INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




A Machine Learning Approach to Affective Computing in the Metaverse

OSVALDO DANIEL MARTINS VIEIRA DA SILVA
Junho de 2025

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

A Machine Learning Approach to Affective Computing in the Metaverse

Osvaldo Daniel Martins Vieira da Silva

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Data Engineering

Advisor: Prof. Dr. Ivo Pereira

Porto, June 28, 2025

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

I have not plagiarised or applied any form of undue use of information or falsification of results along the process leading to its elaboration.

Therefore the work presented in this document is original and authored by me, having not previously been used for any other end.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P.PORTO.

ISEP, Porto, June 28, 2025

Abstract

Integrating Affective Computing in the Metaverse presents revolutionising opportunities in virtual interactions by enabling emotionally intelligent systems that can recognize, interpret and respond to human emotions. This study explores the development of a Machine Learning model for accurate and reliable affect recognition for future deployment in the Metaverse and likewise environments, while also exploring the potential of implementing such systems in order to enhance user engagement, user well-being and general interaction quality in virtual environments.

The state of the art in the interdisciplinary field of Affective Computing and its potential to be applied in the Metaverse was explored, while investigating the impacts this implementation could have and also taking into account the ethical concerns that rise from dealing with personal, thus sensitive, data and its interpretation.

A development pipeline based on the CRISP-DM methodology was implemented and the project focused on physiological datasets which were first evaluated independently, then merged and lastly augmented by SMOTE. A range of of models was tested across the different scenarios which in turn allowed for a comprehensive comparison of their effectiveness in affect recognition.

The evaluation showed that one of the SMOTE-augmented datasets classified with a RandomForest model achieved he best performance and that ensemble methods generally improved generalization and overall robustness specially when applied to the more feature rich datasets. The results highlight the importance of fine-tuning modeling strategies to each modality and also the feasibility of implementing affect-aware systems in immersive digital environments.

Keywords: Affective Computing, Metaverse, Emotion Recognition, Emotional Engagement

Resumo

A integração da Computação Afetiva no Metaverso apresenta oportunidades revolucionárias nas interações virtuais, permitindo sistemas emocionalmente inteligentes que podem reconhecer, interpretar e responder às emoções humanas. Este estudo explora o desenvolvimento de um modelo Machine Learning para o reconhecimento preciso e fiável de afectos para futura implementação no Metaverso e em ambientes semelhantes, explorando também o potencial de implementação de tais sistemas para melhorar o envolvimento do utilizador, o seu bem-estar e a qualidade geral da interação em ambientes virtuais.

Foi explorado o estado da arte no domínio interdisciplinar da Computação Afectiva e o seu potencial para ser aplicado no Metaverso, investigando ao mesmo tempo os impactos que esta implementação poderia ter e tendo também em conta as preocupações éticas que surgem quando se lida com dados pessoais, portanto sensíveis, e a sua interpretação.

Foi implementado um pipeline de desenvolvimento baseado na metodologia CRISP-DM e o projeto centrou-se em conjuntos de dados fisiológicos que foram primeiro avaliados de forma independente, depois fundidos e, por fim, aumentados pelo SMOTE. Foi testada uma gama de modelos nos diferentes cenários, o que, por sua vez, permitiu uma comparação abrangente da sua eficácia no reconhecimento de afectos.

A avaliação mostrou que um dos conjuntos de dados aumentados pelo SMOTE classificados com um modelo RandomForest obteve o melhor desempenho e que os métodos de conjunto melhoraram geralmente a generalização e a robustez global, especialmente quando aplicados aos conjuntos de dados mais ricos em caraterísticas. Os resultados destacam a importância de ajustar as estratégias de modelação a cada modalidade e também a viabilidade de implementar sistemas sensíveis aos afectos em ambientes digitais imersivos.

Contents

List of Figures

List of Tables

List of Acronyms

AC	Affective Computing.
AI	Artificial Intelligence.
AR	Augmented Reality.
CMA	Circumplex Model of Affects.
CRISP-DM	Cross Industry Standard Process for Data Mining.
ECG	Electrocardiogram.
FPR	False Positive Rate.
GSR	Galvanic Skin Response.
ML	Machine Learning.
MLP	Multilayer Perceptron.
PCA	Principal Component Analysis.
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses.
RBF	Radial Basis Function.
RFE	Recursive Feature Elimination.
ROC	Receiver Operating Characteristic.
SMOTE	Synthethic Minority Oversampling Technique.
SVM	Support Vector Machines.
TPR	True Positive Rate.
VR	Virtual Reality.
VREED	VR Eyes: Emotions Dataset.
WBS	Work Breakdown Structure.
XAI	Explainable AI.

Chapter 1

Introduction

The main premise of this dissertation is the development of Affective Computing (AC) models designed to be used in immersive digital environments such as the Metaverse, an evolving virtual realm that blends technologies like Virtual Reality (VR) and Augmented Reality (AR), transforming the way users interact in digital spaces and among themselves. But as great as the latter promises rich, interactive experiences, the emotional engagement of existing Metaverse platforms is still mediocre today. AC, understood as a field interested in recognizing, interpreting, and responding to human emotions, has the potential to enhance these experiences by enabling emotionally intelligent systems.

This research is based on recent literature showing the convergence between Metaverse technologies and AC as an area of innovation. Coutinho and Boukerche [1] lays out the foundational principles for integrating emotional intelligence into the Metaverse, arguing that the future success of such platforms may depend on their ability to emotionally engage users. Pervez, Shoukat, Usama, *et al.* [2] further propose emotionally intelligent avatars capable of adapting to users' emotional states in real time, which could improve social interaction and learning in virtual spaces.

The focus is on detecting affective states from physiological signals, using existing multimodal datasets, to explore challenges such as signal interpretation, and model reliability while the end goal is to support future implementations of emotionally responsive systems in immersive virtual environments.

1.1 Motivation

The rapid development of the Metaverse offers new opportunities for immersive and interactive user experiences. AC, which focuses on recognizing, interpreting, and responding to human emotions, could be important in shaping immersive and emotionally intelligent virtual experiences.

The Metaverse currently offers immersive and interactive experiences, but there is still untapped potential: the ability to capture and respond to human emotions. AC, which focuses on recognising, interpreting and responding to emotions, can be an essential tool for shaping these emotional experiences in the virtual world [1].

In this context, integrating AC into Metaverse platforms could enable avatars and systems that adjust to user emotions in real time, improving learning, social presence, and satisfaction [2]. However, such integration raises technical and ethical challenges, including the reliability of emotion recognition from physiological signals, data imbalance, user privacy, and the robustness of Artificial Intelligence (AI)-driven decision making.

This research is motivated by the vision of leveraging AC to improve user interaction and emotional well-being in virtual environments. The aim is to lay the foundation for future integration by exploring and validating emotion recognition techniques through physiological signals such as Galvanic Skin Response (GSR), Electrocardiogram (ECG), and eye-tracking data.

1.2 Problem

The main challenge is the development of an accurate and reliable model for affect recognition, while understanding how AC can be applied in the context of Metaverse to improve user engagement, interaction and emotional well-being.

Interpreting the problem involves identifying the current limitations in emotional interactions in virtual environments and how the lack of emotional recognition can negatively affect the user experience. By reviewing the state-of-the-art and analysing use cases, the project interprets the problem in several dimensions: Technological, how to correctly interpret human emotions using multimodal data (e.g. voice, gestures); Psychological, how to ensure that the emotional response generated by the system contributes to a positive and emotionally balanced experience; Interdisciplinary, how to integrate various technologies and knowledge (AI, VR/AR, psychology) in a coherent way, while respecting ethical and privacy concerns.

1.3 Objectives

This project's aim is to investigate how AC can be applied to Metaverse to improve user engagement and interaction. As such, investigating the current state of the art of AC, focusing on its main methods, technologies and applications is one of the project's objectives, followed by identifying use cases of AC in Metaverse, exploring its potential in sectors such as education, games, therapy, social interaction, among others, taking into account the technological and psychological implications.

Then, carrying out a systematic review of AC applied in the Metaverse, examining interdisciplinary research in AI, Human-Machine Interaction and Emotional Psychology to understand existing advances and limitations. This, aims to, together with research and analysis of existing datasets on AC, while identifying gaps or limitations and comparing them with the development needs of immersive and emotionally intelligent virtual environments, allow the building of a model that accurately interprets emotional responses in Metaverse environments, using multimodal data such as the aforementioned GSR, ECG, and eye-tracking data.

Moving along to the implementation, the study aims to develop a Machine Learning (ML) model that may be integrated into Metaverse environments, combining emotional recognition and AI-based responses to create more immersive and personalised user experiences.

1.4 Methodology

When it comes to choosing and applying the correct methodologies for this study it is important to understand that two different phases can easily be identified for this project and as such two different methodologies were employed. The research phase which will adopt the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA)

methodology to facilitate the identification and selection of relevant literature to support the theoretical foundation of the project. This allows the Implementation phase, which follows, to have a baseline guide, allowing, in turn, for the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology to be applied to develop and evaluate the ML model given how it is suited for data-centric tasks, such as the one this research focuses on.

The CRISP-DM process provides a structured approach for developing data-driven models, guiding the implementation through sequential, adaptable phases. This aligns with the experimental nature of the project by ensuring each stage is systematically addressed. Both methodologies are further detailed in the following chapters.

1.5 Structure of the Document

The remaining chapters of this document include:

- Chapter 2 goes through the process of the systematic review, detailing the use of the PRISMA methodology to conduct the gathering, analysis and selection of literature relevant to our study, showing the research questions which it aims to answer while also detailing the selected search terms, keywords and inclusion and exclusion requirements.
- Chapter 3 details the development methodology, with a brief overview of business and data understanding followed by an extensive explanation of the data modeling, going over the models selected, hyperparameter tuning, the overall modeling development and its "pipelines".
- Chapter 4 explores the results obtained on the previous chapter, evaluating the contents and drawing conclusions from the modeling and development carried out.
- Chapter 5 wraps up the research by presenting the main conclusions drawn from the whole process, highlighting the key findings, discussing the implications of the results for future applications, and outlining potential directions for future work.

Chapter 2

State of the Art

An overview of the current State of the Art and contextualization is provided, outlining the research methodology used to investigate AC in the Metaverse. The key findings are presented, followed by a detailed discussion of both common and unique themes, establishing a foundation of knowledge to inform future developments in the field.

2.1 Key Concepts

This section introduces and defines the two main concepts that this research explores: Affective Computing and the Metaverse. Understanding these concepts is fundamental for understanding the objectives, motivation and overall development and direction of the project, especially as they converge.

2.1.1 Affective Computing

Affective Computing refers to the study and/or development of computing systems that are capable of recognizing, interpreting, interpreting process and even simulate human affects.

The field is composed by a range of techniques and modalities, ranging from speech recognition to natural language processing or even facial expression detection. Recognizing emotional information requires extracting meaningful patterns from gathered data which is done by using ML techniques that process these different modalities to produce labels that would match how a human perceiver would also label it in the same situation. Labels can be something like "happy" or "positive" if a person makes an expression like slightly smiling, and the "sad" or "negative" for the opposite, given an expression such as furrowing the brow.

The concept was formally introduced by Rosalind Picard, who defined it as computing that “relates to, arises from, or deliberately influences emotions” [3].

2.1.2 Metaverse

The Metaverse is a collective virtual space formed by the convergence of physical environments and virtual worlds. It is characterized by immersive experiences, continuity of identity, and shared interaction among users within real-time 3D environments.

From a technical point of view, the Metaverse uses advanced 3D graphics, spatial audio, haptic feedback, and other technologies to simulate lifelike experiences. It is often framed as one of the possible next evolutions of the internet, transitioning from 2D interfaces to immersive digital spaces. The potential applications of the Metaverse are diverse, including education, remote work, gaming, and even healthcare.

The term itself has evolved over time, but in recent literature, the Metaverse is described as "a collective virtual space formed by the convergence of physical environments and virtual worlds" [4].

2.1.3 Affect vs. Emotion

For a better clearer understanding of our research, it is also important to be able to distinguish an "affect" from an "emotion".

Literature proposes that affect is a more fundamental, continuous state that underlies discrete emotions. Affect is often defined as the valence (pleasant-unpleasant) and arousal (activation-deactivation) experience while emotions are more complex and involve cognitive interpretation and labeling. Some also suggest that affects are more basic while emotions are socially and cognitively constructed. [5].

[6] says "Core affect is a neurophysiological state accessible to consciousness as a simple, nonreflective feeling (good–bad, energized–tired) that blends into emotions when interpreted in context." , while [7] adds to this by saying "The term 'affect' is often used as an umbrella term to include emotion, mood, and preference".

2.2 Research Methodology

The systematic review conducted employed the PRISMA methodology to consolidate the most recent and valuable scientific contributions to AC in the Metaverse. The main objective is to seek answers by contextualizing works related to the practical applications of AC and the Metaverse combined, examining aspects such as their integration and implementation, and the ethical challenges they present. The opportunities and obstacles that these factors create in the development and implementation of such technologies are also explored.

The methodology consists of a set of procedures that guarantees replication and comparative analysis of the knowledge gathered and documented. According to Page, McKenzie, Bossuyt, *et al.* [8], it highlights the importance of articulating very clear research questions that capture the central aspects of the pertinent domains. These questions can effectively be answered through a focus on important, major areas of research that help to give context to the retrieved articles through creation of a computational framework linking context, population, and relevant concepts. This yields a specific repository for the information, providing the initial pool of publications processed through the selection procedure. Selection involves inclusion or exclusion criteria that weigh evidence as to whether articles should be validated or rejected in formal review. This dynamic application of criteria results in including only the most relevant and highest quality resources.

2.2.1 Research Questions

In accordance with the objectives of the project, the proposal on exploring multimodalism in realizing AC tasks will take the next step towards the integration of emotional intelligence within the Metaverse, thus further enhancing the immersion within both virtual and real environments. In order to evaluate how well these domains are incorporated within the proposed systems, some research questions have been created to guide the application of the PRISMA methodology, as it is possible to see on table 2.1.

Table 2.1: Research Questions

Identifier	Research Question
RQ1	How can Affective Computing technologies be effectively implemented in the Metaverse to enable real-time recognition, interpretation, and response to user emotions?
RQ2	What are the impacts of integrating emotionally intelligent systems on user engagement, interaction quality, and emotional well-being within virtual environments?
RQ3	What ethical challenges arise from the use of Affective Computing in the Metaverse, and how can issues such as data privacy, emotional manipulation, and algorithmic bias be addressed to ensure responsible use?

First, the investigation regards AC technologies that are being studied and used in future contexts for real-time user emotion recognition, interpretation, and reaction. This study synthesizes papers on approaches and models for dynamic emotional interactions in virtual environments to enhance the immersive experience for the users.

Then, this study investigates the impact of integrating emotionally intelligent systems with user engagement, interaction quality, and emotional well-being within the Metaverse by investigating the impact of emotionally responsive systems on the way users interact with avatars and the environment, improving both the quality of interactions and the emotional experiences of users in these spaces.

Lastly, addressing ethical issues resulting from the use of AC in the Metaverse is crucial. As such, issues like data privacy, emotional manipulation, algorithmic bias, and others are identified and taken into account. The investigation aims to clarify the ethical aspects of using emotional data in virtual worlds and hence seek to provide an explanation and define how these technologies may be used responsibly, transparently, and fairly.

2.2.2 Scientific Repositories

The selection of scientific repositories is one of the first steps in conducting a systematic review. For this review, IEEE Xplore and ScienceDirect were chosen as the primary data repositories, as shown in Table 2.2 The parameters for each database have been carefully defined to minimize significant differences between them, and these will be discussed further in this section.

