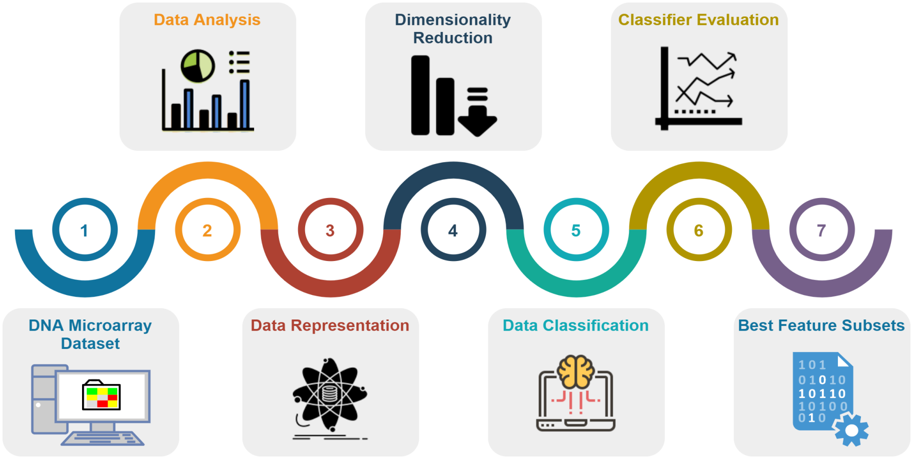
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Departamento de Engenharia Eletrónica e de Telecomunicações e Computadores



Clinical Data Mining and Classification

Adara Stéfanny Rodrigues Nogueira

(Bachelor's degree)

Dissertação para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores

Orientador : Prof. Doutor Artur Jorge Ferreira

Júri:

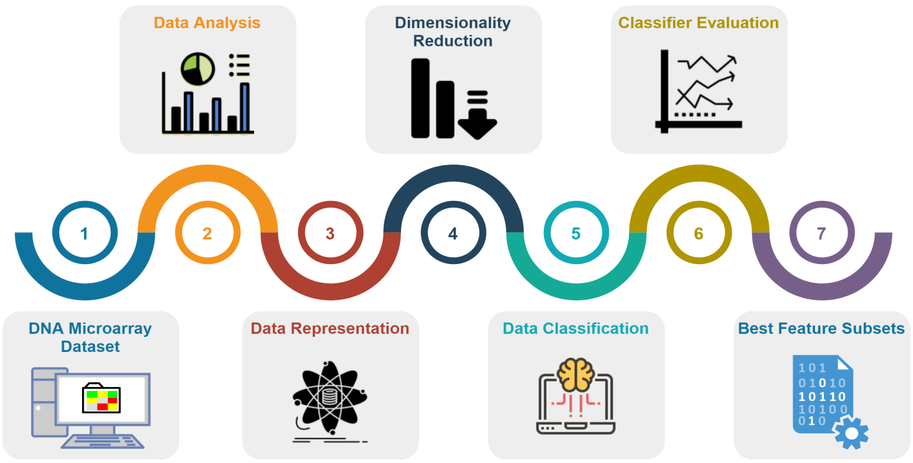
Presidente: Prof. Doutor Tiago Miguel Braga da Silva Dias

Vogais: Prof. Doutor Pedro Mendes Jorge
Prof. Doutor Artur Jorge Ferreira

March, 2022

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Departamento de Engenharia Eletrónica e de Telecomunicações e Computadores



Clinical Data Mining and Classification

Adara Stéfanny Rodrigues Nogueira

(Bachelor's degree)

Dissertação para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores

Orientador : Prof. Doutor Artur Jorge Ferreira

Júri:

Presidente: Prof. Doutor Tiago Miguel Braga da Silva Dias

Vogais: Prof. Doutor Pedro Mendes Jorge
Prof. Doutor Artur Jorge Ferreira

March, 2022

To my dear mom and dad.

Acknowledgments

As I was writing this thesis, I have received a great deal of support and assistance. Thus, I would like to thank the following:

To Prof. Doctor Artur Jorge Ferreira, my thesis supervisor, for his outstanding support, guidance, and patience. I thank him for teaching me and helping me in many aspects of this work. I'm very grateful for everything.

To Prof. Doctor Mário Figueiredo, for his support and comments on the paper associated with this thesis.

To my best friend, Diogo Guerreiro, for his love, support, encouragement, and for believing in me.

To my friend, Luis Loureiro, for his support and encouragement along the past years.

To my grandmother, Maria Isabel, for her unconditional love and support.

Finally, to my parents, Claudiana Silva and Pedro Pires, for their unconditional love, patience, support, and commitment in my education. I thank them for inspiring my life and believing in me. This thesis is dedicated to them.

Thank you, very much

Adara Nogueira

Abstract

Determining which genes contribute to the development of certain diseases, such as cancer, is an important goal in the forefront of today's clinical research. This can provide insights on how diseases develop, can lead to new treatments and to diagnostic tests that detect diseases earlier in their development, increasing patients chances of recovery.

Today, many gene expression datasets are publicly available. These generally consist of DNA microarray data with information on the activation (or not) of thousands of genes, in specific patients, that exhibit a certain disease. However, these clinical datasets consist of high-dimensional feature vectors, which raises difficulties for clinical human analysis and interpretability - given the large amounts of features and the comparatively small amounts of instances, it is difficult to identify the most relevant genes related to the presence of a particular disease.

In this thesis, we explore the usage of feature discretization, feature selection, and classification techniques applied towards the problem of identifying the most relevant set of features (genes), within DNA microarray datasets, that can predict the presence of a given disease. We propose a machine learning pipeline with different feature discretization, feature selection, and classification techniques, to different datasets, and compare/interpret the achieved results with different combinations of techniques.

On most datasets, we were able to obtain lower classification errors by applying either feature discretization or feature selection techniques (but not both). When applying feature selection techniques, we were also able to reduce the number of features fed to each classifier, while maintaining or improving the classification results. These smaller subsets of genes are thus easier to interpret by human clinical experts, improving the explainability of the results.

Keywords: feature selection, feature discretization, microarray data, cancer, explainability of classification.

Resumo

Determinar os genes que contribuem para o desenvolvimento de certas doenças, como o cancro, é um objectivo importante na vanguarda da investigação clínica de hoje. Isto pode fornecer conhecimentos sobre como as doenças se desenvolvem, pode levar a novos tratamentos e a testes de diagnóstico que detectam doenças mais cedo no seu desenvolvimento, aumentando as hipóteses de recuperação dos pacientes.

Hoje em dia, muitos conjuntos de dados de expressão genética estão disponíveis publicamente. Estes consistem geralmente em dados de microarray com informação sobre a activação (ou não) de milhares de genes, em pacientes específicos, que exibem uma determinada doença. No entanto, estes conjuntos de dados clínicos consistem em vetores de características de elevada dimensionalidade, o que levanta dificuldades à análise humana clínica e à interpretabilidade - dadas as grandes quantidades de características e as quantidades comparativamente pequenas de instâncias, é difícil identificar os genes mais relevantes relacionados com a presença de uma determinada doença.

Nesta tese, exploramos a utilização da discretização de características, selecção de características e técnicas de classificação aplicadas ao problema de identificação do conjunto mais relevante de características (genes), dentro de conjuntos de dados de microarray, que podem prever a presença de uma dada doença. Construímos um pipeline onde aplicamos diferentes técnicas de discretização, selecção e classificação, a diferentes conjuntos de dados, e comparamos/interpretamos os resultados obtidos com cada combinação de técnicas.

Na maioria dos conjuntos de dados, conseguimos obter erros de classificação mais baixos aplicando quer técnicas de discretização quer técnicas de selecção (mas não ambas). Ao aplicar técnicas de selecção, conseguimos também reduzir o número de características alimentadas a cada classificador, mantendo ou melhorando os resultados da classificação. Estes pequenos subconjuntos de genes são assim mais fáceis de interpretar pelos especialistas clínicos humanos, melhorando a explicabilidade dos resultados.

Palavras-chave: seleção de características, discretização de características, dados de microarray, cancro, explicabilidade da classificação.

Contents

List of Figures

List of Tables

List of Acronyms

AI	artificial intelligence.
ALL	acute lymphocytic leukemia.
AML	acute myelogenous leukemia.
cDNA	complementary DNA.
CNS	central nervous system.
CV	cross-validation.
DISR	double input symmetrical relevance.
DNA	deoxyribonucleic acid.
DT	decision trees.
EFB	equal frequency binning.
FD	feature discretization.
FiR	Fisher′s ratio.
FN	false negative.
FNR	false negative rate.
FP	false positive.
FPR	false positive rate.
FR	feature reduction.
FS	feature selection.
IDE	integrated development environment.
KNN	K-nearest neighbors.
LOOCV	leave-one-out cross-validation.
LR	logistic regression.
LS	Laplacian score.
MDLP	minimum description length principle.
ML	machine learning.

