INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA INFORMÁTICA

Machine learning models in decision support systems for diagnosing colorectal cancer based on metabolic profiles

RUI XAVIER FERREIRA BARBOSA

Outubro de 2023

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Machine learning models in decision support systems for diagnosing colorectal cancer based on metabolic profiles

Rui Xavier Ferreira Barbosa

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

Supervisor: Professor José Reis Tavares
Co-Supervisor: Professor Isabel Praça

Porto, October 10, 2023

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

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

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

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

ISEP, Porto, October 10, 2023

Dedicatory

I dedicate this work to all the patients affected by this devastating disease and to the healthcare professionals who tirelessly strive to develop new techniques to save lives.

Abstract

In today's ever-evolving technological landscape, the volume of data across sectors is growing, particularly in healthcare. Here, the gathering and processing of biochemical data aim to refine decision-making for patient treatments, especially using tools based on Machine Learning (ML). As a subset of Artificial Intelligence, ML harnesses algorithms to predict outcomes or unearth patterns that might otherwise remain concealed.

The interpretability of ML models is pivotal, enabling healthcare professionals to place confidence in and decipher the model's predictions. This assumes particular significance when decisions could directly affect patient lives.

This research embarked on an in-depth exploration of various ML algorithms and techniques to discern whether the combined metabolic profiles of amino acids and acylcarnitines might serve as new biochemical indicators for predicting colo-rectal cancer prognosis.

Throughout this study, several algorithms and data preprocessing techniques were evaluated. Four distinct experiments validated the predictions of the models in different scenarios. These scenarios involved predicting Colorectal Cancer using amino acids with and without the age parameter, and similarly, using acylcarnitine with and without the age parameter. Each scenario's predictions were elucidated using SHAP, both for overarching feature significance and individual instances.

Preliminary analyses indicated that the constructed models demonstrated promising predictive power, with notable variations for the different scenarios. Amongst the algorithms tested, Random Forest, Support Vector Machine, Gaussian Naive Bayes, and Gradient Boosting emerged as the top performers.

Keywords: Colorectal Cancer, Machine Learning, Amino Acids, Acylcarnitines, ExplainableAI

Resumo

No atual panorama tecnológico em constante evolução, o volume de dados em diversos setores está a aumentar, particularmente na saúde. Aqui, a recolha e processamento de dados bioquímicos visam aprimorar a tomada de decisão para tratamentos de pacientes, especialmente utilizando ferramentas baseadas em Aprendizagem Automática. Como um subconjunto da Inteligência Artificial, a Aprendizagem Automática utiliza algoritmos para prever resultados ou descobrir padrões que de outra forma poderiam permanecer ocultos.

A interpretabilidade dos modelos de Aprendizagem Automática é fundamental, permitindo que os profissionais de saúde confiem e decifrem as previsões do modelo. Isto assume uma importância particular quando as decisões podem afetar diretamente a vida dos pacientes.

Esta investigação levou a cabo uma exploração aprofundada de vários algoritmos e técnicas de Aprendizagem Automática para determinar se os perfis metabólicos combinados de aminoácidos e acilcarnitinas poderiam servir como novos indicadores bioquímicos para a previsão e prognóstico do cancro colo-retal.

Ao longo deste estudo, vários algoritmos e técnicas de pré-processamento de dados foram avaliados. Quatro experiências distintas validaram as previsões dos modelos em diferentes cenários. Estes cenários envolveram a previsão de Cancro Colorretal usando aminoácidos com e sem o atributo idade, e de forma semelhante, usando acilcarnitinas. As previsões de cada cenário foram elucidadas usando o SHAP, tanto para a importância geral dos atributos como para amostras individuais.

Análises preliminares indicaram que os modelos construídos mostraram um poder preditivo promissor, com variações notáveis nos diferentes cenários. Entre os algoritmos testados, *Random Forest*, *Support Vector Machines*, *Naive Bayes* e Gradient Boosting destacaram-se com melhor desempenho.

Acknowledgement

At the end of this long journey, I would like to thank everyone who accompanied me throughout the process.

To my advisor and co-advisor, Professor José Reis Tavares and Professor Isabel Praça, my deep gratitude for all the support provided, availability, and knowledge shared over this year.

To Professor Lúcia Lacerda and Professor Marisa Santos for providing information, clarifying doubts, and assisting in the development of this dissertation.

To my family and friends, a special thanks for all the love, affection, and encouragement to move forward.

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence.
CRC	ColoRectal Cancer.
CRISP-DM	Cross Industry Standard Process for Data Mining.
DSS	Decision Support System.
M0	The sample collection moment when the CRC is detected.
M1	The sample collection moment after the neoadjuvant Chemoradiotherapy treatment.
M2	The sample collection moment after the Surgery.
M3	The sample collection moment for vigilance purposes.
M4	The sample collection moment after the recurrence of disease.
ML	Machine Learning.
SHAP	Shapley Additive Explanations.
XAI	Explainable Artificial Intelligence.

Chapter 1

Introduction

The topic of the dissertation is introduced in this chapter along with its context, problem, objectives, information sources and the approach used in its realization. The document's structure is then presented.

1.1 Context

The digital era has been feeding worldwide databases with very large volumes of data. Extracting information and knowledge from this data has become one of the most predominant study fields. Multiple areas can take advantage of the emerging technologies and healthcare is no exception.

Cancer is a decease with very high death rate which is grouped accordingly with the organ or tissue. ColoRectal Cancer (CRC) is a particular aggressive one [1].

Metabolic profiles such amino acids and acylcarnitines, which can be discovered in blood, urine, and other bodily fluids, can give information about a person's health. Therefore, they are candidates to feed Machine Learning (ML) algorithms.

ML is a type of artificial intelligence that involves training algorithms to identify patterns in data and make predictions or decisions based on the data.

The healthcare field still has not widely adopted predictive tools. There is plenty of room to explore the use of metabolic profiles in CRC diagnosis and treatment using ML.

While ML algorithms have the potential to revolutionize early disease detection, such as in the case of CRC, the explanatory aspect ensures that these tools are safely and effectively integrated into clinical workflows.

1.2 Problem Definition

CRC is a type of colon or rectal cancer, which may also be called colon or rectal cancer (RC) depending on location [2]. CRC is the third most commonly diagnosed cancer worldwide and has been increasing its incidence over time. The dietary habits from western lifestyles may be contributing for this increase [3].

With the lack of symptoms at early stages, it is critical to explore the digital metabolic print, provided by the screening and thus the collected data, along side with the help of the available computational power. This can provide the so needed support in the prematurely diagnose of the decease.

In the context of machine learning (ML), especially in healthcare, the importance of explanations cannot be overstated. Explaining ML models ensures that healthcare professionals can trust and interpret the predictions made by these models. This is especially crucial when patient lives are at stake, as transparent and interpretable ML models can aid in clinical decision-making, reinforcing confidence in the digital tools, and lead to more informed medical interventions.

1.3 Objectives

This dissertation aims to carry out an investigation, using the combined metabolic profiles amino acids and acylcarnitines, with two main goals:

- Predict the diagnosis of CRC in a set of patients;
- Explain the predictions, so healthcare staff can understand why those predictions are being made

1.4 Information Sources

All data is provided by the Hospital and University Center of Porto (CHUPorto) by the following departments:

- Colorectal Surgery Unit of the General Surgery Service
- Genetic Biochemistry Unit of the Medical Genetics Center

The datasets are spread across multiple files containing the results of real patients and being distinct by the measured biomarkers (amino acids and acylcarnitines).

The datasets are divided in two categories:

1. Patients with several types of diseases (including no disease)
2. Patients already diagnosed with CRC

1.5 Approach

There are several project management frameworks used to help teams to execute their data science projects. According to the study made at [4], the Cross Industry Standard Process for Data Mining (CRISP-DM) is the most commonly used.

This framework consists on the following steps:

1. **Business Understanding** - Starting by understanding the customer's needs.
2. **Data Understanding** - Explore the RAW data.
3. **Data Preparation** - Select, clean, format the data into the datasets that will be used.
4. **Modeling** - Determine which algorithms to try for building the models.
5. **Evaluation** - Check if the models meet the business success criteria by the use of performance metrics.
6. **Deployment** - How the models can be deployed for being used by the customer.

Figure 1.1 shows a diagram to visualize the several CRISP-DM steps.

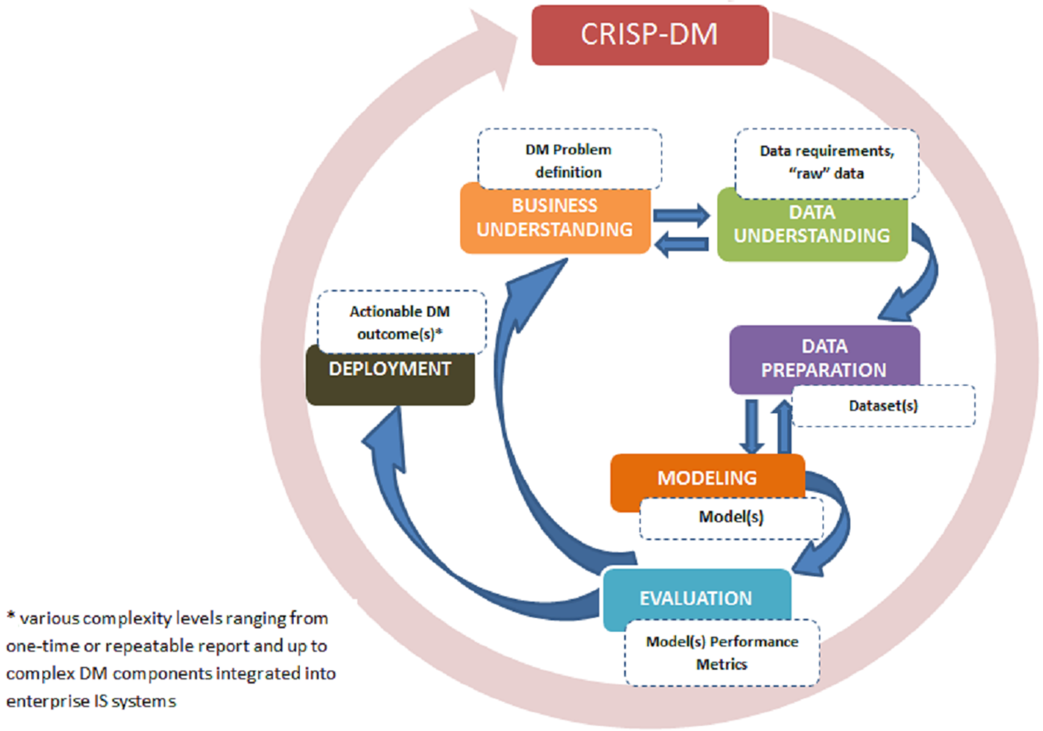
CRISP-DM
DM Problem definition
BUSINESS UNDERSTANDING
Data requirements, "raw" data
DATA UNDERSTANDING
Actionable DM outcome(s)*
DEPLOYMENT
DATA PREPARATION
Dataset(s)
MODELING
Model(s)
EVALUATION
Model(s) Performance Metrics
* various complexity levels ranging from one-time or repeatable report and up to complex DM components integrated into enterprise IS systems

Figure 1.1: CRISP-DM Diagram [5]

1.6 Document Structure

The document organized by the following items:

- **Introduction** - Describes the topic of this dissertation, including its context, the problem, the objectives, the information sources and the approach used in its realization
- **State of the Art** - Introduces the CRC folowed by amioacids and acylcarnitines. Then presents the ML, exploring algorithms, frameworks and explanations. Finally, literature examples of the application of ML in the prediction and prognosis of cancer are given.
- **Data Understanding & Preparation** - Gives an overview of the data used, as well as the pre-processing tools.
- **Modeling** - Provides an insight on how the bench for running the tests was structured.
- **Evaluation & Explanations** - This is where the collected results will be explored in the search for the best models for each scenario, followed by their respective explanations.
- **Deploy** - Describes the system architecture allowing to understand how the models and explanations are going to be deployed.
- **Conclusions** - Presents the conclusions about the work carried out, highlighting what objectives were achieved and what the next steps should be.

Chapter 2

State of the Art

This chapter exposes the fundamental concepts for a better understanding of the work developed: CRC, amino acids, acylcarnitines, ML and Explanations. Finally, examples of ML applications in cancer prediction are presented.

2.1 Colorectal Cancer

This section presents a generic definition of cancer and a more detailed one for the colorectal.

2.1.1 What is a cancer

An aberrant mass of cells that grow and divide uncontrollably is referred to as a tumor. They can be brought on by a variety of things, such as genetic mutations, environmental influences, and way of life decisions, and they can happen in any portion of the body.

There are two types of tumors.

- Benign - Usually grow slowly and do not invade nearby tissues or spread to other parts of the body. They can also be called as Polyps.
- Malign - Grow rapidly and invade nearby tissues. They can also spread to other parts of the body through the bloodstream or lymphatic system, a process known as metastasis.

Cancer is a malign tumor [6]. Its mortality is very related with its detection timing. The survivability increases the early the patient is diagnosed. Depending on the cancer, usually the symptoms start appearing in not so early stages. Moreover, the symptoms may be generic by being present in several other diseases, making it more difficult to relate to them to the cancer [7]. Therefore, it's important to screen each type of cancer. Screening is looking for cancer before a person has any symptoms [8].

2.1.2 What is a Colorectal Cancer

Definition

CRC is a type of cancer that starts in the colon or rectum. It usually begins as small clumps of cells called adenomatous polyps that slowly develop into cancer over time.

The study [9] presents the CRC stages in 2.1.

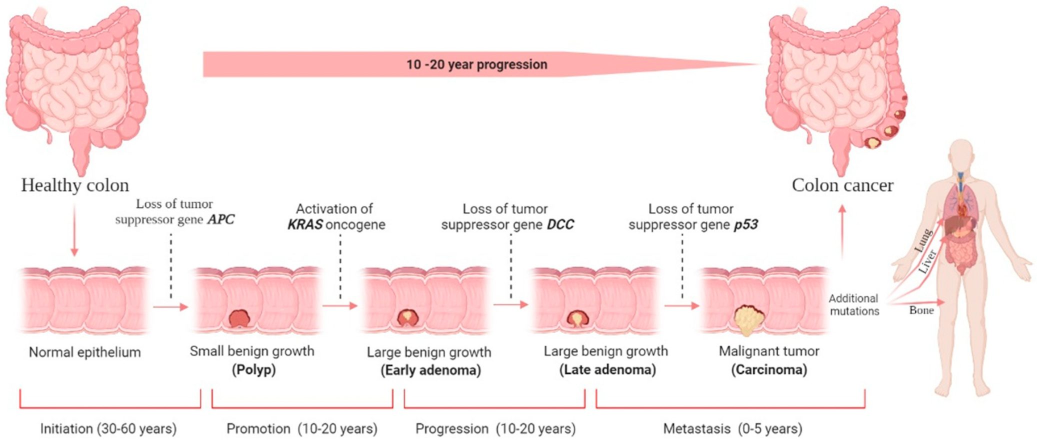
10 -20 year progression
Healthy colon
Colon cancer
Loss of tumor suppressor gene *APC*
Activation of *KRAS* oncogene
Loss of tumor suppressor gene *DCC*
Loss of tumor suppressor gene *p53*
Additional mutations
Lung
Liver
Bone
Normal epithelium
Small benign growth **(Polyp)**
Large benign growth **(Early adenoma)**
Large benign growth **(Late adenoma)**
Malignant tumor **(Carcinoma)**
Initiation (30-60 years)
Promotion (10-20 years)
Progression (10-20 years)
Metastasis (0-5 years)

Figure 2.1: Colorectal Cancer Stages - [9]

Incidence and mortality

It's the third and second most commonly diagnosed cancer in males and females respectively [10]. It comprised 10% (1.9 million) of global new cancer cases and 9.4% (0.9 million) of cancer deaths in 2020 [11]. The figure 2.2 confirms the increasing trend.

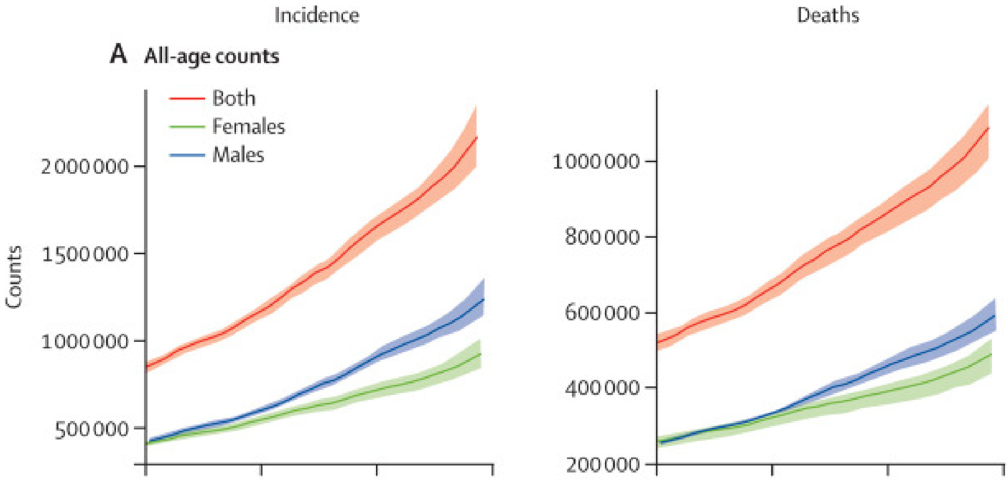

Figure 2.2: Colorectal Cancer Incidence and Mortality adapted from [12]

Risk Factors

There are several risks factors such as family clinical history, lifestyle, age and sex play. Obesity, sedentarism, smoking, alcohol consumption and inappropriate dietary are among them [13]. The 2.3 figure shows the main risk factors associated with CRC.

Symptoms

CRC symptoms may vary depending on the location of the cancer and the stage of the disease. The study [14] identifies the following common symptoms:

- Bleeding from Back Passage

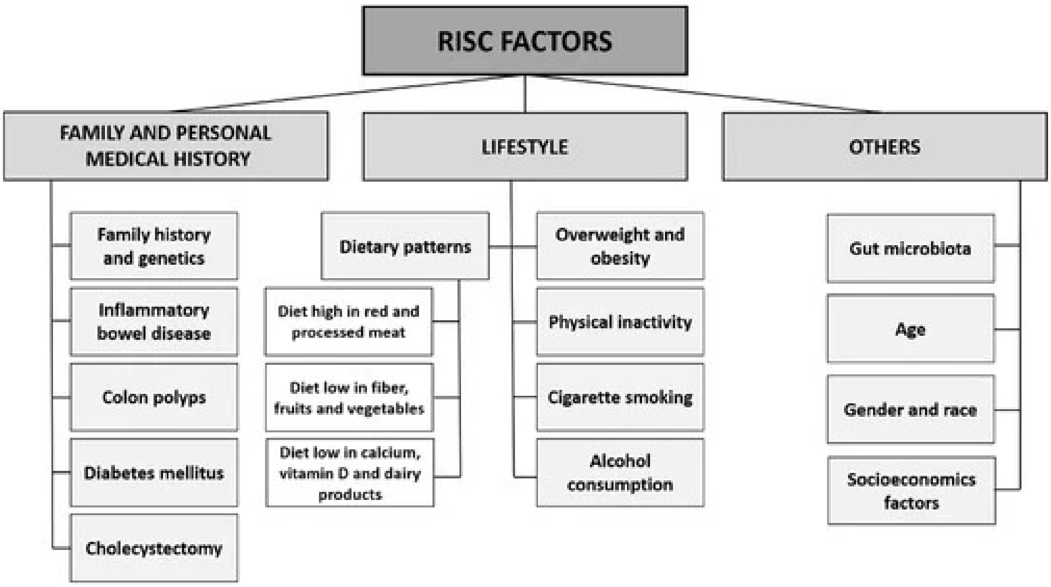

Figure 2.3: Classification vs Regression Algorithms [13]

- Change in Bowel Habit
- Back pain
- Indigestion/Heartburn/Tummy Ache
- Decrease in Appetite
- Weight loss
- Fatigue or Tiredness
- Feeling Different

Treatments

Treatment plans differ greatly depending on the stage. Surgery and chemotherapy have been the most common treatments for this disease [9]. There are different strategies on the order where these two common treatments are applied. The study [15] found that Neoadjuvant ChemoradioTherapy followed by surgery and postoperative chemotherapy was associated with a lower risk of local recurrence and improved disease-free survival compared to postoperative chemotherapy alone.

There other treatments such as Immunotherapy [16] and Targeted therapy [17].

Screening

Cancer screening is the use of medical tests to detect cancer in asymptomatic individuals. The goal of cancer screening is to identify cancer early, when it is more treatable and curable. [18]

In CRC screening is different from surveillance which refers to the interval use of colonoscopy in patients with previously detected with the disease [19]. The study [19] also recommends that CRC screening should start at age 50.

2.2 Metabolic Profiles

This section introduces the concept of metabolic profiles specializing on amino acids and acylcarnitines.

> "Metabolic profiling (metabolomics/metabonomics) is the measurement in biological systems of the complement of low-molecular-weight metabolites and their intermediates that reflects the dynamic response to genetic modification and physiological, pathophysiological, and/or developmental stimuli." - Christopher J Clarke and John N Haselden [20]
>
> Amino acids and acylcarnitines are examples of metabolites and can be used for clinical diagnosis [21].

2.2.1 Amino acids

Amino acids are the basic building blocks of proteins and serve as the nitrogen backbone of compounds such as neurotransmitters and hormones. The name comes from its organic compound that contains both an amino (-NH2) and carboxylic acid (-COOH) functional group. [22].

Although homoeostasis regulates the levels of amino acids in the blood, dietary, metabolic, behavioral, and genetic factors also have an impact. Cancers need a plentiful supply of amino acids to maintain their proliferation drive. They can play a direct function in promoting the synthesis of nucleosides and maintaining cellular redox homoeostasis in addition to their direct involvement as substrates for protein synthesis. Cancer cells occasionally coexist with other metabolic community members in complicated and frequently nutrient-poor microenvironments, developing interactions that can be both symbiotic and parasitic. [23]

2.2.2 Acylcarnitines

Fatty acid metabolites called acylcarnitines play a role in the synthesis of energy needed to maintain cell function. Nowadays, they are employed in the research of numerous illnesses, including diabetes, cardiovascular disease, depression, and a few malignancies, as well as metabolic and neurological disorders. Traditionally, they have been used as diagnostic indicators of fatty acid oxidation mistakes. They are also being researched as indicators of deficiencies in energy metabolism, insulin resistance, peroxisomal and mitochondrial b-oxidation activity, and physical activity. [24]

2.2.3 Relatioship between Amino Acids and Acylcarnitines with CRC

Amino acids and acylcarnitines are not directly related, but they are both important molecules involved in metabolic processes in the body. They can provide information about a person's health and can be found in blood, urine, and other body fluids. These biomarkers can be used to diagnose and monitor various diseases, including CRC [25].