Table 2.2: Scientific Repositories

Repository	URL
IEEEXplore	`https://ieeexplore.ieee.org/`
Science Direct	`https://www.sciencedirect.com/`

2.2.3 Search Terms

Defining search terms is done to give context between the project objectives and the investigation. Providing different combinations of keywords correlated with specific domains within our scope dictates the retrieval of articles, enhancing the selection of scientific contributions [8]. Three domains were identified as being the principal areas to explore. "Metaverse" and its virtual worlds are there areas where we intend to implement our project, while "Affective Computing" is the interdisciplinary field intended to be explored and implemented.

Lastly, evaluating the effectiveness of our system is critical and will imply the use of metrics. For example, by the time spent by the user interacting with the system, active feedback from the user and, as it is pretended, by positive emotional recognition. All of these fit in the "User Engagement" domain. In order to conduct the search, keywords must be extracted from these domains, which, in fact, do not drift much apart from the actual domain designation, as it is possible to see in Table 2.3.

Table 2.3: Search Terms

Domain	**Keywords**
Metaverse Virtual Worlds	"Metaverse"
Affective Computing	"Affective Computing"
User Emotional Engagement	"User Engagement"

Table 2.4 reflects the final research query used, resulting from the concatenation of these keywords.

Table 2.4: Search Query

"Metaverse" AND "Affective Computing" AND "User Engagement"

2.2.4 Inclusion and Exclusion Requirements

One of the most important roles in the selection of studies to be included in the review is played by the inclusion and exclusion requirements set. These requirements can set rules to easily identify and exclude unfit articles or studies without even reaching the screening phase, while further along the PRISMA process these requirements speed up the decision-making in the screening of each article. Tables 2.5 and 2.6 list these requirements, some of which can be observed in Figure 2.1, detailing the full PRISMA process.

Table 2.5: Inclusion Requirements

Identifier	Inclusion Requirement
IR1	The article is part of peer-reviewed journal articles, conferences, papers or books
IR2	The article belongs to the interdisciplinary field of Affective Computing
IR3	The article's scope is relevant to the study
IR4	The article has open access

Table 2.6: Exclusion Requirements

Identifier	Exclusion Requirement
ER1	The article is not written in English
ER2	The article is older than 5 years
ER3	The article is not from a peer-reviewed source
ER4	The article's focus is out of the study's scope
ER5	The article does not have the necessary regard for the ethic and responsibility challenges

2.2.5 Publications Extraction

As mentioned before, the methodology employed in the Research stage of this project is the PRISMA methodology, which divides the publications into three phases: Identification, Screening and Included, as it is visible in Figure 2.1.

In the initial Identification phase it is possible to see the total amount of articles collected from the repositories which are also identified. To enhance this initial search, advanced search criteria were set for the scientific repositories to retrieve recent (last 5 years) and language-appropriate articles, guaranteeing that these articles do not violate the set exclusion requirements, specifically ER1 and ER2. After gathering these articles a duplication check was performed and only 3 were identified as such, and as such the Identification phase resulted in 421 articles selected.

Moving on to the Screening phase, this stage was conducted through two different iterations with different granularity each. Starting off, the titles of the articles were screened and those seemed not fit were not included for the second iteration, which analyzed the abstract of the remaining articles, giving a better, more detailed description of what to expect in each article screened, allowing for a more informed decision-making. After these iterations 42 articles were selected for retrieval, 1 of which was behind a paywall, thus not being available.

The remaining 41 articles were then fully read which made it possible to select the articles fit for inclusion, given the defined inclusion and exclusion criteria, which justify why the final number of articles selected was 13. These 13 articles are the final selected articles where information was sought to answer our research questions.

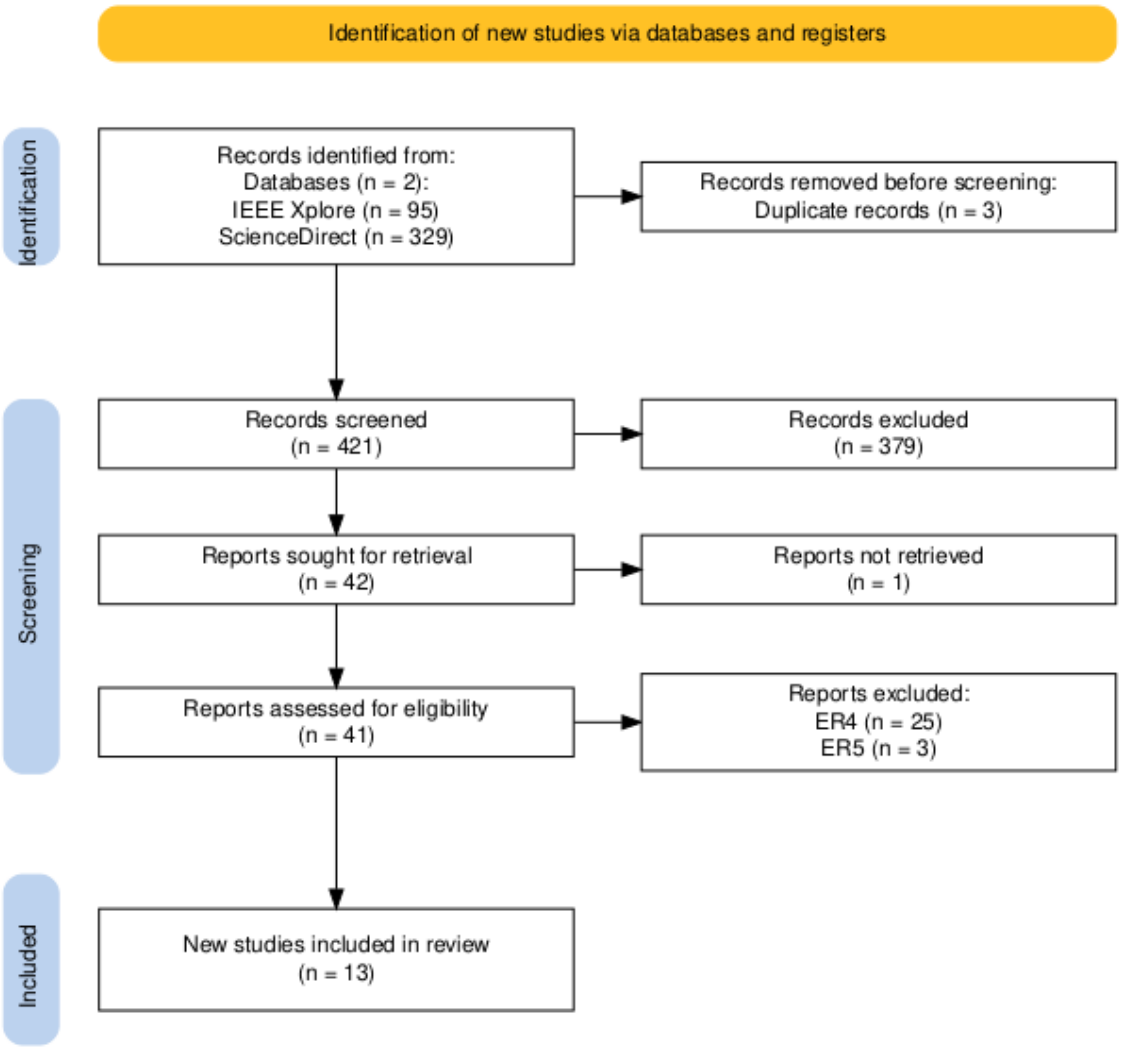


Figure 2.1: Diagram of the finalized PRISMA process

2.3 Results

This section outlines the articles obtained through the application of the proposed methodology. It aims to document the retrieved publications and explain their relevance in addressing the research questions effectively.

2.3.1 Implementation of Affective Computing technologies to enable real-time recognition, interpretation, and response to user emotions

The implementation of AC technologies in the Metaverse has to do with recognition, interpretation and the response of user emotions in real-time. It requires complex emotion recognition and immersive technologies that are implemented in virtual environments.

It involves the deployment of emotion recognition techniques from the integration of physiological, behavioral, and contextual data for better performance. This multimodal approach enriches the possibility of recognition success and its technology involves, for example, deep-learning-based face emotion detection to analyze the facial expression of a person to find

out its emotion state [9]. Such a combination can infer the emotion of the user through real-time processing of physiological signals. However, measuring heart rate or the galvanic skin response from the user will significantly alter the user's perception. This means, in addition to a very quick recognition, that the the immersive capabilities of AR and VR that enhance real-world experiences will also capture real moments outside the artificial nature of these very same technologies [10]. They create a very real engaging environment that can capture the emotional response of the system and then react to it responding interactively based on the user's sensory feedback such as those immersive environments in VR which may be changed in real-time to adjust according to the emotional state of the user, thus making the experience more empathetic [10].

The adoption of this Artificial Intelligence in a system can thus provide the Metaverse environments dynamic, interpreted emotions [2]. Further, they can use the continuous emotional data stream as indicators to change their interaction or environment in adaptation between context and emotion [2]. Such adaptive quality makes the virtual experience interesting but emotionally tuned with the user thus improving personalization and emotionality [11].

These above implementations depend somewhat on the systems being able to respond in real-time and adapt to the user's emotional changes instantly [12]. Such immediate responding systems make interactions in the Metaverse appear to be more natural, and thus the experiences become even more immersive [12]. Such naturalness is vital in keeping the user absorbed and satisfied as it makes communication virtually seamless [2][11].

AC can thus, with the help of advanced technologies and methodologies, change the character of the Metaverse into an intensely interactive and responsive emotional space [2][11]. This platform will make not just the user engagement better through the improved understanding of emotional and interactive means, but it will also prepare the ground for future digital-human interactions to become more complex and nuanced. A comprehensive approach to AC in the Metaverse sounds promising regarding the develop of a more emotionally intelligent and interactive digital world [12].

2.3.2 Impact of integrating emotionally intelligent systems on user engagement, interaction quality, and emotional well-being within virtual environments

The effects of emotional intelligence systems in virtual environments can impact engagement, the quality of the interaction, and the user's emotional well-being. These systems intend to analyze emotions in real time by physiological, behavioral, and contextual measures for recognition of emotions and have made a difference through the integration of the currently available AC technologies. The ability to dynamically adapt these environments contributes to deeply immersive and personalized experiences and is reformulating the ways users interact with digital spaces [13][14].

Emotionally intelligent systems are designed to excel in improving emotional well-being. They have the ability to detect stress, anxiety, or discomfort emotions and respond by introducing calming audiovisual stimuli, such as music or interactive elements. The use of virtual environments for therapeutic purposes, for example, depends on real-time feedback to regulate emotional arousal in ways unique to each users' needs, particularly for users experiencing anxiety or other emotional challenges. These systems not only create conditions for relaxing, but can also help in the long term by building emotional resilience [13][15]. In addition, these virtual environments provide a sense of emotional release and psychological

safety in a form of escapism, which has been shown to assist with real-life stressors [16]. Overall, these emotionally intelligent systems convert virtual environments from being super high-tech to include quite a bit of emotional support. This raises user engagement by producing spaces that are emotionally resonant, improves the quality of interaction through empathetic communication, and encourages emotional well-being by meeting and adjusting to psychological needs of users. This is the perfect place where technology co-relates with human-centered design and thus places the Metaverse within itself in terms of interaction and emotional resonance [14][15].

To a much greater extent, that improves user engagement. It is dependent, however, on emotionally intelligent systems interpreting the emotional status of a user and being able to respond in accordance. For example, multimodal emotion-recognizing systems use several physiological signals, such as heart rate variability, skin conductance, and facial expressions, and dynamically adapt virtual environments to these inputs [15][16]. Ambient changes in lighting, soundscape, or even the agent's responsiveness enhance the sense of presence and connection for the user. Such personalized experiences drive up the amount of use interaction and increase engagement by making users feel understood and very often connected emotionally [14].

On the contrary, the quality of interactions goes up again by several degrees through these intelligent emotional systems. Virtual agents and avatars equipped with such systems will recognize slight cues of emotion, such as a tone of voice or micro-expressions, to provide a response that is both empathetic and authentic. Along with the aforementioned capability, trust is created in an otherwise productive interaction by increasing productivity because, through this ability, mutual trust can be established in a collaborative setup such as virtual workplaces or educational platforms. For instance, it has been shown that emotionally adaptive avatars reduce miscommunication and build rapport in teams within a virtual environment [2][10].

2.3.3 Ethical challenges, addressing data privacy, emotional manipulation and algorithmic bias and ensuring responsible use

The introduction of AC in the Metaverse might completely change the way individuals interact by building emotionally intelligent environments. However, it raises several ethical concerns, particularly regarding data privacy, emotional manipulation, pontential algorithmic bias and others. These need to be faced and addressed as challenges that must be overcome in order for this project to be developed responsibly.

As mentioned, one of the main concerns is related to data privacy. Emotional data such as physiological signals, facial cues, and voice tones can be connected in order to identify a user and as such pose a serious threat, as this is sensitive and prone to surveillance, unauthorized access, and misuse [2][17]. The joining of the Metaverse and AC creates a new context and, consequently, new vulnerabilities, as its immersive nature increases exposure. Encryption and secure data storage mechanisms for emotional data should be made available, but making sure they are only used for implementation [2]. Furthermore, it is important that the users perform informed consent processes to ensure transparency and accountability [2]. Included are privacy-preserving models, such as federated learning, which can limit centralized data storage and exposure [12].

On another front, the most intricate problem is emotional manipulation. Recognizing emotions in real-time allows the exploitation of users for commercial or political gain, which is

made worse by the absence of explicit consent mechanisms or transparency [2]. Restricting emotional data usage is crucial to prevent manipulative practices. Enhancing transparency by making the decision-making processes in AC systems understandable to the users is a plus and can be made possible by using Explainable AI (XAI) [2].

Another critical problem can betray emotion recognition systems: bias regarding race, gender, or cultural backgrounds which can result in unfair treatment or discrimination. These biases are mostly brought by the representations of the datasets and a lack of consideration for social contexts [9][12]. To address these issues, ensuring that the training datasets for this project's emotion recognition systems must be heterogeneous and representative is crucial [12]. Routine algorithmic audits should be undertaken to detect and rectify biases. Further, if the systems are to become culturally sensitive and fair, multidisciplinary teams, including ethicists and social scientists, should be involved [9].

There are, however, other issues beyond that specific concern regarding the ethical and social implications that weaving AC into the Metaverse might unravel. Immersive and pervasively hyped technologies can lead to issues such as addiction, emotional burnout, and loss of autonomy [2][18]. These effects also depend on cultural contexts, influencing how systems are perceived and accepted. Ethical frameworks must not only embrace global standards but also adapt them locally. Features such as breaks or control options that enable users to moderate their immersion and emotional engagement would help minimize this potential risk [17]. Finally, it would be of utmost importance to create open dialogue between all stakeholders, including users, developers, and regulators, to assure trust and inclusive design practices [17].

Turning AC in the metaverse into reality should be done with the utmost care and responsibility. The ethical issues previously described and their broader social implications should be addressed to make sure these systems are developed in an inclusive and ethical manner [2][17].

Chapter 3

Methodology & Development

To guide the data-driven focus of this research, the CRISP-DM methodology was considered, as seen in Figure 3.1. This framework structures the workflow into six phases that allow for an approach to explore, process, model, and evaluate data. In this context, CRISP-DM was applied to the VR Eyes: Emotions Dataset (VREED)[19], in which emotions were triggered using immersive 360° video-based virtual environments delivered via VR. In the context of this research, the objective is to investigate how affective states can be recognized within immersive environments like the Metaverse. This can support broader applications such as adaptive interfaces, emotionally intelligent avatars, and individual user-centered design in virtual spaces. The use of the VREED dataset aims to support this objective by enabling the development and validation of affective computing models based on physiological and behavioral data.