MLL	mixed lineage leukemia.
MM	mean-median.
mRNA	messenger RNA.
MSE	mean squared error.
NB	naive Bayes.
PIP	pip installs packages.
RF	random forests.
R-LBG	relevance-based Linde-Buzo-Gray.
RNA	ribonucleic acid.
RRFS	relevance-redundancy feature selection.
SRBCT	small round blue cell tumors.
SVM	support vector machines.
TN	true negative.
TP	true positive.
U-LBG1	unsupervised Linde-Buzo-Gray 1.

List of Symbols

Δ	minimum allowed distortion or minimum increase in relevance
$\frac{d}{n}$	ratio of the number of features to the number of instances in a dataset
$\overline{\mathrm{A}}$	average value
σ	standard deviation
c	number of classes in a dataset
d	number of features in a dataset
k	number of folds in the cross-validation procedure
m'	percentage of selected features
m	number of selected features
ms	maximum similarity
n	number of instances in a dataset
q	maximum number of bits per feature
X	$n \times d$ matrix (data matrix)
y	class label vector

1

Introduction

This Chapter describes the motivation, the context in which the problem arises, as well as the key goals of this work. In addition, the problem formulation and the handled datasets are also presented. Finally, an overview of the entire document is provided as well as the scientific contributions of this work.

1.1 Motivation

Nowadays, disease detection from clinical data has assumed great importance. New techniques have been developed that allow healthcare professionals to analyze each patient's *deoxyribonucleic acid* (DNA) and to identify the presence of specific genes, which may be indicative of certain diseases, such as cancer.

Given these developments, many gene expression datasets are publicly available. These generally consist of microarray data that contains information on the activation (or not) of thousands of genes, in specific patients, that exhibit a certain disease. Ideally, one would like to use these datasets, to predict the presence of diseases on new patients, given their microarray data.

However, these clinical datasets consist of high dimensionality feature[1] vectors, which raises analysis problems for humans. It is laborious or almost impossible to identify the most important, decisive factors that lead to the appearance of a particular disease.

[1] Also referred as dimensions in the literature, and genes in this work. Henceforth, for the sake of consistency, we will use the term "features".

In addition to their high dimensionality, the publicly available datasets usually have a small number of instances[2]. This is due to the high cost of acquiring a new instance using this process.

Thus, applying data mining and machine learning techniques on these datasets poses equally high challenges, due to the curse of dimensionality issues [12, 20, 48]. For instance, the performance of the classifiers is sub-optimal and it is often not possible to determine, in detail, which genes are relevant to the presence of a given disease.

Given these constraints, the main goal of this work is to determine the smallest set of features that are indicative of a given disease. The identified features can also be used to the human interpretability of the data and to explain the presence of a given disease.

1.2 Datasets

The clinical data used for this work consists in publicly available DNA microarray datasets, which can be found in [72].

This type of data, reviewed in [6], stores information about the gene expression, collected from cell samples. Thus, it is very helpful in the process of disease diagnosis, in this particular study, cancer.

Table 1.1 presents the name and description for each microarray dataset considered in this work. More details about the datasets can be found in Appendix A.

Table 1.1: Microarray Datasets.

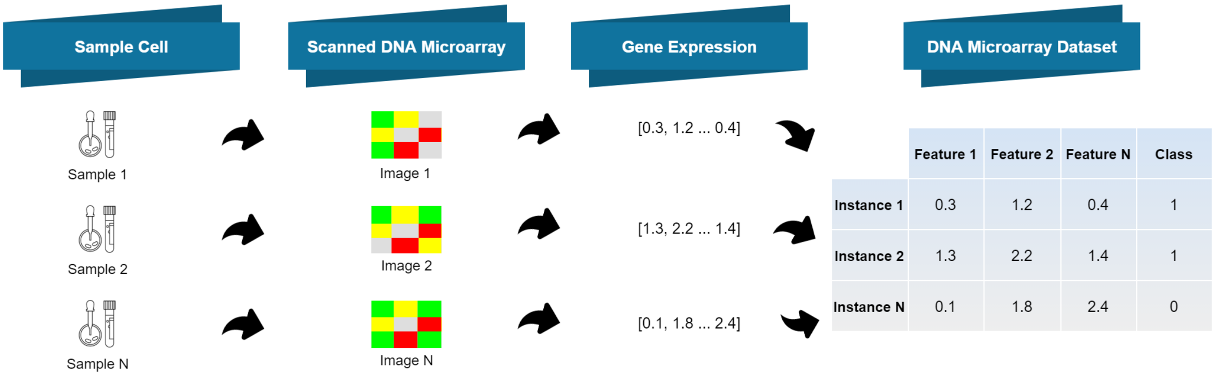
Name	Description
Breast	Breast cancer diagnosis
CNS	*Central nervous system* (CNS) tumor diagnosis
Colon	Colon tumor diagnosis
Leukemia	*Acute lymphocytic leukemia* (ALL) and *Acute myelogenous leukemia* (AML) diagnosis
Leukemia_3c	Distinguishes types of blood cells which became cancerous
Leukemia_4c	Distinguishes types of blood cells which became cancerous
Lung	Lung cancer diagnosis
Lymphoma	Distinguishes subtypes of non-Hodgkin lymphoma
MLL	Distinguishes types of acute leukemia, including *Mixed lineage leukemia* (MLL)
Ovarian	Ovarian cancer diagnosis
SRBCT	Distinguishes types of *small round blue cell tumors* (SRBCT)

[2] Also referred as patterns, examples, and samples in the literature. Henceforth, for the sake of consistency, we will use the term "instances".

1.3 Problem Formulation

Given a DNA microarray dataset, consisting of a relatively small number of instances, each represented by a row in the dataset per individual, as depicted in Figure 1.1, each containing a high dimensionality feature vector with the activation/expression of a set of genes, classified under specific criteria (e.g. the presence/absence of a disease or the phase of a disease) we would like to determine the smallest subset of features (gene activations) that can be used to correctly classify new instances. With this approach, we aim to find the most relevant and disease related genes (features) to explain the presence or absence of a given disease.

Figure 1.1 depicts the process of generating a dataset from a DNA microarray [47]. The datasets considered in this work were obtained using this process.



	Feature 1	Feature 2	Feature N	Class
Instance 1	0.3	1.2	0.4	1
Instance 2	1.3	2.2	1.4	1
Instance N	0.1	1.8	2.4	0

Figure 1.1: DNA Microarray Dataset Generation.

Cancer comes in second as the leading cause of mortality worldwide. Approximately 10 million people died from cancer in 2020, from which lung, colon, and breast cancer were the most common causes [70].

Early and accurate disease detection often leads to higher chances of survival and lower cost of treatment. As an example, according to the World Health Organization, early detection of cancer increases survival rates, decreases morbidity and typically leads to less expensive treatments (for instance, see "Early Diagnosis" section of [70]).

In studying this problem, we hope we can improve the accuracy of DNA based clinical tests while minimizing the risk of false negatives, and thus making a positive contribution in society worldwide.

1.4 Goals

Given the problem formulated above, the goal of this study is to apply several data mining and machine learning techniques to publicly available datasets, compare their performance and conclude which one is best fitted to the task of identifying the smallest set of features capable to correctly classify new data.

In order to achieve this goal, we will have the following approach:

- review the scientific literature on this subject;
- choose which techniques to evaluate;
- build a machine learning pipeline using data representation/discretization, dimensionality reduction and data classification techniques;
- compare the performance of each technique;
- and finally, identify the best suited technique as well as the best subset of features to the problem and datasets under consideration.

1.5 Thesis Contribution

A condensed version of the contents of this work was published as followed:
Adara Nogueira, Artur Ferreira and Mário Figueiredo, *"A Step Towards the Explainability of Microarray Data for Cancer Diagnosis with Machine Learning Techniques"*, International Conference on Pattern Recognition Applications and Methods (ICPRAM), February 2022, pages 362-369, ISBN 978-989-758-549-4, ISSN 2184-4313, DOI:10.5220/0010980100003122.

This paper was the recipient of the best poster award.

1.6 Thesis Structure

The remainder of the main text of this document is organized in five chapters, followed by three appendix.

Chapter 2, State of the Art, focuses on both the clinical, as well as the computational, aspects of this problem. It starts with a brief explanation of what is a DNA microarray, what types of data it produces and how it is used in clinical studies. It is followed by a description on the machine learning techniques mentioned in the remainder of this thesis. This chapter ends with a review on the scientific literature on the data representation, dimensionality reduction, and classification techniques applied to microarray data.

Chapter 3, Data Analysis, consists in the analysis of the publicly available datasets, exploring their characteristics and correlations.

In Chapter 4, Machine Learning Pipeline Implementation, we describe the proposed approach to compare and select the best techniques. In particular, the proposed pipeline and each of its phases are described in detail.