While associations between amino acids and acylcarnitines profiles with CRC have been identified, causative relationships are more challenging to establish.

The altered metabolite profiles might be a consequence of the cancer, or they might play roles in cancer initiation and progression. Furthermore, factors such as diet, microbiota composition and medication use can influence these metabolite levels. The study [26] discusses how amino acid metabolism is dysregulated in colorectal cancer patients and the study [27] desribes the promising role of carnitine and acylcarnitine levels in understanding and potentially diagnosing and treating. Both point out the need of further investigations.

2.3 Machine Learning (ML)

In this section, ML will be introduced followed the classification algorithms and libraries.

2.3.1 Supervised Algorithms

In the context of this work, only the ML supervised algorithms are going to be used, therefore, only these ones will be explored.

Supervised learning uses the approach of a mapping between a set of input variables and an output variable. By applying this mapping, it can predict the outputs for unseen data [28].

There are two types of algorithms [29]:

- Classification - Assumes discrete values from a non ordered dataset. Fraud detection and Medical diagnosis are examples of applications.
- Regression - Assumes infinite values from an ordered dataset. Sales/Weather forecasting are examples of applications.

The figure 2.4 shows the difference between these two types of algorithms.

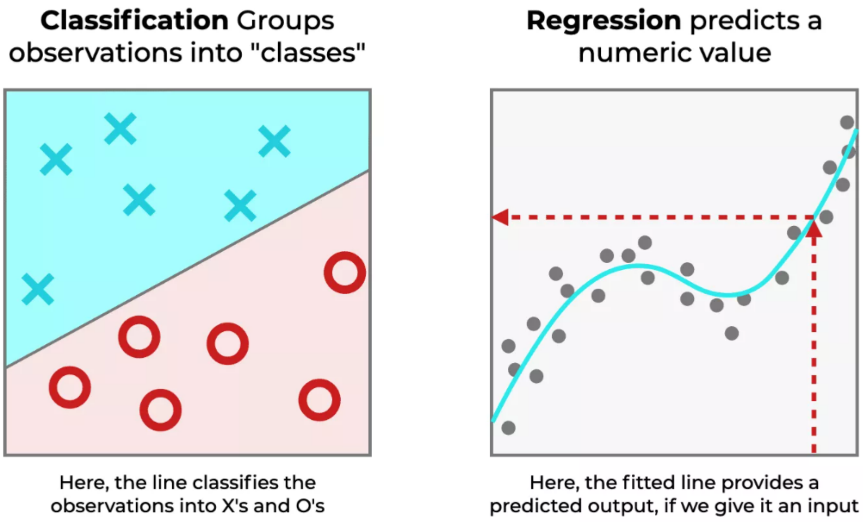
Classification Groups observations into "classes"
Here, the line classifies the observations into X's and O's
Regression predicts a numeric value
Here, the fitted line provides a predicted output, if we give it an input

Figure 2.4: Classification vs Regression Algorithms [30]

The figure 2.5 shows in, a simple way, the typical flow of the implementation and use of ML Classification Supervised Algortihms.

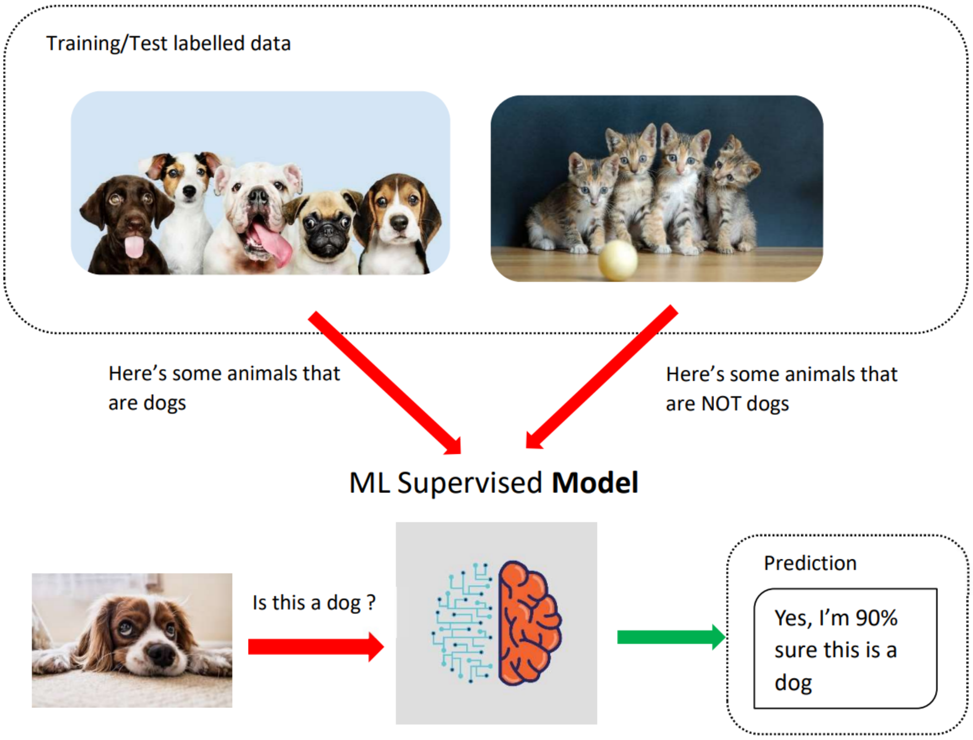
Training/Test labelled data
Here's some animals that are dogs
Here's some animals that are NOT dogs
ML Supervised **Model**
Is this a dog ?
Prediction
Yes, I'm 90% sure this is a dog

Figure 2.5: Typical flow of the implementation and use of ML Supervised systems with Classification

Bellow some common supervised learning algorithms from the book [29]:

Distance Based Algorithms

- **k-NN** k-Nearest Neighbors was first introduced by Evelyn Fix and Joseph Hodges in their paper titled "Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties" published in 1951. It is a simple and effective classification algorithm that is based on the idea that similar inputs are likely to have similar outputs. It classifies an input based on the class of the k-nearest neighbors in the training data. A new case is classified considering the distance between the point to be classified and the K closest points of the different classes, which were previously defined during the learning process. The classification result is given by the class that has the highest number of points represented in the defined distance [31].

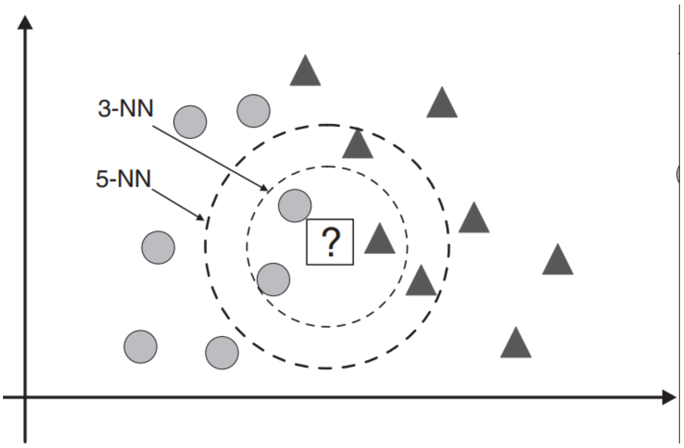

Figure 2.6: k-NN adapted from [29]

Advantages:

- Simplicity - simple and easy-to-understand algorithm.
- Versatility - can be used for both classification and regression problems.

Disadvantages:

- Computationally expensive
- Sensitivity to noise: k-NN is sensitive to noise and outliers in the data, which can affect its classification accuracy.
- Choice of k: The choice of k can affect the performance of the algorithm, and selecting the optimal value of k can be challenging.

Probabilistic Algorithms

- **Naive Bayes** - A classification algorithm based on Bayes' theorem. Introduced by Reverend Thomas Bayes in the 18th century and later extended by Laplace. It assumes that input variables are independent and calculates the probability an input belongs to a class, given the variables [32].

  Bayes' Theorem: The formula is defined as:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

  Where:

  - $P(A|B)$ represents the posterior probability of A given B.
  - $P(B|A)$ signifies the likelihood of B occurring given A.
  - $P(A)$ is the prior probability or initial belief about A.
  - $P(B)$ is the marginal likelihood or evidence of B.

  To illustrate the application of Bayes' Theorem, consider an example adapted from [29]:

Given:

- $P(\text{HeartRisk}) = 0.20$
- $P(\text{High BP} \mid \text{HeartRisk}) = 0.90$
- $P(\text{Elderly} \mid \text{HeartRisk}) = 0.80$
- $P(\text{High BP} \mid \text{No HeartRisk}) = 0.30$
- $P(\text{Elderly} \mid \text{No HeartRisk}) = 0.20$

Compute:

- Joint probability with HeartRisk: $0.90 \times 0.80 = 0.72$
- Joint probability without HeartRisk: $0.30 \times 0.20 = 0.06$

The marginal probability of both High BP and being Elderly is:

$$P(\text{High BP, Elderly}) = 0.72 \times 0.20 + 0.06 \times 0.80 = 0.192$$

Using Bayes' theorem:

$$\begin{aligned} P(\text{HR} \mid \text{HBP, Elderly}) &= \frac{P(\text{HBP, Elderly} \mid \text{HR}) \cdot P(\text{HR})}{P(\text{HBP, Elderly})} \\ &= \frac{0.72 \cdot 0.20}{0.192} \\ &= 0.75 \end{aligned}$$

There are different types of Naive Bayes classifiers, like Gaussian (used for continuous data) and Multinomial (often used for text data).

Advantages:

- Robust to irrelevant features and noise in the data [33].
- Computationally efficient.

Disadvantages:

- The strong independence assumptions might not always hold in real-world scenarios, potentially leading to suboptimal performance [34].

Search Based Algorithms

- **Decision Trees** - Peter E. Hart, Richard O. Duda, and David G. Stork developed the concept of the decision tree as a machine learning algorithm in the 1970. This is a popular algorithm for classification and regression tasks. It uses a tree-like structure to represent a set of decisions and their possible consequences, based on the input data [35]. The construction of a decision tree involves a recursive process that optimally selects features and splits the dataset at each node to create a tree-like structure. The root node serves as the initial starting point, representing the entire dataset. Internal nodes, also known as decision nodes, are non-leaf nodes within the tree. Each leaf node corresponds to a specific predicted class or value [36].

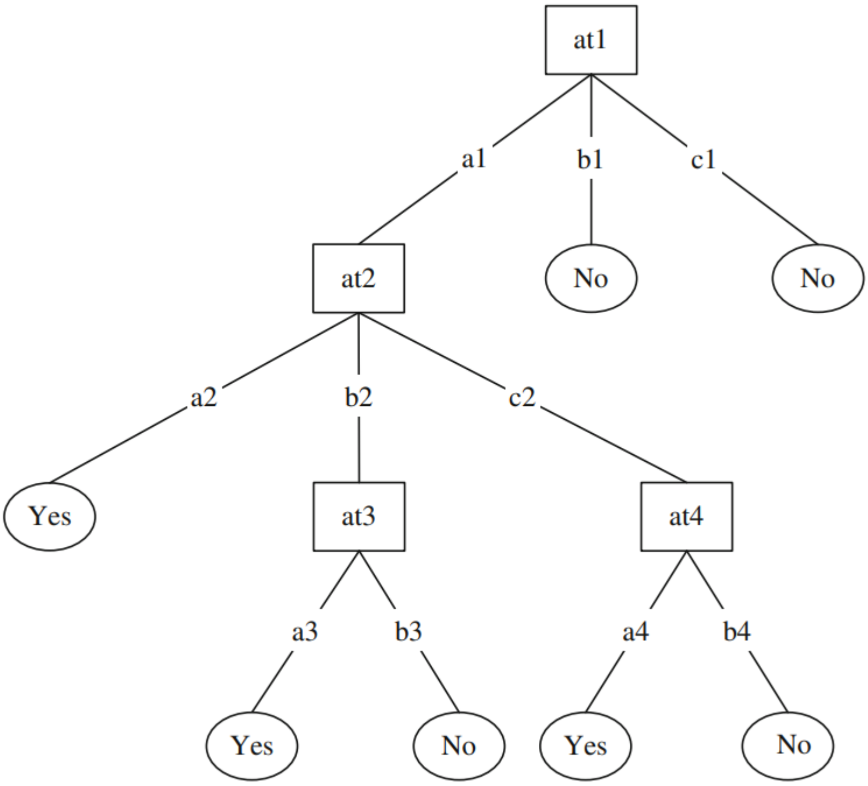

Figure 2.7: DecisionTrees adapted from [36]

Advantages:

- Easy to visualize and understand, even for non-experts, providing a clear graphical representation of the decision-making process, making it easy to see how decisions are being made [36].
- Can handle both categorical and numerical data [37].

Disadvantages:

- Instability: Small changes in the training data can lead to significantly different decision trees. This instability can make decision trees less reliable than other machine learning algorithms. [35].
- Bias: Decision trees can be biased towards features with many levels, as these features can have a larger impact on the split decisions [38].

Ensemble Methods

- **Random Forests** - Created in 1995 by Tin Kam Ho using the random subspace method. Combines multiple decision trees to improve the accuracy of the predictions. Each decision tree is trained on a random subset of the input data, and the final prediction is based on the average or majority vote of the individual trees [39].

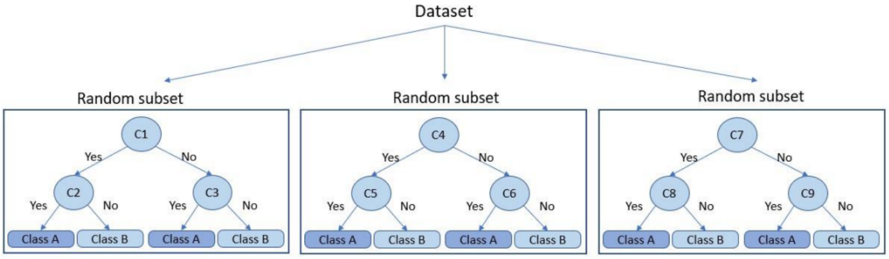

Figure 2.8: RandomForests adapted from [39]

Advantages:

- Robust to overfitting [39].
- Can handle missing data and outliers in the input features [40].

Disadvantages:

- Computationally expensive and slow to train on large datasets, particularly when the number of features or trees is high [41].
- Difficult to interpret and visualize, particularly when the number of trees in the forest is large [42].

- **Gradient Boosting** - Gradient Boosting is a family of algorithms such as XGBoost, LightGBM, and CatBoost. It was introduced in the context of machine learning by Jerome H. Friedman in a series of papers between 1999 and 2002. It builds an ensemble of decision trees in a sequential manner, where each tree corrects the errors of its predecessor. In each iteration, it fits a new tree to the negative gradient of the loss function, essentially pointing in the direction that minimizes the error. By summing the predictions of each tree, the final model produces a combined output that, ideally, offers improved accuracy [43].

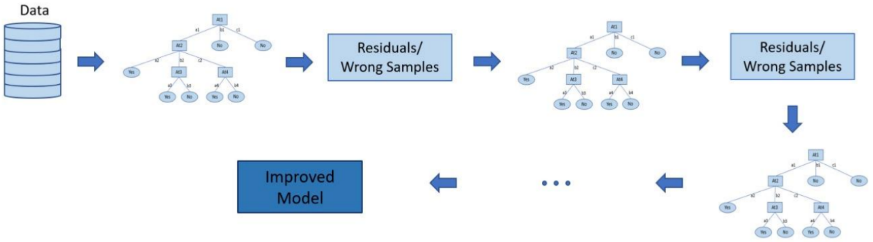

Figure 2.9: GradientBoosting adapted from [44]

Optimization Based Algorithms

- **SVMs** Support Vector Machines - were developed by a team of computer scientists, including Vladimir Vapnik and Corinna Cortes, while working at Bell Labs in the 1990s. It separates data points using hyperplanes. They work by finding the hyperplane that maximizes the margin between the two classes of data points. [45].

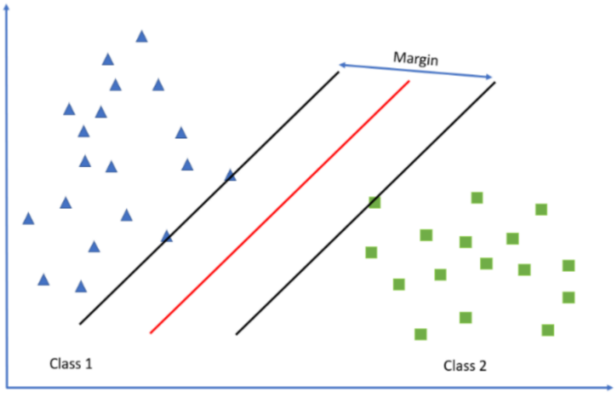

Figure 2.10: Support Vector Machines adapted from [46]

Advantages:

- Perform well in high-dimensional spaces, such as those commonly found in image recognition, text classification, and bioinformatics [45].
- High accuracy and generalization performance in many applications[45].

Disadvantages:

- Computationally expensive, especially for large datasets, which can make training and optimization time-consuming [47].
- Sensitive to noisy and outlier data, which can lead to overfitting or poor generalization performance [47].
- Difficult to interpret and understand, as the decision boundaries are defined in a high-dimensional space [48].

- **ANNs** Artificial Neural Networks - Were first proposed in the 1940s by Warren McCulloch and Walter Pitts in their seminal paper "A Logical Calculus of the Ideas Immanent in Nervous Activity. They are a class of machine learning models that are inspired by the structure and function of the human brain. They consist of multiple layers of interconnected nodes. [49]. Each layer is responsible for a different task, and in addition to the input and output layers, there are hidden layers where several data transformations are carried out. This way, the input data is supplied to the input layer, processed, transformed in the hidden layers and, later, sent to the output layer, where the classification result will be produced [50].

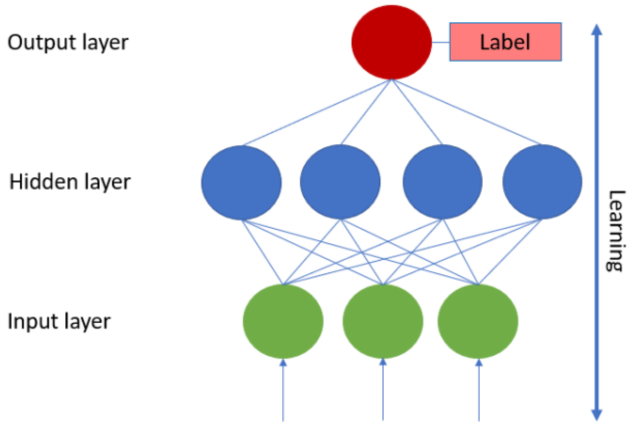

Figure 2.11: Artificial Neural Networks adapted from [51]

Advantages:

- Capable of handling large amounts of data and can be used with a wide range of input types [52].
- Flexible and can be used for both classification and regression tasks [49].

Disadvantages:

- Can be computationally intensive and requires large amounts of data. [53].
- Difficult to interpret, making it hard to understand how they arrive at their predictions [54].

2.3.2 Machine Learning Libraries

There are several ML libraries. While TensorFlow, Keras, PyTorch, and ML.Net are powerful frameworks, they are primarily tailored for deep learning tasks. Deep learning models often need large volumes of data to perform optimally. For projects not based on vast datasets, such as this one, diving deep into these frameworks might not be the most efficient approach. In contrast, scikit-learn is designed for traditional machine learning algorithms, offering a simplified and consistent API that integrates seamlessly with the Python ecosystem. Given the nature and data constraints of this project, scikit-learn emerges as the best and most appropriate choice. In the subsection bellow, some key attributes and features of scikit-learn will be explored.

Scikit-learn

This library for Python, is one of the most popular for both beginners and advanced users. It has the following key attributes and features [55]:

- **Easy to Use:** Scikit-learn provides a consistent interface for different kinds of machine learning algorithms, making it easy to switch between models with minimal code changes.

- **Tools and Algorithms:** The library offers a wide variety of supervised and unsupervised learning algorithms:
  - **Classification:** e.g., SVM, KNN, Random Forests, etc.
  - **Regression:** e.g., linear regression, ridge regression, Lasso, etc.
  - **Clustering:** e.g., k-means, spectral clustering, mean-shift, etc.
- **End-to-End Workflow: Pipelines**
  - Ensure consistency by streamlining the sequence of transformations applied to data.
  - Prevent data leakage by encapsulating the entire workflow.
  - Simplify the machine learning workflow, making code more readable.
  - Enhance reusability by allowing the same steps and processes to be applied consistently across different contexts.
- **Model Selection:** Tools to help in choosing between models, such as cross-validation, grid search, and metrics.
- **Preprocessing:** Tools for normalization, feature selection, etc.
- **Deployment:** Models can be easily serialized (using libraries like joblib) and deployed in various environments.
- **Maturity and Stability:** It's a mature library with a long track record, which means it's stable and has been tested across various scenarios.
- **Integration with Python Ecosystem:** Built on top of NumPy and SciPy, it allows for seamless integration and interoperability with other scientific libraries in the Python ecosystem.
- **Documentation and Community:** Comprehensive online documentation with both tutorials and detailed method information, supported by a large and active community.
- **Extensibility:** The library's architecture is designed to be extensible, allowing users to add their own algorithms and tools.

2.4 Explainable artificial intelligence

In the past decade, Artificial Intelligence (AI) has seen remarkable and continuous advancements, resulting in a wider use of its algorithms, including ML algorithms, to address a variety of problems. This significant progress has its downsides: it results in the creation of more complex models and the use of 'black-box' AI models that are not transparent. Consequently, it is becoming increasingly important to develop solutions to tackle this challenge, as this would enable the expanded use of AI systems in critical and sensitive sectors, such as healthcare and security [56].

