



Machine Learning for Shoulder Rehabilitation

JORGE FILIPE PEREIRA CARDOSO

Setembro de 2024

Machine Learning for Shoulder Rehabilitation

Jorge Filipe Pereira Cardoso

**A dissertation submitted in partial fulfilment of
the requirements for the degree in Master of Science,
Specialisation in the Area of Information and Knowledge Systems**

Supervisor: Maria De Fátima Coutinho Rodrigues

Supervisor: Bruno Miguel Almeida Cunha

Abstract

This project addressed the critical challenge of monitoring and evaluating home-based shoulder rehabilitation exercises through the application of Machine Learning tools.

The rising prevalence of stroke-related disabilities and the shortage of qualified therapists, exacerbated by the COVID-19 pandemic, underscore the need for innovative solutions. The research explores the feasibility of leveraging Machine Learning models to assess the quality of rehabilitation exercises performed by patients in a home-based setting. Utilizing MediaPipe Pose framework, the study captures and analyzes patients' movements during exercises, comparing them with recordings of healthcare professionals executing the same tasks. The proposed Machine Learning model aims to provide insightful progress reports, aiding therapists in delivering focused and intensive therapy remotely. This project involved designing and implementing Machine Learning models, developing an application to be used by patients and healthcare professionals, prescribing exercises, evaluating the quality of rehabilitation exercises, and testing the solution end-to-end. This project was conducted in a partnership with physiotherapists from Center for Rehabilitation Research from Politécnico do Porto.

Keywords: Home-based rehabilitation, shoulder rehabilitation, machine learning, MediaPipe Pose, healthcare technology.

Resumo

Este projeto abordou o desafio crítico da monitorização e avaliação de exercícios de reabilitação do ombro, em casa, através da aplicação de ferramentas de aprendizagem automática.

A crescente prevalência de deficiências relacionadas com o Acidente Vascular Cerebral e a escassez de terapeutas qualificados, exacerbada pela pandemia de COVID-19, sublinham a necessidade de soluções inovadoras. A investigação explora a viabilidade de utilizar modelos de aprendizagem automática para avaliar a qualidade dos exercícios de reabilitação realizados por pacientes num ambiente domiciliário. Utilizando a *framework* MediaPipe Pose, o estudo capta e analisa os movimentos dos doentes durante os exercícios, comparando-os com gravações de profissionais de saúde que executam as mesmas tarefas. O modelo de aprendizagem automática proposto tem como objetivo fornecer relatórios de progresso perspicazes, ajudando os terapeutas a fornecerem remotamente uma terapia focada e intensiva. Este projeto envolveu a conceção e implementação de modelos de aprendizagem automática, o desenvolvimento de uma aplicação a ser utilizada por pacientes e profissionais de saúde na prescrição de exercícios e na avaliação da qualidade dos exercícios de reabilitação e o teste da solução de ponta a ponta. Este projeto foi realizado em parceria com fisioterapeutas do Centro de Investigação em Reabilitação do Politécnico do Porto.

Palavras-chave: Reabilitação em casa, reabilitação do ombro, aprendizagem automática, MediaPipe Pose, tecnologia de cuidados de saúde.

Contents

Contents	vii
List of Figures	xi
List of Tables	xiii
List of Acronyms	xv
1 Introduction	1
1.1 Context.....	1
1.2 Problem Definition	2
1.3 Objectives	2
1.4 Methodology.....	3
1.5 Document Structure.....	3
2 State of the art.....	5
2.1 Rehabilitation.....	5
2.1.1 Home-based rehabilitation.....	6
2.2 Machine Learning.....	8
2.2.1 Supervised Algorithms.....	9
2.2.1.1 Support Vector Machines	9
2.2.1.2 Artificial Neural Networks.....	10
2.2.1.3 K-Nearest Neighbours	11
2.2.1.4 Gradient Boosting.....	12
2.2.2 Unsupervised Algorithms.....	13
2.2.2.1 Gaussian Mixture Models.....	14
2.2.2.2 Principal Component Analysis	15
2.3 Qualisys Track Manager.....	15
2.4 MediaPipe.....	16
2.5 Distance Time Warping.....	17
2.6 Related Works.....	17
3 Machine Learning model for shoulder rehabilitation exercises assessment .	21
3.1 Dataset characterization.....	21
3.2 Ground truth	21

3.3	Data Pre-processing.....	22
3.3.1	Keypoint Extraction.....	23
3.4	Modelling.....	24
3.4.1	Distance calculation approach.....	24
3.4.2	Gaussian Mixture Model approach.....	27
3.5	Final remarks.....	31
4	Problem Analysis	33
4.1	Requirements Analysis.....	33
4.1.1	Functional Requirements	33
4.1.2	Non-functional Requirements.....	35
4.1.2.1	Functionality.....	35
4.1.2.2	Usability	35
4.1.2.3	Reliability	35
4.1.2.4	Performance	35
4.1.2.5	Supportability.....	35
4.1.2.6	“+”	35
4.2	Problem’s Domain	36
4.2.1	Domain Model.....	36
5	Solution Design	37
5.1	Level 1: Context Diagrams.....	38
5.1.1	Logic View.....	38
5.1.2	Process View	39
5.2	Level 2: Container Diagrams.....	39
5.2.1	Logic View.....	40
5.2.2	Implementation View.....	40
5.2.3	Process View	41
5.2.4	Physical View	42
5.3	Level 3: Component Diagram.....	43
5.3.1	Logic View.....	43
5.3.2	Implementation View.....	44
5.3.3	Process View	44
6	Solution Implementation	47
6.1	Technologies Chosen.....	47

6.2	Implementation.....	48
6.2.1	ML Model.....	48
6.2.2	Web API.....	48
6.2.3	Mobile App.....	51
6.2.4	Database.....	53
6.2.5	Fileserver.....	53
7	Conclusion	55
7.1	Limitations and Future Work.....	55
	References	57

List of Figures

Figure 1: Home-based shoulder rehabilitation technologies, adapted from (Cunha et al., 2023)	7
Figure 2: Classification vs Regression Algorithms (“Regression vs. Classification in Machine Learning,” 2021)	9
Figure 3: Support Vector Machine (Tan et al., 2019)	10
Figure 4: Artificial Neural Network (Dertat, 2017).....	11
Figure 5: K-Nearest Neighbours (Tech-AI-Math, 2023)	12
Figure 6: Gradient Boosting (“Flow diagram of gradient boosting machine learning method. The ensemble...,” n.d.).....	13
Figure 7: Definition of landmarks in MediaPipe Pose	16
Figure 8: Example of MediaPipe’s keypoints mapped on video during exercise	24
Figure 9: Distance calculation process.....	25
Figure 10: Mean values of distance vs QTM software result classification for diagonal exercise	26
Figure 11: Mean values of distance vs QTM software result classification for rotation exercise	26
Figure 12: Cumulative explained variance by PCA Components for the diagonal exercise	28
Figure 13: Atypical frame for the patient and corresponding frame for the physiotherapist .	29
Figure 14: Typical frame for the patient and corresponding frame for the physiotherapist ..	29
Figure 15: Mean Log-Likelihood for the 3 repetitions of the diagonal exercise for all patients	30
Figure 16: Mean Log-Likelihood for the 3 repetitions of the rotation exercise for all patients	30
Figure 17: Progress of patient over three sessions.....	31
Figure 18: Use Case Diagram.....	34
Figure 19: Domain Model	36
Figure 20: Level 1 Logic View.....	38
Figure 21: Level 1 Process View for UC3.....	39
Figure 22: Level 1 Process View for UC4.....	39
Figure 23: Level 2 Logic View.....	40
Figure 24: Level 2 Implementation View	41
Figure 25: Level 2 Process View for UC3.....	41
Figure 26: Level 2 Process View for UC4.....	42
Figure 27: Level 2 Physical View	42

Figure 28: Level 3 Logic View.....	43
Figure 29: Level 3 Implementation View	44
Figure 30: Level 3 Process View for UC3.....	45
Figure 31: Level 3 Process View for UC4.....	45
Figure 32: Sign Up and Log in pages.....	51
Figure 33: Different pages of the Mobile App.....	52
Figure 34: Entity Relationship Diagram of the Database.....	53

List of Tables

Table 1: Results provided by QTM software..... 22

List of Acronyms

ANN	Artificial Neural Networks
CNN	Convolutional Neural Networks
DTW	Dynamic Time Warping
GMM	Gaussian Mixture Model
KNN	K nearest neighbours
LSTM	Long Short-Term Memory
ML	Machine Learning
NAS	Network-Attached Storage
PCA	Principal Component Analysis
QTM	Qualisys Track Manager
RNN	Recurrent Neural Networks
SVM	Support Vector Machine
WHO	World Health Organization

1 Introduction

This chapter briefly describes the context, problem definition and objectives. The technological approaches used to develop the final solution and schedule of activities are also mentioned.

1.1 Context

According to the World Health Organization (WHO), stroke is the second most common cause of mortality worldwide and the main cause of disability (Das et al., 2023). A stroke is defined as a blockage or rupture of arteries or blood vessels in the brain, which results in a shortage of blood supply to the brain and damages brain cells. Stroke frequently results in sequelae, which vary according to the affected part of the brain, even though it is not necessarily fatal. These include dizziness, speech issues, lack of coordination, and visual impairments. Treatments for non-fatal cases vary depending on the underlying reason and typically involve speech therapy and physical rehabilitation (most often repeated) (Rodrigues et al., 2021).

Chronic pain is a common consequence of stroke-related disabilities and a public health question of great importance. It has serious individual consequences and a large socio-economic burden. Chronic pain has reached epidemic proportions worldwide and its economic costs have proven to be excessively high, with some authors even stating that chronic pain could significantly contribute to "the end of the welfare state". It has been shown that pain due to musculoskeletal disorders represents, in European countries, costs of up to 2% of the national gross domestic product per year (Bevan, 2015).

The shoulder joint is vital for performing everyday tasks including lifting, throwing, and reaching. For the shoulder to operate as best it can, there must be coordinated movement between the humerus, scapula, clavicle, thoracic wall, and thoracic spine. Injuries to these anatomical structures can seriously limit shoulder motion. Common shoulder diseases such as osteoarthritis and tendinopathy increase with age and impact a significant proportion of people with 65 years of age or older. Of particular concern are rotator cuff tears, which affect about

25% of adults worldwide and as much as 50% of people with 66 years of age and older. Adults experience upper limb pain at a startlingly high rate (36% of the population) (Cunha et al., 2023).

Shoulder pain accounts for a significant percentage of the prevalence of chronic pain among painful conditions of musculoskeletal origin, it is the 3rd most common pain condition presented in primary health care (Cunha et al., 2023).

1.2 Problem Definition

The need for focused, intensive therapy is always growing due to changes in the world's population and the rise in stroke sufferers. However, the shortage of qualified therapists, especially in the wake of the COVID-19 pandemic, is making it difficult for therapists to meet demand. As a result, it's becoming more and more clear that therapists need the right kind of technical assistance to give patients insightful progress reports (Islam et al., 2023).

Home-based rehabilitation is acknowledged as a substitute or supplementary method to conventional care-unit-based therapy, enabling patients to regain their capacity to conduct daily duties by promoting autonomy and aiding in their recovery (Cunha et al., 2023).

Patients and healthcare organizations can benefit from this strategy in many ways, such as increased therapist availability, shorter travel times, more flexible appointment scheduling, and timely feedback delivery. Because of this, patients who would previously need inpatient care can now receive therapy in an outpatient setting through home-based rehabilitation, where they can actively participate in recommended rehabilitation exercises at home, encouraging adherence to the treatment plan (Cunha et al., 2023).

A problem that arises is that there are no practical and affordable ways to monitor and evaluate the rehabilitation process (Davis et al., 2022).

1.3 Objectives

The main objective of this study is to answer the following question: is it possible to reliably measure the quality of home-based shoulder rehabilitation exercises using a Machine Learning (ML) tool, compared to medical evaluation systems? This objective will be accomplished by creating a ML model that will be able to evaluate the quality of the exercises prescribed by the healthcare professional and performed by the patient.

At the end of this project, it is expected that the following results will have been obtained:

- Literature review on existing approaches to upper limb rehabilitation using machine learning.

- Develop a machine learning model to evaluate shoulder exercise rehabilitation by comparing the patient's exercises with those of a healthcare professional.
- Design an architecture that allows an intelligent agent to assist the rehabilitation process of patients, based on an assessment of the quality of rehabilitation exercises.
- Implement and test the components of the planned solution to obtain a system to be used by patients and healthcare professionals. The system will allow healthcare professionals to prescribe exercises and to evaluate the quality of rehabilitation exercises.

1.4 Methodology

This project focuses on the development of a machine learning model that can evaluate the effectiveness of home-based rehabilitation exercises. Was conducted in a partnership with physiotherapists from Center for Rehabilitation Research from Politécnico do Porto, that provided the dataset used as well as their expertise in the matter. The CRISP-DM methodology (Wirth and Hipp, 2000) will be employed for this purpose. In this study, a dataset, composed of recordings of healthcare professionals and patients executing rehabilitation exercises, will be used. The MediaPipe Pose Landmarker task is a tool that allows the detection of landmarks of human bodies in videos. This tool can be used to identify key body locations, analyse posture, and categorize movements. The tool outputs body pose landmarks in image coordinates in 2 or 3-dimensional world coordinates.

1.5 Document Structure

The document organized by the following items:

- Introduction - Describes the topic of this dissertation, including its context, the problem definition, the objectives, the methodology and the structure of this document.
- State of the Art - Introduces rehabilitation and home-based rehabilitation. Then presents the ML, exploring some algorithms and the other subjects relevant for the project. Finally, literature examples of the application of ML home-based rehabilitation.
- Machine Learning model for shoulder rehabilitation exercises assessment – Gives an overview of the data used, as well as the pre-processing tools and data transformations performed, and the process of developing the ML models.
- Problem Analysis – Chapter where the system to be developed requirements are analysed.

- Solution Design – Chapter where the components of the application, API, database and mobile application will be conceptualized.
- Solution Implementation – Chapter in which the components designed are implemented and the technologies chosen are justified.
- Conclusions - Presents the conclusions about the work carried out, highlighting what objectives were achieved and outlining the subsequent actions that should be taken.

2 State of the art

This chapter exposes the fundamental concepts for a better understanding of the work developed: rehabilitation and ML. Finally, some examples of ML applications in home-based rehabilitation.

2.1 Rehabilitation

This section briefly describes what rehabilitation is and the most relevant types of rehabilitation. It will also be described, in more detail, what is home-based rehabilitation since it is the focus of this project.

Rehabilitation is a broad term that refers to the process of helping individuals recover from illnesses, injuries, or surgeries to regain or improve their physical, mental, or cognitive abilities. It involves a multidisciplinary approach to address various aspects of a person's well-being and functionality (Sunder, 2008).

The primary goal of rehabilitation is to restore individuals to their optimal level of functioning and independence. This may involve regaining physical strength, improving mobility, enhancing cognitive skills, and addressing emotional well-being (Sunder, 2008).

It is the third phase of medical care, after preventive and curative. The initial stage of preventive medicine involves preventing the interaction of an agent, host and environment to prevent the development of a disease. Trying to cure the disease is the main goal of the second phase of curative medicine. Most physicians pursue curative care. However, there are several illnesses, such as poliomyelitis, for which the disease-causing agent has been removed from the host but residual symptoms, such as paralysis, continue to exist. Other illnesses, such as rheumatoid arthritis, have no known cure. As a result, a third phase, rehabilitation, is required. This phase involves social as well as medical responsibilities.

There are multiple types of rehabilitation, of which the following stand out:

- **Physical Rehabilitation:** Focuses on restoring physical functions, such as mobility, strength and flexibility. It is commonly employed after injuries, surgeries, or conditions affecting the musculoskeletal system.
- **Occupational Rehabilitation:** Aims to help individuals regain the skills necessary for daily living and work-related activities. Occupational therapists assist in improving fine motor skills, coordination and adaptive techniques.
- **Cognitive Rehabilitation:** Targets cognitive functions, including memory, attention, problem-solving, and language skills. It is often utilized for individuals recovering from brain injuries or neurological conditions.
- **Psychological Rehabilitation:** Addresses emotional and psychological well-being, assisting individuals in coping with the mental challenges associated with illness, injury, or disability (Sunder, 2008).

2.1.1 Home-based rehabilitation

Access to rehabilitation facilities and professionals has become increasingly harder to get. Home-based rehabilitation takes place in the individual's home environment rather than in a clinical or institutional setting. This approach allows individuals to receive rehabilitation services tailored to their needs within the familiar surroundings of their own homes (Cunha et al., 2023).

The biggest challenge to home-based rehabilitation is having the ability to provide the equivalent services to patients that are provided in rehabilitation facilities. Advances in e-health and other remote monitoring technologies have made it easier to move various devices from hospital settings to home-based paradigms in recent years. These gadgets, which are frequently operated by non-health professionals, are now made to be more ergonomic, user-friendly, and small, meeting the needs of a wide range of users (Cunha et al., 2023).

In the specific field of shoulder rehabilitation, the focus of this project, numerous technological solutions are under development. These technologies can be divided into two main categories: physical category and virtual category.

Figure 1 presents an overview of these key technologies.

