UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Informatics

Emotion recognition system through voice for the reproduction of emotionally tuned music

Master's Thesis

Master's Degree in Informatics Engineering

AUTHOR: Zaragozá Portolés, Miguel

Tutor: Juan Lizandra, María Carmen

External cotutor: SAITO, DAISUKE

ACADEMIC YEAR: 2022/2023

Dedication

To my dear parents and sister, whose love and constant support have guided me through the most trying moments, thank you for giving me the education and the basis to develop myself.

To my friends and colleagues from Valencia, with whom I have shared long hours of work. Thank you for the memories we have built.

To my tutors, both M. Carmen, guiding and assisting me from the Universitat Politècnica de València, and Daisuke Saito, who has allowed me to work with artificial intelligence and emotional recognition at The University of Tokyo. I am very grateful for their patience, wisdom, and support.

To my friends at the Tokyo residence, Hakusan House, with whom I have shared and discovered the wonders of Tokyo. Thank you for being my home in a different country.

Acknowledgements

This master's thesis has been developed during an academic exchange at the University of Tokyo financed by the "PROMOE" grant from Santander Bank.

Resumen

El proyecto realizado consiste en el análisis, diseño e implementación de un sistema orientado a la ayuda a personas con trastorno del espectro autista (TEA). Utilizando como entrada la voz, se deberá reconocer las emociones transmitidas a través de esta y reproducir o generar música en respuesta.

Sabemos que la interpretación de emociones puede ser una tarea complicada para personas que presentan estos tipos de trastornos y, a menudo, en contextos sociales, puede significar situaciones incómodas y malentendidos entre ambas partes. Esto puede ocasionar fisuras en sus relaciones interpersonales y una clara desigualdad respecto a personas neurotípicas.

Este TFM, nace de la combinación de diversas áreas, como son la informática gráfica, la inteligencia artificial, el reconocimiento emocional en la voz humana y la musicoterapia. Está orientado especialmente hacia personas que se encuentran en el espectro del autismo, aunque puede extrapolarse a otros campos de la musicoterapia (p.ej., depresión, estrés, ansiedad, Alzheimer, entre otros). Para ello, se utilizarán algoritmos avanzados basados en las redes neuronales que permitan identificar matices emocionales y experimentar la música generada como respuesta.

En primera instancia, el objetivo del sistema es reducir desigualdades entre personas, mejorar la inserción social de individuos con TEA, asistir en su bienestar y motivar la investigación en los campos mencionados anteriormente. También, se contempla la inclusión de sistemas similares en plataformas de *streaming* de música, concretamente en su apartado de recomendaciones.

Palabras clave: reconocimiento, emociones, voz, reproducción, generación, música, trastorno, autista, TEA, algoritmos, redes neuronales, informática gráfica, inteligencia artificial, musicoterapia.

Abstract

The performed project revolves around the analysis, design, and implementation of a system oriented to help people with autism spectrum disorder (ASD). Using the voice as input, it should recognize the emotions transmitted and play or generate music in response.

We know that interpreting emotions can be difficult for people with these types of disorders and, often in social contexts, can mean awkward situations and misunderstandings between both parties. This can cause cracks in their interpersonal relationships and an evident inequality concerning neurotypical people.

This TFM is born from the combined knowledge of different areas, such as computer graphics, artificial intelligence, emotional recognition in the human voice, and music therapy. It is mainly oriented towards people in the autism spectrum, although it can be extrapolated to other fields of music therapy (e.g., depression, stress, anxiety, Alzheimer's, among others). For this purpose, advanced algorithms based on neural networks will be used to identify emotional nuances and experience the music generated in response.

In the first instance, the purpose of the system is to reduce inequalities between people, improve the social insertion of individuals with ASD, assist in their welfare, and motivate research in the preceding fields. The inclusion of similar systems in music streaming platforms is also contemplated, specifically in its recommendations section.

Keywords: recognition, emotions, voice, playback, generation, music, disorder, autistic, ASD, algorithms, neural networks, computer graphics, artificial intelligence, music therapy.

Resum

El projecte realitzat consisteix en l'anàlisi, disseny i implementació d'un sistema orientat a l'ajuda a persones amb trastorn de l'espectre autista (TEA). Utilitzant com a entrada la veu, s'haurà de reconèixer les emocions transmeses a través d'aquesta i reproduir o generar música en resposta.

Sabem que la interpretació d'emocions pot ser una tasca complicada per a persones que presenten aquests tipus de trastorns i, sovint, en contextos socials, pot significar situacions incòmodes i malentesos entre totes dues parts. Això pot ocasionar fissures en les seues relacions interpersonals i una clara desigualtat respecte a persones neurotípiques.

Aquest TFM, naix de la combinació de diverses àrees, com són la informàtica gràfica, la intel·ligència artificial, el reconeixement emocional en la veu humana i la musicoteràpia. Està orientat especialment cap a persones que es troben en l'espectre de l'autisme, encara que pot extrapolar-se a altres camps de la musicoteràpia (p. ex., depressió, estrès, ansietat, Alzheimer, entre altres). Per això, s'utilitzaran algorismes avançats basats en les xarxes neuronals que permeten identificar matisos emocionals i experimentar la música generada com a resposta.

En primera instància, l'objectiu del sistema és reduir desigualtats entre persones, millorar la inserció social d'individus amb TEA, assistir en el seu benestar i motivar la investigació en els camps esmentats anteriorment. També, es contempla la inclusió de sistemes similars en plataformes de *streaming* de música, concretament en el seu apartat de recomanacions.

Paraules clau: reconeixement, emocions, veu, reproducció, generació, música, trastorn, autista, TEA, algorismes, xarxes neuronals, informàtica gràfica, intel·ligència artificial, musicoteràpia.

Index of Contents

Index of figures

Index of tables

1. Introduction

Autism, or autism spectrum disorder (ASD), is a chronic neurodevelopmental disorder that manifests during the initial stages of childhood and involves countless disorders, including impaired sociability, social cognition, communication, and motor and cognitive skills [1] [2] [3]. The German psychiatrist Eugen Bleuler used the concept of autism in 1911 to refer to a symptom of the most severe cases of schizophrenia [4]. In his definition, Bleuler, categorized autistic thinking as infantile dreaming and replacing unsatisfactory reality with fantasies. Later, in 1925, child psychiatrist Grunya Sukhareva detailed the symptoms associated with this condition, giving importance to a treatment that also involves familial and systematic education for people with ASD [5].

The definition of autism changed in the 1960s when many British psychologists challenged previously coined definitions and created new methods to validate child psychology as a science [6]. The interpretation of autism was completely reformulated and came to describe the opposite of what it had previously meant. According to Michael Rutter, the autistic child has a deficiency of fantasy rather than an excess [7]. More current studies indicate, however, that people with ASD are, on average, only slightly less accurate and slower in sorting out the emotions of others [8]. In Spain alone, it is estimated that about 83800 people (aged six years and older) have ASD[1].

This project is a collaboration with Professor Daisuke Saito of the Department of Electrical Engineering and Information Systems (EEIS) of the Graduate School of Engineering at The University of Tokyo; intending to integrate neural network-based models with the recognition of emotional nuances in the voice and the generation of emotionally tuned music. The system will allow users to detect the feelings transmitted through the human voice and listen to a song that matches those emotions. This matching will be done through quantitative values called valence and arousal.

Valence and arousal are two dimensions used to describe emotions. The former refers to the level of liking that an event generates. It is a continuous value that can be either negative or positive. On the other hand, arousal is the level of activation that the event produces [9]. When the value is low, tranquility is conveyed. As it increases, the sentiment is more enthusiastic. In summary, valence represents whether the emotion is positive or negative, and arousal represents the intensity of the emotion.

Every emotion can be quantified using these values. Therefore, the reactions that a person may experience while listening to a song contain valence and arousal. Fields such as musicology, cognitive science, and psychology, among others, study emotional recognition in music (MER) [10]. There are approaches called dimensional emotional recognition in music that try to model the emotions conveyed by music in continuous variables. They are usually formulated as regression problems. Using scales, listeners rate the emotions they feel while listening to a piece. To this information can be added the recording and analysis of brain waves.

[1] https://www.ine.es/index.htm

Motivation

The choice to develop this application capable of recognizing emotions and generating music in tune is due to the certainty that it can become a crucial tool to improve the quality of life of people with ASD, making social interaction with their peers more accessible and helping the development of their emotional self-awareness.

The motivation to develop this project stems from the possibility of applying a large amount of the knowledge acquired during the entire formative period, both the undergraduate and the master's degree, especially those obtained during the exchange, and to lead them to a professional level praxis. The specific reasons are detailed below:

1.- Assimilate specific technologies for voice analysis, advanced signal processing algorithms, and machine learning based on neural network models, both for the emotion recognition module and the music generation system. Using these tools can motivate the research to different applications, such as individualized therapies or group sessions, from an educational point of view.

2.- During the educational stage, we acquired a deep understanding of the different development methodologies, technologies, algorithms, and project management. This base allows us to respond to the present needs of society, such as developing applications and tools that facilitate and improve the quality of life of people with autism.

3.- Apply the knowledge learned during the master's degree and the academic exchange at the University of Tokyo, working with various architectures, regression models, and regularizers to create and train models. We will use libraries for machine learning for the first time. Therefore, facing it is seen as a personal challenge.

4.- To develop a system for psychotherapeutic sessions in which, by using voice recognition and the generation of music intoned with these emotions, people with ASD can improve their interpretation of social situations.

5.- Evaluate models using metrics, such as accuracy, mean squared error, and loss function. The music generated will be evaluated by employing surveys.

6.- Analyze the results, considering the data obtained from the previous evaluation and comparing them with other published studies using emotional classification.

Each of these motivations helps us to give way to the justification of the project and the design of the objectives to be achieved with this master thesis.

1.1 Justification

According to the World Health Organization (WHO), one out of every hundred children have autism spectrum disorder[2]. This number demonstrates the rising demand for tools and strategies to help this population communicate and comprehend one another. Several studies suggest that persons with autism may struggle to recognize and understand their own and others' emotions, which can create considerable issues in their everyday lives.

This tool may be provided to therapists and educators as a beneficial supplement to their treatments, giving an interactive and engaging approach to work on emotional understanding. Eventually, incorporating this technology into educational and therapeutic programs might enhance persons with ASD's quality of life and social integration.

1.3 Objectives

The general objective of this project is to develop a system that assists people with autism spectrum disorders by analyzing and recognizing emotions expressed through the voice, followed by creating musical compositions adapted to these emotions. This system will be used in therapeutic and educational sessions.

For this purpose, we plan to implement advanced audio processing algorithms, as well as the adoption of machine learning techniques. With this, the aim is to boost the accuracy of emotional recognition and improve the ability to generate contextualized music.

To achieve this goal, we must fulfill the following specific objectives:

- To conduct a literature study on emotional recognition in autistic people and the impact of music on emotional modulation.
- To develop speech recognition algorithms that can accurately extract emotional quantitative values, valence, and arousal.
- Implement a music generation system that can respond to data collected from speech analysis.
- Conduct pilot tests with a representative sample of users to ensure the tool's effectiveness.
- Analyze the data obtained from the pilot tests and adjust the system according to the recommendations detected.

Each of the objectives, both general and specific, is fundamental and must be achieved to fulfill the purpose outlined.

1.4 Methodology

Given the project's complexity and the availability of only one programmer, we deemed it reasonable to use an agile approach that would allow for gradual, effective, and efficient development. It identified the necessity for quick and continuous iterations and

[2] https://www.who.int/news-room/fact-sheets/detail/autism-spectrum-disorders

progressive product deliveries to test established functionality and optimize product delivery satisfaction. As a result, we set the time limit for each iteration at one week.

A strategy that emphasizes adaptation and continuous product improvement was chosen, which is one of the goals of the Scrum methodology[3]. The product progresses incrementally, and each "Sprint" allows the product to evolve and improve dramatically.

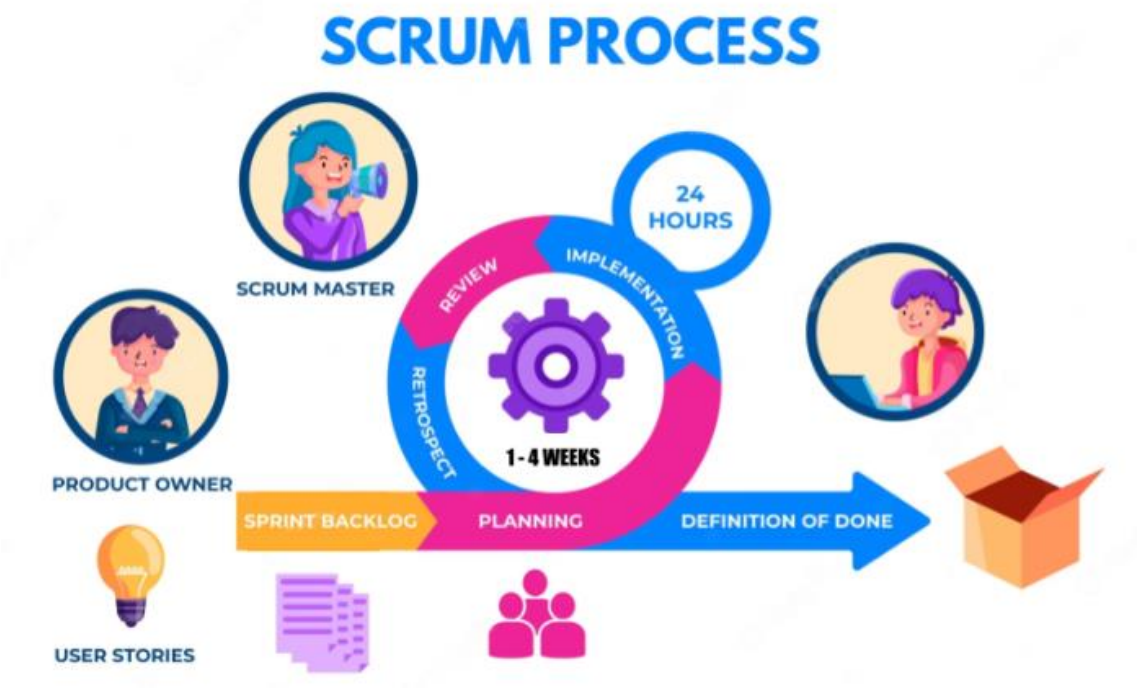

Figure 1.- Scrum process diagram.

The Scrum methodology, as seen in figure 1, is built on early product requirement segmentation and establishing a task priority that controls the sequence of development. We may accomplish this by putting the tasks or product requirements in a backlog. Each iteration contains those requirements with the highest priority and will then move through the steps of analysis, design, implementation, and testing. If a problem is reported in any unit of work during the weekly review, that unit is moved to the next sprint and reviewed from the corresponding phase, depending on the anomaly.

If we achieve satisfaction, we will select the upcoming work units with the highest priority in the backlog to implement for the subsequent sprint.

In the figure 2, we can see the backlog with the different work units of the project ordered by importance, the highest in the pyramid being the ones with the highest priority and urgency. When they are removed from the backlog, we can observe how they follow a workflow represented by a Kanban board, in which each work unit must pass through all the stages stated on the board.

[3] https://www.atlassian.com/es/agile/scrum

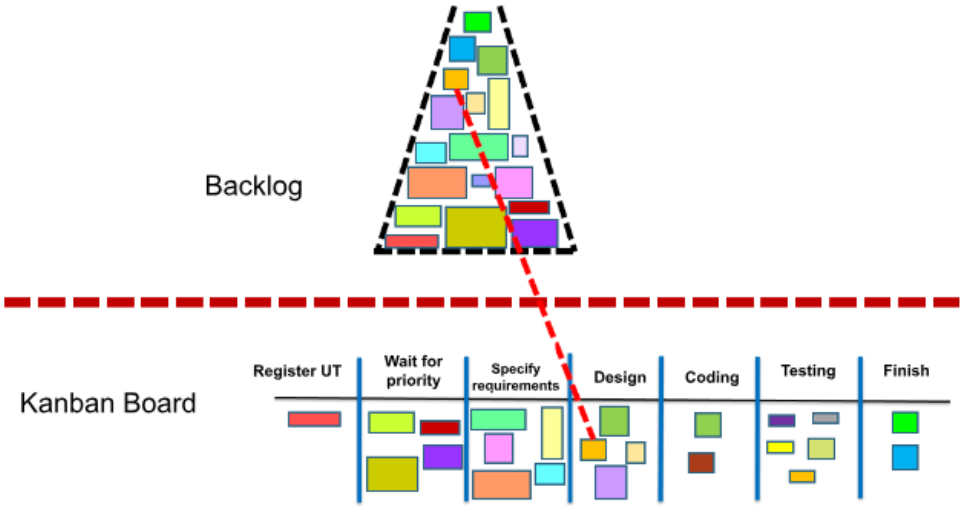

Figure 2.- Distribution of work units in the Scrum Methodology.

This process, together with the short time of evaluation of the requirements specified over the week, makes it easy to spot anomalies and evaluate satisfaction with the product's results. Otherwise, a longer duration would suggest more repairs due to dissatisfaction and errors, causing an increment in working hours in a project with a single programmer. It also enables effective management of complex objectives and agile adjustments. The Scrum approach offers a clear advantage by reducing the risks associated with dissatisfaction and errors through constant reviews and timely adjustments.

The suitability of the Scrum methodology lies in its ability to effectively manage projects with complex and variable objectives while allowing agile adjustments according to emerging realities. This approach offers a critical advantage by mitigating the risks associated with dissatisfaction and errors through constant reviews and timely adjustments. Consequently, its adoption as a working methodology was essential to ensure a controlled and efficient development process aimed at obtaining a quality product that meets the needs and expectations of the project.

1.5 Memory structure

Following this first introductory chapter, in which we discussed the environment in which the final degree project is developed, as well as the motivation, justification, objectives, and methodology, a second chapter, State of the Art, is presented, in which the theory on audio processing algorithms, machine learning, languages and environments used in the development, as well as a technical comparison of other similar models, are discussed. The third chapter reflects the formal statement of requirements and explores the application's functional design. The fourth chapter explains the implementation, focusing on some of the routines and crucial parts of the implementation, and the fifth chapter details the validation, explaining how the research was carried out, the methods, and the analysis of the data acquired. Finally, in the sixth and seventh chapters, we provide the study's conclusions and the bibliography employed in this work.

Finally, there is an appendix with a glossary of essential terminology used in the TFM.

2. State of art

In this section, we present a review of the literature on emotion recognition models, music generation, and applications aimed at assisting the communication of people with ASD, paying particular attention to those with functionalities that focus on helping the user to improve their social integration. We will analyze the applications that serve this purpose, evaluating their advantages and disadvantages and highlighting the significant advantages of our system over its competitors and the aspects of the latter that could be incorporated to increase the usefulness of our prototype.

Likewise, we will review the state of the art of development tools, such as the development and execution environments themselves, to obtain the best possible implementation, using these technologies in the most efficient way and with the least possible friction.

2.1 Emotion recognition models

There are various technologies currently employed for emotion recognition, each with distinct methodologies. In the forthcoming discourse, we will delve into these approaches and explore specific technological instances associated with each one. Given that the speech recognition model serves as the foundation for this project's emotion recognition model, we will dedicate particular emphasis to this segment.

2.1.1 Natural Language Processing (NLP)

IBM Watson Natural Language Understanding

IBM[4] offers an innovative solution that extends beyond merely pinpointing keywords in a text. This innovative tool dismantles linguistic structures to achieve a more profound comprehension of the content at hand. Besides discerning emotions and sentiments, Watson's Natural Language Understanding is also capable of identifying entities, concepts, categories, and so much more. It boasts extensive usage across various business intelligence applications, social network sentiment analysis endeavors, and chatbot implementations.

Google Cloud Natural Language API

The Google Cloud artificial intelligence toolkit encompasses the inclusion of the natural language API[5]. This tool not only performs sentiment analysis but also can recognize entities, determine content categories, and decode the syntactic structure of the text. Its proficiency in comprehending expressed emotions makes it well-suited for scrutinizing user feedback, monitoring brand perception, and enhancing customer interaction within text-based applications.

https://www.ibm.com/products/natural-language-understanding
[5] https://cloud.google.com/natural-language/docs/reference/rest

VADER (Valence Aware Dictionary and sEntiment Reasoner)

VADER stands as an open-source instrument for analyzing sentiment, with a particular aptitude for deciphering texts originating from social networks [11]. Unlike alternative tools, VADER holds the capability to discern content that is both explicit and implicit while considering emoticons, colloquialisms, and slang terms commonly utilized on social media platforms. By merging VADER's word dictionary with a comprehensive collection of grammar and syntax rules, this tool demonstrates tremendous accuracy in gauging the intensity of sentiments. Therefore, it proves remarkably effective in situations where achieving the appropriate interpretation of sentiment relies heavily upon tonality and context optimization.

