INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA INFORMÁTICA

Machine Learning na previsão de Cancro Colorretal em função de alterações metabólicas

PEDRO MANUEL NOGUEIRA LOPES

Outubro de 2022

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Machine Learning na previsão de Cancro Colorretal em função de alterações metabólicas

Pedro Manuel Nogueira Lopes

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Engenharia de Software

Orientador: Professor Doutor José Reis Tavares
Coorientador: Professora Doutora Isabel Praça
Supervisora: Professora Doutora Marisa Santos

Porto, outubro 2022

Resumo

No mundo atual, a quantidade de informação disponível nos mais variados setores é cada vez maior. É o caso da área da saúde, onde a recolha e tratamento de dados biomédicos procuram melhorar a tomada de decisão no tratamento a aplicar a um doente, recorrendo a ferramentas baseadas em *Machine Learning*.

Machine Learning é uma área da Inteligência Artificial em que através da aplicação de algoritmos a um conjunto de dados é possível prever resultados ou até descobrir relações entre estes que seriam impercetíveis à primeira vista.

Com este projeto pretende-se realizar um estudo em que o objetivo é investigar diversos algoritmos e técnicas de *Machine Learning,* de modo a identificar se o perfil de acilcarnitinas pode constituir um novo marcador bioquímico para a predição e prognóstico do Cancro Colorretal.

No decurso do trabalho, foram testados diferentes algoritmos e técnicas de pré-processamento de dados. Foram realizadas três experiências distintas com o objetivo de validar as previsões dos modelos construídos para diferentes cenários, nomeadamente: prever se o paciente tem Cancro Colorretal, prever qual a doença que o paciente tem (Cancro Colorretal e outras doenças metabólicas) e prever se este tem ou não alguma doença. Numa primeira análise, os modelos desenvolvidos apresentam bons resultados na triagem de Cancro Colorretal.

Os melhores resultados foram obtidos pelos algoritmos *Random Forest* e *Gradient Boosting*, em conjunto com técnicas de balanceamento dos dados e *Feature Selection*, nomeadamente *Random Oversampling*, *Synthetic Oversampling* e *Recursive Feature Selection*.

Palavras-chave: Acilcarnitinas, Cancro Colorretal*, Feature Selection, Gradient Boosting, Machine Learning, Random Forest, Random Oversampling, Recursive Feature Selection, Synthetic Oversampling*

Abstract

In today´s world, the amount of information available in various sectors is increasing. That is the case in the healthcare area, where the collection and treatment of biochemical data seek to improve the decision-making in the treatment to be applied to a patient, using Machine Learning-based tools.

Machine learning is an area of Artificial Intelligence in which applying algorithms to a dataset makes it possible to predict results or even discover relationships that would be unnoticeable at first glance.

This project's main objective is to study several algorithms and techniques of Machine Learning to identify if the acylcarnitine profile may constitute a new biochemical marker for the prediction and prognosis of rectal cancer.

In the course of the work, different algorithms and data preprocessing techniques were tested. Three different experiments were carried out to validate the predictions of the models built for different scenarios, namely: predicting whether the patient has Colorectal Cancer, predicting which disease the patient has (Colorectal Cancer and other metabolic diseases) and predicting whether he has any disease. As a first analysis, the developed models showed good results in Colorectal Cancer screening.

The best results were obtained by the Random Forest and Gradient Boosting algorithms, together with data balancing and feature selection techniques, namely Random Oversampling, Synthetic Oversampling and Recursive Feature Selection.

Keywords: Acylcarnitines, Colorectal Cancer, Feature Selection, Gradient Boosting, Machine Learning, Random Forest, Random Oversampling, Recursive Feature Selection, Synthetic Oversampling

Agradecimentos

No final deste longo percurso, gostaria de agradecer a todos aqueles que me acompanharam no decurso do mesmo.

Ao meu orientador e coorientadora, Professor Doutor José Reis Tavares e Professora Doutora Isabel Praça, o meu profundo agradecimento por todo o apoio prestado, disponibilidade e transmissão de conhecimento ao longo deste ano.

À Professora Doutora Lúcia Lacerda, Professora Doutora Marisa Santos e também aos alunos de doutoramento Mestre Pedro Brandão e Mestre Ivo Barros, um agradecimento pelo fornecimento de informação, esclarecimento de dúvidas e constante ajuda no desenvolvimento desta dissertação.

À minha namorada, Milene, um agradecimento especial por toda a paciência, compreensão, ajuda e presença nos altos e baixos deste longo percurso.

Aos meus pais, irmã e restante família, um especial obrigado por todo o amor, carinho e incentivo para seguir em frente.

Index

List of Figures

List of Tables

Acronyms and Symbols

Acronyms List

ACC	Accuracy
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
BCD	Binary-class Classification of Disease
BCRC	Binary-class Classification of Rectal Cancer
BN	Bayesian Network
CI	Consistency Index
CR	Consistency Reason
CRC	Colorectal Cancer
CRISP-DM	Cross Industry Standard Process for Data Mining
DL	Deep Learning
DT	Decision Tree
ET	Execution Time
FEI	Front End Innovation
FFE	Fuzzy Front End
FN	False Negative
FP	False Positive
GB	Gradient Boosting
iFOBT	Immunochemical Faecal Occult Blood Test
KNN	K-Nearest Neighbour
LR	Logistic Regression

MCD	Multi-class Classification of Diseases
ML	Machine Learning
NB	Naïve Bayes
NCD	New Concept Development
NPPD	New Product and Process Development
PC	Pearson Correlation
QFD	Quality Function Deployment
RC	Rectal Cancer
RF	Random Forest
RFE	Recursive Feature Elimination
RI	Random Index
RL	Reinforcement Learning
RO	Random Oversampling
RT	Retention Time
RU	Random Undersampling
SL	Supervised Learning
SO	Synthetic Oversampling
SSL	Semi-supervised Learning
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
UFS	Univariate Feature Selection
UL	Unsupervised Learning
VA	Value Analysis
VC	Value for the Customer
VE	Value Engineering

VP	Value Proposition

Acronyms List

λ	Eigenvalue
σ	Standard Deviation
$\bar{x}$	Mean

1 Introduction

This chapter presents an introduction to the dissertation´s topic, including its context, problem, objectives, and the approach used in its realization. Finally, the structure of the document is presented.

1.1 Context

The quantity of data available has been growing exponentially in various sectors in the world today. With this growth, new ways of dealing with this large amount of information arise, considering that its processing using traditional analysis methods can be inefficient and omit certain patterns that other techniques can reveal [1].

The healthcare field is entering a new era where the abundance of biomedical data plays an increasingly important role. With all the information available, healthcare professionals try to ensure that the right treatment is given to the right patient at the right time, considering various aspects of data, including variability in molecular traits, environment, electronic health records and lifestyle. Thus, it is possible to explore associations between different types of information, allowing the development of reliable medical tools based on Machine Learning (ML) [2].

ML is a branch of Artificial Intelligence (AI) that focuses on the study of algorithms and statistical models that computer systems use so that they can learn from data, without being explicitly programmed [3]. Through the application of ML algorithms, it is possible to predict results or even discover relationships in the data that would be imperceptible at first sight.

1.2 Problem

Colorectal Cancer (CRC) is a type of cancer that originates in the colon or the rectum, which can also be called colon cancer or rectal cancer (RC), depending on the location [4]. CRC is the third most deadly and fourth most commonly diagnosed cancer in the world. CRC incidence has been steadily rising worldwide, especially in developing countries that are adopting the "western" way of life. Recent advances in early detection screenings and treatment options have reduced CRC mortality in developed nations, even in the face of growing incidence [5].

Given the possibility that this cancer does not show any symptoms at an early stage, its screening becomes essential. This is recommended for Portugal's asymptomatic citizens between 50 and 74 years old. This screening process may vary from case to case, with the application of methodologies that have different levels of effectiveness, such as the immunochemical faecal occult blood test (iFOBT) and colonoscopy [6]. This screening is not yet perfect in terms of efficiency and convenience for the patient. Therefore, it is necessary to find other ways that add greater efficiency at reduced costs to the current methodology. The analysis of the acylcarnitine profile in peripheral blood may be an interesting tool to assist in CRC screening.

Metabolic plasticity describes the ability of cells to respond and adapt their metabolism to allow proliferation, continuous growth, and survival in hostile conditions. This behaviour seems to be increasing in tumour cells, to satisfy their energy needs and maintain their uncontrolled proliferation. Components of the carnitine system are involved in bidirectional transport from the cytosol to the mitochondria, thus playing a key role in articulating the shift between glucose and fatty acid metabolism [7].

The analysis of the acylcarnitine profile in peripheral blood is used in the biochemical screening of diseases of fatty acids oxidation and organic acids metabolism. These pathologies have in common acylcarnitine abnormalities [8].

1.3 Objective

This dissertation aims to carry out an investigation whose purpose is to identify whether the acylcarnitine profile can constitute a new biochemical marker for the prediction of CRC. To achieve this, several ML algorithms and techniques will be analysed to choose the ones most suited to the problem.

In the context of [9], an initial study was made, focusing on identifying if the amino acid profile can constitute a biochemical marker to predict CRC, with positive results. This work is a continuation of the previous study, focusing on different biomedical data.

The result of this investigation can be used in the future in the development of a decision support system that will help healthcare professionals detect and prevent the disease, as well as outline the ideal recovery strategy for each patient.

1.4 Sources of information

The analytical data used in this dissertation come from patients with inherited metabolic disorders, studied at the Genetic Biochemistry Unit of *Centro de Genética Médica Jacinto Magalhães*, integrated into *Centro Hospitalar Universitário do Porto* (CHUPorto); along with data from patients with Locally Advanced Rectal Cancer, studied at the rectal Surgery Unit of the General Surgery Service of CHUPorto. The data was obtained by PhD students Pedro Brandão, MD, and Ivo Barros, MSc, in the development of the project entitled "MetLARC – Metabolic abnormalities on tumour response and resistance to neoadjuvant chemoradiotherapy in Locally Advanced Rectal Cancer".

The theme proposal and guidance in the proposing institutions are the responsibility of Marisa Santos, MD, PhD, who is responsible for one of the Emergency Teams of CHUPorto, coordinator of the Rectal Surgery Unit of the General Surgery Service of CHUPorto and coordinator of the Reference Center for Treatment of Rectal Cancer of CHUPorto; as well as Lúcia Lacerda, PhD, who is a Clinical Laboratory Geneticist at the Genetic Biochemistry Unit of *Centro de Genética Médica Jacinto Magalhães* of CHUPorto and member of the Reference Center for Diagnosis and Treatment of Inherited Metabolic Disorders at CHUPorto.

Prof. Marisa Santos and Lúcia Lacerda are respectively members of the "Oncology Research" and "Clinical and Experimental Human Genomics" groups of UMIB - *Unidade Multidisciplinar de Investigação Biomédica* / Unit for Multidisciplinary Research in Biomedicine (ICBAS, University of Porto), ITR- *Laboratório para a Investigação Integrativa e Translacional em Saúde Populacional* / Laboratory for Integrative and Translational Research in Population Health.

1.5 Approach

The approach used is based on CRISP-DM (Cross Industry Standard Process for Data Mining). This standard provides guidelines to be followed in ML projects, consisting of six phases that are detailed below, according to [10]:

- Business Understanding – the goal is to understand the project requirements from a business point of view;
- Data Understanding – consists of exploring the data to become familiar with it, discover possible problems and formulate hypotheses about hidden information;
- Data Preparation – includes all the activities carried out to build the final dataset, such as selecting the information to be used, fixing inconsistencies, and transforming the data in a way that is suitable to be used in the modelling phase;
- Modelling – a selection of ML techniques, algorithms and respective parameters is made, resulting in different ML models. These models will consume the data that was prepared in the previous steps;
- Evaluation – an evaluation of the previously built models is carried out, according to predefined acceptance criteria. In case the criteria are not met, the previous steps can be revisited, to search for possible improvements;
- Deployment – the knowledge acquired by the models is organized and presented in a way that the client can use.

Figure 1 illustrates all the CRISP-DM steps.

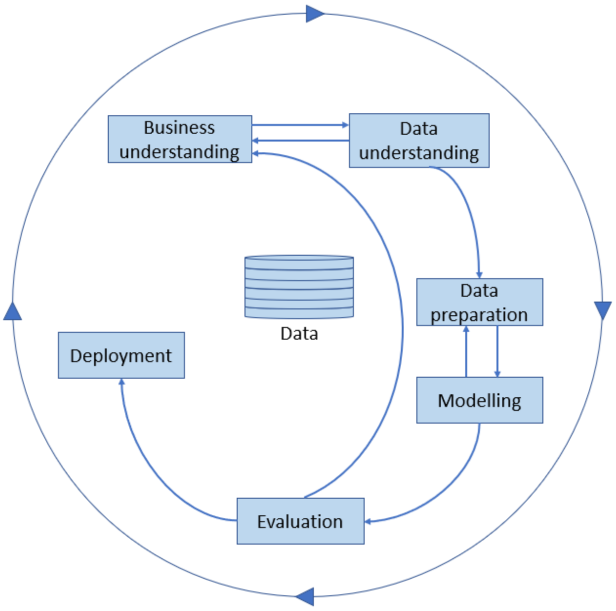


Figure 1 – CRISP-DM steps (adapted from [10]).

1.6 Document Structure

The document structure is as follows:

- Introduction – describes the topic of this dissertation, including its context, the problem, the objectives to be achieved and the approach used in its realization;

- State of the Art – presents the fundamental concepts to better understand the work carried out. Initially, an introduction to CRC and acylcarnitines is made. Then, an overview of ML and Deep Learning (DL) is presented. Finally, literature examples of the application of ML in the prediction and prognosis of cancer are given;
- Value Analysis – analyses the value of this dissertation, by demonstrating the benefits that it can bring to healthcare professionals and how it can translate to a business idea;
- Data Analysis and Design – gives an overview of the data used, as well as the design details of the software application that was developed to conduct the experiments;
- Experimentation and Evaluation – describes the experiments made, alongside an evaluation of the results obtained;
- Conclusion – presents the conclusions about the work carried out, highlighting what objectives were achieved and what the next steps should be.

2 State of the Art

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

2.1 Colorectal Cancer

Colorectal Cancer is one of the most common cancers worldwide. In 2020 it was the third most diagnosed cancer in men and the second in women [11], as shown in Figure 2.

This type of cancer generally develops from a cell or group of cells in the inner lining of the intestinal wall that, when multiplying, form a small group of epithelial cells called a polyp. This polyp is characterized by elevations or projections of the said inner lining detected on colonoscopy [12]. Adenomatous polyps, often known as adenomas, are a type of polyps that can turn into cancer. This type of polyp becomes capable of invading neighbouring tissues, through the intestinal wall or the lymphatic and/or blood circulation. The causes of these transformations seem to be due to the sum of mutations, hereditary or acquired over the years, in the genes that control cells [12].

For that reason, screening is essential to detect cancer early, preferably before adenomas become adenocarcinomas. In Portugal, the most utilized test for screening is the iFOBT and colonoscopy [6].

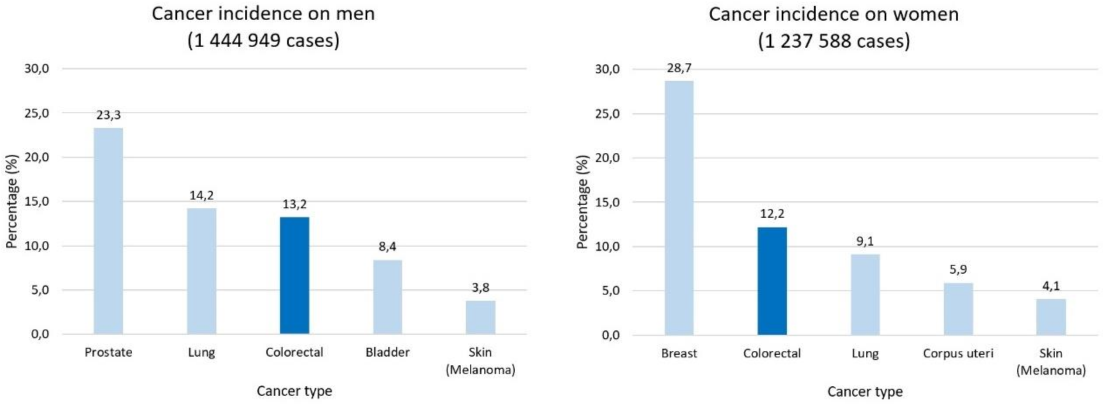


Figure 2 – Cancer incidence in 2020 (adapted from [11]).

2.2 Acylcarnitines

Acylcarnitines are fatty acid metabolites that are involved in the production of energy to maintain cell activity [13]. Currently, they are used in the study of many diseases, such as metabolic and neurological disorders, diabetes, cardiovascular diseases, depression and certain cancers. Historically, they are diagnostic markers for errors of fatty acid oxidation; also, they are being studied as markers of deficits in mitochondrial and peroxisomal b-oxidation activity, energy metabolism, insulin resistance and physical activity [14].

2.3 Machine Learning (ML)

ML can be defined as the scientific study of algorithms and statistical methods used by computer systems to perform a certain task without being explicitly programmed. ML applications are many and can range from web page classification in search engines, to image processing, data mining, and predictive analysis, among others [15].

Different algorithms can be used in ML, so it is necessary to have a good understanding of the problem to select the most suitable algorithm [15]. We can divide the types of problems into the following categories, according to [16]:

- Classification – the objective is to separate the data into distinct classes, known beforehand such as Yes/No and True/False. If the number of output classes is equal to 2, the problem is defined as a binary-class classification problem; if it is greater than 2 it is called a multi-class classification problem;
- Anomaly detection – consists of analysing certain patterns in the data and detecting changes or anomalies in that pattern. An anomaly can be defined as a rare observation that deviates significantly from most of the data;

- Regression – deals with continuous and numeric outputs. An example would be to predict the price of a house;
- Clustering – the goal is to discover structures in the information unknown *a priori* and attempt to divide each data point into distinct groups, called clusters, based on their similarities;
- Reinforcement – the algorithm's decisions consider past learning experiences. This way, without knowing how to solve the problem, each action the algorithm performs is associated with a reward or penalty, which helps it learn how to reach the desired solution.

2.4 Machine Learning Paradigms

According to [3], the different ML algorithms can be classified according to the quantity and type of supervision used. There are four main paradigms: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning, which will be detailed in this section.

2.4.1 Supervised Learning (SL)

SL is a type of learning used in ML where the training data provided to the algorithm, containing a set of attributes (features), has the expected output (classes). Based on the training performed, the algorithm can perform predictions/classifications on the test data, which does not contain the expected solution. This process is iterative, and the given parameters can be adjusted until reaching the desired outcome [15]. This process is shown in Figure 3.