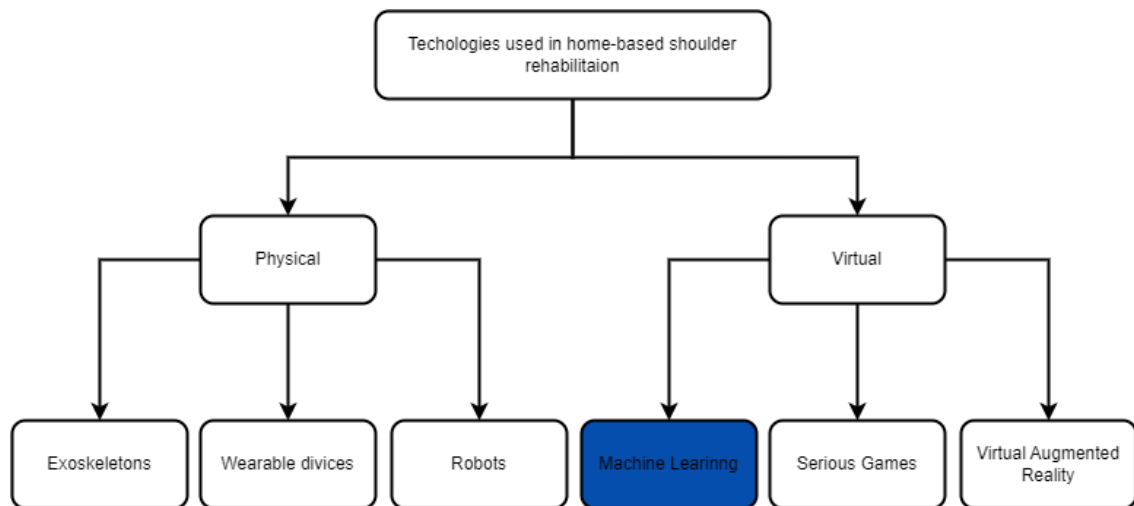


Figure 1: Home-based shoulder rehabilitation technologies, adapted from (Cunha et al., 2023)

On the physical side of rehabilitation are technologies that leverage robotics and mechanical devices to enhance treatment for muscular, neuromuscular, and osseous conditions. Core components, including wearable devices, robots, and exoskeletons, offer personalized interventions and real-time feedback, with wearables crucial in shoulder rehabilitation for assessing motor abilities (Cunha et al., 2023).

Wearable sensors, integral to shoulder rehabilitation, assess motor abilities through kinematics data extraction. Various devices, such as upper body orthotics, soft-orthotic systems, and motion sensors, aid home-based exercises (Cunha et al., 2023).

Advancements in robotics and orthotics positively impact telerehabilitation. Robotic devices provide real-time feedback, enhancing patient understanding and correction of movements. They offer physical assistance, boosting patient motivation and adherence to therapy. Telerehabilitation, facilitated by telecommunication technologies and robotics, enables remote therapy with appropriate safety measures (Cunha et al., 2023).

Exoskeletons, wearable robotic devices, facilitate and enhance physical activities, especially in patients with reduced mobility. Categorized as active or passive, they contribute to upper limb rehabilitation by sustaining proper posture and providing strength to the shoulder. Inputs, such as electromyography signals, ensure effective functionality (Cunha et al., 2023).

In the virtual side of rehabilitation are technologies powered by informatics, introducing immersive experiences, personalized interventions, and intelligent data analysis. Machine learning stands out as a transformative force in developing intelligent rehabilitation systems, especially in the context of remote access and continuous patient monitoring. Virtual and augmented reality, along with serious games, offer innovative approaches for motor recovery and patient engagement in rehabilitation practices (Cunha et al., 2023).

Virtual and augmented reality technologies create controllable environments for rehabilitation. They simulate real-world scenarios, providing engaging platforms for tasks like balance training and motor skill development. These technologies contribute to innovative methods for motor recovery, emphasizing patient engagement (Cunha et al., 2023).

Serious games, integrated into clinical practice, motivate patients to complete exercises. With applications on gaming consoles like Kinect®, Wii®, and PlayStation Move®, these games tailor the environment to specific disease conditions. Challenges, rewards, and progression elements enhance patient engagement, potentially boosting the effectiveness of shoulder rehabilitation programs (Cunha et al., 2023).

Machine learning research has accelerated in the past years using it for classification, prediction, and treatment planning, to develop intelligent rehabilitation systems that automatically adapt to patients' needs and offer tailored interventions based on individual features. Noteworthy examples, such as the iJoint telerehabilitation system, showcase the diverse range of techniques employed in this context. The adoption of rehabilitation systems integrating machine learning has accelerated, particularly in response to the global COVID-19 outbreak. These systems provide advantages such as remote access to patients' exercise information, enabling health professionals to monitor progress without direct physical contact. This remote supervision capability ensures continuous care and allows professionals to simultaneously oversee multiple patients, significantly enhancing productivity. Overall, the integration of machine learning and remote monitoring is transforming healthcare professionals' approach to delivering care, especially during challenging times (Krittanawong et al., 2021).

2.2 Machine Learning

This section briefly describes what is machine learning, as these are the specific kind of algorithms that will be used in the context of this project. The algorithms covered in this section are the ones that were most frequently found in related works, described in section 2.6.

Machine learning is a branch of artificial intelligence that focuses on the development of algorithms and models capable of learning from data and making predictions or decisions without being explicitly programmed. It involves the use of statistical techniques and computational methods to enable machines to improve their performance on a specific task over time through experience. Machine learning encompasses a range of approaches, including supervised learning, unsupervised learning, and reinforcement learning, each suited to different types of problems. The ultimate goal is to enable machines to automatically recognize patterns, generalize from examples, and adapt to new information, facilitating intelligent decision-making and problem-solving in diverse domains such as image recognition, natural language processing, and autonomous systems (Goodfellow et al., 2016).

2.2.1 Supervised Algorithms

Supervised learning comprises applying a mapping to predict the outputs for unseen data by learning a set of input variables, X , and an output variable, Y (Cord and Cunningham, 2008).

There are two types of supervised algorithms:

Classification: assumes discrete values from a non-ordered dataset. They are used to forecast or classify distinct values such as Hot or Cold, Real or False, Male or Female, etc.

Regression: assumes infinite values from an ordered dataset. they are used to determine continuous values such as price, income, age, etc.

Figure 2 shows the difference between these two types of algorithms.

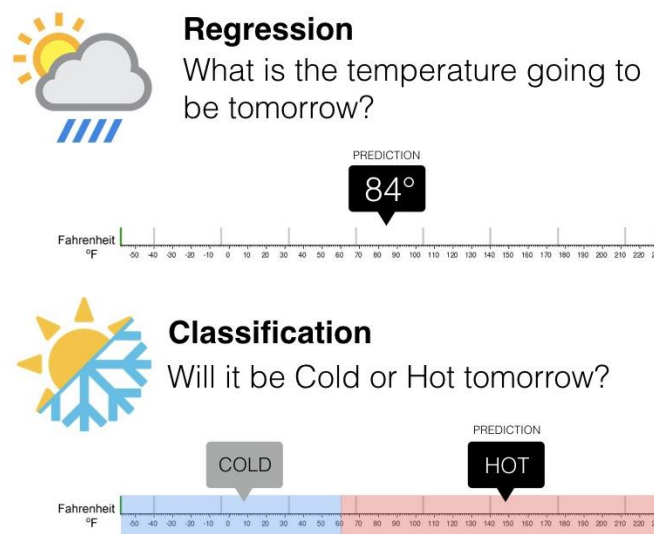


Figure 2: Classification vs Regression Algorithms (“Regression vs. Classification in Machine Learning,” 2021)

In the next sub-sections are presented some of the most used supervised algorithms, and the ones that were found the most in related works, described in section 2.6.

2.2.1.1 Support Vector Machines

Support Vector Machines (SVM) is a type of learning algorithm developed in the 1990s (Cord and Cunningham, 2008). It separates data points using hyperplanes. They work by finding the hyperplane that maximizes the margin between the two classes of data points. Imagine having two classes of points on a graph, and SVM aims to draw the most effective line or boundary between them (Cortes and Vapnik, 1995).

SVM works to maximize the distance between this hyperplane and the nearest points from each class. These critical points that are closest to the boundary are called support vectors. By

maximizing this margin, SVM creates a robust and well-defined separation between classes (Tan et al., 2019).

SVM can handle scenarios where a simple straight line isn't enough to separate classes. It employs a mathematical trick called the kernel trick to map the data into a higher-dimensional space, allowing for the discovery of more complex boundaries (Tan et al., 2019).

The algorithm involves adjusting parameters, like the C parameter, which balances the need for a smooth boundary versus accurate classification of training points. Additionally, SVM can be extended to handle multiple classes and can be applied to regression tasks as well (Tan et al., 2019).

In essence, SVM searches for the optimal hyperplane that best distinguishes different classes.

Figure 3 shows a SVM representation.

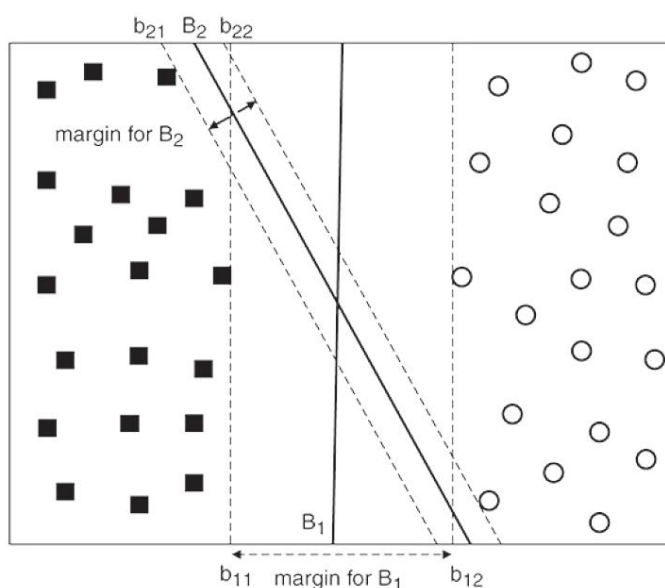


Figure 3: Support Vector Machine (Tan et al., 2019)

2.2.1.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are strong classification technique and may learn extremely complicated and nonlinear decision boundaries just from the data. Since they consistently exceed other classification models (and occasionally even human performance), they have become widely accepted in a variety of applications, including speech, language processing, and vision (Tan et al., 2019).

The study of artificial neural networks was inspired by attempts to emulate biological neural systems. They consist of interconnected nodes called neurons organized into layers: the input layer, hidden layers, and output layer. Each connection between neurons has a weight, and

during training these weights are adjusted to enable the network to learn patterns from input data (Goodfellow et al., 2016).

In the network, neurons apply an activation function to the weighted sum of their inputs, introducing non-linearity for learning complex patterns (Goodfellow et al., 2016).

The process of passing input data through the network, layer by layer, is called feedforward. The output is compared to true values using a loss function, measuring the difference. The goal during training is to minimize this loss (Tan et al., 2019).

Backpropagation is a key process where the network updates weights to minimize the loss. It involves calculating the gradient of the loss concerning the weights and adjusting them accordingly. Training occurs iteratively on the dataset until the network performs satisfactorily. Optimization algorithms like stochastic gradient descent facilitate this process (Goodfellow et al., 2016).

ANNs come in various architectures, including feedforward networks, Convolutional Neural Networks (CNNs) for images, and Recurrent Neural Networks (RNNs) for sequential data.

Figure 4 shows a representation of an ANN.

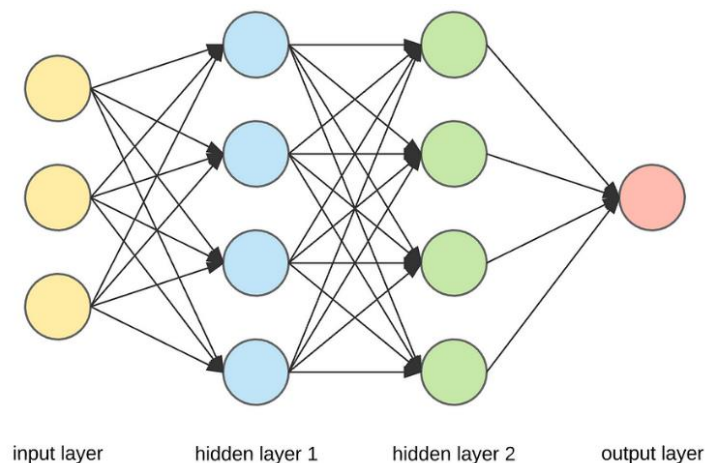


Figure 4: Artificial Neural Network (Dertat, 2017)

2.2.1.3 K-Nearest Neighbours

K-Nearest Neighbours (KNN) is one of the simplest yet powerful supervised ML algorithms. The underlying principle is quite straightforward, examples are classified based on the class of their nearest neighbours (Cord and Cunningham, 2008).

Unlike algorithms previously described, KNN is not trained beforehand, it runs at the time of execution, memorizing the entire training dataset, to find the query output. There's no point in training the model earlier as an input of the model is training data, a hyperparameter (k), and a query point (x) (Goodfellow et al., 2016).

When it comes to making predictions for a new query point, K-NN calculates the distance between that point and all the points in the training data. The most common distance measure is Euclidean distance.

Once distances are calculated, the algorithm identifies the K-nearest neighbours of the new data point. For classification tasks, it assigns the class label that is most common among these neighbours. For regression tasks, it averages the target values of the K-nearest neighbours (Goodfellow et al., 2016).

Figure 5 shows a representation of K-Nearest Neighbours.

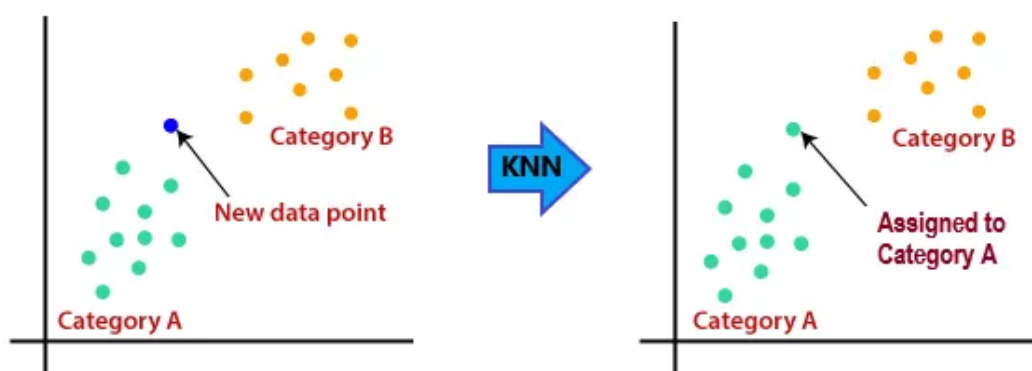


Figure 5: K-Nearest Neighbours (Tech-AI-Math, 2023)

2.2.1.4 Gradient Boosting

Gradient Boosting is a family of algorithms that can be used for both classification and regression tasks. It is an ensemble learning method that combines the predictions of other techniques, typically decision trees, to create a better model. Gradient Boosting builds trees sequentially, with each tree attempting to correct the errors of the previous ones (Natekin and Knoll, 2013).

Gradient Boosting is called "Gradient" because it minimizes the loss function (error) by using gradient descent optimization. In each iteration, the algorithm adjusts the parameters of the method used to minimize the remaining errors.

Common implementations of Gradient Boosting include:

- Gradient Boosted Decision Trees (GBDT): The weak learners are typically decision trees.

- XGBoost (Extreme Gradient Boosting): An optimized and efficient implementation of Gradient Boosting.
- LightGBM: Another efficient and distributed Gradient Boosting framework.
- CatBoost: A Gradient Boosting library that handles categorical features well.

Gradient Boosting is powerful and often produces accurate models, but it can be computationally expensive and prone to overfitting if not properly tuned. Regularization techniques and hyperparameter tuning are important for achieving optimal performance (Tan et al., 2019).

Figure 6 shows a representation of Gradient Boosting using decision trees.

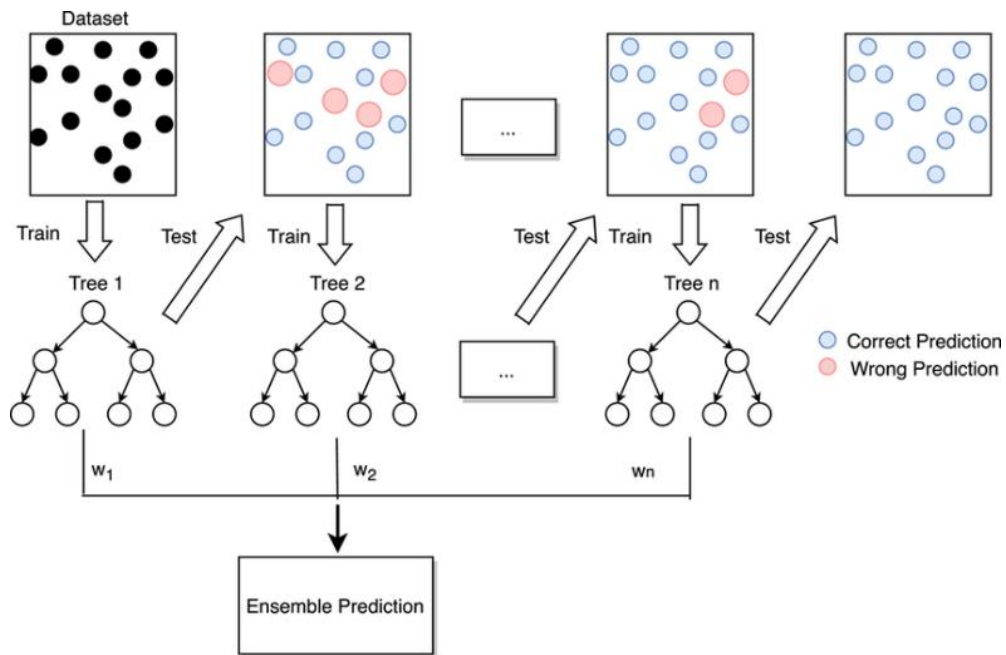


Figure 6: Gradient Boosting (“Flow diagram of gradient boosting machine learning method. The ensemble...,” n.d.)

2.2.2 Unsupervised Algorithms

Unsupervised machine learning is a type of machine learning where the model is trained on unlabelled data, meaning that the algorithm learns patterns, structures or relationships from the data without predefined outputs. Unlike supervised learning, where the goal is to predict a target variable based on input-output pairs, unsupervised learning aims to understand the intrinsic structure of the data. It is primarily used for clustering, dimensionality reduction and anomaly detection (Naeem et al., 2023) (Mahesh, 2020).

There are three types of unsupervised algorithms:

Clustering algorithms group data into clusters by identifying patterns or structures within the dataset. These algorithms determine clusters based on similarity, aiming to separate data points into distinct groups where points within the same group share common characteristics, while those in different groups exhibit greater dissimilarity. Approaches may involve optimizing certain metrics, such as minimizing variance within clusters or identifying regions with high data density.

