UNIVERSITAT DE BARCELONA

Final Degree Project

Biomedical Engineering Degree

"MRI-based radiomics machine learning model for tumour response prediction to neoadjuvant chemoradiotherapy (nCRT) in locally advanced rectal cancer (LARC): a retrospective study"

Barcelona, 11 de Juny de 2025

Author: Paula Sanahuja Rosich

Director/s: Dr. Josep Munuera del Cerro

Tutor: Dra. Núria Gavara Casas

Abstract

Rectal cancer (RC) is one of the most commonly diagnosed malignant tumours worldwide. Its most aggressive form, Locally Advanced Rectal Cancer (LARC), is treated with neoadjuvant chemoradiotherapy (nCRT) to downstage the tumour and improve results of the total mesorectal excision (TME). Approximately 20% to 25% achieve complete response and may be eligible for a more conservative watch and wait strategy. Early prediction of response to nCRT using information from the staging MRI could help adjust neoadjuvant treatment andimprove response, potentially avoiding surgeries.

The aim of this project is to develop a machine learning (ML) model capable of predicting response to nCRT using clinical data and radiomics characteristics extracted from the pre-treatment MRI. The radiomics are extracted from manually delineated tumour masks (Core Tumour Radiomics) and from border masks computed from the manual segmentations (Border Radiomics). Nine ML models have been optimized and tested across the seven feature set combinations. The models with best performance were Random Forest for the core tumour radiomics dataset (Accuracy = 0.77, AUC = 0.70, Sensitivity = 0.67, Specificity = 0.86) and Multilayer Perceptron for the dataset with all features (Accuracy = 0.77, AUC = 0.70, Sensitivity = 0.83, Specificity = 0.71). However, the predictive capability of tumour borders could not be confirmed as these models yielded worse performance. Furthermore, via a feature importance analysis, it has been concluded that both shape and texture related radiomic features are predictors of treatment response although no specific marker has been identified.

KEYWORDS:

Rectal Cancer – Locally Advanced Rectal Cancer (LARC) – Neoadjuvant Chemoradiotherapy (nCRT) – Magnetic Resonance Imaging (MRI) – Radiomics – Artificial Intelligence – Machine Learning – Tumour Segmentation – Watch and Wait Strategy – Treatment Response

Resum

El càncer de recte és un dels tumors malignes més diagnosticats arreu del món. El tipus més agressiu, el càncer de recte localment avançat, es tracta amb quimioradioteràpia neoadjuvant per disminuir l'estadi del tumor i millorar els resultats de la cirurgia d'excisió total del mesorecte. Aproximadament, del 20% al 25% dels pacients presenten resposta completa al tractament i poden optar a una estratègia conservadora de vigilància activa. Predir la resposta primerenca a la teràpia neoadjuvant utilitzant informació obtinguda de la RM d'estadiatge pot assistir en la prescripció del tractament neoadjuvant i així millorar la resposta, potencialment evitant cirurgies.

L'objectiu d'aquest projecte és desenvolupar un model d'aprenentatge automàtic capaç de predir la resposta al tractament neoadjuvant utilitzant dades clíniques i característiques radiòmiques extretes de la RM pre-tractament. Les variables radiòmiques s'han extret de les segmentacions manuals i de les màscares de les vores tumorals. S'han optimitzat i entrenat nou models per cada una de les set combinacions de conjunts de variables. Els models amb millors resultats son *Random Forest* per el conjunt de dades radiòmiques extretes de tumor (Exactitud = 0.77, AUC = 0.70, Sensibilitat = 0.67, Especificitat = 0.86) i *Multilayer Perceptron* pel conjunt amb totes les variables (Exactitud = 0.77, AUC = 0.70, Sensibilitat = 0.83, Especificitat = 0.71). Tanmateix, la capacitat predictiva de les vores tumorals no s'ha pogut confirmar, ja que els models amb aquest conjunt de dades han donat els pitjors resultats. Addicionalment, mitjançant una anàlisi de la importància de les característiques, s'ha pogut concloure que tant les variables radiòmiques de forma com les de textura, en general, prediuen la resposta al tractament, tot i que no s'han pogut identificar marcadors concrets.

PARAULES CLAU:

Càncer de Recte – Càncer de Recte Localment Avançat – Quimioradioteràpia Neoadjuvant – Ressonància Magnètica – Radiòmica – Intel·ligència Artificial – Aprenentatge Automàtic – Segmentació de tumors – Estratègia de Vigilància Activa – Resposta al Tractament

Acknowledgments

The development and completion of this project would not have been possible without the support of the team at the *Hospital de la Santa Creu i Sant Pau* who provided guidance throughout the entire process.

First, I want to sincerely thank my director Dr. Josep Munuera for his guidance and expertise, without which this research could not have been completed. On the same note, I also want to thank my tutor Dra. Núria Gavara for the support in the development of this report.

I would also like to extend my gratitude to the other members of the Radiology Department of the hospital who created an exceptional work environment that enabled this research and who encouraged me to move forward through this long journey. Particularly, I want to thank Dr. Miguel Angel Rios for his support in the medical training required for this work and for correcting the segmentations, and Lucia Borrego for her guidance during the data analysis and machine learning steps. Furthermore, I would also like to thank Dr. Daniel Selva and Dra. Nataly Reyes, residents from the hospital, for their assistance with the segmentation process.

Last but not least, I would like to genuinely thank all my family and friends who have accompanied me and given their unconditional support throughout this final degree project.

List of figures

List of tables

List of equations

List of abbreviations

LARC: Locally Advanced Rectal Cancer
AI: Artificial Intelligence
ML: Machine Learning
DL: Deep Learning
LLMs: Large Language Models
XAI: Explainable Artificial Intelligence
NLP: Natural Language Processing
RC: Rectal Cancer
MRI: Magnetic Resonance Imaging
T2w: T2-weighted
nCRT: neoadjuvant chemoradiotherapy
TME: Total Mesorectal Excision surgery
pCR: pathologic complete response
DWI: Diffusion Weighted Imaging
ADC: Apparent Diffusion Coefficient
TNM: Tumour, Node, Metastasis
TRG: Tumour Regression Grade
TEM: Transanal Endoscopic Microsurgery
W&W: Watch & Wait strategy
TSE: Turbo Spin Echo
FSE: Fast Spin Echo
TR: Repetition Time
DICOM: Digital Imaging and Communications in Medicine
ROI: Region of Interest
3D: Three dimensional
IBSI: Imaging Biomarker Standardization Initiative
PCA: Principal Component Analysis
PC: Principal Components
LDA: Linear Discriminant Analysis
CNN: Convolutional Neural Network
DSC: Dice Similarity Coefficient
CV: Cross-Validation
DT: Decision Tree
RF: Random Forest
LR: Logistic Regression
SVM: Support Vector Machine
RBF: Radial Basis Function
XGBoost: Extreme Gradient Boosting
LightGBM: Light Gradient Boosting Machine
MLP: Multilayer Perceptron
TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative
PPV: Positive Predicted Value
NPV: Negative Predicted Value
SHAP: SHapley Additive Explanations
FMR: Mesorectal Fascia
EMVI: Extramural Vascular Invasion
CEA: Carcinoembryonic Antigen
MSI: Microsatellite Instability
ROC curve: Receiver Operating Characteristic curve
AUC: Area Under the Curve
GLDM: Gray Level Dependence Matrix
GLRLM: Gray Level Run Length Matrix
GLCM: Gray Level Co-occurrence Matrix
GLSZM: Gray Level Size Zone Matrix
NGTDM: Neighbouring Gray Tone Difference Matrix
CTR: Core Tumour Radiomics
BR: Border Radiomics
WBS: Work Breakdown Structure
PERT: Program Evaluation and Review Technique
CPM: Critical Path Method
SWOT: Strengths, Weaknesses, Opportunities, Threats
EU-GDPR: EU General Data Protection Regulation
LOPDGDD: Spanish Organic Law on Data Protection and Digital Rights
ESR: European Society of Radiology
EU-MDR: EU Medical Device Regulat

Table of contents

1. Introduction

Colorectal cancer is one of the most commonly diagnosed malignant tumours worldwide. Although it affects mostly people over 50 years of age, cases of people below 50 are increasing in recent years. According to the *Observatorio del Cáncer de la Asociación Española Contra el Cáncer*, in 2024, 41167 new cases of colorectal cancer were diagnosed, becoming the most common malignant tumour in Spain. It is the second most common cancer for men and women, after prostate and breast cancer, respectively. It is also an important cause of death as, in 2024, 15401 people died of colorectal cancer in Spain [1].

Colorectal cancer encompasses both rectal and colon cancers. This project is focused on rectal cancer (RC), which are the tumours affecting the last 15 cm of the colon, the rectum. Patients with rectal cancer experience changes in bowel habits, such as diarrhoea or constipation, changes in faeces' shape, abdominal pain, presence of blood in the stool, unexplained weight loss and fatigue [2].

There are several risk factors associated with rectal cancer: family or personal history of rectal cancer or related cancers such as colon or ovarian cancer, high alcohol intake, smoking tobacco, obesity, old age, black ethnicity, a diet low in vegetables and high in red or processed meat or having syndromes or diseases related to rectal cancer such as Chron's disease or Lynch syndrome [2] [3].

Early diagnosis is crucial to induce a better response to the treatment. Many countries, including Spain, have a screening program for populations with a higher risk: people of ages between 50 and 69 years. This has yielded a growing number of early diagnosis rectal cancers which may benefit from less invasive treatments. Endoscopy is the gold standard in diagnosis of RC although other imaging techniques, particularly Magnetic Resonance Imaging (MRI), are routinely used to stage the tumour, assess prognosis and treatment response [4]. The most used sequences are high resolution T2-weighted (T2w) images and Diffusion Weighted Imaging (DWI) using b-values and ADC maps.

Staging is carried out using the TNM standard and tumours can be classified into early rectal cancer or locally advanced rectal cancer (LARC). The treatment for the latter is neoadjuvant chemoradiotherapy (nCRT), to downsize the tumour, prior to the total mesorectal excision (TME) surgery [5]. This project will be focused on LARC patients.

Response to nCRT is assessed in a second MRI using Mandard's Tumour Regression Grading system (TRG) [77]. Patients with TRG1 and TRG2 can be classified as good responders whereas those with TRG3, TRG4 and TRG5 can be classified as non-responders to nCRT [6].

1.1 Motivation

It is estimated that around 20% to 25% of LARC patients will have a complete response to nCRT, known as pathologic complete response (pCR), and might be eligible to skip surgery and instead opt for a more conservative Watch and Wait strategy (W&W) or a less invasive surgery (like transanal endoscopic microsurgery (TEM)) with comparable survival outcomes [7]. Avoiding surgery also improves the quality of life of the patient as they avoid the morbidity associated with it: stoma creation, risk of sexual dysfunction or altered bowel habits.

Predicting the response to nCRT using data obtained from the staging MRI [8] could potentially help professionals personalize the treatment to each patient and improve the outcome of the

chemoradiotherapy. If patients that will likely become non-responders are identified early, the doctors can adjust the treatment accordingly, reducing potential adverse effects. Identifying which factors are significant for tumour outcome can help create more accurate models, improve therapy prescriptions and potentially increase the number of pCR.

Another way of avoiding unnecessary excision surgeries is to improve differentiation between pCR and non-responders before the surgery, using information extracted from the second MRI [5].

1.2 Objectives

The main objective of the project is to build an Artificial Intelligence model capable of predicting the response of LARC after neoadjuvant chemoradiotherapy. This breaks down into several subobjectives:

i. Determine clinical features//image markers that are key in predicting LARC evolution.
ii. Verify similar studies already carried out using data obtained from *Hospital de la Santa Creu i Sant Pau*.

And if the goal is achieved, the result of this project will be the development of a tool to help tailor neoadjuvant therapy to each patient.

The initial hypothesis of the project is that the response to neoadjuvant chemoradiotherapy in patients with locally advanced rectal cancer can be predicted using machine learning models that use radiomics and/or clinical data. Moreover, the model that is expected to work better is the one combining both radiomics and clinical data.

1.3 Methodology and structure

The project described in this report follows a methodology similar to a radiomics workflow and an AI pipeline. It includes 84 patients diagnosed with LARC that underwent nCRT between June 2018 to September 2024. Moreover, all data comes directly from or is derived from the 84 patient's clinical histories of *Hospital de Santa Creu i Sant Pau* in Barcelona. This study is a proof of concept for validation of a methodological approach using a real case database of tumour patients. The database is a retrospective cohort of oncological patients acquired from the hospital. This study is under evaluation of the Ethical Committee of the hospital.

The project is divided in four main parts: tumour segmentation, preparation of the databases, model training/validation, and analysis of results. Tumours were segmented on the T2w sequence using the open-source software 3DSlicer (version 5.6.2). All segmentations were corrected by a senior radiologist.

Afterwards, the data was extracted. The chosen clinical features were collected from the hospitals databases and the radiomics features were extracted from the tumour segmentation masks and the calculated tumour border masks. Radiomic extraction was carried out with the Python library *PyRadiomics* after normalizing the MRI studies. To complete the databases, the samples were classified as responders or non-responders using the Tumour Regression Grade (TRG) indicator. The most relevant radiomic characteristics were selected using dimensionality reduction strategies: Kruskall-Wallis non-parametric statistical test and the Spearman's correlation coefficient.

Finally, with the post-processed datasets, Machine Learning models were trained. Nine models (Decision Tree, Random Forest, Support Vector Machine, Logistic Regression, Naïve Bayes, Multilayer Perceptron,

Gradient Boosting, XGBoost, LightGBM) were trained for each of the seven possible feature set combinations (only tumour radiomics, only border radiomics, only clinical, tumour + border, clinical + tumour, clinical + border, clinical + tumour + border). Each model was optimized using hyperparameter tuning via 5-Fold Stratified Cross Validation grid search. Afterwards, the performance metrics and the confusion matrix were computed with the validation set, consisting of 15% of the data. For the models with the best results, the model's feature importance plots and ROC curves were calculated. Feature importance plots were used to analyse the relevant features and compare with findings from the literature.

This report is structured in sections that discuss the different steps of the process and the viability of it. Firstly, after the present introduction, the general theory concepts necessary to understand the aim of the work and a review of the available literature will be discussed in section 2. Afterwards, sections 3 and 4 discuss the different possible solutions and explain in detail the methodology applied during the project. Section 4 also contains the results of every step and the final analysis of results.

Then, four sections detailing the technical aspects of project management, such as its planning or budget will be included. In section 5, the planning of the project is discussed with the development of a WBS dictionary, and PERT and GANTT diagrams. Section 6 contains a SWOT analysis to evaluate technical feasibility of the project. Section 7 is dedicated to explaining project budget. Finally, in section 8 the different laws and regulations that have to be considered for the development of the project are reviewed.

To wrap up, section 9 discusses the conclusions of the project. Afterwards the references used during the report and the annexes containing additional information will also be included.

1.4 Scope and limitations

This project will take place in the imaging centre facilities of *Hospital de la Santa Creu i Sant Pau* and will last 4 months, from February 2025 to May 2025. However, the MRI studies included in the project are taken from June 2018 to September 2024.

This project is focused exclusively on predicting treatment response using features extracted from the pre-treatment staging MRI (T2w sequence), prior to neoadjuvant therapy. The project scope includes the obtention of patients' data and MRIs, tumour segmentation in pre-nCRT MRI studies and obtention and analysis of radiomics data, the design and development of a database storing all the data to train the ML models, the development and training of the models and the extraction of conclusions from the models.

Using genetical, pharmacological or histological characteristics to predict response is outside of the scope of this project as it is limited to radiomic characteristics and demographic and staging clinical features.

As discussed in the motivation, Using the data from both staging and post-nCRT MRIs to be able to properly differentiate between pathological complete responders (pCR) and non-responders before the surgery can help avoid unnecessary surgeries in cured patients and prescribe excision surgeries, with increased confidence, in patients that could have been falsely classified as complete responders elsewise [5]. However, the project will not include the segmentation nor analysis of the post-nCRT MRI studies thus this research direction is outside of the scope.

2. Background

2.1 General concepts

In this section, the main background theory to comprehend what is going to be covered throughout this project will be explained. Firstly, the basis of rectal tumour anatomy will be explained, later the clinical protocols to acquire rectal MRI images and the systems of tumour staging will be discussed. Finally, radiomics and Artificial Intelligence will be explained.

2.1.1 Rectal tumour anatomy

The normal rectal wall is formed by five layers, from the lumen to outside they are in order: the mucosa (containing the epithelium and lamina propria), the muscularis mucosae, the submucosa, the muscularis propria (including the circular and longitudinal muscle) and the serosa/perirectal fat (*Figure 1*). All five layers can be seen in an T2w MRI but usually the mucosa and submucosa are seen as a single dark layer. Rectal cancers form in the mucosa, usually from existing polyps, and advance radially through the layers of the rectal wall. Longitudinal tumour spread is uncommon. If they breach the bowel wall, spread continues into the mesorectum, which is the fatty tissue surrounding the rectum that also contains the rectal vessels, and after breaching the mesorectal fascia, which surrounds the mesorectum, they advance progressively into adjacent pelvic structures, like the prostate or the seminal vesicles, in men, or the uterus, in women. Metastases occur via Locoregional lymphatic spread, i.e., spread to the nearby lymph nodes, and Hematogenous spread, i.e., spread through the bloodstream to the liver and the lungs, the latter mostly in tumours of the lower rectum. Metastases to the brain and skeleton are less common [3].

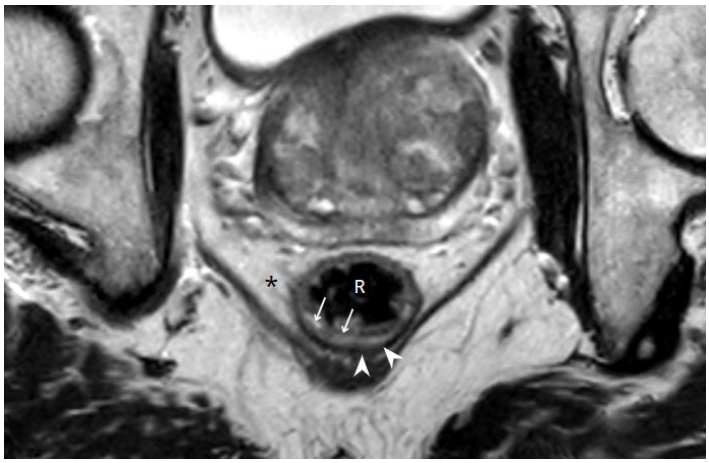

Figure 1. T2w MRI of a rectum. R: lumen, Arrows: mucosa and submucosa, Arrowheads: muscularis propria, Asterisk: mesorectum.

2.1.2 Imaging Protocol

Magnetic Resonance Imaging is the gold standard for rectal tumour staging and assessment of treatment response in rectal cancer [9]. MRI detects the realignment of protons with the magnetic field generated by the machine, after their alignment is temporarily disrupted by a radiofrequency pulse. The process by which the protons return to equilibrium is called relaxation and there are two types: longitudinal relaxation (z plane), described by the T1 time constant, and transverse relaxation (x-y plane), described by the T2 time constant. In MRI, T1-weighted and T2-weighted images measure the T1 and T2 relaxation times of the tissues, respectively [10].

Pelvic phased-array multichannel coils are the standard in rectal tumour imaging as they provide good signal-to-noise ratio, high spatial resolution and are comfortable for the patient [11]. Regarding bowel preparation, using antiperistaltic drugs can be beneficial to reduce motion artifacts related to rectal peristalsis [12]. Endorectal filling with gel or contrast is not indicated as, although it can help detect small tumours, it compresses mesorectal fat due to rectal distension which may hinder the detection of lymph nodes and the estimation of mesorectal fascia infiltration [11].

High-resolution scans are taken at a slice thickness of 3 mm or below as higher slice thickness can lead to loss of anatomical information [12]. The MRI sequences routinely used in tumour diagnosis and staging are T2-weighted Turbo Spin Echo (T2w-TSE), also known as Fast Spin Echo (FSE), and Diffusion Weighted Imaging (DWI), which comprise b-values and ADC maps. TSE is an MRI pulse sequence used routinely in medical practice which collects multiple echoes during a single repetition time (TR), as depicted in *Figure 2*.

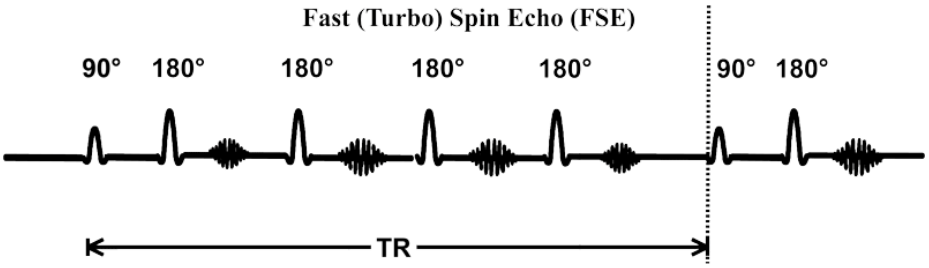

Figure 2. Schematic of the Fast Soin Echo sequence [13].

Firstly, T2w images in the axial, coronal and sagittal planes are taken to localize the tumour and its long-axis. In T2w images, the tumour presents an intermediate signal intensity, a characteristic tone often referred as "evil-grey". Mucinous tumours return a higher signal, appearing whiter, due to their fluid content [3]. This can be seen in *Figure 3,* where the white spots on the tumour in the long-axis image (left) are mucus-like material that is surrounded by solid tumour, with its characteristic "evil-grey" signal. Then two sequences, parallel (known as long-axis) and perpendicular (known as short-axis) to the axis of the tumour, are acquired as shown in *Figure 3* [11]. These sequences, especially the perpendicular one, help visualize the tumour better and to correctly stage the proximity of the tumour to the mesorectal fascia. These sequences can also be used to assess lymph node morphology [3].

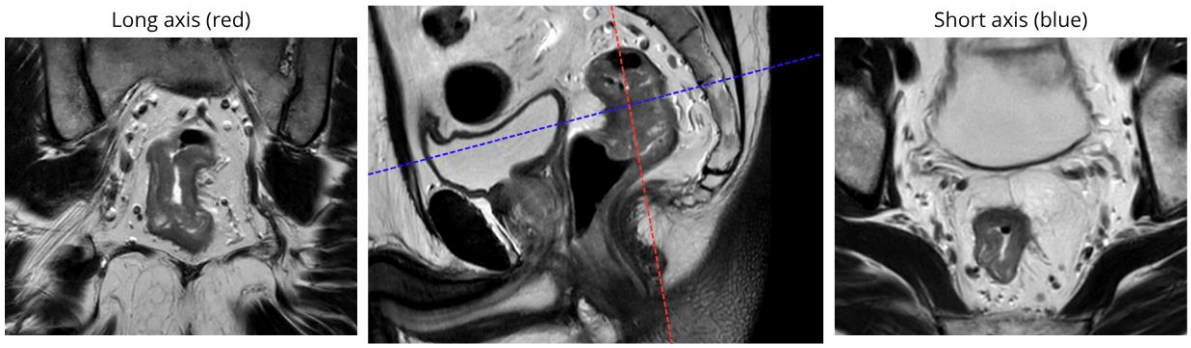

Figure 3. Short and Long-axis of a rectal tumour. Left: Long-axis of the tumour. Middle: slice in the sagittal plane of the tumour with the tumour axes. Right: short-axis of the tumour

DWI images should be taken in the perpendicular plane to the axis of the tumour with the same angulation [12]. DWI images measure the diffusion of water molecules, which follows a Brownian motion. Water molecules move freely in unconstrained tissues (isotropic movement) and move in a restricted manner

in structured environments (anisotropic movement). Since not all structures are oriented in the same manner, several directions have to be measured to evaluate diffusion. Diffusion weighting is expressed as a b-value (measured in s/mm²), which increases with stronger diffusion weighting. When b-values increase, signal from freely moving water is attenuated whereas signal from areas where diffusion is restricted is brighter. Apparent Diffusion Coefficient (ADC) maps can be calculated from different b-values and provide a visual representation of tissue water diffusivity [10]. Rectal tumours appear bright in images with high b-values and dark in ADC maps, as they have restricted water diffusion, as seen in *Figure 4* [3].

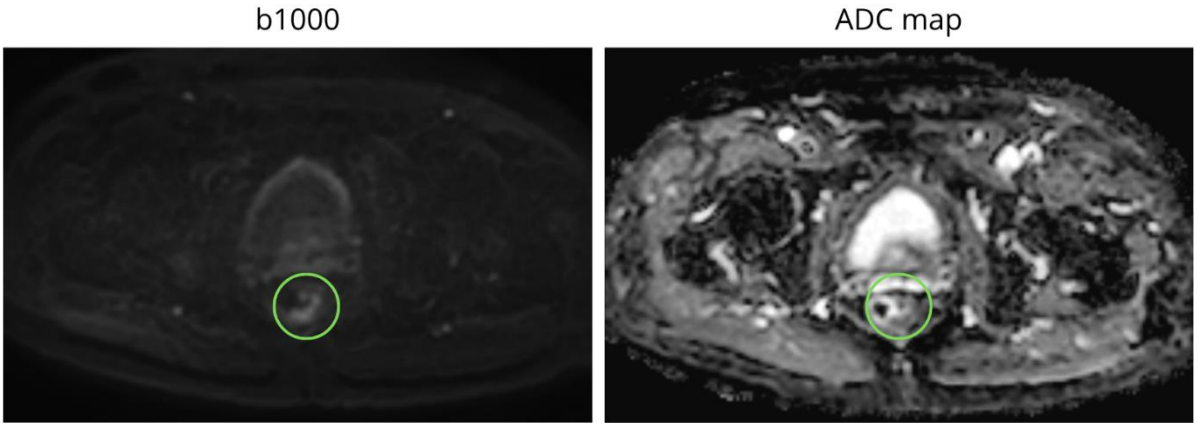

Figure 4. DWI sequences b1000 (left) and ADC map (right). Green circle surrounds the tumour.

2.1.3 Tumour Staging, Treatment and Restaging

Two MRI scans are taken in clinical routine. The first MRI scan is used to stage the tumour before neoadjuvant therapy using the TNM (Tumour, Node, Metastasis) staging system to determine the state of the tumour and avoid prescription of unnecessary nCRT in patients with early cancer. More detailed description of the staging can be found in *Annex 1*.

- T is used to describe the size of the tumour and its invasion into adjacent tissues. T0 indicates no presence of tumour whereas T4 indicates that the tumour has extended through all the layers of the submucosa and invaded the peritoneum or adjacent structures (*Figure 5*).
- N indicates the dissemination of cancer to nearby lymph nodes. Lymph nodes act as filters of harmful substances, including cancer cells, and are a common site of metastases. The presence of cancer cells in these structures is related with worse tumour prognosis. It can take values from N0 (no evidence of nodal spread) to N3 (high distal nodal spread).
- M is used to identify the presence of metastases, the spread of the tumour beyond the regional lymph nodes. It can be M0, no presence of metastasis, or M1, evidence of metastasis [14].

According to TNM staging, RC can be classified into early rectal cancer (T1–2 and N0), in which the standard treatment is surgical excision, and locally advanced rectal cancer or LARC (T3–4 and/or N1-2), in which the standard treatment is neoadjuvant chemoradiotherapy (nCRT) followed by total mesorectal excision surgery (TME) [5]. With the first MRI, other parameters are also assessed related to the localization of the tumour, the length of the tumour, extramural venous invasion and the involvement of the mesorectal fascia (the latter only in T3 tumours).

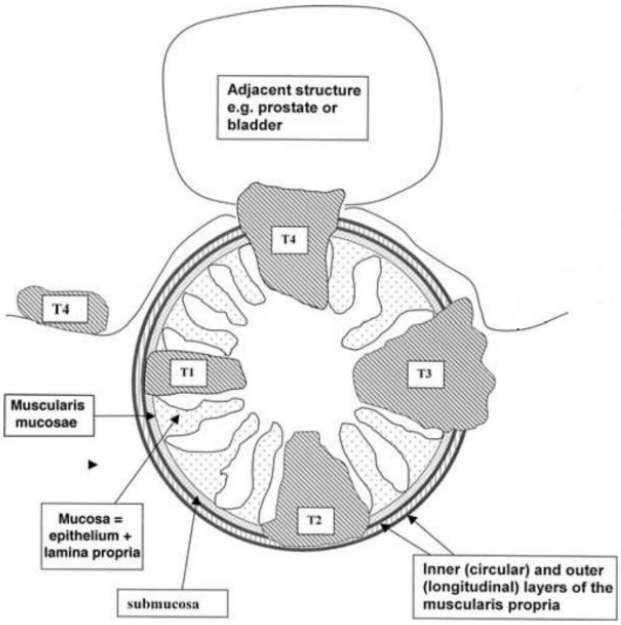

Figure 5. Schematic of the T-staging of rectal cancer [3].

The goal of nCRT in patients with LARC is to downsize and downstage the tumour before the surgery to reduce the risk of leaving tumoral tissue behind during the surgery. The response to nCRT is assessed in a second MRI using the Tumour Regression Grading system (TRG) which describes the extent of tumour in the first MRI that is replaced by fibrosis signal in the second MRI. TRG0 shows no residual tumour whereas TRG5 shows predominant tumour in the second MRI [6]. More detailed description of TRG can be found in *Annex 1*. Patients with TRG1 and TRG2 can be classified as good responders whereas those with TRG3, TRG4 and TRG5 can be classified as non-responders to nCRT. yTNM, tumour staging after nCRT can also be determined in the second MRI after nCRT. Both the TRG and the yTNM are later confirmed during the total mesorectal excision surgery (ypTNM) [6]. This surgery includes the removal of the whole tumour with the surrounding mesorectum invested in the mesorectal fascia. In TNM, the prefix y indicates staging if the patient has had neoadjuvant chemotherapy and the prefix p indicates tumour staging applied to the histological specimen, in other words, the tumour that has been removed during surgery. It is worth of noting that the TNM staging carried out before nCRT is often referred as cTNM, with the c prefix meaning that it is the clinical staging [3].

2.1.4 Radiomics

Traditionally, medical images have been analysed qualitatively by visually assessing the shape, size, intensity of tumours or other structures. Radiomics offers an alternative way to analyse the image by extracting not only quantitative features related to the shape and size but also histogram related features or texture analysis. These features allow the extraction of information that is not apparent to the human eye. Some shape features that can be extracted, which are more complex than what visual inspection allows, are sphericity, compactness, elongation or axis length. Texture features are statistical relationships of voxel intensities within the region of interest (ROI). Thus, radiomics offer a non-invasive and cost-effective way to mine big amounts of meaningful data from medical images [15].

There are many softwares and libraries capable of extracting radiomic features. In this project, features will be extracted using the library PyRadiomics, an open-source Python package [16]. Features are extracted from a segmentation mask by calculating single values for a region of interest ("segment-based"

extraction) or by generating feature maps ("voxel-based" extraction) [17]. In this project the features extracted are First Order Statistics (19 features), Shape-based (3D) (16 features), Gray Level Co-occurrence Matrix (24 features), Gray Level Run Length Matrix (16 features), Gray Level Size Zone Matrix (16 features), Neighbouring Gray Tone Difference Matrix (5 features) and Gray Level Dependence Matrix (14 features). More information about which features are extracted, and their meaning can be found in the library's documentation [17].