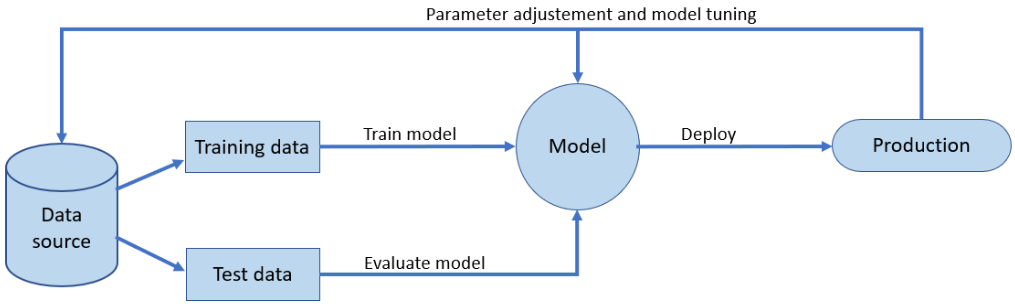


Figure 3 – SL process (adapted from [15]).

According to [17], some of the common applications of SL are:

- Predictive analysis based on regression or categorical classification;
- Pattern detection;

- Natural language processing (for example spam detection);
- Sentiment analysis;
- Automatic classification of images;
- Automatic processing of sequences (for example music and voice).

Some of the algorithms used in this paradigm will be detailed in the following sections. The algorithms were selected according to their relevance and possible application in this project.

Decision Tree (DT)

DT is an algorithm in which the result of its learning is represented by a graph with the shape of a tree. This graph is used to classify new cases according to the value of their features [18].

Each tree is composed of nodes representing the dataset features and branches containing the possible values that each node can have. For each new case, the process starts from the root and using the values of each feature, the tree is traversed until reaching the respective classification [18]. An example of a decision tree is illustrated in Figure 4.

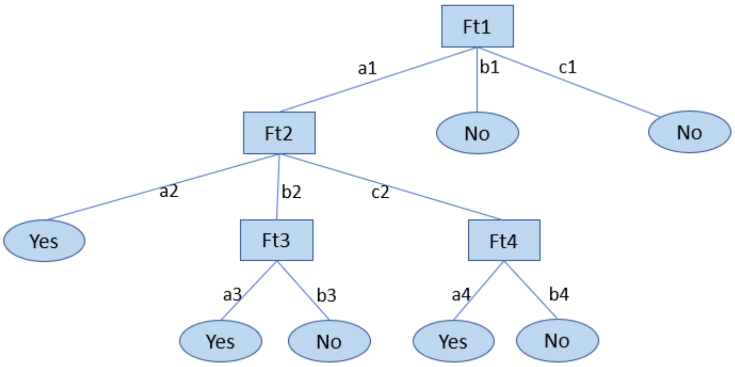


Figure 4 – DT algorithm example (adapted from [18]).

Naïve Bayes (NB)

NB is an algorithm based on Bayes´ Theorem (Figure 5). This theorem describes the probability of an event happening, based on prior knowledge of conditions that may be related to that event. The term Naïve implies that independence between features is assumed, despite the uncertainty of the veracity of this fact [15].

This algorithm is widely used for text classification, namely in the construction of search engines and filters for spam [19].

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad \text{given} \quad \begin{array}{ll} P(c|x) & \text{- Probability of } c \text{ given } x \\ P(x|c) & \text{- Probability of } x \text{ given } c \\ P(c) & \text{- Probability of } c \\ P(x) & \text{- Probability of } x \end{array}$$

Figure 5 – Bayes´ Theorem (adapted from [15]).

Random Forest (RF)

RF is an ensemble method, which combines the use of multiple learning algorithms to improve the results. Specifically, it is represented by a set of DTs, each one using a distinct subset of the input data and producing a classification result. Finally, the classification with the most ‘votes’ (for a discrete classification result) or the average of all trees (for a numerical classification result) is chosen [15]. Figure 6 shows an example of a Random Forest algorithm.

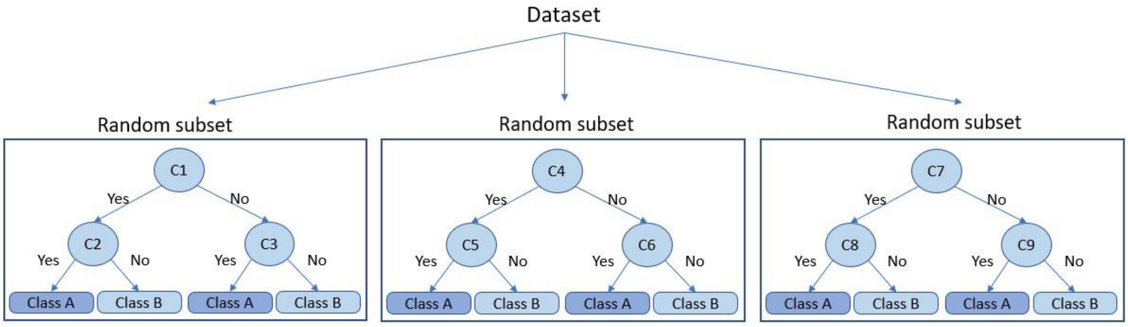


Figure 6 – RF algorithm example (adapted from [20]).

Gradient Boosting (GB)

Similarly to RF, GB is an ensemble method. It works by combining multiple weak learners, typically individual DT, into strong learners. A weak learner is a model that performs slightly better than random chance. The learners are connected in series, so that each one of them tries to minimise the errors (residuals) of the predecessor, with the use of a loss function. The loss function optimization is done using gradient descent, hence the name of the algorithm. Typical loss functions can be the mean squared error for regression tasks and logarithmic loss for classification tasks [21]. This process is exemplified in Figure 7.

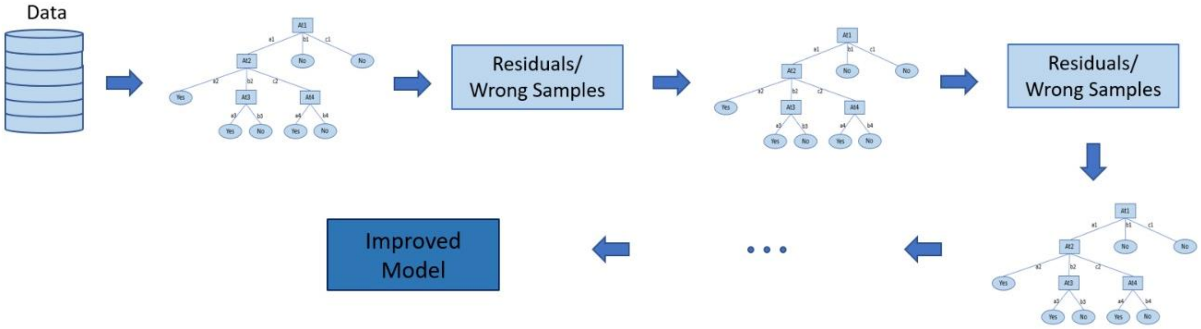


Figure 7 – GB algorithm example (adapted from [21]).

Support Vector Machine (SVM)

SVM is an algorithm that works with the concept of margin calculation. Each case is represented by a point in an n-dimensional space (being n the number of features in the dataset), with coordinates corresponding to the value of the features. Training data is classified into different classes by finding a line that (hyperplane) that leaves the best margin of separation between them [16]. Figure 8 illustrates this process.

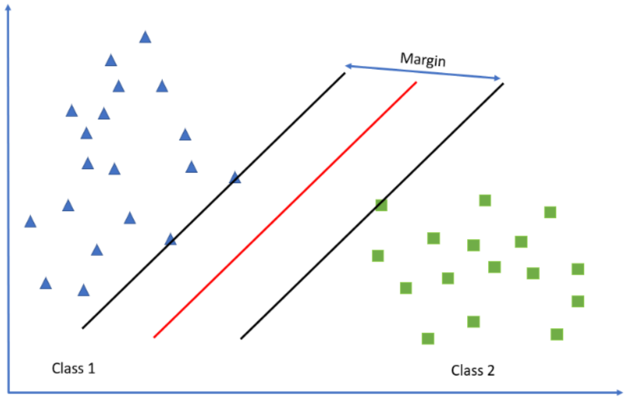


Figure 8 – SVM algorithm example (adapted from [15]).

This is an algorithm capable of performing non-linear classification, regression and even anomaly detection [3].

Logistic Regression (LR)

LR is a regression algorithm that is used to predict the probability of a case belonging to a certain class, such as the probability of an email being spam. It is seen as an extension of linear regression, replacing a continuous value with a value between 0 and 1. If this probability is greater than 50%, the algorithm predicts that the case belongs to the class 1 and vice versa, making this algorithm a binary classifier [20]. The probability is calculated according to a logistic function that outputs a value between 0 and 1, represented in Equation (2.1) [3].

$$f(x) = \frac{1}{1 + \exp(-x)}$$ Equation (2.1)

The LR model can be generalized to model a categorical variable with more than two values, known as Multinomial Logistic Regression [20].

K-Nearest Neighbours (KNN)

KNN is a non-parametric algorithm, that is, it does not make any assumptions about the distribution of the data. In this algorithm, 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 [22]. An example of this algorithm is presented below, in Figure 9.

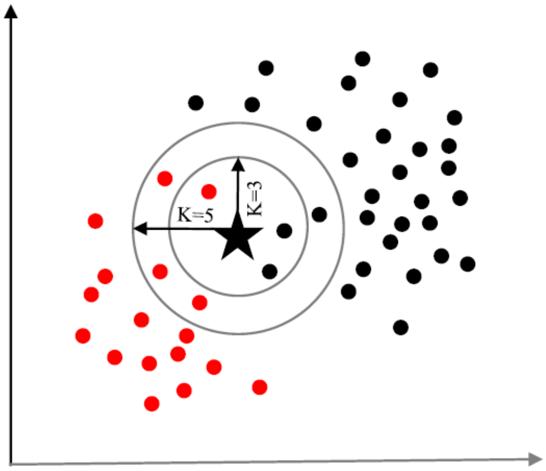


Figure 9 – K-Nearest-Neighbours algorithm example [22].

Considering the previous example, where the objective is to classify a star object as belonging to the "red" or "black" class, different values of K produce different results. For K=3, the distance is defined by the 3 closest points, which are mainly from the "black" class, leading to the star being classified as "black". On the other hand, for K=5, the star is classified as "red", since the nearest points are predominantly from that class.

This algorithm can be used for regression and classification problems [22].

Artificial Neural Network (ANN)

ANN is an algorithm inspired by the functioning of neural networks in the human brain, where neurons are connected and communicate with each other through signals [20].

Similarly, ANNs consist of a set of connected nodes that are grouped into distinct layers. Each layer is responsible for a different task, and in addition to the input and output layers, there are hidden layers where various 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 [20].

The nodes and their connections have associated weights that are adjusted according to the learning made by the algorithm, which allows the identification of the connections that are more relevant to the classification result [16]. Figure 10 shows an example of an ANN.

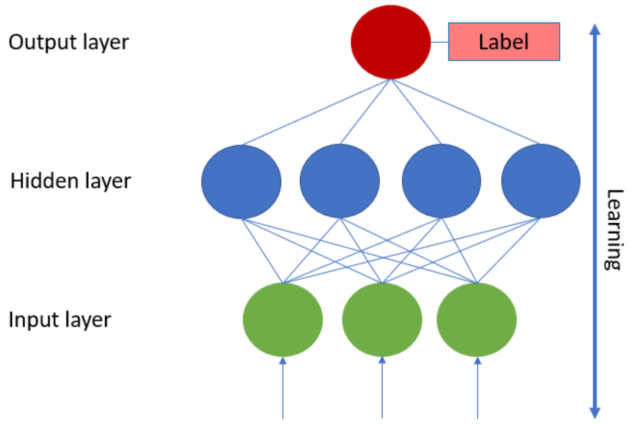


Figure 10 – ANN algorithm example (adapted from [2]).

Supervised Machine Learning Algorithms Comparison

In this section, a summary of the previously mentioned SL algorithms' advantages and limitations is presented in Table 1.

Table 1 – Advantages and limitations of SL algorithms, according to [20] and [21].

Supervised Algorithm	Advantages	Limitations
DT	• Resultant tree is easy to interpret and understand; • Simple data preparation process; • Multiple data types supported: numeric, nominal, and categorical; • Ability to generate robust classifiers and be validated using statistical tests.	• Classes are required to be mutually exclusive; • Algorithm cannot branch if any feature value is missing; • Dependant on the order of the features; • Does not perform as well as other classifiers, such as ANN [23].
NB	• Simple and very useful for large datasets; • Can be used for binary and multi-class classification problems; • Does not require much training data; • Can handle continuous and discrete data.	• Classes are required to be mutually exclusive; • Dependency between features negatively affects the performance of the algorithm; • Numeric attributes are assumed to follow a normal distribution.
RF	• Performs better than an individual DT; • Scales well for large datasets; • Can provide estimates of the most important features for the classification.	• More computationally expensive than DT; • Number of trees must be defined; • Susceptible to overfitting, a term used when the algorithm is well adjusted to the training data but performs poorly on new data.
GB	• Support numerical and categorical features with no preprocessing needed; • Can perform better than RF; • Highly efficient in classification and regression tasks.	• Requires careful tuning of parameters; • Can overfit if too many trees are used; • Sensitive to outliers.

SVM	• Can handle multiple feature spaces; • Less risk of overfitting compared to LR; • Performs well in the classification of semi-structured or unstructured data, such as texts and images.	• Computationally expensive for large datasets; • Susceptible to noise in data; • Generic SVM cannot classify more than two classes unless it is extended.
LR	• Easy to implement; • Easy to update; • Does not make assumptions regarding independent variables; • Provides a probabilistic interpretation of model parameters.	• Accuracy is low when handling complex relationships; • Does not consider the linear relationship between variables; • Prediction accuracy may be affected by sampling bias; • Unless a multinomial logistical regression model is used, it can only classify two classes.
KNN	• Can handle noise in data and missing feature values; • Simple and quick to perform the classification; • Can be used for classification and regression.	• Computationally expensive when the number of features increases; • Features are given equal importance which can degrade performance; • Does not provide information on which features are important for the classification.
ANN	• Can detect complex relationships between dependent and independent variables; • Doesn’t require much training; • Multiple training algorithms available; • Can be applied to regression and classification problems.	• Requires computationally expensive resources for a complex classification problem; • User doesn´t have access to the decision-making process; • Independent variables require preprocessing.

2.4.2 Unsupervised Learning (UL)

UL is a machine learning paradigm in which the data provided in the learning phase does not contain the expected label. The algorithm is responsible for finding structures and patterns in the available information. When new data is presented, the previously obtained knowledge is used to recognise the class of said data. This type of learning is mainly used for clustering and feature reduction tasks [15]. Table 2 displays some of the algorithms used for these types of tasks.

Table 2 – Unsupervised ML algorithms [3].

Task	Algorithms
Clustering	K-means Clustering Hierarchical Cluster Analysis Expectation Maximization
Feature reduction	Principal Component Analysis (PCA) Kernel PCA Locally Linear Embedding t-distributed Stochastic Neighbor Embedding

2.4.3 Semi-supervised Learning (SSL)

SSL is a ML paradigm that works as a combination of supervised and unsupervised learning. This means that this paradigm is used when a small part of the data has the expected label, but the vast majority does not, due to the high cost of obtaining it. This type of learning is used for classification and regression problems [16].

2.4.4 Reinforcement Learning (RL)

RL is a ML paradigm in which the learning process follows a trial-and-error method. Each decision the algorithm makes is associated with a reward or penalty, which helps discover the best way to obtain the expected solution. This paradigm is commonly used in games and temperature control programs [16].

2.5 Deep Learning (DL)

DL is a subset of ML, based on the functioning of an ANN, but with the difference of having more layers and parameters [24].

DL consists of multiple layers of non-linear processing units that are used for feature extraction and transformation. While in the lower layers the complexity of learned features is reduced, in the upper layers the previously obtained knowledge is used to discover more complex features. This makes DL suitable for cases where there is a large amount of data from different sources [25]. Using the recognition of a human face as an example, learning in DL is done progressively, with the contours of the face being recognised first, followed by the nose, ears, and mouth, until the representation of a complete face [25].

According to [24], some of the applications of DL are image, video, voice recognition and natural language processing.

Figure 11 shows a comparison of a traditional ANN and a DL model.

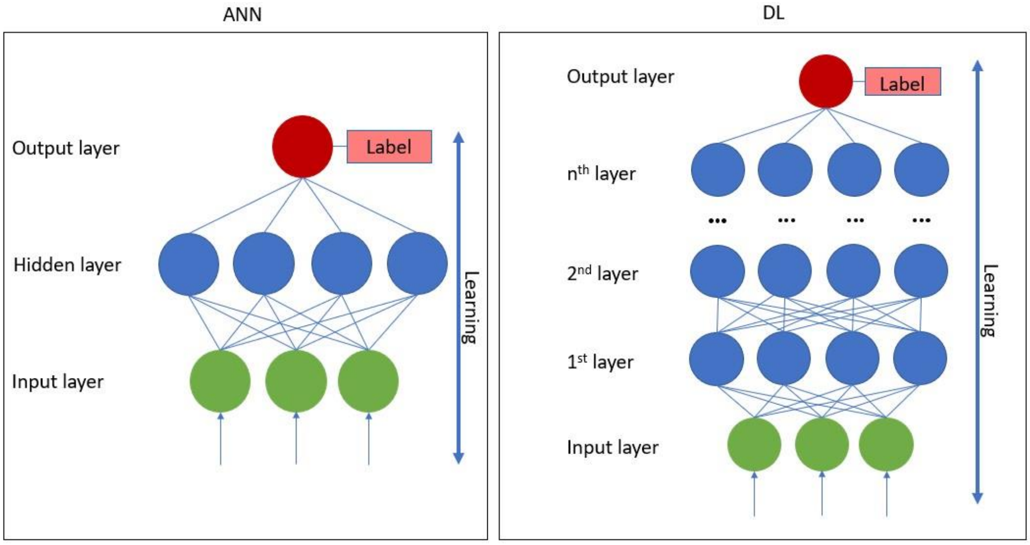


Figure 11 – Comparison of a traditional ANN and a DL model (adapted from [2]).

2.6 Machine Learning Applications in Cancer Prediction and Prognosis

In the last decades, ML techniques have been widely applied to cancer prognosis and predictions. In fact, with the appearance of new technologies in the field of medicine, large amounts of cancer data are available to the medical community. With this data, scientists applied different ML methods to discover patterns and relationships that help them effectively predict future outcomes of a cancer type [26].