2.1.2 Analysis of Facial Expressions

In this part, we explore emotion recognition technologies using visual data applications. These innovative systems employ computer vision algorithms to identify and analyze various emotions exhibited through facial expressions. Some notable instances where this type of recognition is implemented include Microsoft Azure Face API and Affectiva.

Microsoft Azure Face API

Microsoft Azure Cognitive Services provides a solution known as Face API[6], which is focused on facial recognition and its various applications. One of the crucial features of this API is its ability to detect emotions, allowing it not only to identify individuals but also to differentiate between an extensive range of human emotions, including happiness, sadness, anger, contempt, and more. Additionally, the Face API can execute supplementary functions in relation to facial recognition, including validating one's identity and conducting searches for comparable faces within a given database. It also has the capability to analyze groups based on shared characteristics. The accuracy of this API is achieved through leveraging artificial intelligence and machine learning technologies that have been trained using large datasets consisting of facial images.

Affectiva

Affectiva stands out as a leader in emotional analysis powered by computer vision. It relies on advanced algorithms and deep learning techniques to comprehend and assess human emotions through facial expressions[7]. Common devices like cameras and webcams are utilized by Affectiva to conduct real-time analysis of a wide array of emotions, encompassing sadness, contempt, joy, surprise, and more. To enhance the accuracy of its detection capabilities, Affectiva considers not only facial features and expressions but also head movements along with other gestures. This cutting-edge technology has discovered numerous applications across various industries, such as market research, user experience analysis, mental health services, as well as interactions with robots and virtual assistants.

[6] https://learn.microsoft.com/es-es/azure/ai-services/computer-vision/overview-identity
[7] https://www.affectiva.com/

2.1.3 Voice and Tone Detection

It is feasible to identify tools that analyze the emotions transmitted in one's speech by using this approach. Through evaluating factors such as pitch and its fluctuations, as well as other characteristics of vocal communication, these technologies enable the identification of expressed feelings. Several instances showcasing voice emotion recognition include IBM Watson Tone Analyzer and Beyond Verbal.

IBM Watson Tone Analyzer

IBM Watson Tone Analyzer[8] is a highly innovative tool created by IBM as part of its artificial intelligence solutions. Its primary function is to examine texts and voices to identify and distinguish various emotional tones, including joy, sadness, and confidence. Through focused analysis of one's speech, this analyzer assesses factors like modulation, pitch, and cadence to gain insights into the underlying emotions conveyed by the speaker's speech. Although it also possesses the ability to analyze texts and decipher the emotions they convey, this feature may not be crucial when it comes to assessing emotions through tone and voice. Nevertheless, its versatility highlights the extensive range of analytical capabilities this tool possesses.

Beyond Verbal

Beyond Verbal[9] sets itself apart in speech analysis by focusing solely on decoding and comprehending emotions through voice and speech. By examining various distinctive characteristics, Beyond Verbal achieved accurate detection. These characteristics include wave frequency, velocity, voice rhythm, intensity, pressure variations, spectrum, and wave envelope. Wave frequency reveals emotions through pitch, while speed changes can indicate nervousness or confidence. Voice rhythm reflects anxiety or calmness based on pace. Fluctuations in loudness unveil levels of enthusiasm or apathy. Timbre provides insight into the unique quality of one's voice, offering further clues about the emotions expressed. Concerning these criteria, Beyond Verbal had the capability to accurately decipher diverse emotions such as anxiety, delight, sorrow, and assurance. As a result, this technology holds immense value in fields including psychotherapy, marketing, and human communication research.

2.1.4 Wav2vec 2.0

Wav2vec 2.0 [21] is an innovative deep-learning model designed to capture rich representations from audio recordings without manual labeling. Its primary purpose is to serve as a tool for extracting features for analyzing emotion in speech.

The Wav2vec 2.0 model comprises several key stages, each playing a vital role:

- Local Encoder: This component divides the audio signal into small windows and utilizes a convolutional neural network (CNN) to extract distinctive features from each segment.

[8] https://www.ibm.com/docs/en/app-connect/cloud?topic=SSTTDS_cloud/com.ibm.appconnect.dev.doc/how-to-guides-for-apps/watson-tone-analyzer.htm
[9] https://www.predictiveanalyticstoday.com/beyond-verbal/

- Global Encoder: Here, a transformative neural network known as Transformer takes over, capturing contextual information within the sequence of feature windows.
- Vector Quantizer: This stage employs a hierarchical clustering algorithm to categorize feature vectors into discrete codes.

Through these interconnected stages, Wav2vec 2.0 can effectively learn high-quality representations directly from raw audio data.

To validate its performance, the authors applied Wav2vec 2.0 to two benchmark datasets - IEMOCAP and RAVDESS. The outcomes attained demonstrate that this approach surpasses other models relying solely on acoustic information when evaluating these specific datasets.

Wav2vec 2.0 was trained to employ a loss function that incorporates contrast loss, diversity loss, and L2 regularization terms. Such a function effectively enables the acquisition of superior representations from unprocessed audio.

Pepino, Riera & Ferrer [12] accentuates a groundbreaking strategy for discerning emotions in a speech via the utilization of the Wav2vec 2.0 model. This particular approach exhibits notable strengths compared to alternatives reliant solely on acoustic data, and it is conceivable that the suggested loss function can be applied to other natural language processing predicaments involving raw audio as well.

2.1.5 Valence and Arousal-Based Detection

Li & Akagi [13] introduce a groundbreaking approach to developing an advanced system capable of detecting emotions in multiple languages. The main objective of their study is to present a system that overcomes linguistic obstacles while surpassing current systems in terms of accuracy and efficiency.

At the core of their study lies the emotion perception model, which can be divided into three integral layers as outlined below:

- Acoustic Elements: This layer is responsible for analyzing and processing different components of speech, such as F0 (fundamental frequency), power envelope, voice quality, power spectrum, and duration. These features play a crucial role in capturing nuances and variations in the pitch and modulation of speech.
- Semantic Underpinnings: Moving beyond mere detection of pitch and modulation, this layer delves into the realm of semantics. Its primary objective is to comprehend the meaningful essence behind words, thereby facilitating a deeper interpretation of emotions.
- Emotional Aspects: This final layer acts as an interpreter where data processed from the previous two layers culminates into distinct emotions.

Built upon human perception and emotional processing mechanisms, this model presents a multidimensional and sophisticated approach to analyzing emotions across diverse languages.

The study emphasized the importance of identifying essential and effective features for detecting emotions in different languages. A wide range of features were carefully examined, including those related to rhythm, sound energy distribution, and vocal fold movements. Through this analysis, it became clear that certain characteristics are universally applicable and not specific to any language. Utilizing these generalized features boosted the accuracy of emotion detection compared to using language-specific feature sets.

Overall, the research supports the proposed framework's effectiveness and potential use for developing advanced multilingual systems capable of translating emotional speech. This discovery has paved the way for more precise and adaptive systems, potentially transforming emotion recognition in speech across a wide range of practical applications.

2.1.6 Interactions and Biometric Sensors

The advancement in technology has introduced fresh opportunities for exploring the correlation between our feelings and physical reactions. By using portable devices equipped with advanced sensors and algorithms, it is now within our reach to monitor and examine diverse physiological reactions including heart rate, skin conductance, and body temperature, among others. We can gain insights into an individual's mental state and emotional disposition by studying the data obtained.

An excellent demonstration of these technological strides can be seen in Fitbit[10], a widely recognized brand celebrated for its state-of-the-art activity-tracking devices. While its primary purpose is to aid users in monitoring their health and fitness objectives, the embedded sensors within Fitbit's devices can also detect alterations in physiological responses linked to emotions. For example, a sudden surge in heart rate or changes in skin conductance may indicate episodes of stress or anxiety.

2.1.7 Artificial Intelligence and Machine Learning

The advent of machine learning has given rise to the development of advanced systems that possess the ability to perceive and comprehend human emotions. These systems rely on algorithms that acquire knowledge from extensive datasets, thereby identifying various emotional patterns within different input types, including text, voice, and images.

This approach is grounded in training machine learning models using meticulously labeled data sets that reflect diverse emotions or moods. Once these models have undergone sufficient training with relevant data, they can novel inputs and accurately determine the most appropriate sentiment that aligns with said input based on previously learned patterns.

Microsoft's Azure Cognitive Services[11] platform stands as a prominent example where this technology finds practical application. Within this suite of tools resides "Text

[10] https://www.fitbit.com/global/es/technology/health-metrics
[11] https://azure.microsoft.com/es-es/products/cognitive-services/

Analytics," a specialized service dedicated to profound textual analysis. One notable feat offered by this service is its proficiency in sentiment detection.

Text Analytics[12], in sentiment detection, goes beyond mere identification of keywords. It employs sophisticated machine learning models that delve into the comprehensive context of a text to categorize it as positive, negative, or neutral accurately. This classification hinges on recognizing the emotional undertones embedded within words and their interplay within sentences or paragraphs.

The potential for leveraging machine learning algorithms in emotion detection is boundless. Advancements go beyond textual analysis alone and encompass systems capable of deciphering tone of voice, facial expressions, and other indicators of emotions. These tools find application in various business settings, enabling measurement of customer satisfaction, comprehension of product reviews, and evaluation of team spirit through internal communication channels.

2.1.8 Chatbots and Virtual Assistants

One of the most captivating implementations of this technology lies in chatbots and virtual assistants. These applications, traditionally employed for answering inquiries or completing specific tasks, now possess the ability to comprehend and react adaptively based on the user's emotional state.

In this scenario, the chatbot not only responds to the query but also acknowledges whether the user feels frustrated, happy, or confused. As a result, it adjusts its tone and replies accordingly. This emotional flexibility dramatically enhances the users' experience by creating a sense of being understood and valued.

IBM has established itself as a pioneering force in this realm through the introduction of its Watson Assistant platform[13]. Going beyond mere chatbot and virtual assistant creation, it goes beyond by offering the unique ability to seamlessly incorporate real-time sentiment analysis into conversations.

One aspect that sets IBM Watson Assistant apart is its capacity to analyze user input, discern the inherent sentiment or emotion conveyed within, and subsequently adapt its response accordingly. Consequently, whenever a user expresses frustration, for example, the chatbot can empathetically react by providing more extensive solutions or adopting a comforting tone.

2.2 Music generation applications

Numerous technologies exist in the market that harness artificial intelligence and machine learning algorithms to create music. These advancements possess the ability to compose melodies, harmonies, and even produce entire music tracks. Among these innovations are several noteworthy ones:

[12] https://azure.microsoft.com/en-us/products/ai-services/text-analytics
[13] https://www.ibm.com/products/watsonx-assistant

OpenAI's MuseNet

MuseNet[14] is a music generation platform developed by OpenAI. It uses deep neural networks to compose music. It can generate four minutes of musical compositions with ten different instruments and combine styles from country to Mozart to the Beatles. MuseNet was not explicitly programmed with knowledge of music but instead discovered patterns of harmony, rhythm, and style by learning to predict the next token in hundreds of thousands of MIDI files. MuseNet uses the same general unsupervised technology as GPT-2, a large-scale transformer model trained to predict the next token in a sequence, whether audio or text.

Some of MuseNet's limitations include: the instruments you request are strong suggestions, not requirements. MuseNet generates each note by calculating probabilities on all possible notes and instruments. The model changes to make your instrument choices more likely, but there is always the possibility that it will choose something else. MuseNet has more difficulty with odd combinations of styles and instruments (like Chopin with bass and drums). Generations will be more natural if instruments closer to the composer's or band's usual style are chosen.

Amper Music

Amper Music[15] is a platform that uses artificial intelligence to generate original music. It allows users to create custom music tracks by adjusting parameters such as genre, tempo, instrumentation, and more. Amper's technology allows anyone to create computer-generated music. Amper Music uses AI models to create all types of music, including music for podcasts, music videos, video game soundtracks, movie soundtracks, etc.

Similar to other music generation systems, Amper operates through deep learning networks, a form of artificial intelligence that depends on analyzing extensive quantities of data. The software is provided with a substantial amount of source material encompassing various genres, such as dance hits and disco classics. Subsequently, this input undergoes analysis to ascertain consistent patterns relating to chords, tempo, duration, and interrelations between notes. By assimilating the gathered information, Amper acquires the ability to compose its own unique melodies.

AIVA (Artificial Intelligence Virtual Artist)

AIVA[16] is a tool that uses specific algorithms to create musical compositions in distinctive styles. It specializes in classical and symphonic music composition. AIVA became the first artificial composer in the world to be recognized by *Société des auteurs, compositeurs et éditeurs de musique* (SACEM).

Its primary purpose is to facilitate the creative process. Utilizing preset styles or external influences, one can produce music by employing algorithms tailored for Modern Cinematic, Electronic, Pop, Ambient, Rock, Fantasy, Jazz, Sea Shanty, 20th

[14] https://openai.com/research/musenet
[15] https://ampermusic.zendesk.com/hc/en-us
[16] https://www.aiva.ai/

Century Cinematic Tango, and even Chinese musical genres. A noteworthy benefit of AIVA lies within its Pro plan as it grants users complete and eternal copyright ownership over all compositions generated via the platform, effectively eliminating any potential licensing complications.

Jukedeck

Jukedeck[17] originated from the University of Cambridge, UK, as a project that employs artificial intelligence to generate original musical pieces. This web-based tool is incredibly user-friendly and has received esteemed recognition, including the Innovation Award at the Cannes Advertising Festival. By inputting specific criteria such as genre, mood, and duration, users were able to swiftly produce unique music compositions for their projects without any cost.

Regrettably, Jukedeck is no longer accessible as it was assimilated by TikTok's owners. In numerous conversations regarding his mission to democratize music composition, CEO Ed Newton-Rex frequently mentioned how TikTok would be an ideal platform to fulfill this aspiration.

Google Magenta

Google's Magenta[18] project is an open-source research that delves into the integration of machine learning within artistic endeavors. Specifically, it emphasizes leveraging artificial intelligence to generate both music and artwork. The overarching goal of the Magenta project lies in highlighting how machine learning can act as a facilitator, empowering individuals across all backgrounds with exceptional creative capabilities. Through its platform, Magenta provides users with a repertoire of tools and models tailored to fuel musical expression and unravel AI-driven domains of creativity.

Magenta Studio, an innovative music production tool powered by Google's innovative AI and machine learning technologies, is now accessible to the general public at no cost. Utilizing sophisticated cognitive systems, Magenta Studio presents itself in two distinct variations: as a plugin integrated with Ableton Live or as a stand-alone application. However, for individuals who have access to Ableton Live, the plugin option surpasses the stand-alone alternative in terms of versatility and customization options.

Magenta does not aim to serve as a comprehensive music production solution, unlike AIVA or Amper Music. Instead, it functions as a valuable tool that can be employed in songwriting or music production processes. For instance, it excels at creating fresh melodic concepts or crafting unique drum tracks.

LANDR

LANDR[19] is a revolutionary music production tool that employs artificial intelligence to assist composers in their creative process. One aspect that distinguishes it from other

[17]. https://en.wikipedia.org/wiki/Jukedeck
[18] https://magenta.tensorflow.org/studio
[19] https://www.landr.com/es/

options is its audio mastering technology, which leverages extensive data collected from countless mastered songs to produce professional-grade sound quality.

Moreover, LANDR provides an extensive array of creative resources for musicians, encompassing uncomplicated yet influential plugins, track generation functions, and automated mixing and mastering capabilities. By collaborating with LANDR, artists can upload their music on popular platforms like Spotify and Apple Music while retaining full copyright ownership. However, LANDR is not available free of charge.

In music generation, various tools like Ekrett Music, Soundraw, Amadeus Code, and Soundful exist. Although these technologies may appear comparable, they exhibit notable distinctions across platforms: certain ones provide MIDI output while others deliver audio. Moreover, their approach to learning differs considerably - some rely solely on data examination for comprehension, while others adhere to coded rules derived from music theory to govern their outputs. It is worth noting that those reliant on data examination could potentially encounter challenges if the available information proves insufficient unless coupled with a foundation in music theory.

2.3 Programming languages

This project is conceived as an application in which model training is needed to enable the generation of music based on speech emotion recognition. For this reason, we can recognize the great need for the use of machine learning tools to train the models with a large dataset.

Machine learning languages are designed to help practitioners develop, train and deploy models effectively and efficiently. Each language has its own approach and advantages. The market offers several options that make this process easier to complete.

Python

Python is widely regarded as one of the programming languages for machine learning. It offers a range of libraries that make it easier to implement algorithms and efficiently train intelligence models. Due to its user nature, it has become the language for developers and researchers alike.

Advantages

Python boasts an array of libraries and frameworks, within its ecosystem. Specifically tailored to machine learning these encompass TensorFlow, Keras, PyTorch scikit learn and XGBoost. These libraries offer to use tools for model development and training.

Pythons' syntax is user friendly and highly readable. It closely resembles language in its writing style making code comprehension effortless and reducing the learning curve for those to the machine learning domain.

One of Pythons greatest advantages lies in its vibrant community and wealth of learning resources. With a network of developers and scientists a plethora of online materials such as tutorials, documentation, books, and discussion forums are readily

available. This makes Python an ideal language for beginners venturing into machine learning.

The flexibility and versatility exhibited by Python are truly remarkable. Beyond its application in machine learning this language finds utility in web development, automation tasks, data analysis endeavors among areas. Its adaptability proves invaluable for projects necessitating integration, with technologies.

Python is capable of seamlessly blending with various other technologies. It has the capability to integrate with databases, version management systems, visualization libraries, and data analysis tools. This enables the creation of comprehensive workflows that encompass all necessary components.

Disadvantages

When compared to compiled languages such as C++ or Java, Python's performance is substantially inferior. Because it is an interpreted language, its execution speed is slower. Nevertheless, various machine-learning libraries have been optimized to enhance the efficiency of numerical operations.

Another drawback of Python lies in its tendency towards high memory consumption. Compared to languages such as C++, it may not exhibit the same level of efficiency in terms of memory usage due to factors like overhead from object systems and memory management.

However, while Python remains excellent for research and development purposes, it may not always be the most suitable choice for production implementations that require high-speed and optimal efficiency. Its performance limitations can hinder its suitability for such scenarios.

Conclusion

To recapitulate, Python is a strongly suggested option for training machine learning models because of its extensive collection of libraries, accessible syntax, and vibrant community. Nevertheless, when it comes to demanding production applications that require significant resources, it might be necessary to explore alternative solutions to maximize performance.

R

R serves as a programming language and development environment primarily focused on statistical analysis and data visualization. Despite Python's widespread application in machine learning, R continues to prevail as a favored option for tackling data analysis and statistical modeling tasks.

Advantages

The creation of the R language was primarily for statistical analysis, thus making it highly proficient in data handling and manipulation. This makes it an excellent alternative for applications involving data exploration and statistical modeling.

The R language possesses a vast collection of libraries and packages specifically designed for tasks involving statistics and data analysis. This feature proves to be particularly advantageous if our project is focused on traditional methods that pertain to statistics, exploratory data analysis, or predictive modeling.

Through packages such as ggplot2, R offers sophisticated abilities regarding data visualization. These capabilities facilitate the creation of superior-quality graphs and visualizations, which aid in understanding the data better while also effectively communicating results.

Comparable to Python, R boasts a large community composed of active users who are readily available to assist. Additionally, there is a wealth of online resources, including forums, tutorials, and documentation geared towards facilitating learning within this programming language universe.

The functions it offers contribute to its effortless manipulation of data which facilitates the cleaning and transformation of data, a crucial step in preparing data sets for model training.

Disadvantages

R is not as focused on deep machine learning as other languages like Python. While it does offer some libraries for machine learning, such as caret and randomForest, it may lack the same level of support for deep machine learning found in frameworks like TensorFlow and PyTorch.

In terms of performance, R can be slower than compiled languages like C++. As an interpreted language, its execution speed might not match that of a compiled language.

R has certain limitations when it comes to production applications. Its lower performance and scalability make it less ideal for implementing large-scale systems. However, R remains suitable for exploratory analysis and statistical modeling tasks.

R features an integrated development environment (IDE) named RStudio when it comes to development tools. However, compared to the Python ecosystem, there may be fewer options available concerning tools and libraries within the R community.

Conclusion

In summary, R is an excellent choice if our primary focus lies in data analysis, statistics, and traditional modeling. However, as we are interested in deep machine learning and large-scale production applications, alternative options like Python and its specialized libraries may be worth considering.

Java

Java is an extensively utilized programming language known for its object-oriented features, which have gained widespread recognition in various software development areas. Initially designed for web applets, Java has extended its influence to encompass

web development as well by utilizing frameworks such as Spring and JavaServer Faces (JSF) to empower contemporary web applications.

Concerning machine learning, Python and R are often hailed as the top programming languages due to their abundance of libraries and community support. However, we should not disregard the potential that Java brings to the table. The Java ecosystem presents useful libraries like Deeplearning4j, Weka, and MOA (Massive Online Analysis) that enable efficient implementation and deployment of machine learning models. These libraries cover an extensive range of tasks, including preprocessing data, extracting features, training models, and deploying them.

