



Previsão Inteligente das alterações metabólicas no cancro retal com base em modelos de machine e deep learning

JOSÉ GONÇALVES

Outubro de 2021

Previsão Inteligente das alterações metabólicas no cancro retal com base em modelos de machine e deep learning

José Gonçalves

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

Orientador: José Reis Tavares

Co-orientador: Isabel Praça

Porto, outubro 2021

Abstract

Machine learning, broadly speaking, applies statistical methods to training data to automatically adjust the parameters of a model, rather than a programmer needing to set them manually.

Deep Learning is a sub-area of Machine Learning that studies how to solve complex and intuitive problems. The methodologies adopted, using computational means, such as the machines learned and those understood in the world in specific contexts from previous experiences and based on the hierarchy of concepts, use the most used concepts for the form and efficient solution of more varied complex problems.

The main objective in this work is to study various classification algorithms in the area of machine learning, and validate until these points can use a solution for choosing more accurate methods in the selection of tests and in new statistics to improve the therapeutic response. The data involved in the training of classification algorithms refer to all patients with metabolic diseases shredding between the years 2003-2021 and the retrospective part. The best classification algorithms to develop are used in the decision support system in the most effective way in choosing the appropriate therapy for each of the future patients who predicted an approximate rate of 20 patients per year.

Keywords: Machine Learning, Deep Learning, Cancer, Feature Selection

Resumo

Machine Learning, em termos gerais, aplica métodos estatísticos aos dados de treino para ajustar automaticamente os parâmetros de um modelo, em vez de um programador necessitar de defini-los manualmente.

Deep Learning é uma subárea de Machine Learning que estuda como solucionar problemas complexos e intuitivos. As metodologias propostas permitem, com recurso a meios computacionais, que as máquinas aprendam e compreendam o mundo em determinados contextos a partir de experiências anteriores e com base na hierarquia de conceitos possam compreender conceitos mais complexos de forma a solucionarem eficientemente A mais variadíssima gama de problemas.

O principal objetivo neste trabalho consiste no estudo de vários algoritmos de classificação na área de machine learning de forma a validar até que ponto estes podem representar uma solução para a escolha de métodos mais precisos na selecção dos doentes e em novas estratégias para melhorar a resposta terapêutica. Os dados envolvidos para treino dos algoritmos de classificação referem-se a todos os doentes tratados com doenças metabólicas entre os anos 2003-2021 na parte retrospectiva. Os melhores algoritmos de classificação a desenvolver serão usados num sistema de apoio à decisão que ajude de forma mais efetiva na escolha da terapia adequada para cada um dos futuros pacientes que se prevê surgirem a uma taxa aproximada de 20 pacientes por ano.

Palavras-chave: Machine Learning, Deep Learning, Cancro, Feature Selection

Agradecimentos

Gostaria de começar por agradecer à minha família pelo apoio, e por sempre me terem dado todas as condições para que conseguisse fazer o meu percurso académico.

Agradecer também à minha namorada, por ter tido uma paciência e apoio inesgotáveis durante o tempo todo que despendi a desenvolver este projeto.

Ao professor José Reis Tavares e à professora Isabel Praça por todo o apoio prestado durante o projeto, e principalmente por todas as sugestões, disponibilidade e compreensão que foram prestados ao longo mesmo.

À doutora Lúcia Lacerda pela sua disponibilidade para o esclarecimento de dúvidas, além do seu olhar crítico de apoio e de guia ao longo da dissertação.

À doutora Marisa Santos, à doutora Lúcia Lacerda, assim como aos alunos de doutoramento Pedro Brandão, MD e Ivo Barros, MSc, do projecto intitulado “MetLARC - Metabolic abnormalities on tumour response and resistance to neoadjuvant chemoradiotherapy in Locally Advanced Rectal Cancer”, pelo fornecimento dos dados analíticos utilizados neste trabalho.

Por fim, quero deixar um grande agradecimento também aos meus amigos, colegas de curso, em especial ao Carlos Vicente e Daniel Dias, que me ajudaram, me apoiaram e confiaram em mim neste meu percurso académico.

Index

Introduction.....	17
1.1 Context	18
1.2 Motivation and Goals	18
1.3 Document Structure	19
Value Analysis	20
2.1 The New Concept Development (NCD)	20
2.1.1 Opportunity Identification	21
2.1.2 Opportunity Analysis.....	21
2.1.3 Idea Generation and Enrichment.....	21
2.1.4 Idea Selection.....	22
2.1.5 Concept Definition	22
2.2 Value.....	23
2.2.1 Business Value Analysis Models.....	23
2.2.2 Value Proposition.....	25
2.2.3 Perceived Value	25
2.2.4 Value for the customer.....	26
2.2.5 Canvas Business Model.....	27
Context and state of the art.....	29
3.1 Colorectal Cancer	29
3.1.1 Amino acid profile in CRC detection	30
3.2 Machine Learning.....	30
3.2.1 Overview	30
3.2.2 Existing solutions and approaches in cancer diagnosis.....	31
3.3 Types of Learning	33
3.3.1 Supervised Learning.....	33
3.3.2 Unsupervised Learning.....	34
3.3.3 Semi-Supervised Learning.....	34
3.3.4 Reinforcement Learning.....	34
3.4 Machine Learning Algorithms	35
3.4.1 Decision Tree	35
3.4.2 Random Forest (RF)	36
3.4.3 Naïve Bayes	36
3.4.4 K-Nearest Neighbors (kNN).....	37
3.4.5 Logistic Regression.....	37
3.4.6 Support Vector Machine (SVN).....	37
3.4.7 Artificial Neural Networks (ANN).....	38
3.4.8 Algorithms Comparison	39
3.5 Deep Learning	41
3.6 Machine Learning vs Deep Learning	42

3.6.1	Decision.....	43
3.7	Machine Learning Libraries	43
3.7.1	TensorFlow	43
3.7.2	PyTorch	44
3.7.3	Keras	44
3.7.4	Scikit Learn.....	44
3.7.5	Decision.....	45
Design		46
4.1	Functionalities	47
4.2	System Architecture	48
4.2.1	Planned Architecture	48
4.2.2	Prototype.....	50
Experiences and Evaluation		53
5.1	Dataset	54
5.2	Preprocessing	61
5.2.1	Data Cleaning.....	61
5.2.2	Data Normalization	62
5.2.3	Feature Selection	62
5.3	Imbalanced Data	69
5.4	Performance Evaluation.....	71
5.4.1	Split Dataset	71
5.4.2	Performance Metrics.....	73
5.5	Results	75
5.5.1	Data Balancing Evaluation	75
5.5.2	Feature Selection Evaluation	78
5.5.3	CRC Model Validation	85
Conclusion.....		87
6.1	Conclusion.....	87
6.2	Future Work.....	87

List of Figures

Figure 1 - The New Concept Development [2].....	20
Figure 2 - Verna Allee's value network [5]	23
Figure 3 - Porter's value chain.....	24
Figure 4 - Canvas Business Model.....	27
Figure 5 - 2020 cancer incidence in EU [11].....	29
Figure 6 - ML methods used in relevant publications for cancer susceptibility prediction.....	31
Figure 7 - ML methods used in relevant publications for cancer recurrence prediction	31
Figure 8 – Deep Learning methods used in relevant publications for cancer prognosis prediction [15].....	32
Figure 9 - Machine Learning Types [17]	33
Figure 10 - Machine Learning Algorithms	35
Figure 11 - Decision Tree [22]	36
Figure 12 - Bayes Theorem [24]	36
Figure 13 – Support Vector Machine [28]	38
Figure 14 - Structure of an Artificial Neuron.....	38
Figure 15 – Simple Neural Network vs Deep Learning Neural Network [31].....	42
Figure 16 – System use cases diagram.....	47
Figure 17 – System Architecture	49
Figure 18 – Prototype architecture.....	51
Figure 19 - Web Application user interface	52
Figure 20 – User interface response example	52
Figure 21 - Workflow of machine learning algorithm [21]	53
Figure 22 – Partial example of the dataset	54
Figure 23 - Absolute Frequency of the 2 classes in Binary classification of disease detection .	58
Figure 24 - Relative Frequency of the 2 classes in Binary classification of disease detection ...	58
Figure 25 – Relative frequency for Multi class Classification of diseases classes	59
Figure 26 - Absolute Frequency of the 2 classes in Binary classification colorectal cancer detection	60
Figure 27 - Relative Frequency of the 2 classes in Binary classification colorectal cancer detection	61
Figure 28 – Feature Selection	63
Figure 29 - Feature Selection: Filter and Wrapper methods [43]	63
Figure 30 - Pearson's Correlation Coefficient in Binary classification for disease detection.....	66
Figure 31 - Pearson's Correlation Coefficient in Multi class classification for disease detection	66
Figure 32 - Pearson's Correlation Coefficient in Binary classification for colorectal cancer detection	67
Figure 33 – Holdout method [48].....	72
Figure 34 – 10-fold cross-validation [48]	73
Figure 35 – Confusion matrix [23].....	74

List of Tables

Table 1 - Longitudinal perspective of value	26
Table 2 - Advantages and limitations of different supervised machine learning algorithms [23]	39
Table 3 – Differences between Machine Learning and Deep Learning [34]	42
Table 4 - Machine Learning Libraries Comparison [36]	45
Table 5 – Distribution of patients with disease	55
Table 6 – Dataset’s Classes and its distribution	56
Table 7 – Class distribution in Multi class Classification of diseases	59
Table 8 – Feature Selection techniques used	64
Table 9 – Feature id and corresponding name	65
Table 10 – Top 10 Features with best score using Univariate Feature Selection	68
Table 11 – Features with greatest importance using Recursive Feature Selection for each run	68
Table 12 – Data processing techniques applied to each of the datasets.....	70
Table 13 – Alterations to reduce the imbalance bias at algorithm level	71
Table 14 – Binary classification for colorectal cancer detection (control).....	75
Table 15 – Binary classification for colorectal cancer detection with oversampling.....	76
Table 16 - Binary classification for colorectal cancer detection with undersampling	76
Table 17 - Binary classification for colorectal cancer detection with balanced algorithms	77
Table 18 - Multi class Classification for disease detection (control).....	77
Table 19 - Multi class Classification for disease detection with undersampling	78
Table 20 - Multi class Classification for disease detection with oversampling.....	78
Table 21 – Performance comparison with the use of Feature Selection (Pearson’s correlation coefficient) in Binary Classification for disease detection	79
Table 22 - Performance comparison with the use of Feature Selection (Pearson’s correlation coefficient) in Multi class Classification for disease detection	80
Table 23 - Performance comparison with the use Top 10 features with the best average scores using Univariate Feature Selection in Binary Classification for disease detection	81
Table 24 - Performance comparison with the use 12 features present in all runs in the tier-1 using Recursive Feature Elimination in Binary Classification for disease detection.....	82
Table 25 – Percentage of False Negatives for each model	83
Table 26 – Performance using only the aminoacid profile features in the model generation..	84
Table 27 – Percentage of False Negatives for each model using only the aminoacid profile features in the model generation	84
Table 28 – Model detection of patient with colorectal cancer using Binary Classification for colorectal cancer models	85
Table 29 - Model detection of patient with colorectal cancer using Binary and Multi Class Classification for disease detection	86

Acronyms and Symbols

Acronyms List

ANN	Artificial Neural Networks
ML	Machine Learning
DL	Deep Learning
AI	Artificial Intelligence
CRC	Colorectal Cancer
NLP	<i>Natural Language Processing</i>
EU	European Union
FS	Feature Selection
FP	False Positives
FN	False Negatives

Introduction

Computer aid technology is widely applied in decision-making and outcome assessment of healthcare delivery, in which modelling knowledge and expert experience is technically important. However, the conventional rule-based models are incapable of capturing the underlying knowledge because they are incapable of simulating the complexity of human brains and highly rely on feature representation of problem domains. Thus, we attempt to apply a machine learning or deep model to overcome this weakness. That model can simulate the thinking procedure of human and combine feature representation and learning in a unified model.

The theme proposal and guidance in the proposing institutions are the responsibility of:

1. **Marisa Santos**, MD, PhD,

Graduated Hospital Assistant in General Surgery,
Responsible for the Emergency Team of the Hospital and University Center of Porto (CHUPorto),
Coordinator of the Colorectal Surgery Unit of the General Surgery Service of CHUPorto,
Coordinator of the Reference Center for Treatment of Rectal Cancer of CHUPorto,
Associate Professor of the Integrated Masters in Medicine at the Abel Salazar Biomedical Science Institute (ICBAS) of the University of Porto (UP),
Researcher in the “Oncology Research group” of the UMIB-Multidisciplinary Unit for Biomedical Research at ICBAS-UP, integrated in the associated laboratory ITR-Laboratory for Integrative and Translational Research in Population Health

2. **Lucia Lacerda**, PhD

Clinical Laboratory Geneticist of the Genetic Biochemistry Unit of the Medical Genetics Center Jacinto Magalhães, of the Hospital and University Center of Porto (CHUPorto),
Member of the Reference Center for Diagnosis and Treatment of Hereditary Metabolism Diseases at CHUPorto,
External collaborator of the Integrated Masters in Medicine at the Abel Salazar Biomedical Science Institute (ICBAS) of the University of Porto (UP),

Researcher at the “Clinical and Experimental Human Genomics group” of the UMIB-Multidisciplinary Biomedical Research Unit of the ICBAS-UP, integrated in the associated laboratory ITR-Laboratory for Integrative and Translational Research in Population Health

1.1 Context

Nearly all aspects of modern life are in some way being changed by big data and machine learning. It is no surprise then that medicine is awash with claims of revolution from the application of machine learning to big health care data. Recent examples have demonstrated that big data and machine learning can create algorithms that perform on par with human physicians.

Health care is coming to a new era where the abundant biomedical data are playing more and more important roles. In this context, for example, precision medicine attempts to ‘ensure that the right treatment is delivered to the right patient at the right time’ by taking into account several aspects of patient’s data, including variability in molecular traits, environment, electronic health records and lifestyle [49]. The large availability of biomedical data brings tremendous opportunities and challenges to health care research. In particular, exploring the associations among all the different pieces of information in these data sets is a fundamental problem a to develop reliable medical tools based on data-driven approaches and machine learning.

Predictive tools based on machine learning techniques have not been widely applied in medicine. In fact, there remain many challenges in making full use of the biomedical data, owing to their high-dimensionality, heterogeneity, temporal dependency, sparsity and irregularity [1].

There are many aspects of machine and deep learning that could be helpful in health care, such as its superior performance, end-to-end learning scheme with integrated feature learning, capability of handling complex and multi-modality data and so on.

1.2 Motivation and Goals

Colorectal cancer is the fourth most common type of cancer and one of the main causes of death from malignant disease worldwide. Patients with locally advanced rectal cancer are treated in a multimodal manner with radiotherapy, chemotherapy and surgery. The therapeutic response can be very variable - some showing a complete response, while others have little or no response - and the pathological response has become important in assessing the prognosis.

The knowledge about the molecular mechanisms of colorectal cancer has been increasing, without, however, translating this into more precise methods in the selection of patients and new strategies to improve the therapeutic response.

The proliferative drive of tumour cells requires an abundant supply of amino acids. This work aims to evaluate changes in the plasma amino acid profile correlated with tumour activity.

Deep Learning is a sub-area of Machine Learning that studies how to solve complex and intuitive problems. The proposed methodologies allow, using computational means, that the machines learn and understand the world in certain contexts from previous experiences and based on the hierarchy of concepts can understand more complex concepts in order to efficiently solve the most varied range of problems.

The main objective of this work is to identify an amino acid profile that constitutes a biomarker of tumour activity in the diagnosis and assessment of response to treatment.

To accomplish this goal a study of several classification algorithms in machine and deep learning in order to validate the extent to which they can represent a solution to the problem described will be conducted. The data involved in training the classification algorithms are from cases diagnosed with inherited metabolic diseases.

The best classification algorithms to be developed will be used in a decision support system that helps more effectively in the classification of cases with colorectal cancer.

1.3 Document Structure

Introduction: The first chapter of this document presents an introduction and interpretation of the problem, presenting the context involved in this thesis, the problem with which it will be worked and the objectives for the project.

Value Analysis: The value analysis of this development is presented succinctly, as well as the potential of this dissertation as a business idea and how it would fit into the market.

Context and State of the art: In this chapter, state-of-the-art modules will be presented. Initially, an introduction to cancer, specifically colorectal cancer, and its relation with the aminoacid profile is presented. The evolution and the emergence of Machine Learning and Deep Learning. The different types of classification systems are also addressed. The study of the various algorithms used in ML and existing frameworks is presented, as well as the comparative study carried out.

Design: In this chapter, it is presented the information about the system view, from a structural point of view, analyzing the intended features, the architecture and alternatives to the architecture of the system.

Experiences and Evaluation: In this chapter, it is presented the experiments that were conducted as the analysis of the performance of the models built having in consideration the considered performance metrics.

Conclusion: In this chapter, a wrap up of the dissertation is made, presenting what conclusions can be made and what should be the next steps.

Value Analysis

In this chapter, a value analysis of the solution proposed in this project will be carried out. In this analysis, the five key elements of the NCD model (New concept development model) will be identified, the value proposition for the customer will be presented, and finally, the Canvas business developed.

2.1 The New Concept Development (NCD)

The new concept development model, as shown in Figure 1, defines five key points: identification and analysis of the opportunity, creation and selection of the idea, and concept definition.

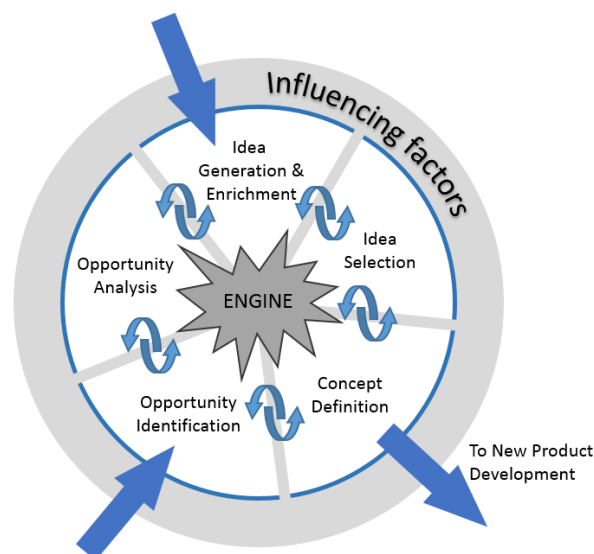


Figure 1 - The New Concept Development [2]

2.1.1 Opportunity Identification

The opportunity analysis process starts with market analysis, to identify market needs, in order to explore opportunities for new or different tools capable of adding value to a market.

Colorectal cancer is one of the most common types of cancer and a major cause of death from malignant disease worldwide. Patients with locally advanced rectal cancer are treated in a multimodal manner with radiotherapy, chemotherapy and surgery. The therapeutic response can be very variable - some showing a complete response, while others have little or no response - and the pathological response has become important in assessing the prognosis. The knowledge about the molecular mechanisms of colorectal cancer has been increasing, without, however, translating this into more precise methods in the selection of patients and new strategies to improve the therapeutic response.

With this, it is possible to identify opportunities in the detection and treatment of cancer patients, making use of the data gathered in the traditional analysis process to create an algorithm that can help the healthcare professionals work.

2.1.2 Opportunity Analysis

Opportunity analysis is a process that aims to study business opportunities, in order to understand, how the market will be able to understand and take advantage of the solution, so that the solution design is profitable.

It is necessary a retrospective component for the identification of amino acid diseases in cases diagnosed with inherited diseases of the metabolism and a prospective component in cases diagnosed with colorectal cancer. The ultimate goal is to identify an amino acid profile that constitutes a biomarker of tumor activity in the diagnosis and assessment of response to treatment.

Recently, machine and deep learning have promised algorithms that not only can make health decisions based on data and free from human error, but can also process data sets that are much larger than anyone could. As is shown in chapter 1.4.2 (Existing solutions and approaches in cancer diagnosis) there are many studies using machine and deep learning technology for cancer diagnoses. Also several studies suggest that multivariate analysis of amino-acid profiles may be useful for the early detection of cancer [3]. There is an opportunity to combine the power of machine and deep learning technologies to create a model using amino-acid profiles to help in the early cancer detection.

2.1.3 Idea Generation and Enrichment

The factor of generation and enrichment of ideas is related to the creation of new ideas linked to the original idea, thereby creating complements or changes to the project idea so that it can serve the client better.

A set of tools and platforms were available to carry out the recommended project. There was an opportunity to use open source tools that fell within the scope of the proposed

solution, some of which were selected as potential choices. These offer not only algorithms but also environments to execute them.

Currently, there are several studies related to cancer diagnostic and prognostic systems, which use other methodologies for data processing and analysis, using for example deep learning for pattern recognition. The main difference present in this dissertation is the use of an amino-acid profile data.