2.1.5 Artificial Intelligence

The term Artificial Intelligence (AI) was formally proposed in a conference at Dartmouth University in 1956. Artificial Intelligence is defined as the capability of computer systems to perform complex tasks such as decision making or creating content, that would usually require or are associated with human intelligence [18]. AI has experienced a big development over recent years although the origins of the technology can be traced back to 70 years ago. Nowadays, the amount and dimensionality of data has increased, known as big data, enabling the development of more complex AI models. The development of advanced hardware capable of supporting the computational power needed to analyse and train the algorithms also has enabled the rapid evolution of AI [19].

Machine Learning (ML) is an important part of AI which refers to the use of an algorithm that learns from data. ML's types of problems are prediction (classification and regression), clustering, and dimensionality reduction and the main learning methods are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Supervised learning is when the algorithm learns from labelled data, usually to solve a classification or regression problem. When the data is unlabelled, the learning method is unsupervised learning, usually used for clustering problems to reveal groups in the dataset or for dimensionality reduction problems to reduce the number of features in a dataset. Semi-supervised learning is a mix of the last two, but it is not used frequently [19]. Lastly, in reinforcement learning, the computer explores an environment and interacts with it receiving rewards, either positive or negative, from the actions. The algorithm is optimized based on these rewards. This idea comes from behavioural psychology and imitates the way that humans and animals learn. Reinforcement learning is used to train a computer to perform tasks such as playing different kinds of games or autonomously driving a car around a specific track [20].

Another subset of Machine Learning is Deep Learning (DL) which uses multilayer neural networks, inspired by the structure of human neurons, as the architecture for the algorithms. DL can process large batches of data, such as images or text, without the need of complex data preprocessing or feature engineering [21]. Examples of widely used Deep Learning applications are Large Language Models (LLMs), which are capable of understanding and generating human-like text and are used to translate, summarize, rewrite, analyse or classify text [21], and Generative AI, which learns patterns from training data to generate new content such as text, images, or audio. [22].

Nowadays, AI is used in multiple industries for various purposes. Natural Language Processing (NLP) refers to a technology that enables computers to understand natural human language. It represents an intersection between language information processing and artificial intelligence. This involves both written text and speech recognition, where the computer hears sounds, converts them into words and then extracts the meaning of the message [19]. In the medical industry, AI is used to assist diagnosis, to

optimize treatment dosage, to assist in deciding personalized treatments, to predict and assess the risk of diseases in big populations, to develop new drugs, among many more [21].

In imaging AI is widely used in computer vision, which is the ability for computers to understand the world with images, similarly to how humans use vision. It is mostly used for facial, pattern and image recognition. Specifically, in medical imaging, AI is used to detect lesions in medical images such as tumours, fractures, pneumonia or retinal diseases, to plan surgeries by building detailed 3D models obtained from medical images or to carry out automatic organ and lesion segmentation [22].

2.2 State of the art

Some sources argue that the methods described in the previous section are not optimal and present some disadvantages that have to be considered. Firstly, it is discussed that while the TRG is a good predictor of pathologic response with high specificity, it may not be the optimal ground truth as it presents low sensitivity [6]. In other words, TRG is good at identifying people without positive response but not at identifying subjects with bad response. Moreover, TRG is assessed via visual inspection and the value can change based on the experience of the reviewer. The quality of the MRI study also affects this parameter [23]. Using response based on the ypTNM as ground truth could help mitigate this error. However, it is not yet clear what a good response would be based on the evolution between cTNM and ypTNM.

Another drawback is that since radiomic-based datasets often have a small number of samples and the number of extracted features exceeds it; datasets fall into the “curse of dimensionality” and the models are prone to overfitting. This is the case for the dataset used in this project. Dimensionality reduction and feature selection strategies can be implemented to identify and remove redundant features such as ranking features by correlation coefficients, by importance metrics or using methods to transform the feature space to a new set of features with reduced dimensionality such as principal component analysis (PCA) or linear discriminant analysis (LDA) [15]. The dimensionality reduction strategies implemented will be explained in later sections.

A disadvantage of Artificial Intelligence, particularly important in diagnostic medical applications, is model explainability which refers to the techniques used to interpret and understand the model’s decision process. In AI, the more complex the model the less self-explainable it is. For instance, models such as Decision Trees are already explainable but Neural Networks, which are more complex and provide better accuracy, are often regarded as a “black box” meaning the user cannot know the reasons behind the predicted output. Explainable AI methods are used to open this box and understand the decision-making process [24]. This is important in diagnostic medical applications as the doctors should be able to tell the patients the cause of their health problem. Some explainable AI methods (XAI) include feature importance methods such as LIME or SHAP or the development of a second model to provide explanations of the original model like TREPAN [24].

2.3 State of the situation

Similar studies to this project have been carried out. Several algorithms and methodologies have been used in these studies and this section is reviewing some of them.

A study used tumour segmentations on pretreatment T2w MRIs to train a Deep Learning model to predict survival in patients with rectal cancer. They used three data cohorts: a first cohort used for model selection and hyperparameter optimization, a second cohort used as a separate internal test cohort, and a third cohort with data collected from a different hospital used as an external test cohort. Patient survival was predicted by classifying patients into high or low risk groups based on a fixed cutoff calculated in the training set. The best model had a C-index of 0.82 and performance was improved using a multimodal model, which also included the pretreatment carcinoembryonic antigen level as input, with a C-index of 0.86, for the validation set, and 0.67, for the external test set [25].

Another study used both T2w and clinical features to train a model to predict treatment response of patients with LARC at an early stage. Models were trained using radiomics features extracted from the tumour core and the tumour border, using clinical features based on T2w images and with both datasets simultaneously. The models yielded an AUC of 0.7, 0.684 and 0.793 respectively, showing that when using both radiomics data and clinical data, the predictions improve. Moreover, the pipeline used in this study is very similar to the one applied in this project, providing a useful benchmark [26].

A separate study focused on discriminating non-responders from complete-responders and partial-responders. Data was extracted from segmentations in the pre-nCRT and during nCRT MRI studies. Ratios and differences between these data points were also calculated to obtain extra datapoints. In this study, a Random Forest classifier with 2000 decision trees was used to build the model. Despite the limited number of patients in the study, 55, the model obtained a main AUC of 0.85 [7].

Another single-centre study trained Machine Learning models to assess complete and positive response to nCRT with 131 patients. Seven separate models were trained using different features: one with exclusively clinical data, one with exclusively tumour radiomics (from T2w), one with mesorectum radiomics data (from T2w), one with clinical and tumour data, one with clinical and mesorectum data and one with the three datasets. The model with the best performance was the one trained with exclusively clinical features, with an AUC of 0.69 for complete response to nCRT and 0.64 for positive response. This is uncommon as similar studies, such as the previously explained, found that a combination of clinical and radiomics features often yields better results. The authors discussed that the reason is likely because of the way that the radiomic regions were segmented, as it is not standardized and the technique varies significantly within literature [9].

An AI radiopathomics model using data form pretreatment MRI and haematoxylin and eosin-stained biopsy slides showed that it could predict pCR with high accuracy. The model, called RAPIDS, was validated on two cohorts showing accuracies of 0.86 and 0.87, respectively, and an AUC of 0.888 [27]. Another study focused in mesorectal fat T2w MRI radiomics showed that these features can predict pCR with an AUC of 0.89, local and distant recurrence and post-treatment T and N categories [28]. Another study used both T2w and DWI MRI sequences to extract radiomics data and used the surgical histopathologic analysis as the ground truth response. They built three models: a T2w, a DWI and a merged one. The best model was the T2w, with am AUC of 0.82 [29].

2.4 Market analysis

Since the aim of this project is not to develop a sellable product but an algorithm/software that can be used as a predictor of response to therapy, the market is comprised of patients whose quality of life could

potentially improve thanks to the software and of hospitals who could reduce expenses in treatment by opting for a more conservative treatment approach.

However, similar technologies capable of detecting lesions in medical images are commercially available in the market. *Sycai Medical* offers a software capable of detecting abdominal cancers at early stages. Moreover, it also offers an automatic single-lesion tracking over time, a patented technology [30]. *SimBioSys* has developed a tool that automatically segments breast tumours and surrounding tissues from DCE MRIs, creates a 3D model and, using information at the cellular level obtained from the tumour biopsy, it creates a digital twin. This twin allows doctors to answer questions related to tumour staging, treatment prescription, treatment response and patient survival [31]. *Veye Lung Nodules*, a product by a company called *aidence*, is a CE certified product used in hospitals in Europe that is able to detect lung nodes in lung CT-scans. Furthermore, it quantifies the size and volume of the nodes, it segments them, it classifies them into solid or sub-solid nodes and also assesses nodal growth by calculating the growth percentage and the volume doubling time. This tool is integrated in the routine clinical workflow without adding any workload to the doctors [32]. Lastly, there are also companies dedicated to improving the radiology department's workflow like *Enlitic*. Some solutions, based on AI, offered by the company are *ENDEX*™*,* which standardizes DICOM images and data into a universal ontology, *ENCOG*™*,* which uses Computer Vision and NLP technologies to anonymize patient data from medical imaging data, and *Migratek*®, which facilitates DICOM data migration between platforms. According to the company, these systems optimize worklists by showing only relevant patients, automate study routing by assigning studies to the appropriate radiologists and allow customized display of studies [33].

There are also related patented technologies. Informatics radiomics integration system is designed to analyse and integrate medical data such as radiomic and clinical data for classification into different groups, using machine learning methods, to improve visualization using 3D representations or heatmaps. This helps doctors understand better the complexity of the data. It can be used for disease diagnosis and staging, personalized medicine and research [34]. Another patented technology is a platform that generates 3D models of tumours from 2D medical images segmentations and identifies radiomic features that can be used as imaging biomarkers for cancer diagnosis and prognosis, assessment of treatment response and personalized therapy design. These features are compared to eliminate redundancies and the selected ones are then compared to separate genomic data and/or outcome data [35].

In 2025, it is forecasted that 4938 patients will be diagnosed with colorectal cancer in Catalonia, according to the *Pla contra el Càncer de Catalunya* [36]. The price of a standard treatment includes the first MRI study, the nCRT, the second MRI and the potential surgery, later post-surgical care and stoma care. By tailoring the treatment to each patient, using the predictions of the software developed, patients may be eligible to skip surgery and all the post treatment care, reducing overall treatment cost.

3. Concept engineering

In this section, the different possible solutions will be discussed and the best will be chosen to optimize the pipeline.

3.1 Data acquisition

3.1.1 Database features

The final dataset will include both radiomics data and clinical data. However, the clinical data has to be chosen. According to similar studies [3, 9, 5, 26], the clinical features most commonly associated to nCRT response are: the tumour location (high rectum = 3, middle rectum = 2, low rectum = 1), whole tumour volume (mm^3), tumour cranio-caudal extension (mm), distance from the internal anal sphincter (mm), mesorectal fascia infiltration (FMR) (absent = 2, present = 1), extramural vascular invasion (EMVI) (absent = 2, present = 1), extramural depth of invasion (mm), T-stage (1–4), N-stage (1–2), overall clinical stage (TNM), nCRT to surgery interval, carcinoembryonic antigen (CEA) level, microsatellite instability (MSI), KRAS mutation status, statin use, tumour grade (degree of cellular differentiation), nCRT radiation dose, age, gender, body mass index, mucinous histology, family history (first or second-degree relative with colon or rectal cancer) and the pathological tumour type.

Whole tumour volume (mm^3) was not selected as this feature is already extracted with radiomics. Carcinoembryonic antigen (CEA) level is a non-specific blood biomarker that is related to cancer, but it is not used to diagnose it, so it was discarded [37]. Microsatellite instability (MSI), detected in 15% of colorectal cancers, and KRAS mutation status are genetical variables [38] that are usually tested in clinical practice, especially in advanced cancers. They are discarded as this study does not focus on genetic factors. Tumour grade (degree of cellular differentiation), mucinous histology and the pathological tumour type were discarded as they are histological variables and the study is not focused on tumour histology. Statin use refers to the use of statin medications, used to lower cholesterol levels [39], and it was discarded because it is not readily available for most patients in the study. Extramural depth of invasion (mm), nCRT to surgery interval, nCRT radiation dose, body mass index, and family history were discarded, despite being relevant, because of lack of data in the hospital's databases. The rest of the variables were chosen for extraction and were available in the hospital's databases as they are usually part of the staging report.

Another aspect to consider when building the database is the ground truth. For this project, the ground truth can be based on the TRG, of the second MRI, or on the ypTNM, obtained after surgery. As explained in the section 2.2, the TRG may not be the best indicator. However, the classification of responders and non-responders is clear [6] thus TRG will be used as it is the most straightforward method.

3.1.2 MRI sequence to segment

As explained in the background section, during a rectal MRI several sequences are acquired. Selecting the proper sequence to segment is a pressing matter in this work. After a bibliography review and expert consultation, it was concluded that the most commonly used sequence in clinical practice and segmentation is the short-axis plane as it is where the tumour is better visualised. This is also consistent with the methodologies of the similar studies reviewed in section 2.3 thus T2w and DWI (ADC + b1000)

images in the short-axis plane preliminarily were selected for segmentation. This would allow more data to be extracted as they are three separate sequences that provide different information.

However, since this process would require three separate segmentations, registering the images to a common space was attempted. Both rigid (only image rotation and translation) and affine (image rotation, translation, shearing and scaling) [40] registrations were carried out. After reviewing some registered cases, it was concluded that registration was not possible because, although the protocol is to acquire DWI images using the same angulation as the T2w short axis [12], most of the available diffusion sequences were acquired in a natural axial plane instead. Thus, the images were not correctly superposed, and segmentations could not be reused.

Another difficulty, innate to the rectum, was that it is a very changing organ because, unlike the brain for example, it is not constrained in a rigid structure [3] and, since the T2w and the DWI sequences are not acquired simultaneously, a gas bubble or a peristaltic movement can change the shape of the rectum [5] and render the first segmentation unusable for the other sequences even if they were properly registered. Moreover, the slice thickness of the available T2w sequences was of 3 mm whereas for DWI was of 4-5 mm. Although this should not pose a problem during registration because information would be interpolated, it could lead to loss of spatial resolution or interpolation artifacts [40].

Finally, it was decided that, as most of the studies only use T2w images to extract radiomics data and this is the most relevant sequence in routine clinical practice, the T2w short-axis sequence was the only chosen sequence for segmentation.

3.2 Tumour segmentation

Segmentation in image processing is the process of dividing an image into one or multiple regions of interest (ROIs) based on criteria such as region homogeneity or edges between heterogeneous regions [41]. There are multiple methods, strategies and programs to carry out this process that will be discussed in this section, specifically for medical imaging.

3.2.1 Segmentation methods

Segmentations can be carried out manually, in a semi-automatic way or automatically. Manual segmentations are executed by experts manually delineating the ROI usually slice by slice. The resulting segmentations are very precise as complex cases are evaluated by an expert. However, this method is very time-consuming, it is subjective to user variability, and observer bias can be introduced [42].

Automatic segmentations are carried out by an algorithm. The model, previously trained, automatically identifies and delineates the ROI without human intervention. This allows faster processing of large datasets and the introduction of real-time segmentation applications. However, segmentations may be less accurate and are especially compromised in images with complicated anatomy or poor quality [42]. Depending on the training data, bias might also be introduced [43].

Semi-automatic segmentation combines user input with an algorithm. There are several methods such as placing a seed in the ROI for the algorithm to delineate the rest of the area [42]. These methods are faster than manual but slower than automatic. Moreover, the user should have some level of expertise to

correct the resulting ROIs. The output also depends on the image quality and the algorithm performance [43].

Several papers describe developments of automatic methods. One study described a convolutional neural network (CNN) architecture based on a densely connected NN to segment LARC in T2w 3D images. The method also included a 3D level-set algorithm to refine the contours of the predicted segmentations. They reported good results (Dice similarity coefficient (DSC), recall rate (RR), and average surface distance (ASD) of 0.8585 ± 0.0184, 0.8719 ± 0.0195, and 2.5401 ± 2.402, respectively) when comparing the predictions with the ground truth (segmentations by a radiologist) [44]. In another paper they developed a U-net based algorithm to automatically segment the outer rectal wall, lumen, and perirectal fat regions on post-nCRT 2D T2w MRI scans. They reported good performance of the algorithm (wall DSC = 0.920, lumen DSC = 0.895) in comparison with radiologist segmentation. They reported better results than multi-class algorithms, i.e., algorithms that can segment multiple ROIs simultaneously [45]. Finally, a different study used a U-net algorithm to segment LARC in DWI images. The results were compared with a semi-automatic method based on a grey-level threshold. The U-net showed better results (DSC = 0.675 ± 0.144) than the semi-automatic method (DSC = 0.614 ± 0.22) when compared with the ground truth segmentations made by radiologists [46].

Despite the good results reported by the studies, none of the models is medically approved thus segmentations will be carried out manually or, if the segmentation software allows, using assistance of semi-automatic tools. This will allow more control and precision of the segmentations over using an automatic method.

3.2.2 Segmentation programs

Several software programs are used to segment medical images both in research and in clinic. One of them is 3DSlicer, an open-source software for image visualization, segmentation, registration, and analysis widely used in research projects with medical images. It is important to point out that the software is not FDA approved, and it is not intended for clinical use although it can be used in research [47] [48]. This software is used in some of the papers reviewed during this report [7, 26, 45].

3DSlicer is structured in modules that permit the different functionalities. In the software, multiple image planes can be seen simultaneously and overlayed. Moreover, there are statistics modules that allow extraction of quantitative data from the images. Most importantly, the software has manual and semi-automatic segmentation tools, located in the Segmentations and the Segment Editor modules. The learning curve of this program is quite steep but there are plenty of tutorials and documentation available. Additionally, it has a broad variety of available extensions that add extra functionalities such as a radiomics module used to extract radiomics features, based on the PyRadiomics python package [49].

Another software is syngo.via by Siemens Healthineers. It is FDA-approved and has a CE-marking, making it usable in clinical practice. The platform offers multi-modality reading with a friendly user interface with accessibility tools that mark anatomical structures in images. It is also structured in modules depending on the specialization: oncology, cardiology, neurology… Moreover, it offers the possibility to perform 3D manual segmentations and quantification of the lesions. It is an easy-to-use software and requires little training [50].

For clinical use, the software is capable of automatically finding similar cases in the hospital database references. It is mainly used for diagnostic, clinical report writing and to plan treatments or procedures. Image scans can also be rendered into a 3D volume for better visualization. However, this software requires a commercial licence to be used which is not free [50].

Although it requires more training, the chosen software was 3DSlicer because it is open-source and versatile.

3.3 Image normalization

Before extracting the features, images have to be normalized, an especially important process in this study as the MRIs are taken from five different machines from the hospital. In the reviewed studies, two methods of normalization are mainly carried out: Z-score normalization [5] or using the mean intensity of the obturator internus muscle as a reference value [7, 45].

Z-score normalization can be applied using the *StandardScaler()* function from the Scikit-learn Python library [5] and it standardizes the image so that it has a mean intensity of zero and a standard deviation of one. This function applies the following equation through all the data

$$z = \frac{x - \mu}{\sigma}$$

Equation 1. Z-score normalization.

were x is the non-standardized pixel, μ is the mean intensity, σ is the standard deviation and z is the standardized pixel [51].

The other option, using the mean intensity of the obturator internus muscle, is carried out by drawing a specific sized ROI on the muscle, extracting the mean and standard deviation of the intensity and using the parameters to normalize the data with an equation like *Equation 1*. However, in some of the MRI, the obturator internus is not visualized well. Thus, the gluteus maximus muscle should be used instead as it is quite uniform for all the patients and easy to spot, located on the rear part of the images [3].

Both methods are very similar, but the latter would increase the workload of the project significantly as additional ROIs would have to be drawn on each study and additional parameters would have to be extracted. Moreover, the studies where the method was applied had good results, but they were not substantially higher than studies that used Z-score normalization. In conclusion, for simplicity and to reduce workload the Z-score method was preferred.

3.4 Radiomic feature extraction

Regarding radiomic feature extraction, the features will be extracted using the PyRadiomics library [16]. The features extracted with this module are in compliance with feature definitions as described by the Imaging Biomarker Standardization Initiative (IBSI) [52]. The IBSI looks to provide standardized image biomarker nomenclature and definitions to ensure that the extraction of imaging biomarkers can be reproductible across different studies [52].

As explained, feature extraction can be applied with a 3DSlicer extension or via Python code. For this project, the extraction will be coded in Python as it is a faster method that returns a single file with all the

features. With the 3DSlicer extension, individual feature files for each segmentation would be obtained and they would have to later be merged, increasing time and resource consumption unnecessarily.

Features can be extracted from different areas apart from the tumour ROI. As reviewed, a study used radiomics extracted from the tumour and tumour borders [26], another also used features from the mesorectum, apart from the tumour [9], and another used features from the mesorectal fat instead [28]. In the present study, the ROI is exclusively drawn on the tumour thus the only extra features that could be extracted are tumour borders. These will be calculated from the original tumour segmentations without needing to delineate an additional mask. Tumour borders also contain some perirectal tissue, specifically from the mesorectum [3], thus the predictive capacity of the surrounding tissues can also be assessed with this method. Thus, borders will be included in the extraction because, without much additional workload, the predictive capability of tumour borders and surrounding tissues will be assessed.

To recap this section, the features will be extracted using the library PyRadiomics via a Python script and radiomics will be extracted from the segmented tumour mask and the calculated tumour borders.

3.5 Feature selection

As discussed in the state of the art section, radiomic based databases often have a large number of features and a small number of samples and the database in the present study is not an exception. Moreover, these types of datasets are prone to overfitting and may yield models that are not generalizable [15]. However, there are several methods to decrease dimensionality that will be discussed in this section. In general, the methods can be classified into two strategies: feature selection and feature extraction.

3.5.1 Feature selection

Feature selection methods are used to select a subset with the most relevant features from all the dataset [15]. Although some of these methods include training a machine learning model with different subsets of features to find de optimal subset (wrapper methods) or training a machine learning model with all the dataset to find what features the model chose as most important (embedded methods), these are highly computationally expensive and model dependent. Model dependency can yield a dataset that is not able to generalize well and might not perform effectively with different models [53]. There are other methods, independent of machine learning algorithms, known as filter methods.

Filter methods assess the relevance of features with respect to the target variable using statistical criteria such as correlation coefficients, mutual information, or variance. These methods are computationally efficient. However, unlike wrapper and embedded methods, they fail to capture interactions between features yielding less precise results [53].

Variance thresholding is used to remove features with low variance, that is features that only consist of noise and therefore have very little variation [54].

Correlation measures the positive and negative linear relationship between two variables. Features with high correlation to the target variable can be considered more relevant. Correlation between feature pairs can also be calculated and used to remove highly correlated features. However, this method has limitations in capturing non-linear relationships [53]. There are two types of correlation coefficients: Pearson's and Spearman's. On the one hand, Pearson's is used when both variables are normally distributed. Moreover, it is influenced by extreme values (outliers) which may exaggerate or dampen the

strength of relationship. On the other hand, Spearman's is appropriate when one or both variables are skewed and is robust to outliers [55].

Mutual information measures the amount of information that two random variables provide about each other [56] or the dependency between them. It captures both linear and non-linear relationships. Features with higher mutual information scores contribute more to reducing uncertainty about the target variable [53]. It is calculated from the joint probability distributions of the feature (independent variable) and the ground truth (dependent variable) [56].

Finally, statistical tests can also be carried out to find significant features (p-value < 0.05). There are several tests to be used depending on the type of data to be analysed. They can be classified into parametric and non-parametric tests. In parametric tests it is assumed that the data follows a specific distribution (normal distribution is commonly assumed). They are applied in well-known populations and are more powerful than their non-parametric counterparts. In non-parametric tests the population's distribution is not known [57]. In radiomics, the distribution of the data cannot be assumed as data is usually skewed, with outliers or the datasets are not balanced thus non-parametric tests are often used.

An example is the Kruskal-Wallis H test, which is the non-parametric version of the one-way analysis of variance or F-test. It assesses whether the distribution of a feature is significantly different across the data groups by ranking all data together [58].

3.5.2 Feature extraction

In feature extraction methods, the original feature space is transformed into a new smaller set of relevant features which are combinations of the original ones [15]. Principal Component Analysis (PCA) is an unsupervised learning technique that creates principal components (PC), which are linear combinations of the original features that maximally explain the variance of the data [59]. The technique projects the original dataset to a new space where the new features or orthogonal axes are the directions of maximum variance. The first PC has the highest variance, and the subsequent PCs have decreasing variances [60].

Linear Discriminant Analysis (LDA) is a supervised learning technique that focuses on finding a new feature space that maximizes class separability. It is considered to be more robust than PCA as it uses the data labels to compute the new feature space. In LDA, the number of produced components is smaller than the number of classes therefore, for binary classification problems, only one new feature will be created regardless of the features in the original dataset [60].

However, these methods may hinder explainability, which is especially important in healthcare. Since the new features are linear combinations of the original ones, it is hard to know exactly how much weight an original feature has in the decision of the later trained model [61]. Since one of the goals of this study is to identify potential biomarkers and image markers for response, these methods cannot be used as extracting which features are actually relevant and their respective importance would be unfeasible.

3.5.3 Proposed pipeline

To decrease the dataset dimensionality, filter methods were preferred because of the low computational cost, because they are model independent and because they preserve the original features. From the available statistical tests, non-parametric tests will be used due to the nature of radiomic data.

The proposed process is first to apply the Kruskal-Wallis test to find the radiomic features whose distributions differ significantly across tumour response groups, i.e., that they have a p-value below 0.05. Afterwards, to avoid selecting features that are highly correlated, the Spearman's correlation coefficient will be used as it is robust to outliers and skewed data [55]. In feature pairs with correlation over 0.8, which can be considered as strong correlation [62], the feature with the highest p-value, the least significant feature according to the Kruskal-Wallis test, will be removed [63].

3.6 Model training

This study consists in a binary classification problem. The two classes are non-responders to chemotherapy (0) and responders to chemotherapy (1).

3.6.1 Machine Learning models

First, some well-established Machine Learning models used in classification tasks will be considered. Most of these are used in the related studies and seem suitable for the current task. A brief description of the models is as follows:

- Decision Tree

Decision tree (DT) is a supervised learning method used for both classification and regression tasks [64]. They are named after their tree-like structure. Decision nodes represent decisions based on input features, branches represent the outcome of the decisions and leaf nodes represent the final predicted classification label [65]. For splitting decision nodes into branches, the most popular criteria are Gini impurity ("gini") and information gain ("entropy") [64]. One of the main advantages of using a decision tree is that it can generate classification rules that are easy to understand and explain [65].

- Random Forest

A Random Forest (RF) classifier is an ensemble classification technique that fits several decision trees in parallel on different dataset sub-samples and uses majority voting (classification tasks) or average (regression tasks) to calculate the outcome or final result. It minimizes the overfitting problem of the decision trees and increases the prediction accuracy, making it more accurate than a single decision tree model. It combines bootstrap aggregation (or bagging), i.e., random sampling of a dataset with replacement, and random feature selection to build the series of decision trees [64].

- Logistic Regression

Logistic Regression (LR) is a probabilistic statistical model commonly used to solve classification tasks. It typically uses a logistic function (sigmoid function) to estimate the probabilities of a given class. However, it can overfit high-dimensional datasets but the L1 (Lasso) and L2 (Ridge) regularization techniques can be used to mitigate it. The model assumes a linear relationship between the probabilities

of the dependent variable and independent variables thus it works well when the dataset can be separated linearly [64].

- Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm that constructs a hyperplane or set of hyperplanes in a high-dimensional space to separate the data into classes. The best hyperplane is the one that achieves a greater separation (margin), i.e., that has the greatest distance from the nearest training data points in any class. A larger margin generally leads to better generalization and lower classification error. SVM is effective in high-dimensional spaces and behaves differently based on different mathematical functions, known as kernels, such as linear, polynomial, radial basis function (RBF) or sigmoid. Kernel functions transform the input data into a higher dimensional space where linear separation might be possible. However, when the dataset contains noise or overlapping target classes, SVM does not perform well as the separation between classes cannot be found [64].

- Gradient Boosting, XGBoost and LightGBM

Gradient Boosting is an ensemble learning method that generates a final model by sequentially combining multiple individual weaker models, typically decision trees. Each new model is trained to minimize the loss function (mean squared error, cross-entropy…) of its predecessor using gradient descent [64]. In every iteration, the algorithm computes the gradient of the loss function with respect to the predictions and trains a new weak model to minimize it. The new predictions are then added to the rest (ensemble) and the process is repeated until a stopping criterion is met. This iterative approach allows the model to progressively correct its own mistakes, improving accuracy [66].

Extreme Gradient Boosting (XGBoost) is a form of gradient boosting that takes more detailed approximations into account when determining the best model as it computes second order gradients of the loss function, apart from the first order gradients already computed by the base model. It also applies L1 (Lasso) and L2 (Ridge) regularization, reducing overfitting and improving model generalization [64].

Light Gradient Boosting Machine (LightGBM) is another form of gradient boosting. It is a framework designed by Microsoft to be more efficient, faster and requires less memory usage. It also uses a tree level framework. It is faster as it supports parallel, distributed and GPU learning [67].

- Naïve Bayes

The Naive Bayes algorithm is based on the Bayes' theorem, with the assumption of conditional independence between every pair of features given the value of the class variable. The key benefit is that, compared to more sophisticated approaches, it needs a small amount of training data to estimate the necessary parameters. However, its performance may be affected due to its strong assumptions on feature independence. Some variants of this model are Gaussian Naïve Bayes, for continuous features, or Multinomial Naïve Bayes, for text data [64].

- Multilayer perceptron

Multilayer Perceptron (MLP), also known as the feed-forward artificial neural network, is the base architecture of deep learning. It consists of an input layer, one or more hidden layers and an output layer that are fully connected through weighted connections [64]. To optimize the model parameters and learn

the complex relationships in data, MLP uses the backpropagation technique via an algorithm such as gradient descent. The model computes the error in the prediction and adjusts the model using the computed gradients [68]. Furthermore, an activation function (ReLU, sigmoid, tanh…) can also be added at the neurons of each hidden layer to introduce nonlinearities to the model [64].

