UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA BIOMÉDICA

TRABAJO FIN DE GRADO

DESIGN AND DEVELOPMENT OF AN EMOTION-DRIVEN SONG RECOMMENDATION SYSTEM FOR MUSIC THERAPY

ROSHNI MAHTANI VASHDEV

JUNIO 2024

TRABAJO DE FIN DE GRADO

Título:	Diseño y Desarrollo de un Sistema de Recomendación Musical basado en Emociones destinado a Musicoterapia
Título (inglés):	Design and Development of an Emotion-Driven Song Recommendation System for Music Therapy
Autor:	Roshni Mahtani Vashdev
Tutor:	Óscar Araque Iborra
Departamento:	Departamento de Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:	—
Vocal:	—
Secretario:	—
Suplente:	—

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

TRABAJO FIN DE GRADO

DESIGN AND DEVELOPMENT OF AN EMOTION-DRIVEN SONG RECOMMENDATION SYSTEM FOR MUSIC THERAPY

Roshni Mahtani Vashdev

Junio 2024

Resumen

Desde el comienzo de los tiempos, la música ha servido para mejorar el bienestar de las personas. A día de hoy, que la salud mental es una cuestión primordial, se necesitan urgentemente soluciones efectivas, siendo una de estas vías la musicoterapia. En este trabajo se ha diseñado, desarrollado y evaluado MoodRoot, una aplicación web capaz de crear listas de reproducción diseñadas para viajar entre estados emocionales con el objetivo de mejorar el estado de ánimo y acompañar en la depresión y la ansiedad. El sistema de recomendación musical integra dos componentes clave: el análisis emocional según el modelo de Valencia, Activación y Control (VAD, por sus siglas en inglés) [1], así como el análisis textual de las letras de las canciones. A través de este contenido emocional y lírico, el sistema proporciona recomendaciones personalizadas que se adaptan a los estados emocionales y las necesidades terapéuticas individuales, teniendo así un impacto positivo en la salud mental.

Para crear una ruta de canciones, el usuario debe elegir sus emociones inicial y objetivo, así como la primera canción y el número de canciones en su lista. La arquitectura del sistema se conforma por un servidor Streamlit para interactuar con el usuario, integración con la API de Spotify para generar las playlists en la cuenta personal del usuario, componentes de acceso a los datos de las canciones, y un módulo de cálculo algorítmico para generar las listas. Este último utiliza el modelo VAD para mapear las canciones según sus etiquetas de emociones para viajar de un estado emocional a otro, además de analizar la similitud entre las letras para que las canciones fluyan con naturalidad.

Por último, se realizaron pruebas con usuarios para evaluar la efectividad de MoodRoot en inducir emociones y se obtuvo retroalimentación para futuras mejoras. Los resultados de esta evaluación indican el potencial de MoodRoot como una herramienta para mejorar el bienestar emocional a través de la música, y enfatizan la importancia de seguir desarrollando y refinando este sistema para crear una experiencia emocional fluida para los oyentes.

Palabras clave: Musicoterapia, Sistema de Recomendación, Modelo VAD, BERT, Procesamiento de Lenguaje Natural (NLP), Embeddings de Letras, Extracción de Características, API de Spotify, OAuth (Autorización Abierta).

Abstract

The use of music to enhance the well-being of individuals can be traced back to the earliest periods of human history. At a time when mental health is a major concern and effective solutions are urgently needed, one such avenue lies in the realm of music therapy. In this project, MoodRoot has been designed, developed and evaluated. It is a web application that creates playlists that navigate between emotional states to improve mood and manage depression and anxiety. The music recommendation system seamlessly integrates two key components: the emotional analysis of Valence (pleasantness), Arousal (activation), and Dominance (control) (VAD) features [1], and the textual analysis of song lyrics. By harnessing the power of both emotional and lyrical content, the system aims to provide personalised recommendations tailored to individual emotional states and therapeutic needs, thereby fostering positive mental health outcomes.

In order to create a route of songs, the user must indicate the initial and target emotions, select the first song, and specify the number of songs in the playlist. The system's architecture incorporates a Streamlit server for user interaction, integration with the Spotify API for playlist generation in the user's personal account, data access components for song data retrieval, and an algorithmic calculation box. This last component uses the VAD model to map songs according to their emotion tags in order to navigate from one emotional state to another one. It also analyses the similarity between song lyrics so that the songs flow naturally.

Finally, user testing and evaluation was used to assess the effectiveness of MoodRoot in inducing emotional transitions and to gather feedback for future improvements. The results of the evaluation process demonstrate the potential of MoodRoot as a tool for enhancing emotional well-being through music, and emphasise the importance of continuing to develop and refine this system to create a seamless emotional journey for listeners.

Keywords: Music Therapy, Recommendation System, VAD (Valence-Arousal-Dominance) Model, BERT (Bidirectional Encoder Representations from Transformers), Natural Language Processing (NLP), Lyrics Embeddings, Feature Extraction, Spotify API, OAuth (Open Authorization).

Agradecimientos

A mis padres, por su amor incondicional y su apoyo inquebrantable. Gracias por creer en mí y por estar siempre a mi lado en cada paso del camino.

A mi hermana, Anjana, por ser la luz que ilumina mis días y mi inspiración constante.

A todos los profesores que me han acompañado y guiado hasta el fin de esta etapa. En especial, a mi tutor, Óscar Araque, por su paciencia, confianza y entrega.

A todas las personas que han aportado su granito de arena para que este trabajo saliese a la luz, gracias.

A las canciones, por ser refugio.

Contents

List of Figures

CHAPTER 1

Introduction

1.1 Context

In today's world, the focus on mental health has reached a crescendo, with concerns about psychological well-being at the forefront of societal discourse. In the context of this prevailing situation, there is a clear necessity for the development of innovative and effective solutions with the potential to enhance mental well-being. One avenue that holds immense promise in this endeavour is the realm of music, since art has the unique ability to evoke emotions.

Throughout history, music has been recognised for its therapeutic potential, dating back to ancient civilisations where philosophers like Plato and Aristotle acknowledged its healing power [10]. However, it was not until the 20$^{\text{th}}$ century, amidst the turmoil of World Wars I and II, that music therapy gained widespread acceptance as a formal practice [11]. Today, the therapeutic benefits of music are well-documented, with research demonstrating its efficacy in reducing stress, alleviating symptoms of depression and anxiety, and improving overall emotional well-being [12].

Given the significant role that music can play in emotional welfare, there is a clear need for systems that can harness this potential effectively. A sophisticated music recommendation system that aligns with users' emotional states and therapeutic needs could offer substantial benefits. Such a system would not only assist individuals in managing their mental health but also provide a valuable tool for therapists and mental health professionals.

1.2 Issue

We live in times where global crises like wars, climate change, and economic instability are prominent, leading to an increase in mental health issues. Suicide rates are alarmingly high, with more than 700,000 people taking their own lives each year, and for every suicide, there are many more attempted suicides [13]. Indeed, it is a grim reality that suicide ranks as the fourth leading cause of death among young people aged 15 to 29 globally, according to the World Health Organization.

Moreover, the COVID-19 pandemic has exacerbated mental health challenges, amplifying feelings of depression, anxiety, and stress across communities worldwide [14]. The disruption of daily routines, financial strain, social isolation, and uncertainty about the future have taken a significant toll on mental well-being.

It is crucial to recognise that mental health issues affect people from all walks of life and can arise from a variety of factors, including socioeconomic disparities, traumatic experiences, discrimination, and chronic health conditions. Unfortunately, not everyone has access to mental health professionals or feels comfortable seeking help due to stigma or financial constraints.

However, despite these challenges, there is hope. Music, with its universal appeal and therapeutic qualities, offers a promising avenue for supporting mental well-being. While it is essential to acknowledge that music is not a substitute for professional medical care, it can serve as a valuable tool to enhance emotional expression, reduce stress, and promote relaxation.

Given its accessibility and ability to evoke powerful emotions, music has the potential to reach individuals who may not have access to traditional mental health services. By incorporating music into mental health interventions, we can expand the reach of support networks and empower individuals to take an active role in their well-being.

1.3 Proposal

In response to the pressing need for innovative mental health solutions, this project proposes the development of a sophisticated music recommendation system: MoodRoot. Harnessing the therapeutic potential of music, the system aims to offer personalised interventions aligned with users' emotional states and therapeutic needs. By integrating models like the Valence, Arousal, and Dominance (VAD) framework [15] and advanced Natural Language Processing (NLP) techniques, the system will provide tailored music playlists designed to regulate mood and promote positive mental health outcomes.

The core objective of the proposed system is to create a seamless interface between music and mental health management. Utilising insights from research on the impact of music on emotional well-being, the system will analyse both musical features and lyrical content to generate personalised routes of songs that induce specific emotions. By offering users a curated selection of music tailored to their emotional states, MoodRoot aims to empower individuals in managing their mental health and fostering emotional resilience.

Moreover, this project seeks to contribute to the broader landscape of mental healthcare by providing a scalable and accessible tool for individuals and mental health professionals alike. By offering a user-friendly interface and evidence-based recommendations, the system aims to bridge the gap between music therapy and mainstream mental health interventions. Through rigorous testing and evaluation, the project endeavors to establish the efficacy and utility of MoodRoot in promoting positive mental health outcomes across diverse populations.

1.4 Project goals

The objectives pursued by this project are the following:

1. Analyse the impact of music on emotional well-being.

2. Develop an algorithm that uses Valence, Arousal and Dominance of songs, as well as lyrics embeddings, to create a route of songs that induces a certain emotion.

3. Design a friendly interface to use the algorithm.

4. Test the system to evaluate its functionality and effectiveness.

1.5 Structure of this document

This section offers a brief overview of the chapters included in this document. The structure is as follows:

Chapter 1. Introduction: Establishes the context in which this project is framed, outlines the main challenges to be solved and sets out the objectives to be achieved.

Chapter 2. Enabling Technologies: Provides information on the technologies used in the development of this project, including relevant frameworks and tools.

Chapter 3. Related Work: Reviews existing literature and previous research that is relevant to this project, highlighting gaps that this work aims to address and drawing comparisons with similar systems and methodologies.

Chapter 4. Architecture: Describes the software architecture of the components of the system and their integration and functionality. The algorithm and flow of information over time through the various components is also detailed.

Chapter 5. Evaluation: Outlines the methodology used to evaluate the system, followed by a presentation of the results.

Chapter 6. Conclusions: Summarises the main findings and contributions of the project, revisits the objectives to discuss their fulfillment, and proposes directions for future work.

CHAPTER 2

Enabling Technologies

The strategic utilisation of enabling technologies stands as a cornerstone in ensuring the success and efficacy of this project. In an ever-evolving landscape where technological advancements continuously redefine the boundaries of innovation, it is of paramount importance to leverage the right tools and frameworks. These enabling technologies serve as the building blocks upon which the project's infrastructure is constructed, shaping its capabilities, scalability, and overall effectiveness.

On the one hand, Python [16] serves as the primary programming language, supported by essential libraries such as Pandas [17], NumPy [18], and Scikit-learn [19] for data analysis and algorithm development, alongside Matplotlib [20] and Seaborn [21] for visualisation.

On the other hand, the integration of BERT (Bidirectional Encoder Representations of Transformers) [4] from HuggingFace enhances the Natural Language Processing (NLP) capabilities to analyse song lyrics. The project interface is constructed using Streamlit [22] and the Spotify API, with a focus on user experience.

These technologies collectively underpin the project's objectives, advancing music recommendation systems and exploring the intersection of music and emotion. This chapter will delve into each of these aspects in greater detail.

2.1 Programming Language: Python

Python [16] is a high-level, interpreted programming language that is renowned for its simplicity, readability, and versatility. Its syntax is clear and straightforward, making it an ideal choice for web development, data analysis, artificial intelligence, scientific computing, and automation. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, providing the flexibility to approach projects in the most optimal manner. Furthermore, Python is dynamically typed and interpreted, facilitating the development and debugging process.

The language supports modules and packages, which promote program modularity and code reuse. Python's interpreter and extensive standard library are freely available for all major platforms, enhancing its productivity and ease of debugging. Python's versatility is augmented by numerous libraries such as Pandas [17], NumPy [18], Scikit-learn [19], Matplotlib [20], Seaborn [21], and Streamlit [22], which provide powerful tools for data manipulation, machine learning, visualisation, and interface development, respectively. These libraries greatly enhance Python's capabilities and make it a popular choice for a wide range of tasks across different domains.

2.2 Data Analysis and Algorithm Development: NumPy, Pandas, and Scikit-learn libraries

In order to comprehend and preprocess the datasets, analyse their features and variables, and integrate them into a unified system, several Python libraries were required.

2.2.1 NumPy

NumPy [18], which stands for Numerical Python, is a fundamental library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a vast collection of high-level mathematical functions to operate on these arrays. NumPy's efficient array-processing capabilities make it an essential tool for scientific computing, data analysis, and general-purpose numerical tasks. Its performance and ease of use have established it as a foundational building block in the Python data science ecosystem, underpinning many other libraries like Pandas and Scikit-learn.

2.2.2 Pandas

Pandas [17] is a robust, open-source data manipulation and analysis library built on top of NumPy. It introduces two primary data structures: Series (one-dimensional) and DataFrame (two-dimensional), which are designed to handle structured data such as tables and time series. Pandas provides an extensive set of functions for data cleaning, transformation, merging, and aggregation, making it indispensable for data munging tasks. Its intuitive and flexible interface allows users to perform complex data operations with ease, thereby facilitating efficient data preparation and exploratory analysis in a wide range of applications.

2.2.3 Scikit-learn

Scikit-learn [19] is a robust and user-friendly machine learning library in Python, designed to facilitate the implementation of machine learning algorithms. It builds on NumPy, Pandas, and other scientific libraries to provide a comprehensive suite of tools for data mining and data analysis. The Scikit-learn library is suitable for classification, regression, clustering, and dimensionality reduction tasks, including model selection and evaluation tools. Its consistent API and extensive documentation make it accessible and enable rapid development and deployment of machine learning models. In this project Scikit-learn has been used to import the cosine similarity function.

2.3 Data Visualisation: Matplotlib and Seaborn libraries

The effective visualisation of data is of paramount importance for the comprehension of datasets and the facilitation of information understanding at a glance. Python offers a range of libraries that facilitate this process.

2.3.1 Matplotlib

Matplotlib [20] is a highly versatile plotting library for the creation of static, interactive, and animated visualisations in Python. It offers a comprehensive range of plotting functions to generate a wide variety of graphs and charts, from simple line plots to complex multi-axis plots. Matplotlib's highly customisable interface allows users to fine-tune every aspect of their plots, including colours, labels, and scales, ensuring precise control over the appearance of visualisations. Widely used in scientific research, data analysis, and engineering, Matplotlib serves as a foundational tool for data visualisation in Python.

2.3.2 Seaborn

Seaborn [21] is a high-level statistical data visualisation library built on top of Matplotlib, designed to facilitate the creation of informative and aesthetically pleasing figures. The software provides a simplified interface for the generation of complex plots, including heatmaps, time series plots, and categorical plots, with minimal code. Seaborn integrates seamlessly with Pandas DataFrames, facilitating the rapid and straightforward generation of graphics based on structured datasets. Additionally, it incorporates a range of pre-defined themes and colour palettes, which can be employed to enhance the aesthetic appeal of charts. This makes it an optimal choice for the production of publication-quality visualisations and the acquisition of deeper insights into data.

2.4 Song Lyrics Analysis: BERT

BERT [4], an acronym for Bidirectional Encoder Representations from Transformers, is a state-of-the-art pre-trained Natural Language Processing (NLP) model developed by Google in 2018 and made available through HuggingFace. BERT's major innovation lies in its ability to comprehend the context of a word based on all its surrounding words, enabling a bidirectional understanding of text. This capability allows BERT to capture complex linguistic patterns and semantic relationships, making it highly effective for a wide range of NLP tasks, including text classification, sentiment analysis, question answering, and language translation.

2.4.1 Model Architecture

The architecture of BERT is based on the Transformer model [2], which relies on self-attention mechanisms to process input text. Self-attention allows the model to focus on different parts of the input sequence simultaneously and capture complex relationships between words in the sequence.

The building block in BERT is the encoder part of the Transformer architecture [view Figure 2.1], since it transforms input sequences into continuous, abstract representations that encapsulate learnt information from the entire sequence. BERT's encoder comprises multiple identical layers, each consisting of a multi-headed attention mechanism and a fully connected network. The encoder in BERT is crucial for processing input sequences bidirectionally and capturing contextual information effectively.