Cancer prognosis and prediction deal with three different types of tasks: the prediction of cancer susceptibility, in which the goal is to find the likelihood of developing a certain type of cancer; the prediction of cancer recurrence, which objective is to predict the probability of redeveloping a type of cancer after complete or partial remission; and the prediction of cancer survival [26].

Table 3, Table 4 and Table 5 present some relevant publications regarding the three different types of prediction tasks mentioned above. A variety of ML algorithms and techniques were used, such as ANN, SVM, DT, Bayesian Network (BN), and some SSL algorithms as well.

Table 3 – Relevant publications that used ML methods for cancer susceptibility prediction (adapted from [26]).

Publication	ML method	Cancer type	Number of patients	Accuracy
Ayer T et al. [27]	ANN	Breast cancer	62 219	96.5%
Waddell M et al. [28]	SVM	Multiple myeloma	80	71%
Listgarten J et al. [29]	SVM	Breast cancer	174	69%
Stojadinovic A et al. [30]	BN	Colon carcinomatosis	53	71%

Table 4 - Relevant publications that used ML methods for cancer recurrence prediction (adapted from [23]).

Publication	ML method	Cancer type	Number of patients	Accuracy
Exarchos K et al. [31]	BN	Oral cancer	86	100%
Kim W et al. [32]	SVM	Breast cancer	679	89%
Park C et al. [33]	Graph-based SSL algorithm	Colon cancer/ Breast cancer	437/374	76.7%/80.7%
Tseng C-J et al. [34]	SVM	Cervical cancer	168	68%
Ahmad LG et al. [35]	SVM	Breast cancer	547	95%

Table 5 – Relevant publications that used ML methods for cancer survival prediction (adapted from [26]).

Publication	ML method	Cancer type	Number of patients	Accuracy
Chen Y-C et al. [36]	ANN	Lung cancer	440	83.5%
Park K et al. [37]	Graph-based SSL algorithm	Breast cancer	162 500	71%
Chang S-W et al. [38]	SVM	Oral Cancer	31	75%
Xu X et al. [39]	SVM	Breast Cancer	295	97%
Gevaert O et al. [40]	BN	Breast cancer	97	85.1%
Rosado P et al. [41]	SVM	Oral Cancer	69	98%
Delen D et al. [42]	DT	Breast Cancer	200 000	93%
Kim J et al. [43]	SSL Co-training algorithm	Breast Cancer	162 500	76%

As demonstrated by Table 3, Table 4 and Table 5, ML has been used for handling different types of prediction tasks regarding multiple types of cancer, namely: Oral, Breast, Colon, Cervical and others. Among all the publications presented, those related to Breast Cancer were the ones that used the larger number of patient data to build the ML models. Different ML algorithms were used; although it is possible to see that SVM was the algorithm that was more times selected.

2.6.1 Amino acid profile for CRC prediction

As mentioned in section 1.3, a previous study was made to check if the amino acid profile can constitute a biochemical marker to predict CRC. For that, different ML algorithms and techniques were studied and tested. With the data available, three different experiments were made, testing the effectiveness of ML models in the following scenarios: predict the presence of disease in a patient (CRC and other metabolic diseases), predict which disease affects the patient, and predict the diagnosis of CRC. For the last experiment, only patients 18 years of age or more were considered. It was concluded that the amino acid profile can be a good indicator of CRC [9].

Table 6 displays the experiments that were done, the number of samples used, the ML algorithm that produced the best results and the accuracy that was obtained.

Table 6 – Amino acid profile for CRC prediction study results [9].

Experiment	ML algorithm	Number of samples	Accuracy
Presence of disease	SVM	3842	85.6%
Diagnosis of which disease affects the patient	RF	3408	99.4%
CRC diagnosis	RF	175	95.9%

As discussed with healthcare professionals, the gathering of data related to the amino acids profile is harder compared to the acylcarnitine profile, as the last one can be achieved by a simple collection of blood in filter paper, which can be done at home without any preparation. For that reason, it will be extremely beneficial for them if this data can be used to help in the diagnosis of patients with CRC.

2.7 Machine Learning Libraries

Manually coding ML and DL algorithms can be very time-consuming and inefficient. For that reason, libraries were created to help with this process. Each library has different features and optimization techniques, which leads to performance variations between them when using the same algorithm. Some of the most used libraries will be presented in the following sections.

2.7.1 TensorFlow

TensorFlow is an open-source library originally developed by Google´s divisions and later released in 2015 as free open-source software under Apache License 2.0. It is mainly used for DL and allows the use of graphics cards in calculations, due to its computational core being written in C++, using CUDA technology. The interface part is implemented in Python [44].

This library is based on the principle of dataflow. According to this principle, the program is organized in computational blocks associated with each other in the form of a directed graph, named computational graph. The data structure used is called a tensor, which represents a multidimensional array of elements [44].

The architecture used in this library is suited for parallel calculations on both multi-core processors and distributed cluster systems and the building of neural networks [44].

2.7.2 PyTorch

PyTorch was created based on Torch, which was developed in C and used Lua as the interface. Python´s growth in popularity led to Torch being rewritten in C++/CUDA and Python. It was initially developed by Facebook and currently, it is an open-source library [44].

The basic principle of this library is the same as TensorFlow, the dataflow concept. The main difference is that TensorFlow uses a static computational graph, compared to the dynamic one used by PyTorch. This means that the graph can be modified when running the model, adding or removing nodes as needed, while in TensorFlow the entire graph must be specified before [44].

2.7.3 SciKit Learn

SciKit Learn library is suitable for traditional ML and data pre-processing tasks. It is mainly written in Python and uses some other libraries for the algorithm's implementation, namely NumPy and SciPy. It implements various methods of classification, regression analysis, clustering, and other classical ML algorithms [44].

This library doesn´t support the concept of dataflow, making it unsuitable for scaling the models for multi-core processors and graphics accelerators. The degree of parallelism is limited to what is implemented in NumPy [44].

2.7.4 Libraries Comparison

In this section, a summary of the previously mentioned libraries' advantages and limitations is presented in Table 7.

As demonstrated by Table 7, both PyThorch and TensorFlow would be the more adequate libraries if the use of DL was considered. However, due to the low volume of data used in this dissertation, this was not the case. As such, the library selected to be used was Scikit-Learn, as it provides all the tools needed to implement several ML algorithms, its easy to use and has a large and active community.

Table 7 – Advantages and limitations of ML libraries, according to [44], [45].

Library	Advantages	Limitations
TensorFlow	• Dataflow programming which provides the bases for DL research and development; • Works efficiently with mathematical expressions involving multi-dimensional arrays; • Flexible architecture that can run on multi-core processors and graphic cards; • More complete visualization tools compared to PyTorch; • Offers the possibility of using low-level APIs or high-level ones such as Keras.	• Debugging can´t be made with typical Python methods, instead it is necessary to use tfdbg (TensorFlow debugger); • Static computational graph; • Parallel execution of tasks is more difficult to achieve, compared to PyTorch.
PyTorch	• Dataflow programming using dynamic computational graphs; • Typical python debugging methods can be used; • Simpler to achieve data parallelism, compared to TensorFlow; • Highly extensible, programmers can develop in C/C++ using an extension API; • Simpler to execute parallel tasks, compared to TensorFlow.	• Data visualization tools are not as complete as TensorFlow; • From the production side, the TensorFlow community is larger and more active.
Scikit Learn	• General purpose, open-source, commercially usable and popular Python ML tools; • Closely coupled with statistic and scientific Python packages; • Well-updated and comprehensive set of algorithms and implementations.	• Does not support graphic cards; • Only basic tools for building neural networks.

3 Value Analysis

In this chapter, the value analysis of the project will be carried out, from the process to the methods applied for such. The main objective of value analysis is to evaluate the increase in value of a product/service, identifying unnecessary associated costs, without sacrificing its quality [46].

The core product of this work is under development and not yet in use by end-users. As such, the model chosen for value analysis is Value Engineering (VE). This is based on the application of the Value Analysis (VA) model, which is applied to existing products/services [46]. VA is a systematic and organized process of analysis and evaluation, that requires an understanding of the product´s purpose, as well as the specifications needed to deliver value to the customer. This process should result in improvements to reduce the cost of production, without compromising the quality and functionality of the product [46]. Figure 12 shows a diagram with the phases to be implemented in this process. It consists of five phases, and the second phase called functional analysis is subdivided into two parts.

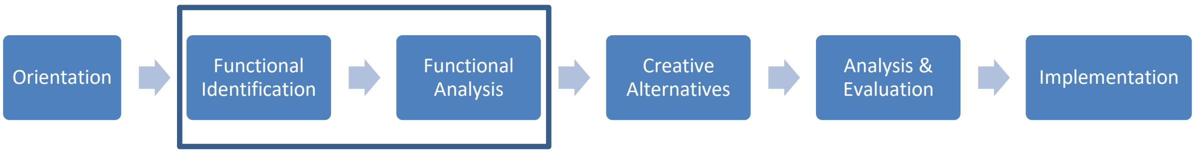


Figure 12 – Phase diagram of the VA model (adapted from [46]).

According to [46], the main objectives of each phase are:

- Orientation – product identification and selection;
- Functional Analysis – identification of the most important functions of the product;

- Creative Alternatives – creating alternatives and more cost-effective ways to achieve the product´s function;
- Analysis and Evaluation – evaluation, prioritization, and pre-selection of ideas from the previous phase considering their cost/value potential;
- Implementation – reporting the findings and gaining permission to implement them.

3.1 Orientation

According to [47], the innovation process can be divided into three parts: Front End Innovation (FEI), New Product and Process Development (NPPD) and commercialization. FEI, commonly called Fuzzy Front End (FFE), is defined as the activities that precede the formal and structured NPPD process and is also considered one of the great opportunities to improve the overall innovation process. To clarify FFE components, describing them in a common language, Koen developed the New Concept Development (NCD) model, represented in Figure 13. This was the model applied for the first stage.

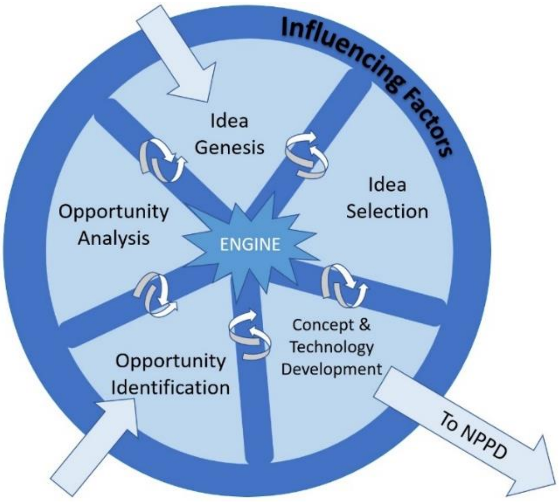


Figure 13 – NCD model (adapted from [47]).

The NCD model is divided into three different areas [47]:

- The engine – responsible for driving the five main elements of the NCD and powered by the organization´s leadership and culture;
- Influencing factors – consist of factors that influence the engine and the wheel, such as business strategy, and organizational capabilities, among others;
- The wheel – consists of the five elements of the FFE: opportunity identification, opportunity analysis, idea genesis, idea selection and concept & technology development.

Next, the areas of the NCD will be presented in the context of this project.

3.1.1 Influencing Factors

Influencing factors consist of factors that influence the engine and the wheel of the NCD model. They are represented by the organization’s capabilities, the influence of competitors and clients and the maturity of the technologies to be used [47].

For this project to be carried out, an effort to collect patient data progressively needs to be made, so that the developed solution is based on up-to-date information. The quantity and quality of the data will impact the accuracy of the algorithms used.

Technology advancements are also an influencing factor. The appearance of new technologies or algorithms, or even improvements to the ones used in the current solution need to be foreseen.

Lastly, the adherence and satisfaction of healthcare professionals are essential. Their feedback will add value to the solution, by suggesting improvements and corrections. This feedback will also have a direct impact on the quality of life of their patients.

3.1.2 Engine

The engine is responsible for driving the five core elements of the NCD and is fuelled by the organization´s leadership and culture [47]. The main goal of this project is to take advantage of the quantity of information available and combine it with available technologies that will improve the way healthcare professionals work, which will benefit the quality of life of their patients.

This project will be realized with the guidance of healthcare professionals of *Centro Hospitalar Universitário do Porto* (CHUPorto), which is a central university hospital with aims at providing “humanized, competitive and reference healthcare, promoting articulation with other partners in the system, valuing pre-and post-graduate education and professional training, stimulating and encouraging research and scientific development in the healthcare area” [48].

3.1.3 Opportunity Identification

Opportunity Identification occurs when the organization recognizes opportunities that it wants to pursue. This can be a response to a competitive threat, a possibility to gain a competitive advantage or ways to improve its current operations [47].

As mentioned in section 2.1, CRC is one of the most common and deadliest cancers in the world, which can be treated in various ways with different levels of effectiveness. The abundance of biomedical data and knowledge about the molecular mechanisms of this type of cancer has been increasing, and, with that in mind, there is an opportunity to use the data available to help healthcare professionals in the detection and treatment of CRC.

3.1.4 Opportunity Analysis

To translate Opportunity Identification into specific business and technological opportunities, additional information is needed. This information can be gathered by focus groups, market studies and/or scientific experiments and the effort expended on those is dependent upon the attractiveness of the opportunity, development effort, fit with business strategy and culture, and risk tolerance of decision-makers [47].

Many studies were developed in the healthcare area regarding the use of ML in cancer prognosis and prediction, some of them mentioned in section 2.6. In the context of [9], an initial study was made to check if amino-acid profile data can be used to predict if a patient has CRC. As a continuation of that work, this dissertation has the main goal of identifying if acylcarnitine profile data can be used as a biomarker for CRC prediction. As discussed with healthcare professionals, the acylcarnitine profile data is easier to collect compared to amino acids, as it is a simple collection of blood in filter paper that can be done at home, without any preparation, as described in section 2.2.

For a better assessment of the value this project brings to the customer, the following chapters will present the following topics: value for the customer, perceived value, and value proposition.

Value for the Customer (VC) and Perceived Value

According to Woodwall [49], the term “Value for the Customer” (VC) is a personal perception that the customer has of an organization´s offer that can be seen as a reduction of sacrifices, presence of benefits or the result of a combination between benefits and sacrifices. The perceived value is the evaluation of a product or service by the consumer according to his needs and expectations and is based on what is received versus what is given.

In Table 8, an overview of the customer´s benefits and sacrifices in the context of this project is presented.

Table 8 – Benefits and sacrifices associated with this project.

Benefits	Sacrifices
• Early CRC diagnosis; • Better decision-making regarding patient treatment; • Reduced costs in unnecessary exams/procedures; • Increased knowledge about CRC; • Patient´s satisfaction; • Patient´s quality of life.	• Effort to collect data; • Learning curve of the application; • Software maintenance and support costs.

Value Proposition (VP)

Value Proposition is “an overall view of a company´s bundle of products and services that are of value to the customer” [50]. With it, the following questions must be answered, according to [51]:

- What is your product?
- Who is your target customer?
- What value do you provide?
- Why is your product unique?

With those questions in mind, the product to be developed is a decision support system that helps healthcare professionals in the prediction and treatment of CRC based on patient´s clinical data (amino-acid profile, acylcarnitine profile and other types of data that can be studied in the future) with a high accuracy rate and reliable predictions. At the time of writing, no system is available in the market that uses this type of data for the prediction and treatment of this cancer.

With this decision support system, healthcare professionals have one more tool to help them early detect CRC cases and find the right strategy of treatment for them. This will be beneficial for all persons involved, by having a higher degree of control of the disease, as well as greatly improving the quality of life of the patients.

3.1.5 Idea Genesis

The goal of this phase is to transform an opportunity into a concrete idea. The generation of this idea can be an iterative process as it is examined, studied, discussed, and developed in conjunction with other elements of the NCD model. Direct contact with customers and users, as well as other companies and institutions, can enhance this activity [47].

In the context of this phase, the project was discussed with a healthcare professional and the following requirements were identified:

- Obtain the diagnosis of the disease based on different parameters (amino-acid and acylcarnitine profile);
- Identify the features that are most correlated with the diagnosis;
- Detect if a treatment applied to a patient was effective (post-chemoradiotherapy and post-surgery);
- Divide patients into different clusters, according to their clinical data;
- Recommend the best type of treatment for the patient;
- Employ the clinical data to diagnose other diseases;
- Develop a user-friendly interface for the application.

3.1.6 Idea Selection

After the generation of ideas, it is necessary to identify which ones are the most important to achieve the most business value, considering the need, cost, and benefits of each of them [47].

With the application of that process, it was defined that the most important requirements, in the context of this dissertation, would be the diagnosis of CRC based on acylcarnitine profile data and the identification of features that are most correlated with the diagnosis.

3.1.7 Concept and Technology Development

The NCD model ends with the development of a business case based on estimates of market potential, customer needs, investment requirements, competitor assessments, technology unknown and overall project risk [47]. The concept associated with this product is the development of a decision support system that analyses and processes patients´ clinical data using ML algorithms and techniques and helps healthcare professionals in the prediction and treatment of CRC, improving the control of the disease and patient quality of life.

3.2 Functional Identification and Analysis

The next step of VA is the functional identification and analysis of the product. In this phase, the product is analysed through the identification of its most important functions. A function is defined as the use demanded of a part of a product as well as the esteem value it provides [52].

First, functional identification is carried out, which consists of the identification of the most important functions and requirements of the product. After, there is an analysis of those requirements, evaluating and comparing them [52].

3.2.1 Functional Identification

One of the tools used in the identification of the functionalities of a product is Qualify Function Deployment (QFD). QFD is a quality tool to project a product according to customer requirements, involving in the process all the product´s organization members [46].

To achieve the expected results of the application of this method (better comprehension of the customer needs, optimized production because of the fewer changes that occurred during the project, business increase, etc.), each requirement proposed by the customer is segmented and analysed, identifying ways to achieve that segmentation [53].

To start the application of QFD, it is necessary to survey the requirements with the interested parties, in this case, the healthcare professionals, as well as the attribution of the weight/degree of importance for each requirement. To this end, a meeting with healthcare professionals was held to identify the requirements, which were presented in subchapter 3.1.5. Considering that the sum of the weights must be 100, the value of each requirement was given according to its importance to the customer, as well as its complexity and development time. The requirements (“Whats”) and the respective weights are presented in Table 9.