For this phase, the project was discussed with a professional in the sector, who suggested:

- in addition to being able to obtain the diagnosis of the disease,
- to understand which features in the amino-acid profile are the most correlated with the diagnosis,
- to recommend the best type of treatment for the patient,
- to present this service in an application with a user friendly interface for health technicians,
- and besides that, was also suggested the use of this clinical data to be employed in the diagnostic of other diseases.

2.1.4 Idea Selection

After generating ideas, it is necessary to define and prioritize ideas, in order to transform an idea, in the development of a solution, taking into account that the choice of options must respect a relationship between need, benefit and cost.

Thus, for the solution, it was defined that it would be an asset for a first phase, to be able to obtain a diagnosis of colorectal cancer with the highest possible rate of success and trying to understand which clinical data are most relevant for that same diagnosis.

These ideas were selected based on technological risk, development costs and the benefit they bring to health professionals. The remaining ideas will remain as future work to be carried out after this project.

2.1.5 Concept Definition

The NDC model ends with the definition of the concept and this means that there must be several assessments, including the needs and benefits of customers, investment requirements as well as the assessment of existing competition for the market segment, technologies and necessary mechanisms and, finally, what is the inherent risk of the project.

The diagnosis of cancer is a sensitive area, since the values of amino-acid analysis (among other data) can be affected by numerous factors that are not easily identified, such as the transport of blood samples to the analysis sites.

This development will be carried out using machine and deep learning, applying different algorithms for the processing and classification of data.

With this development, it will be possible to improve the diagnostic process and the detection of colorectal cancer by health technicians, as well as support the treatment decision process, contributing to a reduction of possible complications of the disease.

2.2 Value

This concept can be defined in many different ways, and according to different theoretical contexts where it was used, was defined as need, desire, interest, criteria, beliefs, attitudes and preferences [4].

2.2.1 Business Value Analysis Models

Verna Allee's value network, shown in Figure 2, is defined by any network of relationships that create both tangible and intangible value, through exchanges between individuals, groups and organizations.

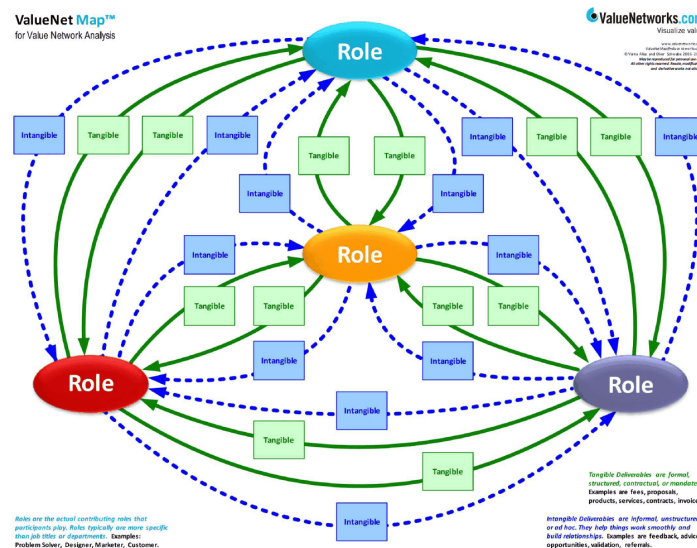


Figure 2 - Verna Allee's value network [5]

The tangible value can be interpreted as any type of interaction on which it is possible to assign a numerical value. Be it the sale of goods or services, payments, requests for proposals, invoices, etc. However, the intangible value is more directed to the human factor, and is based on the interactions and exchange of information between the stakeholders in the business process, who pass on knowledge and / or benefits to each other. The purpose of value networks is to create the maximum benefit for the actors involved in the network, from suppliers to producers and finally to customers. The value network differs from the value chain in that it accepts the value produced by the interrelationships of the stakeholders, not only by the activities performed by them. While according to the value chain, the business process can

be optimized only by changing the *modus operandi* of activities, in the value network, the sharing of information and knowledge also translates into adding value to the process. If an employee does not know how to perform a task, it is more feasible to provide someone who can teach, or offer training, than to be always firing and hiring new employees until he finds one who knows how to perform all tasks [6].

Porter's value chain, as shown in Figure 3, is defined by the set of activities that a company / organization performs in order to distribute a product or service to the market.

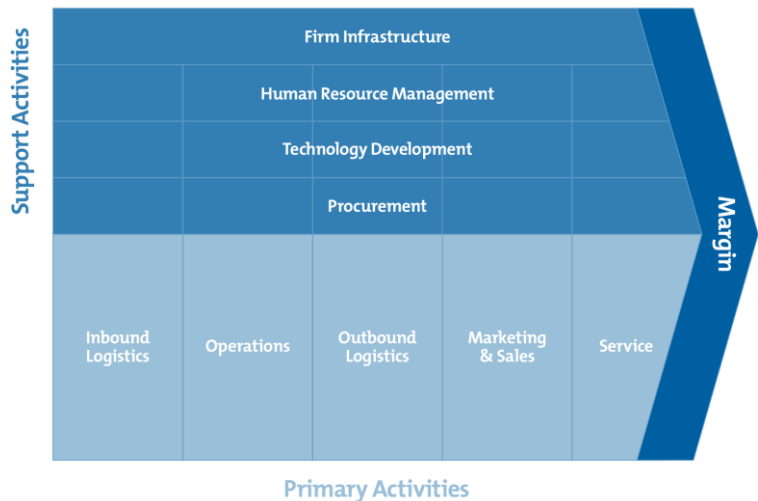


Figure 3 - Porter's value chain

Products go through an order of activities (the chain) where each activity adds value to the products. For example, the activity of cutting diamonds, despite having a low cost, the activity itself provides enormous value, since a rough diamond has a much lower value than a processed one. Thus, since the activities themselves are important to assign value to the products, then the optimization and reduction of costs and time of activities will add even more value. Porter's value chain allows you to model a company's activities as a sequence by analyzing the links between activities, looking for ways to improve or optimize them. If in the analysis of the activities it is found that one of them is stopped due to another needing the same resource, then it can be concluded that an optimization possibility would be the availability of additional relevant resources [7].

“People naturally network as they work so why not model itself as network” (V.Allee)

According to the two models presented, the Verna Allee’s model seems to be the one that best fits with this project, since the exchange of information and value as one of the central aspects and is one of the reasons why its implementation brings several advantages for all your actors. The partnerships and relationships between health professionals, clinics that hold a high amount of data from their patients and other stakeholders would be the basis of success in a business linked to this project, as it would be through a web of influencing factors that it would be possible to discover knowledge, more specifically, the correlation of patterns

of amino-acid profiles to the development of cancer, contributing to an early detection of this disease that would benefit all the actors of this system.

2.2.2 Value Proposition

Develop a decision support system that helps more effectively in the classification of cases with colorectal cancer based on changes in the plasma amino acid profile, with a high accuracy rate, to ensure the reliability of the forecast.

In this way, providing users, medical personnel who handles patients' clinical data, a diagnostic system, which will ensure the early detection of colorectal cancer, as well as the contribution to more accurate methods in the selection of patients and new strategies to improve therapeutic response.

With the increase in the amount of correctly catalogued patient data, the greater the volume of data to train the machine (or deep) learning model on which the system is based, with constant updates being predictable over time that will make it increasingly effective in the detection and support of medical decision.

In short, an early detection of a malignant tumour, aid in supporting the decision of associated therapies, with the consequence of improving the treatment performed, reducing possible side effects, which translate into a significant improvement in the quality of life and greater control of the disease.

There are many studies using machine and deep learning technology for cancer diagnoses, that are presented in chapter 1.4.2 - *Existing solutions and approaches in cancer diagnosis*. Nevertheless, there are not many systems in the market, but there are already companies that using Deep Learning (DL) and Machine Learning (ML) platforms, have developed specialized algorithmic solutions to analyse medical images for oncology for faster and better than human accuracy. Companies like Ibex-AI (<https://ibex-ai.com/>) and CancerCenter (<https://cancercenter.ai/>). The principal differentiation factor in comparison to these companies, are that this project will have its machine/ deep learning algorithm learning process based on changes in the plasma amino acid profile and the input data for this learning will be in the format of value data (age value, levels of specific amino acids) instead of images, like is the case in the listed companies.

2.2.3 Perceived Value

This concept represents, according to the needs and expectations, the differences between the benefits and sacrifices perceived by customers [8].

Perceived value is measured according to the product's usefulness to the customer, and by the ratio of expected benefits to the necessary sacrifices, by the quality and effort of acquisition [4].

The definition of perceived value varies according to the type of product or service. However, the definition of perceived value is measurable from a behavioral and utilitarian

perspective. From a behavioral perspective, the definition of perceived value is the "global assessment of the usefulness of a product based on perceptions of what is received versus what is provided". From a utilitarian perspective, perceived value is calculated based on utility, acquisition, and transaction.

The service provided offers information on decision support in the health area, with the possibility of diagnosing cancers in early stage. Since this service is still rare, it further enhances the perceived value on the part of the customer.

2.2.4 Value for the customer

This concept represents the advantage that arises between personal association with the organizational offer. It can occur through the reduction of sacrifices, the existence of benefits, the result obtained through the weighted combination of sacrifice and benefit [9].

As already mentioned, the value of the product offered is quantified in a balance between benefits and sacrifices for the customer before, during and after the purchase of the product or service [4]. The way a customer evaluates a product or service varies from customer to customer. This variation in perceived value is so fragile that the perceived value of that product or service varies from a longitudinal perspective. The customer's perception of value of the product or service changes during the acquisition and use process. A customer has an idea of the value of the item before buying it, it can vary at the time of purchase, due to several factors, such as price or construction. The perception of value can also change after purchase and also after using the product or service.

Table 1 presents the longitudinal perspective of the value with the benefits and sacrifices for the client in this context.

Table 1 - Longitudinal perspective of value

	Benefits	Sacrifices
Before Purchasing	<ul style="list-style-type: none"> • Reliability • Less risk 	<ul style="list-style-type: none"> • Price
Transaction	—	<ul style="list-style-type: none"> • Acquisition costs
After Purchasing	<ul style="list-style-type: none"> • Customization • Support 	<ul style="list-style-type: none"> • Learning time
Utilization	<ul style="list-style-type: none"> • Early diagnosis / detection of cancer • Reliability 	<ul style="list-style-type: none"> • Effort to insert new data • Updates cost

2.2.5 Canvas Business Model

For an idea to become a successful business it is necessary to deliver value to the customer in order to generate the expected result. For this to happen, it is necessary that the entire structure behind the business is directed towards this. In short, you must clearly identify the target audience, that is, the customer, the problem you are trying to solve, your resolution strategy and how this business will remain structured, generating profit.

The analysis of the value of a given product allows to identify the reason why the market should accept the product that we are analyzing at the expense of a possible competitor. In this sense, the canvas business model, presented in Figure 4, is a great help to demonstrate a value proposal, describing the strategic plan in a quick, agile and elucidative way.

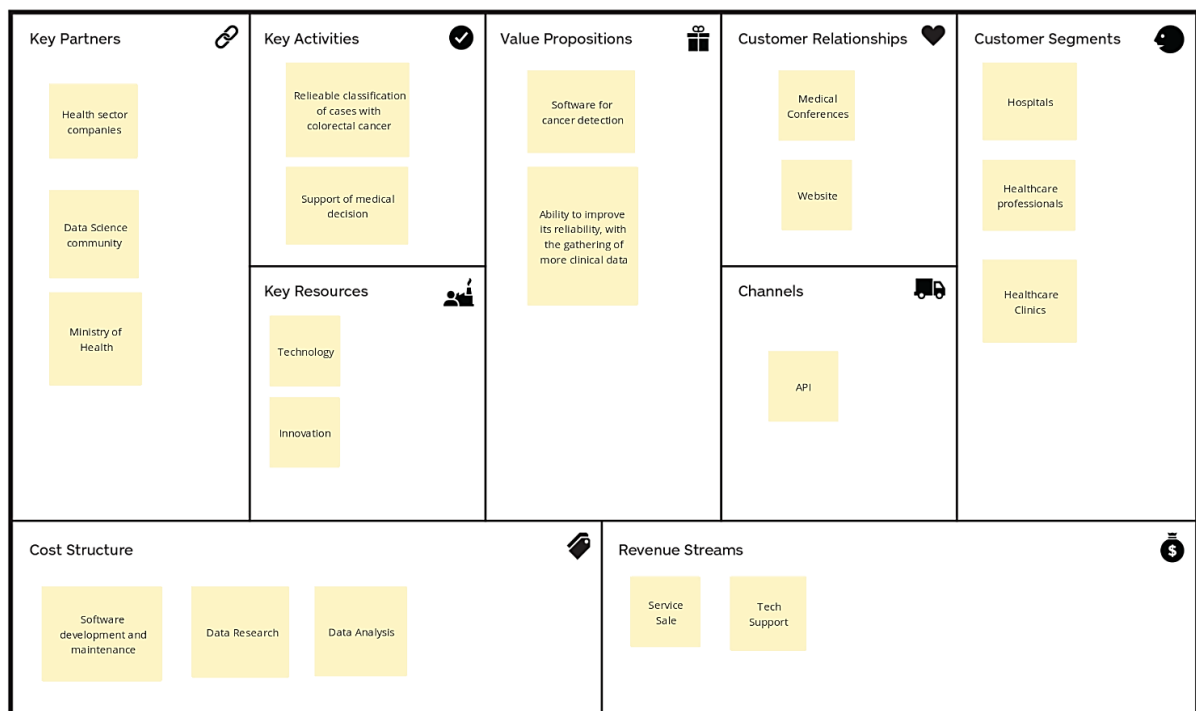


Figure 4 - Canvas Business Model

The business idea that can be associated with the theme of this dissertation is the commercialization of a service capable of assisting in the early detection of cancer based on the patient's clinical data (such as age, gender, etc.), with a special focus on their plasmatic amino acid profile.

The proposal is to conduct a study of several classification algorithms in the area of deep learning, which will be used in a decision support system that helps more effectively in the classification of cases with colorectal cancer.

For the study and construction of the best algorithm, access to a wide range of clinical patient data will be necessary, which will be provided through cooperation with Key Partners, of which protocols with the Ministry of Health, companies related to the medical research area and reliable data science communities.

This business plan has costs associated with infrastructure, payments to software engineers for software development and maintenance, in addition to the costs associated with possible acquisitions, collection and processing of data to be used in the construction of the deep learning algorithm.

The service, based on the best classification algorithm, would be possible to be used through an online API with authentication. The revenue stream would be based on the sale of the service for a certain period of time and the necessary technical support, if necessary a specific integration with the client's existing software.

Context and state of the art

3.1 Colorectal Cancer

Cancer is defined as a group of diseases characterized by uncontrolled growth and spread of abnormal cells. This pathology can be caused by both external and internal factors that may act together or in sequence to initiate or promote cancer [10].

Colorectal cancer (CRC) remains a serious health concern. As is shown in Figure 5, CRC was the third most commonly diagnosed cancer in 2020 in the 27 countries constituting the European Union. [11]

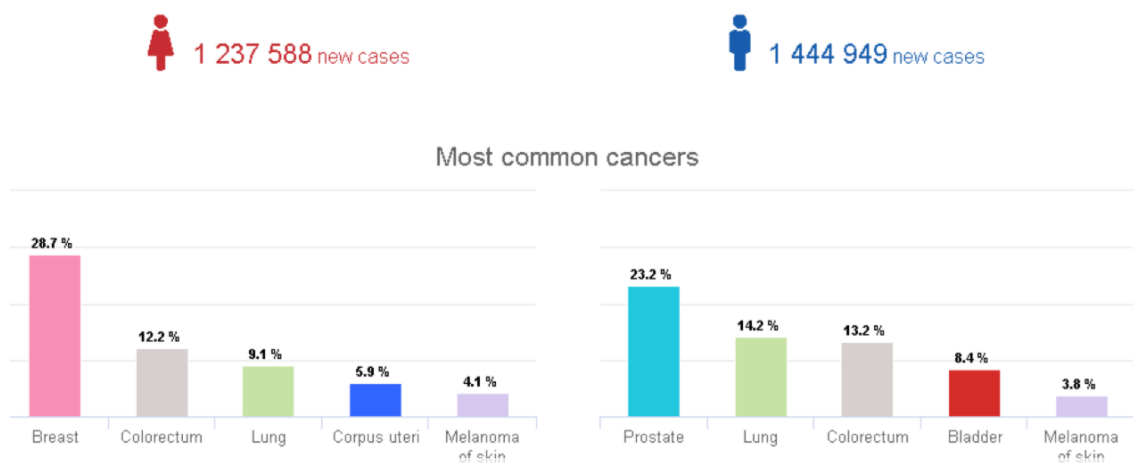


Figure 5 - 2020 cancer incidence in EU [11]

3.1.1 Amino acid profile in CRC detection

Amino acids are the building blocks of proteins and are additionally utilised as a source of energy. They are necessary for the synthesis of a wide variety of compounds, including neurotransmitters, haem and DNA. Although circulating concentrations of amino acids are subject to homeostatic control, they are also affected by diet, metabolism, lifestyle and genetic factors. An abundant supply of amino acids is important for cancers to sustain their proliferative drive. Alongside their direct role as substrates for protein synthesis, they can have roles in energy generation, driving the synthesis of nucleosides and maintenance of cellular redox homeostasis. As cancer cells exist within a complex and often nutrient-poor microenvironment, they sometimes exist as part of a metabolic community, forming relationships that can be both symbiotic and parasitic. Indeed, this is particularly evident in cancers that are auxotrophic for particular amino acids [12].

Several studies have demonstrated significant changes in the plasma amino-acid profiles of cancer patients without cachexia. This suggests that multivariate analysis of amino-acid profiles may be useful for the early detection of cancer [3].

In addition to the possible immunological effects, there is evidence that cancers originating from different organs might lead to different alterations of the amino-acid profile. The results presented demonstrate that plasma free amino-acid profiling is useful for detecting colorectal cancer [3].

3.2 Machine Learning

3.2.1 Overview

A machine learning algorithm is a computational process that uses input data to achieve a desired task without being literally programmed (i.e., “hard coded”) to produce a particular outcome.

These algorithms are in a sense “soft coded” in that they automatically alter or adapt their architecture through repetition (i.e., experience) so that they become better and better at achieving the desired task. The process of adaptation is called training, in which samples of input data are provided along with desired outcomes. The algorithm then optimally configures itself so that it cannot only produce the desired outcome when presented with the training inputs, but can generalize to produce the desired outcome from new, previously unseen data. This training is the “learning” part of machine learning. The training does not have to be limited to an initial adaptation during a finite interval. As with humans, a good algorithm can practice “lifelong” learning as it processes new data and learns from its mistakes. There are many ways that a computational algorithm can adapt itself in response to training. The input data can be selected and weighted to provide the most decisive outcomes. The algorithm can have variable numerical parameters that are adjusted through iterative optimization. It can have a network of possible computational pathways that it arranges for optimal results. It can determine probability distributions from the input data and use them to predict outcomes [13].

3.2.2 Existing solutions and approaches in cancer diagnosis

The early diagnosis and prognosis of a cancer type have become a necessity in cancer research, as it can facilitate the subsequent clinical management of patients. The importance of classifying cancer patients into high or low risk groups has led many research teams, from the biomedical and the bioinformatics field, to study the application of machine learning (ML) methods. Therefore, these techniques have been utilized as an aim to model the progression and treatment of cancerous conditions. In addition, the ability of ML tools to detect key features from complex datasets reveals their importance.

As is stated in Figure 6 and Figure 7, a variety of these techniques, including Artificial Neural Networks (ANNs), Bayesian Networks (BNs), Support Vector Machines (SVMs) and Decision Trees (DTs) have been widely applied in cancer research for the development of predictive models, resulting in effective and accurate decision making [14].

Method	Cancer type	No of patients	Type of data	Accuracy	Validation method	Important features
ANN	Breast cancer	62,219	Mammographic, demographic	AUC = 0.965	10-fold cross validation	Age, mammography findings
SVM	Multiple myeloma	80	SNPs	71%	Leave-one-out cross validation	snp739514, snp521522, snp994532
SVM	Breast cancer	174	SNPs	69%	20-fold cross validation	snpCY1182 (+) 4536 T/C snpCYP181 (+) 4328 C/G
BN	Colon carcinomatosis	53	Clinical, pathologic	AUC = 0.71	Cross-validation	Primary tumor histology, nodal staging, extent of peritoneal cancer

Figure 6 - ML methods used in relevant publications for cancer susceptibility prediction.

