to have a zero mean and unit variance, allowing for meaningful comparisons of features.

Following [38], different thresholds of explained variance are tested to evaluate the different levels of dimensionality reduction achieved on the sub-datasets. Table VI presents the resulting number of principal components needed to explain different portions of variance across all the block-level feature datasets. These experiments led to slightly different results from those obtained by the authors of the original paper, highlighting the need for more principal components to explain the same portion of the variance. This discrepancy may be caused by the different datasets used and by the songs' characteristics, which might include acoustic properties influencing the relationships between features.

TABLE VI: NUMBER OF PRINCIPAL COMPONENTS PER EXPLAINED VARIANCE THRESHOLD PER BLF DATASET

Dataset	**60%**	**65%**	**70%**	**75%**	**80%**	**85%**	**uncompressed**
Correlation Pattern	19	29	45	71	112	182	1325
Spectral Contrast Pattern	2	3	4	5	7	9	800
Spectral Pattern	5	8	10	14	18	25	980
Delta Spectral Pattern	8	12	18	27	46	84	1372
Variance Delta Spectral Pattern	14	19	25	33	42	54	1342
Logarithmic Fluctuation Pattern	9	13	19	29	46	75	3626
TOTAL	**58**	**84**	**121**	**179**	**271**	**426**	**9,445**

The most found baseline in literature for unsupervised tasks on audio features is K-Means clustering [78][79], which can be leveraged in various ways as discussed in Section 2.1.4. Because of its widespread usage, it is considered a baseline method for the purpose of this thesis, and is applied to the PCA-reduced block-level features datasets to obtain a starting benchmark for other experiments. The number of PCs for each dataset is the one granting the preservation of 75% of explained variance.

K-means is a partitional, center-based clustering algorithm that divides a set of points into a predefined number of clusters K. It works by selecting K initial centroids[1] for the K clusters

[1]Either randomly or by considering the empirical probability distribution of each point's contribution to the total inertia to speed up convergence.

and by associating each data point to the closest[1] centroid; after all the data samples have been sorted, the coordinates of each centroid are recomputed as the mean of the coordinates of the points belonging to its cluster. The algorithm terminates when a stopping criteria is met, usually when the change in the centroids coordinates between two sub-sequential iterations falls below a certain tolerance threshold or when the maximum number of iterations is reached. A criticality of K-Means clustering is the necessity to set the parameter K a priori, which may be challenging when data is high-dimensional or complex. Other situations in which K-Means does not perform optimally are those in which clusters have varying sizes and shapes, when outliers are present, or when clusters have non-globular shapes.

Due to the lack of labels, the task of evaluating unsupervised methods requires ad hoc metrics that do not rely on a ground truth but evaluate the clustering outcome, such as the *silhouette score*, which is an internal index[2] measuring the goodness of a clustering structure. The formula is defined in Equation 5.4, where i is a sample, $a(i)$ is the average dissimilarity of i from all other objects within its cluster, and $b(i)$ is the average dissimilarity of i from all the clusters to which i does not belong. $s(i)$ has ranges in $[-1, 1]$, with values closer to 1 indicating a better result.

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{5.4}$$

---

[1]Using a discretional distance metric.

[2]Evaluation is performed without respect to external information.

The silhouette score can be used as an evaluator of the overall clustering structure by considering its average value over all data samples i of the dataset, or of the goodness of a single cluster by considering its average values over each point i of said cluster; in this case, the evaluation of K-Means is carried out in the first manner.

A grid search is performed for each BLF dataset to determine the optimal number of clusters K, using the silhouette score as the evaluator; the results are presented in Table VII.

TABLE VII: EVALUATION OF K-MEANS ON THE BLOCK-LEVEL FEATURES DATASETS

BLF data	**# PCs**	**Optimal K**	**Silhouette**
Spectral contrast pattern	5	2	0.37
Correlation pattern	71	2	0.23
Spectral pattern	14	2	0.28
Delta spectral pattern	27	2	0.43
Variance delta spectral pattern	33	2	0.19
Logarithmic fluctuation	29	2	0.29

For a more intuitive evaluation of the clustering results, the visual representation of clusters can be employed; however, representing high-dimensional data is challenging as it requires projecting it into two-dimensional spaces. For simplification purposes, the visual representation

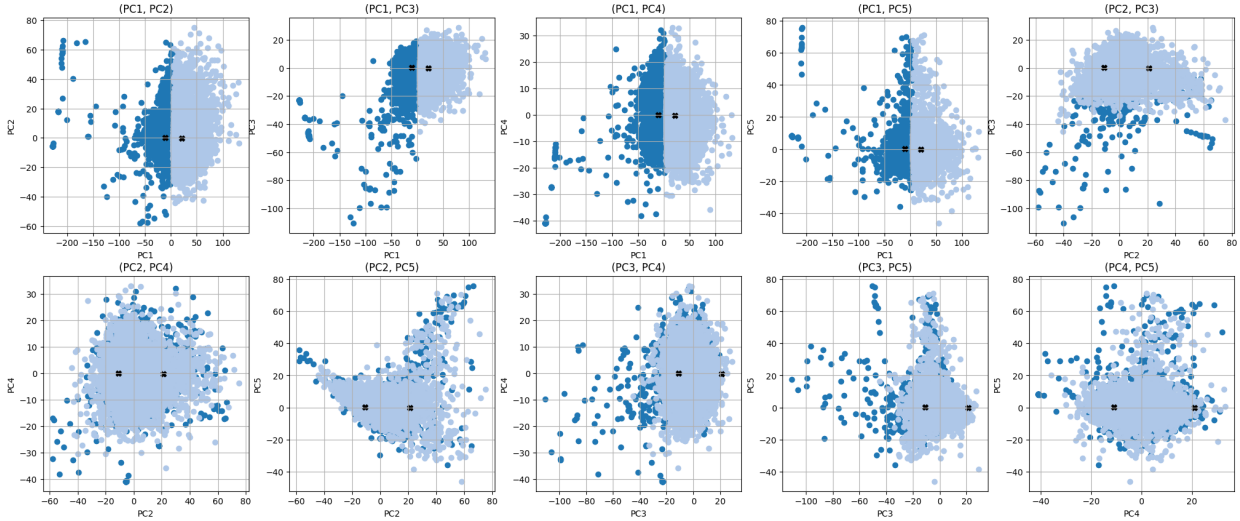

Figure 34: K-Means clustering results on the spectral contrast pattern BLF dataset after PCA.

of K-Means clustering is provided only for the spectral contrast pattern dataset, which has been reduced to 5 dimensions. Figure 34 shows how data samples have been sorted into two clusters, allowing a clearer interpretation of the structure and the potential patterns in data.

As visible from Figure 34, no evident patterns or groups can be identified in the spectral contrast pattern dataset, which is a reflection of the silhouette score of just 0.37. Because of the unsuccessful outcome of the combination of PCA and K-Means, many improvements can be obtained by varying the dimensionality reduction and clustering techniques, as will be discussed in Section 6.1.

The baselines introduced in this chapter constitute a valuable instrument for the evaluation of the models developed in the following sections, as they provide a reference to assess the

performance of new methodologies on the chosen dataset. Furthermore, this section underlined the existing reproducibility and comparison issues in the MER field, caused by both the lack of standardized data and evaluation metrics, and by the limited availability of models pre-trained on the specific emotional information pursued in this work.

CHAPTER 6

METHODS

This section highlights all the methodology and design choices performed to develop the desired multimodal classifier. This represents the initial goal of this thesis and thus forms the core of the research, for which the preliminary operations on the datasets were necessary due to the lack of suitable data discussed in the previous chapters. The results of the experiments described in this section will be presented and discussed in Section 7.

Before discussing in detail the proposed approach, some key considerations from the experiments and insights of the previous chapters can be summarized to provide a clearer understanding of the chosen methodology. The first observations concern the feasibility of different multimodal models given the constraints introduced by the available datasets and the operations described in Sections 3 and 4.

Firstly, the nature of the labeling operations performed poses some dilemmas on the scope of validity of the labels; considering the labeling attempt using user tags described in Section 4.1 but disregarding the reliability issues already mentioned, it is true that the labels extracted describe the tracks in their entirety, meaning that emotions encompassed both by lyrics and by audio are embedded in the datasets. This is deducible by the assumption that people tagging a given song consider it as a whole rather than split into the two separate entities of audio and lyrics; even though the weight given to each part is unknown, unpredictable, and strictly

subjective[1], the large number of samples makes it possible to estimate that the labels on average are comprehensive of both elements.

The same conclusions cannot be inferred for the final labeling process defined in Section 4.2, consisting of leveraging transfer learning from the Edmonds Dance lyrics dataset to the target Music4All-Onion lyrics dataset. According to the authors, the Edmonds Dance dataset was annotated through crowd-sourcing by readers who were exclusively presented with the lyrics of the tracks [65]. The labels, hence, only refer to the lyrics of the songs, and so does the knowledge extracted by the transfer learning model and applied to the target dataset. Consequently, only the lyrics layer of the Music4All-Onion can be considered correctly labeled and suitable for a classification task.

However, the emotional element in the audio component is as essential as the lyrics for listeners and thus cannot be disregarded. Instead of creating a pipeline exclusively focused on pure classification, the audio data are hence clustered using features relevant to the emotional component of the analysis, following the publications mentioned in Section 2.1.4.

6.1 Unsupervised Audio Modality

6.1.1 Dimensionality Reduction

The first open issue that needs to be addressed is the high dimensionality of the audio sub-datasets of the Music4All-Onion dataset. Two possible approaches can be outlined to mitigate the dimensionality problem: the first possibility is reducing the total number of features and

[1]Different users could label the same song differently (e.g. one that pays more attention to the lyrics may notice nuances which are different from those detected by a user focusing on the audio track).

datasets deemed essential for the model, whereas the second option consists of preserving all the features and operating dimensionality reduction to compress the total information. This thesis proposes a hybrid approach: after investigating the literature on the topic to highlight only the strictly relevant information for the task, dimensionality reduction is applied to one single final dataset obtained by concatenating the features resulting from the literature review.

The main reference for the selection of features that are specifically relevant to the music emotion recognition task is [37]; this work provides an overview of emotionally relevant musical features leveraging the eight fundamental musical dimensions: melody, harmony, rhythm, dynamics, tone color, expressivity, texture, and form. For each dimension, the authors list some relevant quantitative estimators, which guide the selection of features from the corpus of data of the Music4All-Onion dataset; the process and resulting features are presented in Table VIII and Table IX. The final dataset, obtained by joining the selected features of the *essentia* dataset with the *emobase* and BLF datasets, is composed of a total of 2.441 features for 33.694 samples[1].

Once the dataset is obtained, dimensionality reduction techniques alternative to the PCA can be applied to it. One characteristic of the PCA that makes it unsuitable in some cases is its linearity, which limits its effectiveness on data with complex nonlinear relationships. To improve the dimensionality reduction results obtained with the baseline method in Section 5.2

---

[1]The slight decrease in number of samples is caused by discrepancies across the datasets.

it is thus necessary to explore nonlinear dimensionality reduction tools, such as those presented in the following paragraphs: t-SNE, UMAP, or autoencoders.

The t-Distributed Stochastic Neighbor Embedding (t-SNE) [80] is a nonlinear method that operates dimensionality reduction by preserving the local structure of data, namely the local relationships between points. It was introduced to aid the visualization of high-dimensional data by mapping it onto a two or three-dimensional space. It works using a probabilistic approach to model the pairwise similarities between points in the original high-dimensional space using a Gaussian distribution, whereas the pairwise similarities between points in the low-dimensional space are modeled using a Cauchy distribution. The final step consists of minimizing the differences between the two distributions. Despite its ability to capture nonlinearity in data, one open issue is the inability to preserve the global structure of data, along with the local one.

The Uniform Manifold Approximation and Projection (UMAP) [81] technique improves t-SNE by capturing both the local and the global structure of data; it was introduced not with the aim of data visualization but as a dimensionality reduction method to be included in machine learning pipelines. Therefore, great attention is being paid to the computational complexity issue. UMAP leverages a graph built from high-dimensional data, where connections between nodes (points) are weighted according to the likelihood of them being connected. The connection between two points is modeled as follows: first, a radius is extended outwards from each point; the selection of the radius length is determined according to each point's distance from its n nearest neighbors. If radii from two different points meet, then the likelihood of the points being connected is computed based on the radius value: the more it grows, the less likely

the connection is, leading to the creation of a fuzzy set of edges. Once the process is performed on each point, each point is forced to be connected to at least its closest neighbor, ensuring the preservation of the local structure. Finally, a low-dimensional graph is built to be as similar as possible to the high-dimensional one, minimizing the cross-entropy loss between the two.

Because of its ability to preserve both the global and the local structure, and thanks to its computational efficiency[1], UMAP is preferred as a dimensionality reduction technique over t-SNE in this work.

Another dimensionality reduction tool tested is autoencoders [82], a type of neural network used to learn efficient data representations, often used for dimensionality reduction. An autoencoder is composed of two main components: an encoder and a decoder; the encoder is, in turn, formed by a set of layers of decreasing sizes, able to gradually reduce the dimension of the input data until it reaches the latent space[2]. The decoder then performs the inverse process, leveraging a set of layers of increasing sizes that reconstruct the original data from the latent space representation [83]. The network is trained to minimize the reconstruction error, which quantifies the difference between the original and reconstructed data, allowing to obtain a reliable low-dimensional representation.

---

[1]The authors report lower runtime for UMAP thanks to its non-use of global normalization, possible thanks to the use of fuzzy graphs instead of probability distributions.

[2]Layer in which data is mapped to the desired lower-dimensional representation, with dimensions corresponding to the number of neurons in the latent space.

For this task, the autoencoder architecture is formed by nine layers each for the encoder and decoder; the outermost ones have as many neurons as the original dataset's dimensions, and the following layers going inwards have each half the neurons of the previous one; this setting allows the latent space to only have two dimensions, in order to make the results comparable with those obtained via UMAP dimensionality reduction. The network is trained for 100 epochs[1], and the metric used to evaluate its performance is the mean square error (MSE). An important note concerns the complexity of the network: the proposed architecture is intentionally shallow to provide a general evaluation of the methodology. As discussed further in the following paragraphs, it does not achieve the same dimensionality reduction level as UMAP, but, even when tested on a simpler task, the reconstruction errors are high, therefore excluding autoencoders from further exploration in this context.

Different configurations are tested for both UMAP and the autoencoder; in the case of UMAP, the parameters that need to be set and the tested values are:

- n_neighbors: the number n of closest neighbors to consider for each point; $[5, 20, 100]$.
- min_dist: the minimum distance that points are allowed to have in the low-dimensional representation; $[0.0, 0.33, 0.67]$.
- metric: the metric used to compute distances between points; [euclidean, correlation].
- n_components: the number of dimensions in the compressed representation; $[2, 3]$.

---

[1]Occasionally, training has been carried out for 200 epochs with no significant improvement.

It is worth reporting that different metrics were tested, such as the Camberra or the Mahalanobis distance, but consistently bad results were obtained during the first trial runs and therefore they have been purposely excluded from further tests.

As for what concerns the autoencoder architecture, different configurations have been tested as well, varying optimizer, learning rate, decay strategy, batch sizes, and scheduling strategy. The final combination leading to the best performance combines a scheduler with *Adam* optimizer with a starting learning rate of 10^{-3} and a decay rate of 0.95 every 850 steps, and a batch size of 32.

6.1.2 Clustering

Once the dimensionality issue in audio data is addressed, it is now possible to test some clustering algorithms on the low-dimensional data obtained via UMAP.

Before proceeding, a visual test using the labels for the lyric component is conducted; the objective is to identify potential patterns in the unlabeled audio data and assess if there is any correlation between the emotions encompassed in the textual data and the audio features. In practice, this consists of plotting all the samples associated with each emotion using their two-dimensional representation. Figure 35 presents the results, showing no particular correlation between the emotional labels associated with the lyrics and the audio features. Once this scenario has been excluded, it is possible to proceed with the clustering experiments.

Each clustering experiment uses a different algorithm and is composed of two stages. The first one is one in which different parameter configurations are tested to select the better-performing one, which is then used in the second step to label the data and plot the obtained

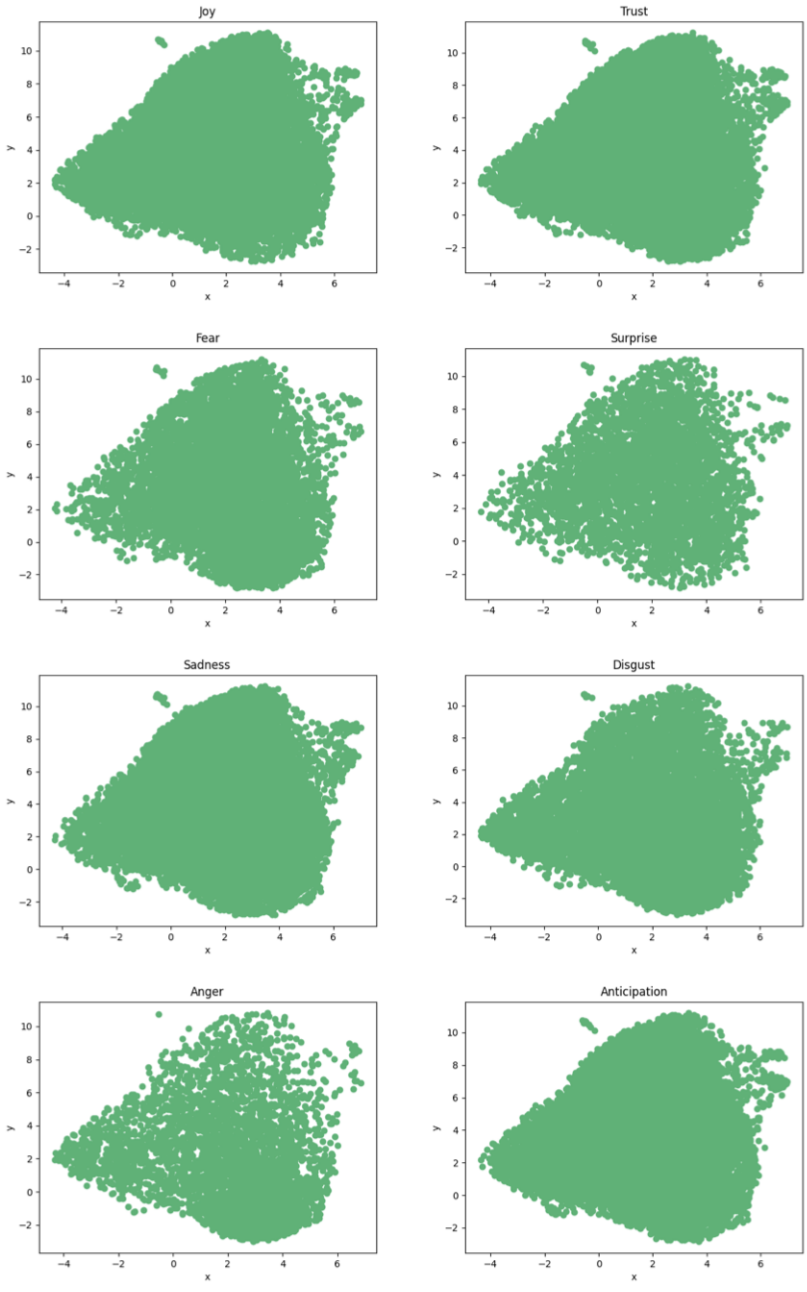

Figure 35: Distribution of samples per emotional label in the lyrics dataset.

clusters. The clustering algorithms tested are chosen according to the type of clustering performed[1] to maximize the likelihood of identifying meaningful patterns in data.

The first algorithm used is K-Means, a proximity-based clustering technique described in Section 5.2. The primary parameter that needs to be tuned is the number of clusters to group the data into, which is chosen among [2, 4, 8]; while eight would ideally align with the emotional classes used for lyrics classification, additional experiments with 2 and 4 clusters are conducted to explore the possibility of a simpler audio model. The evaluation is performed leveraging the silhouette score defined in Equation 5.4, Section 5.2, and two additional metrics: the Calinski Harabasz score and the Davies Bouldin score. The Calinski Harabasz score [84] is an internal metric having values ranging from 0 to infinity and computed as described in Equation 6.1, leveraging the within-cluster dispersion (WCSS) defined in Equation 6.2 and the between-cluster separation BCSS defined in Equation 6.3; n is the total number of points, k is the total number of clusters, n_i is the number of points in the cluster C_i having centroid c_i, and c is the centroid of the data distribution. Higher values of the Calinski Harabasz score correspond to better clustering structures.

$$CH = \frac{\frac{BCSS}{1-k}}{\frac{WCSS}{n-k}} \tag{6.1}$$

$$BCSS = \sum_{i=1}^{k} n_i ||c_i - c|| \tag{6.2}$$

---

[1]Proximity-based, density-based, hierarchical, graph-based.