3.6.2 Evaluation Metrics

For a binary classification problem, the labels are usually interpreted as positive or negative results. In this case, responders would be considered as positive whereas a non-responder would be negatives. Thus, the results when comparing the label predicted by the model and the true ground-truth labels can be classified in four categories: true positives (TP) that are correctly predicted positive outcomes, true positives (TN) that are correctly predicted negative outcomes, false positives (FP) that are negative outcomes wrongly predicted as positive, and false negatives (FN) that are positive outcomes wrongly predicted as negative [69]. These results can be summarized in a 2x2 matrix called a confusion matrix, as seen in Figure 6. The most common classification metrics that reflect the model's performance can be calculated from this matrix [69]:

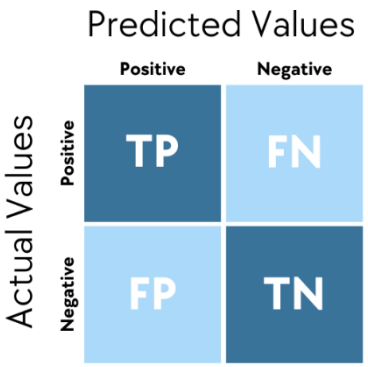

Figure 6. Schematic of a confusion matrix. [72]

- *Accuracy:* proportion of correctly classified instances in the set of all instances.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Equation 2. Accuracy equation.

- *Sensitivity (recall):* proportion of correct positive predictions out of all positive instances.

$$Sensitivity = \frac{TP}{TP + FN}$$

Equation 3. Sensitivity (or recall) equation.

- *Specificity:* proportion of correct negative predictions out of all negative instances.

$$Specificity = \frac{TN}{TN + FP}$$

Equation 4. Specificity equation.

- *Precision or Positive Predicted Value (PPV):* proportion of correct predictions out of all positive predicted instances.

$$Precision = \frac{TP}{TP + FP}$$

Equation 5. Precision equation.

- *Negative Predicted Value (NPV):* proportion of correct predictions out of all negative predicted instances.

$$NPV = \frac{TN}{TN + FN}$$

Equation 6. Negative Predicted Value equation.

- *F1-score:* harmonic mean of precision and recall.

$$F1\ score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Equation 7. F1-score equation.

Sensitivity and specificity, on one hand, and precision and NPV, in the other, are usually used as metric pairs. Moreover, while accuracy is the most common metric, sensitivity and specificity often give more information about the model especially if the dataset is very imbalanced [69].

Furthermore, as previously explained, some machine learning models classify samples into classes based on numeric probabilities and a threshold. Receiver operating characteristic (ROC) curves can be obtained by plotting sensitivity on the y-axis and 1-specificity (or false positive rate) on the x-axis at all possible threshold values. From the ROC curve, the area under the curve (AUC) can be computed and used as a measure of overall model performance and robustness across thresholds. Unlike the others, AUC does not depend on the probability threshold. It has a value between 0 and 1 but values over 0.5 indicate increasing discriminative ability of the model [70]. *Figure 7* shows an example of different ROC curves and their meanings.

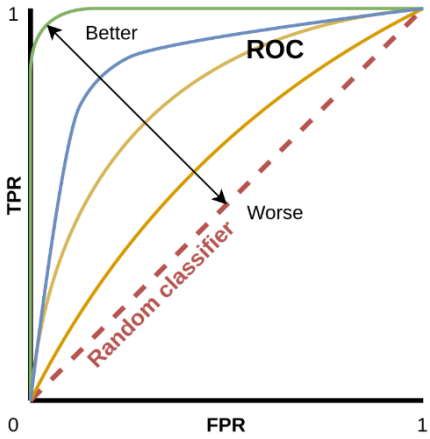

Figure 7. Schematic of a ROC curve and its meanings [73].

3.6.3 Model optimization

Machine Learning models can be tuned by changing the hyperparameters. Hyperparameters are user defined model characteristics that affect the model's functionality. For instance, hyperparameters are the number of decision trees used in a random forest classifier or the number of layers and neurons in each layer in a neural network. Hyperparameter tuning is the process of optimizing the model's hyperparameters to find the best performance and reduce overfitting [71].

The user defines a grid of hyperparameter to test, and the combinations are assessed using several methods. Grid search tests all possible combinations of hyperparameters. It is an exhaustive and time-consuming optimization method. In random search only some random hyperparameter combinations are evaluated. Bayesian optimization uses a probabilistic model to choose which combinations to test. First, it chooses a random subset of combinations. Based on the performance of the first models, it chooses the next combination by taking similar values to the one with best performance [71].

Regardless of the method, the different combinations are usually tested using k-fold Cross Validation in which training data is split into k-folds. Each parameter combination is evaluated by training it using k-1 folds and testing it using the remaining fold in a way that, after k evaluations, all data has been used once to test. After each evaluation the accuracy is computed. Afterwards, the mean accuracy between all k evaluations is assigned to the current hyperparameter combination. After all combinations are tested, the one with the best average accuracy is chosen as best [71]. Cross Validation can also be applied to the model to train and test it across folds without the need for hyperparameter tuning.

3.6.4 Proposed pipeline

Based on the literature reviewed and the methodology of the project, the pipeline to train, optimize and test the models is: train a set of robust models using 5-fold cross validation, carry out hyperparameter tuning via grid search 5-fold cross-validation to optimize the model, train the optimized models with the best hyperparameters using 5-fols cross-validation, and compute the performance metrics and feature importance analysis of the best models.

4. Detail engineering

In this section, the procedure and results of each step will be explained in detail.

4.1 Data acquisition and selection of cases

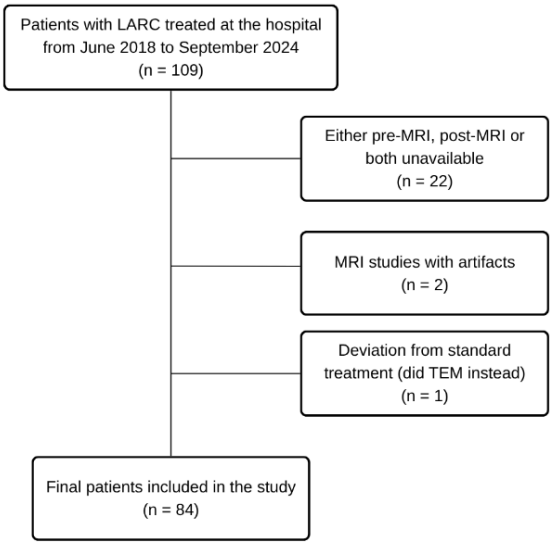

Figure 8. Patient selection diagram.

Before downloading the data, the patients enrolled in the study had to be chosen (*Figure 8*). Firstly, 110 patients diagnosed with Locally Advanced Rectal Cancer (LARC) treated in the hospital between June 2018 and September 2024 were selected. Given all the clinical record numbers of the patients, the MRI studies, called *RM RECTE/TERRA PELVIS,* were manually downloaded from the hospital's application *Enterprise Imaging Cloud* in DICOM format (Digital Imaging and Communications in Medicine), the standard medical imaging format for handling, storing, and transmitting information [74]. During this step, 22 patients were found to have unavailable MRIs, either pre-nCRT, post-nCRT or both, and were removed. Upon revision of the studies, 2 patients had imaging artifacts that impeded the visualization and segmentation of the tumour and were also discarded. After looking into the available data of the patients, one more patient was discarded as they underwent Transanal Endoscopic Microsurgery (TEM) instead. 84 patients remained.

All the studies were downloaded using the anonymization option in the *Enterprise Imaging Cloud* to ensure that personal data was removed. Afterwards, the chosen clinical features were acquired using both the *Enterprise Imaging Cloud* and the *SAP* from the hospital. The TRGs were also downloaded and later classified into responders (TRG1 and TRG2) and non-responders (TRG3, TRG4 and TRG5) [6] for the models' ground truth. *Table 1* shows a description of the clinical features acquired.

Responders (label = 1) (n=36) (42.86 %)					
Categorical Features			***Continuous Features***		
Feature name	Count	%	Feature name	Mean ± std	Range
Sex			Age at diagnosis	69.44 ± 10.69	[49, 92]
Female	18	50 %			
Male	18	50 %			
T			Tumour extension	45.06 ± 12.23	[26, 67]
3a	10	27.8 %			
3b	19	52.7 %			
3c	4	11.1 %			
3d	3	8.3 %			
4a	0	0 %			
4b	0	0 %			
N			Sphincter distance	44.17 ± 38.75	[0, 144]
0	6	16.7 %			
1a	5	13.9 %			
1b	11	30.6 %			
1c	0	0 %			
2a	7	19.4 %			

2b	7	19.4 %			
EMVI			Anal distance	78.72 ± 38.27	[20, 172]
Absent	26	72.2 %			
Present	10	27.8 %			
FMR					
Absent	23	63.9 %			
Present	13	36.1 %			
Localization					
Low	10	27.8 %			
Medium	16	44.4 %			
High	10	27.8 %			

Non-Responders (label = 0) (n=48) (57.14 %)					
Categorical Features			***Continuous Features***		
Feature name	Count	%	Feature name	Mean + std	Range
Sex			Age at diagnosis	69.83 ± 13.41	[40, 90]
Female	16	33.3 %			
Male	32	66.67 %			
T			Tumour extension	52.29 ± 15.72	[25, 88]
3a	10	20.8 %			
3b	14	29.2 %			
3c	15	31.3 %			
3d	2	4.1 %			
4a	3	6.3 %			
4b	4	8.3 %			
N			Sphincter distance	35.04 ± 34.06	[0, 125]
0	4	8.3 %			
1a	9	18.8 %			
1b	16	33.3 %			
1c	3	6.3 %			
2a	12	25 %			
2b	4	8.3 %			
EMVI			Anal distance	68.67 ± 35.15	[15, 145]
Absent	27	56.3 %			
Present	21	43.8 %			
FMR					
Absent	24	50 %			
Present	24	50 %			
Localization					
Low	21	43.8 %			
Medium	17	35.4 %			
High	10	20.8 %			

Table 1. Description of clinical features. The percentages are with respect to the responders and non-responders subset.

The correlation matrix representing the correlation of the clinical features between them and with the response can be found in *Figure 9.*

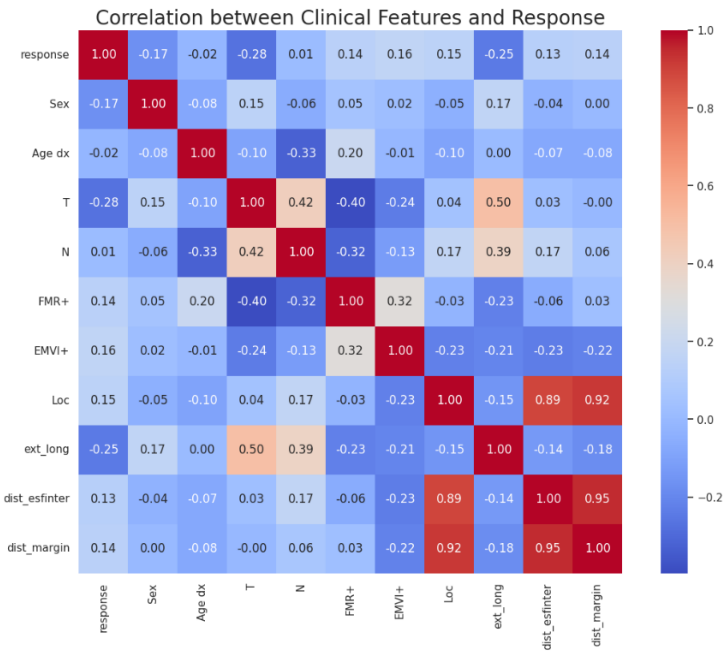

Figure 9. Correlation matrix between the extracted clinical features and response.

After, all anonymized MRI studies were imported into Slicer3D and the short-axis T2w sequences were selected for later segmentation. During this process, it was found that depending on the localization of the tumour in the rectum; low, middle or high, the short-axis and the long-axis could be confused. In tumours located in the low rectum, the short-axis resembles an axial plane sequence and the long-axis a coronal plane sequence, as explained in the background section. However, if the tumour is in the high rectum, the short-axis resembles a coronal plane and the long-axis an axial plane, as seen in *Figure 10.* This is because of the anatomy of the rectum, and it is not a mistake in sequence acquisition as is. Despite this confusion, the selected sequence was always the T2w short-axis.

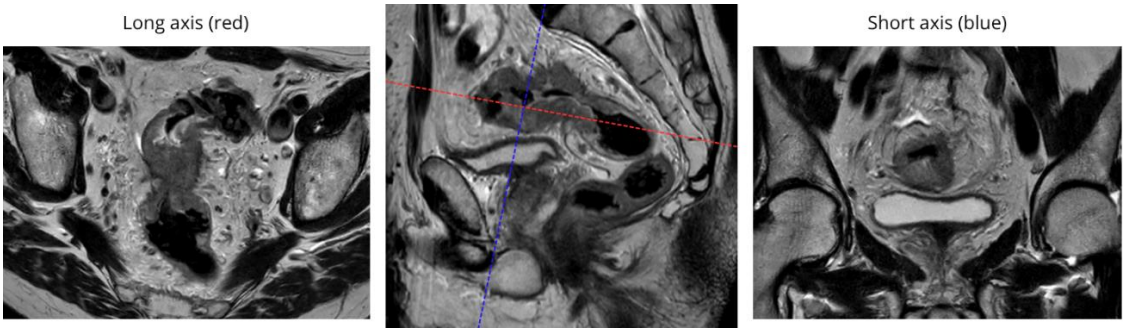

Figure 10. Short and Long-axis of a rectal tumour located in the high rectum. Left: Long-axis of the tumour. Middle: slice in the sagittal plane of the tumour with the tumour axes. Right: short-axis of the tumour

4.2 Tumour segmentation

As discussed in the concept engineering section, segmentations were carried out in 3DSlicer (version 5.6.2) and in the T2w short-axis sequence. The *Segmentations* and *Segment Editor* modules of 3DSlicer were used to delineate the regions of interest (ROI). Specifically, the tools used from *Segment Editor* were Paint, a circular brush of custom radius, Draw, to draw the tumour contour that the program automatically fills, and Erase, a circular brush of custom radius to erase parts of the segmentation.

Semi-automatic methods such as Grow from Seeds, Fill Between Slices or Level Tracing are also available in the program. In Grow from Seeds the user manually paints a small ROI (seed) and the program dilates it to achieve the complete segmentation. In Fill Between Slices the user segments the structure in some slices and skips others. The program will fill the skipped slices by interpolating between segmentations. Level Tracing is used to find areas where the pixels all have the same intensity. By clicking on these areas, they are automatically added to the segmentation.

However, these methods were not useful as in T2w the colorectal tumours have very heterogeneous intensities and similar tones to surrounding tissues. The program would confuse the different regions of the image and the segmentations had to be corrected manually either way. These methods would be useful to segment structures with homogeneous tone and high contrast in comparison with their surroundings, like bone. Thus, the segmentations were carried out using the manual methods.

The main challenge during the segmentations was understanding rectum anatomy and rectal cancer morphology. Segmentation was restricted solely on known tumour regions, excluding regions of uncertainty to avoid adding errors. Moreover, in the cases where the tumour would infiltrate nearby organs, only the part of the tumour that was inside the rectum was included in the segmentation. Another consideration was what are known as “partial volumes”. The MRI slices cover the signal included in a depth of 3 mm, but this is not necessarily a uniform area in the real body thus, especially in the areas where tumour is starting or ending, the tumour appears with a rough texture with a signal intensity halfway between the tumour and healthy tissue. Partial volumes also take place when there are liquid pools inside the tumour, common if the tumours are mucinous, or if there is liquid in the rectal lumen. The partial volumes of liquid pools would have a light grey tone as liquid appears white in the MRI [3]. Partial volumes were not included in the segmentations as it would be including healthy tissue into the tumour ROI, which may alter the results. Segmentation examples can be found in *Figure 11.* Each segmentation was validated and corrected by a senior radiologist from the radiology department of the hospital. When finished, the segmentation masks were saved in *.nrrd* format.

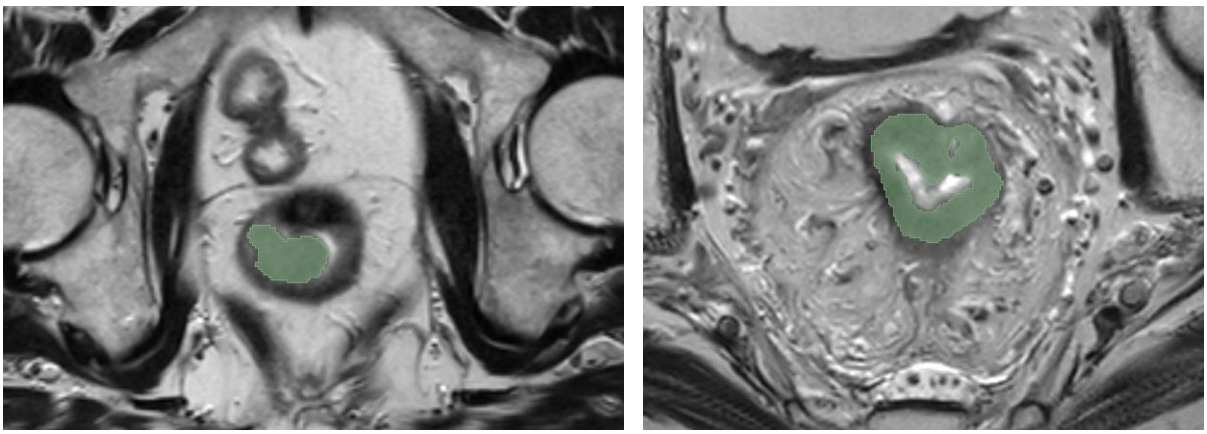

Figure 11. Example segmentations in 3DSlicer. The green area is the segmented tumour. Left: segmentation of a clearly defined tumour. Right: segmentation with liquid infiltrations (white), tumour infiltrating nearby tissues and partial volumes (bottom part of the segmentation).

4.3 Image normalization

Prior to the normalization step, all studies were converted to *.nii.gz* format using a Python script, to avoid errors in processing. Normalization has been carried out on each study before extracting the radiomics features. The images were imported, converted to a Numpy array and then they were normalized by

applying the *.fit_transform()* method from *StandardScaler()* [51]. Finally, the arrays were converted back to an image. The conversions and image processing were achieved using the Python's library SimpleITK, used to carry out multi-dimensional image analysis in multiple coding languages [75].

4.4 Radiomic data extraction

The code to extract the radiomics features was implemented with Python and its library PyRadiomics. All code to extract radiomic features and normalize the images can be found in *Annex 2*. The code iterated over all the MRI scans, saved in a specific folder, checked that their respective segmentation mask was available, which were saved in a separate folder, and extracted the features on the mask region using the previously initialized *RadiomicsFeatureExtractor(),* which is the PyRadiomics class used to store the radiomics extraction parameters [76]. The extractor was configured so that all available radiomics features were extracted, adding up to a total of 110 [17]. The features were saved to a .csv file.

To extract radiomics from tumour borders, these had to be calculated first. Firstly, a 2 mm dilation and a 2 mm erosion were applied to the original segmentation mask using the SimpleITK functions *.BinaryDilate()* and *.BinaryErode()*, respectively. The physical size in millimetres was converted to the number of voxels using the voxel spacing of each MRI, obtained with the *.GetSpacing()* method. Then, the physical size was divided by the spacing in each dimension (x, y and z) to obtain the conversion in number of voxels. Finally, the eroded mask was subtracted from the dilated mask to obtain the tumour masks. With this method, the tumour border mask includes the outermost part of the tumour and the closest surrounding tissues. The masks will usually include part of the rectal mucosa or of the perirectal tissues. An example of the process can be found in *Figure 12.* The last step after both datasets were acquired was to manually add the ground truth column, which contains the target variable for feature selection and supervised learning. The distributions of the radiomic features were plotted. Some did not follow a normal distribution and were skewed (*Figure 13*).

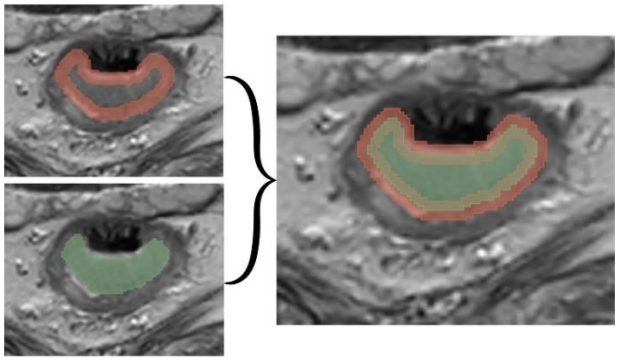

Figure 12. Example of the calculated tumour border mask (top left), the original manual segmentation (bottom left) and an overlay of both (right).

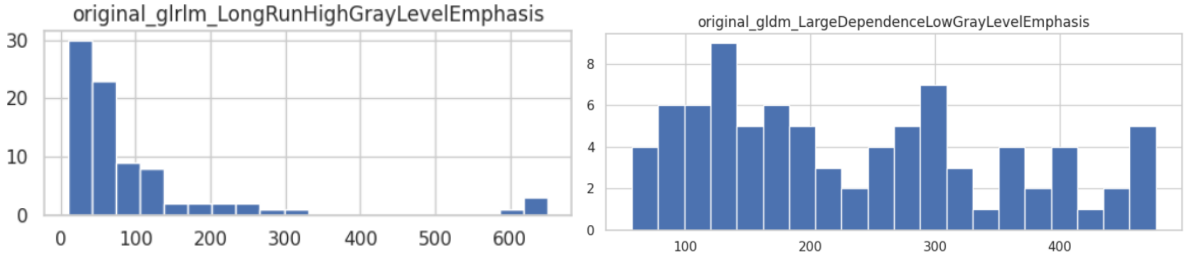

Figure 13. Example distribution plots of the extracted radiomic variables.

4.5 Feature selection

After building the radiomics datasets, the most significant radiomic features have to be selected to decrease dimensionality. This process will be carried out with the pipeline proposed in the concept engineering section both for the tumour core dataset and the tumour borders dataset. First, the Kruskal-Wallis test is applied using the *kruskal()* function from the *scipy.stats* Python library [78]. This test provides a p-value for each feature that signifies whether the distribution of a feature is significantly different across

the data groups (separated by tumour response). The features that have p-values below 0.05 are considered significant and are kept. Afterwards, to remove collinearity, the Spearman correlation matrix is computed and, in feature pairs with an absolute correlation coefficient greater than 0.8, the least significant feature is removed. This is carried out using the *spearman()* function from the *scipy.stats* Python library [78]. The final features are saved in another .csv file. The code can be found in *Annex 2*.

As a result, 17 and 10 radiomic features were selected for the tumour core and tumour borders datasets, respectively. After removing collinearity only 8 and 4 features remained for the tumour core and tumour borders datasets, respectively. *Table 2* shows the selected features after removing collinearity for both datasets and their p-value and mean. After this step, three datasets were available: one with radiomics from the tumour core, one with radiomics from the tumour border and one with clinical features.

Tumour Core Dataset			
Feature Name	**Mean ± std (responders)**	**Mean ± std (non-responders)**	**p-value**
Major Axis Length (shape)	40.06 ± 12.52	46.08 ± 15.76	0.038
Maximum 2D Diameter Slice (shape)	34.84 ± 13.90	40.39 ± 14.79	0.030
Surface Volume Ratio (shape)	0.53 ± 0.13	0.48 ± 0.15	0.049
Total Energy (firstorder)	1476.88 ± 1748.62	4264.07 ± 6727.58	0.014
Dependence Non-Uniformity Normalized (gldm)	0.07 ± 0.03	0.09 ± 0.05	0.035
Large Dependence High Gray Level Emphasis (gldm)	624.00 ± 336.58	886.64 ± 528.61	0.025
Long Run High Gray Level Emphasis (glrlm)	61.71 ± 44.70	135.47 ± 165.89	0.021
Short Run Low Gray Level Emphasis (glrlm)	0.33 ± 0.05	0.29 ± 0.09	0.009
Tumour Borders Dataset			
Feature Name	**Mean ± std (responders)**	**Mean ± std (non-responders)**	**p-value**
Maximum 2D Diameter Slice (shape)	39.48 ± 14.09	44.88 ± 14.93	0.045
Minimum (firstorder)	-1.30 ± 0.31	-1.17 ± 0.32	0.025
Dependence Variance (gldm)	37.52 ± 4.50	35.30 ± 4.64	0.034
Large Dependence Low Gray Level Emphasis (gldm)	266.66 ± 120.74	211.01 ± 114.68	0.036

Table 2. Selected significant radiomic features after removing collinearity for both datasets.

The selected features contain information of the tumour shape, morphology and texture of the region they represent. A brief description of each feature, based on PyRadiomics documentations, can be found below in order to give meaning to the datasets at hand. The features are classified according to the feature class. Firstly, the shape features which are independent of the grey level intensity distribution and are extracted from the segmentation mask (and the mesh calculated from it) are the following [17]:

- *Major Axis Length (shape):* It is the measure of the largest axis length of the ellipsoid that encloses the segmented ROI. It indicates the tumour orientation and elongation [17].
- *Maximum 2D Diameter Slice (shape):* The largest Euclidean distance between tumour vertices in the row-column plane (axial plane). Used to estimate maximum spread [17].
- *Surface Volume Ratio (shape):* Defined as the quotient between surface area and volume (A/V). A lower value indicates a more compact, sphere-like shape whereas higher values suggest irregular shapes [17].

Next, the features in the first order class are the first order statistics that describe the distribution of voxel intensities within the ROI defined by the mask. These are common statistical metrics such as maximum, minimum, percentiles, energy, variances, etc [17]:

- *Total Energy (firstorder):* Energy is the sum of the squares of the voxel values in an image. A larger value means that the image has high intensity voxels overall. Total Energy is the value of Energy feature scaled by the volume of the voxel (in mm^3). This feature is volume-confounded: the higher the volume of the ROI, the higher the energy [17].
- *Minimum (firstorder):* The minimum intensity value of a voxel in the ROI [17].

Gray Level Dependence Matrix (GLDM) features quantify grey level dependencies in an image. Grey level dependencies are defined as the number of connected voxels within distance δ that are dependent on the centre voxel. These features describe the texture homogeneity of a region and the local patterns present in it by analysing how similar the intensities of nearby voxels are [17]:

- *Dependence Non-Uniformity Normalized (gldm):* Measures the similarity of dependencies throughout the image, with a lower value indicating more homogeneity among dependencies in the image, i.e., a more uniform texture [17].
- *Large Dependence High Gray Level Emphasis (gldm):* Measures the joint distribution of large dependencies with higher grey level values, highlighting how common large homogeneous regions with high voxel intensity are in the ROI [17].
- *Large Dependence Low Gray Level Emphasis (gldm):* Measures the joint distribution of large dependencies with lower grey level values, highlighting how common large homogeneous regions with low voxel intensity are in the ROI [17].
- *Dependence Variance (gldm):* Measures the variance in dependence size in the image. A high value suggests heterogeneous texture [17].

Finally, Gray Level Run Length Matrix (GLRLM) features quantify grey level runs, which are defined as the length, in number of pixels, of consecutive, colinear pixels that have the same grey level value in a certain direction [17]:

- *Long Run High Gray Level Emphasis (glrlm):* Measures the joint distribution of long run lengths with higher grey levels. High values suggest uniform texture with large streaks of high grey level intensities [17].
- *Short Run Low Gray Level Emphasis (glrlm):* Measures the joint distribution of shorter run lengths with lower grey levels. High values suggest heterogeneous texture with short streaks of voxels with low grey level intensities, such as grainy or speckle-like regions [17].

Gray Level Co-occurrence Matrix (GLCM), Gray Level Size Zone Matrix (GLSZM), and Neighbouring Gray Tone Difference Matrix (NGTDM) feature classes were also extracted by PyRadiomics [17] but none were selected in the statistical tests.

The correlation matrixes of both feature sets can be found in *Figure 14* and *Figure 15,* which show how, after feature selection using statistical tests and the correlation coefficients, features that are highly correlated with each other, and thus redundant, are removed.

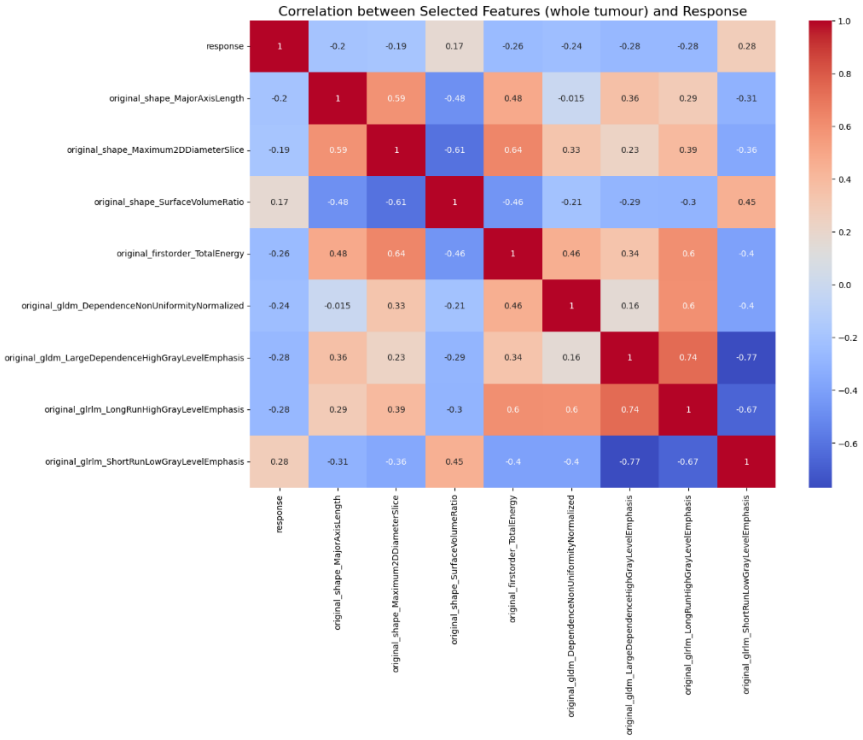

Figure 14. Correlation matrix of the selected core tumour features and response.

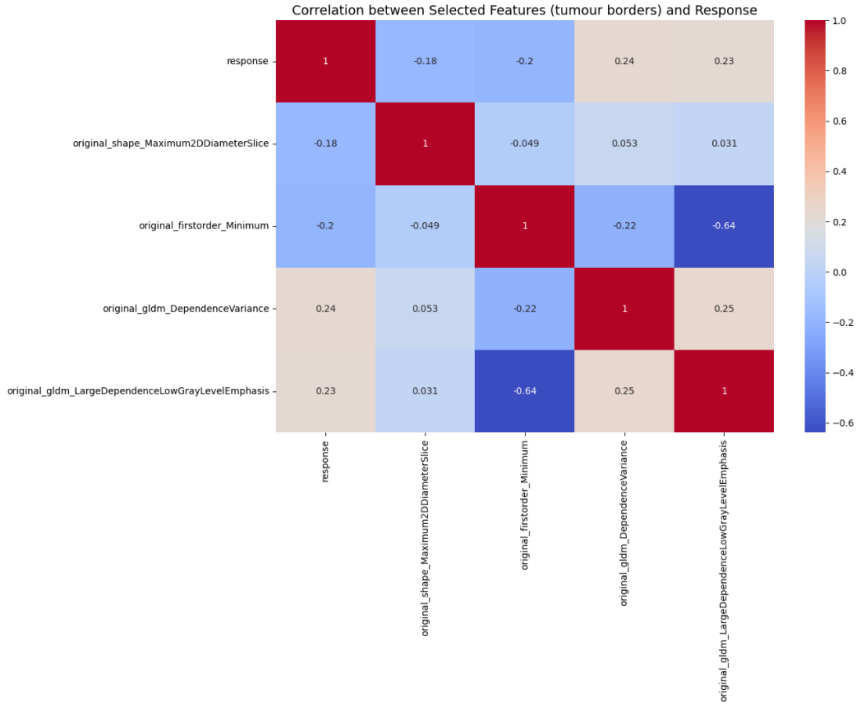

Figure 15. Correlation matrix of the selected tumour border features and response.