The study [57] states the following:

> *In many applications, an explanation of how an answer was obtained is crucial for ensuring trust and transparency. An example of one such application is a medical application.*

Explainable Artificial Intelligence (XAI) is a field of research dedicated to making the results of AI systems more comprehensible to humans. The term was originally introduced in 2004 [58], which used it to describe their system's capacity to elucidate the behavior of AI-controlled entities in simulation game applications. Technically, there is no universally accepted definition of explainable AI. In fact, the term XAI is more often associated with the movement, initiatives, and efforts undertaken to address concerns about AI transparency and trust, rather than a formal technical concept [59]. The following terms are also used, by the research communities, to address the issue of explainability:

- Understandable AI
- Comprehensible AI
- Accurate AI/ML
- Transparent AI
- Interpretable ML
- Responsable ML
- Interactive ML

In the mid-1990s, an artificial neural network (ANN) was developed to predict which pneumonia patients required hospital admission and which could be treated as outpatients.Initial results suggested that neural networks were significantly more accurate than traditional statistical methods. However, a comprehensive test revealed that the neural network incorrectly deduced that pneumonia patients with asthma had a lower risk of death and, therefore, did not require hospital admission. This conclusion was medically counterintuitive but reflected a pattern in the training data. Asthma patients with pneumonia were often admitted directly to the ICU (Intensive Care Unit), received aggressive treatment, and survived. Consequently, it was decided to abandon the AI system as it was deemed too dangerous for clinical use. This incident highlighted the importance of interpreting the model to identify and rectify such critical issues [59].

There are several tools and libraries available for model explanation and interpretation. The following sub section will explore one of them.

2.4.1 SHAP

Shapley Additive Explanations (SHAP) method was introduced in the paper "A Unified Approach to Interpreting Model Predictions" by Scott M. Lundberg and Su-In Lee (2017) [60]. This method is based on Shapley values.

Shapley values originate from cooperative game theory and were introduced by Lloyd S. Shapley in 1953 in the paper: Shapley, L. S. (1953). A value for n-person games. In H. W. Kuhn & A. W. Tucker (Eds.), Contributions to the Theory of Games (Vol. II, pp. 307–317). Princeton University Press[61].

It is a concept from cooperative game theory that provides a way to fairly allocate the rewards or payoffs among the players, based on their marginal contributions to the coalition. The contribution of a player is the additional value that the player brings to a coalition. For example, if a group of players has a certain value without a specific player, and a higher value with that player, the contribution of that player is the difference between the two values. The Shapley value is computed by considering all possible orders in which the players can join the coalition, and averaging the marginal contributions of each player across all these orders. [62]

In ML, SHAP is a method used to explain the contributions of each feature (input variable) to the predictions made by a model.The SHAP value of a feature is comparable to a player's Shapley value in a cooperative game. It represents the average contribution of that feature to the model's prediction across all possible feature combinations. By calculating the SHAP values for all features, it is possible to comprehend how each feature influences the model's prediction for a particular instance [60].

The contribution of individual features can be measured by the expected value which represents the average prediction that the model would make over the entire dataset, acting as a baseline or reference point. The value is calculated by averaging its predictions over the training set or any representative dataset. It is essentially the model's "base rate" prediction without considering any specific features of an instance [60].

The sign (positive/negative) of the SHAP value point the direction of the relationship. In a binary classification problem, such as CRC detection, a positive value means that feature increases the model's likelihood of predicting a patient with CRC (assuming the value 1 for), while a negative value means it increases the likelihood of not having CRC (assuming value 0).

The magnitude of the SHAP value (ignoring the sign) describes the importance of that feature. Larger absolute values mean that a feature plays a more crucial role in the model's decision. regarding the presence or absence of CRC.

SHAP can be useful to explain the overall feature importances for a given dataset or a single instance. The next subsections will explore how each can be achieved.

Overall Feature Importances

If we gather the shap values for every feature of every dataset instance, we're able to see which features impacted the model's prediction and in which way. It gives a sense of which features, on average, have the most influence on the model's predictions

SHAP Library has several plots [63]. SummaryPlot is one that provides a clear visualization of the features importances for multiple predictions, and therefore can be applied to an entire dataset. An example of this plot can be seen at 2.12. In this example, the Breast Cancer Wisconsin dataset from scikit-learn was utilized. It represents a binary classification problem where the objective is to distinguish between malignant and benign tumors based on various cell nucleus. Only the four top most influencing features are shown.

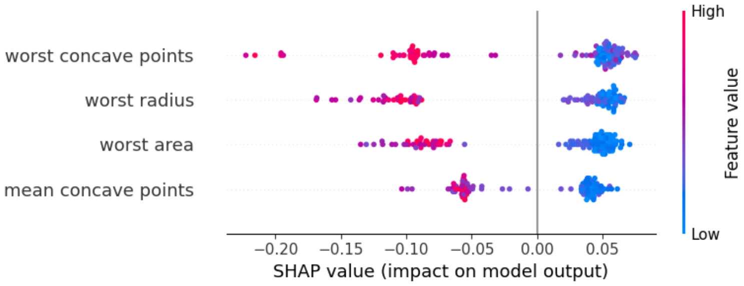

Figure 2.12: SHAP SummaryPlot example

Each feature is listed on the y-axis, often sorted by the average absolute SHAP value. This gives a sense of which features are, on average, most influential in making predictions. The x-axis shows the SHAP value. A feature's SHAP value on the left (negative side) suggests it pushes the model's prediction lower than the base value (expected value), while a SHAP value on the right (positive side) indicates it pushes the prediction higher. The color represents the value of the feature for each data point. A gradient is typically used, where one color (e.g., red) represents higher values of a feature, and another color (e.g., blue) represents lower values of the feature. In this example, the feature "worst concave points" is on top, meaning that it's the most influence feature. The colors all four features have higher values (color red) when the shap values are negative pushing the prediction to 0 (the tumor is benign).

Single Instance

For a single instance explanation, several shap plots can be used. One of them is the the ForcePlot. Positive SHAP values are shown in red and the negative as blue. The range is typically from 0 to 1 in binary classification problems, however it can change depending on the classifier. For example the RandomForests classifier presents the range between 0 and 1 while the GradientBoosting can present a range with negative values. The importance interpretation doesn't change.

The force plots 2.13 and 2.14 represent two examples of the same Wisconsin dataset where the first is explaining a predicting of tumour as being malign and the second as benign.

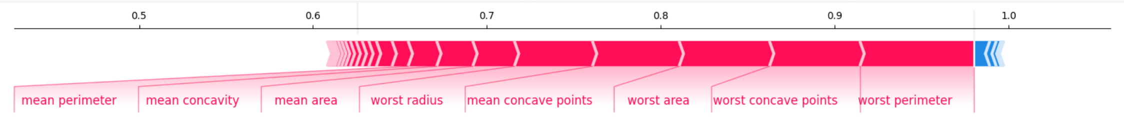

Figure 2.13: SHAP ForcePlot example with positive class

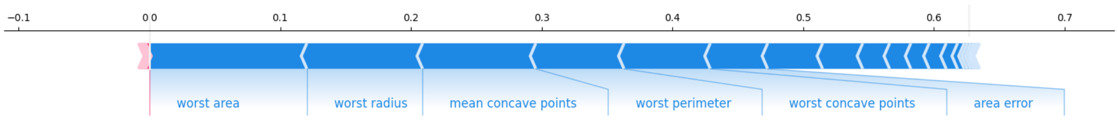

Figure 2.14: SHAP ForcePlot example with negative class

In the example 2.13, features like "worst perimeter", "mean concave points" and" worst radius" with their positive SHAP values and red coloration significantly drive the prediction towards 1. Blue features, with their negative SHAP values, act in opposition, trying to reduce the prediction, they're not appearing in the plot due to their low values. The collective strength of the positive contributors outweighs the negative ones, resulting in a prediction of 1 (malign tumout).

2.5 Machine Learning in Cancer Diagnosis

Over recent decades, ML methods have become prevalent in predicting cancer. The emergence of new medical technologies has provided the healthcare community with vast amounts of cancer-related data. Using this data, researchers have employed various ML techniques to identify patterns and correlations, enhancing their ability to forecast outcomes for specific cancer types [64]. The table 2.1 presents relevant publications that used ML methods for cancer susceptibility prediction.

Publication	ML method	Cancer type	Number of patients	Accuracy
Ayer T et al. [65]	ANN	Breast cancer	62 219	96.5%
Waddell M et al. [66]	SVM	Multiple myeloma	80	71%
Listgarten J et al. [67]	SVM	Breast cancer	174	69%
Stojadinovic A et al. [68]	BN	Colon carcinomatosis	53	71%

Table 2.1: Relevant publications that used ML methods for cancer susceptibility prediction (adapted from [64])

2.5.1 Colorectal Cancer Diagnosis

When it comes to CRC diagnose, the studies [23] and [24] explore the use of amino acids and acylcarnitines, repectively, for CRC diagnosis using the same data mentioned at section 1.4. They both had positive results with the [23] having an accuracy of 95% using 175 samples with RandomForests and the [24] with an accuracy of 90% using 182 samples with also RandomForests.

The study by H. Hussan [69] enrolled 3,116 adults aged 35-50 at average-risk for CRC where machine learning models were constructed to predict CRC and high-risk polyps, using data from electronic health records (EHR) such as demographics, obesity, laboratory values (hemoglobin and cholesterol panels), medications, and zip code-derived factors. Random forest, neural network, and gradient boosting decision tree, were compared against a reference logistic regression model. The results indicated that all machine learning models, except for gradient boosting, showcased superior discriminative ability in predicting CRC or high-risk polyps than the reference model. Particularly, factors like income per zip code, colonoscopy indication, and body mass index quartiles were significant predictors in the regularized discriminant analysis. The findings suggest that machine learning, using EHR data, can enhance CRC risk prediction for adults in the specified age group, but further refinement and validation in primary care settings are essential before clinical deployment.

The study by S Bosch [70] aimed to identify potential biomarker panels for CRC and adenoma detection and understand the interplay between gut microbiota and human metabolism when these lesions are present. Conducted between February 2016 and November 2019, it involved participants grouped into CRC, adenomas, and controls, all matched by age, gender, body-mass index, and smoking status. Fecal samples from 1093 participant were used and analyzed for their proteome, microbiota, and amino acid composition. Only 14 out of the 1093 were diagnosed with CRC. For the comparison between CRC samples and controls, three amino acids: sulfo-l-cystine, proline, and ethanolamine were selected from the machine learning pipeline. The combination of these amino acids resulted in an Area Under the Curve (AUC) of 0.6, suggesting moderate predictive power. It concluded that the integration of biomarkers, derived from fecal microbiota, proteome, and amino acids, offer a hopeful pathway for non-invasive screening of CRC and adenomas, which could potentially decrease the rate and fatalities of CRC.

The article by Zugang Yin [71] explores the advancements of AI in addressing CRC diagnosis and treatment. It emphasizes the value of machine learning and bioinformatics in pinpointing CRC biomarkers, which aids in non-invasive screenings. Additionally, Convolutional Neural Networks are noted for their capability in analyzing histopathologic tissue images, mitigating variability in doctors' interpretations. The article also touches on the rise of robotic surgical systems, such as da Vinci, due to their precision in CRC treatments. The integration of AI in neoadjuvant chemoradiotherapy has also elevated CRC treatment outcomes. It also concluded that deep learning in gene sequencing research offers a new treatment option.

Chapter 3

Data Understanding & Preparation

All data came from medical equipment in the form of several files from the sources indicated at 1.4 These files were pre-cleaned by the previous studies and also by the medical personel, meaning that a treatment to deal with missing, duplicated or inaccurate values, was already applied. This resulted in two CSV and one XLSX files with the follow characteristics:

- Non CRC Amino acids - Contains amino acids samples of patients without CRC (csv) totalizing 118 rows, from [23]
- Non CRC Acylcarnitines - Contains acylcarnitines samples of patients without CRC (csv) totalizing 118 rows, from [24]
- Combined Amino acids and Acylcarnitines - Contains multiple metabolic profiles (including amino acids and acylcarnitines) samples of patients with CRC (xlsx) totalizing 214 rows of multiple stages of the disease, prepared by the healthcare professionals

The two CSV files have a simple structure with features and their corresponding values. However, the XLSX file follows a more complex structure. This structure includes more data that enhances human readability but also makes it more challenging for machines to interpret. The multiple stages found at the Combined Aminoacids and Acylcarnitines (xlsx) file are the following:

- The sample collection moment when the CRC is detected (M0) - 99 samples
- The sample collection moment after the neoadjuvant Chemoradiotherapy treatment (M1) - 48 samples
- The sample collection moment after the Surgery (M2) - 61 samples
- The sample collection moment for vigilance purposes (M3) - 5 samples
- The sample collection moment after the recurrence of disease (M4) - 1 sample
- Blanks (no moment specified) - 2 samples

Since the objective of this work is to help diagnosing patients with CRC, only the M0 samples will be used for training the models. Upon agreement with healthcare professionals, it was decided to use the M1 samples to check the predictions made by those models.

In order to be able to start the data analysis and the model construction, the three different datasets were transformed into two. These two new datasets comprise the samples of patients with and without CRC for each type of metabolic profiles. The merge of the datasets was done by the common columns. The figure 3.1 illustrates this.

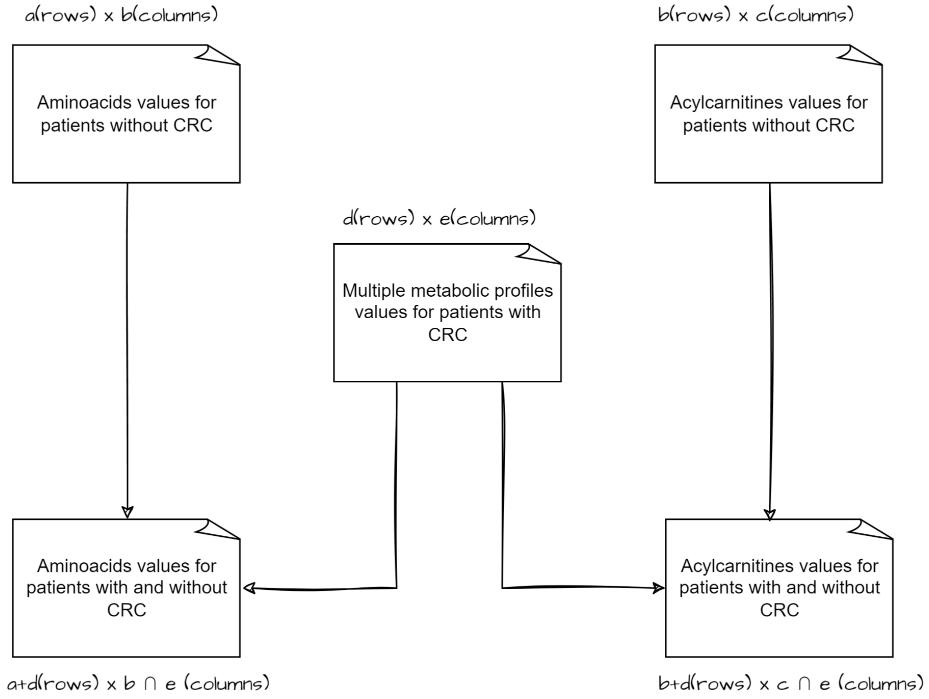
a(rows) x b(columns)
Aminoacids values for patients without CRC
b(rows) x c(columns)
Acylcarnitines values for patients without CRC
d(rows) x e(columns)
Multiple metabolic profiles values for patients with CRC
Aminoacids values for patients with and without CRC
a+d(rows) x b ∩ e (columns)
Acylcarnitines values for patients with and without CRC
b+d(rows) x c ∩ e (columns)

Figure 3.1: Datasets Construction

This process was automated by a tool that receives the three input files and automatically merges the common columns and outputs the two standardized files adding a new column called "CRCDetected" which is the target. Several features from each metabolic profiles were drop since they were not present in the files of patients without CRC. The common columns along with the dropped ones, are described in different sections that are dedicated for the two different metabolic profiles, Amionacids and Acylcarnitines. The dynamic feature of the tool, allows it to receive more columns if, in the meanwhile, data becomes available.

The next to sections will describe and explore the data. However, the column "Date of Birth" was previously transformed to represent the patients age. Also, the percentage of outliers for each feature was calculated using the z-score statistical measurement. More details about transformation, encoding and normalization at section 3.3

3.1 Amino acids

In this section the Amino acids dataset will be explored.

The table 3.1 lists the features for the Aminoacids dataset resulted from the construction described at the section 3.1 which has a total number of 217 samples. It presentes the feature description along side its abbreviation, the range of values (min and max), the median, average and percentage of outliers. The table 3.2 lists the features that were not present in the NonCRC dataset and therefore they won't be consider.

Table 3.1: Amino acids Dataset - Features

Feature Description	**Abbreviation**	**Range** (min - max)	**Median**	**Average**	**% of Outliers**	**Measurement Unit**
1-Methylhistidine	1mhis	(1 - 81)	12.0	16.11	1.38%	micromoles / litre
3-Methylhistidine	3mhis	(0 - 42)	5.0	5.49	1.38%	
Alpha-Aminoadipic acid	aaa	(0 - 38)	6.0	7.39	1.84%	
Aminobutyric acid	abu	(0 - 58)	15.0	17.91	2.30%	
Alanine	ala	(131 - 723)	361.0	367.69	0.92%	
Arginine	arg	(0 - 202)	46.0	48.44	0.92%	
Asparagine	asn	(34 - 179)	65.0	66.92	0.46%	
Aspartic Acid	asp	(0 - 52)	7.0	7.48	0.92%	
Citrulline	cit	(1 - 80)	28.0	30.07	1.38%	
Cystine	cys2	(10 - 147)	59.0	61.73	1.38%	
Cystathionine	cysta	(0 - 7)	1.0	1.31	0.92%	
Glutamine	gln	(293 - 1177)	532.0	536.07	1.38%	
Glutamic acid	glu	(13 - 564)	55.0	70.11	0.92%	
Glycine	gly	(117 - 474)	243.0	257.59	0.92%	
Histidine	his	(37 - 159)	81.0	81.60	0.92%	
Hydroxyproline	hyp	(0 - 46)	8.0	10.22	1.38%	
Isoleucine	ile	(28 - 145)	62.0	65.25	1.38%	
Leucine	leu	(62 - 289)	126.0	130.29	0.46%	
Lysine	lys	(100 - 535)	213.0	217.71	0.92%	
Methionine	met	(10 - 100)	24.0	26.05	1.38%	
Ornithine	orn	(43 - 238)	123.0	127.18	0%	
Phenylalanine	phe	(27 - 166)	55.0	57.00	1.38%	
Proline	pro	(61 - 469)	173.0	191.27	1.38%	
Serine	ser	(44 - 236)	117.0	122.83	0.46%	
Taurine	tau	(29 - 235)	97.0	100.75	0.92%	
Threonine	thr	(56 - 424)	141.0	145.49	0.92%	
Tyrosine	tyr	(8 - 156)	62.0	65.42	0.92%	
Valine	val	(78 - 377)	215.0	217.66	0.46%	
Age of the Patient	age	(18 - 90)	53.0	52.38	N/A	positive integer
The sex of the Patient	sex	N/A	N/A	N/A	N/A	Male or Female

Table 3.2: Amino acids Dataset - Features not present in the Non CRC dataset

Feature Description	Feature Abbreviation	Measurement Units
Serine ester phosphoric	pps	micromoles / litre
Phosphatidylethanolamine	pea	
Ureia	ureia	
Sarcosine	sar	
Beta-Alanine	bala	
Beta-aminoisobutyric acid	baib	
Homocystine	hcy2	
Gamma-Aminobutyric Acid	gaba	
Ethanolamine	etn	
Ammonia	nh3	
Hydroxylysine	hyl	
Anserine	ans	
Carnosine	car	
Sulfocysteine	sulfocys	

Age and sex are well-established demographic factors that often have clinical relevance. Many diseases vary in prevalence, severity, or manifestations depending on age and sex. By plotting these, we can quickly identify patterns or trends related to these factors. The next subsections will explore the number of patients with and without CRC, the age and sex features.

3.1.1 CRC cases

From the plots 3.2 and 3.3 we can see that there are slightly more cases without CRC as shown on the graphs bellow but in general, the dataset is balanced.

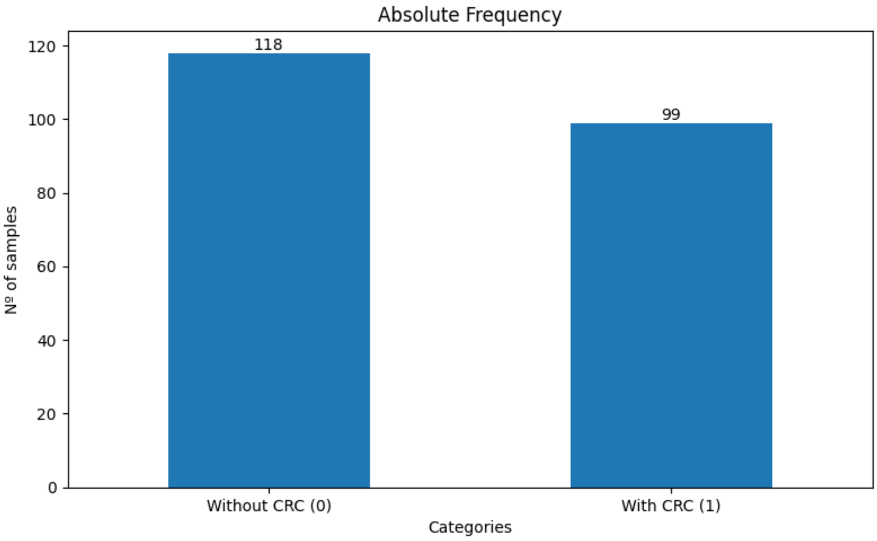

Figure 3.2: Amino acids - Absolute Frequency of the 2 classes

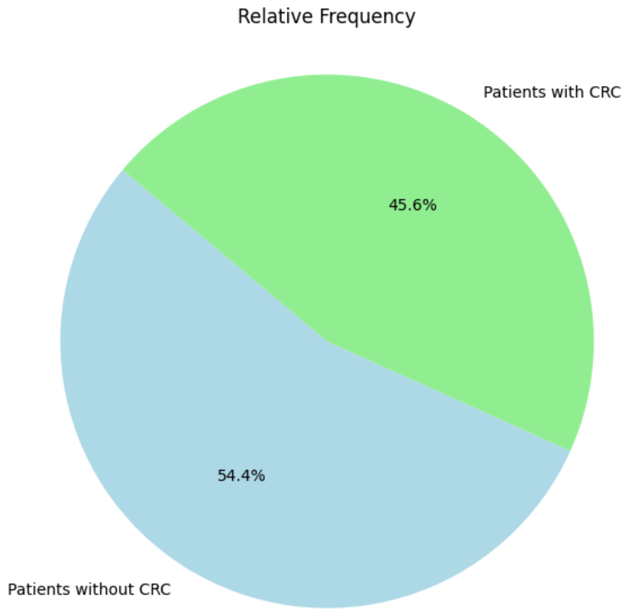

Figure 3.3: Amino acids - Relative Frequency of the 2 classes

3.1.2 Age

The plot 3.4 shows that patients with CRC are older than the healthier ones. This might indicate that this feature has a high information gain meaning that it is good splitter.