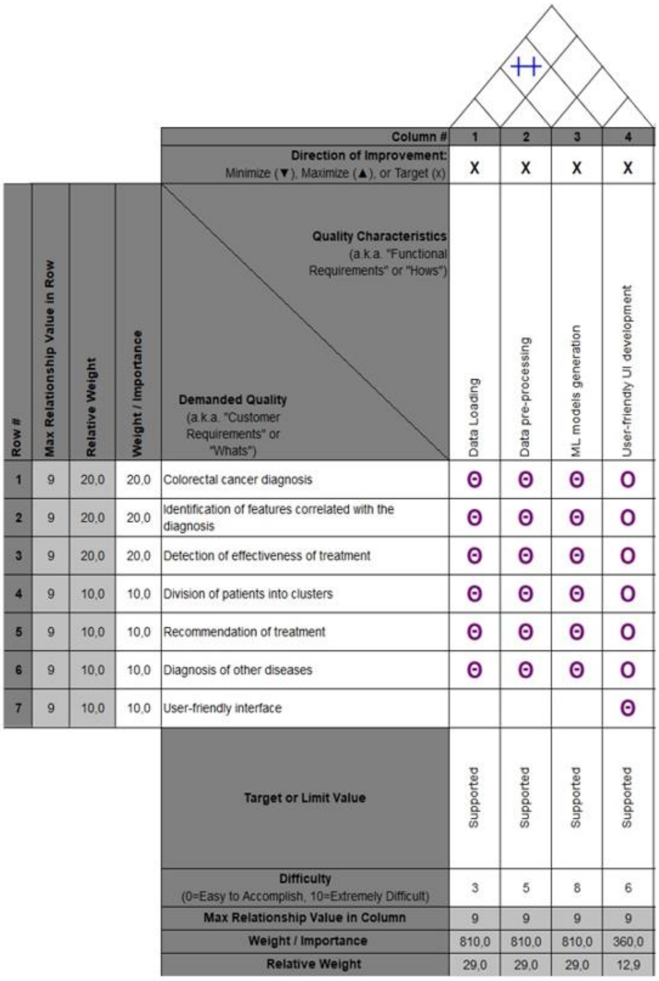
Table 9 – QFD customer requirements and respective weights.

Requirement	Weight
Diagnosis of CRC	20
Identification of features correlated with the diagnosis	20
Detection of the effectiveness of the applied treatment	20
Division of patients into clusters	10
Recommendation of treatment	10
Diagnosis of other diseases	10
Development of a user-friendly interface	10

In the next phase of this method, it is necessary to identify the quality requirements, that represent how the customer requirements will be achieved from a technical point of view (“Hows”). These are the following: data loading; data pre-processing, ML models generation and user-friendly user interface (UI) development.

3.2.2 Functional Analysis

For the functional analysis phase, the “House of Quality” method is going to be used. This method has the objective of comparing quality requirements with the customer requirements, previously mentioned, and evaluating the relationships between them [54].



				Column #	1	2	3	4
				Direction of Improvement: Minimize (▼), Maximize (▲), or Target (x)	X	X	X	X
Row #	Max Relationship Value in Row	Relative Weight	Weight / Importance	Quality Characteristics (a.k.a. "Functional Requirements" or "Hows") Demanded Quality (a.k.a. "Customer Requirements" or "Whats")	Data Loading	Data pre-processing	ML models generation	User-friendly UI development
1	9	20,0	20,0	Colorectal cancer diagnosis	Θ	Θ	Θ	O
2	9	20,0	20,0	Identification of features correlated with the diagnosis	Θ	Θ	Θ	O
3	9	20,0	20,0	Detection of effectiveness of treatment	Θ	Θ	Θ	O
4	9	10,0	10,0	Division of patients into clusters	Θ	Θ	Θ	O
5	9	10,0	10,0	Recommendation of treatment	Θ	Θ	Θ	O
6	9	10,0	10,0	Diagnosis of other diseases	Θ	Θ	Θ	O
7	9	10,0	10,0	User-friendly interface				Θ
				Target or Limit Value	Supported	Supported	Supported	Supported
				Difficulty (0=Easy to Accomplish, 10=Extremely Difficult)	3	5	8	6
				Max Relationship Value in Column	9	9	9	9
				Weight / Importance	810,0	810,0	810,0	360,0
				Relative Weight	29,0	29,0	29,0	12,9

Figure 14 – House of Quality.

Based on the results, it is possible to conclude that most of the customer requirements are strongly related to ML-associated tasks (data loading, data pre-processing and ML models generation), so that will be the focus of the development.

3.3 Creative Alternatives

In this phase, the goal is to identify alternatives. For healthcare professionals to access and use the developed ML models, it was proposed, as a complement to these dissertation objectives, the development of a web user interface. The frameworks that were considered for this development were React, Angular and Vue.js.

3.4 Analysis and Evaluation

The main objective of this phase is to analyse the alternatives identified in the previous step and choose one. Making that decision is a process that has the goal of finding the most efficient alternative, and for that, the Analytic Hierarchy Process (AHP) will be used [51]. In Figure 15 the decision to make, the different criteria and alternatives are presented in a DT.

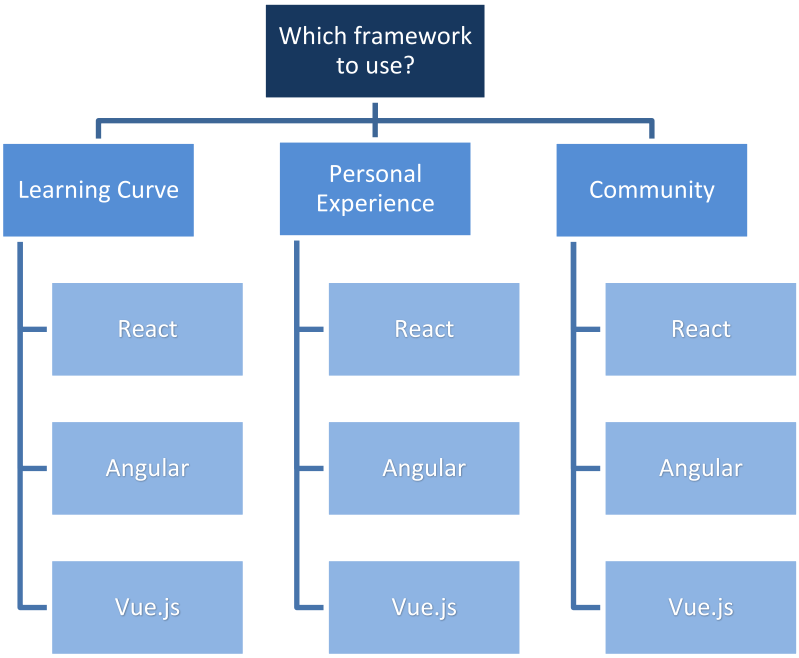


Figure 15 – Hierarchical DT for AHP.

The frameworks are going to be evaluated in three aspects: learning curve, personal experience, and community. The first one is associated with the complexity of the framework and the time and effort that an inexperienced user would have to learn the basics. Personal experience is the second one, as the more experienced someone is with a framework, the less effort and time it takes in the development process. Finally, the community aspect relates to the number of members of the community that are actively contributing to the framework´s growth and preventing it from becoming stale and outdated.

The next phase consists of a comparison of the criteria and alternatives. For that, a scale of numbers is used that shows different degrees of importance, demonstrating how many times an alternative is more important than another in a certain criterion [55].

Table 10 – Fundamental scale of absolute numbers [55].

Importance Level	Definition	Explanation
1	Equal importance	The two activities contribute equally to the objective.
3	Weak importance	The experience and judgement slightly favour one activity over the other.
5	Strong importance	The experience and judgement strongly favour one activity over another.
7	Very strong importance	The experience and judgement very strongly favour one activity over another.
9	Absolute importance	Evidence favouring one activity over the other is of the highest possible order of affirmation.
2, 4, 6, 8	Intermediate Values	Compromise condition between two definitions.

According to this scale, Table 11 presents the different criteria and their respective importance.

Table 11 – Comparison matrix between different criteria.

	Learning Curve	Personal Experience	Community
Learning Curve	1	$1/3$	2
Personal Experience	3	1	4
Community	$1/2$	$1/4$	1
Sum	$9/2$	$19/12$	7

Personal experience is the most important criterion, followed by the learning curve and community. The reasoning behind this decision was the reduced time and effort spent in the development process.

The last line of the matrix represents the sum of each column. To continue the AHP method, the data needs to be normalized, by dividing each value by the sum of the respective column. This is represented in Table 12.

Table 12 – Normalized comparison matrix between different criteria.

	Learning Curve	Personal Experience	Community
Learning Curve	$2/9$	$4/19$	$2/7$
Personal Experience	$2/3$	$12/19$	$4/7$
Community	$1/9$	$3/19$	$1/7$

With the normalized data, the priority vector is obtained with the value of the arithmetic mean of each line in the matrix. This vector identifies the order of importance of each criterion and is illustrated in Table 13.

Table 13 – Normalized comparison matrix between different criteria and respective priority vectors.

	Learning Curve	Personal Experience	Community	Priority Vector
Learning Curve	$^2/_9$	$^4/_{19}$	$^2/_7$	0.24
Personal Experience	$^2/_3$	$^{12}/_{19}$	$^4/_7$	0.62
Community	$^1/_9$	$^3/_{19}$	$^1/_7$	0.14

It is possible to see that personal experience is the most important criterion.

The next step of the AHP method is to verify the consistency of the priorities over large samples of completely random judgements [51]. For that, the consistency reason (CR) is calculated, which must be <0,1 to have consistent and trustworthy priorities. This is calculated by Equation (3.1).

$$CR = \frac{CI}{RI} \qquad \text{Equation (3.1)}$$

CI is the consistency index and RI is the random index. The values for CI are established by the National Laboratory of Oak Ridge, in the USA. Table 14 displays the values of RI according to the number of criteria [51].

Table 14 – RI values defined by the National Laboratory of Oak Ridge [46].

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

In this case, the value of RI is 0.58. The value of CI is calculated by Equation (3.2).

$$CI = \frac{\lambda_{max} - n}{n - 1} \qquad \text{Equation (3.2)}$$

Where λ_{max} is the eigenvalue of the array and n is the number of criteria evaluated.

Based on the previous priority matrix and priority vector, it is possible to calculate the λ_{max}, as shown below.

$$Priority\ matrix\ \times Priority\ vector\ \cong\ \lambda_{max}\ \times Priority\ vector\ \Rightarrow$$

$$\Rightarrow \begin{bmatrix} 1 & 0.33 & 2 \\ 3 & 1 & 4 \\ 0.50 & 0.25 & 1 \end{bmatrix} \times \begin{bmatrix} 0.24 \\ 0.62 \\ 0.14 \end{bmatrix} \cong \lambda_{max} \times \begin{bmatrix} 0.24 \\ 0.62 \\ 0.14 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0.72 \\ 1.90 \\ 0.42 \end{bmatrix} \cong \lambda_{max} \times \begin{bmatrix} 0.24 \\ 0.62 \\ 0.14 \end{bmatrix} \Leftrightarrow \lambda_{max} \cong 3.02$$

Replacing the value of λ_{max} in Equation (3.2), the following result is achieved:

$$CI = \frac{\lambda_{max} - n}{n - 1} \Rightarrow CI = \frac{3.02 - 3}{3 - 1} = \frac{0.02}{2} = 0.01$$

Finally, the values are replaced in Equation (3.1):

$$CR = \frac{CI}{RI} \Rightarrow\ CR = \frac{0.01}{0.58} = 0.02$$

The value of CR is less than 0.1, so it is possible to consider that the values of the priorities are consistent and reliable.

The next step consists of the construction of a comparison matrix for each criterion, considering each alternative [51]. The values considered for each alternative regarding the criterion learning curve and community were based on the comparison of the three frameworks made in [56]. The process of normalizing the data and obtaining the respective priority vectors is the same that was followed above. Table 15, Table 16, and Table 17 illustrate this process applied to the different criteria.

Table 15 – Initial matrix, normalized matrix and priority vector for the criterion learning curve.

Initial Matrix				Normalized Matrix				Priority Vector
	React	Angular	Vue.js		React	Angular	Vue.js	
React	1	2	$^{1}/_{2}$	React	$^{2}/_{7}$	$^{1}/_{3}$	$^{3}/_{11}$	0.10
Angular	$^{1}/_{2}$	1	$^{1}/_{3}$	Angular	$^{1}/_{7}$	$^{1}/_{6}$	$^{2}/_{11}$	0.05
Vue.js	2	3	1	Vue.js	$^{4}/_{7}$	$^{1}/_{2}$	$^{6}/_{11}$	0.18
Sum	$^{7}/_{2}$	6	$^{11}/_{6}$					

Table 16 – Initial matrix, normalized matrix and priority vector for the criterion personal experience.

Initial Matrix				Normalized Matrix				Priority Vector
	React	Angular	Vue.js		React	Angular	Vue.js	
React	1	2	3	React	$^{6}/_{11}$	$^{4}/_{7}$	$^{1}/_{2}$	0.18
Angular	$^{1}/_{2}$	1	2	Angular	$^{3}/_{11}$	$^{2}/_{7}$	$^{1}/_{3}$	0.10
Vue.js	$^{1}/_{3}$	$^{1}/_{2}$	1	Vue.js	$^{2}/_{11}$	$^{1}/_{7}$	$^{1}/_{6}$	0.05
Sum	$^{11}/_{6}$	$^{7}/_{2}$	6					

Table 17 – Initial matrix, normalized matrix and priority vector for the criterion community.

Initial Matrix				Normalized Matrix				Priority Vector
	React	**Angular**	**Vue.js**		**React**	**Angular**	**Vue.js**	
React	1	1	3	**React**	$^{3}/_{7}$	$^{3}/_{7}$	$^{3}/_{7}$	0.14
Angular	1	1	3	**Angular**	$^{3}/_{7}$	$^{3}/_{7}$	$^{3}/_{7}$	0.14
Vue.js	$^{1}/_{3}$	$^{1}/_{3}$	1	**Vue.js**	$^{1}/_{7}$	$^{1}/_{7}$	$^{1}/_{7}$	0.05
Sum	$^{7}/_{3}$	$^{7}/_{3}$	7					

The replacement of the obtained values in the decision tree is illustrated in Figure 16.

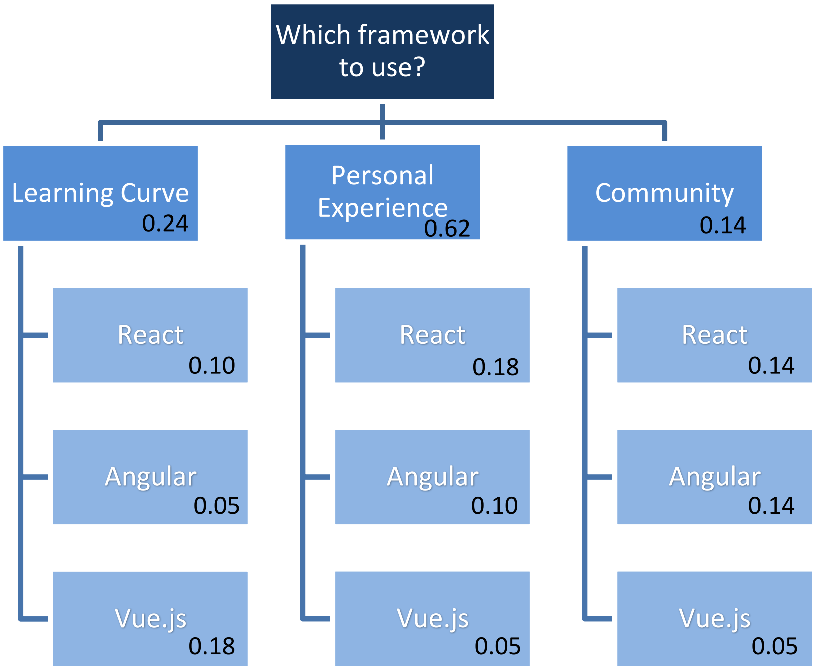


Figure 16 – Hierarchical decision tree with associated values.

From those values, it is possible to calculate the priority vector, by multiplying the matrix of the values of each alternative with the matrix of values of each criterion. The results are shown below.

$$\begin{bmatrix} 0.10 & 0.18 & 0.14 \\ 0.05 & 0.10 & 0.14 \\ 0.18 & 0.05 & 0.05 \end{bmatrix} \times \begin{bmatrix} 0.24 \\ 0.62 \\ 0.14 \end{bmatrix} = \begin{bmatrix} 0.1552 \\ 0.0936 \\ 0.0812 \end{bmatrix}$$

According to the AHP method, React is the best alternative with a value of 0.1552, followed by Angular with 0.0936 and Vue.js with 0.0812.

3.5 Implementation

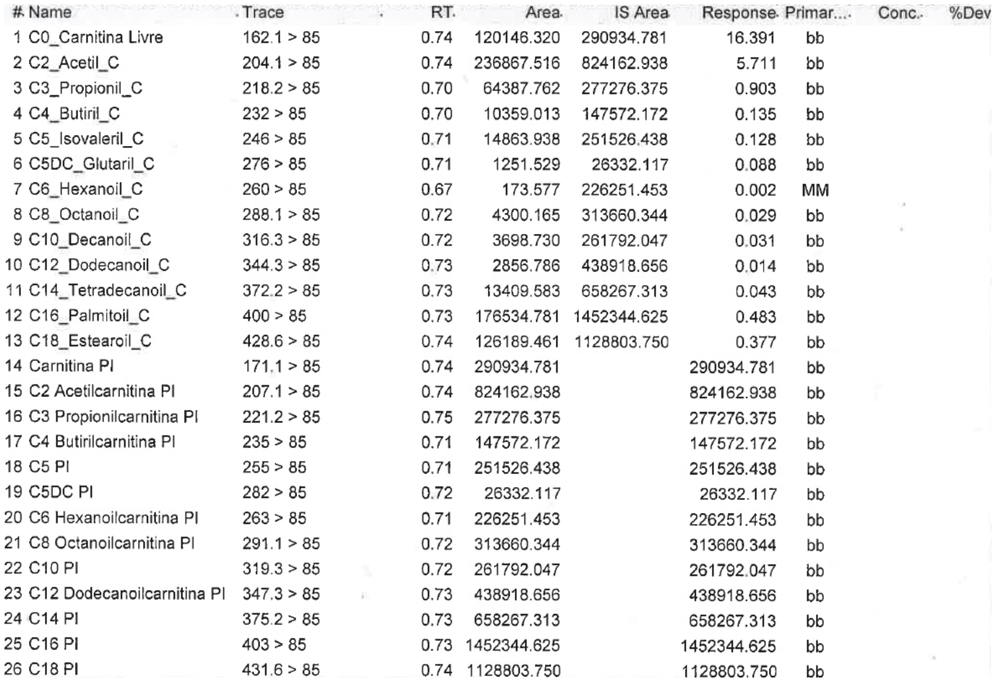
The last stage of VA consists of the implementation of the product. The development of this product will have its focus on ML-related tasks, and, in the end, the result will be delivered to the customers as a web application. Hosting plans need to be analysed to make sure that the software is accessible to everyone and has the necessary resources to work with.