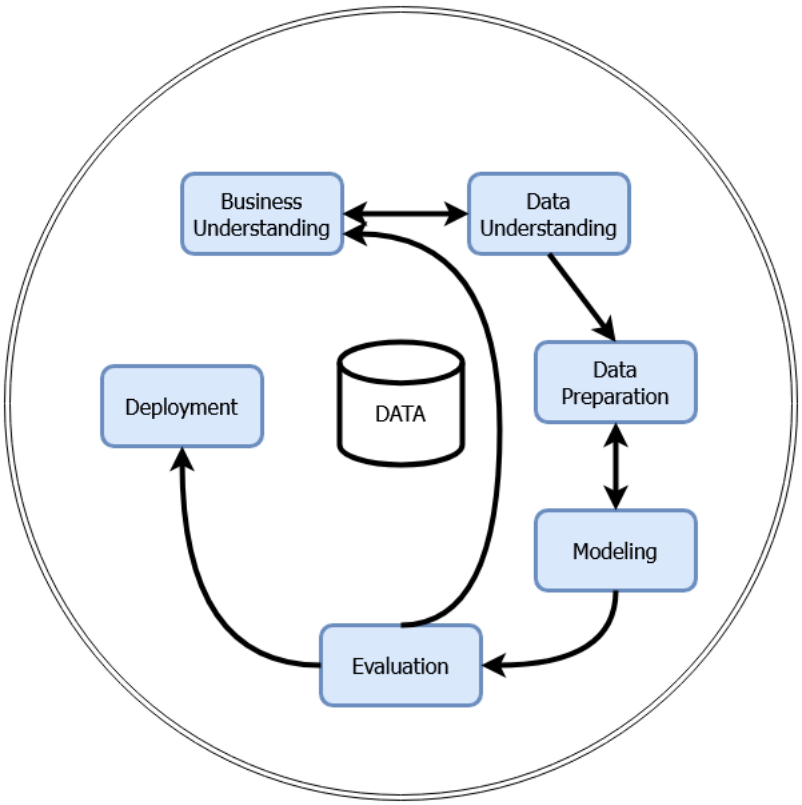


Figure 3.1: Diagram of CRISP-DM methodology, based on [20]

3.1 Business & Data Understanding

The VREED dataset consists of multimodal recordings including eye-tracking data, ECG signals, and GSR, all annotated with a categorical variable that represents emotional states. These emotional states are based on the Circumplex Model of Affects (CMA) which is a model that maps emotions and proposes that "all affective states arise from two fundamental neurophysiological systems, one related to valence (a pleasure–displeasure continuum) and the other to arousal, or alertness. Each emotion can be understood as a linear combination of these two dimensions, or as varying degrees of both valence and arousal" [21]. For example, an emotion such as excitement is associated with high valence and high arousal, whereas depression reflects low valence and low arousal. The categorical variable reflects the affective responses into four quadrants of this 2D space:

- High arousal / high valence
- Low arousal / high valence
- Low arousal / low valence
- High arousal / low valence

Understanding the nature and distribution of these states is essential in order to identify how physiological patterns correlate with affect, which in turn allows for modeling emotionally responsive systems to adapt to users' states in real time. Figure 3.2 displays a CMA diagram similar to the one which the VREED datasets were partially based on.

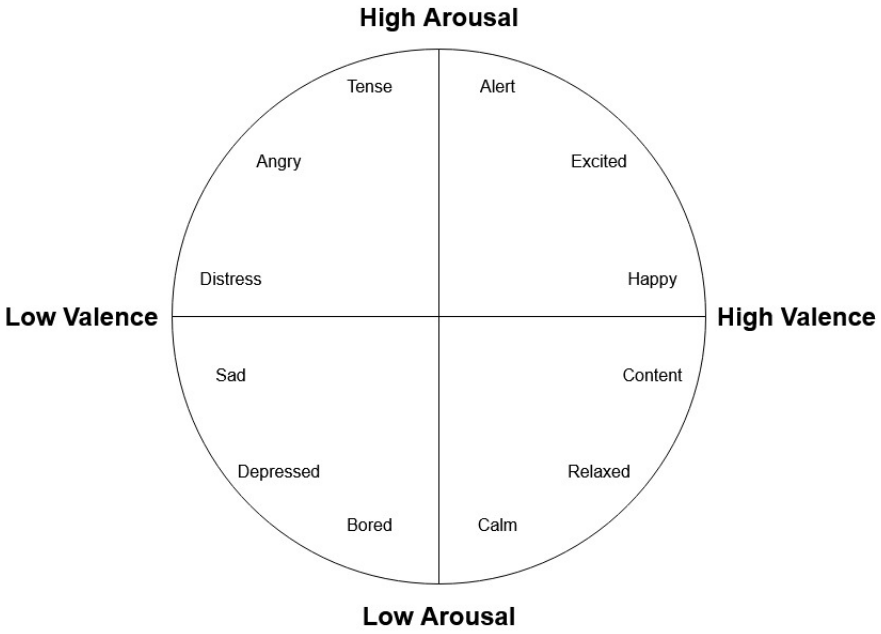


Figure 3.2: Circumplex Model of Affects, based on [21]

3.2 Data Preparation

VREED provides three preprocessed subsets corresponding to three modalities: eye-tracking, ECG and GSR. Each of these contains features already extracted from the raw data, aligned with a common target variable "Quad_Cat" (Quadrant Category) representing an emotional state. After loading the three subsets an initial base inspection was performed to understand the structure and overall content of each dataset. Then, a check for missing values revealed that the eye-tracking dataset contained some null entries, which were handled by assuming the column-wise mean, given its simplicity and suitability for continuous features. The datasets were well structured, with no data type or feature formatting issues.

Each dataset was constituted by 312 samples of extracted features, given that the number of features varies per dataset, with the Eye-tracking being the more "feature-rich" dataset with 50 features, followed by the ECG dataset with 19 features and finally the GSR dataset with 9 features.

Following that, the distribution of the target variable was checked, as is visible from Figures 3.3 to 3.5 which confirmed balanced datasets with 78 samples per class, evenly representing the four quadrants of CMA, which furthermore supports a fair training and evaluation of the classification models.

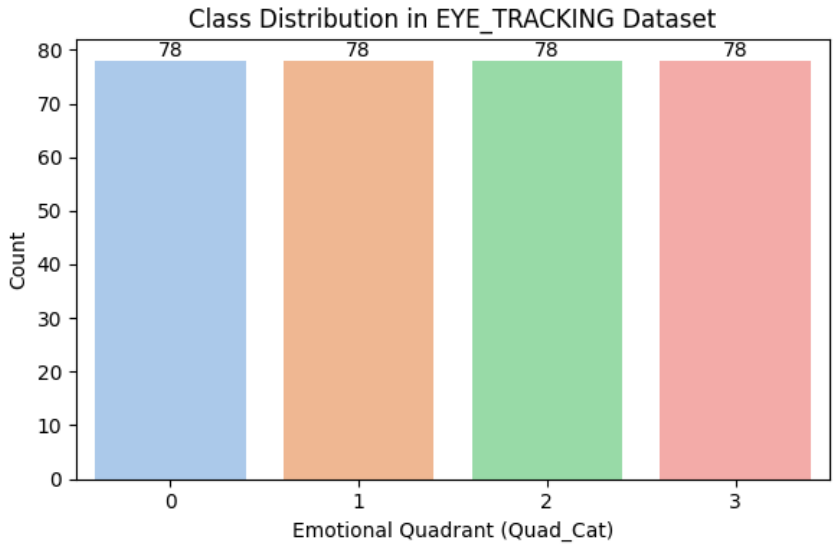


Figure 3.3: Eye-tracking dataset class distribution

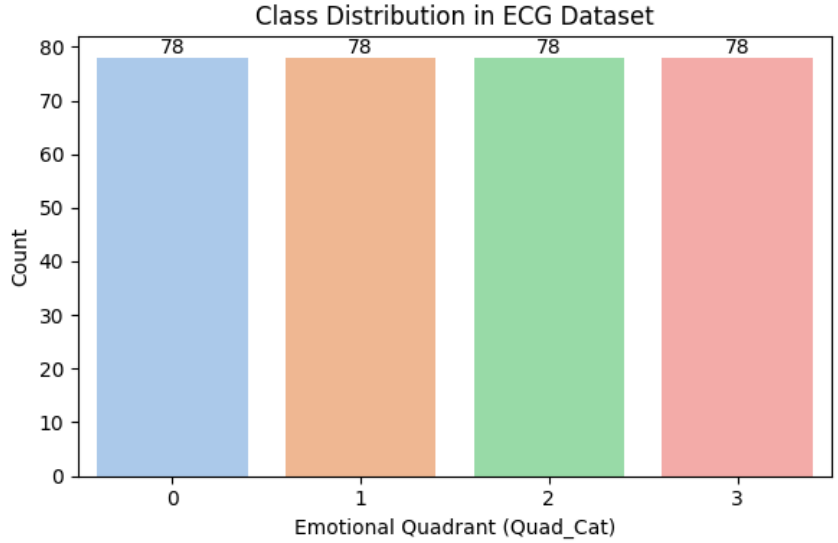


Figure 3.4: ECG dataset class distribution

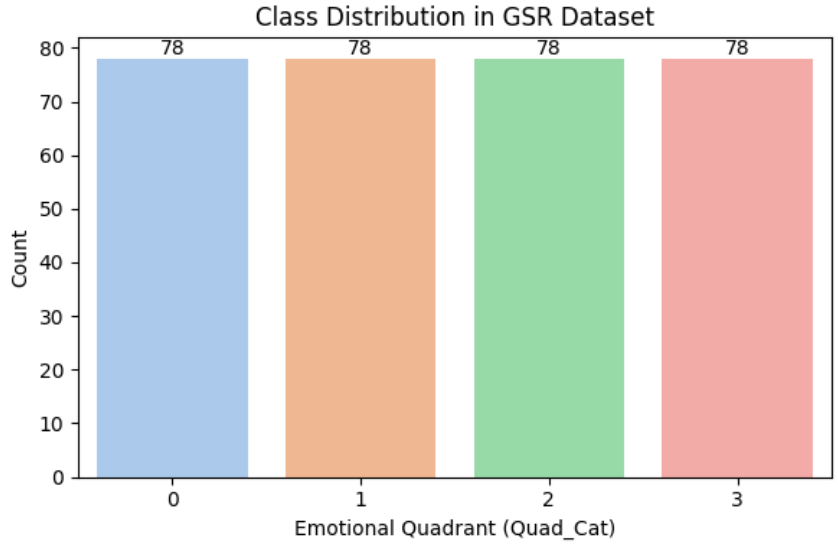


Figure 3.5: GSR dataset class distribution

3.3 Data Modeling

The goal of the data modeling phase was to assess different machine learning models' abilities in classifying affective states from physiological signals. This section provides the selection of a diversified set of supervised classification models and explains their implementation on both the raw and the augmented datasets.

3.3.1 Model Selection

A set of diverse supervised classification models was selected, spanning multiple algorithmic families, each with distinct learning paradigms and profiles, such as :

- Tree-based methods, known for robustness and the ability to handle nonlinear relationships (Random Forest, Gradient Boosting, Extra Trees) [22]
- Support Vector Machines (SVM) chosen for their effectiveness in high dimensional spaces and suitability for small to medium sized datasets (with linear and Radial Basis Function (RBF) kernels) [23]
- Linear models that are valued for their interpretability and efficiency and provide a strong baseline (Logistic Regression, Ridge Classifier) [24]
- Probabilist models which, while assuming feature independence, offer simplicity and fast training times (Gaussian Naive Bayes) [25]
- Neural models to evaluate how well deep learning techniques can learn affective representations from the available data (Multilayer Perceptron (MLP) with varying hidden units) [26][27]

This variety was selected in order for a broader comparison between model types, so it is possible to identify the models that best capture the affective patterns present in the multimodal data.

3.3.2 Modeling

Initially, the modeling was split into two different instances, one in which only the raw feature-extracted data would be taken into account, and another in which synthetic data would be considered, via Synthethic Minority Oversampling Technique (SMOTE), which is an algorithm that generates synthetic samples by interpolating between a data point and its nearest neighbors in the feature space, effectively creating new instances [28]. This would allow for a comparison of the performance of the models on raw vs. synthetically augmented data.

As a first step in the modeling process, each dataset was individually evaluated using all selected models to establish baseline performance. A 5-fold cross-validation strategy was used to ensure consistency and robustness. For models sensitive to feature scaling (SVMs, Logistic Regression, MLPs), the data was standardized prior to training. The goal of this initial stage was to measure each model's "base" performance, providing a reference point before applying any more advanced processing techniques.

Hyperparameter tuning was executed to improve model performance, and, as such, only the best performing model was selected (e.g. the Random Forest algorithm with 100 estimators outperformed other versions with 25 and 50, so it is the one presented). Table 3.1 shows the parameters used. Furthermore, some of the aforementioned algorithms in 3.3.1 consistently underperformed across all datasets, and, as such are not presented, to focus on the most relevant results.

Table 3.1: Hyperparameter tuning values

Algorithm	**Parameter**	**Values**
RandomForest	Estimators	25 50 **100**
GradientBoosting	Estimators	25 50 **100**
AdaBoost	Estimators	25 **50** 100
ExtraTrees	Estimators	25 **50** 100
MLP	Hidden Layers	25 50 **100**
LogisticRegression	Regularization	0.1 **1** 10
DecisionTree	Max Depth	3 5 **No limit**

After gathering the results, the next modeling phase involved the application of ensemble techniques for each dataset. Stacking [29], Bagging [30], and Voting [31] methods were explored to leverage the strengths of diverse base learners. The five top-performing models from the previous evaluation were selected as base estimators for these ensembles. This strategy allowed the ensembles to potentially correct for individual model biases and improve overall robustness.

Following the ensemble evaluation, the next procedure was to try a feature reduction step for all the base datasets, aiming to eliminate noisy or redundant features, especially given the relatively small sample size. A combination of techniques were applied: tree-based feature selection using Random Forest classifier, recursive feature elimination with a logistic regression model and a principal component analysis retaining 95% of the variance. To the reduced datasets produced by these procedures, ensemble methods were re-applied and a subsequent evaluation was realized.

Given these procedures, a trend was visible in the results of each of these modeling processes. The ECG dataset performed significantly worse than the remaining two. As such, the two best performing datasets were merged to form a new dataset to which the same modeling sequence was applied: initial evaluation, ensembling, feature reduction, and further ensembling. This decision is how a new instance (from now on referred to as "scenario(s)") came to be, splitting the research's approach to development into a total of three scenarios. One for raw dataset evaluation, another for the newly merged dataset evaluation and lastly a SMOTE enhanced dataset evaluation, aligning with the multi scenario approach visible in Figure 3.6.

As mentioned previously, each dataset had a balanced count of 78 samples per class, which, given the 4 quadrants of the CMA leaves us with a total of 312 samples per dataset. While balanced, the relatively small number of samples might impose limitations for more complex models and generalization. To address this, SMOTE was applied to the base datasets, which allowed a sample size increase while maintaining class balance. The entire modeling and evaluation process was then re-applied to the synthetic datasets, allowing for a direct comparison of model performance between the original and augmented data conditions.

The full modeling pipeline, including model evaluation, is visible in Figure 3.6, providing an overview of the approach.

