Norwegian University of Life Sciences

Master's Thesis 2023 30 ECTS
Faculty for Science and Technology

Predicting Treatment Outcome Using Interpretable Models for Patients with Head and Neck Cancer

Alida Karlstrøm Martinsen
Miljøfysikk og fornybar energi

Preface

This thesis was conducted at the Faculty of Science and Technology at the Norwegian University of Life Sciences in the srping of 2023. It marks the end of a six year long stay in Ås, and a finished M.Sc degree in Environmental physics and renewable energy.

I would like to thank my supervisor, Cecilia Marie Futsæther, for her insightful feed-back and undying enthusiasm throughout the whole process. A big thanks to my assistant supervisor Oliver Tomic for always answering any questions and providing interesting views on every weekly meeting. I would also like to thank ph.d students Bao Hgoc Huynh and Aurora Rosvoll Grøndahl for always answering my questions and assisting me if I neede it. This thesis would not have been possible without all of their guidance and support throughout the process. A thank you to professor Eirik Malinen (Universitetet i Oslo) and doktor Einar Dale (Oslo University Hospital) for providing me with the datasets used in this thesis.

Lastly, I would like to thank my parents and my partner for being so supportive throughout the process. A genuine thanks to all my peers at Miljøfysikk, who I was so lucky to spend my years at university with.

Alida Karlstrøm Martinsen
Ås, 15th of May 2023

Abstract

Head and neck cancer accounts for around 3%of cancers worldwide, resulting in many deaths each year. The increasing number of patients receiving a cancer diagnosis increases the demand for accurate diagnosis and effective treatment. Intra-tumor heterogeneity is said to be one of the issues in cancer therapy, an issue that needs to be solved. Radiomics pave the way for extracting features based on the shape, size, and texture of the entire tumor.

Radiomics extracts features from tumors based on the gray levels in a medical image. The process of radiomics is intended to capture texture and heterogeneity in the tumor that would be impossible to deduce from a simple tumor biopsy. Feature extraction by radiomics has been proven to enrich clinical datasets with valuable features that positively impact the performance of predictive models.

This thesis investigates the use of clinical and radiomics features for predicting treatment outcomes of head and neck cancer patients using interpretable models. The radiomics algorithm extracts first-order statistical, shape, and texture features from PET and CT images of each patient. The 139 patients in the training dataset were from Oslo University Hospital (OUS), whereas the 99 patients in the test set were from the MAASTRO clinic in the Netherlands. All the clinical features, together with the radiomics features, counted 388 features in total. Feature selection through the repeated elastic net technique (RENT) was performed to exclude irrelevant features from the dataset. Seven different tree-based machine learning algorithms were fitted to the data, and the performance was validated by the accuracy, ROC AUC, Matthews correlation coefficient, F1 score for class 1, and F1 score for class 0. The models were tested on the external MAASTRO dataset, and the overall best-performing models were interpreted.

On the external dataset from the MAASTRO clinic, the highest-performing models obtained an MCC of 0.37 for OS prediction and 0.44 for DFS prediction. For both OS and DFS, the highest predictions were made on only the clinical data. Transparency in machine learning models greatly benefits decision-makers in clinical settings, as every prediction can be reasoned for. Predicting treatment outcomes for head and neck patients is highly possible with interpretable models. To determine if the methods used in this thesis are suited for predicting treatment outcomes for head and neck cancer patients, it is necessary to test the methods and models on more datasets.

Sammendrag

Hode- og halskreft står for rundt 3% av krefttilfellene over hele verden, og forårsaker mange dødsfall hvert år. Det økende antallet pasienter som får en kreftdiagnose øker etterspørselen etter nøyaktig diagnosering og effektiv behandling. Intra-tumor heterogenitet sies å være et av problemene innen kreftbehandling, et problem som må løses. Radiomics gjør det mulig å ekstrahere egenskaper basert på formen, størrelsen og teksturen til hele svulsten.

Radiomics ekstraherer egenskaper fra svulster basert på gråtonenivåene i et medisinsk bilde. Radiomics algoritmen fanger tekstur og heterogenitet i svulsten som ikke ville være mulig å utlede fra en biopsi. Egenskapsuthenting med Radiomics har vist seg å berike kliniske datasett med nyttige egenskaper som positivt påvirker ytelsen til prediktive modeller.

Denne oppgaven undersøker bruken av kliniske og radiomics egenskaper for å predikere behandlingsutfall for hode- og halskreftpasienter ved å bruke tolkbare modeller. Egenskaper ble esktrahert fra bildene med radiomics De 139 pasientene i treningsdatasettet var fra Oslo Universitetssykehus (OUS), mens de 99 pasientene i testsettet var fra MAASTRO-klinikken i Nederland. Alle de kliniske egenskapene sammen med radiomicsegenskapene utgjorde totalt 388 egenskaper. Egenskapsutvelging gjennom repetert elastisk nett teknikk (RENT) ble utført for å ekskludere irrelevante egenskaper fra datasettet. Syv forskjellige trebaserte maskinlæringsalgoritmer ble tilpasset dataene, og ytelsen ble validert av nøyaktigheten, ROC AUC, Matthews korrelasjonskoeffisient, F1-score for klasse 1 og F1-score for klasse 0. Modellene ble deretter testet på et eksternt fra MAASTRP klinikken, og modellene med høyest ytelse ble tolket.

På det eksterne datasettet fra MAASTRO-klinikken oppnådde modellene en MCC på $0,37$ for OS-prediksjon, og $0,44$ for DFS-prediksjon. Datasettene som oppnådde denne ytelsen var kun basert på de kliniske egenskapene. Åpenhet og tolkbarhet i maskinlæringsmodeller er viktig dersom modellene skal brukes til klinisk beslutningsstøtte, ettersom alle prediksjoner kan begrunnes. Å predikere behandlingsutfall for hode- og halskreftpasienter er mulig med tolkbare modeller, men for å avgjøre om metodene anvendt i denne oppgaven er skikket til dette, må metodene og modellene testes på flere datasett.

List of Abbreviations

Abbreviation	Definition
HPV	Human papillomavirus
PET	Positron emission tompgraphy
FDG	Fluordeoxy glucose
CT	Computer tomography
PMT	Photomultiplier tubes
LOR	Line of response
keV	Kilo electron volt
SUV	Standard uptake value
MTV	Mean tumor volume
TLG	Tumor lesion glycolysis
ROI	Region of interest
VOI	Volume of interest
ML	Machine learning
DT	Decision tree
RF	Random forest
XGB	XGBoost
HGB	Histogram Gradient Boosting
FIGS	Fast-interpretable greedy tree sums
HSTree	Hierarchical shrinkage tree
BR	Boosted rules
TP	True positive
TN	True negative
FP	False positive
FN	Falsk negative
ACC	Accuracy
AUC	Area under the curve
MCC	Matthews correlation coefficient
PRE	Precision
REC	Recall
OUS	Oslo University Hospital (Oslo Universitetssykehus)
OS	Overall survival
DFS	Disease-free survival
RENT	Rpeated elastic net technique

Continued on next page

Table 0.0.1 Continued from previous page

Abbreviation	Definition
LBP	Local binary pattern

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In 2020, cancer took the lives of nearly 10 million people worldwide [3]. In Norway, 38 265 people were affected by cancer in 2022, which is slightly higher than the numbers for 2021 [4]. Cancer can be diagnosed using medical imaging, such as PET and CT images [5]. Cancer can form in all parts of the body, including the head and neck region.

Head and neck cancer is a term for all cancers originating in the mouth, lip, nose, sinuses, and throat. It affected 744 994 patients worldwide in 2020 [3], and it accounts for around 3% of cancer cases [3]. Risk factors for cancers in the head and neck include smoking, humano papillomavirus (HPV) infection, obesity, and alcohol consumption [6]. Incidence rates of head and neck cancer are rising, and some studies have contributed it to the increase of sexually transmitted HPV-infections [6].

Cancers are most commonly treated with surgery, chemotherapy, or radiation therapy [6]. Cancer treatment aims to remove or destroy cancerous cells without harming surrounding tissues and organ function [6]. Cancer manifests differently in every patient, and the need for accurate diagnosis and personalized treatment is more significant than ever [7]. The distinct tumor phenotypes in human cancers calls for imaging techniques that can accurately portray the intra-tumor heterogeneities non-invasively [8]. Through increased knowledge of the spatial tumor heterogeneity, it is possible that personalized cancer treatment can minimize treatment resistance and improve clinical outcomes [9].

1.2 Method

Machine Learning has the potential to be used for decision support in the medical field [10]. Many machine learning models make accurate predictions but at the cost of interpretability [10]. For high-stakes decisions, black box models should be avoided, and inherently interpretable models should be used instead [10]. Interpretable models are defined in this thesis as rule-based models that can be understood by a human.

Extracting statistical, shape and textural features from a tumor in a medical image can reveal properties of a tumor that cannot be found by inspection of medical images or biopsies of tissues [8]. Radiomics is a data mining technique that uses mathematical and statistical algorithms to extract features from medical images [11]. The extracted features describe the tumor size, shape, density, and texture [12]. Several studies have demonstrated that radiomics features that describe the non-uniformity and heterogeneity within tumors, are significantly associated with overall survival and disease-free survival in head and neck cancers ([13], [14], [15], [16]).

1.3 Aim

The iModels package, proposed by Singh et al. [2], contains many models whose primary aim is to make interpretable models while still obtaining state-of-the-art performance. Rudin [10] argued that making interpretable models without sacrificing performance is possible. This thesis aims to investigate these statements using already well-established algorithms, as well as some newer candidates proposed in the iModels package [2]. Interpretable and transparent models that can provide meaningful explanations behind its predictions without sacrificing performance are appropriate for decision support in the medical field.

1.4 Outline

Chapter 2 of this thesis will present the theoretical background for all the concepts used to obtain the final results. Chapter 2 starts with Section 2.1, a quick explanation of cancer, diagnosis, and treatment. In Section 2.3, the basic concepts of nuclear physics are explained. The imaging modalities used in this thesis are then covered in Section 2.5 on Nuclear Medicine. Further, Section 2.6 explains how radiomics can extract tabular features of the tumor and affected lymph nodes from medical images. Finally, Chapter 2 is concluded with Section 2.7 where the machine learning methods used in the thesis are described in detail. Chapter 3 outlines the methods used in this thesis. The results are presented in Chapter 4, followed by a discussion in Chapter 5. Finally, Chapter 6 summarizes the conclusions of this thesis.

Chapter 2

Theory

2.1 Cancer

Cancer is a term used for many diseases caused by uncontrolled cell division in different types of cells [17]. A mutation is always possible when cells divide, and specific mutations can cause rapid cell division [17]. When a cell divides faster than normal, it can form a cluster of cancerous cells known as a tumor. Tumors can be malignant (cancerous) or benign (noncancerous) [18]. Cancerous tumors can spread to nearby tissues or travel with the bloodstream or lymph system to other distal parts of the body. This process is called metastases [19]. Metastases cause secondary tumors to form in other parts of the body. Cancerous tumors can form in all body parts, including the head and neck, which is the case for the patients in this thesis.

2.2 Head and neck cancer

Head and neck cancer refers to cancers that arise in the head, throat, mouth, lips, sinuses, and nasal cavities [6]. In 2017-2021, 4 170 Norwegian patients were diagnosed with head and neck cancer [20]. Multiple risk factors are connected to head and neck cancer, including smoking, drinking, poor diet, or being infected by the human papillomavirus (HPV) [6]. Diagnosing head and neck cancer can be done by PET/CT, where combining a functional PET image and a structural CT image can precisely determine the tumor size and activity. PET/CT is useful for staging and planning treatment for the patient [21].

There are multiple sites where cancer can arise, and the datasets in this thesis deals with *cavum oris*, *hypopharynx*, *oropharynx*, and *larynx*. Cavum oris refers to cancer in the oral cavity, hypopharynx refers to the lower part of the neck, and larynx refers to the voice box. The oropharynx refers to the throat's middle part, the back of the mouth and tongue, and the tonsils [22]. Oropharyngeal cancer is often related to HPV [22].

2.3 Nuclear Physics

This section explains the relevant terms for Section 2.5.

2.3.1 Radioactivity and radioactive decay

In the nucleus of every atom, there are a number of neutrons and a number of protons. The protons and neutrons are held together by forces, and when these forces are imbalanced, the nucleus becomes unstable and thereby radioactive [23], [24]. An unstable parent nucleus will attempt to become stable by emitting radiation and transforming into a more stable daughter nuclide. This is the process of radioactive decay [24]. The nuclear radiation emitted in the process exists in one of three forms; alpha and beta particles, or gamma rays.

2.3.2 Alpha and Beta decay

In alpha decay, a particle with two neutrons and two protons is emitted [24]. It is equivalent to a ${}^{4}_{2}He$ nuclei [24]. Alpha particles have low penetrative abilities and cannot pass through a sheet of paper [24].

Beta (β) decay is a process that converts a proton to a neutron or vice versa by emitting a beta particle of either positive or negative charge [24]. The negative beta particle, β^-, is identical to an electron in mass and charge, whereas the positive beta particle, β^+, is the antiparticle of the electron [24]. This β^+-particle is called a positron. When a radionuclide emits a positron, it follows the reaction

$$p \rightarrow n + e^+ + \nu \tag{2.3.1}$$

where p denotes the proton, n denotes the neutron, e^+ is the positron, and ν is the neutrino. Equation 2.3.1 shows β^+ decay [24]. The neutrino is an electrically neutral particle with a negligible mass that carries some of the energy from the reaction [24]. In Equation 2.3.1, a proton is converted to a neutron by emitting a positron and a neutrino. The process in Equation 2.3.1 is referred to as positron emission [24]. Positron emission is the foundation of Positron Emission Tomography (PET) imaging [25].

2.3.3 Gamma decay

Gamma decay occurs when an excited nucleus emits a gamma photon, entering a lower energy state [24]. A nucleus becomes excited when energy is absorbed into its structure, which can then be released [24]. Photons have the highest energy and penetrative power among the three nuclear radiation types. A photon is released from an excited nucleus A^* by

$$A^* \rightarrow A + \gamma \tag{2.3.2}$$

where γ denotes the photon released in the process, and A denotes the nucleus in its ground state. Gamma photons are emitted during annihilation of a positron and an electron, which is a process relevant to PET imaging [25].

2.3.4 X-rays

An X-ray is an electromagnetic wave with an energy of 10 - 150 keV [26]. Because of their high energy, X-rays can penetrate through matter and have been used to depict the insides of a body in an X-ray image. X-ray photons are ionizing radiation that reacts with matter differently depending on their energies [24]. X-rays form the basis of Computed Tomography (CT) imaging [27].

2.4 Interaction with matter

The energy of a photon and X-rays makes them able to interact with matter. The way in which photons react with matter are determined by their energy. The most common ways photons react with matter is by photoelectric effect, Compton scattering, pair production, and annihilation [28].

2.4.1 Photoelectric effect

One of the ways photons react with matter is by the process of the photoelectric effect. In the photoelectric effect, the energy of an incoming photon is absorbed by the electron in an atom of the receiving matter. The electron takes up energy from the photon and detaches into space [24]. Its kinetic energy, T, is given by

$$T = E_\gamma - B_e \tag{2.4.1}$$

where E_γ is the incident photons energy and B_e is the energy needed to release the electron from the atom, the binding energy of the electron.

2.4.2 Compton scattering

Compton scattering is a process where a photon collides with a free electron, causing both the electron and the photon to scatter [24]. The photon releases some of its energy to the electron, and it is deflected in another direction with lower energy. The electron now has kinetic energy [24].

2.4.3 Internal pair formation

Internal pair formation is the process where a photon turns into a positron-electron pair in the presence of a nucleus [29], as illustrated in Figure 2.4.1. Only photons of energies exceeding $1.02 MeV$, or $2m_ec^2$, two times the rest energy m_ec^2 for an electron, can induce pair formation [29].

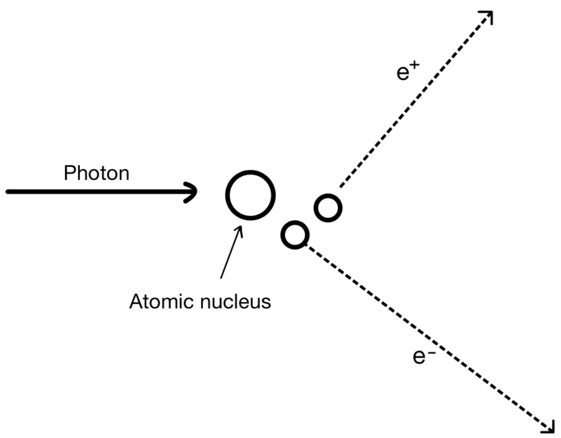


Figure 2.4.1: The process of pair formation. Adapted from [29].

2.4.4 Annihilation

When a released positron, e^+ collides with its anti-particle, the electron, they react with each other. This reaction is called an annihilation [24]. The positron and electron annihilate each other, resulting in two 511 keV photons radiating in opposite directions. Each photons energy is equivalent to the resting energy of one electron. The process is illustrated in Figure 2.4.2.

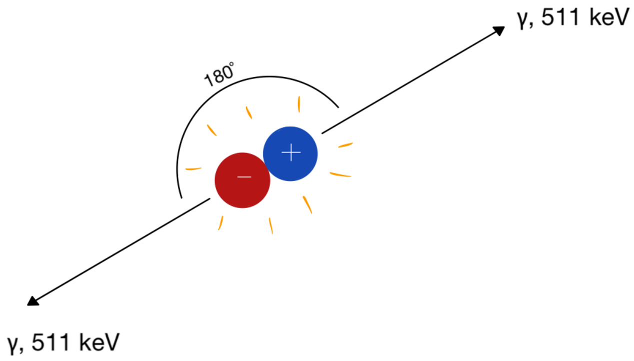


Figure 2.4.2: The process of annihilation. A red electron and a blue positron meet and two photons, γ are emitted to opposite sides, each with an energy of 511 keV. Adapted from [25].

$$e^- + e^+ = \gamma + \gamma \tag{2.4.2}$$

The process of annihilation is given in Equation 2.4.2 where e^- is the electron, e^+ is the positron, or anti-electron, and γ is the resulting photon that are emitted from the interaction [30].

2.5 Nuclear medicine

Nuclear physics principles can be utilized for various applications, including medicine. Nuclear medicine is a branch in the medical field that uses radionuclides to diagnose, treat, and monitor disease progression after treatment [28]. The concepts of the imaging modalitites proton emission tomography (PET), computed tomography (CT), and a combination of the two (PET/CT) are relevant to this thesis.

2.5.1 Positron Emission Tomography

Positron emission tomography (PET) is a functional imaging technique that detects organ function in a patient's body [21]. The most central principle in PET imaging is the proton emission from a proton-rich radionuclide injected into the patient as a radioactive tracer [28]. Several types of tracers are used in PET scans, and the one used in this case is the 18F-fluorodeoxyglucose (18F-FDG) [13]. The 18F-FDG molecule is a glucose molecule with a positron-emitting fluorine-18 atom attached. The fluorine-18 is an isotope of fluorine, and it has a half-life of 110 minutes [28]. Its radioactive decay follows equation 2.5.1 [28]

$$^{18}_{9}F \xrightarrow[T_{1/2}=110min]{} {}^{18}_{8}O + e^{+} + \nu \tag{2.5.1}$$

where $T_{1/2} = 110\ min$ gives the half-life of the radionuclide, $^{18}_{8}O$ is the oxygen nucleus that is produced from the fluorine isotope, e^{+} is the released positron, and ν is the neutrino released in the reaction. As seen in Equation 2.5.1 a proton-rich radionuclide, $^{18}_{9}F$, decays by beta decay, also referred to as positron emission [24], which was described by Equation 2.3.1. The positron, e^{+}, from reaction 2.5.1 travels through surrounding matter and comes to rest after a short distance. It then reacts with a close by electron, causing both particles to annihilate into two photons emitted in opposite directions [28], as shown in Figure 2.4.2. These photons are gamma rays with energies of 511k keV [28]. The PET camera can detect these photons in a PET scan [28].

The PET scanner

The PET camera consists of several components; scintillation crystals coupled with photomultiplier tubes (PMTs) in a ring formation [28]. Typical materials for scintillation crystals in PET detectors are bismuth germanate (BGO), lutetium oxyorthosilicate (LSO), or gadolinium oxyorthosilicate (GSO) [25].

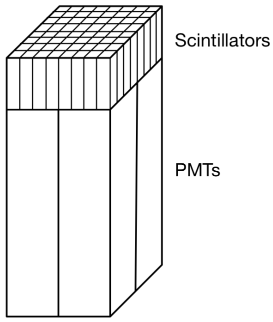


Figure 2.5.1: The structure of a PET camera component, consisting of scintillators and photomultiplier tubes (PMTs). Adapted from [28].

The scintillation crystal produces photons when hit by the incoming photons from the patient. The crystal then produces light with the same energy that the photon lost in the crystal [26]. This scintillation light is then multiplied in the photomultiplier tubes. The photomultiplier tubes convert the photons to voltage by exploiting the photoelectric effect where a photon releases its energy to an electron. [24].

In PET cameras, several crystals are coupled to the same PMT, mainly due to cost and space restraints [28]. Many of the detector elements from Figure 2.5.1 are then attached in a ring formation to form a ring detector, as seen in Figure 2.5.2, covering one transversal slice of the patient's body [28]. Several of these rings are stacked together to cover multiple transversal slices simultaneously. The diameter of the ring varies greatly depending on the scanner's designated use. For a human whole body scan, it is typically around 90 cm [28]. Moving the ring of detectors slightly during the scanning process ensures coverage of the blind spots between the scintillation crystals [29]. The image resolution of modern day PET scanners is around $3 - 5mm$ [28].

The PET scan

The patient is injected with the radioactive tracer 18F-FDG. The tracer is then distributed throughout the body according to the patients specific pharmacokinetics [25]. Since cancer cells have higher metabolic activity than normal cells [17], their sugar uptake is more elevated, causing the sugar-bound tracer to accumulate in the cancer cells [31]. When the radionuclide inside the tracer starts to decay, the cancerous cells will contain more radionuclides, emitting more radioactivity than the normal cells. The photons emitted by the radioactive decay are detected in the PET camera [28].

Suppose the PET camera detects two photons within a specific time interval. In that case, they are assumed to be from the same annihilation, meaning that the annihilation must have happened somewhere on the line between them [28], as seen in the left Figure in 2.5.2. This line is called the line of response (LOR), and this method of detection is called coincidence detection [28]. The time interval for coincidence detection is denoted 2τ, and needs to be kept as small as possible to limit noise from random coincidences. The time it takes from the first photon from the annihilation hits the detector to the second one is detected is called time of flight (TOF) [28].

A digital pulse is formed when a photon is detected by one of the detectors. This digital pulse occurred at the time t and can be coupled with other photons that occurred at $t \pm \tau$, and thus 2τ is often referred to as the coincidence detection time window [28]. If no other photon occurred in the interval of time $t \pm \tau$, the incoming photon is ignored by the system [28]. If the detector captures more than two photons within the time interval, the event is again ignored. There are multiple types of coincidences; true, scattered, and random.

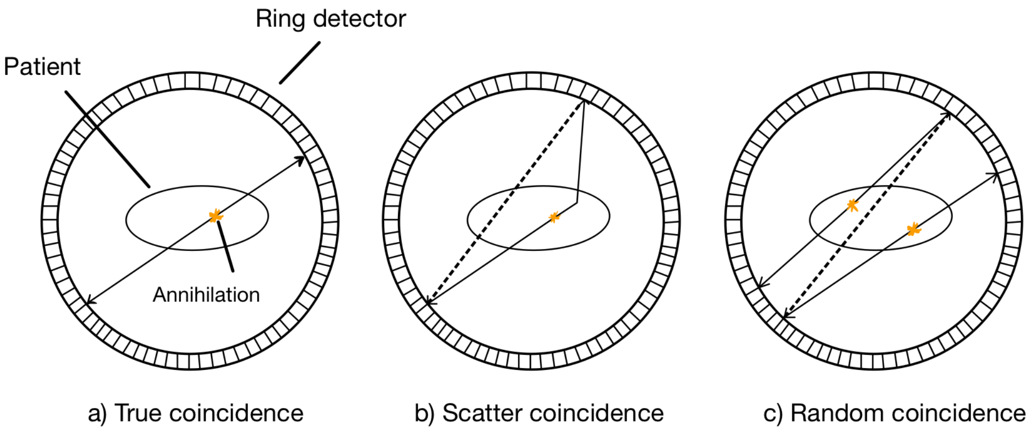


Figure 2.5.2: The types of coincidences detected in PET imaging. The ring detector consists of many PET cameras. A true coincidence occurs when the annihilated photons travel directly to the detector without interacting with the tissues of the patient's body. A scatter coincidence occurs when one or both of the photons are scattered in the body, and causes a false LOR to be registered by the detector, the dotted line in subfigure b). A random coincidence occurs when two unrelated photons arrive at the detectors within the time interval, and the detector assumes they originated from the same annihilation. Again this results in a false LOR, the dotted line in subfigre c) Adapted from [28].

A true coincidence occurs when two photons travel directly from the annihilation site to the detectors without interruption, as seen in Figure 2.5.2. A scattered coincidence happens when one or both of the photons are scattered in the body's tissues and therefore are detected along a different LOR, the dotted line in Figure 2.5.2b. Since photons lose energy when interacting with matter, the energy levels that are measured in the detector determine which photons are true and which are scattered. An energy interval for the detector is typically defined at around $350-650\ keV$. A random coincidence occurs when photons from different reactions are detected at the same time. The random coincidences are limited by keeping the time interval, $t \pm \tau$, as low as possible [28]. Only the true coincidences are beneficial for the final image [28].

Attenuation correction

When a photon passes through the patient's body, it loses some energy to absorption and scattering, called attenuation [25]. Compton scattering, outlined in Section 2.4.2, is the most common interaction for annihilation photons. As the photons lose energy, the photoelectric effect may also occur [28], as explained in Section 2.4. Some photons get entirely absorbed by the tissues [25]. An attenuation correction technique is applied to the image to correct for the attenuation of the photons. In conventional PET imaging, this is done by mapping out the attenuation in a patient on a transmission image [25]. For newer PET systems, this is done by computed tomography [28].

Parameters

The three main parameters in the PET image are based on the concentration of the radiotracer measured in Bq/ml, and the tumor volume [28]. The standard uptake value, SUV, is the radiotracer concentration when the patient's weight and the injected dose are taken into account, and it is calculated by

$$SUV = \frac{C_i(kBq/mL)}{A(kBq)/W(g)} \tag{2.5.2}$$

where C_i is the mean or maximum concentration within the region of interest (ROI), A is the injected dose of radioactivity in the tracer, and W is the patient's weight [28]. The SUV-value is coupled to the glucose uptake in the ROI [28]. SUV-mean is defined as the mean concentration value of the ROI, whereas SUV-max is defined as the maximum concentration value in the ROI [28]. SUV-peak is a more stable alternative to the SUV-max, and is found by averaging the concentration values around the SUV-max. [28]. The metabolic tumor volume is defined as the parts of the tumor that are most metabolically active. The TLG is defined as the SUV-mean multiplied by the metabolic tumor volume (MTV), resulting in a measure of glucose uptake in the entire tumor [28].