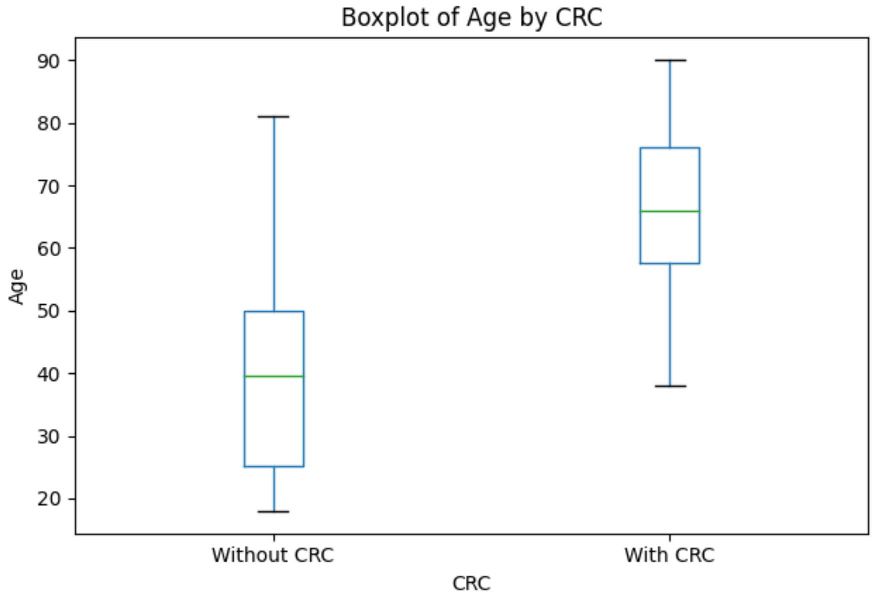

Figure 3.4: Amino acids - Presence of CRC by Age

3.1.3 Sex

Males exhibit a more pronounced difference in the presence and absence of the disease compared to females, as shown in figure 3.5. However, we can not conclude that males are more likely to have the disease.

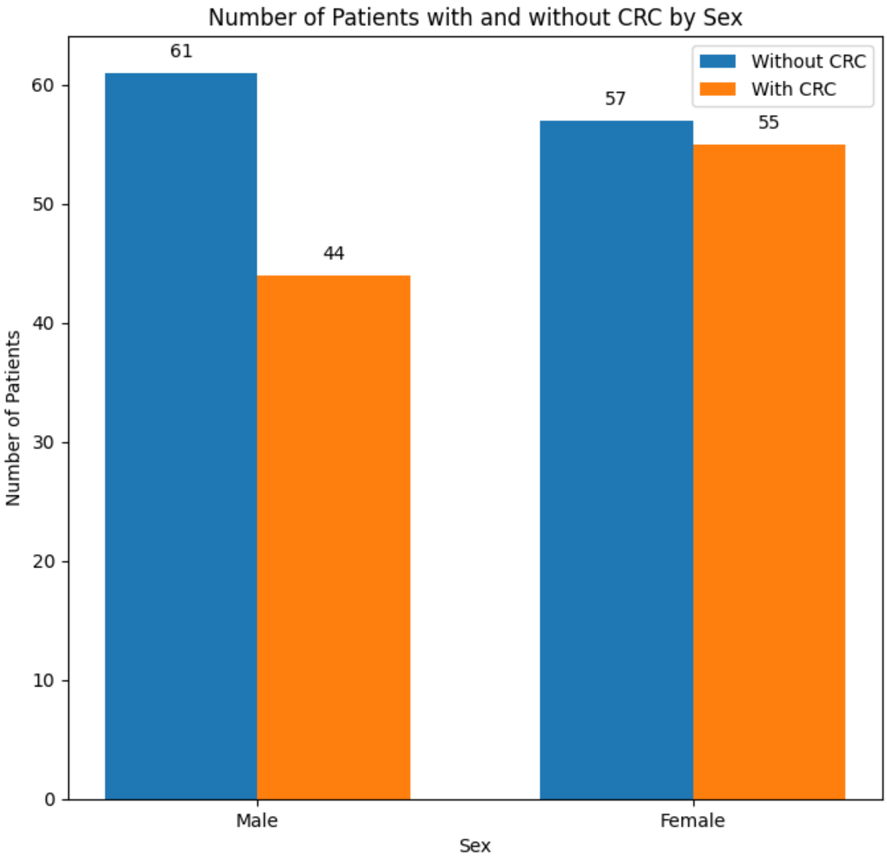

Figure 3.5: Amino acids - Number of patients with and without CRC by Sex

3.2 Acylcarnitines

In this section the Acylcarnitines dataset will be explored. The resulting dataset has a total number of 216 samples. The table 3.4 lists the features that were not present in the NonCRC dataset.

Table 3.3: Acylcarnitines Dataset - Features

Feature Description	Abbreviation	Range	Median	Average	% of Outliers	Measurement Unit
Decenoylcarnitine	c10:1	(0.0 - 0.90)	0.052	0.08	1.84%	micromoles / litre
Tiglylcarnitine/3-Methylcrotonylcarnitine	c5:1	(0.0 - 0.05)	0.0	0.00	3.23%	
Decadienoylcarnitine	c10:2	(0.0 - 0.30)	0.0	0.01	2.76%	
Glutarylcarnitine	c5dc	(0.0 - 0.65)	0.0	0.03	1.84%	
Tetradecenoylcarnitine	c14:1	(0.0 - 1.91)	0.103	0.15	1.84%	
Methylmalonylcarnitine/Succinylcarnitine	c4dc	(0.0 - 57.12)	0.2435	1.15	2.30%	
Malonylcarnitine	c3dc	(0.0 - 0.34)	0.0	0.01	1.84%	
Tetradecenoylcarnitine	c14	(0.0 - 0.85)	0.049	0.07	2.76%	
Octadecenoylcarnitine	c18:1	(0.0 - 3.21)	0.874	0.89	1.38%	
Hexadecenoylcarnitine	c16:1	(0.0 - 0.82)	0.0485	0.07	2.30%	
Octenoylcarnitine	c8:1	(0.0 - 0.50)	0.0405	0.08	1.38%	
Dodecanoylcarnitine	c12	(0.0 - 0.83)	0.015	0.04	2.30%	
Acetylcarnitine	c2	(0.0 - 28.15)	9.2915	9.76	1.38%	
Octadecanoylcarnitine	c18	(0.0 - 1.16)	0.366	0.38	1.38%	
Tiglylcarnitine/ 3-Methylcrotonylcarnitine	c5	(0.0 - 1.47)	0.01	0.07	1.38%	
Hexanoylcarnitine	c6	(0.0 - 0.80)	0.0	0.02	2.76%	
Decanoylcarnitine	c10	(0.0 - 2.26)	0.057	0.14	2.76%	
Tetradecadienoylcarnitine	c14:2	(0.0 - 0.53)	0.012	0.04	4.15%	
Octadecadienoylcarnitine	c18:2	(0.0 - 0.66)	0.233	0.24	0.92%	
Propionylcarnitine	c3	(0.0 - 5.0)	0.898	0.97	0.92%	
Octanoylcarnitine	c8	(0.0 - 2.10)	0.0395	0.10	2.76%	
3-Methylglutarylcarnitine	c6dc	(0.0 - 0.70)	0.0	0.02	1.84%	
Butyrylcarnitine Isobutyrylcarnitine	c4	(0.0 - 0.88)	0.0	0.08	3.23%	
Dodecenoylcarnitine	c12:1	(0.0 - 3.18)	0.0705	0.27	0.92%	
Hexadecanoylcarnitine	c16	(0.0 - 2.37)	0.694	0.75	2.30%	
Age of the Patient	age	(18 - 90)	47.0	47.46	N/A	positive integer
The sex of the Patient	sex	N/A	N/A	N/A	N/A	Male or Female

Table 3.4: Acylcarnitines Dataset - Features not present in the Non CRC dataset

Feature Description	Feature Abbreviation	Measurement Unit
Free Carnitine	C0	micromoles / litre
3-Hydroxybutyrylcarnitine/3-Hydroxyisobutyrylcarnitine	C4-OH	
3-Hydroxyisovalerylcarnitine/3-Hydroxy-2-methylbutyrylcarnitine	C5OH	
3-Hydroxytetradecenoylcarnitine	C14:1-OH	
3-Hydroxytetradecenoylcarnitine	C14-OH	
3-Hydroxyhexadecanoylcarnitine	C16-OH	
Octadecanoylcarnitine	C18	
3-Hydroxyoctadecenoylcarnitine	C18:1-OH	
3-Hydroxyoctadecanoylcarnitine	C18-OH	

Just like the amino acids, the next subsections will explore the number of patients with and without CRC and the age and sex features.

3.2.1 CRC cases

Similar to the amino acids, there are slightly more cases without CRC as shown on the graphs bellow. However we can still consider that the dataset is balanced.

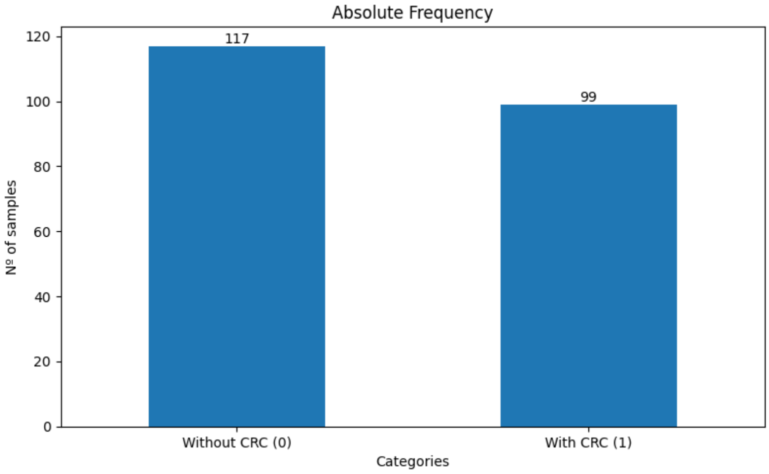

Figure 3.6: Acylcarnitines - Absolute Frequency of the 2 classes

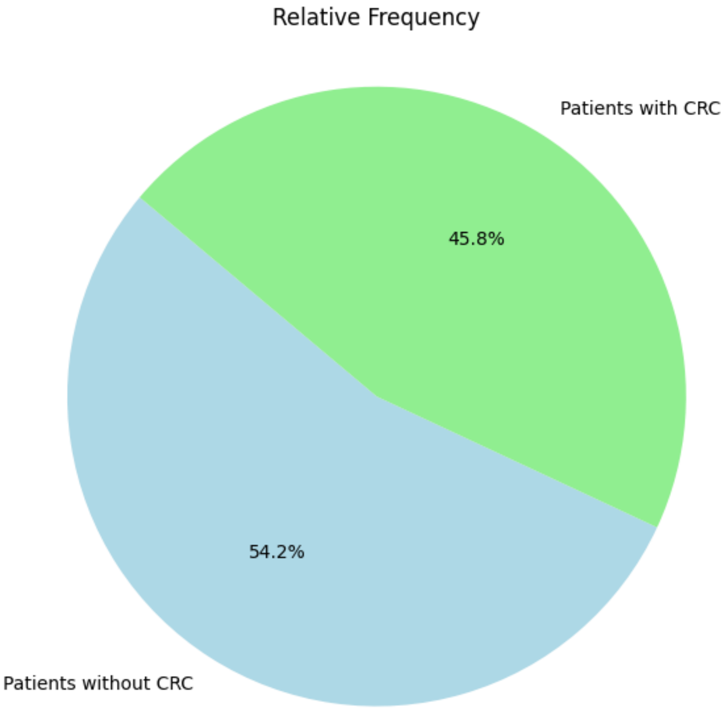

Figure 3.7: Acylcarnitines - Relative Frequency of the 2 classes

3.2.2 Age

The boxplot 3.8 shows that patients with CRC are older than the healthier ones. Also similar to the amino acids, this feature might be a good splitter.

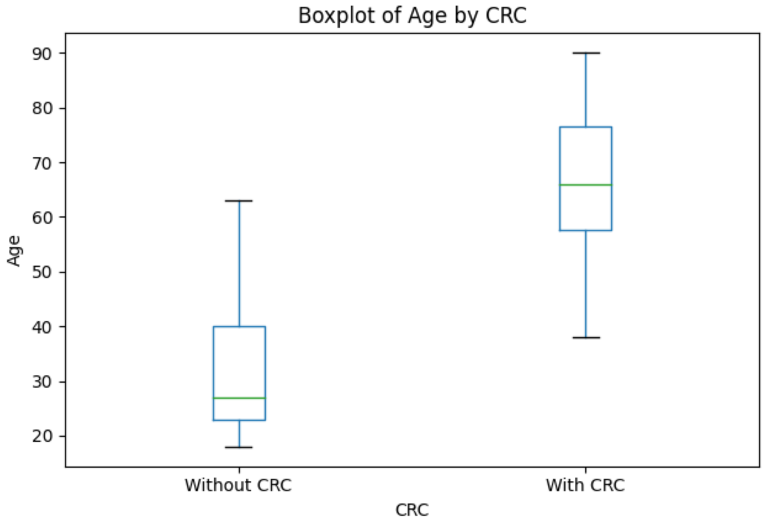

Figure 3.8: Acylcarnitines - Presence of CRC by Age

3.2.3 Sex

Males exhibit a more pronounced difference in the presence and absence of the disease compared to females, as shown in Figure3.9, showing a very similar pattern to the amino acids.

3.3 Encoding & Normalization

Most machine learning algorithms require numerical input. Encoding helps convert non-numerical data (e.g., categorical or textual data) into a numerical format, making it suitable for these algorithms [29].

There's only one categorical feature, the "Sex". Males will be encoded with the value 0 and Females with 1.

Normalization is the process of transforming features to be on a similar scale. This process is crucial because the scale of the features can significantly affect the performance of many ML models. KNeighborsClassifier and Support Vector Machines are examples of classifiers that are sensitive to the scale, in the other hand, DecisionTrees and RandomForest classifiers are not. Normalization techniques are primarily applied to numeric data. The goal is to change the values of numeric columns in the dataset to a common scale without losing information [72].

There are several scikit-learn normalization methods. The following thre are among the most popular ones [73]:

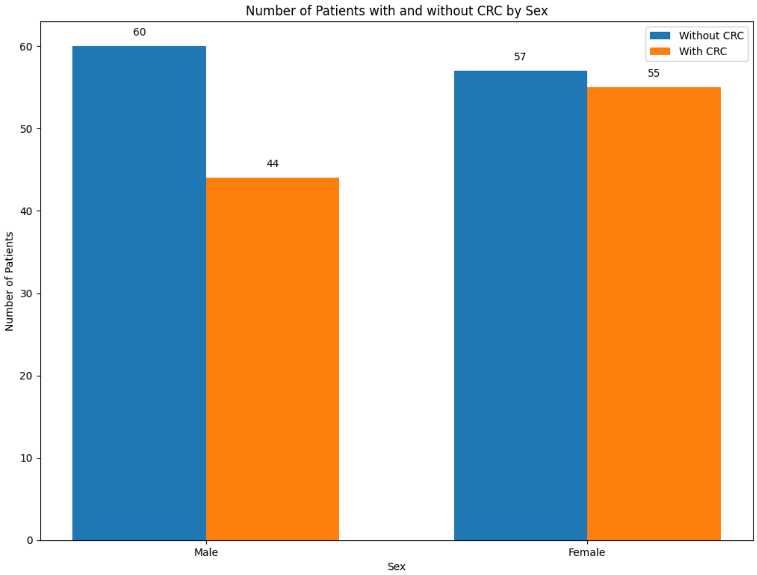

Figure 3.9: Acylcarnitines - Number of patients with and without CRC by Sex

- RobustScaler: Useful when the data contains many outliers.
- StandardScaler: Useful when the data distribution is normal. Sensitive to outliers.
- MinMaxScaler: Useful when the data does not follow a normal distribution. Sensitive to outliers.

The Z-score is a commonly used statistical measurement to describe a value's relationship to the mean of a group of values. If a Z-score is 0, it indicates that the data point's score is identical to the mean score. If the Z-score of a data point is more than 3, it indicates that the data point is quite different from the other data points. Such a data point can be an outlier. Z-score can be both positive and negative. The farther away from 0, higher the chance of a given data point being an outlier [74].

Both aminoacids (table 3.1) and acylcarnitines (table 3.3) present very low presence of outliers.

There are several statistical tests to determine if data follows a normal distribution. The Shapiro-Wilk is among the most powerful tests. [75]. Upon applying this test to given feature, it will output the p-value. If this p-value is less than the commonly used 0.05, there is evidence that the data is not normally distributed.

After applying this test to both aminoacids and acylcarnitines, none of them returned a p-value less than 0.05. This indicates that both datasets are not normally distributed.

Given that there are no significant outliers and the data is not well distributed, the **MinMaxScaler** will be used as the normalization method.

3.4 Feature Selection

Feature selection is a crucial step in building a ML model. It refers to selecting the most important features (or variables) from the original set of features in the dataset. This process helps in reducing the dimensionality of the data, which in turn helps in building a more interpretable, simpler, and more efficient model. [76]

It maintains the initial representation of the variables by simply choosing a subset of them. Consequently, the original meaning of the variables is preserved, which provides the benefit of being easily interpretable by an expert in the field. It also improves the model's prediction performance. [77]

There are several categories of feature selection methods, such as: [78]

- **Filter Methods:** Simple and fast as they evaluate features independently of the classifier, making it a preprocessing step.
  - *Examples:* Variance Threshold, Univariate Feature Selection
- **Wrapper Methods:** Dependent on a classifier and computationally more expensive than filter methods but can lead to better results.
  - *Examples:* Recursive Feature Elimination (RFE), Forward Selection / Backward Elimination
- **Embedded Methods:** Provide a good trade-off between model performance and computational costs.
  - *Examples:* Lasso Regression, Tree-based models

There is another feature selection method called "Permutation Feature Importance" that does not fit in any of the categories previous listed. It's a model-agnostic method that can be applied to any fitted model. However it requires a model to be trained and does not create a subset leading to a more complex implementation. [79] For these reasons it will be excluded from this work.

Since computational power is not an issue, the wrapper methods would be the choice. Specifically the RFECV which is a variant of RFE but with cross-validation. However, it's a classifier dependent method and not all classifiers mentioned at section 2.3.1 are directly compatible since they lack "_coef" or "_feature_importances" attributes.

Filter methods, even with their lower performance, can be considered as an "universal" solution. Based on the scikit-learn library [80], the **SelectPercentile** will be used.

SelectPercentile works by computing the univariate statistical measure (parameter score_func) between each feature and the target variable, and then selecting the specified percentile of features with the highest scores (parameter percentile). [81] For the parameter score_func the default value will be used (f_classif - ANOVA F-value) since it's well suited for a binary classification problem. Since automatic feature selection is desirable, the percentile value cannot be precisely pre-determined. Therefore, a range of values will be used from 10 up to 100 (all features).

3.5 Final Remarks

In this section we described the two initial data steps of the CRISP methodology where the amino acids and acylcarnitines datasets were built and explored, the encoding, normalization and feature selection methods were defined. Next step will be the modeling.

Chapter 4

Modeling

Model selection is the process of choosing the most suitable model from a set of candidates for a specific dataset. This involves a comparison of different models, selecting the most appropriate one based on certain metrics (such as accuracy, precision, and recall), and then validating its performance on both the training and unseen data.

To ensure robustness, techniques such as Cross-Validation are employed, where the model is tested on multiple subsets of the data to confirm its consistent performance. Additionally, Hyper Parameter Tuning (running the model with different specific classifier parameters) is essential to optimize the model's performance. The diagram 4.1 illustrates how the process is arranged.

The following subsections will describe these techniques and how they're going to be applied in this context.

4.1 Evaluation Metrics

A machine learning model is evaluated based on its ability to make accurate predictions on new data that it has not seen before [82].

Bellow a list of some commonly used evaluation metrics for supervised machine learning models:

1. Confusion Matrix - This is a table that shows the number of true positives, true negatives, false positives, and false negatives.

	Predicted	
Actual	Yes	No
Yes	TP	FN
No	FP	TN

2. Accuracy - The ratio of the number of correct predictions to the total number of predictions. It's a common metric, but not always the best one for imbalanced classes or when false negatives are particularly costly. Correct Predictions / Total Predictions.

3. Precision (Positive Predictive Value) - The ratio of the number of true positive predictions to the total number of positive predictions (true positives + false positives). It is important in situations where minimizing false positives is crucial.

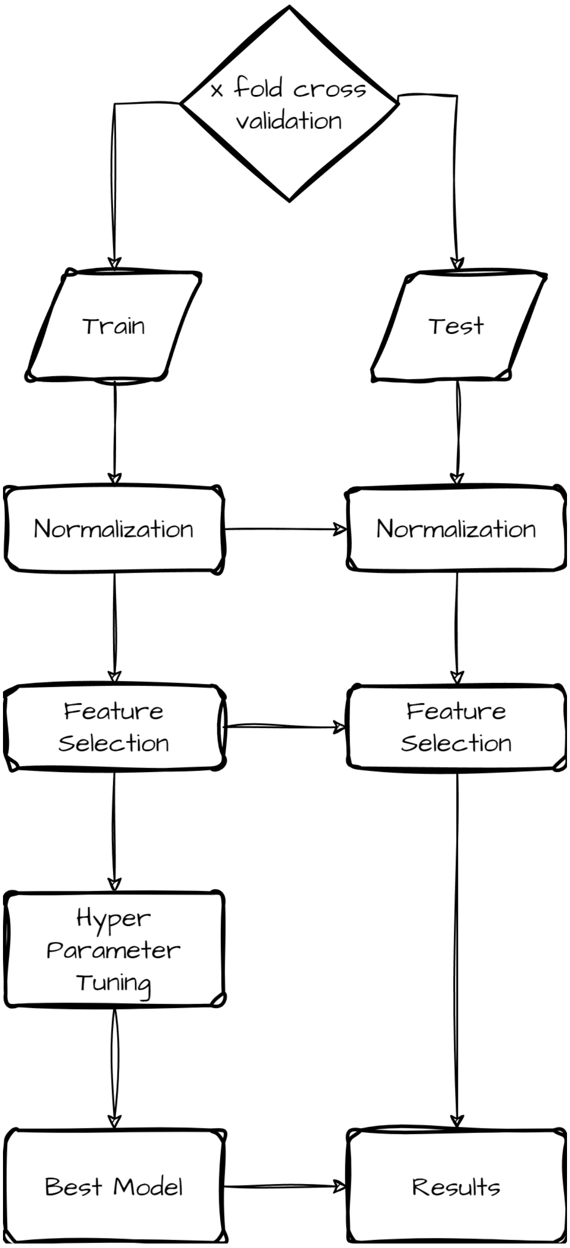

Figure 4.1: Datasets preparation

4. Error - In a binary classification problem like CRC detection, the error rate would be the sum of the false positives and false negatives divided by the total number of sample.