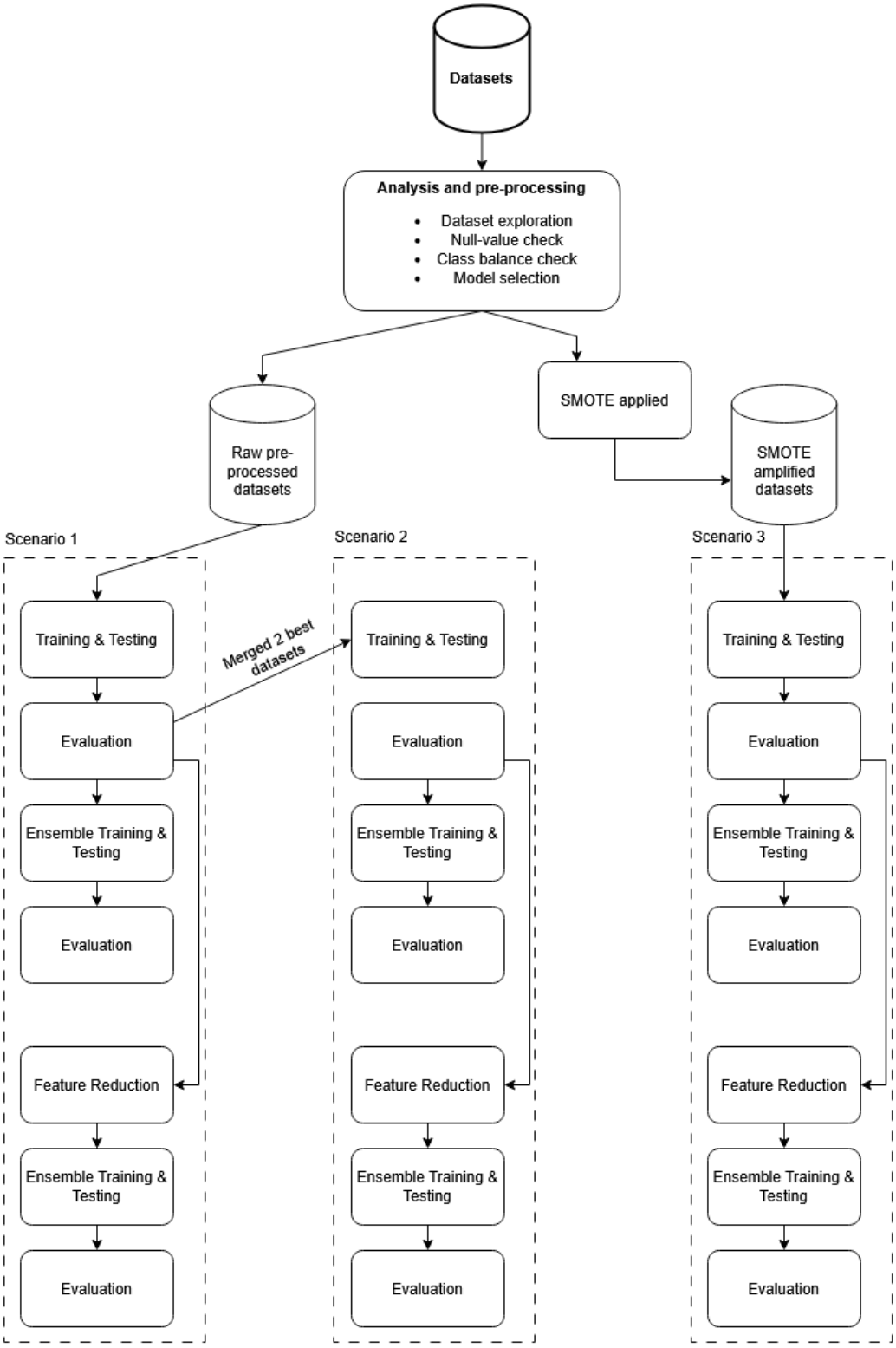


Figure 3.6: Modeling Diagram

3.4 Development

As aforementioned, the development followed a three-pronged development pipeline, known as the three "scenarios". These scenarios are:

1. The evaluation of the raw feature-extracted datasets,
2. The merging of the two best-performing datasets and re-evaluation,
3. The augmentation of the datasets using SMOTE and subsequent modeling.

Each of these scenarios was designed to explore different challenges in modeling, from the base performance on the real-world data to the effects of feature combination and data augmentation on model performance. The same core procedures were applied to each scenario, while each has its own set of variations.

3.4.1 Data preprocessing

Before the modeling phase, all datasets (GSR, ECG and eye-tracking) were subject to a set of preprocessing steps to improve overall data quality by eliminating data issues such as missing/inconsistent values, verifying class balance and feature scaling.

First, the datasets are checked for missing values.

```
datasets = {
    "eye_tracking": eye_data_df,
    "ecg": ecg_data_df,
    "gsr": gsr_data_df
}

for name, df in datasets.items():
    print(df.isnull().sum()[df.isnull().sum() > 0])
```

Listing 3.1: Checking for null values

Given that the eye-tracking dataset proves to have some missing data in four of its features we fill the data with the mean value of each of those features.

```
eye_data_df.fillna(eye_data_df.mean(), inplace=True)
```

Listing 3.2: Filling null values

The class balance was checked for biased learning prevention in the models and, as mentioned before and as it is visible from figures 3.3 to 3.5, all the classes were balanced.

```
all_labels = pd.concat([df["Quad_Cat"] for df in datasets.values()])

for name, df in datasets.items():
    plt.figure(figsize=(6, 4))
    ax = sns.countplot(x=df["Quad_Cat"], palette="pastel")

    plt.title(f"Class Distribution in {name.upper()} Dataset")
    plt.xlabel("Emotional Quadrant (Quad_Cat)")
    plt.ylabel("Count")
    plt.tight_layout()
    plt.show()
```

Listing 3.3: Checking class balance via plots

Then, the models were defined along with their respective hyperparameters. Note that not all the models mentioned in 3.3.1 are present due to the "filtering" based on performance mentioned on 3.3.2.

```
models = [
    ("RandomForest_100", RandomForestClassifier(n_estimators=100,
    random_state=42)),
    ("GradientBoosting_100", GradientBoostingClassifier(n_estimators=100,
    random_state=42)),
    ("SVM_linear", SVC(kernel="linear")),
    ("LogisticRegression", LogisticRegression(max_iter=10000, solver='saga
    ', random_state=42)),
    ("DecisionTree", DecisionTreeClassifier(random_state=42)),
    ("AdaBoost_50", AdaBoostClassifier(n_estimators=50, random_state=42)),
    ("ExtraTrees_50", ExtraTreesClassifier(n_estimators=50, random_state
    =42)),
    ("MLP_100", MLPClassifier(hidden_layer_sizes=(100,), max_iter=3000,
    random_state=42))
]
```

Listing 3.4: Defining the models to be tested

Some of these models are sensitive to the magnitude of input features. For these models, feature standardization was applied in a future step using "StandardScaler" to center features around zero and scale them to unit variance. Models such as tree-based classifiers (e.g, Random Forest, Extra Trees) are not sensitive to feature scaling, and as such, no transformation was applied for those cases. As such, the models that, within the list of previously defined models, need the application of such scaling were defined. Also, as it will be explored further in 4, the evaluation metrics (Accuracy, Precision, Recall and F1 Score) are defined here, along with the cross-validation strategy mentioned in 3.3.2.

```
models_requiring_scaling = {"SVM_linear", "LogisticRegression", "MLP_100"}

scoring = ['accuracy', 'precision_macro', 'recall_macro', 'f1_macro']

kf = KFold(n_splits=5, shuffle=True, random_state=42)
```

Listing 3.5: Defining the models to be scaled, evaluation metrics and cross-validation strategy

Each dataset was loaded and separated into features and the target variable "Quad_Cat", representing the emotional quadrant based on the CMA, was set.

```
X = df.drop(columns=["Quad_Cat"])
y = df["Quad_Cat"]
```

Listing 3.6: Setting the target variable

These steps were common between all three scenarios, and now the details regarding the different individual approaches to the datasets are explored.

3.4.2 Raw datasets

After the preprocessing steps defined 3.4.1 were performed the raw datasets were processed for each model, which ultimately provides a total of 96 results (8 models applied to 3 datasets while outputting 4 different metrics).

```
for name, df in datasets.items():
    for model_name, model in models:
        X_train = X.copy()

        if model_name in models_requiring_scaling:
            scaler = StandardScaler()
            X_train = scaler.fit_transform(X_train)

        scores = cross_validate(
            model,
            X_train,
            y,
            cv=kf,
            scoring=scoring,
            n_jobs=-1
        )

        results.append({
            "Dataset": name,
            "Model": model_name,
            "Accuracy": scores['test_accuracy'].mean(),
            "Precision": scores['test_precision_macro'].mean(),
            "Recall": scores['test_recall_macro'].mean(),
            "F1": scores['test_f1_macro'].mean(),
        })
```

Listing 3.7: Cross-validating the models on feature-extracted raw datasets

Given the completion of the cross-validation of these models on this data some plots are created for the visualization of results and are further explored in 4.1. For the sake of simplicity and to avoid introducing redundancy (since all plots are in the development are created with very similar code) none of the rest of plots' code will be presented. These results are also saved to .csv files simply to facilitate data loading in future steps without the need to keep everything constantly saved on memory.

We will use this saved data for ensemble training and testing and also for applying the models to these same datasets after their transformation via feature reduction.

```
results_df = pd.DataFrame(results)
results_df.to_csv("model_results.csv", index=False)

metrics = [
    ("Mean Accuracy", "Model Accuracy Comparison"),
    ("Mean F1", "Model F1 Score Comparison"),
    ("Mean Precision", "Model Precision Comparison"),
    ("Mean Recall", "Model Recall Comparison"),
]

palette = sns.color_palette("tab10")

for metric, title in metrics:
    for dataset_name in results_df["Dataset"].unique():
        subset = results_df[results_df["Dataset"] == dataset_name]

        plt.figure(figsize=(8, 6))
        ax = sns.barplot(data=subset, x="Model", y=metric, palette=palette
    )

        ax.set_title(f"{title} - {dataset_name}", fontsize=14, fontweight=
    'bold')
        ax.set_xlabel("Model", fontsize=12)
        ax.set_ylabel(metric, fontsize=12)
        ax.set_xticklabels(ax.get_xticklabels(), rotation=45, fontweight='
    bold')

        for container in ax.containers:
            ax.bar_label(container, fmt="%.2f", label_type="edge", padding
    =3, fontsize=10, fontweight='bold')

        y_max = max([bar.get_height() for bar in ax.patches]) * 1.15
        ax.set_ylim(0, y_max)

        plt.tight_layout()
        plt.show()
```

Listing 3.8: Plotting evaluation metrics for the models

After obtaining and visualizing the results, the next step in this scenario's pipeline (visible in Figure 3.6) was to verify the effect of ensemble models on the raw datasets. To proceed with the building of these ensemble models the five best performing models for each dataset (performance measured by the harmonic mean of precision and recall, known as the F1 score). Three methods of ensembling were selected for even more variety and broader comparison: Stacking, Voting and Bagging. The latter of which does not require multiple estimators and, as such, for this method, only the best performing model for each dataset was taken into account. Like on the previous approach, the results are also saved to a .csv file.

Note, it is also in these plots that the trend of the ECG dataset heavily underperforming is first observed and consequently where the decision to merge the two remaining datasets and explore their performance was made.

```python
results_df = pd.read_csv("model_results.csv")

for dataset_name, df in datasets.items():
    dataset_results = results_df[results_df["Dataset"] == dataset_name]
    top_5_models = dataset_results.nlargest(5, "F1")

    best_models = [(row["Model"], dict(models)[row["Model"]]) for _, row in top_5_models.iterrows()]

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    if any(model_name in models_requiring_scaling for model_name, _ in best_models):
        scaler = StandardScaler()
        X_train = scaler.fit_transform(X_train)
        X_test = scaler.transform(X_test)

    stacking_model = StackingClassifier(estimators=best_models, final_estimator=LogisticRegression(), n_jobs=-1)
    stacking_model.fit(X_train, y_train)
    y_pred = stacking_model.predict(X_test)
    ensemble_results.append({
        "Dataset": dataset_name,
        "Ensemble Model": "Stacking",
        "Accuracy": accuracy_score(y_test, y_pred),
        "Precision": precision_score(y_test, y_pred, average="macro"),
        "Recall": recall_score(y_test, y_pred, average="macro"),
        "F1": f1_score(y_test, y_pred, average="macro")})

    bagging_model = BaggingClassifier(estimator=best_models[0][1], n_estimators=10, n_jobs=-1)
    bagging_model.fit(X_train, y_train)
    y_pred = bagging_model.predict(X_test)
    ensemble_results.append({
        "Dataset": dataset_name,
        "Ensemble Model": "Bagging",
        "Accuracy": accuracy_score(y_test, y_pred),
        "Precision": precision_score(y_test, y_pred, average="macro"),
        "Recall": recall_score(y_test, y_pred, average="macro"),
        "F1": f1_score(y_test, y_pred, average="macro")})

    voting_model = VotingClassifier(estimators=best_models, voting="hard")
    voting_model.fit(X_train, y_train)
    y_pred = voting_model.predict(X_test)
    ensemble_results.append({
        "Dataset": dataset_name,
        "Ensemble Model": "Voting",
        "Accuracy": accuracy_score(y_test, y_pred),
        "Precision": precision_score(y_test, y_pred, average="macro"),
        "Recall": recall_score(y_test, y_pred, average="macro"),
        "F1": f1_score(y_test, y_pred, average="macro")})

ensemble_results_df = pd.DataFrame(ensemble_results)
ensemble_results_df.to_csv("ensemble_results.csv", index=False)
```

Listing 3.9: Building and processing of the ensemble methods

After that, plots were created similarly to the pre-ensemble approach, once again for result visualization.

Continuing the flow of the pipeline of development, the next step was trying feature reducing the datasets followed by the same ensemble methodology described above. The idea is to explore the impact of simplifying the datasets while retaining their essential characteristics, which can help improve the performance the models. Three feature reduction techniques were employed:

- Tree-based (using a Random Forest classifier [32]) that selects features whose importance is above the median.
- Linear (Recursive Feature Elimination (RFE) [33]) which removes the least important features and keeps only the top ten features.
- Principal Component Analysis (PCA) [34] which reduces dimensions while keeping 95% of variance.

```
reduced_datasets = {}

for dataset_name, df in datasets.items():

    selector = SelectFromModel(RandomForestClassifier(n_estimators=100,
    random_state=42), threshold="median")
    selector.fit(X, y)
    selected_features_tree = X.columns[selector.get_support()]

    rfe = RFE(LogisticRegression(max_iter=1000, random_state=42),
    n_features_to_select=10)
    rfe.fit(X, y)
    selected_features_rfe = X.columns[rfe.support_]

    pca = PCA(n_components=0.95, random_state=42)
    X_pca = pca.fit_transform(X)
    X_pca_df = pd.DataFrame(X_pca, index=X.index)


    reduced_datasets[dataset_name] = {
        "tree_based": X[selected_features_tree].join(y),
        "linear_models": X[selected_features_rfe].join(y),
        "pca": X_pca_df.join(y)
    }
```

After feature reducing, the next procedure is the consequent evaluation. In this case, there was no interest in the direct interpretation of the results obtained but rather in using these results for the ensemble process that follows. Only the models corresponding to the feature version they were reduced are run, hence the conditional checking.

```
for dataset_name, versions in reduced_datasets.items():
    for version, df in versions.items():

        if version == "linear_models" or version == "pca":
            X_scaled = scaler.fit_transform(X_scaled)

        for model_name, model in models:

            if model_name.startswith(("RandomForest", "GradientBoosting",
    "ExtraTrees", "DecisionTree", "AdaBoost")):
                if version != "tree_based":
                    continue
            elif model_name.startswith(("LogisticRegression", "SVM_linear"
    )):
                if version != "linear_models":
                    continue
            else:
                if version != "pca":
                    continue

            cv_results = cross_validate(
                model,
                X_scaled,
                y,
                cv=kf,
                scoring=scoring,
                n_jobs=-1
            )

            results.append({
                "Dataset": dataset_name,
                "Feature Version": version,
                "Model": model_name,
                "Accuracy": cv_results["test_accuracy"].mean(),
                "Precision": cv_results["test_precision_macro"].mean(),
                "Recall": cv_results["test_recall_macro"].mean(),
                "F1": cv_results["test_f1_macro"].mean(),
            })

refined_results_df = pd.DataFrame(results)
refined_results_df.to_csv("refined_model_results.csv", index=False)
```

After this phase, the next step is the subsequent ensembling, as previously mentioned, and, obviously, the evaluation of those results as well. Since the ensemble methodology is pretty similar to the one done on the non-feature reduced datasets it is not shown here.