Advantages

Java stands out for its performance and runtime efficiency compared to interpreted languages like Python. This attribute becomes highly advantageous in machine learning applications with heavy computational demands.

Another remarkable characteristic is Java's robustness and security, boosting its suitability for sensitive data handling in machine learning or within enterprise settings.

Moreover, thanks to being an object-oriented language, Java aids developers in organizing and structuring code amidst larger-scale and intricate projects.

Additionally, the extensive adoption of Java within enterprise applications and production systems further solidifies it as a viable choice. Hence, if we aim to smoothly incorporate models into existing corporate systems, relying on Java makes perfect sense due to its popularity among enterprises.

Regarding machine learning libraries, Java may not have the same recognition as Python. However, there are a few notable ones, like Deeplearning4j and Weka, which enable model training and data analysis.

Disadvantages

Java may pose a more challenging learning experience for newcomers to programming, compared to languages like Python, with its steeper learning curve.

Regarding machine learning libraries, although Java does offer such tools, the variety and scope they provide tend to be narrower compared to the options available in Python. As a result, staying up to date on the newest techniques and approaches in this field might be difficult for Java users.

Moreover, when it comes to data analysis and manipulation, Java lacks the same level of significance traditionally found in languages such as Python and R. Because of this limitation, certain data preparation activities may be less optimized when using Java.

Another factor worth noting is that although Java boasts an extensive community covering several topics, it does not specifically revolve around machine learning. Therefore, resources focusing on this aspect might not be as abundant within the Java community sphere.

Conclusion

To sum up, Java can constitute a viable alternative for machine learning tasks that demand efficiency, safeguarding, and seamless integration with enterprise platforms. Nevertheless, there may be several issues due to the variety of machine learning frameworks and tools accessible, as well as the ease of analysis and data manipulation elements when compared to languages specially designed for this purpose, such as Python and R.

C++

C++ is a flexible programming language known for its remarkable efficiency and ability to create remarkably efficient applications while exerting fine-grained control over systems. While C++ is not as frequently used in machine learning as Python, it is widely used in demanding computational tasks where optimum performance is critical.

Advantages

C++ gains its reputation for remarkable performance due to its ability to exercise fine control over hardware and optimize resources efficiently. This feature proves exceptionally advantageous in machine learning, where demanding computational tasks, like image and video processing, are prevalent.

Unlike languages like Python, which rely on garbage collection, C++ gives developers more control over memory management. Such autonomy is a crucial advantage since it avoids concerns associated with high memory utilization while maintaining continuous efficiency.

Furthermore, C++ is frequently selected as the primary programming language in projects that have a codebase written in this dialect. This preference is particularly noticeable in high-performance applications or embedded systems where C++ reigns supreme.

Furthermore, various scientific and mathematical computation libraries exclusively available within the realm of C++, such as Armadillo and Eigen, aptly support tasks encompassing matrices manipulation and scientific calculations, integral components for a broad array of machine learning endeavors.

This language enables parallelization and optimization, allowing algorithms to be built effectively for processing enormous amounts of data.

Disadvantages

Learning C++ can be challenging due to its complexity and numerous advanced features. In comparison to simpler, high-level programming-focused languages, mastering C++ requires navigating a steep learning curve.

Another factor that sets C++ apart is its limited range of specialized machine-learning libraries. While there are some options available in this domain, they don't match the vast variety and accessibility found in languages like Python.

Furthermore, development time tends to be extended when working with C++. The language's lower-level approach necessitates delving into intricate memory management details, resulting in slower progress as compared to higher-level programming languages.

Unlike Python and R, both known for their strong focus on data analysis and manipulation capabilities, C++ does not place as much emphasis on these domains. Consequently, certain data preparation tasks may prove more complex within the context of C++.

Conclusion

In essence, C++ proves to be a fitting choice for machine learning uses that demand outstanding efficiency and meticulous management of hardware. Nevertheless, its substantial degree of difficulty in mastering and the absence of dedicated libraries exclusive to machine learning might present obstacles, particularly for those seeking a quicker purposeful on productivity.

Julia

The Julia programming language was developed with a particular focus on numerical computation. Its performance and user-friendly nature have made it increasingly popular in machine learning and data science.

Advantages

Outstanding performance: This characteristic renders the language suitable for scientific and numerical computation applications. The notable level of performance achieved through its JIT (Just-In-Time) implementations enables one to obtain results comparable to those produced by compiled languages.

Its syntax is user-friendly, resembling popular programming languages like Python and MATLAB. This similarity makes it easier for individuals familiar with these languages to transition smoothly into using this one.

The exceptional ease of writing and understanding code serves as a distinct advantage for users. The language has been purposely designed to prioritize clarity and readability, allowing for an accelerated development process.

It offers seamless integration with other programming languages, offering an interface through which one can call code from platforms such as Python, R, or C. This feature proves particularly valuable when utilizing specialized libraries available in these external languages.

One of the advantages of this feature is that it helps with distributed and parallel computations, which can prove advantageous when it comes to training models using vast datasets.

Disadvantages

Despite the increasing size of the Julia community, it falls short of maturity and diversity in libraries compared to well-established languages like Python. Consequently, locating specialized machine-learning libraries and similar tools can be challenging.

In addition to its limitations in library development, the Julia community also lacks a comparable level of online support and resources that more established languages, such as Python, possess. This deficit includes a scarcity of tutorials, documentation, and other readily available information for newcomers.

Conclusion

To sum up, Julia presents a highly viable choice when building machine learning models, especially for those who prioritize performance and are open to experimenting with a relatively new programming language. Nonetheless, it is crucial to consider the accessibility of libraries and the level of development in their ecosystem before deciding whether to utilize Julia for a particular project.

MATLAB

MATLAB is a programming environment specially built for simulation, data visualization, and numerical analysis. This software has garnered widespread usage in industrial and academic settings to conduct detailed analyses and construct models.

Advantages

One of the advantages it offers is the availability of tools for data analysis and visualization, which can be highly beneficial in gaining insights and exploring datasets before model training.

Another key advantage lies in its simplicity when it comes to algorithm prototyping and experimentation. The language is renowned for its ease of use in these aspects, while its syntax resembling mathematical notation makes it particularly intuitive for mathematicians and scientists.

Furthermore, there are numerous libraries and specialized functions specifically designed for signal analysis and statistical analysis. This extensive range proves valuable when dealing with machine learning tasks.

Moreover, its integration with Simulink enhances its utility by providing an environment where dynamic systems can be modeled, simulated, and verified seamlessly. This feature becomes invaluable when tackling projects involving complex systems.

Disadvantages

MATLAB can impose a financial burden, especially for individuals or small enterprises due to its nature as commercial software. Despite the availability of student licensing options, cost remains a restricting factor.

Although MATLAB offers some tools for machine learning, it lacks the extensive assortment of deep learning-specific libraries and frameworks that are found in languages like Python.

In contrast to languages such as Python, which possess vibrant communities engaged in machine learning practices, MATLAB may have a less active user base focused on this field.

Compared to compiled languages or efficiency-driven programming languages like Python with specialized libraries such as NumPy, MATLAB exhibits inferior performance.

Conclusion

Hence, possessing a solid grasp of MATLAB syntax and seeking to leverage its prowess in data analysis and visualization, it can prove to be an advantageous language. Nonetheless, it is crucially necessary to contemplate variables like expense and the accessibility of specialized libraries when contemplating the adoption of MATLAB for machine learning model training.

Desirable features for machine learning languages

In the past, we have extensively covered the widely recognized and commonly employed tools used for these specific purposes. However, it is critical to remember that some of these tools are intended for a broad spectrum of users. Let us examine the essential characteristics that make a programming language suitable for machine learning goals. We shall consider the following features:

- Development simplicity and encouragement of experimentation through clear syntax and specialized libraries: Machine learning languages should be adaptable and user-friendly to allow developers and data scientists to test algorithms, methodologies, and approaches efficiently.
- Optimization capabilities and support for numerical as well as scientific computations: The notion of optimized languages becomes highly desirable when performing calculations related to machine learning tasks due to their involvement with matrix operations and other computationally intensive numerical processes.
- Specialized libraries and frameworks can be accessed to assist in data manipulation and offer pre-existing machine-learning algorithms. These tools prove invaluable as they streamline the process of building and training models.
- By utilizing various tools, we can visualize and analyze data effectively. This helps us identify recurring patterns and gain insights, empowering us to make informed decisions while adequately refining our models.
- Having an active community and readily available learning resources is crucial for facilitating knowledge acquisition. It encourages interaction between experienced professionals and newcomers in the field, leading to an abundance of online resources, thorough documentation, as well as extensive tutorials.
- When choosing a language for software development and data analysis, it becomes advantageous if the language enables seamless integration with

various technologies and tools utilized in these fields. Examples of such technologies include release management systems or cloud platforms.

- Scalability, high performance, and compatibility with existing systems become crucial considerations when selecting a language that prioritizes the creation of models suitable for production environments.
- Considering efficiency and performance aspects holds significant importance, particularly in scenarios involving real-time applications or dealing with substantial datasets.

Analysis of alternatives and final technology decision

After presenting the key features desired in machine learning languages, we will proceed with evaluating the aforementioned technologies based on their alignment with our project's requirements. We will use a comparative table to visualize and facilitate the decision-making process and to highlight each language's compatibility with the most favorable traits.

Languages / Features						
Ease of development and clear syntax	✓				✓	
Optimization and support for numerical calculations		✓				✓
Wide variety of specialized libraries and frameworks	✓	✓				✓
Enabling data visualization and analysis tools	✓	✓				✓
Large community active in machine learning and learning resources	✓	✓				
Enable integration with other technologies	✓		✓	✓	✓	✓
Efficient performance language or thanks to	✓		✓	✓	✓	

libraries						
Language known by the developer	✓		✓	✓		✓
TOTAL	7	4	3	3	3	5

Table 1.- Comparative table of languages and their characteristics.

After careful analysis of the table showcasing various characteristics relevant to the project's nature and the languages utilized for model training, Python emerges as the preferred technology. However, in addition to this basic review, specific criteria were given more weight in our final decision process. We prioritized attributes such as an extensive collection of libraries tailored specifically for audio feature extraction and processing, data analysis, and modeling of emotion detection. Additionally, a dynamic community support system aligned with our decision-making process, along with ample resources accessible for skill development, further solidified our choice of Python. Lastly, familiarity with the programming language was considered to accelerate specialized learning initiatives.

2.4 Development environments

As mentioned above, choosing popular programming languages means that we will have more tools, libraries, documentation, and other resources to help us program more efficiently.

When choosing a development environment, the tool must be:

- Free.
- Extensible.
- Easy to use.
- A tool that has not given us problems in the past.

Thus, we decided to discard popular IDEs such as Netbeans or Eclipse since our previous experiences were not entirely positive, either due to configuration issues, unexpected failures, or problems when running the final software outside the development environment.

Choosing the right development environment is crucial to optimize the workflow. If we could use the same IDE for all parts of the system development, we would highly mitigate the frictions that different tools could generate during the creation of the project. As a result, we picked VSCode since it offers Python extensions, which we needed for the project. It also contains characteristics such as debuggers and version control, which makes it like other popular programming environments. This set of tools and capabilities notably facilitates the development process, allowing a smoother integration of the different components and modules of our system.

We have also extensively used the operating system shell for tasks such as version control or running the system under development. VSCode has its window to run a terminal Shell, so we can perform all these tasks so we can perform all these tasks comfortably from the same editor, bringing together all the activities in the same working environment.

2.5 Communication assistance applications for people with autism

There exist numerous applications designed to aid individuals with autism spectrum disorder, encompassing various functionalities spanning communication assistance, learning support, socialization tools, and routine management. The primary objective of this project is to facilitate communication and comprehension during social interactions by emphasizing the aspect of communication. Nonetheless, we will also touch upon a few alternative applications in unrelated domains.

2.5.1 Communication Applications

Proloquo2Go [20] is an app designed to assist individuals with autism or speech impairments in communication by incorporating visual aids, audio resources, and symbolic tools. The application can be personalized to address the unique requirements of each user. Proloquo2Go belongs to the category of augmentative and alternative communication (AAC) applications.

TouchChat[21] is another AAC app that allows customization of communication options based on individual necessities. It offers a variety of communication boards comprising images, symbols, and text components. Additionally, it includes features like word prediction and synthesized speech capabilities to enhance effective interaction.

AACORN[22] is a tool that aids children with autism in enhancing their communication skills, employing a combination of words and pictures. Additionally, it supplies a collection of social stories to aid in comprehending and predicting various social situations.

MyChoicePad[23] is an application that employs both imagery and linguistic elements to facilitate the formation of sentences and the expression of needs and desires for individuals with autism. It also encompasses educational exercises aimed at improving communication abilities.

SpeakAll! [24] serves as an app geared toward individuals on the autism spectrum, enabling them to construct complete phrases and sentences by selecting images and symbols. Moreover, it offers word prediction functionality to streamline their means of communication.

2.5.2 Socialization and Social Skills Applications

Through Social Stories Creator & Library[25], users have access to videos of social stories so that they can visually understand certain social situations. This application also allows users to create their own stories.

[20] https://www.assistiveware.com/es/productos/proloquo2go
[21] https://touchchatapp.com/
[22] https://at-aust.org/items/11541
[23] https://www.mychoicepad.com/
[24] http://www.speakmod.com/speakall/
[25] https://apps.apple.com/us/app/social-story-creator-library/id588180598

2.5.3 Other applications of assistance to people with ASD outside the focus of the project

Planning and Routine Applications

Choiceworks[26]: a mobile application that assists individuals in adhering to routines and reaching decisions through visual sequences and options.

iPrompts[27]: tailored for aiding people with routine adherence and transitioning, this app presents visual cues and timers.

Learning and Education Applications

Autism iHelp[28]: an array of applications that concentrate on instructing academic, social, and functional abilities via interactive tasks.

ABA Flashcards[29]: this app engages interactive flashcards to impart knowledge and skills using principles rooted in applied behavior analysis.

Relaxation and Stress Management Applications

Calm Counter[30]: supports individuals in managing their anxiety levels and stress by engaging in slow counting techniques alongside deep breathing exercises.

Breath Ball[31]: employs visually represented breathing exercises to induce relaxation among users.

It is worth noting that each person diagnosed with ASD possesses unique needs. Therefore, it is highly advised to seek advice from professionals within the field who can identify those specific requirements. By doing so, an appropriate application can be recommended that tailors its solution according to individual demands.

[26] https://www.beevisual.com/
[27] https://www.neurodevelop.com/iPrompts
[28] https://apps.apple.com/us/app/autism-ihelp-play/id521485216
[29] https://apps.apple.com/us/app/aba-flash-cards-games-emotions/id446105144
[30] https://apps.apple.com/us/app/calm-counter-social-story-anger-management-tool/id470369893
[31] https://play.google.com/store/apps/details?id=com.fundriven.breath_ball

3. Requirements Specification and Design

3.1 Requirements analysis

Vision Document

This section provides a general description of the system's user entities, which we will refer to as stakeholders. It also provides a global vision of the characteristics and functionalities that the system will offer while graphically expressing the relationships between the different concepts of the system using a domain model.

Actors

User with ASD: a person who uses the system to recognize and understand their emotions and those of others through music generated in response to their voice.

Therapist or Educator: specialist who uses the system to help people with ASD identify and manage their emotions. They have full access to the emotional records and music generated for each session, allowing them to adapt and customize the therapy or training according to the progress and needs of each user.

Family members or Caregivers: people close to the user with ASD who can use the system to improve the understanding of their emotions. They may have limited access depending on the privacy settings set by the primary user or therapist.

System: entity representing the developed tool.

User: generalization of the actors, user with ASD, therapist, family members, or caregivers.

Characteristics

User management

- The system shall allow the creation of an initial configuration, including audio parameters and music preferences.
- The system shall allow modification of configuration parameters previously set by the user.
- The system shall save and load user-customized configurations.
- The system shall provide a user-friendly interface for the user to initiate and terminate each session.
- The system shall allow the visualization and review of previous sessions, showing valence and excitation results and the generated musical pieces.

Emotion Analysis

- The system must allow the capture of the user's voice through an integrated or connected microphone.
- The system shall process in real time the captured voice to extract valence and arousal values.
- The system shall visually display the obtained valence and excitation values to the user.
- The system must keep records of the emotion analysis sessions for later review or study.

Music Generation and Management

- The system must generate musical pieces based on valence values, excitation, and notes extracted by MIDI.
- The system must allow the user to adjust musical generation parameters such as tempo or duration.
- The system must allow real-time playback of the generated musical pieces.
- The system must offer the option to save generated music pieces in appropriate formats (e.g., .mp3, .wav, etc.).
- The system must allow the loading of musical pieces in MIDI format for analysis and subsequent generation of sentimentally intoned music.

Auxiliary Model

- The system must allow the loading of musical pieces and extract valence and excitation values from these pieces.
- The system must provide a graphical display of the extracted values for given pieces of music.
- The system must allow comparing valence and excitation values between different sessions and/or musical pieces.

Non-functional requirements

- The system must be available in English due to the availability of data sets containing valence and excitation values in that language.
- The system must be able to process and analyze English speech samples to extract valence and excitation values.
- The system must accept and process MIDI format files for music generation based on the extracted notes.
- The system must be able to load and analyze musical pieces to extract valence and excitation values.
- The system's user interface must be straightforward and intuitive, allowing the user to engage with the program's capabilities.
- A help section should be provided to instruct the user on how to utilize each of the system's functionalities.
- The system must perform speech analysis and music generation in real-time, ensuring fast and efficient response times.

- It must be capable of processing musical pieces of varying lengths and complexities without significantly degrading performance.
- Despite being local, the system must guarantee that the processed data is securely kept and safeguarded from unwanted access.
- To protect data security and integrity, the system should use encryption methods.
- The system must be capable of interfacing seamlessly with usual audio devices such as microphones and loudspeakers.
- The system should be developed in a modular fashion, allowing for future updates and the addition of new features without compromising present operation.
- It should be possible to integrate other models or datasets if information for additional languages or emotions becomes available.

3.2 Functional design

Domain Model

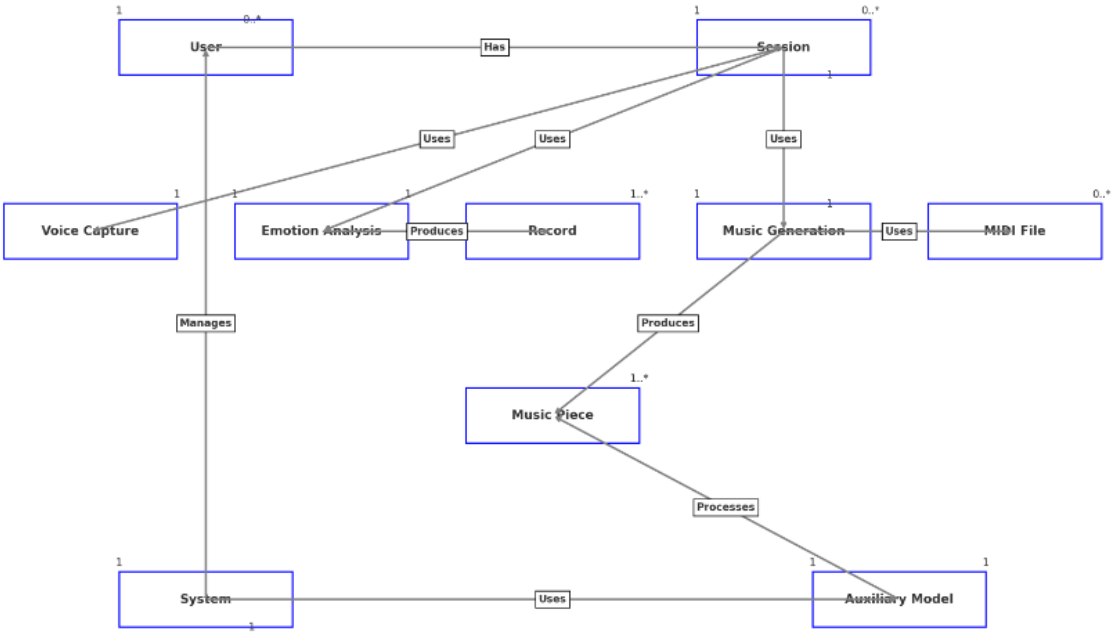

Figure 3.- Domain Model with Multiplicity for ASD Emotion Recognition System.

Use cases

This section intends to represent all the use cases related to the aforementioned characteristics. To do this, first, we will make clear the inheritance between the actors participating in the system:

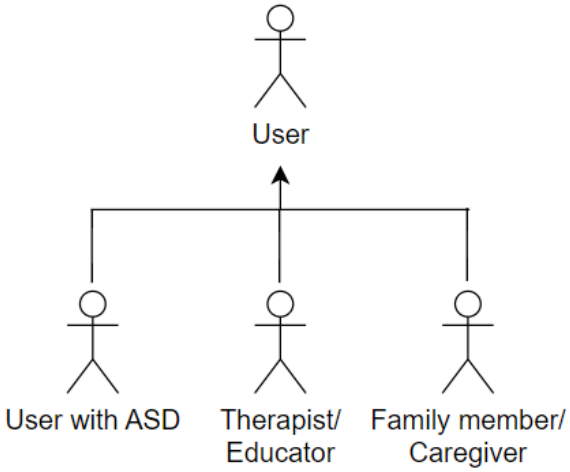

Figure 4.- Actor Diagram for ASD Emotion Recognition System.