4.6 Model training

Nine machine learning models were trained, optimized and tested to assess the predictive capability of each of the seven dataset combinations available:

- One dataset contains only radiomic features from the tumour core (which will also be called whole tumour during this work), capturing exclusively the intratumoral characteristics.
- One dataset contains exclusively tumour border radiomic characteristics, focusing on nearby tissues and tumour-patient interface information, to assess the predictive capabilities of the surrounding tissues.
- A third dataset is composed only of the clinical features acquired from the hospital's database, including demographic information and features extracted during tumour staging, to assess non-imaging predictors.
- The other four datasets contain all combinations of the previous features: whole tumour + tumour borders, whole tumour + clinical features, tumour borders + clinical features, and a final dataset containing all data (whole tumour + tumour borders + clinical features).

This approach was chosen to systematically evaluate all the individual and combined feature sets and their contribution to predicting response to neoadjuvant treatment. Each dataset had 84 samples. 15 % of these samples, 13 samples, were reserved as a final test set to assess the metrics of the best models and avoid data leakage in model assessment. Specifically, out of 13 samples in the test set, 7 were non-responders and 6 were responders. This data split was carried out using the *train_test_split()* function from the Python package scikit-learn. The hyperparameter stratify was used to have the same percentage of samples from each class in the validation set [79]. The remaining set was not split into train and validation as the models were trained and optimized using a 5-fold Cross Validation strategy.

First and foremost, the models were trained using the default hyperparameters with a 5-fold Cross Validation strategy. Since almost all showed possible overfitting to training data and mediocre performance in test and validation sets, the models were optimized using hyperparameter tuning to try and mitigate overfitting, improve generalization and overall performance.

To optimize the models, a grid of hyperparameters was defined and each combination was tested using a gird search 5-fold Cross Validation strategy. This was implemented with the *StratifiedKFold()* and *GridSearchCV()* functions from *scikit-learn.* The stratified K-fold function was implemented to have balanced classes in each of the train and validation folds [80]. The hyperparameters that were optimized for each model and the functions to initialize them can be found in *Table 3*. All functions are from *scikit-learn except for XGBClassifier()* and *LGBMClassifier()* that are from the xgboost and lightgbm modules, respectively. A short list of hyperparameters was chosen to avoid extremely high computational times.

Model	Hyperparameters tuned
Decision Tree *DecisionTreeClassifier()*	Maximum Depth Minimum Samples Split Minimum Samples Leaf Model Criterion
Random Forest *RandomForestClassifier()*	Number of Estimators Maximum Depth Minimum Samples Split

Support Vector Machine *SVC()*	Model Kernel C Gamma
Logistic Regression *LogisticRegression()*	C Penalty Solver
Naïve Bayes *GaussianNB()*	No hyperparameters to tune
Multilayer Perceptron *MLPClassifier()*	Hidden Layer sizes Activation Function Alpha Learning Rate
Gradient Boosting *GradientBoostingClassifier()*	No hyperparameters to tune
XGBoost *XGBClassifier()*	Number of Estimators Maximum Depth Learning Rate Subsample Colsample Bytree
LightGBM *LGBMClassifier()*	Number of Estimators Number of Leaves Learning Rate Boosting Type Minimum child samples

Table 3. List of hyperparameters tuned for each model.

Radiomic features and continuous clinical features were scaled for training and test using *MinMaxScaler()* and *StandardScaler()* [51], respectively, that are also functions from *scikit-learn*. *MinMaxScaler()* transforms data so that all values are within a given range, in this case between [0 1]. This does not change de distribution of the data nor the effect of the outliers, conserving information, especially important in radiomics data [81]. *StandardScaler()* performs z-score normalization, as previously discussed, scaling data to unit variance and zero mean [51]. The scaling process was applied inside each training fold separate from the test set to avoid data leakage.

The best hyperparameters for each model were the ones that yielded the best average test accuracy. After the hyperparameter optimization, the models were retrained with the best configurations using 5-fold Stratified Cross Validation. During Cross Validation (CV), mean and standard deviations of the training and test accuracies were also tracked. Later, the models were assessed using the previously reserved test set to calculate the confusion matrix and the evaluation metrics: accuracy, sensitivity, specificity, precision, NPV and f1-score. Afterwards, the model with the best performance overall, taking into account all of the metrics, was selected for each dataset and the area under the ROC curve and the feature importance were calculated. The code for hyperparameter optimization and model training can be found in *Annex 3.*

4.7 Results

This section contains the results of the best models for each dataset. The results of all models, used to assess which model performed best, are in the *Annex 4*. Note that label 1 (responders) is considered as the positive class whereas label 0 (non-responders) is considered as the negative class for the calculation of performance metrics.

- *Whole tumour radiomics only dataset:*

For this dataset, the model with the best performance is Random Forest as it the best model overall. Gradient Boosting, also based on decision trees, has de same validation accuracy, sensitivity, and specificity but by taking a look at the CV metrics, it can be seen that the model probably overfits the training data and is not able to generalize well to the test data. It fits perfectly to the training data in all folds as it has an accuracy of 1.00 ± 0.00. However, the performance decreases with the test folds with an average accuracy of 0.56 ± 0.11, which means that the model is barely capable of making predictions.

Random Forest, on the other hand, shows a training accuracy that is quite high with little variance through all folds (0.91 ± 0.02) and an average test accuracy of 0.71 ± 0.11 which is very similar to the validation accuracy of 0.77. This shows that, despite some slight overfitting, the model is capable to generalize to new data and predict well. The other metrics are summarized below.

Best Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	AUC	Sensitivity	Specificity	Precision	NPV	F1-score
Random Forest	0.91 ± 0.02	0.71 ± 0.11	0.77	0.70	0.67	0.86	0.80	0.75	0.73

Table 4. Metrics for the best model (Random Forest) for the whole tumour radiomics only dataset.

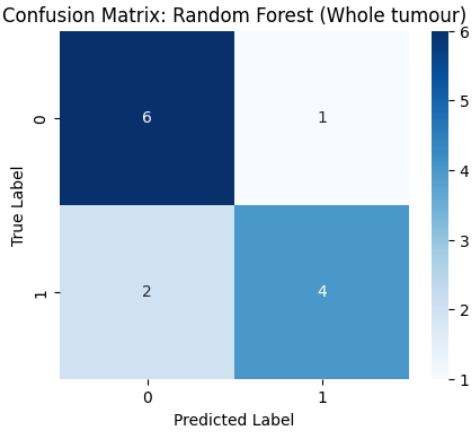

Figure 16. Confusion Matrix of the best model (Random Forest) for the whole tumour dataset.

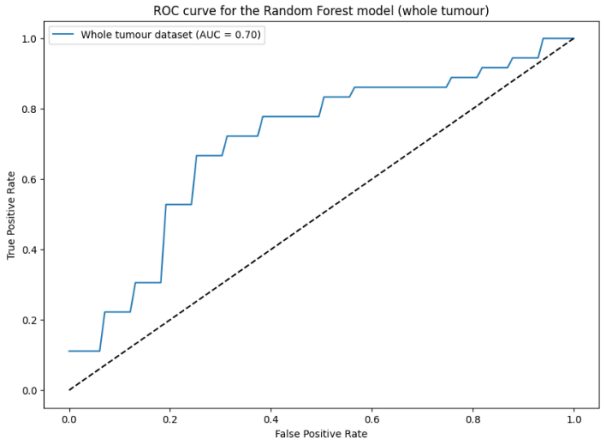

Figure 17. ROC curve and AUC of the best model (Random Forest) for the whole tumour dataset.

The confusion matrix shows that only three cases had incorrect predictions and that the model is more likely to predict that a new sample is a non-responder. An AUC of 0.70 indicates that the model has overall robust performance.

- *Tumour borders radiomics only dataset:*

In this case, the best validation accuracy is 0.62, achieved by four models that also have the same sensitivity and specificity values. Multilayer Perceptron, Naïve Bayes and Logistic Regression show underfitting as they are not able to reach average CV train accuracies over 0.70. On the other hand, XGBoost seemingly shows overfitting as the average train accuracy is 0.99 and the test and validation accuracies are much lower. All four models have similar average test accuracies, with values ranging from 0.60 to 0.66. Therefore, the choice for the best model comes down to whether underfitting or

overfitting is preferred. XGBoost was chosen because it shows capability to learn the patterns in the data, despite having low ability to generalize to new data. The metrics of this model are summarized below.

Best Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	AUC	Sensitivity	Specificity	Precision	NPV	F1-score
XGBoost	0.99 ± 0.01	0.66 ± 0.08	0.62	0.54	0.50	0.71	0.60	0.62	0.55

Table 5. Metrics for the best model (XGBoost) for the tumour border radiomics only dataset.

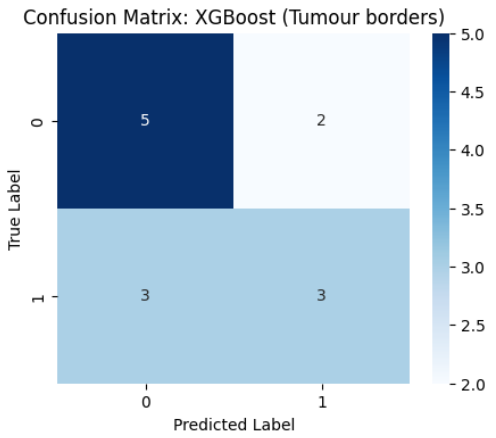

Figure 18. Confusion Matrix of the best model (XGBoost) for the tumour borders dataset.

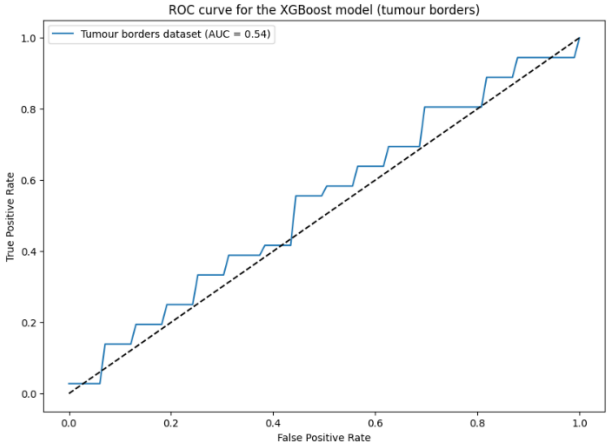

Figure 19. ROC curve and AUC of the best model (XGBoost) for the tumour borders dataset.

Figure 18 shows that 5 cases were predicted wrongly and that the model has a tendency to assign new samples as non-responders. An AUC of 0.54 and the ROC curve shape of *Figure 19* show that the model has poor predictive capability overall.

- *Tumour core + borders radiomics dataset:*

The highest validation accuracy for this dataset is 0.69 and has been achieved by four models. Naïve Bayes is discarded as, with a sensitivity of 1.00 and a specificity of 0.43, the model is biased towards predicting new samples as responders.

The other three models have identical validation metrics. XGBoost is discarded because of apparent overfitting, with an average CV training accuracy of 1.00 and an average CV test accuracy of 0.55. The chosen model is Multilayer Perceptron because it has less gap between training and test accuracies than Decision Tree, indicating better generalization. The latter possibly overfits to the training data and has smaller test accuracy. However, it is worth to point out that the test accuracies for both models are very variable between folds (standard deviations of 0.14 and 0.15 for the Decision Tree and Multilayer Perceptron, respectively) indicating that results are very different across folds.

Best Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	AUC	Sensitivity	Specificity	Precision	NPV	F1-score
Multilayer Perceptron	0.71 ± 0.04	0.62 ± 0.15	0.69	0.67	0.67	0.71	0.67	0.71	0.67

Table 6. Metrics for the best model (Multilayer Perceptron) for the whole tumour + border radiomics dataset.

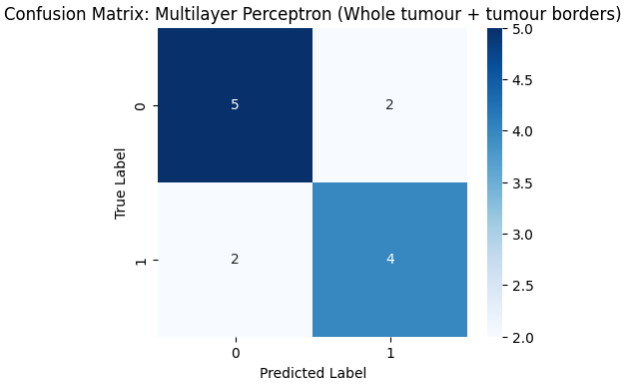

Figure 20. Confusion Matrix of the best model (Multilayer Perceptron) for the core tumour + tumour borders dataset.

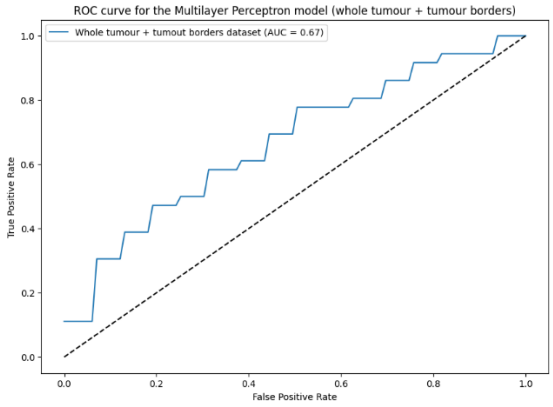

Figure 21. ROC curve and AUC of the best model (Multilayer Perceptron) for the core tumour + tumour borders dataset.

This model presents a performance halfway between the core tumour only and the tumour borders only: four cases are wrongly classified, and the AUC has a value of 0.67. It has worse results than the core tumour only model likely because of the addition of the border features that perform worse on their own and add noise to the data which negatively affects the predictions.

- *Clinical features only dataset:*

There is only one model that has the best accuracy, of 0.69, in this dataset for the validation set, which is Decision Tree. It has an average CV training accuracy of 0.77 and test accuracy of 0.57 showing that it can still generalize to new data, despite the low accuracy values that seemingly indicate some underfitting. The other models have overall poor validation performance, so they are discarded.

Best Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	AUC	Sensitivity	Specificity	Precision	NPV	F1-score
Decision Tree	0.77 ± 0.02	0.57 ± 0.09	0.69	0.59	0.83	0.57	0.62	0.80	0.71

Table 7. Metrics for the best model (XGBoost) for the clinical features only dataset.

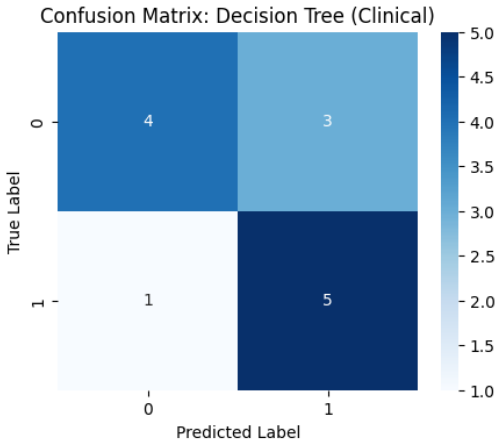

Figure 22. Confusion Matrix of the best model (Decision Tree) for the clinical features only dataset.

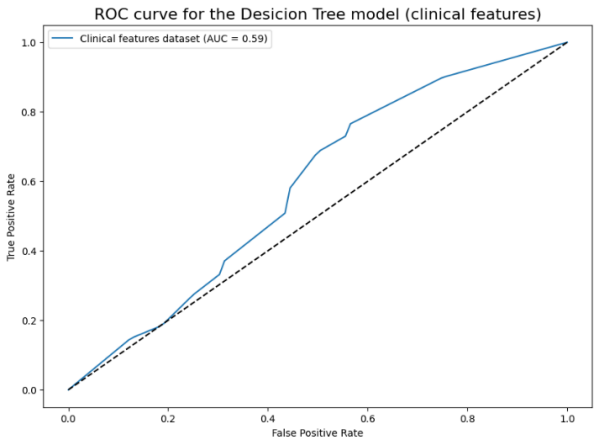

Figure 23. ROC curve and AUC of the best model (Decision Tree) for the clinical features only dataset.

This model tends to predict new samples as responders as it can clearly be seen in the confusion matrix, 8 predicted as responders vs 5 predicted as non-responders, and in the ROC curve, showing that for high sensitivities, the model has lower specificity.

- *Clinical features + tumour radiomics dataset:*

Three models have the maximum accuracy for this dataset, of 0.69. Naïve Bayes is discarded due to the average training and test accuracies of 0.62 and 0.63 that indicate that the model cannot learn the data well and underfits. From the other two, Random Forest is chosen over Multilayer Perceptron because of higher average CV test accuracy, despite the fact that Random Forest likely shows overfitting.

Best Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	AUC	Sensitivity	Specificity	Precision	NPV	F1-score
Random Forest	0.99 ± 0.01	0.73 ± 0.07	0.69	0.75	0.50	0.86	0.75	0.67	0.60

Table 8. Metrics for the best model (Multilayer Perceptron) for the clinical features + whole tumour radiomics dataset.

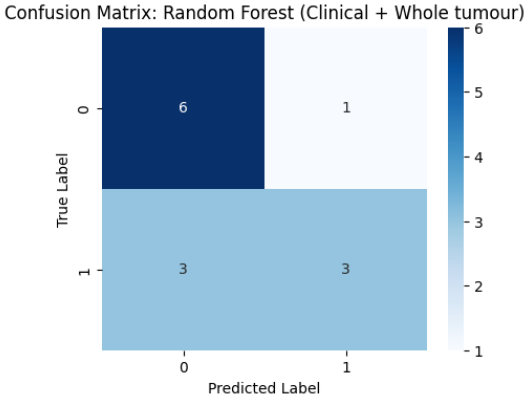

Figure 24. Confusion Matrix of the best model (Random Forest) for the clinical features + whole tumour radiomics dataset.

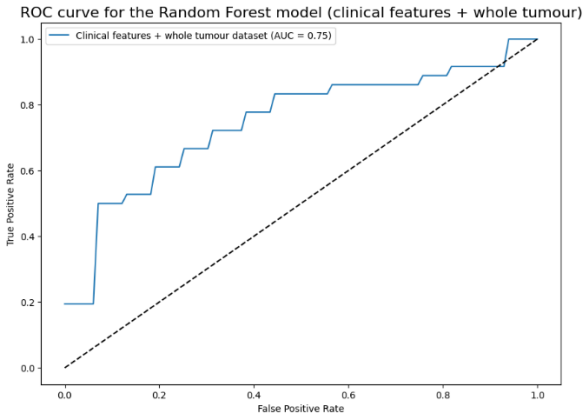

Figure 25. ROC curve and AUC of the best model (Random Forest) for the clinical features + whole tumour radiomics dataset.

The confusion matrix shows that the model tends to classify new samples as non-responders. The ROC curve also shows that the model tends to have higher specificity confirming the fact that it predicts more cases as non-responders. Surprisingly, the AUC is very high. This probably because the class classification threshold, the probability cut that the model uses to assign a label to each class, might not be correct. Since the ROC is calculated across all thresholds, the AUC is high.

- *Clinical features + border radiomics dataset:*

In this case, three models achieve an accuracy of 0.69, the highest for the dataset. However, Multilayer Perceptron is chosen because it is the only model with balanced sensitivity and specificity. In the other models, Decision Tree and Random Forest, these metrics indicate a bias towards classifying new samples as responders and non-responders, respectively.

Best Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	AUC	Sensitivity	Specificity	Precision	NPV	F1-score
Multilayer Perceptron	0.76 ± 0.03	0.65 ± 0.07	0.69	0.67	0.67	0.71	0.67	0.71	0.67

Table 9. Metrics for the best model (Multilayer Perceptron) for the clinical features + tumour border radiomics dataset.

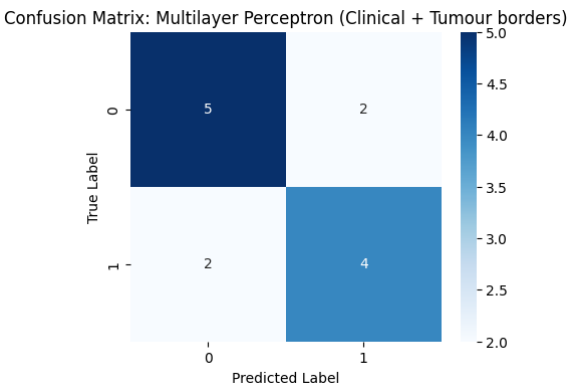

Figure 26. Confusion Matrix of the best model (Multilayer Perceptron) for the clinical features + tumour border radiomics dataset.

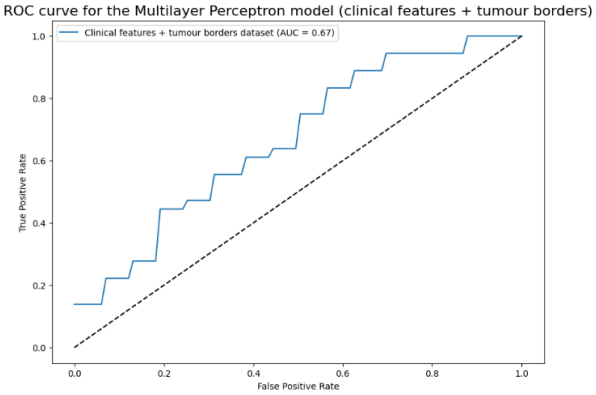

Figure 27. ROC curve and AUC of the best model (Multilayer Perceptron) for the clinical features + tumour border radiomics dataset.

As explained, the model does not show a clear tendency to predict one of the two classes, confirmed by the confusion matrix and the ROC curve. Four validation samples are misclassified.

- *Clinical features + whole tumour radiomics + tumour border radiomics dataset:*

Finally, for the dataset that includes all features, the highest accuracy is 0.77, achieved only by Multilayer Perceptron. However, Multilayer Perceptron probably presents overfitting due to the high gap between CV train and test average accuracies. Despite this, it is the model that generalizes better to new data and has better metrics overall.

Best Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	AUC	Sensitivity	Specificity	Precision	NPV	F1-score
Multilayer Perceptron	0.92 ± 0.01	0.63 ± 0.10	0.77	0.70	0.83	0.71	0.71	0.83	0.77

Table 10. Metrics for the best model (Multilayer Perceptron) for the clinical features + whole tumour + tumour border radiomics dataset.

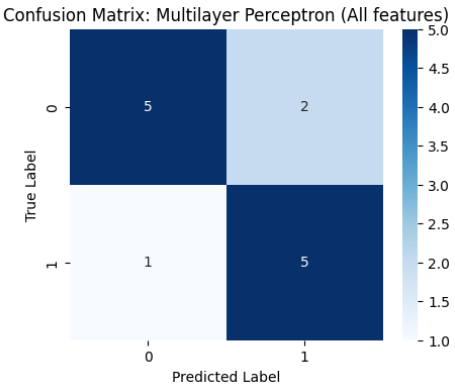

Figure 28. Confusion Matrix of the best model (Multilayer Perceptron) for the clinical features + whole tumour + tumour border radiomics dataset.

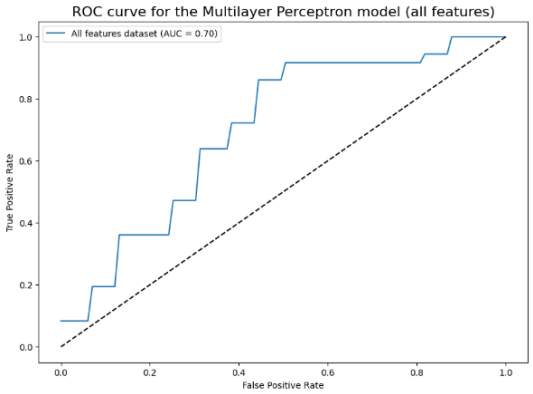

Figure 29. ROC curve and AUC of the best model (Multilayer Perceptron) for the clinical features + whole tumour + tumour border radiomics dataset.

This model only predicts three cases as incorrect and has a slight bias towards predicting the positive class, as seen in the ROC curve. This cannot be seen clearly using the confusion matrix as the validation set only has 13 cases, 7 non-responders and 6 responders.

Overall, the results are summarised in *Table 11*. It has been seen that the models with consistently better performance are Decision Tree-based or the Multilayer Perceptron. The datasets that achieve the best performances are the core tumour radiomics only and the one with all features combined. The border radiomics dataset shows the worst performance alone and in combination with other features. However, it increases the performance of the tumour radiomics and clinical features model, achieving the best accuracy in combination.

Dataset	Best Model	Validation Accuracy	AUC	Sensitivity	Specificity
Core tumour radiomics (CTR)	Random Forest	0.77	0.70	0.67	0.86
Border radiomics (BR)	XGBoost	0.62	0.54	0.5	0.71
CTR + BR	Multilayer Perceptron	0.69	0.67	0.67	0.71
Clinical	Decision Tree	0.69	0.59	0.83	0.57
Clinical + CTR	Random Forest	0.69	0.75	0.50	0.86
Clinical + BR	Multilayer Perceptron	0.69	0.67	0.67	0.71
Clinical + CTR + BR	Multilayer Perceptron	0.77	0.70	0.83	0.71

Table 11. Summary of the model results for each dataset.

4.8 Analysis of results and discussion

4.8.1 Dataset discussion

Firstly, the results presented in the previous section indicate that a big amount of the tested models seemingly present overfitting. There are several reasons behind overfitting such as having noisy training data, limited size of the training set, models that are too complex or training and validation sets that are not representative [82]. The datasets at hand have a small number of samples (84 in total, split into 71 to train and 13 to validate) and have unrepresented groups in the clinical features, as seen in *Table 1*. For example, there are notably less high T stages both for responders and non-responders. This is because, in clinical practice, lower T stages are more common.

Moreover, it could be expected that the radiomic datasets have noise that hinders the performance of algorithms, especially in the tumour borders dataset. This noise does not necessarily come from the image quality or normalization carried out during preprocessing because the core tumour dataset shows good performance and the features were extracted from the same images. The noise is likely due to the nature of tumour borders. A lot of tissues can be included in them: mesorectal fat (light grey), mucosa or submucosa (dark grey), and liquid pools or lumen (very bright) [3]. This heterogeneity probably adds noise to the data, and the models are not able to generalize to new samples that might have different border tissues (and consequently different radiomic features).

The dataset also relies on the segmentations. Although they are validated by a professional to avoid errors, the segmentation style or technique may not be optimal, introducing noise or inaccuracies that hinder the performance.

Moreover, the training dataset is not balanced based on ground truth labels. The validation dataset has six responders and seven non-responders. However, the train dataset contains 30 responders and 41 non-responders. This probably pushes the models to predict the non-responders class more, achieving higher specificities. This is the case for almost all tested and selected models.

However, since data is extracted from real clinical cases, the dataset is not expected to be balanced but to reflect the situation of current clinical practice in the hospital. Nonetheless, these datasets could be balanced by removing samples from overrepresented classes, but this would greatly decrease the number of available samples, which could still result in overfitting and poor performance.

4.8.2 Model results discussion

Despite the nature of the dataset and the overfitting, some models achieved good results. These are the Random Forest from the tumour core radiomics dataset and the Multilayer Perceptron from the dataset with all features. Feature importance plots were calculated for the most successful model of each dataset to aid in understanding the decision-making process behind the predictions.

- *Whole tumour radiomics only dataset:*

The Random Forest model is the one with best performance in the core tumour radiomics dataset. Random Forest models are quite explainable because they consist of an ensemble of Decision Trees, which are very explainable on their own [83]. The feature importance can be directly extracted using the *.feature_importances_* method. The resulting plot is displayed in *Figure 30.*

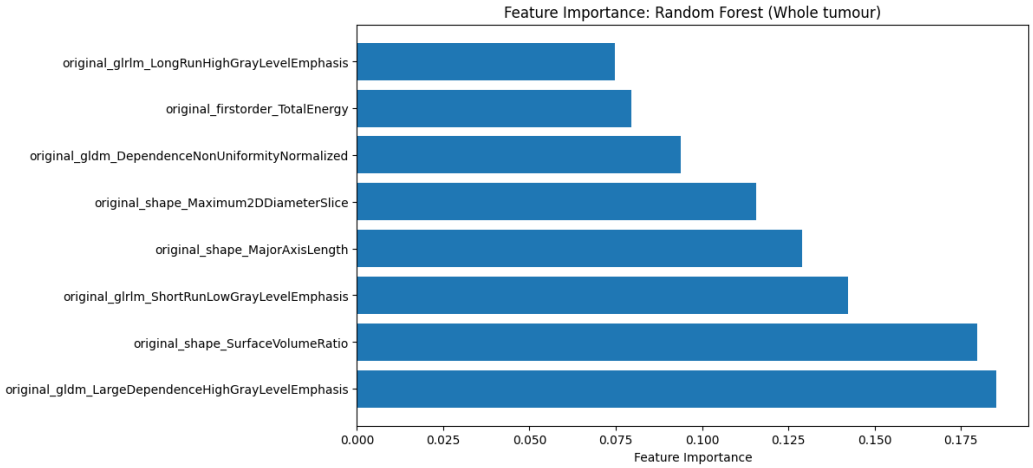

Figure 30. Feature importance graph of the best model (Random Forest) for the whole tumour dataset.

The top four features with most importance are Large Dependence High Gray Level Emphasis (gldm), Surface Volume Ratio (shape), Short Run Low Gray Level Emphasis (glrlm), and Major Axis Length (shape). Two are related to tumour shape and size and the other two are related to tumour texture.

One of the ways to see how they predict the response is by looking at the correlation matrix of *Figure 14.* Large Dependence High Gray Level Emphasis (gldm) and Major Axis Length (shape) have negative correlations of -0.28 and -0.20, respectively. This means that when these values increase, the response value decreases and thus tends to predict non-responders. This could mean that if the tumour is bigger in size, because it has a larger major axis length, the patient is less likely to respond. Some studies concluded that there is a significant inverse relationship between tumour response and tumour size [84].

On the other hand, if the tumour has more large homogeneous regions with high voxel intensity (bright regions), as explained by the large dependence high grey level feature, the patient is also less likely to respond. Bright regions on an MRI indicate liquid content or mucinous tumours, which have been associated with worse nCRT response by existing literature [85] [86].

On the other hand, Surface Volume Ratio (shape) and Short Run Low Gray Level Emphasis (glrlm) show positive correlations with response, of 0.17 and 0.28, respectively. Irregular tumours that are not sphere-like have high surface volume ratio and, according to these results, this may be associated with positive response. This is inconsistent with findings reported in literature as tumour surface irregularity is commonly associated with worse response [87] [88]. Tumours with high values of short run low grey level emphasis have lots of regions with dark speckles which, in MRI, might correspond to areas with fibrotic tissue. According to these results, this phenomenon is associated with positive tumour response. Literature confirms that tumour fibrosis is associated with improved outcomes as the tumour seems to be less aggressive [89].