ML method	Cancer type	No of patients	Type of data	Accuracy	Validation method	Important features
BN	Oral cancer	86	Clinical, imaging tissue genomic, blood genomic	100%	10-fold cross validation	Smoker, p53 stain, extra-tumor spreading, TCAM, SOD2
SVM	Breast cancer	679	Clinical, pathologic, epidemiologic	89%	Hold-out	Local invasion of tumor
Graph-based SSL algorithm	Colon cancer, breast cancer	437 374	Gene expression, PPIs	76.7% 80.7%	10-fold cross validation	BRCA1, CCND1, STAT1, CCNB1
SVM	Cervical cancer	168	Clinical, pathologic	68%	Hold-out	pathologic_S, pathologic_T, cell type RT target summary
SVM	Breast cancer	547	Clinical, population	95%	10-fold cross validation	Age at diagnosis, age at menarche

Figure 7 - ML methods used in relevant publications for cancer recurrence prediction

With the advent of new technologies in the field of medicine, large amounts of cancer data have been collected and are available to the medical research community. However, the accurate prediction of a disease outcome is one of the most interesting and challenging tasks for physicians. As a result, ML methods have become a popular tool for medical researchers. These techniques can discover and identify patterns and relationships between them, from complex datasets, while they are able to effectively predict future outcomes of a cancer type [14].

Recent years, there is a significant increase of computational power and rapid advancement in the technology of artificial intelligence, particularly in deep learning. In addition, the cost reduction in large scale next-generation sequencing, and the availability of such data through open source databases offer opportunities to possibly build more powerful and accurate models to predict cancer prognosis more accurately prognosis.

Figure 8 shows some deep learning methods used in relevant publications to build models for cancer prognosis prediction. Deep learning has been suggested to be a more generic model, requires less data engineering, and achieves more accurate prediction when working with large amounts of data [15].

Type of Cancer	Type of Data	Methods	Outputs	Validation	NN Model Performance
Melanoma	Clinical data of tumors	3 layers NN	Survival time	Not reported	Achieved similar performance as Cox and Kaplan Meier statistical methods
Breast cancer	Cell images to measure 30 nuclear features	3 layers NN	Survival time	10 fold cross validation	As good as conventional methods
Astrocytic brain tumor	A list of genes expression from microarray data	A single layer perceptron and an output (multiple binary models)	Tumor grades	Leave-one-out cross validation	44, 9 and 7 probe sets have achieved 93.3%, 84.6%, and 95.6% validation success rates, respectively.
10 types of cancer	TCGA gene expression data, clinical data and survival data	NN	Survival time	5-fold cross validation	Similar or in some cases better performance than Cox-PH, Cox-boosting or RF
Breast cancer	METABRIC ³ , 4 genes data and clinical information, GBSG ⁴ : clinical data	NN	Survival	20% of METABRIC patients used as test set GBSG has split test dataset	C-index: 0.654 for METABRIC and 0.676 for GBSG, both are better than CoxPH
Breast cancer, nasopharyngeal carcinoma	METABRIC, GBSG, NPC ⁷ : 8-9 clinical features	DNN	Survival	After removed patients with missing data, 20% used as test set	C-index: 0.661 for METABRIC and 0.688 for GBSG, both are better than CoxPH and DeepSurv. c-index ranges 0.681-0.704 depends on input data for NPC, better than CoxPH.

Figure 8 – Deep Learning methods used in relevant publications for cancer prognosis prediction [15]

3.3 Types of Learning

The ability to learn through input from the surrounding environment is the main key to developing a successful machine learning application [16].

As shown in Figure 9, Machine Learning algorithms can be divided into several categories according to the type of learning used.

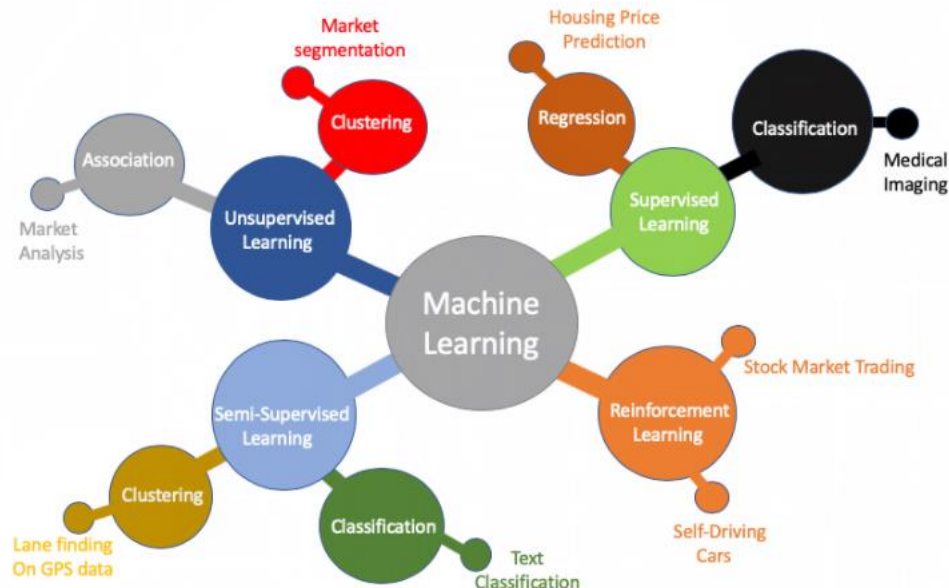


Figure 9 - Machine Learning Types [17]

3.3.1 Supervised Learning

Supervised learning algorithms are algorithms that use a set of data that contain characteristics (attributes), where each example is associated with a category (classes). The model is constructed by defining the classes and the examples of each class.

Based on prior knowledge learned from the training data set, the machine is expected to evaluate and predict the outputs (classes / categories) for input data it does not know. The idea is that the machine “learns” from the training experience, by the labeled values of the data set, being able after identifying / classifying unlabeled examples with high precision [18].

According to Jason Brownlee [19], supervisory learning problems can be grouped into two types of algorithms:

- Classification: algorithm that classifies the result of a forecast, that is, when the output variable is a category / label;
- Regression: algorithm that tries to predict a given result based on the previous variables.

3.3.2 Unsupervised Learning

The unsupervised learning algorithms learn few features from the data. When new data is introduced, it uses the previously learned features to recognize the class of the data. It is mainly used for clustering and feature reduction.

Unsupervised learning algorithms are algorithms that use a data set that contains many characteristics, and then learn the useful properties of the structure of that data set. The goal of unsupervised learning is to model the structure or distribution underlying the data in order to learn more about it. Unlike supervised learning, these algorithms do not have information about the class associated with each example, and learning is done through the discovery of similarities in the data (clusters of data with similar characteristics) [18].

According to Jason Brownlee [19], unsupervised learning problems can be grouped into the following types of algorithms:

- Clustering: a clustering problem is where we want to discover the clusters inherent in the data, such as the grouping of customers by purchasing behavior;
- Association: it is an algorithm that tries to learn through association rules, in large portions of the data, for example, people who buy product X also tend to buy product Y.

3.3.3 Semi-Supervised Learning

Semi – supervised learning algorithms is a technique which combines the power of both supervised and unsupervised learning. It can be fruit-full in those areas of machine learning and data mining where the unlabelled data is already present and getting the labelled data is a tedious process [15].

In such a scenario, the labelled part can be used to aid the learning of the unlabelled part. This kind of scenario lends itself to most processes in nature and more closely emulates how humans develop their skills. There are two particularly important advantages to a successful algorithm. First, it can substitute for laborious and repetitive human effort. Second, and more significantly, it can potentially learn more complicated and subtle patterns in the input data than the average human observer is able to do.

3.3.4 Reinforcement Learning

Reinforcement learning is a type of learning which makes decisions based on which actions to take such that the outcome is more positive. The learner has no knowledge which actions to take until it is given a situation. The action which is taken by the learner may affect situations and their actions in the future. Reinforcement learning solely depends on two criteria: trial and error search and delayed outcome [20].

3.4 Machine Learning Algorithms

This dissertation will focus in the two types of Machine Learning most used in the Healthcare area, the supervised and unsupervised learning. The supervised learning deals deal with labelled data while the unsupervised learning deals with unlabelled data. Figure 10 presents an exemple of algorithms related with those two types of machine learning.

Machine learning algorithms are organized into a taxonomy based on the desired outcome of the algorithm. Supervised learning generates a function that maps inputs to desired outputs.

In this section, some of the main supervised learning algorithms are presented, more specifically classification algorithms.

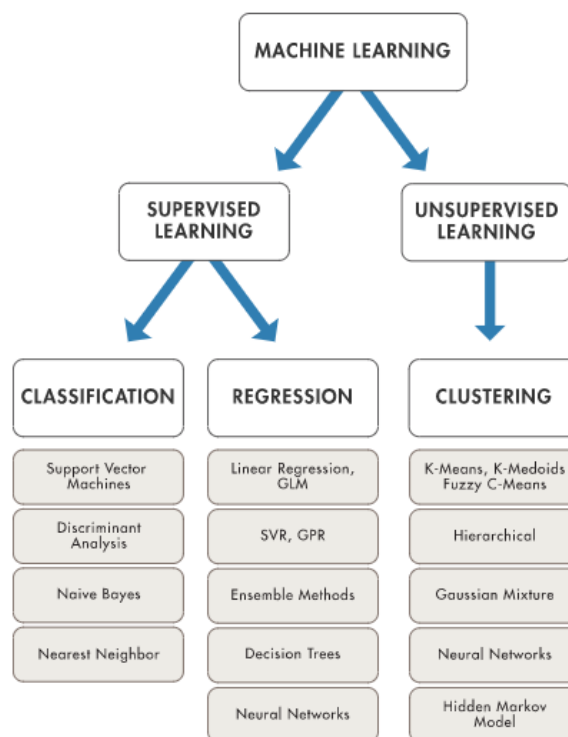


Figure 10 - Machine Learning Algorithms

3.4.1 Decision Tree

Decision trees are those type of trees which groups attributes by sorting them based on their values. Decision tree is used mainly for classification purpose. Each tree consists of nodes and branches. Each nodes represents attributes in a group that is to be classified and each branch represents a value that the node can take [21]. An example of decision tree is shown in Figure 11.

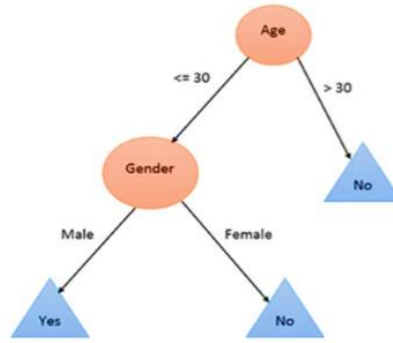


Figure 11 - Decision Tree [22]

3.4.2 Random Forest (RF)

A random forest is an ensemble classifier and consisting of many DTs similar to the way a forest is a collection of many trees. The different DTs of an RF are trained using the different parts of the training dataset. To classify a new sample, the input vector of that sample is required to pass down with each DT of the forest. Each DT then considers a different part of that input vector and gives a classification outcome. The forest then chooses the classification of having the most 'votes' (for discrete classification outcome) or the average of all trees in the forest (for numeric classification outcome) [23].

3.4.3 Naïve Bayes

Naïve Bayes mainly targets the text classification industry. It is mainly used for clustering and classification purpose. This model is called 'naïve' because it assumes independency between all measured attributes. In real world, data attributes are not always perfectly independent. With this assumption, conditional probabilities of attributes can be easily computed as long as all values of attributes are given. In Naïve Bayes model, the model designer is required to decide which attributes to view as dependent attributes to other attributes when computing conditional probabilities. In practice, in some cases, the model works surprisingly well even with this independency assumption, but in other cases, it totally fails to work. The model solely depends on characteristics of data. In Figure 12 is illustrated the Bayes' theorem in which this algorithm is based. [24]

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Likelihood
Class Prior Probability
Posterior Probability
Predictor Prior Probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Figure 12 - Bayes Theorem [24]

3.4.4 K-Nearest Neighbors (kNN)

This algorithm consists of a set of techniques that can be used for classification or regression. It is a non-parametric learning algorithm, that is, it does not make any assumptions in the distribution of the underlying data, the structure of the model being determined from them [18].

Therefore, it can be assumed that this algorithm is recommended for a classification study when there is insufficient knowledge about the data. In the classification using this algorithm, the result is a class association. An object is classified by the majority of its neighbors' votes, the object being assigned to the most common class among its closest neighbors [25].

3.4.5 Logistic Regression

Logistic regression algorithms are appropriate to perform a regression analysis when the dependent variable takes binary values (0 or 1). Like other types of regression, logistic regression is widely used to make predictions, but unlike other types, logistic regression only allows forecasts to be obtained in a binary form. The fact that the result is a value between 0 and 1 makes these algorithms very useful when you want to obtain a probability, because the probabilities are also a value between 0 and 1. This model uses the logistic function which is a sigmoid function hence the binary nature of its output. When the dependent variable is binary, linear regression cannot be used [26].

3.4.6 Support Vector Machine (SVN)

The Support vector machines (SVM) algorithm is a very powerful classification algorithm, developed by Cortes and Vapnik, being one of the most influential approaches to supervised learning. It is a very popular algorithm in machine learning for pattern recognition, especially for binary classification, that is, classification for two classes [27].

The learning machine receives a set of training input data, belonging to two classes, with associated categories. The input data is in the form of attribute vectors and SVM finds the hyperplane separating the input data and leaving the best margin of separation between them, as can be seen in Figure 13. If the data are not linearly separable, the data points are projected in a generally larger space, where the data points become effectively linearly separable [29].

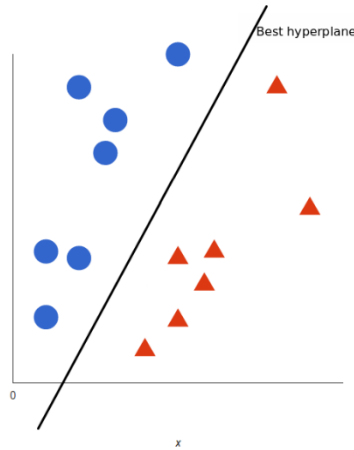


Figure 13 – Support Vector Machine [28]

3.4.7 Artificial Neural Networks (ANN)

The neural network (or artificial neural network or ANN) is derived from the biological concept of neurons.

In the creation of a neural network, it is essential to correctly determine which input data, how many layers can be used and what type of activation function will be implemented. In this way, an ANN is characterized by its architecture, its processing algorithm and its learning algorithm. The architecture specifies the number of neurons as well as how they are connected. The processing algorithm specifies how the network calculates the outputs for a given set of inputs, using a set of weights. Finally, the learning algorithm specifies how the network adapts its weights based on the training input data sets [30].

The basic unit of computation in a neural network is a neuron, commonly referred to as a “node” or “unit.” As shown in Figure 14, the node receives input from other nodes or receives input from an external source and then calculates the output. Each input is complemented with “weight” (w), the weight of which depends on the relative importance of the other inputs. The node applies a function (activation function) to the weighted input sum.

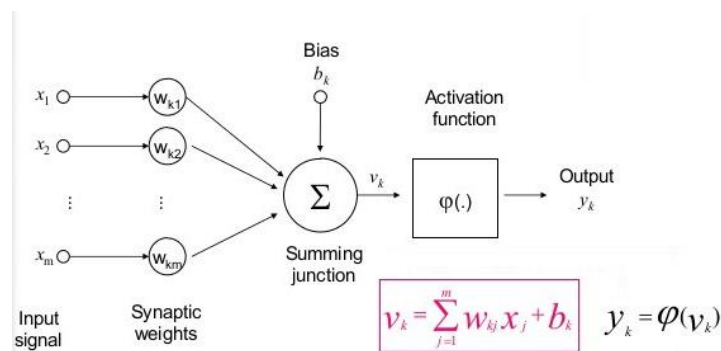


Figure 14 - Structure of an Artificial Neuron

Biasing provides a trainable constant value for each node (beyond the normal input received by the node) and the activation function introduces nonlinearity into the output of the neuron. Because most of the real world data is non-linear, is important that neurons can learn non-linear functional representations.

The main advantage of ANNs is their adaptability, since they can learn from the examples presented to them, often finding very detailed relationships between the data, which are lost even by experts. This feature is very useful for problems where there are many potential parameters that can affect the solution. However, they bring some disadvantages, such as the lack of a definitive way of choosing the ideal architecture and finding the best solution, in addition to always depending on the accuracy of the training set [30].

3.4.8 Algorithms Comparison

The advantages and limitations of different supervised machine learning algorithms are shown in Table 2.

Table 2 - Advantages and limitations of different supervised machine learning algorithms [23]

Supervised Algorithm	Advantages	Limitations
Artificial neural network (ANN)	<ul style="list-style-type: none"> • Can detect complex nonlinear relationships between dependent and independent variables. • Requires less formal statistical training. • Availability of multiple training algorithms. • Can be applied to both classification and regression problems. 	<ul style="list-style-type: none"> • Have characteristics of 'black box' - user cannot have access to the exact decision-making process and therefore, • Computationally expensive to train the network for a complex classification problem. • Predictor or Independent variables require pre-processing.
Decision tree (DT)	<ul style="list-style-type: none"> • Resultant classification tree is easier to understand and interpret. • Data preparation is easier. • Multiple data types such as numeric, nominal, categorical are supported. • Can generate robust classifiers and can be validated using statistical tests. 	<ul style="list-style-type: none"> • Require classes to be mutually exclusive. • Algorithm cannot branch if any attribute or variable value for a non-leaf node is missing. • Algorithm depends on the order of the attributes or variables. • Do not perform as well as some other classifier (e.g., Artificial Neural Network)

Supervised Algorithm	Advantages	Limitations
K-nearest neighbour (KNN)	<ul style="list-style-type: none"> • Simple algorithm and can classify instances quickly. • Can handle noisy instances or instances with missing attribute values. • Can be used for classification and regression. 	<ul style="list-style-type: none"> • Computationally expensive as the number of attributes increases. • Attributes are given equal importance, which can lead to poor classification performance. • Provide no information on which attributes are most effective in making a good classification.
Logistic regression (LR)	<ul style="list-style-type: none"> • Easy to implement and straightforward. • LR-based models can be updated easily. • Does not make any assumptions regarding the distribution of independent variable (s). • It has a nice probabilistic interpretation of model parameters. 	<ul style="list-style-type: none"> • Does not have good accuracy when input variables have complex relationships. • Does not consider the linear relationship between variables. • Key components of LR - logic models, are vulnerable to overconfidence. • May overstate the prediction accuracy due to sampling bias. • Unless multinomial, generic LR can only classify variables that have two states (i.e., dichotomous).
Naïve Bayes (NB)	<ul style="list-style-type: none"> • Simple and very useful for large datasets. • Can be used for both binary and multi-class classification problems. • It requires less amount of training data. • It can make probabilistic predictions and can handle both continuous and discrete data. 	<ul style="list-style-type: none"> • Classes must be mutually exclusive. • Presence of dependency between attributes negatively affects the classification performance. • It assumes the normal distribution of numeric attributes.

Supervised Algorithm	Advantages	Limitations
Random forest (RF)	<ul style="list-style-type: none"> • Lower chance of variance and overfitting of training data compared to DT, since RF takes the average value from the outcomes of its constituent decision trees. • Empirically, this ensemble-based classifier performs better than its individual base classifiers, i.e., DTs. • Scales well for large datasets. • It can provide estimates of what variables or attributes are important in the classification. 	<ul style="list-style-type: none"> • More complex and computationally expensive. • Number of base classifiers needs to be defined. • It favours those variables or attributes that can take high number of different values in estimating variable importance. • Overfitting can occur easily
Support vector machine (SVM)	<ul style="list-style-type: none"> • More robust compared to LR • Can handle multiple feature spaces. • Less risk of overfitting. • Performs well in classifying semi-structured or unstructured data, such as texts, images etc. 	<ul style="list-style-type: none"> • Computationally expensive for large and complex datasets. • Does not perform well if the data have noise. • The resultant model, weight and impact of variables are often difficult to understand. • Generic SVM cannot classify more than two classes unless extended.

3.5 Deep Learning

Deep learning (deep structured learning, deep neural networks or deep machine learning) consists of multiple layers of artificial neurons (as is shown in Figure 15) that mimic neurons in human brain. Similar to linear regression, each neuron has a weight value that is updated by gradient descent algorithm during backpropagation to minimize global loss function. By applying nonlinearity using activation function to the multiple layers of each neuron, more abstract mathematical relationship was extracted from the input data to map to the output. A well-trained model can therefore be used to predict new unlabeled data [16].

Deep learning is a branch of machine learning, and therefore inherits some common knowledge foundation in machine learning, including basic probability and statistics, loss/cost function and etc., but in the meantime has more flexibility and can be built towards more complex layers and multiple neurons in each layer to have better predictive power.