4 Data Analysis and Design

This chapter provides an analysis of the data used, as well as an overview of the methods used to transform it in a way that is suitable to be consumed by ML models. Finally, the design of the prototype developed to carry out the experiments is specified.

4.1 Data Understanding

Initially, the data provided by healthcare professionals was spread across multiple files, containing the results of about 800 acylcarnitine profile exams of real patients from October 2019 to June 2022. Figure 17 illustrates an example of the results of an exam.



#	Name	Trace	RT.	Area	IS Area	Response	Primar....	Conc.	%Dev
1	C0_Carnitina Livre	162.1 > 85	0.74	120146.320	290934.781	16.391	bb		
2	C2_Acetil_C	204.1 > 85	0.74	236867.516	824162.938	5.711	bb		
3	C3_Propionil_C	218.2 > 85	0.70	64387.762	277276.375	0.903	bb		
4	C4_Butiril_C	232 > 85	0.70	10359.013	147572.172	0.135	bb		
5	C5_Isovaleril_C	246 > 85	0.71	14863.938	251526.438	0.128	bb		
6	C5DC_Glutaril_C	276 > 85	0.71	1251.529	26332.117	0.088	bb		
7	C6_Hexanoil_C	260 > 85	0.67	173.577	226251.453	0.002	MM		
8	C8_Octanoil_C	288.1 > 85	0.72	4300.165	313660.344	0.029	bb		
9	C10_Decanoil_C	316.3 > 85	0.72	3698.730	261792.047	0.031	bb		
10	C12_Dodecanoil_C	344.3 > 85	0.73	2856.786	438918.656	0.014	bb		
11	C14_Tetradecanoil_C	372.2 > 85	0.73	13409.583	658267.313	0.043	bb		
12	C16_Palmitoil_C	400 > 85	0.73	176534.781	1452344.625	0.483	bb		
13	C18_Estearoil_C	428.6 > 85	0.74	126189.461	1128803.750	0.377	bb		
14	Carnitina PI	171.1 > 85	0.74	290934.781		290934.781	bb		
15	C2 Acetilcarnitina PI	207.1 > 85	0.74	824162.938		824162.938	bb		
16	C3 Propionilcarnitina PI	221.2 > 85	0.75	277276.375		277276.375	bb		
17	C4 Butirilcarnitina PI	235 > 85	0.71	147572.172		147572.172	bb		
18	C5 PI	255 > 85	0.71	251526.438		251526.438	bb		
19	C5DC PI	282 > 85	0.72	26332.117		26332.117	bb		
20	C6 Hexanoilcarnitina PI	263 > 85	0.71	226251.453		226251.453	bb		
21	C8 Octanoilcarnitina PI	291.1 > 85	0.72	313660.344		313660.344	bb		
22	C10 PI	319.3 > 85	0.72	261792.047		261792.047	bb		
23	C12 Dodecanoilcarnitina PI	347.3 > 85	0.73	438918.656		438918.656	bb		
24	C14 PI	375.2 > 85	0.73	658267.313		658267.313	bb		
25	C16 PI	403 > 85	0.73	1452344.625		1452344.625	bb		
26	C18 PI	431.6 > 85	0.74	1128803.750		1128803.750	bb		

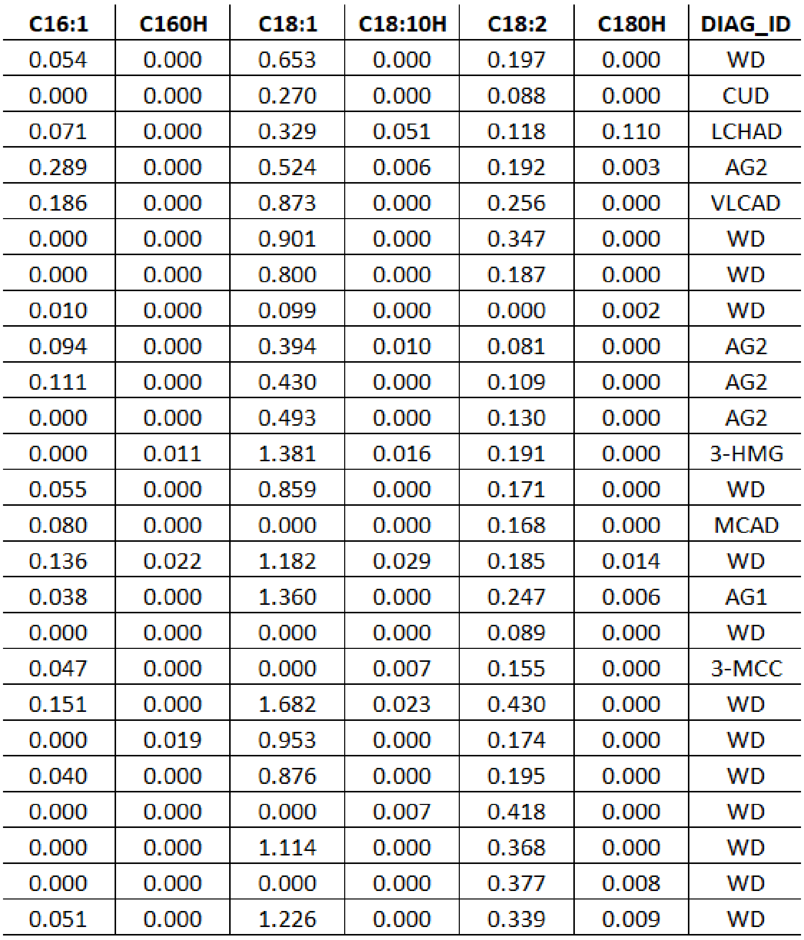
Figure 17 – Example of the results of an acylcarnitine profile exam.

From this data, the “RT” (retention time) and “Response” column values were collected. The first one is used to validate the response value, by comparing the retention time with a standard that was previously agreed upon with healthcare professionals. When the value is not valid, the response is considered 0.

The other files provided contained information about the patient sex, age, and diagnosis. All this information was used to generate a dataset prepared to be used by ML models, as explained in the next section.

4.1.1 Dataset Construction

Manually collecting and merging all the provided data would be a time-consuming task. For that reason, this process was automated with the development of a software application, whose design and implementation details are explained in section 4.3, by validating and compiling all the information in one dataset, ready to be consumed by ML algorithms. This dataset contains information about each acylcarnitine response value, as well as the sex, age and diagnosis of the patient. Figure 18 shows a partial view of the dataset.



C16:1	C160H	C18:1	C18:10H	C18:2	C180H	DIAG_ID
0.054	0.000	0.653	0.000	0.197	0.000	WD
0.000	0.000	0.270	0.000	0.088	0.000	CUD
0.071	0.000	0.329	0.051	0.118	0.110	LCHAD
0.289	0.000	0.524	0.006	0.192	0.003	AG2
0.186	0.000	0.873	0.000	0.256	0.000	VLCAD
0.000	0.000	0.901	0.000	0.347	0.000	WD
0.000	0.000	0.800	0.000	0.187	0.000	WD
0.010	0.000	0.099	0.000	0.000	0.002	WD
0.094	0.000	0.394	0.010	0.081	0.000	AG2
0.111	0.000	0.430	0.000	0.109	0.000	AG2
0.000	0.000	0.493	0.000	0.130	0.000	AG2
0.000	0.011	1.381	0.016	0.191	0.000	3-HMG
0.055	0.000	0.859	0.000	0.171	0.000	WD
0.080	0.000	0.000	0.000	0.168	0.000	MCAD
0.136	0.022	1.182	0.029	0.185	0.014	WD
0.038	0.000	1.360	0.000	0.247	0.006	AG1
0.000	0.000	0.000	0.000	0.089	0.000	WD
0.047	0.000	0.000	0.007	0.155	0.000	3-MCC
0.151	0.000	1.682	0.023	0.430	0.000	WD
0.000	0.019	0.953	0.000	0.174	0.000	WD
0.040	0.000	0.876	0.000	0.195	0.000	WD
0.000	0.000	0.000	0.007	0.418	0.000	WD
0.000	0.000	1.114	0.000	0.368	0.000	WD
0.000	0.000	0.000	0.000	0.377	0.008	WD
0.051	0.000	1.226	0.000	0.339	0.009	WD

Figure 18 – Dataset partial view.

4.1.2 Dataset Features

There is a total of 35 features in the dataset, 33 of them related to acylcarnitine response values measured in μmol/L, the remaining being the patient´s sex and age. Table 18 displays all the features used, as well as their possible values.

Table 18 – Dataset Features.

Feature Description	Feature Abbreviation	Possible Values
Free Carnitine	CARNITINE	Positive real number
Acetylcarnitine	C2	Positive real number
Propionylcarnitine	C3	Positive real number
Butyrylcarnitine Isobutyrylcarnitine	C4	Positive real number
Tiglylcarnitine/3-Methylcrotonylcarnitine	C5	Positive real number
Glutarylcarnitine	C5DC	Positive real number
Hexanoylcarnitine	C6	Positive real number
Octanoylcarnitine	C8	Positive real number
Decanoylcarnitine	C10	Positive real number
Dodecanoylcarnitine	C12	Positive real number
Tetradecenoylcarnitine	C14	Positive real number
Hexadecanoylcarnitine	C16	Positive real number
Octadecanoylcarnitine	C18	Positive real number
Malonylcarnitine	C3DC	Positive real number
3-Hydroxybutyrylcarnitine/3-Hydroxyisobutyrylcarnitine	C40H	Positive real number
Methylmalonylcarnitine/Succinylcarnitine	C4DC	Positive real number
Tiglylcarnitine/3-Methylcrotonylcarnitine	C5:1	Positive real number
3-Hydroxyisovalerylcarnitine/3-Hydroxy-2-methylbutyrylcarnitine	C50H	Positive real number
3-Methylglutarylcarnitine	C6DC	Positive real number
Octenoycarnitine	C8:1	Positive real number
Decenoylcarnitine	C10:1	Positive real number
Decadienoylcarnitine	C10:2	Positive real number
Dodecenoylcarnitine	C12:1	Positive real number

Tetradecenoylcarnitine	C14:1	Positive real number
3-Hydroxytetradecenoylcarnitine	C14:10H	Positive real number
Tetradecadienoylcarnitine	C14:2	Positive real number
3-Hydroxytetradecenoylcarnitine	C140H	Positive real number
Hexadecenoylcarnitine	C16:1	Positive real number
3-Hydroxyhexadecanoylcarnitine	C160H	Positive real number
Octadecenoylcarnitine	C18:1	Positive real number
3-Hydroxyoctadecenoylcarnitine	C18:10H	Positive real number
Octadecadienoylcarnitine	C18:2	Positive real number
3-Hydroxyoctadecanoylcarnitine	C180H	Positive real number
Patient sex	SEX	1 = Male, 0 = Female
Patient age when the sample was collected	AGE	Positive integer

4.1.3 Dataset Classes

The dataset classes correspond to all distinct diagnoses that were given to the patients. Table 19 displays all the classes present, as well as their relative and absolute frequency.

Table 19 – Dataset classes.

Classes	Code	Absolute Frequency	Relative Frequency (%)
3-Hydroxy-3-methylglutaric aciduria	3-HMG	13	1.62
3-Methylcrotonylglycinuria	3-MCC	13	1.62
ArgininoSuccinic Aciduria	AAS	5	0.62
Isovaleric acidaemia	AC_IVA	1	0.12
Glutaric aciduria type I	AG1	19	2.37
Glutaric aciduria type II	AG2	16	1.99
Methylmalonic aciduria	AMM	14	1.75
Methylmalonic aciduria due to methylmalonyl-CoA mutase	AMM_MUT	8	1.00
Cobalamin C deficiency	CBL_C	2	0.25
Cystathionine Beta-Synthase deficiency	CBS	2	0.25
Coenzyme Q10 deficiency	CM_DEF_Q10	1	0.12
Mitochondrial cytopathy – LYRM4	CM_LYRM4	5	0.62
Carnitine palmitoyltransferase II deficiency	CPT II	6	0.75

RC	RC	136	16.98
Carnitine Transporter deficiency	CUD	8	1.00
Intracellular B12 metabolism deficit	D.MET B12	6	0.75
Citrin deficiency – Citrullinemia type II	D_CITR	1	0.12
Glycerol kinase deficit	GK	2	0.25
Glycogen storage disease type V	GSD5	2	0.25
Hypermethioninemia	HIPER_MET	1	0.12
Long-chain 3-hydroxyacyl-CoA dehydrogenase deficiency	LCHAD	18	2.25
Medium-chain acyl-CoA dehydrogenase deficiency	MCAD	59	7.37
Leucinosis	MSUD	1	0.12
Without diagnosis	WD	430	53.68
Ornithine transcarbamylase deficiency	OTC	8	1.00
Pyruvate dehydrogenase deficiency	PDH	2	0.25
Short-chain acyl-CoA dehydrogenase deficiency	SCAD	3	0.37
Trimethyllysine dioxygenase deficiency	TMLHE	3	0.37
Very long-chain acyl-CoA dehydrogenase deficiency	VLCAD	16	1.99

Regarding the patients with a diagnosis of RC, there is a distinction in the moments in which the sample was collected, according to the different stages of the disease, as well as the types of treatment the patient has been through. Table 20 describes these moments.

Table 20 – RC diagnosis moments.

Moment	Moment description	Absolute Frequency	Relative Frequency (%)
M0	Before chemoradiotherapy	64	47.06
M1	After chemoradiotherapy	28	20.59
M2	After surgery	41	27
M3	Relapse	1	0.8
M4	After relapse treatment	2	1.5

As the main objective of this dissertation is to test ML models on patients who were not previously diagnosed with RC, only the cases in the M0 moment were used for training and testing of the models. Upon agreement with healthcare professionals, it was decided to use the other cases to check the predictions made by those models, evaluating possible differences between each moment.

4.2 Data Preparation

The success of ML algorithms relies heavily on the quality of data that they use. Issues like noisy data, redundant data and missing values will negatively affect the performance of the ML models [57]. For that reason, this section will describe the data preprocessing methods that were used to address those issues.

4.2.1 Data Cleaning

Data cleaning is a preprocessing method used to detect missing values, outliers, inconsistencies and noise in the data [58].

Data Inconsistencies

While building the dataset, as described in section 4.2, the following inconsistencies were observed:

- Samples missing half or more acylcarnitine response values;
- Samples with no diagnostic;
- Acylcarnitine profile exams that were repeated.

For the first two problems, as they represent a small portion of the total cases, it was decided to discard all samples. Regarding the last one, after discussion with the healthcare professionals, it was concluded that a repetition only happens when there are issues with the first analysis. For that reason, only the repetitions were considered.

Classes Filtering

The dataset contained diseases that were only diagnosed in a few patients, making it difficult for a ML model to correctly identify them. For that reason, it was decided to only consider the diseases with at least 8 cases, which corresponds to approximately 1% of the total ones. The following classes were removed from the dataset: AAS, AC_IVA, CBL_C, CBS, CIT_I, CM, CM_DEF_Q10, CM_LYRM4, CPT II, D.MET B12, D_CITR, GK, GSD5, HIPER_MET, MSUD, PDH, SCAD and TMLHE.

4.2.2 Data Balancing

Data imbalance occurs when there is an unequal distribution of the target classes, negatively influencing the performance of the algorithms, by making them more biased towards the majority class [59].

In the case of this study, the number of samples classified as not having a diagnosis is much higher than the others. To make the dataset more balanced, some sampling techniques were tested, to see if their use would be beneficial to the performance of the ML models. The next sections will describe the techniques that were tested.

Undersampling

Undersampling is a method whose principle is to reduce the number of samples of the majority class. This can be achieved by doing the reduction randomly, called random undersampling (RU), or by applying statistical knowledge, called informed undersampling [60].

Oversampling

In oversampling, new samples are generated from the minority class. This is done by randomly replicating existing samples, called random oversampling (RO), or by generating artificial samples, called synthetic oversampling (SO) [60].

4.2.3 Data Normalization

Data normalization is used to transform feature values to a similar range. Features can often have a large difference between the maximum and minimum values (e.g., 0.01 and 1000), which can be a problem for some ML algorithms that use distance measures, such as KNN, by making the feature with a larger range dominate over the other ones [57]. The data normalization techniques that were tested will be explained next.

Min-max Normalization

Min-max normalization has the objective of scaling the numerical values of a feature to a specified range (e.g., [0,1]). The new value is calculated according to Equation (4.1), where v is the old feature value and v' is the new one [57].

$$v' = \frac{v - min}{max - min}(new\_max - new\_min\,) + new\_min \qquad \text{Equation (4.1)}$$

Z-score Normalization

Z-score normalization is the process of transforming feature values so that they have a mean equal to 0 and a standard deviation of 1. The new value is calculated using Equation (4.2), where v is the old feature value, v' is the new one, $\bar{x}$ is the mean and σ is the standard deviation [57].

$$v' = \frac{v - \bar{x}}{\sigma} \qquad \text{Equation (4.2)}$$

4.2.4 Feature Selection

Feature selection is a technique used to reduce the dimension of the dataset, by only selecting the most important features and removing irrelevant or redundant ones, as they provide no value for the output prediction. This can lead to better performance and reduced execution times [61]. The feature selection techniques that were tested will be introduced in the next sections.

Pearson Correlation (PC)

Pearson Correlation method assigns a value between -1 and 1, 0 indicating that there is no correlation between features, 1 a total positive correlation and -1 a total negative correlation. This mean that with a positive correlation whenever a feature increases, the correlated feature will also increase. With a negative correlation, whenever a feature increases, the correlated feature will decrease [62].

This method was applied to the dataset features, selecting the ones with a correlation score equal to or above 0.75. For each pair of correlated features, the removal of one of them should not impact the performance. This assumption will be tested in chapter 5. The list of correlated features, according to the application of PC, are shown in Table 21.

Table 21 – PC scores.

Correlated features	PC score
C6 and C8	0.79
C18:10H and C180H	0.75
C160H and C180H	0.75

Univariate Feature Selection (UFS)

UFS selects the best features based on univariate statistical tests, comparing each feature with the target variable, to check if there is any statistically significant relationship between them. Each feature is assigned a score and only the best ones are selected [63]. It was opted to test this technique by selecting the top 25 features of the dataset, from a total of 35.

Recursive Feature Elimination (RFE)

The objective of RFE is to select features by recursively considering smaller sets of the total features. Initially, a ML algorithm is trained with all features, giving an importance to each of them. The least important features are removed from the current set and this process is repeated until the optimal number of features is reached [63].

4.3 Design

This section will present the design and implementation details of the software application developed to treat the data and build ML models to run predictions on that data.

4.3.1 Technologies

The prototype was developed with the Python programming language, version 3.10.2, using the following main libraries: Scikit-learn (version 1.1.2); Imbalanced-learn (version 0.9.0); Pandas (version 1.4.1); Numpy (version 1.22.2).