Dimensionality reduction algorithms are used to simplify large and complex datasets by reducing the number of variables or features. The goal is to retain the most significant information from the original dataset while transforming it into a lower-dimensional form. This can improve the efficiency of machine learning models and facilitate data visualization by focusing on the most important relationships between variables.

Anomaly detection: focuses on uncovering hidden relationships and patterns between variables within large datasets. It seeks to identify frequent co-occurrences or dependencies among variables, often used in contexts where understanding interactions between items or attributes is critical, such as identifying common patterns of behaviour or transactions.

2.2.2.1 Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is a probabilistic model that assumes data is generated from a mixture of several Gaussian distributions, each representing a different cluster or component in the data. GMM is a flexible approach that models complex data by combining multiple Gaussian distributions, allowing for the representation of clusters with varying shapes, sizes, and orientations. Unlike simpler clustering methods like K-Means, which assume uniform cluster shapes, GMM can handle more complex cluster structures by leveraging covariance matrices for each Gaussian component (Reynolds, 2009).

The key feature of GMM is its use of soft clustering, meaning each data point has a probability of belonging to each Gaussian component rather than being strictly assigned to one cluster. This is achieved by calculating the likelihood of each point under the Gaussian components and assigning probabilities based on those likelihoods. The parameters of the GMM, such as the means, covariances, and mixing coefficients, are typically estimated using the Expectation-Maximization (EM) algorithm. The EM algorithm iteratively refines these parameters by maximizing the likelihood of the observed data under the model (Bishop, 2006)

GMM is particularly useful in applications such as image segmentation, anomaly detection, and speaker recognition, where complex and overlapping clusters are common (Reynolds, 2009). However, the model requires the number of components to be specified beforehand, which can be a limitation if this value is not known. Additionally, GMM is sensitive to initialization and may converge to local optima, depending on the starting points of the parameters (McLachlan & Peel, 2000). Despite these challenges, its flexibility and ability to model non-spherical clusters make GMM a valuable tool in data analysis (Bishop, 2006).

2.2.2.2 Principal Component Analysis

Principal Component Analysis (PCA) is a technique used for reducing the number of variables in a dataset while retaining the most important information. It does this by transforming the original variables into a new set of variables called principal components. These components are ordered by how much of the data's variability they explain, with the first component capturing the most variance, and each subsequent component explaining progressively less. This transformation helps simplify complex datasets without losing too much information (Jolliffe and Cadima, 2016).

PCA works by calculating the covariance matrix of the data, followed by finding the eigenvectors and eigenvalues of this matrix. The eigenvectors indicate the directions of the principal components, while the eigenvalues show how much variance each component captures. By projecting the data onto a smaller number of principal components, PCA reduces the dimensionality of the data, making it easier to work with. This is especially useful for tasks like data visualization and improving the performance of machine learning algorithms (Abdi and Williams, 2010).

While PCA is a powerful tool, it has some limitations. One challenge is that it assumes that the directions of maximum variance are the most important for understanding the data, which might not always be the case, especially for complex or non-linear relationships. Moreover, PCA can be sensitive to the scale of the data, so it's important to standardize the dataset before applying the technique to ensure meaningful results (Shlens, 2014).

2.3 Qualisys Track Manager

Qualisys Track Manager (QTM) is an advanced motion capture software used extensively in biomechanics, including shoulder rehabilitation, to track and analyse human movement. Developed by Qualisys, QTM allows users to capture 2D, 3D, and six degrees of freedom (6DOF) data in real time, making it a highly precise and versatile tool for monitoring shoulder rehabilitation exercises. The software integrates seamlessly with a range of hardware, including force plates, electromyography systems, and high-speed cameras, which enables healthcare professionals to gather detailed kinematic and kinetic data. This makes QTM ideal for assessing shoulder movement patterns and ensuring patients are performing rehabilitation exercises correctly ("Qualisys Track Manager (QTM)," n.d.).

In shoulder rehabilitation, the accuracy and real-time feedback provided by QTM can significantly enhance the assessment of range of motion, joint angles, and movement irregularities. It uses passive and active markers to track movements with high precision, which is crucial in diagnosing improper movement patterns or asymmetries that could hinder recovery. QTM's robust inverse kinematic solvers allow clinicians to visualize skeletal data during movement, aiding in the assessment of how well the shoulder joint moves in various rehabilitation exercises ("Qualisys Track Manager (QTM)," n.d.).

Another key feature of QTM in shoulder rehabilitation is its integration with external systems like EMG devices, enabling the simultaneous collection of muscle activity data alongside motion tracking. This provides a comprehensive understanding of how muscles interact with joint movements during rehabilitation exercises. Additionally, its built-in gap-filling algorithms help manage missing data, which can occur if markers are momentarily obscured during complex shoulder movements (“Qualisys Track Manager (QTM),” n.d.).

2.4 MediaPipe

This section explains what MediaPipe is and what will be used for in the context of this project.

MediaPipe is an open-source framework developed by Google that provides tools and components for building applications with perceptual computing features, primarily focused on computer vision and machine learning. It offers a range of pre-built solutions for tasks like face detection, hand tracking, pose estimation, and more (“MediaPipe,” n.d.).

MediaPipe Pose is a component of the MediaPipe framework that focuses on human pose estimation. Pose estimation involves identifying and tracking key points on a person's body, such as joints and limbs, in images or video frames (Kim et al., 2023).

Figure 7 shows the MediaPipe Pose body landmarks definition.

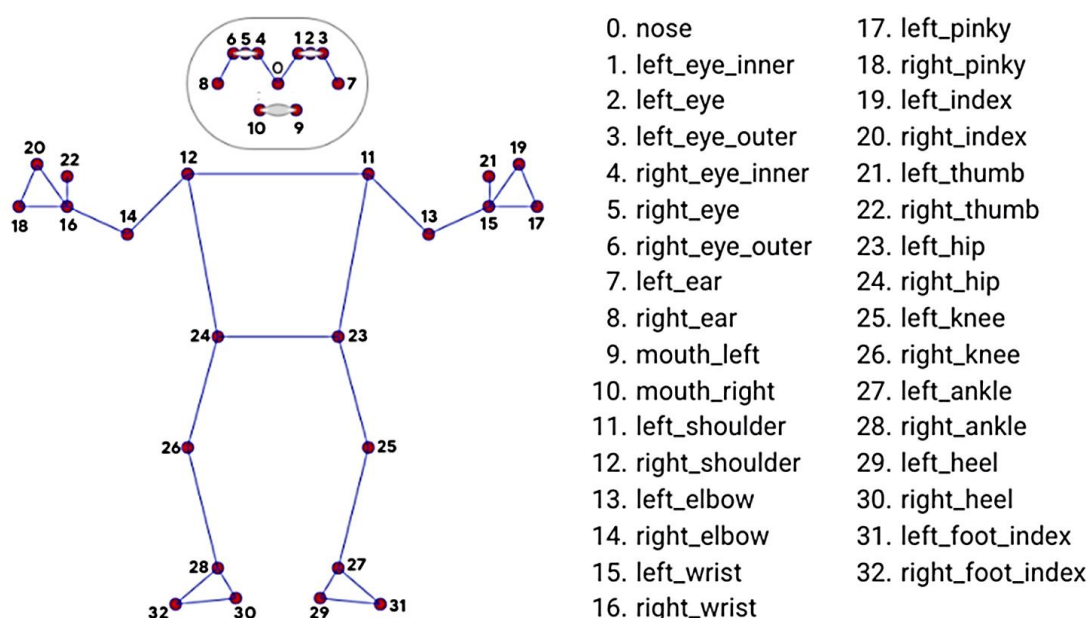


Figure 7: Definition of landmarks in MediaPipe Pose

In this project, MediaPipe Pose will be used to map the patient's movements while performing the rehabilitation exercises. This will be achieved by continuously estimating pose landmarks, on the patients' recorded exercises, representing key points on patients' bodies, across

successive frames. This process involves tracking the temporal sequence of pose information and capturing the dynamic changes in body posture and movement throughout the video.

This process will also be executed in the recordings of healthcare professionals executing the same rehabilitation exercises.

2.5 Distance Time Warping

Dynamic Time Warping (DTW) is an algorithm used to measure the similarity between two time series that may differ in speed or length. Unlike traditional distance metrics, DTW can handle variations in time by aligning sequences non-linearly, allowing it to match sequences that are out of sync. This makes DTW especially useful in fields such as speech recognition, where two utterances of the same word may vary in length or timing, and in time-series analysis for financial or physiological data (Rakthanmanon et al., 2012)

The algorithm works by constructing a cost matrix between the two sequences, where each element represents the distance between points. DTW then finds an optimal warping path that minimizes the cumulative distance between the sequences, allowing parts of one sequence to stretch or compress to match similar parts of the other. This flexibility allows DTW to capture similarities even when there are shifts in timing, which is why it is used in applications where alignment is not straightforward (Rakthanmanon et al., 2012).

In video processing, DTW has been applied to tasks like action recognition and video retrieval, where the timing of actions may differ between videos. For example, two videos showing the same activity (such as walking or running) may vary in the speed at which the action occurs. DTW helps by aligning the frames of the videos to match the timing of similar movements, even when the actions happen at different speeds. This makes it a powerful tool for analysing videos with variable timing and has applications in sports analytics, surveillance, and human activity recognition (Tychola et al., 2022).

2.6 Related Works

This section briefly describes other works that use machine learning techniques to aid in the home-based rehabilitation of patients.

In literature, there are many implementations of systems that use machine learning in aiding home-based rehabilitation in very different contexts. The related works selected, and here described, are the ones that were considered to be closer and more interesting to the work made throughout this project.

Rodrigues et al., (2022) presents a methodology for implementing a system that employs RGB cameras and machine learning models, specifically using the MediaPipe framework, to track

and classify human upper and lower limb movements. The proposed system, developed with the Unity game engine, captures motion data through pose estimation algorithms and records a predefined set of movements for training a supervised machine learning model. The MediaPipe framework is chosen for its compatibility with smartphones and support for Android and iOS platforms. This study emphasizes the importance of precise tracking, utilizing datasets with high-precision tracking values generated from specialized cameras, and discusses the application of the developed system in a motor rehabilitation context. The system incorporates an interface, and remote-control solution, and utilizes a non-relational database for storing movement-related metadata and joint data.

The dataset creation involves participants performing four types of movements, recorded with various devices such as smartphones, Kinect, and wearables. The movements are categorized as Coronal, Sagittal, Transverse, and Elbow, involving shoulder and elbow articulations. The study also details the pre-processing steps, including values normalization, fixed sample size adjustments, and visualization of movements. The machine learning model, an ANN, is trained to classify movements with an accuracy of 100% in a four-class dataset and 89% in an eight-class dataset. The results suggest the potential applicability of motion capture data analysis in motor rehabilitation, with plans for future studies to explore the system's limitations and incorporate more complex movements and lower limb tracking.

Islam et al., (2023) proposes a system that aims to facilitate neurorehabilitation exercises for stay-at-home patients using commodity RGB cameras without the need for special sensors or equipment. The system pipeline comprises three main steps: participant's exercise data acquisition, body joint recognition and tracking using the MediaPipe machine-learning framework, and data extraction and formulation for analysis and evaluation. The authors emphasize the use of regular RGB cameras and zoom video conferencing platforms for data acquisition due to the ongoing pandemic, providing participants with detailed instructions. The MediaPipe framework is employed to track body joint movements, and a comprehensive data processing pipeline is described, including the calculation of angles and validation checks for exercise accuracy.

In the evaluation and analysis section, the system's accuracy is assessed based on different factors such as distance from the camera, lighting conditions, and user groups. Results show that the system's measured angles align well with the ground truth angles, demonstrating its potential for remote monitoring of neurorehabilitation exercises. The study identifies challenges, including variations in participant-side camera resolution and differences in body movement understanding between IT and medical staff. The proposed system offers a promising solution for recording and analysing patients' exercise sessions remotely, with plans to extend its capabilities to incorporate additional exercises and diverse user scenarios.

Das et al., (2023) presents a project of a mobile application aimed at aiding post-stroke rehabilitation patients, with a specific emphasis on upper limb exercises. The application utilizes pose-tracking joint line drawing, employing a pose net model to detect key points and track the position of body parts in real time. The training model, based on a CNN architecture, classifies

images into three categories: normal position, elbow flexion, and elbow extension. Training involves 2500 images, with an 85% training data percentage and 15% test data percentage over 150 epochs. The resulting mobile application demonstrates high accuracy in classifying exercise poses, as evidenced by a loss reduction from 2,5% to less than 0,5% and a testing accuracy approaching 100%. The confusion matrix further validates the model's performance, showing 100% accuracy across all three pose classifications. The authors discuss the successful deployment of the trained AI model as a home rehabilitation mobile application and present promising results in a laboratory environment, encouraging further testing with physically fit individuals before clinical applications.

The research proposes a comprehensive plan for upper-limb home rehabilitation for stroke patients through the development of an AI-based mobile application. The application exhibits significant potential for improving motor skills in post-stroke patients, enabling correct exercise performance at home without the need for sophisticated devices. The study acknowledges the importance of tailoring exercise regimens to different disabilities and outlines future work, aiming to enhance AI techniques for lower limb rehabilitation and other relevant areas. The authors emphasize the potential societal benefits of advancing stroke rehabilitation techniques and align their work with sustainable development goals related to industry innovation, infrastructure, and good health and well-being.

As shown in these three works, analysing videos of patients performing rehabilitation exercises and using the information gathered to train machine learning models shows very good results.

This way it will be used a similar approach with the objective of evaluating the efficiency of home-based rehabilitation exercises, as proposed in this project.

3 Machine Learning model for shoulder rehabilitation exercises assessment

This chapter gives a characterisation of the dataset used in the development of the ML model. In addition, provides an overview of the two different approaches taken to answer the research question of this thesis and a comprehensive analysis of the results obtained.

3.1 Dataset characterization

The dataset used in this project consists of ninety-two videos, ninety of which were provided by fifteen different patients and the remaining two are performed by healthcare professionals and will be used as reference exercises. Each of the fifteen patients performed two different types of physiotherapy exercises, performing three repetitions of each exercise. The videos were recorded in high definition to ensure the clarity and precision of the movements.

The data was collected in a controlled environment, where each patient was instructed to perform two specific types of exercises. These were selected based on their relevance and frequency in common rehabilitation programs, ensuring that the movements captured represent typical physiotherapy practices.

The videos were recorded using an iPhone 14, with the camera positioned two meters away from the participants. The camera was oriented vertically, capturing the front view of the participants' bodies, who were instructed to perform the exercises facing the camera.

3.2 Ground truth

The exercise video recordings, from all 15 patients, were previously subjected to a QTM software designed for motion capture and analysis, that enables detailed analysis of movement.

With this software, all recordings were given a score between 0-100 % regarding how close they were to corresponding healthcare professional's recording, for three body segments, trunk, head and shoulder.

The results of QTM were used as a base line for what the ML model should return as a result. The exercises were classified in good or bad based on the average score, below 80 % the exercise was classified as bad and above 80 % as good. Table 1 shows these results. The values that are in red were identified as outliers.

Table 1: Results provided by QTM software

ID	Exercise	Head	Trunk	Shoulder	Average	Classification
ID1	Diagonal	54,3	77,9	99,9	77,4	Bad
ID2	Diagonal	71,4	80,4	97,5	83,1	Good
ID3	Diagonal	95,8	95,7	87,2	92,9	Good
ID4	Diagonal	90,7	92,4	98,1	93,7	Good
ID5	Diagonal	57,5	79,0	99,9	78,8	Bad
ID6	Diagonal	81,1	59,1	99,9	80,0	Good
ID7	Diagonal	74,0	92,0	98,0	88,0	Good
ID8	Diagonal	67,1	75,7	97,3	80,0	Bad
ID9	Diagonal	85,8	94,2	94,6	91,5	Good
ID10	Diagonal	44,2	79,9	92,2	72,1	Bad
ID11	Diagonal	79,1	95,6	97,5	90,7	Good
ID12	Diagonal	83,0	83,2	95,5	87,3	Good
ID13	Diagonal	70,1	64,6	98,8	77,9	Bad
ID14	Diagonal	78,1	76,5	98,1	84,2	Good
ID15	Diagonal	9,5	65,0	92,4	55,6	Bad
ID1	Rotation	50,7	94,6	99,4	81,6	Good
ID2	Rotation	77,9	88,8	96,0	87,6	Good
ID3	Rotation	87,5	98,4	95,8	93,9	Good
ID4	Rotation	94,9	98,8	96,8	96,8	Good
ID5	Rotation	14,7	88,4	58,9	54,0	Bad
ID6	Rotation	86,1	99,3	83,0	89,5	Good
ID7	Rotation	84,3	98,7	94,3	92,4	Good
ID8	Rotation	94,0	99,6	90,5	94,7	Good
ID9	Rotation	78,7	95,4	61,2	78,4	Bad
ID10	Rotation	76,5	83,0	71,8	77,1	Bad
ID11	Rotation	93,0	91,5	57,2	80,6	Good
ID12	Rotation	76,9	82,9	96,8	85,5	Good
ID13	Rotation	62,8	95,7	99,2	85,9	Good
ID14	Rotation	-178,4	98,2	99,5	6,4	Bad
ID15	Rotation	-107,2	98,8	87,3	26,3	Bad

3.3 Data Pre-processing

The initial phase in developing the ML model involved processing the videos of patients and healthcare professionals performing the rehabilitation exercises. Video processing plays a fundamental role in the pipeline, as it is through this medium that is captured the essential movement information required for evaluation.

The initial step in processing the exercise videos involved breaking down the videos into individual frames. This step is critical, as it allowed for the detailed analysis of the motion sequences. In this project, OpenCV (Open Source Computer Vision Library) (“OpenCV,” n.d.) was used to extract frames from the videos. OpenCV is a highly optimized library that supports real-time image and video processing.