This leads on to the next scenario in the development stage, in which the merged dataset was tackled with the same ideas and overall concepts.

3.4.3 Merged dataset

The first step in approaching this scenario is to, before anything else, check if the datasets are compatible to be merged since it only makes sense to proceed if that is the case, otherwise feature misalignment or mismatched sample correspondence could lead to invalid or misleading model training and evaluation results.

Validation is followed by the merging process and the definition of features and the target variable.

```
assert all(eye_data_df.index == gsr_data_df.index), "Index mismatch"

eye_features = eye_data_df.drop(columns=["Quad_Cat"])
gsr_features = gsr_data_df.drop(columns=["Quad_Cat"])

merged_df = pd.concat([eye_features, gsr_features], axis=1)

merged_df["Quad_Cat"] = eye_data_df["Quad_Cat"]

X = merged_df.drop(columns=["Quad_Cat"])
y = merged_df["Quad_Cat"]
```

After this validation and merging, the next step is intransitive to the first step referred to in 3.4.2, which is running a base validation and evaluation on the merged dataset, in this case.

```
for model_name, model in models:
    X_train = X.copy()

    if model_name in models_requiring_scaling:
        scaler = StandardScaler()
        X_train = scaler.fit_transform(X_train)

    scores = cross_validate(
            model,
            X_train,
            y,
            cv=kf,
            scoring=scoring,
            n_jobs=-1
        )

    results_2best.append({
            "Model": model_name,
            "Accuracy": scores['test_accuracy'].mean(),
            "Std": scores['test_accuracy'].std(),
            "Precision": scores['test_precision_macro'].mean(),
            "Recall": scores['test_recall_macro'].mean(),
            "F1": scores['test_f1_macro'].mean(),
            "Duration": duration
    })

results_2best_df = pd.DataFrame(results_2best)
```

As was the case for 3.4.2 this stage was then followed by the corresponding steps:

- Applying ensemble learning techniques
- Evaluation
- Feature reducing the dataset, then applying ensemble learning techniques
- Evaluation, once more

The biggest difference between these steps and those performed in 3.4.2 (after the base evaluation) is in the ensembling methodology, that, given the dataset consolidation, ends up being a bit more simplified, without needing nested looping structures.

```
top_5_models_merged2 = results_2best_df.nlargest(5, "F1")
best_models_merged2 = [(row["Model"], dict(models)[row["Model"]]) for _, row in top_5_models_merged2.iterrows()]

ensemble_results_merged2 = []

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

if any(model_name in models_requiring_scaling for model_name, _ in best_models):
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)

stacking_model = StackingClassifier(estimators=best_models, final_estimator=LogisticRegression(), n_jobs=-1)
stacking_model.fit(X_train, y_train)
y_pred = stacking_model.predict(X_test)
ensemble_results_merged2.append({
    "Ensemble Model": "Stacking",
    "Accuracy": accuracy_score(y_test, y_pred),
    "F1": f1_score(y_test, y_pred, average="macro"),
    "Precision": precision_score(y_test, y_pred, average="macro"),
    "Recall": recall_score(y_test, y_pred, average="macro"),
})

bagging_model = BaggingClassifier(estimator=best_models[0][1], n_estimators=10, n_jobs=-1)
bagging_model.fit(X_train, y_train)
y_pred = bagging_model.predict(X_test)
ensemble_results_merged2.append({
    "Ensemble Model": "Bagging",
    "Accuracy": accuracy_score(y_test, y_pred),
    "F1": f1_score(y_test, y_pred, average="macro"),
    "Precision": precision_score(y_test, y_pred, average="macro"),
    "Recall": recall_score(y_test, y_pred, average="macro"),
})

voting_model = VotingClassifier(estimators=best_models, voting="hard")
voting_model.fit(X_train, y_train)
y_pred = voting_model.predict(X_test)
ensemble_results_merged2.append({
    "Ensemble Model": "Voting",
    "Accuracy": accuracy_score(y_test, y_pred),
    "F1": f1_score(y_test, y_pred, average="macro"),
    "Precision": precision_score(y_test, y_pred, average="macro"),
    "Recall": recall_score(y_test, y_pred, average="macro"),
})

ensemble_results_merged2_pd = pd.DataFrame(ensemble_results_merged2)
```

3.4.4 SMOTE augmented dataset

As mentioned in 3.3.2 one of the scenarios to be considered would be approaching the raw datasets and generating synthetic data using SMOTE which would allow for a direct comparison of performance between models developed on raw data vs models developed on synthetically augmented data. The initial step is different from the previous scenarios given that the whole premise of this scenario is data augmentation and, as such, that's the first procedure. While SMOTE was used to increase the sample size rather than correct class imbalance, multiple values of target size were tested (by multiplying the size of the original dataset by different values) and, to avoid overfitting, the training vs. testing gaps were monitored. The optimum value of samples was found to be 312 (4 times the original sample size so 75% synthetic data is present in the augmented dataset). In this case a pipeline was necessary to prevent leaking synthetic data across folds.

```
for name, df in datasets.items():
    sampling_strategy = {label: 312 for label in np.unique(y)}

    for model_name, model in models:
        steps = []

        if model_name in models_requiring_scaling:
                steps.append(('scaler', StandardScaler()))

        steps.append(('smote', SMOTE(sampling_strategy=sampling_strategy,
    random_state=42)))
        steps.append(('classifier', model))
        pipeline = Pipeline(steps)

        scores = cross_validate(pipeline, X, y, cv=kf, scoring=scoring,
    return_train_score=True, n_jobs=-1)

        results_smote.append({
            "Dataset": name,
            "Model": model_name,
            "Accuracy": scores['test_accuracy'].mean(),
            "Precision": scores['test_precision'].mean(),
            "Recall": scores['test_recall'].mean(),
            "F1": scores['test_f1'].mean(),
        })

        train_acc = scores['train_accuracy'].mean()
        test_acc = scores['test_accuracy'].mean()
        train_pre = scores['train_precision'].mean()
        test_pre = scores['test_precision'].mean()
        train_rec = scores['train_recall'].mean()
        test_rec = scores['test_recall'].mean()
        train_f1 = scores['train_f1'].mean()
        test_f1 = scores['test_f1'].mean()
        gap_acc = train_acc - test_acc
        gap_pre = train_pre - test_pre
        gap_rec = train_rec - test_rec
        gap_f1 = train_f1 - test_f1

        y_pred = cross_val_predict(pipeline, X, y, cv=kf, n_jobs=-1)

results_smote_df = pd.DataFrame(results_smote)
```

After that, the same overall process is once again applied in the same order:

1. Ensembling techniques applied, where, as is it was done in previous scenarios, the top performing models were selected
2. Evaluation of these results
3. Feature reduction of the dataset and another round of ensembling techniques
4. Final evaluation

Finally, given the nature of this iterative, three-way divided process, multiple evaluation phases could be observed for each scenario and these are explored and discussed in the next chapter.

Chapter 4

Evaluation

Following the modeling flow outlined before, the performance of each model and the ensemble, feature reduction and SMOTE techniques was assessed based on multiple metrics: Accuracy, Precision, Recall and F1 score. Given the multi-class nature of the datasets, precision, recall and F1 score were computed using macro-averaging in order to make sure each class contributes equally to the final score. Accuracy, on the other hand, was reported as the overall proportion of correctly classified instances across all classes.

As described in Figure 3.6, the modeling process was "divided" into three major scenarios: raw datasets, two merged best-performing datasets, and SMOTE-augmented datasets. Within each scenario, individual models and ensembles were evaluated to measure how each strategy affected performance. Given the volume of results due to the repetitive nature of the modeling scenarios (three scenarios, three evaluation phases each, and four metrics per model) to ensure clarity only the best performing evaluation stage for each scenario is focused and presented in detail, while reporting the most relevant metrics: accuracy and F1 score. However, since the raw dataset represents the scenario of baseline performance of the individual models their results are also reported, giving a benchmark for interpreting changes introduced by subsequent modeling stages.

4.1 Scenario 1 - Raw datasets

In the raw datasets approach, the goal was to establish baseline performance using the original extracted features. This helps highlight how well the models generalize, without any performance tweaking or data synthetization. Figures 4.1 to 4.6 present the accuracy and F1 score obtained during cross-validation.

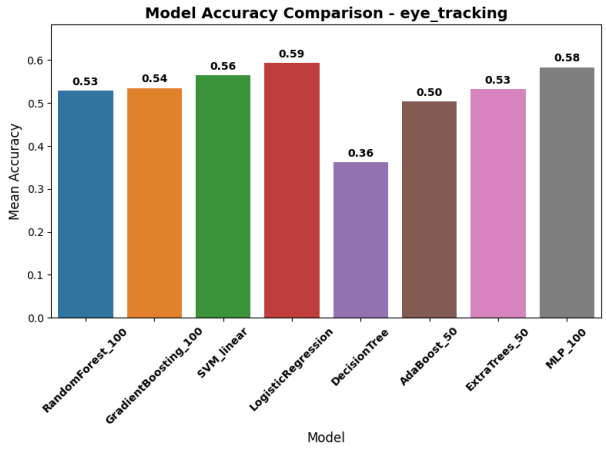


Figure 4.1: Accuracy of individual base models evaluated on the eye-tracking dataset

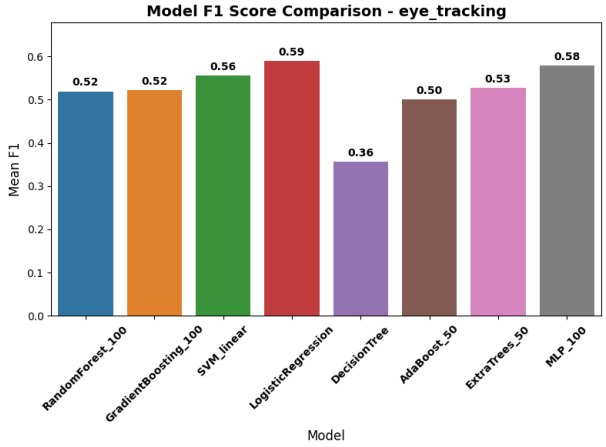


Figure 4.2: F1 score of individual base models evaluated on the eye-tracking dataset

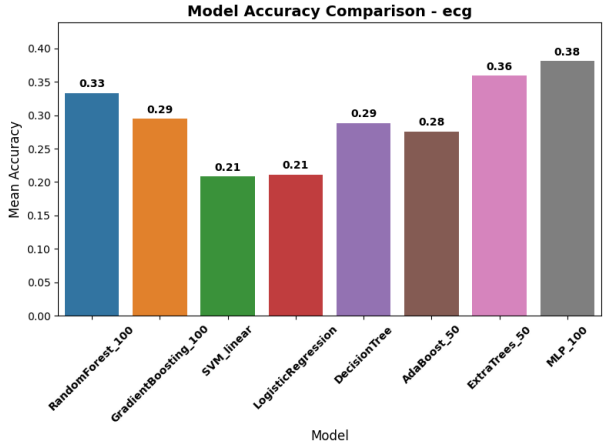


Figure 4.3: Accuracy of individual base models evaluated on the ECG dataset

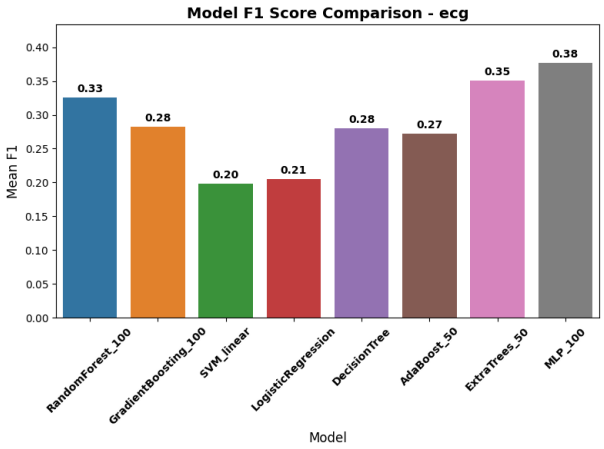


Figure 4.4: F1 score of individual base models evaluated on the ECG dataset

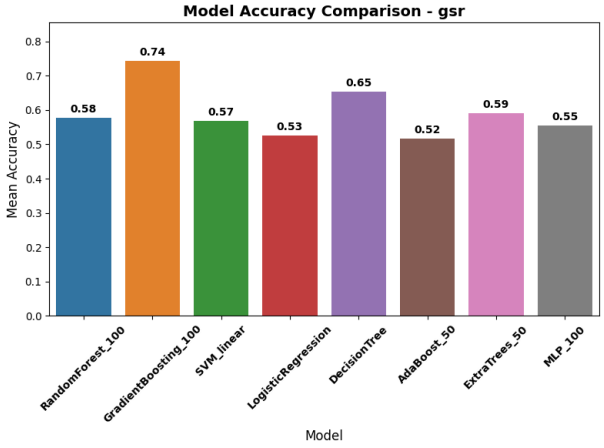


Figure 4.5: Accuracy of individual base models evaluated on the GSR dataset

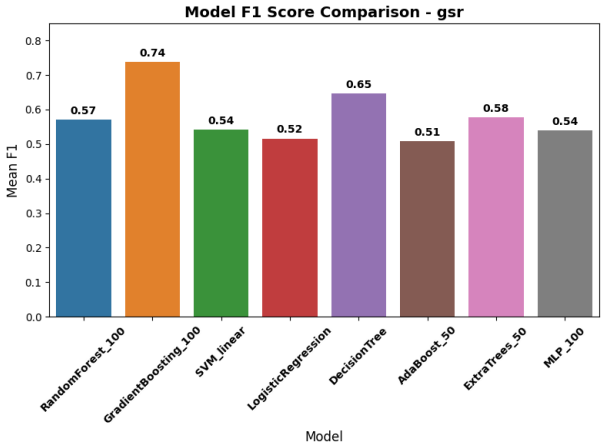


Figure 4.6: F1 score of individual base models evaluated on the GSR dataset

It is possible to notice a slight pattern in the results, seeing how the better performing algorithms are tree-based, regarding the GSR dataset (which, overall, was the dataset with

best performing models in this "raw" approach), but the inverse is visible in the Eye tracking dataset, where Logistic Regression and MLP visibly performed the best. Overall, the ECG dataset produced mediocre results, which, as mentioned before, is a trend that will repeat itself.

However, the best results for this scenario were obtained by the ensemble techniques applied to the datasets, specifically the Bagging method, which emerged as the best performer, once more regarding the GSR dataset, as it is possible to observe in Figures 4.11 and 4.12. This suggests that high-variance models are performing well individually and their aggregation reduces overfitting without sacrificing performance.