As can be seen in the diagram, all those actors that represent a human entity will be considered users of the system to simplify the design of the use cases and facilitate the understanding of those functionalities that can be performed by all these entities. The latter is also an actor since it must perform certain specific functions.

1. Use cases related to "User management"

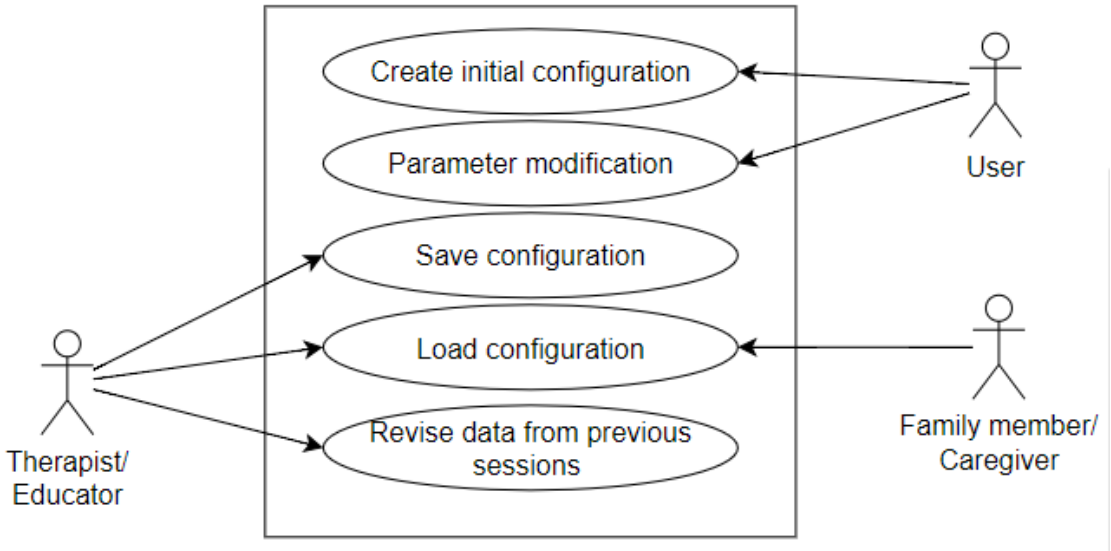

Figure 5.- Use case diagram for "User management"

1.1. The system shall allow the creation of an initial configuration, including audio parameters and music preferences.

Reference:	UC-1.1
Name:	Create Initial Configuration
Description:	Allows the user to set and customize parameters for voice processing and music generation for the first time.
Actor:	User

Relations:	UC-1.3
Preconditions:	User installs and executes the software.
Postconditions:	Initial configuration has been created and saved.
Flow:	1) User selects “Initial configuration” option. 2) The system displays an interface with different parameters to configure. 3) The user customizes the parameters according to his/her preferences. 4) The user confirms and saves the configuration.

Table 2.- Use case “Create Initial Configuration”

1.2. The system shall allow modification of configuration parameters previously set by the user.

Reference:	UC-1.2
Name:	Modify Configuration Parameters
Description:	The user changes the parameters previously set in the configuration.
Actor:	User
Relations:	UC-1.3
Preconditions:	The user has established an initial configuration.
Postconditions:	The configuration parameters have been modified.
Flow:	1) The user accesses the configuration section. 2) Selects and modifies the desired parameters. 3) The user confirms and saves the changes.

Table 3.- Use case “Modify Configuration Parameters”

1.3. The system shall save user-customized configurations.

Reference:	UC-1.3
Name:	Saving the Configuration
Description:	The user saves the current system configuration for later use.
Actor:	Therapist/Educator
Relations:	UC-1.1, UC-1.2
Preconditions:	The user has created or modified a configuration.
Postconditions:	The configuration has been successfully saved.
Flow:	1) After making changes or setting the initial configuration, the user selects the save configuration option. 2) The system saves and confirms the success of the operation.

Table 4.- Use case “Saving the Configuration”

1.4. The system shall load user-customized configurations.

Reference:	UC-1.4
Name:	Load Configuration
Description:	The user loads a previously saved configuration.
Actor:	Therapist/Educator, Family member/Caregiver
Relations:	UC-1.3
Preconditions:	Previously saved configurations exist.
Postconditions:	The selected configuration has been successfully loaded.
Flow:	1) The user accesses the configurations section. 2) Selects the load configuration option. 3) Chooses one of the saved configurations. 4) The system loads and applies the selected configuration.

Table 5.- Use case “Load Configuration”

1.5. The system shall allow the visualization and review of previous sessions, showing valence and excitation results and the generated musical pieces.

Reference:	UC-1.5
Name:	Review Previous Sessions Data
Description:	The user reviews the data and results of previous sessions in which he/she used the system.
Actor:	Therapist/Educator
Relations:	UC-2.4
Preconditions:	The user has used the system in previous sessions and there is saved data.
Postconditions:	The user has reviewed the information without making changes to it.
Flow:	1) The user accesses the history or previous sessions section. 2) The system displays a list of previous sessions with data and results. 3) The user selects and views the data for a specific session.

Table 6.- Use case "Review Previous Sessions Data"

2. Use cases related to "Emotion Analysis"

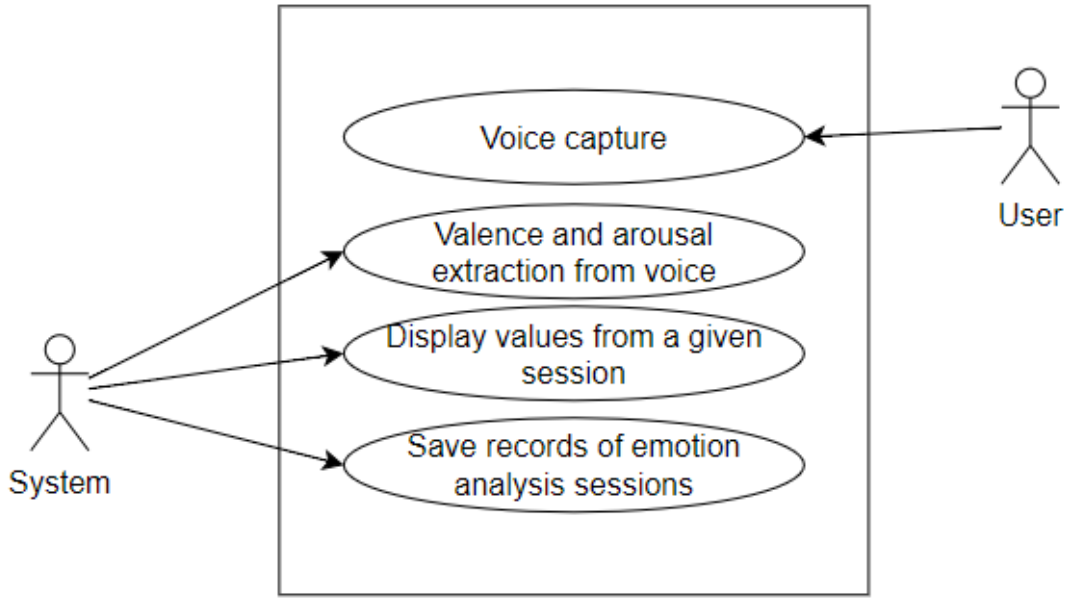

Figure 6.- Use case diagram for "Emotion Analysis"

2.1. The system must allow the capture of the user's voice through an integrated or connected microphone.

Reference:	UC-2.1
Name:	User Voice Capture through a Microphone
Description:	Allows the user to use a microphone to capture and send his voice to the system for processing.
Actor:	User
Relations:	UC-2.2
Preconditions:	The system is powered on and in listening mode, and the microphone is connected and working properly.
Postconditions:	The user's voice has been captured and sent for processing.
Flow:	1) The user activates the voice capture option. 2) They speak or make vocal sounds into the microphone.

	3) The system captures the voice and sends it for processing.

Table 7.- Use case "User Voice Capture through a Microphone"

2.2 The system shall process in real time the captured voice to extract valence and arousal values.

Reference:	UC-2.2
Name:	Process in real time the captured voice to extract values of valence and excitation
Description:	Once the user's voice is captured, the system processes the information to determine the valence and excitation values.
Actor:	System
Relations:	UC-2.3
Preconditions:	The user's voice has been captured.
Postconditions:	The valence and excitation values have been determined.
Flow:	1) The system receives the user's voice data. 2) Real-time processing of the voice is initiated. 3) Valence and excitation values are determined. 4) The system displays the obtained values to the user.

Table 8.- Use case "Process in real time the captured voice to extract values of valence and excitation"

2.3 The system shall visually display the obtained valence and excitation values to the user.

Reference:	UC-2.3
Name:	Visually display to the user the values of Valencia and Excitation obtained in a given session
Description:	The system displays graphically or numerically the valence and excitation values obtained from the user's voice.
Actor:	System
Relations:	UC-2.1, UC-2.2
Preconditions:	The valence and excitation values have been determined.
Postconditions:	The user has displayed the valence and excitation values.
Flow:	1) The system processes and obtains the valence and excitation values. 2) A visual representation (e.g., a graph or numerical indicators) is generated. 3) The user visualizes the results on the system interface.

Table 9.- Use case "Visually display to the user the values of Valencia and Excitation obtained in a given session"

2.4 The system must keep records of the emotion analysis sessions for later review or study.

Reference:	UC-2.4
Name:	Saving Records of Emotion Analysis Sessions for Later Review or Study
Description:	After obtaining and displaying the valence and arousal values, the system automatically saves the session record for future review.
Actor:	System
Relations:	UC-1.5
Preconditions:	The analysis session has concluded, and valence and excitation values have been obtained.
Postconditions:	The session record has been saved correctly.
Flow:	1) The system determines the end of the analysis session. 2) A log is created with the valence data, excitation, date, time and any other relevant information.

	3) The system saves the log to a local database or file. 4) It notifies the user that the log has been successfully saved.

Table 10.- Use case "Saving Records of Emotion Analysis Sessions for Later Review or Study"

3. *Use cases related to "Music Generation and Management"*

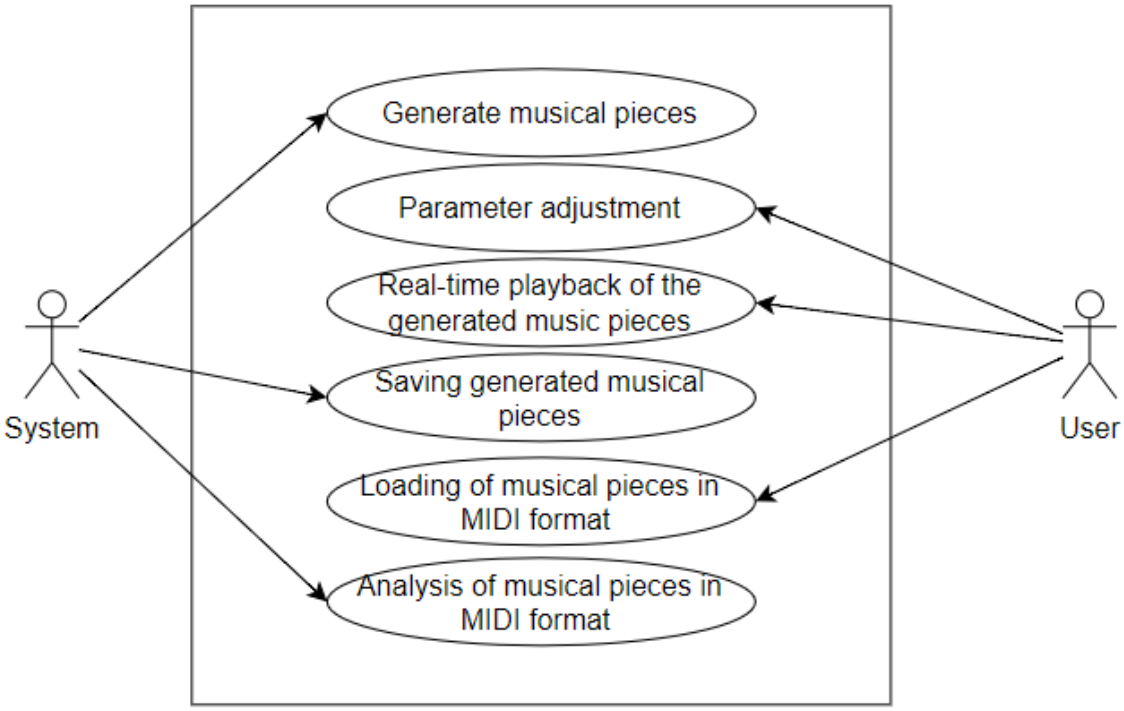

Figure 7.- Use case diagram for "Music Generation and Management"

3.1 The system must generate musical pieces based on valence values, excitation, and notes extracted by MIDI.

Reference:	UC-3.1
Name:	Generate Musical Pieces Based on Valencia Values, Excitation and MIDI Extracted Notes
Description:	The system produces a piece of music reflecting the valence and excitation values, supplemented with notes obtained from a MIDI file.
Actor:	System
Relations:	UC-3.3, UC-3.5
Preconditions:	The user has provided valence and excitation values, and a MIDI file has been loaded.
Postconditions:	A piece of music reflecting the indicated emotion has been generated.
Flow:	1) The system receives and processes the values of valence and arousal. 2) The piece of music is produced. 3) The system notifies the user that the piece of music is ready.

Table 11.- Use case "Generate Musical Pieces Based on Valencia Values, Excitation and MIDI Extracted Notes"

3.2 The system must allow the user to adjust musical generation parameters such as tempo or duration.

Reference:	UC-3.2
Name:	Adjusting Music Generation Parameters such as Tempo, Instruments or Duration

Description:	After obtaining and displaying the valence and arousal values, the system automatically saves the session record for future review.
Actor:	User
Relations:	UC-3.1
Preconditions:	The user is in the music generation interface.
Postconditions:	Music generation parameters have been set or modified.
Flow:	1) The user selects the parameter to be modified (tempo, instrument, duration). 2) They make the necessary adjustments. 3) They confirm the changes. 4) The system applies the set parameters to the next music generation.

Table 12.- Use case "Adjusting Music Generation Parameters such as Tempo, Instruments or Duration"

3.3 The system must allow real-time playback of the generated musical pieces.

Reference:	UC-3.3
Name:	Real Time Playback of the Generated Music Pieces
Description:	The user can listen to the generated musical piece immediately after its creation.
Actor:	User
Relations:	UC-3.2
Preconditions:	A piece of music has been generated.
Postconditions:	The piece of music has been played.
Flow:	1) The user selects "Play". 2) The system plays the piece of music. 3) The user can pause or stop the playback at any time.

Table 13.- Use case "Real Time Playback of the Generated Music Pieces"

3.4 The system must offer the option to save generated music pieces in appropriate formats (e.g., .mp3, .wav, etc.).

Reference:	UC-3.4
Name:	Saving Generated Music Pieces in Appropriate Formats (e.g., .mp3, .wav, etc.)
Description:	The system can save the generated music piece in different formats.
Actor:	System
Relations:	UC-3.2
Preconditions:	A piece of music has been generated.
Postconditions:	The piece of music has been saved in the selected format.
Flow:	1) Chooses the desired format. 2) Generates name of the file. 3) The system saves the piece in the selected format and location.

Table 14.- Use case "Saving Generated Music Pieces in Appropriate Formats"

3.5 The system must allow the loading of musical pieces in MIDI format for analysis and subsequent generation of sentimentally intoned music.

Reference:	UC-3.5
Name:	Loading Musical Pieces in MIDI Format
Description:	The user loads a piece of music in MIDI format to the system for analysis or use in music generation.
Actor:	User
Relations:	UC-3.1
Preconditions:	The user has a MIDI file available.
Postconditions:	The MIDI file has been successfully uploaded to the system.

Flow:	1) The user browses and selects the desired MIDI files. 2) Confirms the upload. 3) The system loads and processes the MIDI file.

Table 15.- Use case "Loading Musical Pieces in MIDI Format"

Reference:	UC-3.6
Name:	Analysis of Musical Pieces in MIDI Format
Description:	The system analyzes a piece of music in MIDI format to obtain information about notes and structure.
Actor:	System
Relations:	UC-3.1, UC-3.5
Preconditions:	A MIDI file has been loaded.
Postconditions:	The MIDI file has been successfully uploaded to the system.
Flow:	1) After the MIDI file has been loaded, the system starts the analysis. 2) Notes and file structure are extracted. 3) The details of the analysis are stored and/or displayed to the user. 4) The system notifies that the analysis is complete.

Table 16.- Use case "Analysis of Musical Pieces in MIDI Format"

4. *Use cases related to "Auxiliary Model"*

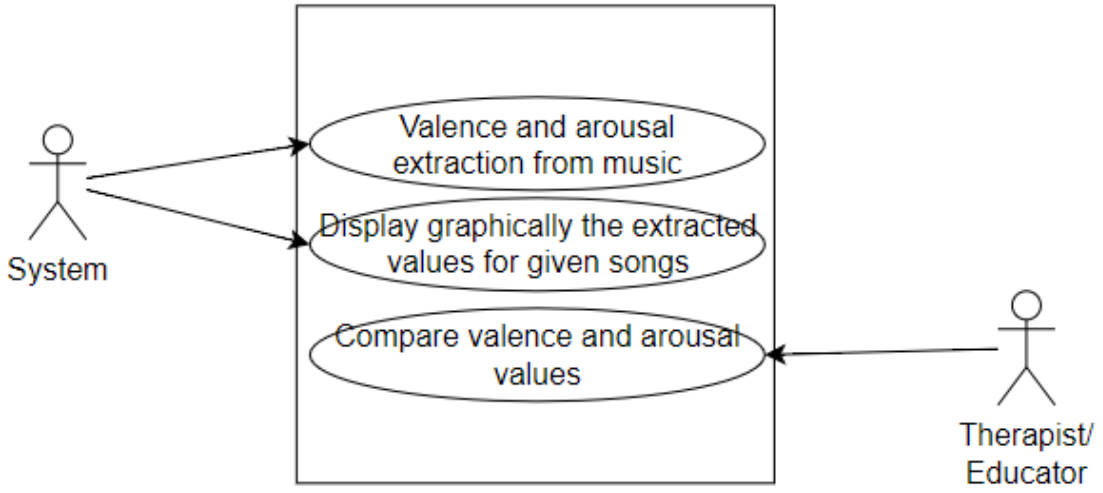

Figure 8.- Use case diagram for "Auxiliary Model"

4.1. The system must allow the loading of musical pieces and extract valence and excitation values from these pieces.

Reference:	UC-4.1
Name:	Extract Valence and Arousal Values from Musical Pieces
Description:	The system processes a piece of music to extract the valence and excitation values associated with it.
Actor:	System
Relations:	UC-3.5, UC-4.2
Preconditions:	The user has loaded a piece of music into the system.
Postconditions:	The corresponding valence and excitation values have been extracted and stored.
Flow:	1) After the MIDI file has been loaded, the system starts the analysis.

	2) The system processes the piece of music. 3) It extracts valence and arousal values. 4) The system notifies the user that the extraction is finished and displays the obtained values.

Table 17.- Use case "Extract Valence and Arousal Values from Musical Pieces"

4.2 The system must provide a graphical display of the extracted values for given pieces of music.

Reference:	UC-4.2
Name:	Graphical Display of Extracted Values for Given Musical Pieces
Description:	The system generates a graphical representation the valence and excitation values for a given piece of music.
Actor:	System
Relations:	UC-3.5, UC-4.1, UC-4.3
Preconditions:	The valence and excitation values of the musical piece have been previously extracted or these values are available.
Postconditions:	A graphical representation of the extracted values is displayed.
Flow:	1) The system generates the graphical representation from the extracted values of valence and arousal.

Table 18.- Use case "Graphical Display of Extracted Values for Given Musical Pieces"

4.3 The system must allow comparing valence and excitation values between different sessions and/or musical pieces.

Reference:	UC-4.3
Name:	Compare Valencia and Excitement Values between different sessions and/or musical pieces
Description:	The user can compare valence and excitation values between different musical pieces or analysis sessions.
Actor:	Therapist/Educator
Relations:	UC-4.2
Preconditions:	More than one set of valence and excitation values are available, either from different musical pieces or from different sessions.
Postconditions:	A graphical or tabular comparison of the values is presented.
Flow:	1) The user selects the musical pieces or sessions to be compared. 2) They instruct the system to perform the comparison. 3) The system generates a graphical or tabular representation showing the comparison. 4) The user visualizes and analyzes the compared values.