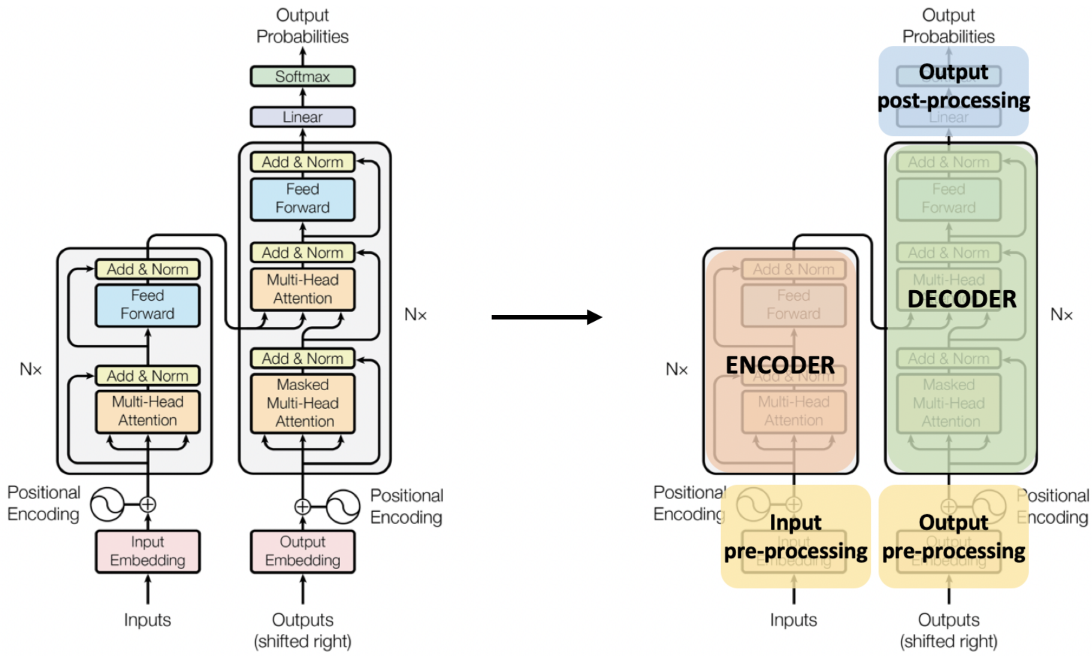

Figure 2.1: Transformer Model Architecture [2]

There are two main versions of BERT:

- **BERT-base:** Consists of 12 Transformer layers (L), 768 hidden layers (H), and 12 self-attention heads (A), with a total of 110 million parameters. When the model was initially developed, it was comparable in size to the OpenAI GPT model [23]. However, the GPT model has since undergone significant expansion, with the GPT-3 model comprising 175 billion parameters [24]. There are also key differences between BERT and GPT architectures and functionalities. In contrast to BERT, GPT (Generative Pre-trained Transformer) operates as a decoder in an auto-regressive manner, generating text by predicting the next word in a sequence based on the preceding words [3] [see Figure 2.2].

- **BERT-large:** Consists of 24 Transformer layers (L), 1024 hidden layers (H), and 16 self-attention heads (A), with a total of 340 million parameters. This large variant offers a greater capacity and accuracy for more complex NLP tasks.

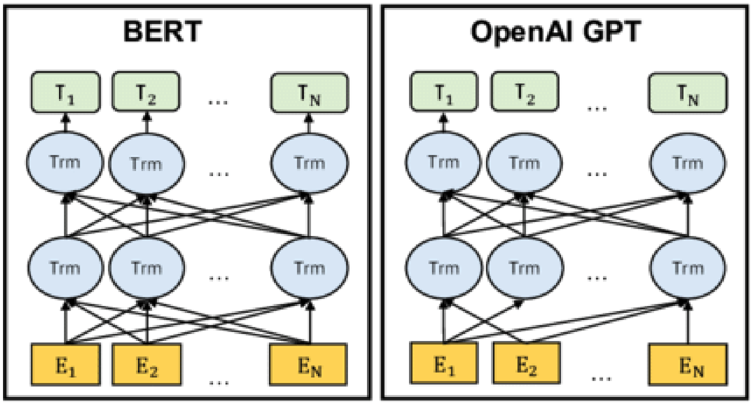

Figure 2.2: BERT vs GPT Architecture [3]

2.4.2 BERT Framework

The BERT framework comprises two main stages: pre-training and fine-tuning [view Figure 2.3].

1. **Pre-training**

   During the pre-training phase, BERT is trained on an enormous corpus of unlabelled text from the English Wikipedia [25] and BookCorpus [26]. This phase involves two unsupervised tasks: Masked Language Modelling (MLM) and Next Sentence Prediction (NSP).

   - **Masked Language Modelling (MLM):** In this task, 15% of the words in the input sequence are randomly masked, and the model is trained to predict these masked words based on their context.
   - **Next Sentence Prediction (NSP):** This task involves predicting if a pair of sentences is sequential in the original text. This helps the model understand sentence relationships.

   The pre-training phase allows BERT to learn deep bidirectional representations by conditioning on both the left and right context of each word, unlike traditional unidirectional models [27].

2. **Fine-tuning:**

   In the fine-tuning stage, the singular pre-trained BERT model is subjected to further training on labelled data specific to the target task. This process enables the creation of a distinct fine-tuned model for each NLP task, as the fine-tuning process adjusts the pre-trained parameters and adds an output layer specific to the task in question.

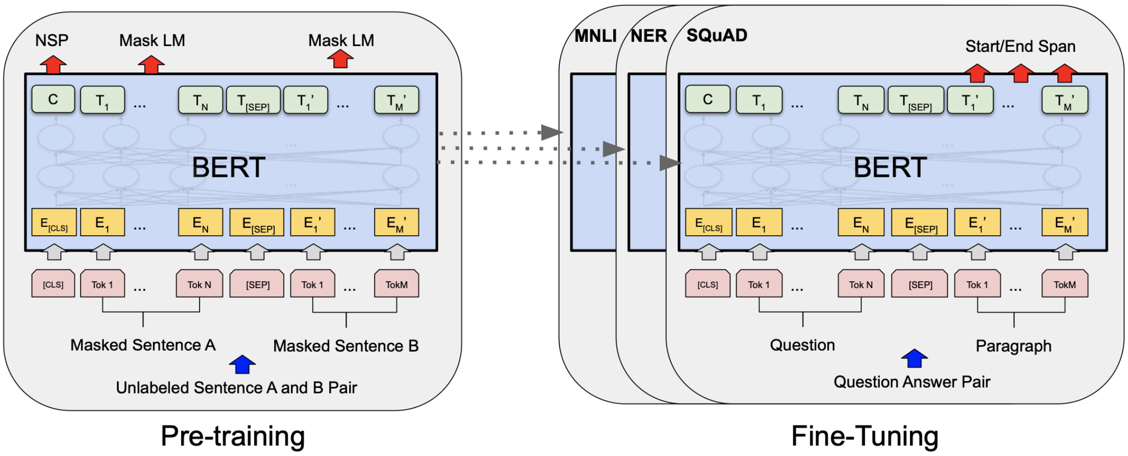

Figure 2.3: BERT Framework: Pre-training and Fine-tuning [4]

2.4.3 BERT Base Uncased and Feature Extraction Pipeline

In this project, the BERT base (uncased) pre-trained model from HuggingFace [28] is used to analyse song lyrics. The feature extraction pipeline [29] is implemented to obtain numerical representations from the lyrics. The key steps in the feature extraction process are as follows:

1. **Text Input Processing:** The song lyrics are inputted into the pipeline as raw text. BERT requires input text to be tokenized into a format it can process, typically involving splitting the text into words or subwords and converting these into numerical token IDs.

2. **Tokenization:** BERT's tokenizer splits the lyrics into subword tokens and maps each token to a unique identifier. It also adds special tokens [CLS] (classification) at the beginning and [SEP] (separator) at the end of the sequence to help BERT understand the structure of the input.

3. **Embedding Extraction:** The tokenized lyrics are then converted into numerical feature vectors. BERT provides contextual embeddings, where each token is represented by a dense vector that captures the semantic meaning of the word in the given context.

4. **Contextual Encoding:** BERT processes the tokenized input through multiple layers of transformers, capturing bidirectional contextual information. This means each token's representation is influenced by the entire sequence, providing rich, contextual embeddings.

5. **Feature Aggregation:** Given that each token in the lyrics is embedded individually, these embeddings are typically aggregated (in this case, by averaging) to reduce the dimensionality of the extracted features. This process results in the generation of a single, 768-dimensional feature vector, which represents the entirety of the song lyrics.

In conclusion, BERT serves as an effective tool in this project due to its capabilities for processing and understanding song lyrics. By leveraging its bidirectional transformer architecture, BERT captures intricate contextual nuances and semantic relationships within the text. This enables the extraction of rich, meaningful embeddings that are then utilised for lyric comparison.

2.5 Interface Design: Streamlit and Spotify API

In order to develop the web application and enable users to test and evaluate the system, the Streamlit library from Python has been employed. Furthermore, the application has been linked to the Spotify Web API, enabling the creation of playlists within the user's Spotify account.

2.5.1 Streamlit

Streamlit [22] is an open-source Python library that enables the creation and sharing of web applications for data science and machine learning. It facilitates the transformation of Python scripts into interactive, visually engaging web apps without requiring front-end development expertise. With its simple syntax, Streamlit enables rapid app development by incorporating interactive widgets such as sliders, buttons, and text inputs, which update in real-time as users interact with them. Seamlessly integrating with popular Python libraries, including NumPy, Pandas, and Matplotlib, Streamlit makes it easy to visualise data and build machine learning models. Additionally, it offers various layout options for customisable interfaces and provides easy deployment options through platforms like Streamlit Cloud, facilitating the sharing of applications with a broader audience. This capability makes Streamlit a powerful tool for data exploration, model demonstration, and rapid prototyping.

2.5.2 Spotify API

Spotify Web API [30] allows the development of applications that can interface with Spotify's streaming service. It is a set of RESTful web services provided by Spotify with different endpoints that return JSON metadata about music artists, albums, and tracks, directly from the Spotify Data Catalogue. This allows a wide range of functionalities which include fetching content metadata, obtaining recommendations, managing and creating playlists, or controlling playback. To interact with the Spotify API, one must have an access token, which is a string that contains the credentials and necessary permissions.

In this work, the Streamlit application establishes a connection to the Spotify Web API, enabling the authentication of users and the direct creation of emotion-inducing playlists in their personal accounts. Therefore, the user must grant the application access permissions to their Spotify data and features. This process is known as authorization. Spotify implements the OAuth 2.0 [6] authorization framework [see Figure 2.4].

Figure 2.4: OAuth 2.0 Authorization Framework Implementation by Spotify [5]

2.6 Authorization Flows: OAuth 2.0

In the conventional client-server authentication model, the client seeks to access a protected resource on the server by verifying its identity using the resource owner's credentials. This implies that to grant third-party applications access to these restricted resources, the resource owner must provide its credentials to the third party. This approach leads to various issues and constraints since third-party applications gain overly broad access to the resource owner's protected resources.

These issues are addressed in the OAuth (Open Authorization) framework [6] by introducing an authorization layer that separates the role of the client from that of the resource owner. In OAuth, the client requests access to resources controlled by the resource owner and hosted by the resource server, and is issued a different set of credentials than those of the resource owner. An authorization server issues an access token to the third-party application with the approval of the resource owner. This way, the client uses the access token to access the protected resources hosted by the resource server [view Figure 2.5].

```
+--------+                               +---------------+
|        |--(A)- Authorization Request ->|   Resource    |
|        |                               |     Owner     |
|        |<-(B)-- Authorization Grant ---|               |
|        |                               +---------------+
|        |
|        |                               +---------------+
|        |--(C)-- Authorization Grant -->| Authorization |
| Client |                               |     Server    |
|        |<-(D)----- Access Token -------|               |
|        |                               +---------------+
|        |
|        |                               +---------------+
|        |--(E)----- Access Token ------>|    Resource   |
|        |                               |     Server    |
|        |<-(F)--- Protected Resource ---|               |
+--------+                               +---------------+
```

Figure 2.5: OAuth 2.0 Protocol Flow [6]

2.6.1 OAuth Roles

There are four defined roles in OAuth:

1. **Resource owner:** An entity capable of granting access to a protected resource. In the context of this application, the resource owner is the end-user who grants access to their Spotify account.

2. **Resource server:** The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens. In this case, Spotify is the resource server, hosting the user's playlists and music data.

3. **Client:** An application making protected resource requests on behalf of the resource owner and with its authorization. In this project, the client is the developed application that creates playlists for the user.

4. **Authorization server:** The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization. In this case, Spotify's authorization server is responsible for issuing these tokens.

2.6.2 Authorization Grant

The authorization grant is a necessary credential that the client must obtain from the resource owner in order to get an access token. This specification defines four grant types:

1. **Authorization code:** The client directs the resource owner to an authorization server, which in turn directs the resource owner back to the client with the authorization code. This way, the authorization code is obtained by utilising an authorization server as an intermediary between the client and the resource owner. The application developed uses this type of authorization grant, since it is the safest option.

2. **Implicit:** The resource owner issues an access token directly to the client, there are no intermediate credentials like an authorization code. This approach is less secure, since the server does not authenticate the client.

3. **Resource owner password credentials:** In cases where the client is highly trustworthy or when other authorization grant types are not available, the resource owner password credentials can serve as an authorization grant. Even though this grant type requires direct client access to the resource owner credentials, these are used for a single request and are exchanged for an access token, so the client does not store the credentials for future use.

4. **Client Credentials:** These can be used as an authorization grant when the client and the resource owner are the same person. It requests access to protected resources based on an authorization previously arranged with the authorization server.

2.6.3 Access Tokens and Refresh Tokens

As previously stated, access tokens are credentials that are used to access protected resources. An access token is a string that represents an authorization issued to the client, and it is usually opaque to the client. Tokens represent specific scopes and duration of the access, granted by the resource owner and enforced by the resource server and authorization server.

These access tokens expire, which is why refresh tokens are needed. These credentials are employed to obtain a new access token when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope. It should be noted that access tokens may have a shorter lifetime and fewer permissions than those authorized by the resource owner.

2.7 Development Environment: Jupyter Notebook

Jupyter Notebook [31] is an open source web application that allows users to create and share documents that integrate live code, equations, visualisations, and narrative text. Named after its support for Julia, Python, and R, it extends its versatility to numerous languages via different kernels. Throughout this project, since Python serves as the core language, it is driven by the Python3 ipykernel.

Built upon an open document format based on JSON, Jupyter notebooks archive users' entire sessions, capturing code, textual narratives, equations, and rich output. These notebooks interact with computational kernels via the Interactive Computing Protocol, an open network protocol that uses JSON data over ZeroMQ (a high-performance asynchronous messaging library) and WebSockets [32] (a communication protocol that allows real-time interaction between the client and server over a single TCP connection). Kernels are processes that run interactive code in specific programming languages, return results to the user, and handle tasks such as tab completion and introspection requests. This integration enables users to create comprehensive, interactive documents that combine executable code with explanations and visuals, streamlining workflows for data analysis, research, and education.

CHAPTER 3

Related Work

This chapter presents related work to provide an understanding of the foundations of MoodRoot. Firstly, the relation between music and mental health will be reviewed by examining studies that investigate the biological effects of music on mental health and the brain. Secondly, emotion models such as the Valence-Arousal-Dominance model [1] will be examined in detail, as it serves as the foundation for mapping emotions and therefore songs to create playlists. The study that generated the dataset on which MoodRoot is based is also described. Finally, other music recommendation systems will be analysed in order to gain an understanding of their functioning.

3.1 Biological Effects of Music

The efficacy of music therapy has been demonstrated in the treatment of a range of conditions, including mental disorders such as depression [33], anxiety [34], and schizophrenia [35], as well as neurological conditions such as stroke [36], epilepsy [37], multiple sclerosis [38], and Parkinson's disease [39]. The benefits of music therapy include improved heart rate [40], motor skills [39], stimulation of the brain [41], and enhancement of the immune system [12].