The frame extraction process can be outlined in the following steps:

1. Loading the video file: Using OpenCV's function, the video file was loaded, and basic information such as the frame rate and total number of frames was retrieved.
2. Reading frames: Each frame was read one by one using a loop, and these frames were stored in a memory as array.
3. Key point extraction: Inside the loop, the keypoints were extracted. This extraction is detailed in the next chapter.

3.3.1 Keypoint Extraction

The MediaPipe Pose module detects 33 keypoints across the body, including key joints like the shoulders, elbows, wrists, and hips. The process begins with a raw video input of the patient performing shoulder exercises, which is fed into MediaPipe for pose estimation. MediaPipe's pose tracking algorithm, based on deep learning, detects human poses in real time.

The following steps outline the keypoint extraction process:

1. Pose detection: MediaPipe employs a detector that identifies a human body pose in the image frame. Once a pose is detected, the framework proceeds to track the keypoints across subsequent frames.
2. Keypoint identification: Once the pose is detected, MediaPipe extracts 33 keypoints from the body. These keypoints represent different anatomical landmarks, including the shoulders, elbows, wrists, and other major joints.
3. Tracking and recording keypoints: For each frame in the video, MediaPipe generates a set of 2D coordinates representing the detected keypoints. These coordinates are relative to the video frame and reflect the spatial positioning of the joints. Since the videos were collected from a single camera, 2D keypoints were sufficient to analyse planar motions.
4. Data collection and storage: Since it takes about three minutes to extract the keypoints from a video, these are stored in .xlsx files for each individual video, for later use.

Figure 8 shows an example of a video frame with MediaPipe's keypoints mapped, during the execution of an exercise.

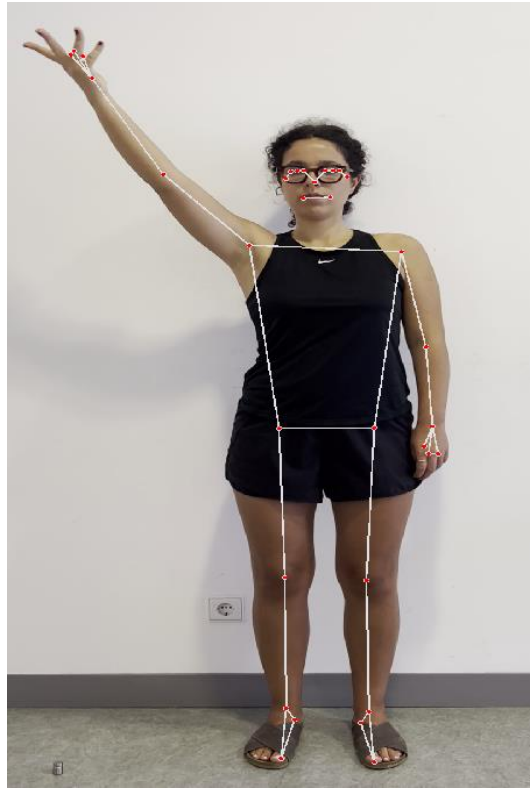


Figure 8: Example of MediaPipe's keypoints mapped on video during exercise

After extracting frames from the videos using OpenCV and then extracting keypoints from those frames using MediaPipe, we ended up with a dataset with one row for each frame and 66 columns, with each pair columns representing a keypoint position (x, y) .

3.4 Modelling

The next phase in developing the ML model investigated two distinct approaches to assess the quality of the patient's exercise execution.

The first approach combined pose estimation and DTW to assess the patients' exercises in comparison with those of the healthcare professionals, while the second approach employed a probabilistic Gaussian Mixture Model.

3.4.1 Distance calculation approach

The first approach to assess the quality of exercise execution was calculate the distance between each corresponding keypoint. Since the objective of this project is to find a simple and fast method to assess the quality of the execution of the rehabilitation exercise, this approach seemed appropriate due to its simplicity.

Like mentioned above, the results given by the QTM software were for three different body segments and not for the body as a whole. This way, prior to the distance calculation, the keypoints extracted from the videos were also segmented into the same three body segments to allow for a comparison of the results.

The Euclidean distance is a measure of the straight-line distance between two points in a multidimensional space. When applied to time series, it calculates the sum of the squared differences between corresponding points (i.e., values at the same time steps) in two sequences. One limitation to this distance function is that it assumes that both time series are of equal length and aligned. One problem that quickly arose was exercises can be performed at different speeds, resulting in different length videos. To test whether this approach could have good results, even though important information could be lost, the longer video was shortened to the length of the shorter video and the distance function was applied to each body segment for each exercise video to calculate the distance.

To try and overcome the problem of the video having different sizes, the FastDTW, a python library that implements the DTW algorithm, was used to calculate the distance between the two videos. This algorithm measures similarity between two sequences, which may vary in speed or length. It's an ideal method for analysing motion trajectories since DTW dynamically aligns the sequences by stretching or compressing them as needed to find the optimal match. The result of the comparison between the two videos is a value of distance. Other problem that was identified with both distance function was, since the distance function returned a distance value, there was no direct way to compare these values to the values returned by the QTM software that returns percentage values. Other limitation of this approach is that, since the distance function result is a cumulative value of all the small differences that exist in each video frame, the bigger the video bigger the distance value.

Figure 9 illustrates the complete distance calculation process.

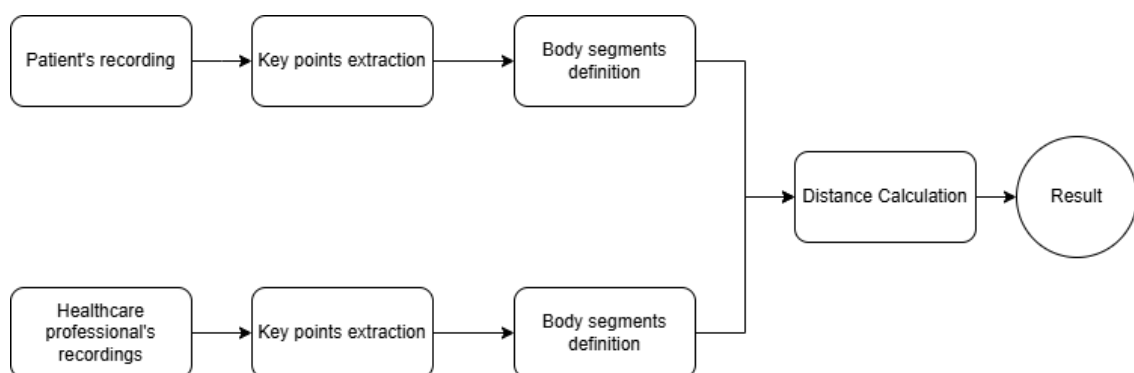


Figure 9: Distance calculation process

Finally, the same process was applied to all patients' videos.

To be able to compare the with the provided classification mentioned in chapter 3.2, the average value for the three repetitions of each patient and, and the average value for the three

body segments were calculated, for both exercises. Figure 10 and Figure 11 show plots of the mean values of distance vs the QTM software result classifications.

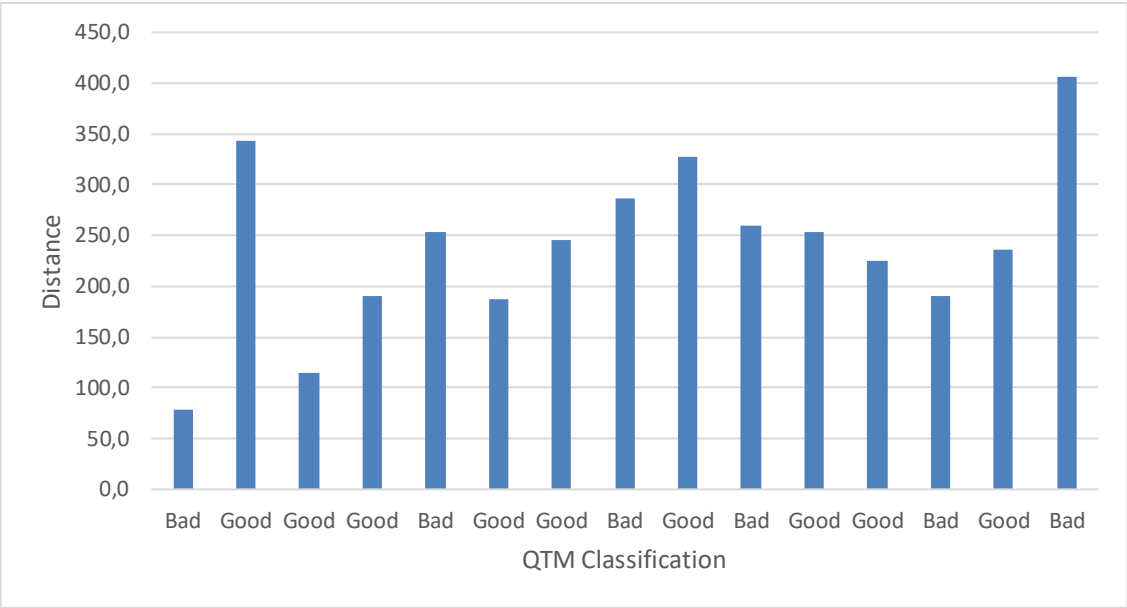


Figure 10: Mean values of distance vs QTM software result classification for diagonal exercise

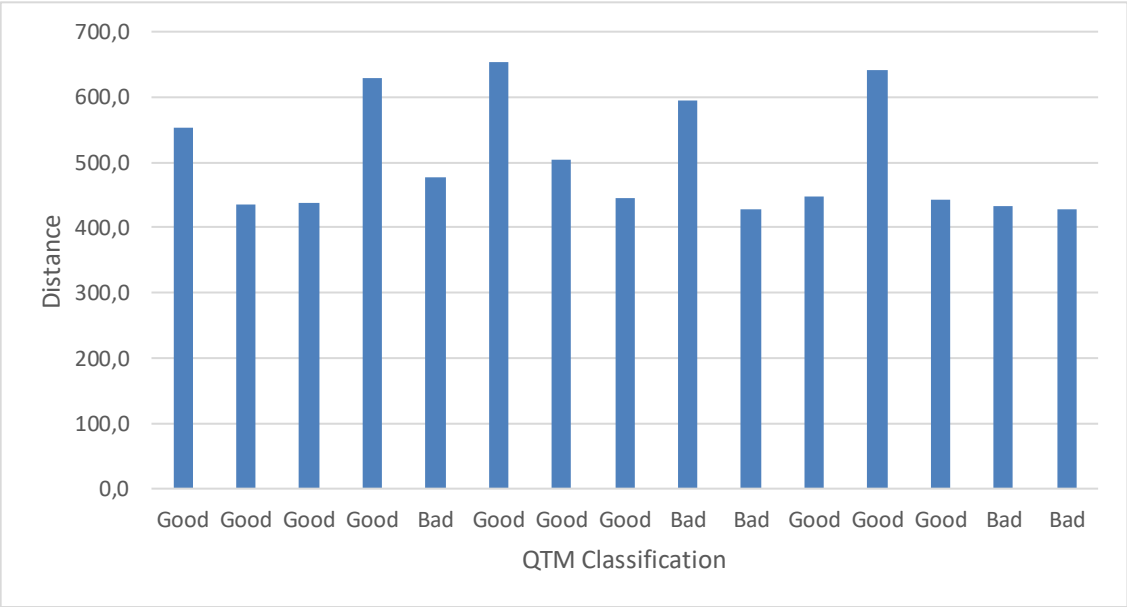


Figure 11: Mean values of distance vs QTM software result classification for rotation exercise

With these results it was possible to find a trend and define value of distance, for each exercise, that maximized the cases in which the model correctly classified the exercise. For the diagonal exercise, this value was 250 and the model correctly classified 10 out of 15 and for the rotation exercise this value was 470 and the model correctly classified 7 out of 13 since 2 exercises had QTM value that were outliers.

Due to all identified limitations and the results obtained not being conclusive, it was decided to explore an alternative approach.

3.4.2 Gaussian Mixture Model approach

The second explored approach to evaluate the patient's movements, a probabilistic approach was used the GMM. This allows for a frame-by-frame evaluation of how well the patient's movements match the healthcare professional's reference movements.

To successfully apply this model, the following steps were taken.

1. Handle missing values and data scaling

Before applying machine learning techniques, it is essential to handle missing values and scale the data. Both datasets were cleaned by dropping any rows containing missing values, ensuring consistency for further processing. Subsequently, the data was standardized using the StandardScaler function from sklearn library ("StandardScaler," n.d.). This standardization technique is used to standardize or normalize data by removing the mean and scaling it to have unit variance. In simpler terms, it transforms the data so that it has an average of 0 and a standard deviation of 1. Standardization is crucial because the keypoint coordinates may have different ranges, and scaling ensures that all features contribute equally to the model.

2. Dimensionality reduction with PCA

Given the high dimensionality of the data (with a large number of keypoints in each frame), PCA was applied to reduce the number of features while preserving as much variance as possible. This step helps in simplifying the analysis and reduces the risk of overfitting. This was achieved using the PCA function from the sklearn library ("PCA," n.d.).

PCA is applied to the scaled healthcare professional's data. The number of components is chosen to match the number of features in the dataset. Figure 12 exemplifies the cumulative explained variance, which shows how much of the total variance is captured by the principal components. This helped determine the number of components to retain for further analysis.

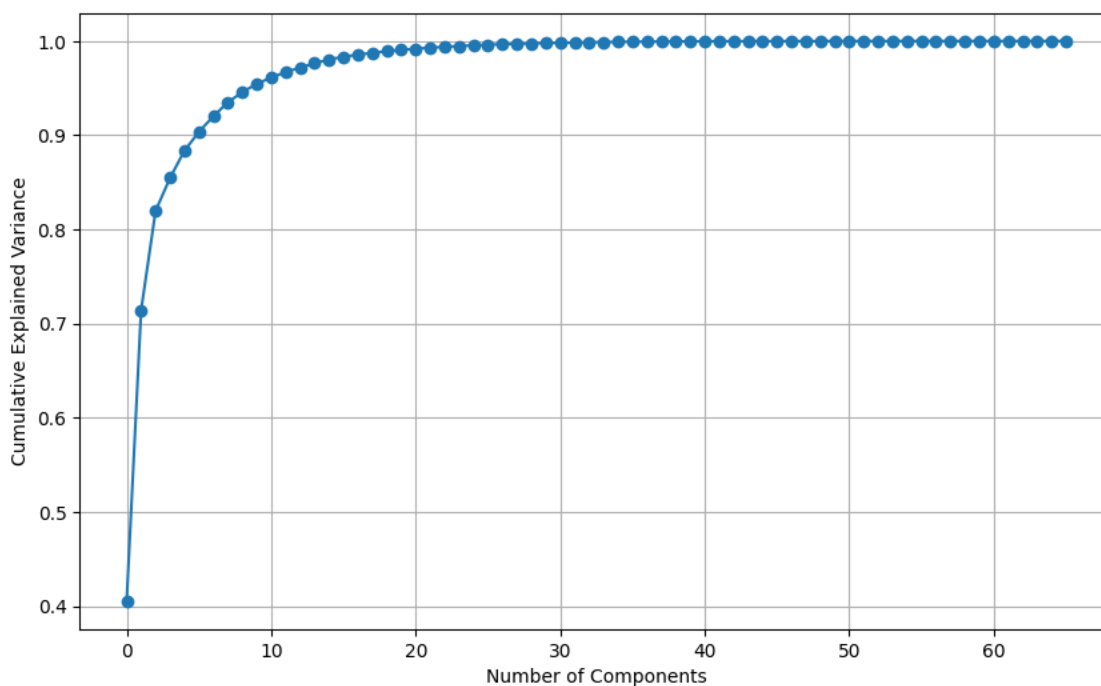


Figure 12: Cumulative explained variance by PCA Components for the diagonal exercise

With ten components the Cumulative Explained Variance was 95,4% for the diagonal exercise and 97,2% for the rotation exercise. This was the number of components used to reduce the dimensions of the dataset.

Once the PCA model was trained on the healthcare professional's data, the same transformation was applied to the patient data to ensure both datasets are represented in the same lower-dimensional space.

3. Probabilistic movement evaluation using GMM

The GMM was fitted, with 10 components, on the healthcare professional's PCA-transformed data to model the density of the keypoints. This was achieved using the GaussianMixture function from the sklearn library ("GaussianMixture," n.d.). The model provides a log-likelihood score for each frame of the patient's movements, which indicates how well the GMM fits the patient's data. Higher log-likelihoods correspond to frames where the patient's movements are more similar to the healthcare professional's movements.

To detect atypical movements, a threshold based on the 3rd percentile of the healthcare professional's log-likelihoods was established. Frames with a log-likelihood below this threshold were considered atypical.

4. Frame Alignment using DTW

Like mentioned in the previous approach, the videos of the patient and the healthcare professional may have different numbers of frames. DTW was again used, this time to align the

corresponding frames in both datasets. The DTW distance between the scaled physiotherapist and patient data was calculated. The path returned by DTW contained pairs of indices, indicating the aligned frames from the physiotherapist and patient sequences. This allowed for a direct comparison between the patient's atypical movements and the corresponding movements of the physiotherapist. Figure 13 shows the plot of an example of one atypical frame and Figure 14 a plot of an example of one atypical frame.