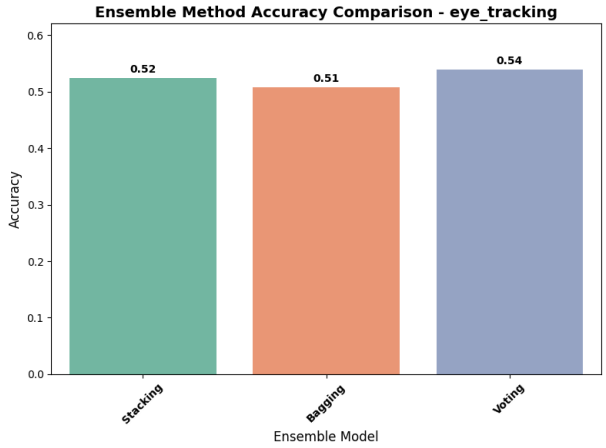


Figure 4.7: Accuracy of ensemble methods on the eye-tracking dataset

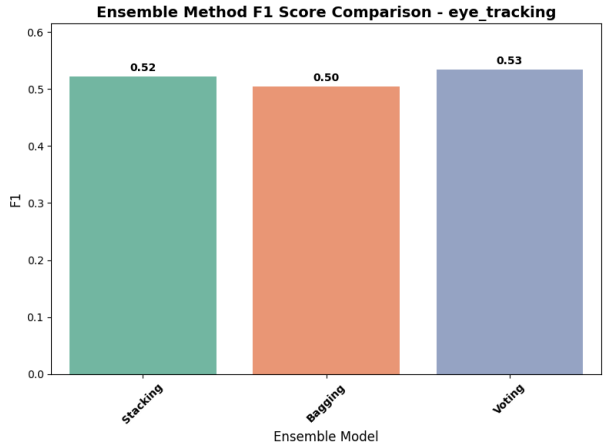


Figure 4.8: F1 score of ensemble methods on the eye-tracking dataset

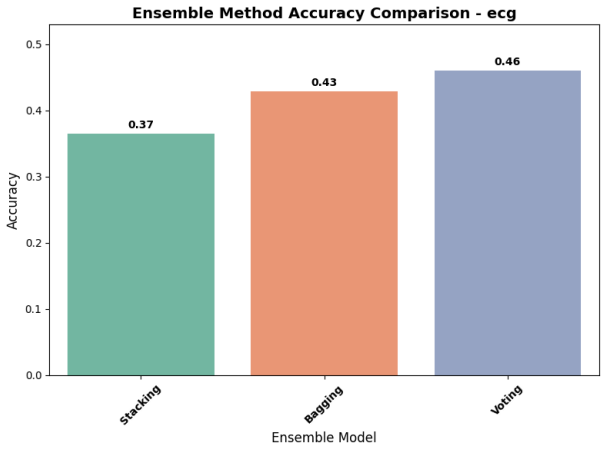


Figure 4.9: Accuracy of ensemble methods on the ECG dataset

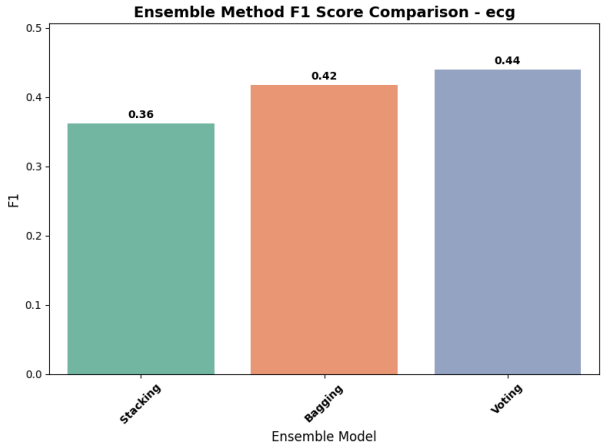


Figure 4.10: F1 score of ensemble methods on the ECG dataset

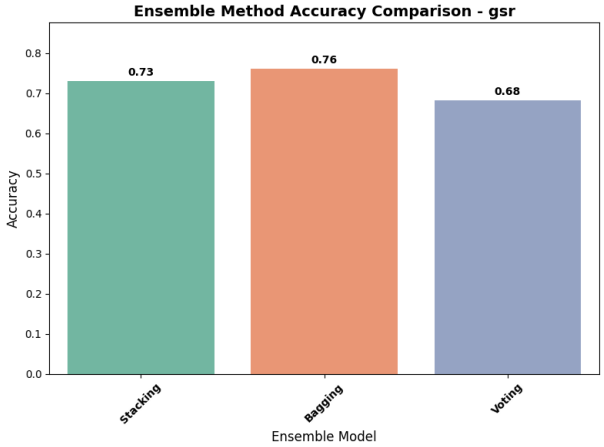


Figure 4.11: Accuracy of ensemble methods on the GSR dataset

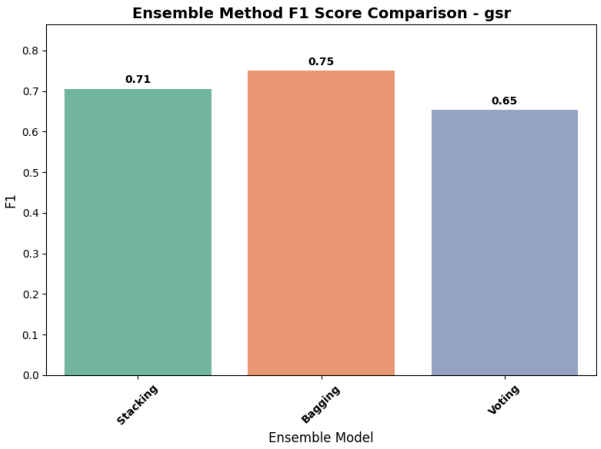


Figure 4.12: F1 score of ensemble methods on the GSR dataset

4.2 Scenario 2 - Merged datasets

Moving on to the second scenario of the data modeling, where the two best performing datasets overall (GSR and Eye-tracking) were merged into a single dataset, the goal was to explore if combining complementary signals could improve classification performance. This merged dataset was subjected to the same modeling steps as before, including baseline evaluation, ensemble methods, and feature reduction. Comparing the baseline split dataset evaluation to the baseline merged dataset evaluation, there was an overall gain across almost all models, however, some models, such as Gradient Boosting, performed slightly worse than in the GSR dataset alone, which highlights that while merging may enhance general model robustness it can also introduce some dilution to dataset-specific strengths.

As in the previous scenario, only the best performing evaluation phase is presented. The results, visible in Figures 4.13 and 4.14 show that ensemble methods once again prove to produce the best results, although this time, Voting outperformed the remaining two methods, which suggests that while no single model dominates their consensus is strong, showing that the models' different strengths complement each other, allowing for a reduced bias and reliable outcome.

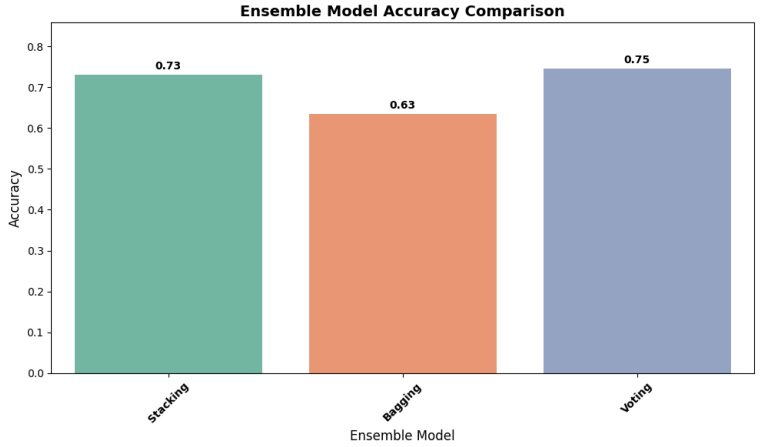


Figure 4.13: Accuracy of ensemble methods evaluated on the merged datasets

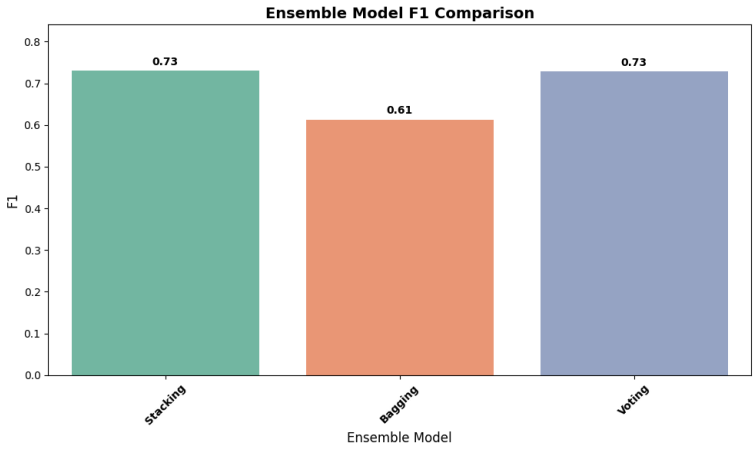


Figure 4.14: F1 score of ensemble methods evaluated on the merged datasets

4.3 Scenario 3 - Augmented datasets

Finally, analyzing the last scenario, where SMOTE was applied to generate synthetic data. The objective was not to address class imbalance, given that the original datasets were already balanced, but rather to increase the overall sample size and improve model generalization. As in the previous scenarios, this augmented data was processed through the same modeling flow, including the evaluation phases. Out of which, as previously done with the other scenarios, the best performing evaluation phase was selected.

The results, visible from Figures 4.15 to 4.20 show that when directly comparing the performance of GSR dataset before vs. after the application of SMOTE there is a clear improvement in several base models, particularly tree-based algorithms. These gains are some of the most significant throughout the modeling process and suggest that SMOTE enhanced the models' ability to generalize. However, although some cases did show substantial improvements, the impact of SMOTE was not consistent across all datasets. A decline in performance in the ECG dataset and a mixed performance in the Eye-tracking dataset likely reflects differences in how clearly each dataset's signals distinguish between classes and how meaningful the original features are for classification.

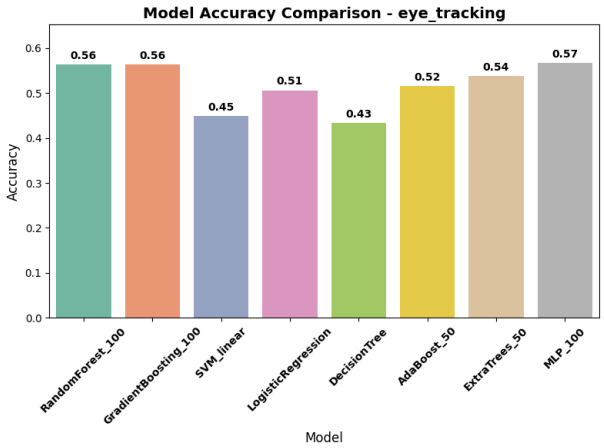


Figure 4.15: Accuracy of individual base models evaluated on the augmented eye-tracking dataset

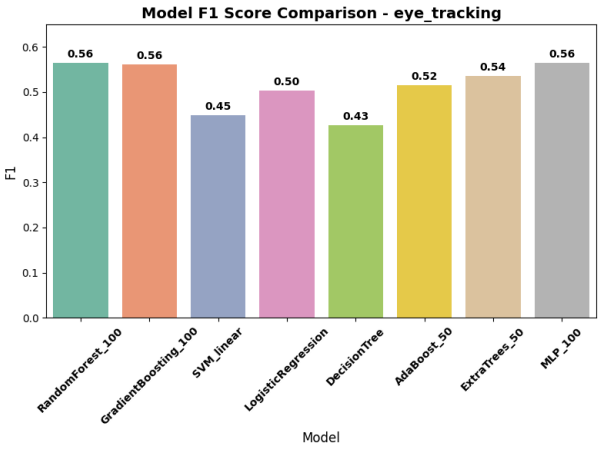


Figure 4.16: F1 score of individual base models evaluated on the augmented eye-tracking dataset

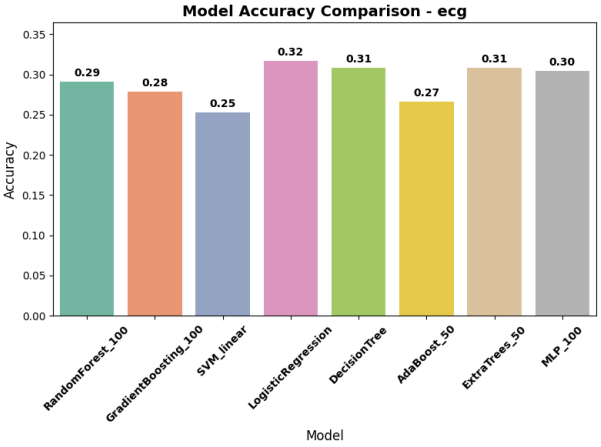


Figure 4.17: Accuracy of individual base models evaluated on the augmented ECG dataset

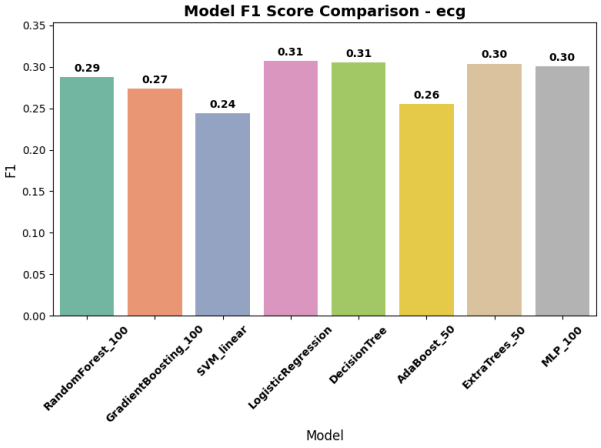


Figure 4.18: F1 score of individual base models evaluated on the augmented ECG dataset

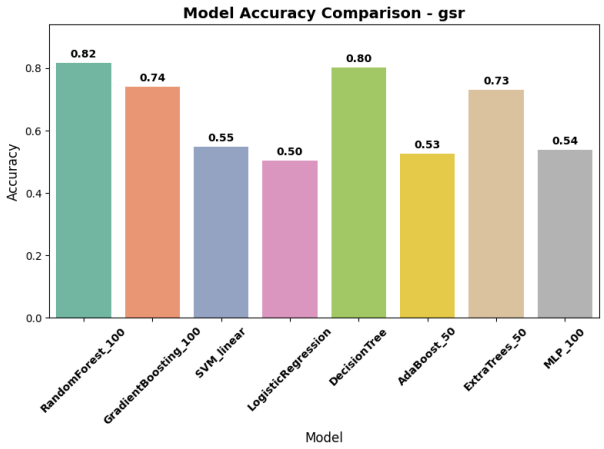


Figure 4.19: Accuracy of individual base models evaluated on the augmented GSR dataset

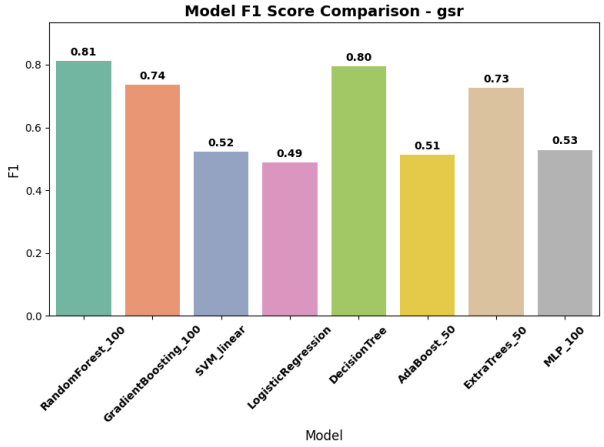


Figure 4.20: F1 score of individual base models evaluated on the augmented GSR dataset

To further support the findings, Figure 4.21 presents the feature importance of this scenario. The most influential feature aligns with physiological expectations, given that "Ratio" refers to the "Ratio of peaks and time"[19], which is defined as "a burst or a peak in the phasic response approximately 1 to 5 seconds after exposure to emotional stimuli. The greater the number of peaks, the greater the arousal during that experience"[35]. Additionally, the confusion matrix in Figure 4.22 provides a detailed breakdown of the model's classification performance, highlighting some interesting details. As it is clearly visible, the most common wrong predictions were "3" when the actual value was "0", vice versa, and also predicting "2" when the actual label is "1" and vice versa. Given that, as shown on Figure 3.2, "0" represents the top-right (High arousal / high valence) quadrant of the CMA and the remaining classes follow in a clockwise direction, these misclassifications appear to occur most frequently between adjacent emotional quadrants, which indicates that valence is less easily perceived or distinguishable than arousal.