2.5.2 Computed tomography

Computed tomography (CT) is a structural imaging technique often used for diagnosis and to determine treatment response [24]. CT solves many problems with traditional X-ray imaging [27].

The CT scanner

The CT scanner is made up of a high-voltage generator, an X-ray tube, filters, a collimator, and detectors [26]. The X-ray tube consists of an anode and a cathode in a chamber, where the cathode is negatively charged, and the anode is positively charged. Under high temperatures, the cathode produces an electron cloud in a thermionic process. A high voltage accelerates these electrons towards the anode, and as they hit, they release their energy in the form of an X-ray.

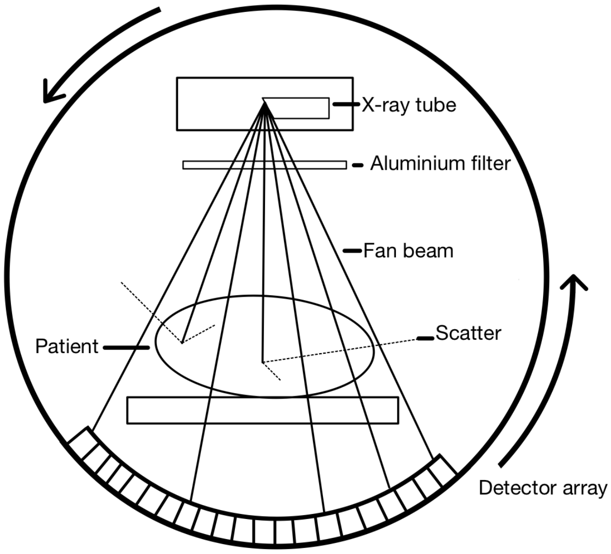


Figure 2.5.3: The CT scanner in the transversal plane. Adapted from [27] and [26].

The X-rays travel toward the filter, as seen in figure 2.5.3. The filter is present to remove X-rays of lower energy, called soft X-rays, as these have low penetrative abilities and will only contribute to the total radiation dose to the patient [26]. Filters like this are typically made of aluminum [26]. Further, a collimator is designed to focus the beam and limit the amount of scattering before the X-rays enter the patient. Limiting scattering by collimation increases the resolution of the final image [27]. From here, the radiation from the fan beam enters the patient as seen in Figure 2.5.3.

The detector catches the radiation that went through the patient's body. The energy of these X-rays reflects what tissues they passed through as different tissues have different attenuation coefficients [27]. The detector may be a xenon gas detector, where an incident photon ionizes the xenon gas. The ions create a current proportional to the incident photon's energy, collected as raw data. The detector can also be a solid-state detector, similar to the scintillation detector in PET scans, but with photo-diode instead of PMTs [27].

Similar to PET, during CT imaging, the patient is passed through a circular detector designed to capture every angle of the body in the final image [27]. The CT scanner produces transversal image slices by rotating the detector around the patient [27], as seen in Figure 2.5.3. The resulting slices are stacked to form a three-dimensional image of the patient. CT images allow a complete reconstruction of the patient's body that can be viewed from all angles [27].

Image reconstruction

When the X-rays hit the detector, their energy and position are recorded as raw data. The detector can measure how much the beam is attenuated to get the ray sum [27]. The ray sum is then correlated to the ray's position, creating an attenuation profile. All attenuation profiles can be projected onto a matrix in the process of back projection [27]. Over many rounds of back projection, the different views with their attenuation profiles are projected onto the matrix, and eventually, an image should form. Back projecting over many small angles tends to create shadows or artifacts in the final image. The artifacts can be removed or reduced by applying a mathematical filter to the data before back projecting it onto the matrix [27]. The spatial resolution of CT images is 0.5–0.625mm [32].

2.5.3 Head Neck cancer and PET-CT

When combining the functional imaging of PET with the structural imaging of CT, a more advanced imaging technique arises. The PET scan lights up hot spots where there might be cancerous cells, and CT pinpoints the area structurally in the body with high precision [27]. The result is a three-dimensional image of the exact anatomy with visible potential cancerous regions. PET/CT is a powerful tool for diagnosing, staging, and post-therapy monitoring of cancer patients [31].

2.6 Radiomics

Previously in Section 2.3, the PET parameters SUVmax, TLG, and MTV were described. These parameters, derived from PET images, express the tumor uptake of sugar, which reflects the tumor metabolic activity and the tumor size [33]. In addition to these parameters, other tumor characteristics can be reveal essential information about the tumor's shape, size, and texture. These characteristics typically lie in the tumor heterogeneity, meaning the complex variations in gene expression, biochemistry, histopathology, and structure in a malignant tumor [34]. Medical images can depict such heterogeneities [33].

Radioimics is the process of extracting features from a *region of interest* (ROI), typically a tumor or malignant lymph nodes in medical images [11]. The extracted features from the ROI can describe the tumor's shape, size, and texture, which is essential information when diagnosing and treating cancer [8]. The radiomics process starts with a medical image of a patient with a tumor. The ROI is then identified in the image. A three-dimensional image of the tumor is then rendered. Features are then extracted from the 3D image, and these features can be stored in a table to form a dataset that can be used for modeling and prediction [11]. The flow chart for the radiomics process is shown in Figure 2.6.1. Radiomics can be used on any imaging modality; magnetic resonance, CT, or PET images [11]. For head and neck cancer PET/CT is the preferred image modality [31].

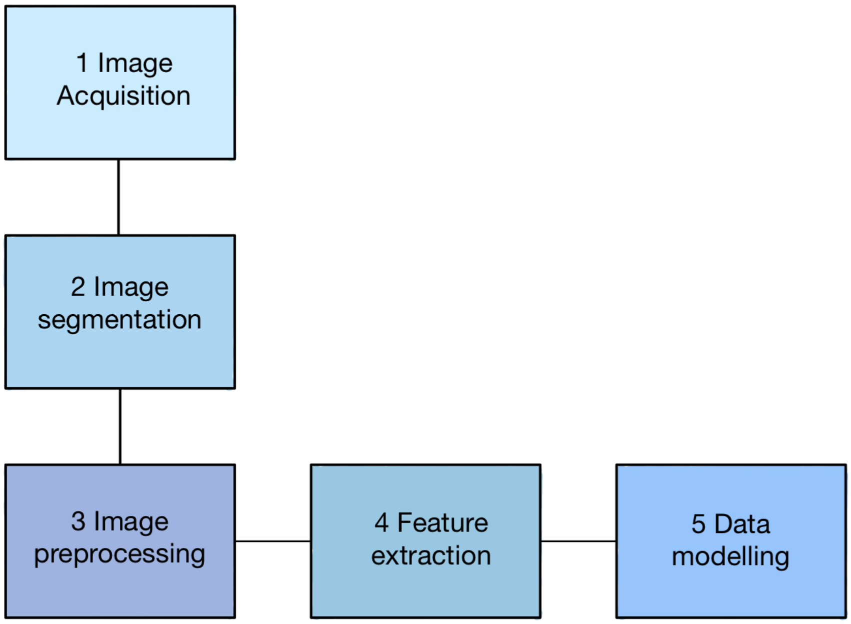


Figure 2.6.1: A flow chart for the radiomics process. Adapted from [12].

2.6.1 Image segmentation

Image acquisition is the first step of the radiomics process, as seen in the flow chart in Figure 2.6.1. When the image is acquired, the next step is to define the tumor and, thereby, the region of interest (ROI) in the image. In the case of metastases, identifying all lesions is necessary, resulting in multiple regions of interest [11]. The region of interest is the region in which the tumor is located. If the image is three-dimensional, the ROI is a volume or VOI. The ROI defines the part of the image where all the features are extracted. The extracted features can be divided into histogram-based, texture-based, and shape-based.

2.6.2 Image preprocessing

Before feature extraction can be performed, the images needs to be preprocessed, by discretizing the intensity values of the images. Intensity discretization is the process of grouping the image intensity values into bins. This can be done in two ways. The first one groups the intensity values into a fixed number of bins, N_g, where each intensity value is assigned to a bin depending on its magnitude [12]. The second method defines a bin width, ω_b. The first bin starts at the smallest intensity values and holds all the intensity values up to ω_b. [12]. The bins from either method form a histogram.

2.6.3 Image features

Feature extraction is step 4 in the radiomics process, illustrated by the flow chart in Figure 3.4.1. Image features can be extracted from the image by using statistical operations on the voxel gray intensity values [11]. The medical image consists of voxels of different gray values. Statistical measures can be calculated from these gray levels, which is the basis of the radiomics features. All formulas for the features described in this section can be found in the Pyradiomics documentation [35].

Histogram features

Representing a digital image by its histogram allows statistical information about that image to be extracted through mathematical operations [12]. A histogram of a digital image shows the distribution of gray-level values of the pixels in the image. The histogram depends on the *intensity discretization*, as explained in Section 2.6.2.

Using statistical methods on the histogram results in descriptive features of the image, such as *mean intensity*, *intensity variance*, *minimum* and *maximum intensity*. *Median intensity* is the median value of the grey level distribution, and *mode intensity* is the most common intensity value in the histogram. These properties are first-order statistical properties because they are based on single voxel values [36]. *Skewness* and *kurtosis* are related to the shape of the distributions. The

Table 2.6.1: The equations for finding the Compactness 1 and 2, the spherical disproportion, the sphericity and the asphericity. All equations for all radiomics features are found in the Pyradiomics documentation.

Feature	Equation
Compactness 1	$F_{morph.comp1} = \frac{V}{\pi^{1/2}}$
Compactness 2	$F_{morph.comp2} = 36\pi \frac{V^2}{A^3}$
Spherical disproportion	$F_{morph.sph.dispr} = \frac{A}{(36\pi V^2)^{1/3}}$
Sphericity	$F_{morph.sphericity} = \frac{(36\pi V^2)^{1/3}}{A}$
Asphercity	$F_{morph.asphericity} = \left(\frac{1}{36\pi}\frac{A^3}{V^2}\right)^{1/3} - 1$

asymmetry of the intensity distribution is described by the skewness. The kurtosis of the histogram reflects how tailed the distribution is relative to a normal distribution [36]. Further, the *entropy* and the $uniformity$ describe the distribution of values among the bins.

Shape-based features

Morphological features are shape-based features that describe the tumor's geometrical properties. These properties include *volume*, *surface area*, *elongation*, *flatness*, *compactness*, the *center of mass shift*, and *diameter* in multiple directions [35].

The surface area of the tumor is mapped out by triangles lining the surface of the entire ROI. From the surface area, the volume can be calculated. For ROIs of 1000 voxels and more, the volume can also be found by pixel counting. Pixel counting is not recommended for smaller ROIs (10-100 voxels), as it tends to overestimate the volume of the tumor [12]. From the surface and volume, the surface-to-volume ratio is also easily found by dividing surface by volume.

In addition to volume and surface, some features describe the tumor's shape. Multiple features represent the tumor's *sphericity. Compactness 1* and 2, *spherical disproportion*, *sphericity*, and *asphericity* are all features concerning the roundness of the tumor [12]. The center of mass shift describes the shift between the center of the ROI centroid and the center of the intensity-weighted ROI.
The *maximum 3D diameter* is the distance between the two vertices in the surface mesh that lay furthest away. This distance describes the extent of the tumor. Principal Component Analysis (PCA) on the three-dimensional image gives three

eigenvectors with corresponding eigenvalues that provide information about the tumor's shape. The eigenvalues are typically coined $\lambda_{major} > \lambda_{minor} > \lambda_{least}$ [12]. The *Major Axis Length*, λ_{major}, is the vector and eigenvalue that explains most of the variance in the ROI. The *minor axis length*, λ_{minor}, describes the direction of the ROI with the second most variance. The *Least axis length*, λ_{least} is the direction along the ROI that is the shortest. The *elongation* and $flatness$ can be found from the eigenvalues derived from PCA. The equations for elongation and flatness are found in Table 2.6.2.

Table 2.6.2: The definitions of elongation and flatness.

Feature	Equation
Elongation	$F_{morph.pca.elongation} = \sqrt{\frac{\lambda_{minor}}{\lambda_{major}}}$
Flatness	$F_{morph.pca.flatness} = \sqrt{\frac{\lambda_{least}}{\lambda_{major}}}$

Further, the *integrated intensity*, called total lesion glycolysis (TLG) for PET images, is the average intensity of the gray levels in the image multiplied by the tumor volume. Lastly, there are two features for describing the ROI *spatial autocorrelation*. Spatial autocorrelation is defined as the correlation between two neighboring voxels [37]. A high spatial autocorrelation refers to the voxel as similar to its neighbor, and vice versa for a low spatial autocorrelation. A low spatial autocorrelation in a tumor would mean that areas of the tumor differ from the rest of the tumor, indicating local abnormalities or deformities. The features that describe the spatial autocorrelation are the *Moran's I index* and the *Geary's C measure* [12]. All equations used for calculation of the radiomics features can be found in the Pyradiomics documentation [35].

Texture features

The texture in an image can be described by the absolute gradient [12]. The absolute gradient quantifies the difference of pixel values in an image. The absolute gradient reaches its maximum if two neighboring pixel values are black and white. This gradient is used to construct a second-order histogram for the image, the *Gray-Level Cooccurence Matrix* (GLCM) [36].

The GLCM describes how often two gray values are found on neighboring voxels (voxels with a set distance) in the image along a particular direction. The GLCM has dimensions N_g x N_g where each gray level holds a spot in the columns and the rows of the GLCM. The GLCM can find texture in the image by comparing intensity values. Many different features can be extracted from the GLCM, including *averages*, *variances*, and *entropies* [12]. The entropy reveals the randomness of the gray levels. The *angular second moment*, similar to the $uniformity$, describes how homogenous the image voxel distribution is. The *contrast* and *dissimilarity* give higher values to more different gray levels. The *inverse difference* does the opposite and gives lower values to high-contrast voxels, measuring homogeneity. The autocorrelation was described for morphological features above, and the explanation also asserts itself for texture features. The GLCM also has some clustering features, *cluster tendency*, *shade*, and *prominence*, which describe the clustering of the grey levels in the voxels.

Another similar texture matrix is the *Gray-Level Run-Length Matrix* (GLRLM). The GLRLM quantifies the occurrence of consecutive voxels with the same grey level in a given direction of the image [12]. The GLRLM has dimensions N_g x N_r where N_r is the maximum run length for grey level i in direction m [12]. In the top left quadrant of the GLRLM, the values represent short run lengths of low grey levels, and in the bottom right quadrant, the values represent long run lengths of high grey levels [12]. Features are extracted based on the short runs, the long runs, the low grey levels, and the high grey levels from all matrix quadrants. The $non-uniformity$ of the grey levels and run lengths are extracted into multiple features, and lastly, the run *entropy* describes the randomness in the matrix [12].

The *Gray Level Size Zone Matrix* (GLSZM) describes how the gray levels occur in a zone of the ROI [12]. The GLSZM has dimensions N_g x N_z where N_z is the maximum size of the zone. The features of the GLSZM are similar to the GLRLM features, but instead of run length, the emphasis is on zone size. The upper left quadrant of the GLSZM contains values for low grey levels in small zones, and the bottom right quadrant contains high gray levels in large zones. In addition to the GLRLM-based features, the GLSZM has a *zone percentage*, which describes how many actual zones there are in the matrix compared to how many there could be (potential zones) [12].

Grey Level Distance Zone Matrix (GLDZM) is similar to the GLSZM but incorporates the voxels' distance from the ROI edge [12]. The voxels in a distance zone have the same grey level and distance from the ROI edge. The dimensions of the GLDZM are N_g x N_d, where N_d is the maximum distance a voxel can be from the border of the ROI. The GLDZM requires a grey-level zone map, which the GLSZM generates, and a distance map of the ROI. The top left quadrant of the GLDZM holds values for zones of low grey levels occurring close to the ROI border, and the bottom right quadrant represents zones of high gray levels occurring towards the center of the ROI. The features extracted from the GLDZM are similar to the GLRLM features, but the distance to the ROI border is emphasized instead of run lengths.

The next texture matrix is called *Neighborhood Grey Tone Difference Matrix* (NGTDM). The NGTDM focuses on finding the difference between a voxel grey value and its neighbors [12]. By considering voxels with neighbors on all sides, the difference between the voxel grey level value and the mean of the neighboring voxels can be calculated [12].

The features extracted from the NGTDM are *coarseness*, *contrast*, *busyness*, *complexity*, and *strength* [12]. Coarseness is a measure of change in the intensity of different areas of the image. The contrast measures how the intensity changes throughout the image, giving a high value for images with abrupt changes in intensity. The busyness describes the heterogeneity of the image, meaning neighboring voxels of very different intensities are identified as busy. Complexity and strength describe how complex, or non-uniform, the texture is [12].

The last texture features are derived from the *Neighboring Grey Level Dependce Matrix* (NGLDM). The NGLDM also uses the concept of voxel neighbors, similar to the NGTDM [12]. However, the NGLDM finds the connection between the center and neighboring voxels if their grey levels fit a set criterion. The result is a matrix with the dimensions N_g x N_n, where N_n is the maximum number of dependencies one gray level has. The features extracted from the NGLDM are similar to the GLRLM, but the dependence between voxels is emphasized instead of the run lengths [12]. All equations for all features extracted by the radiomics algorithm are found in the Pyradiomics documentation [35].

2.7 Machine learning

Machine learning (ML) is a type of artificial intelligence that allows a computer to learn patterns and relationships in data and use them to make a prediction [1]. Within machine learning, there are three main branches, called supervised, unsupervised, and reinforcement learning. With supervised learning, the data has a predetermined label, and the algorithm attempts to identify data patterns that can distinguish the different labels [1]. In unsupervised learning, the data has no label, and the algorithm aims to identify subgroups or patterns within the data without a predetermined output [1]. The last branch, reinforcement learning, is where an agent makes decisions and receives feedback to create a sequence of right decisions [1]. In this thesis, the methods of supervised learning are the ones used.

2.7.1 Fitting a machine learning model

When feeding data to an ML model, it is commonly split into training and test data, typically denoted X_{train} and X_{test}. This allows for evaluating the model's performance by training it on one part and testing it on a different, unseen part of the data. The model's performance on unseen data determines its generalizability and, therefore, its performance [1]. The target variables, denoted y_{train} and y_{test}, are the ground truth that the model's prediction is compared to [1]. The prediction can be a binary classification problem, a multi-class problem, or a regression problem. This thesis deals with binary classification.

When training on a data set, a model attempts to learn all the valuable information to make the correct prediction [1]. If the model trains too much on the data, it can adapt to the noise and randomness in the dataset, as seen in Figure 2.7.1, which weakens the model's ability to predict well on unseen data. This is called overfitting. Training a machine learning model requires a balance between learning and overfitting, often called the bias-variance trade-off [1]. The model should be complex enough to pick up most data patterns without memorizing the training data's noise.

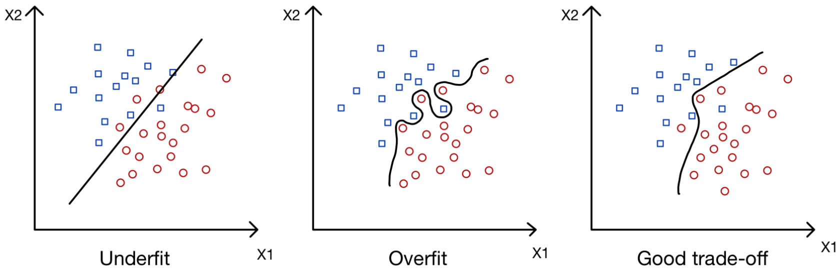


Figure 2.7.1: A graphical representation of one underfitted, one overfitted, and one well-adapted model plotted in the feature space of feature X_1 and X_2. Note that class 0 and class 1 are here represented by blue and red data points, respectively. Adapted from [1].

The bias-variance trade-off is illustrated in figure 2.7.1. The left model has a linear decision boundary that splits the features by a straight line, which in Figure 2.7.1 results in low variance and high bias This means the model needs to learn more of the information in the data, or the model is underfitting. The second model has high variance and low bias, meaning it memorizes some noise and randomness in the training data, lowering its ability to predict well on unseen data. This model is overfitting on the data set. The last model has the optimal bias-variance trade-off, where it learns most of the valuable information in the training data and can use this to predict well on the test data [1].

Throughout machine learning history many different models and algorithms have been introduced, including the Decision Tree, the Random Forest, the XGBoost, the Histogram Gradient Boosting, and the interpretable models from the iModels package [2].

2.7.2 Decision Trees

Decision trees are models that use a tree-like structure to represent a series of decisions and their possible consequences [1]. Each node in the tree represents a decision, and each leaf node represents a possible outcome. As the model is trained on more data, it becomes better at predicting outcomes based on input features. Decision trees are also the easiest models to interpret, as their structure is comprehensible to the brain. In Figure 2.7.2, a simple decision tree structure for weather someone should get a dog or not is illustrated.

A decision tree algorithm finds the most optimal feature split in a dataset. The most optimal feature split can be found by calculating the impurity of the potential

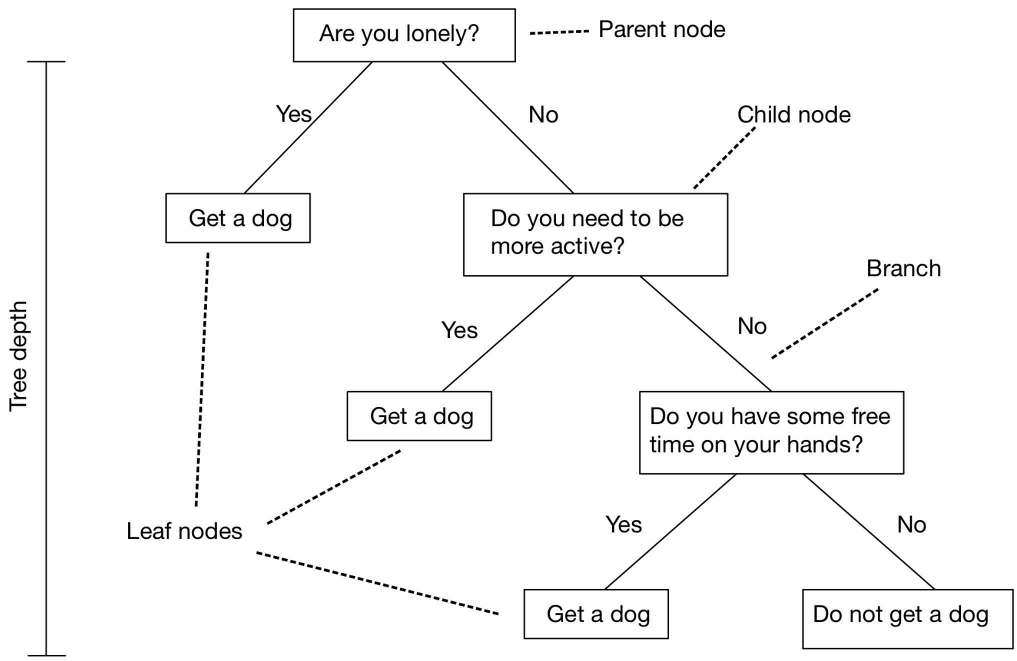


Figure 2.7.2: An illustration of a decision tree, where the classes are split into weather or not someone should get a dog.

child nodes [1]. The impurity can be defined in three ways; entropy, Gini, and classification error. The entropy is calculated for each feature, and is given by

$$I_H(t) = -\sum_{i=1}^{c} p(i|t) log_2 p(i|t), \tag{2.7.1}$$

where $p(i|t)$ is the fraction of samples in class i in node t, and c is the number of classes. When there is an even mix of samples from all classes in the node, the impurity is at its maximum value of 1. If the node contains samples from one class only, the impurity is 0 and the node is called a pure leaf node [1]. The Gini impurity, $I_G(t)$ can be found by

$$I_G(t) = \sum_{i=1}^{c} p(i|t)(1 - p(i|t)) = 1 - \sum_{i=1}^{c} p(i|t)^2 \tag{2.7.2}$$

The Gini impurity reaches its maximum value, 0.5, when the classes are perfectly mixed in the node. The last definition of impurity is the classification error, $I_E(t)$, which can be found by

$$I_E(t) = 1 - max\{p(i|t)\} \tag{2.7.3}$$

The most used impurity measures are the Gini and the entropy, and these definitions yield very similar results. Classification error is not used very often [1]. From the impurity, the *information gain (IG)* can be defined as

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right}) \tag{2.7.4}$$

where D_p is the dataset in the parent node, f is the feature, I denotes the impurities, N_p, N_{left}, and N_{right} are the number of samples in the parent, the left and right child node respectively, D_{left}, and D_{right} are the datasets in the left and right child nodes. The IG is lowest when the child nodes are a perfect mixture of the classes, and highest when the child nodes are pure. The model always aims to split at the feature and value that creates the highest information gain [1].

The algorithm goes through the features and values, one by one, to find the one with the highest information gain, which creates the first feature split. The process repeats on the child nodes until all nodes are pure. This creates an overfitted decision tree. To avoid overfitting, a maximum depth can be set to stop the tree from overgrowing. The best value for the maximum depth depends entirely on the dataset and can be tuned to fit the data [1]. When the model has undergone training, it can be used to classify new samples. When a new sample is classified, it is tested against all the conditions of the decision tree until it ends up in a leaf node. In the leaf node, the sample gets a predicted class label based on the most common class in that node in the process of majority voting [1].

One of the significant advantages of decision trees is their interpretability. Decision trees are inherently logical in their structure, and can be understood easily by the human brain. The decision tree algorithm is the base of many other ML algorithms, including the Random Forest, XG Boost, and all of the models from the iModels package [2].

2.7.3 Random Forest

The Random Forest algorithm is a forest of decision trees. The trees in the random forest are sometimes referred to as estimators. Algorithms that combine multiple other classifiers into one classification process are called ensemble algorithms [1], and Random Forest is an ensemble of Decision Trees. The idea behind ensemble learning is to combine many weak classifiers into one strong classifier [1]. The algorithm grows multiple decision trees, allowing for a more complex decision tree model without overfitting on the training data [1].

The algorithm starts by drawing a random set of samples and features with replacement from the original dataset. These subsets are used to generate and train multiple decision trees, and this process is called bootstrapping [1]. When classifying a new sample, every tree in the forest makes a prediction. The final class label is determined in a majority voting, which is called aggregation [1]. Bootstrapping and aggregation together are called bagging. Bagging makes the Random Forest algorithm more stable than single decision trees because the model is less sensitive to random variations in the training dataset [1]. Bagging promotes a model with high generalizability.