5. Recall (Sensitivity or True Positive Rate) - The ratio of the number of true positive predictions to the total number of actual positives (true positives + false negatives). This is a particularly important metric in CRC detection, as it is crucial to minimize false negatives. TP / (TP + FN)

6. F1-score - This is the harmonic mean of precision and recall. It is a good metric to consider for the balance between precision and recall. 2TP / (2TP + FP + FN)

7. Area Under the Receiver Operating Characteristic Curve (AUC-ROC) - The ROC curve is a plot of the true positive rate (recall) against the false positive rate. The AUC is the area under this curve. A model with perfect classification ability will have an AUC-ROC of 1, whereas a model with no classification ability will have an AUC-ROC of 0.5.

As previously mentioned, selecting the best model requires the use of certain metrics that will indicate if one model is outperforming others.

For a binary classification problem like CRC detection, it is crucial to minimize false negatives as it could be life-threatening to incorrectly classify a cancerous sample as benign.

From the metrics previously described, the F1-Score and Recall will be used.

While the F1-score is a good overall measure of a model because it balances precision and recall, in this case, recall is more important due to the potential consequences of false negatives. However, it is also important to consider the overall performance of the model, and not just focus on a single metric. Therefore, if one model outperforms the other on recall, it would generally be more advisable to choose that model, even if its F1-score is slightly lower.

4.2 x-Fold Cross Validation

The process of building a model requires a train and test subsets from the original dataset. The first is used to train the model, the second to evaluate it. This train-test split can be done in several ways. However, the model's performance can vary upon different subsets even if they come from the same dataset. Therefore, different train-test splits should be used. One of these processes is the x-Fold Cross Validation which works by dividing the dataset into k equally sized "folds" or subsets. In each iteration, one of these folds is used as the test set, while the remaining (k-1) folds are combined to form the training set. The model is trained on the training set and then evaluated on the test set, resulting in one performance metric for that iteration. This process is repeated k times, with each fold taking a turn as the test set. The final evaluation score is typically an average of the scores from all iterations. This helps in obtaining reliable performance estimates and reduces the risk of overfitting.[83]. The figure 4.2 illustrates the behaviour of K-fold Cross-Validation with K=5.

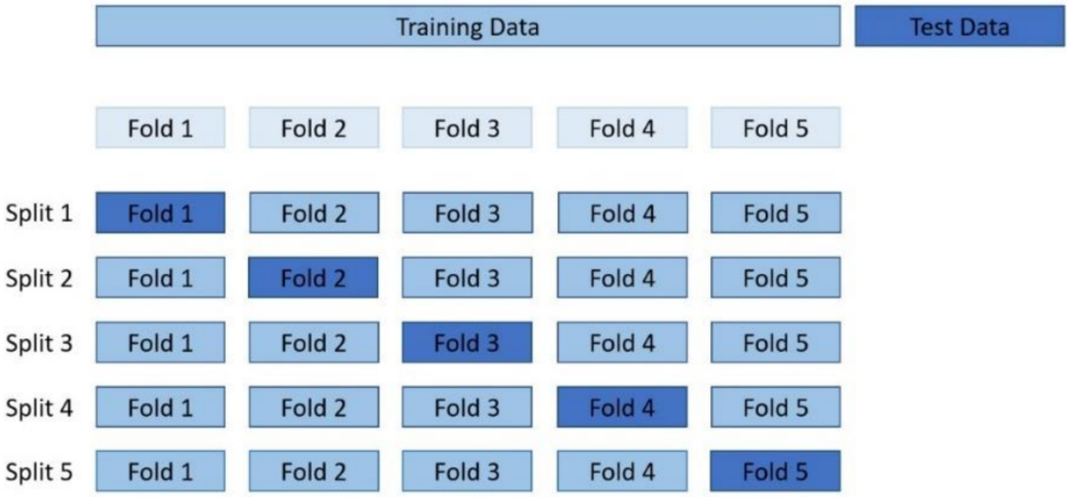

Figure 4.2: K-fold Cross-Validation for K=5 (adapted from [84])

Overfitting is a common issue in ML, occurring when a model excessively adapts to the training data, to the point of capturing the data's noise instead of the fundamental pattern. This leads to excellent performance on the training data but subpar performance on new, unseen data. Essentially, the model becomes overly complex, fitting the training data so precisely that it fails to generalize effectively to new data.[85]

The StratifiedKFold from scikitlearn, which is a variation of k-fold cross-validation that returns stratified subsets (also known as folds) where they're made by preserving the percentage of samples for each class. [86]. In this work, the number of splits will be the default, 5.

For each split, the normalization, feature selection and hyper parameter tuning is applied upon evaluating the different classifiers described at subsection 2.3.1.

4.3 Hyper Parameter Tuning

Hyperparameters are specific parameters whose values are predetermined before initiating the training process. Hyperparameter tuning involves searching for the optimal set of hyperparameters from all possible combinations. The objective is to maximize the evaluation score. It is a crucial step in optimizing a model's performance and it's considered a model-centric approach [87].

While hyperparameter tuning focuses on optimizing the model and its configuration, it is often recommended to prioritize data-centric approaches for optimizing a model's performance. Data-centric optimization involves cleaning, sampling, augmenting, or modifying the data. Despite the complexity and sophistication of a model, its performance will be limited if the quality of the data or features is poor. This concept is the well-known expression, "garbage in, garbage out (GIGO)" [87].

Scikit-learn provides several hyperparameter tuning classes. Since computational resources and time are not an issue, the **GridSearch** can be used. This method performs an exhaustive search over a specified parameter grid. It tries every combination of hyperparameters in the grid and selects the best combination based on cross-validated performance [88].

The following hyperparameters will be used:

- **KNeighbors**
  - **Number of neighbors** (scikit-learn: *n_neighbors*): The values 3, 5(default), 7, 9 and 11 will be used.
  - **Weights** (scikit-learn: *_weights*): The values 'uniform' (default) and 'distance' will be used.
- **DecisionTree**
  - **Maximum depth** (scikit-learn: *max_depth*): The values 'None', 5 and 10 will be used.
- **GaussianNB (Gaussian Naive Bayes)**
  - **Variance smoothing** (scikit-learn: *var_smoothing*): The values '1e-9' (default), '1e-8' and '1e-7' will be used.
- **SVC (Support Vector Machine)**
  - **Kernel type** (scikit-learn: *kernel*): The values are ‘linear‘, ‘poly‘, ‘rbf‘ (default), ‘sigmoid‘.
  - **Regularization parameter** (scikit-learn: *C*): The values 0.1, 1 (default) and 10 will be used.
- **MLP(Multi-Layer Perceptron, a type of artificial neural network)**
  - **Activation function** (scikit-learn: *activation*): The values ‘identity‘, ‘logistic‘, ‘tanh‘, ‘relu‘ (default) will be used.
  - **Size of hidden layers** (scikit-learn: *hidden_layer_sizes*): The values (50,) and (100,) will be used where (100,) is the default.
- **RandomForest**
  - **Number of trees** (scikit-learn: *n_estimators*): The values 50, 100 (default), 150 will be used.
  - **Maximum depth** (scikit-learn: *max_depth*): The values None (default), 10 and 20 will be used.
- **GradientBoosting**
  - **Number of boosting stages** (scikit-learn: *n_estimators*): The values 50, 100 (default), 150 will be used.

4.4 Final Remarks

In this section we described the modeling step CRISP methodology where the metrics, the train/test splits and the algorithms to be used were presented. Next chapter the evaluation and explanations will be presented.

Chapter 5

Evaluation & Explanations

The results were recorded for each classifier. After running all the folds, described at section 4.2, a mean was applied to both F1-score and Recall. These metrics allowed to understand the overall performance of each classifier. A "Control" version was also recorded, so a baseline could be established. This record contains the classifier's performance without the application of any preprocessing steps. The performance with normalization,described at section 3.3, and feature selection were recorded separately, resulting in three pairs (F1-Score and Recall) for each classifier. For the feature selection, describer at section 3.4, the most common percentile for each classifier was also recorder as well as the most common features removed.

For the explanations, three different approaches were applied:

- Overall Feature Importance on the original dataset. SHAP was applied with the entire dataset that train the model.
- Overall Feature Importance on the M1 cases. SHAP was applied to the patients in M1 phase. The reason for the use of these cases for testing the model's performance and explanations, is described at the beginning of chapter 3.
- Single M1 instance. SHAP was applied to explain a single instance of the M1 phase, specifically for wrong predicted instances. This plot is the one that is going to be shown once the system is deployed. Also, only the top 10 most influencing features are going to be presented.

The diagram 5.1 illustrates the overall process of results evaluation and explanations.

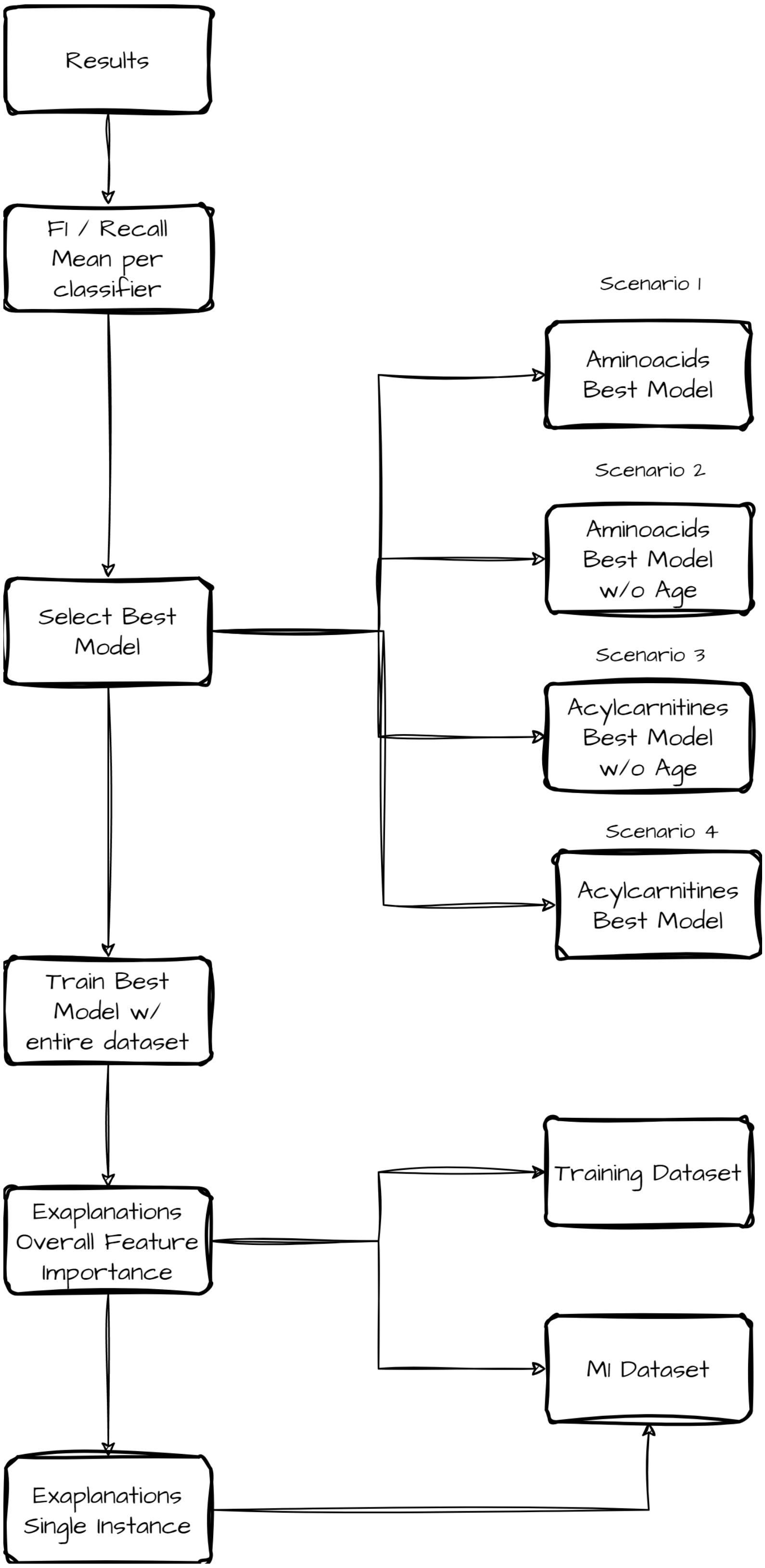
Results
F1 / Recall Mean per classifier
Select Best Model
Scenario 1
Aminoacids Best Model
Scenario 2
Aminoacids Best Model w/o Age
Scenario 3
Acylcarnitines Best Model w/o Age
Scenario 4
Acylcarnitines Best Model
Train Best Model w/ entire dataset
Exaplanations Overall Feature Importance
Training Dataset
M1 Dataset
Exaplanations Single Instance

Figure 5.1: Evaluation Diagram

Upon agreement with healthcare professionals, it was decided to build models with and without the feature "age". This feature is considered a discriminative feature, meaning that it has a strong correlation with the target. Even though this feature can enhance the performance of classification models, when applying explanations, they'll translate something that healthcare professionals already know, older patients are typically more susceptible to have the disease as we can see at the feature blox plots for amino acids 3.4 and acylcarnitines 3.8. For these reasons, there were four runs at total, with and without "age" for each of the metabolic profiles. These runs are called scenarios and their results will be described at the sections bellow.

5.1 Scenario 1 - Amino acids All Features

This section will explore the results for Amino acids using all features, presenting the classifiers performances first followed by the explanations.

5.1.1 Model Selection

This subsection will present the classifiers performances and the best model for the amino acids dataset.

Normalization

Overall, the impact of normalization on classifier performance appears to be classifier-specific, with some benefiting and others experiencing declines as we can see at table 5.1. KNeighbors, for instance, experiences a noticeable performance decline upon normalization, with both F1 and Recall dropping by around 5%. This is surprising since KNN usually benefits from normalization as stated at section 3.3. While KNeighbors experienced a substantial improvement of approximately 22% in both F1 and Recall metrics, DecisionTree and RandomForest saw slight declines. GaussianNB's performance remained unchanged, whereas SVC and MLP demonstrated gains, with SVC having a notable 14% rise in F1. GradientBoosting displayed a marginal decrease in its performance metrics post-normalization. Overall, normalization proved beneficial for some classifiers and neutral or slightly detrimental for others in the context of this dataset.

ClassifierName	Mean F1-Recall Control	Mean F1-Recall Norm
KNeighbors	0.65-0.65	0.79-0.80 (22.52% - 22.32%)
DecisionTree	0.80-0.83	0.79-0.82 (-1.78% - 0.92%)
GaussianNB	0.72-0.73	0.72-0.73 (0.00% - 0.00%)
SVC	0.71-0.72	0.81-0.82 (14.27% - 13.68%)
MLP	0.49-0.53	0.52-0.55 (7.88% - 3.28%)
RandomForest	0.88-0.91	0.87-0.90 (-1.76% - -0.95%)
GradientBoosting	0.83-0.86	0.82-0.85 (-0.64% - -0.72%)

Table 5.1: Comparison of Classifier Performance with and without Normalization for Amino acids with All Features

Feature Selection

Upon applying feature selection, classifiers showed varied responses. KNeighbors and GaussianNB witnessed marked improvements, with KNeighbors gaining nearly 25% in F1. SVC also saw a significant rise of over 15% in its F1 metric. MLP, although starting with lower baseline scores, showcased a big increase of over 60% in its F1 score. Conversely, DecisionTree and RandomForest experienced minor declines in certain metrics. GradientBoosting's F1 improved by 6.5%, ensuring consistency with its high baseline. The optimal feature subsets, as indicated by the best percentiles, ranged from 20% to a complete set at 100%, underscoring the uniqueness of each classifier's feature preference.

These results can be seen in the table 5.2.

ClassifierName	Mean F1-Recall Control	Mean F1-Recall FS	Best Percentile
KNeighbors	0.65-0.65	0.81-0.80 (+24.90% - +22.54%)	50
DecisionTree	0.80-0.83	0.81-0.80 (+0.64% - -2.93%)	20
GaussianNB	0.72-0.73	0.82-0.73 (+13.39% - 0.00%)	30
SVC	0.71-0.72	0.82-0.82 (+15.54% - +13.68%)	50
MLP	0.49-0.53	0.78-0.65 (+61.48% - +23.36%)	100
RandomForest	0.88-0.91	0.90-0.88 (+1.78% - -2.96%)	60
GradientBoosting	0.83-0.86	0.88-0.87 (+6.50% - +1.34%)	100

Table 5.2: Comparison of Classifier Performance with and without Feature Selection for Amino acids with All Features

The features "sex", "pro" (Proline), "abu" (Aminobutyric acid) were among the most frequently removed features across the classifiers, suggesting that these features might not be as influential for the given dataset.

Hyperparameters and Processing Times

The table 5.3 summarizes the processing time and the most common parameters for the different classifiers. KNeighbors and DecisionTree classifiers have the least processing time, each taking 10 seconds and 4 seconds, respectively. On the other hand, RandomForest took notably longer with a processing time of 3 minutes and 25 seconds. The classifiers SVC, MLP, and GradientBoosting took moderate processing times ranging between 14 to 52 seconds.

For the KNeighbors classifier, the most common hyperparameters were 9 neighbors with uniform weights. For the DecisionTree classifier, the most frequent configuration did not limit the maximum depth of the tree. The GaussianNB often applied the default variance smoothing factor. The SVC classifier typically utilized a linear kernel with a regularization parameter of 1. The MLP classifier commonly had an identity activation function and hidden layer sizes of 100 units. The RandomForest was frequently set with a maximum depth of 10 and 50 estimators. Lastly, the GradientBoosting classifier commonly employed 50 estimators.

ClassifierName	Processing Time	Most Common HyperParameters
KNeighbors	0m:10s	{'neighbors': 9, 'weights': 'uniform'}
DecisionTree	0m:4s	{max_depth': None}
GaussianNB	0m:4s	{'var_smoothing': 1e-09}
SVC	0m:17s	{'C': 1, 'kernel': 'linear'}
MLP	0m:14s	{'activation': 'identity', 'hidden_layer_sizes': (100,)}
RandomForest	3m:25s	{'max_depth': 10, 'n_estimators': 50}
GradientBoosting	0m:52s	{'n_estimators': 50}

Table 5.3: Processing Times and Most Common HyperParameters for Amino acids with All Features

Best Model

Based on the previous listed results, the best classifier was the **RandomForests**. Even though it had the biggest computational cost, it outperformed the others with a F1-Score 0.90 of and Recall of 0.88. Without normalizing, with a percentile of 50%, the _max_depth set to 10 and n_estimators set to 50. If the computational cost was to be considered, the GradientBoosting could be an alternative since it took three times less to process and the performance is not that lower.

5.1.2 Explanations

For this dataset, the dominating feature is the "age". Details about the explanations at the subsections bellow. The process took about 5 seconds for the original dataset and about 1 second for the M1 samples using TreeExplainer.

Overall Feature Importance on the original dataset

The top four most influencing features for the original dataset are the "age", "cys2" (Cystine), "his"(Histidine) and "phe" (Phenylalanine). When the value of the age increases, the model is more likely to predict a sample as having CRC. The same happens for "cys2". On the contrary,when the values of the features "hys" and "phe" increase, the model is more likely to predict a sample as not having CRC. This confirms the age as being a discriminative feature as stated at the beginning of this chapter. The summary plot for the original dataset values can be seen at 5.2

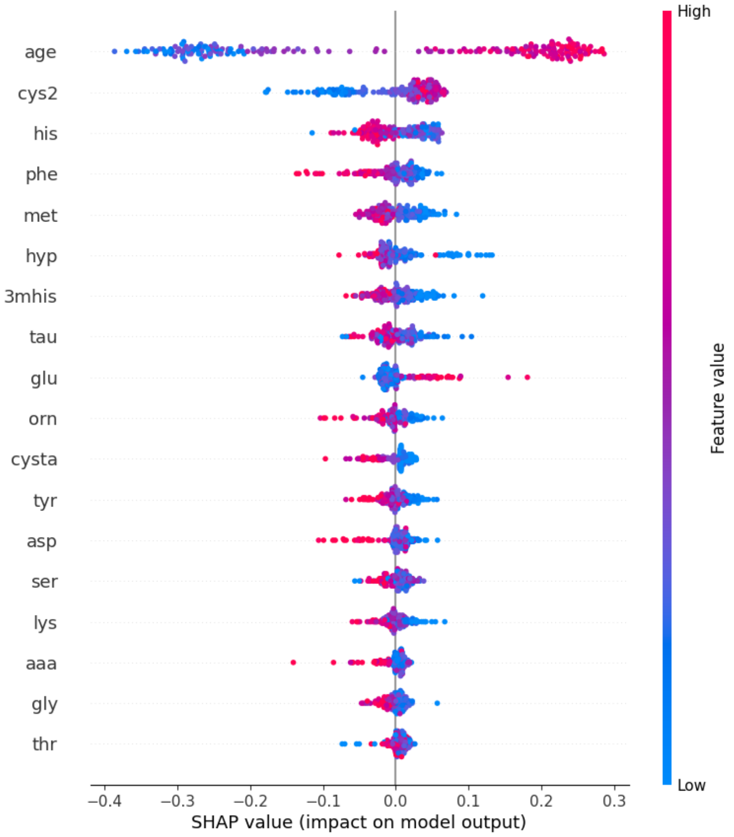

Figure 5.2: Shap Summary Plot for the amino acids original dataset

Overall Feature Importance on the M1 samples

When applying to the unseen M1 samples, there's no big difference in the features importances. The age is still the most influential feature, however, as we can see at the summary plot 5.3, no matter the value of the age, it still pushed almost all instances towards the positive class (CRC detected).