4.3.2 Use Cases Diagram

As previously mentioned, it was necessary to compile and validate the data provided by healthcare professionals, to create a dataset ready to be consumed by ML models. For that reason, the prototype was developed taking into consideration two main use cases: compiling patient data into a suitable dataset, as well as using ML models to run predictions on a dataset. Figure 19 displays the use case diagram for the developed prototype.

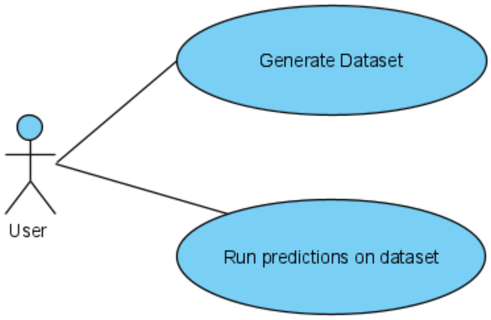


Figure 19 – Prototype use case diagram.

4.3.3 Package Diagram

A package diagram illustrates the way a system is structured, by organizing high-level system elements into packages [64]. The prototype package diagram is illustrated in Figure 20.

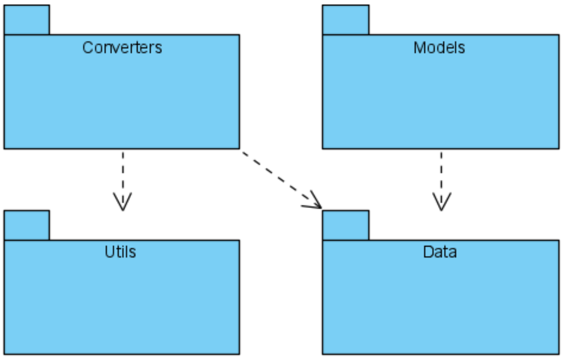


Figure 20 – Prototype package diagram.

The “Data” package is where all the files necessary to run the prototype are stored, from the raw unprocessed data to the final datasets. The “Converters” package is responsible to convert the unprocessed data into a dataset that is ready to be consumed by ML models, using functions present in the “Utils” package. Finally, the “Models” package is where all the developed ML models are located.

4.3.4 Deployment Diagram

A deployment diagram is used to specify the configuration of the system in runtime, as well as its components and interactions [65]. The prototype deployment diagram is presented in Figure 21.

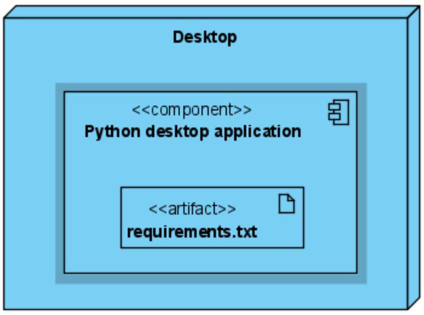


Figure 21 – Prototype deployment diagram.

The prototype is a desktop application, that can be run through a command line prompt. The list of dependencies necessary to run the software is specified in the artifact “requirements.txt”.

4.3.5 Proposed System Architecture

Although not the focus of this dissertation, it was discussed that this investigation can lead to an implementation of a decision support system, to help healthcare professionals in the prognosis and treatment of CRC patients.

For that reason, it is presented in Figure 22, a proposal for the architecture to be used in the development of the decision support system.

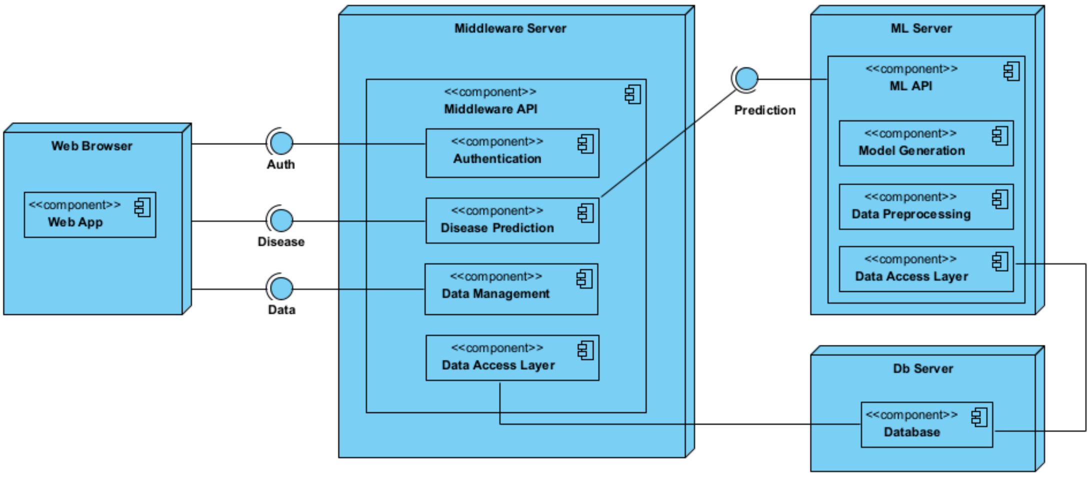


Figure 22 – Proposed system architecture.

The main part of the proposed architecture is the "ML Server", which is a server where all the components necessary to handle ML-related tasks are. This server is isolated from the rest of the system for decoupling purposes, so that other future external components can communicate with it, via an API described as "ML API". This API will be responsible to receive incoming requests and handle all the loading of data, data preprocessing and model generation tasks.

The system will be accessible to the users by a web application, that communicates with an API, described as "Middleware API", responsible to handle different requests: authentication, disease prediction and data management. For disease prediction tasks, requests to "ML API" will be made.

The "Database" component is where all the data is stored, such as users and respective credentials, as well as the data used for the training of algorithms and generation of ML models.

5 Experimentation and Evaluation

This chapter details the experiments made, as well as an evaluation of the results obtained.

5.1 Experiments

The experiments were performed on a laptop with an Intel® Core™ i7-9750H CPU, 16 GB of RAM, running a Windows 11 Pro 64-bit operating system. To conduct said experiments, different Python libraries were used, containing implementations of ML algorithms and data preprocessing methods. Table 22 displays the algorithms and data preprocessing methods used, as well as the classes where they are implemented and respective library.

Table 22 – Libraries used in the experiments.

Algorithm/Data preprocessing method	Class	Library
LR	LogisticRegression	Scikit-learn
SVM	SVC	Scikit-learn
DT	DecisionTreeClassifier	Scikit-learn
RF	RandomForestClassifier	Scikit-learn
NB	GaussianNB	Scikit-learn
KNN	KNeighborsClassifier	Scikit-learn
ANN	MLPClassifier	Scikit-learn
GB	XGBClassifier	XGBoost
Z-score normalization	StandardScaler	Scikit-learn
Min-max normalization	MinMaxScaler	Scikit-learn
RU	RandomUnderSampler	Imbalanced-learn
RO	RandomOverSampler	Imbalanced-learn
SO	SMOTE	Imbalanced-learn
UFS	SelectKBest	Scikit-learn
RFE	RFECV	Scikit-learn

Based on the data available, three experiments were conducted (only using M0 RC cases), applying different techniques to ML models, which will be presented in the next sections.

5.1.1 Binary-class Classification of Rectal Cancer (BCRC)

The goal of this experiment is to test different ML models in the task of identifying if a patient has RC, based on the acylcarnitine profile exam results. Since all patients in the dataset diagnosed with that disease are adults, only samples of patients 18 years of age or more were considered. The distribution of the classes used in this experiment is illustrated in Figure 24.

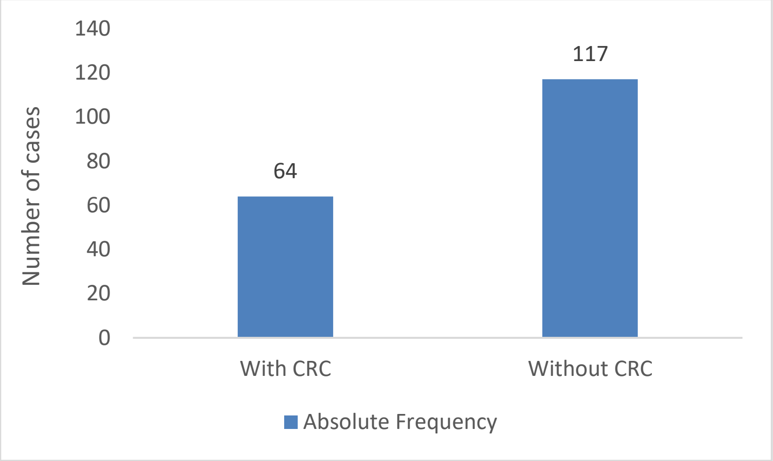


Figure 23 - Distribution of classes in the BCRC experiment.

5.1.2 Multi-class Classification of Diseases (MCD)

The objective of this experiment is to test how well a ML model can identify which disease the patient has, based on the acylcarnitine profile exam results. The distribution of the classes used for this experiment is illustrated in Figure 24.

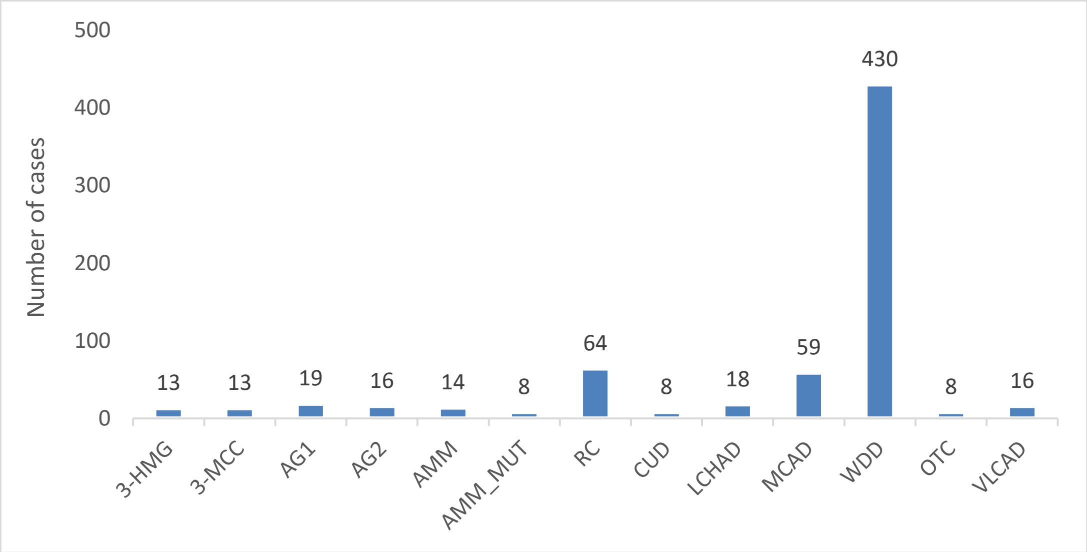


Figure 24 – Distribution of classes in the MCD experiment.

5.1.3 Binary-class Classification of Disease (BCD)

In this experiment, the goal is to test ML models by predicting if a patient has a disease or not, based on the acylcarnitine profile exam results. The distribution of the classes used for this experiment is illustrated in Figure 25.

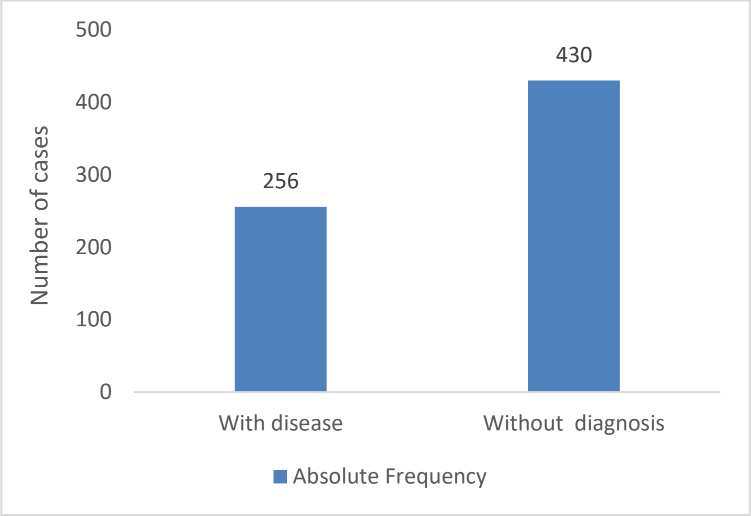


Figure 25 - Distribution of classes in the Binary classification of disease experiment.

5.2 Evaluation Methods and Metrics

In this section, two different methods to split the dataset into training and test data are presented, followed by the definition of the metrics used.

5.2.1 Dataset Split

Using the whole dataset to train and evaluate a ML model results in an uncertainty of how said model would behave with new unseen observations. Two main problems can arise, according to [66]:

- Underfitting, which occurs when the model is poorly adjusted to the data, with high error both with training and test data;
- Overfitting, a problem that happens when the model is well adjusted to the training data but performs poorly on new unseen data.

To avoid having these issues, it was decided to split the data into different sets used for training and testing purposes. To achieve this, different methods can be used. An overview of some of those methods is presented in the next sections, as well as the decision made for which one to use.

Holdout

The holdout method is one of the simplest data resampling techniques. In this method, the data is separated into two sets, called the training and test set. The first one is used to train the ML models, while the second one is used to evaluate the performance of the models on unseen data. Figure 26 illustrates this process.

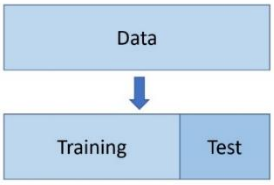


Figure 26 – Holdout method.

K-fold Cross-Validation

One of the main issues with the use of the holdout method is that the number of samples used for the training is reduced and the results obtained can be dependent on the random selection of the training and test set. Considering the small sample size of the dataset and the imbalance between classes, it was decided to use the K-Fold-Cross-Validation method, which will be explained next.

In K-fold Cross-Validation the data is partitioned into K smaller sets, called folds. The ML model is trained with K-1 training folds and evaluated with the remaining one, used as a test fold. This process is repeated K times until all folds are used for testing exactly once, ensuring all data is used for testing purposes. After all the iterations are complete, the results are combined, usually by averaging them [66].

Typical values used for K are 5 and 10 [66]. Given the reduced size of the data used, as well as the fact that some diagnosed diseases have less than 10 cases, it was decided to use 5 as the value of K. Figure 27 illustrates the behaviour of K-fold Cross-Validation with K=5.

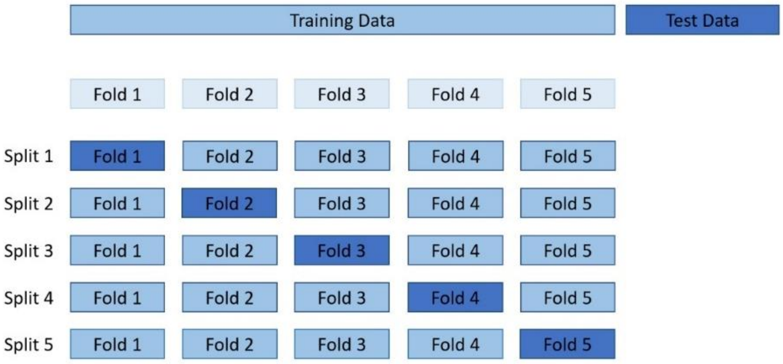


Figure 27 – K-fold Cross-Validation for K=5 (adapted from [67]).

5.2.2 Performance Metrics

In this section, the performance metrics used to analyse the performance of the ML models will be described.

Confusion Matrix

The metrics used are based on the confusion matrix, a framework where true positives (TP) are the positive cases correctly identified; true negatives (TN) are the negative cases correctly identified. Similarly, false positives (FP) and false negatives (FN) are the positive and negative cases that were not correctly identified. This is represented in Table 23.

Table 23 – Confusion matrix.

Predicted values	Actual values	Definition
Yes	Yes	TP
Yes	No	FP
No	No	TN
No	Yes	FN

Accuracy (ACC)

ACC is the number of correct predictions divided by the number of total cases, as presented by Equation (6.1).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$ Equation (6.1)

F1-score

F1-score is the harmonic mean of precision and recall, being:

- Precision: The number of true positive predictions divided by the number of all cases predicted as positive. It is expressed by Equation (6.2).

$$Precision = \frac{TP}{TP + FP}$$ Equation (6.2)

- Recall: The number of true positive predictions divided by all the results that should have been identified as positive. It is given by the Equation (6.3).

$$Sensitivity = Recall = True\ positive\ rate = \frac{TP}{TP + FN}$$ Equation (6.3)

Considering the above, F1-score is calculated by Equation (6.4).

$$F1\,score = \frac{2\ \times TP}{2\ \times TP + FN + FP} \qquad \text{Equation (6.4)}$$

Metrics used in each experiment

Since the BCRC and BCD experiments relate to binary-class classification problems, it is important to understand how well the models are at predicting the class with fewer cases (patients with RC/disease). Due to the class imbalance, the ACC metric is not sufficient, since high accuracy values can be achieved by just predicting that cases belong to the most represented class.

For that reason, in the above-mentioned experiments, the F1-score metric is used alongside the ACC. To achieve a high F1-score, both precision and recall metrics must be high. A high number of FP and/or FN will negatively influence the score.

Apart from the ACC and F1-score metrics, the execution time (ET) to train the models is also used, measured in seconds. If two models have similar ACC and F1-score results, a lower ET can be the differential factor.

5.3 Evaluation

In this section, the results obtained from the experiments will be presented in four parts. The first two demonstrate the effectiveness of different data preprocessing techniques applied to several ML algorithms; the third has the objective of optimizing the parameters of the algorithms and, in the last one, the models are validated with new cases of RC patients.

5.3.1 Data normalization and balancing results

The first set of tests had the goal to evaluate the best algorithms to use for each experiment, as well as verifying if the use of data normalization and balancing techniques would produce performance improvements. Table 24, Table 25 and Table 26 present the results for each experiment. The “Control” column displays the results that each algorithm had without the use of any data balancing and normalization technique. For the ACC and F1-score metrics, it is presented the standard deviation of the results between the 5 runs of K-fold Cross-Validation.

Table 24 – Data normalization and balancing results for the MCD experiment.