2.7.4 XGBoost

XGBoost is another ensemble algorithm. The name, XGBoost, stands for Extreme Gradient Boosting, as this algorithm is a further developed gradient boosting algorithm [38]. The name gradient stems from the method of optimization called gradient descent, which is used for both XGBoost [38] and the HistGradientBooster [39] explained below. XGBoost works, similarly to other boosting algorithms, by iteratively growing trees. Each new tree is designed to correctly classify the samples that were misclassified in the previous iteration. The prediction of a sample is made by combining each prediction f_k from K different trees [38]. The total prediction, $\hat{y}_i$ for a sample i in dataset X can be found by equation 2.7.5,

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{X}_i) \tag{2.7.5}$$

where, f_k corresponds to the prediction from a decision tree q. Each tree q is weighted by a weight ω [38]. The optimization objective of an XGBoost model is finding a value for ω that minimizes the loss function, L^t. The loss function describes the prediction error and can be expressed as in 2.7.6,

$$L^t = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{t-1} + f_t(\mathbf{X}_i)) + \Omega f(t), \tag{2.7.6}$$

where t denotes the t^{th} iteration, i denotes the sample, l measures the difference between the ground truth y_i and the prediction of last iteration y_i^{t-1} added to the prediction of the current iteration f_t. $\Omega f(t)$ is the regularization term of the loss function. XGBoost uses both $L2$ and $L1$ regularization [38]. $L2$ regularization keeps weights from getting too large by punishing large weights. $L1$ regularization deals with the complexity of the model and works as a feature selection method for datasets with many features. Optimizing the loss is done through gradient descent [38]. The idea behind gradient descent is that there is a parameter combination that minimizes the prediction errors. XGBoost uses first and second-order gradients meaning it knows in which direction the gradient is increasing or decreasing [38].

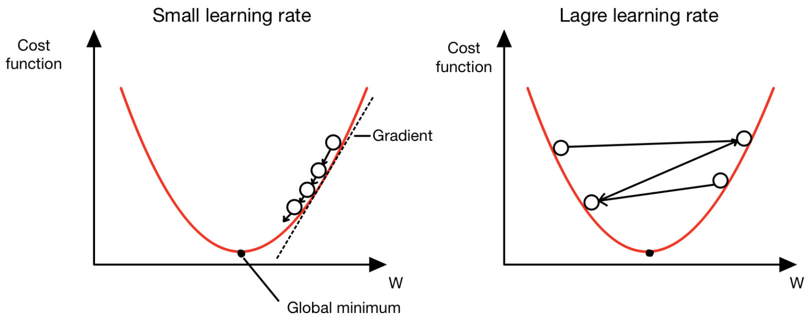


Figure 2.7.3: Gradient descent is used to find the global minimum of the cost function. Adapted from [1].

The ciricles indicate the weight of the model, which is slowly approaching the global minimum in the left diagram of Figure 2.7.3. The arrows in Figure 2.7.3 represent the learning rate, where a high learning rate, shown in the right diagram, shows that the model overshoots the global minimum of the loss function. The loss function is illustrated by the red lines. The learning rate is a hyperparameter in XGBoost that needs to be tuned to the specific dataset. The gradient of a function is its first derivative, and the hessian of a function is its second derivative [40]. Including the gradients and hessians in the loss function results in equation 2.7.7 and 2.7.8,

$$L^t(q) = -\frac{1}{2}\sum_{j=1}^{T} \omega_j^* + \gamma T \tag{2.7.7}$$

$$\omega_j^* = -\frac{\sum_j g_i}{\sum_j h_i + \lambda} \tag{2.7.8}$$

$L^t(q)$ in equation 2.7.7, calculates the total loss of tree q over all T leaf nodes. The total loss can also be used as a score for the quality of tree q. Equation 2.7.8 computes the weight, w^*, of leaf node j by the gradient g_i and hessian h_i. The regularization term λ is applied to keep the model from overfitting. Equation 2.7.7 is also used to identify the best split during training [38].

2.7.5 Histogram Gradient Boosting Classifier

Histogram Gradient boosting classifier (HGBC) works similarly to XGBoost in many ways. However the HGBC discretizes the feature space by binning the data into a given number of bins, [39]. Figure 2.7.4 shows a graphical representation of binning.

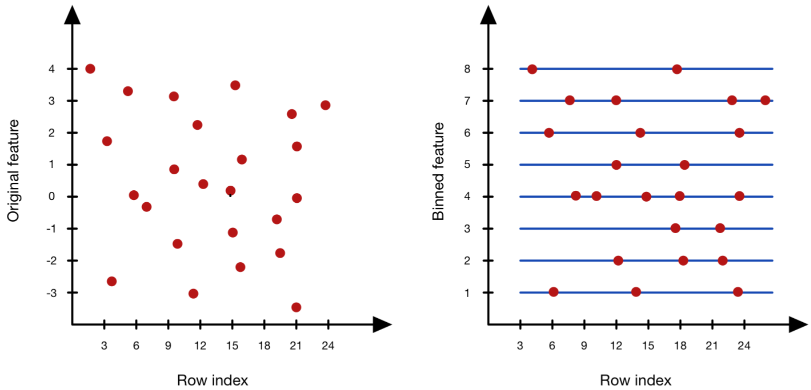


Figure 2.7.4: The binning of feature values into bins. The original feature values on the y-axis are replaced by the bin numbers. Adapted from [41].

Within each bin, the gradients of the samples are summed. The sum of the samples' gradients in each bin forms a histogram. Two separate histograms can be made for the gradients and hessians, respectively. Like XGBoost, HGBC finds the best feature split by minimizing the gradient of the child nodes, but because the features are binned, the number of potential splits to calculate is far less than that of an unbinned feature space [39]. This makes HGBC very computationally efficient.

2.7.6 iModels

The iModels package, proposed by Singh et al. [2], is a set of rule-based models aiming to make interpretable ML decision tree models while maintaining a state-of-the-art performance. The package contains many models, including the Fast Interpretable Greedy Sums (FIGS), the Hierarchical Shrinkage Tree (HSTree), and the Boosted Rules Tree.

Fast Interpretable Greedy-Tree Sums

Fast Interpretable Greedy-Tree Sums (FIGS) is an algorithm that grows a flexible number of trees simultaneously [42]. FIGS builds a decision tree, splitting nodes at the most informative criteria. The algorithm can keep adding nodes to an existing tree or start a new tree at every iteration, which avoids repeated splits and makes the models more compact [42]. An illustration of the FIGS algorithm compared to the conventional Decision tree is presented in Figure 2.7.5.

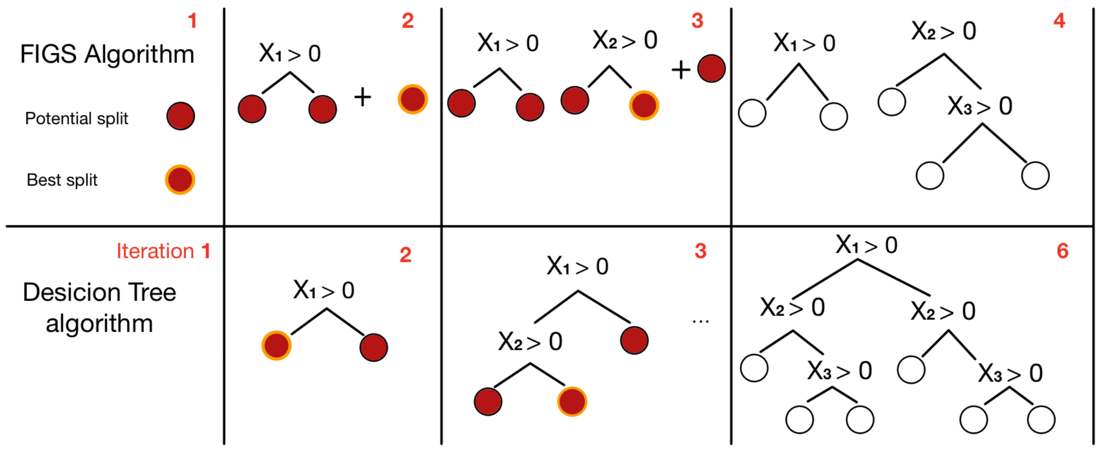


Figure 2.7.5: The FIGS algorithm vs the Decision tree algorithm on a demonstration function $y = \mathbb{1}_{X_1>0} + \mathbb{1}_{X_2>0} \cdot \mathbb{1}_{X_3>0}$. The FIGS algorithm requires three splits across two trees to implement this function, whereas the Decision tree algorithm requires five splits. Adapted from [42]

In Figure 2.7.5, the algorithms are built on a function $y = \mathbb{1}_{X_1>0} + \mathbb{1}_{X_2>0} \cdot \mathbb{1}_{X_3>0}$. This function consists of two components that can be implemented by two trees using the FIGS algorithm [42]. The first tree contains one split $X_1 > 0$, and the second tree contains two splits $X_2 > 0$ and $X_3 > 0$, as seen in Figure 2.7.5 in the top right. Implementing the same function using the Decision tree algorithm requires five splits, many of which are repeated, as seen in Figure 2.7.5 in the bottom right.

The FIGS algorithm has the advantage that it avoids repeated splits. It keeps the model smaller and more compact while keeping the complexity of a decision tree with the same rule set [42]. Avoiding complicated decision trees promotes interpretability and allows for clear visualizations.

Hierarchical Shrinkage Tree

The Hierarchical shrinkage (HS) algorithm works by regularizing a tree-model's prediction [43]. Unlike FIGS, it does not affect the tree structure, but instead, it affects the prediction. HS is a post-hoc algorithm. For a given tree model f, the

HS regularizes the prediction on each tree leaf node [43]. It can be applied to any tree algorithm.

Boosted Rules Classifier

The Boosted Rules Classifier (BR) is another algorithm proposed by Singh et al. [2]. The algorithm uses the Adaboost algorithm to fit a set of rules sequentially [44]. Adaboost works by initializing a decision tree stump, C_1, which makes a decision boundary as seen in figure 2.7.6 subfigure a). Before the first iteration, all samples are equally weighted, whereas for iteration 2, the two misclassified samples from the previous iteration are assigned a larger weight. Iteration 2 grows a decision stump, C_2, that defines a new decision boundary, as seen in subfigure b) in Figure 2.7.6, attempting to correctly classify the previously misclassified samples. The process repeats for the third iteration, with a larger weight assigned to the samples misclassified in during iteration 2. The result of three rounds of boosting is a model shown in the bottom right diagram of Figure 2.7.6. A majority vote decides the final prediction of the model [1].

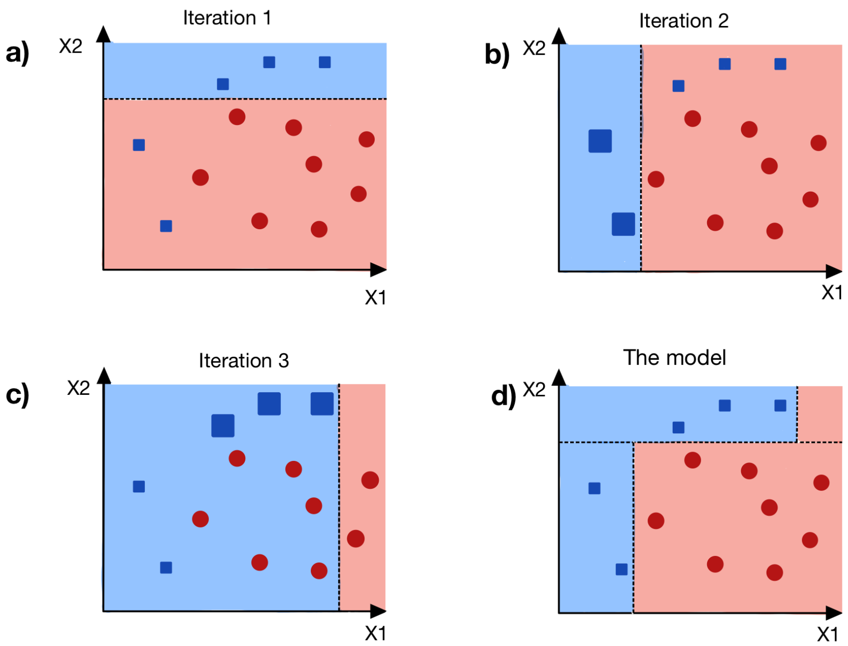


Figure 2.7.6: The Adaboost algorithm. The red circles denote samples of class 0, and the blue squares denote samples of class 1. The shaded areas are the different decision boundaries, where a red area shows that the samples within that area were predicted as class 0, and vice versa for the blue area. Adapted from [1]

2.7.7 Model Performance

Measuring model performance is crucial for all ML models. Performance metrics are needed for tuning and optimization of the models, where the criteria of optimization is to maximize a certain metric. Many metrics measure the model's performance, including the accuracy, the F1 score for class 1 and class 0, the receiver operating curve, and the Matthews Correlation Coefficient [1].

Accuracy

The accuracy of a model is the simplest form of metric. The accuracy compares the predicted outcome to the ground truth by dividing the sum of the correct predictions by the total number of samples. A Confusion Matrix can visualize the predicted and true classes, shown in Figure 2.7.7.

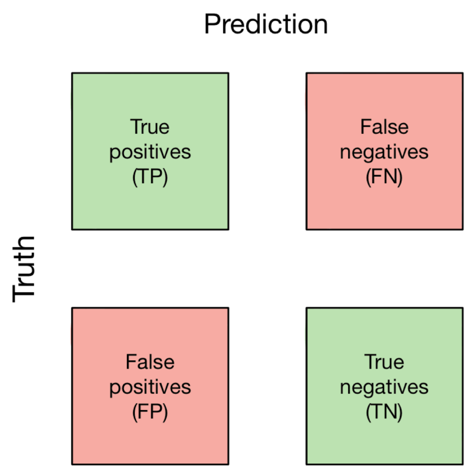


Figure 2.7.7: A Confusion Matrix gives a visualization of the model's right and wrong classifications. Adapted from [1].

Class 0 and class 1 are also referred to as the negative and positive class. In Figure 2.7.7, class 0 is referred to as the negative class, and class 1 is referred to as the positive class. All performance metrics described below are based on the information given by the Confusion matrix in Figure 2.7.7. The accuracy, ACC, is given by

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} \tag{2.7.9}$$

where TP and TN are the true positives and negatives, and FP and FN are the false positives and negatives, respectively.

F1 score

The F1 score is defined through the precision and recall metrics [1]. The precision of a model shows the proportion of the predicted positives that were true positives, as seen in equation 2.7.10. However, recall shows the proportion of the positives the model could predict correctly, as shown in equation 2.7.11, given by

$$PRE = \frac{TP}{TP + FP} \tag{2.7.10}$$

$$REC = \frac{TP}{FN + TP} \tag{2.7.11}$$

From here the F1 of the positive class can be defined as

$$F1 = 2\frac{PRE \times REC}{PRE + REC} \tag{2.7.12}$$

The F1 score defeats issues with using precision and recall separately. Optimizing a model by maximizing the recall increases the probability of the model predicting false positives, and maximizing the precision increases the probability of predicting false negatives. The F1 score combines both of these metrics to create a more stable and balanced metric for the model. [1]. To get the F1 for class 0, the equations for precision and recall are changed to

$$PRE = \frac{TN}{TN + FN} \tag{2.7.13}$$

$$REC = \frac{TN}{FN + TN} \tag{2.7.14}$$

These new definitions are plotted into the equation 2.7.12 to find the F1 score for class 0.

Receiver operating characteristic

The receiver operating characteristic visualizes the true positive rate (TPR) and the false positive rate (FPR) in a graphic representation [1]. The optimal model would have TPR=1 and FPR=0, like the *perfect performance* line in Figure 2.7.8. The line across the diagonal of the plot represents random guessing, and a line under the diagonal would then be worse than random guessing [1]. From the TPR/FPR graph, the area under the curve (ROC-AUC) can be calculated, giving a performance estimate for a model.

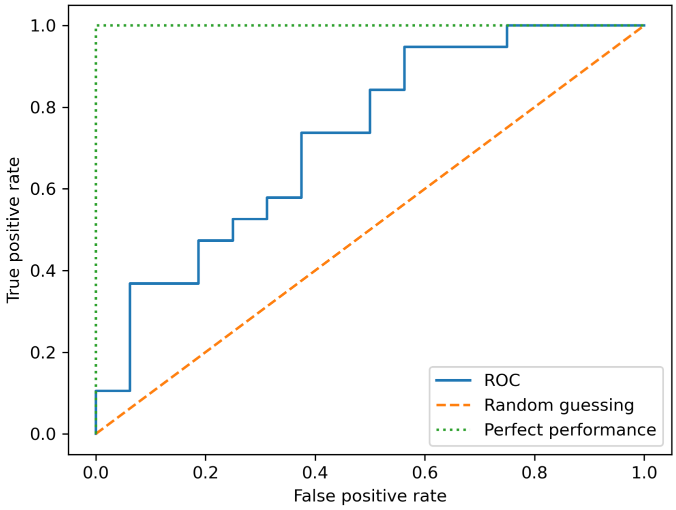


Figure 2.7.8: The ROC curve. Adapted from [1].

Matthews Correlation Coefficient

The Matthews Correlation Coefficient (MCC) ranges between $[-1, 1]$ [45]. The MCC score is less vulnerable for class imbalances than other metrics such as accuracy [45]. If 90% of the samples in a dataset belong to class 1, predicting all samples as class 1 would give an accuracy of 90% [1]. This example illustrates the importance of using several performance metrics. The MCC can be found by

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}. \qquad (2.7.15)$$

Chapter 3

Materials and Methods

This thesis investigated the performance of seven ML classifiers on two head and neck datasets. The datasets were collected from Oslo University Hospital (OUS) and the Maastricht University Medical Center (MAASTRO). The dataset from OUS was used to train and validate the models, and the MAASTRO data was used for external testing. Radiomics was used to extract features from the images of each patient. A feature selection algorithm, RENT, was then used to select the most important features. Seven ML classifiers were tuned by Optuna[1], validated on the OUS data, and tested on the MAASTRO data.

3.1 The datasets

The OUS dataset contained patients with head and neck cancer treated at Oslo University Hospital (OUS) from January 2007 to December 2013 [13]. Similarly, the MAASTRO dataset contained patients treated at the MAASTRO clinic from January 2008 to December 2014. There were originally around 400 patients receiving treatment for head and neck cancer at OUS during this time, and since this thesis follows the work of Moan et al. [13], the same inclusion criteria were used. Patients diagnosed with cancer in the oral cavity (cavum oris), oropharynx, hypopharynx, and larynx whose radiotherapy treatment plans were based on 18FDG-PET/CT were included. Patients with oropharyngeal cancer and unknown *HPV-status* were excluded, as well as patients without contrast-enhanced CT images along with the PET images. In total, 139 patients from the OUS dataset and 99 patients from the MAASTRO dataset fit the inclusion criteria.

[1] https://github.com/optuna/optuna

Clinical information was collected from these patients: age, gender, tobacco use (pack years), Charlson comorbidity index, tumor site, cancer stage, and HPV status. Two response variables were used in the datasets, namely *Disease-free Survival* (DFS) and *Overall survival* (OS), as described in table 3.1.1. A value of 1 in either of these response variables indicated an event, relapse, or death.

Table 3.1.1: Distribution of the targets for the OUS and MAASTRO datasets. Here DFS refers to Disease-free survival and OS refers to Overall survival.

Target	Type	OUS	MAASTRO
DFS	Binary Target		
Disease free (Class 0)		51.1%	40.4%
Local, regional or metastatic failure or death (Class 1)		48.9%	59.6%
OS	Binary Target		
Survive (Class 0)		59.0%	46.5 %
Death (Class 1)		41.0%	53.5 %

The clinical features used for this analysis are listed in Table 3.1.2. The two continuous features *age* and *pack years* are presented by their means for both the OUS and MAASTRO data. The pack-years feature describes the patient's tobacco use, where smoking 20 cigarettes (one pack) a day for one year grants you one pack-year. The remaining categorical features consist of four different cancer sites; *Cavum Oris*, *Oropharynx*, *Hypopharynx*, and *Larynx*. The tumor sites were outlined in Section 2.2. Patients who had HPV-related oropharyngeal cancer had a registered HPV status. The histologic grade and cancer stage were dichotomized as seen in Table 3.1.2. Three PET parameters were also included in the clinical dataset, the *SUV peak*, *MTV*, and *TLG*, which were all outlined in Section 2.5.1.

Table 3.1.2: Table of features in the clinical dataset. The continuous features are represented by their mean and standard deviation, and the categorical/binary features are represented by their distribution, for both the OUS and MAASTRO dataset.

Feature	Type	OUS	MAASTRO
Age (mean)	Continuous	60.2 ± 7.7	61.6 ± 9.5
Gender	Binary		
Female		23.0%	26.3%
Male		77.0%	73.7%
Cancer Stage (TNM8)	Binary		
I-II		51.8 %	19.2 %
III-IV		48.2 %	80.8 %
Tumor Site	Categorical		
Cavum Oris		7.9%	3.0 %
Oropharynx		65.6%	44.4%
Hypopharynx		11.5%	15.2%
Larynx		15.1%	37.4%
HPV related	Binary		
Yes:		58%	22.2%
No:		42%	77.8%
Pack Years (mean)	Continuous	25.0 ± 22.8	46.1 ± 47.7
Histologic grade	Binary		
High		69.1%	40.4%
Low/moderate		30.9%	59.6%
Charlson comorbodity index	Binary		
0		61.9 %	25.3 %
1-6		38.1 %	74.7 %
PET parameters			
SUV peak	Continuous	11.0 ± 5.4	11.2 ± 6.2
MTV	Continuous	11.9 ± 13.5	15.1 ± 10.9
TLG	Continuous	121.0 ± 194.7	109.9 ± 74.1

The patients were staged according to the tumor-node-metastasis (TNM) system [46].
HPV: Humanopapilus virus.
Charlson Comobidity index: degree of abnormality in cancerous cells.

The clinical features were preprocessed by encoding the categorical variables into binary variables. *Gender* was named *female*, where a value of 1 indicated female, and 0 indicated male. *Tumor site* was separated into the four categories shown in Table 3.1.2, where a value of 1 in one of these indicated that the patient had cancer at that particular site. The cancer stage, as shown in Table 3.1.2, was dichotomized into two categories *I-II* and *III-IV*, according to the tumor-node-metastasis system, version 8 [46]. The category *Cancer stage* had value 0 for patients in lower cancer stages (I-II), whereas it had value 1 for patients in the higher cancer stages (III-IV). The histologic grade describes how abnormal the cancer cells look under a microscope [47]. Histologic grade was dischotomized into two levels, *0* and *1-6*. In total, the clinical dataset contained 14 features, and will from here be referred to as dataset *D1*.

3.2 Software

For this thesis, Python 3.9.13 was used as the programming language of choice. The Jupyter Notebook IDE from Anaconda was used as the main programming tool. Pandas [48] was used for preparing and preprocessing the data. The DecisionTree, RandomForest, HistGradientBoosting classifiers, and the metrics were used through Scikit learn [49], whereas the FIGS, HSTree, and BoostedRules classifiers were used through the iModels package [2]. The XGBoost was used through a separate package, the XGBoost package [38]. Feature selection was done through RENT [50], and for tuning of the models, the Optuna framework [51] was used. Visualizations were done using Matplotlib [52], Seaborn [53], dtreeviz [54], and Microsoft Excel [55].

3.3 FDG PET/CT

A Siemens Biograph 16 was used to perform FDG PET/CT on all patients in the OUS dataset [13]. The tumor volume was then delineated by an experienced nuclear medicine specialist, based on the PET images. An oncologist refined the delineation of the gross tumor volume (GTV) based on the CT image and clinical information. The PET values were originally in the unit *Bq/ml* before the conversion to standard uptake value (SUV). An image from a patient in the OUS dataset is shown in Figure 3.3.1.

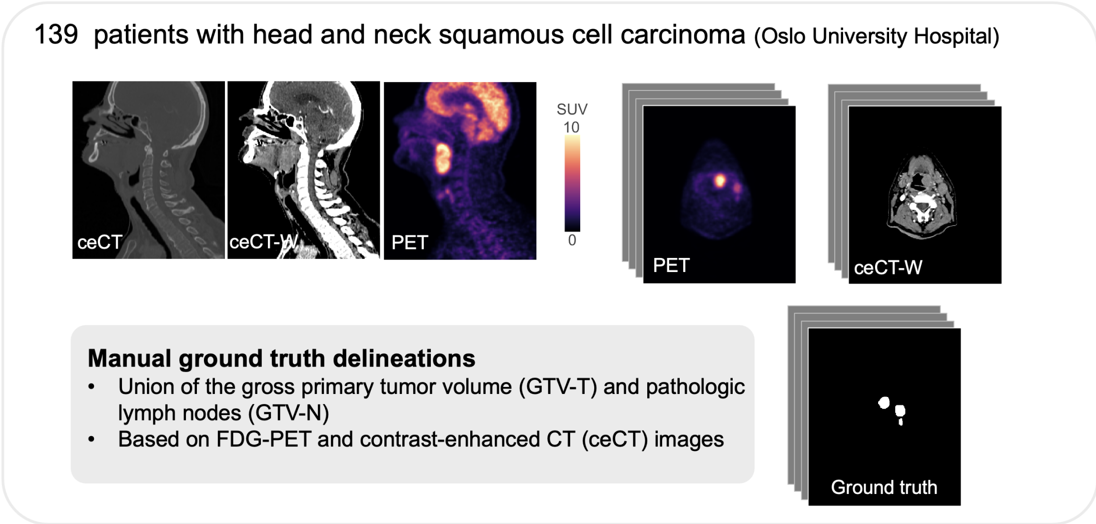


Figure 3.3.1: An example image from one of the patients at OUS.

3.4 Workflow

The workflow of this thesis is outlined in Figure 3.4.1. It consisted of 9 steps starting with data preprocessing. Further, a performance baseline model was established. Feature selection was performed through the RENT feature selection framework [50], where unimportant features were eliminated. The models were then tuned and validated on the training data from OUS, before their interpretability was assessed. Testing happened in step 8 of the analysis where the models were evaluated on the unseen data from the MAASTRO clinic.

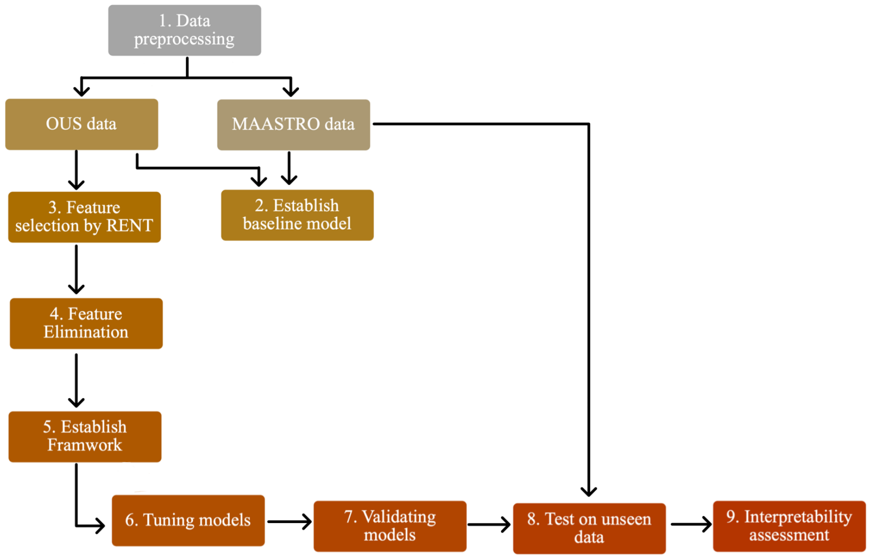


Figure 3.4.1: The workflow of this thesis illustrated by 9 steps. Note that the model selection was done using the OUS data, and the MAASTRO data was used for external testing.