Table 19.- Use case "Compare Valencia and Excitement Values between different sessions and/or musical pieces"

4. Implementation of the System

In this section, we will provide an elaborate account of the hardware and software options that have been selected for the design and implementation of the system. Additionally, we will delve into the intricacies of various neural network models employed in shaping the application's structure. Thorough descriptions regarding learning phases and their assessment shall also be included.

4.1 Detailed aspects of the system

For the development of this master's thesis project, we focused on the recognition of emotions through voice and the subsequent generation of music based on such recognition. As we have previously detailed, the main goal of this system is to facilitate emotional recognition in people with autism. In the "State of the Art" chapter, we examined several tools and libraries suitable for constructing this type of application.

Among the libraries that we consider relevant for audio processing and emotion recognition through machine learning techniques, we selected the following: blosc2 for data compression, cython for optimizing some critical parts of the code, pretty_midi for working with MIDI files, librosa for audio analysis, numpy for matrix manipulation and numerical calculations, Tensorflow as our principal tool for developing machine learning models, os for file and directory management, and csv for handling CSV files in which we store certain datasets.

After analyzing various tools for development, we explored multiple devices and platforms suitable for speech-emotion recognition. Nonetheless, to meet the requirement of local operation and prioritize privacy as well as data integrity, we decided to create a desktop application.

For the final implementation, these libraries were utilized with an emphasis on ensuring the reliability, efficiency, and most importantly effectiveness of the system in assisting individuals with autism in recognizing emotions.

Hardware

To effectively recognize emotions through speech and create music based on those recognitions, our project aims to assist individuals with autism in identifying their emotions. We must have specialized hardware to tackle the intricate computational requirements of these tasks.

After thoroughly evaluating various options, we have opted to utilize the subsequent equipment:

Computer

The central component of this entire procedure will be the brain, which must possess substantial power to efficaciously analyze audio data instantly and create music that reflects the identified emotions.

- Processor:
  - Model: Intel® Core™ i5-10300H
  - Number of cores: four
  - Turbo frequency (maximum): 4.5 GHz
- Memory:
  - Type: DDR4-SDRAM
  - Slots: 2x SO-DIMM
  - Maximum Memory: 16 GB
  - RAM: 8 GB
- Graphics
  - Video card: NVIDIA® GeForce® GTX 1650
  - On-board graphics: Intel® UHD Graphics
  - Discrete graphics memory type: GDDR6

Microphone: Audio-Technica AT2020USB+

The microphone chosen is renowned for its exceptional recording capabilities. Its precision and clarity allow for intricate vocal capture, particularly crucial in precise emotion identification. USB connectivity allows seamless integration with the computer system whilst assuring uncompromised transmission quality of audio data.

Figure 9.- USB microphone AT2020USB+.

The presence of these elements guarantees the effective functioning of the system. Its ability to capture and evaluate the user's vocal expressions fast leads to real-time music generation that aligns with detected emotions while upholding a seamless and precise user experience.

Software

To advance the progress of our system, we have incorporated a variety of third-party software and libraries, which we will now elucidate below.

Librosa

Librosa, a Python library dedicated to audio and music analysis, stands out as a versatile tool suitable for various applications involving audio signals. Whether it be feature extraction or signal transformation, Librosa shines in facilitating these tasks.

A standout feature of Librosa lies in its capacity to convert audio into Melspectrograms. Through Melspectrograms, the information within the audio is visually represented in an alternative manner. This depiction assigns greater importance to frequencies humans perceive with clarity and sensitivity. The utilization of this representation proves particularly valuable when analyzing emotions tied to auditory perception by accentuating the relevant characteristics of the audio input.

Tensorflow

TensorFlow, an innovative platform created by Google, has become indispensable in machine learning and neural networks due to its adaptability and durability. Serving as an open-source library, it facilitates the effective handling of vast datasets and intricate models for researchers and developers alike. Concerning our project scope, TensorFlow assumes vital significance across multiple stages of the process.

To address audio-related tasks about emotion analysis specifically, we have leveraged TensorFlow's capabilities extensively. Our efforts involved devising and training three distinct neural network models targeting various aspects within this domain:

- The model for extracting music valence and arousal aims to interpret and analyze the audio signals in musical compositions. Through automated training, our neural network is trained to recognize and extract the degrees of positivity/negativity (valence) and intensity (arousal) found in the music. These measurements play a crucial role in grasping the emotions or moods a musical piece can evoke from listeners.
- The human voice valence and arousal extraction model is like the previous model but revolves around human vocal audio. This network is trained to identify and scrutinize the emotions and moods present in the human voice. Consequently, we become more adept at comprehending individuals' feelings by analyzing their vocal pitch and modulation, an invaluable asset for deciphering emotional recognition within a given context.
- The model for generating music based on valence and arousal: After determining the valence and arousal levels, this model creates music that corresponds to these emotions. By employing generative neural networks, it can generate musical compositions that effectively capture the identified emotional state. As a result, this creates a listening encounter that strongly resonates with the listener's emotions.

Keras

Keras, a Python-based high-level API, serves as an essential component in our system by enabling the effortless and efficient creation and training of machine learning

models. It runs on top of TensorFlow and aids us in various stages through its ability to easily define models by combining different layers:

- Utilizing Keras, the loading and saving of models can be carried out effortlessly, enabling us not only to reload previously trained models but also to save any modified or trained model we have, ensuring the persistence of our models and enabling future reuse without the need for retraining from the ground up. Loading a model is particularly advantageous when starting our application promptly or employing effective models for comparable tasks. It also eases model lifecycle management, fostering agile iterations, tests, and deployments with greater efficiency.
- Optimizing a model: after establishing a model, Keras equips us with tools to optimize it. With these tools, we can choose the most suitable algorithms for optimization, specify loss functions, and configure metrics to assess the performance of our model.
- Definition of Layers: within Keras, there exists a diverse selection of preexisting layers that can be modified to suit individual requirements. These layers span the gamut from dense (which is fully connected) to convolutional (suitable for signal or image processing). To achieve emotion recognition in our application, we have implemented a spectrum of these versatile layers and adjusted their parameters accordingly, ensuring they effectively accommodate the audio data type we are working with.

Incorporating Keras into our workflow has brought about two significant benefits: streamlining the process of designing and optimizing models, as well as expediting system development, thus enabling us to iterate and experiment swiftly. The compatibility it shares with TensorFlow and its user-friendly approach spare us from getting lost in the complexity of neural network design while enabling us to concentrate on the core logic behind our emotion recognition system.

Pretty_midi

Pretty_midi, a Python library focused on handling MIDI format, stands out due to its unique capacity for decoding and encoding intricate musical compositions. Individual notes, instrument changes, and dynamics are all examples of this.

When our system identifies a specific emotion through voice analysis, we rely on Pretty_midi to generate a musical piece. It is worth noting that this composition is not random in nature. Pretty_midi allows us to arrange and organize the music in a way that portrays the identified emotional state. For instance, if an underlying melancholic tone is detected, it could manifest as a slow-paced melody consisting of minor keys; conversely, if there is an experience of joyfulness identified, the resulting piece might exhibit more exuberance by utilizing major keys.

Music21

Music21 serves as an asset in our digital music toolkit. While Pretty_midi is outstanding for generating and manipulating MIDI compositions, music21 elevates its capabilities by offering multiple tools for examining music.

With music21, apart from producing melodies, we can also disassemble and assess each piece we create. This proves particularly advantageous in understanding and ensuring that the tunes generated by our system align with detected emotions. For instance, we can evaluate the key signatures, rhythms, and harmonic structures of a composition to guarantee they harmonize appropriately with the intended emotional response.

By combining Pretty_midi and music21, we adopt a comprehensive yet meticulous approach to crafting and dissecting musical elements within our system. Both libraries collaborate seamlessly to ensure that every musical piece delivered resonates emotionally and possesses coherence and enrichment musically.

Other Useful Libraries

Numpy stands as an essential Python library when it comes to scientific computing. Its primary purpose is to facilitate the manipulation of multidimensional arrays and offer a broad range of mathematical functions that can be applied to these arrays.

Blosc2 serves as a compression library created for reading and processing data rapidly. It particularly shines in efficiently managing substantial amounts of data.

Cython serves as a comprehensive toolset that simplifies the process of creating C extensions for Python. It is crucial for enhancing execution velocity, particularly when confronted with computationally complex tasks.

The '**os**' module enables developers to tap into operating system-specific features, like file system operations such as reading and writing. It forms the backbone of file and directory management tasks.

For handling tabular data storage, the '**csv**' package comes to the forefront by providing convenient functions for reading from and writing to CSV files, which are widely used across various domains.

These supplementary libraries are crucial to ensure the system's effectiveness, scalability, and extensive capability. Together, they provide means for proficient data handling, mathematical computations, interfacing with the OS, and optimizing performance.

These libraries have been selected for their high reliability and effectiveness in our system. When combined, they form a smooth process that spans from the initial processing of voice signals to the generation of music, all while maintaining local operations and safeguarding user data privacy and security.

Datasets

The underlying essential element of any system based on machine learning lies in the data it operates with. In this segment, we will direct our attention toward the particular sets of data that were chosen and implemented to train and improve the models of our system. We aim to emphasize their importance and composition and explain the reasons behind their selection.

IEMOCAP

IEMOCAP, a multimedia database known as Interactive Emotional Dyadic Motion Capture Database, aims to investigate emotional interactions during conversational exchanges involving two individuals. Its primary purpose lies in the research and development of emotion recognition and analysis systems.

IEMOCAP contains recorded sessions where actors converse while assuming various roles and scenarios. They aim to mimic real-life dialogues. During these interactive sessions, participants embody diverse emotional states, rendering a comprehensive range of emotions accessible for study. The recordings encompass expressions, facial movements, speech intonation, rhythm (prosody), and other acoustic signs that contribute to identifying distinct emotions.

The dataset holds much information that proves incredibly valuable when researching in fields like deciphering emotions based on voice or body language. It has been used in several studies focused on improving our understanding of emotional communication across fields such as linguistics, psychology, and computer science.

Within the scope of our system, we have employed the IEMOCAP dataset to train a sophisticated model capable of extracting valence and arousal values from human speech. By adapting this dataset for our purposes, we now possess the ability to interpret and quantify emotions expressed through verbal communication.

Emotional Speech Database

The Emotional Speech Database (ESD) is a resourceful collection of speech data researchers use to study speech conversion. ESD contains 350 parallel utterances spoken by ten native English speakers and ten native Chinese speakers, encompassing five emotional categories: neutral, happy, angry, sad, and surprised. The database includes over 29 hours of meticulously recorded speech in a controlled acoustic setting. This extensive database proves invaluable for studies exploring emotional voice conversion across multiple speakers and languages.

The field of emotional speech recognition is gaining traction as it aims to recognize and categorize the emotions expressed through an individual's voice. This capability has enormous potential in numerous areas, including assisting people with ASD in improving their communication skills and benefiting technologies such as speech synthesis, animation, and games.

To enhance our system, we have utilized the ESD dataset to address a deficiency of certain emotions in the IEMOCAP set. By incorporating it, we can equilibrate our training samples and enhance the precision and resilience of our system when identifying and generating distinct emotions.

DEAM

DEAM, short for Database for Emotional Analysis of Music, is a comprehensive repository dedicated to analyzing emotions conveyed through music. The DEAM dataset encompasses 1802 musical pieces annotated with valence and arousal values.

These annotations are provided in two forms, for the entire duration of each song and on a second-by-second basis. Additionally, metadata records additional information concerning these audio samples, such as their duration or genre.

The study of emotional analysis in music investigates how music influences our emotional state, and it aims to leverage this knowledge to convey emotions better and elevate our overall musical experiences. The wide array of emotionally rich data provided inside DEAM can be used for future study initiatives and is invaluable to academics in this discipline.

Within our system framework, we employed the DEAM dataset to train a model that extracts valence and arousal levels from diverse music genres. This model will be used to complete the information given by the VGMIDI and generate emotionally tuned music.

VGMIDI

VGMIDI is an extensive collection of piano arrangements for video game soundtracks. It comprises 200 MIDI pieces categorized based on emotion and an additional 3,850 songs without any specific labels. The emotional categorization was done by gathering input from 30 individuals who used a custom web tool to apply the Circumplex model involving valence-arousal.

The VGMIDI dataset offers numerous emotionally charged musical data that can be highly valuable in analyzing emotions conveyed through music. Its massive size, along with the regulated acoustic environment, makes it an incredibly useful resource.

For our project's framework, we have opted to make VGMIDI the centerpiece for generating music. This decision stems from the conveniences provided by the MIDI format, which facilitates analysis and feature extraction processes. With this format streamlining comprehension and manipulation of musical information, we can focus on interpreting and eliciting emotions through compositions.

4.2 Design of the system

The modular structure is a crucial aspect of the architecture of any robust application or system. Incorporating a design that emphasizes individual modules not only does it lead to a more understandable and more logical code structure but also enhances scalability, maintainability, and efficiency. This section will examine our system's underlying design by diving into its many components. Each module was created with a specific goal in mind, and they all work together to fulfill the system's overall objective. We will investigate and clarify the functions and tasks performed by each module, providing a thorough grasp of how they interconnect and interact to produce the ultimate result.

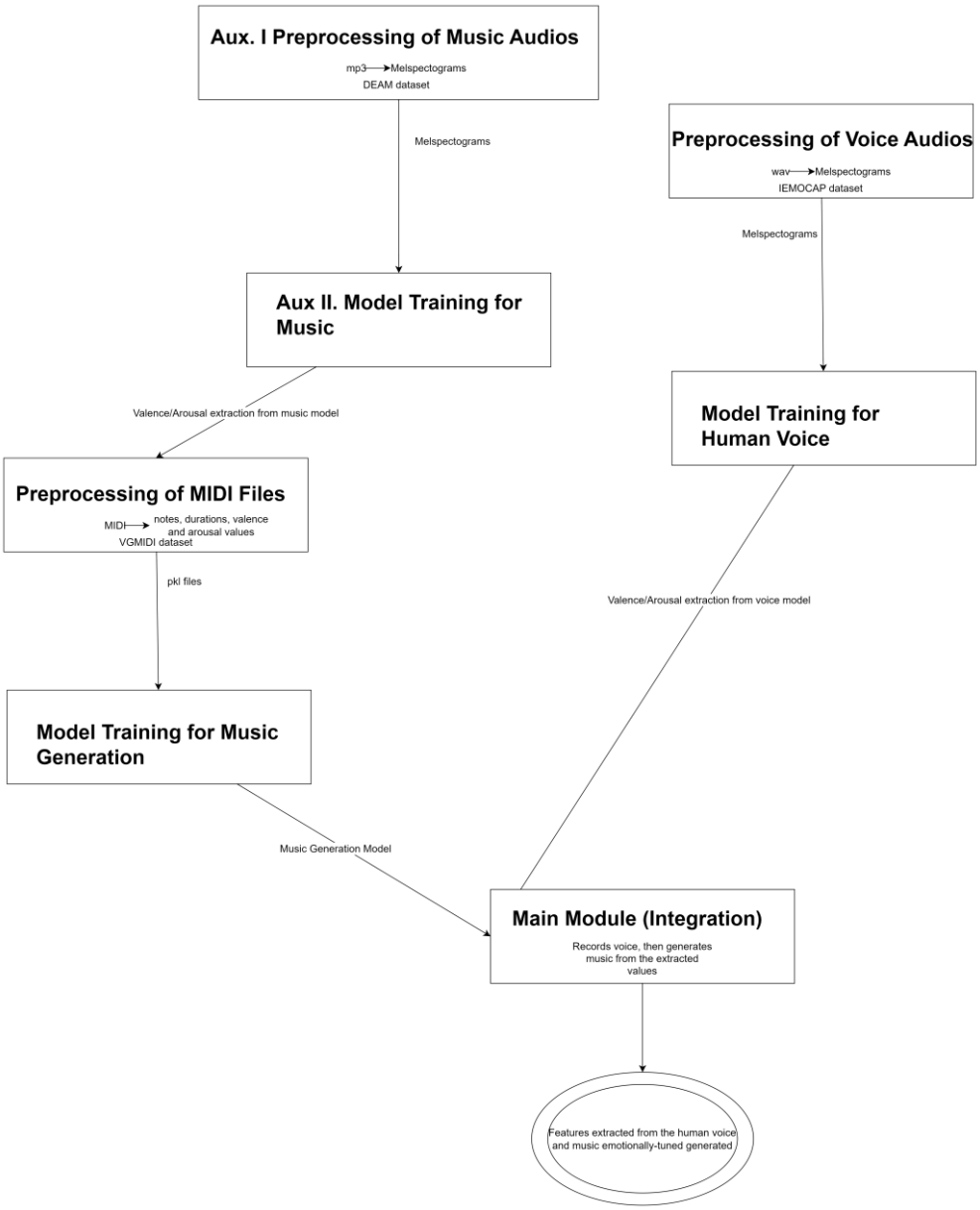

Figure 10.- System architecture diagram.

Module I: Preprocessing of audio files containing human voice from IEMOCAP dataset

The objective of this module is to prepare and handle an audio dataset, specifically the IEMOCAP dataset, for emotional analysis. The code's structure and functionality are broken down into several crucial phases:

- Eliminating Undesirable or Underrepresented Emotions: before starting any processing, an initial stage is performed, in which those emotions deemed unimportant or do not have enough representation in the dataset are filtered and discarded. For instance, certain emotions might have been labeled inconsistently or possess a significantly low number of associated audio samples, leading to potential distortions or imbalances in subsequent analysis.

This stage guarantees that only audios of relevant emotional categories and sufficient representation undergo further processing.

- Extracting Emotional Annotations: emotional annotations relevant to each audio segment or 'turn' are extracted from designated text files accompanying the dataset. These annotations contain information regarding the turn's name, associated emotion, valence, and arousal.
- Preprocessing Audio: each .wav audio file is loaded and processed. In this stage, the audio undergoes resampling at a frequency of 16,000 Hz and is then standardized to a consistent length of 3 seconds. Files that exceed this length will be truncated, while those shorter will have zeros inserted for padding.
- Extracting Features: from each preprocessed audio segment, we obtain a Melspectrogram, a visual representation displaying how the frequency spectrum of sound evolves. These spectrograms capture significant characteristics that are useful in tasks like emotional analysis.
- Storing Results: once extraction is complete, every Melspectrogram is saved as a .npy file for future use in model analysis and training.

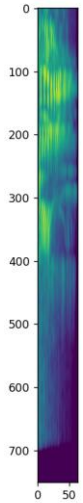

Figure 11.- Melspectogram derived from one of the IEMOCAP audios.

Now let us delve into why we chose to work with exclusively 3-second fragments when extracting information from each audio file. There are various reasons behind this choice:

- Computational and Storage Limitations: choosing shorter audio segments helps prevent the system from overloading during training, especially if employing deep neural networks. Moreover, managing and storing vast amounts of data poses potential problems and needlessly consumes storage space.
- Consistency in the dataset: maintaining a uniform length for all audio segments eases processing and minimizes issues that may arise during training. For instance, when feeding data into a neural network, all inputs must have identical dimensions.
- Capturing Emotional Essence: although three seconds may appear brief, it often proves sufficient to capture the emotional essence in many instances.

Emotions within speech typically reveal themselves through condensed fragments, thus offering a representative window into the overall emotion conveyed by the segment.

Additionally, the valence and arousal values of all the audio samples were represented and colored with the assigned emotions (see Illustration 12). By averaging the values, we extracted the centers of the emotional zones assigned to each feeling, as shown in Illustration 13.

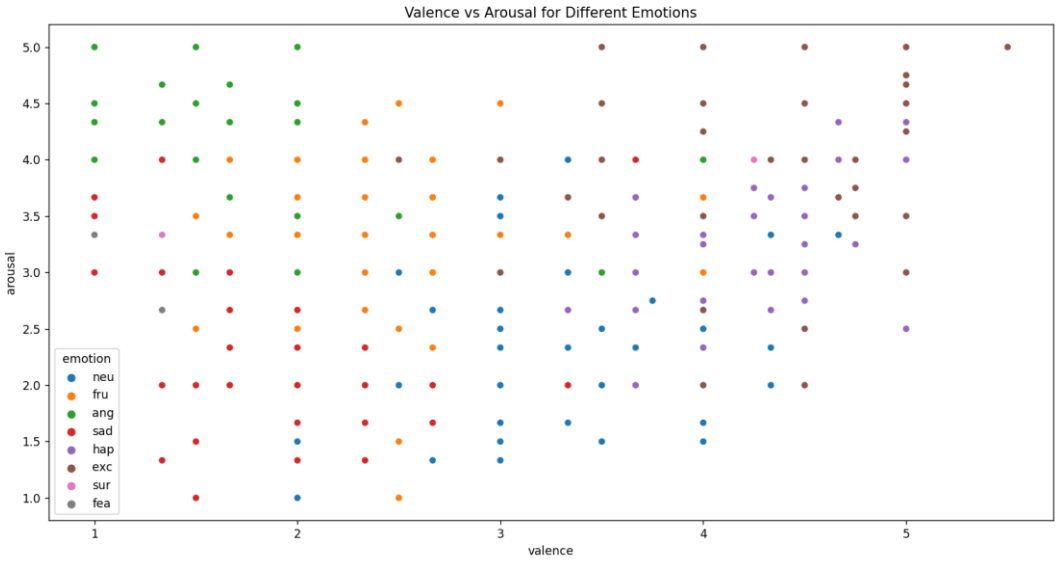

Figure 12.- Valence vs Arousal Graph for each IEMOCAP audio.