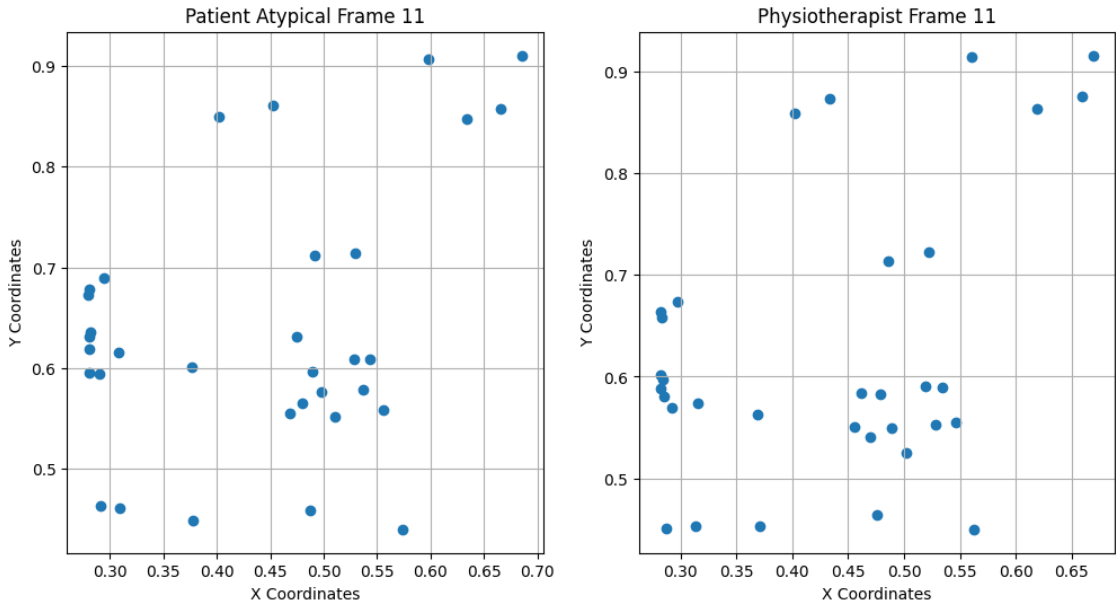


Figure 13: Atypical frame for the patient and corresponding frame for the physiotherapist

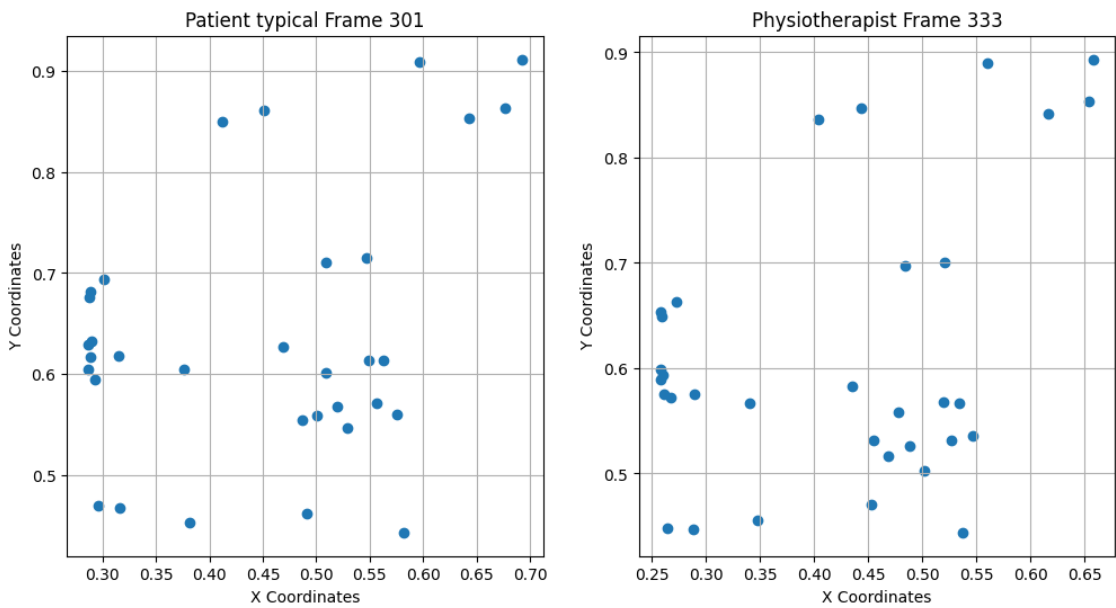


Figure 14: Typical frame for the patient and corresponding frame for the physiotherapist

Finally, the same process was applied to all patients' videos, for both exercises. Figure 15 and Figure 16 show the Mean Log-Likelihood for the 3 repetitions performed by all patients for diagonal and rotation exercises.

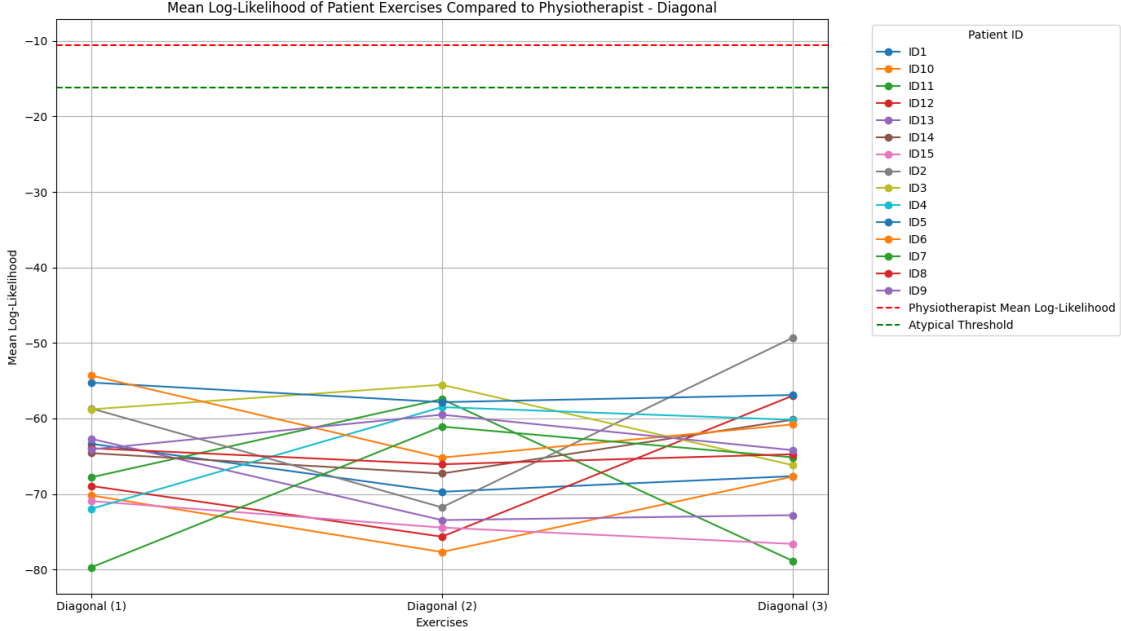


Figure 15: Mean Log-Likelihood for the 3 repetitions of the diagonal exercise for all patients

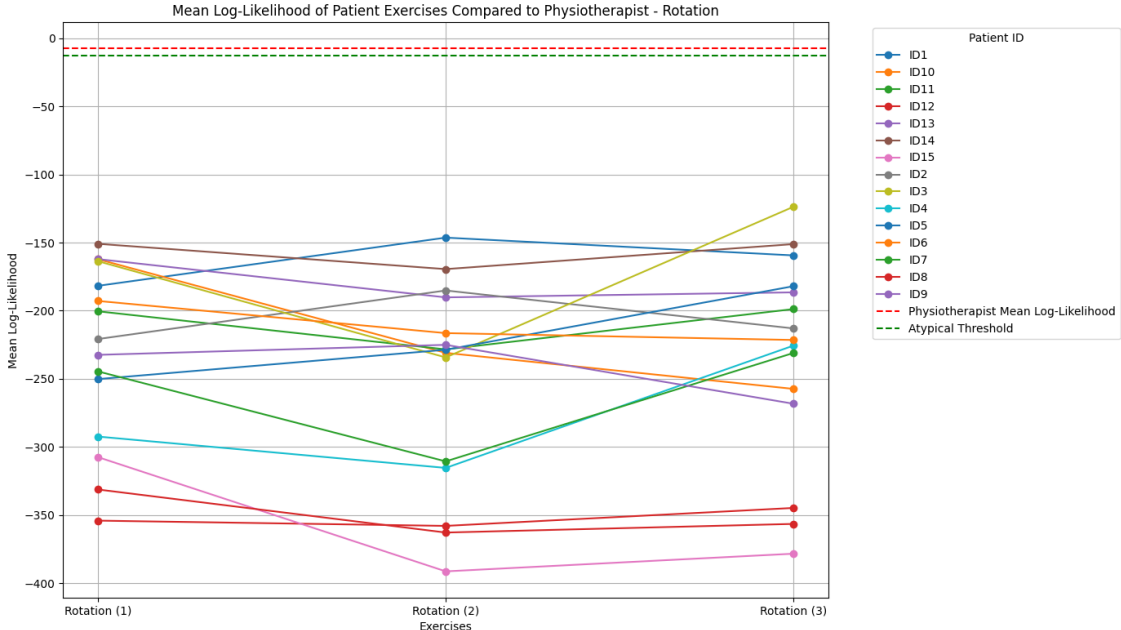


Figure 16: Mean Log-Likelihood for the 3 repetitions of the rotation exercise for all patients

As shown in both figures, the model returned Mean Log-Likelihood values, for all repetitions of both exercise, that are below the minimum threshold, which means that all exercises were

poorly executed. These results are not compatible with the ones delivered by the QTM software shown in Table 1.

3.5 Final remarks

As shown in the previous two sections the results return by both models were not reliable measuring the quality of home-based shoulder rehabilitation exercises. One promising new possible application for the GMM model that was developed is to evaluate the progress of the rehabilitation process. Despite the results were not compatible with the ground truth provided, since the models said that the patients performed worst that they did, the GMM model developed could be used to access the progress of the patients from one session to another. Figure 17 shows the results of the three exercise repetitions for one patient.

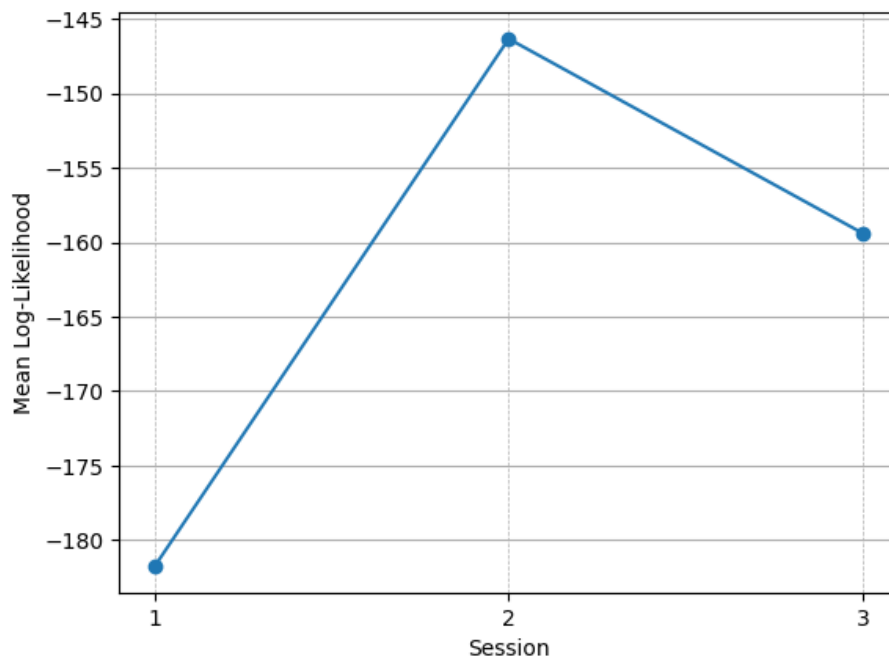


Figure 17: Progress of patient over three sessions

This way, patients and healthcare professionals could use this model, and the design system, in a different way that was initially planned but have a practical and very useful application in the rehabilitation process.

4 Problem Analysis

In this chapter the system developed is analysed. Begins with the requirements analysis, which includes the functional and non-functional requirements that must be achieved by the application, definition of the domain of the problem and the System Architecture designed to meet the requirements identified.

4.1 Requirements Analysis

Requirements analysis is a critical phase in the systems engineering and software development. This analysis will be split into two sections, Functional Requirements and Non-functional Requirements.

4.1.1 Functional Requirements

Functional requirements define what the system must do to meet user needs. Based on the project proposal, it was possible to identify the following use cases represented in the Use Case Diagram in Figure 18.

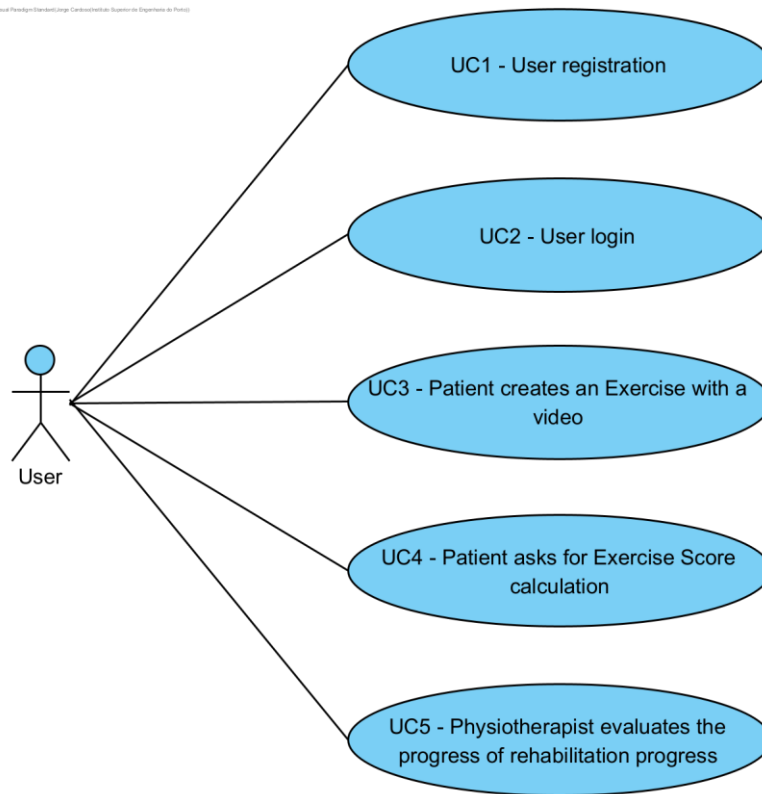


Figure 18: Use Case Diagram

UC1 – User registration

The system must be able to allow users to register an account.

UC2 – User login

The system must be able to allow users to login with the credentials used in the registration process.

UC3 – Patient creates an Exercise with a video

The systems must be able to allow users to create Exercises with an associated video.

UC4 – Patient asks for Exercise Score calculation

The system must be able to allow user to ask for a score calculation for a specific Exercise that was previously created.

UC5 – Physiotherapist evaluates the progress of rehabilitation progress

The system must be able to allow for a healthcare professional to evaluate the progress of the rehabilitation process. This use case was not contemplated in the implementation because was

added in a later stage of the project due to being a byproduct of the ML model development and not planned in the original objectives of this project.

4.1.2 Non-functional Requirements

Non-functional requirements specify the quality attributes and operational criteria of a system, and not its functionalities (Grady, 1992). The requirements analysis was made based on the FURPS+ model which stands for Functionality, Usability, Reliability, Performance and Supportability (Lee, 2014). Later, it was added the “plus,” which extends into other requirements: Design, Implementation, Interface, and Physical requirements. They ensure the system operates effectively and meets user expectations beyond its basic functions.

4.1.2.1 Functionality

- This item is described in the Functional Requirements section.

4.1.2.2 Usability

- The mobile app should be simple and intuitive, and any user must be able to use it with ease.

4.1.2.3 Reliability

- The system should correctly match patient Exercise list and Scores to their respective patients.

4.1.2.4 Performance

- The Score calculation operation should not impact the performance of the rest of the application’s functionalities.

4.1.2.5 Supportability

- Every system’s part should be modular and able to be deployed to a different server without affecting the applications functionality, to facilitate the systems expansion.

4.1.2.6 “+”

- An SQL Server database is adopted.
- The backend of the system is implemented using an ASP.NET Web API.
- The mobile application is implemented using the React Native framework.

- A version control system shall be used.

4.2 Problem's Domain

The main aim of the application is to allow users to upload videos of themselves performing rehabilitation exercises, which are then compared with a reference video of a healthcare professional performing the same exercise and given a score.

4.2.1 Domain Model

After the requirements analysis was possible to develop the Domain Model of the system in question. The Figure 19 presents the Domain Model.

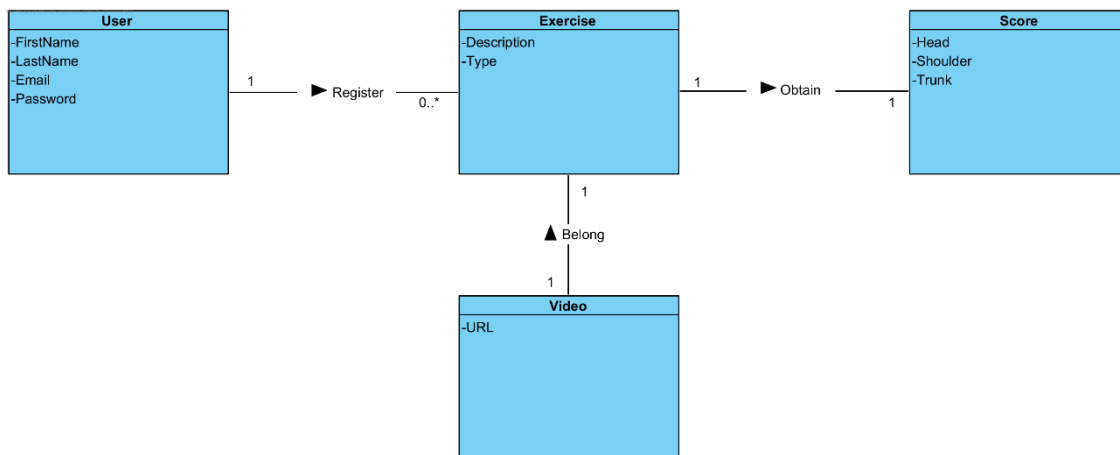


Figure 19: Domain Model

The User, in the context of this system, is the patient who is registering the Exercises. The Exercise is composed of a video of the patient performing a rehabilitation exercise. Each Exercise obtains a score.

5 Solution Design

This chapter specifies the global architecture of the solution. To better specify the multiple components of the solution and their interactions and implementation, the C4 model in conjunction with the 4+1 model was used.