3.5 Radiomics

Radiomics is a feature extraction technique that extracts firt order internsity, shape and texture features from medical images. The features extracted from radiomics are outlined in Section 2.6. The equations for all features are found in the Pyradiomics[1] documentation [35]. This thesis used radiomics to extract features from both the PET and the CT images.

From the radiomics process, 354 features were extracted (40 first order, 14 shape, and 300 texture features from the primary tumor). Radiomics was used through imskaper[2], which is an NMBU developed software based on PyRadiomics [35]. In addition to the radiomics features, imskaper extracts another 20 three-dimensional local binary pattern features [56]. In total, radiomics through imskaper extracted 374 features from the medical images. These 374 features formed a radiomics dataset, which will from here be referred to as dataset *D2*. A third dataset consisting of both the clinical and the radiomics features will be called dataset *D3*. An overview of the datasets is found in Table 3.5.1.

[1]https://pyradiomics.readthedocs.io/en/latest/index.html
[2]https://github.com/NMBU-Data-Science/imskaper

Table 3.5.1: The three datasets from OUS, together with a description, and their number of features.

Dataset	Description	Features
D1	Clinical features	14
D2	Features extracted from images by radiomics	374
D3	D1+D2	388

3.6 Feature selection

Using radiomics to extract features from images significantly increases the number of features in the datasets. High dimensional data causes models to easily overfit, which leads to poor results on unseen data [1]. Further, high dimensional datasets have lower interpretability, which is a key factor in the scope of this thesis. For these reasons feature selection is necessary, and it is step 3 of the flowchart in Figure 3.4.1.

Feature selection can be done in many ways, one of which is by the *repeated elastic net technique* (RENT) [50]. RENT trains an ensemble of models based on subsets of data, and can provide information on how frequently features are selected for all the models. Frequently selected features are assumed to be more important to the prediction, and features that were not selected are assumed to be less important. Feature selection is then performed based on the weight distribution of a feature across all models.

The agressiveness of RENT can be controlled through parameters τ_1, τ_2 and τ_3 [50]. These three parameters form conditions that determine if a feature is selected or not. The first condition sets a limit for the selection frequency of a feature. If a feature is selected more frequently than the limit τ_1, the first condition is fulfilled. The second condition τ_2 determines how many of the weights of a feature must have the same sign. The third condition is based on the Student's t-test, which ensures that the weights of the feature is consistently high with low variance across all models. Typically a significance value of $\alpha = 0.05$ is used, and therefore $\tau_3 = 0.975$ [50].

When running RENT, the user selects values for the regularization hyperparameters of L2 and L1 regularization for the underlying logistic regression model. The values used for RENT in this thesis are defined in Table A.1.1 in Appendix A.1.

Based on the parameters in Table A.1.1, RENT found the optimal values for the regularization through k-fold cross validation with 5 folds and 20 rounds. RENT was run in a brute-force manner, meaning it tests all combinations of hyperparameters from Table A.1.1, in order to find the best one. For each hyperparameter combination, RENT trains 100 models. All the 100 models selected features from the original dataset, with each model selecting a different combination. Features selected by all the models were considered most important, and features that were selected by none of the models were considered unimportant. The unimportant features were eliminated from the dataset.

Step 4 in the in the workflow in Figure 3.4.1, was feature elimination. RENT was run in a brute-force manner on the three datasets *D1*, the clinical data, *D2*, the radiomics data, and *D3*, the combination of D1, and D2, as described in Table 3.5.1. This was done for two targets; Disease-free survival and Overall survival. This resulted in three new subsets of data for each response. All features selected by RENT at least once were included in the datasets *DR1*, *DR2*, and *DR3*, and all features that were not selected by RENT were eliminated. All further analysis was based on the RENT-selected features in datasets DR1, DR2, and DR3 for OS and DFS.

3.7 Establishing a model framework

Establishing a framework was step 5 of the workflow in Figure 3.4.1. The scope of this thesis is to provide accurate predictions with models that can be interpreted and understood. Decision trees are the most obvious choice of algorithm for this, because of their inherent interpretability and visualization possibilities. Decision trees form the basis of many other, more complicated algorithms, which were all outlined in Chapter 2.7. The classifiers used in this thesis are the Decision Tree, Random Forest, XGBoost, and HistGradientBoosting , and the FIGS, HSTree, and the Boosted Rules from the iModels package, described in Section 2.7.

3.8 Tuning the model

Step 6 in the workflow in Figure 3.4.1 was tuning the ML models. There are several methods for tuning the ML model. To tune a model means determining the hyperparameters best suited for the dataset. ML models can have an array of different hyperparameter options and different effects of tuning. There are many different algorithms for tuning, one of which is Optuna.

The Optuna package[3] is a framework designed to optimize the hyperparameter combination of an ML classifier. The user defines an *objective* for optimization. The objective in this case is the classifier whose hyperparameters are to be optimized. Within the objective, the hyperparameters are defined within a plausible range for the given classifier. The classifier is created and fit with a hyperparameter combination. A scoring or validation method is defined. To optimize each model, Optuna creates a *study*. A study is the process of optimizing the objective, and is run in a user-defined number of *trials*. A trial is one execution of the objective function [51]. The study is created and optimized over a number of trials, and the hyperparameters that gave the best performing model are stored in the study object [51].

The efficient hyperparameter sampling and pruning mechanisms set Optuna apart from other optimization methods. For hyperparameter sampling, Optuna implements both *relational sampling* and *independent sampling*, where *relational sampling* utilizes the hyperparameter correlations and *independent sampling* samples each hyperparameter independently [51]. Pruning mechanisms are designed to terminate trials with unpromising results, to keep the algorithm from wasting time searching for optimal hyperparameter combinations in useless directions. Optuna features an *Asynchronous successive Halving algorithm* (ASHA) for early stopping based on ranking of previous trials.

Optuna version 3.1.0 was used for hyperparameter tuning in this thesis. All input hyperparameters for every classifier is found in Appendix B.1. The input hyperparameters were intentionally held at a lower level to avoid large complicated models, and to promote model interpretability. The objective of optimization was defined to be the *cross_val_score* from ScikitLearn [49].

[3]https://github.com/optuna/optuna

3.9 Validating the model with k-fold cross validation

Validating the models was step 7 in the flowchart in Figure 3.4.1. K-fold cross-validation is a type of validation method for machine learning models. The method splits the data into k folds [1]. In Figure 3.9.1, $k = 5$. During the first iteration, labeled 1 in figure 3.9.1, the first four folds are used for training the model, and the fifth fold is used to validate that model's performance. The second iteration uses folds 1, 2, 3, and 5 for training and the fourth for testing. The process repeats five times, giving five estimates, $M_1 - M_5$, for the performance. The model's overall performance, M, on the dataset is found by averaging the performance for every round [1].

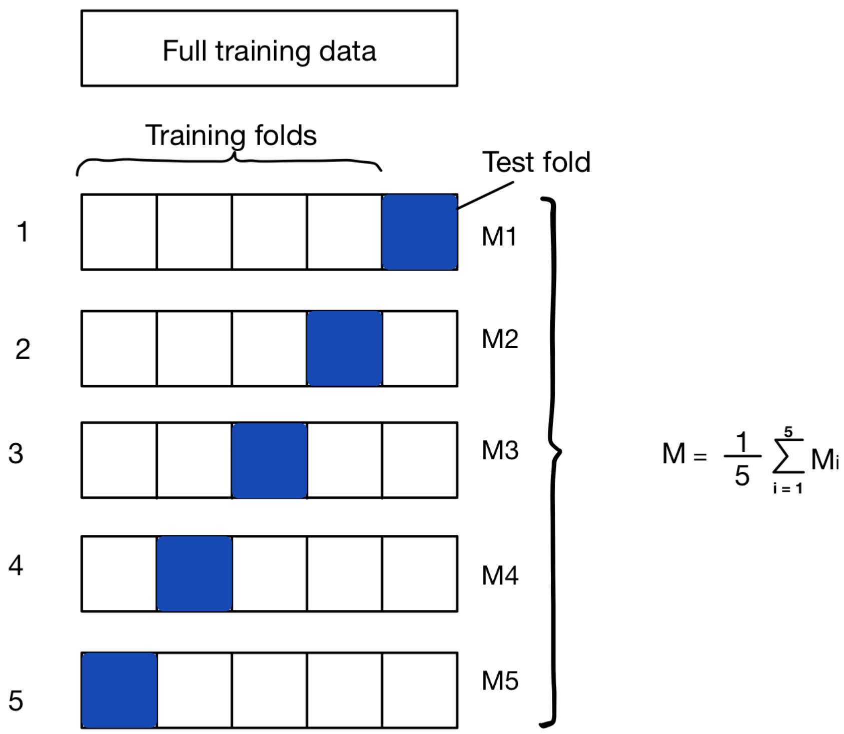


Figure 3.9.1: The process of k-fold cross validation with five folds, where the blue fold is the test fold, and the rest are the training folds. M1-M5 refer to the performance metrics that measure the performance of the classifiers on each data fold. Adapted from [1].

In this thesis, k-fold cross validation was used with five folds, over 100 repeats through the RepeatedStratifiedKFold cross validator from ScikitLearn [49]. The models were made for datasets DR1, DR2, and DR3 for OS and DFS respectively.

3.9.1 Measuring performance

Five performance metrics were used when measuring the model's performance. Accuracy, ROC score, F1 for the positive and negative class, and the Matthews Correlation Coefficient were described in detail in Section 2.7.7. Evaluating the model with multiple metrics gives a more complete picture of how the model is performing, than evaluating with one metric. These five metrics were chosen because they highlight different aspects of prediction errors. The F1 score specifically targets class imbalances because it works as a trade off between the Precision and Recall. The F1 score for the negative class compared to the F1 of the positive class can reveal if the model is better at predicting one class over the other.

3.10 Prediction on MAASTRO data

Final evaluation of all the models was performed on unseen data from the MAASTRO clinic, which was step 9 in the flowchart in Figure 3.4.1. The predictive performance of each model was evaluated using the same five performance metrics as presented in Section 3.9.1, the accuracy, the AUC, the MCC, the F1:1, and the F1:0.

3.11 Interpretability assessment

The models were validated on their performance, and the best performing models were interpreted based on their structure and prediction. In this thesis, ML models were intepreted based on visualizations of the models. For this reason, only tree-based algorithms were chosen, as they can provide insight to how predictions are made. The Decision Tree can easily be plotted into a tree, which is also the case with the FIGS model from the iModels package, and several other models.

Some models have built in functions to visualize their structure or predictions, like the XGBoost and the decision tree. The ensemble models can be visualized tree by tree, but if there are too many trees, the visualization loses its point. In that case, the feature importances can be comouted and visualized as a form of interpretation of the model. Feature importances were computed here with the Scikit learn feature permutation [49].

Chapter 4

Results

In this section, the results from the methods presented in Section 3 are presented. Firstly a baseline performance of each of the targets on the full, unreduced dataset D3, defined in Table 3.5.1, will be presented. The baseline model was a FIGS classifier, and the model was tuned by Optuna. The feature selection results using RENT are then presented for each of the targets. Results from five-fold cross-validation are then presented, before the models are assessed on their interpretability. Finally, the results from testing the models on the external MAASTRO data concludes this section.

4.1 The baseline model

A baseline for comparison of model performance was first made. The baseline was based on the full dataset D3, defined in Table 3.5.1, containing both clinical and radiomics features, without any feature selection. The MCC scores were used as a baseline comparison metric, and the FIGS classifier was the baseline classification algorithm, due to its high interpretability [42].

Table 4.1.1: A baseline performance of the FIGS classifier on the full dataset with both clinical and radiomics features, for each of the targets OS and DFS. The scores are given by the MCC, one for the five-fold cross validation on the OUS data, and the other on the MAASTRO test data.

Response	Cross validation MCC	External testing MCC
OS	0.180	0.139
DFS	0.191	0.156

From Table 4.1, it is apparent that the MCC of the baseline model is lower for the external testing than for the training data.

4.2 OS performance results

The results from the steps 2-9 in the workflow illustrated by Figure 3.4.1, are presented for the OS target in this section. The results for DFS are presented in Section 4.3.

4.2.1 RENT feature selection results

RENT was used for feature selection for all three datasets, D1, D2, and D3. The optimal C parameter of RENT was 0.1 for all datasets, whereas the *L1_ratios* had different optimal values for each dataset. RENT was run in a brute-force manner on all three datasets, D1, D2, and D3, with the response variable OS, as explained in Section 3.6. The result was three new and reduced datasets *DR1*, *DR2*, and *DR3*, which contained features selected by RENT at least once across 100 models. The features with selection frequency higher than 30% are displayed in Table 4.2.1, and the full table of all selected features can be found in Appendix 4.2.1.

Table 4.2.1: Table of features selected by RENT with a frequency higher than 30% for response OS. The results are divided by dataset, where D1 is the clincial dataset, D2 is the radiomics data, and D3 is the combination of both D1 and D2.

Data	Feature	Frequency (%)
D1	Cancer stage	100
	HPV-related	95
	Pack years	47
	Oropharynx	36
D2	Shape: Tumor Sphericity	100
	Texture: GLCM Joint Average (CT)	79
	Texture: GLCM Sum Average (CT)	79
	Shape: Tumor Major Axis Length (CT)	57
	Intensity: Maximum Discrete HU (CT)	35
	Texture: GLRLM High Gray Level Run Emphasis (PET)	34
	Shape: Maximum Tumor 3D Diameter	31
D3	Shape: Tumor Sphericity	100
	Cancer Stage	88
	HPV-related	86

For dataset D1, which contained only clinical data, seven of 14 features were selected at least once by RENT. For OS, *Cancer Stage* was selected in every model, followed by *HPV status*, which was selected in 95% of the models, as seen in Table 4.2.1. For the second dataset D2, containing only the radiomics data, 32 features were selected across the 100 models for the OS response. The *tumor sphericity*, a shape feature, was selected by every RENT model, corresponding to a 100% frequency. The four features below are all textural features from the CT images, followed by a texture feature from the PET images, and another shape feature, the maximum diameter.

The third dataset D3 contained clinical and radiomics features, and out of all 388 features in the input data, RENT chose seven features at least once. The top ranking feature was the shape feature *tumor sphericity*, the same as for D2. The *Cancer stage* and *HPV status* followed closely with 88% and 86%, respectively. Only three features were selected at a frequency higher than 30%, and all three were selected in more than 85% of the models. The full table of RENT selected features is found in Appendix A.2.

4.2.2 Reduced dataset

The feature selection performed by RENT, with the results presented in Section 4.2.1, resulted in three new datasets, DR1, DR2, and DR3. The three new datasets are described in Table 4.2.2. Dataset DR1, DR2, and DR3 were further used in this analysis to predict the OS target.

Table 4.2.2: The three datasets of RENT selected features from the three original datasets D1, D2, and D3. See Appendix A.2 for the full list of features.

Dataset	Description	Features
DR1	Clinical features selected by RENT at least once from dataset D1	7
DR2	Radiomics features selected by RENT at least once from dataset D2	32
DR3	Features selected at least once from dataset D3	7

4.2.3 Model validation

Seven ML classifiers were used for validation and prediction in this thesis, which were: Decison Tree, Random Forest, XGBoost, Histogram Gradient Boosting, FIGS, HSTree, and Boosted rules. All models were tuned using the Otuna package, which was outlined in Section 3.8. The result from running Otuna on all classifiers were an optimized combination of hyperparameters, which is found in Appendix B.2.

After optimizing the hyperprameters with Optuna, all models were tested again by a five-fold stratified cross validation, with 100 repeats through the RepeatedStratifiedKFold validator. The prediction results were then measured by five different performance, that were averaged over the 100 repeats, for all three datasets, DR1, DR2, and DR3. The results from the five-fold cross validation are presented in Table 4.2.3 by the five different performance metrics accuracy, AUC, MCC, F1:1, and F1:0, for datasets DR1, DR2, and DR3 respectively. The classifiers are ranked by their accuracy and MCC.

Table 4.2.3: Results from the Decision Tree, Random Forest, XGBoost, Histogram Gradient Boosting, FIGS, Hierarchical shrinkage, and Boosted rules classifiers after five-fold stratified cross validation with 100 repeats on dataset DR1, DR2, and DR3, respectively. The classifiers are ranked on the MCC scores for each dataset.

Dataset	Algorithm	Accuracy	AUC	MCC	F1:1	F1:0
DR1	XGBoost	0.7405	0.7219	0.4625	0.6557	0.7885
	Boosted Rules	0.7339	0.7249	0.4567	0.6725	0.7725
	Random Forest	0.7324	0.7158	0.447	0.6501	0.7795
	Decision Tree	0.7005	0.6872	0.3836	0.6059	0.7465
	FIGS	0.7081	0.6953	0.4035	0.6282	0.7524
	HistGradientBoosting	0.6918	0.6867	0.376	0.6319	0.7300
	HSTree	0.6900	0.6841	0.3707	0.6250	0.7295
DR2	HistGradientBoosting	0.7914	0.7750	0.5682	0.7256	0.8302
	XGBoost	0.6882	0.6598	0.3457	0.5620	0.7543
	RandomForest	0.6823	0.65528	0.3348	0.5572	0.7478
	HSTree	0.6725	0.6693	0.3526	0.597	0.6897
	BoostedRules	0.6725	0.6528	0.3187	0.5724	0.7279
	FIGS	0.655	0.6403	0.3526	0.597	0.6897
	Decision Tree	0.6441	0.6286	0.2667	0.5493	0.6990
DR3	BoostedRules	0.7707	0.7609	0.5324	0.7140	0.8057
	RandomForest	0.7525	0.7396	0.4915	0.6848	0.7932
	HistGradientBoosting	0.7446	0.7328	0.4763	0.6775	0.7849
	XGBoost	0.7414	0.7181	0.4763	0.6775	0.7849
	Decision Tree	0.7409	0.7316	0.4694	0.6771	0.7797
	FIGS	0.7252	0.7154	0.4384	0.6594	0.7657
	HSTree	0.6975	0.6642	0.3612	0.5375	0.7676

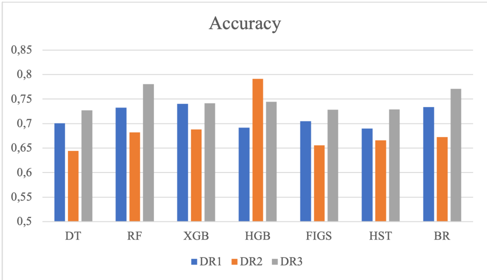
For the clinical dataset, DR1, the top three classifiers had very similar performance, all with accuracy and AUC above 0.73. FIGS came in fourth, with slightly poorer performance. Regarding interpretability, FIGS is a more interpretable model, as the other models are ensembles of trees, and thereby loose some interpretability. The boosting process used in both XGBoost and Boosted rules is not inherently interpretable, but according to the results here, bossting greatly benefits the decision tree algorithms used on dataset DR1.

For dataset DR2, all model performances are found in Table 4.2.3. The Histogram

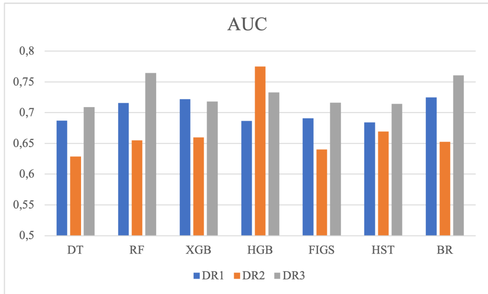
Gradient Boosting algorithm outperformed the other classifiers by nearly 10% for all performance metrics. Again, the boosting algorithms increased decision trees' performance. For dataset DR3, the Boosted Rules gave the highest performance, with an AUC of 0.76 and an MCC above 0.5. The Random Forest followed closely behind, as seen in Table 4.2.3. All classifiers had an accuracy and AUC above 0.7, except for the HSTree from iModels.

Note that Random Forest, XGBoost, and Boosted Rules were among the top dour classifiers for all three datasets. XGBoost and Boosted Rules are both based on the boosting concept, indicating that boosted decision trees perform better than non-boosted decision trees across all models. For all classifiers in Table 4.2.3, the F1:0 score is consistently higher than the F1:1 score. A higher F1:0 score indicates that the model predicts class 0 best. For OS, class 0 corresponds to survival. All models predicted the surviving patients slightly better than the dying patients. All models in Table 4.2.3 outperformed the baseline MCC score of 0.180 from Table 4.1. This indicated that the models benefited from the RENT feature selection.

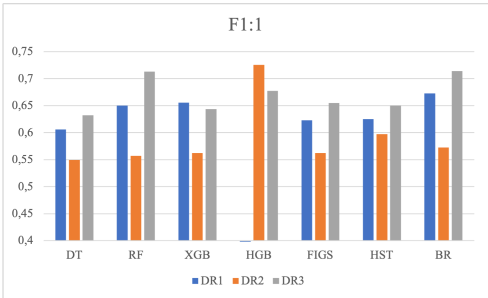
A plot of the five performance metrics, accuracy, AUC, MCC, F1:1, and F1:0, across the three datasets DR1, DR2, and DR3 are presented in Figure 4.2.1 for all seven ML classifiers used in this analysis. The classifiers are DT (Decision Tree), RF (Random Forest), XGB (XGBoost), HGB (Histogram Gradient Boosting), FIGS, HST (HSTree), and BR (Boosted Rules).



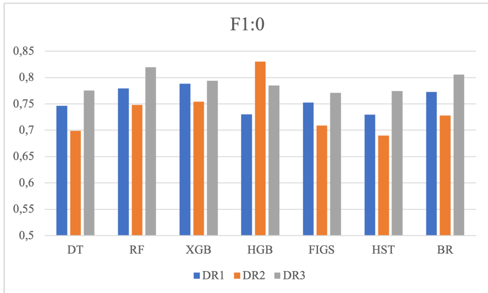
(a) Accuracy



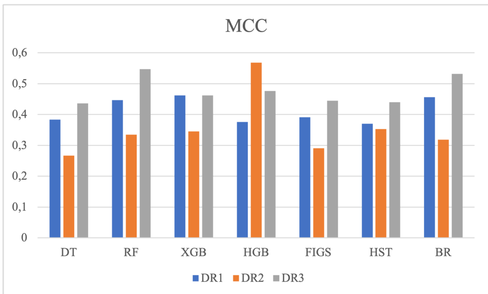
(b) AUC



(c) F1:1



(d) F1:0



(e) MCC

Figure 4.2.1: The performance from five-fold cross validation over 100 repeats for classifiers DT (DecisionTree), RF (RandomForest), XGB (XGBoost), HGB (HistGradientBoosting), FIGS, HST (HSTree), and BR (BoostedRules) on datasets DR1, DR2, and DR3 from OUS. The FIGS, HST, and BR are the models from the iModels package [2].

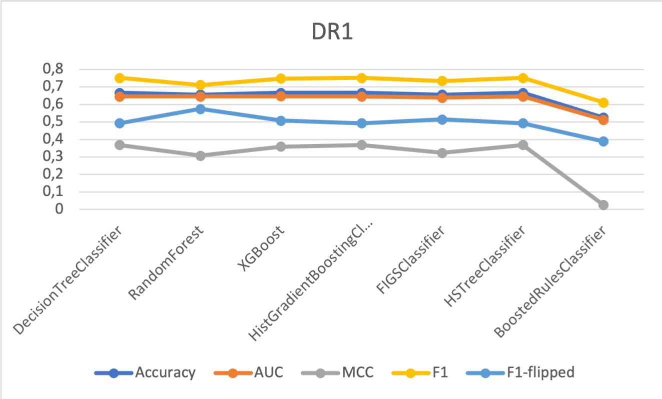
In general, most of the classifiers had poorer performance on the radiomics only dataset, DR2, than on DR3 and DR1, except for the Histogram Gradient Boosting which predicted very well on the dataset DR2. From Figure 4.2.1, the classifiers had higher performance for DR3 than DR1. The best performing classifier in this plot was the HistGradientBoostingClassifier when used on dataset DR2, followed by the RandomForestClassifier for dataset DR3. The DecisionTreeClassifier performed poorly in general, and the HSTree followed it closely.

4.2.4 Testing on external data

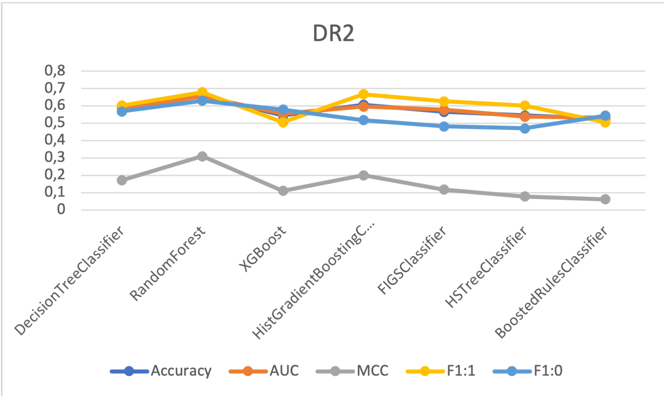
The OS models were finally tested on the external dataset from the MAASTRO clinic, and the results are presented below in Tigure 4.2.4.

Table 4.2.4: Results from the Decision Tree, Random Forest, XGBoost, Histogram Gradient Boosting, FIGS, HSTree, and Boosted rules classifiers when tested on external data from the MAASTRO clinic. The FIGS, HSTree, and Boosted rules are the models from the iModels package [2].

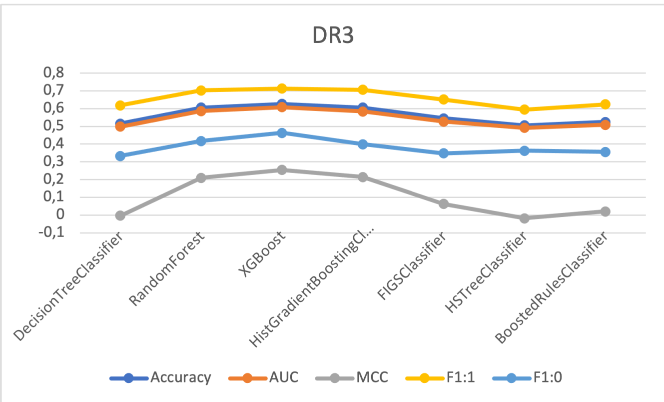
Dataset	Algorithm	Accuracy	AUC	MCC	F1:1	F1:0
DR1	Decision Tree	0.6667	0.6456	0.3688	0.7519	0.4923
	Random Forest	0.6566	0.6462	0.3072	0.7119	0.5750
	XGBoost	0.6667	0.6470	0.3588	0.7481	0.5075
	HistGradientBoosting	0.6667	0.6456	0.3688	0.7519	0.4923
	FIGS	0.6566	0.6390	0.3237	0.7344	0.5143
	HSTree	0.6667	0.6456	0.3699	0.7519	0.4923
	Boosted Rules	0.5252	0.5121	0.0260	0.6116	0.3896
DR2	Decision Tree	0.5859	0.5859	0.1714	0.6019	0.5684
	RandomForest	0.6566	0.6548	0.3097	0.6792	0.6303
	XGBoost	0.5454	0.5539	0.1106	0.5055	0.5784
	HistGradientBoosting	0.6061	0.5962	0.2007	0.6667	0.5185
	FIGS	0.5657	0.5770	0.1176	0.6261	0.4819
	HSTree	0.5455	0.5381	0.0779	0.6018	0.4706
	BoostedRules	0.5253	0.5308	0.0621	0.5053	0.5437
DR3	Decision Tree	0.5152	0.4984	-0.0037	0.6190	0.3333
	RandomForest	0.6061	0.5861	0.2102	0.7023	0.4179
	XGBoost	0.6869	0.6803	0.3680	0.7257	0.6353
	HistGradientBoosting	0.6061	0.5847	0.2145	0.7068	0.4000
	FIGS	0.5455	0.5267	0.0629	0.6512	0.3478
	HSTree	0.5051	0.4918	-0.0176	0.5950	0.3636
	BoostedRules	0.5253	0.5092	0.0207	0.6240	0.3561



(a) Performance on DR1



(b) Performance on DR2



(c) Performance on DR3

Figure 4.2.2: Performance of the seven classifiers for predicting OS on the external MAASTRO dataset, measured in accuracy, AUC, MCC, F1:1, and F1:0, for the Decision tree, Random Forest, XGBoost, Histogram Gradient Boosting, FIGS, HSTree, and Boosted rules. The three last models were from the iModels package [2].