For instance, a study conducted by the 'Creative Practice as Mutual Recovery' project, which was funded by the UK's Arts and Humanities Research Council [34], demonstrated that group drumming had a positive impact on the well-being of mental health service users. During the course of 10 weeks, patients who participated in group drumming exhibited a reduction in depression and anxiety, as well as an increase in social resilience and mental well-being. Besides, the patients' immune profiles shifted from pro-inflammatory to anti-inflammatory.

3.1.1 Neurochemical Effects of Music

There is also a study that reviews the scientific evidence on the neurochemical effects of music on health and well-being [42]. It states that the neurochemical systems engaged in music-based activities are the following:

- **Reward, motivation, and pleasure:** Dopamine is crucial in reward and motivation circuits during musical activities, interacting with the endogenous opioid system (the body's natural painkillers) to mediate pleasure associated with music.
- **Stress and arousal:** Music influences stress and arousal through cortisol, corticotrophin-releasing hormone, and adrenocorticotropic hormone.
- **Immunity:** Music-based activities cause immunological changes, though the specific mechanisms remain unclear.
- **Social affiliation:** Music enhances social bonding, linked to the release of neurohormones like oxytocin.

In summary, the multifaceted neurochemical impacts of music underline music's potential as a therapeutic tool for enhancing health and wellness.

3.1.2 Music and the Brain

The human brain harmonises advanced rational capabilities with primal emotional responses due to its unique structure [43]. The prefrontal cortex, responsible for rational decision-making, coexists with older limbic structures, which are the core of our emotional and memory circuitry. This combination enables humans to engage in complex abstract thinking while still being driven by basic instincts. An understanding of the interaction between these brain regions is essential for comprehending human behaviour in both health and disease.

The brain's response to music is complex, engaging multiple areas, including the auditory, visual, and motor cortices, the cerebellum, emotional centres such as the amygdala, and memory structures like the hippocampus. These regions collaborate to interpret sequences of sounds as music. The activation patterns observed in these areas are dependent on the type of music and the activity being performed, whether listening, rehearsing, or improvising. This understanding can enhance music therapy by optimising brain function through the selection of specific types of music to evoke desired cognitive, motor, or emotional responses.

3.2 Emotion Models

Emotions are conscious mental reactions experienced by subjects as strong feelings directed toward objects or situations that are associated with physiological, behavioural, and cognitive changes [44]. Throughout time, different strategies have been adopted in order to model emotions. In NLP, emotion modelling has evolved from two main approaches: the categorical model and the dimensional model.

3.2.1 Categorical Emotion Models

The categorical model, based on the work of Ekman [45] and Plutchik [46], suggests that human emotions can be represented as a set of basic emotions. On the one hand, Paul Ekman defines the six primary emotions (sadness, happiness, disgust, anger, fear and surprise) based on pan-cultural facial expressions [45]. On the other hand, Plutchik presents a wheel model [46] to describe eight discrete primary emotions (joy, trust, fear, surprise, sadness, anticipation, anger, and disgust), which can then be combined to form more complex emotions [view Figure 3.1]. Although categorical models offer a clear method for identifying emotions, they have limitations in accounting for ambiguity or mixed emotions.

3.2.2 Dimensional Emotion Models: Valence-Arousal-Dominance

In contrast, the dimensional model, based on the work of Russell and Mehrabian [1], proposes that emotions can be captured as points in a multidimensional space. This dimensional approach allows the conceptualisation and measure of emotions, and it is defined by three independent dimensions:

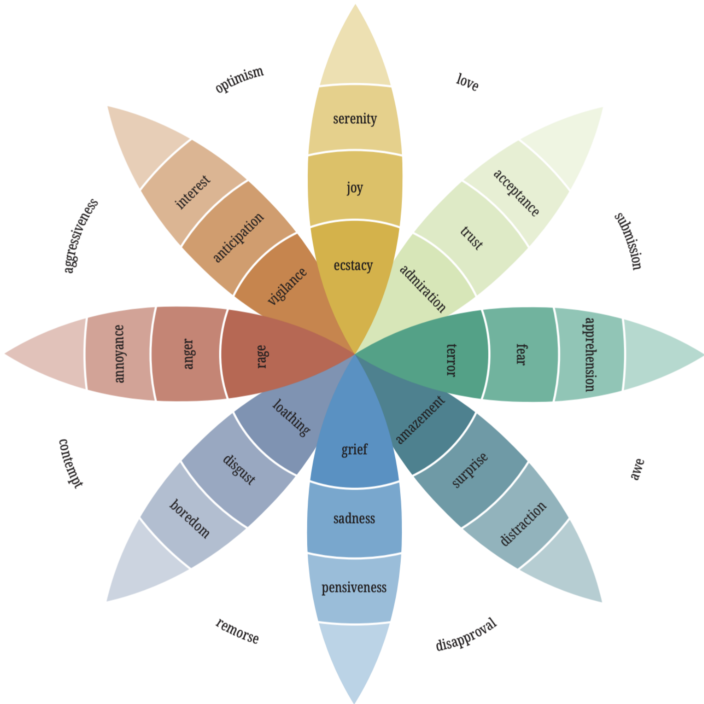

Figure 3.1: Plutchik's Wheel of Emotions [7]

- **Valence:** Represents the pleasantness or unpleasantness of an emotional state, ranging from negative/unpleasant to positive/pleasant.
- **Arousal:** Reflects the intensity or activation level of the emotion, ranging from calm/unaroused to excited/aroused.
- **Dominance:** Indicates the degree of control or influence felt in the emotional state, ranging from controlled/submissive to in control/dominant.

This dimensional model, known as the Valence-Arousal-Dominance (VAD) model, overcomes the limitations of the categorical model by allowing for the representation of mixed or ambiguous emotions. In the VAD model, the term "emotion" is not restricted to occasional feelings of excitement or intense pleasure. Instead, an individual is perceived as being in a constant emotional state, which can be described as a region within a three-dimensional space.

As shown in Figure 3.2, emotions can be mapped in this three-dimensional space. In this case, the figure presents the distribution of Ekman's six basic emotions [45] within the VAD space from Russell and Mehrabian's model [1].

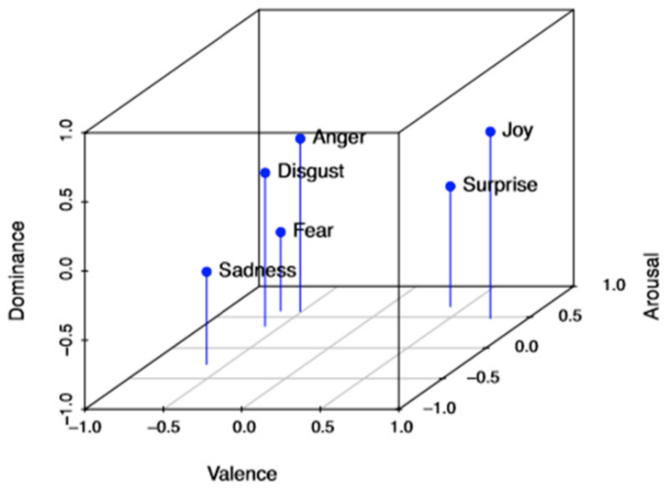

Figure 3.2: Ekman's Six Basic Emotions Mapped in the VAD Model [8]

Another characteristic of the VAD model is that the relationship between valence and arousal is not linear, but presents a U-shaped pattern. Therefore, words with very high or low valence tend to be more arousing than neutral words. A similar phenomenon occurs between dominance and arousal, with high and low dominance words being more arousing. In contrast, the relationship between valence and dominance is more linear, with positive emotions associated with higher dominance.

What makes the VAD model unique is the inclusion of the dominance dimension. Most other models do not explicitly consider this aspect of emotional experience, such as Russell's circumplex model of affection [47]. Moreover, the VAD model has been extensively validated through empirical studies that have collected ratings of large numbers of words on the three dimensions [48]. This provides a rich dataset for studying emotional semantics.

In summary, the VAD model offers a dimensional, orthogonal, and empirically-grounded approach to conceptualising emotions that differs from categorical models and provides a more nuanced representation of emotional experiences. The aforementioned characteristics permit the application of the VAD model in domains as diverse as psychology, affective computing, and sentiment analysis.

3.3 The Musical Sentiment Dataset

MoodRoot uses the MuSe (Musical Sentiment) dataset [9] as a foundation to create playlists of songs that pursue to induce a certain emotion. This dataset uses the VAD model to map songs within the VAD space based on their associated emotion tags [49], which are the third most prevalent descriptors of songs in Last.fm [50].

The MuSe dataset [9] comprises 90,408 songs along with sentiment information and various metadata. Understanding the steps involved in creating this dataset is crucial, as it constitutes the framework for developing this entire project:

1. **Seed stage: Mood labels collection from Allmusic.com**

   The first step involves collecting mood labels from Allmusic.com [51]. AllMusic is an American online music database that catalogues more than 3 million album entries and 30 million tracks. Alongside the platform and users' reviews, it also contains a section of "Mood and Themes" related to a song, album or artist. In total, 279 mood labels were created manually by editors at Allmusic.com. This approach ensures the collection of a wide range of moods and a balanced dataset.

2. **Expansion stage: User-generated tags collection from Last.fm**

   The 279 mood tags from AllMusic are used to collect song objects from the Last.fm API [52] with more tags, with the objective of obtaining a maximum of 1,000 songs per mood label. In addition, other metadata is also collected, including the artist and album names, as well as the number of listeners.

   A total of 131,883 songs were collected using this approach, but after removing duplicate songs (there are songs collected for multiple tags), the corpus consisted of 96,499 songs. A total of 261,000 unique tags were identified, with the most frequent being genre-related, including rock (29,810 times), alternative (24,763 times) and indie (23,006). Nevertheless, affective tags are also among the highly frequent tags, including "chill" (11,841 times), "sad" (8,350 times) and "melancholy" (8,344 times).

3. **Filtering stage: Mood tags identification with WordNet-Affect**

   In order to identify which labels are emotion-related, it is necessary to preprocess them [53]. This can be achieved by comparing the list of tags to the lemmas in WordNet-Affect [54]. WordNet-Affect is an extension of WordNet [55], an English lexical database that provides sets of synonyms for English words. WordNet-Affect comprises 1,606 emotional words and, when matched to the corpus, the 216,000 tags are reduced to 873.

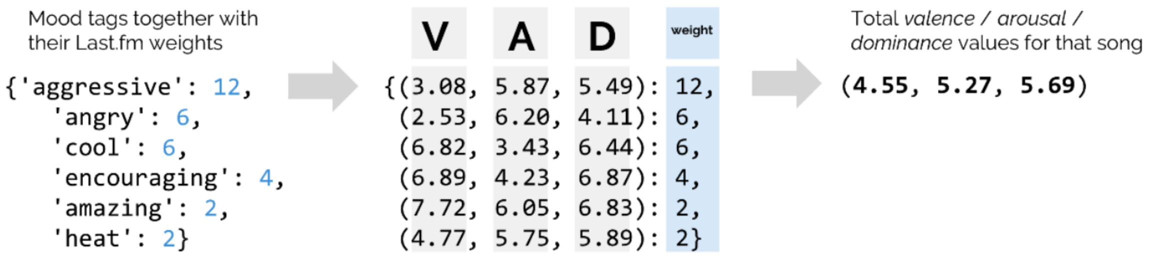

Figure 3.3: Example weighted average calculation for valence, arousal and dominance of the song ”Till I Collapse”, by Eminem [9]

4. **Mapping stage: Mood labels embedding into VAD space**

   The next step is to map these mood labels into the VAD space. For this purpose, Warriner et al. wordlist [48], which will be referenced hereinafter as the VAD list, has been used. This list was constructed through the recruitment of native English speakers through Amazon Mechanical Turk, who provided 1,085,998 affective ratings of 13,915 lemmas across all three VAD dimensions. A total of 88.1% of the mood tags from the MuSe dataset matched the VAD list.

   After filtering the initial 96,499 songs to remove those with at least one tag that matched the VAD list, 90,408 songs (93.7%) remained. Since multiple mood tags are typically used to define a song, the weighted average for each dimension of each word is calculated separately. The weights are obtained from the Last.fm API, which provides scores for each tag [see Figure 3.3]. The higher the Last.fm weight, the higher the relevance of a tag.

5. **Metadata stage: Further metadata incorporation via the Spotify API**

   Finally, after collecting the emotional metadata the Spotify ID is added. This enables the retrieval of additional information from the Spotify API [30]; however, it was not found for all songs, but for 61,484 songs.

It is of the utmost importance to have a clear understanding of the creation of this dataset, since it is the MuSe dataset which is used in this project. The final MuSe dataset comprises a number of key elements, including basic metadata such as artist, title, and genre, as well as mood labels, the three affective dimensions, and the Spotify ID. In MoodRoot, however, the lyrics of each song are also added, merging the MuSe dataset with the Genius Song Lyrics dataset [56] that contains the lyrics of 5,134,856 songs. Following this merge, the resulting dataset comprises 29,130 songs. Finally, when removing those that have missing Spotify IDs, 26,826 songs remain.

3.4 Music Recommendation Systems

Music recommendation systems are algorithms that suggest songs, artists, or playlists to users based on their preferences and listening history. There are a few key approaches to music recommendation:

1. **Collaborative Filtering:** Predicts what a user might like based on the preferences of similar users [57]. It collects data on user behaviour, such as ratings, likes, and listening time, in order to determine which users have matching tastes. Users who have previously liked the same songs are considered similar, so the system can recommend songs that other users with similar tastes are listening to.

2. **Content-based Filtering:** Uses songs metadata, such as descriptions, keywords, and tags to determine similarity [58]. When a user likes a song, the system looks for other songs with similar metadata to recommend.

3. **Contextual Approach:** Considers the user's current context, like their mood, activity, goal, or interactions with the music application [59]. The objective is to provide recommendations that fit the user's profile in terms of music taste, popularity preferences, familiarity with songs, and diversity.

4. **Emotion-Based Filtering:** Tailors music recommendations based on the user's emotional state or the emotional content of the music. Recommendation strategies could match the user's current emotional state or regulate a user's desired mood [60].

MoodRoot is an emotion-based system. The user states their actual emotional state and the desired emotional state to be achieved. The system does not take into account the user's preferences by acquiring their listening data; however, it does allow the user to choose their first song. In order to determine the suitability of subsequent songs, the system calculates the similarity between the song's current lyrics and those of the next song in the playlist, with the intention of adding songs with a similar emotional impact.

There are other recommendation systems that diverge slightly from the proposed model but ultimately pursue a similar objective. Moodify [61], for instance, employs a reinforcement learning approach to induce emotions in users by selecting music tracks that align with a target emotional state. The system assumes that emotions are determined by a sequence of recently played music tracks. Besides, the method proposed by Nakata and Nakanishi [62] uses VAD scores to extract emotional data from lyrics and recommend songs with similar emotional transitions.

An innovative approach in recent studies involves AI-driven music generation informed by biophysiological measures, specifically using galvanic skin response to gauge emotional arousal. In Duncan Williams' research [63], a machine learning algorithm dynamically generates music in real-time based on a perceptual similarity model, demonstrating the potential for AI to create personalised, emotionally congruent music that effectively induces emotional states to reduce anxiety and stress while addressing the problem of familiarity and repetition found in traditional music therapy.

Similarly, another system proposed in the 2020 IEEE International Conference on Consumer Electronics [64] employs EEG signals and Convolutional Neural Networks to recommend music that alleviates anxiety and depression. This system uses a commercialised EEG Bluetooth headset to acquire brain wave data and a CNN for detailed classification analysis, allowing for precise mood-based music recommendations. Both approaches showcase the promise of AI in developing tailored music experiences to enhance mental well-being, demonstrating the effectiveness of personalised, data-driven music therapy.

In conclusion, the evolution of music recommendation systems from collaborative and content-based filtering to emotion-driven and biophysiologically informed models represents a significant advancement in personalised music therapy. By analysing and comparing audio features or lyrics using the VAD model, or even integrating machine learning techniques that consider physiological signals, these innovative approaches hold substantial promise for enhancing mental well-being and managing conditions such as anxiety and depression. Finally, the integration of real-time data and personalised algorithms underscores the transformative potential of AI in creating more effective and engaging music therapy solutions.

CHAPTER 4

Architecture

This chapter examines the design phase of the project, as well as the implementation details pertaining to its architecture. Firstly, an overview of the project is presented, divided into several modules. This overview is intended to provide a general understanding of the project architecture. Secondly, the algorithm of the system is explained in greater detail. Finally, MoodRoot's workflow is presented in greater detail to facilitate a more thorough comprehension of the project's operational process.