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - c_i|| \tag{6.3}$$

The Davies-Bouldin score [85] is another internal clustering evaluation metric ranging from 0 to infinity and is computed as reported in Equation 6.4, leveraging Equation 6.5 and $R_{i,j}$ Equation 6.6 representing how good a clustering scheme is, where $M_{i,j}$ is the separation between cluster i and cluster j, and S_i is the within-cluster scatter for a cluster i. Lower values of the Davies Bouldin score indicate a better clustering structure.

$$DB = \frac{1}{N} \sum_{i=1}^{N} D_i \tag{6.4}$$

$$D_i = max_{j \neq i} R_i, j \tag{6.5}$$

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}} \tag{6.6}$$

The results of the parameter tuning procedure on K-Means are presented in Figure 36; it is crucial to report that the Calinski Harabasz score had values of the order of magnitude of 10^4, and they have therefore been scaled such that the maximum value is equal to 1 for visualization purposes; the scaling of the Calinski Harabasz score will be performed for every clustering algorithm evaluation from now on. The silhouette score has comparable results for 2

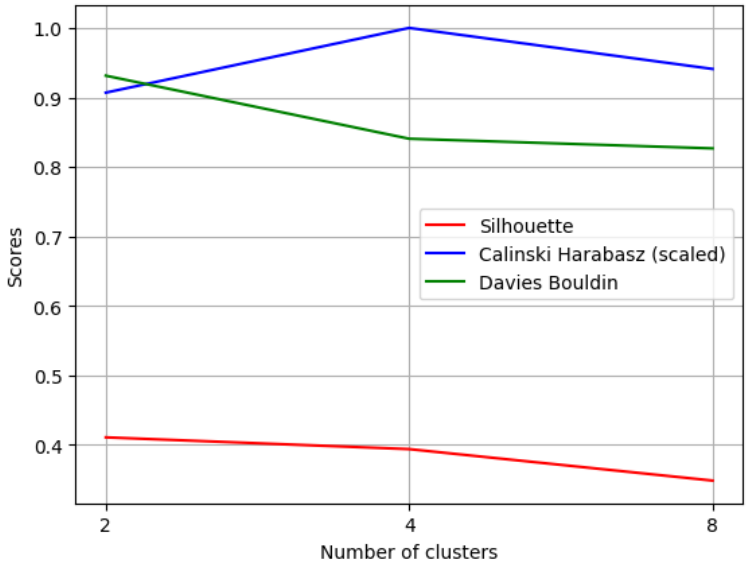

Figure 36: Evaluation metrics over different numbers of clusters for K-Means.

or 4 clusters, similarly to the Davies Bouldin score for 4 and 8 clusters; the number of clusters is therefore set to 4, which is the setting corresponding to the maximum Calinski Harabasz score.

The second algorithm tested is DBSCAN, a density-based clustering algorithm that does not require to specify the number of clusters a priori, but instead works by clustering together points that are densely close to each other. Specifically, it requires setting a radius *eps* and a minimum number of points to consider *min_points*: if a certain point has more than *min_points* at a distance lower than *eps*, then it is labeled as a *core point*. If a point has instead less than *min_points* in the neighborhood created by *eps* but is in the neighborhood of a core point, it is labeled as a *border point*; all other points are *noise points*. A remarkable advantage of

DBSCAN, in addition to not having to specify the number of clusters, is its ability to handle clusters of complex shapes.

The parameters that need to be tuned are *eps*, which is chosen from $[0.3, 0.6, 0.9]$, and *min_points*, chosen from $[5, 10, 20]$. Figure 37 shows the metrics values for different configurations. The graph shows default values for the metrics in the case of *eps* equal to 0.9 because it leads to a single cluster being found. Since the results don't change significantly when *eps* is equal to 0.6, the final configuration arbitrarily chosen combines a *eps* of 0.6 and 20 *min_points*.

HDBSCAN is a variation of DBSCAN that converts it into a hierarchical clustering algorithm to improve its ability to identify density-based clusters having different densities. The parameters that need to be tuned and the tested values are:

- min_cluster_size: minimum number of samples per cluster; [50, 100, 200].
- min_samples: minimum number of samples to consider when computing each point's distance to its closest neighbors; [5, 10, 20].
- cluster_selection_epsilon: minimum distance threshold to observe when merging together close clusters; [0, 0.3, 0.6, 0.9].

Additionally, the algorithm is forced to return at least two clusters to provide more meaningful results and to use the excess of mass (eom) algorithm to identify the most persistent clusters. The average metrics corresponding to different values of the parameters to tune are presented in Figure 38; since the value of epsilon does not seem to yield significantly different results, its final value chosen arbitrarily to be 0.9, with a minimum cluster size of 50, and a minimum number of samples for closest neighbors computation of 5.

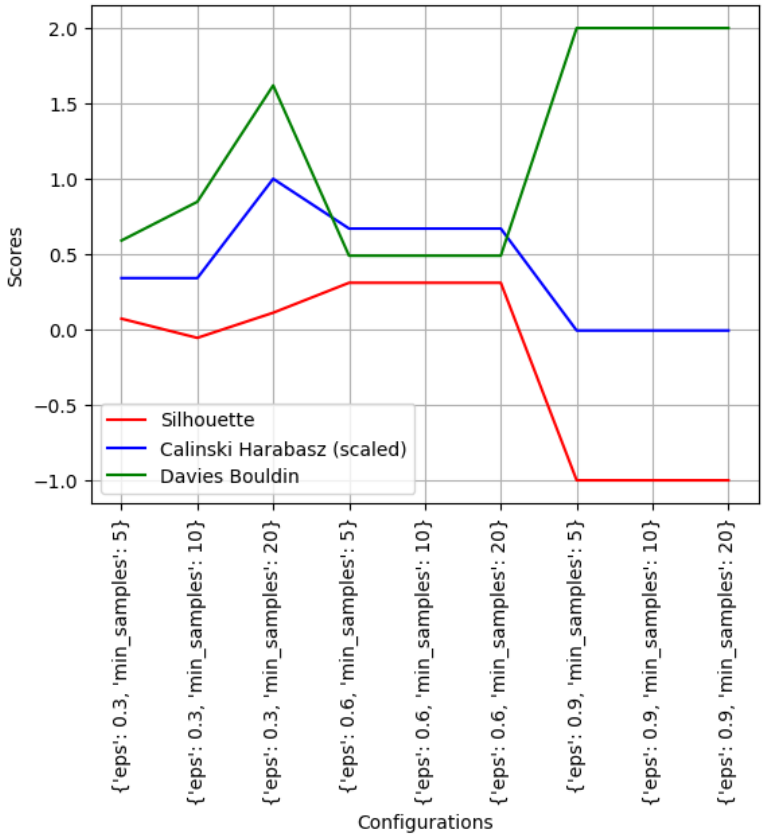

Figure 37: Evaluation metrics over different configurations for DBSCAN.

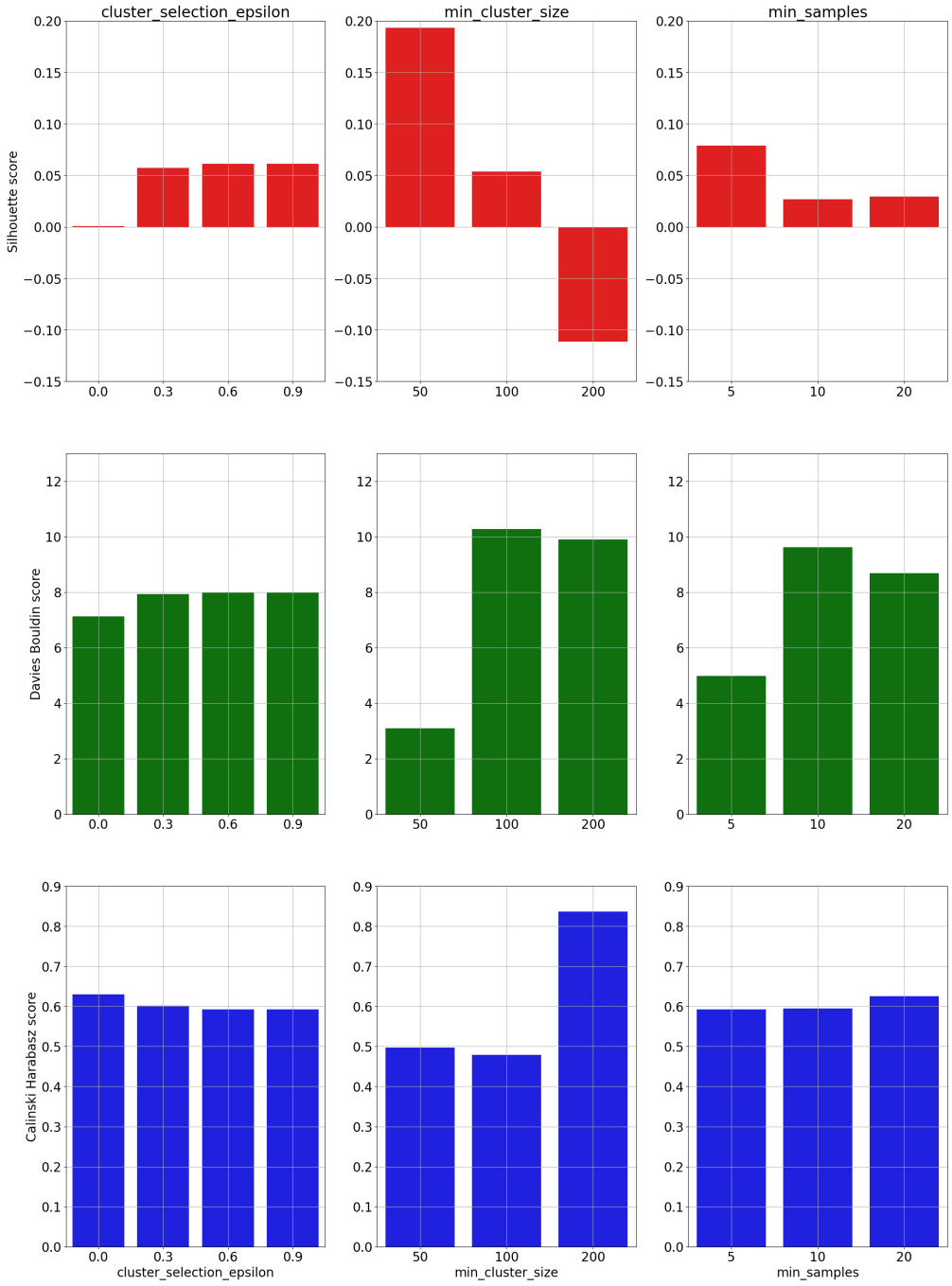

Figure 38: Average evaluation metrics over different parameter values for HDBSCAN.

The final algorithm tested is spectral clustering, which is based on the computation of a similarity matrix and the corresponding similarity graph to model the data distribution: the nodes of the graph represent the data samples, and the edges are weighted according to the similarity between the two points they connect, with similarities below a certain threshold leading to the edge being discarded. The base idea behind spectral clustering is grouping together points that belong to the same connected component of the similarity graph. This is implemented by projecting the normalized Laplacian matrix obtained from the similarity matrix onto an n-dimensional space, where n is the number of clusters to be obtained. The parameters that need to be tuned and the tested values are:

- n: the number of clusters; [2, 4, 8].
- n_neighbors: number of neighbors to consider for the affinity matrix; [5, 10, 20].
- assign_labels: strategy for label assignment; [kmeans, cluster_qr[1]].

The metrics values over the different configurations are shown in Figure 39; the final configuration chosen is the one using cluster_qr as a strategy to assign the labels with no additional parameters specified other than the affinity metrics.

---

[1]Clusters are extracted directly from the eigenvectors of the Laplacian matrix, representing the connected components in the graph; if no parameters are specified, it can automatically identify the number of clusters based on the number of connected components.

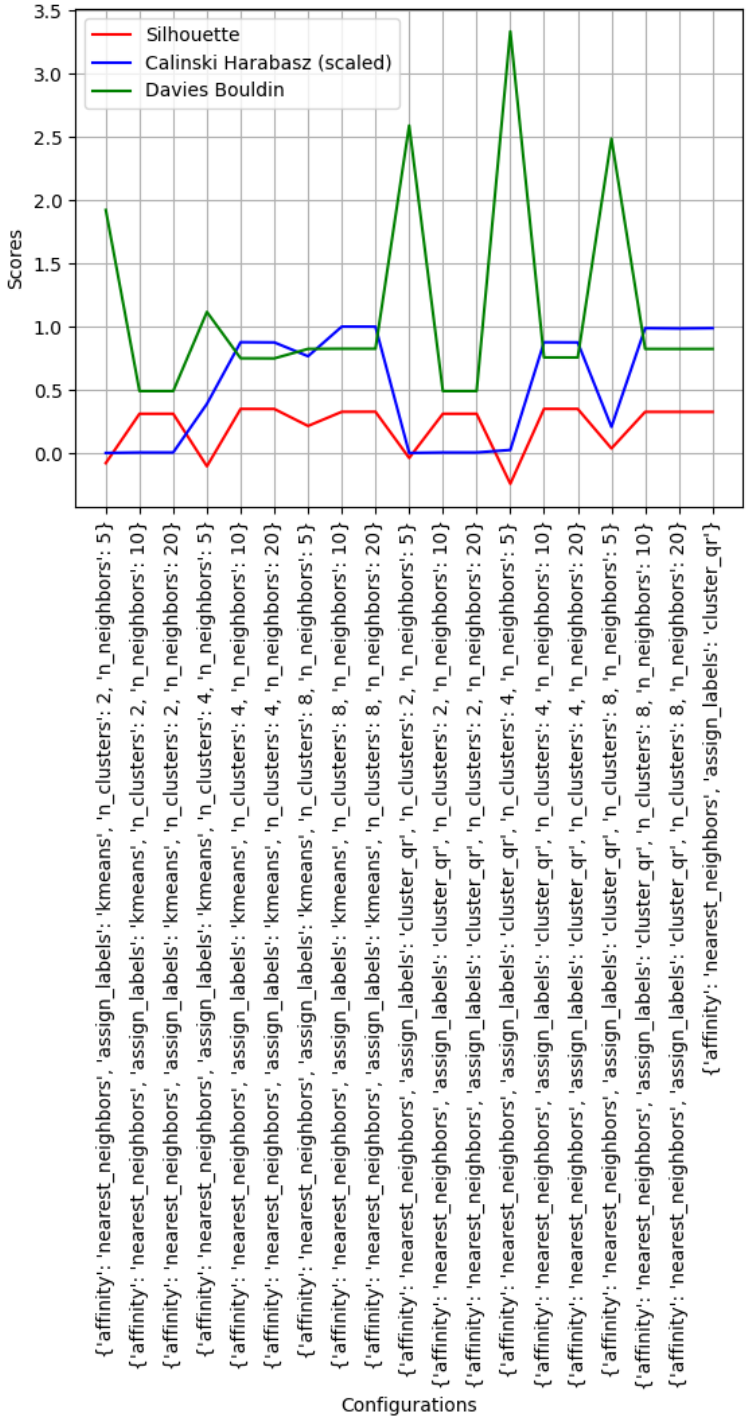

Figure 39: Evaluation metrics over different configurations for spectral clustering.

6.2 Multimodal Classification

6.2.1 Supervised Audio Classification

For the final classification task, the labels obtained through clustering are initially leveraged as an auxiliary tool for the supervised classification task in a similar manner to the approach described in Section 2.1.4 for [30]. The motivation behind this choice stems from the use of the UMAP dimensionality reduction technique: despite its strong performance in the tackled task, its use poses a two-fold problem.

First, the way high-dimensional data is mapped into a two-dimensional space makes the obtained representation meaningful only when considered in the context of the complete data distribution; in other words, the mapping is not absolute, but it depends on the global and local relationships between samples. Consequently, if the data distribution changed or if a new distribution was introduced, the two-dimensional representation on which clustering is performed would lose meaning.

Second, UMAP does not provide information on how the single features contribute to the low-dimensional mapping of data, unlike, for instance, PCA, which allows for a direct interpretation of the results. This lack of interpretability is due to the non-linearity of the transformations performed by the algorithm, which complicates the understanding of both the structure and the clustering results.

These two challenges, while being the expression of distinct issues, are intrinsically linked to the same underlying cause: the lack of a direct correlation between objective and invariant audio descriptors and the clustering structure. A way to tackle both problems thus involves

attempting to find correlations between the original, high-dimensional data and the cluster labels. Depending on the specific aspect to address, different yet complementary approaches can be employed.

Starting from the first challenge regarding the limited validity of the low-dimensional representation, the objective is to find a connection between the obtained cluster labels and absolute, distribution-invariant data features, such as the original ones composing the final emotional audio dataset described in Section 6.1.1. One potential solution is to train a classifier that is able to predict the cluster labels assigned to the two-dimensional mapping directly from its original high-dimensional representation. However, this poses the same issues discussed in Section 3.2.2 concerning the curse of dimensionality and the challenges of working with such high-dimensional data. Furthermore, training a classifier on high-dimensional data may not resolve the second issue of explainability, especially when using deep networks as classifiers: while they may provide a functional mapping from the original space to the clustering results, it does not necessarily offer insight into why specific features contribute to a given classification.

To gain insight into the relationships between the clustering labels and the original features before dimensionality reduction, approaches exploring the correlation between each feature and the target label can be employed: mutual information computation and feature importance extraction.

Mutual information quantifies the dependency between two variables and is able to capture both linear and non-linear relationships. In this scenario, it can compute the impact of each individual feature on the structure identified by the spectral clustering algorithm [86]. Features

having a higher mutual information score are strongly associated with the clustering labels, suggesting they might be meaningful descriptors of the data, whereas variables having null mutual information are statistically independent.

Feature importance analysis follows a slightly different strategy: instead of investigating the correlation between features and labels directly, it leverages a supervised learning approach. Training a classifier to predict the labels from the high-dimensional data allows later extraction of the feature importance scores to determine which attributes gave the greatest contribution to the classification task. One of the models allowing this is the Extreme Gradient Booster classifier (XGBoost) [87], which implements gradient-boosted decision trees. The base concept behind it is similar to a random forest classifier, since they both employ multiple decision trees following an ensemble learning approach, but while random forests leverage bagging to obtain multiple decision trees working in parallel on bootstrap samples of data, XGBoost uses boosting. Instead of training trees in parallel, they are trained sequentially so that each estimator can focus on correcting the mistakes of the previous ones. Additionally, each tree is assigned a weight based on its contribution to lowering the overall loss, different from random forests, which average predictions from the parallel trees or employ voting schemas. XGBoost is a robust classifier that is able to handle missing data, noise, and a large number of features.

By combining mutual information analysis with feature importance from XGBoost, a more comprehensive understanding of the relationships between high-dimensional features and clustering labels can be obtained. While mutual information provides insight from a statistical perspective, feature importance ranks features using their contribution to the prediction per-

formance, providing insight into the attributes that influence the classification outcome more significantly.

It is worth noting the overlap between the strategies suggested for addressing the validity and the explainability issues, as both rely on the need for a classification model to connect the original features with the clustering label. This further proves the interdependence of the issues to tackle and the complementarity of the solution. In fact, the approaches are combined to maximize interpretability and explainability, allowing the extraction of more meaningful insights from the clustering results.

To further increase the comprehension of the analysis, the emotional audio dataset is split into two separate parts; this choice is motivated by the two different natures of the features, which can either be descriptive of the song as a whole[1], or of a section of the song[2]. These two datasets will be referred to as the general dataset and the block-level dataset, respectively. The general dataset is composed of 56 features, excluding the dimensionality issues repeatedly discussed in this thesis, whereas the block-level dataset contains the remaining 2385 attributes.

First, an attempt at training classifiers to predict the clustering labels from the original audio features is performed, starting from the general dataset. Since, as previously discussed, it is crucial to choose a model able to capture non-linear relationships between features and target, a multi-layer perceptron classifier (MLP) is leveraged for this task. This model is a

---

[1]e.g., average loudness, dissonance, entropy.

[2]e.g., block-level features, MFCCs, F0.

simple type of artificial neural network composed of an input and an output layer separated by a number of hidden layers of customizable dimensions, all fully connected to each other.

After experimenting with different numbers of hidden layers of various dimensions, the best results are obtained with a simple architecture with one hidden layer composed of 30 neurons. The train component of the 56-dimensional general dataset, containing 80% of the data, is fed to the network to train it to predict the cluster labels; after 200 iterations, the model is used to predict the labels on the remaining test samples.

Another tool than can be leveraged for the same purpose of increasing interpretability is the *lime* Python library [88], developed with the purpose of explaining machine learning classifiers and their predictions. It implements a local linear approximation of the model's predictions to explain its behavior: once an instance is provided for prediction, it is slightly modified to build a linear model around it that is able to approximate the model's decision boundaries locally. While the global structure may be different, this tool provides local approximate explainability of the classification process, mitigating the tackled issue.

The *lime* library is used with the multi-layer perceptron classifier built for the general dataset to provide insights on the features that contribute most to the predicted outcome. Once the classifier has been trained, it is fed to the explainer of the *lime* library with a sample from the train set[1], and a graphical explanation of the features that contribute the most to the prediction is provided.

---

[1]The train set is used instead of the test set to ensure that the model outputs a correct prediction and guarantee reliability of the results.

While, as already said, the global decision boundaries of the model remain hidden, this still provides some notions for specific samples and can be leveraged as a qualitative indicator of which features contribute the most to the output prediction.

Furthermore, by referencing the same literature source used in Section 6.1.1 to identify the emotionally relevant features, it is possible to infer some insights on the overall emotional content of songs based on the feature values. For instance, [37] suggests that a pronounced rhythmic pattern generally corresponds to more intense emotions, such as joy or anger, and that songs in major keys are typically associated with positive emotions; consequently, a song characterized by a strong rhythmic pattern in a major key is likely to evoke a joyful sensation in listeners. Unfortunately, the data and the tools available at this moment prevent such a straightforward interpretation, but an effort is made anyway to extract some general descriptors for each class and offer preliminary insights on the potential emotional implications.

Because of the promising techniques highlighted for this purpose, an attempt is performed on the block-level dataset as well. The expectations for training a multi-layer perceptron on such high-dimensional data are considerably lower due to both the potential curse of dimensionality and the risk of overfitting, and to the computational complexity. Furthermore, repeating the same feature relevancy discovery process implemented for the general data is tricky because of the nature of the dataset: since it is composed of features describing different sections of songs, the importance of a feature may vary depending on the segment being analyzed, and different sections may be considered important according to different evaluation strategies. In order to

mitigate this, the more impacting features will be handled considering only what they measure and not the index of the associated section.

6.2.2 Final Multimodal Classifier

The model developed to tackle the multimodal classification task draws inspiration from those set as baselines in Section 5.1, leveraging the strong performances achieved on lyrics data.