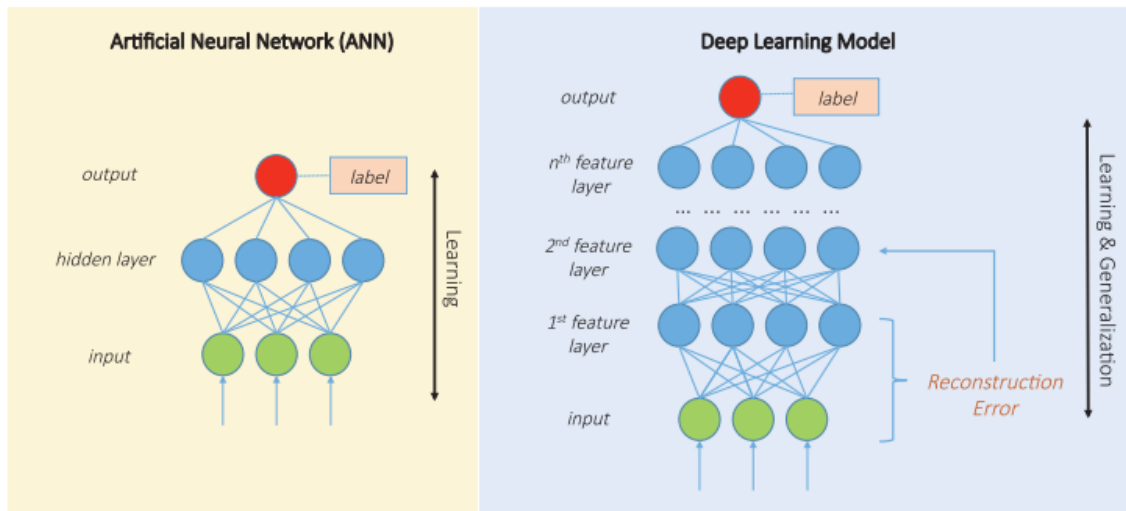


Figure 15 – Simple Neural Network vs Deep Learning Neural Network [31]

One of the potentials of deep learning is replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction [32].

Compared with other machine learning techniques in the literature, deep learning has witnessed significant advances. These successes have prompted researchers in the field of computational medical imaging to investigate the potential of deep learning in medical images acquired with, for example, CT, MRI, PET, and X-ray [33].

Deep learning can serve as a guiding principle to organize both hypothesis-driven research and exploratory investigation in clinical domains based on different sources of data.

3.6 Machine Learning vs Deep Learning

In both cases, data is the most important component, it is the quality of data which ultimately determines the quality of the result. In Table 3 some differences in these types of technologies are presented to help in its selection for the use in this dissertation.

Table 3 – Differences between Machine Learning and Deep Learning [34]

	Machine Learning	Deep Learning
Data Dependencies	Works on a smaller amount of dataset for accuracy, generally a few thousand instances	Works on a large amount of dataset, generally with millions of instances
Time to Train	Takes less time to train (ranging from a few minutes to a few hours)	Takes longer time to train (take a couple of days to train completely from

		scratch)
Human Intervention	Requires features engineering, needing to be fed with structured data (represented in the figure X)	No feature engineering is required, nor is a necessity for structured data and minimal human intervention which minimizes the likelihood of human bias to the model
Hardware Requirements	Needs much less computational power (CPUs meet the performance needs)	Need to have high end infrastructure to train in reasonable time (GPUs are suited to meet the performance needs)
Output	The output is usually a numeric value, like a score or a classification	The output can be anything from a score, an element, free text or sound

When the learning data is small, machine learning algorithms outperform deep learning because deep learning thrives on big data to understand it and generalization error bound shrinks as the training data size increases. On the other hand, machine learning algorithms with their handcrafted rules start to collapse as size of the data increases. [35]

3.6.1 Decision

In this dissertation the initial intent was to use both Machine Learning and Deep Learning to generate models able to predict the colorectal cancer. Nevertheless, at a mid-stage in the progress of this work due to the volume of our dataset not being **not very large**, it was decided that the experiences will be done using **only the machine learning approach** and using the machine learning algorithms presented in the *Machine Learning Algorithms* chapter.

3.7 Machine Learning Libraries

Machine Learning libraries help developers to define models in precise, transparent, and concise ways.

Each library has different features and performance characteristics. Also, each framework tries different techniques to optimize its implementation of ML algorithms. So, although the same algorithm is implemented in different frameworks, the performance of the different implementations can vary greatly.

This section will present some of most used machine learning libraries.

3.7.1 TensorFlow

TensorFlow is an open source library used primarily for deep machine learning. It was originally developed on by Google's divisions, but in 2015 it was released as free open source software under the Apache License 2.0. The computational core is written in C++ (60% of all

code) using CUDA technology, which allows one to utilize graphics cards in calculations. The interface part is implemented in Python (30% of all code base). There are also unofficial bindings for other languages, but only C++ and Python interfaces are officially supported. The library is based on the principle of data flows(dataflow), according to which the program is organized in the form of computational blocks associated with each other in the form of a directed graph which is called computational graph. Data is processed by passing from one block to another. Such application architecture makes it easy to use parallel calculations on both multi-core CPUs and distributed cluster systems. In addition, it is well suited for building neural networks in which each neuron is presented by an independent component. In addition to the computational graph, TensorFlow uses a data structure called tensor. It is similar to the tensor from differential geometry in the sense that it is a multidimensional array [36].

3.7.2 PyTorch

The PyTorch library was created on the basis of Torch. The original Torch library was developed in C and used Lua as the interface. With the growth of Python popularity in machine learning, Torch has been rewritten in C++11/CUDA (60% code) and Python (32% code). Initial development was conducted in the company of Facebook, but currently PyTorch is an OpenSource library, distributed under a BSD-like license. PyTorch, as well as TensorFlow, is built on the basis of dataflow concept. The main difference from TensorFlow is that in TensorFlow computational graph is static, then in PyTorch the graph is dynamic. This means that one can modify the graph on the fly, adding or removing nodes as needed. In TensorFlow, the entire graph must be specified before the model run. The developers of PyTorch emphasize that Python is tightly integrated into the library (library is more pythonic). This makes it easier to use than TensorFlow, as the user does not have to dive into low-level parts written in C++. It is worth noting, however, that TensorFlow surpasses PyTorch in popularity, as it appeared earlier and is used in many educational courses on machine learning [36].

3.7.3 Keras

The Keras library provides a high-level programming interface for building neural networks. It can work on top of TensorFlow, Microsoft Cognitive Toolkit (CNTK) or Theano. The library is written entirely in Python and is distributed under the MIT license.

The library is based on the following principles: ease of use, modularity, extensibility. The modularity principle allows you to separately describe the neural layers, optimizers, activator functions, etc, and then combine them into a single model. The model is fully described in Python. The created model can be saved to disk for further use and distribution [36].

3.7.4 Scikit Learn

SciKit Learn is the library for data processing. It implements various methods of classification, regression analysis, clustering and other algorithms related to classical machine learning. It is written almost entirely in Python (98% of all code base), but uses NumPy and SciPy for algorithms implementation. The project is very stable, as it has been developing since

2007. SciKit Learn is suitable for traditional machine learning and data preprocessing tasks. This library does not support the concept of dataflow and does not allow one to create his own models. The absence of a computational graph does not allow flexible scale of models for multi-core processors and graphics accelerators and forces to limit the degree of parallelism that is implemented in NumPy [36].

3.7.5 Decision

The table 4 presents in a condensed form the comparison between the machine learning libraries presented.

Table 4 - Machine Learning Libraries Comparison [36]

Property	Scikit Learn	TensorFlow	PyTorch	Keras
Core	Python	C++	Python	Python
Cost	Free Open Source	Free Open Source	Free Open Source	Free Open Source
Platform	Linux, Windows, MacOS. Does not support GPU computing.	Linux, Windows, MacOS. Nvidia GPUs and CUDA recommended	Linux, Windows, MacOS. CPUs and Nvidia GPUs	Linux, Windows, MacOS. Backend: CNTK, TensorFlow, etc
Pros	Great for beginners and for those looking to explore machine learning algorithms.	Continued Community Support. Superior Computational Graph Visualizations.	Lots of pretrained models. Much better suited for small projects and prototyping.	Prototyping is facilitated to the limit (few lines of code). It's ideal for learning and prototyping simple concepts
Cons	It's only suitable for small projects with small datasets, and for tasks that are not particularly computationally intensive.	Hard to learn, much code to create a model.	No commercial support. In relation to cross-platform solutions, TensorFlow looks like a more suitable choice.	Less configurable environment than low-level frameworks.

Due to my inexperience with machine learning, one of the drivers for this library selection is connected to the easiness of use and community so that any difficulty is more easily overcome with some search.

Having this in mind the **choice made was to use the Scikit-Learn library**, since Scikit-Learn is a higher-level library that is mainly used for machine learning and that includes implementations of several machine learning algorithms, being easy to define a model object in few lines of code, then use it to predict a value.

Design

This chapter contains information about the system view, from a structural point of view, analyzing the intended features, the architecture and alternatives to the architecture of the system.

Then the approach considered to solve the problem is presented. Still in this chapter, it is possible to view the implementation of the proposed solution that contains details about each one of the procedures performed in order to solve the problem.

4.1 Functionalities

The two distinct roles identified, the system content manager and the common user, and use cases for the system are presented in Figure 16.

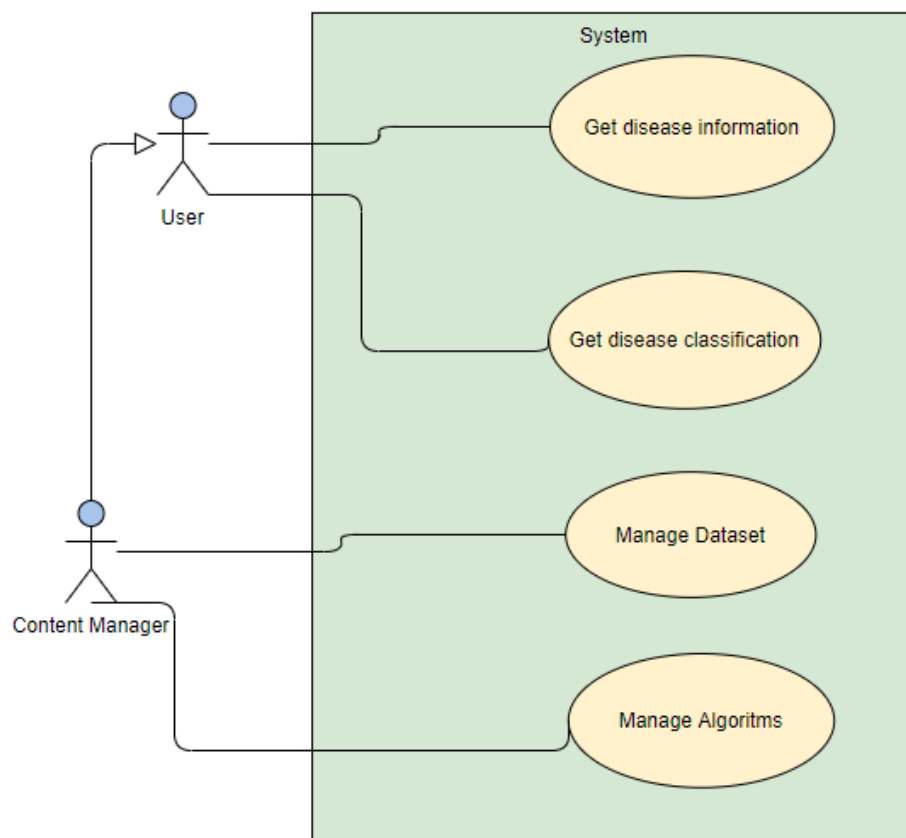


Figure 16 – System use cases diagram

The common user must have access to obtain the classification of a disease based on the clinical data submitted. There will be different types of classification, binary ones to predict the presence (or absence) of disease and multi classification to predict the specific disease. With the diagnostic, it will be also presented the percentage of accuracy of the prediction to

provide support for the user, that is expected to be a healthcare professional, in its decision-making process about the best course of action.

Besides that, if the user wants more information about the diagnosed disease (in case of being classified with one), a brief description can be provided by the system.

The content manager is a role that, in addition to the common user's functionalities, has access to two distinct functionalities that are very relevant to the System's operability, which are dataset management and algorithms management. The first is related to the possibility of adding new instances to the dataset, in addition to being able to remove or edit existing ones, so that models are generated based on an increasingly larger and more reliable volume of data. Algorithm management is related to the possibility of removing or adding the algorithms that are used to generate the model, so that the model's performance can be improved, as well as the time it takes to calculate the best of the generated models.

One of the features that have been discarded for the content manager role is model management, which would allow to designate which model the system must use for disease prediction, as it is considered that the best procedure is, based on existing data and defined algorithms, to generate several models and be selected by the system the one that presents the best performance.

4.2 System Architecture

In this section, it is intended to assess possible architectural approaches to respond to the projected system's use cases.

4.2.1 Planned Architecture

The system architecture is divided into 4 main components, represented in Figure 17.

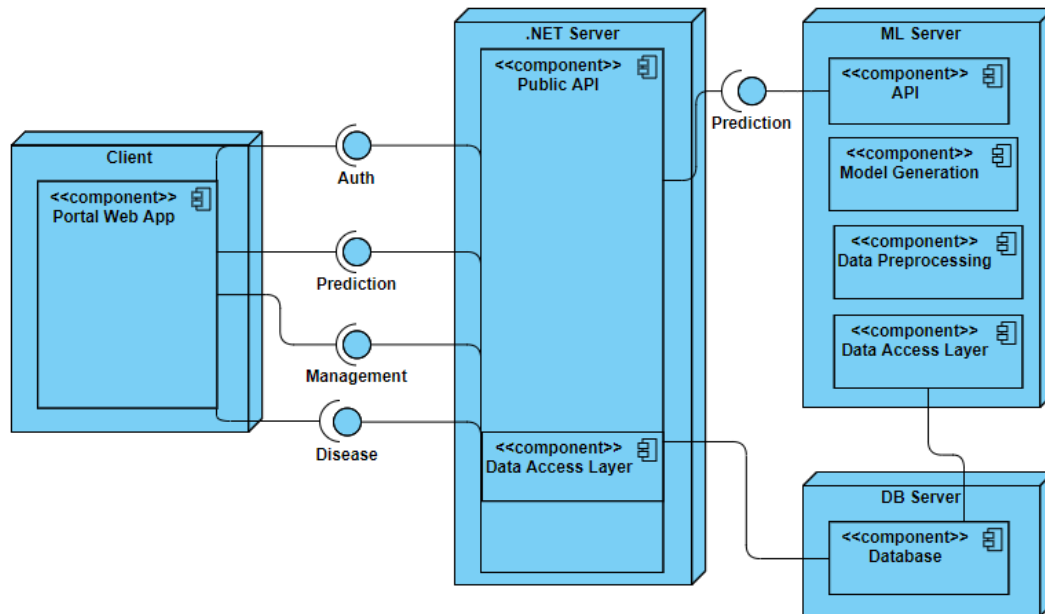


Figure 17_– System Architecture

One of the components consists of a web app, which will allow the user to interact with the system through a browser. To use the application, the user will have to authenticate, from which the type of user and their level of permissions will be determined.

Through the web app, the user will be able to enter the clinical data of the patient for which it intends to predict the existence of colorectal cancer (due to other models being studied, it is possible that metabolic diseases can also be predicted).

For this disease prediction/diagnosis to be carried out, a request will be made to another component of the system, the backend server, which is described as a .Net server, as it is one of the backend technologies I am most familiar with, but another backend technology can be used to build the system. This server works as a gateway for the remaining components that belong to the system's backend, and the web application's interaction with any of them is mediated by this component, providing 4 different endpoint groups under its responsibility:

- Auth: to handle the registration and authentication process.
- Prediction: to deal with the system's most important function, that is the disease diagnosis.
- Management: only accessible to users who are content managers, to allow management of training data and algorithms used.
- Disease: to make it possible to consult more information about a specific disease, especially information associated with colorectal cancer.

When predicting disease, the .Net server needs to communicate with the most important component of the system, the Machine Learning Server (ML Server). The ML server will expose an API for communicating with external components, in this architecture it only

receives requests from the .Net Server, but it is designed from scratch as an isolated component, to allow decoupling and facilitate its evolution for possible future communications with other external systems.

In this component is realized all the processing related to the loading of the data, its pre-processing and its use in the generation of models according to the existing algorithms. Furthermore, it is responsible for evaluating the generated models and choosing the best model to be used to provide the forecast based on the metabolic data received by the API.

The generation and choice of the best model can be triggered by having the API request as the process initiator or using a task scheduling strategy, which would be responsible for running this all process at a defined time interval, for when a prediction was requested by a user, the response time to be minimal.

Finally, we have the database where all data to be used by the system would be kept, such as users (and their permission levels), information about diseases and data for training algorithms and generating ML models.

In this architecture, this component communicates with the ML Server to provide data for the model generation models and with .Net Server to provide data related with authentication diseases description.

4.2.2 Prototype

The emphasis of this dissertation is on training machine learning models, so due to time constraints, it was not possible to create a system based in the Planned Architecture. Being only created a prototype to enable an easier interaction to test the generated model, whose architecture is depicted in the Figure 18.

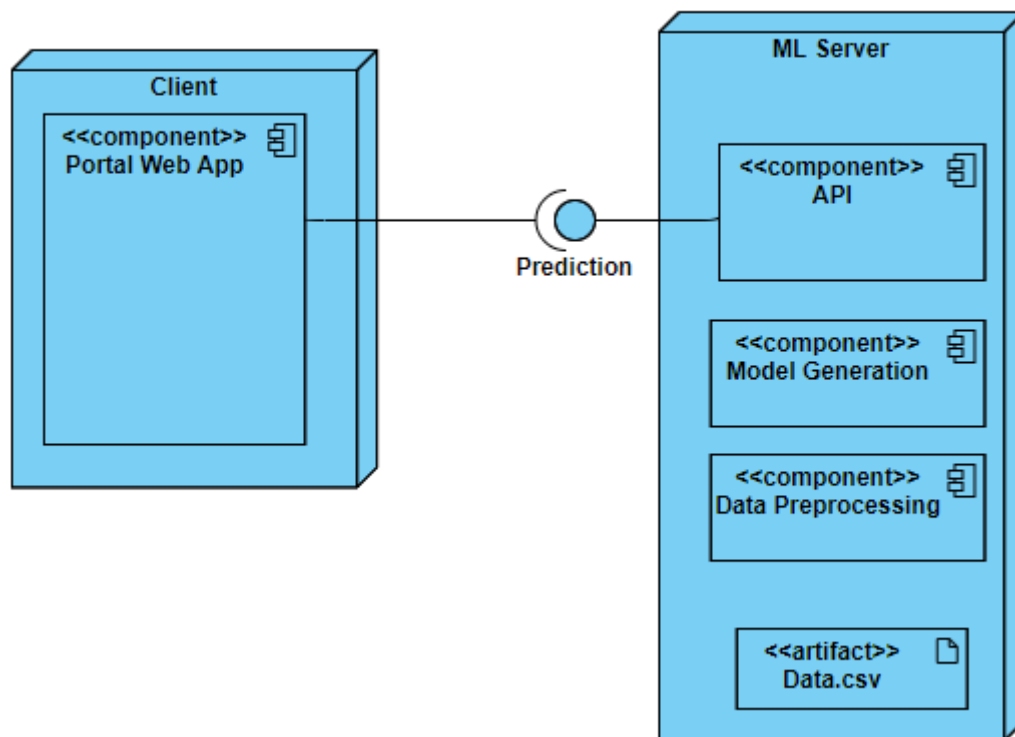


Figure 18 – Prototype architecture

In development of this prototype, the main technologies used were the following:

- Python, version 3.9;
- Scikit-learn, version 0.24.2;
- Pandas, version 1.1.5;
- Flask, version 2.0.1;
- Numpy, version 1.19.3;

The prototype is composed of two components a simple web app that allows the user to input the patient's clinical data. As shown in Figure 19, the user interface has a button to, after the data is inserted, request a prediction of diagnosis for colorectal cancer.

127.0.0.1:5000

Colorectal cancer disease detection

Predict the probability of having CLR cancer

Tau	Asp	Hyp	Thr	Ser	Asn
Glu	Gln	AAA	Pro	Gly	Ala
Cit	ABU	Val	Cys2	Met	Cysta
Ile	Leu	Tyr	Phe	Orn	Lys
1Mhis	His	3Mhis	Arg	Gender	Age

PREDICT PROBABILITY

Figure 19 - Web Application user interface

When the prediction is requested, the Web App communicates with the API of the Machine Learning Server, that based in the best model generated based in data stored in a .csv file, returns the prediction result. As shown in Figure 20, this result is presented to the user in the user interface.

PREDICT PROBABILITY

Your diagnostic is: Cancer detected

Select a file:

Escolher Ficheiro

NÃO FOI ESCOL...NHUM FICHEIRO

Submeter

MEI Dissertation

Previsão Inteligente das alterações metabólicas no cancro retal com base em modelos machine learning

Figure 20 – User interface response example

Experiences and Evaluation

In this chapter, the measures that will be used in the evaluation of the experiences made in the various classification models implemented are identified, as well as the hypotheses and evaluation methodologies used in them. The figure 21 shows the followed workflow of a machine learning algorithm.