Chapter 5, Experimental Results, contains the experimental results, their evaluation and analysis. Several metrics, such as error and false negative rates, will be used to derive conclusions for each dataset.

Chapter 6, Conclusions, provides an overview of what was achieved, describes the conclusions and directions of future work.

Appendix A, Dataset Description, contains the description of each DNA microarray dataset used in the context of this work.

In Appendix B, Pipeline's Configurations, we describe the analysis of the best fitted techniques for each dataset used in the machine learning pipeline.

Finally, Appendix C, Additional Experimental Results, provides additional experimental results about the most relevant features.

2

State of the Art

This Chapter describes the main concepts underlying this work. Section 2.1 addresses the DNA microarray technique from a clinical perspective. Section 2.2 outlines the machine learning commonly techniques used in this area. Finally, Section 2.3 describes the existing approaches to this problem.

2.1 DNA Microarray Technique

Every biological organism has a set of genes encoded in its DNA. These may be expressed, i.e. active, in different cells at different points in time. In the context of biological/medical research it's important to understand which genes are being expressed (active/inactive) in a given cell, at a given point in time.

However, living beings have thousands of genes, e.g. a human has approximately 21000 [15, 49]. Each one of these genes is responsible for encoding a protein, which is in charge of a specific functionality. Given the complexity and amount of information, healthcare professionals are unable to analyze this data one gene at a time. Even if it were possible, it would take a very long time and the efficiency and accuracy of the analysis would be extremely low.

The DNA microarray technique [47] addresses this issue. A DNA microarray allows researchers and healthcare professionals to carry out an investigation on thousands of genes at the same time, i.e. in one single experiment[1], and determine which genes are being expressed by a cell.

A DNA microarray has the following characteristics:

- a microarray is a solid surface with thousands of spots arranged in well-ordered columns and rows;
- each spot on this microarray characterizes only one gene and contains multiple strands of the same DNA, i.e. the DNA sequence is unique;
- each spot location and its respective DNA sequence is recorded in a database.

Among other uses, DNA microarrays can identify dissimilarities between cancer cells and healthy cells, more specifically, which genes in a cancer cell are being expressed which are not in a healthy cell.

Figure 2.1 presents an overview of the DNA microarray technique.

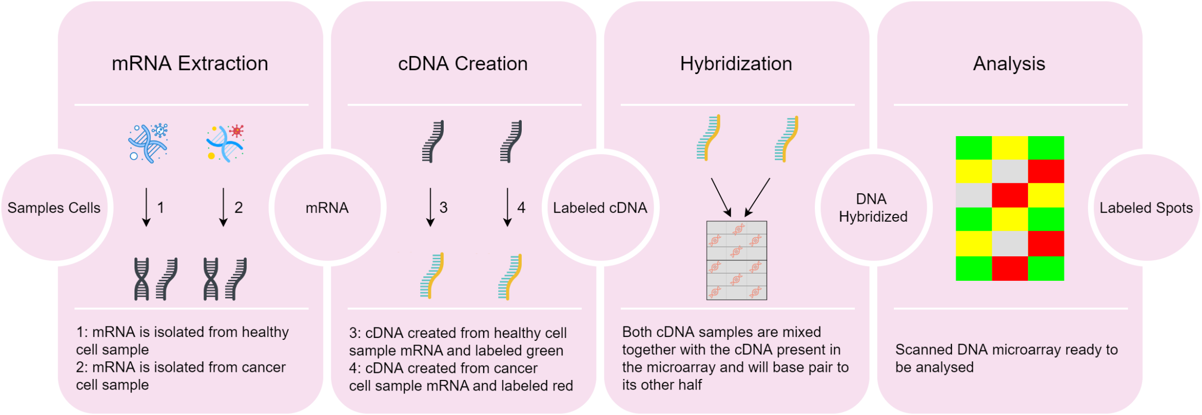

mRNA Extraction
cDNA Creation
Hybridization
Analysis
Samples Cells
1
2
mRNA
3
4
Labeled cDNA
DNA Hybridized
Labeled Spots
1: mRNA is isolated from healthy cell sample
2: mRNA is isolated from cancer cell sample
3: cDNA created from healthy cell sample mRNA and labeled green
4: cDNA created from cancer cell sample mRNA and labeled red
Both cDNA samples are mixed together with the cDNA present in the microarray and will base pair to its other half
Scanned DNA microarray ready to be analysed

Figure 2.1: DNA Microarray Technique.

First, it's necessary to extract the *ribonucleic acid* (RNA) from the sample cells and then draw out the *messenger RNA* (mRNA) from the existing RNA, because only the mRNA develops gene expression.

Second, a DNA copy is made from the mRNA with the aid of the reverse transcriptase enzyme, which will generate the *complementary DNA* (cDNA). In this process, a label is added in the cDNA representing each cell sample, e.g. a fluorescent red for the cancer cell and a fluorescent green for the healthy cell. This step is necessary because the DNA is a more stable molecule than RNA and the labeling allows us to identify the genes in each sample, at a later stage.

[1] An example of a DNA microarray experiment can be found in [16]

Third, both cDNA types previously created are added to the DNA microarray and because each spot of it already has many unique cDNA, when mixed together they will base pair each other due to the DNA property, designated complementary base pairing. This process is denominated "hybridization". Not all cDNA strands will bind to each other, some may not hybridize therefore they need to be washed off.

Finally, the DNA microarray is analyzed with a scanner, which can find patterns of hybridization by detecting the fluorescent colors. As a result, we can observe the following:

- only a few red cDNA molecules bound to a spot, which means that the gene was being expressed only in the red (cancer) cell;
- only a few green cDNA molecules bound to another spot, which means that the gene was being expressed only in the green (healthy) cell;
- some of both red and green cDNA molecules bound to a single spot on the microarray (forming a yellow spot), which means that the gene was being expressed both in the cancer and the healthy cell;
- several spots of the microarray don't have a single red or green cDNA strand bound to it, because the gene is not being expressed in either cell.

Thus, on the one hand the red color on a spot indicates the higher production of mRNA in the cancer cell as compared to the healthy cell. On the other hand, the green color specifies the higher production of mRNA in the healthy cell as compared to the cancer cell. However, a yellow spot suggests that the gene is expressed equally in both cells and therefore, they are not relevant as the cause of the disease, because when the healthy cell becomes cancerous their activity does not undergo a change.

In conclusion, with DNA microarray we can analyze a large amount of genes at the same time, find which genes are being expressed and decide on a better prognosis based on the previous analyzes.

2.2 Machine Learning Techniques

Machine Learning (ML) [36, 63] is a branch of *artificial intelligence* (AI) [59] that focuses on designing algorithms that learn from past experience, or existing data, in order to solve new tasks.

Rather than explicitly "programming" a computer, or telling it what to do, ML techniques rely on showing a computer a set of data associated with a given task, and having the computer analyze it in order to learn and build its own internal model. This model can be used to solve future tasks based on new data, that was not used on the model building phase.

ML algorithms can be applied to different tasks, such as data representation, dimensionality reduction and data classification, among others.

The following concepts are relevant to machine learning:

- Instance - refers to a vector containing the data from one patient (in the context of this work);
- Feature - refers to the set of dimensions associated to an instance, e.g. in this work each feature corresponds to one specific gene activation;
- Class label[2] - refers to the values that tag each data instance;
- Dataset - refers to the set of instances available to be used by the learning algorithm. In the context of this work, datasets contain gene activations from several DNA microarrays and we have labeled instances;
- Training Dataset - refers to the subset of instances and the corresponding class labels used to train the learning algorithm;
- Test Dataset - refers to the subset of instances and the corresponding class labels used to evaluate the learning algorithm's performance.

ML techniques can be grouped in two major approaches: supervised learning and unsupervised learning.

In supervised learning, the algorithm receives a set of labeled instances, i.e. it receives the input and the expected output/classification for each instance. Its goal is to learn an internal model/representation that can be used to classify/identify the label for unseen instances, correctly.

In unsupervised learning, the training dataset which the algorithm receives does not have classes and the learning process is conducted based on the analysis of the instances present in the data.

[2] Also referred as label, for the sake of simplicity.

In the context of this work, the focus will be on supervised learning techniques, since the problem to be solved relies on labeled datasets of DNA microarray data. Unsupervised learning and other machine learning techniques, will not be explored further here.

Figure 2.2 summarizes the supervised learning process, regarding the use of the available data.