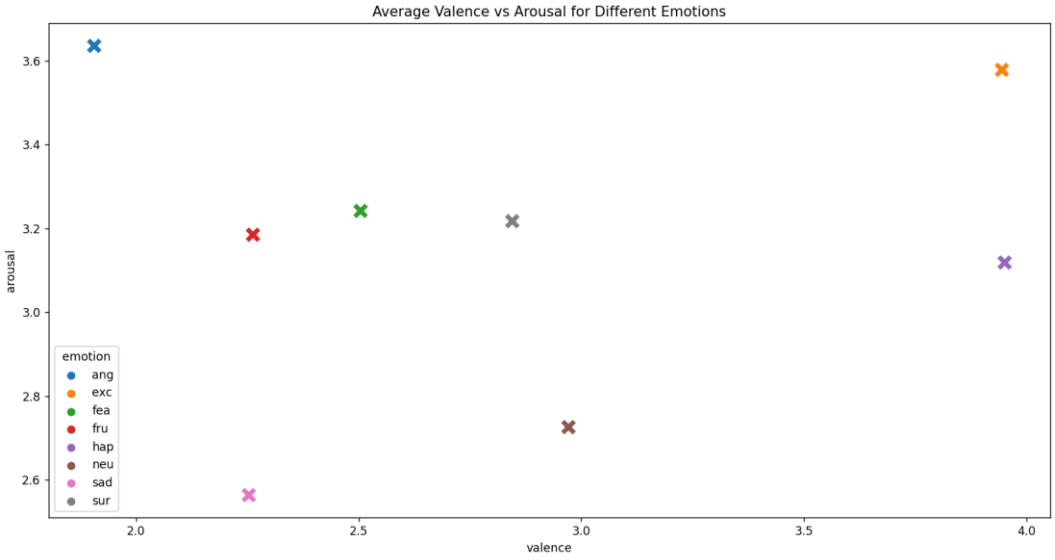

Figure 13.- Average Valence vs Arousal Graph for each IEMOCAP emotion.

Module II: Processing and Training of an Emotional Detection Model based on IEMOCAP.

In this module, we employ a specific approach to process, train, and confirm the accuracy of a regression model that anticipates two emotional measurements: valence and arousal.

To begin with, we establish both the lowest and highest values for valence and arousal. These limits will be employed to normalize the value ranges for both metrics.

Next, we define a function that retrieves valence and arousal labels from designated file paths. These files contain annotations of audio segments or "turns" along with their corresponding emotions and information on valence and arousal. We employ regular expressions to extract pertinent details accurately and consistently from each text file.

Upon availability of the labels, the spectrograms of each turn are loaded. These spectrograms have been previously generated and stored in .npy format by the first module of the system. They serve as a visual representation showcasing the frequency content transformation of audio samples.

The core of this module lies in establishing and training a regression model. Within the `create_model` function, we define an architecture reliant on convolutional neural networks using TensorFlow and Keras. It encompasses several layers, including convolutional layers, normalization processes, pooling techniques, dropout mechanisms, and densely connected layers towards its culmination. It is worth mentioning that we implemented regularization methods to combat overfitting issues and enhance overall model generalization capabilities. Figure 14 shows the chosen architecture of the model.

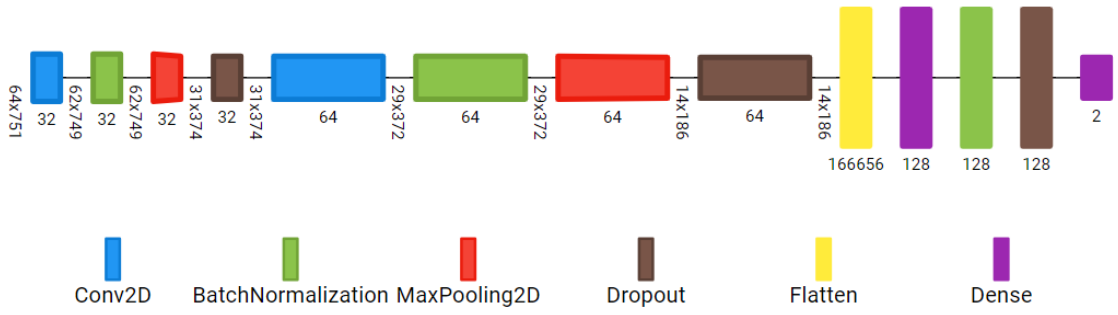

Figure 14.- Valence and arousal model architecture diagram.

K-Fold cross-validation holds significant importance within this module. By employing this technique, the model can be evaluated using different training and testing data combinations. This enhances the reliability of performance evaluation and also strengthens its robustness.

After performing cross-validation, the model undergoes training using a distinct training and validation data set. Throughout this process, we track the progression of loss (error) in both the training and validation sets. Ultimately, these observations are visualized through a graph that helps us assess how the loss alters as the training advances.

Upon completion of training, we save the model in a .h5 file for future use or application.

Auxiliary Module I: Preprocessing of audio files containing music from DEAM dataset.

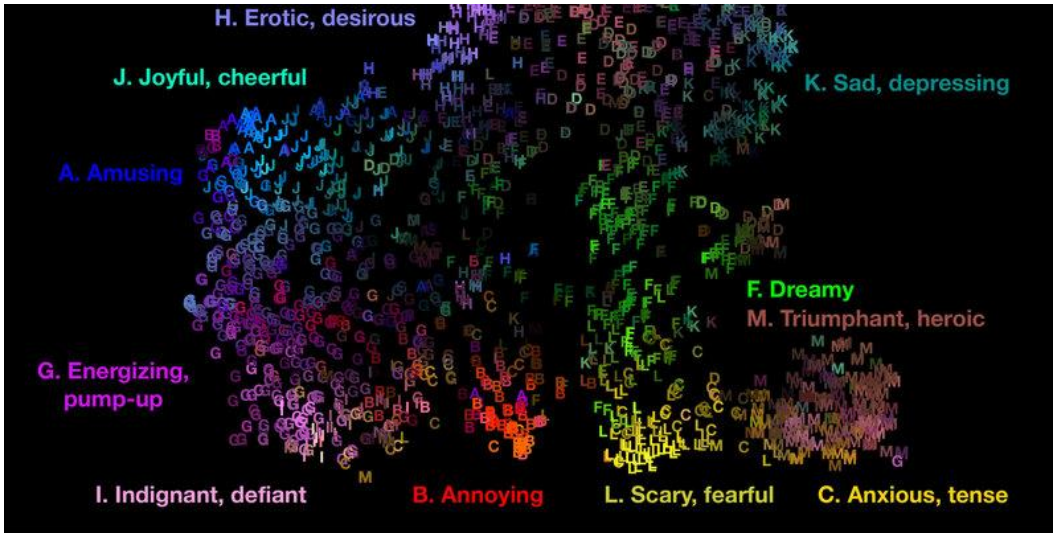

Figure 15.- Valence vs arousal graph of the songs included in DEAM.

The purpose of this module is to preprocess audio files in MP3 format from the DEAM dataset. The main goal is to convert these audio files into spectrograms, which we will use for emotion analysis based on audio features.

Firstly, we use the librosa library to load each audio file and convert them to waveform (wav) format with a sampling rate of 16,000 Hz. Afterward, to keep the file size manageable and minimize resource consumption during training, only 7 seconds of audio data are extracted from each file. Each section is chosen from the middle portion to ensure that represents the entirety of the audio file accurately. If a file is shorter than the desired target duration, constant values are added as padding until reaching the desired length.

Using the Melspectrogram transform on this audio snippet, we can generate a spectrogram. This Melspectogram is then converted to decibels to provide a more meaningful representation for perception. Additionally, another dimension is added to the expanded Melspectrogram to make it compatible with future models like convolutional neural networks. To conclude, each spectrogram is saved as a .npy file in the output directory.

The decision to extract only 7 seconds from each file has two main justifications. Firstly, by limiting the audio's length, we reduce computational load during training and

enhance efficiency in the process. Secondly, this limitation helps control storage size and prevents the flooding of large files within the system. Selecting an audio fragment from the middle ensures that we capture the essence of any song since many songs feature quieter introductions and endings, while their middle parts are often more illustrative.

Auxiliary Module II: Processing and Training of an Emotional Detection Model based on DEAM.

The current module trains a deep learning model to examine and predict emotions within music.

First, CSV files with valence and arousal annotations averaged for each song are loaded and merged into a consolidated dataset. These annotations provide emotional details about every individual melody.

Simultaneously, the code imports preprocessed spectrograms from the Auxiliary Module I. It goes through each spectrogram file, extracting the song ID and using it to fetch valence and excitation values from the annotation set. As a result, it generates two lists: one containing spectrograms and the other comprising corresponding valence and excitation values.

Next, the data is split into training and testing sets. This division is crucial for assessing how well the model performs on fresh data that it has not previously encountered.

The core of the code lies in the formulation of the deep learning model. We tailored a convolutional neural network structure for processing spectrograms and predicting both valence and excitation values, as presented in Table 20. Once this architecture is established, the model is compiled by implementing mean square error as a loss function, given its relevance to regression tasks.

Layer	Depth	Height	Width	Filter Height	Filter Width	Vector Length
Input	1	128	3501	-	-	-
Conv2D	32	126	3499	3	3	-
MaxPooling2D	32	63	1749	2	2	-
Conv2D	64	61	1747	3	3	-
MaxPooling2D	64	30	873	2	2	-
Flatten	-	-	-	-	-	1672320
Dense	-	-	-	-	-	128
Output	-	-	-	-	-	2

Table 20.- LeNet-style tabular representation for the valence and arousal extraction from music model.

Throughout the training process, designated checkpoints allow for intermediate versions of the model's parameters to be saved and retrieved if necessary. Upon completion of training, an overview detailing the model's progress throughout each epoch is provided through printed records displaying information such as loss and mean absolute error (MAE).

Finally, the trained model is saved to a file, allowing its use in future applications or analyses. This model was mainly used to add valence and arousal values and classifying the musical piece from VGMIDI.

Module III: Preprocessing of MIDI files from VGMIDI dataset

The purpose of this module is to handle MIDI files from a collection, extract their musical characteristics (such as notes and durations), as well as the associated emotional values (namely valence and arousal), and convert them into numerical sequences to use in machine learning models. One specific area where these transformed sequences find application is within recurrent neural networks, particularly long short-term memory (LSTM) networks.

To start, we import the required libraries and establish some initial parameters. These parameters include determining the length of the desired sequences and specifying the maximum and minimum values for valence and arousal, which we will subsequently use to normalize these attributes within a range of 0 to 1. Additionally, we define the file locations necessary for preprocessing the data resources.

The code incorporates additional functions that execute the following tasks:

- `scale_value`: Adjusts a value to fit within a specified range.
- `transpose_to_c`: Changes the key of a musical score to C, aiding in data standardization.
- `process_corpus`: This principal function handles MIDI files. It extracts notes, durations, valence, and arousal values from each file. The extracted information is stored in lists and saved as .pkl files for future reference.
- `unique` and `lookup`: These functions generate unique numerical representations for observed notes, durations, valence values, and excitations.
- `tables`: Generates and keeps track of tables displaying distinct values alongside their respective numerical counterparts.
- `sequences`: Converts note lists, duration lists, valence lists, and excitation lists into number sequences suitable for input into a neural network. These converted sequences are stored for later use.

Finally, the script undergoes a series of steps. Initially, it handles the collection of MIDI files, then creates tables containing numerical representations, and ultimately transforms the data into sequences suitable for model training.

Module IV: Processing and Training of an Emotionally Tuned Music Generation Model

The primary purpose of this module is to construct and train a neural network model that seeks to compose music using notes, durations, valence, and activation. The following explanation sheds light on how it functions:

Constants

Within the code are various constants that serve as parameters for the model. These constants encompass factors like the size of embeddings (`EMBED_SIZE`), the number of RNN units (`RNN_UNITS`), the inclusion or exclusion of attention (`USE_ATTENTION`), the total number of epochs (`NUM_EPOCHS`), batch size (`BATCH_SIZE`), fraction allocated for validation data (`VAL_SPLIT`), as well as an adaptable mixing option (`SHUFFLE`).

Input Data

Multiple datasets are extracted from pickle files in this process. These datasets include information about individual musical notes, durations, valence levels, and arousal degrees in terms of intensity or excitement level—stored under "uniques.pkl"—as well as training input and output datasets denominated "inputs.pkl" and "outputs.pkl," respectively.

Building the Model

The method `build_model` is responsible for establishing the structure of the neural model:

- We provide inputs for notes, durations, valence, and arousal.
- Notes and durations are embedded and merged with valence and arousal.
- An LSTM layer is introduced, to add complexity to the design.
- Additionally, there is an option to incorporate an attention mechanism. If enabled, the model can calculate attention weights, which emphasize certain features before sending them to subsequent layers.
- Output layers are designated to make predictions regarding both scores and durations.
- Lastly, optimization of the model occurs through compilation using the RMSProp optimizer.

Layer	Type	Output Shape	Connected to
input_1	InputLayer	(None, None)	-
input_2	InputLayer	(None, None)	-
embedding	Embedding	(None, None, 100)	input_1
embedding_1	Embedding	(None, None, 100)	Input_2
input_3	InputLayer	(None, None, 1)	-
input_4	InputLayer	(None, None, 1)	-
concatenate	Concatenate	(None, None, 202)	embedding, embedding_1, input_3, input_4
lstm	LSTM	(None, None, 256)	concatenate
dense	Dense	(None, None, 1)	lstm
reshape	Reshape	(None, None)	dense
activation	Activation	(None, None)	reshape
repeat_vector	RepeatVector	(None, 256, None)	activation
permute	Permute	(None, None, 256)	repeat_vector
multiply	Multiply	(None, None, 256	lstm, permute
lambda	Lambda	(None, 256)	multiply
pitch	Dense	(None, 15797)	lambda

duration	Dense	(None, 120)	lambda

Table 21.- LeNet-style tabular representation for the emotionally tuned music generation model.

Training the Model

Using the `train` function, the model is trained throughout this process:

- We define two checkpoints (`ModelCheckpoint`) to store the model weights during training, based on their loss.
- Additionally, we implemented an option called `EarlyStopping`, which automatically halts the training if no improvement in loss occurs after a certain number of epochs.
- Using input and output data previously loaded, the model undergoes training.
- To assess its performance on unobserved data, a fraction of it is held back for validation purposes.
- Once the training is completed, the model is saved in a .h5 file.

Visualizing Progress

To visually represent how well our model has developed during training, we utilize a `plot_loss` function. This tool aids us in determining whether our model tends to overfit or underfit while highlighting instances where learning occurs correctly.

Module V: Integration. "Main Module"

This module employs the valence and arousal parameters to customize music generation according to a specific emotional state. As a result, the generated pieces can mirror feelings such as tranquility or intensity.

When initiating the system, two pre-trained models are loaded: one for music generation and another for extracting valence and arousal from a human voice. Following this, every 3 seconds within a span of 3 minutes, we capture recordings of the user's voice. By the end of these 3 minutes, we will have accumulated a collection of 60 data points representing valence and arousal levels that will steer the process of musical generation.

With this information, the code proceeds to create music. It starts by utilizing a "seed" consisting of an initial sequence of notes and durations, which is then provided to the model alongside valence and excitation values to anticipate the following notes and durations for the composition. As it generates additional notes, it continually modifies the input sequence and repeats this process until it has generated sufficient notes for a three-minute musical piece or until it detects a "START" token that indicates a new sequence should begin.

After generating these predictions, they transform a data structure representing a score using the 'music21' library. Based on these predictions, individual notes, chords, and pauses are incorporated into this score while being assigned specific instrument associations, specifically piano in this case.

Ultimately, the produced score is stored as a MIDI file, which serves as a widely used format to depict music through digital events. The outcome is a three-minute musical composition that captures fluctuations in emotional qualities and intensity throughout the observation period.

5. Validation of the System

5.1 Validation of the valence and arousal extraction model from speech

The main objective of the developed model is to predict valence and arousal values based on spectrograms, which is essential for recognizing emotions in different domains. The model architecture includes convolutional layers, grouping layers, regularization, and fully connected layers. The validation results are presented below:

K-Fold Cross-Validation:

- A 5-partition cross-validation was performed to evaluate the overall model performance on different subsets of the dataset.
- The Mean Squared Error (MSE) obtained through this validation was approximately 0.324. This value gives an idea of the average error of the model in predicting valence and arousal.

Training Loss History:

- As shown in figure 16 the training loss (represented by the dashed red line) decreases with each epoch, indicating that the model is learning and adjusting its weights appropriately.
- The validation loss (represented by the solid blue line) also shows a decreasing trend. However, it is crucial to note if there is any point where the validation loss begins to increase while the training loss continues to decrease, as this could be indicative of overfitting.

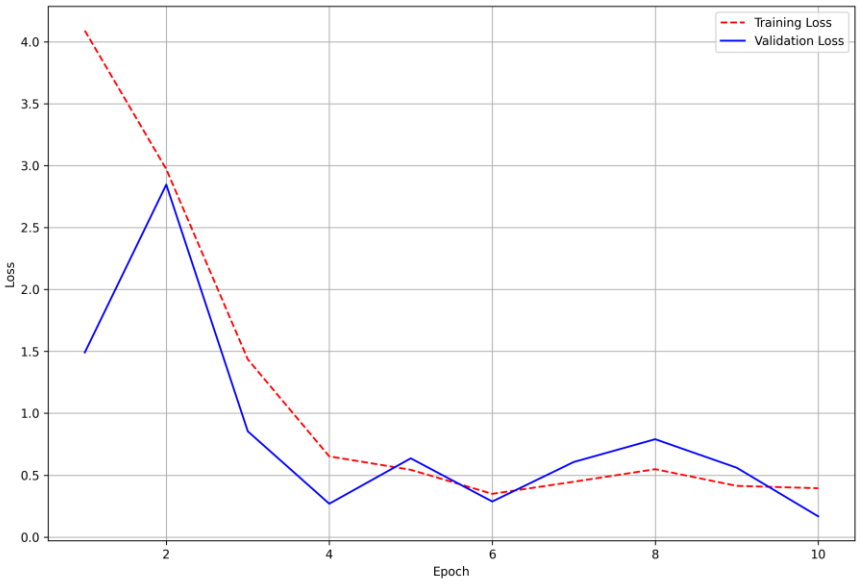

Figure 16.- Training loss and Validation loss history plot.

Conclusion:

- The model appears to perform reasonably well, with an MSE of 0.324 in cross-validation.
- The decreasing trend in training and validation losses suggests that the model is learning effectively and shows no clear signs of overfitting.

5.2 Validation of the valence and arousal extraction model from music

The developed model was designed to predict valence and arousal values from musical spectrograms. These values are essential for emotion recognition in musical contexts. The model structure comprises convolutional layers, max-pooling layers, and densely connected layers. The results obtained during the validation of the model are detailed below:

Mean Squared Error and Mean Absolute Error:

- The MSE is a commonly used metric for evaluating regression models. During the first epoch, the MSE was extremely high (4775363.0) but reduced drastically in the following epochs. At the end of training, at epoch 200, the MSE reached a value of approximately 0.339.
- The MAE, which provides a more interpretable idea of model error, ranged around values such as 0.447 at the end of the training, indicating that the model predictions have an average error of approximately 0.447 on the valence and arousal scale.

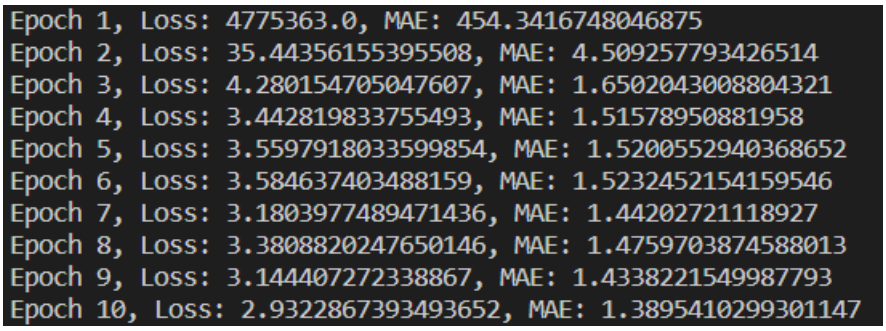

Figure 17.- Loss and Mean Absolute Error from the first 10 epochs.