While the convolutional architecture outperformed the two leveraging recurrence on text classification, it is expected that incorporating audio data into the model increases the task complexity, requiring architectures able to handle temporal and spectral interdependencies.

The goal is to create an architecture combining convolutional layers and recurrent layers, implemented as bidirectional LSTMs: the will to include both elements is justified by the attempt to develop a model able to perform more informed predictions, leveraging local patterns captured by the convolutional layers and broader contextual awareness provided by the bidirectional LSTM. Despite the absence of a well-defined temporal dimension in the used datasets, this architecture has the potential to yield good results on real-world audio data, where a more structured temporal component should be present.

The proposed architecture is composed of three parallel convolutional layers that first receive the input vectors and process them with the same number of filters but different kernel sizes; this aims at building a model able to capture patterns of varying dimensions and longer dependencies.

Once the convolutional layers have processed the input, it is necessary to define the order of the following layers: the bidirectional LSTM and the global pooling layer. The first option consists of performing max pooling on the output of the convolutional layers, forcing the pooler to maintain a unit-rank temporal dimension for compatibility with the LSTM. The second option places the LSTM directly after the convolutional layer to better leverage the extracted information, and max pooling is performed on the output of the bidirectional LSTM instead.

After the pooling and LSTM layers have processed the output of the convolutional layers in the selected order, the three parallel blocks consisting of all three elements each are concatenated, passed through a dense layer with a ReLU activation function, a dropout layer to avoid overfitting and, finally, through the output layer.

Multiple runs are conducted to evaluate the best model composition, testing the performance of various configurations across different types of data; this approach provides valuable insights into the interplay between different model compositions and the two different modalities.

During the evaluation of models, a key observation was the underperformance of the architectures tested on the audio datasets, with results shown in Table X. On the one hand, the general features dataset yielded results comparable to those obtained with the multi-layer perceptron architecture, averaging an F1 score of 57%, but at a higher computational cost due to the complexity of the convolutional-recurrent architecture. On the other hand, the architecture tested on the block-level dataset yielded results significantly worse than those obtained with the MLP, averaging an F1 score of just 51%. These results prove once again how the type of audio data available is not suitable to be processed by such a complex architecture, and that the

patterns and context that can be extracted from them are not sufficient to capture meaningful relationships that would justify the added complexity of convolutional and recurrent layers.

As a result, an alternative solution was developed: given that the optimal results in audio classification are achieved by a multi-layer perceptron, a relatively simple yet effective architecture with low computational cost, the multimodal architecture is revised accordingly to align with these findings.

To incorporate this result in the final architecture, a hybrid model consisting of two sequential components is designed. First, a dataset including both the lyrics and the emotional features of each sample is built. Once the data is fed into the model, it is split into different inputs: lyrics and audio features. The audio features are processed by a multi-layer perceptron (MLP) to predict the cluster indexes for each sample, leveraging the high reliability of this approach, as proved by the results achieved in Section 6.2.1, Figure 65 and Figure 52.

Once the cluster labels are obtained, they are incorporated into the lyrics dataset as an additional auxiliary feature. The enriched dataset is now processed by the final model, where the lyrics are first converted into embeddings and subsequently analyzed through the three parallel convolutional-recurrent blocks. After the lyrics are processed, the outputs of these blocks are concatenated with one another and with the cluster labels obtained with the MLP. The concatenated representation is then passed through the final dense layers, which generate the definitive multi-label emotional prediction for the samples.

It is important to notice the reduced role of audio features in the classification task, as their inclusion did not enhance the performance of the complex architecture. Rather than

serving as a crucial component of a multimodal system, the audio data in this setting serves an auxiliary purpose, contributing to the classification of samples without fundamentally altering the outcome in order to preserve the role of textual data, which is the most valuable element to the performance in this setting. While this implementation does incorporate both modalities, it is not a fully multimodal model in the strict sense, as the primary decision-making process is mainly driven by textual data. Given the limitations of the available audio features, this is a practical compromise that ensures the best possible results without introducing unnecessary complexity or compromising the performance of the classifier.

This section detailed the setup of the different experiments conducted to achieve the final multimodal architecture. Starting from the baselines defined in Section 5, new models and approaches have been introduced, providing important insights into the role of the two different modalities by initially analyzing them separately. The results of the described experiments are presented and discussed in the following chapter, Section 7.

TABLE VIII: LITERATURE-GUIDED FEATURE SELECTION PROCESS FROM MUSIC4ALL-ONION CORPUS (1)

Musical Dimension	**Estimators**	**Matching Feature(s)**	**Dataset**
Melody	f0	*	emobase_f0
Harmony	hpcp	hpcp.mean, hpcp.stdev	essentia
	tuning frequency	tuning_frequency	essentia
	key	key_krumhansl.strength, key_temperley.strength	essentia
	modality	tuning_diatonic_strength	essentia
Rhythm	beat spectrum	bpm_histogram_first_peak_bpm, bpm_histogram_first_peak_weight, bpm_histogram_second_peak_bpm, bpm_histogram_second_peak_weight	essentia
	onset	onset_rate	essentia
	pulse	beats_count, bpm	essentia
Dynamics	RMS energy	spectral_rms.mean, spectral_rms.stdev	essentia
	loudness	average_loudness, loudness_ebu128.loudness_range, loudness_ebu128.short_term.mean, loudness_ebu128.short_term.stdev	essentia
Expressivity	average silence ratio	silence_rate_20dB.mean, silence_rate_20dB.stdev, silence_rate_30dB.mean, silence_rate_30dB.stdev, silence_rate_60dB.mean, silence_rate_60dB.stdev	essentia
Texture	musical layers	NA	NA
Form	structural information	NA	NA
Bonus	vocal features	*	emobase_voice

* Whole dataset.

TABLE IX: LITERATURE-GUIDED FEATURE SELECTION PROCESS FROM MUSIC4ALL-ONION CORPUS (2)

Musical Dimension	**Estimators**	**Matching Feature(s)**	**Dataset**
Tone Color	zero-crossing rate	zerocrossingrate.mean, zerocrossingrate.stdev	essentia
	spectral flatness, crest factor	spectral_strongpeak.mean, spectral_strongpeak.stdev, spectral_decrease.mean, spectral_decrease.stdev, spectralcontrastvalleys.mean, spectralcontrastvalleys.stdev	essentia
	spectral entropy	spectral_entropy.mean, spectral_entropy.stdev	essentia
	spectral flux	spectral_flux.mean, spectral_flux.stdev	essentia
	spectral rolloff	spectral_rolloff.mean, spectral_rolloff.stdev, barkbands.mean	essentia
	energy in Mel, Bark, ERB bands	barkbands.mean, barkbands.stdev, melbands128.mean, melbands128.stdev, erbbands.mean, erbbands.stdev, erbbands_spread.mean, erbbands_spread.stdev, barkbands_spread.mean, barkbands_spread.stdev, melbands_spread.mean, melbands_spread.stdev	essentia
	MFCC	*	emobase_mfcc
	lsp	*	emobase_lsp
	spectral contrast	*	blf_spectralcontrast
	sensory dissonance	dissonance.mean, dissonance.stdev	essentia

* Whole dataset.

TABLE X: PERFORMANCE OF THE CONVOLUTIONAL-RECURRENT ARCHITECTURE ON PREDICTING CLUSTER INDEXES FOR THE AUDIO DATASETS

Metric	**Audio (general dataset)**	**Audio (block dataset)**
Precision	0.46	0.35
Recall	0.76	0.91
F1-score	0.57	0.51
Hamming loss	0.21	0.25

CHAPTER 7

EXPERIMENTAL RESULTS AND DISCUSSION

This chapter presents the results obtained with the models discussed in Section 6 and provides an overview of the experimental setups implemented.

7.1 Unsupervised Audio Modality

7.1.1 Dimensionality Reduction

This section specifically presents the results obtained with the implementation of the techniques described in Section 6.1.1.

The final UMAP configuration chosen for the dimensionality reduction task on audio data is, as outlined in Section 6.1.1, the one using the Euclidean distance, 100 as the value of the n_neighbors parameters, 0.67 as the minimum distance, and two final components for the low-dimensional representation. This configuration yields a reconstruction MSE of 0.67. Figure 40 shows the data projected onto the final two dimensions, whereas Figure 41 shows it onto three dimensions, keeping all other parameters unchanged, to verify that no important information is lost with the simpler representation.

Then, the autoencoder architecture is evaluated using the final configuration that combines a scheduler with *Adam* optimizer having a starting learning rate of 10^{-3} and a decay rate of 0.95 every 850 steps and a batch size of 32.

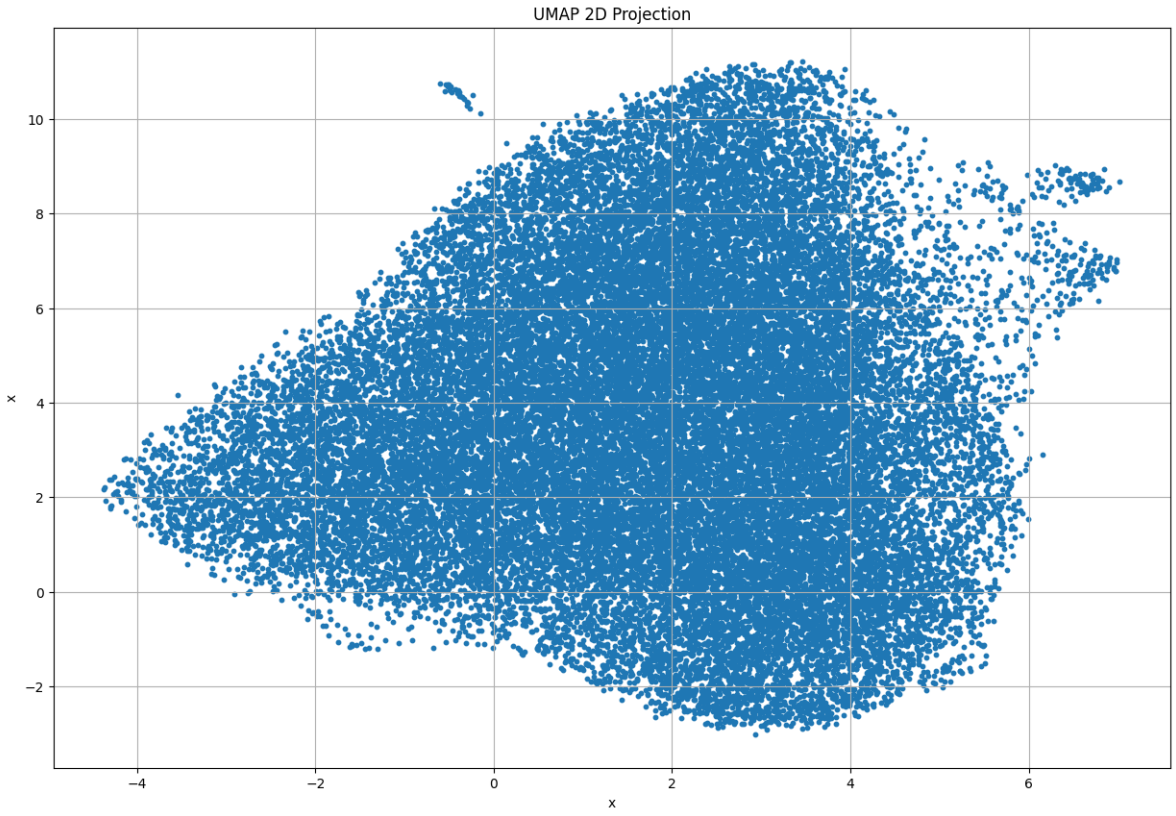

Figure 40: Data after being reduced to two dimensions using UMAP.

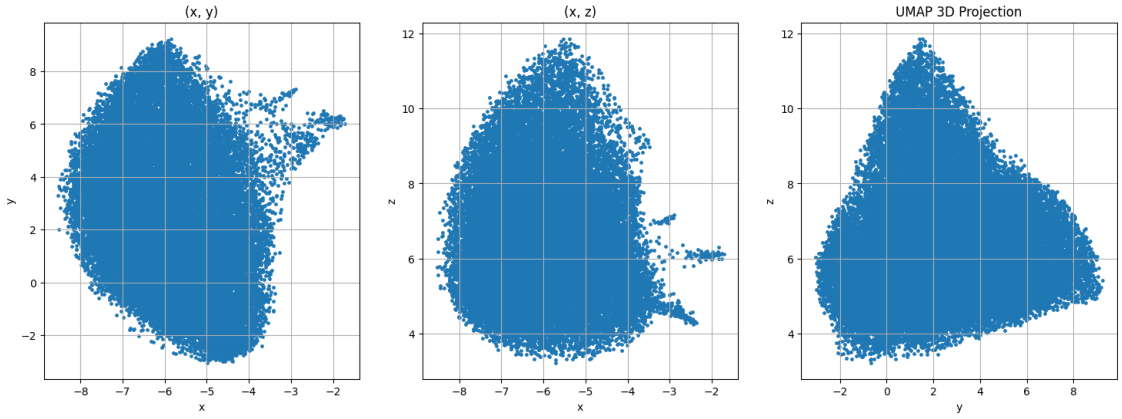

Figure 41: Data after being reduced to three dimensions using UMAP.

After training the model for 100 epochs with the goal of minimizing the reconstruction error, the final mean value of the reconstruction MSE across all features is 2.8×10^5, remarkably larger than the one obtained by UMAP: further investigation shows an imbalance in MSE for different features, with some having really high reconstruction errors and the majority having errors comparable the UMAP case. Figure 42 shows the reconstruction MSE per feature obtained after removing the 204 features having a mean square error over 10^3 for visualization purposes. Among the 2.237 features included in the graph, 1.834 have reconstruction errors lower than 1.

A surprising result emerges when plotting the low-dimensional representation obtained with the autoencoder: as shown in Figure 43, all the points collapse to the origin of the axes $(0, 0)$ in the two-dimensional space. This clearly indicates that the selected configuration for the autoencoder fails to learn a meaningful representation of the data and highlights that quantitative

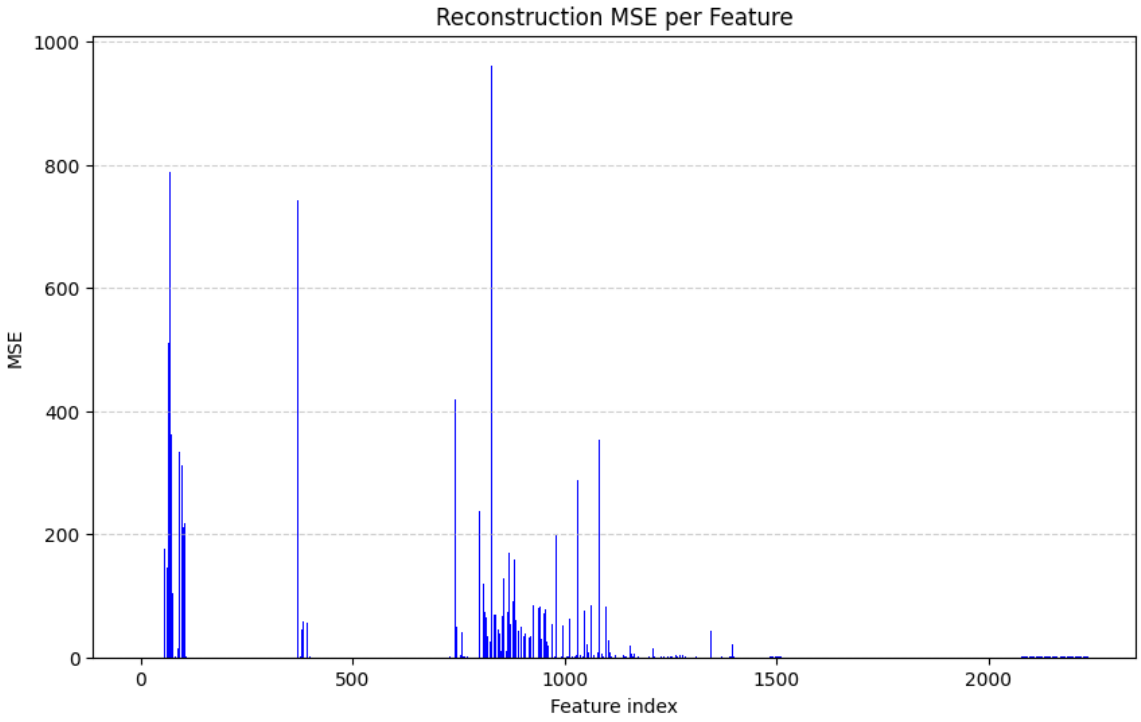

Figure 42: Autoencoder reconstruction mean square error per feature (truncated).

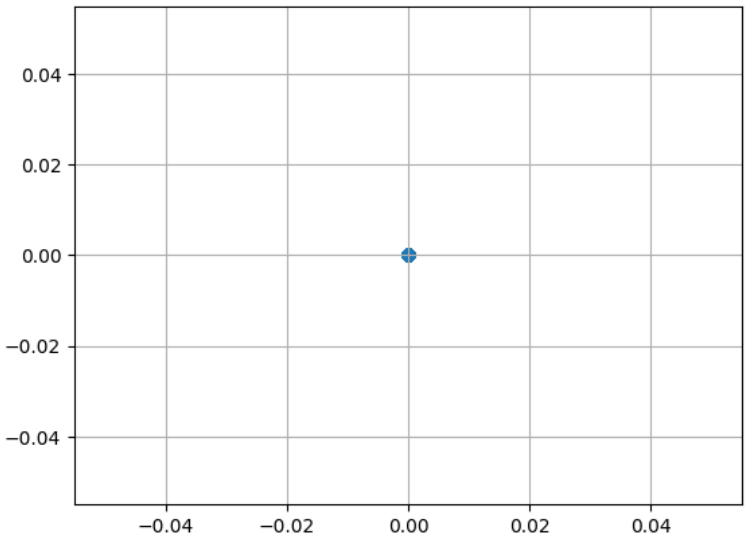

Figure 43: Two-dimensional representation of data obtained with the autoencoder.

metrics alone are insufficient for the evaluation of such techniques. Despite the poor quality of the result, it provides some insights into the imbalance in reconstruction errors. Since the dataset was normalized to have zero mean and unit variance, many features have values close to zero, and thus the reconstruction error in terms of MSE is low for those features and higher for those having values in wider ranges. To exclude that normalizing data is the cause for the poor autoencoder performance, a trial on unnormalized data is performed; Figure 44 presents the results of the test, showing that the autoencoder mapped the data in two linearly dependent components. The outcome indicates that the network struggles to identify meaningful underlying patterns in the data regardless of the normalization strategy.

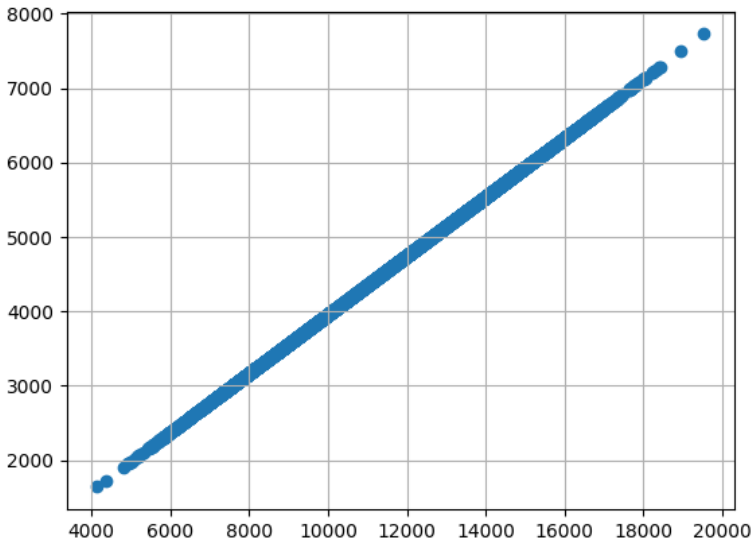

Figure 44: Two-dimensional representation of data obtained with the autoencoder on unnormalized data.

Even disregarding the poor reconstruction quality, UMAP is still preferred for multiple reasons: first, it is remarkably faster than a network that needs to be trained. Second, one drawback typical of neural networks, and thus autoencoders, is the unpredictability of their behavior, as they are considered black boxes, and their operation is opaque [89]. While this is not always a critical flaw, UMAP is favored for its clear mathematical framework, allowing both interpretability and reproducibility without the need for parameter tuning and extensive training.

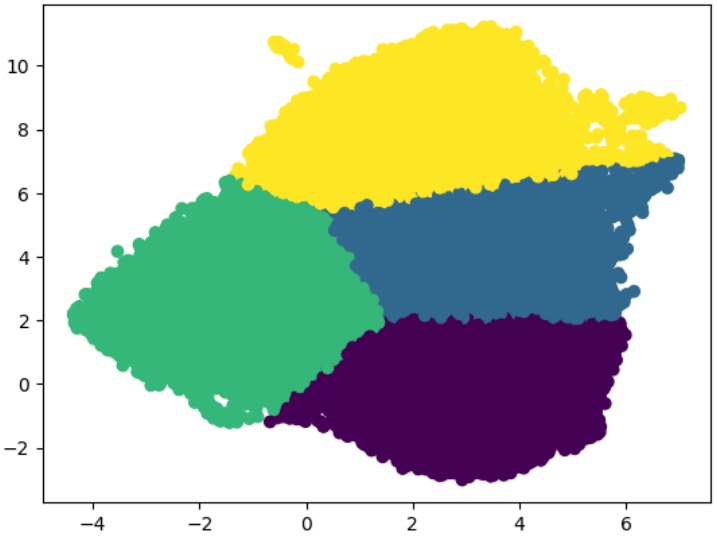

Figure 45: Final clustering structure obtained with K-Means clustering.