4.1 General View

MoodRoot is a web application that aims to induce a specific emotion in the user through music. Upon entering MoodRoot, users must first log in to their Spotify account. This is followed by a redirect to the MoodRoot web page, where users input their initial and target emotions, select the first song for the playlist (based on the initial emotion) and specify the number of songs for the playlist. Once the playlist has been created, the link to it is displayed in the Spotify account. To describe MoodRoot's architecture, its components and connections must be understood. Figure 4.1 displays a general view of the system.

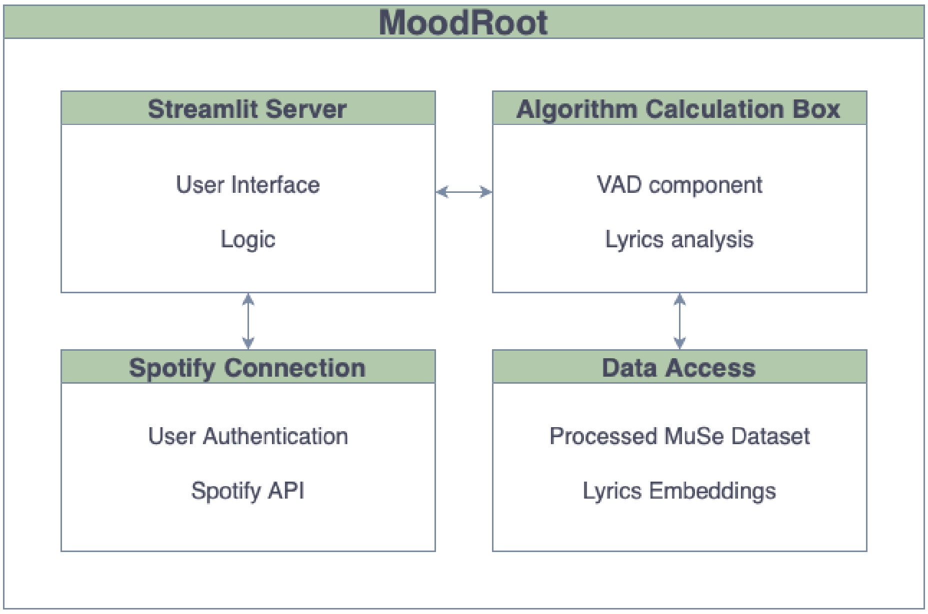

Figure 4.1: MoodRoot Architecture

4.1.1 Streamlit Server

The Streamlit server functions as the user interface for the system, and it also contains some logic. Initially, it facilitates the user's connection to their personal Spotify account. Subsequently, it gathers user inputs, including the initial emotional state, the first song, the target emotional state, and the desired number of songs for the playlist. This information is transmitted to the algorithm calculation module, which then returns the Spotify IDs of the songs comprising the playlist. Finally, the server interfaces with the Spotify API to create a playlist in the user's account, and provides the user with a link to the newly created playlist.

4.1.2 Algorithm Calculation Box

The algorithm calculation box contains the core logic for generating the playlist. It uses the VAD model [1] to calculate the emotional transition and select appropriate songs. It also considers similarities between song lyrics. Detailed insights into the algorithm are provided in Section 4.2.

This component receives user inputs from the Streamlit server, and accesses emotional data (VAD scores) and lyrics embeddings from the data storage component. Finally, it sends the Spotify IDs of the songs to the Streamlit server.

4.1.3 Data Access

This component provides access to the necessary data for the algorithm calculation box to create the route of songs. The data required includes the processed MuSe Dataset [9] and the lyrics embeddings file obtained using BERT [4]. The main dataset (porcessed MuSe dataset) stores data related to song attributes, VAD scores, genres, emotion seeds, etc.

4.1.4 Spotify Connection

This component provides integration with Spotify for user authentication and playlists creation. As explained in Section 2.6, OAuth 2.0 authentication flow [6] is handled for accessing the user's Spotify account. Besides, the Spotify API [30] is accessed to create playlists in the authenticated user's account.

For Spotify authentication, the user logs in to Spotify when entering the MoodRoot web application. Spotify provides an access token to the user for creating playlists. The Spotify API receives song IDs from the algorithm calculation box, extracts the song URIs, and creates the playlist in the user's Spotify account. Finally, it sends the playlist link back to the Streamlit server for user interaction.

4.2 Data Preparation and MoodRoot Algorithm

In order to comprehend the functioning of MoodRoot, it is necessary to elucidate the data preparation phase and the algorithm.

4.2.1 Data Preparation

As mentioned in Section 3.3, the MuSe dataset [9] is the foundation to create the route of songs. The dataset includes the following attributes:

- **lastfm_url:** Last.fm page of the song.
- **track:** Song title.
- **artist:** Artist(s) name(s).
- **seeds:** Initial keyword(s) that seeded the scraping of the song.

- **number_of_emotion_tags:** Number of words that contributed to the emotion score of the song.
- **valence_tags:** Pleasantness dimension of the song.
- **arousal_tags:** Intensity dimension of the song.
- **dominance_tags:** Control dimension of the song.
- **mbid:** MusicBrainz Identifier of the song.
- **spotify_id:** Spotify Identifier of the song.

Some of these attributes were not used, such as *lastfm_url* or *mbid*. However, others, such as *number_of_emotion_tags*, were instrumental in identifying the VAD values for individual tags among the seeds, like "happy", "sad", "angry", etc. To find the VAD values for a specific tag, it can be searched within the seeds, and then the rows where *number_of_emotion_tags* is equal to 1 can be identified (the VAD score will relate uniquely to that specific word). These VAD values (rescaled to a range of -1 and 1) are then used to determine the VAD point of the target emotion, which is one of the tags from the seeds. In general, the algorithm uses the VAD values to create a route of songs transitioning from an initial emotional state to a target emotion within a specified number of songs.

If the selected emotions to be shown in the web application as a dropdown menu are represented in the VAD space, the result can be viewed in Figure 4.2.

During the creation of MoodRoot, the lyrics of the songs were also incorporated by merging the MuSe dataset with the Genius Song Lyrics dataset [56]. This merge resulted in a dataset containing 29,130 songs, but when those with missing Spotify IDs were removed, 26,826 songs remained.

To analyse the lyrics, Natural Language Processing (NLP) was employed using BERT [4] to extract embeddings. The lyrics were preprocessed by removing repeated lines and unusual characters, as BERT-Base-Uncased [28] can only process up to 512 tokens, with any excess truncated. The integration of lyrics analysis through transformers enabled the representation of song lyrics as vectors, thereby enhancing the accuracy of recommendations.

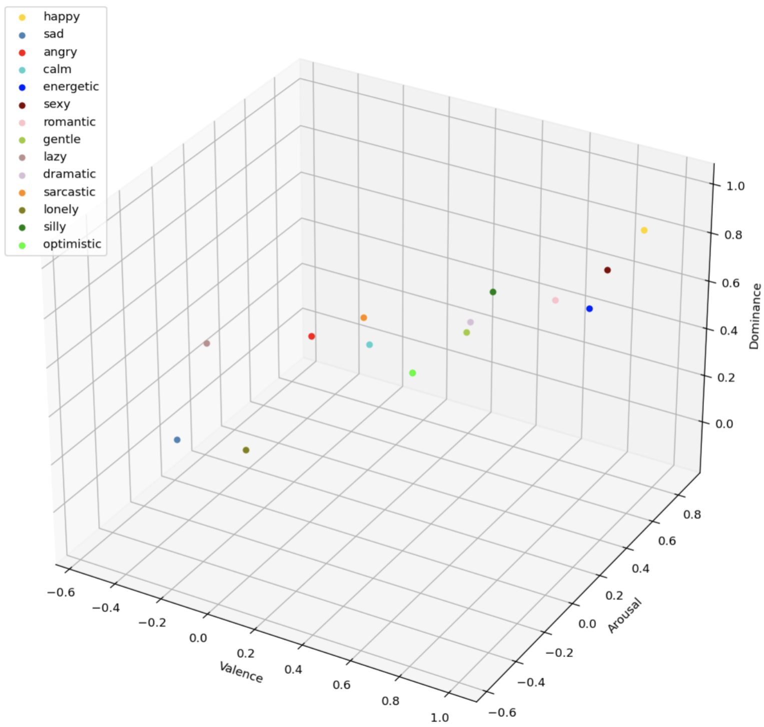

Figure 4.2: Emotions displayed in the MoodRoot app mapped in the VAD space

4.2.2 MoodRoot Algorithm

The algorithm establishes a sequence of steps designed to create a route of songs that induces a specific emotion. In the process of selecting subsequent tracks on the route, a preliminary filtration stage is executed, identifying songs proximate to the subsequent point on the route, established through prior calculations. The first three-quarters of the playlist are allocated for the transitional progression between emotions, while the remaining quarter is designated for the inclusion of songs corresponding to the final emotion.

Subsequently, the analysis turns to a comparison of the lyrics embeddings, with songs selected based on the cosine similarity metric between their embeddings and those of the preceding song. In particular, songs exhibiting a cosine similarity of at least 0.85 with the previous song's embedding are considered. In the event that no such songs meet this criterion, the song displaying the highest similarity is appended to the route. This approach ensures a coherent progression in the playlist, favouring songs that align closely with the lyrical themes of the preceding tracks.

Algorithm 1: Songs Route Generation

Data: Initial song ID $first\_song\_id$, Target emotion vector $target\_emotion$, Number of songs in the route num_songs_route, DataFrame of songs df
Result: DataFrame containing the route of songs

```
initial_emotion ← VAD values of the initial song;
;                                    /* Get initial emotion vector */
points ←
  calculate_points(initial_emotion, target_emotion, ⌈num_songs_route × 3/4⌉);
;                                    /* Calculate transition points */
route_df ← Empty DataFrame;
;                                    /* Initialize route DataFrame */
for i ← 0 to |points| do
    if i == 0 then
        route_df ← first song from df with first_song_id;
        ;                            /* Add the first song to the route */
    else
        closest_song_vad ← point[i];
        ;        /* Get the VAD values for the closest song to the
         current point */
        song_info ← search_song(closest_song_vad, route_df, df);
        ;                            /* Select a song near the closest song */
        if song_info is not None then
            route_df ← route_df ∪ song_info;
            ;                        /* Add the chosen song to the route */
        end
    end
end
for j ← 0 to num_songs_route × 1/4 do
    song_info ← search_song(target_emotion, route_df, df);
    ;                                /* Search for songs with target emotion */
    if song_info is not None then
        route_df ← route_df ∪ song_info;
        ;                            /* Add the song to the route */
    end
end
return route_df;
;                                    /* Return the final route */
```

Algorithm 2: Function to Search for a Song

```
Data: Emotion vector closest_song_vad, Route DataFrame route, DataFrame of
      songs df
Result: Song information closest to the given emotion vector
margin ← 0.05;
;                                            /* Set initial margin */
filtered_df ← Empty DataFrame;
;                                  /* Initialize filtered DataFrame */
while |filtered_df| == 0 margin ≤ 0.5 do
    filtered_df ← df filtered by closest_song_vad within margin;
    ;                                  /* Filter songs within margin */
    merged_df ← merge filtered_df with route;
    ;         /* Merge filtered songs with route to avoid repeats */
    filtered_df ← rows not in route;
    ;                           /* Remove songs already in the route */
    margin ← margin + 0.1;
    ;                                        /* Increment the margin */
end
if |filtered_df| == 0 then
    return None;
    ;                              /* Return None if no songs found */
end
previous_id ← last song's id from route;
;                                          /* Get previous song's ID */
song_info ← similar_lyrics(previous_id, filtered_df);
;                                 /* Find a song with similar lyrics */
return song_info;
;                                    /* Return the song information */
```

Algorithm 3: Function to Find a Song with Similar Lyrics

Data: Previous song's ID $previous\_id$, DataFrame of filtered songs $filtered\_df$, Array of songs embeddings $embeddings$

Result: Information about the song with the similar lyrics

```
filtered_embeddings ← Embeddings of songs in filtered_df;
;                        /* Extract embeddings of filtered songs */
previous_embedding ← Embedding of previous song;
;                               /* Get embedding of previous song */
similarities ← [];
;                      /* Initialize array to store similarities */
foreach embedding in filtered_embeddings do
    similarity ← cosine_similarity([previous_embedding], [embedding]);
    if similarity > 0.85 then
        similarities.append(similarity);
    end
end
if |similarities| == 0 then
    similar_lyrics_row ← row in filtered_df with maximum similarity;
    ;                     /* Choose song with maximum similarity */
end
else
    similar_lyrics_row ← random row in filtered_df with similarity > 0.85;
    ;              /* Choose random song with similarity > 0.85 */
end
return similar_lyrics_row;
;               /* Return the chosen song with similar lyrics */
```

4.3 Workflow

1. **User Interaction:** The user enters MoodRoot through the Streamlit server user interface and encounters the message: "Please, log in to Spotify" [see Figure 4.3]. When the user click on "Spotify" (it is a hyperlink) it redirects them to Spotify's login page.

2. **User Authentication:** The user logs in to their Spotify account [see Figure 4.4] and authorises MoodRoot to access their account. Spotify provides an access token, and the user is redirected back to the MoodRoot page.

3. **Input Collection:** Once back on the MoodRoot page, the user inputs their initial emotional state, first song, target emotional state, and the desired number of songs for the playlist [see Figure 4.5].

4. **Data Transmission:** The Streamlit server sends these inputs to the algorithm calculation box.

5. **Playlist Generation:** The algorithm calculation box processes these inputs and consults the data access component for VAD scores, song metadata, and lyrics embeddings. Based on this data, the algorithm calculates the emotional transition path and selects appropriate songs. The algorithm calculation box sends the Spotify IDs of the selected songs to the Streamlit server.

6. **Spotify Integration:** The Streamlit server receives the array of Spotify IDs from the algorithm calculation box. Using the access token, the Streamlit server requests the Spotify API to create a playlist in the authenticated user's Spotify account with the provided Spotify IDs. Once the playlist is created, the Spotify API returns the playlist link to the Streamlit server [see Figure 4.7].

7. **Display Playlist Link:** The Streamlit server receives the playlist link from the Spotify API and it displays the playlist link to the user, allowing them to access the playlist directly on Spotify [see Figure 4.8].