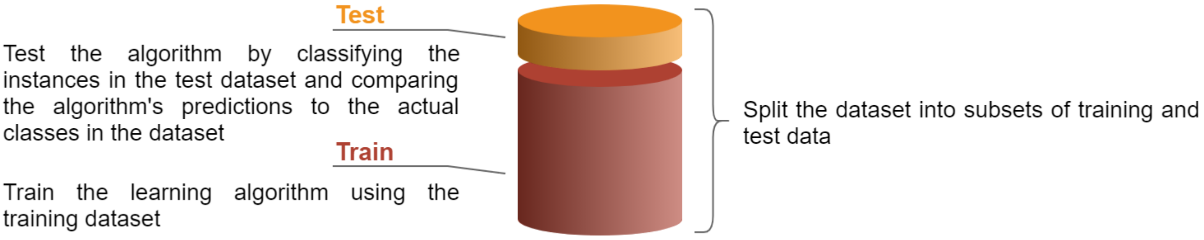

Test
Test the algorithm by classifying the instances in the test dataset and comparing the algorithm's predictions to the actual classes in the dataset
Train
Train the learning algorithm using the training dataset
Split the dataset into subsets of training and test data

Figure 2.2: Supervised Learning Process.

As depicted in Figure 2.2, the process of supervised learning relies on a labeled dataset that is divided into two subsets of training and test data. The former is used to train a classifier, whereas the latter is used to evaluate the trained classifier by classifying the instances in the test set and comparing the classifier's predictions to the actual class labels in the dataset.

This is the general approach, however we can partition the training and test set using a different methodology, e.g. applying a *cross-validation* (CV) technique, such as 10-fold or *leave-one-out cross-validation* (LOOCV) [13].

The CV is a resampling procedure used to assess the performance of an algorithm, i.e., to estimate the accuracy of a learning algorithm. In this methodology, the original dataset is randomly partitioned into k subsets of data, each called a fold. Then, the learning algorithm is applied k times, using k-1 folds as a training set and the remaining as a test set. A 10-fold technique is a common case of CV where the number of folds is equal to 10 (k=10). In comparison, a LOOCV is a particular case of CV in which the number of folds is equal to the number of instances in the dataset. Hence, the learning algorithm is applied as many times as the number of instances, and for each time a single unique instance is used as a test set and the remaining as a training set.

Training and evaluating a classifier with the LOOCV procedure, we have no bias in the results, caused by the sampling procedure.

With a CV technique, k tests are performed, and the final result is obtained by averaging the evaluation metric from these k tests, e.g. the error rate. As long as the dataset is representative of the problem, this average result is a reliable estimate. Therefore, if by any chance a new instance from the same distribution as the original dataset is created, the algorithm is capable to manifest, in this new instance, a performance as good and as reliable as the averaged one attained with the CV procedure.

The advantage of applying a CV technique is that by performing k number of tests, biased conclusions can be avoided, and we will be able to understand how well our algorithm generalizes to new data. With only one test set the result of the algorithm's performance might only be a fluke. In addition, it also allows us to evaluate and compare different algorithms' performance instead of only one per time.

On the other hand, the disadvantage of applying a CV technique is the high computation cost, which increases as the number of folds (k) grows.

2.2.1 Data Representation

The datasets considered in this work consist of high dimensionality feature vectors. Since these features contain a large amount of information regarding gene expressions, they may contain irrelevant fluctuations (noise) [47]. These fluctuations lead to complexity hence bad performance, when applying some machine learning algorithms. Data representation techniques allow us to handle such problems [2, 37].

Data representation refers to how real world information is stored inside a computer system, for the purposes of storing, transforming or transmitting such information. In the context of this thesis, several techniques are used to generate, or transform between different representations of the same real world information, such as feature normalization and *feature discretization* (FD) [2, 53]. In addition, data analysis (e.g. statistical analysis [37]) is also regarded as a form of data representation.

There are many feature normalization methods in the literature [25]. However, in this work we consider only the MinMaxScaler. This procedure consists in scaling each feature between a minimum and maximum range (such as 0 and 1).

The FD procedure [2] consists in transforming continuous features into discrete ones. This transformation can be achieved through unsupervised and supervised approaches.

The unsupervised *equal frequency binning* (EFB) method [26] divides continuous features into a given number of intervals (bins) which contain approximately the same number of instances.

The *unsupervised Linde-Buzo-Gray 1* (U-LBG1) [3] is a technique that discretizes each feature into a specified number of intervals. This discretization process is applied until the minimum *mean squared error* (MSE) between the original and the discretized feature falls under a determined threshold (Δ) or when the maximum number of bits per feature (q) is reached.

The supervised *minimum description length principle* (MDLP) method recursively divides the features values into multiple intervals, using a information gain minimization heuristic (entropy). Please, refer to [65] for a formal description of this method.

The *relevance-based Linde-Buzo-Gray* (R-LBG) [5] is a supervised FD technique similar to U-LBG1. The difference is that, instead of relying on the MSE, it relies on a relevance function and a stop criterion: each feature is discretized until there is no significant increase on its relevance. The relevance function can be computed using, for instance, the *Fisher's ratio* (FiR) [52], a supervised metric.

Please refer to [26, 53] for some additional insights on other FD approaches.

2.2.2 Dimensionality Reduction

In the presence of high-dimensional data, dimensionality reduction techniques [18, 24] have proven essential to generate adequate representations of the data and to improve the machine learning results. There are many dimensionality reduction techniques in the literature, which can be broadly categorized as *feature selection* (FS) approaches or *feature reduction* (FR) approaches. We now briefly review some of these dimensionality reduction techniques.

FS techniques [18, 24] consist in selecting a subset of features from the original one. One way to perform feature selection is to rank features according to their relevance. This relevance is computed using some similarity based methods. Some well-known methods in the literature are:

- Unsupervised methods - *Laplacian score* (LS) [71], Spectral (also known as SPEC) [73], and term-variance [33];
- Supervised methods - FiR [52] and ReliefF [23].

Relevance-redundancy feature selection (RRFS) [1] works in unsupervised mode using the *mean-median* (MM), an unsupervised relevance metric, and in supervised mode using the FiR metric.

FR techniques [18, 24] consist in generating a new and smaller subset of features, computed from the original ones. Some well-known methods in the literature are:

- Unsupervised methods - principal component analysis [21], factor analysis [18], and multidimensional scaling [18];
- Supervised methods - linear discriminant analysis [18], canonical correlation analysis [18], and partial least squares [24].

Please refer to [18, 24] for additional insights on dimensionality reduction techniques.

2.2.3 Data Classification

2.2.3.1 Support Vector Machine

Support vector machines (SVM) [13, 18] are a set of supervised methods used in ML for the task of classification. They treat the input as a set of d-dimensional feature vectors in a d-dimensional vector space and try to identify a hyperplane that can separate the instances of each class, maximizing the distance between the hyperplane and each feature vector.

The method separates the data by finding support vectors to move them apart between different classes, in the form of f(x) = wx + b. The support vectors are calculated with an iterative process using an optimization algorithm. That is, the support vectors are given by a kernel function, which divides the feature space.

The kernel functions, also called cost functions, considered in this work are: linear, polynomial, radial-basis function (rbf), and sigmoid. In kernel functions, two parameters must be considered, C and gamma. The parameter C can be used in any of the four kernel functions while gamma is used in all but the linear kernel function. The C parameter is a regularization parameter - the higher this parameter is, the more the classifier tries to correctly classify the training examples. Fine tuning the gamma parameter may yield better results.

2.2.3.2 Decision Trees

Decision trees (DT) [13, 18] is another classification technique. This method employs a hierarchical tree-like model, which contains decisions and their corresponding labels. The DT will work from top to bottom, testing each feature and comparing them. It is also an iterative technique, like SVM, dividing the data among its classes, with rules defined through iterative decisions.

We have four well-known DT algorithms. The ID3 (search for each node, the categorical feature), C4.5 (successor to ID3, removing the categorical feature exception), C5.0

(uses less memory, but less precise than C4.5), and CART (similar to C4.5, but supports numeric variables). Each algorithm uses different criteria for splitting the nodes of the tree (Gini index and entropy).

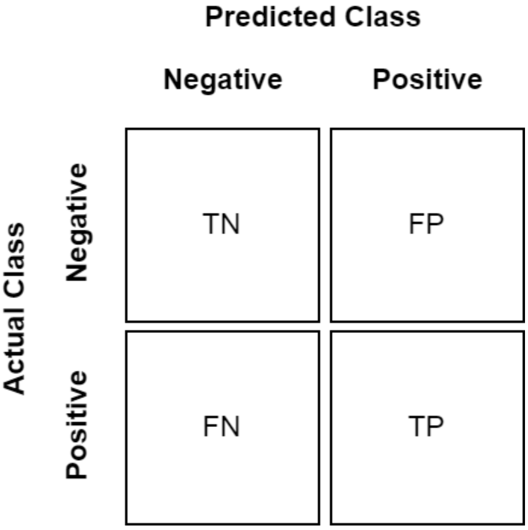
The parameters to take into consideration are: the depth of the tree and randomization, which controls the randomness of the estimator.

2.2.4 Evaluation Metrics

We now describe commonly used metrics for evaluating the classifiers' performance.

A confusion matrix [13] summarizes the performance of a classification algorithm. In this work, its rows represent the actual class labels and the columns the predicted class labels.

Figure 2.3 shows the structure of a confusion matrix for a binary classification task: a (2x2) matrix.