The C4 model is a framework for visualizing the architecture of software systems using a set of hierarchical diagrams. It is designed to provide a clear and structured way to represent the different levels of system architecture, focusing on simplicity and clarity. The C4 model consists of four levels (“The C4 model for visualising software architecture,” n.d.):

- Context Diagram: Provides a high-level view of the system within its environment, showing the system's interactions with external entities like users and other systems.
- Container Diagram: Details the system's major containers (applications, databases, services, etc.) and how they interact with each other.
- Component Diagram: Breaks down each container into its constituent components and illustrates their relationships.
- Code Diagram: Offers a detailed view of the internal structure of individual components, often at the class level in object-oriented systems.

The 4+1 View Model is a software architectural model developed by Philippe Kruchten. It organizes a system's architecture into five concurrent views to address different concerns and stakeholder needs (Kruchten, 1995):

- Logical View: Focuses on the object model of the design, detailing the system's functionality and showcasing how major design elements (classes, objects, etc.) interact.

- **Development View:** Emphasizes the software's static organization in the development environment, representing modules, packages, and layers.
- **Process View:** Addresses the dynamic aspects of the system, focusing on the runtime behaviour and interactions between system processes, including performance, scalability, and concurrency.
- **Physical View:** Illustrates the system's physical deployment on hardware, including nodes, network configurations, and system topology.
- **Scenarios (Use Cases):** The "+1" view, which ties the other four views together by demonstrating how the system's architecture supports specific use cases or user interactions.

5.1 Level 1: Context Diagrams

This section presents the context diagrams that provide a high level of view of the entire system. From now on, the system will be referred to as MLSR (Machine Learning for Shoulder Rehabilitation).

5.1.1 Logic View

Figure 20 represents the level 1 Logic View of the MLSR system.

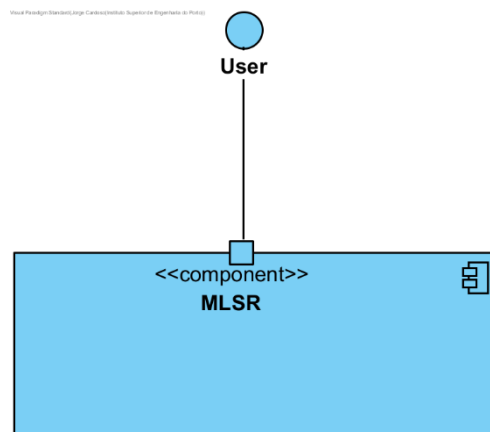


Figure 20: Level 1 Logic View

The MLSR does not interact with external systems, so we can represent the whole system as one component.

5.1.2 Process View

UC3 – Patient creates an Exercise with a video

Figure 21 shows a Process View for the UC3 – User creates an Exercise with a video.

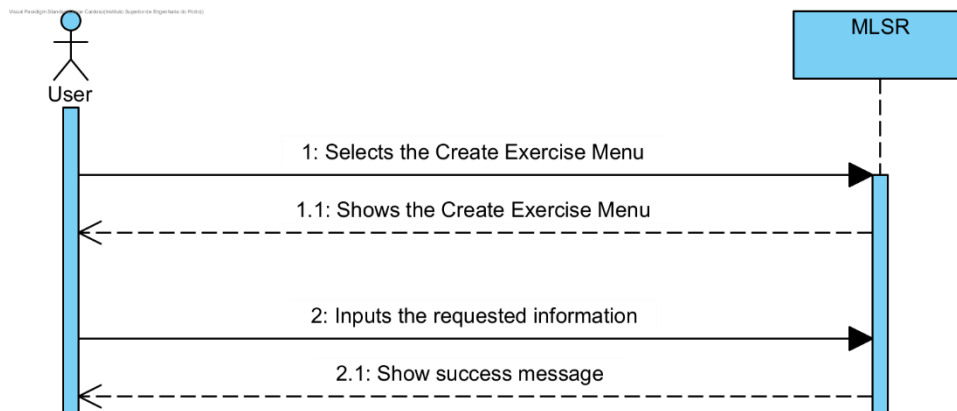


Figure 21: Level 1 Process View for UC3

UC4 – Patient asks for Exercise Score calculation

Figure 22 shows a process view for the UC4 – User asks for Exercise Score calculation.

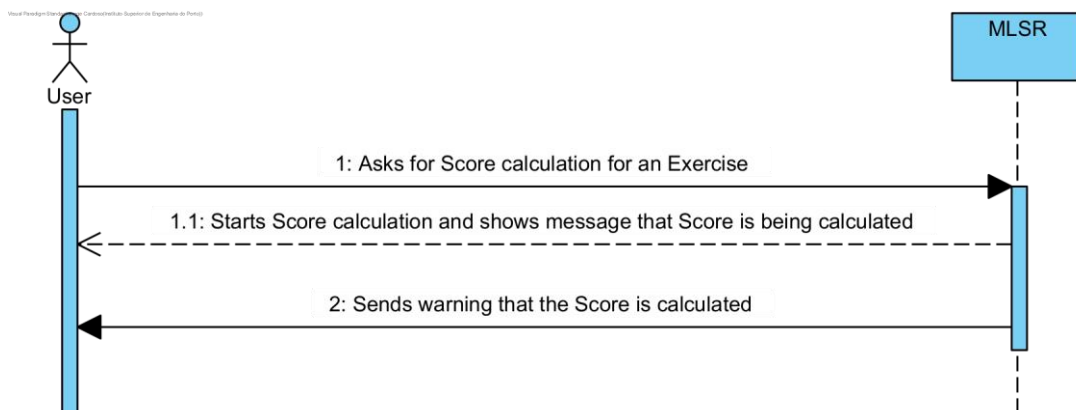


Figure 22: Level 1 Process View for UC4

5.2 Level 2: Container Diagrams

This section presents the container diagrams that provide a more detailed view of the entire system.

5.2.1 Logic View

Figure 23 represents the level 2 Logic View of the MLSR system with the different containers of the system and their dependencies.

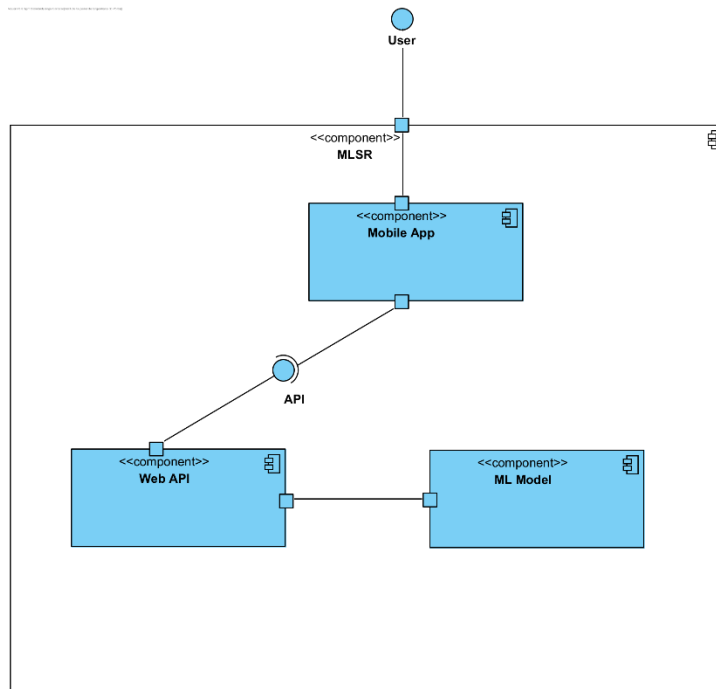


Figure 23: Level 2 Logic View

The system is composed by three different containers, Mobile App, a Web API a Machine Learning model. The User will interact with the system through a Mobile App that communicates with a Web API that is responsible for handling all the business rules and requirements. When a Score calculation is needed, the Web API communicates with a Machine Learning model that performs this calculation and returns the results.

5.2.2 Implementation View

Figure 24 represents the level 2 Implementation View of the MLSR system with the different containers that compose it.

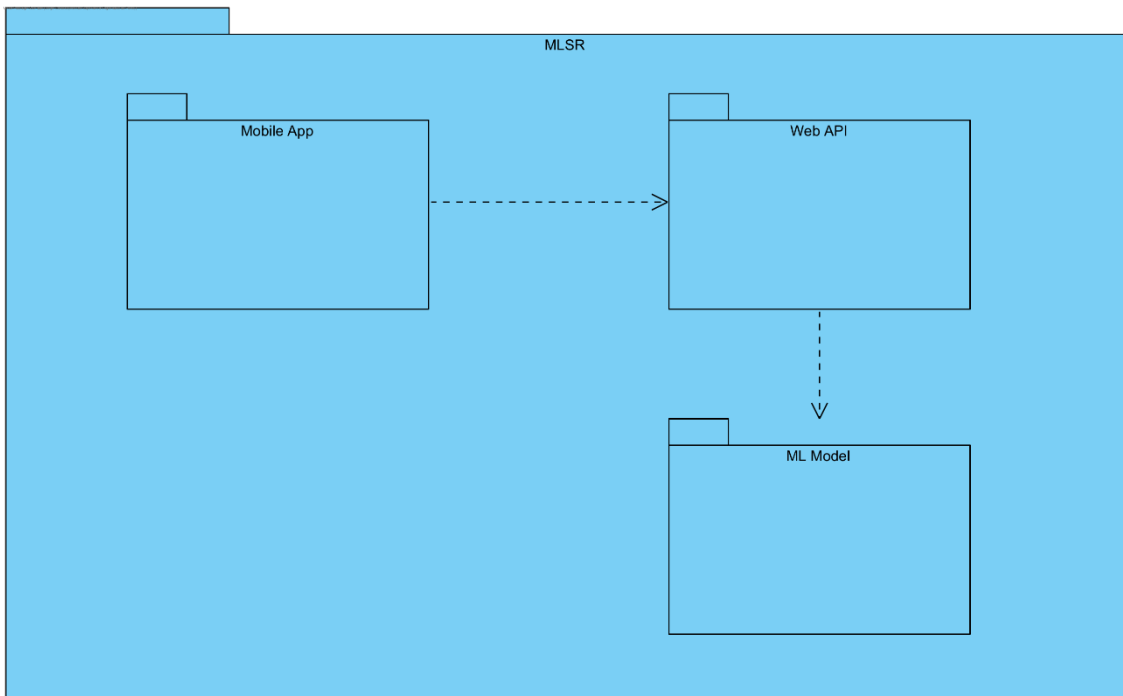


Figure 24: Level 2 Implementation View

It was necessary to divide the Web API from the ML Model because they are implemented in two distinct programming languages, the Web API is implemented in C# and the ML Model is implemented in Python.

The Mobile App must be a different component since will be deployed to a different equipment.

5.2.3 Process View

UC3 – Patient creates an Exercise with a video

Figure 25 shows a Process View for the UC3 – Patient creates an Exercise with a video.

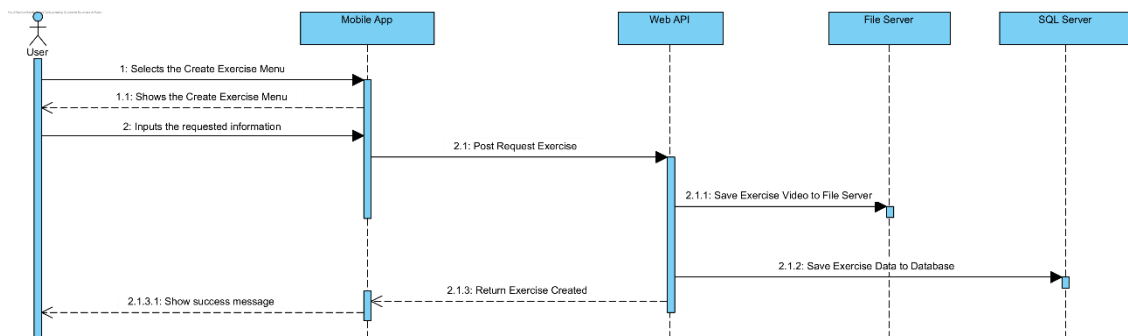


Figure 25: Level 2 Process View for UC3

UC4 – Patient asks for Exercise Score calculation

Figure 26 shows a process view for the UC4 – Patient asks for Exercise Score calculation.

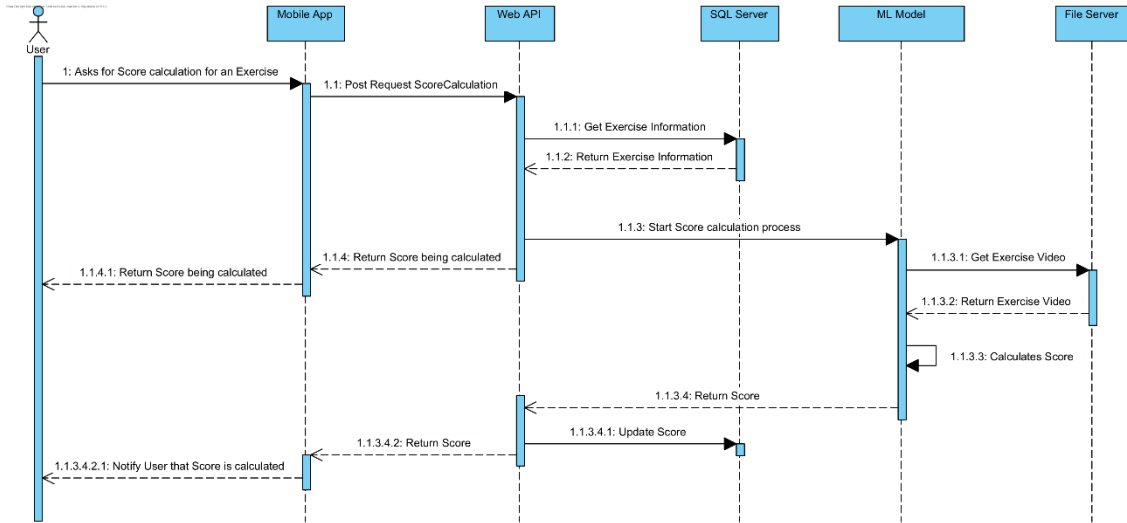


Figure 26: Level 2 Process View for UC4

5.2.4 Physical View

Figure 27 represents the level 2 Physical View of the MLSR system with the different servers necessary to deploy it.

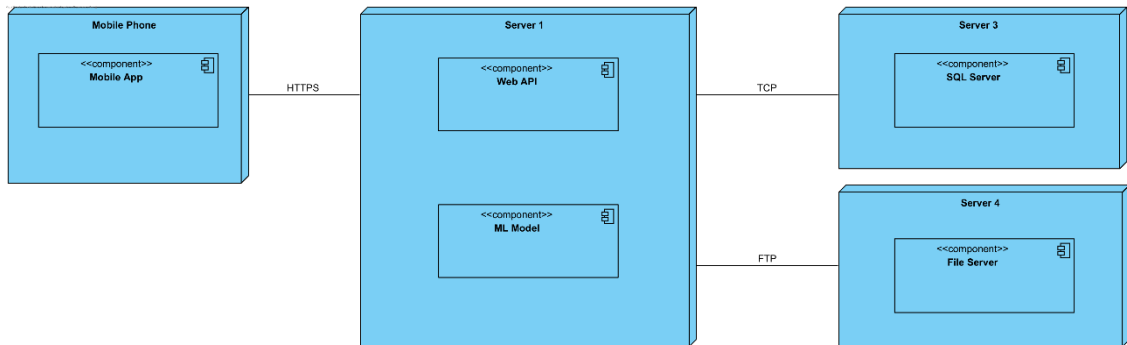


Figure 27: Level 2 Physical View

In the current implementation, the Web API and the ML Model are located on the same server, and the Web API executes the ML Model Python script that calculates the score. A more robust and scalable implementation could be that the Web API and the ML model are on separate servers and communicate via HTTPS. The current implementation was chosen because it is not necessary to handle authentication to the File Server, where the videos of the exercises are stored, on the part of the ML Model.

5.3 Level 3: Component Diagram

This section presents the component diagrams that provide a more detailed view of each container described in the container diagram section.

5.3.1 Logic View

Figure 28 represents the level 3 Logic View of the MLSR system with the different components that compose each container of the system and their dependencies.

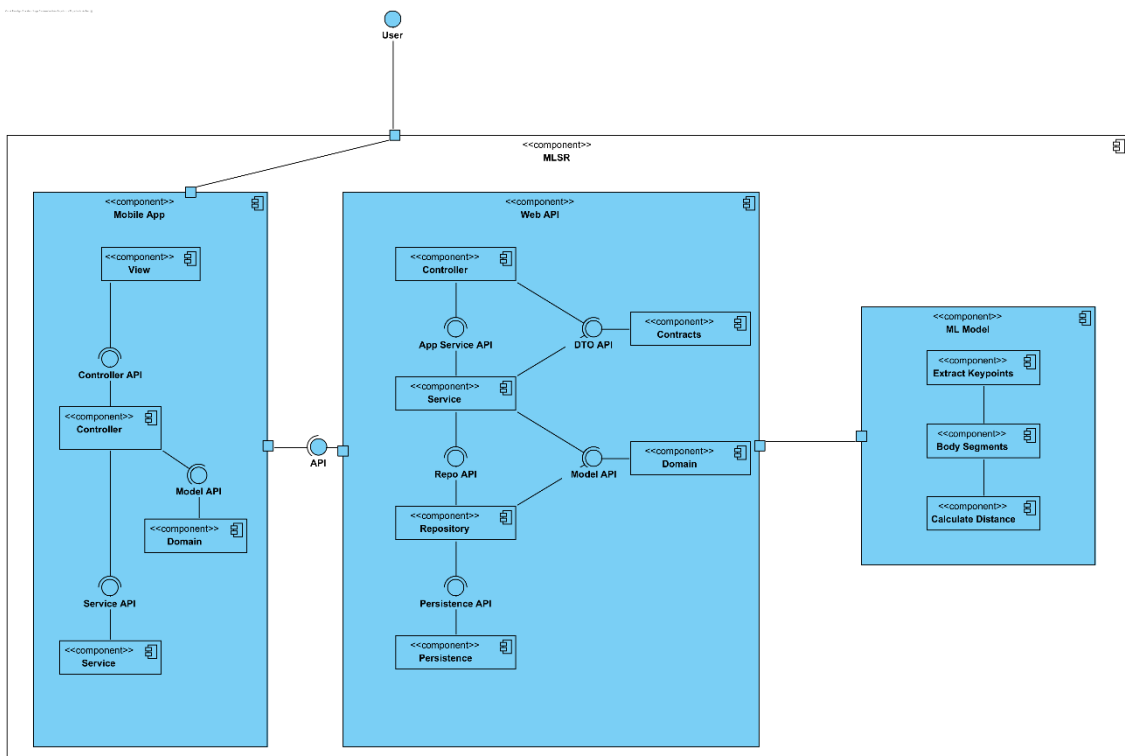


Figure 28: Level 3 Logic View

The Mobile App container adopts a Model-View-Controller (MVC) architecture. In this architecture the Model manages the data and business logic. It communicates with the Services to fetch, process, and store data. The View represents the UI components that the user interacts with, and the Controller handles the user input and interactions, updating the Model and the View accordingly. It was added a Service layer that provides an abstraction layer for making HTTP requests and handling responses.