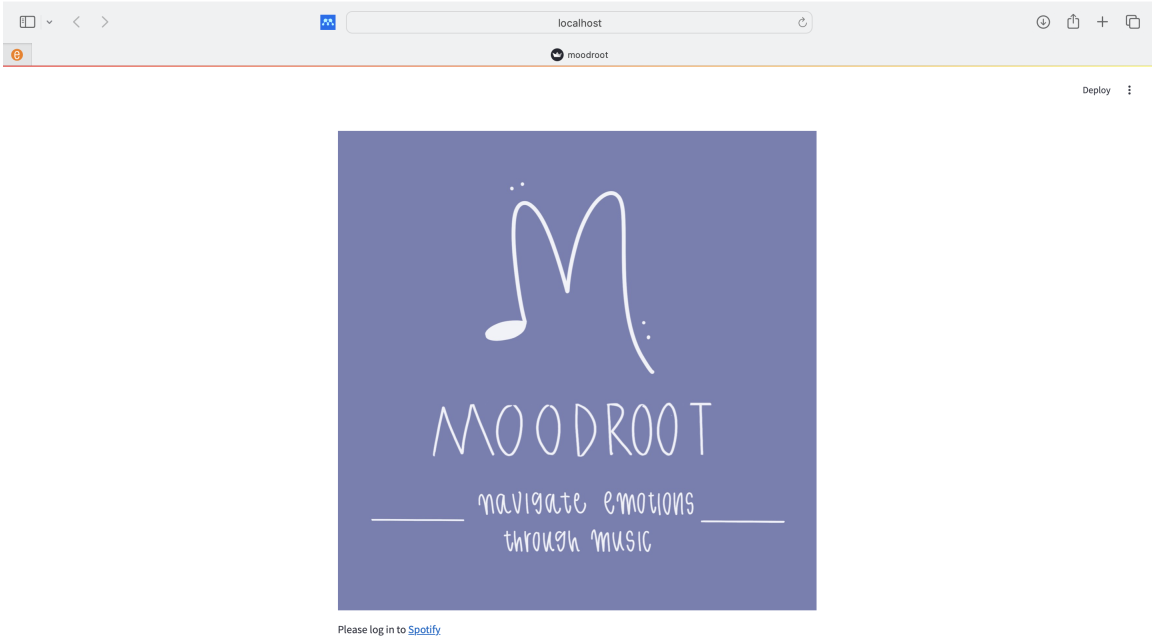

Figure 4.3: User's First Interaction with MoodRoot

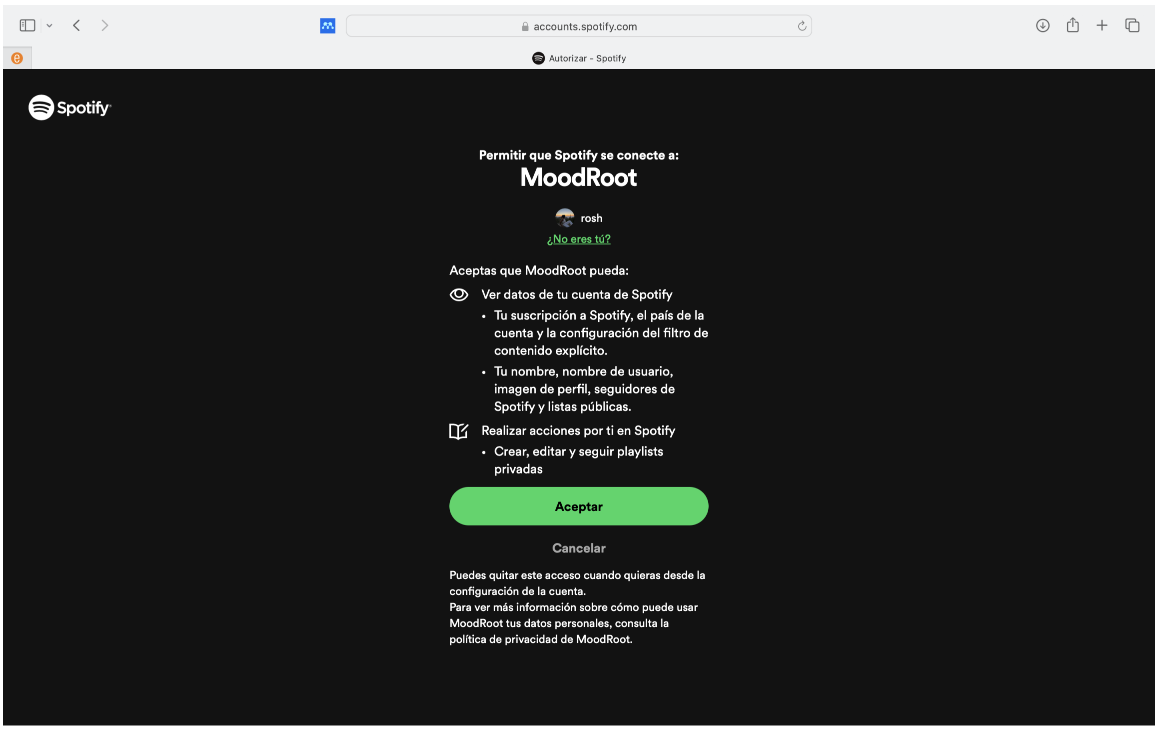

Figure 4.4: User Authentication via Spotify

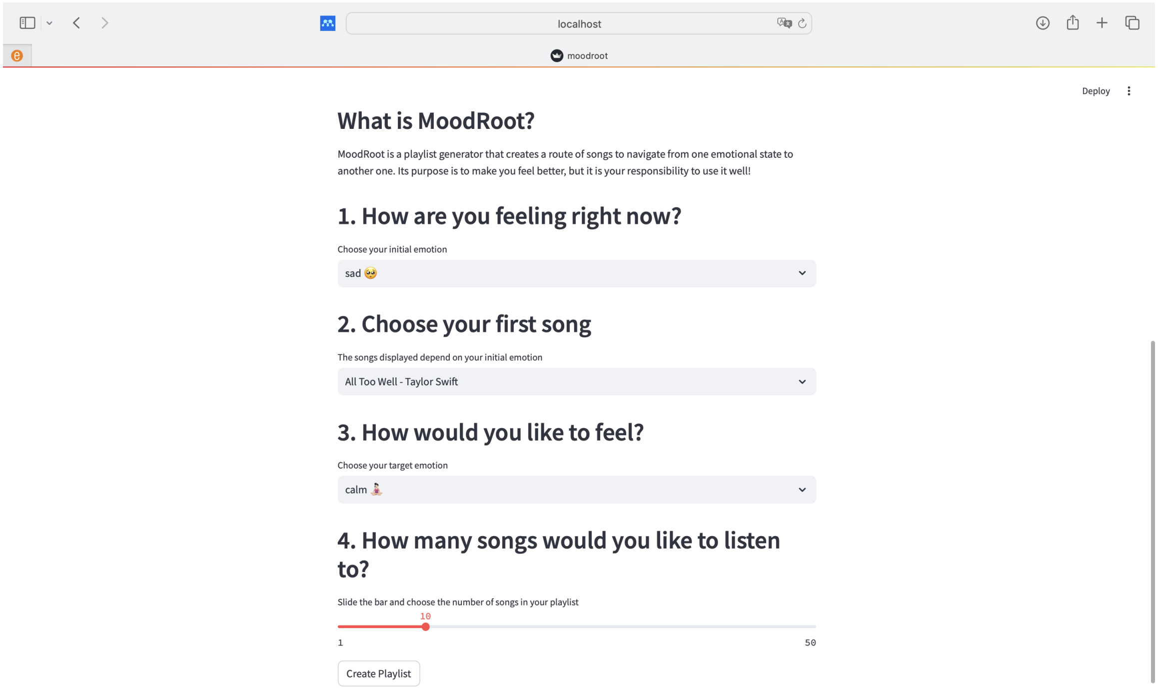

Figure 4.5: Input Collection

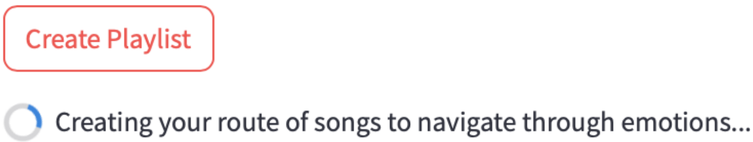

Figure 4.6: Playlist Charging

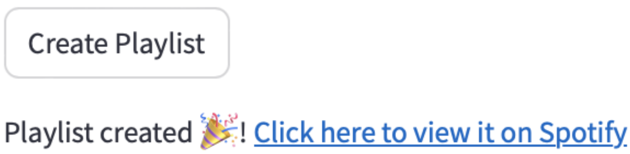

Figure 4.7: Link with Playlist Created

Figure 4.8: Playlist in the User's Spotify Account

CHAPTER 5

Evaluation

Once the MoodRoot system has been described, an evaluation process takes place to gather feedback and determine its effectiveness. This assessment delves into various aspects of the system's functionality, usability, and impact on users' experiences. The aim is to identify strengths, weaknesses, and areas for improvement, ensuring the system meets user needs and expectations effectively.

5.1 Methodology

In order to evaluate MoodRoot, a questionnaire was designed to evaluate latent variables, which are abstract variables directly connected to measurable variables. The values of these variables are quantified by the responses provided by the respondents in the questionnaire. Six latent variables were defined to evaluate the music recommendation system:

1. **Emotional Transition Efficiency:** Measures how effectively the playlist helps users transition from their initial emotional state to their target emotional state.

2. **Playlist Coherence:** Evaluates how well the songs in the playlist flow together in terms of mood, energy, and lyrical themes.

3. **User Satisfaction:** Gauges overall satisfaction with the MoodRoot experience, including the variety of songs and how well the system meets user expectations.

4. **Ease of Use:** Assesses the user interface and user experience of MoodRoot, including the ease with which users can navigate and create a playlist.

5. **Recommendation Intent:** Indicates how likely users are to recommend MoodRoot to others and their intention to use the system again in the future.

6. **Music Discovery:** Measures the extent to which users discover new music through MoodRoot, including finding new artists and adding new songs to their personal playlists.

Each variable comprises two or three statements, which the participants were asked to evaluate on a scale ranging from 1 (strongly disagree) to 10 (strongly agree).

5.2 Participants

Since the evaluation is oriented to assess the recommendation system that aims to induce a specific emotion, the testing was conducted with 45 participants who answered the questionnaire after using MoodRoot. The participants were asked to indicate their initial emotion, target emotion, the number of songs in their playlist, and to select the first song according to the initial emotion. They then listened to their newly created playlist and evaluated it using the questionnaire.

The participants in the study were students aged between 18 and 24 years, with a median age of 21. They were all native Spanish speakers. Of the 45 participants, two-thirds were female (30 participants). Additionally, the majority of users (73,3%) reported listening to music for several hours a day [see Figure 5.1].

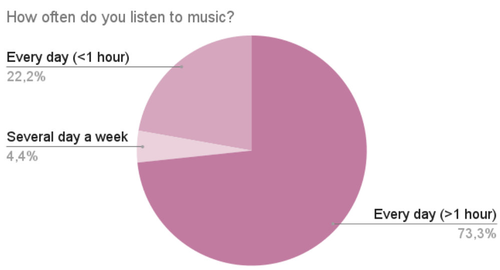

Figure 5.1: Music Listening Frequency of the Participants

The transitions tested by the participants where the following:

Calm to energetic (5)	Happy to calm (2)	Lazy to silly (1)
Calm to happy (4)	Calm to romantic (2)	Lazy to happy (1)
Dramatic to calm (4)	Calm to silly (2)	Sexy to happy (1)
Lazy to energetic (4)	Calm to optimistic (1)	Happy to sexy (1)
Lazy to optimistic (3)	Optimistic to dramatic (1)	Dramatic to romantic (1)
Sad to energetic (2)	Energetic to dramatic (1)	Romantic to angry (1)
Dramatic to energetic (2)	Energetic to happy (1)	Angry to silly (1)
Energetic to calm (2)	Energetic to lazy (1)	Lonely to sad (1)

Not all transitions were tested, since the options for initial emotion and target emotion were: happy, sad, angry, calm, energetic, sexy, romantic, gentle, lazy, dramatic, sarcastic, lonely, silly, and optimistic.

Moreover, the number of songs in each playlist varied from 6 to 50. Most playlists were of 20 songs (16), but many others were of 10 (9), 15 (7) and 30 songs (5). Most users (75,6%) knew the first song they chose.

5.3 Questionnaire and Results

The questionnaire responded by the participants, along with the mean and standard deviation of the answers, is represented in Table 5.1. Furthermore, the mean of each latent variable can be represented in a radial graph, which allows for a visual representation of the evaluation of each variable [see Figure 5.2].

Apart from punctuating the statements in the questionnaire, participants provided comments after using MoodRoot. Overall, the feedback was positive, although there are some aspects that could be improved. Upon meticulous examination of each case and consideration of the remarks of the participants, the following conclusions can be drawn:

- Since no musical attributes are taken into account, various playlists encountered abrupt transitions, particularly from sad to energetic, energetic to happy, happy to calm, dramatic to romantic, lazy to energetic, romantic to angry, and calm to silly.

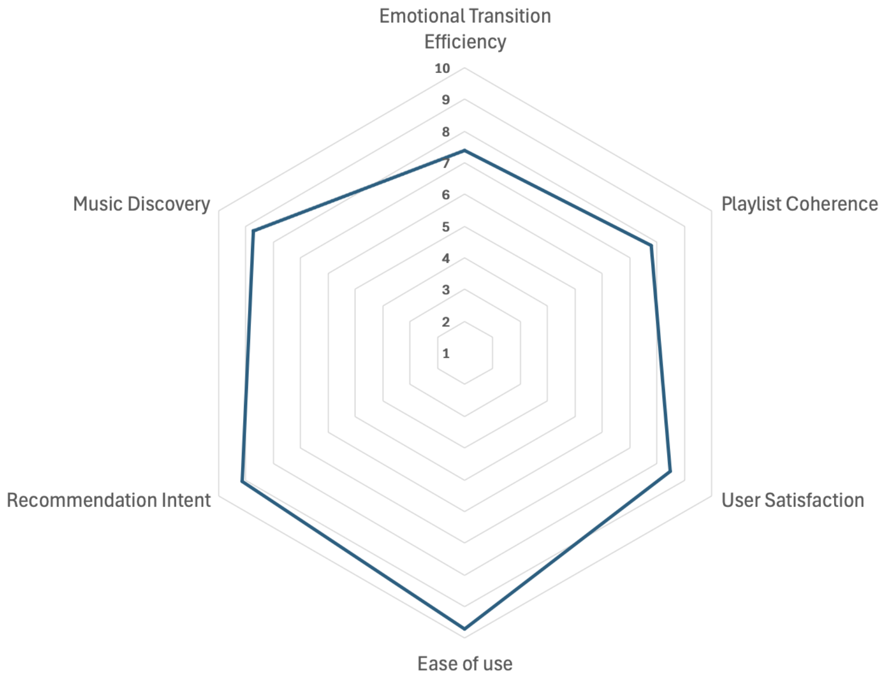

Figure 5.2: Latent Variables Evaluation

- Some participants stated that there were very few songs that matched the first emotion. This is because the algorithm reserves one fourth of the playlist for the target emotion, but only leaves one for the initial emotion. The songs situated in the middle may or may not correspond to the initial emotion, contingent upon the number of songs in the playlist and the proximity of the initial and target emotions. It is also important to consider that there may be other emotions between the initial and target emotions that could be the cause of the abrupt transitions.

- In several cases, the emotional transition was not as effective as it could have been due to the inclusion of songs from genres that users did not enjoy listening to. The implementation of a genre filter would likely prove more effective.

- With regard to the lyrics of songs, the punctuation provided by users was satisfactory. However, the similarities could be more apparent if a higher value for cosine similarity had been selected. When the algorithm was designed, the song with the highest cosine similarity was chosen, and frequently a song by the same artist was retrieved. This is why it was set to choose a song with at least 0.85 cosine similarity with the previous song randomly. This approach would result in a more diverse playlist, with a different one generated each time a user inputted data. It is also important to note that the participants were Spanish speakers, and the majority of the songs were in English.

- In regard to the length of the playlist, the feedback from the participants was not uniform. One participant who listened to 10 songs (from dramatic to calm) felt that the playlist was insufficiently long to facilitate an effective emotional transition. Conversely, another participant who listened to 12 songs (from calm to energetic) felt that the duration was excessive for their transition. Furthermore, a participant who listened to 33 songs (from energetic to calm) also indicated that the playlist was excessively lengthy.

- In general, the majority of participants expressed satisfaction with their experience of using MoodRoot, indicating a willingness to recommend the platform and a likelihood of future use. Additionally, they perceived MoodRoot as a potential means of improving well-being.
- The interface is designed in a user-friendly manner, and the playlist is created rapidly within the user's Spotify account.
- MoodRoot serves as a tool to discover new music, since most of the users added songs to their personal playlists after using MoodRoot.

Latent Variable	**Statement (rated from 1 to 10)**	**Mean**	**Standard Deviation**
Emotional Transition Efficiency	The playlist effectively helped me transition from my initial emotional state to my target emotional state.	7.62	1.81
	Each song in the playlist contributed to the overall emotional transition.	7.16	2.04
Playlist Coherence	The songs in the playlist flowed well together in terms of mood and energy.	7.96	1.85
	In general, there were no abrupt or jarring transitions between songs in the playlist.	7.71	2.22
	The songs lyrics maintained a coherent narrative throughout the playlist.	7.73	2.22
User Satisfaction	I am satisfied with the variety of songs in the playlist.	8.33	1.87
	Overall, I am pleased with the experience of using MoodRoot.	8.64	1.69
Ease of Use	MoodRoot's interface was easy to navigate and use.	9.76	0.57
	The steps to create the playlist were easy to follow.	9.71	0.59
	The playlist was created swiftly.	9.64	0.98
Recommendation Intent	I would recommend MoodRoot to my friends and other people.	9.2	1.12
	I am likely to use MoodRoot again in the future.	9.13	1.42
	I believe MoodRoot could help others feel better.	9.07	1.34
Music Discovery	I discovered new music thanks to MoodRoot.	9.51	0.89
	I added songs from the MoodRoot playlist to my personal track lists.	7.96	2.65

Table 5.1: MoodRoot Evaluation Questionnaire and Results

CHAPTER 6

Conclusions and Future Work

This chapter presents the principal findings and conclusions of this project, along with a discussion of the objectives met and suggestions for future work.