		Predicted Class	
		Negative	Positive
Actual Class	Negative	TN	FP
	Positive	FN	TP

Figure 2.3: Confusion Matrix.

Each cell contains the count of correct/incorrect classifications (being the total value the same as the number of instances). The cells are designated as:

- A *true negative* (TN) occurs when the instance is negative and it is classified as negative;
- A *true positive* (TP) occurs when the instance is positive and it is classified as positive;

- A *false positive* (FP) occurs when the instance is negative and it is classified as positive;
- A *false negative* (FN) occurs when the instance is positive and it is classified as negative.

Good classifier performance corresponds to a scenario with large numbers down the main diagonal, and small numbers (ideally zero) in the remaining cells.

Additional metrics are built on top of this confusion matrix.

The accuracy measures the proportion of correct classifications out of all predictions, and is given by

$$accuracy = \frac{TN + TP}{TN + TP + FN + FP}. \tag{2.1}$$

The error rate (Err) measures the proportion of incorrect classifications out of all predictions, which is given by

$$Err = 1 - accuracy. \tag{2.2}$$

The *false negative rate* (FNR) measures the proportion of actual positive instances which are incorrectly identified as negative, which is given by

$$FNR = \frac{FN}{FN + TP}. \tag{2.3}$$

The *false positive rate* (FPR) measures the proportion of actual negative instances which are incorrectly identified as positive, which is given by

$$FPR = \frac{FP}{FP + TN}. \tag{2.4}$$

Besides the accuracy, the most important metric to be assessed in this type of data is the FNR. When it comes to human beings, the cost of FN is too high, since we would be classifying cancer patients as healthy when in reality they would not be - this could result in a loss of human lives.

Please refer to [14] for additional insights on evaluation metrics.

2.3 Existing Approaches and Related Work

Given its applications in the fields of medical and biological research, the problem of "selecting the smallest set of features, from high dimensionality feature vectors, that can be used to best classify new instances" has been the focus of considerable attention from the research community.

In the last decades, for the purpose of diagnosing diseases such as cancer, there has been considerable research on microarray data classification. In addition, to assess better performance, before classification takes place, many unsupervised and supervised FD and FS techniques have been employed on this type of data.

We now briefly review some of the existing approaches and related work that uses FD, FS, and classification techniques.

The work [26] presents a review of continuous FD techniques. The authors considered three discretization methods: equal width intervals (also known as equal interval binning), one-rule discretizer proposed by Holte [50], and MDLP. According to the authors, MDLP proved to have better performance when applied with DT and *naive Bayes* (NB) [63] classifiers.

It has been found that unsupervised FD performs well when combined with several classifiers. For instance, [41] reports that applying equal interval binning and EFB on microarray data, together with SVM classifiers, yields good results. It has also been reported in [22], that applying the EFB technique with the NB classifier produces very good results.

The approaches proposed on [3] (U-LBG1 technique) and [5] (R-LBG technique) are reported to have yielded good improvements on classification accuracy, when working with high-dimensional datasets.

The work of [4] shows that FS significantly improves the classification accuracy of multi-class SVM classifiers and other classification algorithms.

In [45], FS techniques (such as backwards elimination of features) and classification techniques were explored, both using *random forests* (RF) [31]. The authors applied the chosen method on one simulated and nine real microarray datasets and found that RF has better performance than other classification methods, such as diagonal linear discriminant analysis, *K-nearest neighbors* (KNN) [63], and SVM. They also demonstrated that the FS technique used led to a smaller subset of features than alternative techniques, namely nearest shrunken centroids [51] and a combined method using a filter with a nearest neighbor classifier.

A FS filter for microarray data, with an information-theoretic criterion named *double input symmetrical relevance* (DISR), which measures feature complementarity, was proposed by [41]. The reported experimental results on one synthetic dataset and eleven microarray datasets show that the DISR criterion is competitive with existing FS filters.

The proposed relevance/redundancy approach from [1] (RRFS technique) has proven to be adequate for high-dimensional microarray datasets.

In addition, an overview of some of the most popular FS algorithms (such as LS, SPEC, and FiR) was provided in the study [29]. According to the authors, FS techniques have proven again to be very effective and efficient when working with high-dimensional data.

The study [74] introduces the use of large-scale linear support vector machines and recursive feature elimination with variable step size, as an enhancement to the traditional technique of FS based upon on support vector machines with recursive feature elimination [24], which is considered one of the best methods in the literature, but has vast computational time requirements. The improved approach consists in upgrading the recursive feature elimination process by varying the step size with the goal of reducing the number of iterations (the step size is kept higher in the initial stages of this process where non-relevant features are discarded). In addition, the standard SVM is upgraded to a large-scale linear SVM and thus accelerating the method of assigning weights. The authors compare their approach to FS with SVM and RF, and use the SVM, NB, KNN and *logistic regression* (LR) [13] as classifiers. These techniques were applied on six microarray datasets and the authors demonstrate that the new approach provides better performance with comparable levels of accuracy. They also observed that SVM and LR outperform the other two classifiers.

Recently, in the context of the fight against cancer, [7] considered the problem of finding a small subset of features capable of discerning among six classes of instances. These classes may be healthy or cancerous. The goal is to define a comprehensive set of rules based on the most relevant features (selected by their technique) that can distinguish classes based on their gene expressions. The proposed method combines a genetic algorithm [63] to conduct FS and a fuzzy rule-based system [13] to execute classification on a single dataset with 21 instances and more than 45 thousand features. In this study, ten rules were devised such that each rule always takes into account specific features, which makes these features crucial in explaining the classification results of ovarian cancer detection.

Finally, a survey of common classification techniques and related methods to increase their accuracy for microarray analysis is presented by [6, 67]. The experimental evaluation is carried out in publicly available datasets, as in this work.

3

Data Analysis

This Chapter describes the data analysis performed on each DNA microarray dataset used in this work. This analysis aims to better understand the data and to identify potential problems that may arise (e.g. due to the type of the features and class imbalance). In Section 3.1, we describe the datasets used, in terms of number of features, instances, and classes. Finally, Section 3.2 the statistical analysis on the different datasets is presented.

3.1 Characteristics and Basic Information

It's good practice to understand and identify the basic information about the data we have, such as the total number of instances, features, and classes. This kind of information allows us to recognize the dimension of the dataset and, therefore, to conclude whether if it is necessary or not to apply some technique, such as dimensionality reduction. Furthermore, it also makes possible to identify the different types of classifier to be applied, based on the number of classes, i.e. binary classification or a multi-class classification.

Table 3.1 presents the main characteristic of all microarray datasets used in this work, which were introduced in Chapter 1 (please, see Table 1.1). In Table 3.1, n denotes the number of instances, d indicates the number of features, c the number of classes, and $\frac{d}{n}$ is the ratio of the number of features to the number of instance, which is a measure of how many more features we have than instances.

Table 3.1: Microarray Datasets Characteristics.

Name	# Instance (n)	# Feature (d)	# Class (c)	Ratio $\frac{d}{n}$
Breast	97	24481	2	252.38
CNS	60	7129	2	118.81
Colon	62	2000	2	32.25
Leukemia	72	7129	2	99.01
Leukemia_3c	72	7129	3	99.01
Leukemia_4c	72	7129	4	99.01
Lung	203	12600	5	62.06
Lymphoma	66	4026	3	61.00
MLL	72	12582	3	174.75
Ovarian	253	15154	2	59.89
SRBCT	83	2308	4	27.80

As we can see in the "# Feature" column, all datasets have a large number of features, ranging from 2000 to 24481. In addition, as evidenced by the "# Instance" column, all datasets have a small number of instances, ranging from 60 to 253.

The high number of features and the reduced number of instances will make it difficult to apply standard supervised learning techniques to these data, since we face the curse of dimensionality problem.

Richard E. Bellman was the first to term the phenomenon known as the "curse of dimensionality". He has shown that the number of instances required to achieve the desired knowledge/accuracy grows exponentially with the dimensionality (number of features) [46]. In high dimensional spaces, the dataset may become sparse due to the lack of combinations of feature values (lack of instances), and in addition to computational cost, overfitting problems may arise, therefore resulting in poor performance [12, 48].

Thus, it's necessary to select the most relevant features to avoid the curse of dimensionality and to improve the performance by making the classification feasible [24]. However, as we can see in Table 3.1, the $\frac{d}{n}$ ratio is quite high for some datasets, which convey the difficulty in applying feature selection in these data.

The column entitled "# Class" indicates if the dataset is binary (c equals to 2) or multi-class (c greater than 2). A binary dataset indicates the presence/absence of a specific tumor/cancer (such as in the CNS, Colon, and Ovarian datasets), the re-incidence of a disease (such as in the Breast dataset), or the diagnosis between two types of cancer (such as in the Leukemia dataset). A multi-class dataset distinguishes between different types of cells (such as in the Leukemia_3c, Leukemia_4c, and Lymphoma datasets), and tumors/cancer (such as in the Lung, MLL, and SRBCT datasets). More details on these datasets can be found in Appendix A.