Algorithm	Control	Z-score normalization	Min-max normalization	RU	RO	SO
LR	ACC: 81.05% (± 0.02) ET: 0.19	ACC: 84.84% (± 0.01) ET: 0.16	ACC: 79.16% (± 0.03) ET: 0.11	ACC: 38.77% (± 0.04) ET: 0.13	ACC: 53.05 % (± 0.06) ET: 1.14	ACC: 54.51% (± 0.04) ET: 1.59
SVM	ACC: 2.02% (± 0.02) ET: 0.12	ACC: 81.05% (± 0.02) ET: 0.17	ACC: 81.93% (±0.03) ET: 0.14	ACC: 25.67% (± 0.08) ET: 0.05	ACC: 33.96% (± 0.03) ET: 2.57	ACC: 39.36% (± 0.04) ET: 2.66
DT	ACC: 80.90% (±0.01) ET: 0.07	ACC: 80.76% (± 0.01) ET: 0.08	ACC: 80.90% (± 0.01) ET: 0.08	ACC: 37.47% (± 0.04) ET: 0.04	ACC: 79.89% (± 0.03) ET: 0.17	ACC: 76.10% (± 0.04) ET: 0.86
RF	ACC: 87.61% (± 0.02) ET: 1.02	ACC: 87.47% (± 0.01) ET: 1.00	ACC: 87.61% (± 0.02) ET: 1.09	ACC: 50.29% (± 0.04) ET: 0.59	ACC: 87.90% (± 0.02) ET: 1.98	**ACC: 88.64% (± 0.03)** **ET: 6.32**
NB	ACC: 38.78% (± 0.03) ET: 0.02	ACC: 31.06% (± 0.03) ET: 0.03	ACC: 31.79% (± 0.03) ET: 0.03	ACC: 28.72% (± 0.04) ET: 0.05	ACC: 36.45% (± 0.03) ET: 0.06	ACC: 41.70% (± 0.03) ET: 0.32
KNN	ACC: 74.06% (± 0.02) ET: 0.19	ACC: 81.78% (+/ -0.01) ET: 0.19	ACC: 79.01% (± 0.02) ET: 0.20	ACC: 25.51% (± 0.03) ET – 0.21	ACC: 60.21% (± 0.01) ET: 0.33	ACC: 53.07% (± 0.04) ET: 0.37
ANN	ACC: 83.68% (± 0.04) ET: 4.24	ACC: 85.72% (± 0.01) ET: 4.29	ACC: 84.99% (± 0.01) ET: 3.89	ACC: 37.17% (± 0.04) ET – 1.18	ACC: 84.70% (± 0.02) ET: 31.23	ACC: 84.85% (± 0.04) ET: 31.38
GB	ACC: 87.91% (± 0.02) ET: 2.34	ACC: 87.91% (± 0.02) ET: 2.07	ACC: 87.91% (± 0.02) ET: 2.10	ACC: 48.69% (± 0.04) ET: 1.45	**ACC: 88.34% (± 0.02)** **ET: 6.69**	ACC: 86.59% (± 0.03) ET: 10.0

Table 25 – Data normalization and balancing results for the BCD experiment.

Algorithm	Control	Z-score normalization	Min-max normalization	RU	RO	SO
LR	ACC: 82.08% (± 0.04) F1: 72.90% (± 0.06) ET: 0.23	ACC: 85.43% (± 0.02) F1: 78.84% (± 0.03) ET: 0.21	ACC: 81.05% (± 0.03) F1: 70.04% (± 0.05) ET: 0.14	ACC: 81.35% (± 0.03) F1: 74.49% (± 0.04) ET – 0.31	ACC: 81.93% (± 0.03) F1: 74.74% (± 0.04) ET: 0.31	ACC: 82.22% (± 0.02) F1: 75.54% (± 0.03) ET: 0.50
SVM	ACC: 73.18% (± 0.01) F1: 46.78% (± 0.03) ET: 0.09	ACC: 83.68% (± 0.03) F1: 76.10% (± 0.04) ET: 0.11	ACC: 82.95% (± 0.03) F1: 74.11% (± 0.05) ET: 0.11	ACC: 73.03% (± 0.02) F1: 64.82% (± 0.02) ET: 0.10	ACC: 73.77% (± 0.04) F1: 63.09% (± 0.06) ET: 0.19	ACC: 73.03% (± 0.02) F1: 66.10% (± 0.04) ET: 0.23
DT	ACC: 82.66% (± 0.04) F1: 77.31% (± 0.05) ET: 0.06	ACC: 82.66% (± 0.04) F1: 77.31% (± 0.05) ET: 0.09	ACC: 82.66% (± 0.04) F1: 77.31% (± 0.05) ET: 0.08	ACC: 82.66% (± 0.03) F1: 78.80% (± 0.03) ET: 0.07	ACC: 84.41% (± 0.02) F1: 78.81% (± 0.03) ET: 0.08	ACC: 81.93% (± 0.02) F1: 76.53% (± 0.02) ET: 0.12
RF	ACC: 88.20% (± 0.03) F1: 83.45% (± 0.04) ET – 0.85	ACC: 88.20% (± 0.03) F1: 83.45% (± 0.04) ET: 0.95	ACC: 88.20% (± 0.03) F1: 83.45% (± 0.04) ET: 0.90	ACC: 87.33% (± 0.03) F1: 83.71% (± 0.03) ET: 0.90	**ACC: 89.22% (± 0.03)** **F1: 84.99% (± 0.04)** **ET: 0.96**	ACC: 87.91% (± 0.03) F1: 83.43% (± 0.04) ET: 1.06
NB	ACC: 79.01% (± 0.03) F1: 64.08 % (± 0.05) ET – 0.03	ACC: 79.01% (± 0.02) F1: 64.08% (± 0.05) ET: 0.04	ACC: 79.01% (± 0.02) F1: 64.08% (± 0.05) ET: 0.04	ACC: 79.59% (± 0.03) F1: 65.54% (± 0.06) ET: 0.05	ACC: 79.01% (± 0.02) F1: 64.05% (± 0.05) ET: 0.05	ACC: 79.74% (± 0.03) F1: 65.55% (± 0.06) ET: 0.09
KNN	ACC: 79.31% (± 0.02) F1: 68.01% (± 0.04) ET – 0.20	ACC: 81.50% (± 0.04) F1: 70.28% (± 0.06) ET: 0.25	ACC: 80.47% (± 0.02) F1: 68.90% (± 0.04) ET: 0.25	ACC: 75.22% (± 0.02) F1: 67.28% (± 0.03) ET: 0.28	ACC: 76.53% (± 0.01) F1: 68.86% (± 0.03) ET: 0.27	ACC: 76.24% (± 0.02) F1: 68.46% (± 0.03) ET: 0.15
ANN	ACC: 86.45% (± 0.04) F1: 80.26% (± 0.06) ET: 3.68	ACC: 87.18% (± 0.03) F1: 82.20% (± 0.03) ET: 5.52	ACC: 85.87% (± 0.02) F1: 79.42% (± 0.03) ET: 5.67	ACC: 83.24% (± 0.04) F1: 78.36% (± 0.02) ET: 4.15	ACC: 85.86% (± 0.02) F1: 80.91% (± 0.05) ET: 6.90	ACC: 86.74% (± 0.02) F1: 81.98% (± 0.03) ET: 7.15
GB	**ACC: 88.20% (± 0.04)** **F1: 83.75% (± 0.05)** **ET: 0.69**	**ACC: 88.20% (± 0.04)** **F1: 83.75% (± 0.05)** **ET: 0.59**	**ACC: 88.20% (± 0.04)** **F1: 83.75% (± 0.05)** **ET: 0.59**	ACC: 83.69% (± 0.04) F1: 79.39% (± 0.04) ET: 0.61	ACC: 87.47% (± 0.03) F1: 82.97(± 0.04) ET: 0.73	ACC: 87.47% (± 0.04) F1: 83.29% (± 0.05) ET: 0.76

Table 26 - Data normalization and balancing results for the BCRC experiment.

Algorithm	**Control**	**Z-score normalization**	**Min-max normalization**	**RU**	**RO**	**SO**
LR	ACC: 91.16% (± 0.01) F1: 87.40% (± 0.02) ET: 0.11	ACC: 90.60% (± 0.01) F1: 87.16% (± 0.02) ET: 0.08	ACC: 90.60% (± 0.03) F1: 86.54% (± 0.03) ET: 0.06	ACC: 91.17% (± 0.01) F1: 88.03% (± 0.02) ET: 0.12	ACC: 90.63% (± 0.03) F1: 87.39% (± 0.04) ET: 0.12	ACC: 91.17% (± 0.02) F1: 88.02% (± 0.03) ET: 0.30
SVM	ACC: 92.27% (± 0.02) F1: 88.56% (±0.03) ET: 0.03	ACC: 85.08% (± 0.07) F1: 78.20% (± 0.10) ET: 0.05	ACC: 86.16% (± 0.03) F1: 80.43% (± 0.09) ET: 0.04	ACC: 90.63% (± 0.02) F1: 87.38% (± 0.03) ET: 0.05	ACC: 90.68% (± 0.06) F1: 87.85% (± 0.07) ET: 0.06	ACC: 90.66% (± 0.04) F1: 87.50% (± 0.06) ET: 0.09
DT	ACC: 86.77% (± 0.04) F1: 81.56% (± 0.04) ET: 0.03	ACC: 86.77% (± 0.04) F1: 81.56% (± 0.04) ET: 0.04	ACC: 86.77% (± 0.04) F1: 81.56% (± 0.04) ET: 0.04	ACC: 85.68% (± 0.05) F1: 81.35% (± 0.05) ET: 0.05	ACC: 88.95% (± 0.04) F1: 84.48% (± 0.06) ET: 0.06	ACC: 90.05% (± 0.03) F1: 86.36% (± 0.04) ET: 0.09
RF	ACC: 92.27% (± 0.03) F1: 88.25% (± 0.05) ET: 0.56	ACC: 92.27% (± 0.02) F1: 88.25% (± 0.05) ET: 0.60	ACC: 92.82% (± 0.04) F1: 89.12% (± 0.06) ET: 0.59	ACC: 91.71% (± 0.03) F1: 88.85% (± 0.04) ET: 0.57	ACC: 92.84% (± 0.03) F1: 89.58% (± 0.05) ET: 0.65	**ACC: 92.82% (± 0.04)** **F1: 89.60% (± 0.06)** **ET – 0.68**
NB	ACC: 51.95% (± 0.07) F1: 58.08% (± 0.04) ET: 0.02	ACC: 51.95% (± 0.07) F1: 58.08% (± 0.04) ET: 0.03	ACC: 51.95% (± 0.07) F1: 58.08% (± 0.04) ET: 0.04	ACC: 53.60% (± 0.05) F1: 59.66% (± 0.04) ET: 0.06	ACC: 51.97% (±0.08) F1: 58.53% (± 0.06) ET: 0.05	ACC: 53.05% (± 0.05) F1: 58.18% (± 0.03) ET – 0.09
KNN	ACC: 88.96% (± 0.03) F1: 83.74% (± 0.05) ET: 0.21	ACC: 78.98% (± 0.06) F1: 65.99% (± 0.11) ET: 0.22	ACC: 81.19% (± 0.08) F1: 71.00% (± 0.11) ET: 0.23	ACC: 86.74% (± 0.02) F1: 81.68% (± 0.03) ET: 0.23	ACC: 88.44% (± 0.04) F1: 84.45% (± 0.04) ET: 0.24	ACC: 90.08% (± 0.03) F1: 85.97% (± 0.05) ET: 0.15
ANN	ACC: 86.73% (± 0.02) F1: 82.41% (± 0.04) ET: 1.59	ACC: 82.84% (± 0.06) F1: 75.19% (± 0.09) ET: 1.77	ACC: 86.74% (± 0.04) F1: 80.48% (± 0.06) ET: 1.69	ACC: 85.08% (± 0.03) F1: 81.59% (± 0.03) ET: 1.38	ACC: 86.74% (± 0.02) F1: 83.58% (± 0.02) ET: 1.92	ACC: 87.28% (± 0.02) F1: 84.14% (± 0.03) ET: 1.94
GB	**ACC: 94.49% (± 0.02)** **F1: 91.99% (± 0.04)** **ET: 0.61**	**ACC: 94.49% (± 0.02)** **F1: 91.99% (± 0.04)** **ET: 0.52**	**ACC: 94.49% (± 0.02)** **F1: 91.99% (± 0.04)** **ET: 0.53**	ACC: 92.27% (± 0.04) F1: 89.81% (± 0.04) ET: 0.61	ACC: 94.49% (± 0.02) F1: 91.99% (± 0.04) ET: 0.67	ACC: 93.95% (± 0.03) F1: 91.51% (± 0.04) ET: 0.65

Results Discussion

For the MCD experiment, the best results were achieved by the RF and GB algorithms, the first using SO and the second using RO. RF achieved an ACC of 88.64%, whilst GB had an ACC of 88.34%. Data normalization techniques didn´t produce significant changes in the above-mentioned algorithms but produced better results for LR, SVM, KNN and ANN.

For the BCD experiment, the best results were achieved by the RF and GB algorithms, the first using RO and the second without using any preprocessing technique, achieving an ACC/F1-score of 89.22%/84.99% and 88.20%/83.75%, respectfully. Similarly, to the MCD experiment, data normalization techniques had no significant effect on the results of RF and GB algorithms but brought improvements to LR, SVM, KNN and ANN.

For the BCRC experiment, the best results were also achieved by the GB and RF algorithms, achieving an ACC/F1-score of 94.49%/91.99% and 92.82%/89.60%, respectively. For the first, no data balancing or normalization techniques brought improvements compared to Control results, while the second achieved the best score with the use of SO.

Overall, the algorithms that produced the best results were RF and GB. Applying data normalization techniques didn´t produce any significant benefit on these algorithms; however, oversampling techniques improved the results in some cases.

5.3.2 Feature Selection

Next, the use of feature selection techniques was tested. Only the GB and RF algorithms were used, as these had the best results in the previous tests. Table 27, Table 28 and Table 29 present the results for each experiment.

Table 27 - Feature selection results for the MCD experiment.

Algorithm	Control	PC	UFS	RFE
RF with SO	ACC: 87.61% (± 0.02) ET: 1.02	ACC: 86.74(± 0.02) ET: 1.03	ACC: 87.47 (± 0.02) ET: 1.08	ACC: 88.34 (± 0.02) ET: 1.01
GB	ACC: 87.91% (± 0.02) ET: 2.34	ACC: 87.32 (± 0.03) ET: 2.24	ACC: 87.90 (± 0.02) ET: 2.14	ACC: 87.62 (± 0.03) ET: 1.96

Table 28 - Feature selection results for the BCD experiment.

Algorithm	Control	PC	UFS	RFE
RF with RO	ACC – 89.22% (± 0.03) F1 – 84.99% (± 0.04) ET – 0.96	ACC – 87.91 (± 0.04) F1 – 82.62 (± 0.06) ET – 0.90	ACC – 87.62 (± 0.03) F1 – 82.25 (± 0.04) ET – 0.81	ACC – 88.64 (± 0.03) F1 – 83.88 (± 0.05) ET – 0.87
GB	ACC – 88.20% (± 0.04) F1 – 83.75% (± 0.05) ET – 0.69	ACC – 86.89 (± 0.03) F1 – 82.29 (± 0.03) ET – 0.63	ACC – 87.04 (± 0.05) F1 – 81.87 (± 0.05) ET – 0.53	ACC – 87.91 (± 0.04) F1 – 83.29 (± 0.05) ET – 0.49

Table 29 - Feature selection results for the BCRC experiment.

Algorithm	Control	PC	UFS	RFE
RF	ACC – 92.27% (± 0.03) F1 – 88.25% (± 0.05) ET – 0.56	ACC – 91.71 (± 0.03) F1 – 87.52 (± 0.05) ET – 0.56	ACC – 92.27 (± 0.03) F1 – 88.39 (±0.06) ET – 0.52	ACC – 94.49 (± 0.04) F1 – 91.93 (± 0.06) ET – 0.50
GB	ACC – 94.49% (± 0.02) F1 – 91.99% (± 0.04) ET – 0.61	ACC – 93.95 (± 0.03) F1 – 91.44 (± 0.04) ET – 0.39	ACC – 93.39 (± 0.03) F1 – 90.37 (± 0.04) ET – 0.32	ACC – 94.49 (± 0.03) F1 – 91.79 (± 0.05) ET – 0.31

Results Discussion

Overall, the use of PC and UFS negatively impacted the results of both algorithms. The use of RFE brought performance improvements to RF in all experiences but didn´t improve the results of GB. Despite the worse results, the use of feature selection techniques reduced the ET of the algorithms in most cases, as the number of features used is lower.

5.3.3 Hyperparameter Optimization

ML models have two types of parameters: model parameters are the ones that are learned in the training phase, such as weights in ANN, and hyperparameters are all parameters that can be set before the training process starts. Hyperparameter optimization refers to the process of tuning hyperparameters, to find the combination of their values that achieves the best results [68]. To perform this optimization several methods can be used, according to [68]:

- Manual Search, where the hyperparameters are adjusted manually;
- Random Search, which tests random combinations of hyperparameter values;
- Grid Search, a method that tests every possible combination of hyperparameter values.

Even though Grid Search usually takes longer to run, as it tests every possible hyperparameter combination, this was the method that was chosen, since the dataset size is reduced and generally the algorithms take a low amount of time to train. To apply this method, the GridSearchCV class provided in the Sci-kit learn library was used. Table 30 displays the algorithms, hyperparameters and possible values that were selected for this tuning process.

Table 30 - Algorithm hyperparameters and possible values for GridSearchCV.

Algorithm	Hyperparameters
RF	max_depth: [1;5;10;25] max_features: [0.1;0.2;0.3;0.4;0.5;0.6;0.7;0.8;0.9;1.0] n_estimators: [100;500;1000]
GB	colsample_bytree: [0.6;0.8;1.0] gamma: [0.5;1;1.5;2;5] max_depth: [3;5;7;9] min_child_weight: [1,3,5] subsample: [0.6;0.8;1.0]

Since the use of data balancing and feature selection techniques provided better results with their use, ML models were tested using both, evaluating the best possible combination of techniques to use. The models that generated the best results were the ones selected for the hyperparameter optimization process. Table 31, Table 32 and Table 33 present the results of this process for each experiment.

Table 31 - Hyperparameter optimization results for MCD experiment.

ML Model	With default hyperparameters	With optimized hyperparameters	Best hyperparameters

RF with SO and RFE	ACC = 89.36% (± 0.02)
	ET = 5.62
		ACC = 89.80% (± 0.03)
		ET = 63.86
			max_depth = 25
			max_features = 0.3
			n_estimators = 1000

GB with RO	ACC – 88.34% (± 0.02)
	ET – 6.69
		ACC = 89.65% (± 0.02)
		ET = 9.92
			colsample_bytree = 0.6
			gamma = 1
			max_depth = 5
			min_child_weight = 1
			subsample = 0.6

Table 32 - Hyperparameter optimization results for the BCD experiment.