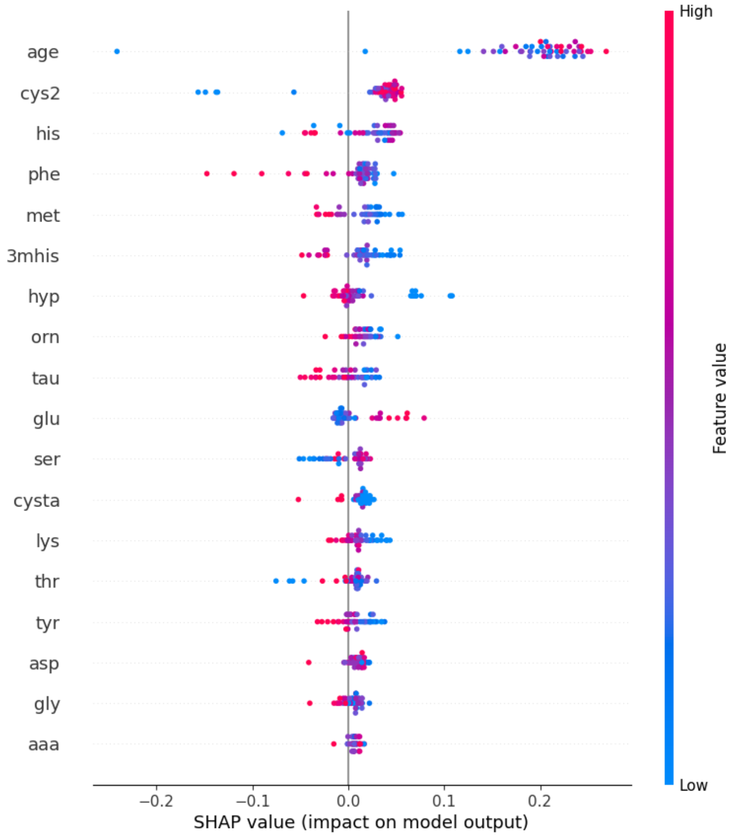

Figure 5.3: Shap Summary Plot for M1 samples for amino acids

Single M1 instance

From the 48 M1 samples, the model correctly predicted 46 as having CRC.

The table 5.4 presents the features and shap values of one of the wrongly predicted samples. The table is ordered by the features that contributed for the prediction to be 0 (without CRC) followed by the ones that contributed for being 1 (with CRC).

Table 5.4: Shap Values for Single Amino acids M1 Wrong Prediction

Feature	Description	Value	SHAP Value
age	Age	46	-0.2049
his	Histidine	101	-0.0928
tau	Taurine	133	-0.0264
glu	Glutamic acid	37	-0.0227
cys2	Cystine	51	-0.0215
ser	Serine	149	-0.0193
hyp	Hydroxyproline	9	-0.0199
gly	Glycine	256	-0.0124
thr	Threonine	152	-0.0018
orn	Ornithine	103	-0.0009
phe	Phenylalanine	45	0.0328
3mhis	3-Methylhistidine	4	0.0163
asp	Aspartic Acid	3	0.0096
tyr	Tyrosine	46	0.0121
met	Methionine	21	0.0070
lys	Lysine	207	0.0054
aaa	Alpha-Aminoadipic acid	6	0.0037
cysta	Cystathionine	0	0.0036

The feature "age" with a SHAP value of -0.1809 is the most influential, pushing the prediction lower. This implies that the age value of 46 is associated with a decrease in the likelihood of the target being 1 (crc detected). The features "his" and "cys2" also have relatively high negative magnitudes. "met" (0.0291), "phe" (0.0205), and "tyr" (0.0108) are features that push the prediction higher, implying that these feature values are associated with an increased likelihood of the target being 1 for this instance. Features with Minimal Influence: Features like "gly" (-0.0001), "aaa" (0.0011), and "orn" (-0.0027) have very low magnitudes, suggesting they had a minimal influence on the prediction for this instance. The force plot 5.4 provides a visual representation of the shap values and their corresponding features that have a bigger influence on the prediction. The features with red color indicate, that they're making the instance more likely to be predicted as the class 1 (with CRC). The blue Indicates the opposite, where they're making the instance more likely to be predicted as the class 0 (without CRC).

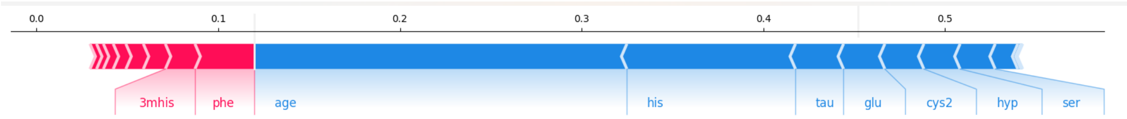

Figure 5.4: Shap Force Plot for Single Amino acids M1 Prediction

5.2 Scenario 2 - Amino acids Without Age

This section will explore the results for Amino acids using all features except the age. The structure is very similar to the previous one.

5.2.1 Model Selection

Normalization

as we can see at table 5.5. KNeighbors exhibited a notable enhancement in its F1 and Recall scores when shifting from the Control to the Norm dataset, with an increase of approximately 26.31% and 25.48%, respectively. Conversely, the DecisionTree and RandomForest classifiers saw a decrease in performance, with the former dropping by about 5.80% in F1 and 5.57% in Recall, and the latter declining by around 5.74% and 5.08%. GaussianNB maintained consistent performance across both datasets. The SVC and MLP classifiers demonstrated modest improvements, while the GradientBoosting classifier had a slight dip in its metrics, with a reduction close to 1.18% in F1 and 1.47% in Recall when applied to the Norm dataset.

ClassifierName	Mean F1-Recall Control	Mean F1-Recall Norm
KNeighbors	0.61-0.61	0.77-0.77 (+26.35% - +25.50%)
DecisionTree	0.67-0.67	0.63-0.63 (-5.92% - -5.49%)
GaussianNB	0.62-0.63	0.62-0.63 (0.00% - 0.00%)
SVC	0.74-0.75	0.78-0.78 (+5.11% - +4.69%)
MLP	0.52-0.54	0.53-0.56 (+1.40% - +3.94%)
RandomForest	0.75-0.75	0.70-0.71 (-5.72% - -5.08%)
GradientBoosting	0.72-0.73	0.72-0.72 (-1.18% - -1.47%)

Table 5.5: Comparison of Classifier Performance with and without Normalization for Amino acids without Age

Feature Selection

With the removal of the "Age" feature, all classifiers started using more features. KNeighbors experienced a 30.05% enhancement in F1 score and a 25.56% rise in recall with the best performance at the 50th percentile, classifiers like DecisionTree and RandomForest exhibited a slightly better F1 score with FS, but at the cost of reduced recall. Specifically, DecisionTree optimized its performance at the 30th percentile, and RandomForest at the 60th. Meanwhile, SVC and MLP demonstrated pronounced improvements, particularly at the 100th and 50th percentiles, respectively. GaussianNB and GradientBoosting also saw benefits from FS, optimally at the 80th and 90th percentiles. More details at table 5.6.

ClassifierName	Mean F1-Recall Control	Mean F1-Recall FS	Best Percentile
KNeighbors	0.6067-0.6130	0.7892-0.7694 (+30.05% - +25.56%)	50
DecisionTree	0.6658-0.6676	0.7437-0.6393 (+11.71% - -4.24%)	30
GaussianNB	0.6204-0.6346	0.6864-0.6346 (+10.61% - 0.00%)	80
SVC	0.7408-0.7474	0.8303-0.7824 (+12.06% - +4.69%)	100
MLP	0.5235-0.5430	0.6700-0.4990 (+27.96% - -8.11%)	50
RandomForest	0.7473-0.7473	0.7844-0.7237 (+4.96% - -3.16%)	60
GradientBoosting	0.7248-0.7290	0.7594-0.7183 (+4.76% - -1.46%)	90

Table 5.6: Comparison of Classifier Performance with and without Feature Selection for Amino acids without Age

The features "sex", "pro" (Proline), "leu" (Leucine) and "cit" (Citrulline) were among the most frequently removed features across the classifiers, suggesting that these features might not be as influential for the given dataset.

Hyperparameters and Processing Times

The table 5.7 summarizes the processing time and the most common parameters for the different classifiers. KNeighbors and DecisionTree classifiers have the least processing time, each KNeighbors classifier took approximately 9.84 seconds to process and commonly used 5 neighbors with uniform weights. The DecisionTree processed in 3.45 seconds, typically without any depth restriction. GaussianNB had a processing time of 3.33 seconds with a frequent default value for var_smoothing. The SVC classifier took 1 minute and 27 seconds (considerably higher than when using all features), predominantly using the RBF kernel with a C value of 10. The MLP processed in 12.08 seconds, regularly with a relu activation function and a single hidden layer of 100 neurons. RandomForest, having the longest processing time of 3 minutes (slightly lower than when using all features), usually had a max depth of 10 and utilized 50 estimators. Finally, the GradientBoosting classifier took 49 seconds with a recurring number of estimators set to 100.

ClassifierName	Processing Time	Most Common HyperParameters
KNeighbors	0m:09s	{'neighbors': 5, 'weights': 'uniform'}
DecisionTree	0m:03s	{'max_depth': None}
GaussianNB	0m:03s	{'var_smoothing': 1e-09}
SVC	1m:24s	{'C': 10, 'kernel': 'rbf'}
MLP	0m:12s	{'activation': 'relu', 'hidden_layer_sizes': (100,)}
RandomForest	3m:1s	{'max_depth': 10, 'n_estimators': 50}
GradientBoosting	0m:49s	{'n_estimators': 100}

Table 5.7: Processing Times and Most Common HyperParameters for Amino acids without Age

Best Model

Based on the previous listed results, the best classifier was the **SVC**. Using normalization, percentile as 100 (no features removed) and using the RBF kernel with a C value of 10 as hyperparameters.

5.2.2 Explanations

For this dataset, the dominating feature is the "cys2" (Cystine). Details about the explanations at the subsections bellow. The process took about 28 minutes for the original dataset and about 3 minutes for the M1 samples using KernelExplainer.

Overall Feature Importance on the original dataset

The top four most influencing features for the original dataset are the "cys2" (Cystine), "his"(Histidine) and "cysta" (Cystathionine). The "cys2" was in second place when the age feature was used, suggesting that this feature may also be a discriminative feature. The "sex" came in fourth place without much discrimination between males and females. The summary plot for the original dataset values can be seen at 5.5

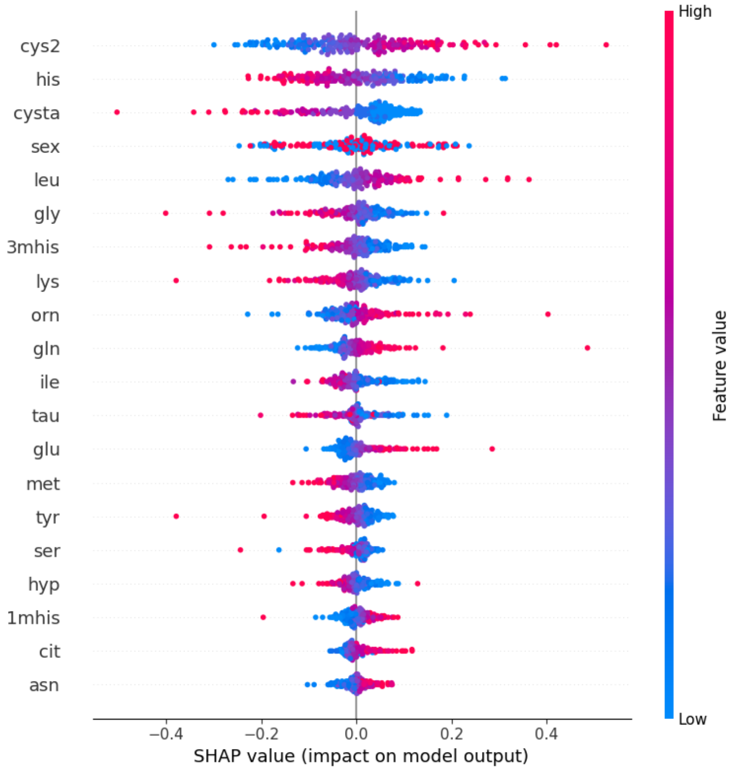

Figure 5.5: Shap Summary Plot for the amino acids original dataset Without Age

Overall Feature Importance on the M1 samples

When applying to the unseen M1 samples, the top feature changed to "leu" (Leucine). The "cys2" came in second place followed by "sex" and "hys" (Histidine). This time the females were more likely to be diagnosed with CRC. The importance of this feature came with surprise since it was one of the most removed features among the several classifiers.

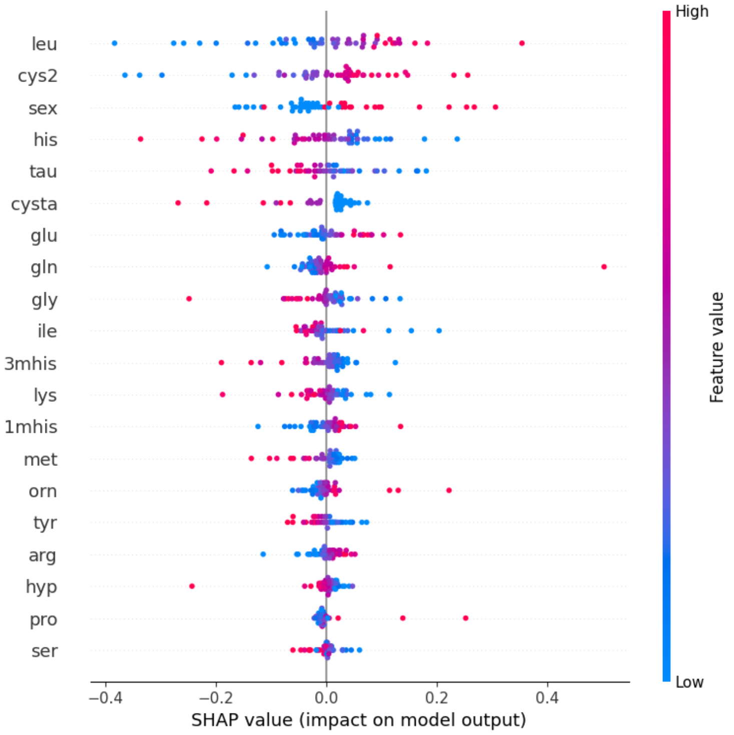

Figure 5.6: Shap Summary Plot for M1 samples for the amino acids dataset Without Age

Single M1 instance

From the 48 M1 samples, the model correctly predicted 41 as having CRC.

The table 5.8 presents the features and shap values of one of the wrongly predicted samples. Almost all features contributed for the wrong prediction.

Table 5.8: Shap Values for Single Amino acids M1 Wrong Prediction Without Age

Feature	Description	Value	SHAP Value
cysta	Cystathionine	2	-0.2780
tau	Taurine	136	-0.2075
sex	Sex	0	-0.1319
orn	Ornithine	75	-0.0718
1mhis	1-Methylhistidine	6	-0.0701
his	Histidine	72	-0.0523
ile	Isoleucine	92	-0.0468
tyr	Tyrosine	69	-0.0360
glu	Glutamic acid	68	-0.0343
ala	Alanine	372	-0.0304
thr	Threonine	95	-0.0210
met	Methionine	23	-0.0205
aaa	Alpha-Aminoadipic acid	3	-0.0192
lys	Lysine	206	-0.0192
3mhis	3-Methylhistidine	5	-0.0181
pro	Proline	120	-0.0180
asn	Asparagine	60	-0.0100
cit	Citrulline	18	-0.0082
phe	Phenylalanine	56	-0.0048
leu	Leucine	169	0.0737
cys2	Cystine	75	0.0679
arg	Arginine	59	0.0457
gly	Glycine	187	0.0333
gln	Glutamine	583	0.0169
ser	Serine	95	0.0064
hyp	Hydroxyproline	6	0.0
asp	Aspartic Acid	4	0.0
val	Valine	244	0.0
abu	Aminobutyric acid	25	0.0

The feature "cysta"(Cystathionine) with a SHAP value of -0.2780, suggesting a strong negative influence meaning it contributed to classify the patient as not having CRC. This is followed by "tau"(Taurine) and "sex" with SHAP values of -0.2075 and -0.1319, respectively. While these features decrease the predicted probability, "leu"(Leucine) with a SHAP value of 0.0737 and "cys2" (Cystine) with 0.0679 increase it. The majority of the features exhibit small SHAP values, indicating their marginal influence on the model's prediction for this sample. Additionally, several features like "hyp"(Hydroxyproline), "asp"(Aspartic Acid), "val"(Valine), and "abu"(Aminobutyric acid) showed no influence with a SHAP value of 0.

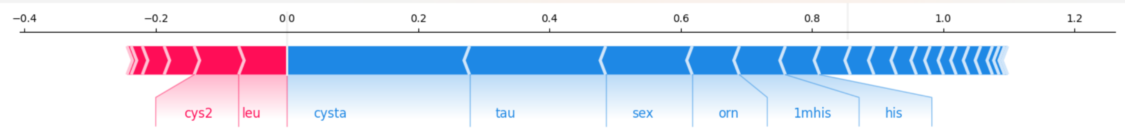

Figure 5.7: Shap Force Plot for Single Amino acids M1 Prediction Without Age

5.3 Scenario 3 - Acylcarnitines All Features

Following the same structure, this section will explore the results for Acylcarnitines using all features.

5.3.1 Model Selection

Normalization

KNeighbors showed a slight decrease in performance after normalization, with Mean F1 and Recall dropping by about 5.04% and 4.65% respectively. DecisionTree and SVC had minimal performance declines, both under 1%. GaussianNB improved by 2.59% in Mean F1 and 2.56% in Recall. The MLP classifier saw the most notable improvement with a 41.80% increase in Mean F1 and a 22.29% rise in Recall. Both RandomForest and GradientBoosting had performance dips, but RandomForest's drop was more significant at around 3.11% in Mean F1 and 3.17% in Recall. GradientBoosting had a near 1% decrease in both metrics.

Table 5.9: Comparison of Classifier Performance with and without Normalization for Acylcarnitines with All Features

ClassifierName	Mean F1-Recall Control	Mean F1-Recall Norm
KNeighbors	0.8324-0.8325	0.7904-0.7940 (-5.04% -4.65%)
DecisionTree	0.8324-0.8324	0.8284-0.8291 (-0.48% -0.39%)
GaussianNB	0.7129-0.7321	0.7314-0.7508 (+2.59% +2.56%)
SVC	0.8518-0.8522	0.8473-0.8468 (-0.54% -0.43%)
MLP	0.4757-0.5857	0.6746-0.7162 (+41.80% +22.29%)
RandomForest	0.9293-0.9277	0.9003-0.8984 (-3.11% -3.17%)
GradientBoosting	0.9342-0.9331	0.9249-0.9238 (-0.99% -1.00%)

Feature Selection

Significant improvements in F1 scores were observed, especially for the GaussianNB and MLP classifiers. The GaussianNB's F1 score saw a remarkable increase of 32.35%, while the MLP experienced an even more impressive surge of 89.68%. Most classifiers' recall scores, however, displayed slight variations with the DecisionTree and GradientBoosting classifiers showing the most significant drops. In terms of the most common percentile used for feature selection, the values ranged from 10 to 60.

ClassifierName	Mean F1-Recall Control	Mean F1-Recall FS	Best Percentile
KNeighbors	0.8324-0.8325	0.8453-0.8398 (+1.54% - +0.87%)	20
DecisionTree	0.8324-0.8324	0.8452-0.8077 (+1.53% - -2.97%)	20
GaussianNB	0.7129-0.7321	0.9435-0.7508 (+32.35% - +2.56%)	10
SVC	0.8518-0.8522	0.8920-0.8468 (+4.71% - -0.63%)	10
MLP	0.4757-0.5857	0.9025-0.6902 (+89.68% - +17.84%)	30
RandomForest	0.9293-0.9277	0.9384-0.9290 (+0.98% - +0.14%)	60
GradientBoosting	0.9342-0.9331	0.9530-0.9288 (+2.01% - -0.46%)	40

Table 5.10: Comparison of Classifier Performance with and without Feature Selection for Acylcarnitines with All Features

"c5:1"(Tiglylcarnitine/ 3Methylcrotonylcarnitine), "c10"(Decanoylcarnitine), "c6dc"(3-Methylglutarylcarnitine) and "sex" were the most removed features.

Hyperparameters and Processing Times

KNeighbors took 11s with the most common parameters being 3 neighbors and uniform weighting. The DecisionTree processed in 5s, favoring a maximum depth of 10. GaussianNB completed in 5s with the default var_smoothing parameter. SVC, with an RBF kernel and C-value of 10, took 8s. The MLP classifier, using an identity activation and 100 hidden layer sizes, took 17s. RandomForest, not setting a maximum depth and using 150 estimators, was the lengthiest at 3m:26s. Lastly, GradientBoosting finished in 53s, commonly employing 50 estimators.

ClassifierName	Processing Time	Most Common HyperParameters
KNeighbors	0m:11s	{’neighbors’: 11, ’weights’: ’distance’}
DecisionTree	0m:5s	{’max_depth’: 5}
GaussianNB	0m:5s	{’var_smoothing’: 1e09}
SVC	0m:8s	{’C’: 0.1, ’kernel’: ’linear’}
MLP	0m:17s	{’activation’: ’identity’, ’hidden_layer_sizes’: (100,)}
RandomForest	3m:26s	{’max_depth’: None, ’n_estimators’: 100}
GradientBoosting	0m:53s	{’n_estimators’: 50}

Table 5.11: Processing Times and Most Common Parameters with Feature Selection (FS) for Acylcarnitines without Age

Best Model

The best classifier was the **GradientBoosting** with a F1-Score 0.9530 and Recall 0.9288. Using Normalization and Feature Selection with percentile as 50 and n_estimators set as 50.

5.3.2 Explanations

Like the amino acids, the dominating feature is the "age". Details about the explanations at the subsections bellow. The process took about 5 seconds for the original dataset and about 1 second for the M1 samples using TreeExplainer, also similar to the amino acids.