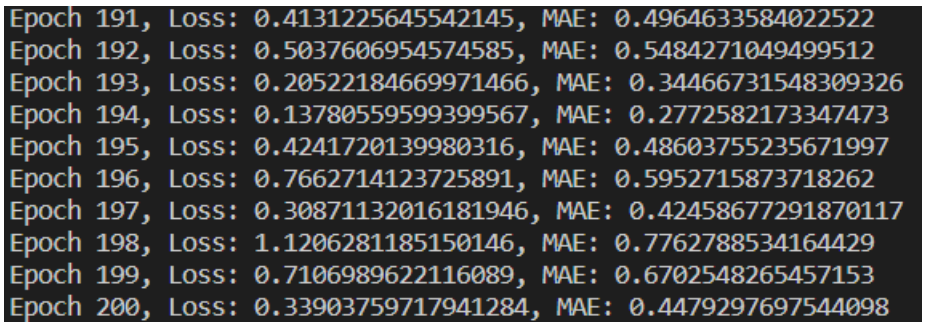

Figure 18.- Loss and Mean Absolute Error from the last 10 epochs.

Loss History During Training:

- By analyzing Figure 19 and 20, it is evident that the training loss (represented in blue) decreases consistently as the epochs progress, indicating that the model is learning and adjusting to the training data. However, the validation loss (represented in orange) shows a more volatile behavior: it starts to decrease along with the training loss but presents peaks and valleys along the process, which may be a sign of overfitting at certain epochs.
- It can be observed that, in the first epochs, there is a drastic reduction in loss, especially in the first epoch, where the model reduces its error significantly. Subsequently, the reduction is more gradual but still evident. However, towards the later epochs, the validation loss seems to stabilize, indicating that the model may not be benefiting significantly from further training epochs.

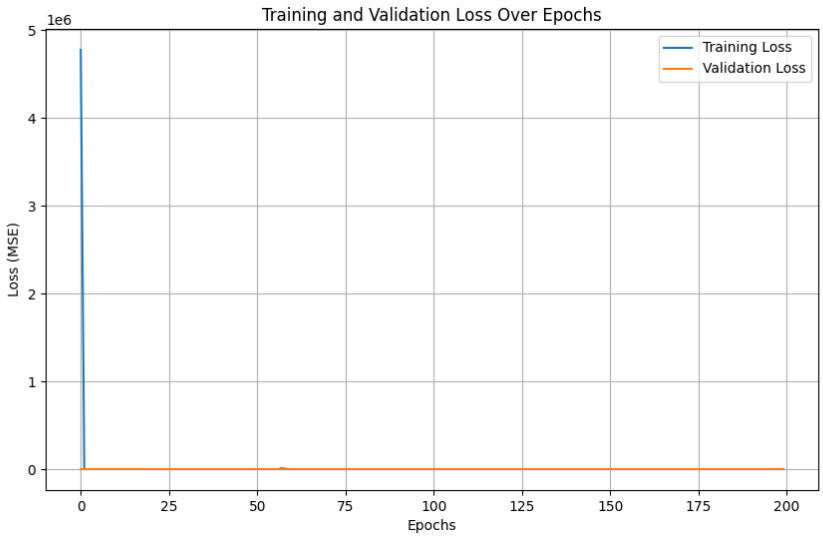

Figure 19.- Training loss and Validation loss history plot.

Figure 20.- Training loss and Validation loss history plot excluding the first epoch.

Conclusions:

- The model shows promising performance, with a clear reduction in loss during training.
- However, it would be beneficial to consider additional techniques to handle the possible overfitting observed at certain stages, such as regularization or fine-tuning of the model architecture.

5.3 Validation of the music generation model

The proposed model is designed to generate music using information about notes, durations, valence, and arousal. The model architecture combines embedding layers with recurrent long short-term memory units, and the option to include an attention mechanism is provided to enhance the model's ability to focus on certain inputs over time.

Early Stopping:

- Early Stopping is a regularization technique used to avoid overfitting by stopping model training once a given metric has stopped improving.
- In the code, we defined a "loss" monitor for Early Stopping, which means that training will stop if the loss does not improve after a given number of epochs.
- When Early Stopping is activated, the best weights of the model are restored thanks to the parameter `restore_best_weights=True`.

Model Checkpoints:

- Model Checkpoints are used to save the model or its weights at different points in the training.
- There are two checkpoints defined in the code:
    - `checkpoint1` saves the model with a name indicating the epoch and loss, but only if that loss is the best observed so far.
    - `checkpoint2` saves the best weights of the model in a file called "weights.h5", again based on the lowest observed loss.
- Both checkpoints are configured to monitor the loss and save only the improvements.

Training Loss History:

- The graph shows the training loss (represented by the blue dotted line) and validation loss (represented by the solid blue line) over epochs.
- It is observed that both training and validation loss decrease over time, indicating that the model is learning effectively. In this case, the validation loss is an aggregate of the pitch and duration loss, which is why it increases over epochs.

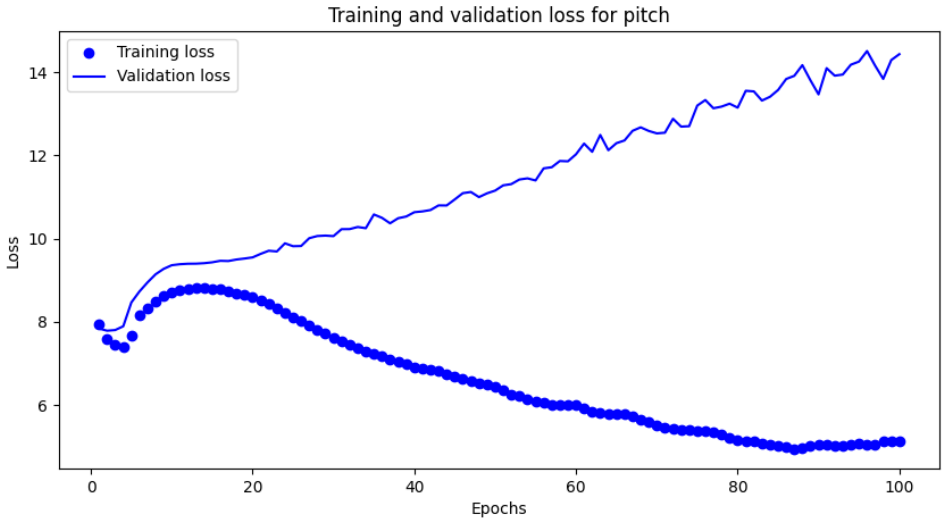

Figure 21.- Training loss and Validation loss history plot.

Conclusion:

- The music generation model appears to be training properly, with consistent decreases in loss in both the training and validation sets.
- The use of an attention mechanism may have contributed to the model's ability to handle complex temporal sequences.

5.4 Validation of the system

Description of the study

The study comprised 11 individuals ranging in age from 19 to 34, with the average age being 22.54 ± 4.32 years old. The sex classification of the statistical sample is as follows:

- 6 men (54.55%)
- 5 women (45.45%)

The system is designed to identify and generate music based on the following emotions:

- Anger (ANG)
- Excitement (EXC)
- Fear (FEA)
- Frustration (FRU)
- Happiness (HAP)
- Neutral (NEU)
- Sadness (SAD)
- Surprise (SUR)

Experiment phases

The experimental design was comprised of two critical phases aimed at addressing the aspects of recognition and emotional tone in music generation:

Recognition phase

During the initial phase of the experiment, individuals listened to music compositions specifically generated to express various emotions. The primary task assigned to the listeners was to recognize and determine the specific sentiment each musical piece aimed to portray. All participants listened to one composition per emotion, following the same order. After listening to all the pieces three times, they had to assign the emotions according to their criteria. By analyzing their responses, we can evaluate the system's proficiency in effectively capturing and transmitting these emotions through its generated music.

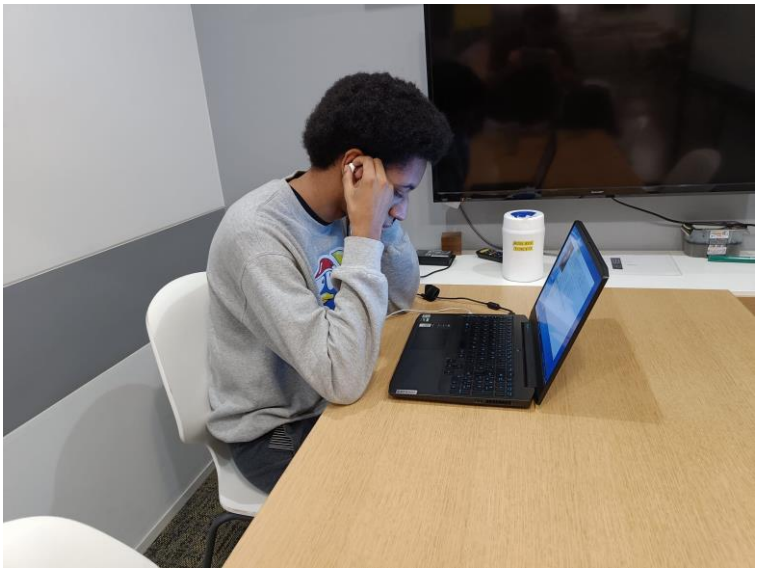

Figure 22.- Recognition test.

Generation phase

During the second stage of the experiment, individuals were instructed to exhibit different emotional states while standing in front of a microphone. Consequently, the software would create a musical composition based on the perceived emotion. Following this creative process, participants would listen to the resulting music piece and evaluate whether it effectively captured and mirrored the intended sentiment they were trying to convey.

Figure 23.- Generation test.

Utilizing this two-fold method, the experiment aimed to delve into the system's capacity for transmuting emotions into music and humans' perception and response to these auditory renderings of feelings.

Results

In the following section, we will examine the data gathered during the conducted experiments.

Results of the Recognition phase

In this stage, we evaluated the participant's ability to recognize specific emotions conveyed by the musical pieces generated by the system. Recognition accuracy was determined as a function of the percentage of hits.

The overall average hit rate of the participants was 71.59%. Specifically, the songs associated with the emotions of "Excitement" (EXC) and "Happiness" (HAP) had perfect accuracy, with 100% accuracy. Whereas the sentiments of "Fear" (FEA) and "Frustration" (FRU) had a lower recognition rate, with 27.27% and 45.45%, respectively.

When analyzing gender disparities, we determined that there were no substantial variations in the capacity to identify emotions between males and females, as seen in Figure 25, indicating that the system proves equally proficient for individuals of both genders.

Regarding age distribution, according to Table 22, younger participants tended to identify emotions slightly more accurately and quickly than older participants. However, the distinction did not demonstrate any substantial statistical significance.

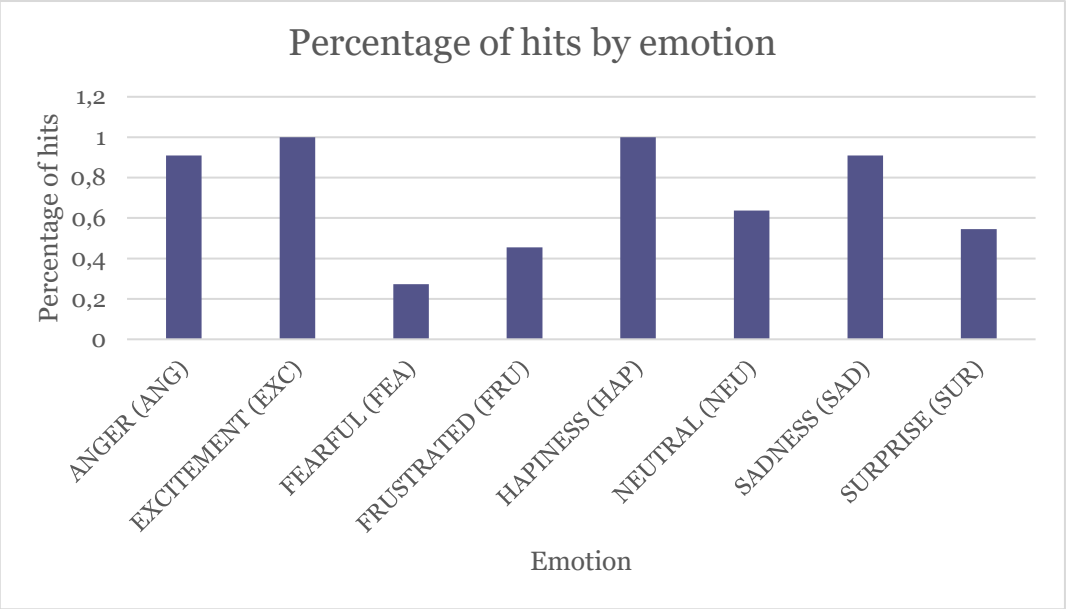

Figure 24.- Graph illustrating the emotional accuracy of listeners to music pieces generated by the system.

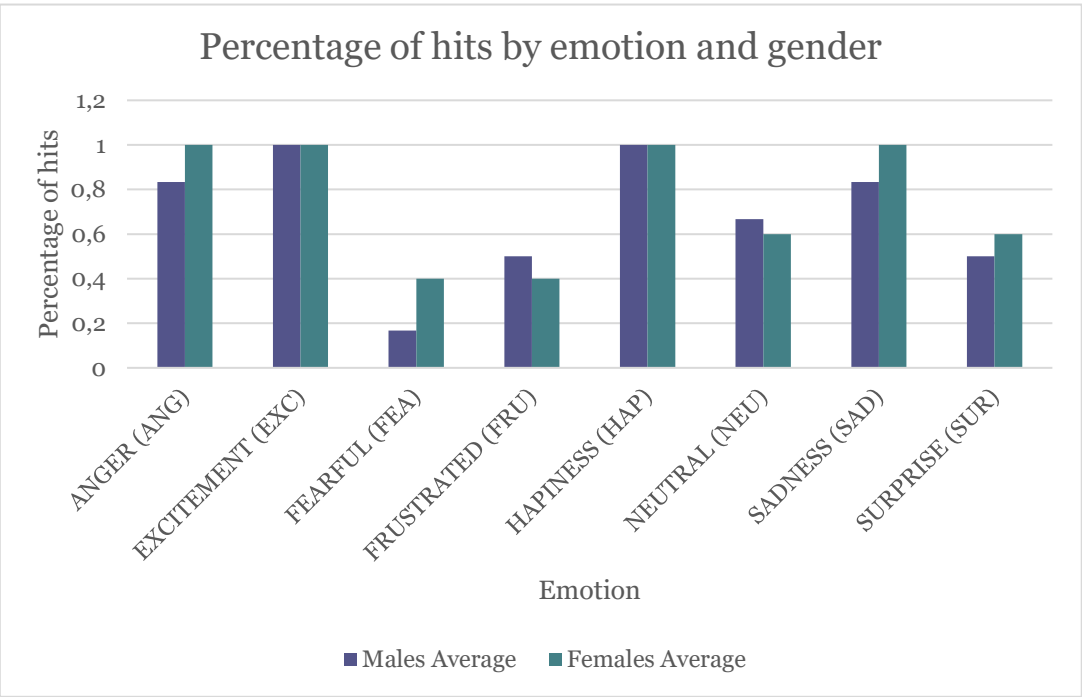

Figure 25.- Graph illustrating the emotional accuracy of listeners, separated by gender, to pieces of music generated by the system.

Age	Approximated Response Time	Accuracy
19	13,11	0,75
21	15,66	1
22	13,92	0,5
21	17,28	1
22	13,35	1
21	18,99	0,75
20	16,29	0,75

20	17,31	0,75
21	18,99	0,625
27	24,48	0,5
34	25,02	0,25

Table 22.- Approximated Response Time and Accuracy in the response from each participant.

Results of the Generation phase

During the second stage of the experiment, participants were asked to convey different emotional states in front of a microphone. In response, the software created a musical composition based on the perceived emotion. After the system completed the music generation procedure, the participants listened to the resultant music composition, assessing its efficacy in encapsulating, and reflecting their intended sentiment. Participants could conduct multiple trials of this evaluation.

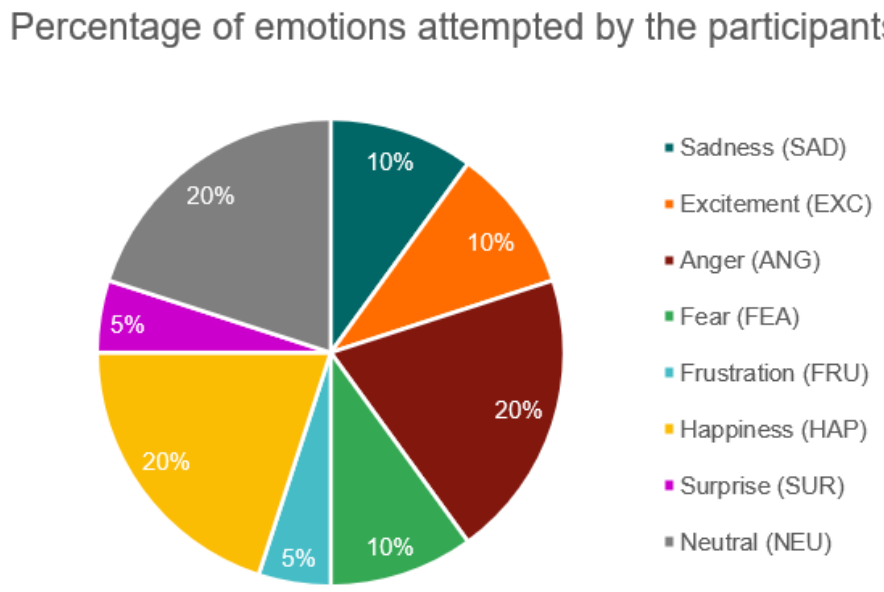

Figure 26.- Emotions attempted by the participants.

According to Figure 26, individuals opted for a diverse range of emotions in order to assess the system. Notably, "Anger" (ANG), "Happiness" (HAP), and "Neutrality" (NEU) stood out as the most chosen emotions due to their straightforward portrayal. Nonetheless, there was relatively less representation of emotions like "Frustration" (FRU) or "Surprise" (SUR).

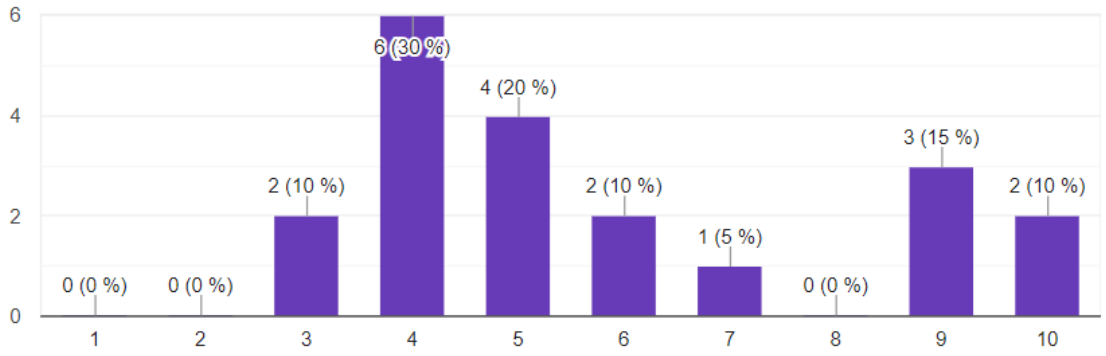

Figure 27.- Graph illustrating the correspondence between attempted emotion and generated music.

Initial results indicate a wide range of responses in the correspondence between the intended emotion and the generated music. On a scale of 1 to 10, where one is "Does not match at all" and ten is "Matches perfectly," a 25% of them felt that the music generated adequately reflected their emotion, giving ratings of 9 or 10. This is a positive indication of the software's ability to capture and reflect participants' emotions.

However, 60% of participants felt that the music did not adequately reflect their emotions, giving ratings between 3 and 5. These responses suggest areas for improvement in the music generation process.

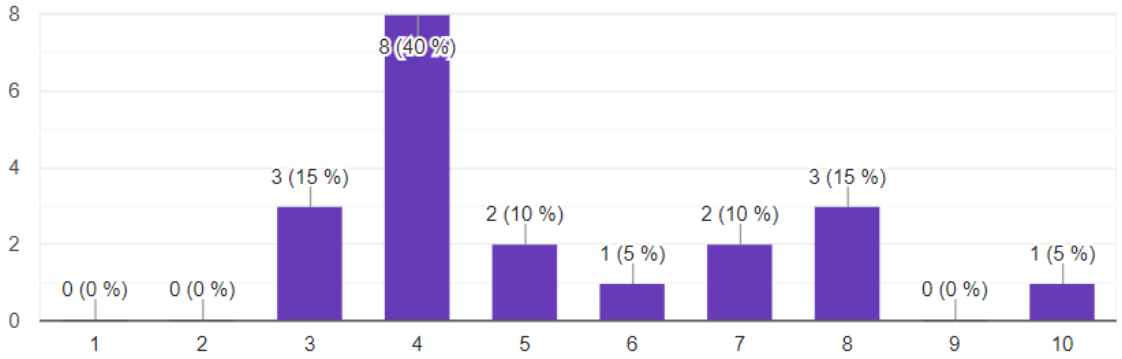

Figure 28.- Graph illustrating the clarity of emotion in the musical piece.