ML Model	With default hyperparameters	With optimized hyperparameters	Best hyperparameters

RF with RO	ACC – 89.22% (± 0.03)
	F1 – 84.99% (± 0.04)
	ET – 0.96
		ACC = 90.24% (± 0.03)
		F1 = 86.52% (± 0.04)
		ET = 11.13
			max_depth = 25
			max_features = 0.3
			n_estimators = 1000

GB	ACC – 88.20% (± 0.04)
	F1 – 83.75% (± 0.05)
	ET – 0.69
		ACC = 89.66% (± 0.02)
		F1 = 85.77% (± 0.05)
		ET = 0.74
			colsample_bytree = 1
			gamma = 0.5
			max_depth = 9
			min_child_weight = 1
			subsample = 1

Table 33 - Parameter optimization for the BCRC experiment.

ML Model	With default hyperparameters	With optimized hyperparameters	Best hyperparameters

RF with RFE	ACC – 94.49 (± 0.04)
	F1 – 91.93 (± 0.06)
	ET – 0.50
		ACC = 94.49% (± 0.03)
		F1 = 92.12% (± 0.04)
		ET = 0.52
			max_depth = 5
			max_features = 0.6
			n_estimators = 100

GB	ACC – 94.49% (± 0.02)
	F1 – 91.99% (± 0.04)
	ET – 0.61
		ACC = 95.05% (± 0.03)
		F1 = 92.86% (± 0.04)
		ET = 0.38
			colsample_bytree = 0.6
			gamma = 0.5
			max_depth = 5
			min_child_weight = 3
			subsample = 1

Results Discussion

Overall, hyperparameter optimization had positive results in all experiments, boosting the performance of the previously built ML models with default parameters. However, in some cases, the increase in performance was minimal, while the ET went up considerably, specifically when the n_estimators hyperparameter of RF was set to 1000. This can be an issue in the future, as the quantity of data increases.

For the MCD experiment, the best result was obtained with the model using RF with SO and RFE, with an accuracy of 89.80%. Regarding the BCD experiment, the model using RF with RO obtained the best performance with an accuracy/F1-score of 90.24%/86.52%. Finally, for the BCRC experiment, the best results were obtained using the model with GB, with an accuracy/F1-score of 95.05%/92.86%.

5.3.4 Model validations

The last step of trialling was to validate the ML models against cases of RC in distinct moments, as explained in section 4.1.3. Apart from the data that was already available, new cases of RC in the M0, M1 and M2 moments were collected by the healthcare professionals. Table 34, Table 35, Table 36 display the results for each experiment.

Table 34 – Model validations for the MCD experiment.

Moment	Number of Cases	Predicted has having RC
M0	7	7
M1	33	31
M2	41	39
M3	1	1
M4	2	2

Table 35 – Model validations for the BCD experiment.

Moment	Number of Cases	Predicted as having disease
M0	7	7
M1	33	32
M2	41	37
M3	1	1
M4	2	2

Table 36 – Model validations for the BCRC experiment using RF with RO and RFE.

Moment	Number of Cases	Predicted has having RC
M0	7	6
M1	33	30
M2	41	39
M3	1	1
M4	2	2

Results Discussion

For the M0 moment, which was the one used during the training phase, both the models used for the MCD and BCD experiments predicted all the cases as having RC. The model built for the BCRC experiment failed one case. This worse result can be derived from the fact that the data used for this experiment was smaller, compared to the other ones.

Regarding the other moments, these cases were collected to check if there was any difference in the predictions made by the models, as explained in section 4.1.3. The results were discussed with healthcare professionals, who expected a lower number of patients diagnosed as having RC/disease, at those moments.

However, the results demonstrate that even in the M1, M2, M3 and M4 moments, the models predict most cases as still having RC/disease. It was concluded that these models could be useful in the screening stage of RC, but not in the other moments.

5.3.5 Feature Importance

Apart from evaluating the predictions made by the ML models, it was also valuable for the healthcare professionals to know which features were more important when making said predictions. The RF algorithm has an attribute called "feature_importances_" that returns a score for each feature, based on the importance they had during the training phase. Table 37 displays the ten most important features of the model used in the MCD experiment, which achieved the best results in the previous section.

Table 37 – Feature importance for the RF algorithm used in the MCD experiment.

Feature	Importance Score
AGE	0.116
C3	0.084
C4DC	0.083
C50H	0.061
C8	0.061
C2	0.059
C160H	0.053
C14	0.047
C12	0.045
C16	0.044

6 Conclusions

In this section, general considerations regarding the dissertation are presented, as well as suggestions for future steps to take, to improve the work developed.

6.1 General Considerations

The main objective proposed in this dissertation was to investigate if acylcarnitine profile data could constitute a biochemical marker for the prediction of CRC, by applying ML algorithms and techniques.

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. This process was automated with the development of a software application, reducing the amount of time and effort necessary to process new information.

With all the information compiled, several data preprocessing techniques and ML algorithms were evaluated, to test their effectiveness in the task of classifying patients with CRC and other metabolic diseases. As a first analysis, the ML models that were built were able to identify most of the CRC cases, which is a good indication that acylcarnitine data can be used in the screening stage of this disease. However, to have more reliable results, these models should be validated with more data, considering that a misdiagnosis can be very costly and have a negative impact on the life of a patient.

The results of this study can be used to develop a decision support system, helping in early CRC detection. The predictions made by the system should not be considered as the only source of truth, but as a complement to the analysis made by a healthcare professional.

6.2 Future Work

To continue the work developed, the following topics can be considered:

- Conduct more experiments with new data, to validate the results obtained in this study;
- Test the use of DL, if data is sufficient;
- Gather more information about the different cancer moments and treatments, to develop ML models capable of suggesting the right treatment to apply to a patient;
- Explore the use of unsupervised ML algorithms, to group the patients into different clusters, according to their characteristics;
- Develop a decision support system, aligned with the proposed architecture, that combines the results of this study and the one made in [9].

7 References

[1] R. Bhardwaj, A. R. Nambiar, and D. Dutta, “A Study of Machine Learning in Healthcare,” in *Proceedings - International Computer Software and Applications Conference*, Sep. 2017, vol. 2, pp. 236–241. doi: 10.1109/COMPSAC.2017.164.

[2] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, “Deep learning for healthcare: Review, opportunities and challenges,” *Brief Bioinform*, vol. 19, no. 6, pp. 1236–1246, May 2017, doi: 10.1093/bib/bbx044.

[3] A. Géron, “Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems,” 2017. [Online]. Available: http://oreilly.com/safari

[4] American Cancer Society, “What Is Colorectal Cancer? | How Does Colorectal Cancer Start?,” 2020. https://www.cancer.org/cancer/colon-rectal-cancer/about/what-is-colorectal-cancer.html (accessed Oct. 05, 2022).

[5] P. Rawla, T. Sunkara, and A. Barsouk, “Epidemiology of colorectal cancer: Incidence, mortality, survival, and risk factors,” *Przeglad Gastroenterologiczny*, vol. 14, no. 2. Termedia Publishing House Ltd., pp. 89–103, 2019. doi: 10.5114/pg.2018.81072.

[6] SPED, “SPED - Cancro Colorretal,” 2020. https://www.sped.pt/index.php/publico/carcinoma-colorretal (accessed Nov. 12, 2021).

[7] M. Anna *et al.*, “The carnitine system and cancer metabolic plasticity,” *Official journal of the Cell Death Differentiation Association*, vol. 1234567890, p. 1234567890, 2018, doi: 10.1038/s41419-018-0313-7.

[8] P. Rinaldo, T. M. Cowan, and D. Matern, “Acylcarnitine profile analysis,” 2008, doi: 10.1097/GIM.0b013e3181614289.

[9] J. Gonçalves, “Previsão Inteligente das alterações metabólicas no cancro retal com base em modelos de machine e deep learning,” 2021.

[10] U. Shafique and H. Qaiser, “A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA),” 2014. [Online]. Available: http://www.ijisr.issr-journals.org/

[11] EU SCIENCE HUB, “2020 Cancer incidence and mortality in EU-27 countries,” 2020. https://ec.europa.eu/jrc/en/news/2020-cancer-incidence-and-mortality-eu-27-countries (accessed Jan. 10, 2022).

[12] Hospital da Luz, “Cancro colorretal: porque tem uma mortalidade tão alta?,” 2019. https://www.hospitaldaluz.pt/pt/dicionario-de-saude/cancro-colorretal-mais-mortifero (accessed Jan. 10, 2022).

[13] C. Indiveri *et al.*, “The mitochondrial carnitine/acylcarnitine carrier: Function, structure and physiopathology,” *Mol Aspects Med*, vol. 32, no. 4–6, pp. 223–233, Aug. 2011, doi: 10.1016/j.mam.2011.10.008.

[14] M. Dambrova *et al.*, “Acylcarnitines: Nomenclature, Biomarkers, Therapeutic Potential, Drug Targets, and Clinical Trials,” *Pharmacol Rev*, vol. 74, no. 3, pp. 506–551, Jul. 2022, doi: 10.1124/PHARMREV.121.000408.

[15] B. Mahesh, “Machine Learning Algorithms-A Review,” 2019, doi: 10.21275/ART20203995.

[16] J. Alzubi, A. Nayyar, and A. Kumar, “Machine Learning from Theory to Algorithms: An Overview,” in *Journal of Physics: Conference Series*, Nov. 2018, vol. 1142, no. 1. doi: 10.1088/1742-6596/1142/1/012012.

[17] B. Giuseppe, *Machine Learning Algorithms*. 2017.

[18] S. B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques,” 2007.

[19] H. Zhang and D. Li, “Naïve Bayes Text Classifier,” Apr. 2008, pp. 708–708. doi: 10.1109/grc.2007.40.

[20] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, “Comparing different supervised machine learning algorithms for disease prediction,” *BMC Med Inform Decis Mak*, vol. 19, no. 1, Dec. 2019, doi: 10.1186/s12911-019-1004-8.

[21] “Gradient Boosted Decision Trees-Explained | by Soner Yıldırım | Towards Data Science.” https://towardsdatascience.com/gradient-boosted-decision-trees-explained-9259bd8205af (accessed Sep. 28, 2022).

[22] D. M. Atallah, M. Badawy, and A. El-Sayed, "Intelligent feature selection with modified K-nearest neighbor for kidney transplantation prediction," *SN Appl Sci*, vol. 1, no. 10, Oct. 2019, doi: 10.1007/s42452-019-1329-z.

[23] L. Atlas *et al.*, "Performance comparison of trained multi-layer perceptrons and trained classification trees," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1989, vol. 3, pp. 915–920. doi: 10.1109/icsmc.1989.71429.

[24] P. Shinde and S. Sah, *A Review of Machine Learning and Deep Learning Applications*. 2018.

[25] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," 2017.

[26] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. v. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Computational and Structural Biotechnology Journal*, vol. 13. Elsevier, pp. 8–17, 2015. doi: 10.1016/j.csbj.2014.11.005.

[27] T. Ayer, O. Alagoz, J. Chhatwal, J. W. Shavlik, C. E. Kahn, and E. S. Burnside, "Breast cancer risk estimation with artificial neural networks revisited: Discrimination and calibration," *Cancer*, vol. 116, no. 14, pp. 3310–3321, Jul. 2010, doi: 10.1002/cncr.25081.

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

[29] J. Listgarten *et al.*, "Predictive Models for Breast Cancer Susceptibility from Multiple Single Nucleotide Polymorphisms," 2004. [Online]. Available: http://www.polyomx.org/.

[30] A. Stojadinovic, A. Nissan, J. Eberhardt, T. C. Chua, J. O. Pelz, and J. Esquivel, "Development of a Bayesian Belief Network Model for Personalized Prognostic Risk Assessment in Colon Carcinomatosis," 2011.

[31] K. P. Exarchos, Y. Goletsis, and D. I. Fotiadis, "Multiparametric decision support system for the prediction of oral cancer reoccurrence," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1127–1134, 2012, doi: 10.1109/TITB.2011.2165076.

[32] W. Kim *et al.*, "Development of novel breast cancer recurrence prediction model using support vector machine," *J Breast Cancer*, vol. 15, no. 2, pp. 230–238, Jun. 2012, doi: 10.4048/jbc.2012.15.2.230.

[33] C. Park, J. Ahn, H. Kim, and S. Park, “Integrative gene network construction to analyze cancer recurrence using semi-supervised learning,” *PLoS One*, vol. 9, no. 1, Jan. 2014, doi: 10.1371/JOURNAL.PONE.0086309.

[34] C. J. Tseng, C. J. Lu, C. C. Chang, and G. den Chen, “Application of machine learning to predict the recurrence-proneness for cervical cancer,” *Neural Comput Appl*, vol. 24, no. 6, pp. 1311–1316, 2014, doi: 10.1007/s00521-013-1359-1.

[35] A. LG and E. AT, “Using Three Machine Learning Techniques for Predicting Breast Cancer Recurrence,” *J Health Med Inform*, vol. 04, no. 02, 2013, doi: 10.4172/2157-7420.1000124.

[36] Y. C. Chen, W. C. Ke, and H. W. Chiu, “Risk classification of cancer survival using ANN with gene expression data from multiple laboratories,” *Comput Biol Med*, vol. 48, no. 1, pp. 1–7, May 2014, doi: 10.1016/j.compbiomed.2014.02.006.

[37] K. Park, A. Ali, D. Kim, Y. An, M. Kim, and H. Shin, “Robust predictive model for evaluating breast cancer survivability,” *Eng Appl Artif Intell*, vol. 26, no. 9, pp. 2194–2205, 2013, doi: 10.1016/j.engappai.2013.06.013.

[38] S. W. Chang, S. Abdul-Kareem, A. F. Merican, and R. B. Zain, “Oral cancer prognosis based on clinicopathologic and genomic markers using a hybrid of feature selection and machine learning methods,” *BMC Bioinformatics*, vol. 14, no. 1, May 2013, doi: 10.1186/1471-2105-14-170.

[39] X. Xu, Y. Zhang, L. Zou, M. Wang, and A. Li, “A gene signature for breast cancer prognosis using support vector machine,” in *2012 5th International Conference on Biomedical Engineering and Informatics, BMEI 2012*, 2012, pp. 928–931. doi: 10.1109/BMEI.2012.6513032.

[40] O. Gevaert, F. de Smet, D. Timmerman, Y. Moreau, and B. de Moor, “Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks,” vol. 22, no. 14, pp. 184–190, 2006, doi: 10.1093/bioinformatics/btl230.

[41] P. Rosado, P. Lequerica-Fernandez, L. Villallain, I. Pena, F. Sanchez-Lasheras, and J. C. de Vicente, “Survival model in oral squamous cell carcinoma based on clinicopathological parameters, molecular markers and support vector machines,” *Expert Syst Appl*, vol. 40, no. 12, pp. 4770–4776, 2013, doi: 10.1016/j.eswa.2013.02.032.

[42] D. Delen, G. Walker, and A. Kadam, “Predicting breast cancer survivability: A comparison of three data mining methods,” *Artif Intell Med*, vol. 34, no. 2, pp. 113–127, Jun. 2005, doi: 10.1016/j.artmed.2004.07.002.

[43] J. Kim and H. Shin, “Breast Cancer Survivability Prediction with Labeled, Unlabeled, and Pseudo-Labeled Patient Data,” 2012.

[44] M. N. Gevorkyan, A. v. Demidova, T. S. Demidova, and A. A. Sobolev, “Review and comparative analysis of machine learning libraries for machine learning,” *Discrete and Continuous Models and Applied Computational Science*, vol. 27, no. 4, pp. 305–315, Dec. 2019, doi: 10.22363/2658-4670-2019-27-4-305-315.

[45] Samaya Madhavan, “Compare deep learning frameworks,” Mar. 12, 2021. https://developer.ibm.com/articles/compare-deep-learning-frameworks/ (accessed Feb. 06, 2022).

[46] N. Rich and M. Holweg, “Value Analysis, Value Engineering,” 2000.

[47] P. Koen *et al.*, “Providing clarity and a common language to the ‘fuzzy front end,’” *Research Technology Management*, vol. 44, no. 2, pp. 46–55, 2001, doi: 10.1080/08956308.2001.11671418.

[48] CHUPORTO, “CHUPORTO - Apresentação,” 2022. https://www.chporto.pt/v0B0A/apresentacao (accessed Feb. 08, 2022).

[49] T. Woodall, “Conceptualising ‘Value for the Customer’: An Attributional, Structural and Dispositional Analysis,” 2003. [Online]. Available: http://www.amsreview.org/articles/woodall12-2003.pdf

[50] A. Osterwalder, “The Business Model Ontology: a proposition in a design science approach,” 2004.

[51] S. Nicola, “Análise Valor,” Porto, Feb. 2021.

[52] N. Rich and M. Holweg, “Value Analysis, Value Engineering,” 2000.

[53] © Warwick, “Quality Function Deployment,”

[54] Warwick Manufacturing Group, “Quality Function Deployment,” in *Product Excellence using Six Sigma*,

[55] T. L. Saaty, “Decision making with the analytic hierarchy process,” 2008.

[56] S. Daityari, “Angular vs React vs Vue: Which Framework to Choose,” 2021. https://www.codeinwp.com/blog/angular-vs-vue-vs-react/ (accessed Feb. 15, 2022).

[57] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Data Preprocessing for Supervised Learning”.

[58] W. S. Bhaya, “Review of Data Preprocessing Techniques in Data Mining.”

[59] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results," in *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, Apr. 2020, pp. 243–248. doi: 10.1109/ICICS49469.2020.239556.

[60] M. S. Shelke, P. R. Deshmukh, and V. K. Shandilya, "A Review on Imbalanced Data Handling Using Undersampling and Oversampling Technique," *International Journal of Recent Trends in Engineering and Research*, vol. 3, no. 4, pp. 444–449, May 2017, doi: 10.23883/ijrter.2017.3168.0uwxm.

[61] V. Kumar, "Feature Selection: A literature Review," *The Smart Computing Review*, vol. 4, no. 3, Jun. 2014, doi: 10.6029/smartcr.2014.03.007.

[62] D. Nettleton, "Selection of Variables and Factor Derivation," *Commercial Data Mining*, pp. 79–104, 2014, doi: 10.1016/B978-0-12-416602-8.00006-6.

[63] "Feature selection — scikit-learn 1.1.2 documentation." https://scikit-learn.org/stable/modules/feature_selection.html (accessed Sep. 19, 2022).

[64] "What is Package Diagram?" https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-package-diagram/ (accessed Sep. 11, 2022).

[65] "What is Deployment Diagram?" https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/ (accessed Sep. 11, 2022).

[66] S. García, J. Luengo, and F. Herrera, "Data Preprocessing in Data Mining." [Online]. Available: http://www.springer.com/series/8578

[67] "3.1. Cross-validation: evaluating estimator performance — scikit-learn 1.1.2 documentation." https://scikit-learn.org/stable/modules/cross_validation.html (accessed Sep. 21, 2022).

[68] "Hyperparameters Optimization. An introduction on how to fine-tune… | by Pier Paolo Ippolito | Towards Data Science." https://towardsdatascience.com/hyperparameters-optimization-526348bb8e2d (accessed Sep. 28, 2022).