7.1.2 Clustering

This section specifically presents the results obtained with the implementation of the techniques described in Section 6.1.2.

After the optimal configuration for each clustering technique has been selected in Section 6.1.2, it is possible to visually present the results in order to compare them.

Figure 45 presents the final clustering structure obtained with the K-Means algorithm using 4 clusters.

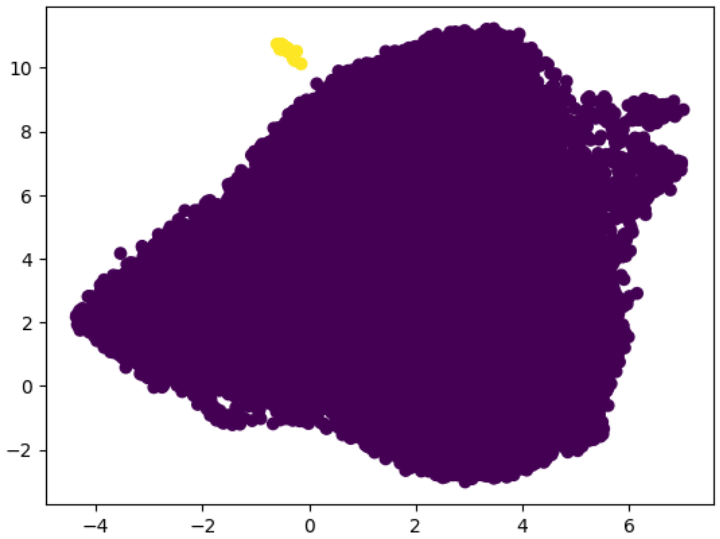

Figure 46: Final clustering structure obtained with DBSCAN clustering.

The final configuration chosen for the DBSCAN clustering algorithm leads to the final clustering structure shown in Figure 46, in which a main cluster contains the majority of points, and a smaller cluster contains the rest.

As for what concerns the HDBSCAN clustering algorithm, the resulting structure is shown in Figure 47, and presents a bigger cluster containing the majority of samples with some border clusters having significantly fewer samples.

The results obtained with the spectral clustering algorithm setting exclusively *cluster_qr* as a label assignment strategy and the nearest neighbors as an affinity measure are presented in

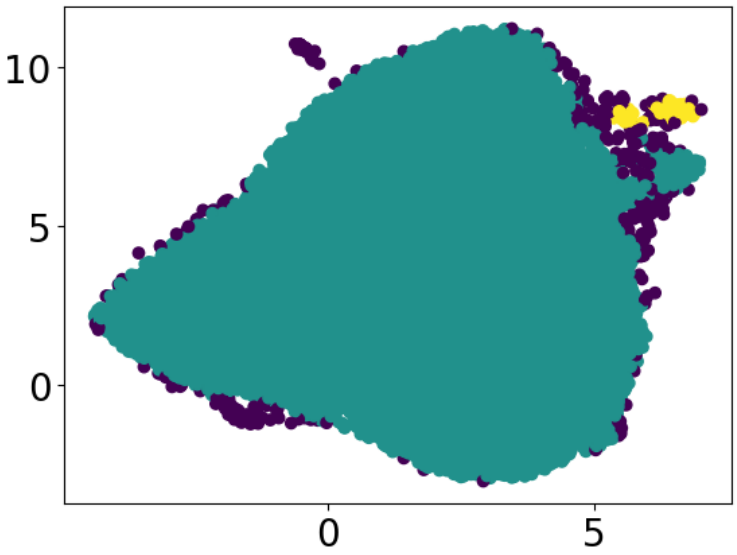

Figure 47: Final clustering structure obtained with HDBSCAN clustering.

Figure 48. The result is different from those of the previous clustering techniques as it presents a structure of 8 clusters.

Since the parameter tuning results for this algorithm indicate the same perforance for the chosen configuration and for the one using K-Means as a label assignment strategy, 8 clusters and 10 or 20 neighbors, this configuration is tested as well and leads to the same results shown in Figure 48.

A comparison of the metrics obtained by the best configuration of each tested algorithm is shown in Figure 49, indicating that the clustering structures obtained with DBSCAN and HDBSCAN are the worst performing ones, with the one resulting from K-Means and spectral

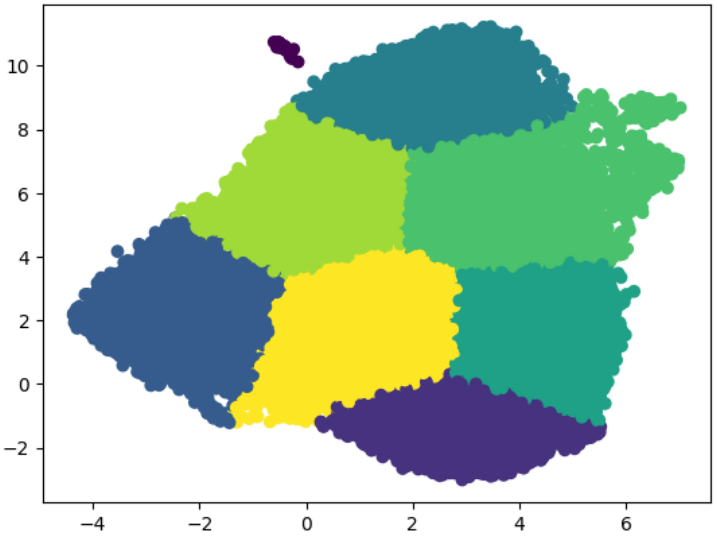

Figure 48: Final clustering structure obtained with spectral clustering.

clustering yielding to better results. Despite the metrics indicating a slightly better performance for K-Means clustering, the final structure chosen is the one obtained through spectral clustering, both because of its meaningful 8-label structure ensuring compatibility with the emotional model used to label the lyrics, and because of the mathematical foundation of spectral clustering, which leverages properties of the affinity matrix and graph to automatically identify the clusters embedded in the data.

The final distribution of the cluster labels is shown in Figure 50; by considering Figure 48, it is possible to notice how cluster 0, having merely 28 samples, corresponds to the uppermost cluster of separate points in the two-dimensional representation. Because of the remarkable

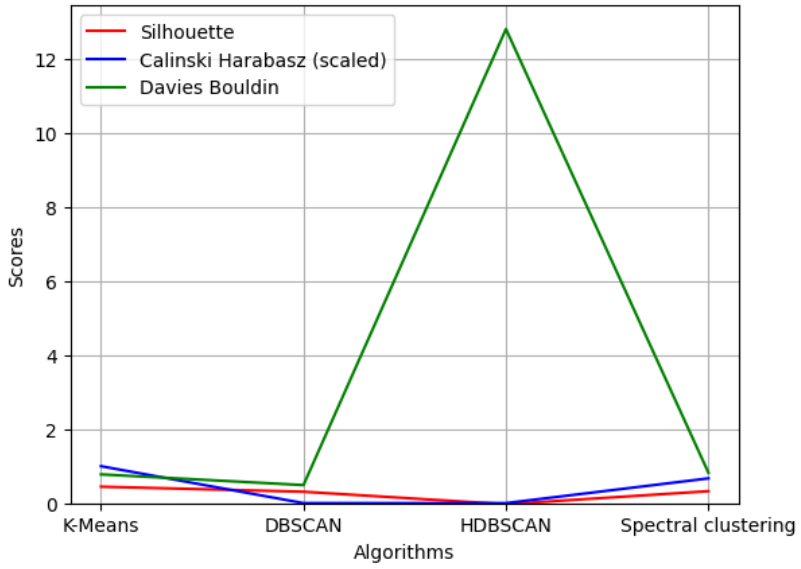

Figure 49: Evaluation metrics over different clustering algorithms.

difference in the cardinality of the cluster, all samples belonging to class 0 are discarded as outliers for further experiments in order not to introduce possible biases in the models discussed in the following sections. From now on, the cluster indexes will be numbered from 0 to 6 in order to ensure compatibility with the models used in the following sections.

7.2 Multimodal Classification

7.2.1 Supervised Audio Classification

This section specifically presents the results obtained with the implementation of the techniques described in Section 6.2.1.

First, the results obtained by using the multi-layer perceptron architecture to predict the cluster labels starting from the *general* dataset are presented and evaluated in terms of accuracy[1], recall, precision, and F1 score. Figure 51 presents the recall, precision and F1 score values for each of the seven classes, and Figure 52 presents their micro-averaged values along with the overall accuracy. The results are quite promising: considering that a small proportion (about 2%) of the original features was used to train the multi-layer perceptron classifier, its performance over 60% indicates that the general features alone can reasonably approximate the complete dataset, providing a good trade-off between dimensionality reduction and model performance. While the results are not perfect, they suggest that the selected general features are able to capture the essential characteristics of the data, allowing the model to achieve competitive results despite the drastic reduction in input dimensions.

Although the multi-layer perceptron classifier does not fully solve the explainability issue associated with UMAP dimensionality reduction, it provides a solution to the need for objective features that can capture the essence of data. This strategy is effective in approximating the mapping obtained through dimensionality reduction and clustering, offering meaningful support to the supervised task.

Despite the surprising results obtained with the first experiments, other attempts are made for research purposes to further investigate the possibility of improving the audio classification performance. One factor that needs to be taken into account before running the experiments

---

[1]In this case, accuracy is considered a reliable metric because of the single-label nature of the classification problem.

is that an issue that remains open concerns the emotional valence of the clustering labels: leveraging the data currently available, it is impossible to infer any quality of the sound that may point a human listener toward certain emotional states.

To gain more insights into the relationship between the original audio data and the clustering labels, the mutual information scores between each feature of the two datasets and the clustering labels are computed. Figure 53 shows the values for the general dataset, whereas Figure 54 concerns the block-level dataset. It can be noticed that there is, in fact, a remarkable difference between the attributes, suggesting that certain features have a stronger correlation with the labels than others. This indicates that those features may be more informative for the clustering process and, thus, for the UMAP dimensionality reduction.

After training an XGBoost classifier for the classification task, the model is leveraged to extract the feature importance scores of the two datasets. Figure 55 shows the values for the general dataset, and Figure 56 for the block-level dataset. A significant difference in scores between features is noticeable in this case as well, supporting the findings from the mutual information observation. It is worth mentioning that the discrepancy in the magnitude of values between the datasets is due to the significantly different number of features: since the sum of all feature importance scores for a dataset equals 1, a dataset with a higher number of features influences the absolute values of the scores, differently from the mutual information case.

Additional insights can be obtained by analyzing the local explanations provided by the *lime* library for each class, shown in Figure 57 to Figure 63.

Because of the complexity of the clustering structure, the analysis is performed only considering the positive indicators for the class (represented by green bars); this aims at simplifying the interpretation of results and ensuring that classes can be differentiated more easily. Given the number of classes, focusing exclusively on descriptors that contribute to a positive prediction for each class allows to identify distinctive characteristics across clusters in a more straightforward way; furthermore, the negative indicators for each class may point to any of the remaining six classes, making their interpretation less direct and less useful to the characterization of each cluster's emotional profile.

When observing the explanation for class 0 in Figure 57, the positive indicators point to songs with high spectral roll-off, sensory dissonance, and spectral entropy alongside a low voice probability. The spectral properties suggest the sharpness of sound, which could be perceived by listeners as energetic and dynamic, with a component of tension or unease given by the high sensory dissonance. The low voice probability suggests that the song is mainly instrumental. Overall, this description defines a dynamic and possibly harsh sound, with a mostly instrumental track, that could evoke energetic feelings in listeners.

As for class 1, depicted in Figure 58, it is characterized by low sensory dissonance, high spectral strong-peak, and moderate values of Melbands spread, spectral decrease, and minimum position of voice probability. The low sensory dissonance may point to songs evoking calmness and serenity; the high spectral strong peak suggests the presence of one prominent frequency or harmony, which can be associated with tonal clarity and stability, evoking positive emotions such as peacefulness or happiness. Moderate values of spectral decrease suggest fullness and

richness of the sound, making it more expressive, and the value of the Melbands spread implies a balance between harmonic and noise components. Such characteristics potentially indicate intense emotional content, with potential elements of tension or excitement. Overall, this profile suggests a harmonically pleasant and balanced sound, potentially expressing relaxation, warmth, or even gentle optimism.

Class 2, whose characterization is shown in Figure 59, presents more positive indicators and, notably, a greater proportion of features considering their standard deviation. In general, a higher standard deviation for a feature indicates greater variation in its values, contributing to a sense of unpredictability of sound. The high zero-crossing rate and relative standard deviation indicate the prevalence of high-frequency content and noise of sounds alternated with contrasting calmer sections. The moderate values of spectral entropy indicate unpredictability of the energy distribution and, therefore, of sounds, which appear more chaotic; the same conclusion can be drawn from the values of both RMS features and of the spectral decrease standard deviation. The dynamic nature of sound is confirmed by the high standard deviation of both Barkbands and Melbands energy spread, as well as by the spectral flux values. On the other hand, features like the low average Melbands spread point to less complex and more controlled spectral content, while the average dissonance value introduces a slight sense of tension. Finally, voice probability statistics indicate a stable yet non-dominant voice presence. Overall, these descriptors depict a complex sound profile marked by high energy, instability, and unpredictability, which may be associated with intense emotions, leaning towards agitation or excitement.

Class 3, shown in Figure 60, is characterized by a high onset rate and spectral flux, pointing to a fast-paced and rhythmically dense sound, with frequent spectral changes evoking liveness. The high standard deviation of ERB bands spread and sensory dissonance indicates fluctuations and dynamicity in the timbral component of the sound, highlighting the dynamic nature of the track. An interesting aspect is the low weight of the first peak in the bpm histogram, representing that the song does not rely on a prominent rhythmic component but is instead more fluid. Overall, the descriptors indicate a dynamic and fast-paced song with an intense rhythm that does not overpower the sound, suggesting that this track may convey intense and high-energy emotions, such as excitement, agitation, or restlessness, with a moderate vocal component.

The descriptors of class 4 in Figure 61 clearly depict a different sound profile: the low values of sensory dissonance, ERB bands spread, and spectral entropy indicate a calmer, harmonically smooth and balanced sound, with mellow components and less complex sound patterns. Overall, this description suggests a peaceful, stable, and possibly soothing song with a soft acoustic style.

Figure 62 shows the description of class 5, characterized by a high sensory dissonance, spectral entropy, and zero-crossing rate, indicating an energetic and dynamic sound again; this impression is reinforced by the moderate-to-high values of the spectral flux, suggesting a sense of instability and motion. Voice probability statistics indicate an intermittent presence of vocal elements, which are not significant in the overall song. Interestingly, the low loudness reported indicates that the dynamicity and energy present in the song are not explicitly expressed through sheer volume, but instead they may be conveyed through timbre, potentially leading to a

more eerie type of sound. Leveraging this information, the overall emotional content might be described as nervous, anxious, or even excited.

Finally, class 6 presents low spectral entropy, sensory dissonance, and spectral flux, as shown in Figure 63, which describe a structured and predictable sound with a smooth harmonic content and low dynamic variability. The importance of voice probability linear regression coefficients suggests a more prominent presence of vocal components, which may contribute to additional emotional expressivity. Overall, this class appears to be characterized by soft, calm, and peaceful songs, evoking positive feelings with low intensity.

As already mentioned, these are high-level and general observations that would undoubtedly need further validation through more rigorous experimentation, possibly leveraging a set of ground truth emotional labels, complete audio tracks, or at least listener feedback. However, the information obtained through this analysis offers a preliminary understanding of the potential emotional content within each cluster.

To further enhance the analysis, it is useful to extract the names of the features having higher mutual information scores and feature importance for the general dataset, in order to compare them with the ones identified by the *lime* library.

The features of the general dataset having mutual information scores higher than 0.25 are listed below, with the feature in bold having the highest mutual information score:

- barkband_spread.mean
- **dissonance.mean**
- dissonance.stdev

- spectral_entropy.mean
- spectral_entropy.stdev
- spectral_flux.mean
- voiceProb_sma_amean[1]

Whereas the features having importance higher than 0.03 according to the XGBoost classifier are:

- **lowlevel.dissonance.mean**
- lowlevel.erbbands_spread.stdev
- lowlevel.spectral_entropy.mean
- lowlevel.spectral_entropy.stdev
- lowlevel.spectral_flux.mean
- voiceProb_sma_de_linregerrQ
- voiceProb_sma_de_stddev

It is immediately noticeable how some features are common to all three evaluation systems, indicating a higher predictive ability for the cluster labels and, therefore, highlighting the differences between the seven resulting classes. It is the case of dissonance, spectral entropy and flux, and voice probability statistics, as well as the energy in ERB bands. However, the

---

[1]sma: smoothed average; amean: arithmetic mean

insights provided by the *lime* library suggest that other features have significant predictive power, such as the zero-crossing rate, the average loudness, onset rate, spectral rms, and bpm statistics, as well as energy measure on different bands. It is worth noting how all methods pose significant importance to the dissonance feature, representing the perceived roughness of the sound. According to the *essentia* documentation [36], it is computed starting from spectral information such as the peak positions and values, therefore explaining the relevancy of other spectral components.

Concerning the replication of the experiments just described on the block-level dataset, different configurations of a multi-layer perceptron have been successfully tested, and the results proved a remarkably good performance: not only does the model converge after fewer iterations (never more than 50 with the tested settings), but the metrics also indicate a significant improvement with respect to the classification using only the general features, reaching metrics values over 80%. The final configuration chosen uses again a single hidden layer, but the size is now set to 100 neurons; Figure 64 shows the recall, precision, and F1 score for each prediction class, and Figure 65 presents their average values and the overall accuracy.

For research purposes, another model is trained and tested on the block-level audio data, leveraging the inter-dependent nature of the features[1]. The model in question is a convolutional neural network similar to the one described in Section 5.2 that has already been tested on lyrics

[1]Since the dataset contains features computed on different sections of a song, it is implied that there are temporal dependencies within these sections, which may capture patterns that are relevant to the overall classification task.

data. After numerous attempts to find the optimal configuration, the best results that were obtained reached accuracy, F1 score, recall, and precision values of 62%, confirming that the MLP classifier achieves the best results despite the simpler architecture. The discrepancy in performance may be due to the non-fully connected nature of the CNN, which could struggle to capture the complex relationships between features, as it relies more heavily on local spatial dependencies, which are not significant enough in this case.

Figure 66 to Figure 72 present the local explanations for one sample for each class. Similarly to the general dataset case, an attempt to describe each class is made in order to identify potential contradictions or confirmations with respect to each class's musical and emotional information obtained earlier.

By analyzing the descriptors for each class, it can be noticed how the vast majority of them are related to the bpm histogram of specific sections of the sound, suggesting that the model classifies samples based on the location of certain rhythmic patterns. For instance, the class 0 descriptors shown in Figure 66 imply that the discriminative sections are characterized by a reduced rhythmic structure, pointing to a song having a softer rhythm in those parts. This does not contradict the hypothesis made for the same class based on the general dataset descriptors, but rather provides additional insights into the sound. When considering all of the classes, it is noticeable how they are differentiated based on which specific sections have a slower rhythmic structure. Consequently, this analysis is less informative and offers limited relevance to the purpose of this research and is, therefore, omitted.

As for what concerns the mutual information scores for the block-level dataset of interest, the features associated with values larger than 0.25 are reported below, where the feature in bold is the one having the highest score, and the *X* represents different song section indexes:

- lowlevel.erbbands.mean_X, lowlevel.melbands.mean_X, lowlevel.barkbands.mean_X
- BLF_spec_ctrsX (block-level features spectral contrast)
- lspFreq_sma_de[X] statistics
- mfcc_sma_de[X] statistics
- **lowlevel_spectral_contrast_valleys.mean_X**

Similarly, below are reported the features having importance greater than 0.003 according to the XGBoost evaluation:

- lowlevel.erbbands.mean_X
- lowlevel.spectral_contrast_valleys.mean_X
- mfcc_sma_de[X] statistics
- lspFreq_sma_de[X] statistics
- **BLF_spec_ctrsX** (block-level features spectral contrast)

Differently from the general dataset case, there is an overlap between features having high mutual information scores and XGBoost importance, but the interpretation provided by the *lime* library poses the focus on different attributes. Specifically, great importance is given to

the bpm histogram, highlighting the impact of the rhythmic structure of data and to the base frequency F0 and its envelope, correlated to the pitch of the songs.

Although great improvements have been made concerning the two issues mentioned at the beginning of the section, giving a full interpretation of the cluster labels able to tie it to the emotional aspects of songs is still an open challenge. The unavailability of the audio tracks limits the possibility of performing a qualitative analysis of the tracks to assess how songs belonging to different clusters sound to the human ear and if the division actually corresponds to different emotional patterns. Further analysis, possibly incorporating human annotators and audio segments of the songs, would be required to move forward and bridge this gap. Unfortunately, the current state of the dataset does not allow for such an approach, and the integration of this research with data or methodologies leverageable to infer emotional properties remains open for future work. For the time being, the information obtained so far about feature relevancy and label predictions for audio data is integrated with the lyrics layer of the Music4All-Onion dataset and the relative labels obtained in Section 4.2.4 to achieve the final multimodal classifier.

The attempt to find a correlation between the cluster labels and their potential emotional content, although not completely successful in providing definitive answers, yielded valuable insights into the clustering structure. This analysis, especially the one leveraging the information from the *lime* library on the general dataset, partially supports the hypothesis that the emotional features used to define the clusters are, in fact, meaningful and relevant indicators to distinguish songs into different groups. A summary of such findings is provided in Table XI.

TABLE XI: SUMMARY OF INFERRED AUDIO QUALITIES FOR EACH CLUSTER

ClusterID	**Sample Description**
0	Sharp, energetic, dynamic sound; tension; possibly mainly instrumental.
1	Tonal clarity and stability; calm, rich, harmonically pleasant sound.
2	Complex, energetic, unpredictable, intense sound.
3	Dynamic, fast-paced, high-energy sound; fluid rhythm.
4	Calm, harmonically smooth, balanced sound; soothing, calm, stable structure.
5	Energetic, dynamic sound; potentially eerie; intermittent vocal component.
6	Structured, predictable, soft sound; smooth harmonic content; important voice component.