3.2 Statistical Analysis

A statistical analysis of the data helps to point out the presence of an unbalanced class distribution. It also allows us to check on the type of values and the distribution of each feature. With this information in mind, it's possible to decide what is the best course of action to handle the data.

It's important to highlight that a numerical or quantitative feature can have one of two types:

- Discrete - which can only take integer values, e.g. 1 and 3152;
- Continuous - which can take any real values, e.g. float values, such as 2.3, 2.4, 2.5, and 2.6.

Also, a categorical or qualitative feature can be:

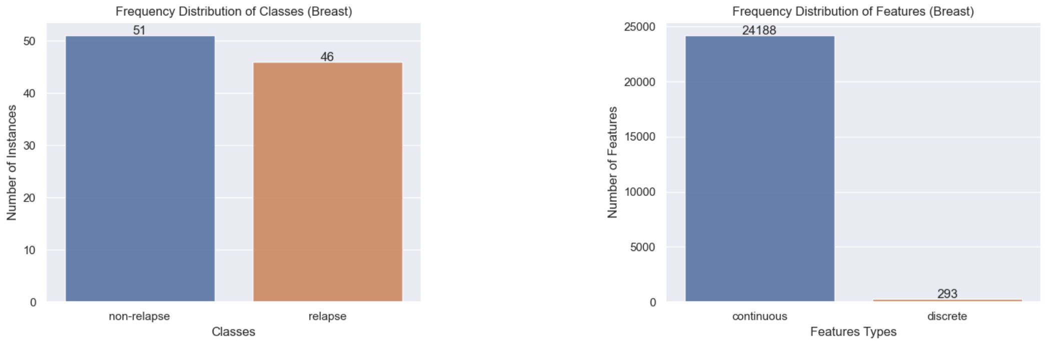
- Nominal - which can take values with no natural order among them, such as colors, e.g. white, red, black, and blue;
- Ordinal - which can take values with an order between them, like a rank, e.g. 1, 2, and 3 or low, medium, and high .

The figures presented in this Section, show the microarray dataset statistical analysis, including the frequency distribution of classes and feature types. These figures were generated using Python's Seaborn [58] and Matplotlib [35] libraries.

As we can observe in Figure 3.1, in the Breast dataset there are 97 instances which are distributed among two classes: label non-relapse (51 instances) and label relapse (46 instances). In addition, there are 24481 features in total, which can be divided in two types with the following distribution:

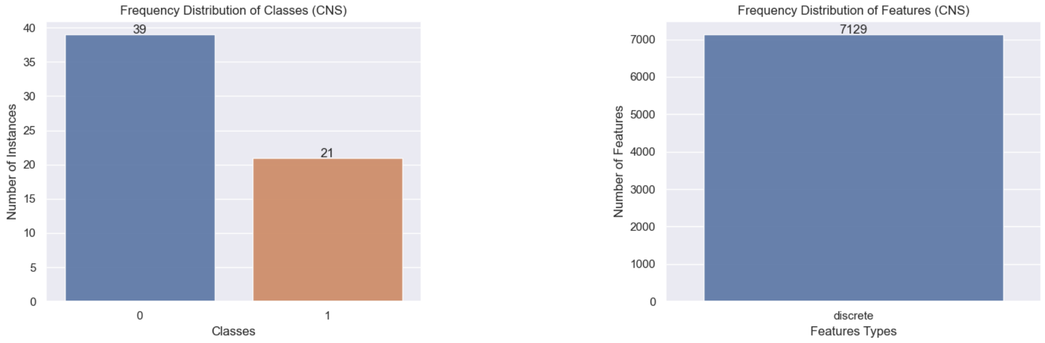
- 24188 continuous features;
- 293 discrete features.

Figure 3.2 shows that in the CNS dataset there are 60 instances which are distributed among two classes: label 0 (39 instances) and label 1 (21 instances). Moreover, there are only discrete features, with a total number of 7129.



(a) (b)

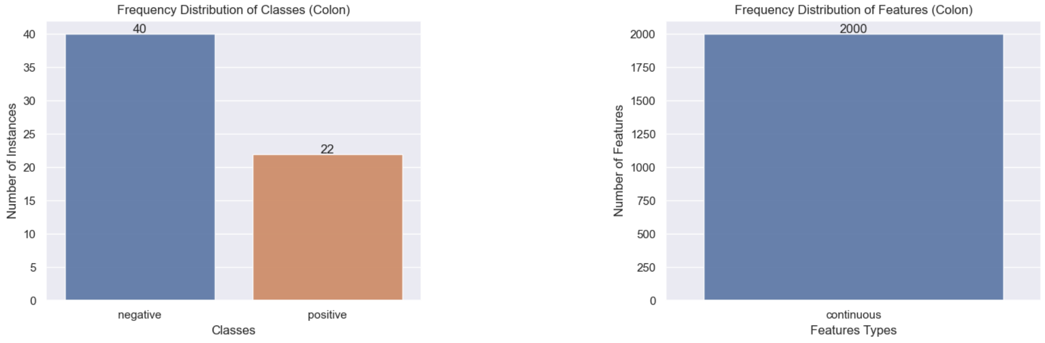
Figure 3.1: Breast Dataset Statistical Analysis.



(a) (b)

Figure 3.2: CNS Dataset Statistical Analysis.

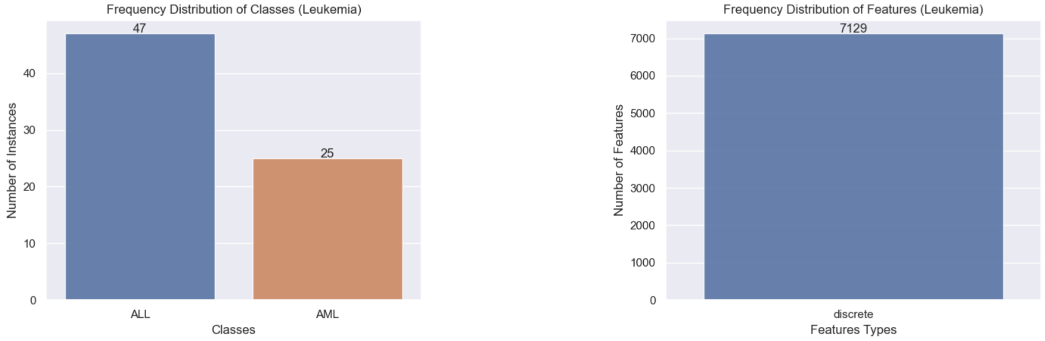
Figure 3.3 describes that in the Colon dataset there are 62 instances which are distributed among two classes: label negative (40 instances) and label positive (22 instances). Additionally, there are only continuous features, totaling up to 2000.



(a) (b)

Figure 3.3: Colon Dataset Statistical Analysis.

Figure 3.4 shows that in the Leukemia dataset there are 72 instances which are distributed among two classes: label ALL (47 instances) and label AML (25 instances). In addition, there are only discrete features, with 7129 in total.



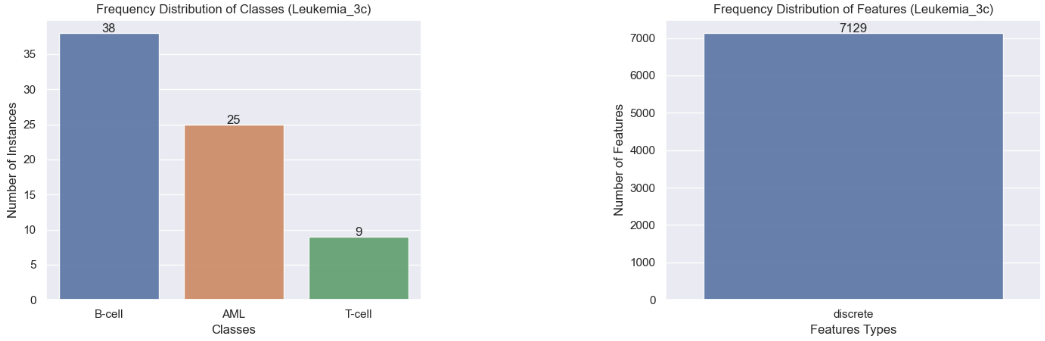
(a) (b)

Figure 3.4: Leukemia Dataset Statistical Analysis.

Likewise, Figures 3.5 and 3.6, present the same number of instances and frequency distribution of feature types as Figure 3.4. On the other hand, the frequency distribution of the class is different.