HSTree from iModels, the Decision Tree, XGBoost, and HistGradientBoosting seemed to perform quite similarly on the dataset DR1. For the training data, XGBoost, HistGradient and Boosted rules had the highest performance, as seen in Table 4.2.3. The Decision Tree and the FIGS classifiers also performed well on the training data with meaning that two of the same classifiers performed well during validation and testing for dataset DR1.

For dataset DR2, RandomForest gave the best performance on the MAASTRO data, whereas for the training data, HistGradientBoosting outperformed all other classifiers, as seen in Figure 4.2.2. The high validation performance of HistGradientBoosting was not transferable to the test data, where this model gave poorer performance than the RandomForest.

XGBoost gave the best performance on dataset DR2 with the MAASTRO data, followed by the RandomForest and the HistGradientBoosting. All three of them also did well on the training data. Boosted rules did very well on the training data, as seen in Table 4.2.3, which did not translate well to the test data. For datasets DR2 and DR3, several classifiers give MCC around zero, and some even below zero. An MCC score below zero is considered worse than random guessing, meaning the model provides no predictions that are better than randomly guessing if the patients survives or not.

Table 4.2.5: MCC scores for all classifiers for OS prediction on external testing of datasets DR1, DR2, and DR3.

Classifier	DR1	DR2	DR3
DecisionTree	0.3688	0.1714	-0.0037
RandomForest	0.3072	0.3097	0.2102
XGBoost	0.3588	0.1106	0.2547
HistGradientBoosting	0.3688	0.2007	0.2145
FIGS	0.3237	0.1176	0.0629
HSTree	0.3688	0.0779	-0.0176
BoostedRules	0.026	0.0621	0.0207

Table 4.2.5 shows the MCC scores of all classifiers for OS prediction on the three datasets, DR1, DR2, and DR3, of the unseen MAASTRO data. The baseline performance for external validation on the OS target, as in Table 4.1, was 0.139. For dataset DR1, all classifiers except BoostedRules outperformed the baseline by over 10%. For the second dataset DR2, the baseline was outperformed by the Decision Tree, Random Forest, and HistGradientBoosting classifiers. For dataset DR3, only three of the classifiers outperformed the baseline, and two classifiers

even had MCC scores below zero. Random Forest, HistGradientBoosting, and XGBoost had the highest overall MCC on the three datasets.

4.2.5 Interpretability assessment

The interpretability assessment consisted of visualizing high performing models to reveal the reasoning for the prediction. The highest performing models throughout the dataset for OS prediction were the Random Forest, the XGBoost, and the Histogram Gradient Boosting classifiers, with some occasionally high performances from the Decision tree model. Both the Random Forest and the XGBoost have visualization possibilities, however, this is not the case for the Histogram Gradient Boosting. Therefore, Random Forest and XGBoost were visualized in this Section, based on the model structure on the DR3 dataset.

Random Forest proved to have high performance across all datasets, as seen in Table 4.2.3. For dataset DR3 Random Forest had an accuracy and an AUC above 0.76, making it the best performing classifier on the OUS data. However, the model did not generalize well, and obtained an AUC of 0.59 on the MAASTRO data. XGBoost performed slightly better on the MAASTRO data with an AUC of 0.61, yet XGBoost obtained a slightly lower performance than the Random Forest during the validation on the OUS data. Boosted rules had a high validation performance, however, it obtained one of the lowest testing performances, as seen in Table 4.2.4 in the bottom row. Random Forest and XGBoost were visualized by their structure and their feature importance.

The models with highest overall performance for OS prediction were based on dataset DR1, the clinical data. From all classifiers on dataset DR1, the Decision tree and the Random forest were visualized, as they were the highest performing models on dataset DR1.

Decision tree on clinical data

The Decision tree classifier obtained an MCC score of 0.3836 during training on the OUS data, and an MCC of 0.3688 on the external testing on the MAASTRO data, as seen in Table 4.2.3 and 4.2.4. The tree decision tree was visualized, and its structure is shown in Figure

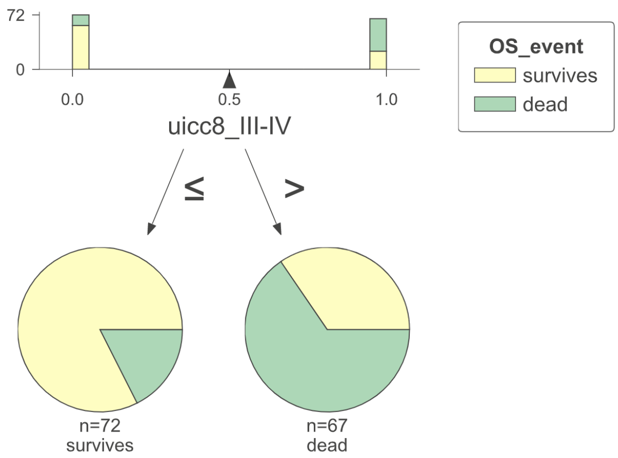
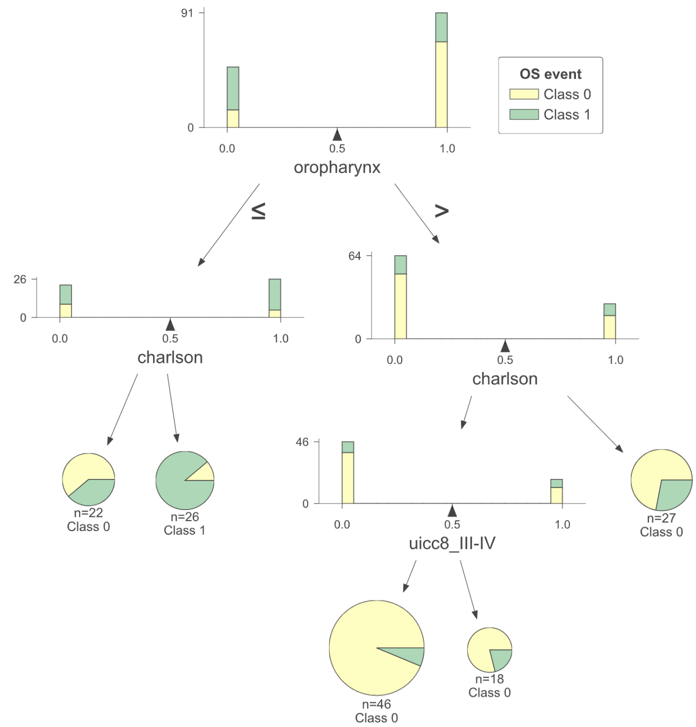


Figure 4.2.3: The Decision tree for dataset DR1 for OS prediction

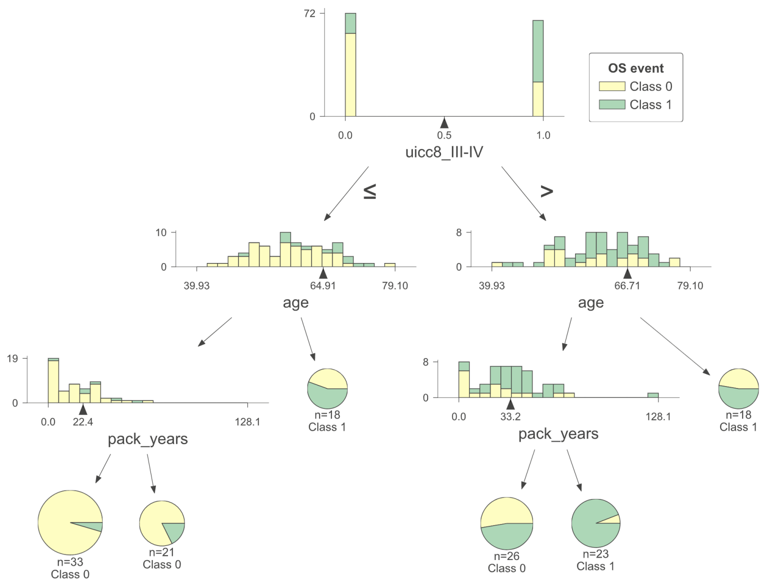
The decision tree has one rule based on the cancer stage feature $uicc8\_III - IV$. This means the model predicts that all patients with cancer stage $I - II$ survive and all patients with cancer stage $III - IV$ do not survive. The model is straightforward and very interpretable.

Random forest on clinical data

The Random forest classifier obtained a training MCC of 0.447 and a test MCC of 0.31, as seen in Table 4.2.3 and 4.2.4. The model consisted of four trees, two of which are visualized in Figure 4.2.4. Each tree had a depth of 3.



(a) Tree 1/4 in the Random Forest.



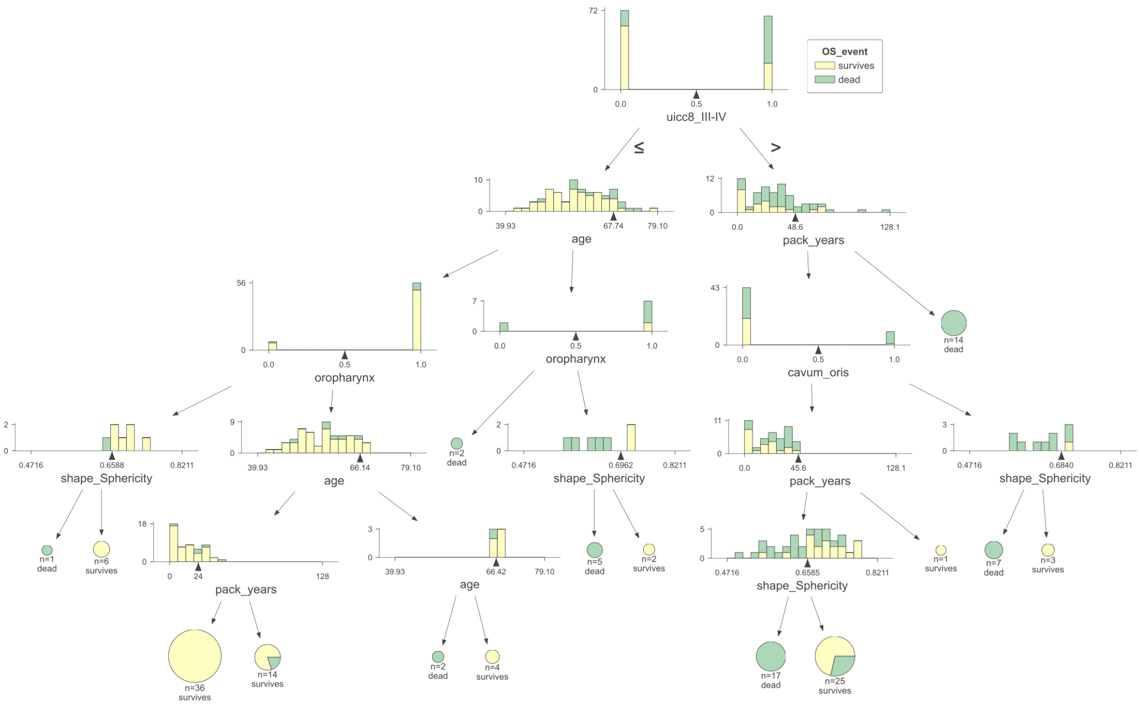
(b) Tree 2/4 in the Random Forest.

Figure 4.2.4: Two of the four trees that make up the Random Forest model on clinical data for OS prediction. Note that the final prediction is made by a majority vote across all four trees, and that these trees are two of those four trees.

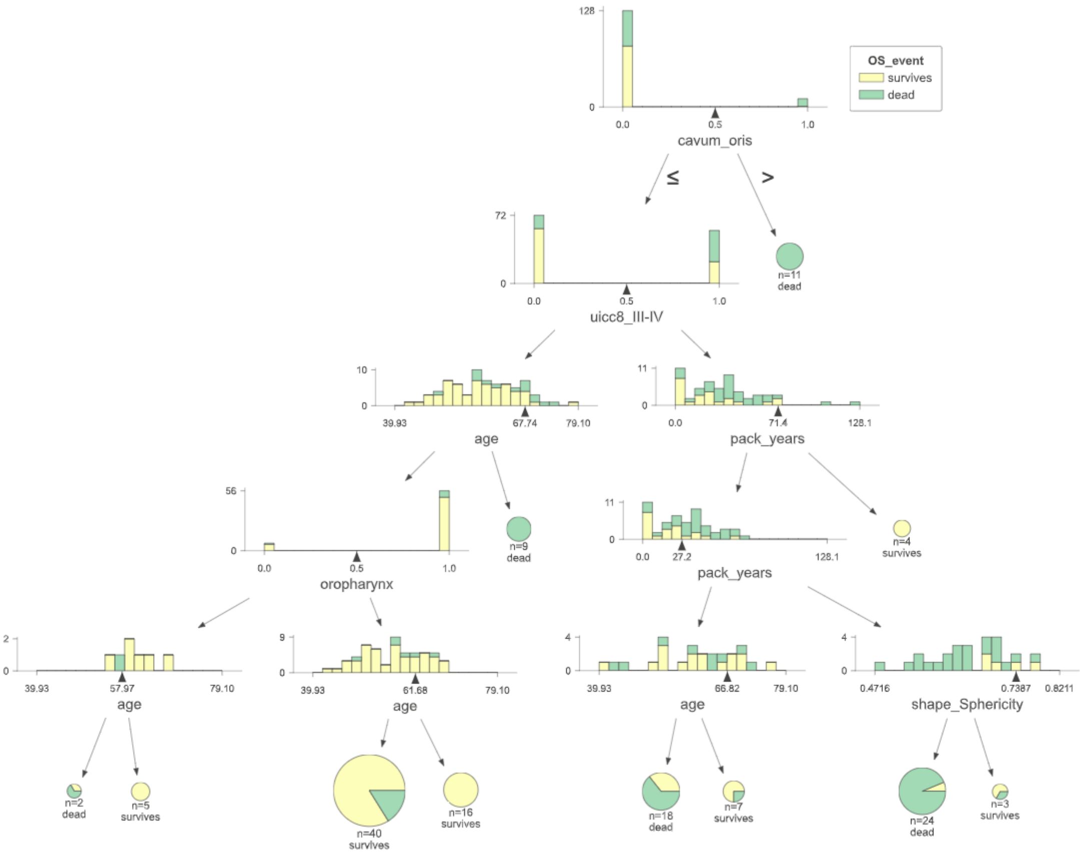
The Random forest model on dataset DR1 consisted of four trees, each with a depth of 4. Tree 1 splits on *oropharynx* and both child nodes are then split on the *charlson comorbidity index*. There are, in total, five leaf nodes on Tree 1. The second tree splits at the cancer stage before both child nodes are split on age. In this tree, pack years are also used slightly further down. This model is highly interpretable as the four trees can easily give an overeview of what the model is doing.

Random Forest on clinical and radiomics data

The Random Forest model consisted of 12 estimators with a maximum depth of 5, as seen in Table B.3.2. The impurity criterion was Gini, which was calculated for each step by Equation 2.7.2 in Section 2.7. The Random forest model obtained a training MCC of 0.49 and a test MCC of 0.21. The plots below show some of the trees generated by the Random Forest classifier. The samples belong to the OUS data, meaning that there are 139 total samples classified in each tree. Note that the final prediction was made by a majority vote amongst all trees, as mentioned in Section 2.7.3. The trees visualized here only contribute towards the final prediction, which is made by all of the 12 trees together, and not by the individual trees.



(a) Tree 5 in the Random Forest.



(b) Tree 10 in the Random Forest.

Figure 4.2.5: Two of the trees in the Random Forest model for dataset DR3 and OS target. Note that uicc8_III − IV is the Cancer stage feature from Table 3.1.2, and the shape_spericity describes the roundness of the tumor. In this visualization, yellow representd the surviving patients, and green represent the patients that did not survive.

In Figure 4.2.5, two of the trees from the Random Forest are visualized. Tree 5 started by separating the patients by cancer stage, where the patients in the lower stages, $I - II$, went to the left node, and the higher stages, $III - IV$, went to the right node. The patients in the lower cancer stage were then split by age, 67.74, creating two new nodes. Each of the new nodes were then split by oropharynx, which created a leaf node of two samples. Further, the shape sphericity, age and pack years were used for splitting until all samples ended up in a leaf node. This tree had a depth of 5 and 14 rules.

Tree 10 in the forest, as seen in Figure 4.2.5b, was a little smaller than tree 5. Tree 10 contained 10 rules, which was four less than tree 5. Tree 10 made the first feature split on the *cavum_oris* feature, which was the rarest cancer site in the dataset. The first split created a pure leaf node immediately, showing that all 11 patients with cancer in the oral cavity in the OUS dataset died. Both trees showed that patients with less sphere-like tumors more often ended up in leaf nodes where the majority of the samples were classified as dead.

The Random forest model containing 12 trees with a depth of five is likely too complicated to interpret directly from the visualizations. Therefore, the feature importance was found for all the features in the dataset. The Random Forest feature importance was found using the ScikitLearn permutation importance. The feature importance was calculated by defining a baseline model with all the features in the dataset, before dropping the features one by one, and comparing the performance of model to the baseline [49]. The feature importance is then expressed by how much the performance decreased when the feature was excluded. The result is shown in Figure 4.2.6.

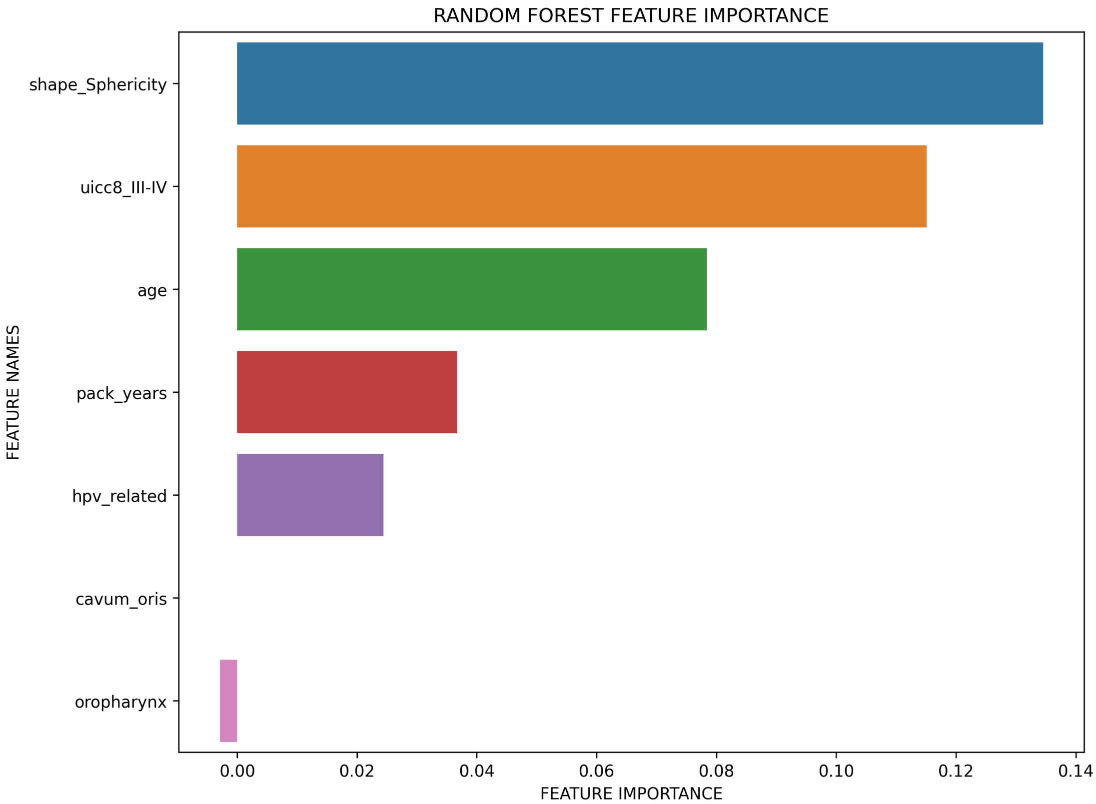
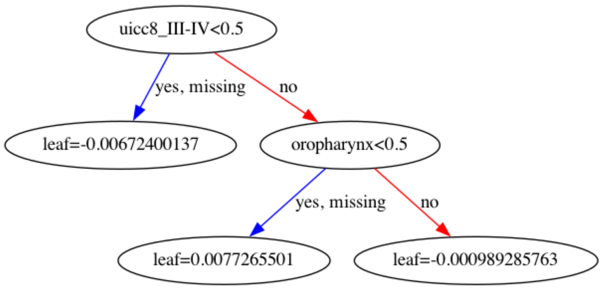


Figure 4.2.6: The feature importances of the Random Forest model on the DR3 dataset for the OS target. Note that the feature importances were found by feature permutation.

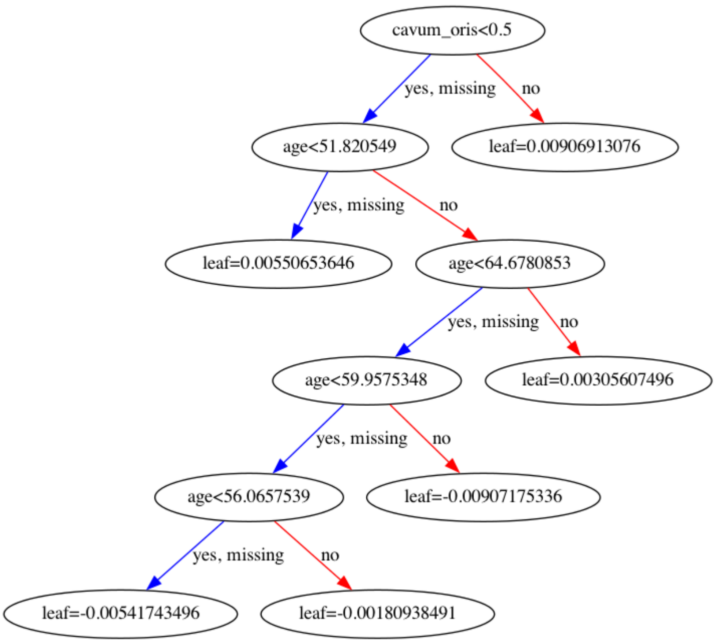
Figure 4.2.6 suggests that the shape sphericity, meaning the tumor's roundness, is the dataset's most important feature. According to Figure 4.2.6, the model performance rises with around 13% by including the shape sphericity. The *uicc8_II-IV* feature, for the Cancer stage, was the second most important feature according to Figure 4.2.6, followed by *age*, *pack years*, and *HPV status*. *Cavum oris* had did not have any impact on the models performance, and *oropharynx* had a negative impact on the model, meaning that excluding the feature improves the models performance slightly, as implied by Figure 4.2.6.

XGBoost on the clinical and radiomics data

The XGBoost model, with all hyperparameters listed in Table B.2.3, consisted of 9 estimators with a maximum depth of 9. The plots in Figure 4.2.7, show the structure of two of the trees in the XGBoost model on dataset DR3 for OS. Note that the final prediction is made in a boosting manner, described in Section 2.7.4, meaning none of the individual trees visualized here made a prediction alone.



(a) Tree 2 in the XGBoost.



(b) Tree 3 in the XGBoost.

Figure 4.2.7: Two of the trees in the XGBoost model for dataset DR3 and OS target. Note that uicc8_$III - IV$ is the Cancer stage feature from Table 3.1.2, and the shape_sphericity describes the roundness of the tumor.

Figure 4.2.7 shows two of the 9 trees that was generated by XGBoost for dataset

DR3 for OS. This plot looks different from the one for Random Forest, as it was visualized by the *plot_tree* function implemented in XGBoost. Tree 2 made the first split on feature *uicc8_III-IV*, the cancer stage, followed by one split on the *oropharynx* feature. Tree 3 started by splitting on the *Cavum oris* feature, similar to the Random Forest model from Figure 4.2.5b, followed by four consecutive splits on different ages. Nine trees in total makes the XGBoost model a little to complicated to grasp through the visualizations only. Therefore the feature importances were found for each feature, in an attempt to add more context to the model.

The feature importances were computed in the same manner as for Random Forest, and the results are plotted in Figure 4.2.8. Like Random Forest, the most important feature for the XGBoost model was the shape sphericity, according to Figure 4.2.8, and excluding it from the model degrades the model performance by around 5%. Several of the RENT selected features had a negative feature importance, as the plot in Figure 4.2.8 shows, including *oropharynx*, *hpv-related*, *pack_years*, and *age*. Several of the features that were considered unimportant by Figure 4.2.8, were used in the trees visualized in Figure 4.2.7.