The Web API container adopts a Clean Architecture where the code is organized in layers, Domain, Application, Infrastructure and Presentation, each with specific responsibilities and dependencies only pointing inwards. The Presentation layer contains the Controller component which is responsible for interacting with the outside world and is where the API endpoints live. This layer only depends on the Application layer. The Application layer contains the Service and

Contracts components and is responsible for orchestrating the operations and uses the domain entities. The Infrastructure Layer that contains the Repository and Persistence components is responsible for providing a concrete implementation for data access and external services defined by the Application layer's interfaces. It handles database interactions and other operational concerns. And lastly the Domain layer that contains the Domain component contains the core business logic and entities.

The ML Model component is a python script. This component can be easily swapped based on the evolution of the ML model developed.

5.3.2 Implementation View

Figure 29 represents the level 3 Implementation View of the MLSR system with the different components that compose each container of the system.

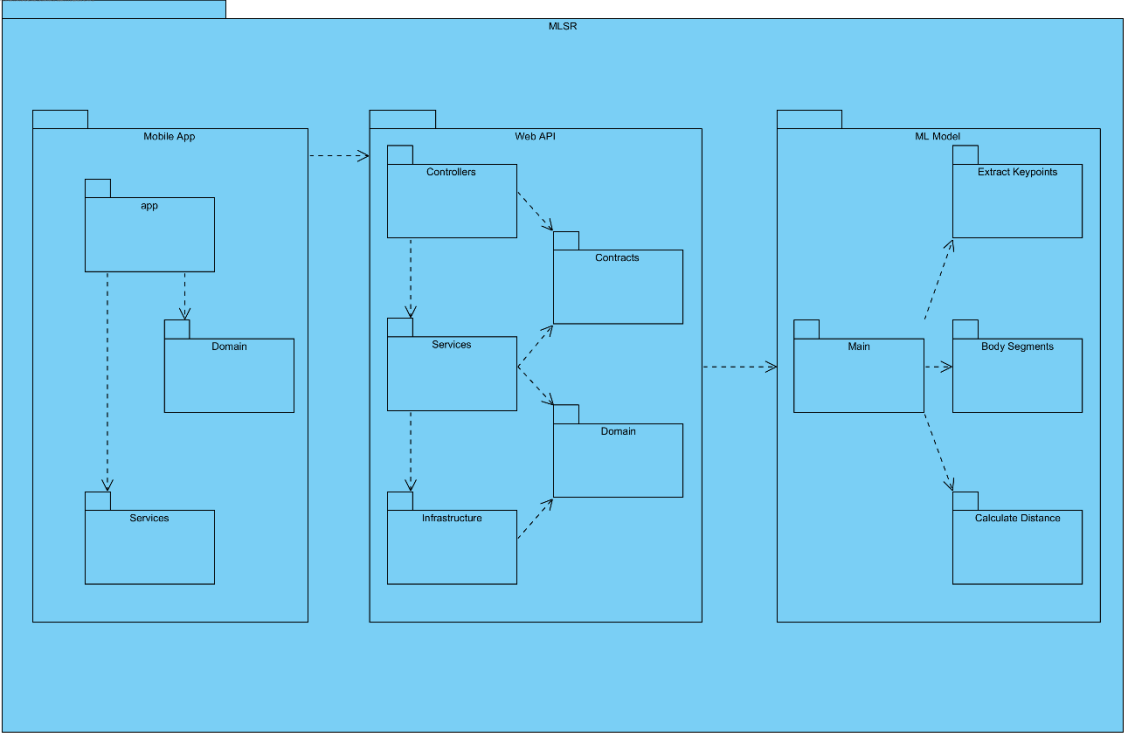


Figure 29: Level 3 Implementation View

5.3.3 Process View

UC3 – Patient creates an exercise with a video

Figure 30 shows a Process View for the UC3 – Patient creates an exercise with a video.

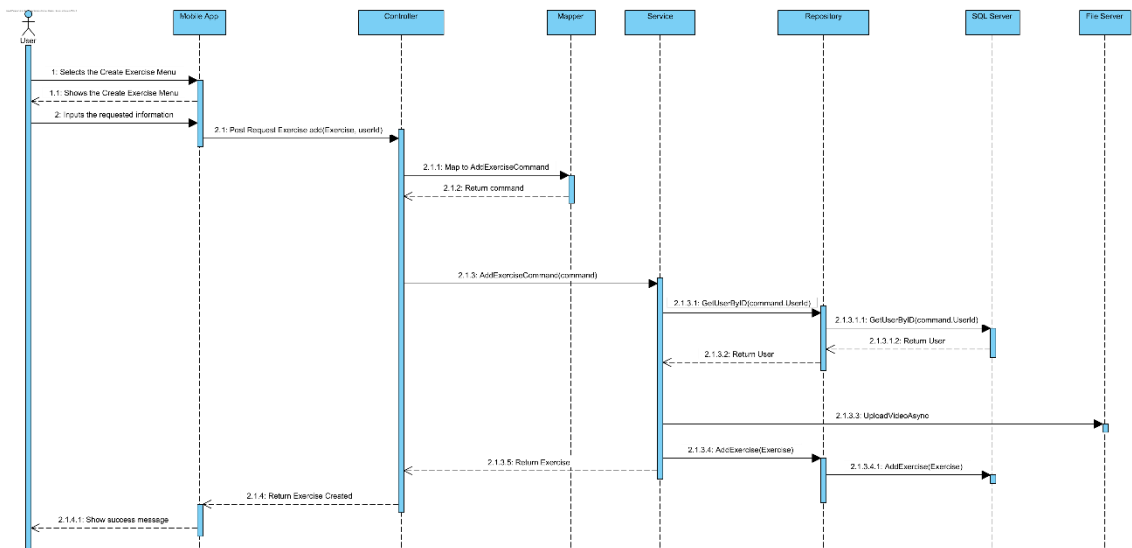


Figure 30: Level 3 Process View for UC3

UC4 – Patient asks for Exercise Score calculation

Figure 31 shows a process view for the UC4 – Patient asks for exercise score calculation.

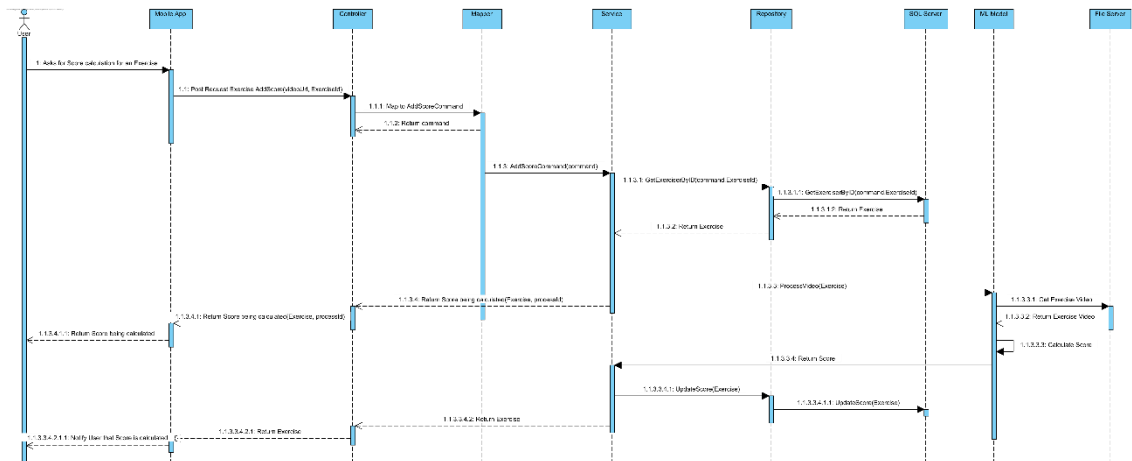


Figure 31: Level 3 Process View for UC4

6 Solution Implementation

This chapter presents the implementation of the solution designed in the Solution Designs section.

6.1 Technologies Chosen

The Solution developed is composed of five different components, a ML Model, a Web API, a Mobile App, a Database and a Fileserver.

Each component, due to its particularities and requirements, was developed in different frameworks and programming languages.

The ML Model was developed using the python programming language due to the very wide range of libraries available for video processing and time series comparison.

The Web API was developed using ASP.NET Web API with C# as programming language. The framework was chosen also due to its popularity and the ability to the desired architecture and code patterns with ease (“ASP.NET Web APIs | Rest APIs with .NET and C#,” n.d.).

The Mobile App was developed in React Native with TypeScript as programming language. There are multiple frameworks to mobile development. React Native was the chosen one due to its popularity, a very large community with a strong support, and the ability to, with one codebase, develop to Android and IOS (“React Native · Learn once, write anywhere,” n.d.).

The Database used is a Microsoft SQL Server relational database. It’s a very well established and robust relational database.

The Fileserver used is a Network-Attached Storage (NAS) Server for its simplicity and availability.

6.2 Implementation

This chapter describes in more detail the implementation of the designed system. The entire implementation is available in a GitHub repository (“jorgecardoso220/Machine-Learning-for-Shoulder-Rehabilitation,” n.d.).

6.2.1 ML Model

The main objective of this solution component is to compare a video from a patient performing a rehabilitation exercise and compare it with a video of a health care professional performing the same exercise and calculate a score (good or bad).

For this, a python scrip was implemented with two different parts:

- 1st part: Video key points extraction.
- ML model that calculates the score.

A main function orchestrates the way these multiples parts work in conjunction and it’s the entry point to the ML model that is called by the API. The specific implementation was discussed in chapter 3 of this work. Machine Learning model for shoulder rehabilitation exercises assessment

6.2.2 Web API

The Web API serves as the backbone of the system, facilitating communication between the ML Model, the Mobile App, the Database, and the File server. It is developed using C# and adheres to the principles of Clean Architecture to ensure maintainability, scalability, and testability.

The implementation was done separating the code into layers, each with distinct responsibilities, to ensure that the core business logic remains independent of external dependencies. The key layers are:

- Domain: Contains the business logic and domain entities.

This layer includes the fundamental business objects: User and Exercise.

- Application: Contains use cases, application services like.

This layer implements the four Use Cases defined in the Requirements Analysis section of the project. It also defines the Contracts (DTOs) to structure the data for input and output. Is in the layer that is defined the service that runs the ML Model to calculate the score of an exercise.

Command Query Responsibility Segregation (CQRS) (RobBagby, n.d.) pattern was used to separate the reads and writes into Queries when data is being fetched from the Database and Commands when new data is being created and stored in the Database.

The ML Model service creates a new Process instance and execute the Python scrip with the ML Model directly

- Infrastructure: Handles external concerns like data access and file storage.

The Infrastructure Layer implements repository interfaces using Entity Framework Core library to connect to the Database. It also manages interactions with the File server.

- Controllers: Exposes the API endpoints and manages HTTP requests.

Is where API endpoints are defined, and HTTP requests are managed. Were created the UserController and ExerciseController to handle these requests. The ErrorOr (Mantinband, 2024) library was used to better handle error in the application and the Mapster ("MapsterMapper/Mapster," 2024) library was used to map all objects in the application.

The API endpoints that were created are:

- Register user:

```
POST {{host}}/api/auth/register
Content-Type: application/json

{
  "firstName": "Jorge",
  "lastName": "Cardoso",
  "email": "jorgecardoso@mail.com",
  "password": "*****"
}
```

- Login:

```
POST {{host}}/api/auth/login
Content-Type: application/json

{
  "email": "jorgecardoso@mail.com",
  "password": "*****"
}
```

- Create Exercise:

```
POST {{host}}/api/exercise/add
Content-Type: multipart/form-data

{
  "Description": "First exercise",
  "Type": "Diagonal",
  "UserId": "1d9d1d07-e14c-4e72-abb0-2df653c1a65b",
  "VideoFile": "FileURI"
}
```

- Get Exercise:

```
POST {{host}}/api/exercise/getExercise
Content-Type: application/json

{
  "UserId": "1d9d1d07-e14c-4e72-abb0-2df653c1a65b",
  "ExerciseId": "eb63968e-9957-4cd1-ad06-38fcc15af037"
}
```

- Get Exercises list:

```
POST {{host}}/api/exercise/getExercises
Content-Type: application/json

{
  "Id": "1d9d1d07-e14c-4e72-abb0-2df653c1a65b"
}
```

- Calculate Score to an Exercise:

```
POST {{host}}/api/exercise/addScore
Content-Type: application/json

{
  "Id": "d8a9327d-f378-41af-9dd4-9299b27fe649",
  "videoUrl": "VideoURL",
}
```

- Get Score Calculation Status:

```
POST {{host}}/api/exercise/getScoreCalculationStatus
Content-Type: application/json

{
  "processId": "a8c13496-218e-4d22-b09f-6b70ba3db042"
}
```

For security measures authentication is enforced through JSON Web Tokens (JWT). Input validation is implemented using the library FluentValidation to ensure data integrity.

6.2.3 Mobile App

The Mobile App serves as the primary user interface, allowing users to interact with the system. Developed using React Native, the app runs on both iOS and Android platforms, providing a seamless and consistent user experience. The app communicates with the Web API the give answer to the Use Cases defined in the requirements analysis.

The Mobile App is composed of a Sign Up page, and a Log in page for handling the user registration and authentication. Figure 32 shows the Sign Up and Log in pages.

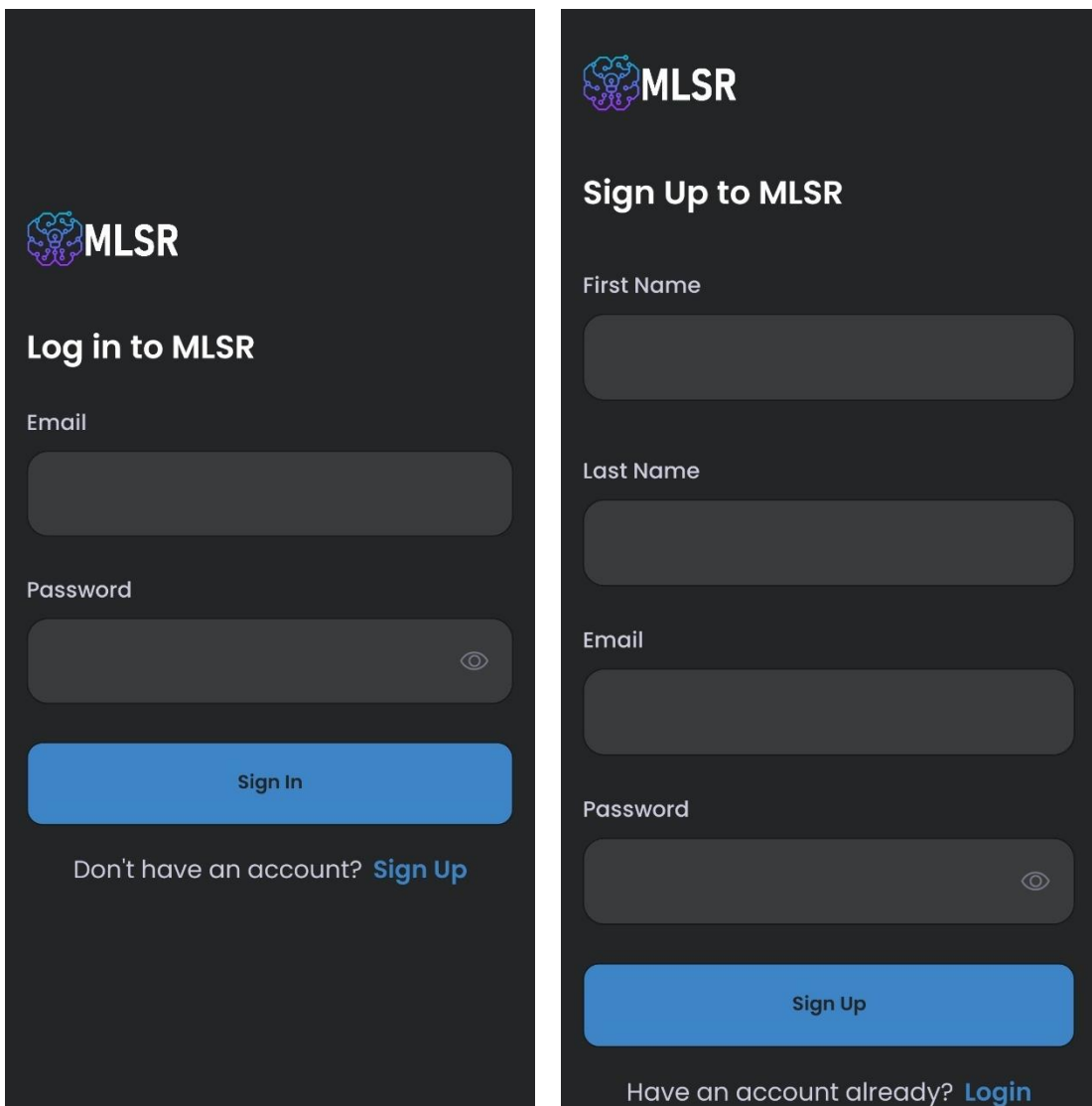


Figure 32: Sign Up and Log in pages

Once a user completes the registration process and successfully logs in to the App, he is presented with a home page where he can see if he has a doctor's appointment (this part isn't implemented; it is just placeholder text) and the last video that was submitted to the App. There is also a Create page for creating new exercises and a profile page where all exercises can be

viewed. If a certain Exercise already has a score calculated that information will be displayed below the video, if not, a button will be displayed for asking for a Score calculation. Figure 33 shows the different pages of the App as well as different stages of score calculation.