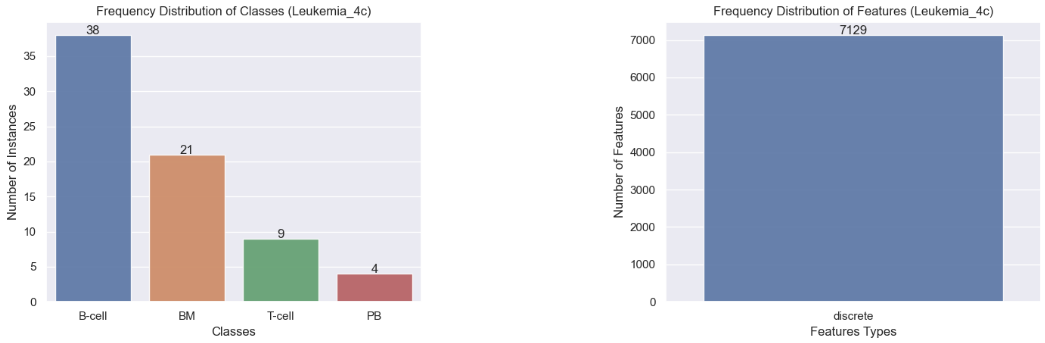
Figure 3.5 shows that in the Leukemia_3c dataset there are 72 instances which are distributed among three classes: label B-cell (38 instances), label AML (25 instances), and label T-cell (9 instances).

Figure 3.6 shows that in the Leukemia_4c dataset there are 72 instances which are distributed among four classes: label B-cell (38 instances), label BM (21 instances), label T-cell (9 instances), and label PB (4 instances).



(a) (b)

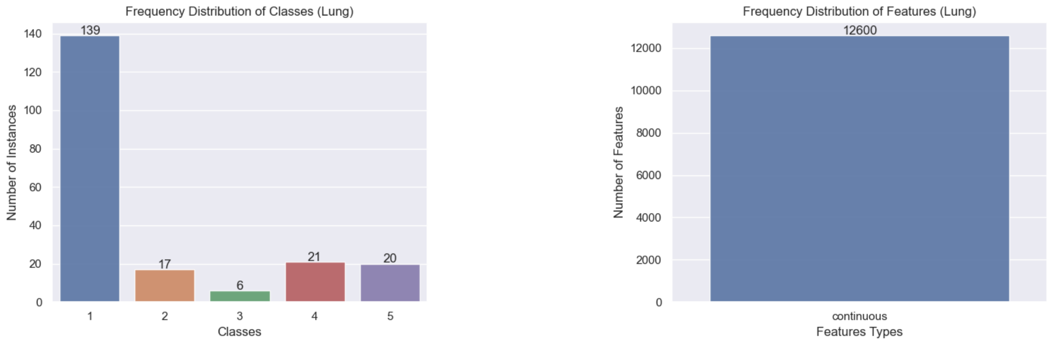
Figure 3.5: Leukemia_3c Dataset Statistical Analysis.



(a) (b)

Figure 3.6: Leukemia_4c Dataset Statistical Analysis.

As we can observe in Figure 3.7, in the Lung dataset there are 203 instances which are distributed among five classes: label 1 (139 instances), label 2 (17 instances), label 3 (6 instances), label 4 (21 instances), and label 5 (20 instances). Moreover, there are only continuous features, with a total number of 12600.



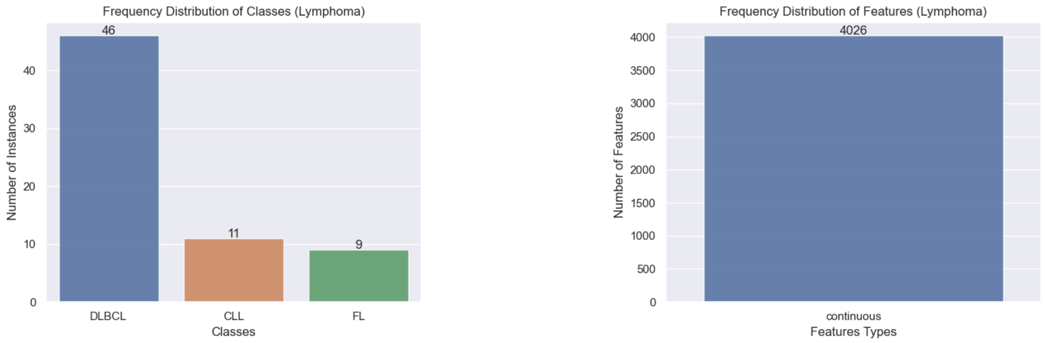
(a) (b)

Figure 3.7: Lung Dataset Statistical Analysis.

Figure 3.8 shows that in the Lymphoma dataset there are 66 instances which are distributed among three classes: label DLBCL (46 instances), label CLL (11 instances), and label FL (9 instances). Additionally, there are only continuous features, totaling up to 4026.

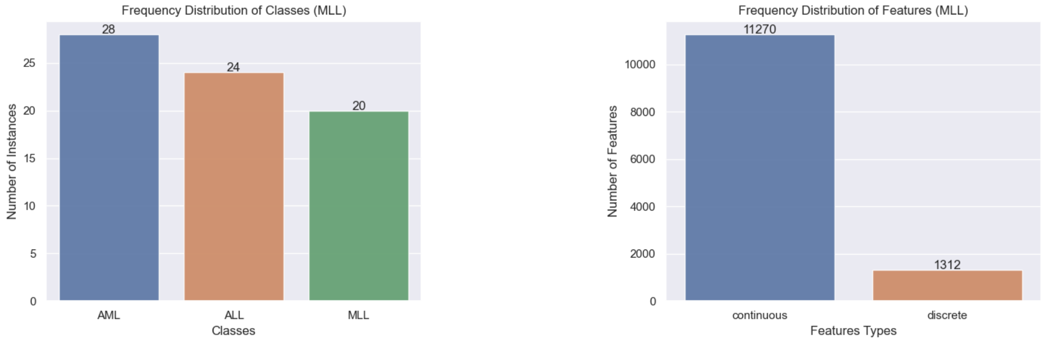
Figure 3.9 shows that in the MLL dataset there are 72 instances which are distributed among three classes: label AML (28 instances), label ALL (24 instances), and label MLL (20 instances). In addition, there are 12582 features in total, which can be divided in two types with the following distribution:

- 11270 continuous features;
- 1312 discrete features.



(a) (b)

Figure 3.8: Lymphoma Dataset Statistical Analysis.



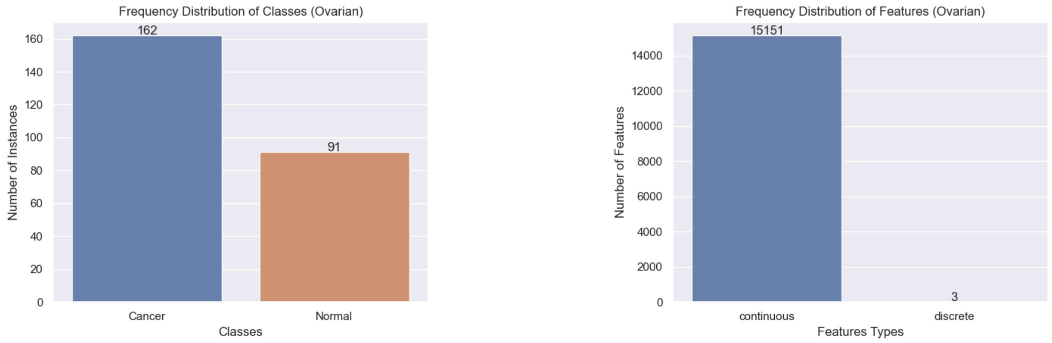
(a) (b)

Figure 3.9: MLL Dataset Statistical Analysis.

Figure 3.10 shows that in the Ovarian dataset there are 253 instances which are distributed among two classes: label Cancer (162 instances) and label Normal (91 instances). Moreover, there are 15154 features in total, which can be divided in two types with the following distribution:

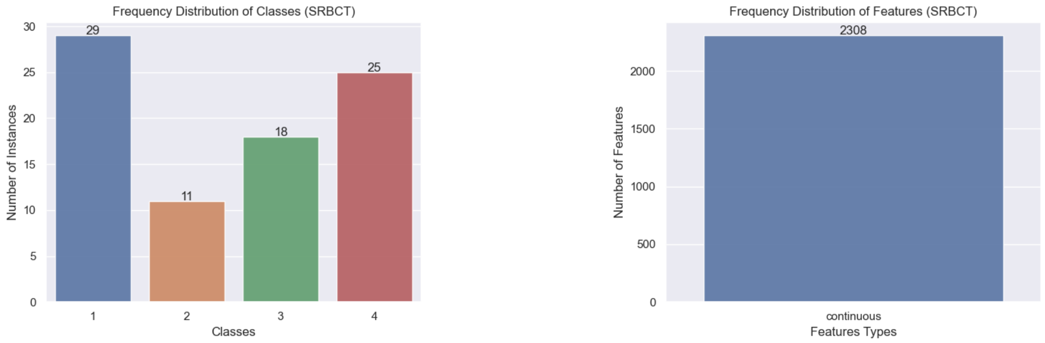
- 15151 continuous features;
- 3 discrete features.