- *Tumour border radiomics datasets:*

The models with border radiomics features showed poor results overall, suggesting that tumour borders do not have predictive capability for tumour response. However, these findings could be due to the fact that the borders were calculated from the tumour ROIs automatically using mathematical operations (erode and dilate) and were not manually delineated nor validated. The findings are consistent with some studies that found that including tumour borders does not improve the prediction of response and, in fact, it decreased predictive capacity due to noisy data coming from the heterogeneous mix in tissues included in the border masks [90]. However, another showed that borders enhance predictions [26]. Borders were manually delineated in both studies.

- *Clinical feature datasets:*

As reviewed in the background section, clinical features often enhance the performance of the models. This, in the present study, is the case for all datasets except the tumour core one. The best model including clinical features is Multilayer Perceptron for the dataset with all features combined. Since Multilayer Perceptron is not an explainable model itself, the SHapley additive explanations (SHAP) technique has been used to compute feature importance. This technique adds a weight, referred to as the shapley value, to each feature in the model which is used to measure its contribution to the final prediction [91]. The resulting SHAP plot can be found in *Figure 31*. The plot shows the features ranked by importance from most important on the top to least important on the bottom. The colours show the value of each data point (high in red and low in blue) and the x-axis shows the weight of the values of each feature on the predictions [92].

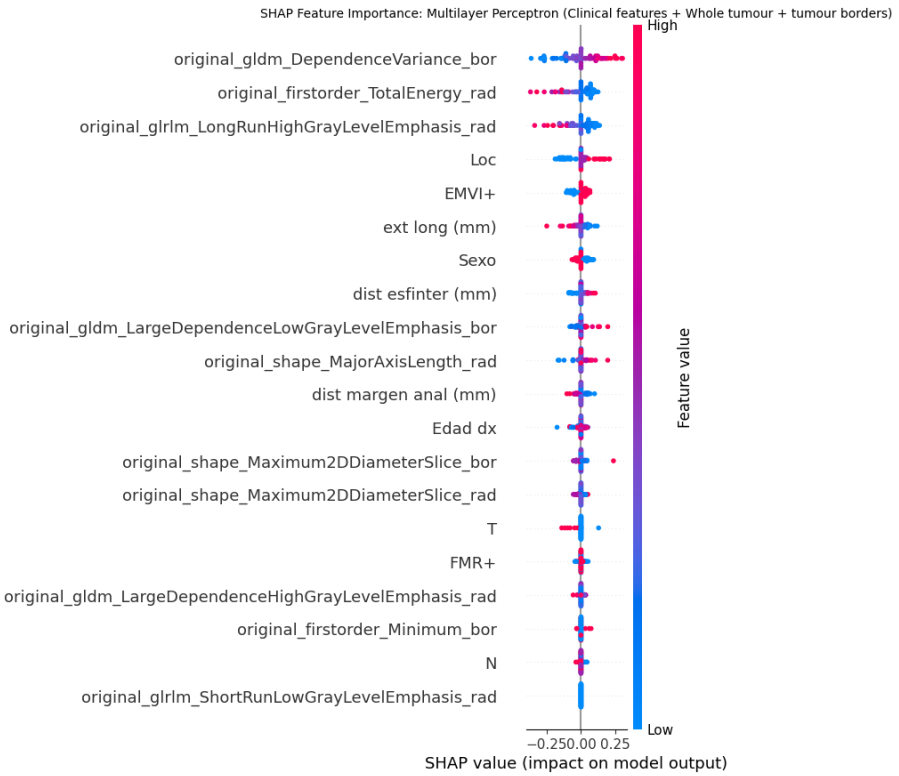

Figure 31. Feature importance graph of the best model (Multilayer Perceptron) for the dataset with all features combined. The plot was obtained using the SHAP technique.

The top five most important features are Dependence Variance (gldm) from the border radiomics dataset, Total Energy (firstorder) from the core tumour dataset, Long Run High Gray Level Emphasis (glrlm) from the core tumour dataset, tumour localization from the clinical dataset, and Extramural Vascular Invasion (EMVI+) from the clinical dataset. The plot is consistent with the correlation coefficients of each variable: when the coefficient is positive, the high feature values in the plot have a positive impact on response, contributing to the prediction of the responders class, and vice versa.

Dependence Variance is a measure of the heterogeneity of the signal texture. According to the plot, more texture heterogeneity in the tumour borders, meaning that for instance there are several different tissues surrounding the tumour, is an indicator of good response. Studies exploring the relationship between tumour border heterogeneity and response were not found. However, literature shows that heterogeneity in tumour borders decreases the predictive capabilities of the models [90].

Tumours with high total energy have large volumes or high grey level intensity values (lots of bright spots). The plot shows that low values of total energy are related to better tumour response, meaning that tumours with smaller volumes or darker intensities (fibrosis) respond better. This is consistent with the sources found previously related to tumour size and tumour fibrosis, discussing that there is an inverse relationship between tumour response and tumour size and that fibrotic tumours seem to be less aggressive, which yields better responses [84] [89].

Tumours with high Long Run High Gray Level Emphasis have lots of large regions of high intensity, probably due to liquid infiltrations or mucinous nature. In fact, the features Large Dependence High Gray Level Emphasis, previously explained, and Long Run High Gray Level Emphasis are highly correlated

with a correlation coefficient of 0.74. As previously explained, literature shows that mucinous tumours are associated with worse nCRT response [85] [86], which is consistent with the results.

The localization of the tumour is another feature that shows high feature importance. In the plot, higher values, meaning tumours placed in the high rectum, have better responses. However, some studies found no differences between tumour location and response [93], while others found that tumours located in the low rectum present better treatment response [94], hence, the results of the plot are not supported by literature.

Finally, EMVI was also found to be relevant in predictions. This feature is categorical and can take a value of 2 (absence of extramural vascular invasion) or 1 (presence of extramural vascular invasion). The SHAP plot indicates that higher values, absence of EMVI, show better treatment response, which is supported by studies affirming that absence of extramural vascular invasion is a predictor of positive response [86] [94].

- *Clinical feature datasets (T and N subcategories):*

Surprisingly, the T and N features, which according to literature are very important markers of tumour response [3, 5, 9, 26], are ranked very low in the feature importance plot. This occurs in all the other feature importance plots of models with clinical features. The reason is that the values introduced are the general categories, which can only take values of 3 or 4 (in the case of T), instead of the subcategories, which are 3a, 3b, 3c, 3d, 4a and 4b (in the case of T). To see the full categories and subcategories and their description, refer to *Annex 1*. Only using the general categories results in low variance in these variables, providing little information, and the model does not take them into account.

To see if it has a real effect on model performance, the models were optimized and retrained using a dataset that split these features into the different subcategories, according to TNM staging data. Subcategories were hypothesised to provide more information and thus improve differentiation and results. The results, found in *Annex 4*, showed no increase in the performance of any model but were able to match the previous results. The best model for the new variables, among all the datasets, was still Multilayer Perceptron for the dataset with all features. The results of this model compared with the original one are displayed below.

Best Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	AUC	Sensitivity	Specificity	Precision	NPV	F1-score
Multilayer Perceptron (subcategories)	0.90 ± 0.02	0.64 ± 0.11	0.77	0.69	0.67	0.86	0.80	0.75	0.73
Multilayer Perceptron (original)	0.92 ± 0.01	0.63 ± 0.10	0.77	0.70	0.83	0.71	0.71	0.83	0.77

Table 12. Metrics for the best model (Multilayer Perceptron) for the clinical features + whole tumour + tumour border radiomics dataset. Top row: results with T and N subcategories. Bottom row: original results.

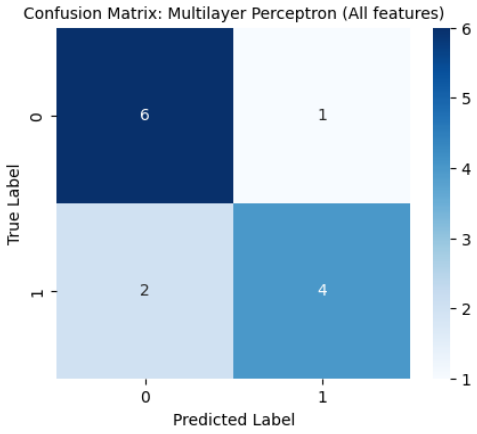

Figure 32. Confusion Matrix of the best model (Multilayer Perceptron) for the clinical features (with T and N subcategories) + whole tumour + tumour border radiomics dataset.

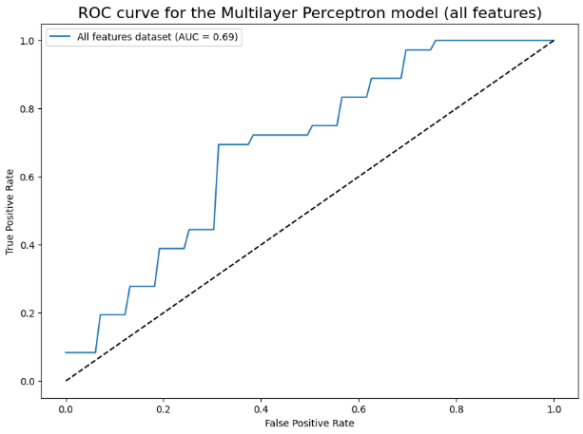

Figure 33. ROC curve and AUC of the best model (Multilayer Perceptron) for the clinical features (with T and N subcategories) + whole tumour + tumour border radiomics dataset

The model only predicts three cases wrong and the ROC curve shows constant performance across all thresholds. This model presents better specificity and precision and worse sensitivity and NPV than the original because it classifies more samples as negative than its original counterpart. The feature importance plot was also calculated using the SHAP technique (*Figure 34).*

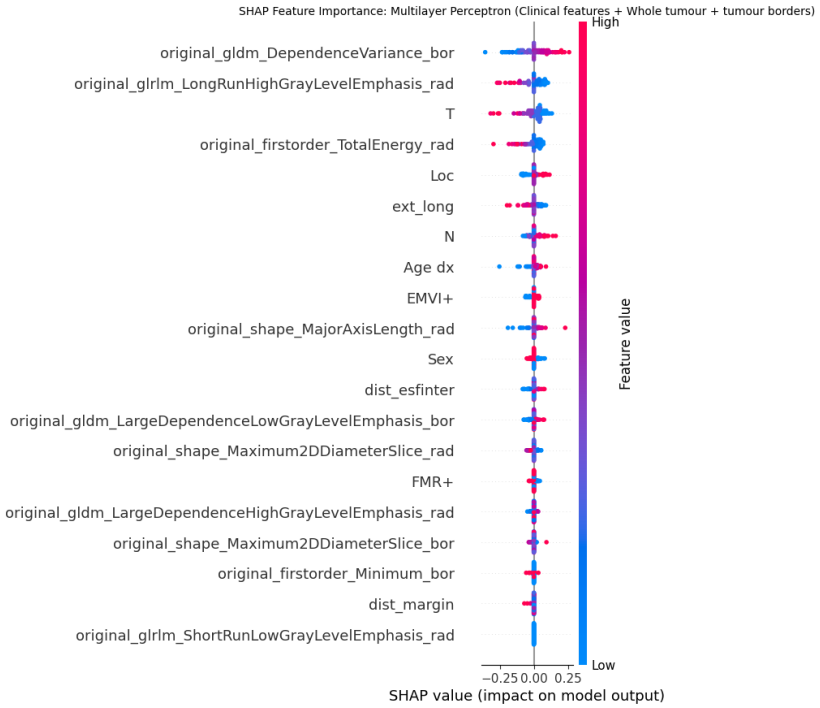

Figure 34. Feature importance graph of the best model (Multilayer Perceptron) for the dataset with all features combined with T and N subcategories. The plot was obtained using the SHAP technique.

T and N are higher in the ranking when including their subcategories in the data, showing that they probably provide information to the model now. This approach did not show better classification, but it highlights that small changes in the dataset have an impact on the model's inner structure.

The top five features are the same as before except for EMVI+ that has been replaced by T. The T feature shows that higher values, i.e., stagings of more advanced tumours, contribute to negative predictions, which is discussed in several papers [84] [85] [95].

5. Execution schedule

The project has taken place from February 1, 2025, to June 10, 2025, a total duration of 130 days. In this section the schedule of the project and its tasks will be discussed.

5.1 Work Breakdown Structure (WBS)

First and foremost, the workpackages and tasks for each package have to be defined. The project is broken down into six workpackages. The first one, project documentation, aims to capture all the literature research and written documents that have to be produced, specifically a report of the final degree project. Next, the preliminary training block is dedicated to the familiarization of the different software and images used. After, a data acquisition section breaks down all the tasks required to obtain the data from the hospital's databases and the segmented MRI studies. The fourth block is Segmentations, where all the tasks carried out to complete and validate the tumour segmentations are detailed. Next, Feature Selection and Dataset Description breaks down the tasks needed to select the variables in the dataset and to understand the dataset. The final workpackage, Model Training, captures the steps taken to select, optimize, train, validate, and explain the Machine Learning models. *Figure 35* shows a schematic of the structure of the project and *Table 13* shows a brief description of the tasks and their estimated deadlines.

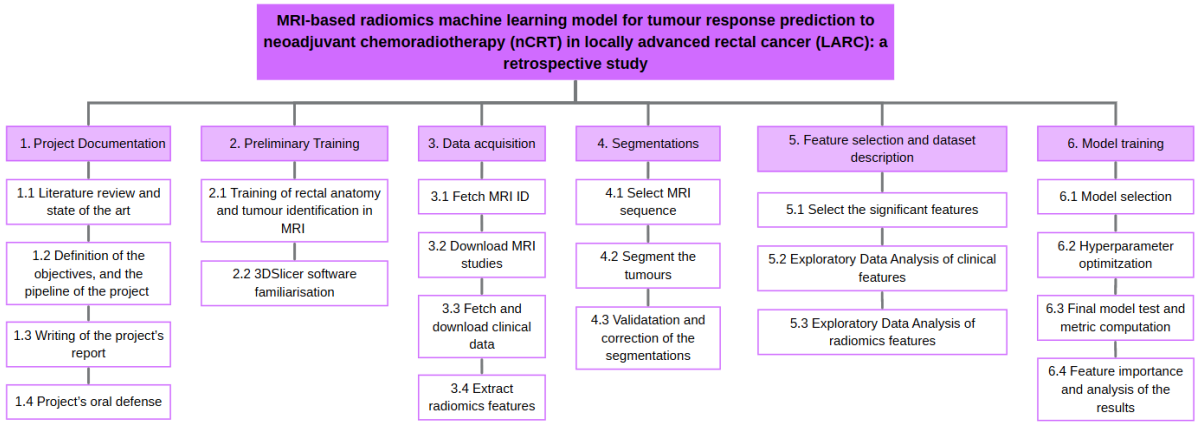

Figure 35. Schematic structure of the WBS of the project.

WBS ID	Task Name	Description	Duration (days)
1.1	Literature review	Bibliographic research on studies predicting nCRT response in patients with LARC, on rectal anatomy, on the state of the art and on the LARC medical basis. **Deliverables:** sources of the background section (2).	18
1.2	Definition of the objectives and pipeline	Determine the goals and scope of the project, what is the hypothesis to solve, and the methodology carried out. **Deliverables:** introduction section (1).	5
1.3	Writing of project's report	Writing of the written report of the project, from March 2025 to June 2025. **Deliverables:** written report	100
1.4	Project's oral defence	Production of the materials (ppt or similar) and script for the oral defence. **Deliverables:** ppt file and script.	7
2.1	Tumour Identification in MRI	Training to learn to identify structures in the MRI studies, particularly the tumour and surrounding structures. **Deliverables:** none.	5
2.2	3DSlicer software familiarisation	Training to learn how to segment the tumours using the software 3DSlicer by watching tutorials and reading the software documentation. **Deliverables:** none.	3

3.1	Fetch MRI ID	Select the MRI citations ID and dates based on the provided patient IDs using the *Enterprise Imaging Cloud* application. **Deliverables:** MRI ID and dates.	2
3.2	Download MRI studies	Use the MRI IDs to download the anonymized studies using the *Enterprise Imaging Cloud* application. **Deliverables:** MRI studies.	4
3.3	Download clinical data	Download the selected clinical features from the *Enterprise Imaging Cloud* and *SAP* applications. **Deliverables:** Clinical database and ground truth.	4
3.4	Extract radiomics features	Produce a code to calculate the border masks, normalize the images and extract the radiomic features from the segmentation and border masks using PyRadiomics. **Deliverables:** Code, radiomics database with ground truth.	4
4.1	Select MRI sequence	From the DICOM studies, select the sequence in which segmentations will be carried out and save it. **Deliverables:** MRI sequences.	2
4.2	Segment the tumours	Manually segment the 84 tumours using the software PyRadiomics. **Deliverables:** tumour segmentations.	40
4.3	Correction of the segmentations	Correct the segmentation masks with the help of a senior radiologist. **Deliverables:** corrected tumour segmentations.	15
5.1	Select significant radiomic features	Produce a code to select the most relevant radiomic features for tumour core and tumour borders. **Deliverables:** relevant radiomic features datasets.	5
5.2	EDA of clinical features	Carry out an Exploratory Data Analysis on the clinical features dataset to understand the nature of the data. **Deliverables:** clinical dataset description.	3
5.3	EDA of radiomics features	Carry out an Exploratory Data Analysis on the radiomics features datasets to understand the nature of the data. **Deliverables:** radiomics datasets description.	3
6.1	ML Model Selection	Select the models to train based on literature and produce code to train them. **Deliverables:** list of models to train and code.	5
6.2	Hyperparameter optimization	Optimize the performance of the models using hyperparameter optimization techniques implemented with Python code. **Deliverables:** optimal hyperparameters dictionary and code.	7
6.3	Model validation and metric computation	Train and validate the models using the best hyperparameter combinations, calculate the confusion matrixes, the metrics and the ROC curves. **Deliverables:** model metrics and code.	7
6.4	Analysis of the results	Produce the feature importance plots and analyse the most important features by performing a literature search. **Deliverables:** feature importance plots and analysis of results section (4.8)	7

Table 13. WBS dictionary with tasks descriptions, deliverables and task duration.

5.2 PERT/CPM Diagram

The Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) are used to optimize the timing of the program and determine the critical path, which is the path defined by the tasks that add up to the minimum time to complete the project [96]. The written report will be developed in parallel with the other tasks for efficiency. *Table 14* shows the task duration and precedencies. *Figure 36* shows the PERT diagram and the critical path highlighted in red.

WBS ID	PERT ID	Task Name (Reduced)	Precedent tasks	Duration (days)
1.1	A	Literature review	-	18
1.2	B	Definition of the objectives and pipeline	A	5
1.3	C	Writing of project's report	B	100
1.4	D	Project's oral defence	C, T	7
2.1	E	Tumour Identification in MRI	A	5
2.2	F	3DSlicer software familiarisation	-	3
3.1	G	Fetch MRI ID	-	2
3.2	H	Download MRI studies	G	4
3.3	I	Download clinical data	B	4
3.4	J	Extract radiomics features	M	4
4.1	K	Select MRI sequence	H	2

4.2	L	Segment the tumours	E, F, K	40
4.3	M	Correction of the segmentations	L	15
5.1	N	Select the significant radiomic features	J	5
5.2	O	EDA of clinical features	I	3
5.3	P	EDA of radiomics features	N	3
6.1	Q	ML Model Selection	O, P	5
6.2	R	Hyperparameter optimization	Q	7
6.3	S	Model validation and metric computation	R	7
6.4	T	Analysis of the results	S	7

Table 14. PERT diagram table and analysis of precedence. All WBS are assigned a PERT ID.

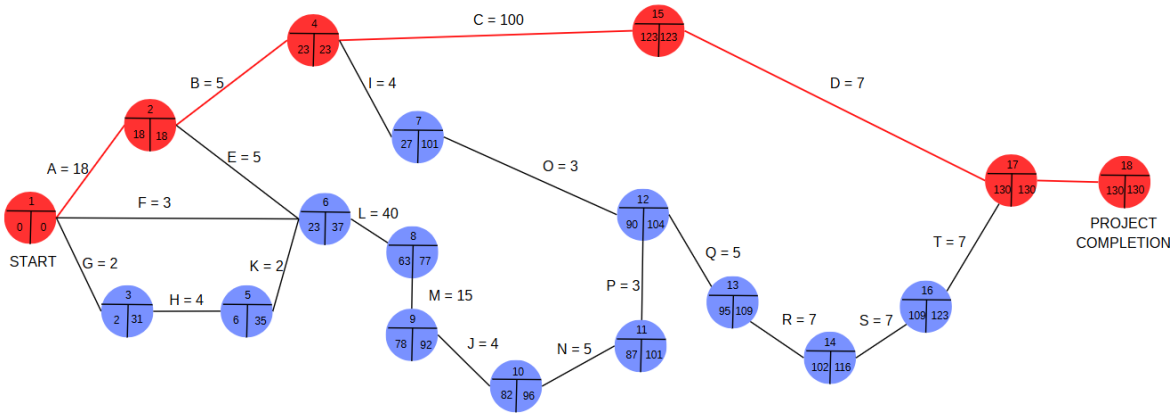

Figure 36. PERT diagram of the project.

5.3 GANTT Diagram

Finally, a GANTT diagram was developed. It is a bar plot used that shows the planning of the project as a timeline of the different tasks [96]. *Figure 37* shows the GANTT diagram of the project. Like in the PERT diagram, the red tasks represent the critical path. The other tasks are non-critical and have margin.

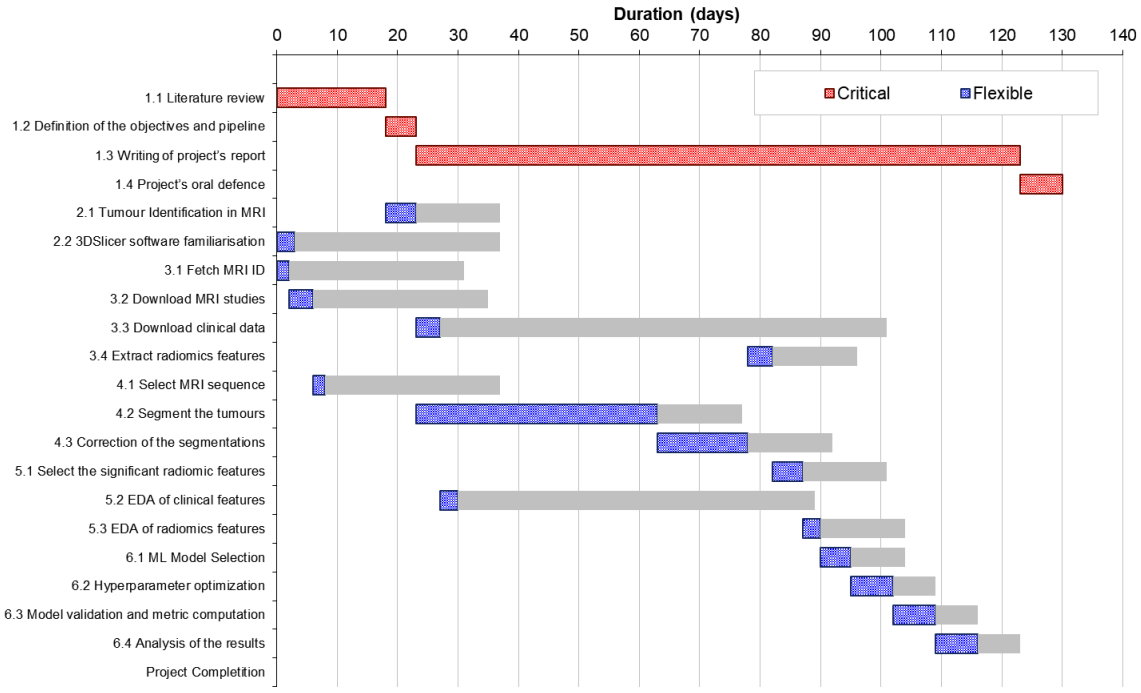

Figure 37. GANTT diagram of the project.

6. Technical feasibility

In this section, the Strengths, Weaknesses, Opportunities and Threats of the project will be discussed, what is known as a SWOT analysis.

The strengths of this project are that the segmentations are manual and validated by a senior radiologist from the Image department of the hospital. This ensures that all segmentation ROIs are strictly on the tumour. Another important aspect of this project is the explainability of the models. Since relevant features are selected before training and most models used are explainable or XAI techniques can be applied, it can be known which features have more predictive value and what they predict. Moreover, the methods applied in this project are reproductible which can help in multicentre or validation studies. This is also important to identify markers of prognosis of the tumour that are clinically relevant. Furthermore, the pipeline is custom made throughout, allowing full control of each step. Moreover, all the data used in the models is routinely acquired in clinical practice thus there is no extra cost for the hospital.

The weaknesses of this project are the small number of available cases to train and validation as machine learning models require, especially in datasets with high dimensionality, a large number of samples. Having few samples may lead to a dataset that is not representative of the studied population, especially in tumour classes. This can also lead to overfitting of the models as they are not able to adapt to new samples and classify them properly [82]. Another point to take into consideration is that to be able to apply this pipeline in a real clinical setting, segmentation should be automatic as manual delineations are highly time consuming and would increase the workload of the radiologists. However, since the goal is to identify features that give information about the prognosis of the tumour, there might not be a need to segment the whole tumour. With a partial representative segmentation such information could be obtained. Moreover, scaling the project to a larger dataset, without automatic segmentation tools, can also be challenging as a big number of studies should be segmented manually. Additionally, the ground truth labels used in this project are based on TRG which may not be the most reliable method.

This project has many future opportunities to develop research further. The pipeline can be generalized to be used in other types of tumours with ease or to be used in other hospitals as it is simple and reproductible. Most of the workload of the project relies on the manual segmentation of the studies that, as mentioned, is a very time-consuming process. Manual segmentation is necessary in this project as there are not enough cases to train an automatic model and no automatic model has been developed and validated for this purpose. However, if this study can be transformed into a multi-centre study, more cases can be added, and developing an automatic segmentation system would be feasible. Other questions arise from this project: can ML help differentiate responders and non-responders using the information from the second MRI study? Moreover, a model like this can be integrated into clinical practice to help doctors choose a personalized treatment plan and to adapt it, if necessary, to increase the chances of positive treatment response.

The threats that the project presents are improper image normalization, as the MRI scans in this work have been acquired from multiple machines by different technicians thus a good image normalization is key to ensure that the features are uniform and that the project can be reproduced in other centres. Moreover, it has been seen that there is some variability in sequence acquisition between studies and centres, especially regarding DWI, slice thickness and tumour axis sequences. Another threat is that the

adoption of an AI tool in routine clinical practice may be refused by some professionals due to the lack of trust in the technology. However, since the models are explainable, doctors may be more willing to adopt the technology as they can know how and why the decision was made. Finally, owing to the fact that AI is in rapid development, the pipeline and the technology used in this project may be obsolete in the near future, thus there is a need to constantly develop and improve the models to keep up with trends.

The points explained above are summarized in the following table:

STRENGHTS	WEAKNESSESS
• Senior Radiologist validated manual segmentations • Model explainability and pipeline reproducibility • Identification of clinically relevant imaging and clinical markers (non-invasive) • Custom made pipeline allows full control of the process. • Use of routinely acquired medical data to train the models	• Small number of samples with high dimensionality (limited diversity in dataset population) • Potential overfitting due to the small number of samples • Difficulty to apply the pipeline in a clinical setting. • Manual segmentation process is time-consuming and non-scalable • Potentially unreliable ground truth
OPPORTUNITIES	**THREATS**
• Generalizable pipeline: other tumour types and multi-centre studies • Possibility of developing an automatic segmentation model using the developed segmentations as ground truth • Differentiation between responders and non-responders in the post-nCRT MRI • Potential pipeline integration to plan personalized treatment and adaptive treatment strategies	• Improper image normalization • Variability in imaging protocols across studies and centres • Implementation resistance from clinicians due to lack of trust in AI tools • Rapid development of AI may make the model obsolete.

Table 15. SWOT analysis of a radiomics-based AI project to predict tumour prognosis.

7. Economic feasibility

The budget for this project encompasses costs related to the physical equipment required, the software used and costs of the workers involved. In particular, the costs of the MRI scans and the computers used to download the data, perform the segmentations, analyse the data and build the models will be regarded as physical equipment. There are two types of computers used: a hospital's workstation used to download the medical data which consists of two monitors (600 € each), and a desk computer (800 €) and a laptop where the models will be trained on, after carrying out bibliography research and data extraction and analysis, with an approximate cost of 700 €. All the software used, 3DSlicer and the Python libraries, is open-source and thus it is free. The personnel involved in the project include an entry-level biomedical engineer and a senior radiologist.

The following table includes the economic value of the different items mentioned and the total calculated budget for the project.

ITEM	COST
Physical equipment:	
Hospital Workstations: 5% * 2000 €	100 €
Laptop: 25 % * 700 €	175 €
MRI scans: 150 € * 86 scans	12900 €
Software:	
3DSlicer	0 €
Python libraries	0 €
Personnel:	
Biomedical Engineer: 300 h * 18€/h	5400 €
Senior Radiologist: 50 h * 25€/h	1250 €
TOTAL	**19825 €**

Table 16. Project budget divided in type of cost.

The percentages of the physical equipment items indicate that they were not bought and exclusively used for this project and the percentage is an estimate of the amortization of the equipment during this project. That is, 5% of the total amortization of the hospital workstations was used to fetch and download the data and 25% of the total amortization of the laptop was used to segment, analyse the data and build the models, as these are more demanding tasks.

The values of all the items in the budget were extracted from the hospital's economic data.

In total, the project has an estimated budget of approximately 20000 €. The most expensive item in the budget is the MRI scans. However, these scans are taken in routine clinical practice therefore they are primarily used to treat the patients and are reused for this project. Thus, the project's budget, without taking into account the price of the MRI scans, is 8140 €.

The budget broken down according to the tasks proposed in the GANTT diagram can be found below.