Overall Feature Importance on the original dataset

The top four most influencing features for the original dataset were the "age", "c3" (Propionylcarnitine), "c14:2"(Tetradecadienoylcarnitine) and "c2" (Acetylcarnitine). Just like the amino acids, when the value of the age increases, the model is more likely to predict a sample as having CRC. The same happens for "c3". On the contrary,when the values of the feature "c14"(Tetradecenoylcarnitine) increased, the model was more likely to predict a sample as not having CRC. The "c14:2" as both low and high values pushing the predictions to a positive class, however, it had lower values when classifying a sample to the negative side (without CRC). The summary plot for the original dataset values can be seen at 5.8

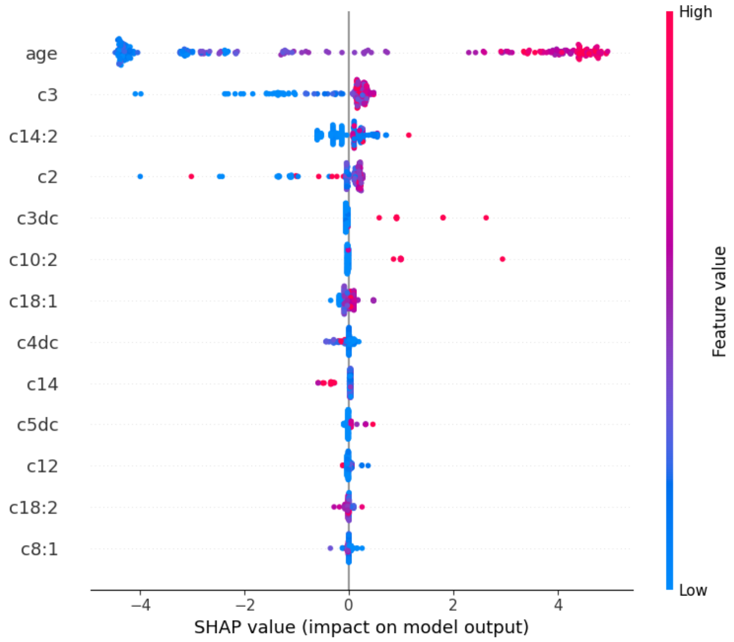

Figure 5.8: Shap Summary Plot for original acylcarnitines dataset with All Features

Overall Feature Importance on the M1 samples

There's no big difference in the features importances when applied to the unseen M1 samples, just like the amino acids. The age was still the most influential feature, however, as we can see at the summary plot 5.9, no matter the value of the age, it still pushed almost all instances towards the positive class (CRC detected), the same goes with the "c3".

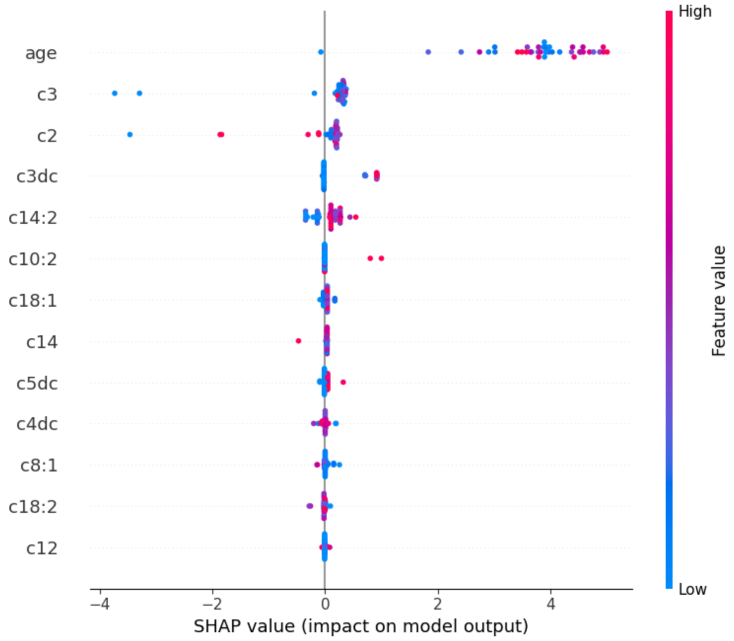

Figure 5.9: Shap Summary Plot for M1 samples acylcarnitines dataset with All Features

Single M1 instance

From the 48 M1 samples, the model correctly predicted 45 as having CRC. One of the wrong predictions is described at the table 5.12.

Table 5.12: Shap Values for Single Acylcarnitines M1 Wrong Prediction

Feature	Description	Value	SHAP Value
c3	Propionylcarnitine	0.00	-3.9058
c14:2	Tetradecadienoylcarnitine	0.00	-0.2111
c18:2	Octadecadienoylcarnitine	0.27	-0.1406
c5dc	Glutarylcarnitine	0.00	-0.1363
c10:2	Decadienoylcarnitine	0.00	-0.0901
c2	Acetylcarnitine	8.49	-0.0493
c3dc	Malonylcarnitine	0.00	-0.0204
c8:1	Octenoylcarnitine	0.00	0.0019
c14	Tetradecenoylcarnitine	0.00	0.0121
12	Dodecanoylcarnitine	0.00	0.0179
c4dc	Methylmalonylcarnitine/Succinylcarnitine	0.00	0.0234
c18:1	Octadecenoylcarnitine	1.06	0.0391
age	Age of the Patient	56	2.5056

Table 5.13: Ordered Features by SHAP values

With a shap values of -3.9058 and, for the "c3" (Propionylcarnitine) and "c14:2" (Tetradecadienoylcarnitine) respectively, were the features that most contributed to the wrong prediction. Curiously, these sample values were both 0. The "age" had the highest positive influence with a value of 2.5056, meaning that feature was the one that "tried" the most to predict this sample has having CRC.

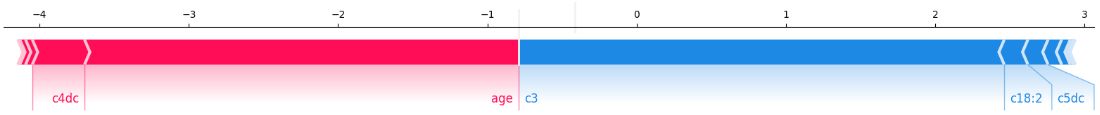

Figure 5.10: Shap Force Plot for Single Acylcarnitines M1 Prediction All Features

5.4 Scenario 4 - Acylcarnitines Without Age

This section will explore the results for Acylcarnitines using all features except the age. The structure is very similar to the previous one.

5.4.1 Model Selection

Normalization

A significant drop in all calssifiers performances can be seen upon removid the "Age" feature. KNeighbors, SVC, and RandomForest observed improvements in both F1 score and Recall upon normalization, with KNeighbors achieving the most significant boost of +3.39% in Recall. On the contrary, DecisionTree and MLP experienced slight performance drops after normalization. Notably, GaussianNB remained unaffected, maintaining the same performance metrics in both scenarios. GradientBoosting's performance slightly decreased post-normalization, with a drop of about 0.53% in Recall.

Table 5.14: Comparison of Classifier Performance with and without Normalization for Acylcarnitines without Age

ClassifierName	Mean F1-Recall Control	Mean F1-Recall Norm
KNeighbors	0.6566-0.6702	0.6744-0.6940 (+2.72% +3.39%)
DecisionTree	0.5870-0.5895	0.5601-0.5625 (-4.59% -4.58%)
GaussianNB	0.5593-0.6055	0.5724-0.6102 (0.00% 0.00%)
SVC	0.6040-0.6107	0.6122-0.6290 (+1.35% +3.00%)
MLP	0.5174-0.5600	0.5139-0.5560 (-0.68% -0.72%)
RandomForest	0.6786-0.6999	0.6975-0.7164 (+2.78% +2.35%)
GradientBoosting	0.6394-0.6596	0.6364-0.6561 (-0.47% -0.53%)

Feature Selection

Like the amino acids, the removal of the "Age" feature made all classifiers using more features. KNeighbors exhibited a 12.38% improvement in F1 score, while GaussianNB showed a significant 39.07% boost. SVC and DecisionTree also benefitted, with respective F1 enhancements of 16.16% and 16.47%. MLP had a substantial uptick of 38.43%. Conversely, RandomForest and GradientBoosting had modest improvements, with 7.21% and 13.13% respectively. The majority of classifiers, including KNeighbors, DecisionTree, and GaussianNB, achieved optimal performance at higher feature selection percentiles such as 80% to 100%.

ClassifierName	Mean F1-Recall Control	Mean F1-Recall FS	Difference (F1-Recall)	Most Common Percentile
KNeighbors	0.6566-0.6702	0.7378-0.6940	+12.38% - +3.57%	100
DecisionTree	0.5870-0.5895	0.6838-0.6233	+16.47% - +5.73%	60
GaussianNB	0.5593-0.6055	0.7780-0.6055	+39.07% - 0.00%	80
SVC	0.6040-0.6107	0.7016-0.6290	+16.16% - +3.00%	60
MLP	0.5174-0.5600	0.7163-0.6150	+38.43% - +9.82%	50
RandomForest	0.6786-0.6999	0.7275-0.6842	+7.21% - -2.25%	50
GradientBoosting	0.6394-0.6596	0.7234-0.6452	+13.13% - -2.18%	80

Table 5.15: Comparison of Classifier Performance with and without Feature Selection for Acylcarnitines without Age

The most common removed features were the "c10:1"(Decenoylcarnitine), the "c12:1"(Dodecenoylcarnitine) and the "c6dc"(3Methylglutarylcarnitine) which was also removed in the All Features dataset.

Hyperparameters and Processing Times

The KNeighbors classifier, with parameters indicating 3 neighbors and uniform weighting, took about 10 seconds, while DecisionTree and GaussianNB each finished in approximately 5 seconds, with the former favoring a maximum depth of 10. The SVC classifier, leveraging an RBF kernel and a C-value of 10, required 8 seconds. The MLP classifier, which utilized an identity activation function and a hidden layer size of 100, took 15 seconds. RandomForest, without a defined maximum depth and using 150 estimators, took the longest at 3 minutes and 12 seconds. Meanwhile, GradientBoosting concluded in 51 seconds, typically employing 50 estimators.

ClassifierName	Processing Time	Most Common HyperParameters
KNeighbors	0m:10s	{'neighbors': 3, 'weights': 'uniform'}
DecisionTree	0m:5s	{'max_depth': 10}
GaussianNB	0m:5s	{'var_smoothing': 1e-09}
SVC	0m:8s	{'C': 10, 'kernel': 'rbf'}
MLP	0m:15s	{'activation': 'identity', 'hidden_layer_sizes': (100,)}
RandomForest	3m:12s	{'max_depth': None, 'n_estimators': 150}
GradientBoosting	0m:51s	{'n_estimators': 50}

Table 5.16: Processing Times and Most Common Parameters with Feature Selection (FS) for Acylcarnitines without Age

Best Model

The best classifier was the **GaussianNB** with a F1-Score 0.7780 and Recall 0.6055. Using Normalization and Feature Selection with percentile at 80 and the default value for var_smoothing. This came with a surprise since data doesn't seem to follow a normal distribution as stated at section 3.3.

5.4.2 Explanations

Like the amino acids, by removing the feature "age", the features importances also changed. Details about the explanations at the subsections bellow. The process took about 7 minutes seconds for the original dataset and about 1 minute for the M1 samples using KernelExplainer, also similar to the amino acids.

Overall Feature Importance on the original dataset

The top four most influencing features for the original dataset were the "c8:1"(Octenoylcarnitine) and "c4dc"(Methylmalonylcarnitine/Succinylcarnitine) , "c14"(Tetradecenoylcarnitine) and "c12" (Dodecanoylcarnitine). All of them increased the probability of diagnosing a patient with CRC the lower the values were. Just like the amino acids, upon the removal of the feature "age", the model's explanations changed. The summary plot for the original dataset values can be seen at 5.11

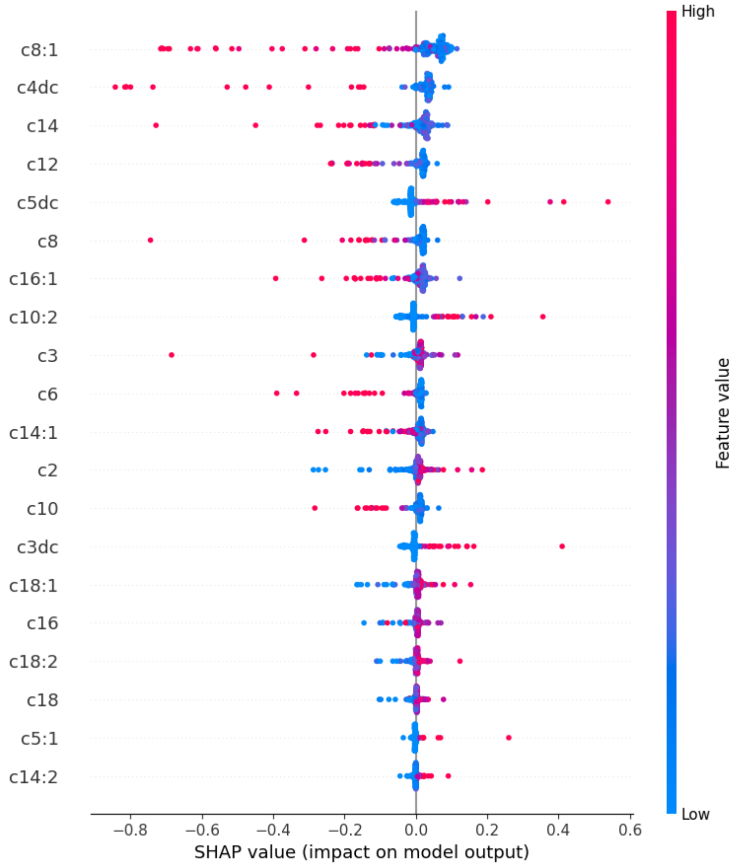

Figure 5.11: Shap Summary Plot for original acylcarnitines dataset Without Age

Overall Feature Importance on the M1 samples

This time the features importances changed for M1 samples, having the "c3dc" (Malonylcarnitine), "c6" (Hexanoylcarnitine) replacing the "c8:1"and "c4dc" as the most influencing features. More details can be seen at the summary plot 5.12.

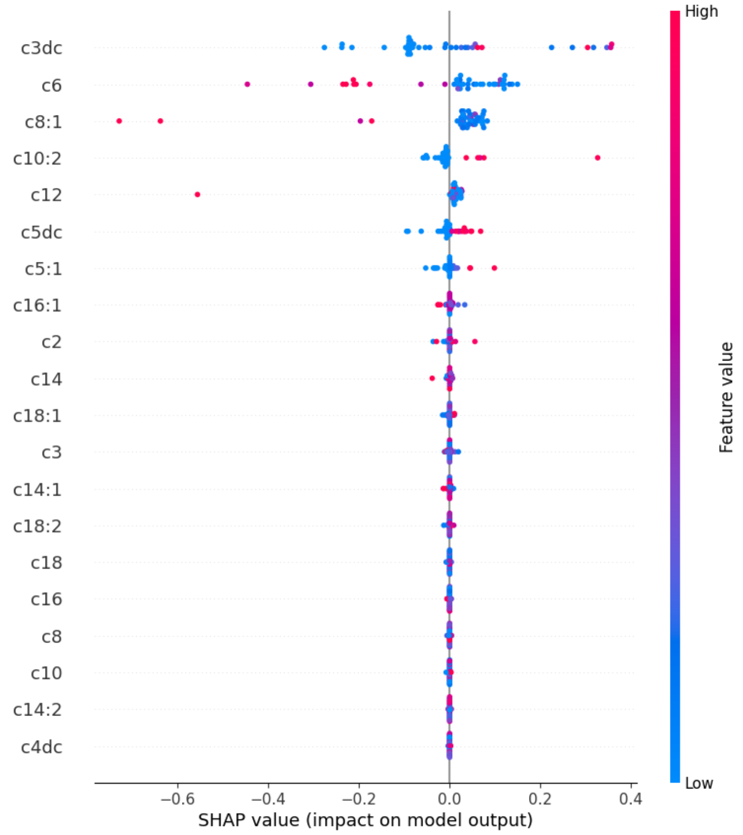

Figure 5.12: Shap Summary Plot for M1 samples Without Age

Single M1 instance

From the 48 M1 samples, only 42 were correctly predicted as having CRC. One of the wrong predictions is described at the table 5.17

Table 5.17: Shap Values for Single Acylcarnitines M1 Wrong Prediction Without Age

Feature	Description	Value	SHAP Value
c8:1	Octenoylcarnitine	0.766	-0.6339
c3dc	Malonylcarnitine	0	-0.2172
c10:2	Decadienoylcarnitine	0	-0.0608
c5:1	Tiglylcarnitine/ 3-Methylcrotonylcarnitine	0	-0.0357
c16:1	Hexadecenoylcarnitine	0.084	-0.0074
c14	Tetradecenoylcarnitine	0	-0.0063
c5dc	Glutarylcarnitine	0	0.0060
c16	Hexadecanoylcarnitine	0.544	0.0063
c18	Octadecanoylcarnitine	0.39	0.0033
c12	Dodecanoylcarnitine	0	0.0128
c6	Hexanoylcarnitine	0.029	0.0580
c14:2	Tetradecadienoylcarnitine	0	0.0
c4dc	Methylmalonylcarnitine/Succinylcarnitine	0.518	0.0
c3	Propionylcarnitine	1.28	0.0
c10	Decanoylcarnitine	0.106	0.0
c18:2	Octadecadienoylcarnitine	0.31	0.0
c18:1	Octadecenoylcarnitine	0.83	0.0
c2	Acetylcarnitine	17.29	0.0
c14:1	Tetradecenoylcarnitine	0.262	0.0
c8	Octanoylcarnitine	0.072	0.0

The features "c8:1"(Octenoylcarnitine) and "c3dc"(Malonylcarnitine) dominated the negative prediction (With CRC) with shaps values of -0.6339 and -0.2172 respectively. The same instance tested with all features at section 5.3.2 was also tested with this model. The model also failed to predict this sample but this time pointed two other features as responsible for the wrong prediction. The features were the "c12" (Dodecanoylcarnitine) and the "c3dc" in contrast with the feature "c3"(Propionylcarnitine) that had the most importance on the model with all features but almost none in this model.

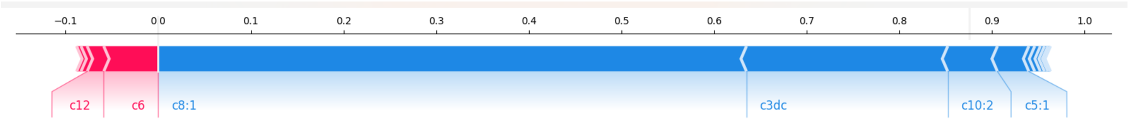

Figure 5.13: Shap Force Plot for Single Acylcarnitines M1 Prediction Without Age

5.5 Final Remarks

In this section we described the evaluation step CRISP methodology where the performances of the different algorithms were compared along side with the explanations. Next is the deploy.

Chapter 6

Deploy

In this chapter, the crucial step of turning the ML models with their respective explanations, into real, working solutions, will be explored. Starting with the definition of the functional requirements followed by the system architecture.

6.1 Functional Requirements

The system will be primarily used by healthcare staff. They should be able to make predictions and obtain explanations for those predictions. They should also be capable of re-training the models whenever new data is available.

The diagram 6.1 presents a visual interpretation of the use cases.

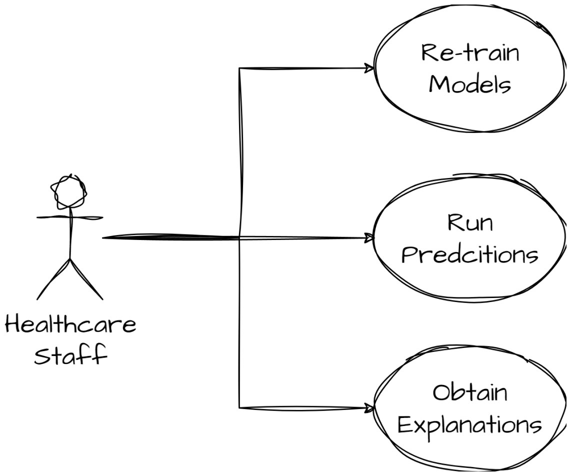

Figure 6.1: Use Case Diagram

6.2 System Architecture

The system architecture will follow a typical Decision Support System (DSS) architecture.

A DSS is a computerized system used to support decision-making in an organization. It assists the users in making informed decisions by providing access to information and tools to analyze that information. At its core, it comprises a Database Management System (DBMS) that stores and manages vital raw data, a Model Management System (MMS) which houses decision models to process and analyze data, an User Interface System that facilitates user-system interaction and ensures efficient tool and information access and a Data Processing Subsystem which processes the data, either by preparing it for analysis (like cleaning or transforming the data) or by conducting the analysis itself. In some systems, a Knowledge Base that contains expert knowledge and domain-specific insights that can help in the decision-making process. Together, these components synchronize to offer users analytical tools and information, enabling informed decision-making across several organizational challenges [89].

The following sub section provides an adapted system architecture for this work.

Input Component

- **Data Sources**:
  - Pulls data from multiple sources.
  - Includes databases, APIs, and user input interfaces.
- **Data Preprocessing Modules**:
  - Clean, preprocess, and transforms data.
  - Ensure data is ready for training models.

Model Component

- **Model Repository**:
  - Houses all models and associated metadata.
  - Includes versioning, training date, and performance metrics.
- **Model Server**:
  - Manages model training and deployment.
  - Uses the tools scikit-learn and joblib.

Explanations Component

- **SHAP Calculation Module**:
  - Calculates SHAP values after model predictions.
- **Explanation Database**:
  - Stores SHAP explanations.
  - Useful for historical tracking and audit trails.
- **Explanation Visualization Tool**:
  - Visualizes SHAP values.
  - Offers visual explanations like force plots, summary plots.

Output Component

- **API**:
  - Serves as the bridge between the system and external users or systems.
  - Enables system consumption by other platforms.
- **Decision Interface**:
  - Showcases model predictions and SHAP explanations.
  - Designed for end-users with clarity and user-friendliness in mind.