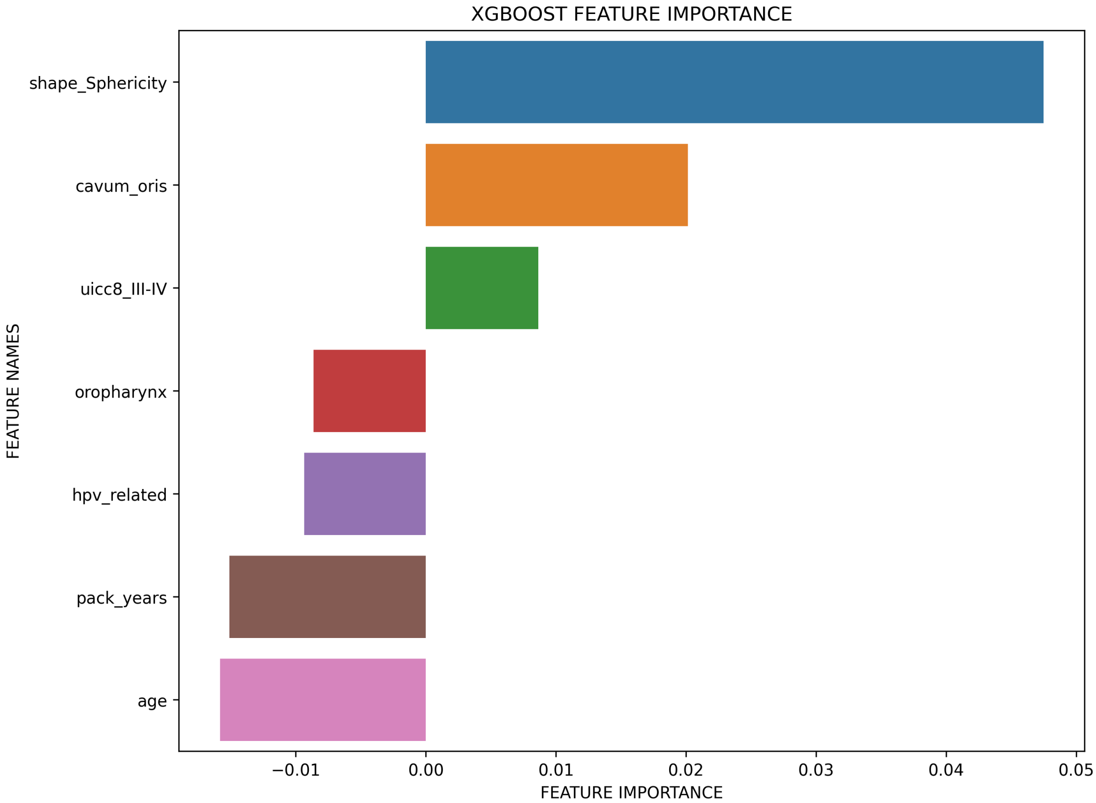


Figure 4.2.8: The feature importances of the features in dataset DR3 for the fitted XG-Boost model for OS target. Note that the feature importances were averaged over 10 rounds of feature permutation, and that the numbers on the X-axis corresponds to the difference in performance of the baseline model and the model where each feature is permutated.

4.3 DFS performance results

All results from step 2-9 in the flowchart in Figure 3.4.1 for DFS are presented in this section.

4.3.1 RENT feature selection results

RENT was used in a brute-force manner for feature selection amongst all datasets, D1, D2, and D3. The features selected with a frequency more than 30% by RENT are presented in Table 4.3.1. The full table of all features selected at least once is presented in Appendix B.3.

Table 4.3.1: Table of features selected by RENT with a frequency higher than 30%. for target DFS.

Data	Feature	Frequency (%)
D1	HPV-related	98
	Cancer Stage	89
	Pack years	41
	Cavum Oris	39
	Oropharynx	37
D2	Shape: Tumor Sphericity	95
	Texture: LBP_102 (PET)	95
	Shape: Tumor Elongation	69
	Texture: GLSZM Small Area Low Gray Level Emphasis (CT)	63
	Texture: LBP_201 (PET)	48
	Texture: GLSZM Gray Level Non-Uniformity Normalized (PET)	31
D3	Shape: Tumor Sphericity	98
	Shape: Tumor Elongation	95
	Texture: LBP_102 (PET)	94
	Texture: GLSZM Small Area Low Grey Level Emphasis (CT)	85
	Texture: LBP_201 (PET)	68
	HPV-related	55
	Texture: GLSZM Grey Level Non-Uniformity Nomalized (PET)	49
	Cancer Stage	47

For the D1 dataset, consisting only of clinical data, there were 10 features that were selected by RENT at least once. *HPV-status* was the most selected feature with a 98% frequency. *cancer stage* was selected at an 89% frequency, and the frequency dropped to 41% for *pack years.* For dataset D2, 25 features were selected at least once by RENT. The *shape sphericity* and the texture feature *LBP_102 (PET)* both had a frequency of 95%. The shape feature tumor spherecity describes the roundness of the tumor, and the *LBP_102* is a PET image texture feature. The shape feature tumor elongation, had a frequency of 69%. The three next features are texture features, for both CT and PET images.

Dataset D3, consisted of both clinical and radiomics features, had the highest number of features selected at least once by RENT. In total, 42 features from D3 were selected at least once by RENT. Again, the shape feature *tumor sphericity* had the highest selection frequency, followed by the *tumor elongation*. The five most selected features in D2 and D3 were the same, however, the frequencies were slightly different, as seen in Table 4.3.1.

4.3.2 Reduced dataset

The feature selection performed by RENT, with the results presented in Section 4.3.1, resulted in three new datasets, DR1, DR2, and DR3. The three new datasets are decribed in Table 4.3.2.

Table 4.3.2: The three datasets of RENT selected features for DFS from the three original datasets D1, D2, and D3.

Dataset	Description	Features
DR1	Clinical features selected by RENT at least once from dataset D1	10
DR2	Radiomics features selected by RENT at least once from dataset D2	25
DR3	Features selected at least once from dataset D3	42

4.3.3 Model validation

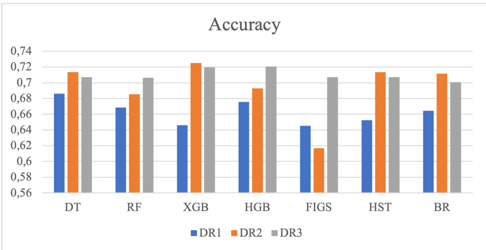
The three reduced datasets *DR1*, *DR2*, and *DR3* were then used to train models. For optimization of hyperparameters, the Optuna framwork was used. The Optuna framework was outlined in Section 3.8, and the optimal hyperparameters for each classifier on all three datasets is presented in Appendix B.2. The models were then tested again in a five-fold cross validation over 100 repeats. The results of the four best classifiers for each dataset are presented in Table 4.3.3.

Table 4.3.3: Results from the Decision Tree, Random Forest, XGBoost, Histogram Gradient Boosting, FIGS, Hierarchical shrinkage, and Boosted rules classifiers after five-fold stratified cross validation with 100 repeats on dataset DR1, DR2, and DR3 respectively. The performances are ranked by MCC.

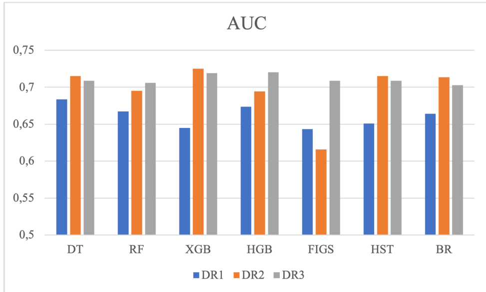
Dataset	Algorithm	Accuracy	AUC	MCC	F1:1	F1:0
DR1	Decision Tree	0.6860	0.6836	0.3852	0.6327	0.7196
	HistGradientBoosting	0.6756	0.6738	0.3618	0.6312	0.7049
	Random Forest	0.6686	0.6674	0.3450	0.6336	0.6919
	Boosted Rules	0.6646	0.6639	0.3349	0.6427	0.6767
	HSTree	0.6524	0.6511	0.3095	0.6193	0.6738
	FIGS	0.6455	0.6433	0.3018	0.5919	0.6780
	XGBoost	0.6459	0.6448	0.2984	0.6113	0.6687
DR2	XGBoost	0.7253	0.7250	0.4575	0.7142	0.7310
	Decision Tree	0.7135	0.7152	0.4462	0.7278	0.6872
	HSTree	0.7135	0.7152	0.4462	0.7278	0.6872
	BoostedRules	0.7117	0.7134	0.4431	0.7270	0.6836
	HistGradientBoosting	0.6930	0.6942	0.3980	0.6942	0.6817
	RandomForest	0.6855	0.6952	0.3971	0.0.6840	0.7003
	FIGS	0.6169	0.6158	0.2387	0.6042	0.6196
DR3	HistGradientBoosting	0.7208	0.7204	0.4478	0.7124	0.7235
	XGBoost	0.7196	0.7191	0.4457	0.7065	0.7266
	Decision Tree	0.7071	0.7086	0.4329	0.7200	0.6818
	FIGS	0.7071	0.7086	0.4329	0.7200	0.6818
	HSTree	0.7071	0.7086	0.4329	0.7200	0.6818
	BoostedRules	0.7009	0.7026	0.4209	0.7158	0.6732
	RandomForest	0.7013	0.6693	0.4104	0.6693	0.7228

The Decision tree classifier was within the top three classifiers for all three datasets, DR1, DR2, and DR3, for predicting DFS. This was not the case for OS, as seen in Table 4.2.4. Further all classifiers outperformed the baseline performance from Table 4.1, which was an MCC of 0.190. In general, the *F1:1* and *F1:0* scores are quite similar, which indicated that the model predicted the dead or relapsed patients.

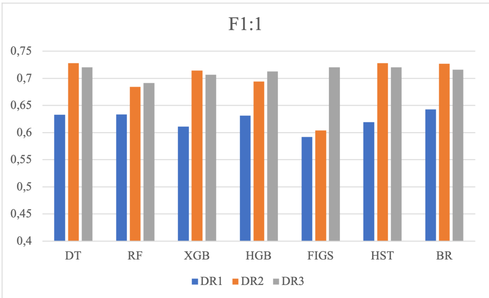
For dataset DR1, the clinical data with features selected by RENT, the four best-performing classifiers all performed very similarly, with accuracies and AUC scores above 0.66, as seen in Table 4.3.4. The Decision Tree and the HistGradientBoosting classifier had F1:0 scores above 0.7. For DR3, three classifiers performed identically: the FIGS, HSTree, and Decision Tree.



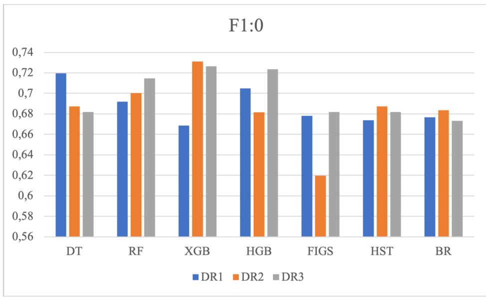
(a) Accuracy



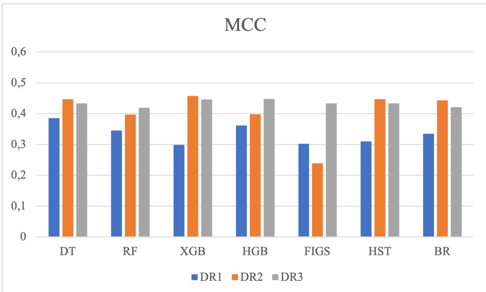
(b) AUC



(c) F1:1



(d) F1:0



(e) MCC

Figure 4.3.1: The performance from five-fold cross validation for DFS prediction over 100 repeats for classifiers DT (DecisionTree), RF (RandomForest), XGB (XGBoost), HGB (HistGradientBoosting), FIGS, HST (HSTree), and BR (BoostedRules) on datasets DR1, DR2, and DR3. The FIGS, HST, and BR are from the iModels package [2].

The performance on datasets DR2 and DR3 were, for many of the classifiers, very similar. Figure 4.3.1 shows that the FIGS classifier performed poorly on DR2, whereas the XGB classifier performs very well on the same dataset. The Decision Tree had very high F1:0 score for dataset DR1, however, the other performances on DR1 were below par. The overall best performance is on dataset DR2 by the XGBoost classifier.

In general, all classifiers performed better on datasets DR2 and DR3, than on DR1, which indicated that the radiomics features contributes positively to the models. Only the FIGS classifier performed better on the DR1, the clinical data, than on DR2, the radiomics data. The Decision tree had a high $F1:0$ score for dataset DR1, indicating that the model predicted the disease-free patients slightly better than the relapsed or passed patients.

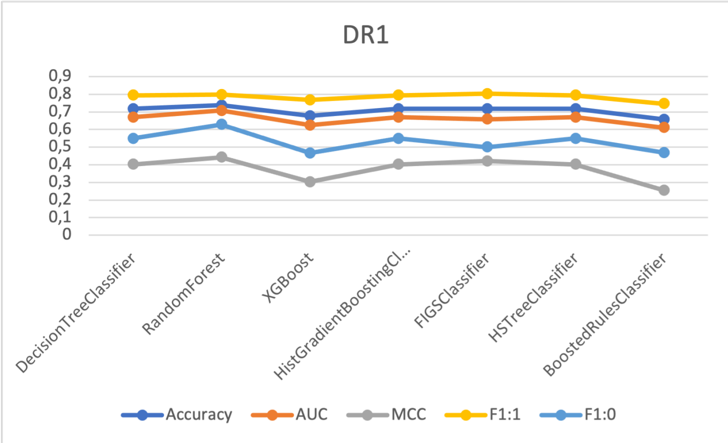
4.3.4 Testing on External dataset

The models were all tested on the external dataset from the MAASTRO clinic. The results from prediction on the MAASTRO data is presented by Figure 4.3.2.

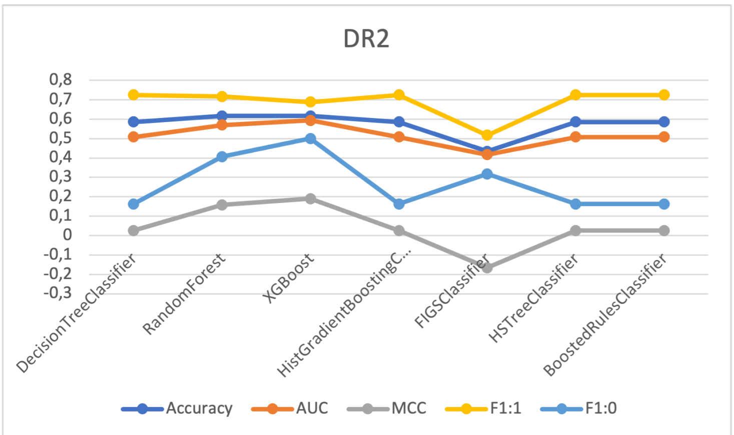
Table 4.3.4: Results from the Decision Tree, Random Forest, XGBoost, Histogram Gradient Boosting, FIGS, HSTree, and Boosted rules classifiers when tested on external data from the MAASTRO clinic for DFS prediction. The FIGS, HSTree, and Boosted rules are the models from the iModels package [2].

Dataset	Algorithm	Accuracy	AUC	MCC	F1:1	F1:0
DR1	Decision Tree	0.7172	0.6701	0.4016	0.7941	0.5484
	Random Forest	0.7374	0.7072	0.4425	0.7969	0.6286
	XGBoost	0.6768	0.6242	0.3035	0.7681	0.4667
	HistGradientBoosting	0.7172	0.6701	0.4016	0.7941	0.5485
	FIGS	0.7172	0.6701	0.4016	0.7941	0.5484
	HSTree	0.7172	0.6701	0.4016	0.7941	0.5485
	Boosted Rules	0.6566	0.6112	0.2575	0.7462	0.4688
DR2	Decision Tree	0.5859	0.5076	0.0260	0.7248	0.1633
	RandomForest	0.6162	0.5693	0.1587	0.0.7164	0.4062
	XGBoost	0.6162	0.5934	0.1906	0.6885	0.5000
	HistGradientBoosting	0.5958	0.5076	0.0260	0.7248	0.1633
	FIGS	0.4343	0.4167	-0.1653	0.5172	0.3171
	HSTree	0.5958	0.5076	0.0260	0.7248	0.1633
	BoostedRules	0.5958	0.5076	0.0260	0.7248	0.1633
DR3	Decision Tree	0.5859	0.5076	0.0260	0.7248	0.1633
	RandomForest	0.6869	0.6487	0.3277	0.7634	0.5373
	XGBoost	0.5960	0.5483	0.1106	0.7015	0.3750
	HistGradientBoosting	0.6061	0.5528	0.1266	0.7153	0.3607
	FIGS	0.5859	0.5076	0.0260	0.7248	0.1633
	HSTree	0.5859	0.5076	0.0260	0.7248	0.1633
	BoostedRules	0.5859	0.5076	0.0260	0.7248	0.1633

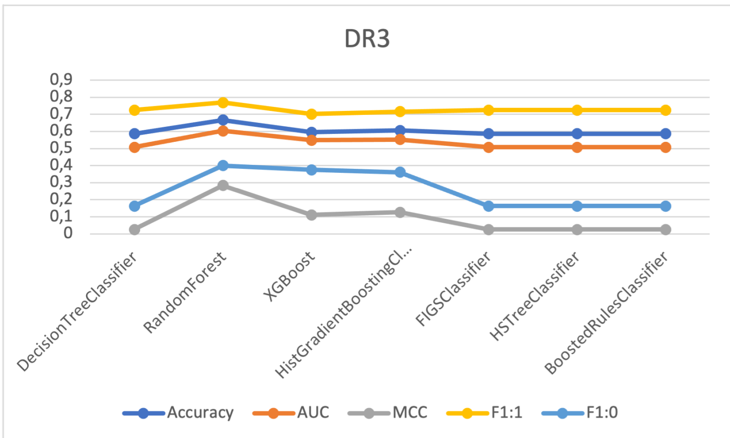
The performance metrics are presented in Table 4.3.4. For dataset DR1 the performance of the RandomForest was somewhat higher than the other classifiers, as shown in Table 4.3.5. Several other classifiers performed similarly on the DR1 data, and the FIGS classifier had the second highest MCC score. For DR1, the classifiers all performed better on the external dataset than on the training data. For the DR2 dataset, with only radiomics features, the XGBoost and RandomForest classifiers had the highest performance on the external data, with an MCC of 0.37. XGBoost performed best on the training data as well, indicating generalizability.



(a) DR1 MAASTRO



(b) DR2 MAASTRO



(c) DR3 MAASTRO

Figure 4.3.2: Results of testing seven classifiers for prediction of DFS on the external MAASTRO dataset.

For dataset DR2 qnd DR3 several of the MCC scores are close to zero, and some are even below. An MCC score below zero indicates that the classifier performs worse than random guessing. For predicting DFS, the radiomics features did not contribute positively to the predictions.

Table 4.3.5: MCC scores for all classifiers for DFS prediction on external testing of datasets DR1, DR2, and DR3.

Classifier	DR1	DR2	DR3
DecisionTree	0.4016	0.026	0.026
RandomForest	0.4425	0.1587	0.2836
XGBoost	0.3588	0.1906	0.1106
HistGradientClassifier	0.4016	0.026	0.1266
FIGS	0.4214	-0.1653	0.026
HSTree	0.4016	0.026	0.026
BoostedRules	0.2547	0.026	0.026

Compared to the baseline from Table 4.1, which was 0.156 for external testing of DFS, all models on the clinical dataset, DR1, performed better than the baseline. For DR2, only Random Forest and XGBoost outperformed the baseline score of 0.156. For DR3, the Random Forest was the only classifier that beat the MCC score.

4.3.5 Interpretability and overall assessment

The highest-performing models for DFS prediction were all on the clinical dataset, as seen by the MCC scores in Table 4.3.5. For the clinical dataset, DR1, the Decision tree, FIGS, and Random Forest were visualized, whereas for the clinical and radiomics dataset, DR3, only the Random forest was visualized.

Decision tree on clinical data

For dataset DR1, the Decision tree achieved a training MCC of 0.39 and a test MCC of 0.41. The Decision tree had the highest performance on the training data, whereas the Random forest performed the highest on the test data. The Decision tree structure for dataset DR1 is presented in Figure 4.3.3. The decision tree had a depth of 2, and the optimization criterion was Gini.

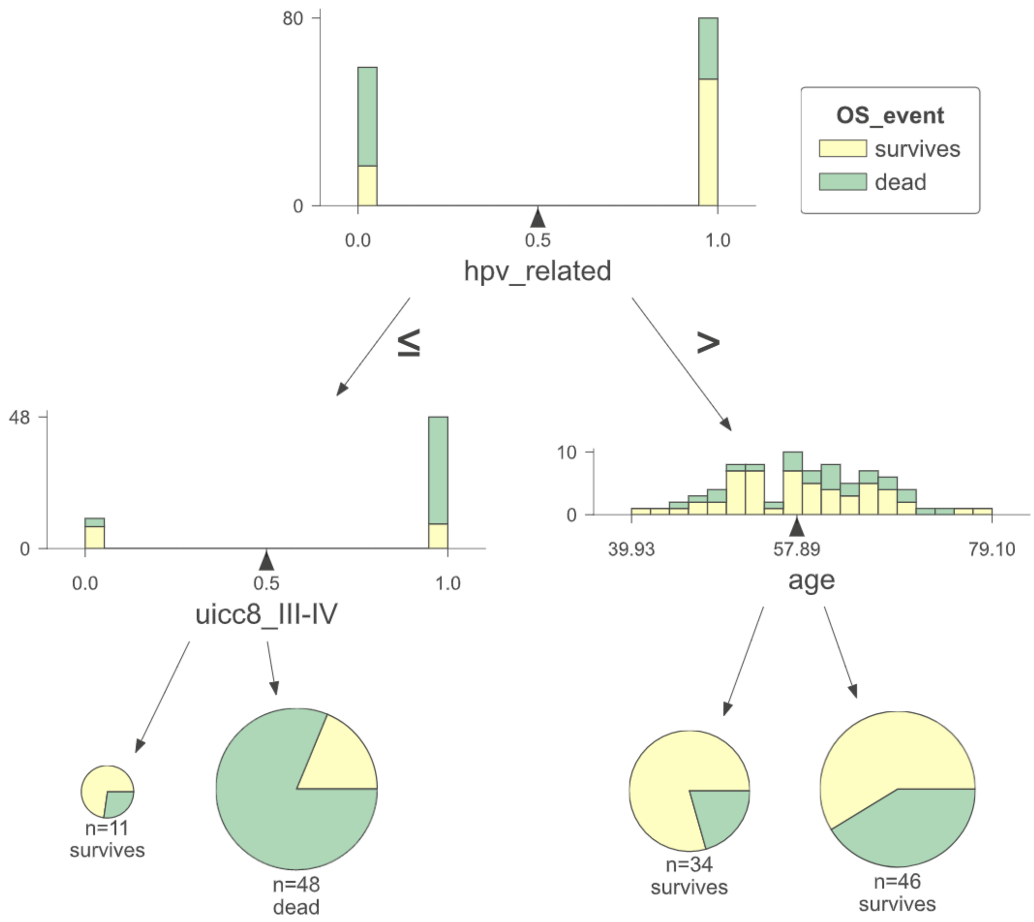
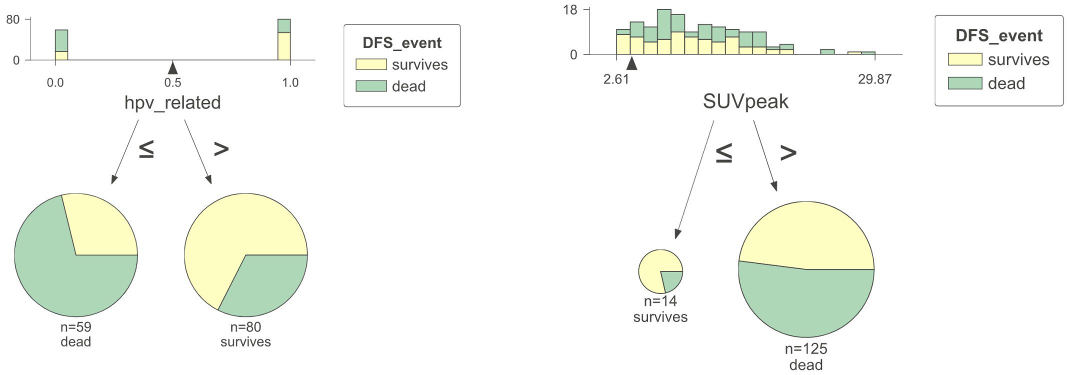


Figure 4.3.3: Decision tree for dataset DR1, the clinical dataset, for predicting DFS.

The Decision tree classifier did the first split on the *HPV-status* feature. Further, the HPV-negative patients were split by cancer stage, and patients with cancer stage *I-II* were predicted to survive disease free. The HPV-positive patients were split by age. However, both leaf nodes predicted surviving patients, as seen in Figure 4.3.3.

FIGS on clinical data

The FIGS model obtained an MCC of 0.30 on the OUS training data and 0.42 on the MAASTRO test data for the clinical dataset. The model consisted of two trees, and the features *HPV-status* and *SUVpeak* were chosen by the FIGS algorithm for the two rules.



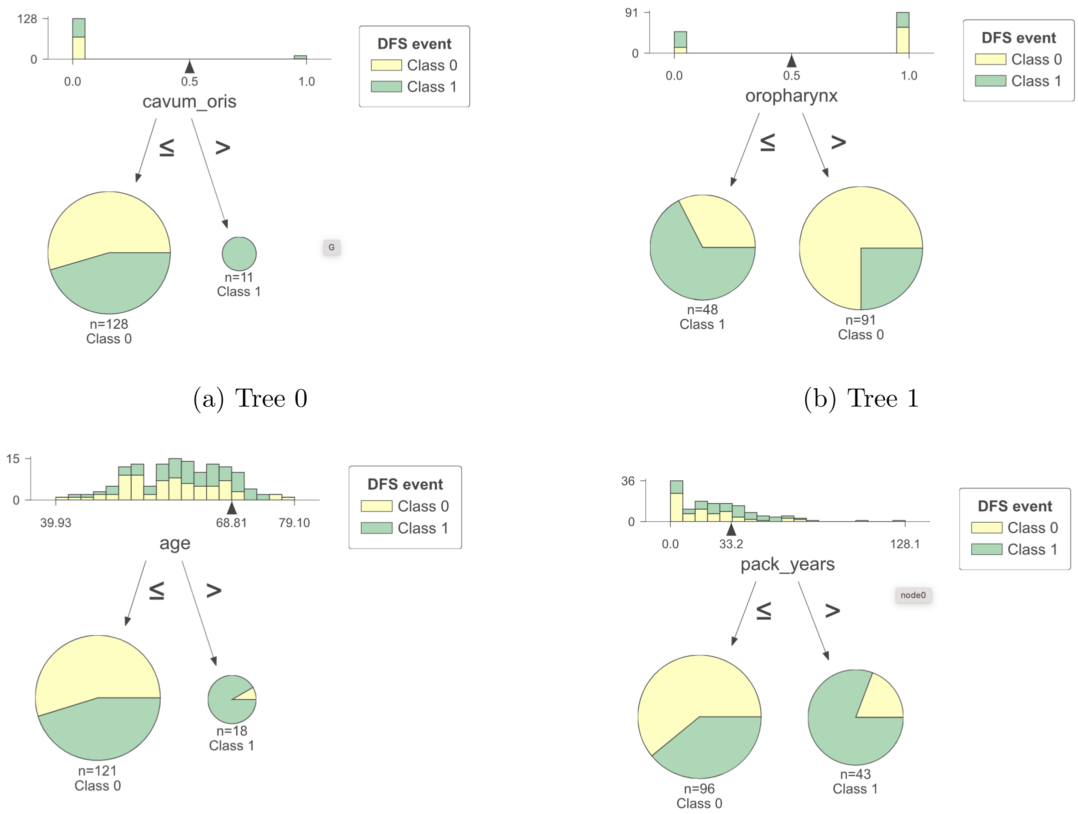
(a) Tree 1/2 (b) Tree 2/2

Figure 4.3.4: The structure of the FIGS model on dataset DR1 for predicting DFS.

The FIGS model on the clinical data found that **??** and *SUVpeak* were the most informative features to split on. The HPV status was a slightly better predictor, as the SUVpeak predicted many survuving patients wrongly, as seen in the right tree in Figure 4.3.4.