WBS ID and Task Name	Cost Breakdown	Cost
1.1 Literature review and state of the art	Biomedical Engineer (18€/h * 15h) + Laptop (700€ * 1 %)	277 €
1.2 Definition of the objectives and the pipeline of the project	Biomedical Engineer (18€/h * 5h) + Laptop (700€ * 1 %) + Senior Radiologist (25€/h * 3h)	172 €
1.3 Writing of the project's report	Biomedical Engineer (18€/h * 70h) + Laptop (700€ * 4 %)	1288 €
1.4 Project's oral defence	Biomedical Engineer (18€/h * 10h) + Laptop (700€ * 1 %)	187 €
2.1 Training of Rectal Anatomy and Tumour Identification in MRI	Biomedical Engineer (18€/h * 10h) + Senior Radiologist (25€/h * 8h) + Hospital Workstation (2000 € * 1%)	400 €
2.2 3DSlicer software familiarisation	Biomedical Engineer (18€/h * 10h) + Laptop (700€ * 1 %)	187 €
3.1 Fetch MRI ID	Biomedical Engineer (18€/h * 5h) + Senior Radiologist (25€/h * 3h) + Hospital Workstation (2000 € * 0.5%)	175 €
3.2 Download MRI studies	Biomedical Engineer (18€/h * 10h) + Senior Radiologist (25€/h * 3h) + Hospital Workstation (2000 € * 1%) + MRI scans (150 € * 86 scans)	13175 €
3.3 Fetch and download clinical data	Biomedical Engineer (18€/h * 10h) + Senior Radiologist (25€/h * 4h) + Hospital Workstation (2000 € * 1%)	300 €
3.4 Extract radiomics features	Biomedical Engineer (18€/h * 5h) + Laptop (700€ * 1 %)	97 €
4.1 Select MRI sequence	Biomedical Engineer (18€/h * 5h) + Laptop (700€ * 0.5 %) + Senior Radiologist (25€/h * 3h) + Hospital Workstation (2000 € * 0.5%)	178.5 €
4.2 Segment the tumours	Biomedical Engineer (18€/h * 50h) + Laptop (700€ * 5 %)	935 €
4.3 Validation and correction of the segmentations	Biomedical Engineer (18€/h * 20h) + Laptop (700€ * 2 %) + Senior Radiologist (25€/h * 16h) + Hospital Workstation (2000 € * 1%)	794 €
5.1 Select the significant features	Biomedical Engineer (18€/h * 5h) + Laptop (700€ * 0.5 %)	93.5 €
5.2 Exploratory Data Analysis of clinical features	Biomedical Engineer (18€/h * 5h) + Laptop (700€ * 0.5 %)	93.5 €
5.3 Exploratory Data Analysis of radiomics features	Biomedical Engineer (18€/h * 5h) + Laptop (700€ * 0.5 %)	93.5 €
6.1 Model Selection	Biomedical Engineer (18€/h * 10h) + Laptop (700€ * 1 %)	187 €
6.2 Hyperparameter optimization	Biomedical Engineer (18€/h * 20h) + Laptop (700€ * 2 %)	374 €
6.3 Final model test and metric computation	Biomedical Engineer (18€/h * 10h) + Laptop (700€ * 2 %)	194 €
6.4 Feature Importance and analysis of the results	Biomedical Engineer (18€/h * 20h) + Laptop (700€ * 2 %) + Senior Radiologist (25€/h * 10h)	624 €
TOTAL	**Biomedical Engineer (18€/h * 250h) + Laptop (700€ * 25 %) + Senior Radiologist (25€/h * 50h) + Hospital Workstation (2000 € * 5%) + MRI scans (150 € * 86 scans)**	**19825 €**

Table 17. Project budget broken down into GANTT diagram tasks.

Again, the most expensive task is the download of MRI studies but it is not a real additional cost as these are acquired during clinical practice.

8. Regulation and legal aspects

This project uses medical information to build the database the models will be trained on. There are laws regulating the data treatment, especially for medical data. The models used to predict are based on Artificial Intelligence, which is not highly regulated yet. In the recent years, the European Union has made an effort to regulate the use of AI, pioneers in the world. The different laws applicable to the project are summarised below.

8.1 EU General Data Protection Regulation (GDPR) (EU Regulation 2016/679) and Spanish Organic Law on Data Protection and Digital Rights 3/2018

The data used in this project contains personal and medical information of patients. There are laws that regulate the treatment of personal data that have to be followed. The *Spanish Organic Law on Data Protection and Digital Rights (LOPDGDD) 3/2018* [97] is the adaptation of the *EU General Data Protection Regulation (GDPR) (EU Regulation 2016/679)* [98] to the Spanish legal system. The GDPR establishes the principles of data treatment and the rights of the patients. In this project, data is processed for scientific research and public interest purposes without explicit consent, as permitted under Articles 6 and 9. Article 89 of the GDPR states the safeguards and derogations related to data processing for research purposes that must be applied to ensure data minimisation. Pseudonymisation is a highly recommended strategy. It is important to note that anonymous data falls outside the scope of GDPR. In medical imagining, total anonymization is difficult because it deals with images from the patient's body, where certain characteristics can be identified, thus pseudonymisation strategies have been carried out. The law also states that some rights can be derogated if exercising them would impair the fulfilment of the research processes. These include the right of access by the data subject, the right to rectification, the right to restriction of processing and the right to object. Moreover, article 35 of the law states that a data protection impact assessment must be carried out by the data processer if the processing suposes a high risk to the rights and freedoms of the data subjects.

The LOPDGDD confirms what has been previously stated and adds that in the case of use of medical data, even pseudonymised, in research for purposes for which explicit consent hasn't been obtained, an ethical committee has to approve such use. This project is under evaluation of the Ethical Committee of the hospital.

8.2 European Artificial Intelligence Act (Regulation (EU) 2024/1689)

The European AI Act is the first ever legal framework on AI worldwide and its goal is to ensure trust in AI. It will be fully applicable by August 2026. This regulation lays down the rules for the use of AI systems in the European Union, the prohibitions of certain AI practices, the specific requirements for high-risk AI systems, the rules for transparency of certain AI systems and the rules to place AI systems on the market. This regulation is applicable to the present project as its goal is to develop an AI system to predict response to a treatment [99].

This regulation classifies AI systems based on their risk. Unacceptable risk, the highest risk category, includes all the systems that are considered a clear threat to the safety, livelihoods and rights of people and are prohibited. This includes harmful practices such as social scoring, untargeted scraping of the internet or CCTV material to create or expand facial recognition databases, emotion recognition in

workspaces or educational institutions or real-time remote biometric identification for law enforcement purposes in public spaces. High risk systems are defined as those that can pose serious risks to health, safety or fundamental rights to the people and are subjected to strict obligations before being deployed on the market. The category includes AI systems that can determine the access to education, to a workplace, to asylum or migration, AI-based safety systems or AI used in democratic processes. Risk mitigation strategies, cybersecurity or submission of detailed documentation of the system are some of the required obligations to ensure safety. Limited risk systems include the systems that need to be transparent about the use of AI such as informing the users that the chatbot they are talking to is an AI or informing that an image has been AI-generated. The final risk category, minimal or no risk, includes almost all AI systems such as AI videogames or spam email filters, for which the law does not introduce rules [100].

Regarding the applicability of this law in the field of medical imaging, the European Society of Radiology (ESR) released a statement identifying the key policies of the AI Act on radiology. According to the statement, AI systems related to medical imaging, such as the one developed in the present project, are in the high-risk category as they have a potential impact on the health of the population. Additionally, the statement reviews the EU regulation and makes recommendations of educating radiology professionals about AI, of data handling, of the required human oversight, of model transparency and quality management in AI and of clinical studies [101].

Furthermore, the ESR demands the European Commission to release specific implementation guidelines of the AI Act for the medical community to facilitate implementation and fulfilment of the legal requirements [101].

8.3 European Union Medical Device Regulation (EU MDR) (EU Regulation 2017/745)

To market a medical device in the EU, manufacturers must obtain CE marking by adhering to the regulations of the European Union Medical Device Regulation (EU MDR). In this project, AI could qualify as a medical device as it is used to make or assist in clinical decisions [102]. However, since the goal of this project is to do a concept evaluation that developing such software is possible and not to market the resulting product, some of the rules are not applicable.

The MDR lays down the main rules governing medical device development, manufacture, commercialization and surveillance. The regulation defines classes based on risk that the products have to be classified in. Software products are part of the higher risk categories, especially if their intended use is to support diagnosis or treatment decisions. Products belonging to high-risk classes, manufacturers are required to submit periodic safety update reports [102].

Moreover, the law increases the medical evidence needed to commercialize a product, when compared to its predecessor: the Medical Device Directive. It also describes the creation of a European Database on medical devices to integrate information on devices, manufacturers, clinical investigations, and market surveillance. It also defines Unique Identification system (UDI system) to improve traceability transparency of the products, enhancing post-market related safety activities and regulatory oversight [102].

8.4 Spanish Law 14/2007 on Biomedical Research

This project falls under the scope of the Biomedical Research law as it involves human medical data extracted from clinical records. The objective of the law is to regulate biomedical research to guarantee one's dignity, respect to integrity, rights and freedoms. It also states that all investigation projects require the evaluation and approval of the Research Ethics Committee, regardless of whether data is anonymized [103]. As discussed, this project is under evaluation of the Ethical Committee of the hospital.

The law also states that explicit informed consent is required for a person to be included in the research study. However, when data is collected from hospitals, the original patient consent must allow the reuse of data for secondary purposes such as research. Data treatment and processing must be protected and carried out in accordance with the rules of the pertinent regulation [103], in this case the GDPR [98] and the Spanish Organic on Data Protection and Digital Rights Law 3/2018 [97]. Moreover, the research's benefits must outweigh the risks and states the special regulations for vulnerable populations, such as minors or the disabled [103].

Furthermore, the law also reviews the rules for biomedical research based on biological samples, which is not applicable in this project. It states that all patients have to sign an explicit consent to use these samples for research, that donation in exchange for money is prohibited, the transportation or storage of the samples (biobanks) and regulate the type of genetical tests that can be carried out. The law also regulates the research that involves invasive procedures and embryonic research, also not applicable in this project [103].

9. Conclusions

The main goal of this project was to build an Artificial Intelligence model capable of predicting the response of LARC after neoadjuvant therapy. Other subobjectives arised from this goal: determining clinical and image markers that are key in predicting LARC evolution and verifying similar studies with data from *Hospital de la Santa Creu i Sant Pau*. Furthermore, it was hypothesized that response can be predicted with machine learning models and that a combination of clinical and radiomics features would yield the best performance.

The findings showed that predicting the response to neoadjuvant chemotherapy of patients with LARC using machine learning models is possible. Using explainable algorithms, such as Decision Tree and Random Forest, and explainable AI techniques, such as SHAP, allowed the extraction of the features that are most relevant to compute the predictions.

However, the combination of radiomics and clinical features did not yield better results than the radiomics model alone. It has been discussed that the reason is that the clinical features present underrepresented classes and thus the model cannot generalize well to new samples. The models also presented probable overfitting, which was slightly mitigated using hyperparameter tuning techniques.

The best models have been obtained in the core tumour radiomics dataset, with Random Forest, and in the dataset with all features, with Multilayer Perceptron. The validation set consisted of 13 samples with 7 non-responders and 6 responders. The validation accuracy in both cases was of 0.77 and the AUC of 0.70. The model with the core tumour radiomics features showed better specificity, 0.86, whereas the model with all features showed better sensitivity, 0.83. These are positive results that fulfil the main goal of developing a ML model capable of predicting response. The results can also be considered positive taking the limitations of the study into account: the small number of samples and the underrepresentation of classes, which are frequent issues when with real patient data, especially in single centre studies.

The most relevant features varied depending on the model. For the Random Forest the Large Dependence High Gray Level Emphasis (gldm), Surface Volume Ratio (shape), Short Run Low Gray Level Emphasis (glrlm), and Major Axis Length (shape) were the most relevant. For the Multilayer Perceptron model Dependence Variance (gldm) from the border radiomics dataset, Total Energy (firstorder) from the core tumour dataset, Long Run High Gray Level Emphasis (glrlm) from the core tumour dataset, tumour localization from the clinical dataset, and Extramural Vascular Invasion (EMVI+) from the clinical dataset were the most relevant. Most of the relationships of these variables with response were supported by literature, confirming that shape characteristics, texture features and clinical features are predictors of response. However, a specific imaging or clinical marker with high predictive capability across all models could not be identified.

T and N staging were one of the least important clinical features due to the low variance in categories. To assess this, models were optimized and trained again with clinical features taking the staging subcategories into account. The overall performance did not improve but T and N staging went up in the feature importance rankings.

The predictive capability of the tumour borders was also assessed but it can be concluded that, for the methodology carried out, the tumour borders show poor predictive capability of response to neoadjuvant treatments in patients with LARC.

One of the most time-consuming steps of the pipeline is manual segmentation. However, the main advantage of it is that it is very precise. To ease the workload, a future line of work could explore the development of an algorithm to automatically segment the tumours using the 84 segmentations produced in this study as ground truth. This would ease workload when adding new samples and in the application of the pipeline in clinical practice.

Another future line of work is to assess the performance of the algorithms when changing the method to classify responders and non-responders. In this study, this has been carried out using Mandard's TRG, which is consistent with similar studies. However, it is believed that classifying response based on the comparison between cTNM and ypTNM could change the results as, sometimes, there is discrepancy between the response indicated with TRG and the one seen in the ypTNM staging after surgery. However, there is no clear method to classify response based on the post-surgery staging yet.

Furthermore, as discussed in the motivation section, there have been studies exploring another research branch to avoid unnecessary surgeries in cured patients. Their aim is to improve differentiation between pathological complete responders (pCR) and non-responders in the post-nCRT MRI before the surgery. To achieve this, the pipeline proposed in this project should be applied to the post-nCRT MRI studies, which could be interesting to assess its generalizability.

To conclude, it is possible to predict, with good results, the response to neoadjuvant chemotherapy in patients with locally advanced rectal cancer using radiomics-based machine learning algorithms, despite the dataset limitations. Furthermore, the addition of clinical data does not improve the results and the tumour borders show poor predictive capability. However, radiomics features extracted from the tumour core are able to predict the response well. Feature importance analysis was also assessed but the identification of imaging and clinical biomarkers for tumour response remains unknown.

10. References

[1] Asociación Española Contra el Cáncer (AECC). (2025, March). *Cáncer de colon: Epidemiología.* contraelcancer.es. https://www.contraelcancer.es/es/todo-sobre-cancer/tipos-cancer/cancer-colon/epidemiologia-cancer-colon Last Accessed: March 2, 2025

[2] National Cancer Institute. (2025, February 12). *Rectal Cancer Treatment (PDQ®) Health Professional Version. National Cancer Institute;* Cancer.gov. https://www.cancer.gov/types/colorectal/hp/rectal-treatment-pdq Last Accessed: March 2, 2025

[3] Mak, S., Hulse, P. A., & Carrington, B. M. (2016). *MRI Manual of Pelvic Cancer*. CRC Press.

[4] Di Costanzo G, Ascione R, Ponsiglione A, Tucci AG, Dell'Aversana S, Iasiello F, et al. Artificial intelligence and radiomics in magnetic resonance imaging of rectal cancer: a review. Explor Target Antitumor Ther. 2023;4:406–21. https://doi.org/10.37349/etat.2023.00142

[5] Ma, X., Shen, F., Jia, Y., Xia, Y., Li, Q., & Lu, J. (2019). MRI-based radiomics of rectal cancer: preoperative assessment of the pathological features. *BMC Medical Imaging, 19(1)*. https://doi.org/10.1186/s12880-019-0392-7

[6] Koh, D.-M. (2020). Using Deep Learning for MRI to Identify Responders to Chemoradiotherapy in Rectal Cancer. *Radiology*, *296*(1), 65–66. https://doi.org/10.1148/radiol.2020200417

[7] Ferrari, R., Mancini-Terracciano, C., Voena, C., Rengo, M., Zerunian, M., Ciardiello, A., Grasso, S., Mare', V., Paramatti, R., Russomando, A., Santacesaria, R., Satta, A., Solfaroli Camillocci, E., Faccini, R., & Laghi, A. (2019). MR-based artificial intelligence model to assess response to therapy in locally advanced rectal cancer. *European journal of radiology*, *118*, 1–9. https://doi.org/10.1016/j.ejrad.2019.06.013

[8] Perez, R. O. (2011). Predicting Response to Neoadjuvant Treatment for Rectal Cancer: A Step Toward Individualized Medicine. *Diseases of the Colon & Rectum*, *54*(9), 1057–1058. https://doi.org/10.1097/dcr.0b013e31822182ce

[9] Peterson, K. J., Simpson, M. T., Drezdzon, M. K., Szabo, A., Ausman, R. A., Nencka, A. S., Knechtges, P. M., Peterson, C. Y., Ludwig, K. A., & Ridolfi, T. J. (2023). Predicting Neoadjuvant Treatment Response in Rectal Cancer Using Machine Learning: Evaluation of MRI-Based Radiomic and Clinical Models. *Journal of gastrointestinal surgery: official journal of the Society for Surgery of the Alimentary Tract*, *27*(1), 122–130. https://doi.org/10.1007/s11605-022-05477-9

[10] Grover, V. P., Tognarelli, J. M., Crossey, M. M., Cox, I. J., Taylor-Robinson, S. D., & McPhail, M. J. (2015). Magnetic Resonance Imaging: Principles and Techniques: Lessons for Clinicians. *Journal of clinical and experimental hepatology*, *5*(3), 246–255. https://doi.org/10.1016/j.jceh.2015.08.001

[11] Jhaveri, K. S., & Hosseini-Nik, H. (2015). MRI of Rectal Cancer: An Overview and Update on Recent Advances. *American Journal of Roentgenology*, *205*(1), W42–W55. https://doi.org/10.2214/ajr.14.14201

[12] Delli Pizzi, A., Basilico, R., Cianci, R. *et al.* Rectal cancer MRI: protocols, signs and future perspectives radiologists should consider in everyday clinical practice. *Insights Imaging* **9**, 405–412 (2018). https://doi.org/10.1007/s13244-018-0606-5

[13] *TSE/FSE*. (2024). Questions and Answers in MRI. https://mriquestions.com/what-is-fsetse.html Last Accessed: April 4, 2025

[14] Rosen RD, Sapra A. TNM Classification. [Updated 2023 Feb 13]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2025 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK553187/ Last Accessed: March 25, 2025

[15] McCague, C., Ramlee, S., Reinius, M., Selby, I., Hulse, D., Piyatissa, P., Bura, V., Crispin-Ortuzar, M., Sala, E., & Woitek, R. (2023). Introduction to radiomics for a clinical audience. *Clinical Radiology*, *78*(2), 83–98. https://doi.org/10.1016/j.crad.2022.08.149

[16] van Griethuysen, J. J. M., Fedorov, A., Parmar, C., Hosny, A., Aucoin, N., Narayan, V., Beets-Tan, R. G. H., Fillion-Robin, J. C., Pieper, S., & Aerts, H. J. W. L. (2017). Computational Radiomics System to Decode the Radiographic Phenotype. *Cancer research*, *77*(21), e104–e107. https://doi.org/10.1158/0008-5472.CAN-17-0339

[17] *Radiomic Features — pyradiomics 2.2.0. documentation*. (2016). https://pyradiomics.readthedocs.io/en/latest/features.html Last Accessed: June 7, 2025

[18] May, K. (2024, May 13). *What Is Artificial Intelligence?* NASA. https://www.nasa.gov/what-is-artificial-intelligence/ Last Accessed: April 7, 2025

[19] Zhang, C., & Lu, Y. (2021). Study on Artificial Intelligence: The State of the Art and Future Prospects. *Journal of Industrial Information Integration*, *23*(23), 100224. https://doi.org/10.1016/j.jii.2021.100224

[20] gymnasium. (2025, March 6). PyPI. https://pypi.org/project/gymnasium/ Last Accessed: April 7, 2025

[21] Alowais, S.A., Alghamdi, S.S., Alsuhebany, N. *et al.* Revolutionizing healthcare: the role of artificial intelligence in clinical practice. *BMC Med Educ* **23**, 689 (2023). https://doi.org/10.1186/s12909-023-04698-z

[22] Pinto-Coelho L. (2023). How Artificial Intelligence Is Shaping Medical Imaging Technology: A Survey of Innovations and Applications. *Bioengineering (Basel, Switzerland)*, *10*(12), 1435. https://doi.org/10.3390/bioengineering10121435

[23] Rengo, M., Picchia, S., Marzi, S., Bellini, D., Caruso, D., Caterino, M., Ciolina, M., De Santis, D., Musio, D., Tombolini, V., & Laghi, A. (2017). Magnetic resonance tumor regression grade (MR-TRG) to assess pathological complete response following neoadjuvant radiochemotherapy in locally advanced rectal cancer. *Oncotarget*, *8*(70), 114746–114755. https://doi.org/10.18632/oncotarget.21778

[24] Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., Alonso-Moral, J. M., Confalonieri, R., Guidotti, R., Ser, J. D., Díaz-Rodríguez, N., & Herrera, F. (2023). Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information Fusion*, *99*(101805), 101805. sciencedirect. https://doi.org/10.1016/j.inffus.2023.101805

[25] Jiang, X., Zhao, H., Saldanha, O. L., Nebelung, S., Kuhl, C., Amygdalos, I., Lang, S. A., Wu, X., Meng, X., Truhn, D., Kather, J. N., & Ke, J. (2023). An MRI Deep Learning Model Predicts Outcome in Rectal Cancer. *Radiology*, *307*(5), e222223. https://doi.org/10.1148/radiol.222223

[26] Delli Pizzi, A., Chiarelli, A. M., Chiacchiaretta, P., d'Annibale, M., Croce, P., Rosa, C., Mastrodicasa, D., Trebeschi, S., Lambregts, D. M. J., Caposiena, D., Serafini, F. L., Basilico, R., Cocco, G., Di Sebastiano, P., Cinalli, S., Ferretti, A., Wise, R. G., Genovesi, D., Beets-Tan, R. G. H., & Caulo, M. (2021). MRI-based clinical-radiomics model predicts tumor response before treatment in locally advanced rectal cancer. *Scientific reports*, *11*(1), 5379. https://doi.org/10.1038/s41598-021-84816-3

[27] Feng, L., Liu, Z., Li, C., Li, Z., Lou, X., Shao, L., Wang, Y., Huang, Y., Chen, H., Pang, X., Liu, S., He, F., Zheng, J., Meng, X., Xie, P., Yang, G., Ding, Y., Wei, M., Yun, J., Hung, M. C., … Wan, X. (2022). Development and validation of a radiopathomics model to predict pathological complete response to neoadjuvant chemoradiotherapy in locally advanced rectal cancer: a multicentre observational study. *The Lancet. Digital health*, *4*(1), e8–e17. https://doi.org/10.1016/S2589-7500(21)00215-6

[28] Jayaprakasam, V. S., Paroder, V., Gibbs, P., Bajwa, R., Gangai, N., Sosa, R. E., Petkovska, I., Golia Pernicka, J. S., Fuqua, J. L., 3rd, Bates, D. D. B., Weiser, M. R., Cercek, A., & Gollub, M. J. (2022). MRI radiomics features of mesorectal fat can predict response to neoadjuvant chemoradiation therapy and tumor recurrence in patients with locally advanced rectal cancer. *European radiology*, *32*(2), 971–980. https://doi.org/10.1007/s00330-021-08144-w

[29] Shin, J., Seo, N., Baek, S. E., Son, N. H., Lim, J. S., Kim, N. K., Koom, W. S., & Kim, S. (2022). MRI Radiomics Model Predicts Pathologic Complete Response of Rectal Cancer Following Chemoradiotherapy. *Radiology*, *303*(2), 351–358. https://doi.org/10.1148/radiol.211986

[30] Sycai Medical. (2025, April 24). *Solutions - Sycai Medical*. Sycai Medical. https://sycaimedical.com/solutions/ Last Accessed: April 30, 2025

[31] SimBioSys. (2024). *SimBioSys – Redefine precision medicine*. https://simbiosys.com/ Last Accessed: April 30, 2025

[32] aidence. (n.d.). Veye Lung Nodules. Aidence. https://www.aidence.com/veye-lung-nodules/ Last Accessed: April 30, 2025

[33] Enlitic. (2025, March 7). *Radiologist Workflow Improved With AI*. Enlitic. https://enlitic.com/radiology/ Last Accessed: April 30, 2025

[34] A. Jacobs, M., & S. Parekh, V. (n.d.). Informatics radiomics integration system (IRIS): a novel combined informatics and radiomics method for integration of many types of data for classification into different groups for improved visualization.

[35] Yip, S. (2022). 3D radiomic platform for imaging biomarker development.

[36] Pla director d'oncologia. Direcció General de Planificació i Recerca en Salut. (2023). *Pla contra el càncer de Catalunya*. https://scientiasalut.gencat.cat/bitstream/handle/11351/9855/pla_contra_cancer_catalunya_2022_2026_ca.pdf?sequence=13&isAllowed=y Last Accessed: April 22, 2025

[37] Kankanala VL, Zubair M, Mukkamalla SKR. Carcinoembryonic Antigen. [Updated 2024 Dec 11]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2025 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK578172/ Last Accessed: May 15, 2025

[38] Boland, C. R., & Goel, A. (2010). Microsatellite instability in colorectal cancer. *Gastroenterology*, *138*(6), 2073–2087.e3. https://doi.org/10.1053/j.gastro.2009.12.064 Last Accessed: May 15, 2025

[39] Mayo Clinic. (2022, March 18). *Statins: Are these cholesterol-lowering drugs right for you?* Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/high-blood-cholesterol/in-depth/statins/art-20045772 Last Accessed: May 15, 2025

[40] Rong, Y., Rosu-Bubulac, M., Benedict, S. H., Cui, Y., Ruo, R., Connell, T., Kashani, R., Latifi, K., Chen, Q., Geng, H., Sohn, J., & Xiao, Y. (2021). Rigid and Deformable Image Registration for Radiation Therapy: A Self-Study Evaluation Guide for NRG Oncology Clinical Trial Participation. *Practical radiation oncology*, *11*(4), 282–298. https://doi.org/10.1016/j.prro.2021.02.007

[41] Ranjbarzadeh, R., Dorosti, S., Jafarzadeh Ghoushchi, S., Caputo, A., Tirkolaee, E. B., Ali, S. S., Arshadi, Z., & Bendechache, M. (2023). Breast tumor localization and segmentation using machine learning techniques: Overview of datasets, findings, and methods. *Computers in Biology and Medicine*, *152*, 106443. https://doi.org/10.1016/j.compbiomed.2022.106443

[42] Mohan, G., & Subashini, M. M. (2018). MRI based medical image analysis: Survey on brain tumor grade classification. *Biomedical Signal Processing and Control*, *39*, 139–161. https://doi.org/10.1016/j.bspc.2017.07.007

[43] Alnazer, I., Bourdon, P., Urruty, T., Falou, O., Khalil, M., Shahin, A., & Fernandez-Maloigne, C. (2021). Recent advances in medical image processing for the evaluation of chronic kidney disease. *Medical Image Analysis*, *69*, 101960. https://doi.org/10.1016/j.media.2021.101960

[44] Soomro, M. H., Coppotelli, M., Conforto, S., Schmid, M., Giunta, G., Del Secco, L., Neri, E., Caruso, D., Rengo, M., & Laghi, A. (2019). Automated Segmentation of Colorectal Tumor in 3D MRI Using 3D Multiscale Densely Connected Convolutional Neural Network. *Journal of healthcare engineering*, *2019*, 1075434. https://doi.org/10.1155/2019/1075434

[45] DeSilvio, T., Antunes, J. T., Bera, K., Prathyush Chirra, Le, H., Liska, D., Stein, S. L., Marderstein, E., Hall, W., Rajmohan Paspulati, Jayakrishna Gollamudi, Purysko, A. S., & Viswanath, S. E. (2023). Region-specific deep learning models for accurate segmentation of rectal structures on post-chemoradiation T2w MRI: a multi-institutional, multi-reader study. *Frontiers in Medicine*, *10*. https://doi.org/10.3389/fmed.2023.1149056

[46] Zhu, H., Zhang, X., Shi, Y., Li, X., & Sun, Y. (2021). Automatic segmentation of rectal tumor on diffusion-weighted images by deep learning with U-Net. *Journal of Applied Clinical Medical Physics*, *22*(9), 324–331. https://doi.org/10.1002/acm2.13381

[47] Slicer. (2019). *3D Slicer*. Slicer.org. https://www.slicer.org/ Last Accessed: June 5, 2025

[48] Fedorov A., Beichel R., Kalpathy-Cramer J., Finet J., Fillion-Robin J-C., Pujol S., Bauer C., Jennings D., Fennessy F.M., Sonka M., Buatti J., Aylward S.R., Miller J.V., Pieper S., Kikinis R. 3D Slicer as an

Image Computing Platform for the Quantitative Imaging Network. Magnetic Resonance Imaging. 2012 Nov;30(9):1323-41. PMID: 22770690. PMCID: PMC3466397.

[49] 3D Slicer. (n.d.). *Welcome to 3D Slicer's documentation! — 3D Slicer documentation*. Slicer.readthedocs.io. https://slicer.readthedocs.io/en/latest/ Last Accessed: June 5, 2025

[50] Siemens Healthineers. (2025). *syngo.via*. Siemens-Healthineers.com. https://www.siemens-healthineers.com/es/digital-health-solutions/syngovia#simplifying-routine Last Accessed: May 20, 2025

[51] scikit-learn. (2019). *StandardScaler*. Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html Last Accessed: June 8, 2025

[52] Zwanenburg, A., Vallières, M., Abdalah, M. A., Aerts, H. J. W. L., Andrearczyk, V., Apte, A., Ashrafinia, S., Bakas, S., Beukinga, R. J., Boellaard, R., Bogowicz, M., Boldrini, L., Buvat, I., Cook, G. J. R., Davatzikos, C., Depeursinge, A., Desseroit, M. C., Dinapoli, N., Dinh, C. V., Echegaray, S., … Löck, S. (2020). The Image Biomarker Standardization Initiative: Standardized Quantitative Radiomics for High-Throughput Image-based Phenotyping. *Radiology*, *295*(2), 328–338. https://doi.org/10.1148/radiol.2020191145

[53] Cheng, X. (2025). A Comprehensive Study of Feature Selection Techniques in Machine Learning Models. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.5154947

[54] Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., & Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, *143*, 106839. https://doi.org/10.1016/j.csda.2019.106839

[55] Mukaka M. M. (2012). Statistics corner: A guide to appropriate use of correlation coefficient in medical research. *Malawi medical journal: the journal of Medical Association of Malawi*, *24*(3), 69–71.