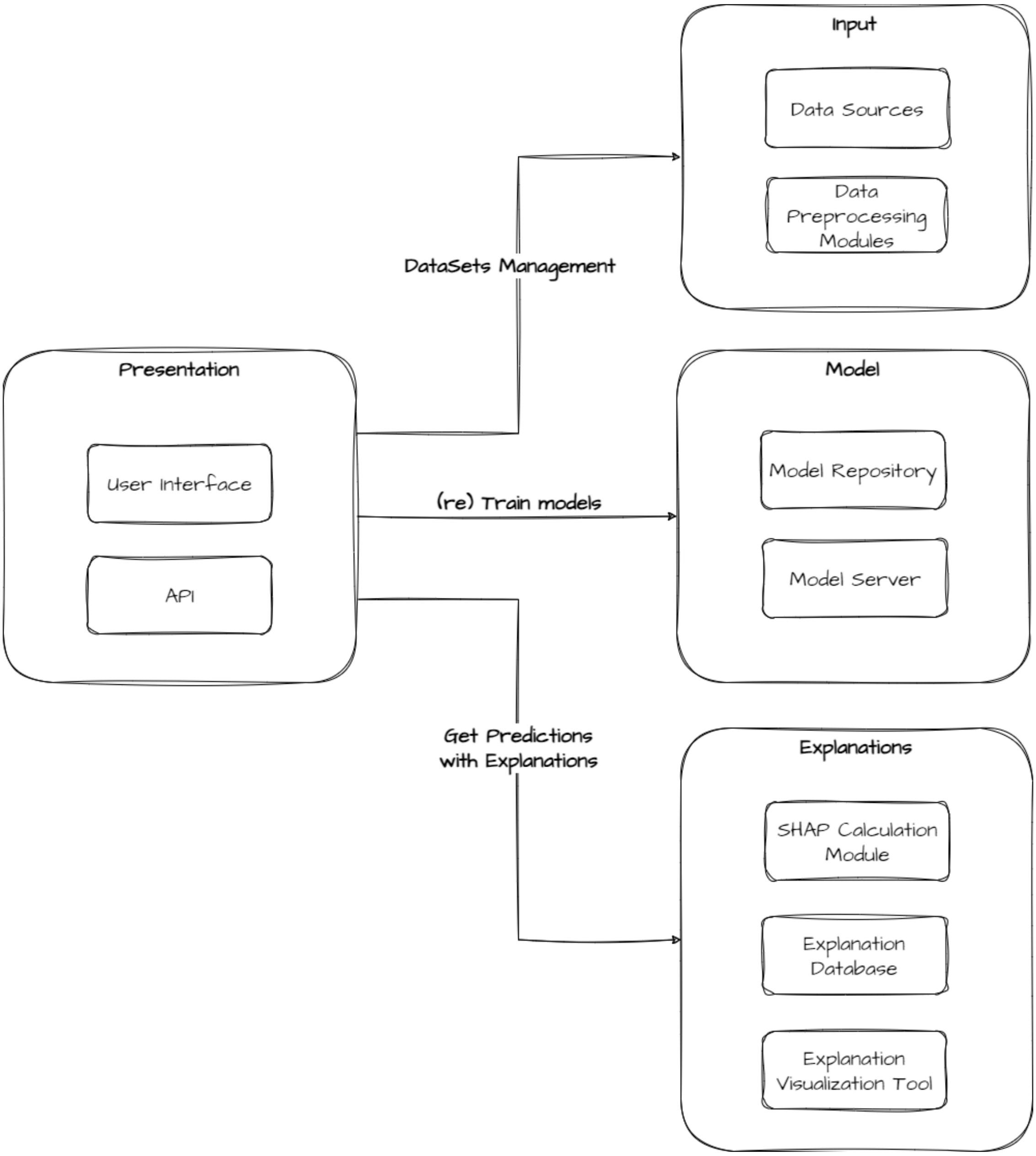

Figure 6.2: System Architecture

6.3 Legal & Ethics

If a bad decision, supported by a DSS is made, who is going to be accounted ? The medical personnel, the system or both ?

There isn't a single universally applicable answer, as responsibility can vary based on jurisdiction, specific circumstances, and the nuances of local laws and regulations. However, there are several relevant frameworks and principles to consider:

- Medical Device Regulation (MDR): If a DSS is classified as a medical device under EU regulations, the MDR will apply. While the MDR sets standards for the safety and efficacy of devices, it doesn't provide a straightforward answer to the allocation of responsibility between healthcare providers and the system. However, manufacturers of medical devices have specific obligations related to post-market surveillance, risk management, and overall system safety [90].
- Proposed AI Regulations (2021): The European Commission's proposal for AI regulations classifies AI systems based on risk. High-risk systems, which include certain applications in healthcare, have stringent requirements, but the proposal mainly emphasizes transparency, documentation, and system quality rather than directly addressing the liability question [91].

For this reason, this project will be deployed only for academic/research purposes.

6.4 Final Remarks

In this section we described the deploy step CRISP methodology. In the next chapter the conclusions will be presented.

Chapter 7

Conclusions

7.1 General Considerations

The main objective of this work is to investigate if the combined metabolic profiles, amino acids and acylcarnitinescould constitute a biochemical marker for the prediction of CRC, by applying ML algorithms and techniques and also to provide explanations.

For that, it was necessary to compile and treat multiple sources of data, transforming it in a way suitable to be used by ML algorithms. Unfortunately, for the combined metabolic profiles, the data was only available for patients with CRC. For a classification problem, both patients with and without CRC are needed. Therefore, the patients without CRC were obtained by the previous works, [23] and [24].

With all the information compiled, we evaluate several data preprocessing techniques and ML algorithms, to test their effectiveness in the task of classifying patients with CRC.

The constructed ML models successfully identified the majority of the CRC cases when tested on two distinct datasets comprising just over 200 samples each. However the models performances substantially decreased when the feature "age" was removed.

The explanations were applied to two different datasets, the original training one and another containing some patients in a later stage of the disease. They were also applied to instances that were wrongly predicted by the models. Overall, they provided a good insight on which features were most relevant upon predicting CRC. As previously suspected, the feature "age" dominated the explanations, however this changed with its removal.

7.2 Results

This section summarizes the results.

7.2.1 Amino acids prediction with age

The best model was obtained by the RandomForests classifier with a F1-Score of approximately 90%. When applying the explanations, the most influencing feature was "age", followed by "cys2" (Cystine), "his"(Histidine) and "phe" (Phenylalanine).

7.2.2 Amino acids prediction without age

The best model was provided by the Support Vector Machine classifier using the RBF kernel with a F1-Score about 83%. This time, the explanations pointed out "cys2" (Cystine), "his"(Histidine) and "cysta" (Cystathionine) for the M0 cases. For the M1 this changed to "leu" (Leucine), "cys2", "sex" and "hys" (Histidine). On this dataset, the females were more likely to have CRC than males.

7.2.3 Acylcarnitines prediction with age

GradientBoosting with a F1-Score around 95% was the best model. Just like the aminoacids, the "age" played a predominant role in the predictions. Followed by the "c3" (Propionylcarnitine), "c14:2"(Tetradecadienoylcarnitine) and "c2" (Acetylcarnitine).

7.2.4 Acylcarnitines prediction without age

The best model was obtained by the Gaussian Naive Bayes with a F1-Score close to 77%. The most important features were the "c8:1"(Octenoylcarnitine), "c4dc"(Methylmalonylcarnitine/Succinylcarnitine) , "c14"(Tetradecenoylcarnitine) and "c12" (Dodecanoylcarnitine). When applied the explanations for the M1 cases, the features importances also changed having the "c3dc" (Malonylcarnitine) and "c6" (Hexanoylcarnitine) as the most influencing ones. The "sex" feature was not considered in these models since it was automatically removed by the feature selection method.

7.3 Future Work

To continue the work developed so far, we plan to have more interactions with the healthcare staff. Some topics for discussion can be considered:

- Collect information from more patients;
- Get samples for patients without CRC with the combined metabolic profiles;
- Test different normalization and feature selection methods;
- Further tune the classifiers hyper parameters;
- Test other explanation tools;
- Deploy the system in appropriate infrastructure so it can be accessed by CHUP and ISEP;

Bibliography

[1] Abhishek Bhandari, Melissa Woodhouse, and Samir Gupta. “Colorectal cancer is a leading cause of cancer incidence and mortality among adults younger than 50 years in the USA: a SEER-based analysis with comparison to other young-onset cancers”. In: *Journal of Investigative Medicine* 65.2 (2017), pp. 311–315.

[2] Nicolas Huyghe, Pamela Baldin, and Marc Van den Eynde. “Immunotherapy with immune checkpoint inhibitors in colorectal cancer: what is the future beyond deficient mismatch-repair tumours?” In: *Gastroenterology report* 8.1 (2020), pp. 11–24.

[3] Emmanuelle Kesse, Françoise Clavel-Chapelon, and Marie-Christine Boutron-Ruault. “Dietary patterns and risk of colorectal tumors: a cohort of French women of the National Education System (E3N)”. In: *American journal of epidemiology* 164.11 (2006), pp. 1085–1093.

[4] Jeffrey S. Saltz and Nicholas Hotz. “Identifying the most Common Frameworks Data Science Teams Use to Structure and Coordinate their Projects”. In: (2020), pp. 2038–2042. doi: 10.1109/BigData50022.2020.9377813.

[5] Veronika Plotnikova, Marlon Dumas, and Fredrik Milani. “Adaptations of data mining methodologies: A systematic literature review”. In: *PeerJ Computer Science* 6 (2020), e267.

[6] Douglas Hanahan and Robert A Weinberg. “The hallmarks of cancer”. In: *cell* 100.1 (2000), pp. 57–70.

[7] Minjoung Monica Koo et al. “Presenting symptoms of cancer and stage at diagnosis: evidence from a cross-sectional, population-based study”. In: *The Lancet Oncology* 21.1 (2020), pp. 73–79.

[8] Sidney J Winawer. “Colorectal cancer screening”. In: *Best practice & research Clinical gastroenterology* 21.6 (2007), pp. 1031–1048.

[9] Md Sanower Hossain et al. “Colorectal cancer: a review of carcinogenesis, global epidemiology, current challenges, risk factors, preventive and treatment strategies”. In: *Cancers* 14.7 (2022), p. 1732.

[10] Freddie Bray et al. “Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries”. In: *CA: a cancer journal for clinicians* 68.6 (2018), pp. 394–424.

[11] Yumo Xie et al. “Gastrointestinal cancers in China, the USA, and Europe”. In: *Gastroenterology Report* 9.2 (Mar. 2021), pp. 91–104. issn: 2052-0034. doi: 10.1093/gastro/goab010. eprint: https://academic.oup.com/gastro/article-pdf/9/2/91/37944287/goab010.pdf. url: https://doi.org/10.1093/gastro/goab010.

[12] Rajesh Sharma et al. “Global, regional, and national burden of colorectal cancer and its risk factors, 1990–2019: a systematic analysis for the Global Burden of Disease Study 2019”. In: *The Lancet Gastroenterology & Hepatology* 7.7 (2022), pp. 627–647.

[13] Tomasz Sawicki et al. "A Review of Colorectal Cancer in Terms of Epidemiology, Risk Factors, Development, Symptoms and Diagnosis". In: *Cancers* 13.9 (2021). issn: 2072-6694. doi: `10.3390/cancers13092025`. url: `https://www.mdpi.com/2072-6694/13/9/2025`.

[14] Fiona M Walter et al. "Symptoms and patient factors associated with longer time to diagnosis for colorectal cancer: results from a prospective cohort study". In: *British journal of cancer* 115.5 (2016), pp. 533–541.

[15] Huaqin Lin et al. "Meta-analysis of neoadjuvant chemotherapy versus neoadjuvant chemoradiotherapy for locally advanced rectal cancer". In: *World Journal of Surgical Oncology* 19.1 (2021), pp. 1–10.

[16] Fumito Ito and Alfred E Chang. "Cancer immunotherapy: current status and future directions". In: *Surgical Oncology Clinics* 22.4 (2013), pp. 765–783.

[17] Yuan-Hong Xie, Ying-Xuan Chen, and Jing-Yuan Fang. "Comprehensive review of targeted therapy for colorectal cancer". In: *Signal transduction and targeted therapy* 5.1 (2020), p. 22.

[18] Robert A Smith et al. "Cancer screening in the United States, 2019: A review of current American Cancer Society guidelines and current issues in cancer screening". In: *CA: a cancer journal for clinicians* 69.3 (2019), pp. 184–210.

[19] Douglas K. Rex et al. "Colorectal Cancer Screening: Recommendations for Physicians and Patients From the U.S. Multi-Society Task Force on Colorectal Cancer". In: *Gastroenterology* 153.1 (2017), pp. 307–323. issn: 0016-5085. doi: `https://doi.org/10.1053/j.gastro.2017.05.013`. url: `https://www.sciencedirect.com/science/article/pii/S0016508517355993`.

[20] Christopher J Clarke and John N Haselden. "Metabolic profiling as a tool for understanding mechanisms of toxicity". In: *Toxicologic Pathology* 36.1 (2008), pp. 140–147.

[21] Minnie Jacob et al. "A targeted metabolomics approach for clinical diagnosis of inborn errors of metabolism". In: *Analytica chimica acta* 1025 (2018), pp. 141–153.

[22] Michael J Lopez and Shamim S Mohiuddin. "Biochemistry, essential amino acids". In: (2020).

[23] João Gonçalves. "Previsão Inteligente das alterações metabólicas no cancro retal com base em modelos de machine e deep learning". In: (2021).

[24] Pedro Manuel Nogueira Lopes. "Machine Learning na previsão de Cancro Colorretal em função de alterações metabólicas". In: (2022).

[25] Shaghayegh Hosseinkhani et al. "Targeted metabolomics analysis of amino acids and acylcarnitines as risk markers for diabetes by LC–MS/MS technique". In: *Scientific Reports* 12.1 (2022), pp. 1–11.

[26] Joseph A Rothwell et al. "Circulating amino acid levels and colorectal cancer risk in the European Prospective Investigation into Cancer and Nutrition and UK Biobank cohorts". In: *BMC medicine* 21.1 (2023), pp. 1–13.

[27] Shuo Zhao et al. "The association between acylcarnitine metabolites and cardiovascular disease in Chinese patients with type 2 diabetes mellitus". In: *Frontiers in endocrinology* 11 (2020), p. 212.

[28] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. "Supervised Learning". In: *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*. Ed. by Matthieu Cord and Pádraig Cunningham. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 21–49. isbn: 978-3-540-75171-7. doi: 10.1007/978-3-540-75171-7_2. url: https://doi.org/10.1007/978-3-540-75171-7_2.

[29] João Gama et al. *Extração de Conhecimento de Dados - Data Mining - 3ª Edição*. Edições Sílabo, 2017.

[30] Joshua Ebner. *Regression vs classification, explained*. July 2021. url: https://www.sharpsightlabs.com/blog/regression-vs-classification/.

[31] Rajvi Shah. *Introduction to K-nearest neighbors (knn) algorithm*. Mar. 2021. url: https://ai.plainenglish.io/introduction-to-k-nearest-neighbors-knn-algorithm-e8617a448fa8.

[32] Irina Rish et al. "An empirical study of the naive Bayes classifier". In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22. 2001, pp. 41–46.

[33] Sayali D Jadhav and HP Channe. "Comparative study of K-NN, naive Bayes and decision tree classification techniques". In: *International Journal of Science and Research (IJSR)* 5.1 (2016), pp. 1842–1845.

[34] Jia Wu, Zhihua Cai, and Xingquan Zhu. "Self-adaptive probability estimation for naive bayes classification". In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2013, pp. 1–8.

[35] J. Ross Quinlan. "Induction of decision trees". In: *Machine learning* 1.1 (1986), pp. 81–106.

[36] Sotiris B Kotsiantis. "Decision trees: A recent overview". In: *Artificial Intelligence Review* 39.4 (2013), pp. 261–283.

[37] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. Elsevier, 2011.

[38] Tan Pang-Ning, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Addison-Wesley, 2006.

[39] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32. doi: 10.1023/A:1010933404324.

[40] Chao Chen, Andy Liaw, and Leo Breiman. "Using random forest to learn imbalanced data". In: *University of California, Berkeley, Tech. Rep* 666 (2004), pp. 1–12.

[41] Andy Liaw and Matthew Wiener. "Classification and regression by randomForest". In: *R news* 2.3 (2002), pp. 18–22.

[42] Carolin Strobl et al. "An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests". In: *Psychological methods* 14.4 (2009), p. 323.

[43] Alexey Natekin and Alois Knoll. "Gradient boosting machines, a tutorial". In: *Frontiers in neurorobotics* 7 (2013), p. 21.

[44] Siddharth Yadav. *Gradient Boosted Decision Trees Explained*. 2021. url: https://towardsdatascience.com/gradient-boosted-decision-trees-explained-9259bd8205af.

[45] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[46] Batta Mahesh. "Machine learning algorithms-a review". In: *International Journal of Science and Research (IJSR).[Internet]* 9.1 (2020), pp. 381–386.

[47] Chih-Wei Hsu and Chih-Jen Lin. "A practical guide to support vector classification". In: *Department of Computer Science, National Taiwan University* 2 (2003), pp. 1–16.

[48] Scott Lundberg. *Welcome to the shap documentation*. 2018. url: `https://shap.readthedocs.io/en/latest/`.

[49] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[50] Shahadat Uddin et al. "Comparing different supervised machine learning algorithms for disease prediction". In: *BMC medical informatics and decision making* 19.1 (2019), pp. 1–16.

[51] Riccardo Miotto et al. "Deep learning for healthcare: review, opportunities and challenges". In: *Briefings in bioinformatics* 19.6 (2018), pp. 1236–1246.

[52] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828.

[53] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444.

[54] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature Machine Intelligence* 1.5 (2019), pp. 206–215.

[55] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[56] Waddah Saeed and Christian Omlin. "Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities". In: *Knowledge-Based Systems* 263 (2023), p. 110273.

[57] Prashant Gohel, Priyanka Singh, and Manoranjan Mohanty. "Explainable AI: current status and future directions". In: *arXiv preprint arXiv:2107.07045* (2021).

[58] Michael Van Lent, William Fisher, and Michael Mancuso. "An explainable artificial intelligence system for small-unit tactical behavior". In: *Proceedings of the national conference on artificial intelligence*. Citeseer. 2004, pp. 900–907.

[59] Amina Adadi and Mohammed Berrada. "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)". In: *IEEE access* 6 (2018), pp. 52138–52160.

[60] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (2017).

[61] Lloyd S Shapley et al. "A value for n-person games". In: (1953).

[62] Alvin E Roth. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[63] Scott Lundberg et al. *SHAP: A unified approach to explain the output of any machine learning model*. Accessed: 2023-08. 2018. url: `https://shap-lrjball.readthedocs.io/en/latest/api.html#plots`.

[64] Konstantina Kourou et al. "Machine learning applications in cancer prognosis and prediction". In: *Computational and structural biotechnology journal* 13 (2015), pp. 8–17.

[65] T. Ayer et al. "Breast cancer risk estimation with artificial neural networks revisited: Discrimination and calibration". In: *Cancer* 116.14 (July 2010), pp. 3310–3321. doi: `10.1002/cncr.25081`.

[66] M. Waddell, D. Page, and John Shaughnessy. "Predicting Cancer Susceptibility from Single-Nucleotide Polymorphism Data: A Case Study in Multiple Myeloma". In: *[ACM]*. 2005.

[67] J. Listgarten et al. *Predictive Models for Breast Cancer Susceptibility from Multiple Single Nucleotide Polymorphisms*. 2004. url: `http://www.polyomx.org/`.

[68] Alexander Stojadinovic et al. "Development of a Bayesian Belief Network Model for personalized prognostic risk assessment in colon carcinomatosis". In: *The American Surgeon* 77.2 (2011), pp. 221–230.

[69] Hisham Hussan et al. "Utility of machine learning in developing a predictive model for early-age-onset colorectal neoplasia using electronic health records". In: *Plos one* 17.3 (2022), e0265209.

[70] Abdellatif Bakkali et al. "Integration of stool microbiota, proteome and amino acid profiles to discriminate patients with adenomas and colorectal cancer". In: (2022).

[71] Zugang Yin et al. "Application of artificial intelligence in diagnosis and treatment of colorectal cancer: A novel Prospect". In: *Frontiers in Medicine* 10 (2023), p. 1128084.

[72] Peshawa Jamal Muhammad Ali et al. "Data normalization and standardization: a technical report". In: *Mach Learn Tech Rep* 1.1 (2014), pp. 1–6.

[73] *Compare the effect of different scalers on data with outliers*. Accessed: August 2023. url: `https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html`.

[74] K Senthamarai Kannan, K Manoj, and S Arumugam. "Labeling methods for identifying outliers". In: *International Journal of Statistics and Systems* 10.2 (2015), pp. 231–238.

[75] Nornadiah Mohd Razali, Yap Bee Wah, et al. "Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests". In: *Journal of statistical modeling and analytics* 2.1 (2011), pp. 21–33.

[76] Jiliang Tang, Salem Alelyani, and Huan Liu. "Feature selection for classification: A review". In: *Data classification: Algorithms and applications* (2014), p. 37.

[77] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. "A review of feature selection techniques in bioinformatics". In: *Bioinformatics* 23.19 (Aug. 2007), pp. 2507–2517. issn: 1367-4803. doi: `10.1093/bioinformatics/btm344`. eprint: `https://academic.oup.com/bioinformatics/article-pdf/23/19/2507/49857541/bioinformatics\_23\_19\_2507.pdf`. url: `https://doi.org/10.1093/bioinformatics/btm344`.

[78] Amina Benkessirat and Nadjia Benblidia. "Fundamentals of Feature Selection: An Overview and Comparison". In: *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*. 2019, pp. 1–6. doi: `10.1109/AICCSA47632.2019.9035281`.

[79] *sklearn.modules.permutation_ importance*. Accessed: 2023-08. url: https://scikit-learn.org/stable/modules/permutation_importance.html.

[80] *sklearn.feature_ selection*. Accessed: 2023-08. url: https://scikit-learn.org/stable/modules/feature_selection.html.

[81] *sklearn.feature_ selection.SelectPercentile*. Accessed: 2023-08. url: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectPercentile.html#sklearn.feature_selection.SelectPercentile.

[82] Yoram Reich and SV Barai. “Evaluating machine learning models for engineering problems”. In: *Artificial Intelligence in Engineering* 13.3 (1999), pp. 257–272.

[83] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[84] scikit-learn. *3.1. Cross-validation: evaluating estimator performance*. Version 1.1.2. Accessed: Sep.2023. 2022. url: https://scikit-learn.org/stable/modules/cross_validation.html.

[85] Xue Ying. “An overview of overfitting and its solutions”. In: *Journal of physics: Conference series*. Vol. 1168. IOP Publishing. 2019, p. 022022.

[86] *sklearn.model_ selection.StratifiedKFold*. Accessed: 2023-08. url: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html.

[87] Louis Owen. 2022.

[88] *sklearn.model*$_{s}election.GridSearchCV$. Accessed: 2023-08. url: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.

[89] Daniel J Power. *Decision support systems: concepts and resources for managers*. Quorum Books, 2002.

[90] R Beckers, Z Kwade, and F Zanca. “The EU medical device regulation: Implications for artificial intelligence-based medical device software in medical physics”. In: *Physica Medica* 83 (2021), pp. 1–8.

[91] Jakob Mökander et al. “Conformity assessments and post-market monitoring: a guide to the role of auditing in the proposed European AI regulation”. In: *Minds and Machines* 32.2 (2022), pp. 241–268.