According to the data presented in Figure 28, the participants displayed a diverse range of opinions when assessing the emotional clarity of the music. While some individuals assigned high scores, indicating that they perceived clear emotional expression in the music, there was also a group who believed that its emotive portrayal lacked transparency, resulting in lower ratings ranging from 3 to 5.

Several individuals were pleasantly surprised upon discovering that the music resonated more with their emotions than anticipated. However, some observed that the songs did not capture their true sentiments entirely. Moreover, the range of emotions perceived while listening to the music varied significantly, as some songs evoked additional or alternative feelings beyond what was intended.

The participants were also asked about their opinions on the produced music. A substantial majority perceived that the melodies, notes, and rhythms were repetitive

while asserting that they strongly drew from vintage video game soundtracks. On occasion, they felt a sense of ambiguity in the music due to an excessive abundance of notes.

6. Conclusions and future work

The proposal made by Dr. Daisuke Saito has inspired us to design and develop the prototype of a tool for the recognition of emotions expressed through voice, which generates music based on such detection. The main objective of this project is to facilitate emotional recognition in people with autism, an area with great therapeutic and social potential.

The development of this project encompasses more than just writing the code. From the creation of the original concept through the conclusion of this master thesis, an investigation into the theoretical framework has been undertaken, examining voice manipulation methods, neural networks, melody creation, and understanding how people on the autism spectrum interpret emotions differently.

Based on our investigation of the state of the art, we have specified a series of requirements that we believe are necessary to address the topic of study. We have then prioritized those that allow us to obtain a robust prototype focused on the functionalities of emotional recognition through speech and the generation of adaptive music. The recommendations and suggestions offered by the participants will be crucial to refine and improve the emotion-based music generation process in future iterations of the experiment.

We used advanced voice analysis tools, neural networks, and machine learning methodologies. We must acknowledge that during our validations, we engaged participants without any prior diagnosis of autism spectrum disorders. Additional testing must be performed to guarantee that our proposed solution is practicable and beneficial for the target audience.

Undoubtedly, our prototype is still at an early stage of development and will need a long process of refinement and evolution. The generation of customized music that accurately reflects users' feelings is a technical and creative challenge, but it is also a promising therapeutic support tool.

From a technical and academic point of view, we think that the best way to consolidate the knowledge acquired during the degree, the master's degree, and the exchange is by carrying out a project like this one. We consider that thanks to our academic learning we have obtained the necessary keys to learn about the tools and technologies chosen to carry out this project and other knowledge, of equal importance, oriented to the rest of the processes that must conducted before starting the development. Thanks to our two tutors, we have been able to develop and work using a flow based on an agile methodology such as SCRUM, with emphasis on the prioritization of requirements and early validation of the prototype.

In conclusion, this project combines scientific knowledge, technical progress, and human compassion. Its goal is to create an innovative and valuable resource that helps improve the well-being of people with autism and others in their immediate social circle.

7. Bibliography

[1] Ferreira, X., & Oliveira, G. (2016). Autismo e Marcadores Precoces do Neurodesenvolvimento [Autism and Early Neurodevelopmental Milestones]. Acta medica portuguesa, 29(3), 168–175. https://doi.org/10.20344/amp.6790

[2] Strathearn, L. (2009). The elusive etiology of autism: nature and nurture? Frontiers in Behavioral Neuroscience, 3:11. https://doi.org/10.3389/neuro.08.011.2009

[3] Waterhouse, L. (2008). Autism Overflows: increasing prevalence and proliferating theories. Neuropsychology Review, 18(4), 273–286. https://doi.org/10.1007/s11065-008-9074-x

[4] Bleuler, E. (1950). Dementia praecox or the group of schizophrenias. International Universities Press.

[5] Ssucharewa, G. (1926). Die schizoiden Psychopathien im Kindesalter. (Part 1 of 2). European Neurology, 60(3–4), 235–247. https://doi.org/10.1159/000190478

[6] Evans, B. (2013). How autism became autism: The radical transformation of a central concept of child development in Britain. History of the human sciences, 26(3), 3–31. https://doi.org/10.1177/0952695113484320

[7] Rutter, M. (1972). Childhood schizophrenia reconsidered. Journal of autism and childhood schizophrenia, 2(4), 315–337. https://doi.org/10.1007/BF01537622

[8] Georgopoulos, M. A., Brewer, N., Lucas, C. A., & Young, R. L. (2022). Speed and accuracy of emotion recognition in autistic adults: The role of stimulus type, response format, and emotion. Autism research: official journal of the International Society for Autism Research, 15(9), 1686–1697. https://doi.org/10.1002/aur.2713

[9] Bestelmeyer, P. E. G., Kotz, S. A., & Belin, P. (2017). Effects of emotional valence and arousal on the voice perception network. Social Cognitive and Affective Neuroscience, 12(8), 1351–1358. https://doi.org/10.1093/scan/nsx059

[10] Bai, J., Peng, J., Shi, J., Tang, D., Wu, Y., Li, J., & Luo, K. (2016). Dimensional music emotion recognition by valence-arousal regression. 2016 IEEE 15th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), 42–49. https://doi.org/10.1109/icci-cc.2016.7862063

[11] Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. Proceedings of the International AAAI Conference on Web and Social Media, 8(1), 216–225. https://doi.org/10.1609/icwsm.v8i1.14550

[12] Pepino, L., Riera, P., & Ferrer, L. (2021). Emotion Recognition from Speech Using Wav2vec 2.0 Embeddings. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2104.03502

[13] Li, X. & Akagi, M. (2018). A Three-Layer Emotion Perception Model for Valence and Arousal-Based Detection from Multilingual Speech. Proceedings of Inerspeech, 3643-3647. https://doi.org//10.21437/Interspeech.2018-1820

Annex I. Questionnaire designed to evaluate the generation of emotionally intoned music.

Please answer the following questions based on the piece of music you heard after attempting to convey a specific emotion in front of the microphone.

Emotion Attempted:

- Anger
- Excitement
- Fear
- Frustration
- Happiness
- Neutral
- Sadness
- Surprise

Emotional Match:

On a scale of 1 to 10, where 1 is "Does not match at all" and 10 is "Matches perfectly," how well do you feel the piece of music generated reflects the emotion you tried to convey?

Clarity of Music:

On a scale of 1 to 10, where 1 is "Very Confusing" and 10 is "Very Clear," how would you describe the clarity of emotion in the piece of music?

Was there any other emotion you felt while listening to the piece, other than the one intended?

Yes (Please specify: _____)

No

Personal Description:

Please briefly describe how you felt when listening to the music generated. Do you think it captured the essence of your emotion? Why or why not?

Suggestions for Improvement:

Do you have any recommendations or suggestions for improving the emotion-based music generation process?

Annex II. Sustainable Development Goals

Degree to which the work is related to the Sustainable Development Goals (SDGs).

Sustainable Development Goals	High	Medium	Low	Not Applicable
ODS 1. **No poverty.**				X
ODS 2. **Zero hunger.**				X
ODS 3. **Good health and well-being.**	X			
ODS 4. **Quality education.**	X			
ODS 5. **Gender equality.**				X
ODS 6. **Clean water and sanitation.**				X
ODS 7. **Affordable and clean energy.**				X
ODS 8. **Decent work and economic growth.**				X
ODS 9. **Industry, innovation and infrastructure.**				X
ODS 10. **Reduced inequalities**.	X			
ODS 11. **Sustainable cities and communities**.		X		
ODS 12. **Responsible consumption and production.**				X
ODS 13. **Climate action.**				X
ODS 14. **Life below water.**				X
ODS 15. **Life on land.**				X
ODS 16. **Peace, justice, and strong institutions.**				X
ODS 17. **Partnerships for the goals.**		X		

Table 23.- Table illustrating the extent to which the work relates to the Sustainable Development Goals (SDGs).

Reflection on the relationship of the TFM with the SDGs and with the most related SDG(s).

Located within the United Nations Sustainable Development Goals, there exists a comprehensive approach to improving communication for those with autism spectrum disorder. These SDGs embody 17 interconnected objectives that tackle essential aspects of human development and sustainability, encompassing realms such as poverty reduction, inequality mitigation, healthcare accessibility, education enhancement, climate change action, and more. Because of its potential to improve

quality of life, increase inclusion, and contribute to more equal and responsive communities, this system is linked with several of these SDGs.

First, Goal 3: "Health and well-being" becomes relevant in this context. Individuals experiencing ASD sometimes encounter challenges in communicating and engaging socially, resulting in frustration and isolation. A specialized application aimed at enhancing their communication abilities holds the potential to raise their overall state of being. Effective communication is a principal component of psychological and emotional growth and enhancing it may result in reduced stress levels and an improved quality of life for people with ASD.

Quality education, which promotes Goal 4, is closely related to this initiative. The system is a beneficial educational tool as it can help people with ASD learn and practice communication skills. They can engage more actively in inclusive educational environments if we can help them improve their communication skills. This can lead to a more equitable and meaningful education, empowering individuals with ASD to reach their full academic and personal potential.

The notion of Reducing Inequalities, addressed by Goal 10, is also essential in this context. People with ASD frequently encounter communication difficulties, which can lead to social isolation and prejudice. This application provides a tool to express themselves and connect with others more fluidly and can, therefore, reduce inequalities. Empowering persons with ASD to communicate offers equitable opportunities and contributes to a more inclusive society.

Goal 11: "Sustainable cities and communities" is also related in this context. The application can contribute to building communities that are more inclusive and responsive to the needs of all their members. Promoting communication in urban and community contexts fosters engaged and mutual understanding. This helps create a sense of belonging and social cohesion in diverse and heterogeneous communities.

Finally, Goal 17: "Partnerships to Achieve the Goals," plays a crucial role in this system. The development of the application needs the involvement of a multitude of individuals who possess different skill sets and perspectives, including specialists in ASD, experts in emotion recognition and speech technologies, professionals in healthcare and education, as well as people diagnosed with ASD and their families. This collaborative effort among various disciplines is crucial for successfully attaining the Sustainable Development Goals within this field. By amalgamating diverse expertise and viewpoints, we can guarantee that the application will be efficacious, fittingly aligned to requirements, and specifically catered to meet the genuine needs of individuals with ASD.

While the Sustainable Development Goals mentioned above have a more direct relationship to the development of the aforestated system, other goals may also have some connection, although they may not be as obvious or straightforward. Below, I will briefly explain why the other SDGs may have less of a relationship to this initiative:

- SDG 1: End poverty: Although the application could indirectly impact improving the quality of life for people with ASD, it is not directly related to addressing poverty or its root causes.
- SDG 2: Zero hunger: There is no explicit connection between improving communication for people with ASD and eradicating hunger.
- SDG 5: Gender equality: Although communication is essential in all situations, this effort is not explicitly designed to combat gender differences.
- SDG 6: Clean water and sanitation: Improving communication for people with ASD is not directly related to providing clean water or sanitation services.
- SDG 7: Affordable and clean energy: This initiative has no explicit connection to affordable energy or the transition to clean energy sources.
- SDG 8: Decent work and economic growth: Although social inclusion of persons with ASD is critical for decent work, this implementation does not address economic development or job creation directly.
- SDG 9: Industry, innovation, and infrastructure: Although the app is a technology innovation, its focus on communication for people with ASD is not directly related to industry or infrastructure.
- SDG 12: Responsible production and consumption: The app is not related to promoting responsible production or consumption.
- SDG 13: Climate Action: Improving communication for people with ASD is not directly linked to climate action.
- SDG 14: Underwater Life and SDG 15: Terrestrial Ecosystem Life: These objectives pertain to the preservation of biodiversity and ecosystems and do not establish a direct correlation with communication among individuals with ASD.

The SDGs highlighted above are the ones with the most evident and direct link to our project. This initiative could improve the quality of life of people with ASD, foster inclusion and equality, and contribute to building societies that are more sustainable and responsive to the needs of all their members. This application can empower persons with ASD and promote their full and meaningful involvement in society by improving communication skills.

Annex III. Glossary of terms

The purpose of this section is to define the most relevant terms in a precise manner to avoid confusion and inconsistencies among project members.

A

Ability: the physical or mental power or skill needed to do something.

Acoustic: relating to sound or hearing.

Algorithm: a set of mathematical instructions or rules that, especially if given to a computer, will help to calculate an answer to a problem.

Analyze: to study something in a systematic and careful way.

Anxiety: an uncomfortable feeling of nervousness or worry about something that is happening or might happen in the future.

Apathy: behavior that shows no interest or energy and shows that someone is unwilling to act, especially over something important.

Arousal: numerical value that represents the intensity of an emotion provoked by a stimulus.

Artificial Intelligence: the use of computer programs that have some of the qualities of the human mind, such as the ability to understand language, recognize pictures, and learn from experience.

ASD (autism spectrum disorder): a range, or one of a range, of conditions, some more severe than others, that are also referred to as autism and that affect the development of social and communication skills and sometimes a person's behavior and interests.

Assist: to help.

Audio: a sound recording, recorded sound or a generated sound.

Autism: a brain condition that affects the development of social and communication skills in ways that can be severe or slight, and that can make someone's behavior and interests different from people without the condition.

Autistic: affected by or relating to the condition of autism, which affects the development of social and communication skills and can affect behavior.

B

Band: a group of musicians who play modern music together.

Bass: the lowest range of musical notes, or a man with a singing voice in this range.

Brain: the organ inside the head that controls thought, memory, feelings, and activity.

C

Camera: A device for taking photographs, using an aperture or lens to focus a visual image on to a light-sensitive material.

Cadence: the regular rise and fall of the voice.

Calmness: the quality of being peaceful, quiet, and without worry.

Chatbot: a computer program designed to have a conversation with a human being, especially over the internet.

Chronic: (especially of a disease or something bad) continuing for a long time.

Cognitive (Skills): connected with thinking or conscious mental processes.

Colloquialisms: an informal word or expression that is more suitable for use in speech than in writing.

Communication: the act of communicating with people.

Composition: a piece of music that someone has written.

Comprehension: the ability to understand completely and be familiar with a situation, facts, etc.

Confidence: the quality of being certain of your abilities or of having trust in people, plans, or the future.

Confusion: a situation in which people do not understand what is happening, what they should do or who someone or something is.

Contempt: a strong feeling of disliking and having no respect for someone or something.

Conversation: talk between two or more people in which thoughts, feelings, and ideas are expressed, questions are asked and answered, or news and information is exchanged.

D

Data: information, especially facts or numbers, collected to be examined and considered and used to help decision-making, or information in an electronic form that can be stored and used by a computer.

Database: a large amount of information stored in a computer system in such a way that it can be easily looked at or changed.

Dataset: a collection of separate sets of information that is treated as a single unit by a computer.

Decipher: to discover the meaning of something written badly or in a difficult or hidden way.

Disorder: an illness of the mind or body.

Duration: the length of time that something lasts.

E

Education: the process of teaching or learning, especially in a school or college, or the knowledge that you get from this.

Educator: a person who teaches people.

Emotion: a strong feeling such as love or anger, or strong feelings in general.

Enthusiasm: a feeling of energetic interest in a particular subject or activity and an eagerness to be involved in it.

Essence: the basic or most important idea or quality of something.

Expression: the act of saying what you think or showing how you feel using words or actions.

F

Facial: of or on the face.

Feelings: emotions, especially those influenced by other people.

Frequency: the number of times that a wave, especially a light, sound, or radio wave, is produced within a particular period, especially one second.

G

Generation: the action of causing something to exist.

Gesture: a movement of the hands, arms, or head, etc. to express an idea or feeling.

Grammar: (the study or use of) the rules about how words change their form and combine with other words to make sentences.

Genre: a style, especially in the arts, that involves a particular set of characteristics.

H

Harmony: a pleasant musical sound made by different notes being played or sung at the same time.

Health: the condition of the body and the degree to which it is free from illness, or the state of being well.

I

Identify: to recognize someone or something and say or prove who or what that person or thing is.

Image: any picture, especially one formed by a mirror or a lens.

Inclusion: the act of including someone or something as part of a group, list, etc., or a person or thing that is included.

Inequality: the unfair situation in society when some people have more opportunities, money, etc. than other people.

Interaction: an occasion when two or more people or things communicate with or react to each other.

Interpreting: to decide what the intended meaning of something is.

Instrument: an object, such as a piano, guitar, or drum, that is played to produce musical sounds.

Issue: a subject or problem that people are thinking and talking about.

J

Joy: great happiness.

K

Key: A set of musical notes based on one particular note.

Keyword: A word or idea that serves as a solution or explanation for something; a word, expression, or concept of particular importance or significance.

L

Life: the period between birth and death, or the experience or state of being alive.

Linguistic: connected with language or the study of language.

Loudness: The amount of noise something or someone makes.

M

Melody: a tune, often forming part of a larger piece of music.

Melspectograms: visual representation of the frequency content of an audio signal over time, where the frequencies are converted to a perceptual scale of pitches judged by listeners to be equal in distance from one another.

MIDI: Musical Instrument Digital Interface: a system for allowing electronic musical instruments to communicate with each other.

Modulation: a change in the style, loudness, etc. of something such as your voice to achieve an effect or express an emotion.

Mood: the way you feel at a particular time.

Movement: a change of position.

Motor (Skills): relating to muscles that produce movement, or the nerves and parts of the brain that control these muscles.

Music: a pattern of sounds made by musical instruments, voices, or computers, or a combination of these, intended to give pleasure to people listening to it.

Musician: someone who is skilled in playing music, usually as a job.

mp3: a brand name for a type of computer file that stores high-quality sound in a small amount of space, or the technology that makes this possible.

N

Neural network: a computer system or a type of computer program that is designed to copy the way in which the human brain operates.

Negative: not expecting good things, or likely to consider only the bad side of a situation.

Note: a single sound at a particular level, usually in music, or a written symbol that represents this sound.

P

Pattern: any regularly repeated arrangement.

Perception: a belief or opinion, often held by many people and based on how things seem.

Person: a man, woman, or child.

Pitch: the degree to which a sound or a musical note is high or low.

Positive: full of hope and confidence or giving cause for hope and confidence.

Psychiatrist: a doctor who is also trained in psychiatry.

Psychiatry: the part of medicine that studies mental illness.

Psychology: the scientific study of the way the human mind works and how it influences behavior, or the influence of a particular person's character on their behavior.

Psychotherapeutic: the methods or practice of psychotherapy; the branch of medicine or science concerned with this.

Q

Quality of life: the level of enjoyment, comfort, and health in someone's life.

R

Recognition: the fact of knowing someone or something because you have seen or heard him or her or experienced it before.

Reason: the ability of a healthy mind to think and make judgments, especially based on practical facts.

Rhythm: a strong pattern of sounds, words, or musical notes that is used in music, poetry, and dancing.

S

Sadness: the feeling of being unhappy, especially because something bad has happened.

Sample: a small amount of something that shows you what the rest is or should be like.

Schizophrenia: a serious mental illness in which someone cannot understand what is real and what is imaginary.

Semantic: connected with the meanings of words.

Sensor: a device that is used to record that something is present or that there are changes in something,

Sentiment: a general feeling, attitude, or opinion about something.

Skill: an ability to do an activity or job well, especially because you have practiced it.

Social: relating to activities in which you meet and spend time with other people and that happen during the time when you are not working.

Social media: websites and computer programs that allow people to communicate and share information on the internet using a computer or mobile phone.

Song: a usually short piece of music with words that are sung.

Speaker: a person who gives a speech at a public event.

Spectrum: a range of waves such as light waves or radio waves.

Speed: how fast something happens.

Speech: a formal talk given usually to a large number of people on a special occasion.

Struggle: to experience difficulty and make a very great effort in order to do something.

Syntactic: relating to the grammatical arrangement of words in a sentence.

Syntax: the grammatical arrangement of words in a sentence.

Symptom: any feeling of illness or physical or mental change that is caused by a particular disease.

Surprise: the feeling caused by something unexpected happening.

Sorrow: to feel great sadness.

T

Tempo: the speed at which a piece of music is played.

Text: The wording of anything written or printed; the structure formed by the words in their order; the very words, phrases, and sentences as written.

Therapy: The medical treatment of disease; curative medical or psychiatric treatment.

Therapist: someone whose job is to treat a particular type of mental or physical illness or disability, usually with a particular type of therapy.

Tone: a quality in the voice that expresses the speaker's feelings or thoughts, often towards the person being spoken to.

V

Valence: numerical value that represents the pleasantness of a stimulus.

Velocity: the speed at which something is traveling.

Voice: the sounds that are made when people speak or sing.

W

Wav: file format and file extension ending with .wav that was created by Microsoft and IBM and widely introduced with the release of Microsoft Windows 95. WAV files were an early way of playing audio files on the computer that was replaced with MP3 and WMV files.

Wave: signal pattern, such as that generated by sound, that changes at regular intervals