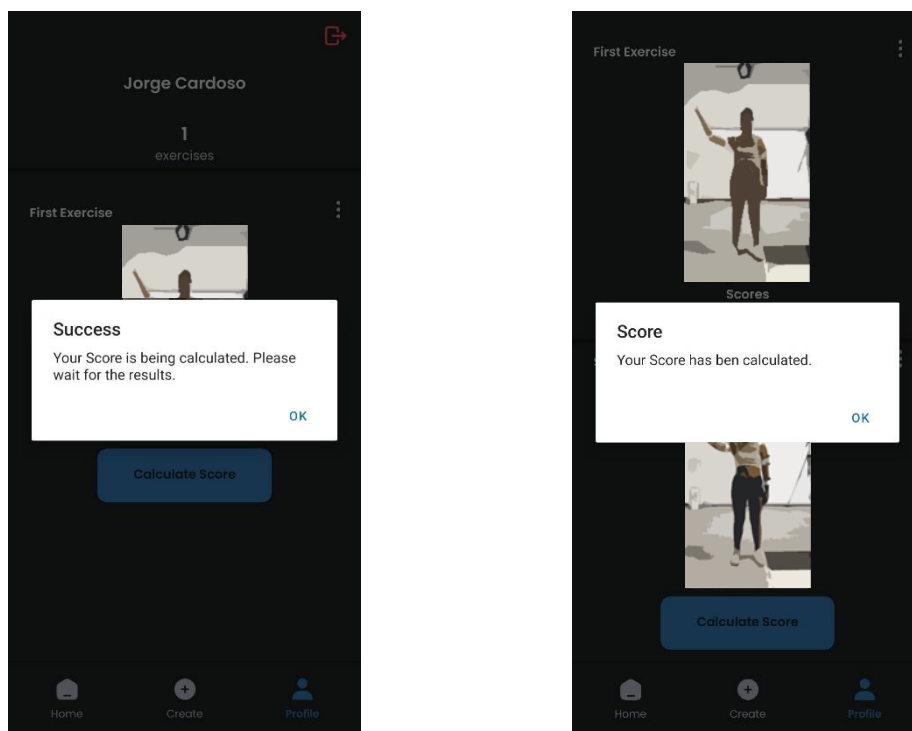
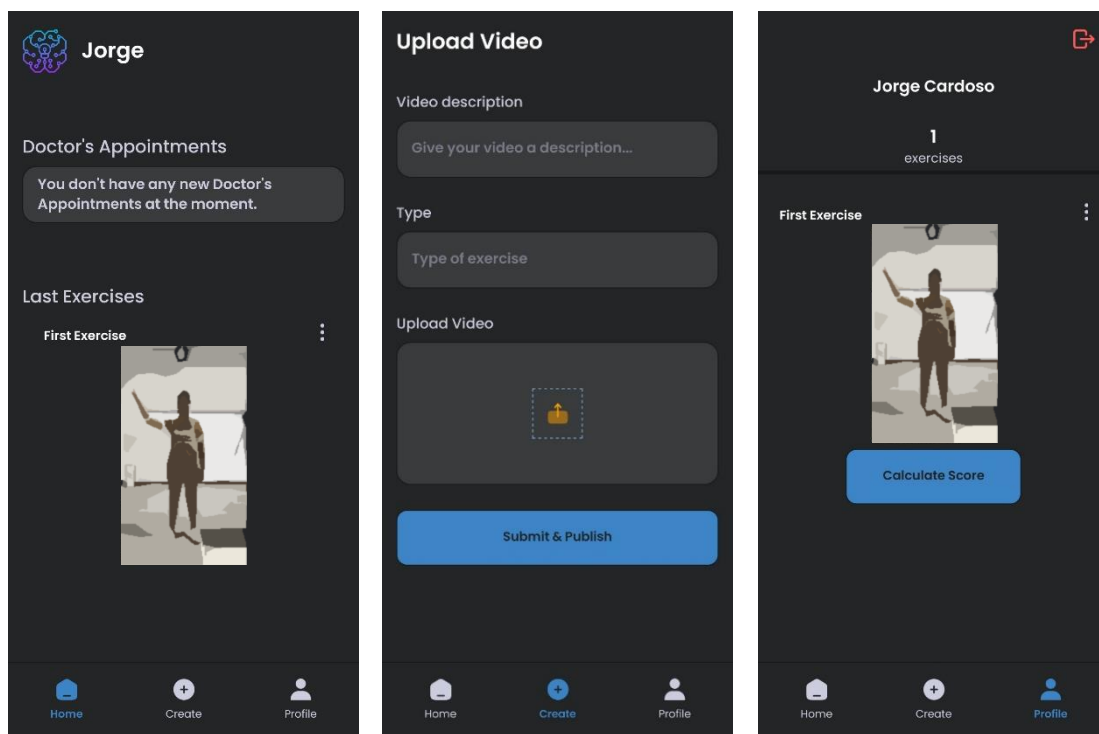


Figure 33: Different pages of the Mobile App

6.2.4 Database

The Database used is a Microsoft SQL Server relational database. Figure 34 shows the Entity Relationship Diagram of the Database constructed for the system.

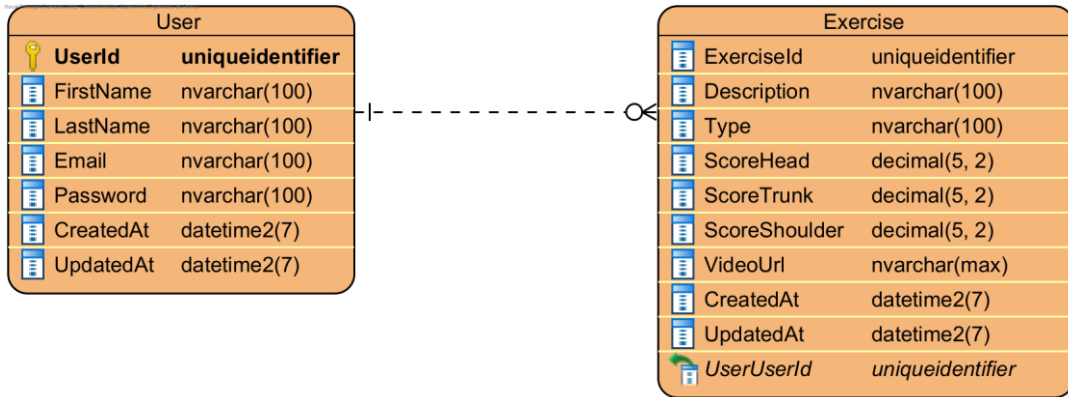


Figure 34: Entity Relationship Diagram of the Database

6.2.5 Fileserver

The Fileserver used for storing the Exercise videos is a NAS server. The Web API establishes an FTP connection to the NAS to store the Exercise videos and retrieve them to use in the ML model for score calculations. To ensure that video from different users will not be mixed up, a new folder is created for each user.

7 Conclusion

This chapter aims to describe the accomplished objectives achieved in relation to the initial proposal.

The main objective of this project was to discover if it was possible to reliably measure the quality of home-based shoulder rehabilitation exercises using a Machine Learning (ML) tool, compared to medical evaluation systems.

A Literature review on existing approaches to upper limb rehabilitation using machine learning was successfully conducted and documented.

The proposed system was successfully designed, documented and implemented.

The developed Mobile App in conjunction with the Web API, database and fileserver allow the patient to create exercises, and request for a score calculation.

The ML models that were developed, did not fully achieve the desired performance accessing the quality of home-based shoulder rehabilitation exercises. On the other hand, one new, not initially planned, application for one of the ML models developed (GMM), was access the evolution of the rehabilitation process section after section.

7.1 Limitations and Future Work

During the development of the ML models some limitations of the followed approach were identified. In this section those limitations are identified.

The MediaPipe framework use to process the exercise videos and extract keypoints may be sensitive to the distance at which the videos are filmed, since the coordinates that provides for the keypoints are measured from the left side of the screen until the keypoint in question. This

way, in a real scenario where a patient uses the application at home, the application should tell the patient at what distance to place the phone and in what position.

In the first approach used to develop the ML model (the distance calculation) the limitations of this model were already pointed out. Some of these limitations could be overcome by finding a way to transform the distance in a score value, like the QTM software returns to see if the performance of the model improves.

Long Short-Term Memory (LSTM) could be a useful alternative approach for comparing two videos of arm movements. The advantage of using LSTMs is that they can automatically learn the relevant features and patterns from the video data, eliminating the need for manual feature engineering or alignment techniques like DTW. However, training an LSTM model requires a larger amount of data, which prevented us from adopting this technique.

Modify the system's architecture to be able to handle different roles, such as patient and healthcare professionals, so the UC5 can be implemented. This approach allows for both patients and healthcare professionals, to use the same system to access the progress of the rehabilitations process.

References

- Abdi, H., Williams, L.J., 2010. Principal component analysis. *WIREs Computational Statistics* 2, 433–459. <https://doi.org/10.1002/wics.101>
- ASP.NET Web APIs | Rest APIs with .NET and C# [WWW Document], n.d. . Microsoft. URL <https://dotnet.microsoft.com/en-us/apps/aspnet/apis> (accessed 6.26.24).
- Bevan, S., 2015. Economic impact of musculoskeletal disorders (MSDs) on work in Europe. *Best Pract Res Clin Rheumatol* 29, 356–373. <https://doi.org/10.1016/j.berh.2015.08.002>
- Bishop, C.M., 2006. Pattern recognition and machine learning. New York: Springer, [2006] ©2006.
- Cord, M., Cunningham, P., 2008. Machine Learning Techniques for Multimedia, 1st ed, Cognitive Technologies. Springer Berlin, Heidelberg.
- Cortes, C., Vapnik, V., 1995. Support-vector networks. *Mach Learn* 20, 273–297. <https://doi.org/10.1007/BF00994018>
- Cunha, B., Ferreira, R., Sousa, A.S.P., 2023. Home-Based Rehabilitation of the Shoulder Using Auxiliary Systems and Artificial Intelligence: An Overview. *Sensors* 23, 7100. <https://doi.org/10.3390/s23167100>
- Das, U.C., Le, N.T., Benjapolakul, W., Vitoonpong, T., Pluempitiwiriyaewej, C., 2023. Stroke Home Rehabilitation Approach Using Mobile Application Based on PostNet Machine Learning Model, in: Proceedings of the 2023 7th International Conference on Medical and Health Informatics, ICMHI '23. Association for Computing Machinery, New York, NY, USA, pp. 222–228. <https://doi.org/10.1145/3608298.3608339>
- Davis, T., Adams, C.-M., Zhao, J., Zhao, Y., O'Rourke, N., Peng, H., Geiger, A., Raiti, J., 2022. Data Signal Processing and Machine Learning for Motion Detection; A Solution for Postoperative Breast Cancer Rehabilitation, in: Proceedings of the 15th International Conference on Pervasive Technologies Related to Assistive Environments, PETRA '22. Association for Computing Machinery, New York, NY, USA, pp. 327–329. <https://doi.org/10.1145/3529190.3534787>
- Dertat, A., 2017. Applied Deep Learning - Part 1: Artificial Neural Networks [WWW Document]. Medium. URL <https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6> (accessed 12.28.23).
- Flow diagram of gradient boosting machine learning method. The ensemble... [WWW Document], n.d. . ResearchGate. URL https://www.researchgate.net/figure/Flow-diagram-of-gradient-boosting-machine-learning-method-The-ensemble-classifiers_fig1_351542039 (accessed 1.6.24).

- GaussianMixture [WWW Document], n.d. . scikit-learn. URL <https://scikit-learn/stable/modules/generated/sklearn.mixture.GaussianMixture.html> (accessed 9.13.24).
- Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.
- Grady, R.B., 1943-, 1992. Practical software metrics for project management and process improvement. Prentice Hall, Englewood Cliffs, NJ.
- Islam, A.T., Bader, S., Kirste, T., 2023. Evaluating neurorehabilitation exercises captured with commodity sensors and machine-learning framework, in: Proceedings of the 7th International Workshop on Sensor-Based Activity Recognition and Artificial Intelligence, iWOAR '22. Association for Computing Machinery, New York, NY, USA, pp. 1–10. <https://doi.org/10.1145/3558884.3558897>
- Jolliffe, I.T., Cadima, J., 2016. Principal component analysis: a review and recent developments. *Philos Trans A Math Phys Eng Sci* 374, 20150202. <https://doi.org/10.1098/rsta.2015.0202>
- jorgecardoso220/Machine-Learning-for-Shoulder-Rehabilitation [WWW Document], n.d. . GitHub. URL <https://github.com/jorgecardoso220/Machine-Learning-for-Shoulder-Rehabilitation> (accessed 6.27.24).
- Kim, J.-W., Choi, J.-Y., Ha, E.-J., Choi, J.-H., 2023. Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model. *Applied Sciences* 13, 2700. <https://doi.org/10.3390/app13042700>
- Krittanawong, C., Rogers, A.J., Johnson, K.W., Wang, Z., Turakhia, M.P., Halperin, J.L., Narayan, S.M., 2021. Integration of novel monitoring devices with machine learning technology for scalable cardiovascular management. *Nat Rev Cardiol* 18, 75–91. <https://doi.org/10.1038/s41569-020-00445-9>
- Kruchten, P., 1995. The 4+1 View Model of Architecture. *IEEE Software* 12, 45–50. <https://doi.org/10.1109/52.469759>
- Lee, M.-C., 2014. Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance. *British Journal of Applied Science & Technology*. <https://doi.org/10.9734/BJAST/2014/10548>
- Mahesh, B., 2020. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*. [Internet] 9, 381–386.
- Mantinband, A., 2024. amantinband/error-or.
- MapsterMapper/Mapster, 2024.
- MediaPipe [WWW Document], n.d. . Google for Developers. URL <https://developers.google.com/mediapipe> (accessed 12.29.23).

- Naeem, S., Ali, A., Anam, S., Ahmed, M., 2023. An Unsupervised Machine Learning Algorithms: Comprehensive Review. *IJCDS Journal* 13, 911–921. <https://doi.org/10.12785/ijcnds/130172>
- Natekin, A., Knoll, A., 2013. Gradient boosting machines, a tutorial. *Front Neurorobot* 7, 21. <https://doi.org/10.3389/fnbot.2013.00021>
- OpenCV [WWW Document], n.d. . OpenCV. URL <https://opencv.org/> (accessed 9.14.24).
- PCA [WWW Document], n.d. . scikit-learn. URL <https://scikit-learn/stable/modules/generated/sklearn.decomposition.PCA.html> (accessed 9.13.24).
- Qualisys Track Manager (QTM) [WWW Document], n.d. . Qualisys. URL <https://www.qualisys.com/software/qualisys-track-manager/> (accessed 9.15.24).
- Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E., 2012. Searching and mining trillions of time series subsequences under dynamic time warping, in: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12*. Association for Computing Machinery, New York, NY, USA, pp. 262–270. <https://doi.org/10.1145/2339530.2339576>
- React Native · Learn once, write anywhere [WWW Document], n.d. URL <https://reactnative.dev/> (accessed 6.26.24).
- Regression vs. Classification in Machine Learning: What's the Difference? [WWW Document], 2021. . Springboard Blog. URL <https://www.springboard.com/blog/data-science/regression-vs-classification/> (accessed 12.27.23).
- Reynolds, D.A., 2009. Gaussian mixture models. *Encyclopedia of biometrics* 741.
- RobBagby, n.d. CQRS pattern - Azure Architecture Center [WWW Document]. URL <https://learn.microsoft.com/en-us/azure/architecture/patterns/cqrs> (accessed 6.27.24).
- Rodrigues, L.G.S., Dias, D., Guimaraes, M. de P., Brandao, A.F., Rocha, L., Iope, R.L., Brega, J.R.F., 2022. Classification of Human Movements with Motion Capture Data in a Motor Rehabilitation Context, in: *Symposium on Virtual and Augmented Reality, SVR'21*. Association for Computing Machinery, New York, NY, USA, pp. 56–63. <https://doi.org/10.1145/3488162.3488210>
- Rodrigues, Luis.G.S., Dias, D.R.C., Guimarães, M.P., Brandão, A.F., Rocha, L.C.D., Iope, R.L., Brega, J.R.F., 2021. Upper Limb Motion Tracking and Classification: A Smartphone Approach, in: *Proceedings of the Brazilian Symposium on Multimedia and the Web, WebMedia '21*. Association for Computing Machinery, New York, NY, USA, pp. 61–64. <https://doi.org/10.1145/3470482.3479618>
- Shlens, J., 2014. A Tutorial on Principal Component Analysis. *Educational* 51.
- StandardScaler [WWW Document], n.d. . scikit-learn. URL <https://scikit-learn/stable/modules/generated/sklearn.preprocessing.StandardScaler.html> (accessed 9.13.24).

- Sunder, S., 2008. Textbook of Rehabilitation. Jaypee Brothers Medical Publishers Pvt. Limited.
- Tan, P.-N., Steinbach, M., Karpatne, A., Kumar, V., 2019. Introduction to data mining, Second edition. ed. Pearson Education, Inc. New York, NY, New York, NY.
- The C4 model for visualising software architecture [WWW Document], n.d. URL <https://c4model.com/> (accessed 6.24.24).
- Tychola, K., Tsimperidis, I., Papakostas, G., 2022. On 3D Reconstruction Using RGB-D Cameras. Digital 2, 401–423. <https://doi.org/10.3390/digital2030022>
- Wirth, R., Hipp, J., 2000. Crisp-dm: towards a standard process modell for data mining.