Random forest on clinical data

Random forest model on dataset DR1 achieved a training MCC of 0.35 and a test MCC of 0.44 as seen in Table 4.3.3 and 4.3.4, respectively. The Random forest model consisted of *8 estimators*, each with a maximum depth of 1. A majority vote amongst the estimators made the final prediction. Four Random forest model estimators are visualized in Figure 4.3.5.



(a) Tree 0

(b) Tree 1

(c) Tree 2

(d) Tree 3

Figure 4.3.5: The structure of four of the estimators in the Random forest model for dataset DR1 predicting DFS.

The estimators in the Random forest model in Figure 4.3.5 split on the features *cavum oris*, *oropharynx*, *age*, and *pack years*. The oropharynx and cavum oris are two cancer site features described in Table 3.1.2. The age was split on 68.8 years, and pack-years were split on 33.

Random forest on clinical and radiomics data.

The chosen model for DR3 was the Random Forest, which had a high performance on the external MAASTRO data. Random Forest obtained a training MCC score of 0.4189, which was the lowest amongst all the classifiers. However, it was the only classifier to perform better than the baseline on the external data. The Random Forest model consisted of 20 estimators, each with a maximum depth of 10, which is quite a large model. One of the trees was visualized in Figure 4.3.6. Note that the final prediction was made from a majority vote between all 20 estimators and not by the tree visualized here.

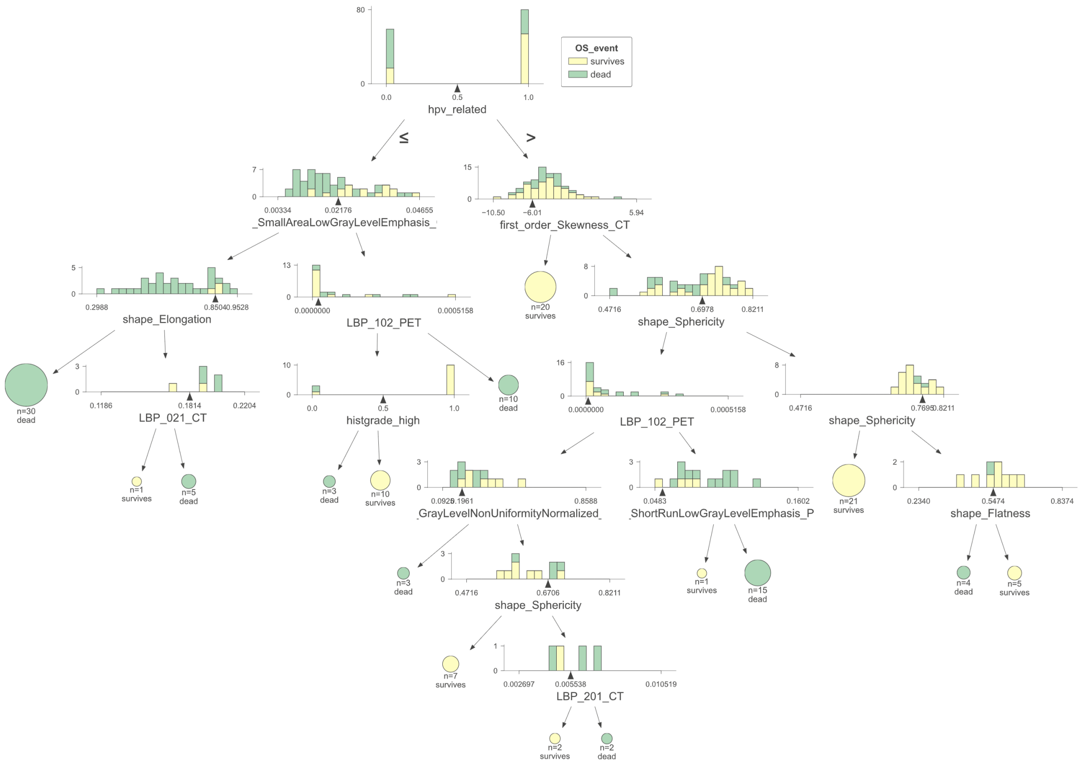


Figure 4.3.6: Tree 7 of 20 from the Random Forest model for predicting DFS on dataset DR3. Note that this tree is not the only tree that makes up the prediction of the Random Forest model. The prediction is made by a majority vote.

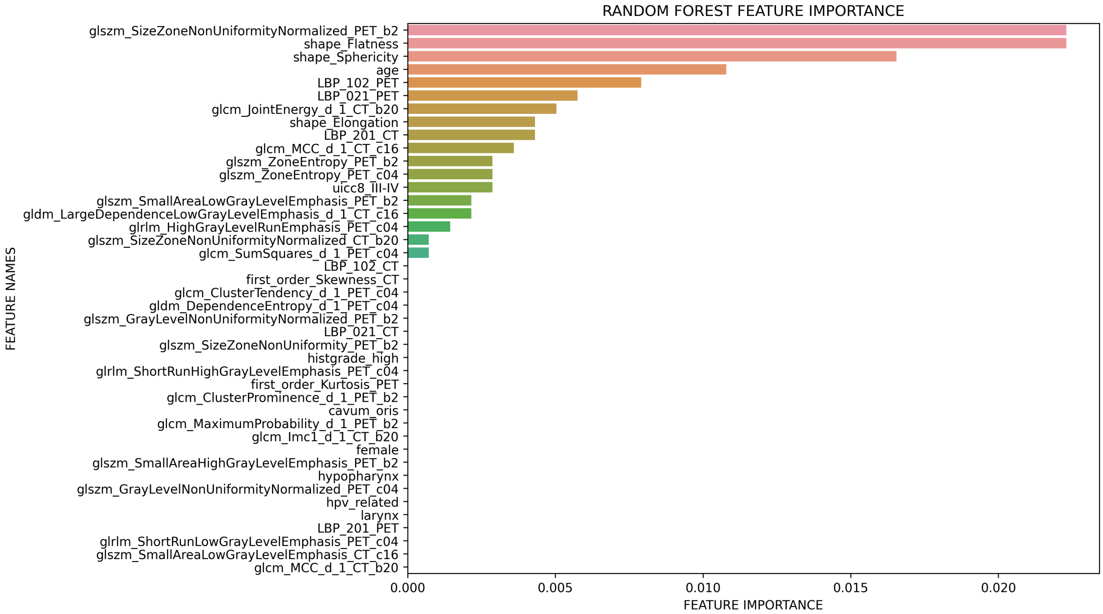


Figure 4.3.7: The feature importances of all the features in dataset DR3 on the Random Forest model on DFS.

The model structure and plot of the feature importance were visualized. Tree 7 of 20 in the Random forest model initially split at the *HPV-status*. The following splits were done on *Small area Low Gray level emphasis* and *first order skewness*. The remaining feature splits occurred on a few shape features concerning the sphericity and LBP features.

As in Section 4.2.5, the feature importance was based on the permutation importance. As seen in Figure 4.3.7, only 18 of the 42 features positively contributed to the model's performance. The most important feature was the *GLSZM Non-uniformity* for PET, meaning the non-uniformity of texture in PET images was an important predictor for DFS for the Random forest model. The two second most important features were the shape features *flatness* and *sphericity*, both considered important by RENT, as seen in Table A.3.3.

Chapter 5

Discussion

This thesis aimed to determine if it was possible to use interpretable models to predict treatment outcomes for patients with head and neck cancer while maintaining state-of-the-art performance. This section discusses the results found in Section 4.3.5.

5.1 Features selected by RENT

This thesis used RENT [50] as the framework of choice for feature selection. In total, three subsets of data were used with RENT for two different targets, yielding six different subsets of data. For the OS target, the results from RENT are presented in Table 4.2.1, with a full version in Appendix A.2. For the DFS target, the complete Table of results can be found in Appendix A.3, with a shorter version in Table 4.3.1.

For the clinical dataset D1, a total of 7 features were selected at least once by RENT for OS, whereas ten features were selected for DFS, suggesting that more features were required for predicting DFS. All seven features selected by RENT for OS were also selected for DFS, in addition to *female*, *larynx*, and *cavum_oris*. Cancer stage and HPV were the two most selected features in RENT for both OS and DFS.

For the radiomics dataset D2, a total of 25 features were selected by RENT for OS and 32 for DFS. Again, RENT selected more features for DFS models than for OS models. For both OS and DFS, the shape feature *tumor sphericity* was the most selected feature, suggesting the roundness of the tumor was an important feature for both OS and DFS prediction. Liu et al. [14] also found that tumor roundness was a signature feature for predicting OS and DFS. Tumor sphericity was also found to be a signature feature for CT radiomics by Keek et al. [57], who also found that rounder tumors had a better prognosis.

For OS on the radiomics dataset, DR2, three texture features were selected with a frequency higher than 30%. These were *glcm_JointAverage_d_1_CT_c16*, which measures the mean gray-level intensity in the intensity distribution of the CT image, *glcm_SumAverage_d_1_CT_c16*, which measures the occurrence of high grey-level pairs and low grey-level pairs, and *glrlm_HighGrayLevelRunEmphasis_PET_c04*, which measures the concentration high gray-level intensity values [35]. Note that texture in CT images is related to the tumor's physical texture, whereas texture in PET images is related to the tumor's uneven tracer uptake. Texture in the PET images reflects the tumor's metabolic activity, as explained in Section 2.5.

For DFS on the radiomics dataset, DR2, other texture features were selected. The local binary pattern features *LBP_102_PET* and *LBP_201_PET* was chosen by RENT with a selection frequency of 95% and 48%, respectively. The LBP features describe texture patterns in the tumor captured by the PET images [58]. The *tumor elongation* was selected at a frequency of 69%, suggesting that it is an important feature for predicting DFS.

For the full dataset D3, which consisted of both radiomics and clinical features, the number of features selected for OS and DFS was quite different. For OS, only seven features were selected from D3, with 6 being clinical features. For DFS, 42 features were selected from the D3 dataset. The OS and DFS target had five common features, *tumor sphericity*, *uicc_III-IV* (cancer stage), *HPV-related*, *cavum_oris*, and *age*, suggesting that these features were important to both OS and DFS prediction.

The PET parameter *SUVpeak* was selected only once by RENT for DFS prediction. The other PET parameters *MTV* and *TLG* were never selected by any of the models in RENT, which suggests that they were not important for the prediction of OS and DFS. The work of Moan et al. [13] also found that the PET parameters had no significance for predicting DFS.

5.2 Performance results

5.2.1 OS

For dataset DR1, every model outperformed the baseline for training and testing. This indicated that models based on RENT-selected clinical features yield a higher performance for OS prediction than the entire dataset of clinical and radiomics features.

During model validation, XGB, RF, and BR from iModels, obtained the highest performance when predicting OS on dataset DR1, the clinical dataset, with AUC around 0.72 and MCC of around 0.45, as seen in Table 4.2.3. However, during external testing on the MAASTRO, the performance dropped for all three classifiers. During external testing, four classifiers had the same performance, the decision tree, XGBoost, HistGrad, and FIGS, as seen in Table 4.2.4. These four had the highest performance among all classifiers during testing, with an AUC of 0.65 and MCC of 0.37. None of them had the same performance during training.

For dataset DR2, consisting of 25 radiomics features, the HistGrad classifier obtained the highest performance with an AUC of 0.78 and MCC of 0.57 during validation on the OUS data. The RF and XGB followed with AUC scores around 0.67, which is not unlike what other studies found. Liu et al. [14] found that for PET and CT radiomics of the primary tumor, an AUC of $0.68 - 0.90$ was achievable, and Vallières et al. [15] obtained an AUC of 0.60 for PET and CT radiomics when predicting OS. The performance of the HistGrad classifier dropped from training to external testing, where the HistGrad achieved an AUC of 0.60. XGBoost did not generalize well and ended up with an AUC of 0.55, just 5% better than random guessing. The highest-performing model across training and testing for dataset DR2 was the Random Forest. The Random forest model generalized well and obtained a test AUC of 0.65, not far from the training AUC and in line with the results from similar studies ([14], [15]).

Dataset DR3 consisted of six clinical and one radiomics feature. The performance during model validation was promising, with BR from iModels obtaining an AUC of 0.76, followed by RF and HistGrad with an AUC of around 0.73, as seen in Table 4.2.3. Again, during external testing on the MAASTRO data, the performance dropped for all classifiers. BR experienced the largest drop, down to an AUC of 0.51, indicating the model did not generalize well. The best performance on the external testing was achieved by XGBoost, with an AUC of 0.68. XGBoost gave a training AUC of 0.71, meaning the model generalized well to the external dataset.

For both datasets, DR2 and DR3, several models were outperformed by the baseline model from Table 4.1. The baseline MCC for validation was 0.180, and the baseline for testing was 0.139 for OS. All classifiers outperformed the baseline during validation for both DR2 and DR3. However, several of the classifiers performed poorly during testing on the MAASTRO data. For dataset DR2, HSTree and BR from iModels obtained an MCC of just 0.078 and 0.062, respectively. Such low MCC scores indicated that the model's predictions were just above random guessing.

In general, for all the datasets, the F1:0 score dropped the most for all classifiers during external validation, indicating that many patients from the MAASTRO clinic were falsely predicted as positives. For the model validation on the OUS data, the F1:0 score was consistently higher than the F1:1 score, indicating that OUS patients were more often predicted as false negatives.

The highest-performing classifier during training when predicting OS was the HistGradBoosting classifier on dataset DR2, the radiomics dataset. The HistGrad achieved an MCC of 0.57, as seen in Table 4.2.3. However, it did not generalize well to the external dataset, where it obtained an MCC of 0.20. The highest performing classifier during testing on the external MAASTRO dataset was the Decision tree, FIGS, and HSTree classifiers on the clinical dataset, DR1, with an MCC of 0.37. The three classifiers had the same performance across all metrics, suggesting they were similar in structure. The three classifiers had training MCCs between $0.37 - 0.39$, suggesting they generalized well to the test data.

5.2.2 DFS

For the DFS target, the classifiers performed very similarly during training on dataset DR1, with all AUCs between $0.64 - 0.68$. For external testing, the model's performances spanned a greater interval with the highest-performing model, the Random forest, achieving an AUC of 0.71 and the lowest-performing model, the Boosted rules from iModels, achieving an AUC of 0.61.

The performance of the classifiers on dataset DR2, the radiomics dataset, was slightly higher during training than those of the clinical dataset, as seen in Table 4.3.3. The XGBoost, decision tree, HSTree, and Boosted rules from iModels, all gave AUCs above 0.71. The models did not generalize well to the test data, where the same models obtained AUCs of $0.51 - 0.59$. XGBoost has the highest performance during testing of all the classifiers with an AUC of 0.59. In general, the test performance for the radiomics dataset was much lower than that of the clinical dataset, indicating that DFS was best predicted for the MAASTRO data on the clinical features rather than the radiomics features.

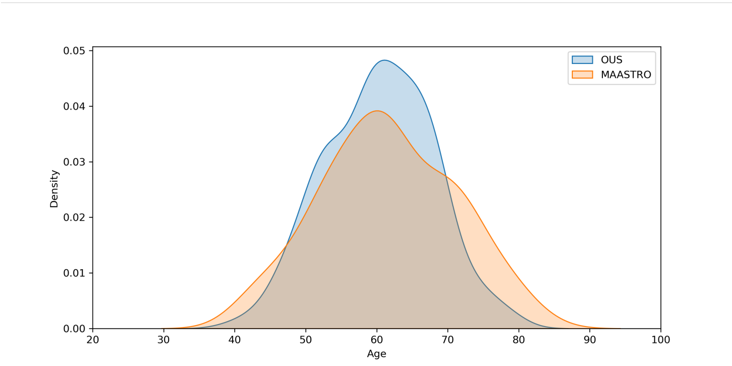
For dataset DR3, the training AUCs for nearly all classifiers were above 0.70, with only RandomForest at 0.67. However, Random Forest achieved a test AUC of 0.64, meaning the model generalized well to the unseen MAASTRO data and performed approximately the same during training and testing. All other classifiers achieved an AUC between $0.51 - 0.55$, similar to randomly guessing the class of each patient. The Random Forest had the highest performance across the training and testing.

For both datasets, DR2 and DR3, the external testing performance of F1:0 was very low for some classifiers. As seen in Table 4.3.4 and Figure 4.3.2, the F1:0 score for the Decision tree, FIGS, HSTree, and Boosted rules was 0.16 for both DR2 and DR3. This indicated that the models predicted many false positives. For datasets DR2 and DR3, several models did not outperform the baseline, with many obtaining an MCC score just above 0, which is the limit for random guessing. For the radiomics-only dataset, DR2, the FIGS classifier even got MCC scores lower than zero, which is worse than random guessing.

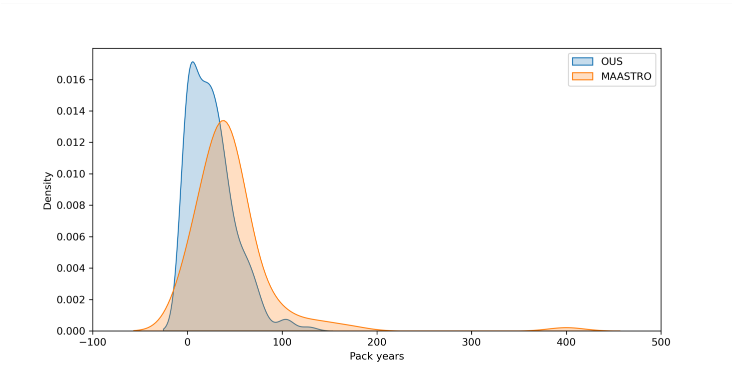
The highest performing classifier on the OUS training data for DFS prediction was the XGBoost classifier on dataset DR2, the radiomics data. XGBoost obtained a training MCC score of 0.46, as seen in Table 4.3.3, on the OUS data. However, it did not generalize well to the test data, which obtained an MCC of 0.19, as seen in Table 4.3.4. The highest-performing classifier on the MAASTRO test data for DFS prediction was the Random forest classifier on the clinical dataset DR1. Random forest obtained an MCC of 0.44 during external testing, as seen in Table 4.3.4, and an MCC of 0.35 during training.

5.3 Possible explanations for performance gap

The two datasets collected from OUS and the MAASTRO clinic had some core differences, as seen in Table 3.1.2. Patients from OUS tended to have fewer pack years, lower stages, and different frequencies of the different cancer sites. For the MAASTRO dataset, there were more patients with laryngeal cancer, which is linked to smoking [59]. This coincides with the higher trends of pack years for the MAASTRO dataset. In the OUS dataset, oropharyngeal cancer was the most common type, accounting for nearly 75% of all the cases. The differences between the features pack years and age in the two datasets are visualized in Figure 5.3.1.



(a) Age distribution



(b) Pack years distribution

Figure 5.3.1: The distributions of age and pack years in the OUS and MAASTRO datasets

Figure 5.3.1a shows that the average age of the OUS and MAASTRO patients was about the same, as seen in Table 3.1.2. The tails of the MAASTRO age curve are slightly longer, meaning the MAASTRO patients had a greater age span than the OUS patients. Figure 5.3.1b shows that the MAASTRO patients, on average, smoked more than the OUS patients and that someone in the MAASTRO dataset had nearly 400 pack years.

Across all the models, the most common pattern was that the models performed higher during training than on validation. For DFS on dataset DR1, the models predicted much better on the MAASTRO data after training on the OUS data. The patients from the MAASTRO clinic were generally sicker than the OUS, with a higher percentage of patients in the later stages of cancer. Predicting the outcome of patients might be easier if the patients, in general, are sicker. In general, models that got accuracies and AUC around 0.7 for training on the OUS data obtained a lower performance on the MAASTRO dataset and vice versa. The two datasets have core differences that can explain the drop in performance or the increase for dataset DR1 on DFS prediction.

Overoptimistic results could cause a reason for the drop in performance between training and testing. Throughout k-fold cross-validation, all of the data has at some point been training the model, meaning there is an information bleed between the training data and the validation data for all classifiers. This could lead to a seemingly over-optimistic performance during the model validation.

5.4 Interpretability and overall assessment of the models

Implementing machine learning in the medical field offers many possibilities but also proposes some challenges. One of these challenges is interpretability. Many ML models are so-called black box models that offer no insight into how the prediction is made. When diagnosing and administering treatment to a patient, the consequences of wrong predictions are severe. The ideal implementation of ML models for clinical decision support is to use easily explainable models with state-of-the-art performance.

The iModels package was explored in this thesis because of its promise of high-performing interpretable models. Several of the models from the iModels package proved less effective than RandomForest, XGBoost, and HistogramGradient models for several datasets in this thesis. This is likely due to the simplicity of the iModels algorithms. The iModels package offers an array of models for interpretable machine learning. In the FIGS article, [42], an interpretable model was coined to have less than 20 decision rules. For complicated datasets, like those enriched with radiomics features, 20 splits may be too few to catch the data patterns. The same is true for the Decision Tree algorithm. As seen in Table 4.2.4 and 4.3.4, the Decision tree performs poorly during external validation of the MAASTRO data for the radiomics datasets.

The models interpreted in Section 4.2.5 and 4.3.5 had different complexities depending on which datasets they were built on. The models built on the clinical datasets had simple structures. The decision tree performed well on the clinical datasets for both OS and DFS prediction, as well as other simple algorithms. The Random forest, XGBoost, and HistGradeBoosting classifiers had the highest performances on the more complicated datasets. The HistGradientBoosting was not visualized here, as it is less interpretable than the other models.

The features found most important by the feature importance permutation were the same for the RENT-selected models. *Sphericity* was one of the most important feature in both feature importance diagrams for DR3, which was confirmed by RENT. Other studies have also found tumor sphericity to be an important predictor for OS and DFS [14], [57]. For the XGBoost on the DR3 dataset for OS prediction, several of the features in the dataset have a negative feature importance, indicating that their presence in the dataset degrades the performance slightly, as seen in Figure 4.2.8. This caused a discrepancy between the features considered important by RENT and the features considered important by the model. Note that this is for the XGBoost model and that the RENT feature selection chose features based on the logistic regression algorithm.

For OS prediction on the clinical data, a Decision tree with just one rule, as seen in Figure 4.2.3 obtained an MCC of $o.37$ during training and testing. The Decision tree's only rule was that patients with cancer stage $I - II$ were predicted to survive, and III_IV were predicted to not. During tuning, the model had the possibility of choosing a more complex structure. However, this simple depth of 1 was chosen. With only one rule, predicting OS with an MCC of 0.37 on these datasets was highly possible.

Further, for OS prediction, the Random forest model on the clinical data only obtained a training MCC of 0.447 and a test MCC of 0.31. The Random forest model on the clinical and radiomics data obtained a training MCC of 0.49 and a test MCC of 0.21. The model on only the clinical data had a 10% better performance on the unseen data, while the two models performed around the same on the training data. The clinical Random forest model also had a simpler structure, as seen in Figure 4.2.4, with four trees with a maximum depth of 3, whereas the clinical and radiomics model consisted of 12 trees with a depth of 5. The most interpretable Random forest model outperformed the more complicated model.

For DFS prediction, both the Decision tree and the FIGS algorithm obtained MCCs of $0.30 - 0.42$ during training and testing. Both models were inherently interpretable in their structure and were visualized in Figure 4.3.3 and 4.3.4. Both trees obtained a high performance while keeping their interpretability.

The Random forest on the clinical data consisted of 8 estimators with one rule each. Four estimators were visualized in Figure 4.3.5. This model obtained a training -MCC of 0.35 and a test MCC of 0.44. The Random forest model on the clinical and radiomics data obtained a training MCC of 0.42 and a test MCc of 0.33. Overall the two Random forest models perform almost identically, where one predicts better on the OUS data, and the other predicts better on the MAASTRO data. The RF model on the clinical and radiomics data consists of 20 trees, each with a maximum depth of 10. This model is considerably more complicated and less interpretable than the model for the clinical data, yet they perform about the same.

For both OS and DFS, the clinical datasets gave rise to the models with the highest performances in this thesis. Several classifiers were simple trees that were easily visualized and understood, indicating that it is possible to predict treatment outcomes at state-of-the-art performance using interpretable models.

5.5 Future Work

Other algorithms promote interpretability, such as Bayesian networks [60] and Bayesian rule lists [61], which both offer interpretable probabilistic classifiers. The minimum description length (MDL)-based rule lists are another classification algorithm that offers interpretable rule lists for classification. MDL-based rule lists do not require any tuning, as it is a hyperparameter-free algorithm [62]. Conducting a comparison between these algorithms and the iModels algorithms would be an interesting approach.

Identifying the misclassified samples across the 100 models during training would lead to more insight into which patients are harder to classify. Determining where the model falls short, through which patients are hard to classify, could be an interesting addition to a model used for decision support. Because the different cancer sites were so differently distributed amongst the two datasets, and cancer sites often were coined as important features for the models, splitting the dataset by the different cancer sites could lead to better predictions.

Chapter 6

Conclusion

This thesis aimed to find interpretable models for predicting patient outcomes of head and neck cancer patients. The analysis was based on decision trees to produce visualizations that could provide context to the model's predictions. Two responses were predicted; overall survival and disease-free survival. Radiomics features were extracted from the images using imskaper [56]. The data was separated into three datasets, one for the clinical data, one for the radiomics data, and one for all of the data. Feature selection on the three was done using the repeated elastic net technique [50]. This resulted in six datasets, three for each response. Seven machine learning classifiers were trained and tested on the datasets. A dataset from Oslo University Hospital was used for training all the classifiers. This dataset contained 139 patients. For testing, a dataset from the MAASTRO clinic was used, which contained 99 patients.

For predicting overall survival, the highest performance was achieved by the HistGradientBoosting classifier during training, with a training MCC of 0.57. However, the model did not generalize well to the external testing. This performance was obtained on the radiomics only dataset. The highest test performance was an MCC of 0.37 which was obtained by the Decision tree, FIGS, and HSTree classifiers on the clinical data.

For predicting disease-free survival, the XGBoost classifier achieved the highest performance on dataset DR2, with an MCC of 0.46 during training. The model did not generalize well to the external MAASTRO data, where it obtained a test MCC of 0.19. The highest performance on the MAASTRO data was achieved by the Random forest classifier, with an MCC of 0.44.

The Decision tree models, for the most part, performed well on the clinical data. However, it struggled to grasp the complexity of the radiomics datasets and was outperformed by other algorithms for the radiomics datasets. The high performance on the clinical dataset and the interpretability makes the decision tree a good candidate for use in various applications.

The iModels did not outperform the other classifiers. However, some did perform well on the clinical datasets for both OS and DFS prediction. For the more complex datasets with many radiomics features, the iModels were outperformed by the Random forest, XGBoost, and the HistGradientBoosting classifiers.

Several interpretable models predicted well on clinical data, whereas more complex models were needed to capture the patterns in the radiomics data. Transparency in machine learning models greatly benefits decision-makers in clinical settings, as every prediction can be reasoned for, contributing to a greater understanding. Predicting treatment outcomes for head and neck patients is highly possible with interpretable models. In order to determine if the methods used in this thesis are suited for predicting treatment outcomes for head and neck cancer patients, it is necessary to test the models on more datasets.

Bibliography

[1] S. Raschka and V. Mirjalli, *Python Machine Learning*, ch. 1, 3. Birmingham B3 2PB, UK: Packt Publishing Ltd., 3 ed., 2019.