7.2.2 Final Multimodal Classifier

The experiments to evaluate the different architectures described in Section 6.2.2 have been carried out leveraging Bayesian Optimization to find the optimal parameter configuration for each model. The parameters that have been tested are:

- number of filters of the convolutional layers, equal for all convolutional layers; [32, 64, 128, 256]
- kernel sizes, different for each convolutional layer; [5, 10, 15, 20, 25]
- number of units in each of the LSTMs, different for each LSTM; [50, 75, 100]
- optimizer; [adam, sgd, adagrad, rmsprop]
- optimizer's learning rate; [1e-2, 1e-3, 1e-4]

- number of neurons in the dense layer processing audio classes; [8, 16, 32]

Additionally, different thresholds are tested on the probability vectors output from the model to set the value above which a label is marked as positive.

The chosen loss function for the multi-label classification task is binary cross-entropy, reflecting the considerations made in Section 5.1.

As shown by the test metrics in Figure 73, the performed experiments do not yield significantly different results when the bidirectional LSTM and pooling layers are inverted, suggesting that the order of these components may not critically impact model performance and highlighting its robustness.

The final configuration chosen applies, therefore, the pooling operation before feeding the information into the bidirectional LSTM: this choice is motivated by the lower computational cost and faster training without negatively impacting the performance.

This result, however, reinforces the idea that the recurrent component does not have a significant impact on the classification task, given the data available.

The optimal configuration, obtained with Bayesian Optimization, uses 128 filters across the three convolutional layers, with kernel sizes equal to 5, 15, and 20, respectively. Each parallel LSTM has 75 units, while the first dense layer processing the audio classes consists of 8 neurons. The model is optimized using Adagrad with a learning rate equal to 1e-3.

To balance performance and computational efficiency while minimizing overfitting, the model is trained for a total of 20 epochs. The optimal trade-off between precision and re-

call is obtained with a threshold of 0.25, meaning that a label is assigned to a sample if the probability output by the model for that class is higher than 0.25.

Table XII shows the performance achieved by the different versions of the model on the different datasets; the audio and lyrics experiments were performed using the hybrid convolutional-recurrent architecture solely on the lyrics dataset to predict the emotional labels or on the audio datasets to predict the cluster indexes, respectively. The integrated experiments, instead, leverage the final definitive architecture, using the acoustic features as auxiliary information for the classifier. The values reported in the table are average indicators of the performance achieved by the models during different runs.

TABLE XII: PERFORMANCE OF THE CONVOLUTIONAL-RECURRENT ARCHITECTURE ON LYRICS-ONLY DATA AND ON DATA INTEGRATING LYRICS AND ACOUSTIC CLUSTER LABELS

Metric	**Lyrics**	**Multimodal architecture**
Precision	0.64	0.42
Recall	0.56	0.74
F1-score	0.60	0.54
Hamming loss	0.21	0.37

Despite the reported results showing that the multimodal architecture yields unsatisfactory results for the emotional label classification task, it provides insights into the role of different modalities in this context. Specifically, they confirm that textual data is the most valuable source of information, and that the introduction of acoustic information in the form executed in this work unfolds as detrimental to the performance of the models. These insights may guide future research toward more effective multimodal fusion techniques leveraging improved data in terms of quality and relevance of audio features for emotional classification tasks.

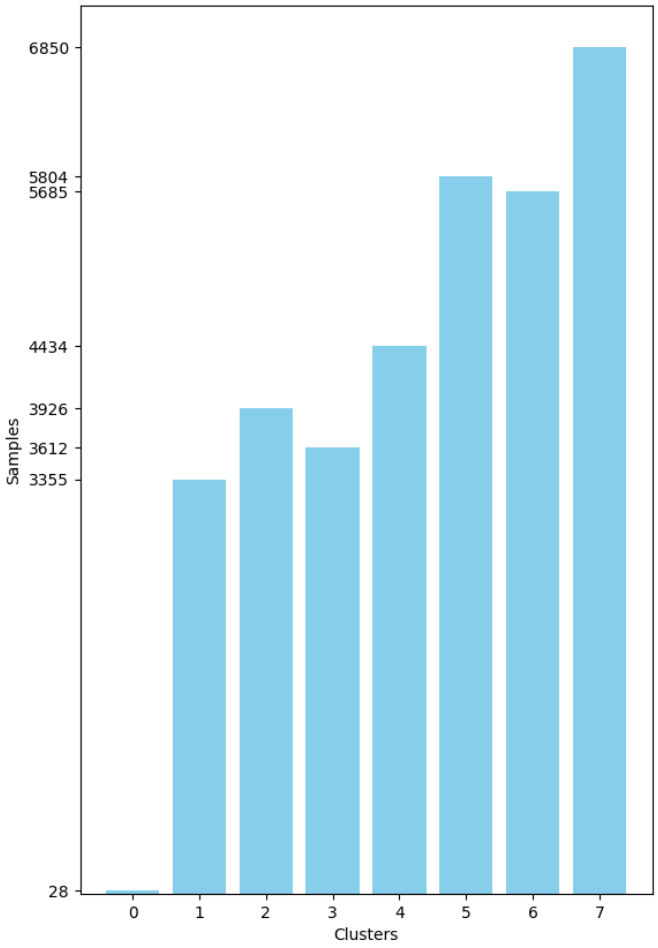

Figure 50: Distribution of samples over different clusters.

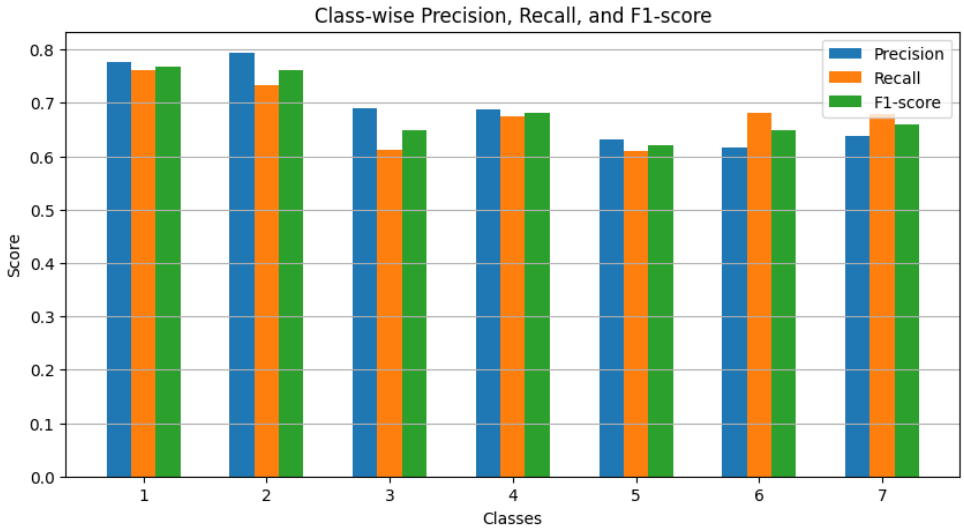

Figure 51: Performance of multi-layer perceptron classifier per cluster class on the general dataset.

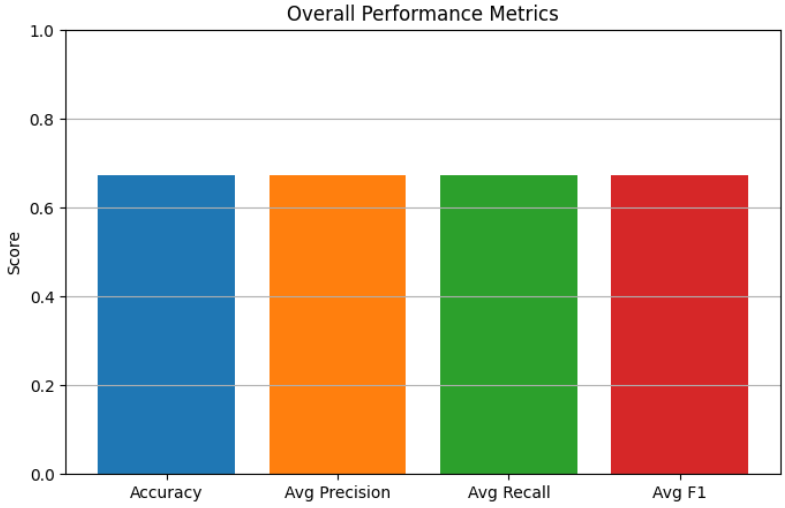

Figure 52: Average performance of multi-layer perceptron on the general dataset.

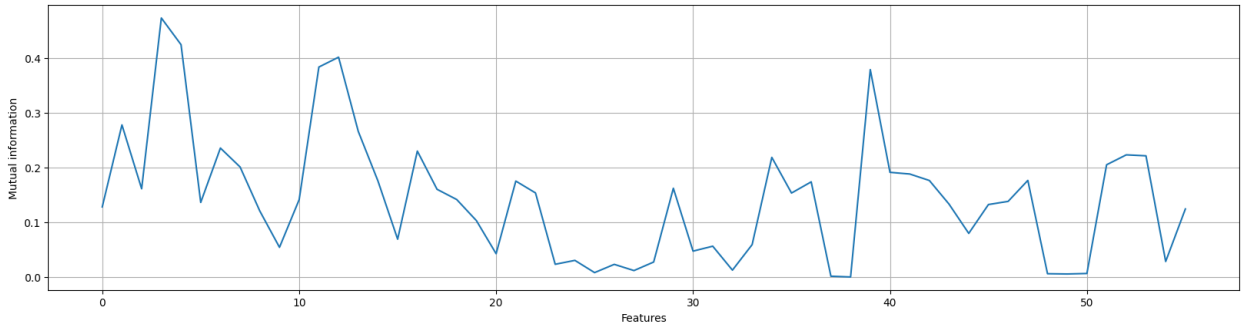

Figure 53: Mutual information per feature of the general dataset.

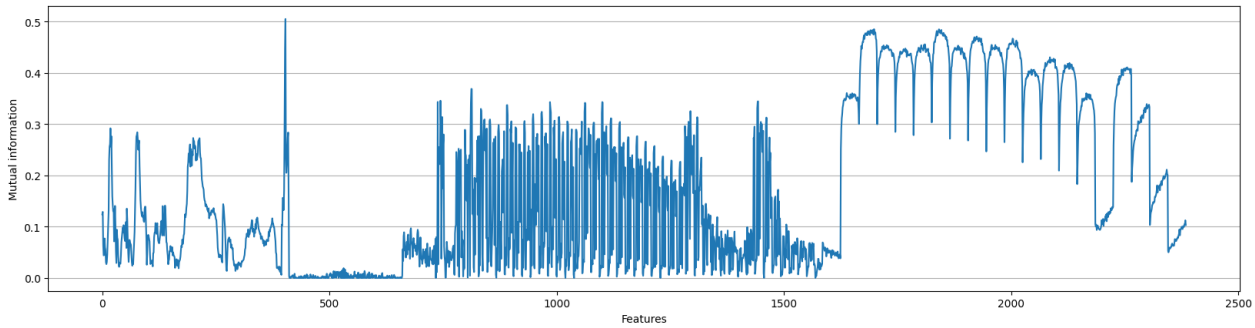

Figure 54: Mutual information per feature of the block-level dataset.

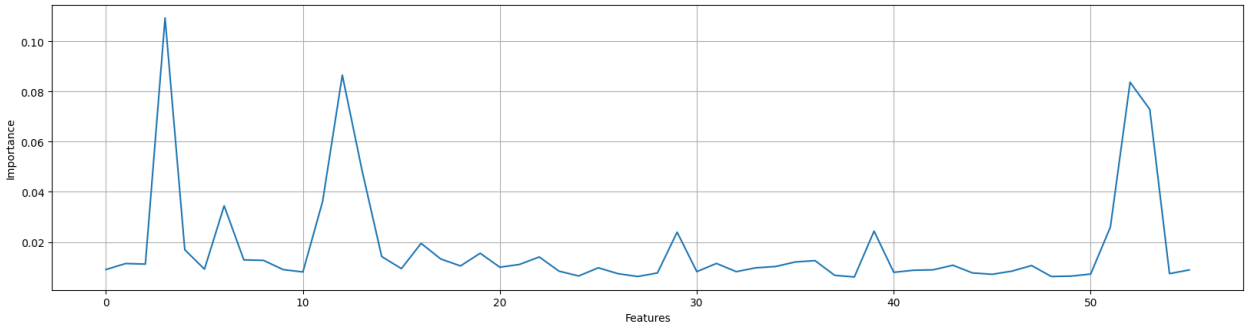

Figure 55: Feature importance per feature of the general dataset.

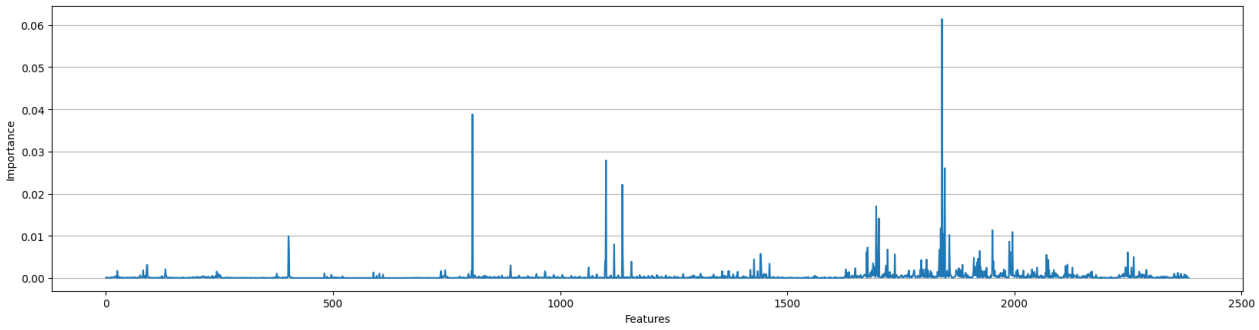

Figure 56: Feature importance per feature of the block-level dataset.

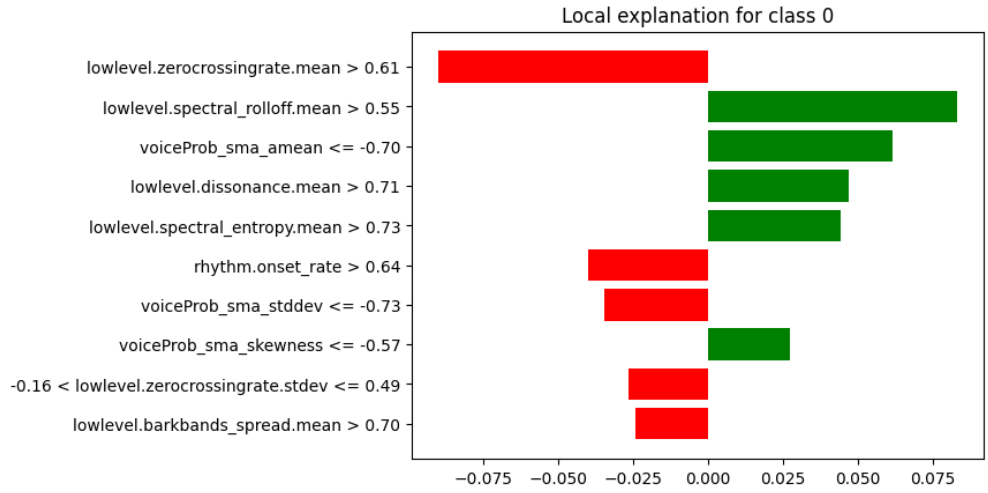

Figure 57: Feature importance in predicting label 0 on the general dataset.

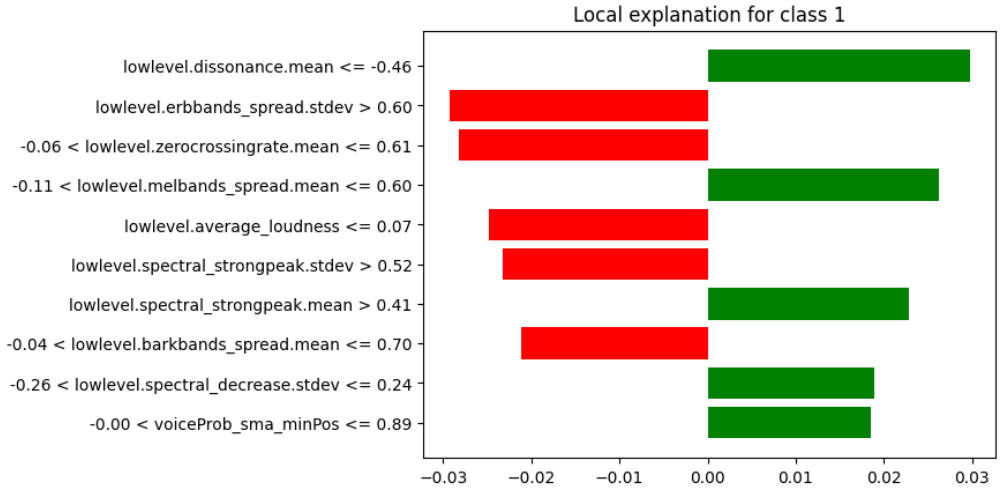

Figure 58: Feature importance in predicting label 1 on the general dataset.

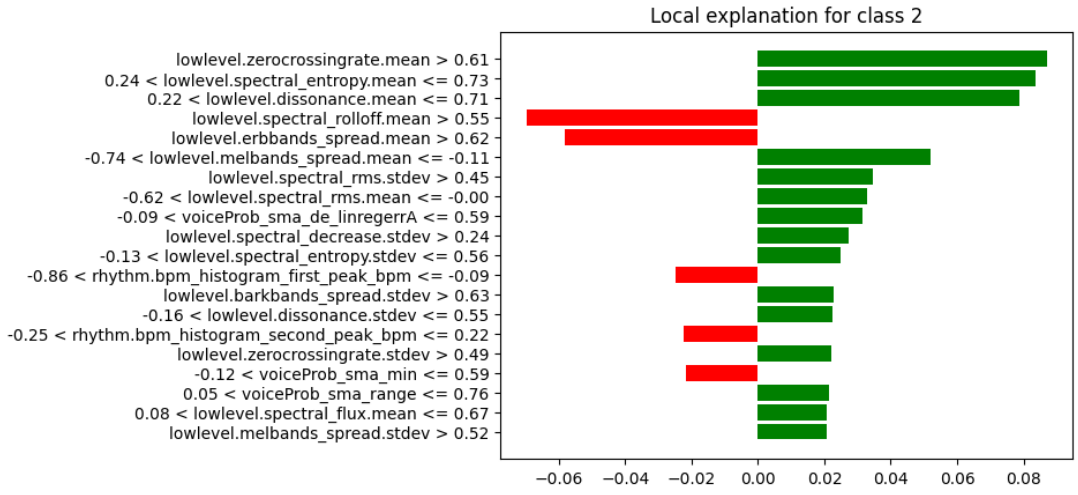

Figure 59: Feature importance in predicting label 2 on the general dataset.

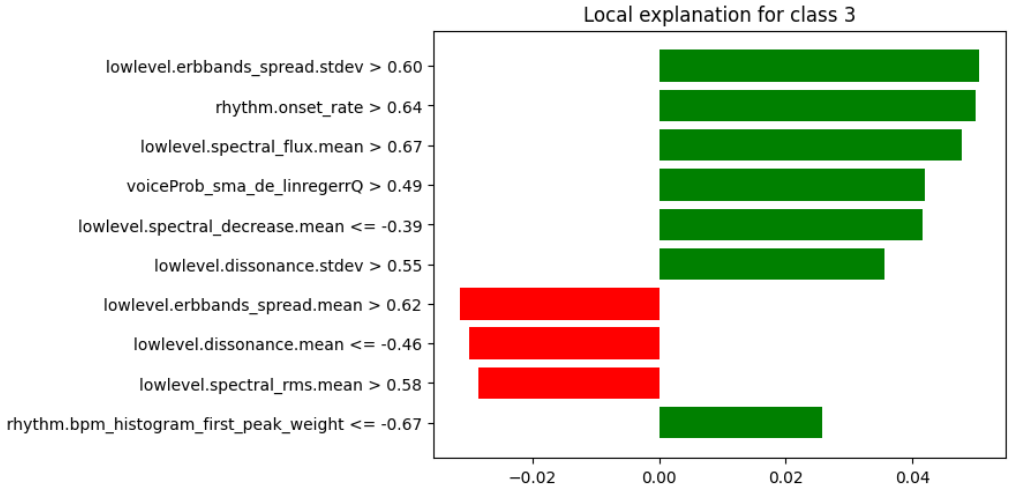

Figure 60: Feature importance in predicting label 3 on the general dataset.

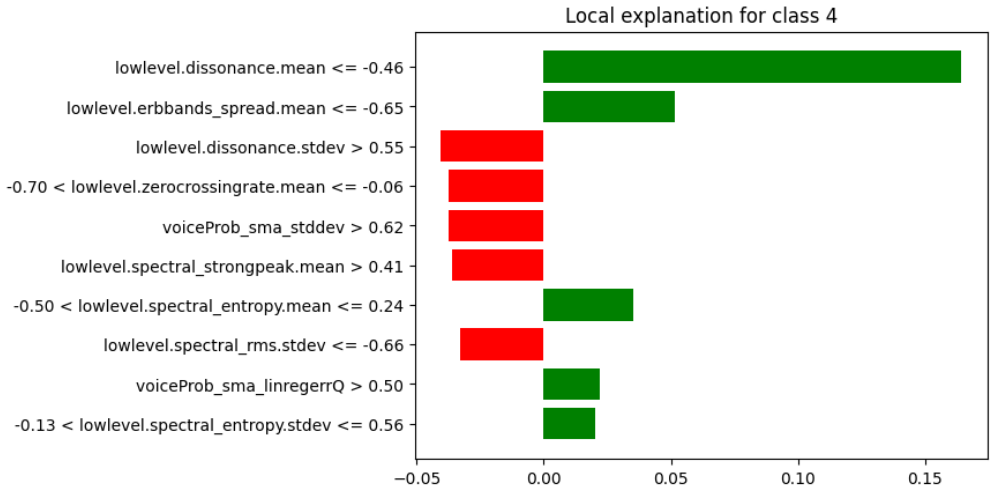

Figure 61: Feature importance in predicting label 4 on the general dataset.