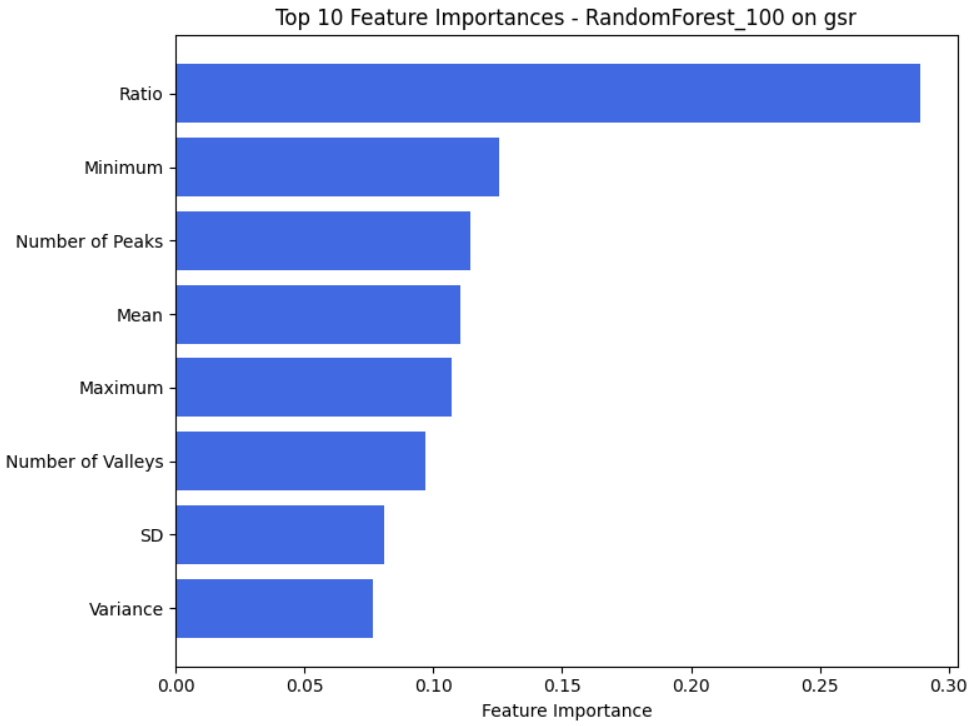


Figure 4.21: SMOTE Feature Importance

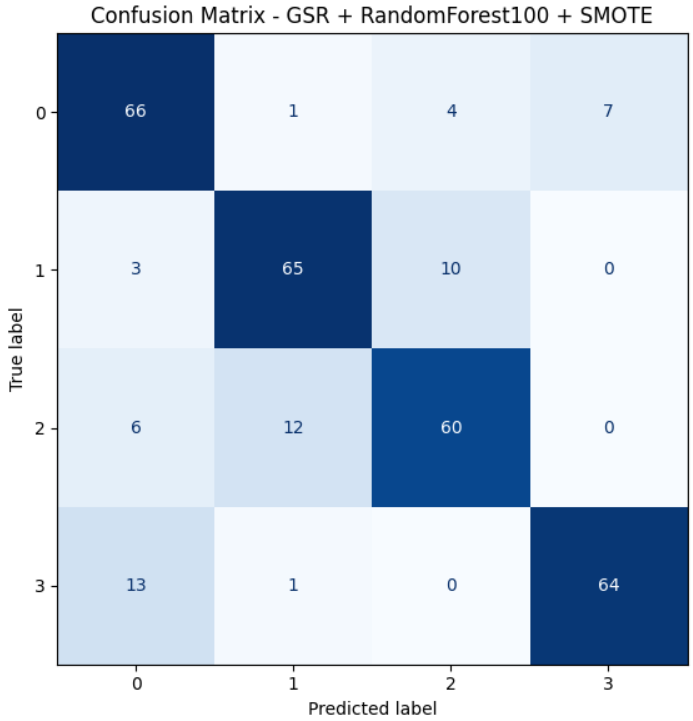


Figure 4.22: SMOTE Confusion Matrix

4.4 Overall

Overall, the evaluation of these three scenarios demonstrated that while the SMOTE-augmented GSR dataset achieved the highest overall scores, no single model or strategy consistently outperformed the others across all datasets or evaluation metrics. Tree-based models generally led in effectiveness, while the merging of the complementary datasets produced moderate yet stable improvements. These findings show the importance of tailoring modeling strategies to the unique properties of each dataset, as well as considering the nature of the features and signal quality, in order to maximize classification performance. Given the results and context, it is easy to deduce that the less feature-rich dataset (ECG), shows less clear differences between emotional states and that heart rate, heart rate variability and RR intervals may not be sufficiently expressive to distinguish emotional states on their own, unlike the GSR dataset, which seems to capture clearer responses to different stimuli or conditions. The eye-tracking dataset falls between GSR and ECG in terms of feature richness and classification performance, reflecting moderate differentiation of emotional states.

The final selected models from each scenario, along with their performance across all four metrics, are summarized in Table 4.1, offering a view of how each strategy impacted classification effectiveness. Also, Figure 4.23 presents the best overall's model Receiver Operating Characteristic (ROC) curve which plots of the True Positive Rate (TPR) against the False Positive Rate (FPR) at each threshold setting, which helps assess the model's discriminative capacity independent of class distribution.

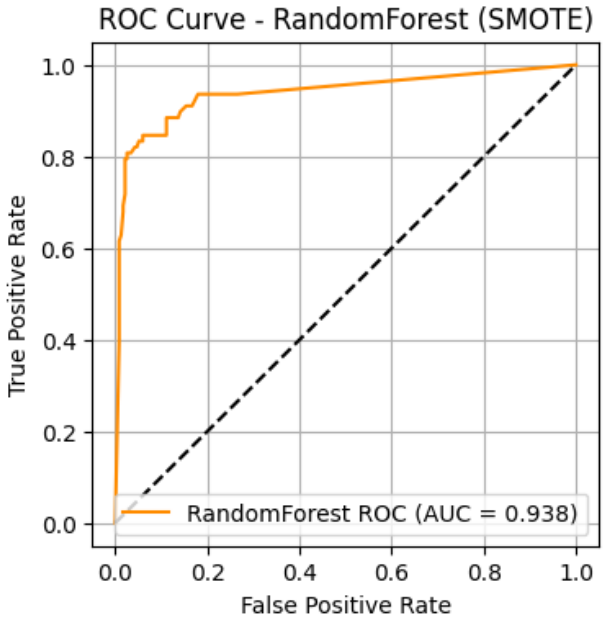


Figure 4.23: SMOTE Model ROC Curve

Table 4.1: Metrics of the best evaluation phases, per scenario

Metric	Base (Bagging-GSR)	Merged (Voting)	SMOTE (RandomForest-GSR)
Accuracy	0.76	0.75	**0.82**
F1 Score	0.75	0.73	**0.81**
Precision	0.78	0.75	**0.83**
Recall	0.76	0.72	**0.81**

The evaluation results indicate that the combination of fine-tuned modeling strategies with appropriate data preprocessing can lead to meaningful affect recognition performance. Applying a RandomForest classifier to the SMOTE augmented GSR dataset proved to stand out as the most effective overall approach, which suggests that physiological responses captured through skin conductance are particularly expressive for distinguished emotional states. Merging datasets lead to better generalization when directly compared to the raw dataset results but was still outperformed by ensemble methods applied to those individual datasets, suggesting that data merging offers benefits but it can also dilute the strengths of more informative signals. These findings reinforce the importance of aligning model selection and modeling techniques with the specific characteristics of each signal type.

Chapter 5

Conclusions

The work carried out thus far was key in setting a foundation for the development of affect recognition models based on physiological signals, with the goal of enabling intelligent systems suitable for immersive environments such as the Metaverse. By exploring multimodal datasets the project investigated various modeling strategies that revealed that certain modalities, particularly GSR and eye-tracking, contributed effectively to emotional state classification. Ensemble methods also showed a strong potential in improving robustness and generalization.

The use of PRISMA for literature review and CRISP-DM for model development proved effective in overall structuring the research. Each scenario revealed some insights into tradeoffs between data quality, data volume and augmentation strategies, which, in turn supports the usage of the iterative (and split) approach of the defined pipeline.

Despite reaching promising results some challenges were encountered during this research, such as the limited size present in the datasets, that being the main reason for the decision of creating the SMOTE-augmented scenario, which limited the models' ability to generalize and increased the risk of overfitting. Also, the feature richness varied across the datasets, with ECG offering fewer cues compared to GSR or eye-tracking, which is likely one of the main reasons it underperformed consistently. These limitations may have affected overall model performance and show that more extensive and diverse datasets can produce even better results.

In the future, the project can then focus on deploying the developed model in immersive environments such as the Metaverse, guided by the insights gathered from this research stage, by integrating the model into interactive systems where affect recognition can enhance user experience. Nevertheless, as previouly mentioned, further improvements to the model could be obtained by trying to nullify one of this research's biggest challenge by collecting additional, broader, data sources, which in turn would allow for better refining model generalization, while still addressing challenges such as ethical considerations.

Bibliography

[1] R. W. Coutinho and A. Boukerche, "When smart metaverse meets affective computing: Opportunities and design guidelines," *IEEE Communications Magazine*, vol. 61, pp. 46–52, 10 Oct. 2023, issn: 15581896. doi: 10.1109/MCOM.004.2300009.

[2] F. Pervez, M. Shoukat, M. Usama, M. Sandhu, S. Latif, and J. Qadir, "Affective computing and the road to an emotionally intelligent metaverse," *IEEE Open Journal of the Computer Society*, vol. 5, pp. 195–214, 2024, issn: 26441268. doi: 10.1109/OJCS.2024.3389462.

[3] R. W. Picard, *Affective computing*. Cambridge, MA, USA: MIT Press, 1997, isbn: 0262161702.

[4] J. D. N. Dionisio, W. G. B. III, and R. Gilbert, "3d virtual worlds and the metaverse: Current status and future possibilities," *ACM Comput. Surv.*, vol. 45, no. 3, 2013, issn: 0360-0300. doi: 10.1145/2480741.2480751.

[5] L. F. Barrett, "Are emotions natural kinds?" *Perspectives on Psychological Science*, vol. 1, no. 1, pp. 28–58, 2006, issn: 1745-6916. doi: 10.1111/j.1745-6916.2006.00003.x.

[6] J. A. Russell, "Core affect and the psychological construction of emotion," *Psychological Review*, vol. 110, no. 1, pp. 145–172, 2003, issn: 0033-295X. doi: 10.1037/0033-295x.110.1.145.

[7] K. Scherer, "Scherer kr. what are emotions? and how can they be measured? soc sci inf 44: 695-729," *Social Science Information*, vol. 44, pp. 695–792, Dec. 2005. doi: 10.1177/0539018405058216.

[8] M. J. Page, J. E. McKenzie, P. M. Bossuyt, *et al.*, "The prisma 2020 statement: An updated guideline for reporting systematic reviews," *The BMJ*, vol. 372, Mar. 2021, issn: 17561833. doi: 10.1136/bmj.n71.

[9] B. B. Gupta, A. Gaurav, K. T. Chui, and V. Arya, "Deep learning-based facial emotion detection in the metaverse," in *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*, Institute of Electrical and Electronics Engineers Inc., 2024, isbn: 9798350324136. doi: 10.1109/ICCE59016.2024.10444217.

[10] K. Gupta, Y. Zhang, T. S. Gunasekaran, N. Krishna, Y. S. Pai, and M. Billinghurst, "Caevr: Biosignals-driven context-aware empathy in virtual reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, pp. 2671–2681, 5 May 2024, issn: 19410506. doi: 10.1109/TVCG.2024.3372130.

[11] T. Baidya and S. Moh, "Comprehensive survey on resource allocation for edge-computing-enabled metaverse," *Computer Science Review*, vol. 54, Nov. 2024, issn: 15740137. doi: 10.1016/j.cosrev.2024.100680.

[12] S. Rokhsaritalemi, A. Sadeghi-Niaraki, and S. M. Choi, "Exploring emotion analysis using artificial intelligence, geospatial information systems, and extended reality for urban services," *IEEE Access*, vol. 11, pp. 92 478–92 495, 2023, issn: 21693536. doi: 10.1109/ACCESS.2023.3307639.

[13] B. J. Keegan, I. P. McCarthy, J. Kietzmann, and A. I. Canhoto, "On your marks, headset, go! understanding the building blocks of metaverse realms," *Business Horizons*, vol. 67, pp. 107–119, 1 Jan. 2024, issn: 00076813. doi: 10.1016/j.bushor.2023.09.002.

[14] Y. K. Dwivedi, L. Hughes, A. M. Baabdullah, *et al.*, "Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy," *International Journal of Information Management*, vol. 66, Oct. 2022, issn: 02684012. doi: 10.1016/j.ijinfomgt.2022.102542.

[15] H. Kim and T. Hong, "Enhancing emotion recognition using multimodal fusion of physiological, environmental, personal data," *Expert Systems with Applications*, vol. 249, 2024, issn: 0957-4174. doi: https://doi.org/10.1016/j.eswa.2024.123723. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742400589X.

[16] D. Pal and C. Arpnikanondt, "The sweet escape to metaverse: Exploring escapism, anxiety, and virtual place attachment," *Computers in Human Behavior*, vol. 150, 2024, issn: 0747-5632. doi: https://doi.org/10.1016/j.chb.2023.107998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0747563223003497.

[17] D. Shanley and D. Meacham, "A place where "you can be who you've always wanted to be. . . " examining the ethics of intelligent virtual environments," *Journal of Responsible Technology*, vol. 18, Jun. 2024, issn: 26666596. doi: 10.1016/j.jrt.2024.100085.

[18] G. Lampropoulos, P. Fernández-Arias, Á. Antón-Sancho, and D. Vergara, "Affective computing in augmented reality, virtual reality, and immersive learning environments," *Electronics (Switzerland)*, vol. 13, 15 Aug. 2024, issn: 20799292. doi: 10.3390/electronics13152917.

[19] L. Tabbaa, R. Searle, S. M. Bafti, *et al.*, "Vreed: Virtual reality emotion recognition dataset using eye tracking & physiological measures," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 4, Dec. 2022. doi: 10.1145/3495002. [Online]. Available: https://doi.org/10.1145/3495002.

[20] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining," in *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining (PADD)*, vol. 1, 2000, pp. 29–39.

[21] J. Posner, J. A. Russell, and B. S. Peterson, "The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology"," *Dev. Psychopathol.*, vol. 17, 2005.

[22] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001. doi: 10.1023/A:1010933404324.

[23] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995. doi: 10.1007/BF00994018.

[24] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009. doi: 10.1007/978-0-387-84858-7.