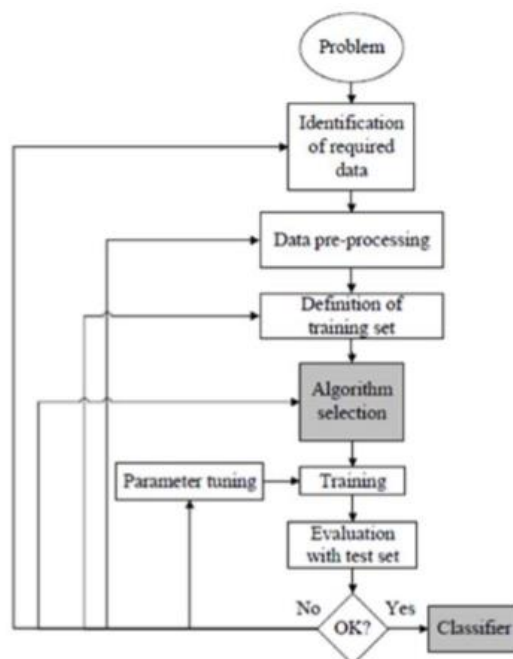


Figure 21 - Workflow of machine learning algorithm [21]

5.1 Dataset

In order to facilitate the understanding and visualization of the execution of the previous algorithms, this subsection reserves itself to demonstrate examples of the data and data sets that will be used, whether these are for training or the actual values that are expected from the forecast.

The analytical data used in this dissertation come from:

1. Patients with hereditary metabolic diseases, studied at the Genetic Biochemistry Unit of the Medical Genetics Center Jacinto Magalhães, integrated into the Hospital and University Center of Porto (CHUPorto).

2. Patients with colorectal cancer, studied at the Colorectal Surgery Unit of the General Surgery Service of CHUPorto. Analytical data were obtained by doctoral students Pedro Brandão, MD and Ivo Barros, MSc, in the development of the project entitled "MetLARC - Metabolic abnormalities on tumor response and resistance to neoadjuvant chemoradiotherapy in Locally Advanced Rectal Cancer", under the collaboration protocol to finance the multi-annual research grants plan for doctoral students, (scholarship with reference UI/BD/150743/2020), signed between the FCT and the R&D Unit – Multidisciplinary Biomedical Research Unit (UMIB) UI215.

A partial view of the data shown in Figure 22, with emphasis for the features or attributes that appear in bold in the image. The dataset presents 30 features or attributes, having two types of outputs:

- one binary, where 0 represents the absence of disease (healthy) and 1 represents the presence of disease (not healthy)
- other multi value, represented with integers, representing each one a different disease.

CYS2 SA16P	MET SA17P	CYSTA SA18P	ILE SA19P	LEU SA20P	TYR SA21P	PHE SA22P	ORN SA24P	LYS SA25P	1MHIS SA26P	HIS SA28P	3MHIS SA29P	ARG SA30P	SEXO	IDADE NA COLHEITA (Anos)	DIAG YES	DIAG ID
27	35	0.001	77	126	100	55	90	184	0.001	70	0.001	67	0		0	0
31	22	0.001	40	101	40	47	48	146	0.001	65	0.001	85	0		5	0
32	18	0.001	43	87	41	45	49	119	0.001	62	3	26	1		36	0
32	29	0.001	72	150	100	77	169	223	0.001	102	0.001	18	1		2	0
37	18	0.001	43	74	53	43	61	110	0.001	34	0.001	39	1		3	0
35	12	0.001	43	184	87	57	67	148	0.001	49	0.001	66	0		1	0
25	19	0.001	49	93	52	44	50	108	0.001	46	0.001	35	0		2	0
24	21	0.001	47	97	54	83	44	89	0.001	37	0.001	45	0		6	0
32	21	0.001	15	31	55	43	46	82	0.001	59	0.001	14	0		0	0
41	38	0.001	62	112	63	49	62	175	3	56	5	81	0		4	0
33	19	0.001	46	95	41	44	35	89	0.001	61	0.001	36	0		2	0
33	11	0.001	42	85	48	34	32	107	0.001	55	0.001	31	0		2	0
41	31	0.001	55	154	94	60	67	173	0.001	71	0.001	58	0		2	0
32	24	0.001	61	94	40	53	62	162	0.001	38	0.001	42	0		0	0
26	18	0.001	36	67	41	37	50	97	0.001	54	0.001	25	0		5	0
35	12	0.001	23	60	32	22	27	91	0.001	35	0.001	31	0		3	0
69	30	0.001	42	92	47	50	49	166	0.001	53	0.001	65	0		8	0
39	25	0.001	34	66	57	43	50	92	0.001	47	0.001	67	0		8	0
36	19	0.001	29	55	35	55	60	73	0.001	33	0.001	32	0		0	1
65	25	0.001	73	141	49	48	150	171	0.001	76	0.001	13	0		38	0
27	25	0.001	56	108	74	58	62	131	0.001	56	0.001	48	0		3	0
22	23	0.001	51	117	56	57	106	147	0.001	43	0.001	21	0		16	0
31	29	0.001	54	105	56	55	169	121	0.001	84	0.001	59	0		10	1

Figure 22 – Partial example of the dataset

The instances of the data have 30 features, of which 28 are plasma amino acid concentrations, the remaining being the patient's age and gender. The table 5 display all the features used in this study with the corresponding abbreviation and unit of measurement in which the values are presented.

Table 5 – Distribution of patients with disease

Feature Name	Feature Abreviation	Unit of measurement
Taurine	Tau	micromoles / litre
Aspartic Acid	Asp	micromoles / litre
Hydroxyproline	Hyp	micromoles / litre
Threonine	Thr	micromoles / litre
Serine	Ser	micromoles / litre
Asparagine	Asn	micromoles / litre
Glutamic acid	Glu	micromoles / litre
Glutamine	Gln	micromoles / litre
Alpha-Aminoadipic acid	AAA	micromoles / litre
Proline	Pro	micromoles / litre
Glycine	Gly	micromoles / litre
Alanine	Ala	micromoles / litre
Citrulline	Cit	micromoles / litre
Aminobutyric acid	ABU	micromoles / litre
Valine	Val	micromoles / litre
Cystine	Cys2	micromoles / litre
Methionine	Met	micromoles / litre
Cystathionine	Cysta	micromoles / litre
Isoleucine	Ile	micromoles / litre
Leucine	Leu	micromoles / litre
Tyrosine	Tyr	micromoles / litre
Phenylalanine	Phe	micromoles / litre
Ornithine	Orn	micromoles / litre
Lysine	Lys	micromoles / litre
1-Methylhistidine	1Mhis	micromoles / litre
Histidine	His	micromoles / litre
3-Methylhistidine	3Mhis	micromoles / litre
Arginine	Arg	micromoles / litre
Age (years) in the blood collection	Age	Positive Integer
Gender	Gender	2 possible: Male or Female

The table 6 displays the dataset's Classes and its distribution. The dataset is composed of 3842 examples, classified into 46 different classes, of which 45 corresponded to diseases (metabolic or colorectal cancer) and the remaining class grouped patients who had not been diagnosed with any disease. Despite the fact that these patients were not associated with any disease, it cannot in fact be concluded that they were healthy patients.

Table 6 – Dataset's Classes and its distribution

Diagnosis Id	Diagnosis	Absolute Frequency	Relative Frequency (in %)
0	Without diagnosed disease	2222	57,83
1	3-Hydroxy-3-methylglutaric aciduria	50	1,3
2	Argininosuccinic aciduria	23	0,6
3	Glutaric Aciduria Type I	48	1,25
4	Glutaric Aciduria Type II	11	0,29
5	Methylmalonic aciduria	77	2
6	Methylmalonic aciduria due to impaired cobalamin metabolism	92	2,39
7	Propionic aciduria	6	0,16
8	Alkaptonuria	3	0,08
9	Argininemia	63	1,64
10	Cystinuria – heterozygosity type B	2	0,05
11	Mitochondrial cytopathy	7	0,18
12	Citrullinemia Type I	54	1,41
13	Citrullinemia Type II	13	0,34
14	CPS deficiency	8	0,21
15	Methionine Adenosyltransferase Deficiency	5	0,13
16	OTC Deficiency	301	7,83
17	GAMT Deficiency	3	0,08
18	MCAD Deficiency	1	0,03
19	Phenylketonuria	485	12,62
21	Galactosemia	1	0,03
22	Glycogenosis Type I	2	0,05
23	Hyperargininemia	1	0,03
25	Hyperglycemia without Ketosis	9	0,23
26	Hypermethioninemia	4	0,1
27	Hyperornithinemia with Gitate Atrophy	29	0,75
28	Hypophosphatasia	1	0,03
29		83	2,16

Diagnosis Id	Diagnosis	Absolute Frequency	Relative Frequency (in %)
	Classical Homocystinuria		
30	Iminodipeptiduria	2	0,05
31	Fructose Intolerance	1	0,03
32	Leucinosi	120	3,12
35	Tyrosinemia Type I	11	0,29
36	Tyrosinemia Type II	2	0,05
37	3-Methylcrotonylglycinuria	2	0,05
40	Chanarin-Dorfman Syndrome	1	0,03
42	Beta-ketothiolase deficiency	7	0,18
44	CLN3 disease	1	0,03
48	Glycerol Kinase Deficiency	1	0,03
49	GM2-Gangliosidosis	1	0,03
50	Glycogenosis Type V	3	0,08
51	HHH Syndrome	14	0,36
52	Hyperprolinemia	3	0,08
54	Niemann-Pick Type B	1	0,03
55	Niemann-Pick Type C	2	0,05
57	Tyrosinemia Type 3	9	0,23
99	Colorectal cancer	57	1,48

Based on these data, it was decided to carry out three different case studies, involving the use of three different subsets of this global dataset, to determine the best machine learning model for:

1) Binary classification of disease detection

Using the entire dataset to build a model that can determine the presence of disease. In which all patients without diagnosed disease (where Diagnosis Id is 0) constitute a class and the remaining patients constitute the other class indicating the presence of disease. The absolute and relative distributions of each class is shown in figures 23 and 24.

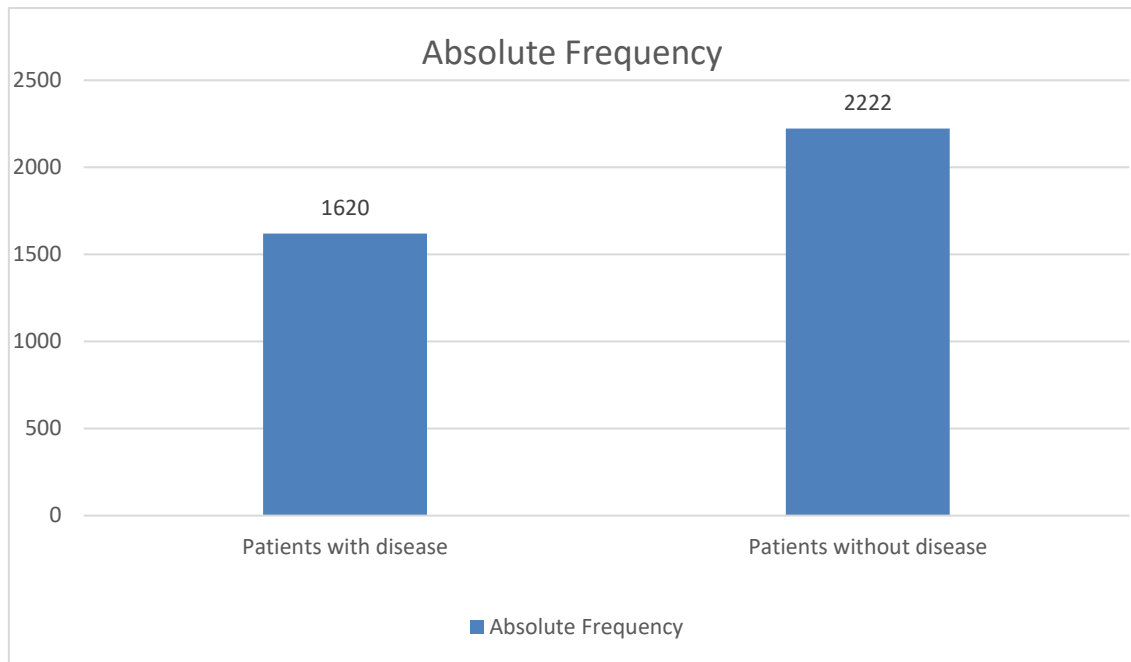


Figure 23 - Absolute Frequency of the 2 classes in Binary classification of disease detection

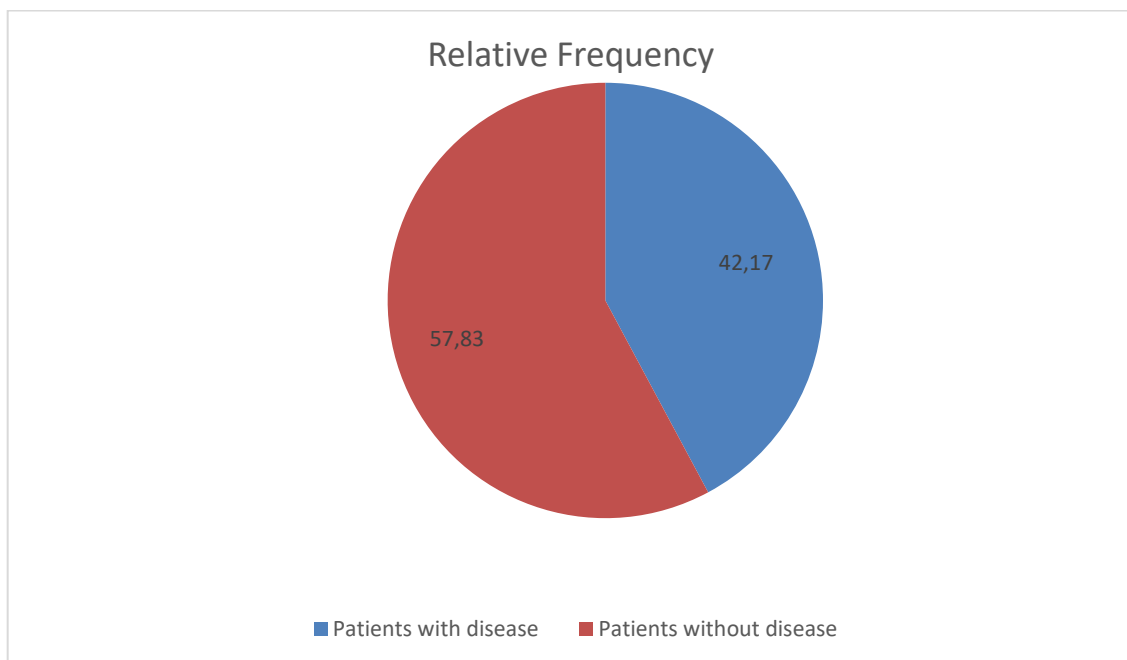


Figure 24 - Relative Frequency of the 2 classes in Binary classification of disease detection

2) Multi class Classification of diseases

Diagnosis of which disease (or absence of disease) affects the patient.

Bearing in mind that many of the classes presented have a very small number of instances, it is impractical to use them to generate a machine learning model capable of identifying them. It was decided to use only disease classes whose absolute value was greater than 55 instances, which in percentage corresponds to approximately 1.5%. The selected classes distribution is shown in table 7.

Table 7 – Class distribution in Multi class Classification of diseases

Diagnosis Id	Diagnosis	Absolute Frequency	Relative Frequency (in percentage) in relation to the entire dataset
0	Without disease	2222	57,83
5	Methylmalonic aciduria	77	2
9	Argininemia	63	1,64
16	OTC Deficiency	301	7,83
19	Phenylketonuria	485	12,62
29	Classical Homocystinuria	83	2,16
32	Leucinosi	120	3,12
99	Colorectal cancer	57	1,48

It should be noted that the relative frequency presented in table 7 is relative to the total available data. In figure 25 is shown the relative frequency of all classes in relation to the subset of data that will be used in the multi classification.

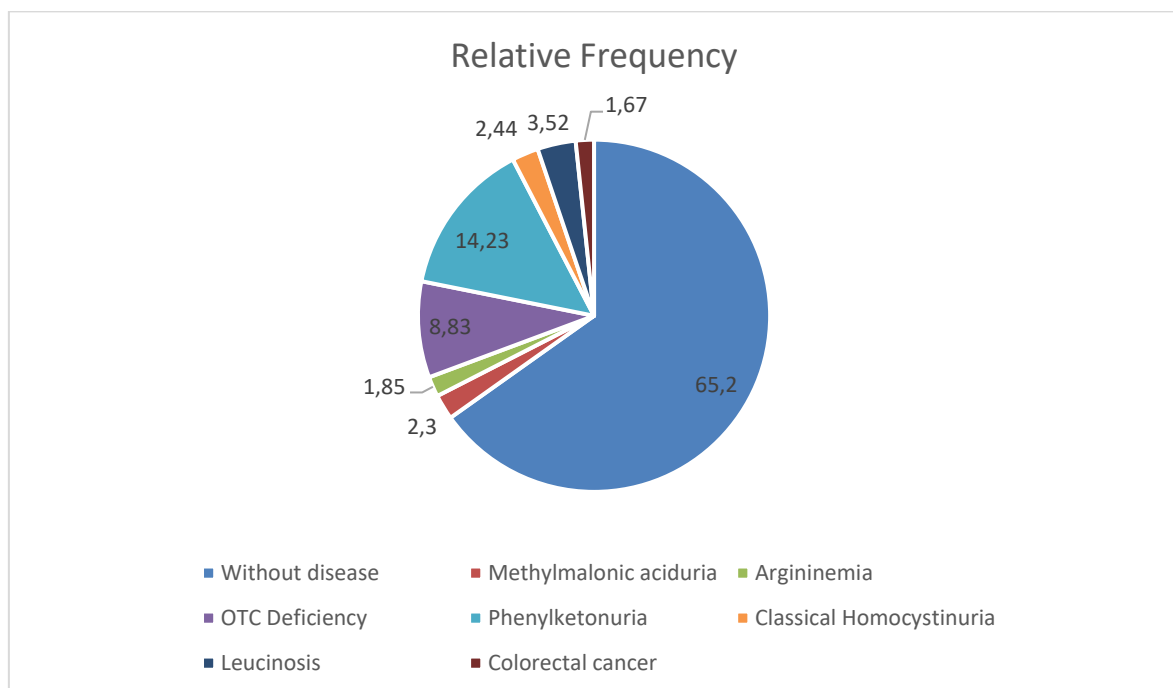


Figure 25 – Relative frequency for Multi class Classification of diseases classes

3) Binary classification for colorectal cancer detection

In this study the class of patients diagnosed with colorectal cancer is used and for the control class (without CRC), instead of using all the others, it was decided to use only the class of patients without diagnosis, as these have the profile closer to the population in which the model is intended to be used if it has satisfactory results.

As the class of undiagnosed patients has a much higher absolute frequency, only the subset that belongs to the most recent samples will be used, as the collection of these data was carried out with an extra precaution. Furthermore, within this subset only adult patients (aged 18 years and over) as these will be able to make a fairer comparison, as patients diagnosed with colorectal cancer are all adults as well. The absolute and relative distributions of each class is shown in figures 26 and 27, respectively.