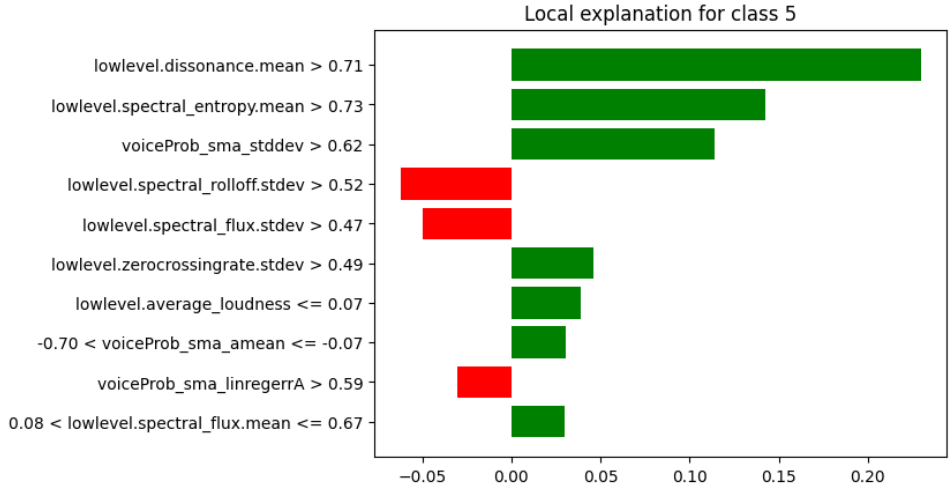

Figure 62: Feature importance in predicting label 5 on the general dataset.

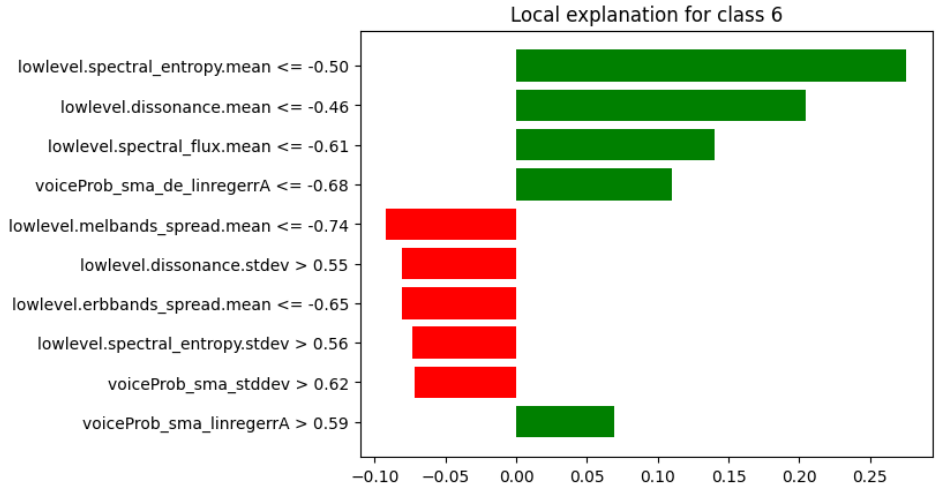

Figure 63: Feature importance in predicting label 6 on the general dataset.

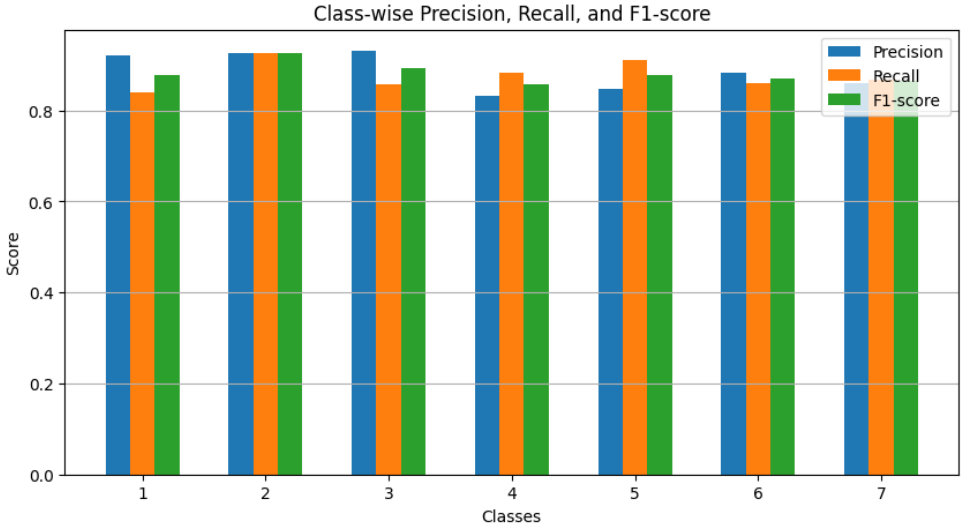

Figure 64: Performance of multi-layer perceptron classifier per cluster class on the block-level dataset.

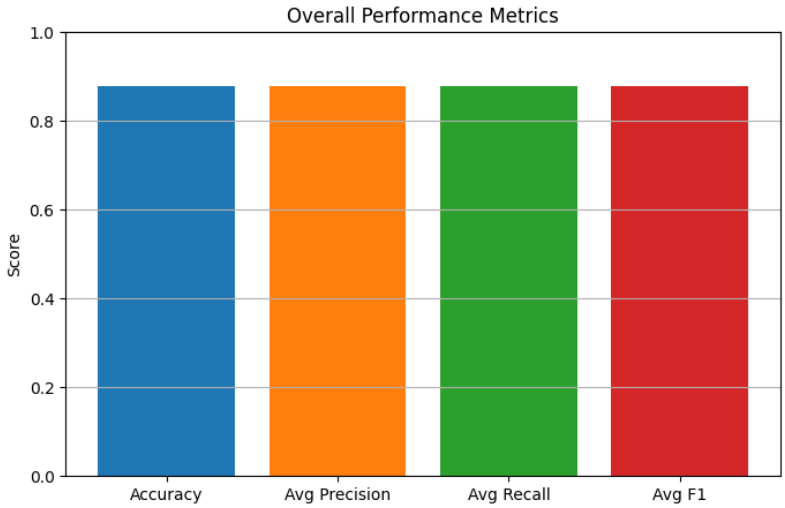

Figure 65: Average performance of multi-layer perceptron on the block-level dataset.

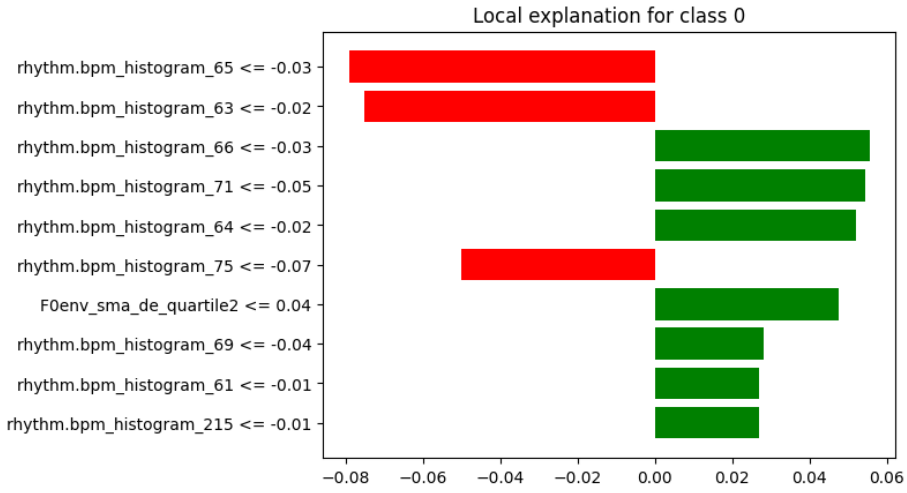

Figure 66: Feature importance in predicting label 0 on the block-level dataset.

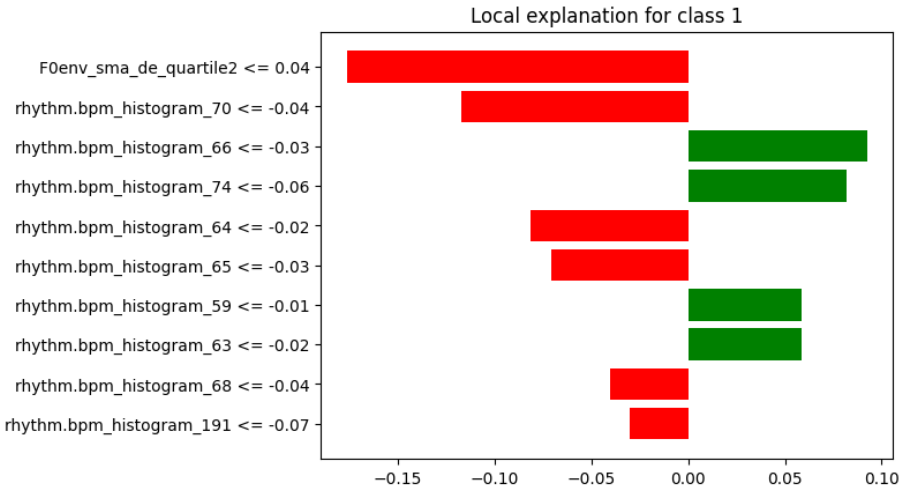

Figure 67: Feature importance in predicting label 1 on the block-level dataset.

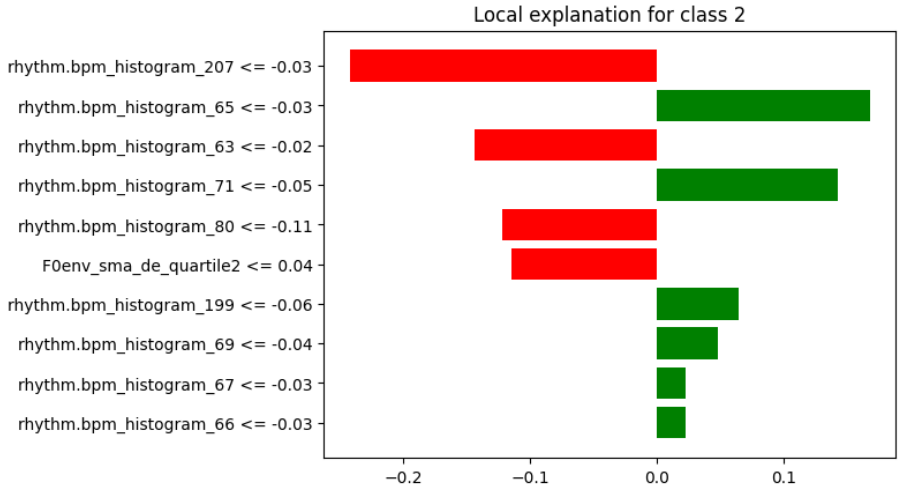

Figure 68: Feature importance in predicting label 2 on the block-level dataset.

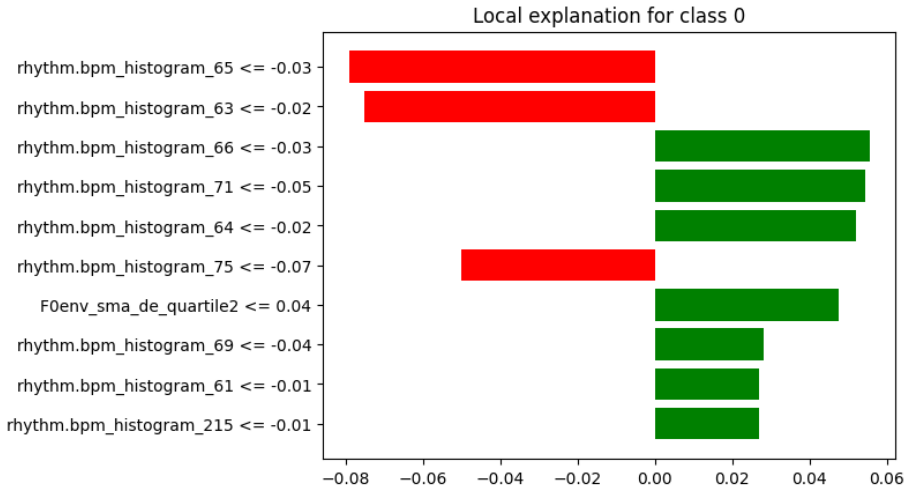

Figure 69: Feature importance in predicting label 3 on the block-level dataset.

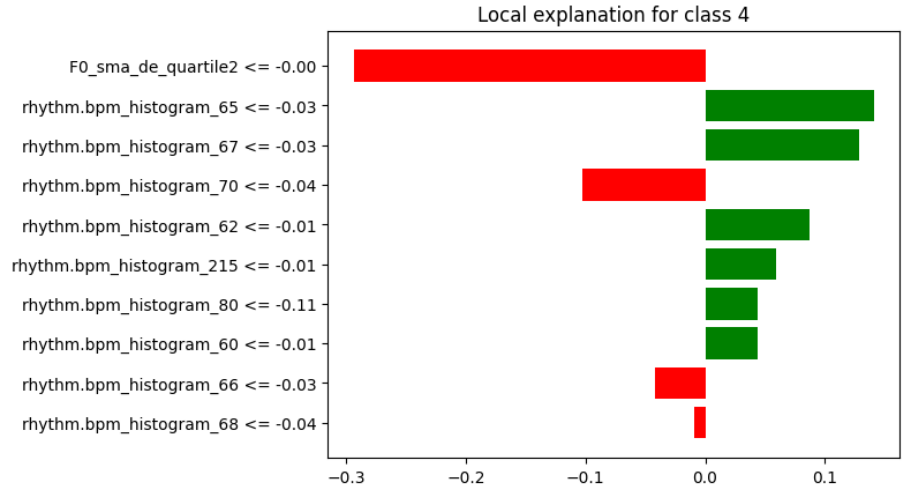

Figure 70: Feature importance in predicting label 4 on the block-level dataset.

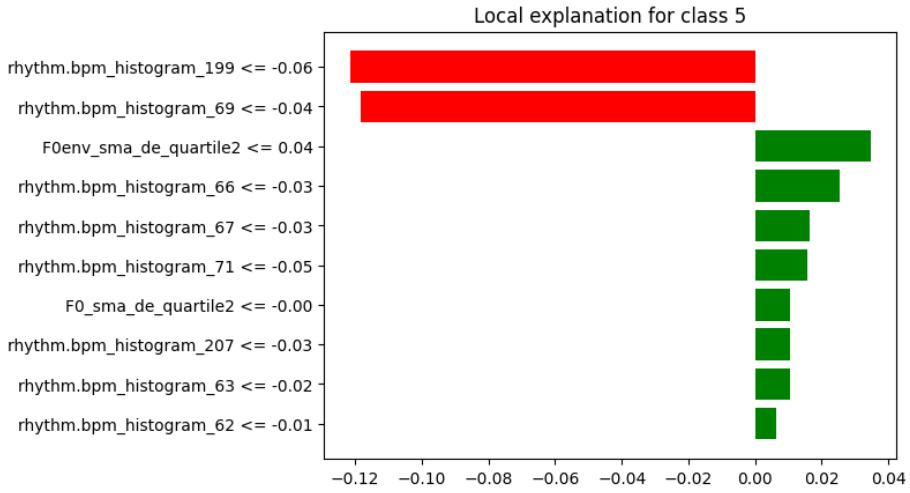

Figure 71: Feature importance in predicting label 5 on the block-level dataset.

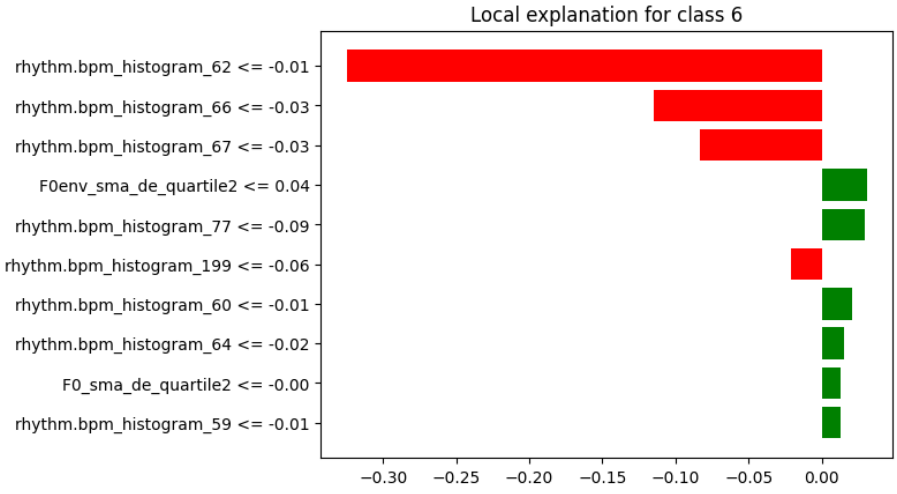

Figure 72: Feature importance in predicting label 6 on the block-level dataset.

Figure 73: Test metrics for two different designs (pooling layer followed by bi-LSTM layer / bi-LSTM layer followed by pooling layer).

CHAPTER 8

CONCLUSIONS

This thesis explored the development of a multimodal classifier for emotion-based music recommendation, leveraging textual and audio features to maximize the effectiveness of emotional classification. By integrating both modalities, this research aimed at capturing a more comprehensive representation of the emotions encompassed by musical content, improving the performance of unimodal models.

The challenges addressed covered the whole development process, ranging from the data scarcity issues explored in Section 3, which posed the most limiting issues to this work, to the need to balance computational complexity and efficiency throughout the experimental research phases.

To address the data scarcity issue, considerable effort was made to curate a suitable dataset for the task at hand, leveraging the best resources available at the time these decisions were taken. As this research progressed, other resources became available to the public, such as new multimodal datasets offering different data formats and quantities for this task, potentially enhancing the model's performance and simplifying the pipeline. However, by the time these new resources were discovered, the research had already advanced too far to allow for a complete redesign of the pipeline, forcing the study to proceed with the originally available dataset. Future work can certainly benefit from these newly available datasets, either by incorporating them into the model to improve generalization or by exploring different data formats for further

experimentation. Additionally, a crucial improvement would be the integration of more reliable emotional labels, which could contribute to enhancing the overall classification quality.

This work evaluated different models on diverse tasks: first, for the lyrics labeling task, different models have been tested on the transfer learning task, proving the superior performance of the GPT-4o-mini large language model on other LLMs and on the BERT model. Subsequently, different dimensionality reduction techniques have been used in an attempt to reduce the dimensionality of audio data in order to give way to multiple clustering algorithms to find meaningful emotional patterns, attempting to create pseudo-labels for the audio component of songs. These pseudo-labels have then been investigated to map the relationship between them and the original audio features, gaining insights into the acoustic properties of samples belonging to different classes. Finally, the definitive multimodal architecture has been developed, joining the good performance of the multi-layer perceptron model in predicting the audio classes for each sample with the ability to process textual data of the convolutional-recurrent model.

The multimodal architecture was originally designed to fully integrate both textual and audio information, leveraging the strengths of both modalities to enhance classification. However, in practice, the model is primarily functions as a text classification system, leveraging audio information only as an auxiliary feature. This outcome highlights the challenges posed by the scarcity of public data sources, which limited the ability to implement a custom architecture for feature extraction. Additionally, the lack of reliable emotional labels for audio data further complicated the task, hindering the development of a robust audio classification model. Future

work could address these limitations and extend this approach by leveraging new audio sources and high-quality labeled data, enabling a classifier in which both modalities contribute significantly to the prediction. To this end, the creation of a customized loss function to balance the importance given to the two modalities could be crucial to the classification task at hand and to further applications for content recommendation based on each user's preferences. Provided that a new model in which the two modalities have comparable importance, a personalized approach to classification allowing users to specify the importance of different modalities based on their listening habits and preferences could be explored.

The evaluation of different architectures led to multiple key findings: first, exclusively convolutional models outperformed both recurrent and hybrid models, including the final multimodal architecture, on the lyrics classification tasks. This suggests that feature extraction played a more significant role when compared to capturing contextual dependencies. A potential contributing factor to this is the preprocessed status of the lyrics data: while techniques such as stopword removal, stemming, and lemmatization are standard practice in multiple natural language processing tasks, their application on song lyrics significantly altered them, often making them unrecognizable even to human eyes. As a result, the reduced linguistic context might have limited the good performance that recurrent architectures often achieve on sequential data. The same considerations can be made for the audio modality: while the final architecture did not perform well on the features already extracted, working with raw audio tracks could allow for more effective feature extraction, able to capture a wider temporal window and, therefore, more spectral patterns. The use of preprocessed data, as continuously mentioned throughout this

thesis, may have caused the loss of important information that could have improved the performances of the tested models. Future work could explore how end-to-end convolutional-recurrent architectures perform on raw acoustic data, where more temporal information can be extracted and leveraged for classification. One promising approach could involve using the convolutional-recurrent block to process spectrogram representations of songs, which were unavailable in the datasets used in this work, to extract deeper insights into the emotional properties of music.