6.1 Conclusions

Music has always been a powerful tool for enhancing well-being, used throughout history to make people feel better. In the modern world, with the increasing prevalence of mental health issues, music can serve as a crucial pillar for emotional support.

MoodRoot has proven to be an effective tool for inducing emotions through personalised music playlists. Users have expressed high levels of satisfaction with the system, indicating its potential for broader application. Particularly, younger users, who typically listen to several hours of music a day, have found MoodRoot beneficial for mood enhancement and music discovery. This underscores the system's relevance in today's music consumption habits and its potential to make a positive impact on users' emotional well-being.

On the one hand, the VAD (Valence-Arousal-Dominance) model has been instrumental in mapping emotions to songs, facilitating the emotional transitions users seek. While the model has generally been effective, certain nuances must be considered when transitioning from one emotional state to another. Additionally, incorporating more musical aspects, such as tempo, key, and genre, could further enhance the emotional impact of the playlists.

On the other hand, lyrics analysis was successful, but the degree of similarity could be improved. The chosen similarity level allows for different playlists with the same user inputs, avoiding immediate repetition of artists. Future iterations should incorporate multilingual support or emphasise instrumental and universally understood musical elements, as the majority of songs were in English and the participants were Spanish speakers, potentially influencing the efficacy of emotional transitions for certain users.

In essence, the incorporation of an emotional component into music recommendation systems can markedly enhance their capacity to support users' emotional well-being. Future research should prioritise the refinement of the emotional mapping, the enhancement of the user interface, and the integration of more sophisticated musical analysis in order to provide an even more effective and satisfying user experience. With the technological resources currently available, systems such as MoodRoot can play a pivotal role in assisting users in feeling better and in discovering new music.

6.2 Achieved goals

Throughout the project, several key objectives were outlined and subsequently achieved, each contributing to the overall success and functionality of MoodRoot.

1. **Analyse the Impact of Music on Emotional Well-Being**

   One of the primary goals was to understand how music affects emotional well-being. Through extensive literature review and empirical research, the project has successfully highlighted the significant role music plays in influencing and enhancing emotional states. This foundational understanding informed the development of the MoodRoot system and its focus on emotional transitions through music.

2. **Develop an Algorithm that Creates Routes of Songs to Induce Emotions**

   The project aimed to create an algorithm that utilises the Valence-Arousal-Dominance (VAD) model alongside lyric embeddings to generate playlists that guide users towards a desired emotional state. This goal was successfully achieved. The algorithm effectively maps emotions to songs and creates a coherent emotional journey for the user. This was validated through user testing, where participants reported effective emotional transitions facilitated by the playlists generated.

3. **Design a User-Friendly Interface for the Algorithm**

   A crucial aspect of the project was to ensure that the powerful capabilities of the algorithm were accessible through a user-friendly interface. The MoodRoot interface, built using Streamlit, meets this objective by providing a simple and intuitive user experience. Users can easily input their initial and target emotions, select a starting song, and specify the desired playlist length, all through a straightforward interface.

4. **Test the System to Evaluate its Functionality and Effectiveness**

   Finally, the system was rigorously tested to evaluate its functionality and effectiveness. User testing involved participants interacting with MoodRoot, providing feedback on their experiences, and rating various aspects of the system. The results demonstrated that MoodRoot is not only functional but also effective in achieving its intended purpose of influencing emotional states through music. The feedback highlighted areas of success as well as opportunities for further refinement and improvement.

In summary, the project successfully achieved its initial goals, culminating in the development of a functional and effective tool that leverages music to enhance emotional well-being.

6.3 Future work

As MoodRoot continues to evolve, several areas have been identified for future development to enhance its functionality and user experience:

1. **Integration of Musical Attributes:** Incorporate analysis of musical features such as loudness, tempo, and genre into the algorithm. This would help in creating playlists with smoother transitions and a more cohesive emotional journey.

2. **Expansion of Dataset:** Expand the dataset to include a greater variety of songs, including those in different languages. This would enhance the diversity and relevance of playlists generated by MoodRoot, catering to a broader audience.

3. **Enhanced User Control:** Provide users with more control over playlist creation by allowing them to choose from a wider selection of songs as the first song. Additionally, introduce filters that enable users to exclude songs from genres they do not enjoy, ensuring a more personalised and enjoyable listening experience.

4. **Flexible Playlist Generation:** Develop algorithms that offer flexibility in playlist generation, allowing for non-linear transitions between emotions and avoiding abrupt shifts in mood. This could involve refining the algorithm to consider intermediary emotions and adjusting playlist lengths based on user preferences.

5. **Optimisation of Playlist Length:** Determine the optimal number of songs for facilitating effective emotional transitions. This could involve conducting further research to understand the relationship between playlist length and emotional impact, thereby refining MoodRoot's playlist generation algorithm.

6. **Preview and Customisation Options:** Implement features that allow users to preview playlists before downloading them and customise playlists by removing or replacing individual songs. This would empower users to tailor playlists to their specific preferences and ensure a more satisfactory listening experience.

7. **User Interface Refinements:** Continuously refine the user interface to enhance usability and intuitiveness. This includes optimising the interface for different devices and screen sizes, as well as incorporating user feedback to address any usability issues or pain points.

8. **Further Integration with Spotify:** Explore opportunities for deeper integration with the Spotify platform, such as allowing users to seamlessly sync MoodRoot-generated playlists with their Spotify accounts and access additional Spotify features within the MoodRoot interface.

9. **Reinforcement Learning:** Integrate reinforcement learning techniques to allow MoodRoot to adapt and improve over time based on user interactions and feedback. This would enable the system to learn from user preferences and behaviors, continuously refining its recommendations to better meet the needs of individual users.

10. **Continued Evaluation and Iteration:** Conduct regular evaluations and gather feedback from users to assess MoodRoot's effectiveness and identify areas for further improvement. Use this feedback to inform ongoing development and iteration of the platform, ensuring that it continues to meet the evolving needs and preferences of its users.

By focusing on these areas for future work, MoodRoot can continue to evolve and improve, ultimately providing users with a more immersive, personalised, and emotionally impactful music listening experience.

APPENDIX A

Impact of this project

This appendix reflects, in quantitative or qualitative terms, the potential social, economic and ethical impacts of the project. A brief analysis has been made to see how the project is aligned with the Sustainable Development Goals [65].

A.1 Social Impact

This project has a significant social impact by addressing mental health through the therapeutic potential of music. It aligns with Goal 3: Good Health and Well-Being, as it aims to contribute to the improvement of mental health, acknowledging that improvements in mental well-being can have cascading effects on overall health. Moreover, music therapy is inherently inclusive, transcending geographical and cultural boundaries, making it a universally accessible medium that benefits individuals regardless of their origins. By integrating music therapy into mental health care, the project promotes mental wellness across diverse communities, aligning with Goal 10: Reduced Inequalities.

A.2 Economic Impact

From an economic perspective, MoodRoot presents both opportunities and benefits. By offering a cost-effective and scalable solution for emotional well-being management, MoodRoot has the potential to reduce healthcare costs associated with traditional therapy and medication. Through its integration with platforms like Spotify, MoodRoot can also drive traffic and engagement, benefiting the music industry through increased streaming revenue and exposure for artists. Moreover, MoodRoot is a web application that involves data processing and machine learning algorithms (NLP) to improve mental health care practices. Therefore, it aligns with Goal 9: Industry, Innovation, and Infrastructure, contributing to economic growth and innovation.

A.3 Ethical Considerations

In navigating the ethical landscape, MoodRoot prioritises user privacy and transparency through the utilisation of Spotify's OAuth 2.0 authorization flow. Although users do not have direct access to the algorithm, their information is safeguarded through the implementation of robust privacy measures provided by Spotify's authentication protocols. This guarantees the security and protection of user data and preferences against unauthorised access or misuse. Transparency is maintained by providing users with clear information about how their data is utilized within the platform, including the processes involved in playlist generation. MoodRoot endeavours to ensure inclusivity and mitigate bias in its recommendations, taking into account factors such as cultural diversity and individual preferences in order to avoid reinforcing stereotypes or marginalising certain groups. By adhering to ethical principles and utilising industry-standard security measures, MoodRoot aims to foster trust and encourage the responsible use of technology in the pursuit of emotional well-being.

APPENDIX B

Economic budget

This appendix details an adequate budget to bring about the project. All the resources required to develop the current project are presented, divided into the following sections: hardware resources, software resources and human resources. Finally, a summary of the total estimated monetary cost is provided.

B.1 Hardware Resources

The only hardware requirement needed for this project is a laptop with the following characteristics:

- **Operating System:** macOS Monterey Version 12.7.1
- **CPU:** 2 GHz dual-core Intel Core i5
- **RAM:** 8 GB 1867 MHz LPDDR3
- **Storage:** 250 GB

The cost of acquiring this computer, and therefore the total cost of hardware components for this project, is 1500€.

B.2 Software Resources

Regarding the acquisition or licensing of software components, all software technologies used in the development of this project are licensed under open source licences and therefore don't have any licensing costs associated with their use. Consequently, the total cost of software resources is 0€.

B.3 Human Resources

In order to estimate the economic budget related to personal time, it is assumed that the entire development has been carried out by a single individual with a professional category of a biomedical engineering student. Given that this project has a workload equivalent to 12 ECTS credits, which correspond to 30 hours each, the total work time is 360 hours. If a cost of 11 euros per hour of work is estimated, the personnel cost is 3960€.

B.4 Total Resources Costs

Upon summation of all costs, the final estimated economic cost for the development of this project is 5560€.

APPENDIX C

Evaluation Graphs

This appendix contains the graphic representations from the questionnaires that were employed in the evaluation of the system.

C.1 General Questions

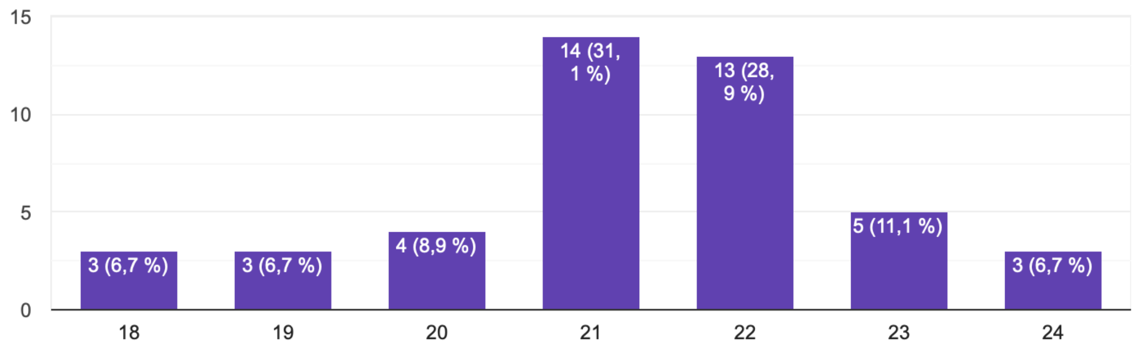

Figure C.1: How old are you?

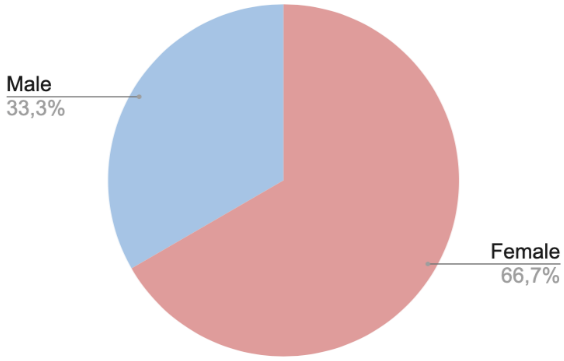

Figure C.2: How do you identify?

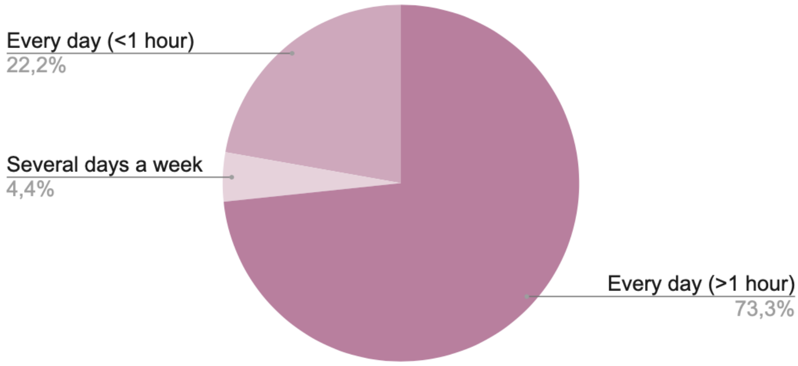

Figure C.3: How often do you listen to music?

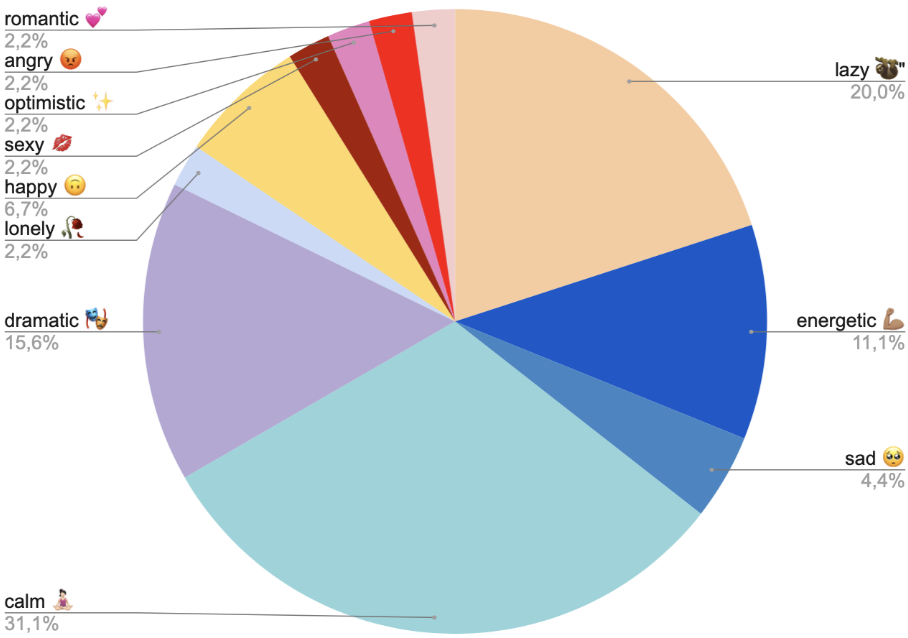

Figure C.4: What was the initial emotion in your playlist?

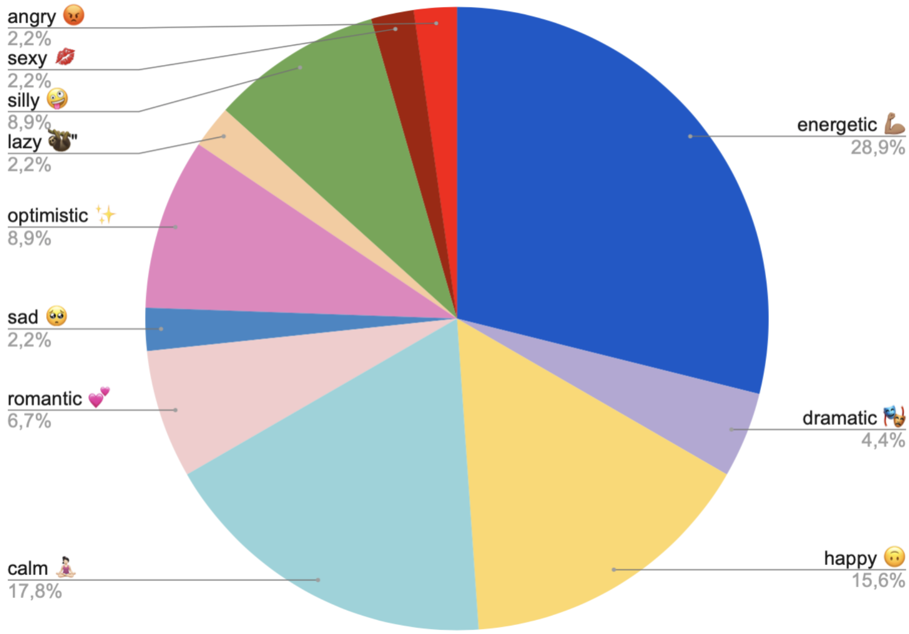

Figure C.5: What was the target emotion in your playlist?