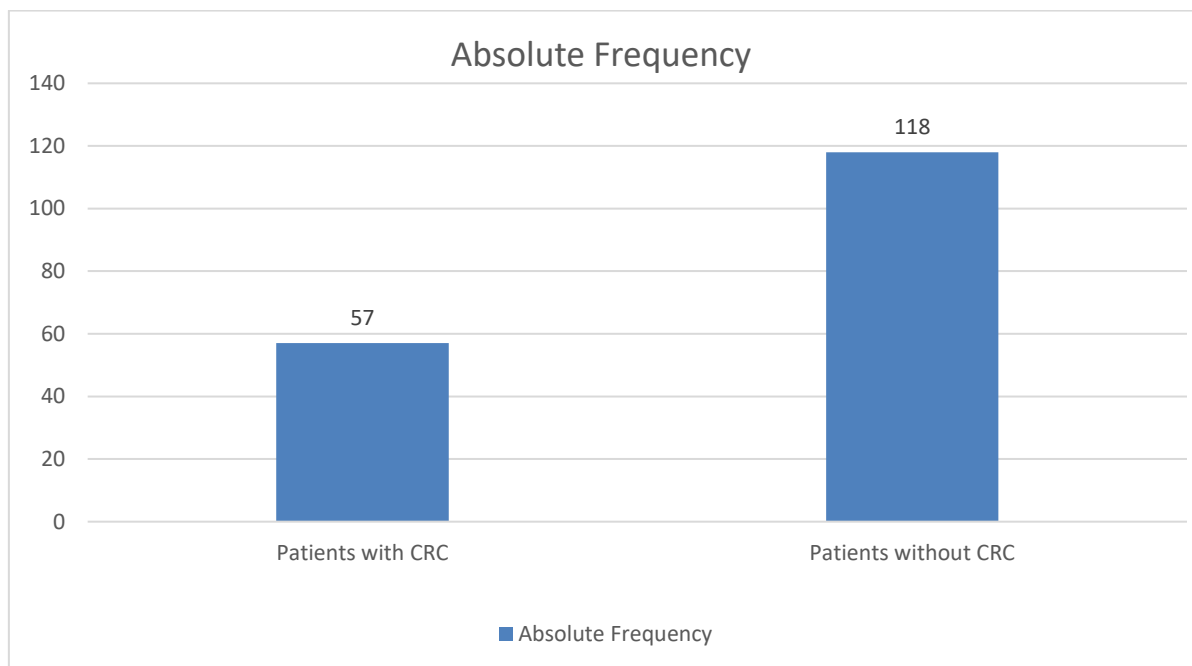


Figure 26 - Absolute Frequency of the 2 classes in Binary classification colorectal cancer detection

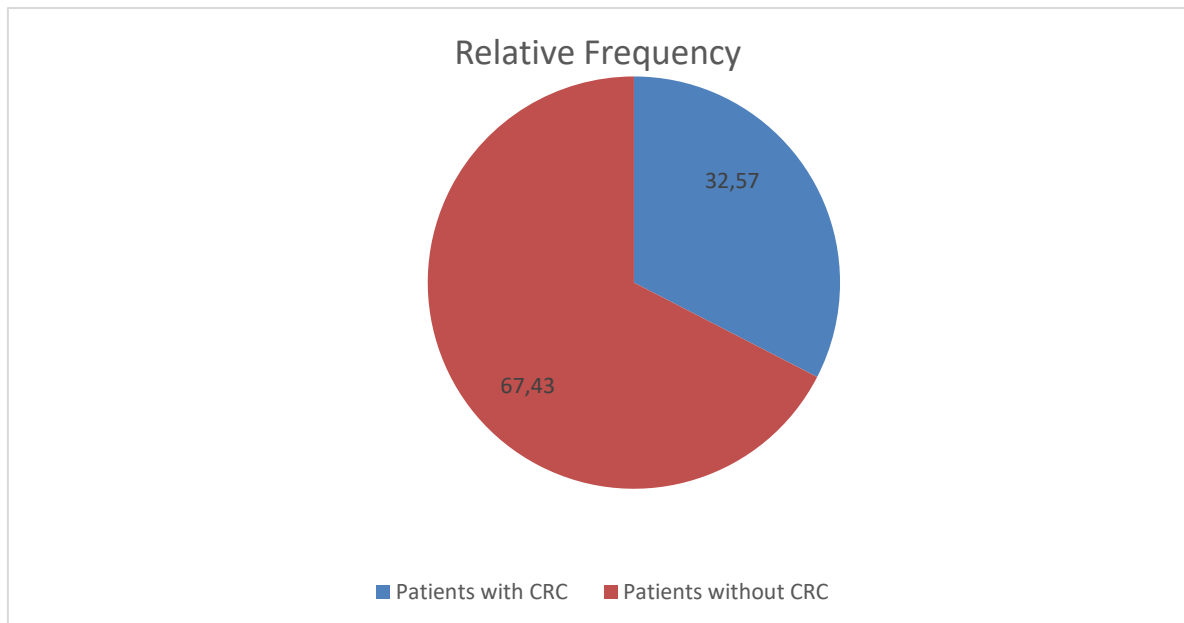


Figure 27 - Relative Frequency of the 2 classes in Binary classification colorectal cancer detection

5.2 Preprocessing

Currently, it is difficult for classification algorithms to extract important information from such large volumes of information. In the preprocessing of the data, unreliable information must be removed, leaving only the relevant information [37].

When improving the data quality, typically the quality of the resulting analysis is also improved. In addition, in order to make the raw data more suitable for further analysis, preprocessing steps should be applied that focus on the modification of the data. A number of different techniques and strategies exist, relevant to data preprocessing that focus on modifying the data for better fitting in a specific ML method [14].

Among these techniques some of the most important approaches include:

- data cleaning
- data normalization
- dimensionality reduction (feature selection)

5.2.1 Data Cleaning

Data cleaning refers to the process that increases the quality of the input data, removing noisy data, completing incomplete data and correcting inconsistencies in the data. If this step is not applied, it becomes complicated to consider that the data is reliable, which consequently leads to a distrust in the results of any algorithm learning process [38].

In this step the data values were analysed in search for some error that could impact the learning process of our algorithm.

Some of the dataset instances presented incomplete data, where some features did not have value. In other cases, the data had negative values, which in the context of the problem dealing with amino acid concentrations could not be correct. For both scenarios, these cases affected less than 1% of the total data and mainly in the patients without disease class, so it was decided that the best was to remove the instances.

5.2.2 Data Normalization

As mentioned, of the 30 features available for training the model, 28 are concentrations of plasma amino acids, the rest being the patient's age and gender. For amino acid concentrations, the same measurement unit was used. For age, it was used the integer value by truncation (for example 24 years and 7 months would count as 24 years). In the case of gender, we are only considering 2 values as possible (male and female), thus they were transformed into numerical values 0 and 1 respectively.

5.2.3 Feature Selection

Higher dimension data leads to the prevalence of noisy, irrelevant and redundant data. Which intern causes overfitting of the model and increases the error rate of the learning algorithm. To handle these problems “Dimensionality Reduction” techniques are applied, and it is the part of the preprocessing stage. Feature selection is a subset of dimensionality reduction that is used to clean up the noisy, redundant and irrelevant data. As shown in figure 28, a subset of features are selected from the original set of features based on features redundancy and relevance. [41]

Three benefits of performing feature selection before modelling data are:

- Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.
- Improves Accuracy: Less misleading data means modelling accuracy improves.
- Reduces Training Time: Less data means that algorithms train faster.

Moreover, this technique enhances the comprehensibility of data, facilitates better visualization of data and improves the performance of prediction. [41]

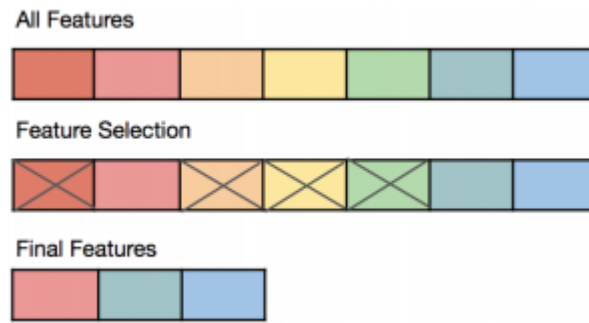


Figure 28 – Feature Selection

From the existing Feature Selection techniques, to analyse the impact of different strategies, techniques belonging to two distinct categories, shown in figure 29, will be addressed:

Filter Method – It is one of the oldest methods of feature selection. In variable selection using filter approach, filtering of features is done before the implementation of any learning algorithm. It ranks features based on a certain evaluation criteria. These methods give fast and efficient results on execution. [42]

Wrapper Method – Wrapper methods does selection of features by giving due consideration to the learning algorithm to be used. The major advantage over filter methods is that it finds the most “useful” features and does optimal selection of features for the learning algorithm. Moreover, this method is very complex and more prone to over-fitting on small training datasets. [42]

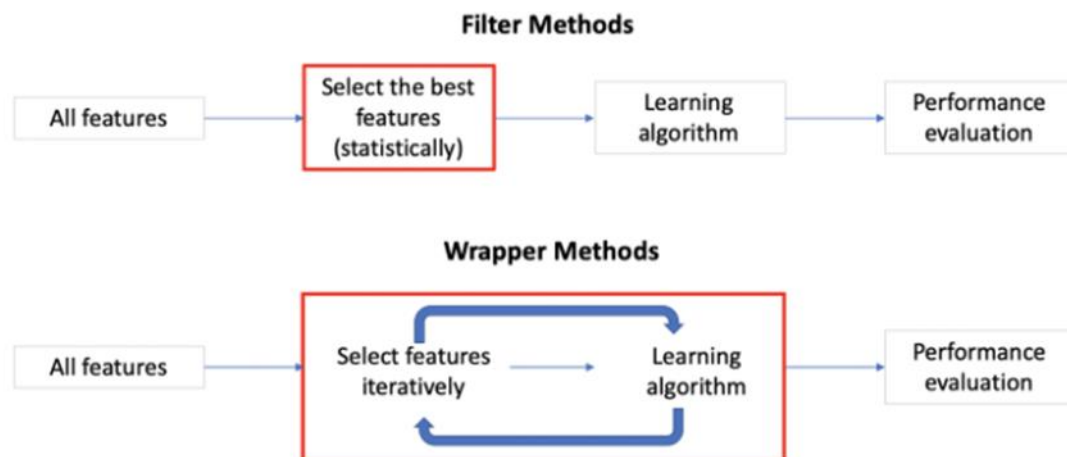


Figure 29 - Feature Selection: Filter and Wrapper methods [43]

In Wrapper Method category will be studied the technique Recursive Feature Elimination (RFE).

RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains. This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking

features by importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remains [39].






Within Filters Methods, two techniques will be studied:

- Pearson's correlation (PC): Correlation thresholds remove features that are highly correlated with others (i.e. its values change very similarly to another's).
- Univariate Feature Selection (with SelectKBest): Univariate Feature Selection is a feature selection method based on the univariate statistical test (e.g: chi2).

The premise with SelectKBest is combining the univariate statistical test with selecting the K-number of features based on the statistical result between the feature and label.

In order not to be a very extensive study, and as the main objective of this dissertation is the detection of colorectal cancer, the distribution of techniques among the studies was done as represented in table 8.

Table 8 – Feature Selection techniques used

	Pearson's correlation	Univariate Feature Selection	Recursive Feature Elimination
Binary classification for disease detection			
Multi class Classification for disease detection			
Binary classification for colorectal cancer detection			

Pearson's correlation (PC) Coefficient

Due to large number of features, it is helpful to label in the PC coefficient diagram each feature as a number. The table 9 will help us relate each number to the corresponding feature.

Table 9 – Feature id and corresponding name

Feature Id	Feature Name	Feature Id	Feature Name
0	Taurine	15	Cystine
1	Aspartic Acid	16	Methionine
2	Hydroxyproline	17	Cystathionine
3	Threonine	18	Isoleucine
4	Serine	19	Leucine
5	Asparagine	20	Tyrosine
6	Glutamic acid	21	Phenylalanine
7	Glutamine	22	Ornithine
8	Alpha-Aminoadipic acid	23	Lysine
9	Proline	24	1-Methylhistidine
10	Glycine	25	Histidine
11	Alanine	26	3-Methylhistidine
12	Citrulline	27	Arginine
13	Aminobutyric acid	28	Age (years) in the blood collection
14	Valine	29	Gender

Pearson's correlation is used for detecting the linear relationship between two variables. Generally, the PC value lies in between $[-1, 1]$ if the value is -1 then the variables are negatively correlated otherwise if the value is 1 then the variables are positively correlated. In case that the value is 0 , then there is no correlation between the variables. [41]

It will be considered a strong correlation if the coefficient value lies between ± 0.75 and ± 1 , being possible to remove one of the features without impacting the performance of the model. This assumption will be tested in the Results chapter.

For the Binary classification for disease detection there are 2 features whose correlation is higher than the set threshold. As it is seen in figure 30, Valine (feature id 14) and Isoleucine (feature id 18) have a correlation of 0.82 . So it is expected that the Isoleucine removal will not impact the model's prediction capabilities.

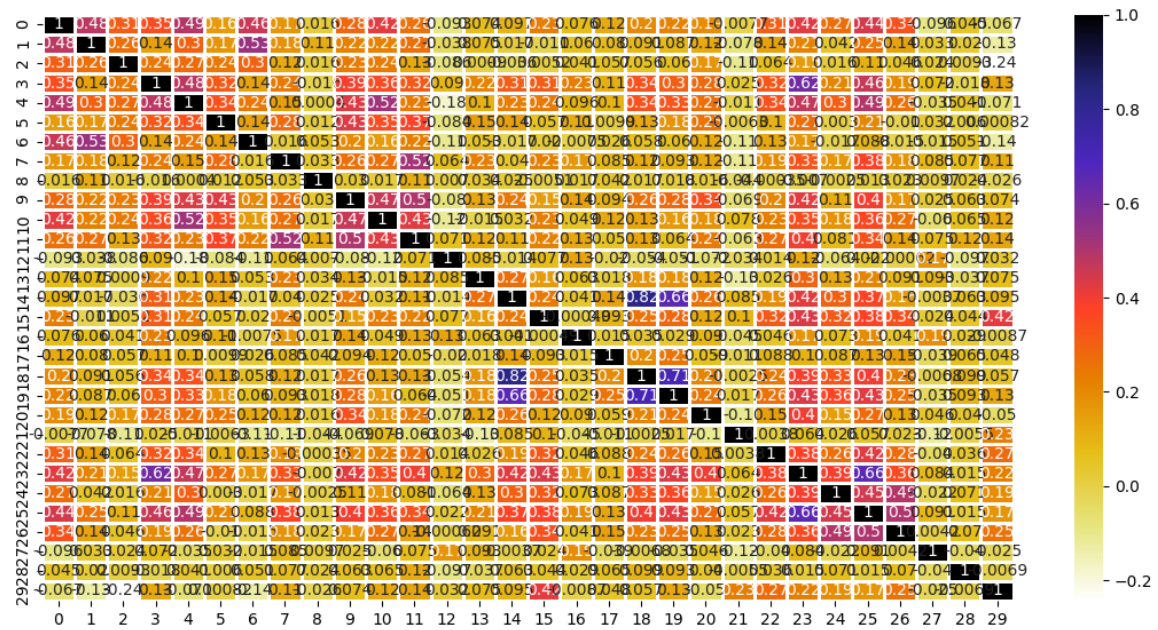


Figure 30 - Pearson's Correlation Coefficient in Binary classification for disease detection

In the case of multi classification, this is the diagnosis of the specific disease affecting the patient, for the same threshold the result is similar to the binary classification for disease detection. As it is seen in figure 30, Valine (feature id 14) and Isoleucine (feature id 18) have a correlation of 0.81. So, as said before, it is expected that the Isoleucine removal will not impact the model's prediction capabilities for multi class classification.

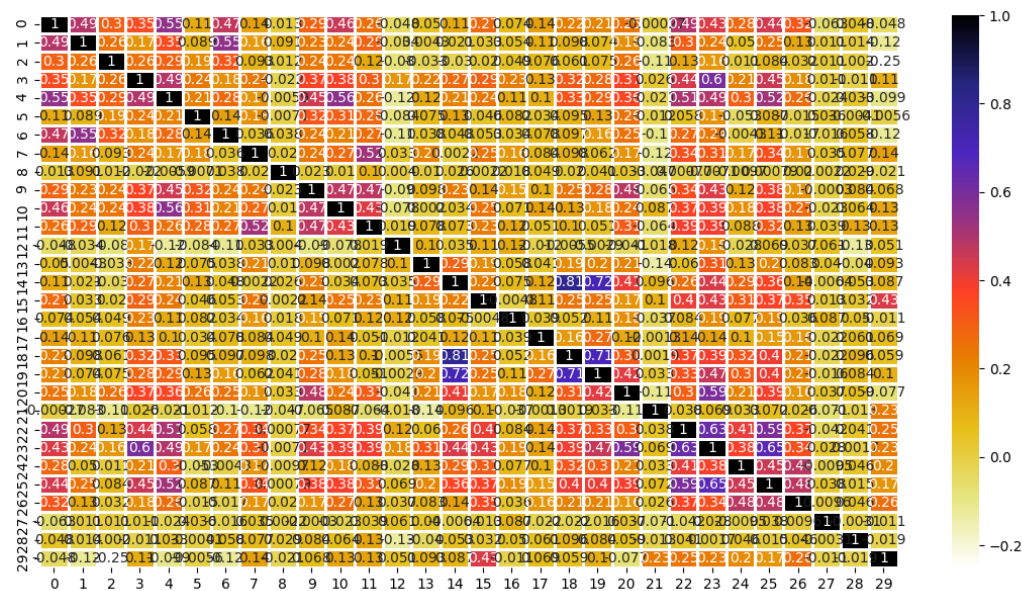


Figure 31 - Pearson's Correlation Coefficient in Multi class classification for disease detection

In the case of Binary classification for colorectal cancer detection, as the Figure 32 shows none of the features have a correlation equal or superior to the threshold stipulated, so

no feature should be removed. Existing two pairs of feature with the highest correlation (0.71). The Aspartic Acid (feature id 1) and Glutamic acid (feature id 6) pair and the other pair, is Valine (feature id 14) and Isoleucine (feature id 18).

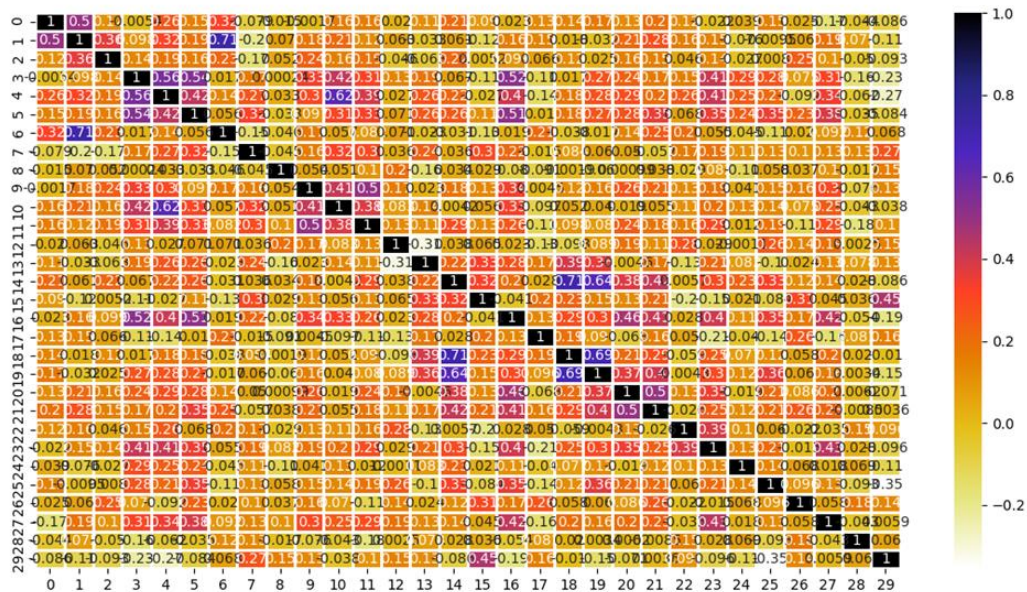


Figure 32 - Pearson's Correlation Coefficient in Binary classification for colorectal cancer detection

Univariate Feature Selection

Univariate feature selection works by selecting the best features based on univariate statistical tests. Each feature is compared to the target variable, to see whether there is any statistically significant relationship between them. It is also called analysis of variance. When analysing the relationship between one feature and the target variable, the other features are ignored, that is why it is called 'univariate'. Each feature has its test score. [40]

Using Univariate feature selection based on chi-squared stats between each feature and the outcome (classification label) for Binary classification for colorectal cancer detection, it is possible to attribute points and rank the features based on its score.

There are great differences in the score between the features with best score and worst score. It was decided to validate how a model generated based in one third of the features would behave, so 5 runs will be done and the Top 10 features with the best average scores of the runs are presented in the table 10.

Table 10 – Top 10 Features with best score using Univariate Feature Selection

Feature	Average Score
Glutamine	992,54
Age	821,46
Cystinine	257,14
Glutamic acid	220,07
Aspartic acid	134,62
Proline	103,64
Lysine	97,31
Hydroxyproline	95,92
Alanine	61,01
Isoleucine	37,25

The table shows that there are two features that stand out clearly: Glutamine and Age.

Recursive Feature Elimination (RFE)

RFE is a method to select features by recursively considering smaller and smaller sets of features. In this way, the final result is a ranking of features grouped according to their importance.

In the table 11 is presented the resulting sets of ranking 1 (with greatest importance) for the 5 runs done.