[56] Alon Orlitsky. (2003). Information Theory. *Elsevier EBooks*, 751–769. https://doi.org/10.1016/b0-12-227410-5/00337-9

[57] IBM. (2022, October 4). *Background of parametric and nonparametric statistics*. Ibm.com. https://www.ibm.com/docs/en/ias?topic=nonparametric-background Last Accessed: June 6, 2025

[58] Teradata. (2025). Rank Tests: Mann-Whitney/Kruskal-Wallis Test. Teradata.com. https://docs.teradata.com/r/Teradata-Warehouse-Miner-User-Guide-Volume-3Analytic-Functions/July-2017/Statistical-Tests-Teradata-Only/Rank-Tests Last Accessed: June 6, 2025

[59] Greenacre, M., Groenen, P.J.F., Hastie, T. *et al.* Principal component analysis. *Nat Rev Methods Primers* **2**, 100 (2022). https://doi.org/10.1038/s43586-022-00184-w

[60] Anowar, F., Sadaoui, S., & Selim, B. (2021). Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). *Computer Science Review*, *40*, 100378. https://doi.org/10.1016/j.cosrev.2021.100378

[61] Ahmed, S. (2025, January 31). *Principal Component Analysis (PCA) Made Easy: A Complete Hands-On Guide*. Medium. https://medium.com/@sahin.samia/principal-component-analysis-pca-made-easy-a-complete-hands-on-guide-e26a3680c0bc Last Accessed: June 4, 2025

[62] Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation Coefficients: Appropriate Use and Interpretation. *Anesthesia & Analgesia*, *126*(5), 1763–1768. https://doi.org/10.1213/ANE.0000000000002864

[63] Jiang, J., Wei, J., Zhu, Y. *et al.* Clot-based radiomics model for cardioembolic stroke prediction with CT imaging before recanalization: a multicenter study. *Eur Radiol* **33**, 970–980 (2023). https://doi.org/10.1007/s00330-022-09116-4

[64] Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN COMPUT. SCI.* **2**, 160 (2021). https://doi.org/10.1007/s42979-021-00592-x

[65] Ye, J., Dobson, S., & McKeever, S. (2012). Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing*, *8*(1), 36–66. https://doi.org/10.1016/j.pmcj.2011.01.004

[66] GeeksForGeeks. (2020, August 25). *ML - Gradient Boosting*. GeeksforGeeks. https://www.geeksforgeeks.org/ml-gradient-boosting/ Last Accessed: June 4, 2025

[67] *LightGBM 3.3.5 documentation*. (n.d.). Lightgbm.readthedocs.io. https://lightgbm.readthedocs.io/en/stable/ Last Accessed: June 4, 2025

[68] GeeksForGeeks. (2024, March 4). *Backpropagation in Neural Network*. GeeksforGeeks. https://www.geeksforgeeks.org/backpropagation-in-neural-network/ Last Accessed: June 4, 2025

[69] Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, *14*(1), 1–14. Nature. https://doi.org/10.1038/s41598-024-56706-x

[70] Nahm F. S. (2022). Receiver operating characteristic curve: overview and practical use for clinicians. *Korean journal of anesthesiology*, *75*(1), 25–36. https://doi.org/10.4097/kja.21209

[71] Elgeldawi, E., Sayed, A., Galal, A. R., & Zaki, A. M. (2021). Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. *Informatics*, *8*(4), 79. https://doi.org/10.3390/informatics8040079

[72] Soni, P. (2024, September 5). *Confusion Matrix, Precision, and Recall - Train in Data's Blog*. Train in Data's Blog. https://www.blog.trainindata.com/confusion-matrix-precision-and-recall/ Last Accessed: May 20, 2025

[73] Madrigal, M. (2024). *Demystifying ROC Curves: Understanding Performance Metrics for AI Classification Models*. Ridgerun.ai. https://www.ridgerun.ai/post/demystifying-roc-curves-understanding-performance-metrics-for-ai-classification-models Last Accessed: May 20, 2025

[74] Softneta. (2019). *DICOM Library - About DICOM format*. Dicomlibrary.com. https://www.dicomlibrary.com/dicom/ Last Accessed: May 20, 2025

[75] *SimpleITK - Home*. (n.d.). Simpleitk.org. https://simpleitk.org/ Last Accessed: May 20, 2025

[76] PyRadiomics. (2023b). *Pipeline Modules — pyradiomics v3.1.0rc2.post5+g6a761c4 documentation*. Readthedocs.io. https://pyradiomics.readthedocs.io/en/latest/radiomics.html Last Accessed: June 7, 2025

[77] Mandard, A. M., Dalibard, F., Mandard, J. C., Marnay, J., Henry-Amar, M., Petiot, J. F., Roussel, A., Jacob, J. H., Segol, P., & Samama, G. (1994). Pathologic assessment of tumor regression after preoperative chemoradiotherapy of esophageal carcinoma. Clinicopathologic correlations. *Cancer*, *73*(11), 2680–2686. https://doi.org/10.1002/1097-0142(19940601)73:11<2680::aid-cncr2820731105>3.0.co;2-c

[78] The SciPy community. (2019). *Statistical functions (scipy.stats) — SciPy v1.3.3 reference guide*. Scipy.org. https://docs.scipy.org/doc/scipy/reference/stats.html Last Accessed: June 7, 2025

[79] scikit-learn. (2018). *sklearn.model_selection.train_test_split — scikit-learn 0.20.3 documentation*. Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html Last Accessed: June 2, 2025

[80] scikit-learn. (n.d.). *sklearn.model_selection*. Scikit-Learn. https://scikit-learn.org/stable/api/sklearn.model_selection.html Last Accessed: June 3, 2025

[81] scikit-learn. (2019a). *sklearn.preprocessing.MinMaxScaler — scikit-learn 0.22.1 documentation*. Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html Last Accessed: June 2, 2025

[82] Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, *1168*(2), 022022. https://doi.org/10.1088/1742-6596/1168/2/022022

[83] Mary M., C., H.S., J., & S., A. (2025). Explainable Optimal Random Forest model with conversational interface. *Engineering Applications of Artificial Intelligence*, *145*, 110134. https://doi.org/10.1016/j.engappai.2025.110134

[84] Boubaddi, M., Fleming, C., Assenat, V. *et al.* Tumor response rates based on initial TNM stage and tumor size in locally advanced rectal cancer: a useful tool for shared decision-making. *Tech Coloproctol* **28**, 122 (2024). https://doi.org/10.1007/s10151-024-02993-5

[85] Tan, Y., Fu, D., Li, D., Kong, X., Jiang, K., Chen, L., Yuan, Y., & Ding, K. (2019). Predictors and Risk Factors of Pathologic Complete Response Following Neoadjuvant Chemoradiotherapy for Rectal Cancer: A Population-Based Analysis. *Frontiers in oncology*, *9*, 497. https://doi.org/10.3389/fonc.2019.00497

[86] Hammarström, K., Imam, I., Mezheyeuski, A., Ekström, J., Sjöblom, T., & Glimelius, B. (2020). A Comprehensive Evaluation of Associations Between Routinely Collected Staging Information and The Response to (Chemo)Radiotherapy in Rectal Cancer. *Cancers*, *13*(1), 16. https://doi.org/10.3390/cancers13010016

[87] Tanaka, H., Fukuda, S., Kimura, K., Fukawa, Y., Yamamoto, K., Fukushima, H., Moriyama, S., Yasuda, Y., Uehara, S., Waseda, Y., Yoshida, S., Yokoyama, M., Matsuoka, Y., Saito, K., Tateishi, U., Campbell, S. C., & Fujii, Y. (2022). Defining Tumour Shape Irregularity for Preoperative Risk Stratification of Clinically Localised Renal Cell Carcinoma. *European urology open science*, *48*, 36–43. https://doi.org/10.1016/j.euros.2022.12.003

[88] Huang, R. Y., Unadkat, P., Bi, W. L., George, E., Preusser, M., McCracken, J. D., Keen, J. R., Read, W. L., Olson, J. J., Seystahl, K., Le Rhun, E., Roelcke, U., Koeppen, S., Furtner, J., Weller, M., Raizer, J. J., Schiff, D., & Wen, P. Y. (2019). Response assessment of meningioma: 1D, 2D, and volumetric criteria for treatment response and tumor progression. *Neuro-oncology*, *21*(2), 234–241. https://doi.org/10.1093/neuonc/noy126

[89] Hernández-Yagüe, X., López-Ben, S., Martínez-Sancho, J., Ortíz-Durán, M. R., Casellas-Robert, M., Aula-Olivar, A., Meléndez-Muñoz, C., Pujolràs, M. B., Queralt-Merino, B., & Felip, J. F. i. (2025). The Prognostic Value of Tumor Fibrosis in Patients Undergoing Hepatic Metastasectomy for Colorectal Cancer: A Retrospective Pooled Analysis. *Cancers*, *17*(11), 1870. https://doi.org/10.3390/cancers17111870

[90] Korsavidou Hult, N., Tarai, S., Hammarström, K. *et al.* Inclusion of tumor periphery in radiomics analysis of magnetic resonance images does not improve predictions of preoperative therapy response in patients with rectal cancer. *Abdom Radiol* (2025). https://doi.org/10.1007/s00261-025-04815-0

[91] Vimbi, V., Shaffi, N., & Mahmud, M. (2024). Interpreting artificial intelligence models: a systematic review on the application of LIME and SHAP in Alzheimer's disease detection. *Brain informatics*, *11*(1), 10. https://doi.org/10.1186/s40708-024-00222-1

[92] Abid Ali Awan. (2023, June 28). *An Introduction to SHAP Values and Machine Learning Interpretability*. Datacamp.com; DataCamp. https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability Last Accessed: June 4, 2025

[93] Ward, W. H., Sigurdson, E. R., Esposito, A. C., Ruth, K. J., Manstein, S. M., Sorenson, E. C., Wernick, B. D., & Farma, J. M. (2018). Pathologic response following treatment for locally advanced rectal cancer: Does location matter?. *The Journal of surgical research*, *224*, 215–221. https://doi.org/10.1016/j.jss.2017.11.072

[94] Yilmaz, S., Liska, D., Conces, M. L., Tursun, N., Elamin, D., Ozgur, I., Maspero, M., Rosen, D. R., Khorana, A. A., Balagamwala, E. H., Amarnath, S. R., Valente, M. A., Steele, S. R., Krishnamurthi, S. S., & Gorgun, E. (2025). What Predicts Complete Response to Total Neoadjuvant Therapy in Locally Advanced Rectal Cancer?. *Diseases of the colon and rectum*, *68*(1), 60–68. https://doi.org/10.1097/DCR.0000000000003395

[95] Peng, H., Wang, C., Xiao, W., Lin, X., You, K., Dong, J., Wang, Z., Yu, X., Zeng, Z., Zhou, T., Gao, Y., & Wen, B. (2018). Analysis of Clinical characteristics to predict pathologic complete response for patients with locally advanced rectal cancer treated with neoadjuvant chemoradiotherapy. *Journal of Cancer*, *9*(15), 2687–2692. https://doi.org/10.7150/jca.25493

[96] Aysen Çeliktaş. (2023, September 12). *GANTT, CPM, PERT in Project Management - Aysen Çeliktaş - Medium*. Medium; Medium. https://medium.com/@aysenceliktas/gantt-cpm-pert-in-project-management-f381ed0363cc Last Accessed: April 8, 2025

[97] España. (2018). *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales*. *Boletín Oficial del Estado*, núm. 294, de 6 de diciembre de 2018.

[98] European Union. (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 (General Data Protection Regulation)*. *Official Journal of the European Union*, L119, 1–88. https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng

[99] European Parliament, & Council of the European Union. (2024). *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts*. Official Journal of the European Union, L 168, 1–161. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689#enc_1

[100] European Commission. (2025, February 18). *AI Act*. European Commission. https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai Last Accessed: April 11, 2025

[101] Kotter, E., D'Antonoli, T.A., Cuocolo, R. *et al.* Guiding AI in radiology: ESR's recommendations for effective implementation of the European AI Act. *Insights Imaging* **16**, 33 (2025). https://doi.org/10.1186/s13244-025-01905-x

[102] Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EEC (Text with EEA relevance.)' (2017) *Official Journal* L 117, 1-175. http://data.europa.eu/eli/reg/2017/745/oj

[103] Boletín Oficial del Estado. (2011). *BOE-A-2007-12945 Ley 14/2007, de 3 de julio, de Investigación biomédica.* Www.boe.es. https://www.boe.es/buscar/doc.php?id=BOE-A-2007-12945

Annex 1. Tumour staging and restaging systems.

TNM staging:

Primary Tumour (T):

Tx Cannot assess primary tumour

T0 No evidence of primary tumour

T1 Involves the mucosa without extending beyond it

T2 Involves the muscularis propria without extending beyond it

T3 Infiltrates the perirectal fat

- T3a Invasion < 1 mm
- T3b Invasion 1-5 mm
- T3c Invasion 6-15 mm
- T3d Invasion > 15 mm

T4 Tumour invades directly into other organs or structures and/or perforates visceral peritoneum

- T4a Perforation of the visceral peritoneum
- T4b Invasion of adjacent organs

Regional Lymph nodes (N):

N0 No perirectal lymph node involvement

N1 1–3 lymph nodes involved (>5 mm, heterogeneous signal, irregular borders)

- N1a 1 lymph node
- N1b 2-3 lymph nodes
- N1c Tumour deposits in the perirectal fat without nodal structures

N2 4 or more lymph nodes involved (>5 mm, heterogeneous signal, irregular borders)

- N2a 4-6 lymph nodes
- N2b 7 or more lymph nodes

Distant Metastases (M):

M0 No distant metastases

M1 Presence of distant metastases:

- M1a Metastases in a single distant organ or group of lymph nodes
- M1b Metastases in more than one organ or in the peritoneum

Mandard's Tumour Regression Grade (TRG):

TRG 1 – Complete response (no visible tumour)

TRG 2 – Good response (dense fibrosis, minimal residual tumour)

TRG 3 – Moderate response (>50% fibrosis or minimal mucin with intermediate signal)

TRG 4 – Minimal response (minimal fibrosis with mostly intermediate signal)

TRG 5 – No response (intermediate signal, similar to the original tumour)

Annex 2. Normalization and radiomic feature extraction code

Core Tumour radiomic features extraction code:

```
# Extract radiomic features from normalized MRI images based on their segmentation
masks

# Import necessary libraries
import os
import SimpleITK as sitk
import pandas as pd
from radiomics import featureextractor
from sklearn.preprocessing import StandardScaler

# Define the paths for the mri folder and the segmentation folder and the output
.csv file
mri_folder = " C:\Users\Usuari\Desktop\TFG\Radiomics\MRI"
seg_folder = " C:\Users\Usuari\Desktop\TFG\Radiomics\Segmentations"
output_csv = "C:\Users\Usuari\Desktop\TFG\Radiomics\radiomics_features_all.csv"

# Define a function to normalize images using StandardScaler
def normalize_image(image):
    array = sitk.GetArrayFromImage(image) # Convert image to NumPy array to be used
to normalize

    # Use StandardScaler to normalize
    scaler = StandardScaler() # Create the scaler
    array = scaler.fit_transform(array.reshape(-1, 1)).reshape(array.shape) # Fit
it to the image and transform

    # Create the normalized image and copy metadata
    normalized_image = sitk.GetImageFromArray(array)
    normalized_image.CopyInformation(image)
    return normalized_image

# Define a function to extract radiomic features from an image and its segmentation
def extract_radiomics_features(image_path, mask_path, extractor):

    # Read the image and the mask
    image = sitk.ReadImage(image_path)
    mask = sitk.ReadImage(mask_path)

    # Normalize the image and extract the radiomic features
    image = normalize_image(image)
    features = extractor.execute(image, mask)

    # Create a dictionary to store the features
    feature_dict = {"ID": os.path.basename(image_path).replace('.nii.gz', '')} #
First column is the patient's ID
    feature_dict.update({k: v for k, v in features.items()}) # Add the rest of
features to the dictionary
    return feature_dict

# List all MRI and segmentation files
mri_files = [os.path.join(mri_folder, f) for f in os.listdir(mri_folder) if
f.endswith(".nii.gz")]
```

```
seg_files =[os.path.join(seg_folder, f) for f in os.listdir(seg_folder) if
f.endswith(".nrrd")]

# Debug
print("MRI Files:", mri_files)
print("Segmentation Files:", seg_files)

# Initialize the radiomics feature extractor
extractor = featureextractor.RadiomicsFeatureExtractor()

# Geometry tolerance to handle minor mismatches (Image/Mask mismatch error)
extractor.settings['geometryTolerance'] = 1e-2

# Debug
print("Enabled features:", extractor.enabledFeatures)

# List to store the results
results = []

# Loop all through all the images
for mri_file in mri_files:
    base_id = os.path.basename(mri_file).replace('.nii.gz', '.nrrd') if
mri_file.endswith('.nii.gz') else os.path.splitext(os.path.basename(mri_file))[0] #
Get the patient ID
    print("File name: ", base_id) # Debug
    matching_seg_file = next((f for f in seg_files if base_id in f), None) # Check
if there is a matching segmentation

    if matching_seg_file:

        # Create the mri and segmentation paths
        mri_path = os.path.join(mri_folder, mri_file)
        seg_path = os.path.join(seg_folder, matching_seg_file)
        try:
            print(f"Processing {mri_file} and {matching_seg_file}")
            feature_dict = extract_radiomics_features(mri_path, seg_path,
extractor) # Extract features
            results.append(feature_dict) # Append features to results
        except Exception as e:
            print(f"Error processing {mri_file} and {matching_seg_file}: {e}")
    else:
        print(f"No matching segmentation found for {mri_file}") # Debug

# Save the features to CSV
if results:
    df = pd.DataFrame(results) # Create a dataframe of results
    df = df.sort_values(by='ID', ascending=True) # Sort results by the 'Id' column
    df.to_csv(output_csv, index=False, float_format="%.6f") # Convert it to .csv
and store it
    print(f"Radiomic features saved to {output_csv}")
else:
    print("No features extracted.")
```

Tumour Borders radiomic features extraction code:

```
# Extract radiomic features from the border regions of segmentation masks
```

```python
# Import the necessary libraries
import os
import SimpleITK as sitk
import pandas as pd
from radiomics import featureextractor
from sklearn.preprocessing import StandardScaler

# Define the paths for the mri folder, the segmentation folder and the output .csv
mri_folder = " C:\Users\Usuari\Desktop\TFG\Radiomics\MRI"
seg_folder = " C:\Users\Usuari\Desktop\TFG\Radiomics\Segmentations"
output_csv = "C:\Users\Usuari\Desktop\TFG\Radiomics\radiomics_border_features.csv"

# Define a function to normalize images using StandardScaler
def normalize_image(image):
    array = sitk.GetArrayFromImage(image) # Convert image to NumPy array to be used to normalize

    # Use StandardScaler to normalize
    scaler = StandardScaler() # Create the scaler
    array = scaler.fit_transform(array.reshape(-1, 1)).reshape(array.shape) # Fit it to the image and transform

    # Create the normalized image and copy metadata
    normalized_image = sitk.GetImageFromArray(array)
    normalized_image.CopyInformation(image)
    return normalized_image

# Define a function to compute a mask of the borders of the segmentation
def get_border_mask(mask, radius_mm=2.0):

    # Convert radius in mm to radius in voxels
    spacing = mask.GetSpacing() # Get image spacing (voxel size in mm)
    radius_voxels = [int(radius_mm / s + 0.5) for s in spacing]

    # Perform a dilation and an erosion to obtain the borders
    dilated = sitk.BinaryDilate(mask, radius_voxels)
    eroded = sitk.BinaryErode(mask, radius_voxels)

    # Subtract the previous operations to obtian the borders of the segmentation
    border = sitk.Subtract(dilated, eroded)
    return border

# Define a function to extract radiomic features from an image and its segmentation
def extract_radiomics_features(image_path, mask, extractor):

    # Read the image
    image = sitk.ReadImage(image_path)

    # Normalize the image and extract the radiomic features
    image = normalize_image(image)
    features = extractor.execute(image, mask)

    # Create a dictionary to store the features
    feature_dict = {"ID": os.path.basename(image_path).replace('.nii.gz', '')} # First column is the patient's ID
    feature_dict.update({k: v for k, v in features.items()}) # Add the rest of features to the dictionary
```

```
    return feature_dict

# Initialize the radiomics extractor
extractor = featureextractor.RadiomicsFeatureExtractor()
extractor.settings['geometryTolerance'] = 1e-2

# Debug
print("Enabled features:", extractor.enabledFeatures)

# List to store the results
results = []

# List all MRI and segmentation files
mri_files = [os.path.join(mri_folder, f) for f in os.listdir(mri_folder) if
f.endswith(".nii.gz")]
seg_files =[os.path.join(seg_folder, f) for f in os.listdir(seg_folder) if
f.endswith(".nrrd")]

# Loop all through all the images
for mri_file in mri_files:
    base_id = os.path.basename(mri_file).replace('.nii.gz', '.nrrd') if
mri_file.endswith('.nii.gz') else os.path.splitext(os.path.basename(mri_file))[0] #
Get the patient ID
    print("File name: ", base_id) # Debug
    matching_seg_file = next((f for f in seg_files if base_id in f), None)  # Check
if there is a matching segmentation

    if matching_seg_file:

        # Create the mri and segmentation paths
        mri_path = os.path.join(mri_folder, mri_file)
        seg_path = os.path.join(seg_folder, matching_seg_file)
        try:
            print(f"Processing {mri_file} and {matching_seg_file}")

            # Load segmentation mask and compute borders
            mask = sitk.ReadImage(seg_path)
            mask = sitk.Cast(mask > 0, sitk.sitkUInt8)  # Ensure mask is binary
            border_mask = get_border_mask(mask)

            feature_dict = extract_radiomics_features(mri_path, border_mask,
extractor) # Extract radiomic features
            results.append(feature_dict) # Store them in the results list
        except Exception as e:
            print(f"Error processing {mri_file} and {matching_seg_file}: {e}")
    else:
        print(f"No matching segmentation found for {mri_file}") # Debug

# Save the features to CSV
if results:
    df = pd.DataFrame(results) # Create a dataframe of results
    df = df.sort_values(by='ID', ascending=True) # Sort results by the 'Id' column
    df.to_csv(output_csv, index=False, float_format="%.6f") # Convert it to .csv
and store it
    print(f"Radiomic features saved to {output_csv}")
else:
    print("No features extracted.")
```

Annex 3. Feature selection and model training code

Feature selection code (for whole tumour radiomics, equivalent for tumour borders):

```
# Eliminate non-significant features (p-value > 0.5) and highly correlated features
(correlation > 0.8) (whole tumour)

# Import the necessary libraries
import pandas as pd
import numpy as np
from scipy.stats import spearmanr, kruskal
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
import matplotlib.pyplot as plt

# Get data from a .csv file
file_path = "C:\Users\Usuari\Desktop\TFG\Radiomics\MRI\radiomics_no_borders.csv"
data = pd.read_csv(file_path, header='infer', sep=';')
data_og_rad = data.copy()

# Identify radiomic and non-radiomic columns
radiomics_start_col = 'original_shape_Elongation'
radiomics_columns = data.loc[:, radiomics_start_col:].columns
non_radiomics_columns = ['response', 'ID']

# Normalize radiomic variables
scaler = MinMaxScaler()
data[radiomics_columns] = scaler.fit_transform(data[radiomics_columns])

# Kruskal-Wallis to select relevant features (response as output)
output_col = 'response'
kruskal_results = {}
response_groups = [data[data[output_col] == group][radiomics_columns] for group in
data[output_col].unique()]

for col in radiomics_columns:
    group_values = [group[col].dropna() for group in response_groups]
    if all(len(values) > 0 for values in group_values):
        stat, p_value = kruskal(*group_values)
        kruskal_results[col] = p_value

# Filter relevant features (p < 0.05)
selected_features = [feature for feature, p_val in kruskal_results.items() if p_val
< 0.05]
print('Number of significant features: ', len(selected_features))
print('Significant features: ', selected_features)

# Correlation matrix to remove colinearity
correlation_matrix = data[selected_features].corr(method='spearman')
variables_to_remove = set()

for i in range(len(selected_features)):
    for j in range(i + 1, len(selected_features)):
        feature1, feature2 = selected_features[i], selected_features[j]
        if abs(correlation_matrix.loc[feature1, feature2]) > 0.8:
            p1, p2 = kruskal_results[feature1], kruskal_results[feature2]
```

```
            variables_to_remove.add(feature1 if p1 > p2 else feature2)

# Final features
final_features = [feature for feature in selected_features if feature not in
variables_to_remove]
print("Number of final selected features after removing collinearity: ",
len(final_features))
print("Final selected features after removing collinearity: \n", final_features)

# Save the selected radiomics variables and the non radiomics ones to a .csv file
output_data = data_og_rad[non_radiomics_columns + final_features]
output_file_path =
"C:\Users\Usuari\Desktop\TFG\Radiomics\MRI\selected_radiomics_features_no_borders.c
sv"
output_data.to_csv(output_file_path, index=False)

print(f"File with the selected features saved in: {output_file_path}")
```

Hyperparameter tuning code (for mixed clinical datasets, equivalent for the other feature sets):

```
# Hyperparameter tuning

# Import the necessary libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, StratifiedKFold, GridSearchCV
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.pipeline import Pipeline
from sklearn.feature_selection import SelectKBest, f_classif
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from lightgbm import LGBMClassifier
from sklearn.base import BaseEstimator, TransformerMixin

# Define a function to plot the confusion matrix of the model
def plot_confusion_matrix(y_true, y_pred, model_name, dataset_name):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(5, 4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=np.unique(y_true), yticklabels=np.unique(y_true))
    plt.xlabel('Predicted Label')
    plt.ylabel('True Label')
    plt.title(f'Confusion Matrix: {model_name} ({dataset_name})')
    plt.show()

# Define a class to perform intra fold scaling of only some columns
class SelectiveStandardScaler(BaseEstimator, TransformerMixin):
    def __init__(self, start_index=-3):
        self.start_index = start_index
```

```
        self.scaler = StandardScaler()

    def fit(self, X, y=None):
        self.scaler.fit(X[:, self.start_index:])
        return self

    def transform(self, X):
        X = X.copy()
        X[:, self.start_index:] = self.scaler.transform(X[:, self.start_index:])
        return X

# Get the clinical data from the .csv file
file_path_clinic_whole =
"C:\Users\Usuari\Desktop\TFG\Radiomics\MRI\radiomics_clinical.csv"
file_path_clinic_borders =
"C:\Users\Usuari\Desktop\TFG\Radiomics\MRI\borders_clinical.csv"
file_path_clinic_whole_borders =
"C:\Users\Usuari\Desktop\TFG\Radiomics\MRI\radiomics_borders_clinical.csv"

data_whole = pd.read_csv(file_path_clinic_whole)
data_borders = pd.read_csv(file_path_clinic_borders)
data_whole_borders = pd.read_csv(file_path_clinic_whole_borders)

# Drop the ID column
columns_to_remove = ['ID']
data_whole = data_whole.drop(columns=columns_to_remove, errors='ignore')
data_borders = data_borders.drop(columns=columns_to_remove, errors='ignore')
data_whole_borders = data_whole_borders.drop(columns=columns_to_remove,
errors='ignore')

# Separate output column and features
X_whole = data_whole.drop(columns=['response']).values
y_whole = data_whole['response'].values

X_borders = data_borders.drop(columns=['response']).values
y_borders = data_borders['response'].values

X_whole_borders = data_whole_borders.drop(columns=['response']).values
y_whole_borders = data_whole_borders['response'].values

X_whole[:,-11:] = X_whole[:,-11:].astype(int) # Convert the last 11 columns to
integers (debug)
X_borders[:,-7:] = X_borders[:,-7:].astype(int) # Convert the last 7 columns to
integers (debug)
X_whole_borders[:,-15:] = X_whole_borders[:,-15:].astype(int) # Convert the last 15
columns to integers (debug)

# Reserve a fixed validation set (15%)
X_train_whole, X_val_whole,  y_train_whole, y_val_whole = train_test_split(X_whole,
y_whole, test_size=0.15, stratify=y_whole, random_state=42)
X_train_borders, X_val_borders,  y_train_borders, y_val_borders =
train_test_split(X_borders, y_borders, test_size=0.15, stratify=y_borders,
random_state=42)
X_train_whole_borders, X_val_whole_borders, y_train_whole_borders,
y_val_whole_borders = train_test_split(X_whole_borders, y_whole_borders,
test_size=0.15, stratify=y_whole_borders, random_state=42)
```

```
# Define the cross-validation strategy
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Define models and hyperparameter grids to test
model_configs = {
    "Random Forest": {
        "model": RandomForestClassifier(class_weight='balanced', random_state=42),
        "params": {
            "model__n_estimators": [50, 100, 150, 200, 300],
            "model__max_depth": [3, 5, 10, 15, 20, None],
            "model__min_samples_split": [1, 2, 4, 5, 10]
        }
    },
    "Logistic Regression": {
        "model": LogisticRegression(max_iter=1000, random_state=42),
        "params": {
            "model__C": [0.001, 0.01, 0.1, 1, 10],
            "model__penalty": ["l2"],
            "model__solver": ["lbfgs", "saga"]
        }
    },
    "Support Vector Machine": {
        "model": SVC(class_weight='balanced', random_state=42),
        "params": {
            "model__kernel": ['linear','poly','rbf'],
            "model__C": [0.1, 1, 10, 50, 100, 500],
            "model__gamma": ['scale', 'auto']
        }
    },
    "XGBoost": {
        "model": XGBClassifier(use_label_encoder=False, eval_metric='logloss',
random_state=42),
        "params": {
            "model__n_estimators": [50, 100, 200],
            "model__max_depth": [3, 5, 7, 10, 20],
            "model__learning_rate": [0.01, 0.05, 0.1, 0.5],
            "model__subsample": [0.6, 0.8, 1.0],
            "model__colsample_bytree": [0.6, 0.8, 1.0]
        }
    },
    "Multilayer Perceptron": {
        "model": MLPClassifier(max_iter=1000, random_state=42),
        "params": {
            "model__hidden_layer_sizes": [(50,), (100,), (50, 50), (100, 50)],
            "model__activation": ["relu", "tanh"],
            "model__alpha": [0.0001, 0.001, 0.01],
            "model__learning_rate": ["constant", "adaptive"]
        }
    },
    "Decision Tree": {
        "model": DecisionTreeClassifier(class_weight='balanced', random_state=42),
        "params": {
            "model__max_depth": [3, 5, 10, None],
            "model__min_samples_split": [2, 5, 10],
            "model__min_samples_leaf": [1, 2, 4],
            "model__criterion": ["gini", "entropy"]
        }
```

```
    },
    "LightGBM": {
        "model": LGBMClassifier(random_state=42),
        "params": {
            "model__n_estimators": [100, 200],
            "model__num_leaves": [15, 31, 63],
            "model__learning_rate": [0.01, 0.05, 0.1],
            "model__boosting_type": ["gbdt", "dart"],
            "model__min_child_samples": [10, 20]
        }
    }
}

# Store a summary of results
summary = []

# Perform grid search for each model
for name, config in model_configs.items():
    for dataset, X_train, X_val, y_train, y_val, scale_columns in [
        ('Whole tumour', X_train_whole, X_val_whole, y_train_whole, y_val_whole,-
11),
        ('Tumour borders', X_train_borders, X_val_borders, y_train_borders,
y_val_borders,-7),
        ('Whole tumour + tumour borders', X_train_whole_borders,
X_val_whole_borders, y_train_whole_borders, y_val_whole_borders,-15)
    ]:
      print(f"\n {name} (5-Fold CV) for {dataset} + clinical features dataset")

      # Define a pipeline to scale and train
      pipeline = Pipeline([
          ('scaler', SelectiveStandardScaler(start_index=scale_columns)),  # Only
scale last x columns
          ('model', config["model"])
      ])

      # Test
      grid_search = GridSearchCV(
          estimator=pipeline,
          param_grid=config["params"],
          cv=cv,
          scoring='accuracy',
          n_jobs=-1,
          verbose=1
      )

      grid_search.fit(X_train, y_train)

      best_model = grid_search.best_estimator_
      print(f"Best Parameters for {name}: {grid_search.best_params_}")

      # Evaluate on final validation set
      y_val_pred = best_model.predict(X_val)
      acc_val = accuracy_score(y_val, y_val_pred)

      print(f"Validation Accuracy ({name}): {acc_val:.4f}")
      print("Classification Report:\n", classification_report(y_val, y_val_pred))
      plot_confusion_matrix(y_val, y_val_pred, name, dataset)
```

```
        # Save to summary
        summary.append({
            "Dataset": dataset,
            "Model": name,
            "Best Parameters": grid_search.best_params_,
            "Validation Accuracy": acc_val
        })

# Final Summary Table (results of hyperparameter tuning)
print("\nModel Comparison Summary:")
summary_df = pd.DataFrame(summary)
print(summary_df.to_string(index=False))
```