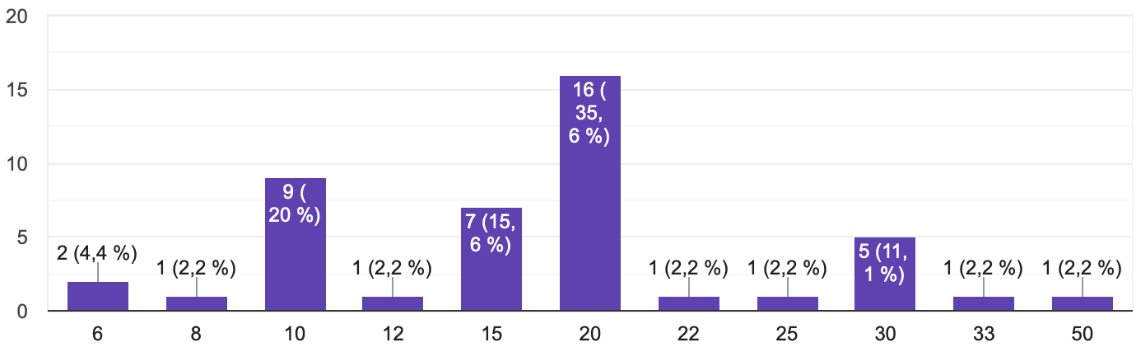

Figure C.6: How many songs did your playlist have?

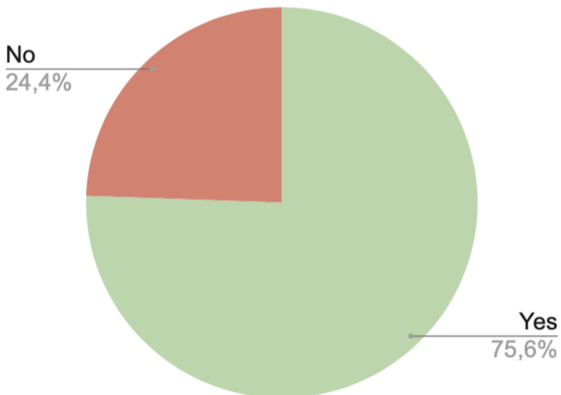

Figure C.7: Did you know the first song you chose?

C.2 Emotional Transition Efficiency

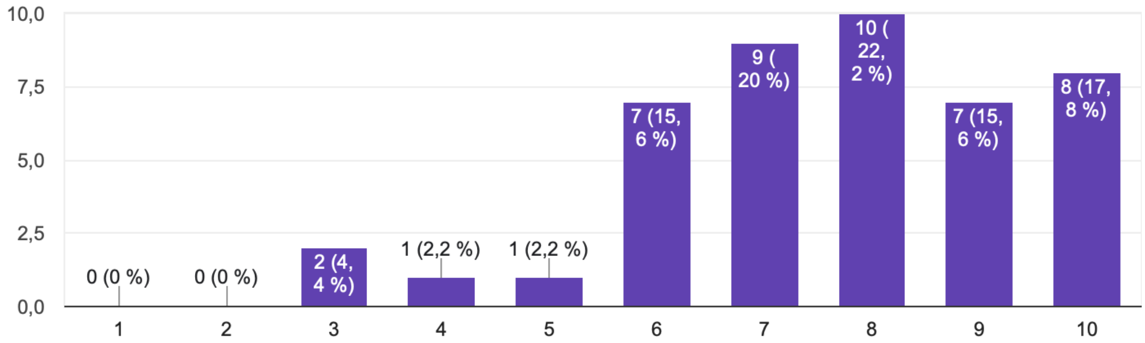

Figure C.8: The playlist effectively helped me transition from my initial emotional state to my target emotional state.

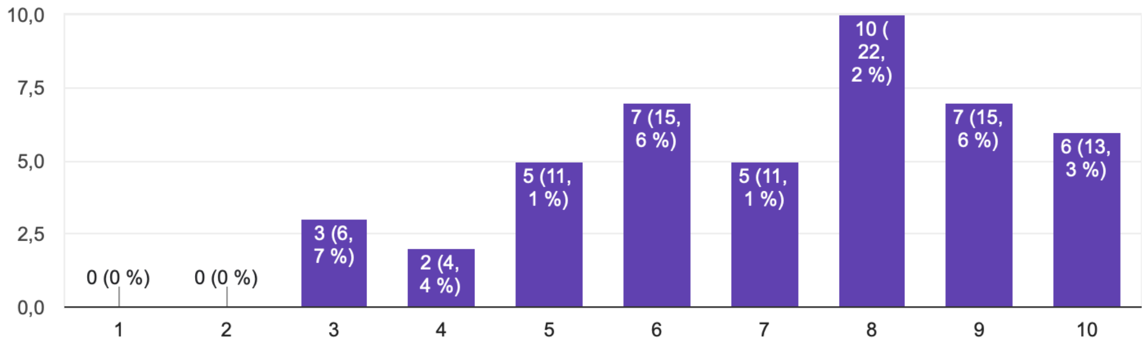

Figure C.9: Each song in the playlist contributed to the overall emotional transition.

C.3 Playlist Coherence

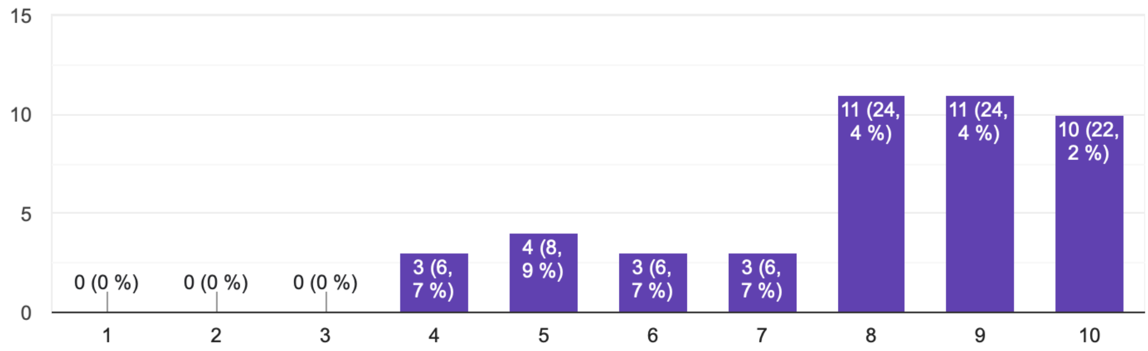

Figure C.10: The songs in the playlist flowed well together in terms of mood and energy.

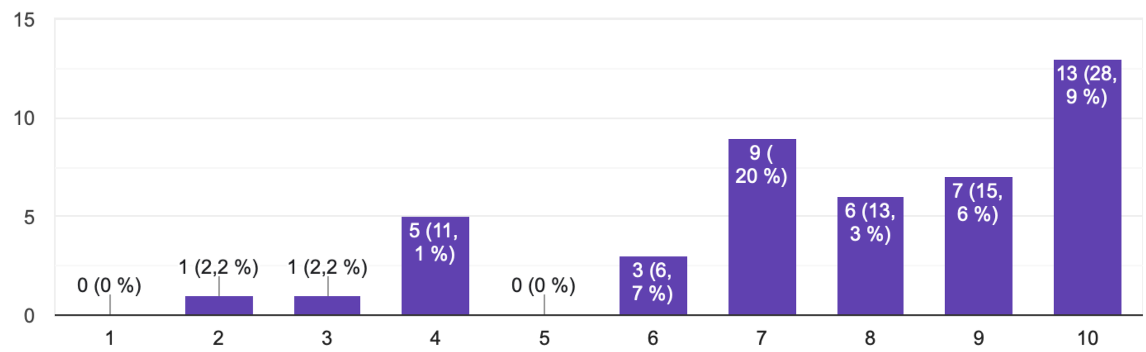

Figure C.11: In general, there were no abrupt or jarring transitions between songs in the playlist.

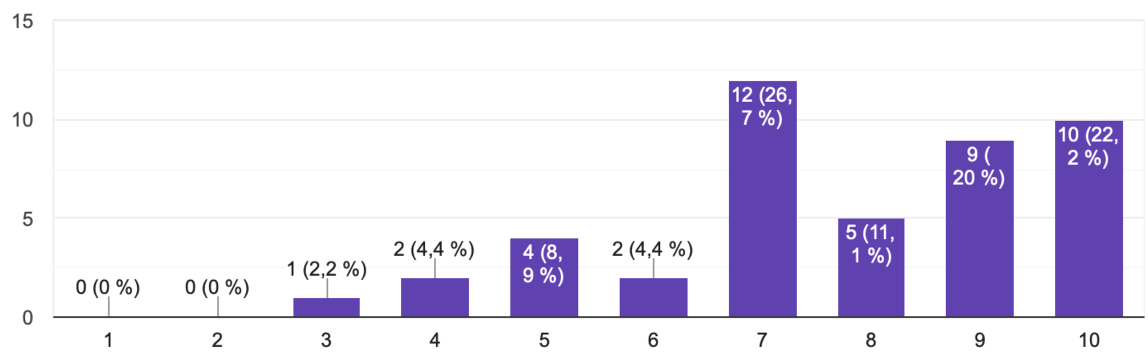

Figure C.12: The songs lyrics maintained a coherent narrative throughout the playlist.

C.4 User Satisfaction

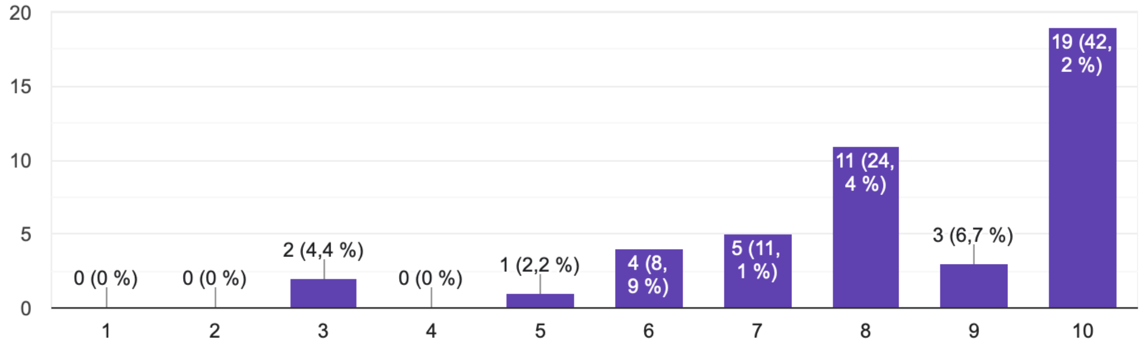

Figure C.13: I am satisfied with the variety of songs in the playlist.

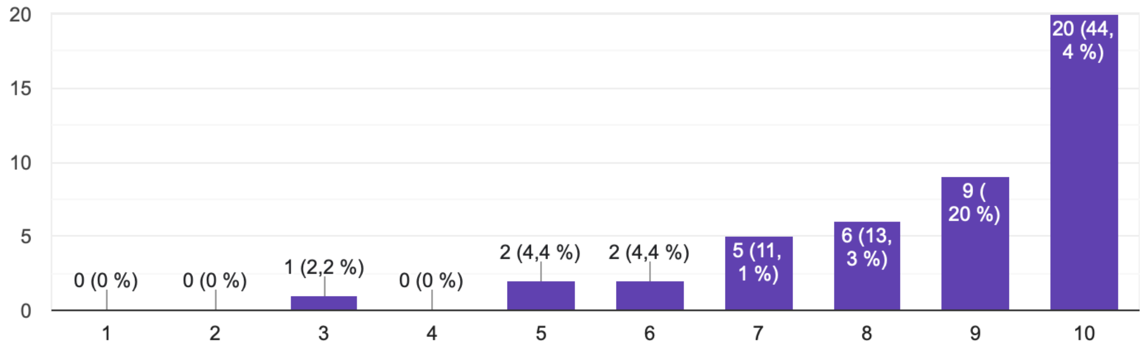

Figure C.14: Overall, I am pleased with the experience of using MoodRoot.

C.5 Ease of Use

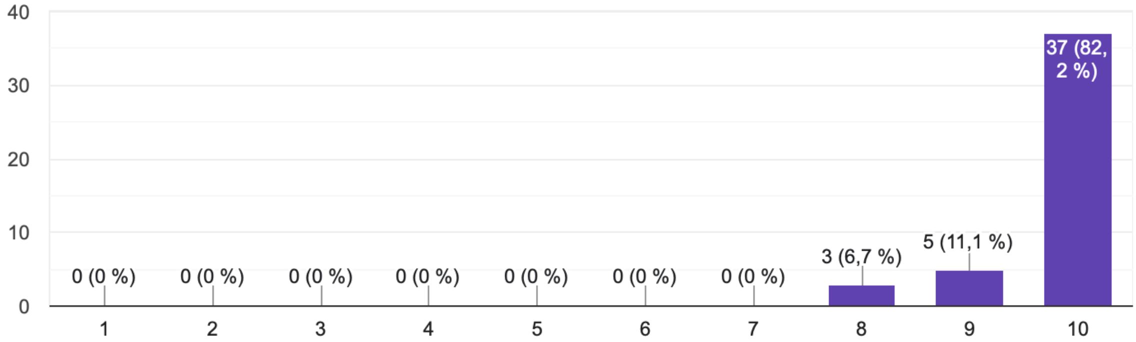

Figure C.15: MoodRoot's interface was easy to navigate and use.

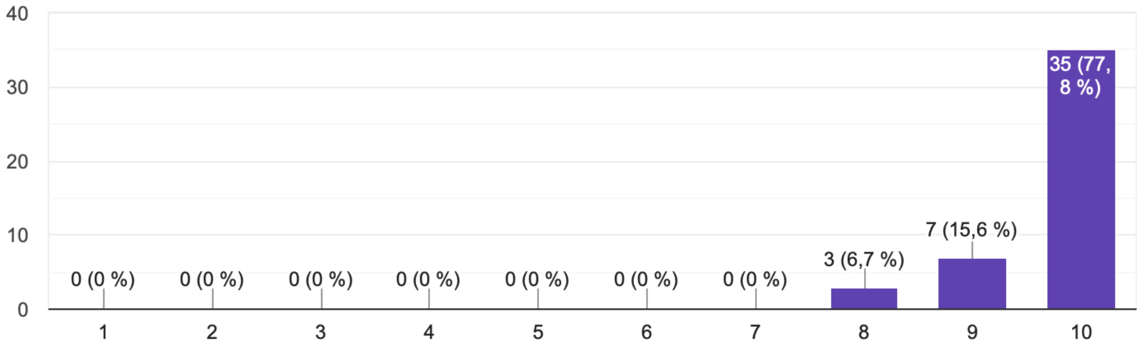

Figure C.16: The steps to create the playlist were easy to follow.

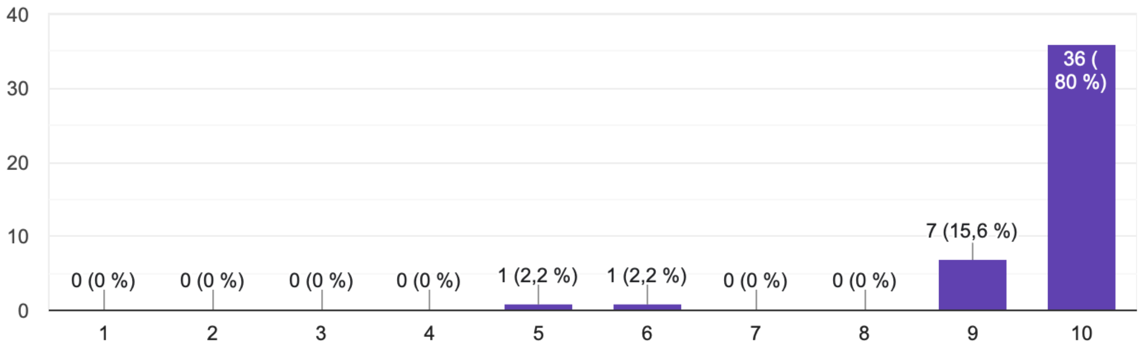

Figure C.17: The playlist was created swiftly.