To conclude, this thesis has tackled various problems in the MER field, specifically achieving the following results:

- creation of an emotion-based lyrics dataset starting from the Music4All-Onion lyrics layer (Section 4);
- identification and implementation of various baselines on the datasets used in this work (Section 5);
- evaluation of different dimensionality reduction techniques on the audio layers of the Music4All-Onion dataset, achieving a simplified representation on which meaningful clustering experiments can be conducted (Sections 6.1.1 and 7.1.1);
- identification of meaningful acoustic properties of different classes of samples based on the values of the audio features (Sections 6.2.1 and 7.2.1);
- definition and implementation of a multimodal architecture able to infer emotional labels for each sample starting from a subsection of the raw features of the Music4All-Onion dataset (Sections 6.2.2 and 7.2.2).

Although the quantitative results obtained in terms of performance did not surpass those of the baseline models tested on the lyrics dataset, this study achieves promising results considering the available resources and provides insights into the impact of multimodal information for music classification. Furthermore, the analysis of the emotional content of songs contributes to an underexplored field with vast potential applications: aside from the intended contributions to emotion-based music recommendation, emotionally aware music classification has possible implications in the creation of mood-based playlists or interactive music experiences, in the creation of personalized soundtracks for storytelling, and in the mental healthcare field, proving its relevance and the necessity for further research.

APPENDIX

EMOTIONAL LABELS EXTRACTION FROM USERS TAGS

Before diving into the labels extraction, some exploration of the dataset's properties can be useful to better understand its composition and have a clearer idea of the specific challenges that it presents. This is a crucial part of the process since it allows the specific design of a suitable methodology.

After removing the songs that did not have any tag assigned to them, each song has, on average, 41 tags, with the songs being tagged the least having 1 tag and the most tagged songs having 100 tags. The frequency distribution of the number of tags per song shown in Figure 74 highlights a very large number of songs having 100 tags, from which it can be inferred that, during the dataset creation, tags exceeding the 100th one were discarded.

The structure of the dataset is then changed so that each sample respects the following structure:

```
<user_tag>: [(song1, weight1), ..., (songN, weightN)]
```

In this way, it is possible to track exactly how many unique tags are in the dataset and how many songs each tag is assigned to on average. This process resulted in 257.510 distinct tags, with an average of 14 songs per tag. The most used tag is associated with 40.031 songs and the least popular ones with 1 song. The frequency distribution is in this case dramatically more unbalanced, as shown in Figure 75 and more clearly in Figure 76, and it reflects an issue in the

APPENDIX (continued)

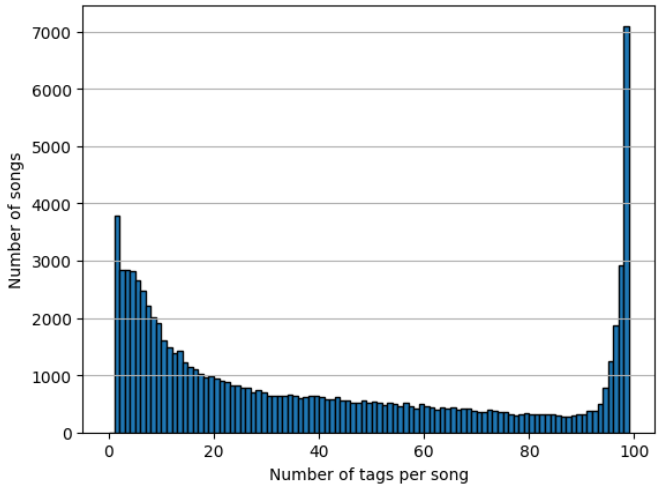

Figure 74: Number of users' tags per song.

data collection stage: the users are able to assign any tag to a song, in opposition to only being able to choose from a predetermined list, and this leads to tags with really high specificity and therefore applicable to few songs, if not only one.

The insight gained through these operations highlights the importance of a mapping process designed to drastically filter and reduce the number of tags in the dataset. Inspired by [90], the tool employed for this purpose and to extract emotional information from the tags is WordNet-Affect (WNA) [62], an extension of the WordNet lexical database [91]. WordNet was originally introduced in 1995 as a tool to collect lexicographic information in a form that is usable by computing systems and consists of organizing words in sets of synonyms, referred to as *synsets*,

APPENDIX (continued)

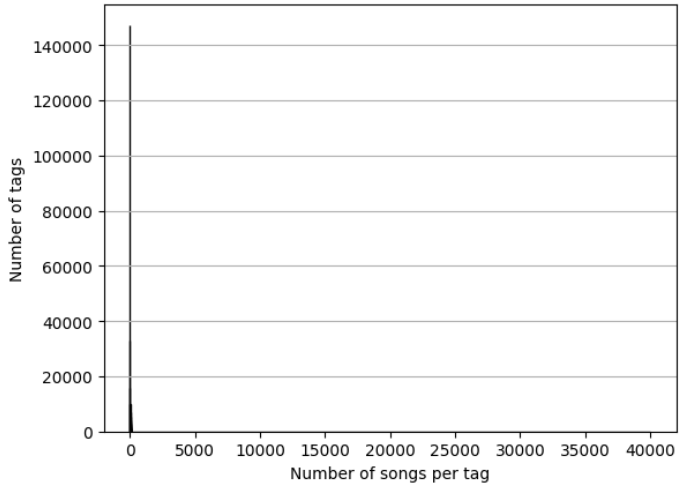

Figure 75: Number of songs per unique tag.

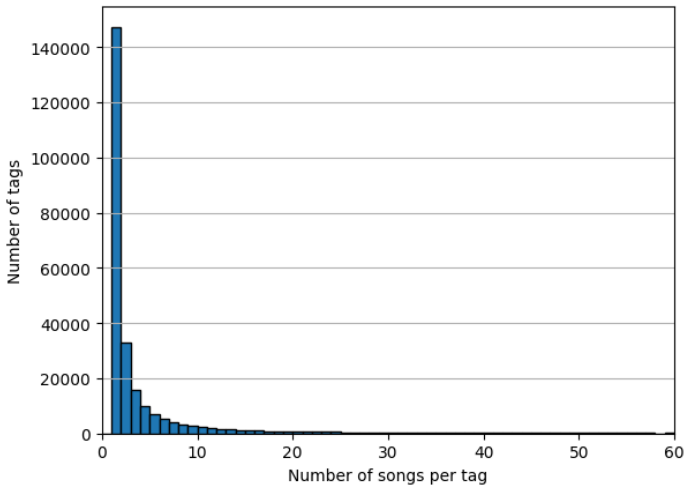

Figure 76: Number of songs per unique tag (zoomed in).

APPENDIX (continued)

linked by semantic relations. The -Affect extension was later introduced to encode emotional and affective knowledge in order to facilitate the operations in the field of affective computing.

The development of WordNet-Affect started with the identification of affective synsets in the original WordNet database with the aid of Affect, a preliminary lexical database containing terms referring to emotional states and the corresponding affective information [92]. Affect is a core element of WordNet-Affect since it is mapped onto the WordNet original database to create its final, emotionally accurate version through the addition of information without the need to change the original structure.

Emotional information is encoded leveraging a hierarchical tree structure, in which more specific emotional states are iteratively organized under more general concepts. As an example, one of the terminal leaf emotional states is guilt, which is hierarchically mapped through the following affective nodes:

```
negative-emotion -> sadness -> sorrow -> regret-sorrow -> compunction -> guilt
```

This allows working at different levels of specificity based on the task at hand and on the desired granularity.

In practice, the WordNet-Affect package, freely available on GitHub[1], identifies whether words of a sentence belong to any of the synsets encoding affective information and by returning the corresponding emotion if found.

---

[1]https://github.com/clemtoy/WNAffect

APPENDIX (continued)

To apply WordNet-Affect to the emotional labeling task discussed in this section, all the distinct users' tags are fed to the library's methods to infer the affective meaning associated with each of them. Each word that corresponds to an emotion, which will be referred to as *trigger word*, is saved in a dedicated file for future investigation and the corresponding emotion at the desired specificity level is memorized. After each tag is processed through WordNet-Affect, another dictionary is used to store the newly found emotions and the associated songs and weights. The tags that did not have any affective meaning according to WNA are discarded.

The resulting emotions to which all the tags were mapped and their occurrence counts are shown in Figure 77. The clear imbalance in the cardinality of songs tagged with *love* raised doubts about the emotion tagging process, which needs to be inspected to verify if the imbalance is due to a conceptual mistake or if it is naturally embedded into the dataset.

The starting point for the investigation of the strong imbalance for the *love* class is the trigger words file. By examining which tags are selected as those having an emotional meaning, it is possible to notice occurrences of words such as *loves, loved, likes* and *liked*: it is clear to human eyes that these words may indicate that a user *loves* a certain song, not necessarily that that song is about *love*. Similarly, quality indicators such as *good* and *bad* may say nothing about the emotional reaction evoked in the users but could have been used simply to express a personal preference. This problem, also reported by the authors of [90], can be solved by ignoring all the ambiguous tags: after manually cleaning the trigger words file to delete the words that may cause equivocation, a filtering step is added to the WNA tagging such that emotions deriving from words which are not in the "allowed" trigger words set are discarded.

APPENDIX (continued)

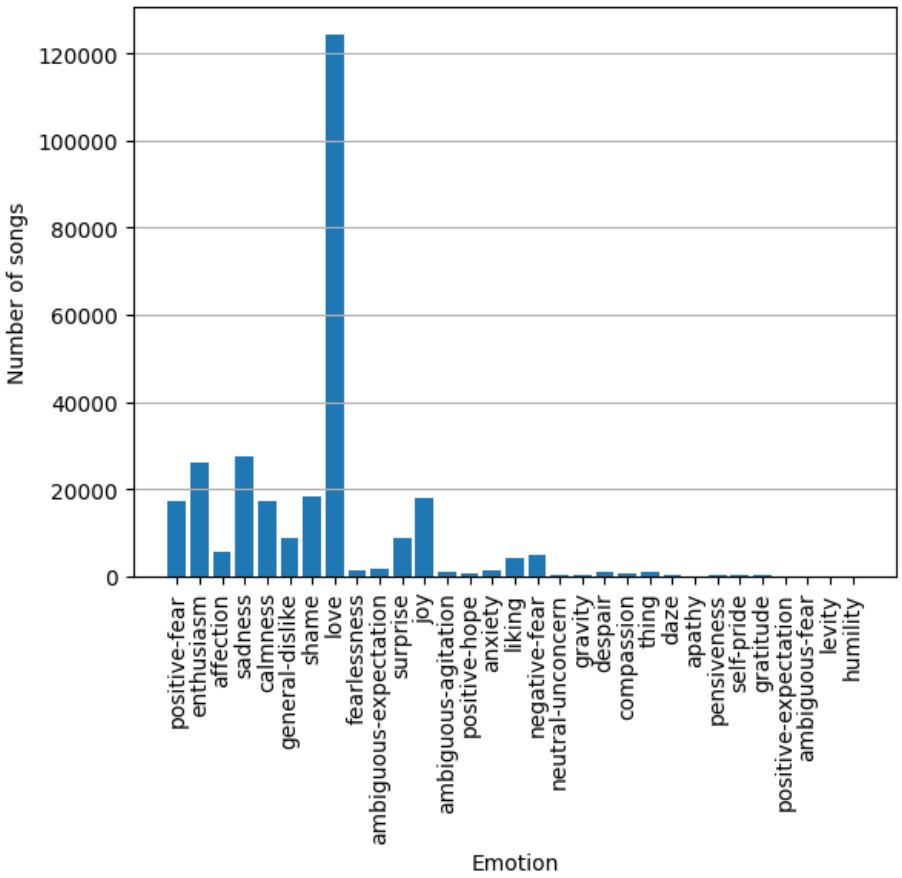

Figure 77: Emotions identified by WordNet-Affect and corresponding frequency in the dataset.

APPENDIX (continued)

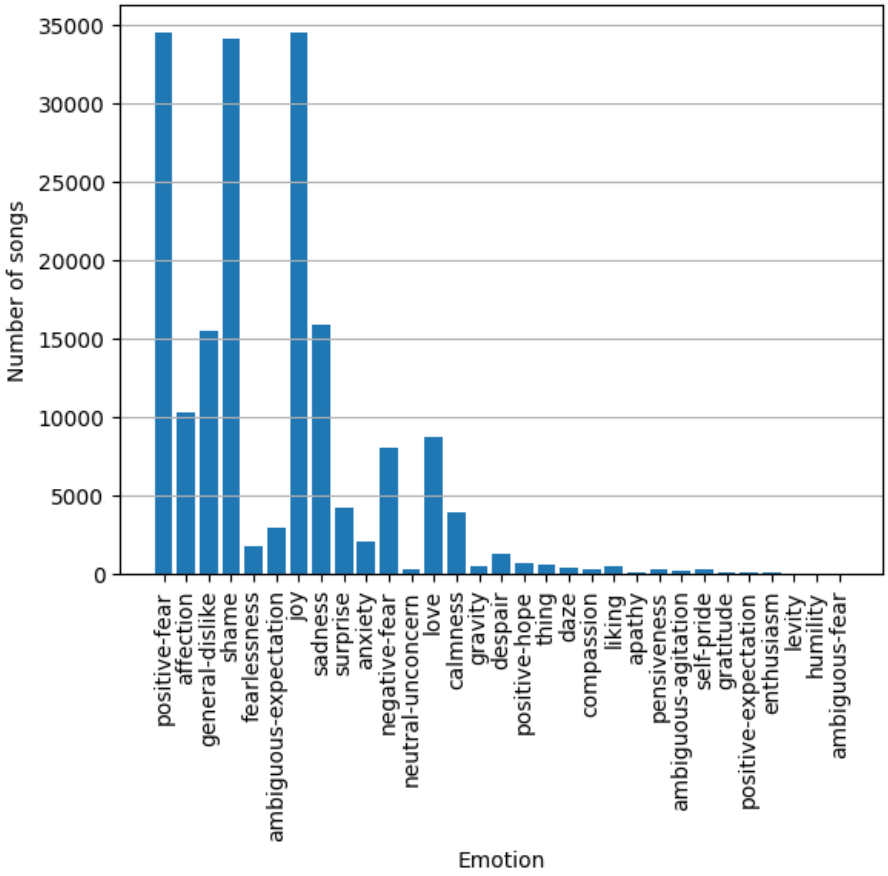

Figure 78: Emotions identified by WordNet-Affect and corresponding frequency in the dataset after cleaning the trigger words.

APPENDIX (continued)

After the cleaning operations, a more balanced distribution of the tags can be observed in Figure 78. Despite a persisting imbalance, the cardinality of the different emotions is now at least comparable.

As mentioned at the beginning of this section, not every song is labeled with tags that have emotional valence, and this process allowed the dataset to be cleaned from all those songs. The resulting working dataset is now composed of 42.572 tracks.

It is now possible to dive deeper into the labels obtained through the WNA classification. Figure 78 shows among the emotions on the x-axis *gravity* and *thing*, which do not correspond to what is usually defined as an emotion. Therefore, these tags are discarded.

Despite the goal of working with an increased level of emotional granularity expressed in Section 3.1.2, a set of 29 distinct emotions would represent a level of specificity that would be hard to work with even as humans, and thus the next goal is to reduce it to a simpler set by clustering together similar emotions.

After reviewing the literature on the matter, eight clusters are identified based on Plutchik's wheel of emotions [55]:

- Joy: fearlessness, joy, liking, self-pride, enthusiasm, levity
- Trust: affection, love, gratitude, calmness, neutral-unconcern
- Fear: ambiguous-fear, negative-fear, anxiety
- Surprise: surprise, daze
- Sadness: sadness, compassion, humility, despair, pensiveness, apathy

APPENDIX (continued)

- Disgust: shame
- Anger: general-dislike
- Anticipation: positive-fear, ambiguous-expectation, ambiguous-agitation, positive-expectation, positive-hope

It is important to note how this clustering operation has been performed on the basis of the WNA library documentation in order to be as faithful as possible with the original intended emotional interpretation given by the authors. Figure 79 shows the resulting dataset's distribution of songs over the final emotional clusters, whereas Figure 80 shows the distribution of number of tags per song.

There still is one final cleaning operation to perform: the weight associated with each tag is a form of measure for the reliability of the tag. Assuming that the tag used most times has a score of 100, and considering the high tags imbalance shown in Figure 74, discarding all the tags having a weight smaller than or equal to 5 seems to provide a good trade-off between ensuring data reliability and maintaining a large enough dataset. This operation leads to a final dataset of 27.657 songs.

After these operations have been performed, it is possible to create two final versions of the dataset: a *multilabel* version, which allows for each track to be tagged with more than one emotion, and a *monolabel* version, in which only the emotion with the highest weight is used to label a song. The distribution of emotional labels in the monolabel version is shown in Figure 81, and the one for the multilabel version in Figure 82, with the final distribution of the number of tags per each track being displayed in Figure 83.

APPENDIX (continued)

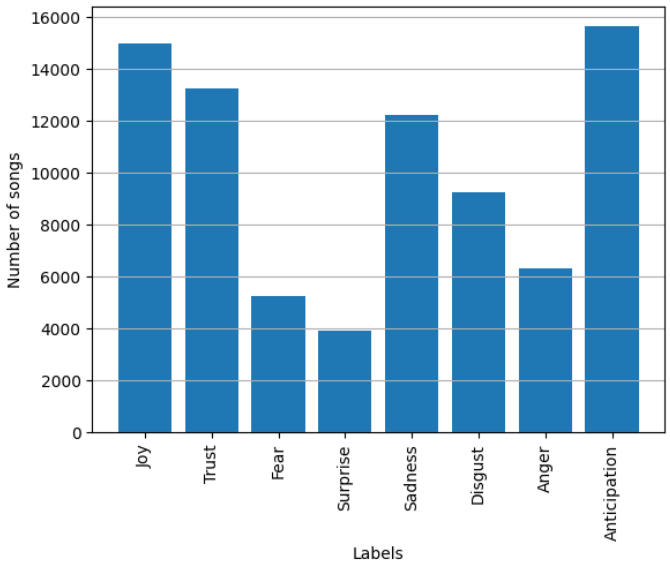

Figure 79: Intermediate distribution of emotional labels.

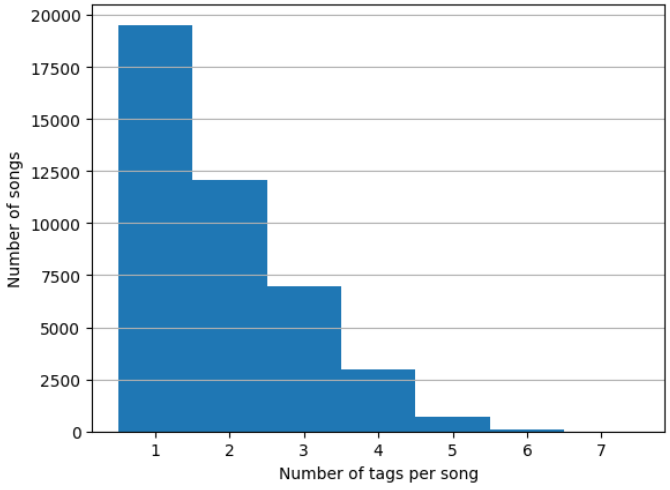

Figure 80: Intermediate distribution of emotion tags per song.

APPENDIX (continued)

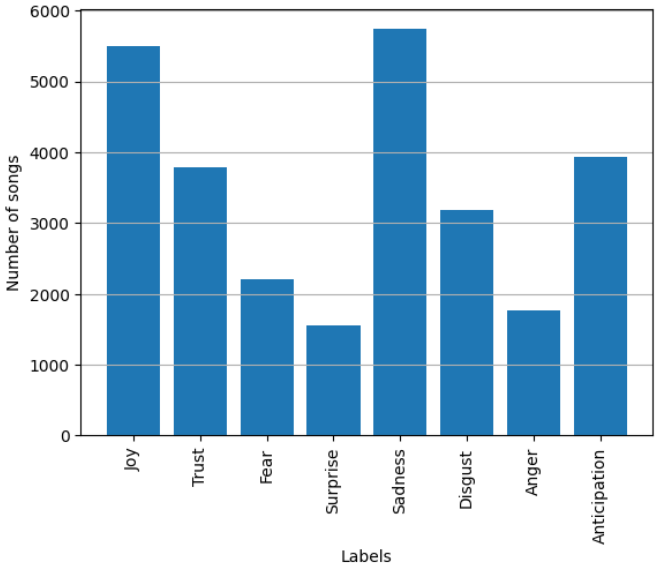

Figure 81: Final distribution of emotional labels in monolabel dataset.

As stated in Section 4.1, a criticality embedded in the data labeling problem is the lack of validation possibilities to check the quality of the extracted tags. One of the few possible workaround to this is verifying which sets of emotions frequently appear together using the Apriori algorithm[1]. After setting minimum support and confidence values to 0.05, which seems a reasonably low threshold, the only couples of emotions that frequently appear together are trust and sadness and sadness and anticipation. This result is somehow reassuring: if the

[1]Frequent item set mining algorithm.

APPENDIX (continued)

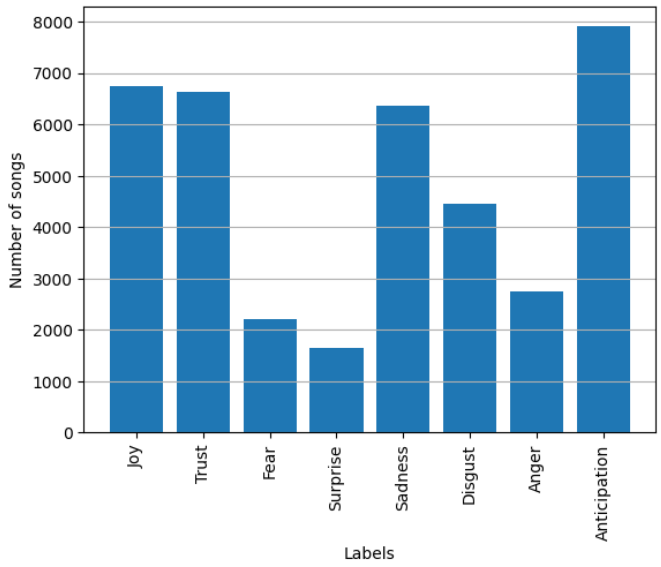

Figure 82: Final distribution of emotional labels in multilabel dataset.