[25] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, 2001, pp. 41–46.

[26] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015. doi: 10.1016/j.neunet.2014.09.003.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org.

[28] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[29] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992. doi: 10.1016/S0893-6080(05)80023-1.

[30] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996. doi: 10.1007/BF00058655.

[31] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, 2012, isbn: 9781439830031.

[32] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010. doi: 10.1016/j.patrec.2010.03.014.

[33] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002. doi: 10.1023/A:1012487302797.

[34] I. T. Jolliffe, *Principal Component Analysis*, 2nd. Springer, 2002. doi: 10.1007/b98835.

[35] Y. S. Can, B. Arnrich, and C. Ersoy, "Stress detection in daily life scenarios using smart phones and wearable sensors: A survey," *Journal of Biomedical Informatics*, vol. 92, p. 103 139, 2019, issn: 1532-0464. doi: https://doi.org/10.1016/j.jbi.2019.103139. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046419300577.

Appendix A

Planning

Planning a clear structure and direction is key in any project that aims to truly achieve its objectives. Here presented is the planning phase of this project, going through topics such as the methodologies adopted, skill development strategies, detailing phases scheduling and outlining potential risks that can hinder the project's execution.

A.1 Project Scope

The goal this project aims to achieve is to to develop, implement and evaluate a framework that integrates AC in the Metaverse. As such, the methodology which most aligns with this project is the Action and Research methodology. As its name implies the project has Act (develop and implement the framework) and Research (evaluate the results and refine, accordingly) stages. This is a cyclical process that ensures more validation and improvement through practical application. There are some considerations to be taken into account, such as defining clear goals for each iteration, avoiding over-extension in these iterations and addressing ethical concerns, the latter revolving around data privacy and the possibility of emotional manipulation and even algorithmic bias.

Improving emotional intelligence and user engagement is the validation sought by this project, which aims to incorporate emotion detection and response. Metrics such as the accuracy of emotion recognition, the time spent in the environment and even user-reported satisfaction are key for our evaluations in each implementation stage. This choice of methodology and its iterations are reflected in the Work Breakdown Structure (WBS) in Figure A.1.

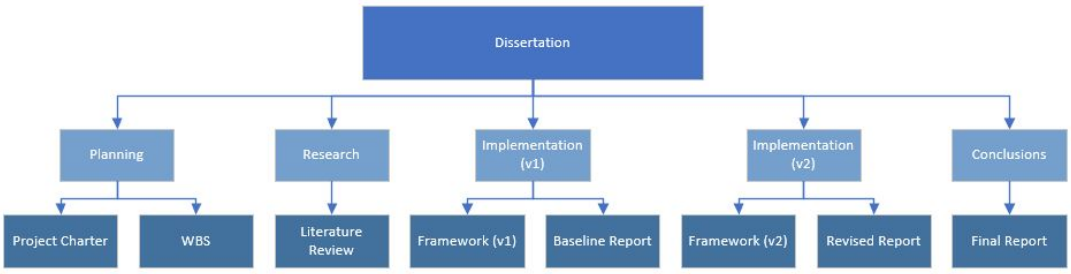


Figure A.1: The developed Work Breakdown Structure

A.2 Project Schedule

In order to outline the key phases identified as necessary for the development of this project a project schedule was developed. It is designed to align with the re-iterative nature of the Action and Research methodology ensuring that the time allocated for each phase reflects the effort demanded, while allowing for a small amount of flexibility, if necessary. As such, a Gantt chart was developed (using Microsoft Project) and as it possible to see in Figure A.2 it details the phases more thoroughly. It is possible to analyze that the implementation iterations occupy most of the time (which was expected, given the adopted methodology) and that there is also time considerations for report reviewing and discussing.

Taking these discussions into account, it is also important to clarify the project monitoring which will consist in status meetings between advisor and advisee every 3 weeks, and progress reporting which will be done by task reviewing and updating the created Microsoft Project document in the "% done" column with percentage values. Figure A.2 is appendix (A) of this document.

ID	Task Mode	Task Name	Duration	Start	Finish
1		**Dissertation**	**196 days**	**Mon 30/09/24**	**Mon 30/06/25**
2		**Planning**	**10 days**	**Mon 30/09/24**	**Fri 11/10/24**
3		Project Charter	5 days	Mon 30/09/24	Fri 04/10/24
4		WBS	5 days	Mon 30/09/24	Fri 04/10/24
5		Plan	10 days	Mon 30/09/24	Fri 11/10/24
6		**Research**	**70 days**	**Mon 14/10/24**	**Fri 17/01/25**
7		Literature review	60 days	Mon 14/10/24	Fri 03/01/25
8		Identifying design	10 days	Mon 06/01/25	Fri 17/01/25
9		**Implementation (v1)**	**45 days**	**Wed 05/02/25**	**Tue 08/04/25**
10		Build framework	30 days	Wed 05/02/25	Tue 18/03/25
11		Test and evaluate	7 days	Wed 19/03/25	Thu 27/03/25
12		Baseline report	5 days	Fri 28/03/25	Thu 03/04/25
13		Report feedback	3 days	Fri 04/04/25	Tue 08/04/25
14		**Implementation (v2)**	**45 days**	**Fri 04/04/25**	**Thu 05/06/25**
15		Build framework	30 days	Fri 04/04/25	Thu 15/05/25
16		Test and evaluate	7 days	Fri 16/05/25	Mon 26/05/25
17		Revised report	5 days	Tue 27/05/25	Mon 02/06/25
18		Report feedback	3 days	Tue 03/06/25	Thu 05/06/25
19		**Conclusion**	**17 days**	**Fri 06/06/25**	**Mon 30/06/25**
20		Final report	15 days	Fri 06/06/25	Thu 26/06/25
21		Prepare presentation	2 days	Fri 27/06/25	Mon 30/06/25
22		**Milestones**	**124 days**	**Fri 03/01/25**	**Thu 26/06/25**
23		PREPD delivery	0 days	Fri 03/01/25	Fri 03/01/25
24		Design identified	0 days	Fri 17/01/25	Fri 17/01/25
25		Implementation (v1) created	0 days	Tue 18/03/25	Tue 18/03/25
26		Baseline report delivered	0 days	Thu 03/04/25	Thu 03/04/25
27		Implementation (v1) created	0 days	Thu 15/05/25	Thu 15/05/25
28		Revised report delivered	0 days	Mon 02/06/25	Mon 02/06/25
29		Final report delivered	0 days	Thu 26/06/25	Thu 26/06/25

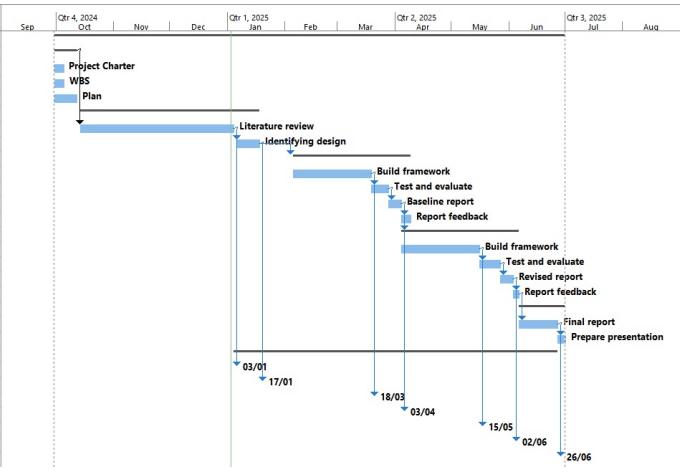


Figure A.2: The project's planned timeline

A.3 Skills management

The development of this project demands certain skills, some of which were determined to still have room for improvement. As such, Table A.1 was developed to reflect the skills that were found lacking and the actions that will be taken into account to combat these gaps. The project document developed (Figure A.2) does not reflect all these development actions as specific tasks because most of them are skills developed during the duration of the project itself. Note that some actions, like creating a detailed schedule have already been developed and consequently presented in this document.

Table A.1: Skills to develop

Skill	Development actions
Time management	Planning aligned with advisor, with temporally well-defined phases and action plan Time blocking techniques
Technical skills in the specific area of knowledge	Deepen knowledge related to Affective Computing Development of prototypes planned for the project
Planning and organization	Structuring the work in stages Creating a detailed schedule
Proactivity	Search for continuous feedback Actively participate as a "user" of the project, not just as an advisee

A.4 Risk analysis

Analyzing potential risks associated with the development of this project and its corresponding effects and outcomes is crucial. Accordingly, a risk register was developed to expose risks and their causes, their probabilities and also their effects and corresponding impacts along with how it is planned to respond to these risks in order to mitigate their effects. Figure A.3 presents the developed risk register, where it is possible to see that "multiple causes possible" are considered for one of the risks.

Given that this project investigates a relatively unexplored area of study there are multiple variables that can influence the development of the project without previously seeming relevant, influencing the level reached by the development, and it is reasonable to expect the challenges previously outlined can introduce new variables that cannot be considered as of now. Figure A.3 is appendix (B) of this document.

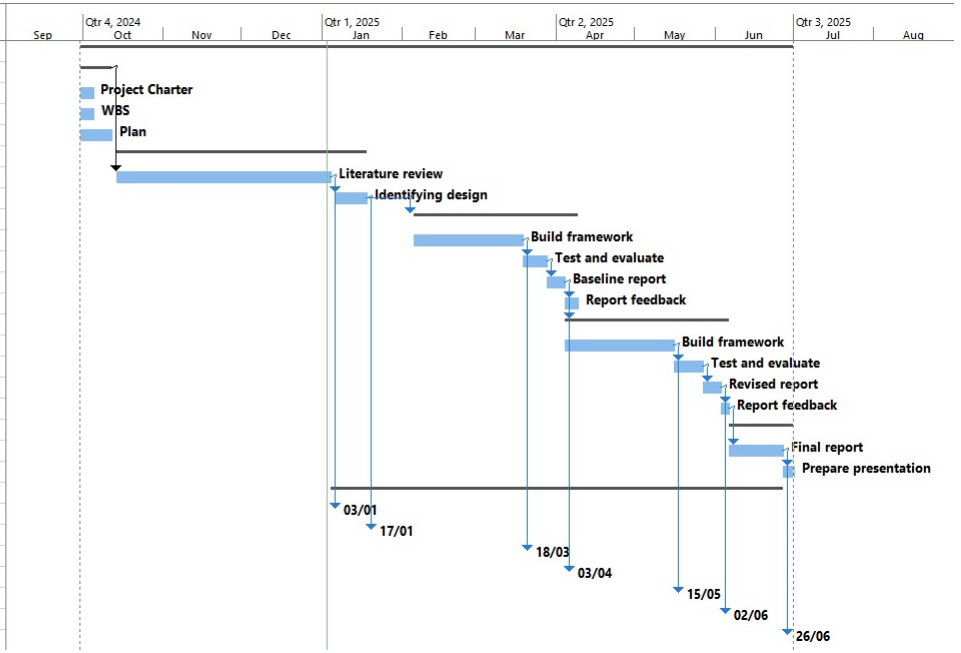
Risk ID	Description	Cause	Effect	Risk Owner	Probability (1-5)	Impact (1-5)	PI Score	Expected Result, No Action	Risk Response Type	Response description
	Description of the risk	Cause of the risk	Effect on the project	Name of person who monitors the risk	Group sourced rough estimate of how likely this is to occur	Rough estimate of how significant the impact of this risk	Probability multiplied by Impact	What will happen if the risk becomes an issue and no action is taken	Decision made by group on how to respond to this risk	How do you know it is time to put the response into play
1	The equipment doesn't have functionalities needed	Wrong Equipment	Deadline compromised	Osvaldo Silva	1	5	5	We will not be able to secure the timeline	Mitigate	Use another equipment
2	The dataset doesn't have quality	Mediocre Dataset	Higher execution	Osvaldo Silva	2	4	8	Not able to analyse the data to the fullest	Mitigate	Investigate other datasets
3	Development doesn't reach desired level	Multiple causes possible	Worse end results	Osvaldo Silva	2	3	6	Results will be worse	Mitigate	Focus on critical functionalities

Figure A.3: Risk Register

Appendix B

Gantt Chart

ID	Task Mode	Task Name	Duration	Start	Finish
1		**Dissertation**	**196 days**	**Mon 30/09/24**	**Mon 30/06/25**
2		**Planning**	**10 days**	**Mon 30/09/24**	**Fri 11/10/24**
3		Project Charter	5 days	Mon 30/09/24	Fri 04/10/24
4		WBS	5 days	Mon 30/09/24	Fri 04/10/24
5		Plan	10 days	Mon 30/09/24	Fri 11/10/24
6		**Research**	**70 days**	**Mon 14/10/24**	**Fri 17/01/25**
7		Literature review	60 days	Mon 14/10/24	Fri 03/01/25
8		Identifying design	10 days	Mon 06/01/25	Fri 17/01/25
9		**Implementation (v1)**	**45 days**	**Wed 05/02/25**	**Tue 08/04/25**
10		Build framework	30 days	Wed 05/02/25	Tue 18/03/25
11		Test and evaluate	7 days	Wed 19/03/25	Thu 27/03/25
12		Baseline report	5 days	Fri 28/03/25	Thu 03/04/25
13		Report feedback	3 days	Fri 04/04/25	Tue 08/04/25
14		**Implementation (v2)**	**45 days**	**Fri 04/04/25**	**Thu 05/06/25**
15		Build framework	30 days	Fri 04/04/25	Thu 15/05/25
16		Test and evaluate	7 days	Fri 16/05/25	Mon 26/05/25
17		Revised report	5 days	Tue 27/05/25	Mon 02/06/25
18		Report feedback	3 days	Tue 03/06/25	Thu 05/06/25
19		**Conclusion**	**17 days**	**Fri 06/06/25**	**Mon 30/06/25**
20		Final report	15 days	Fri 06/06/25	Thu 26/06/25
21		Prepare presentation	2 days	Fri 27/06/25	Mon 30/06/25
22		**Milestones**	**124 days**	**Fri 03/01/25**	**Thu 26/06/25**
23		PREPD delivery	0 days	Fri 03/01/25	Fri 03/01/25
24		Design identified	0 days	Fri 17/01/25	Fri 17/01/25
25		Implementation (v1) created	0 days	Tue 18/03/25	Tue 18/03/25
26		Baseline report delivered	0 days	Thu 03/04/25	Thu 03/04/25
27		Implementation (v1) created	0 days	Thu 15/05/25	Thu 15/05/25
28		Revised report delivered	0 days	Mon 02/06/25	Mon 02/06/25
29		Final report delivered	0 days	Thu 26/06/25	Thu 26/06/25



Appendix C

Risk Register

Risk ID	Description	Cause	Effect	Risk Owner	Probability (1-5)	Impact (1-5)	PI Score	Expected Result, No Action	Risk Response Type	Response description
	Description of the risk	Cause of the risk	Effect on the project	Name of person who monitors the risk	Group sourced rough estimate of how likely this is to occur	Rough estimate of how significant the impact of this risk	Probability multiplied by Impact	What will happen if the risk becomes an issue and no action is taken	Decision made by group on how to respond to this risk	How do you know it is time to put the response into play
1	The equipment doesn't have functionalities needed	Wrong Equipment	Deadline compromised	Osvaldo Silva	1	5	5	We will not be able to secure the timeline	Mitigate	Use another equipment
2	The dataset doesn't have quality	Mediocre Dataset	Higher execution	Osvaldo Silva	2	4	8	Not able to analyse the data to the fullest	Mitigate	Investigate other datasets
3	Development doesn't reach desired level	Multiple causes possible	Worse end results	Osvaldo Silva	2	3	6	Results will be worse	Mitigate	Focus on critical functionalities