C.6 Recommendation Intent

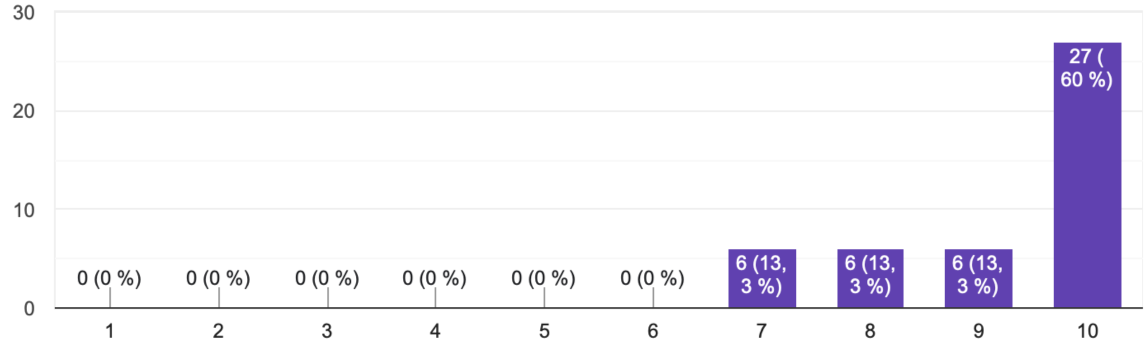

Figure C.18: I would recommend MoodRoot to my friends and other people.

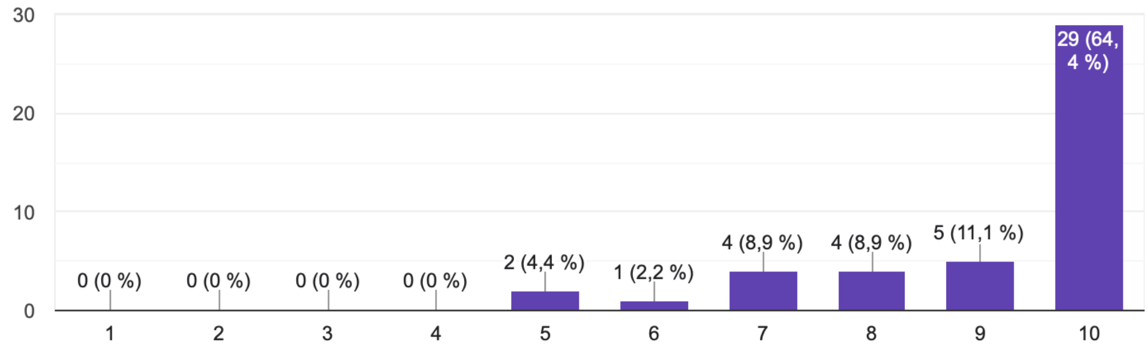

Figure C.19: I am likely to use MoodRoot again in the future.

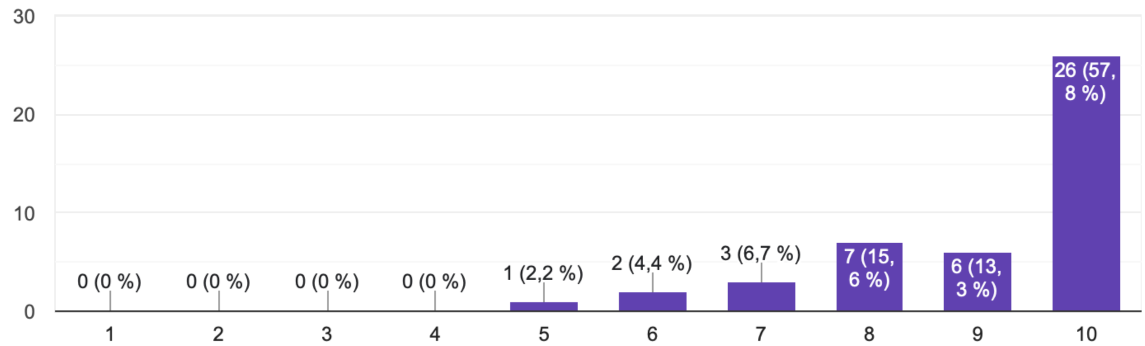

Figure C.20: I believe MoodRoot could help others feel better.

C.7 Music Discovery

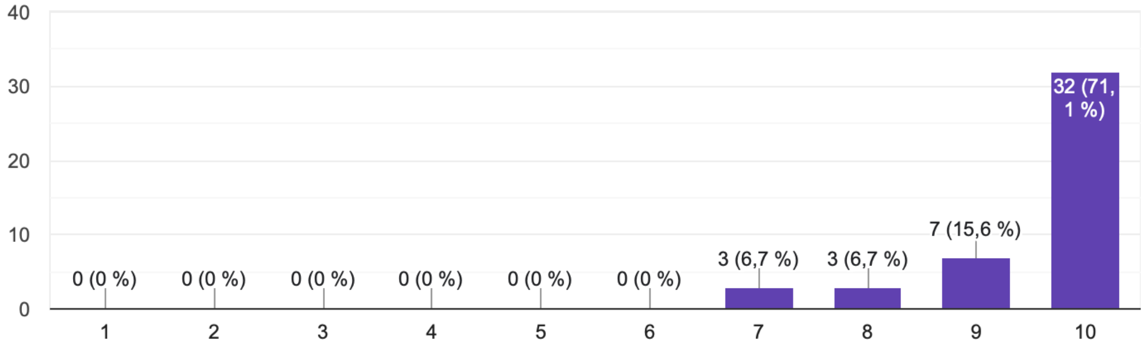

Figure C.21: I discovered new music thanks to MoodRoot.

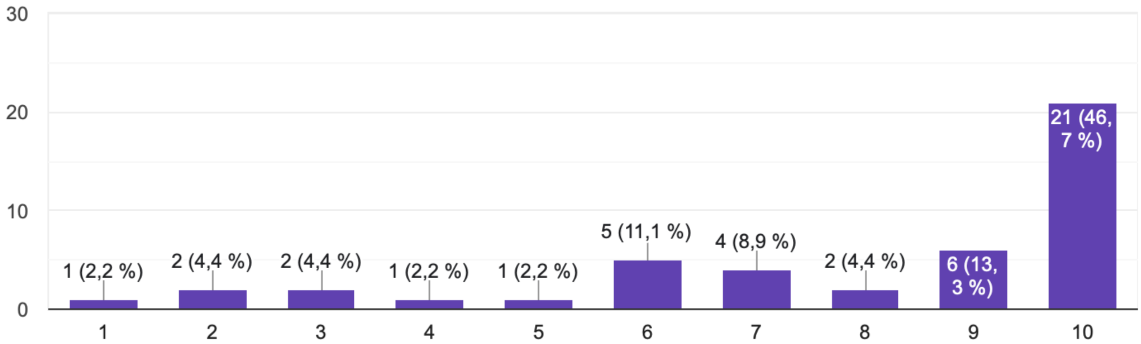

Figure C.22: I added songs from the MoodRoot playlist to my personal track lists.

Bibliography

[1] Albert Mehrabian. Pleasure-Arousal-Dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14:261–292, 1996.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *CoRR*, abs/1706.03762, 2017.

[3] Zhen Huang, Shiyi Xu, Minghao Hu, Xinyi Wang, Jinyan Qiu, Yongquan Fu, Yuncai Zhao, Yuxing Peng, and Changjian Wang. Recent Trends in Deep Learning Based Open-Domain Textual Question Answering Systems. *IEEE Access*, PP:1–1, 04 2020.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805, 2018.

[5] Spotify for Developers. Authorization. `https://developer.spotify.com/documentation/web-api/concepts/authorization`, 2024. [Online; accessed 26-May-2024].

[6] Dick Hardt. The OAuth 2.0 Authorization Framework. RFC 6749, October 2012. [Online; accessed 26-May-2024].

[7] Texas Tech University. Wheel of Emotions. `https://www.depts.ttu.edu/rise/PDFs/wheelofemotions.pdf`. [Online; accessed 31-May-2024].

[8] Oana Bălan, Gabriela Moise, Livia Petrescu, Alin Moldoveanu, Marius Leordeanu, and Florica Moldoveanu. Emotion classification based on biophysical signals and machine learning techniques. *Symmetry*, 12(1):21, 2019.

[9] Christopher Akiki and Manuel Burghardt. Toward a Musical Sentiment (MuSe) Dataset for Affective Distant Hearing. In *CHR*, pages 225–235, 2020. [Online; accessed 20-Feb-2024].

[10] Michael H Thaut. Music as therapy in early history. *Progress in brain research*, 217:143–158, 2015. Epub 2015 Jan 31.

[11] Margaret Ann Rorke. Music and the Wounded of World War II. *Journal of Music Therapy*, 33(3):189–207, 10 1996.

[12] Lavinia Rebecchini. Music, mental health, and immunity. *Brain, behavior, & immunity - health*, 18:100374, Dec 2021.

[13] World Health Organization. Suicide. `https://www.who.int/news-room/fact-sheets/detail/suicide`, august 2023. [Online; accessed 15-May-2024].

[14] World Health Organization et al. Mental health and COVID-19: Early evidence of the pandemic's impact. Technical report, World Health Organization, 2022.

[15] James Russell and Albert Mehrabian. Evidence for a Three-Factor Theory of Emotions. *Journal of Research in Personality*, 11:273–294, 09 1977.

[16] Python. `https://www.python.org/`, 2024. [Online; accessed 20-May-2024].

[17] pandas. `https://pandas.pydata.org/`, Apr 2024. [Online; accessed 20-May-2024].

[18] NumPy. `https://www.numpy.org/`, 2024. [Online; accessed 20-May-2024].

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[21] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[22] Streamlit. `https://streamlit.io/`, 2024. [Online; accessed 20-May-2024].

[23] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. 2018.

[24] Anis Koubaa. GPT-4 vs. GPT-3.5: A concise showdown. 2023.

[25] Wikipedia contributors. English Wikipedia. `https://en.wikipedia.org/wiki/English_Wikipedia`, April 2024. [Online; accessed 25-May-2024].

[26] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. *CoRR*, abs/1506.06724, 2015.

[27] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

[28] Hugging Face. BERT base model (uncased). `https://huggingface.co/google-bert/bert-base-uncased`, 2024. [Online; accessed 25-May-2024].

[29] Hugging Face. What is Feature Extraction? `https://huggingface.co/tasks/feature-extraction`. [Online; accessed 25-May-2024].

[30] Spotify for Developers. Spotify Web API. `https://developer.spotify.com/documentation/web-api`, 2024. [Online; accessed 25-May-2024].

[31] Project Jupyter. `https://jupyter.org`, 2024. [Online; accessed 20-May-2024].

[32] Ian Fette and Alexey Melnikov. The websocket protocol. Technical report, 2011.

[33] Jaakko Erkkilä, Marko Punkanen, Jörg Fachner, Esa Ala-Ruona, Inga Pöntiö, Mari Tervaniemi, Mauno Vanhala, and Christian Gold. Individual music therapy for depression: randomised controlled trial. *The British journal of psychiatry*, 199(2):132–139, 2011.

[34] Daisy Fancourt, Rosie Perkins, Sara Ascenso, Livia A Carvalho, Andrew Steptoe, and Aaron Williamon. Effects of group drumming interventions on anxiety, depression, social resilience and inflammatory immune response among mental health service users. *PloS one*, 11(3):e0151136, 2016.

[35] Karin Mössler, XiJing Chen, Tor Olav Heldal, and Christian Gold. Music therapy for people with schizophrenia and schizophrenia-like disorders. *Cochrane Database of Systematic Reviews*, (12), 2011.

[36] Chengyan Xu, Zixia He, Zhipeng Shen, Fei Huang, et al. Potential benefits of music therapy on stroke rehabilitation. *Oxidative medicine and cellular longevity*, 2022, 2022.

[37] Huan Liao, Guohui Jiang, and Xuefeng Wang. Music therapy as a non-pharmacological treatment for epilepsy. *Expert review of neurotherapeutics*, 15(9):993–1003, 2015.

[38] Josiane Lopes and Ivo Ilvan Keppers. Music-based therapy in rehabilitation of people with multiple sclerosis: a systematic review of clinical trials. *Arquivos de Neuro-Psiquiatria*, 79:527–535, 2021.

[39] C Pacchetti, R Aglieri, F Mancini, E Martignoni, and G Nappi. Active music therapy and Parkinson's disease: methods. *Functional Neurology*, 13(1):57–67, 1998.

[40] Joke Bradt, Cheryl Dileo, and Noah Potvin. Music for stress and anxiety reduction in coronary heart disease patients. *Cochrane Database of Systematic Reviews*, (12), 2013.

[41] Wendy L Magee, Imogen Clark, Jeanette Tamplin, and Joke Bradt. Music interventions for acquired brain injury. *Cochrane Database of Systematic Reviews*, (1), 2017.

[42] Mona Lisa Chanda and Daniel J Levitin. The neurochemistry of music. *Trends in cognitive sciences*, 17(4):179–193, 2013.

[43] Samata R Sharma and David Silbersweig. Setting the stage: neurobiological effects of music on the brain. 2018.

[44] Merriam-Webster. Emotion definition meaning. `https://www.merriam-webster.com/dictionary/emotions`, 2024. [Online; accessed 27-May-2024].

[45] P. Ekman, E. R. Sorenson, and W. V. Friesen. Pan-cultural elements in facial displays of emotion. *Science*, 164(3875):86–88, 1969.

[46] Robert Plutchik. A psychoevolutionary theory of emotions. *Social Science Information*, 21(4-5):529–553, 1982.

[47] James Russell. A Circumplex Model of Affect. *Journal of Personality and Social Psychology*, 39:1161–1178, 12 1980.

[48] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior research methods*, 45:1191–1207, 2013.

[49] Paul Lamere. Social tagging and music information retrieval. *Journal of new music research*, 37(2):101–114, 2008.

[50] Last.fm. Play music, find songs, and discover artists. `https://www.last.fm/`, 2024. [Online; accessed 27-May-2024].

[51] Allmusic.com. AllMusic: Music Search, Recommendations, Videos and Reviews. `https://www.allmusic.com/`, 2024. [Online; accessed 27-May-2024].

[52] Last.fm. Last.fm Music Discovery API. `https://www.last.fm/api`, 2024. [Online; accessed 28-May-2024].

[53] Xiao Hu, J Stephen Downie, and Andreas F Ehmann. Lyric text mining in music mood classification. *American music*, 183(5,049):2–209, 2009.

[54] Carlo Strapparava, Alessandro Valitutti, et al. WordNet-Affect: an Affective Extension of WordNet. In *Lrec*, volume 4, page 40. Lisbon, Portugal, 2004.

[55] George A Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[56] Nikhil Nayak. Genius Song Lyrics. `https://www.kaggle.com/datasets/carlosgdcj/genius-song-lyrics-with-language-information/data`, 2022. [Online; accessed 20-Mar-2024].

[57] Elena Shakirova. Collaborative filtering for music recommender system. In *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 548–550. IEEE, 2017.

[58] Kazuhiro Iwahama, Yoshinori Hijikata, and Shogo Nishida. Content-based filtering system for music data. In *2004 International Symposium on Applications and the Internet Workshops. 2004 Workshops.*, pages 480–487. IEEE, 2004.

[59] Marius Kaminskas and Francesco Ricci. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, 6(2-3):89–119, 2012.

[60] Mikhail Rumiantcev and Oleksiy Khriyenko. Emotion based music recommendation system. In *Proceedings of Conference of Open Innovations Association FRUCT*. Fruct Oy, 2020.

[61] Roberto De Prisco, Alfonso Guarino, Delfina Malandrino, and Rocco Zaccagnino. Induced Emotion-Based Music Recommendation through Reinforcement Learning. *Applied Sciences*, 12(21), 2022.

[62] Hiroki Nakata and Takafumi Nakanishi. Music Recommendation Method for Time-Series Emotions from Lyrics Using Valence-Arousal-Dominance Model. In *2022 12th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 443–448. IEEE, 2022.

[63] Duncan Williams, Victoria J. Hodge, and Chia-Yu Wu. On the use of AI for Generation of Functional Music to Improve Mental Health. *Frontiers in Artificial Intelligence*, 3, 2020.

[64] Shih-Hsiung Lee, Tzu-Yu Chen, Yu-Ting Hsien, and Lin-Roung Cao. A Music Recommendation System for Depression Therapy Based on EEG. In *2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, pages 1–2, 2020.

[65] United Nations. The 17 Goals - Sustainable Development. `https://sdgs.un.org/goals`. [Online; accessed 3-Jun-2024].