[2] C. Singh, K. Nasseri, Y. S. Tan, T. Tang, and B. Yu, “imodels: a python package for fitting interpretable models,” 2021.

[3] International Agency for Research on Cancer, “Data visualization tools for exploring the global cancer burden in 2020,” 2020.

[4] Kreftforeningen, “Kreft i norge,” 2023. (Last accessed 11 May 2023).

[5] Kreftregisteret, “Undersøkelser,” 2023. (Last accessed 11 May 2023).

[6] Helse Norge, “Hode- og halskreft,” 2023. (Last accessed 17 @April 2023).

[7] A. Hoeben, E. A. J. Joosten, and M. H. J. van den Beuken-van Everdingen, “Personalized medicine: Recent progress in cancer therapy.,” *Cancers*, vol. 13, 2021.

[8] H. SAerts, E. Velazquez, and R. e. a. Leijenaar, “Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach.,” *Nature Communications*, vol. 5, 2014.

[9] A. Marusyk, M. Janiszewska, and K. Polyak, “Intratumor heterogeneity: The rosetta stone of therapy resistance,” *Cancer cell*, vol. 37, pp. 471–484, 2020.

[10] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nat Mack Intell*, vol. 1, pp. 206–215, 2019.

[11] R. J. Gillies, P. E. Kinahan, and H. Hricak, “Radiomics: Images are more than pictures, they are data,” *Radiology*, vol. 278, pp. 563–577, February 2016.

[12] A. Zwanenburg, S. Leger, M. Vallières, and S. Löck, "Image biomarker standardization initiative," 2016.

[13] J. M. Moan, C. Dlephin Amdal, E. Malinen, J. Grååådal Svestad, T. Velde Bogsrud, and E. Dale, "The prognostic role of 18f-fluordeoxyglucose pet in head and neck cancer depends on hpv status," *Radiotherapy and oncology*, vol. 140, no. 2, pp. 54–61, 2019.

[14] Z. Liu, Y. Cao, W. Diao, Y. Cheng, Z. Jia, and X. Peng, "Radiomics-based prediction of survival in patients with head and neck squamous cell carcinoma based on pre- and post-treatment 18f-pet/ct.," *Aging*, vol. 12, no. 14, pp. 14593—14619, 2020.

[15] M. Vallières, E. Kay-Rivest, L. J. Perrin, X. Liem, C. Furstoss, H. J. W. L. Aerts, N. Khaouam, P. F. Nguyen-Tan, C. S. Wang, K. Sultanem, J. Seuntjens, and I. El Naqa, "Radiomics strategies for risk assessment of tumour failure in head-and-neck cancer," *Scientific reports*, vol. 7, no. 1, 2017.

[16] S.-W. Chen, W.-C. Shen, Y.-C. Lin, R.-Y. Chen, T.-C. Hsieh, K.-Y. Yen, and C.-H. Kao, "Correlation of pretreatment 18f-fdg pet tumor textural features with gene expression in pharyngeal cancer and implications for radiotherapy-based treatment outcomes," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 44, pp. 1–14, 04 2017.

[17] Mayo Clinic, "Cancer," 2022. (Last accessed 19 January 2023).

[18] American Cancer Society, "What is cancer?," 2022. (Last accessed 17 April 2023).

[19] Cleveland Clinic, "Metastasis (metastatic cancer)," 2021. (Last accessed 17 April 2023).

[20] Cancer Registry of Norway, "Cancer in norway 2021," 2021.

[21] D. A. Anand SS, Singh H, "Clinical applications of pet and pet-ct," *Med J Armed Forces India*, vol. 65, no. 4, 2009.

[22] National Cancer Institute, "Head and neck cancers," 2021. (Last accessed 17 April 2023).

[23] Australian Radiation Protection and Nuclear Safety Agency, "Radioactivity." (Last accessed 17 @April 2023).

[24] J. Lilley, *Nuclear Physics - Principles and Applications.* West Sussex PO19 1UD, England: John Wiley & Sons Ltd, 1 ed., 2001.

[25] R. B. Workman and R. E. Coleman, *Fundamentals of PET and PET/CT Imaging*, pp. 1–22. New York, NY: Springer New York, 2006.

[26] J. G. Webster, *Medical Instrumentation - Application and Design*, ch. 12. John Wiley & Sons Ltd, 4 ed., 2010.

[27] L. E. Romans.

[28] *Nuclear Medicine Physics.* Non-serial Publications, Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY, 2015.

[29] A. Kamal, *Nuclear Physics.* Heidelberg: Springer Berlin, Heidelberg, 1 ed., 2001.

[30] S. S. M. Wong, *Introductory Nuclear Physics*, ch. 2, 9. Germany: Wiley-VCH, 2 ed., 2004.

[31] A. Almuhaideb, N. Papathanasiou, and J. Bomanji, "18f-fdg pet/ct imaging in oncology," *Annals of Saudi Medicine*, vol. 31, no. 1, pp. 3–13, 2011.

[32] E. Lin and A. Alessio, "What are the basic concepts of temporal, contrast, and spatial resolution in cardiac ct?," *Journal of Cardiovascular Computed Tomography*, vol. 3, no. 6, pp. 403–408, 2009.

[33] J. W. Lee and S. M. Lee, "Radiomics in oncological pet/ct: Clinical applications," *Nuclear Medicine and Molecular Imaging*, vol. 52, pp. 170–189, October 2018.

[34] J. P. O'Connor, C. J. Rose, J. C. Waterton, R. A. Carano, G. J. Parker, and A. Jackson, "Imaging intratumor heterogeneity: role in therapy response, resistance, and clinical outcome," *Clinical cancer research : an official journal of the American Association for Cancer Research*, vol. 21, pp. 249–257, January 2015.

[35] J. J. M. van Griethuysen, A. Fedorov, C. Parmar, A. Hosny, N. Aucoin, V. Narayan, R. G. H. Beets-Tan, J. C. Fillon-Robin, S. Pieper, and H. J. W. L. Aerts, "Computational radiomics system to decode the radiographic phenotype," 2017.

[36] M. E. Mayerhoefer, A. Materka, G. Langs, I. Häggström, P. Szczypiński, P. Gibbs, and G. Cook, "Introduction to radiomics," *Journal of nuclear medicine : official publication, Society of Nuclear Medicine,*, vol. 61, pp. 488–495, February 2020.

[37] M. R. T. Dale, P. Dixon, M.-J. Fortin, P. Legendre, D. E. Myers, and M. S. Rosenberg, "Conceptual and mathematical relationships among methods for spatial analysis," *Ecography*, vol. 25, no. 5, pp. 558–577, 2002.

[38] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[39] A. Guryanov, *Histogram-Based Algorithm for Building Gradient Boosting Ensembles of Piecewise Linear Decision Trees*, pp. 39–50. 12 2019.

[40] Q. M. Nguyen, N. Khanh Le, and L. M. Nguyen, "Scalable and secure federated xgboost," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.

[41] T. J. Fan, "Deep dive into scikit-learn's histgradientboosting classifier."

[42] Y. S. Tan, C. Singh, K. Nasseri, A. Agarwal, and B. Yu, "Fast interpretable greedy-tree sums (figs)," *arXiv preprint arXiv:2201.11931*, 2022.

[43] A. Agarwal, Y. S. Tan, O. Ronen, C. Singh, and B. Yu, "Hierarchical shrinkage: Improving the accuracy and interpretability of tree-based methods," *ArXiv:2202.00858*, 2022.

[44] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[45] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 31, pp. 3–13, 2020.

[46] W. Lydiatt, S. Patel, B. O'Sullivan, M. Brandwein, J. Ridge, J. Migliacci, A. Loomis, and J. Shah, "Head and neck cancers—major changes in the american joint committee on cancer eighth edition cancer staging manual," *CA Cancer Journal for Clinicians*, vol. 67, pp. 122–137, Mar. 2017. Publisher Copyright: © 2017 American Cancer Society.

[47] National Cancer Institute, "histologic grade." (Last accessed 6 May 2023).

[48] W. McKinney, "Data structures for statistical computing in python," pp. 56–61, 01 2010.

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

[50] A. Jenul, S. Schrumer, K. H. Liland, U. G: Indahl, C. M. Futsæther, and O. Tomic, "Rent - repeated elastic net technique for feature selection," 01 2021.

[51] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," 2019.

[52] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[53] M. L. Waskom, "seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.

[54] T. Parr and P. Grover, "How to visualize decision tree." `https://explained.ai/decision-tree-viz/`, 2023. [Accessed 13-May-2023].

[55] Microsoft Corporation, "Microsoft excel."

[56] NMBU Data Science, "Imskaper."

[57] S. A. Keek, F. W. R. Wesseling, H. C. Woodruff, J. E. van Timmeren, I. H. Nauta, T. K. Hoffmann, S. Cavalieri, G. Calareso, S. Primakov, R. T. H. Leijenaar, L. Licitra, M. Ravanelli, K. Scheckenbach, T. Poli, D. Lanfranco, M. R. Vergeer, C. R. Leemans, R. H. Brakenhoff, F. J. P. Hoebers, and P. Lambin, "A prospectively validated prognostic model for patients with locally advanced squamous cell carcinoma of the head and neck based on radiomics of computed tomography images," *Cancers*, vol. 13, no. 13, 2021.

[58] K. Chauhan and R. Chauhan, *Image Processing for Automated Diagnosis of Cardiac Diseases*. 07 2021.

[59] P. Menach, H. O. Oburra, and A. Patel, "Cigarette smoking and alcohol ingestion as risk factors for laryngeal squamous cell carcinoma at kenyatta national hospital, kenya.," *Clinical medicine insights. Ear, nose and throat*, vol. 5, no. 14, pp. 17–24, 2012.

[60] B. Mihaljević, C. Bielza, and P. Larrañaga, "Bayesian networks for interpretable machine learning and optimization," *Neurocomputing*, vol. 456, pp. 648–665, 2021.

[61] H. Yang, C. Rudin, and M. Seltzer, "Scalable bayesian rule lists," 2017.

[62] H. M. Proença and M. van Leeuwen, "Interpretable multiclass classification by MDL-based rule lists," *Information Sciences*, vol. 512, pp. 1372–1393, feb 2020.

Appendix A

RENT

A.1 RENT input parameters

The input parameters for running the RENT algorithm for feature selection is presented below in Table A.1.1. Each parameter is presented with a description and the range used for each parameter.

Table A.1.1: The parameter ranges used for training the ensemble of models through the RENT framework. The input parameters are described, and their respective ranges and values are presented.

Parameter	Description	Range
C	Controls the amount of L2 regularization	[0.1, 1, 10, 100, 1000]
L1-ratio	Controls the amount of L1 regularization	[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
K	Number of models in ensemble	100
testsize_range	Allows more randomness into the model by varying the size of the subsets	(0.25, 0.25)
random_state	Sets a value to keep results reproducible	None
autoEnetParSel	Determines how the best regularization parameters are found	False
poly	If ON then combinations of and interactions of features can also be used to train the models	OFF
scoring	Performance metric for the models	MCC
classifier	The chosen classification algorithm	logreg

A.2 Features selected by RENT at least once for OS target

This section presents the full tables with results of running RENT on the clinical data (D1), the radiomics data (D2) and the combined data (D3).

Table A.2.1: Table of features selected by RENT at least once across 100 models for dataset D1, the clincal data, and OS.

Feature	Frequency (%)
Cancer stage	100
HPV-related	95
Pack years	47
Oropharynx	36
age	6
charlson	4
Cavum oris	4

Table A.2.2: Table of features selected by RENT at least once across 100 models on dataset D2, the radiomics data, for OS.

Feature	Frequency (%)
shape_Sphericity	100
glcm_JointAverage_d_1_CT_c16	79
glcm_SumAverage_d_1_CT_c16	79
shape_MajorAxisLength	57
first_order_Maximum_CT	35
glrlm_HighGrayLevelRunEmphasis_PET_c04	34
shape_Maximum3DDiameter	31
glcm_ClusterShade_d_1_PET_b2	25
gldm_LargeDependenceLowGrayLevelEmphasis_d_1_CT_c16	17
gldm_LargeDependenceHighGrayLevelEmphasis_d_1_CT_c16	12
glrlm_LowGrayLevelRunEmphasis_PET_c04	7
glcm_JointAverage_d_1_PET_c04	7
glcm_SumAverage_d_1_PET_c04	7
LBP_102_PET	7
first_order_Minimum_PET	6
glcm_Autocorrelation_d_1_PET_c04	5
glszm_ZoneEntropy_CT_b20	5
glrlm_ShortRunHighGrayLevelEmphasis_PET_c04	4
shape_Maximum2DDiameterSlice	3
gldm_DependenceVariance_d_1_CT_b20	3
first_order_Skewness_PET	3
LBP_210_CT	2
first_order_Skewness_CT	2
glszm_GrayLevelNonUniformityNormalized_CT_b20	2
ngtdm_Busyness_d_1_PET_c04	2
gldm_HighGrayLevelEmphasis_d_1_CT_c16	2
glcm_Autocorrelation_d_1_CT_c16	2
first_order_Range_CT	1
ngtdm_Busyness_d_1_CT_b20	1
ngtdm_Busyness_d_1_PET_b2	1
shape_Elongation	1
gldm_HighGrayLevelEmphasis_d_1_PET_c04	1

Table A.2.3: Table of features selected by RENT at least once for the combination of clinical and radiomics data, dataset D3, for OS

Feature	Frequency (%)
$shape_Sphericity$	100
$Cancer\ Stage$	88
$HPV-related$	86
$oropharynx$	12
$pack\_years$	6
$cavum\_oris$	2
Age	2

A.3 Features selected by RENT at least once for DFS target

This section presents the full tables with RENT selected features for datasets D1, D2, and D3.

Table A.3.1: Table of features selected by RENT at least once across 100 models for dataset D1 for DFS

Feature	Frequency (%)
$hpv\_related$	98
$uicc8\_III-IV$	89
$pack\_years$	41
$cavum\_oris$	39
$oropharynx$	37
$larynx$	4
$charlson$	2
$female$	1
age	1
$SUVpeak$	1

Table A.3.2 presents all features selected by RENT at least once, with their respective frequencies for dataset D2, the radiomics dataset.

Table A.3.2: Table of features selected by RENT at least once across 100 models for dataset D2 for DFS

Feature	Frequency (%)
shape_Sphericity	95
LBP_102_PET	95
shape_Elongation	69
glszm_SmallAreaLowGrayLevelEmphasis_CT_c16	63
LBP_201_PET	48
glszm_GrayLevelNonUniformityNormalized_PET_c04	31
LBP_201_CT	12
shape_Flatness	9
glrlm_ShortRunLowGrayLevelEmphasis_PET_c04	9
glszm_ZoneEntropy_PET_b2	8
$LBP_{0}21_{P}ET$	4
glszm_SizeZoneNonUniformityNormalized_PET_b2	3
glszm_ZoneEntropy_PET_c04	2
gldm_LargeDependenceLowGrayLevelEmphasis_d_1_CT_c16	2
glszm_SmallAreaHighGrayLevelEmphasis_PET_b2	2
glcm_MaximumProbability_d_1_PET_b2	1
LBP_003_CT	1
LBP_300_CT	1
glszm_GrayLevelVariance_PET_c04	1
LBP_102_CT	1
glrlm_ShortRunHighGrayLevelEmphasis_PET_c04	1
glcm_SumSquares_d_1_PET_c04	1
glrlm_HighGrayLevelRunEmphasis_PET_c04	1
first_order_Minimum_PET	1
glszm_GrayLevelNonUniformityNormalized_PET_b2	1

Table A.3.3 presents all features selected by RENT at least once, with their respective frequencies for dataset D3, the combined clinical and radiomics dataset.

Table A.3.3: Table of features selected by RENT at least once across 100 models for dataset D3 for DFS

Feature	Frequency (%)
shape_Sphericity	98
shape$_{E}$longation	95
LBP_102_PET	94
glszm_SmallAreaLowGrayLevelEmphasis_CT_c16	85
LBP_201_PET	68
hpv_related	55
glszm_GrayLevelNonUniformityNormalized_PET_c04	49
uicc_III − IV	47
shape_Flatness	18
glszm_ZoneEntropy_PET_b2	17
glrlm_ShortRunLowGrayLevelEmphasis_PET_c04	16
glszm_SmallAreaHighGrayLevelEmphasis_PET_b2	13
cavum_oris	12
LBP_201_CT	11
age	10
female	9
LBP_021_PET	8
larynx	5
hypopharynx	5
glszm_SizeZoneNonUniformityNormalized_CT_b20	3
glrlm_HighGrayLevelRunEmphasis_PET_c04	3
glszm_ZoneEntropy_PET_c04	2
glcm_SumSquares_d_1_PET_c04	1
glcm_SumSquares_d_1_PET_c04	2
glcm_ClusterProminence_d_1_PET_b2	2
glszm_SizeZoneNonUniformityNormalized_PET_b2	2
glszm_SizeZoneNonUniformity_PET_b2	2
glszm_SmallAreaLowGrayLevelEmphasis_PET_b2	2
glszm_GrayLevelNonUniformityNormalized_PET_b2	2
histgrade_high	2
glcm_Imc1_d_1_CT_b20	1
glcm_MaximumProbability_d_1_PET_b2	1
first_order_Kurtosis_PET	1
gldm_LargeDependenceLowGrayLevelEmphasis_d_1_CT_c16	1
LBP_102_CT	1
glrlm_ShortRunHighGrayLevelEmphasis_PET_c04	1
glcm_JointEnergy_d_1_CT_b20	1

Appendix B

Optuna

B.1 Optuna input parameters

Optuna takes in an array of different hyperparameters for tuning a classifier, and the hyperparameter ranges for all classifiers are presented in this section. The same input hyperparameters were used for all classifiers and targets.

B.1.1 Decision Tree Classifier

The input hyperparameters of the Decision Tree Classifier are presented in Table B.1.1. Description of the hyperparameters are included, along with the ranges and values.
In Table B.1.1, the input hyperparameters for the DecisonTreeClassifier are presented.

Table B.1.1: Input parameter range for running Optuna on a Decision Tree classifier

Hyperparameter	Range
max_depth	1 - 20
criterion	gini, entropy

B.1.2 Random Forest Classifier

Table B.1.2 presents the hyperparameter ranges provided to Optuna to tune the Random Forest Classifier. The desr

Table B.1.2: Input parameter range for running Optuna on a Random Forest classifier. A description of each hyperparameter is included, with the range and values.

Hyperparameter	Range
n_estimators	1 - 20
max_depth	1 - 20
criterion	gini, entropy

B.1.3 XGBoost

Table B.1.3 presents the hyperparameter ranges provided to the optimizer Optuna for the XGB Classifier.

Table B.1.3: Input hyperparameter range for running Optuna on a XGBoost classifier, along with a description of each hyperparameter.

Hyperparameter	Range
booster	gblinear, gbtree, dart
lambda	1e-8 - 1.0, log=True
alpha	1e-8 - 1.0, log=True
subsample	0.2 - 1.0
colsample_bytree	0.2 - 1.0
if booster = gbtree	
max_depth	1 - 9
eta	1e-8 - 1.0, log=True
gamma	1e-8 - 1.0
grow_policy	depthwise, lossguide

B.1.4 Histogram Gradient Boosting Classifier

Table B.1.2 presents the hyperparameter ranges provided to the optimizer Optuna for the Histogram Gradient Boosting Classifier.

Table B.1.4: Input hyperparameter range for running Optuna on a Histogram Gradient Boosting Classifier, along with a description for each hyperparameter.

Hyperparameter	Range
learning rate	0.5 - 0.2
max_depth	1 - 7
min_samples_leaf	1 - 30
l2_regularization	0 - 0.1
max_bins	2 - 255

B.1.5 FIGS Classifier

Table B.1.2 presents the hyperparameter ranges provided to the optimizer Optuna for the Histogram Gradient Boosting Classifier.

Table B.1.5: Input hyperparameter range for running Optuna on a FIGS Classifier, along with a decription of each parameter

Hyperparameter	Range
max_rules	1 - 20
min_impurity_decrease	0.0 - 1.0
max_features	sqrt, log2

B.1.6 Hierarchical Shrinkage Classifier

Table B.1.2 presents the hyperparameter ranges provided to the optimizer Optuna for the Hierarchical Shrinkage Classifier.

Table B.1.6: Input hyperparameter range for running Optuna on a HSTree Classifier, along with a description of each hyperparameter

Hyperparameter	Range
reg_param	0 - 10
max_leaf_nodes	2 - 30
$shrinkage\_scheme$	$node\_based$, $leaf\_based$

B.1.7 Boosted Rules Classifier

Table B.1.2 presents the hyperparameter ranges provided to the optimizer Optuna for the Boosted Rules Classifier.

Table B.1.7: Input hyperparameter range for running Optuna on a Boosted Rules Classifier, along with a description of each hyperparameter

Hyperparameter	Range
n_estimators	1 - 20
learning_rate	0.5 - 2.0

B.2 Optuna output for OS target

The result of running Optuna is a set of hyperparameters that yielded the best results for the classifier on a dataset. In this section the results of ML models run on dataset DR1, DR2, and DR3 are presented for the OS target. The results of Optuna on the DFS target are presented in Section B.3.

B.2.1 Decision Tree

Table B.2.1: Optimal hyperparameters chosen by Optuna for Decision Tree Classifier on datasets DR1, DR2, and DR3 with OS target

Hyperparameter	DR1	DR2	DR3
max_depth	2	7	3
criterion	gini	entropy	gini

B.2.2 Random Forest

Table B.2.2: Optimal hyperparameters chosen by Optuna for Random Forest Classifier on dataset DR1, DR2, and DR3 with OS target

Hyperparameter	DR1	DR2	DR3
n_estimators	4	5	20
max_depth	11	18	10
criterion	gini	entropy	entropy

B.2.3 XGBoost

Table B.2.3: Optimal hyperparameters chosen by Optuna for XGBoost Classifier on dataset D1 with OS target.

Hyperparameter	DR1	DR2	DR3
n_estimators	8	16	9
learning_rate	0.0676	0.0474	0.0046
lambda	-	-	0.0048
alpha	-	-	0.0002
max_depth	13	13	9
subsample	0.5917	0.5640	0.3458
colsample_bytree	0.6051	0.2251	0.3322
gamma	1.2528	0.6578	1.8712e-08
booster	gbtree	dart	gbtree
grow_policy	-	-	depthwise

B.2.4 HisGradientBoosting

Table B.2.4: Optimal hyperparameters chosen by Optuna for HistGradientBoosting Classifier on dataset D1 with OS target

Hyperparameter	DR1	DR2	DR3
learning_rate	0.6207		0.1914
max_depth	6		1
min_samples_leaf	11		30
l2_regularization	0.0742		0.0390
max_bins	87		136

B.2.5 FIGS

Table B.2.5: Optimal hyperparameters chosen by Optuna for FIGS Classifier on dataset DR1 with OS target

Hyperparameter	DR1	DR2	DR3
max_rules	2	18	8
max_trees	11	5	13
min_impurity_decrease	0.9376	0.9604	0.9993
max_features	sqrt	sqrt	log2

B.2.6 HSTree

Table B.2.6: Optimal hyperparameters chosen by Optuna for HSTree Classifier on dataset DR1 with OS target

Hyperparameter	DR1	DR2	DR3
reg_param	6.0168	7.1762	4.5546
max_leaf_nodes	2	2	12
cv	7	8	5
shrinkage_scheme	*leaf_based*	*node_based*	*node_based*

B.2.7 Boosted Rules

Table B.2.7: Optimal hyperparameters chosen by Optuna for Boosted Rules Classifier on dataset DR1 with OS target

Hyperparameter	DR1	DR2	DR3
n_estimators	1	7	16
learning_rate	0.7327	0.7120	0.5502

B.3 Optuna output for DFS target

Optuna was used to tune classifiers on all three datasets DR1, DR2, and DR3 with the DFS target. The optimal parameters are presented in this section.

B.3.1 Decision Tree

The results from running Optuna on the Decision Tree Classifier with the input hyperparameters defined in Table B.1.1, are presented in this section in Table B.3.1.

Table B.3.1: Optimal hyperparameters chosen by Optuna for Decision Tree Classifier on datasets DR1, DR2, and DR3 with DFS target

Hyperparameter	DR1	DR2	DR3
max_depth	2	7	1
criterion	gini	entropy	entropy

B.3.2 Random Forest

This section presents the result from running Optuna on a Random Forest classifier with the input hypparameters from Table B.1.2. The results are shown in Table B.3.2.

Table B.3.2: Optimal hyperparameters chosen by Optuna for Random Forest Classifier on dataset DR1, DR2, and DR3 with DFS target

Hyperparameter	DR1	DR2	DR3
n_estimators	8	19	15
max_depth	1	10	2
criterion	entropy	entropy	gini

B.3.3 XGBoost

In this section, the results from running Optuna on an XGBoost classifier with the input hyperparameters form Table B.1.3 are presented. The results are found in Table B.3.3.

Table B.3.3: Optimal hyperparameters chosen by Optuna for XGBoost Classifier on dataset DR1, DR2, and DR3 with DFS target

Hyperparameter	DR1	DR2	DR3
n_estimators	8	20	
learning_rate	0.0677	0.0198	0.4344
lambda	-	3.6730e-7	1.9295e-05
alpha	-	5.3980	4.0553e-07
max_depth	13	5	2
subsample	0.5917	0.5767	0.7134
colsample_bytree	0.6050	0.6692	0.8715
gamma	1.2527	0.0601	-
booster	gbtree	gbtree	gblinear
grow_policy		depthwise	

B.3.4 HisGradientBoosting

The results from running Optuna on a Histogram Gradient Boosting Classifier on datasets DR1, DR2, and DR2 are presented below in B.2.4. The hyperparameters fed to Optuna were presented in Table B.1.4.

Table B.3.4: Optimal hyperparameters chosen by Optuna for HistGradientBoosting Classifier on dataset DR1, DR2, and DR3 with DFS target

Hyperparameter	DR1	DR2	DR3
learning_rate	0.6206	0.1780	0.0503
max_depth	6	9	13
min_samples_leaf	11	15	30
l2_regularization	0.07423	0.2367	0.04860
max_bins	87	87	218

B.3.5 FIGS

The parameters from running Optuna on a FIGS classifier using the hyperparameter ranges from Table B.1.5

B.3.6 HSTree

Table B.3.6: Optimal hyperparameters chosen by Optuna for HSTree Classifier on dataset DR1 with DFS target

Hyperparameter	DR1	DR2	DR3
reg_param	6.01687	3.4427	5.5848
max_leaf_nodes	2	2	2
cv	7	2	5
shrinkage_scheme	leaf_based	node_based	leaf_based

B.3.7 Boosted Rules

Table B.3.7: Optimal hyperparameters chosen by Optuna for Boosted Rules Classifier on dataset DR1 with DFS target

Hyperparameter	DR1	DR2	DR3
n_estimators	1	1	2
learning_rate	0.7327	1.3015	1.2310

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway

Table B.3.5: Optimal hyperparameters chosen by Optuna for FIGS Classifier on datasets DR1, DR2, and DR3 with DFS target

Hyperparameter	DR1	DR2	DR3
max_rules	2	1	1
max_trees	11	19	13
min_impurity_decrease	0.9376	0.6474	0.3065
max_features	sqrt	None	None