Table 11 – Features with greatest importance using Recursive Feature Selection for each run

Feature – Run 1	Feature – Run 2	Feature – Run 3	Feature – Run 4	Feature – Run 5
Taurine	Taurine	Asparagine	Taurine	Taurine
Lysine	Histidine	Histidine	Histidine	Histidine
Leucine	1-Methylhistidine	1-Methylhistidine	Lysine	Lysine
Methionine	Lysine	Lysine	Phenylalanine	Phenylalanine
Cystine	Phenylalanine	Phenylalanine	Leucine	Tyrosine
Citrulline	Tyrosine	Glutamic acid	Methionine	Leucine
Alanine	Leucine	Leucine	Cystine	Methionine
Glycine	Isoleucine	Isoleucine	Valine	Cystine
Valine	Methionine	Methionine	Citrulline	Citrulline
Hydroxyproline	Cystine	Cystine	Age	Glycine
Aspartic acid	Citrulline	Glutamine	Glutamic acid	Valine
Threonine	Valine	Valine	Glutamine	Serine
Serine	Glycine	Age	Threonine	Aspartic Acid
Age	Aspartic Acid	Aspartic Acid	Aspartic Acid	Glutamine
Glutamic acid	Hydroxyproline	Hydroxyproline		Glutamic acid
Glutamine	Threonine	Threonine		Age

Feature – Run 1	Feature – Run 2	Feature – Run 3	Feature – Run 4	Feature – Run 5
Alpha-Aminoadipic acid	Serine	Serine		Hydroxyproline
Histidine	Alanine			Threonine
Aminobutyric acid	Glutamic acid			Asparagine
Isoleucine	Age			
Proline	Glutamine			
3-Methylhistidine				
Asparagine				
Phenylalanine				
1-Methylhistidine				
Tyrosine				
Cystathionine				
Arginine				

As it can be seen in the table 11, there is some variability between the 5 runs, both in the number of tier-1 of importance features and in the features themselves. Nevertheless there are 12 features that appear on all runs: Lysine, Leucine, Methionine, Cystine, Valine, Aspartic acid, Threonine, Age, Glutamic acid ,Glutamine, Histidine and Phenylalanine. These 12 features will be the considered to study the generated models performance using this feature selection technique.

5.3 Imbalanced Data

Canonical machine learning algorithms assume that the number of instances in considered classes is roughly similar. However, in disease detection situations the distribution of examples is skewed since representatives of some of classes appear much more frequently. In our three studies, the class distribution is shown in *Dataset* chapter.

This poses a difficulty for learning algorithms, as they will be biased towards the majority group. At the same time usually the minority class is the one more important, as despite its rareness it may carry important and useful knowledge. Therefore, when facing such disproportions one must design an intelligent system that is able to overcome such a bias. This domain is known as learning from imbalanced data. [44]






One of the most interesting directions in binary imbalanced classification is the notion that imbalance ratio is not the sole source of learning difficulties. Even if the disproportion is high, but both classes are well represented and come from non-overlapping distributions we may obtain good classification rates using canonical classifiers. [44]

We may distinguish three main approaches to learning from imbalanced data:

- Data-level methods that modify the collection of examples to balance distributions and/or remove difficult samples. With respect to balancing distributions we may distinguish approaches that generate new objects for minority groups (oversampling) and that remove examples from majority groups (undersampling).
- Algorithm-level methods that directly modify existing learning algorithms to alleviate the bias towards majority objects and adapt them to mining data with skewed distributions.
- Hybrid methods that combine the advantages of two previous groups.

The table 12 represents the different data balancing techniques applied to each of the datasets. As the Binary Classification for disease detection dataset is represented by a high number of instances for each class and there is not a great discrepancy between them, it will not be employed any technique.

Table 12 – Data processing techniques applied to each of the datasets

	Data-level (Oversampling)	Data-level (Undersampling)	Algorithm-level
Binary classification for disease detection			
Multi class Classification for disease detection			
Binary classification for colorectal cancer detection			

For the oversampling technique, it will be used the Synthetic Minority Oversampling Technique (SMOTE) that randomly creates artificial samples along the line joining a minority sample and one of its nearest neighbors. [45]

For the undersampling technique, it will be used the Random undersampling that is a sampling technique to improve imbalance levels of the classes to the desired target by randomly removing instances from the majority class(es). [46]

This approach may be more suitable for those datasets where there is a class imbalance although a sufficient number of examples in the minority class, such a useful model

can be fit. A limitation of undersampling is that examples from the majority class are deleted that may be useful, important, or perhaps critical to fitting a robust decision boundary.

At the algorithm level, it is shown in the table 13 what modifications were applied to the algorithms used.

Table 13 – Alterations to reduce the imbalance bias at algorithm level

Algorithm	Balanced Version
Gaussian Naive Bayes	Not used
Logistic Regression	Passed class weight as an argument
Support Vector Machine	Passed class weight as an argument
Decision Tree	Passed class weight as an argument
Random Forest	Used Balanced Random Forest
K-Nearest Neighbor	Not used
Artificial Neural Network	Not used

5.4 Performance Evaluation

As the final stage of ML systems, the model predictive performance evaluation is a necessary stage, no less relevant than the other referred previous stages. Through the experimental data, it provides feedback on how the model correctly predicts or classifies the target label. The ability to assess the performance is vital for guiding the Model Selection phase, which is the process of selecting the algorithm and the values of its parameters. It primarily consists in a “trial and error” strategy, training different algorithms with different parameters. Each created model must be evaluated and compared with others. In the end, the highest accurate model is chosen based on estimative values, such as the generalisation error which provides a sense of how the model performs with unseen data.

5.4.1 Split Dataset

The dataset that is available to build and evaluate a predictive model is referred to as the learning set, assumed to be a sample from a population of interest.

When estimating the performance, it is important to avoid overfitting, which can occur using the same data instances used during the model training, or else the evaluation becomes “biased”. Meaning that, an overfitted model is only fit to its training data but incapable of correctly predicting future observations. The recommended approach for model evaluation is to separate the data randomly into different sets:

- Training Set: used only for the model generation (usually is the largest set).
- Test Set: used to evaluate the performance of the generated model.

Holdout Method

Among the various data resampling strategies, one of the simplest ones is the single hold-out method. Hold-out validation was proposed to remove the problem of over-fitting that was there in re-substitution validation. As shown in figure 33, the data is divided into two non-overlapping parts and these two parts are used for training and testing respectively. The part which is used for testing is the hold-out part. It is so named because we hold-out that part for testing and learn the model using the remaining part of the data. [47]

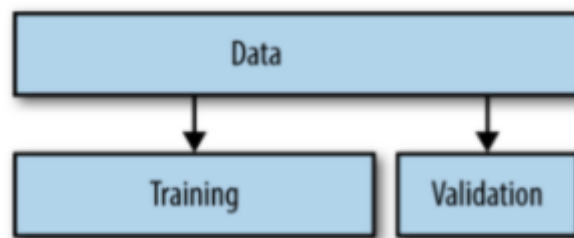


Figure 33 – Holdout method [48]

People usually do not understand this very clearly and assume that hold-out validation is splitting the data into two equal parts. While it's true that it can be called hold-out validation, however, it is a very specific case of hold-out validation where the amount of data being held-out for testing is 50%. In this case, it will be tested different proportions for training data and validation data to identify which presents the best results.

One of the important points about using hold-out validation is that the time taken for learning the model is relatively lesser than the time taken for learning the model using k-fold cross-validation. [47]

K-fold Cross-validation

Cross-validation is one of the most widely used data resampling methods to estimate the true prediction error of models and to tune model parameters. In k-fold cross-validation, the available learning set is partitioned into k disjoint subsets of approximately equal size. Here, "fold" refers to the number of resulting subsets. This partitioning is performed by randomly sampling cases from the learning set without replacement. The model is trained using k – 1 subsets, which, together, represent the training set. Then, the model is applied to the remaining subset, which is denoted as the validation set, and the performance is measured. This procedure is repeated until each of the k subsets has served as validation set. The average of the k performance measurements on the k validation sets is the cross-validated performance. Figure 34 illustrates this process for k = 10, i.e., 10-fold cross-validation. [48]

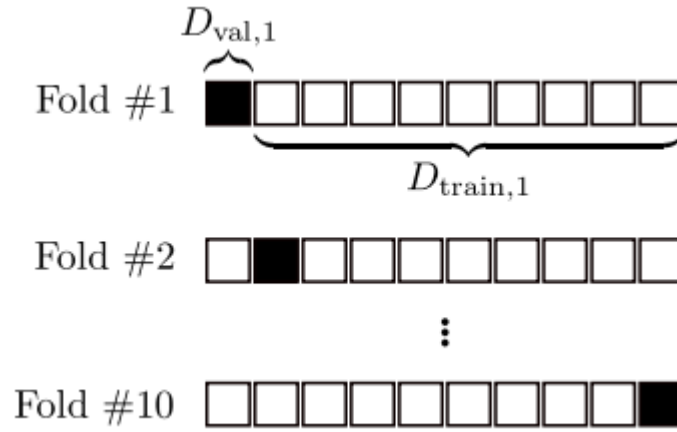


Figure 34 – 10-fold cross-validation [48]

Decision

In scenarios where the data available is scarce, it is preferable to utilize techniques like cross-validation. It performs successive rounds of validation, in a way that both segments cross over validating against each other [40]. Performing cross-validation also has the advantage of providing an indication of how well the model will generalise to an unseen dataset reducing the problem of overfitting.

Having this in mind, the resampling strategy that will be used in all the models generation is the 5-fold cross validation.

5.4.2 Performance Metrics

For the comparison and evaluation of ML models, many performance metrics are available however one must ponder which metric is best for his problem. As such, here are presented some of the most basic and commonly utilised metrics:

- True Positive (TP): number of instances correctly accepted, or predicted/classified as positive.
- True Negative (TN): number of instances correctly rejected or predicted/classified as negative.
- False Positive (FP): number of instances incorrectly accepted or predicted/classified as positive.
- False Negative (FN): number of instances incorrectly rejected or predicted/classified as negative.
- Accuracy: proportion of true results (both positive and negative)

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

- Sensitivity/Recall: true positive rate

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

- Specificity: true negative rate

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

- F1 Score: harmonic mean of precision and recall.

This evaluation measure calculates the effectiveness of a classifier algorithm, combining the accuracy and sensitivity measures.

$$F1\ Score = \frac{2 \times TP}{2 \times TP + FN + FP}$$

The diagnostic ability of classifiers has usually been determined by the confusion matrix and the receiver operating characteristic (ROC) curve. In the machine learning research domain, the confusion matrix is also known as error or contingency matrix. The basic framework of the confusion matrix has been provided in Figure 35. In this framework, true positives (TP) are the positive cases where the classifier correctly identified them. Similarly, true negatives (TN) are the negative cases where the classifier correctly identified them. False positives (FP) are the negative cases where the classifier incorrectly identified them as positive and the false negatives (FN) are the positive cases where the classifier incorrectly identified them as negative. The following measures, which are based on the confusion matrix, are commonly used to analyse the performance of classifiers, including those that are based on supervised machine learning algorithms [23].

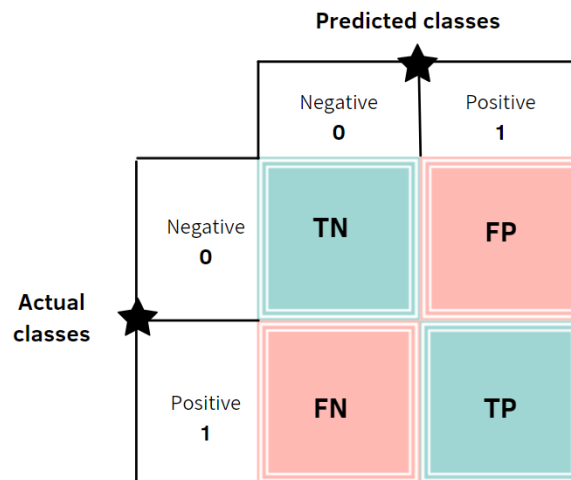


Figure 35 – Confusion matrix [23]

5.5 Results

The experiments were performed on Intel core i7-6500 CPU with 16GB RAM, Windows 10 Education 64bits

To increase the reliability, all the results presented are the mean values of 5 runs for each one.

As stated before, to determine the performance of the model will be used the 5-fold cross validation technique with the scoring function being the F1-Score.

5.5.1 Data Balancing Evaluation

The first aspect studied is the data balancing due to the unbalance of the data, and how impactful the presence of classes with much more data can be in the generated models prediction ability. Based on the presented strategies defined before it will be analysed which strategy offers the best approach to deal with imbalanced data.

For the reasons declared in the *Imbalanced* chapter, this data balancing study is done only for the Binary classification for colorectal cancer detection and Multi class classification for disease detection.

Binary classification for colorectal cancer detection

In table 14 it is presented the results for the various algorithms for the baseline (control) for this experiment, the model generation where no strategy is applied to the dataset.

Table 14 – Binary classification for colorectal cancer detection (control)

Algorithm	Accuracy (standard deviation)	Execution Time
Gaussian Naive Bayes	83.763% (+/-0.04)	0.0103 seconds
Logistic Regression	88.0% (+/-0.03)	4.6874 seconds
Support Vector Machine	86.286% (+/-0.03)	0.0171 seconds
Decision Tree	85.857% (+/-0.03)	0.0121 seconds
Random Forest	93.8% (+/-0.03)	0.8970 seconds
K-Nearest Neighbor	80.0% (+/-0.09)	0.0150 seconds
Artificial Neural Network (Multi-layer Perceptron classifier)	86.857% (+/-0.03)	5.2413 seconds

In table 15 are presented the results for the various algorithms using the SMOTE strategy for oversampling.

Table 15 – Binary classification for colorectal cancer detection with oversampling

Algorithm	Accuracy (standard deviation)	Execution Time
Gaussian Naive Bayes	88.486% (+/-0.03)	0.0105 seconds
Logistic Regression	93.67% (+/-0.03)	3.8483 seconds
Support Vector Machine	90.275% (+/-0.04)	0.0435 seconds
Decision Tree	88.582% (+/-0.04)	0.0194 seconds
Random Forest	95.904% (+/-0.03)	0.7885 seconds
K-Nearest Neighbor	80.496% (+/-0.05)	0.0157 seconds
Artificial Neural Network (Multi-layer Perceptron classifier)	88.98% (+/-0.02)	6.0262 seconds

In table 16 are presented the results for the various algorithms using a Random Undersampling, that randomly removes examples from the most represented classes.

Table 16 - Binary classification for colorectal cancer detection with undersampling

Algorithm	Accuracy (standard deviation)	Execution Time
Gaussian Naive Bayes	83.763% (+/-0.04)	0.0109 seconds
Logistic Regression	88.0% (+/-0.03)	3.8900 seconds
Support Vector Machine	86.286% (+/-0.03)	0.0098 seconds
Decision Tree	86.571% (+/-0.03)	0.0087 seconds
Random Forest	93.714% (+/-0.03)	0.7107 seconds
K-Nearest Neighbor	80.0% (+/-0.09)	0.0106 seconds
Artificial Neural Network (Multi-layer Perceptron classifier)	86.857% (+/-0.03)	3.9012 seconds

In table 17 are presented the results for a different strategy where the data itself is not changed, but the algorithms are changed instead specially to deal with imbalanced data.

Table 17 - Binary classification for colorectal cancer detection with balanced algorithms

Algorithm	Accuracy (standard deviation)	Execution Time
Gaussian Naive Bayes	Not used	Not used
Logistic Regression	89.143% (+/-0.04)	4.7709 seconds
Support Vector Machine	85.714% (+/-0.03)	0.0271 seconds
Decision Tree	88.571% (+/-0.03)	0.0144 seconds
Balanced Random Forest	92.861% (+/-0.02)	1.3061 seconds
K-Nearest Neighbor	Not used	Not used
Artificial Neural Network (Multi-layer Perceptron classifier)	Not used	Not used

In terms of better performance there are no significative differences between the control and the undersampled and even the use of balanced algorithms. The strategy that stands out, being noticeable that offers the **best performance is the oversampling**, and even in terms of execution time the difference to the other strategies is negligenciabile.

Overall, the algorithm that presents the best results is the one generated with Random Forest algorithm using oversampling which presents a high value of accuracy of 95.904% (+/-0.03).

Multi class classification for disease detection

For the multi class classification study, in table 18 it is presented the results for the various algorithms with no data balancing strategy being applied, used as a control for this experiment.

Table 18 - Multi class Classification for disease detection (control)

Algorithm	Accuracy (standard deviation)	Execution Time
Gaussian Naive Bayes	70.967% (+/-0.03)	0.0297 seconds
Logistic Regression	90.589% (+/-0.01)	52.5869 seconds
Support Vector Machine	91.275% (+/-0.01)	2.4445 seconds
Decision Tree	86.556% (+/-0.01)	0.4502 seconds
Random Forest	92.734% (+/-0.01)	5.1296 seconds
K-Nearest Neighbor	89.302% (+/-0.01)	0.4091 seconds
Artificial Neural Network (Multi-layer Perceptron classifier)	91.275% (+/-0.01)	43.4466 seconds

In table 19 are presented the results for the various algorithms in multi class classification for disease detection using a Random Undersampling.

Table 19 - Multi class Classification for disease detection with undersampling

Algorithm	Accuracy (standard deviation)	Execution Time
Gaussian Naive Bayes	73.355% (+/-0.04)	0.0149 seconds
Logistic Regression	79.157% (+/-0.05)	10.9359 seconds
Support Vector Machine	84.639% (+/-0.05)	0.0666 seconds
Decision Tree	74.242% (+/-0.04)	0.0354 seconds
Random Forest	86.839% (+/-0.01)	1.0735 seconds
K-Nearest Neighbor	76.536% (+/-0.01)	0.0236 seconds
Artificial Neural Network (Multi-layer Perceptron classifier)	84.869% (+/-0.02)	12.1863 seconds

In table 20 are presented the results for the various algorithms in multi class classification for disease detection using the SMOTE strategy for oversampling.

Table 20 - Multi class Classification for disease detection with oversampling

Algorithm	Accuracy (standard deviation)	Execution Time
Gaussian Naive Bayes	79.903% (+/-0.02)	0.1394 seconds
Logistic Regression	92.831% (+/-0.01)	301.8681 seconds
Support Vector Machine	96.698% (+/-0.00)	16.8793 seconds
Decision Tree	96.461% (+/-0.01)	1.5497 seconds
Random Forest	99.403% (+/-0.00)	17.5872 seconds
K-Nearest Neighbor	96.382% (+/-0.00)	5.9833 seconds
Artificial Neural Network (Multi-layer Perceptron classifier)	97.836% (+/-0.02)	197.8681 seconds

The use of undersampling although decreased the execution time in relation to control (without data balancing strategy) but the accuracy was penalized for all algorithms in a general way.

In the use of oversampling, the execution time was penalized, having increased greatly, but brought great improvements in terms of accuracy, with the Random Forest algorithm reaching 99.403% of accuracy.

5.5.2 Feature Selection Evaluation

Since the use of oversampling resulted in best results, to study the Feature Selection, all the experiments related to Binary classification for colorectal cancer detection and Multi class classification for disease detection will use this technique.

In this section, it is intended to study if the selection a subset of relevant features, that where determined in the *Feature Selection* chapter, for use in model construction can improve the execution time and mainly the model performance.

Binary Classification for disease detection

For Binary Classification for disease detection, using Pearson's correlation coefficient to analyse how correlated where the features, it was pointed out that the removal of the Isoleucine would not impact the model due to the keeping of Valine.

As can be seen in table 21, with the use of Pearson's correlation coefficient to determine the removal of one of the features, there are not significative changes in relation to both time to generate the model and its accuracy.

Table 21 – Performance comparison with the use of Feature Selection (Pearson's correlation coefficient) in Binary Classification for disease detection

Algorithm	Binary Classification for disease detection (control)		Binary Classification for disease detection with Feature Selection	
	Accuracy (standard deviation)	Execution Time	Accuracy (standard deviation)	Execution Time
Gaussian Naive Bayes	77.085% (+/-0.03)	0.0226 seconds	76.303% (+/-0.02)	0.0206 seconds
Logistic Regression	83.002% (+/-0.04)	6.4618 seconds	82.376% (+/-0.05)	6.7749 seconds
Support Vector Machine	85.635% (+/-0.04)	1.6910 seconds	85.687% (+/-0.04)	1.6807 seconds
Decision Tree	77.501% (+/-0.06)	0.3385 seconds	78.153% (+/-0.05)	0.3392 seconds
Random Forest	83.262% (+/-0.06)	4.8610 seconds	83.001% (+/-0.06)	4.6204 seconds
K-Nearest Neighbor	84.409% (+/-0.05)	0.3563 seconds	84.409% (+/-0.05)	0.4051 seconds
Artificial Neural Network (Multi-layer Perceptron classifier)	84.957% (+/-0.05)	22.7046 seconds	85.009% (+/-0.05)	21.4192 seconds

Multi class Classification for disease detection

For Multi class Classification for disease detection, using Pearson's correlation coefficient to analyse how correlated where the features, it was pointed out that the removal of the Isoleucine would not impact the model due to the keeping of Valine.

As shown in table 22, for the multi class classification for disease detection, the behaviour is very similar to the one observed in binary classification for disease detection, with the removal of one feature not provoking a significative change in both accuracy and execution time.