Optimized model training and metric computation code (for mixed clinical datasets, equivalent for the other feature sets):

```
# Train and test with the best hyperparameters

# Import the necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.pipeline import Pipeline
from sklearn.model_selection import StratifiedKFold, train_test_split
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Define a function to plot the confusion matrix of the model
def plot_confusion_matrix(y_true, y_pred, model_name, dataset_name):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(5, 4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=np.unique(y_true), yticklabels=np.unique(y_true))
    plt.xlabel('Predicted Label')
    plt.ylabel('True Label')
    plt.title(f'Confusion Matrix: {model_name} (Clinical features +
{dataset_name})', fontsize=10)
    plt.show()

# Define a function to plot the feature importance of the model
def plot_feature_importance(model, X_train, feature_names, model_name,
dataset_name):
    # Verify that the model has feature importance
    if hasattr(model, 'feature_importances_'):
        importance = model.feature_importances_
    elif hasattr(model, 'coef_'):  # For models Logistic Regression or SVM
```

```
        importance = np.abs(model.coef_[0])
    else:
        importance = None

    if importance is not None:
        sorted_idx = np.argsort(importance)[::-1]
        n_features = len(feature_names)  # Real number of features
        plt.figure(figsize=(10, 6))
        plt.barh(range(min(n_features, len(importance))),
importance[sorted_idx][:min(n_features, len(importance))])
        plt.yticks(range(min(n_features, len(importance))),
np.array(feature_names)[sorted_idx][:min(n_features, len(importance))])
        plt.xlabel('Feature Importance')
        plt.title(f'Feature Importance: {model_name} ({dataset_name})')
        plt.show()

# Define a function to plot the feature importance of the model using the SHAP
tecnique
def plot_shap_summary(model, X_train, feature_names, model_name, dataset_name):
    try:
        if model_name in ["Decision Tree", "Random Forest"]:
            explainer = shap.TreeExplainer(model)
            shap_values = explainer.shap_values(X_train)

            # For binary classification
            plt.figure(figsize=(10, 6))
            shap.summary_plot(shap_values[:, :, 1], X_train,
feature_names=feature_names, show=False)

            # Plot
            plt.title(f'SHAP Feature Importance: {model_name} (Clinical features +
{dataset_name})', fontsize=12)
            plt.tight_layout()
            plt.subplots_adjust(top=0.88)
            plt.show()

        elif model_name in ["Gradient Boosting", "XGBoost", "LightGBM"]:
            explainer = shap.TreeExplainer(model)
            shap_values = explainer.shap_values(X_train)

            # For binary classification
            if isinstance(shap_values, list):
                shap_values_to_plot = shap_values[1] # Use class one
            else:
                shap_values_to_plot = shap_values

            # Plot
            plt.figure(figsize=(10, 6))
            shap.summary_plot(shap_values_to_plot, X_train,
feature_names=feature_names, show=False)
            plt.title(f'SHAP Feature Importance: {model_name} (Clinical features +
{dataset_name})', fontsize=10)
            plt.tight_layout()
            plt.show()

        elif model_name == "Logistic Regression":
            explainer = shap.LinearExplainer(model, X_train)
```

```
            shap_values = explainer.shap_values(X_train)

            # Plot
            plt.figure(figsize=(10, 6))
            shap.summary_plot(shap_values, X_train, feature_names=feature_names,
show=False)
            plt.title(f'SHAP Feature Importance: {model_name} (Clinical features +
{dataset_name})', fontsize=10)
            plt.tight_layout()
            plt.subplots_adjust(top=0.9)
            plt.show()

        else:  # For SVC, Naive Bayes, MLP
            background = shap.sample(X_train, 70, random_state=42)
            explainer = shap.KernelExplainer(model.predict_proba, background)
            shap_values = explainer.shap_values(X_train[:70]) # Select a data
subsample to explain

            # Plot
            plt.figure(figsize=(10, 6))
            shap.summary_plot(shap_values[:, :, 1], X_train[:70],
feature_names=feature_names, show=False)
            plt.title(f'SHAP Feature Importance: {model_name} (Clinical features +
{dataset_name})', fontsize=10)
            plt.tight_layout()
            plt.subplots_adjust(top=0.9)

            plt.show()

    except Exception as e:
        print(f"SHAP failed for {model_name}. Error: {e}")

# Get the clinical data from the .csv file
file_path_clinic_whole =
"C:\Users\Usuari\Desktop\TFG\Radiomics\MRI\radiomics_clinical.csv"
file_path_clinic_borders =
"C:\Users\Usuari\Desktop\TFG\Radiomics\MRI\borders_clinical.csv"
file_path_clinic_whole_borders =
"C:\Users\Usuari\Desktop\TFG\Radiomics\MRI\radiomics_borders_clinical.csv"

data_whole = pd.read_csv(file_path_clinic_whole)
data_borders = pd.read_csv(file_path_clinic_borders)
data_whole_borders = pd.read_csv(file_path_clinic_whole_borders)

# Drop the ID column
columns_to_remove = ['ID']
data_whole = data_whole.drop(columns=columns_to_remove, errors='ignore')
data_borders = data_borders.drop(columns=columns_to_remove, errors='ignore')
data_whole_borders = data_whole_borders.drop(columns=columns_to_remove,
errors='ignore')

# Separate output column and features
X_whole = data_whole.drop(columns=['response']).values
y_whole = data_whole['response'].values

X_borders = data_borders.drop(columns=['response']).values
y_borders = data_borders['response'].values
```

```
X_whole_borders = data_whole_borders.drop(columns=['response']).values
y_whole_borders = data_whole_borders['response'].values

X_whole[:,-11:] = X_whole[:,-11:].astype(int) # Convert the last 11 columns to
integers (debug)
X_borders[:,-7:] = X_borders[:,-7:].astype(int) # Convert the last 7 columns to
integers (debug)
X_whole_borders[:,-15:] = X_whole_borders[:,-15:].astype(int) # Convert the last 15
columns to integers (debug)

# Reserve a fixed validation set (15%)
X_train_whole, X_val_whole,  y_train_whole, y_val_whole = train_test_split(X_whole,
y_whole, test_size=0.15, stratify=y_whole, random_state=42)
X_train_borders, X_val_borders,  y_train_borders, y_val_borders =
train_test_split(X_borders, y_borders, test_size=0.15, stratify=y_borders,
random_state=42)
X_train_whole_borders, X_val_whole_borders, y_train_whole_borders,
y_val_whole_borders = train_test_split(X_whole_borders, y_whole_borders,
test_size=0.15, stratify=y_whole_borders, random_state=42)

# Define the models with the best hyperparameters to train for the three datasets
models_whole = {
    'Random Forest': RandomForestClassifier(class_weight='balanced',
random_state=42, max_depth = 3, min_samples_split = 10, n_estimators = 100),
    'Support Vector Machine': SVC(class_weight='balanced', random_state=42, C = 1,
gamma = 'scale', kernel = 'rbf'),
    'Logistic Regression': LogisticRegression(random_state=42, C= 0.1, penalty=
'l2', solver = 'lbfgs'),
    'Gradient Boosting': GradientBoostingClassifier(random_state=42),
    'Multilayer Perceptron': MLPClassifier(random_state=42, activation = 'relu',
alpha = 0.0001, hidden_layer_sizes = (100,), learning_rate = 'constant'),
    'Naïve Bayes': GaussianNB(),
    'Decision Tree': DecisionTreeClassifier(class_weight='balanced',
random_state=42, criterion = 'entropy', max_depth = 10, min_samples_leaf = 1,
min_samples_split = 5),
    'XGBoost': XGBClassifier(use_label_encoder=False, eval_metric='logloss',
random_state=42, colsample_bytree = 0.8, learning_rate = 0.05, max_depth = 3,
n_estimators = 100, subsample = 0.8),
    'LightGBM': LGBMClassifier(random_state=42, boosting_type = 'gbdt',
learning_rate = 0.01, min_child_samples = 10, n_estimators = 100, num_leaves = 15)
}

models_borders = {
    'Random Forest': RandomForestClassifier(class_weight='balanced',
random_state=42, max_depth = 5, min_samples_split= 4, n_estimators=300),
    'Support Vector Machine': SVC(class_weight='balanced', random_state=42, C = 10,
gamma = 'auto', kernel = 'rbf'),
    'Logistic Regression': LogisticRegression(random_state=42, C= 0.1, penalty=
'l2', solver = 'lbfgs'),
    'Gradient Boosting': GradientBoostingClassifier(random_state=42),
    'Multilayer Perceptron': MLPClassifier(random_state=42, activation = 'tanh',
alpha = 0.01, hidden_layer_sizes = (50,), learning_rate = 'constant'),
    'Naïve Bayes': GaussianNB(),
    'Decision Tree': DecisionTreeClassifier(class_weight='balanced',
random_state=42, criterion = 'entropy', max_depth = 3, min_samples_leaf = 1,
min_samples_split = 2),
```

```
    'XGBoost': XGBClassifier(use_label_encoder=False, eval_metric='logloss',
random_state=42, colsample_bytree = 1, learning_rate = 0.1, max_depth =5,
n_estimators = 100, subsample = 1),
    'LightGBM': LGBMClassifier(random_state=42, boosting_type = 'dart',
learning_rate = 0.01, min_child_samples = 20, n_estimators = 100, num_leaves = 15)
}

models_whole_borders = {
    'Random Forest': RandomForestClassifier(class_weight='balanced',
random_state=42, max_depth = 3, min_samples_split= 2, n_estimators= 300),
    'Support Vector Machine': SVC(class_weight='balanced', random_state=42, C = 1,
gamma = 'scale', kernel = 'rbf'),
    'Logistic Regression': LogisticRegression(random_state=42, C= 0.1, penalty=
'l2', solver = 'saga'),
    'Gradient Boosting': GradientBoostingClassifier(random_state=42),
    'Multilayer Perceptron': MLPClassifier(random_state=42, activation = 'tanh',
alpha = 0.0001, hidden_layer_sizes = (100,), learning_rate = 'constant'),
    'Naïve Bayes': GaussianNB(),
    'Decision Tree': DecisionTreeClassifier(class_weight='balanced',
random_state=42, criterion = 'entropy', max_depth = 10, min_samples_leaf = 1,
min_samples_split = 5),
    'XGBoost': XGBClassifier(use_label_encoder=False, eval_metric='logloss',
random_state=42, colsample_bytree = 0.8, learning_rate = 0.05, max_depth = 3,
n_estimators = 50, subsample = 0.6),
    'LightGBM': LGBMClassifier(random_state=42, boosting_type = 'gbdt',
learning_rate = 0.01, min_child_samples = 20, n_estimators = 100, num_leaves = 15)
}

# Initialize the 5-Fold CV
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Create a diccionary to store the results
results = {}

# CV in all three datasets
for dataset, X_train_full, X_val, y_train_full, y_val, scale_columns in [
    ('Whole tumour', X_train_whole, X_val_whole, y_train_whole, y_val_whole,-11),
    ('Tumour borders', X_train_borders, X_val_borders, y_train_borders,
y_val_borders,-7),
    ('Whole tumour + tumour borders', X_train_whole_borders, X_val_whole_borders,
y_train_whole_borders, y_val_whole_borders,-15)
]:
    results[dataset]={}
    if dataset == 'Whole tumour':
      for model_name, model in models_whole.items():
        results[dataset][model_name] = {'Train': [], 'CV Test': [], 'Val': None}
        print(f"\n{model_name} (5-Fold CV) for {dataset} dataset")
        for fold, (train_idx, test_idx) in enumerate(skf.split(X_train_full,
y_train_full), 1):
            X_train, X_test = X_train_full[train_idx], X_train_full[test_idx]
            y_train, y_test = y_train_full[train_idx], y_train_full[test_idx]

            # Scale within fold
            scaler = StandardScaler()
            X_train[:,scale_columns:] =
scaler.fit_transform(X_train[:,scale_columns:])
            X_test[:,scale_columns:]  = scaler.transform(X_test[:,scale_columns:])
```

```
            model.fit(X_train, y_train)

            y_train_pred = model.predict(X_train)
            y_test_pred = model.predict(X_test)

            acc_train = accuracy_score(y_train, y_train_pred)
            acc_test = accuracy_score(y_test, y_test_pred)

            results[dataset][model_name]['Train'].append(acc_train)
            results[dataset][model_name]['CV Test'].append(acc_test)

            print(f"Fold {fold} - Train Acc: {acc_train:.4f} | CV Test Acc:
{acc_test:.4f}")

        # Retrain on full training set and evaluate on final validation set
        final_scaler = StandardScaler()
        X_train_full[:,scale_columns:] =
final_scaler.fit_transform(X_train_full[:,scale_columns:])
        X_val[:,scale_columns:] = final_scaler.transform(X_val[:,scale_columns:])

        model.fit(X_train_full, y_train_full)
        y_val_pred = model.predict(X_val)

        acc_val = accuracy_score(y_val, y_val_pred)
        results[dataset][model_name]['Val'] = acc_val

        print(f"Validation Accuracy: {acc_val:.4f}")
        print("Classification Report (Validation Set):\n",
classification_report(y_val, y_val_pred))
        plot_confusion_matrix(y_val, y_val_pred, model_name, dataset)

        # Plot the Feature Importance
        feature_names = data_whole.drop(columns=['response']).columns
        plot_shap_summary(model, X_train_full, feature_names, model_name, dataset)
        plot_feature_importance(model, X_train_full, feature_names, model_name,
dataset)

    elif dataset == 'Tumour borders':
      for model_name, model in models_borders.items():
        results[dataset][model_name] = {'Train': [], 'CV Test': [], 'Val': None}
        print(f"\n{model_name} (5-Fold CV) for {dataset} dataset")
        for fold, (train_idx, test_idx) in enumerate(skf.split(X_train_full,
y_train_full), 1):
            X_train, X_test = X_train_full[train_idx], X_train_full[test_idx]
            y_train, y_test = y_train_full[train_idx], y_train_full[test_idx]

            # Scale within fold
            scaler = StandardScaler()
            X_train[:,scale_columns:] =
scaler.fit_transform(X_train[:,scale_columns:])
            X_test[:,scale_columns:]  = scaler.transform(X_test[:,scale_columns:])

            model.fit(X_train, y_train)

            y_train_pred = model.predict(X_train)
            y_test_pred = model.predict(X_test)
```

```
            acc_train = accuracy_score(y_train, y_train_pred)
            acc_test = accuracy_score(y_test, y_test_pred)

            results[dataset][model_name]['Train'].append(acc_train)
            results[dataset][model_name]['CV Test'].append(acc_test)

            print(f"Fold {fold} - Train Acc: {acc_train:.4f} | CV Test Acc:
{acc_test:.4f}")

        # Retrain on full training set and evaluate on final validation set
        final_scaler = StandardScaler()
        X_train_full[:,scale_columns:] =
final_scaler.fit_transform(X_train_full[:,scale_columns:])
        X_val[:,scale_columns:] = final_scaler.transform(X_val[:,scale_columns:])

        model.fit(X_train_full, y_train_full)
        y_val_pred = model.predict(X_val)

        acc_val = accuracy_score(y_val, y_val_pred)
        results[dataset][model_name]['Val'] = acc_val

        print(f"Validation Accuracy: {acc_val:.4f}")
        print("Classification Report (Validation Set):\n",
classification_report(y_val, y_val_pred))
        plot_confusion_matrix(y_val, y_val_pred, model_name, dataset)

        # Plot the Feature Importance
        feature_names = data_borders.drop(columns=['response']).columns
        plot_shap_summary(model, X_train_full, feature_names, model_name, dataset)
        plot_feature_importance(model, X_train_full, feature_names, model_name,
dataset)

    else:
      for model_name, model in models_whole_borders.items():
        results[dataset][model_name] = {'Train': [], 'CV Test': [], 'Val': None}
        print(f"\n{model_name} (5-Fold CV) for {dataset} dataset")
        for fold, (train_idx, test_idx) in enumerate(skf.split(X_train_full,
y_train_full), 1):
            X_train, X_test = X_train_full[train_idx], X_train_full[test_idx]
            y_train, y_test = y_train_full[train_idx], y_train_full[test_idx]

            # Scale within fold
            scaler = StandardScaler()
            X_train[:,scale_columns:] =
scaler.fit_transform(X_train[:,scale_columns:])
            X_test[:,scale_columns:]  = scaler.transform(X_test[:,scale_columns:])

            model.fit(X_train, y_train)

            y_train_pred = model.predict(X_train)
            y_test_pred = model.predict(X_test)

            acc_train = accuracy_score(y_train, y_train_pred)
            acc_test = accuracy_score(y_test, y_test_pred)

            results[dataset][model_name]['Train'].append(acc_train)
```

```
            results[dataset][model_name]['CV Test'].append(acc_test)

            print(f"Fold {fold} - Train Acc: {acc_train:.4f} | CV Test Acc:
{acc_test:.4f}")

        # Retrain on full training set and evaluate on final validation set
        final_scaler = StandardScaler()
        X_train_full[:,scale_columns:] =
final_scaler.fit_transform(X_train_full[:,scale_columns:])
        X_val[:,scale_columns:] = final_scaler.transform(X_val[:,scale_columns:])

        model.fit(X_train_full, y_train_full)
        y_val_pred = model.predict(X_val)

        acc_val = accuracy_score(y_val, y_val_pred)
        results[dataset][model_name]['Val'] = acc_val

        print(f"Validation Accuracy: {acc_val:.4f}")
        print("Classification Report (Validation Set):\n",
classification_report(y_val, y_val_pred))
        plot_confusion_matrix(y_val, y_val_pred, model_name, dataset)

        # Plot the Feature Importance
        feature_names = data_whole_borders.drop(columns=['response']).columns
        plot_shap_summary(model, X_train_full, feature_names, model_name, dataset)
        plot_feature_importance(model, X_train_full, feature_names, model_name,
dataset)

# Final summary
print("\nSummary:")
for datasets, models in results.items():
    print(f"\n{datasets} dataset:")
    for model, acc in models.items():
      print(f"{model}")
      print(f"  Avg Train Acc: {np.mean(acc['Train']):.4f} ±
{np.std(acc['Train']):.4f} | Avg CV Test Acc: {np.mean(acc['CV Test']):.4f} ±
{np.std(acc['CV Test']):.4f} | Validation Acc: {acc['Val']:.4f}”)
```

Annex 4. Machine Learning model results

Core Tumour Radiomics dataset results:

Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	Validation Sensitivity	Validation Specificity
Decision Tree	0.86 ± 0.01	0.64 ± 0.12	0.62	0.83	0.43
Random Forest	0.91 ± 0.02	0.71 ± 0.11	0.77	0.67	0.86
Support Vector Machine	0.71 ± 0.02	0.62 ± 0.19	0.69	0.67	0.71
Logistic Regression	0.63 ± 0.04	0.65 ± 0.10	0.62	0.33	0.86
Naïve Bayes	0.63 ± 0.02	0.58 ± 0.07	0.62	1.00	0.29
Multilayer Perceptron	0.80 ± 0.02	0.62 ± 0.10	0.62	0.50	0.71
Gradient Boosting	1.00 ± 0.00	0.56 ± 0.11	0.77	0.67	0.86
XGBoost	0.99 ± 0.01	0.58 ± 0.04	0.54	0.50	0.57
LightGBM	0.75 ± 0.03	0.65 ± 0.13	0.54	0.50	0.57

Table 18. Machine Learning model results for the Core Tumour Radiomics dataset.

Tumour Border Radiomics dataset results:

Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	Validation Sensitivity	Validation Specificity
Decision Tree	0.99 ± 0.01	0.59 ± 0.08	0.46	0.33	0.57
Random Forest	0.80 ± 0.02	0.60 ± 0.12	0.54	0.33	0.71
Support Vector Machine	0.66 ± 0.05	0.63 ± 0.20	0.54	0.67	0.43
Logistic Regression	0.66 ± 0.03	0.66 ± 0.13	0.62	0.50	0.71
Naïve Bayes	0.69 ± 0.05	0.60 ± 0.15	0.62	0.50	0.71

Multilayer Perceptron	0.69 ± 0.03	0.65 ± 0.11	0.62	0.50	0.71
Gradient Boosting	1.00 ± 0.00	0.55 ± 0.03	0.46	0.17	0.71
XGBoost	0.99 ± 0.01	0.66 ± 0.08	0.62	0.50	0.71
LightGBM	0.76 ± 0.03	0.63 ± 0.11	0.54	0.33	0.71

Table 19. Machine Learning model results for the Tumour Border Radiomics dataset.

Core Tumour Radiomics + Tumour Border Radiomics dataset results:

Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	Validation Sensitivity	Validation Specificity
Decision Tree	0.87 ± 0.03	0.59 ± 0.14	0.69	0.67	0.71
Random Forest	1.00 ± 0.00	0.64 ± 0.12	0.62	0.50	0.71
Support Vector Machine	0.75 ± 0.02	0.62 ± 0.16	0.62	0.67	0.57
Logistic Regression	0.71 ± 0.03	0.65 ± 0.12	0.62	0.50	0.71
Naïve Bayes	0.68 ± 0.03	0.62 ± 0.12	0.69	1.00	0.43
Multilayer Perceptron	0.71 ± 0.04	0.62 ± 0.15	0.69	0.67	0.71
Gradient Boosting	1.00 ± 0.00	0.55 ± 0.12	0.69	0.67	0.71
XGBoost	1.00 ± 0.00	0.72 ± 0.08	0.62	0.50	0.71
LightGBM	0.84 ± 0.04	0.65 ± 0.10	0.46	0.33	0.57

Table 20. Machine Learning model results for the Core Tumour + Tumour Border Radiomics dataset.

Clinical variables dataset results:

- With T and N subcategories

Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	Validation Sensitivity	Validation Specificity
Decision Tree	0.80 ± 0.04	0.61 ± 0.06	0.69	0.83	0.57

Random Forest	0.86 ± 0.03	0.65 ± 0.04	0.69	0.67	0.71
Support Vector Machine	0.55 ± 0.03	0.52 ± 0.05	0.46	1.00	0.00
Logistic Regression	0.66 ± 0.03	0.60 ± 0.06	0.62	0.17	1.00
Naïve Bayes	0.63 ± 0.01	0.61 ± 0.08	0.62	0.50	0.71
Multilayer Perceptron	0.60 ± 0.03	0.62 ± 0.08	0.54	0.50	0.57
Gradient Boosting	1.00 ± 0.00	0.58 ± 0.14	0.62	0.50	0.71
XGBoost	0.80 ± 0.03	0.65 ± 0.07	0.69	0.67	0.71
LightGBM	0.76 ± 0.06	0.65 ± 0.11	0.69	0.67	0.71

Table 21. Machine Learning model results for the Clinical variables dataset (T and N subcategories).

- Without T and N subcategories

Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	Validation Sensitivity	Validation Specificity
Decision Tree	0.77 ± 0.02	0.57 ± 0.09	0.69	0.83	0.57
Random Forest	0.90 ± 0.01	0.63 ± 0.02	0.62	0.67	0.57
Support Vector Machine	0.63 ± 0.04	0.61 ± 0.15	0.62	0.67	0.57
Logistic Regression	0.61 ± 0.01	0.61 ± 0.03	0.54	0.17	0.86
Naïve Bayes	0.59 ± 0.05	0.59 ± 0.15	0.62	0.67	0.57
Multilayer Perceptron	0.58 ± 0.02	0.58 ± 0.10	0.38	0.17	0.57
Gradient Boosting	0.99 ± 0.01	0.52 ± 0.14	0.54	0.50	0.57
XGBoost	0.82 ± 0.02	0.59 ± 0.04	0.62	0.50	0.71
LightGBM	0.85 ± 0.04	0.62 ± 0.09	0.46	0.50	0.43

Table 22. Machine Learning model results for the Clinical variables dataset.

Clinical variables + Core Tumour Radiomics dataset results:

- With T and N subcategories

Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	Validation Sensitivity	Validation Specificity
Decision Tree	0.9859 ± 0.0070	0.6324 ± 0.0875	0.54	0.67	0.43
Random Forest	0.90 ± 0.03	0.75 ± 0.10	0.62	0.50	0.71
Support Vector Machine	0.54 ± 0.06	0.49 ± 0.10	0.46	1.00	0.00
Logistic Regression	0.70 ± 0.04	0.62 ± 0.06	0.62	0.50	0.71
Naïve Bayes	0.67 ± 0.03	0.63 ± 0.05	0.62	0.67	0.57
Multilayer Perceptron	0.77 ± 0.01	0.65 ± 0.06	0.69	0.50	0.86
Gradient Boosting	1.00 ± 0.00	0.66 ± 0.09	0.46	0.33	0.57
XGBoost	0.99 ± 0.01	0.68 ± 0.10	0.69	0.50	0.86
LightGBM	0.86 ± 0.02	0.61 ± 0.10	0.54	0.17	0.86

Table 23. Machine Learning model results for the Clinical variables (T and N subcategories) + core tumour radiomics dataset.

- Without T and N subcategories

Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	Validation Sensitivity	Validation Specificity
Decision Tree	0.90 ± 0.03	0.62 ± 0.15	0.54	0.50	0.57
Random Forest	0.99 ± 0.01	0.73 ± 0.07	0.69	0.50	0.86
Support Vector Machine	0.54 ± 0.06	0.49 ± 0.10	0.46	1.00	0.00
Logistic Regression	0.69 ± 0.03	0.66 ± 0.05	0.62	0.50	0.71
Naïve Bayes	0.62 ± 0.03	0.63 ± 0.09	0.69	0.83	0.57

Multilayer Perceptron	0.79 ± 0.023	0.66 ± 0.12	0.69	0.50	0.86
Gradient Boosting	1.00 ± 0.00	0.56 ± 0.05	0.46	0.17	0.71
XGBoost	0.91 ± 0.02	0.66 ± 0.08	0.62	0.33	0.86
LightGBM	0.96 ± 0.02	0.63 ± 0.10	0.54	0.33	0.71

Table 24. Machine Learning model results for the Clinical variables + core tumour radiomics dataset.

Clinical variables + Border Tumour Radiomics dataset results:

- With T and N subcategories

Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	Validation Sensitivity	Validation Specificity
Decision Tree	0.78 ± 0.03	0.69 ± 0.13	0.46	0.17	0.71
Random Forest	0.99 ± 0.01	0.72 ± 0.09	0.69	0.50	0.86
Support Vector Machine	1.00 ± 0.00	0.52 ± 0.09	0.62	0.50	0.71
Logistic Regression	0.74 ± 0.05	0.69 ± 0.11	0.46	0.17	0.71
Naïve Bayes	0.67 ± 0.03	0.62 ± 0.12	0.62	0.67	0.57
Multilayer Perceptron	0.79 ± 0.02	0.66 ± 0.12	0.62	0.50	0.71
Gradient Boosting	1.00 ± 0.00	0.66 ± 0.05	0.62	0.33	0.86
XGBoost	1.00 ± 0.00	0.69 ± 0.11	0.69	0.50	0.86
LightGBM	0.68 ± 0.02	0.64 ± 0.09	0.46	0.17	0.71

Table 25. Machine Learning model results for the Clinical variables (T and N subcategories) + tumour border radiomics dataset.

- Without T and N subcategories

Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	Validation Sensitivity	Validation Specificity
Decision Tree	1.00 ± 0.00	0.64 ± 0.10	0.69	0.83	0.57

Random Forest	0.94 ± 0.01	0.68 ± 0.09	0.69	0.50	0.86
Support Vector Machine	1.00 ± 0.00	0.48 ± 0.10	0.62	0.50	0.71
Logistic Regression	0.70 ± 0.03	0.65 ± 0.12	0.54	0.33	0.71
Naïve Bayes	0.64 ± 0.06	0.59 ± 0.11	0.62	0.67	0.57
Multilayer Perceptron	0.76 ± 0.03	0.65 ± 0.07	0.69	0.67	0.71
Gradient Boosting	1.00 ± 0.00	0.58 ± 0.06	0.62	0.33	0.86
XGBoost	0.87 ± 0.03	0.69 ± 0.10	0.62	0.33	0.86
LightGBM	0.68 ± 0.03	0.62 ± 0.09	0.46	0.17	0.71

Table 26. Machine Learning model results for the Clinical variables + tumour border radiomics dataset.

Clinical variables + Core Tumour Radiomics + Border Tumour Radiomics dataset results:

- With T and N subcategories

Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	Validation Sensitivity	Validation Specificity
Decision Tree	0.98 ± 0.02	0.61 ± 0.10	0.62	0.67	0.57
Random Forest	0.96 ± 0.02	0.69 ± 0.12	0.62	0.50	0.71
Support Vector Machine	0.54 ± 0.06	0.49 ± 0.10	0.46	1.00	0.00
Logistic Regression	0.72 ± 0.02	0.68 ± 0.11	0.54	0.33	0.71
Naïve Bayes	0.68 ± 0.03	0.65 ± 0.08	0.62	0.67	0.57
Multilayer Perceptron	0.90 ± 0.02	0.64 ± 0.11	0.77	0.67	0.86
Gradient Boosting	1.00 ± 0.00	0.61 ± 0.08	0.54	0.17	0.86
XGBoost	0.94 ± 0.02	0.66 ± 0.12	0.46	0.17	0.71

LightGBM	0.71 ± 0.01	0.61 ± 0.07	0.62	0.17	1.00

Table 27. Machine Learning model results for the Clinical variables (T and N subcategories) + core tumour radiomics + tumour border radiomics dataset.

- Without T and N subcategories

Model	Avg. Train Accuracy	Avg. Test Accuracy	Validation Accuracy	Validation Sensitivity	Validation Specificity
Decision Tree	1.00 ± 0.00	0.61 ± 0.06	0.54	0.50	0.57
Random Forest	1.00 ± 0.00	0.71 ± 0.07	0.54	0.33	0.71
Support Vector Machine	0.54 ± 0.06	0.49 ± 0.10	0.46	1.00	0.00
Logistic Regression	0.77 ± 0.04	0.66 ± 0.11	0.62	0.50	0.71
Naïve Bayes	0.66 ± 0.02	0.65 ± 0.08	0.69	0.83	0.57
Multilayer Perceptron	0.92 ± 0.01	0.63 ± 0.10	0.77	0.83	0.71
Gradient Boosting	1.00 ± 0.00	0.61 ± 0.11	0.62	0.33	0.86
XGBoost	0.99 ± 0.01	0.66 ± 0.11	0.62	0.33	0.86
LightGBM	0.71 ± 0.01	0.61 ± 0.07	0.62	0.17	1.00

Table 28. Machine Learning model results for the Clinical variables + core tumour radiomics + tumour border radiomics dataset.