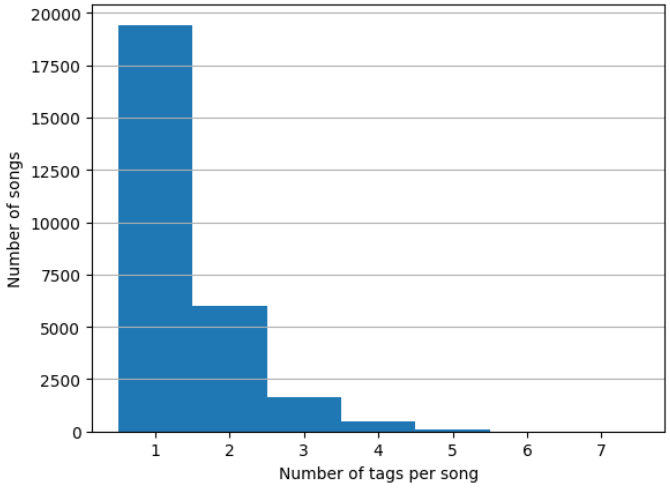

Figure 83: Final distribution of emotion tags per song.

APPENDIX (continued)

frequent co-occurrence of opposite tags, such as joy and sadness, was identified, that would have been an indicator of unreliability.

CITED LITERATURE

1. Lonsdale, A. J. and North, A. C.: Why do we listen to music? a uses and gratifications analysis. British journal of psychology, 102(1):108–134, 2011.

2. Gurgen, E. T.: Social and emotional function of music listening: reasons for listening to music. Eurasian Journal of Educational Research, 16(66):229–242, 2016.

3. Rentfrow, P. J.: The role of music in everyday life: Current directions in the social psychology of music. Social and personality psychology compass, 6(5):402–416, 2012.

4. Covington, P., Adams, J., and Sargin, E.: Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM conference on recommender systems, pages 191–198, 2016.

5. Nikolsky, A.: How emotion can be the meaning of a music work, 01 2016.

6. Corrigall, K.: Music: The language of emotion, pages 299–326. 10 2013.

7. Han, D., Kong, Y., Jiayi, H., and Wang, G.: A survey of music emotion recognition. Frontiers of Computer Science, 16, 12 2022.

8. Aljanaki, A., Wiering, F., and Veltkamp, R.: Emotion based segmentation of musical audio. In Proceedings of the 16th Conference of the International Society for Music Information Retrieval (ISMIR 2015), pages 770–776, 2015.

9. Kang, J. and Herremans, D.: Are we there yet? a brief survey of music emotion prediction datasets, models and outstanding challenges, 2024.

10. Panda, R.: Emotion-based Analysis and Classification of Audio Music. Doctoral dissertation, 01 2019.

11. Gasparetto, A., Marcuzzo, M., Zangari, A., and Albarelli, A.: A survey on text classification algorithms: From text to predictions. Information, 13(2), 2022.

12. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D.: Text classification algorithms: A survey. Information, 10(4):150, 2019.

CITED LITERATURE (continued)

13. Abubakar, H. D., Umar, M., and Bakale, M. A.: Sentiment classification: Review of text vectorization methods: Bag of words, tf-idf, word2vec and doc2vec. SLU Journal of Science and Technology, 4(1):27–33, 2022.

14. Rosenfeld, R.: Two decades of statistical language modeling: Where do we go from here? Proceedings of the IEEE, 88(8):1270–1278, 2000.

15. Fan, Y.: Music Mood Classification Based On Lyrics and Audio Tracks. Master's thesis, University of North Carolina, 2017.

16. Hu, X. and Downie, J.: When lyrics outperform audio for music mood classification: A feature analysis. pages 619–624, 01 2010.

17. Mikolov, T., Chen, K., Corrado, G., and Dean, J.: Efficient estimation of word representations in vector space, 2013.

18. Pennington, J., Socher, R., and Manning, C.: Glove: Global vectors for word representation. volume 14, pages 1532–1543, 01 2014.

19. Akella, R. and Moh, T.-S.: Mood classification with lyrics and convnets. In 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), pages 511–514, 2019.

20. Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5:135–146, 06 2017.

21. Kornkanya Siriket, Vera Sa-ing, S. K.: Mood classification from song lyric using machine learning. 2021 9th International Electrical Engineering Congress (iEECON), pages 476–478, 2021.

22. Blei, D., Ng, A., and Jordan, M.: Latent dirichlet allocation. volume 3, pages 601–608, 01 2001.

23. Kyriakopoulou, A. and Kalamboukis, T.: Text classification using clustering. In Proceedings of the Discovery Challenge Workshop at ECML/PKDD 2006, pages 28–38. Citeseer, 2006.

CITED LITERATURE (continued)

24. Vr, R., Pillai, A., and Daneshfar, F.: Lyemobert: Classification of lyrics' emotion and recommendation using a pre-trained model. Procedia Comput. Sci., 218(C):1196–1208, January 2023.

25. Hu, X., Downie, J., and Ehmann, A.: Lyric text mining in music mood classification. pages 411–416, 01 2009.

26. Corona, H. and O'Mahony, M.: An exploration of mood classification in the million songs dataset. 07 2015.

27. Devlin, J., Chang, M., Lee, K., and Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805, 2018.

28. Edmonds, D. and Sedoc, J.: Multi-emotion classification for song lyrics. In Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, eds. O. De Clercq, A. Balahur, J. Sedoc, V. Barriere, S. Tafreshi, S. Buechel, and V. Hoste, pages 221–235, Online, April 2021. Association for Computational Linguistics.

29. Arzaghi, S.: Audio pre-processing for deep learning. 12 2020.

30. Lin, W.-C., Sridhar, K., and Busso, C.: Deepemocluster: a semi-supervised framework for latent cluster representation of speech emotions. pages 7263–7267, 06 2021.

31. Sharafi, M., Yazdchi, M., and Rasti, J.: Audio-visual emotion recognition using k-means clustering and spatio-temporal cnn. In 2023 6th International Conference on Pattern Recognition and Image Analysis (IPRIA), pages 1–6, 2023.

32. El Haj, A.: Emotions recognition in audio signals using an extension of the latent block model. Speech Commun., 161(C), June 2024.

33. Elbir, A. and Aydin, N.: Music genre classification and music recommendation by using deep learning. Electronics Letters, 56, 06 2020.

34. Vimal, B., Surya, M., Darshan, Sridhar, V., and Ashok, A.: Mfcc based audio classification using machine learning. pages 1–4, 07 2021.

35. Panda, R., Malheiro, R., and Paiva, R. P.: Novel audio features for music emotion recognition. IEEE Trans. Affect. Comput., 11(4):614–626, October 2020.

CITED LITERATURE (continued)

36. Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., and Serra, X.: Essentia: an audio analysis library for music information retrieval. 11 2013.

37. Panda, R., Malheiro, R., and Paiva, R. P.: Audio features for music emotion recognition: A survey. IEEE Trans. Affect. Comput., 14(1):68–88, January 2023.

38. Seyerlehner, K., Widmer, G., Schedl, M., and Knees, P.: Automatic music tag classification based on blocklevel features. 05 2012.

39. Zhang, K. and Sun, S.: Web music emotion recognition based on higher effective gene expression programming. Neurocomputing, 105:100–106, 04 2013.

40. Palanisamy, K., Singhania, D., and Yao, A.: Rethinking cnn models for audio classification, 2020.

41. Laurier, C., Grivolla, J., and Herrera, P.: Multimodal music mood classification using audio and lyrics. In 2008 Seventh International Conference on Machine Learning and Applications, pages 688–693, 2008.

42. Simonyan, K. and Zisserman, A.: Very deep convolutional networks for large-scale image recognition, 2015.

43. Sarkar, R., Choudhury, S., Dutta, S., Roy, A., and Saha, S.: Recognition of emotion in music based on deep convolutional neural network. Multimedia Tools and Applications, 79, 01 2020.

44. Xue, H., Xue, L., and Su, F.: Multimodal music mood classification by fusion of audio and lyrics. In MultiMedia Modeling, eds. X. He, S. Luo, D. Tao, C. Xu, J. Yang, and M. A. Hasan, pages 26–37, Cham, 2015. Springer International Publishing.

45. Shen, T., Jia, J., Li, Y., Ma, Y., Bu, Y., Wang, H., Chen, B., Chua, T.-S., and Hall, W.: Peia: Personality and emotion integrated attentive model for music recommendation on social media platforms. In AAAI Conference on Artificial Intelligence, 2020.

46. Lykartsis, A. and Kotti, M.: Prediction of user emotion and dialogue success using audio spectrograms and convolutional neural networks. pages 336–344, Stockholm, Sweden, September 2019. Association for Computational Linguistics.

CITED LITERATURE (continued)

47. Mohammad, S. M. and Turney, P. D.: Crowdsourcing a word-emotion association lexicon, 2013.

48. Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S.: On the dangers of stochastic parrots: Can language models be too big? . In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery.

49. Russell, J.: A circumplex model of affect. Journal of personality and social psychology, 39(6):1161–1178, 1980.

50. Thayer, R.: The Biopsychology of Mood and Arousal. Oxford University Press, 1990.

51. Dipaola, S. and Arya, A.: "affective communication remapping in musicface system". 01 2004.

52. Russell, J. A. and Mehrabian, A.: Evidence for a three-factor theory of emotions. Journal of Research in Personality, 11(3):273–294, 1977.

53. Ekman, P.: Emotions in the human face. Cambridge University Press, 1982.

54. Hevner, K.: Experimental studies of the elements of expression in music. The American Journal of Psychology, 48(2):246–268, 1936.

55. Plutchik, R.: A general psychoevolutionary theory of emotion. In Theories of Emotion, eds. R. Plutchik and H. Kellerman, pages 3–33. Academic Press, 1980.

56. Moscati, M., Parada-Cabaleiro, E., Deldjoo, Y., Zangerle, E., and Schedl, M.: Music4all-onion - A large-scale multi-faceted content-centric music recommendation dataset. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022, eds. M. A. Hasan and L. Xiong, pages 4339–4343. ACM, 2022.

57. Santana, I. A. P., Pinhelli, F., Donini, J., Catharin, L., Mangolin, R. B., e Gomes da Costa, Y. M., Feltrim, V. D., and Domingues, M. A.: Music4all: A new music database and its applications. 27th International Conference on Systems, Signals and Image Processing (IWSSIP 2020), pages 1–6, 2020.

58. Seyerlehner, K. and Schedl, M.: Block-level audio features for music genre classification.

CITED LITERATURE (continued)

59. Seyerlehner, K., Widmer, G., Schedl, M., and Knees, P.: Automatic music tag classification based on blocklevel features. 05 2012.

60. Shen, T., Jia, J., Li, Y., Ma, Y., Bu, Y., Wang, H., Chen, B., Chua, T.-S., and Hall, W.: Peia: Personality and emotion integrated attentive model for music recommendation on social media platforms. Proceedings of the AAAI Conference on Artificial Intelligence, 34(01):206–213, Apr. 2020.

61. Eyben, F., Weninger, F., Gross, F., and Schuller, B.: Recent developments in opensmile, the munich open-source multimedia feature extractor. In Proceedings of the 21st ACM International Conference on Multimedia, MM '13, page 835–838, New York, NY, USA, 2013. Association for Computing Machinery.

62. Strapparava, C. and Valitutti, A.: Wordnet-affect: an affective extension of wordnet. Vol 4., 4, 01 2004.

63. Thrun, S. and Pratt, L.: Learning to Learn: Introduction and Overview, pages 3–17. Boston, MA, Springer US, 1998.

64. Hutchinson, M. L., Antono, E., Gibbons, B. M., Paradiso, S., Ling, J., and Meredig, B.: Overcoming data scarcity with transfer learning, 2017.

65. Edmonds, D. and Sedoc, J.: Multi-emotion classification for song lyrics. In Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, eds. O. De Clercq, A. Balahur, J. Sedoc, V. Barriere, S. Tafreshi, S. Buechel, and V. Hoste, pages 221–235, Online, April 2021. Association for Computational Linguistics.

66. Provost, F. J.: Machine learning from imbalanced data sets 101. 2008.

67. Araken M Santos, Anne M P Canuto, A. F. N.: A comparative analysis of classification methods to multi-label tasks in different application domains. International Journal of Computer Information Systems and Industrial Management Applications, 3:10, Jan. 2011.

68. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I.: Attention is all you need, 2023.

69. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen,

CITED LITERATURE (continued)

M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T.: Llama 2: Open foundation and fine-tuned chat models, 2023.

70. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V.,

CITED LITERATURE (continued)

Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala,

CITED LITERATURE (continued)

S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z.: The llama 3 herd of models, 2024.

71. Team, G., Riviere, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., Ferret, J., Liu, P., Tafti, P., Friesen, A., Casbon, M., Ramos, S., Kumar, R., Lan, C. L., Jerome, S., Tsitsulin, A., Vieillard, N., Stanczyk, P., Girgin, S., Momchev, N., Hoffman, M., Thakoor, S., Grill, J.-B., Neyshabur, B., Bachem, O., Walton, A., Severyn, A., Parrish, A., Ahmad, A., Hutchison, A., Abdagic, A., Carl, A., Shen, A., Brock, A., Coenen, A., Laforge, A., Paterson, A., Bastian, B., Piot, B., Wu, B., Royal, B., Chen, C., Kumar, C., Perry, C., Welty, C., Choquette-Choo, C. A., Sinopalnikov, D., Weinberger, D., Vijaykumar, D., Rogozińska, D., Herbison, D., Bandy, E., Wang, E., Noland, E., Moreira, E., Senter, E., Eltyshev, E., Visin, F., Rasskin, G., Wei, G., Cameron, G., Martins, G., Hashemi, H., Klimczak-Plucińska, H., Batra, H., Dhand, H., Nardini, I., Mein, J., Zhou, J., Svensson, J., Stanway, J., Chan, J., Zhou, J. P., Carrasqueira, J., Iljazi, J., Becker, J., Fernandez, J., van Amersfoort, J., Gordon, J., Lipschultz, J., Newlan, J., yeong Ji, J., Mohamed, K., Badola, K., Black, K., Millican, K., McDonell, K., Nguyen, K., Sodhia, K., Greene, K., Sjoesund, L. L., Usui, L., Sifre, L., Heuermann, L., Lago, L., McNealus, L., Soares, L. B., Kilpatrick, L., Dixon, L., Martins, L., Reid, M., Singh, M., Iverson, M., Görner, M., Velloso, M., Wirth, M., Davidow, M., Miller, M., Rahtz, M., Watson, M., Risdal, M., Kazemi, M., Moynihan, M., Zhang, M., Kahng, M., Park, M., Rahman, M., Khatwani, M., Dao, N., Bardoliwalla, N., Devanathan, N., Dumai, N., Chauhan, N., Wahltinez, O., Botarda, P., Barnes, P., Barham, P., Michel, P., Jin, P., Georgiev, P., Culliton, P., Kuppala, P., Comanescu, R., Merhej, R., Jana, R., Rokni, R. A., Agarwal, R., Mullins, R., Saadat, S., Carthy, S. M., Cogan, S., Perrin, S., Arnold, S. M. R., Krause, S., Dai, S., Garg, S., Sheth, S., Ronstrom, S., Chan, S., Jordan, T., Yu, T., Eccles, T., Hennigan, T., Kocisky, T., Doshi, T., Jain, V., Yadav, V., Meshram, V., Dharmadhikari, V., Barkley, W., Wei, W., Ye, W., Han, W., Kwon, W., Xu, X., Shen, Z., Gong, Z., Wei, Z., Cotruta, V., Kirk, P., Rao, A., Giang, M., Peran, L., Warkentin, T., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Sculley, D., Banks, J., Dragan, A., Petrov, S., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya, E., Borgeaud, S., Fiedel, N., Joulin, A., Kenealy, K.,

CITED LITERATURE (continued)

Dadashi, R., and Andreev, A.: Gemma 2: Improving open language models at a practical size, 2024.

72. Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms, 2023.

73. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W.: Lora: Low-rank adaptation of large language models, 2021.

74. OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D.,

CITED LITERATURE (continued)

Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B.: Gpt-4 technical report, 2024.

75. Xie, J. and Qiu, Z.: The effect of imbalanced data sets on lda: A theoretical and empirical analysis. Pattern Recognition, 40(2):557–562, 2007.

76. Snoek, J., Larochelle, H., and Adams, R. P.: Practical bayesian optimization of machine learning algorithms, 2012.

77. Pearson, K.: Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin philosophical magazine and journal of science, 2(11):559–572, 1901.

78. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory, 28(2):129–137, 1982.

79. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press, 1967.

80. van der Maaten, L. and Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research, 9(86):2579–2605, 2008.

81. McInnes, L., Healy, J., and Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

82. DeMers, D. and Cottrell, G.: Non-linear dimensionality reduction. In Advances in Neural Information Processing Systems, eds. S. Hanson, J. Cowan, and C. Giles, volume 5. Morgan-Kaufmann, 1992.

83. Goodfellow, I., Bengio, Y., and Courville, A.: Deep Learning. MIT Press, 2016. `http://www.deeplearningbook.org`.

CITED LITERATURE (continued)

84. Caliński, T. and JA, H.: A dendrite method for cluster analysis. Communications in Statistics - Theory and Methods, 3:1–27, 01 1974.

85. Davies, D. L. and Bouldin, D. W.: A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence, (2):224–227, 1979.

86. Zeng, G.: A unified definition of mutual information with applications in machine learning. Mathematical Problems in Engineering, 2015(1):201874, 2015.

87. Chen, T. and Guestrin, C.: Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794, 2016.

88. Ribeiro, M. T., Singh, S., and Guestrin, C.: "why should i trust you?": Explaining the predictions of any classifier, 2016.

89. Chaquet, J., Gimeno, J., Moral-Rubio, S., Muñoz-Romero, S., and Rojo-Álvarez, J.-L.: On the black-box challenge for fraud detection using machine learning (ii): Nonlinear analysis through interpretable autoencoders. Applied Sciences, 12:3856, 04 2022.

90. Hu, X., Downie, J., and Ehmann, A.: Lyric text mining in music mood classification. pages 411–416, 01 2009.

91. Miller, G. A.: Wordnet: a lexical database for english. Commun. ACM, 38(11):39–41, November 1995.

92. Valitutti, A., Strapparava, C., and Stock, O.: Developing affective lexical resources. PsychNology J., 2:61–83, 2004.

93. Gall, J., Yao, A., Razavi, N., Van Gool, L., and Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(11):2188–2202, 2011.

94. Lee, J., Lee, K., Park, J., Park, J., and Nam, J.: Deep content-user embedding model for music recommendation, 2018.

95. Kanwal, S. and Asghar, S.: Speech emotion recognition using clustering based ga-optimized feature set. IEEE Access, PP:1–1, 09 2021.

CITED LITERATURE (continued)

96. Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., and Gao, J.: Large language models: A survey, 2024.

97. Campello, R. J. G. B., Moulavi, D., and Sander, J.: Density-based clustering based on hierarchical density estimates. In Advances in Knowledge Discovery and Data Mining, eds. J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, pages 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

VITA

NAME	Eleonora Quaranta
EDUCATION	
	Master of Science in Computer Science, University of Illinois at Chicago, May 2025, USA
	Master's Degree in Data Science and Engineering, April 2025, Polytechnic of Turin, Italy
	Bachelor's Degree in Management Engineering, September 2022, Polytechnic of Turin, Italy
LANGUAGE SKILLS	
Italian	Native speaker
English	Full working proficiency
	2022 - IELTS examination - band 8.0
	2018 - Cambridge English: Advance (CAE) - C2 level
	A.Y. 2023/24 One Year of study abroad in Chicago, Illinois
	A.Y. 2022/23. Lessons and exams attended exclusively in English
SCHOLARSHIPS	
Fall 2023	Polytechnic of Turin's scholarship for TOP-UIC students
TECHNICAL SKILLS	
Basic level	docker, Kubernetes, C programming, HTML, CSS
Average level	Java programming, database management, SQL, NoSQL, Hadoop, Apache Spark, MATLAB
Advanced level	Python programming, machine learning, deep learning, TensorFlow, Keras, PyTorch, Scikit-learn, NumPy, Pandas, Weights and Biases, data visualization, Tableau, ETL processes, data governance, technical writing, literature review

VITA (continued)

WORK EXPERIENCE AND PROJECTS

Spring 2022	Curricular Internship at Estia S.p.A., Turin, Italy
	Performed data analysis for cost optimization in different spending areas
Fall 2022	Development of a pipeline for spoken language intent detection
	Spectral analysis of audio tracks containing instructions for a smart home assistant with the purpose of intent classification
Spring 2023	Development of a model for semantic segmentation for autonomous driving in a federated learning setting
	Usage of deep learning and computer vision methodologies and Python libraries, extensive literature review on state-of-the-art methods, creation of custom loss function and new batch normalization method to incorporate style information extracted from the dataset and improve prediction accuracy.
Fall 2023	Other projects:
	Conduction of a reproducibility study of a published paper on multimodal sarcasm detection
	Development of a Python pipeline for political sentiment analysis on tweets regarding the 2012 U.S. elections
	Exploration of new causal discovery methods in the presence of selection bias, introducing techniques from ensemble methods into the existing methods
	Usage of various AI technologies to tackle real-world fraud detection task, including rule-based systems, Bayesian networks, influence diagrams and large language models