Table 22 - Performance comparison with the use of Feature Selection (Pearson's correlation coefficient) in Multi class Classification for disease detection

Algorithm	Multi class Classification for disease detection (control)		Multi class Classification for disease detection with Feature Selection	
	Accuracy (standard deviation)	Execution Time	Accuracy (standard deviation)	Execution Time
Gaussian Naive Bayes	70.967% (+/-0.03)	0.0297 seconds	70.386% (+/-0.03)	0.0313 seconds
Logistic Regression	90.589% (+/-0.01)	52.5869 seconds	90.618% (+/-0.01)	52.6890 seconds
Support Vector Machine	91.275% (+/-0.01)	2.4445 seconds	91.304% (+/-0.01)	3.1869 seconds
Decision Tree	86.556% (+/-0.01)	0.4502 seconds	86.241% (+/-0.01)	0.5513 seconds
Random Forest	92.734% (+/-0.01)	5.1296 seconds	92.706% (+/-0.01)	6.3313 seconds
K-Nearest Neighbor	89.302% (+/-0.01)	0.4091 seconds	89.159% (+/-0.01)	0.4379 seconds
Artificial Neural Network (Multi-layer Perceptron classifier)	91.275% (+/-0.01)	43.4466 seconds	91.104% (+/-0.01)	55.2224 seconds

Binary Classification for colorectal cancer detection

For Binary Classification for colorectal cancer detection, using Pearson's correlation coefficient to analyse how correlated where the features, it was pointed out that no pair of features where correlated to the point that one could be removed, so no comparison will be compared using that feature selection technique.

In the table 23 it is presented a comparison between the performance of Binary Classification for colorectal cancer detection (control) and the performance of the Top 10 features with the best average scores using Univariate Feature Selection.

Table 23 - Performance comparison with the use Top 10 features with the best average scores using Univariate Feature Selection in Binary Classification for disease detection

	Binary Classification for colorectal cancer detection (control)		Binary Classification for colorectal cancer detection with Univariate Feature Selection	
Algorithm	Accuracy (standard deviation)	Execution Time	Accuracy (standard deviation)	Execution Time
Gaussian Naive Bayes	88.486% (+/-0.03)	0.0105 seconds	92.343% (+/-0.02)	0.0251 seconds
Logistic Regression	93.67% (+/-0.03)	3.8483 seconds	91.117% (+/-0.03)	0.3249 seconds
Support Vector Machine	90.275% (+/-0.04)	0.0435 seconds	88.972% (+/-0.05)	0.0358 seconds
Decision Tree	88.582% (+/-0.04)	0.0194 seconds	91.135% (+/-0.04)	0.025 seconds
Random Forest	95.904% (+/-0.03)	0.7885 seconds	94.304% (+/-0.02)	0.7871 seconds
K-Nearest Neighbor	80.496% (+/-0.05)	0.0157 seconds	79.645% (+/-0.05)	0.0405 seconds
Artificial Neural Network (Multi-layer Perceptron classifier)	88.98% (+/-0.02)	6.0262 seconds	86.826% (+/-0.06)	5.0116 seconds

Using this technique for feature selection that enabled the removal of the majority of the features, the accuracy was only very slightly impacted, having the best model (generated with Random Forest algorithm) reached 94.304% (+/-0.02) compared with the 95.904% (+/-0.03) of the control.

In the table 24 it is presented a comparison between the performance of Binary Classification for colorectal cancer detection (control) and the performance of the Top 12 features, the features present in all runs in the tier-1 of importance using Recursive Feature Elimination.

Table 24 - Performance comparison with the use 12 features present in all runs in the tier-1 using Recursive Feature Elimination in Binary Classification for disease detection

	Binary Classification for colorectal cancer detection (control)		Binary Classification for colorectal cancer detection with Recursive Feature Elimination	
Algorithm	Accuracy (standard deviation)	Execution Time	Accuracy (standard deviation)	Execution Time
Gaussian Naive Bayes	88.486% (+/-0.03)	0.0105 seconds	88.094% (+/-0.03)	0.0241 seconds
Logistic Regression	93.67% (+/-0.03)	3.8483 seconds	86.897% (+/-0.05)	0.4385 seconds
Support Vector Machine	90.275% (+/-0.04)	0.0435 seconds	88.98% (+/-0.04)	0.0310 seconds
Decision Tree	88.582% (+/-0.04)	0.0194 seconds	90.866% (+/-0.03)	0.0260 seconds
Random Forest	95.904% (+/-0.03)	0.7885 seconds	95.754% (+/-0.02)	0.7027 seconds
K-Nearest Neighbor	80.496% (+/-0.05)	0.0157 seconds	80.106% (+/-0.04)	0.0319 seconds
Artificial Neural Network (Multi-layer Perceptron classifier)	88.98% (+/-0.02)	6.0262 seconds	87.713% (+/-0.03)	4.0477 seconds

The use of Top 12 features according to Recursive Feature Elimination, having into account that less than half of the features were used provided great results, this is no significant differences comparing to the control (with exception to the model generated using Logistic Regression).

The accuracy of the model is important, but other performance metric with great importance when dealing with disease detection is the model's ability to not fail in detecting the patients that actually have the disease, since it is much less costly to falsely diagnose a patient with a disease (false positive) than otherwise. With this in mind in the table 25, is analysed how the algorithms and the use of feature selection behave in the goal of minimizing the false negatives.

Table 25 – Percentage of False Negatives for each model

	Binary Classification for colorectal cancer detection (control)	Binary Classification for colorectal cancer detection with Univariate Feature Selection	Binary Classification for colorectal cancer detection with Recursive Feature Elimination
Algorithm	False Negatives (%)	False Negatives (%)	False Negatives (%)
Gaussian Naive Bayes	6,78%	3,39%	7,63%
Logistic Regression	0,85%	5,08%	10,17%
Support Vector Machine	4,24%	6,78%	5,08%
Decision Tree	8,47%	6,78%	5,93%
Random Forest	2,37%	3,90%	3,39%
K-Nearest Neighbor	4,24%	5,93%	7,63%
Artificial Neural Network (Multi-layer Perceptron classifier)	4,24%	9,32%	6,78%

Analysing the false negatives, the use of feature selection although being able with Random Forest of obtaining a low percentagem, it is notorious that in general perform worse than the control.

Evaluating by accuracy Random Forest is the algorithm that is able to generate the best model, but the Logistic Regression (without feature selection) is the algorithm that generates the model with the best (the lowest) false negatives rate, achieving only 0,85% of false negatives.

Feature selection based in the aminoacid profile

As one of the goals of this dissertation is to study the aminoacid profile and its ability to by itself being able to generate reliable models. It was done an experiment where only were considered the features that are aminoacids, being removed the age and gender features from the control (without feature selection) and the age from both feature selection techniques, since in both it was one of the top features by importance. In the table 26 the results are presented and in the table 27 it is shown how the models perform in terms of false negatives.

Table 26 – Performance using only the aminoacid profile features in the model generation

	Binary Classification for colorectal cancer detection (control)	Binary Classification for colorectal cancer detection with Univariate Feature Selection	Binary Classification for colorectal cancer detection with Recursive Feature Elimination
Algorithm	Accuracy (standard deviation)	Accuracy (standard deviation)	Accuracy (standard deviation)
Gaussian Naive Bayes	74.669% (+/-0.05)	70.321% (+/-0.05)	70.183% (+/-0.07)
Logistic Regression	83.945% (+/-0.07)	78.848% (+/-0.04)	78.023% (+/-0.07)
Support Vector Machine	83.91% (+/-0.05)	77.535% (+/-0.04)	78.83% (+/-0.04)
Decision Tree	86.844% (+/-0.04)	86.445% (+/-0.02)	89.257% (+/-0.04)
Random Forest	93.434% (+/-0.04)	89.415% (+/-0.03)	94.144% (+/-0.02)
K-Nearest Neighbor	76.684% (+/-0.06)	74.566% (+/-0.03)	77.119% (+/-0.03)
Artificial Neural Network (Multi-layer Perceptron classifier)	83.901% (+/-0.05)	81.791% (+/-0.04)	85.186% (+/-0.05)

Table 27 – Percentage of False Negatives for each model using only the aminoacid profile features in the model generation

	Binary Classification for colorectal cancer detection (control)	Binary Classification for colorectal cancer detection with Univariate Feature Selection	Binary Classification for colorectal cancer detection with Recursive Feature Elimination
Algorithm	False Negatives (%)	False Negatives (%)	False Negatives (%)
Gaussian Naive Bayes	10,17%	6,78%	11,86%
Logistic Regression	10,17%	15,25%	17,80%
Support Vector Machine	12,71%	13,56%	19,49%
Decision Tree	9,66%	9,66%	5,42%
Random Forest	4,24%	4,24%	2,20%
K-Nearest Neighbor	5,08%	9,32%	7,63%
Artificial Neural Network (Multi-layer Perceptron classifier)	11,86%	13,56%	10,17%

The gender and age removal from the control had a great impact in the control with, in general, a worse performance in both accuracy and the presence of false negatives.

To note that Random Forest is consistently generating the best models, and used with Recursive Feature Elimination, even with removal of the age feature, there was no improvement in the number of false negatives, achieving only 2,20%.

5.5.3 CRC Model Validation

After the model generation, new 4 instances of patients with colorectal cancer were provided to see how the models would behave with completely new data.

As in general the models generated using the Random Forest algorithm obtained the best results, that is the algorithm selected to do the comparison shown in the table 28 that presents the ability to identify that the patient has colorectal cancer, based in the different number of features used to model generation.

Table 28 – Model detection of patient with colorectal cancer using Binary Classification for colorectal cancer models

	Control (all features)	Control (with all features except age and gender)	Univariate Feature Selection (all selected features)	Univariate Feature Selection (all selected features except age)	Recursive Feature Elimination (all selected features)	Recursive Feature Elimination (all selected features except age)
Patient 1	Not detected	Not detected	Not detected	Not detected	Not detected	Not detected
Patient 2	Detected	Detected	Not detected	Not detected	Not detected	Not detected
Patient 3	Detected	Detected	Not detected	Not detected	Not detected	Not detected
Patient 4	Detected	Detected	Not detected	Not detected	Not detected	Not detected

Although, as shown in table 27 in our tests the models using feature selection for colorectal cancer detection provided good results for the Random Forest algorithm, with a low number of false negatives, when facing new data the results were against the expectations. All of the models using Feature Selection that involved removing features from aminoacid profile were unable to correctly diagnose the patients. On the other hand both the models generated using all the features and all the features belonging to the aminoacid profile (all features except age and gender) were able to correctly identify 3 out of 4 with colorectal cancer.

The same experiment was conducted for the models generated for Binary Disease Detection and Multi class classification of diseases, as can be seen in table 29. Due to the good results, the models were generated using the Random Forest algorithm with Oversampling.

Table 29 - Model detection of patient with colorectal cancer using Binary and Multi Class Classification for disease detection

	Binary Disease Detection (all features)	Binary Disease Detection (with all features except age and gender)	Multi class classification of diseases (all features)	Multi class classification of diseases (all selected features except age)
Patient 1	Detected	Detected	Not detected	Not detected
Patient 2	Detected	Detected	Detected	Detected
Patient 3	Detected	Detected	Detected	Detected
Patient 4	Detected	Detected	Detected	Detected

In the case of disease detection, the model obtained 100% accuracy, being able to diagnose all the patients with disease. In the case of multi class classification for diseases, the model correctly diagnosed 3 out of 4 patients with colorectal cancer. Misclassifying only the Patient 1, at least the patient was not classified wrongly with another disease, but with the absence of associated disease. There was no difference in the results using all the features or only the aminoacid profile (all features except age and gender).

Conclusion

This chapter presents the conclusions and hypotheses resulting from the preparation of the present work, as well as an approach to potential future work that could be carried out within the scope of the stated theme and objectives.

6.1 Conclusion

The results with the application of the final models had a satisfactory accuracy and mainly false negatives rate, and fulfilled the defined objectives, allowing to verify that the amino acid profile can be a good indicator for the disease detection and, more specifically, of colorectal cancer.

This study should then serve only as more a tool to help in the early cancer detection and not as the only analysis made by them.

6.2 Future Work

From a perspective of continuation of this work, the following points should be considered:

- Make a better assessment of the different parameters and settings of the different algorithms in the generation of models, evaluating its possible improvements.
- Improve the system used, to be more in line with the desired architecture and respecting the presented uses cases.
- Test more data pre-processing techniques, especially at the feature selection level, to optimize the prediction with the fewest possible number of features.

- Conduct more experiments, with larger and different sets of data, in order to consolidate the results obtained. Even, if the data is sufficient, test the use of deep learning.
- Explore the alterations in the aminoacid profile after treatments and based on that help the healthcare professionals decision making for what treatment should be provided.

References

- [1] Jensen PB, Jensen LJ, Brunak S. Mining electronic health records: towards better research applications and clinical care. *Nat Rev Genet* 2012;13:395–405.
- [2] “New Product Development Methods” [Online]. Available: <https://www.speakingofproducts.com/new-product-development-methods>. [Accessed: 01-Feb-2021].
- [3] Naoyuki Okamoto, Yohei Miyagi, Akihiko Chiba, Makoto Akaike: Diagnostic modeling with differences in plasma amino acid profiles between non-cachectic colorectal/breast cancer patients and healthy individuals (2009).
- [4] Susana Nicola, Eduarda Pinto Ferreira, J. J. Pinto Ferreira. (2012). A Novel framework for modeling value for the customer, an essay on negotiation. *International Journal of Information Technology & Decision Making*, 11:03, 661-703
- [5] V. Allee, “ValueNet Works Fieldbook,” 2006.
- [6] Allee, Verna (2002). A Value Network Approach for Modeling and Measuring Intangibles.
- [7] Porter, Michael E (1985). *Competitive Advantage - Creating and Sustaining Superior Performance*. doi: 10.1182/blood-2005-11-4354.
- [8] Jozée Lapierre, 2000. Customer-perceived value in industrial contexts
- [9] Woodall, Tony (2003). Conceptualising ‘Value for the Customer’: An Attributional, Structural and Dispositional Analysis . In: *Academy of Marketing Science Review* 12.5, pp. 1 42. issn: 14705931.
- [10] Floor L, Dumont JE, Maenhaut C, Raspe E. Hallmarks of cancer : of all cancer cells , all the time - *Trends in Molecular Medicine*. 2012;18(9):509–15
- [11] 2020 Cancer incidence and mortality in EU-27 countries [Online] Available: <https://ec.europa.eu/jrc/en/news/2020-cancer-incidence-and-mortality-eu-27-countries> [Accessed: 02-Dec-2020]

- [12] New aspects of amino acid metabolism in cancer Lisa Vettore, Rebecca L. Westbrook and Daniel A. Tennant, British Journal of Cancer (2020) 122:150–156;
- [13] Issam El Naqa, Ruijiang Li, Martin J. Murphy. Machine Learning in Radiation Oncology: Theory and Applications, 2015
- [14] Konstantina Kouroua, Themis P.Exarchos, Konstantinos P.Exarchos, Michalis V.Karamouzis, Dimitrios I.Fotiadis (2015). Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal, Volume 13, 2015, Pages 8-17
- [15] X. Zhu, A. B. Goldberg, “Introduction to Semi – Supervised Learning”, Synthesis Lectures on Artificial Intelligence and Machine Learning, 2009, Vol. 3, No. 1, Pages 1-130
- [16] Cherkassky VS, Mulier F. Learning from data: concepts, theory, and methods. 2nd ed. Hoboken: IEEE Press/Wiley-Interscience; 2007
- [17] Machine Learning basics [Online] Available: <https://brianasimba.github.io/MachineLearningblog//Introduction-post/>
- [18] Ian Goodfellow, Yoshua Bengio, Aaron Courville. (2016). Deep Learning, [pdf] MIT Press. Disponível em: <http://www.deeplearningbook.org>
- [19] Supervised and Unsupervised Machine Learning Algorithms. [online] Available in: <https://machinelearningmastery.com/supervised-and-unsupervisedmachine-learning-algorithms/> [December 2nd 2020].
- [20] L. P. Kaelbling, M. L. Littman, A. W. Moore, “Reinforcement Learning: A Survey”, Journal of Artificial Intelligence Research, 4, Page 237-285, 1996
- [21] S.B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques”, Informatica 31 (2007) 249-268
- [22] L. Rokach, O. Maimon, “Top – Down Induction of Decision Trees Classifiers – A Survey”, IEEE Transactions on Systems, 2005
- [23] Comparing different supervised machine learning algorithms for disease prediction, Shahadat Uddin, Arif Khan, Md Ekramul Hossain and Mohammad Ali Moni. BMC Medical Informatics and Decision Making (2019);
- [24] D. Lowd, P. Domingos, “Naïve Bayes Models for Probability Estimation”, 2005
- [25] Jeffrey Tweedale, Lakhmi C. Jain. (2012). Advanced Techniques for Knowledge Engineering and Innovative Applications. em 16th International Conference, KES 2012, San Sebastian, Spain, Setembro 10-12.
- [26] Peng, C.-Y.J., Lee, K.L., Ingersoll, G.M., 2002. An Introduction to Logistic Regression Analysis and Reporting. J. Educ. Res. 96, 3–14.

- [27] Cortes, C. & Vapnik, V. (1995). Support-vector network. *Machine Learning*, 20, 1–25
- [28] Support Vector Machines [Online] Available: <https://www.datacamp.com/community/tutorials/support-vector-machines-r>
- [29] A. Kampouraki, G. Manis, and C. Nikou. (2009). Heartbeat time series classification with support vector machines. *Information Technology in Biomedicine, IEEE Transactions*, 13(4):512–518.
- [30] Song, Y.H., Johns, A., Aggarwal, R. (2016). *Computational Intelligence Applications to Power Systems*, Science Press & Kluwer Academic Publishers, Beijing
- [31] Miotto R et al (2017) Deep learning for healthcare: review, opportunities and challenges. *Briefings in Bioinformatics*
- [32] M. Xiong, J. Chen, Z. Wang, C. Liang, Q. Zheng, Z. Han, et al., Deep Feature Representation via Multiple Stack Auto-Encoders, in *Advances in Multimedia Information Processing-PCM 2015*, ed: Springer, 2015, pp.275-284.
- [33] Dinggang Shen, Guorong Wu and Heung-Il Suk, Deep Learning in Medical Image Analysis, *Annu. Rev. Biomed. Eng.* 2017. 19:221–48
- [34] Machine Learning vs Deep Learning: Its Time You Know the Difference [Online] Available: <https://www.mygreatlearning.com/blog/is-deep-learning-better-than-machine-learning/>
- [35] Dispelling Myths: Deep Learning vs. Machine Learning [Online] Available: <https://www.merkleinc.com/blog/dispelling-myths-deep-learning-vs-machine-learning>
- [36] Migran N. Gevorkyan, Anastasia V. Demidova, Tatiana S. Demidova, Anton A. Sobole. Review and comparative analysis of machine learning libraries for machine learning, 2019
- [37] Guido Dornhege, José del R. Millán, Thilo Hinterberger, Dennis J. McFarland, Klaus-Robert Muller, 2007. *Toward Brain-Computer Interfacing*, A Bradford Book.
- [38] Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques* [pdf]. (pp. 83–124).
- [39] Recursive Feature Elimination (RFE) for Feature Selection in Python [Online] Available: <https://machinelearningmastery.com/rfe-feature-selection-in-python/>

- [40] Feature selection using Python for classification problems [Online]
Available: <https://towardsdatascience.com/feature-selection-using-python-for-classification-problem-b5f00a1c7028>
- [41] B. Venkatesh, J. Anuradha (2019). A Review of Feature Selection and Its Methods. CYBERNETICS AND INFORMATION TECHNOLOGIES - Volume 19, No 1
- [42] Divya Jain, Vijendra Singh (2018). Feature selection and classification systems for chronic disease prediction: A review. Egyptian Informatics Journal. Volume 19, Issue 3, November 2018, Pages 179-189
- [43] Jack Tan (2020). Feature Selection for Machine Learning in Python — Wrapper Methods. Accessed in: 10th May 2021, In: <https://towardsdatascience.com/feature-selection-for-machine-learning-in-python-wrapper-methods-2b5e27d2db31>.
- [44] Bartosz Krawczyk (2016). Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence volume 5, pages221–232.
- [45] Tingting Pan, Junhong Zhao, Wei Wu, Jie Yang (2016). imbalanced datasets based on SMOTE and Gaussian distribution. Information Sciences Volume 512, February 2020, Pages 1214-1233
- [46] Richard Zuech, John Hancock , Taghi M. Khoshgoftaar (2021). Detecting web attacks using random undersampling and ensemble learners. Journal of Big Data volume 8, Article number: 75 (2021)
- [47] Sanjay Yadav and Sanyam Shukla (2016). Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. 2016 IEEE 6th International Conference on Advanced Computing
- [48] Daniel Berrar (2019). Cross-validation. Data Science Laboratory, Tokyo Institute of Technology
- [49] Miotto R et al (2017) Deep learning for healthcare: review, opportunities and challenges. Briefings in Bioinformatics