Figure 3.11 shows that in the SRBCT dataset there are 83 instances which are distributed among four classes: label 1 (29 instances), label 2 (11 instances), label 3 (18 instances), and label 4 (25 instances). In addition, there are only continuous features, totaling up to 2308.



(a) (b)

Figure 3.10: Ovarian Dataset Statistical Analysis.



(a) (b)

Figure 3.11: SRBCT Dataset Statistical Analysis.

Given the fact that the datasets considered in this work consist of a high number of features and a reduced number of instances, the most suitable CV technique to apply is the LOOCV. Although, its computational cost is high, the results attained using this procedure are more accurate than that of 10-fold CV.

As mentioned in Section 2.2, the LOOCV technique uses all but one instance to train the classifier. This allows us to use the largest training set and make full use of a small dataset, thus achieving a better estimate of the generalization error and the other evaluation metrics. Due to this same reason, the chances of overfitting (the classifier performing well in the training set, but poorly in the test set [48]) and underfitting (the classifier not performing well neither in the training and nor the test set [18]) are lower.

In addition, the LOOCV doesn't depend on the data sampling. Thus, the average result will not vary, and we do not need to be concerned with not having an even distribution of classes per dataset skewing the results of this study.

Under other circumstances, such as in the presence of datasets in which the dimensionality is not too high and the number of instances is high enough to do a suitable sampling (to test the algorithm ability to generalize), the 10-fold CV technique is better suited than LOOCV, since it is computationally more efficient (faster).

As stated at the beginning of this Chapter, the type of values of each feature can also generate problems.

This may occur in the data representation phase, when applying discretization techniques. Since in this phase we transform continuous features into discrete features, as explained in Subsection 2.2.1, we must ensure that categorical features are not changed. Categorical features already have an inherent meaning, and therefore, don't need to be discretized, only numerical features need to.

However, since the datasets in this work only consist of numerical features (as we can see in the statistical analysis presented above), this is not an issue.

4

Machine Learning Pipeline Implementation

This Chapter describes the approach followed to achieve the goals set forth in this work. In Section 4.1, we comment on the chosen programming language to develop the machine learning pipeline used in this work. Section 4.2 describes each phase of the aforementioned pipeline, by presenting the techniques that we applied and the procedures that we followed. Lastly, Section 4.3 presents the main distinction between the present work and those described in Section 2.3.

4.1 Programming Languages

The pipeline depicted in Figure 4.1 was developed in Python (version 3.7) [43], using Visual Studio Code [66] as the *integrated development environment* (IDE). In addition, alongside Python, the Weka software [69] was chosen to assist with the data analysis, described in Chapter 3, due to its built-in tools for standard machine learning problems and its intuitive user interface.

Python is the *de facto* standard programming language for machine learning today. It contains numerous state of the art libraries to handle machine learning problems, which reduces development time, e.g. Tensorflow [60], Pytorch [44], Scikit-learn [54], Pandas [42], Numpy [39], and SciPy [55]. Other suitable programming languages would have been Matlab, R, and Java.

Matlab [34], allows us to easily deal with machine learning problems by providing a set of resources such as the capability of building training models with only a point and click approach, signal processing, feature extraction techniques, automatic machine learning (e.g. feature selection, model selection, and hyperparameter tuning), large-scale processing and so on. At its core, it focuses heavily on matrix operations which is useful for machine learning problems.

R [61], had its roots as an open source statistical package, with a strong emphasis on vector (and matrix) operations, and a centralized package system through which contributors can add new functionality. Today, it boasts an impressive set of packages for loading and transforming data, doing exploratory data analysis, executing machine learning algorithms and displaying results graphically.

Java [40], is one of the oldest and most used programming languages for enterprise development. Countless mission critical applications in production today are written in Java. This includes some systems for handling big data (e.g. Apache Hadoop and Apache Spark), which are important components of production ready pipelines. Other machine learning software, like Weka, are also written in this language, and most machine learning stacks, even if not written in Java, have language bindings that allows them to be used from Java.

Ultimately, it's up to the programmer to choose which programming language he/she is more comfortable with. For the problem at hand, Python provides all the necessary and adequate tools.

4.2 The Proposed Pipeline

The machine learning pipeline developed in the context of this work, consists of a set of actions that we apply to each dataset. This allows us to compare different techniques and different parameters and evaluate their performance in order to draw our conclusions. Figure 4.1 depicts the pipeline developed in the context of this thesis.

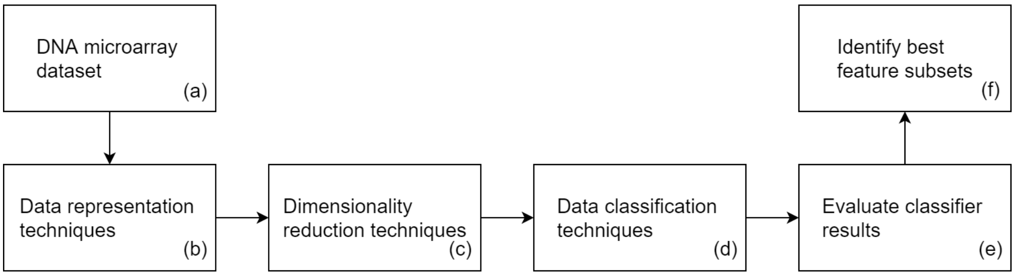

DNA microarray dataset (a)
Data representation techniques (b)
Dimensionality reduction techniques (c)
Data classification techniques (d)
Evaluate classifier results (e)
Identify best feature subsets (f)

Figure 4.1: The pipeline of the proposed approach.

Phase (a), the inputs for the pipeline, consists in one of the eleven publicly available DNA microarray datasets shown in Table 1.1 and Table 3.1, while phase (f), the experimental results and conclusions, are reported in Chapter 5 and Chapter 6.

Algorithm 1 describes the pipeline developed in this work.

Algorithm 1 Machine learning pipeline

```
Input:  11 DNA microarray datasets.
Output: Error rate (Err).
        False negative rate (FNR).
        False positive rate (FPR).
        Percentage of the selected features ($m'$).

 1: for $data \in datasets$ do
 2:     Preprocess the data.
 3:     Instantiate LOOCV.
 4:     for $indexes \in LOOCV$ do
 5:         Split the instances of the data into training and test sets.
 6:         Apply normalization on the training and test sets.
 7:         for $fd \in FD\ techniques$ do
 8:             Apply phase (b) as described in Algorithm 2.
 9:             for $fs \in FS\ techniques$ do
10:                 Apply phase (c) as described in Algorithm 5.
11:                 Select features on the discretized training and test sets.
12:                 Compute how many times a feature was selected.
13:                 for $classifier \in classification\ techniques$ do
14:                     Apply phase (d) as described in Algorithm 7.
15:                     Save the classifier predictions (phase (e)).
16:                 end for
17:             end for
18:         end for
19:     end for
20: end for
21: For all datasets and combinations of techniques applied in this pipeline, compute
    the confusion matrix using the saved predictions, as described in Algorithm 8
    (phase (e)).
22: Compute Err, FNR, and FPR using the confusion matrix.
23: Compute $m'$, the percentage of selected features.
```

At line 2, we preprocess each dataset. This procedure was accomplished in four steps:

1. We splitted the dataset into features set (X) and class labels (y).

2. We mapped all nominal class labels to a representative number (for instance: no cancer is given by 0, whereas having cancer is specified by 1). This was done because some algorithms don't accept nominal class labels.

3. We completed the missing values with the most frequent value in the corresponding feature. We used the SimpleImputer method from Scikit-learn [54]. This was only required for the Lymphoma dataset, as it was the only one with missing values.

4. We removed all constant features, since they didn't provide relevant information for the algorithms applied. This was only required for the Breast dataset, where d, the number of features, was reduced from 24481 to 24188.

In line 3, we instantiate and initialize the LOOCV technique, the methodology we used for training and testing classifiers, which is then applied to all evaluations throughout. By using LOOCV, we achieved a better estimate of the generalization error and the other evaluation metrics, when compared to standard 10-fold CV, since as the number of instances n is small, there is no standard deviation due to the data sampling procedure as it happens on the latter CV technique. More information on this technique can be found at the end of Section 3.2.

At line 12, we compute how many times a feature was selected in the FS phase of the pipeline. For this, we considered a d dimensional counter for each dataset. The counter was implemented as a dictionary, in which each entry was represented by a (key : value) pair corresponding to (feature index : count). At the beginning, we set all count values to zero, and at line 12, for each feature index selected, we increment the counter by 1. Since we are using the LOOCV technique, each feature can be selected (at maximum) n times. In the best case scenario, a feature is selected n times, and at worst, 0 times. Once the pipeline process is finalized, we remove from the map all the pairs (feature index : count) in which the count value is still zero. Next, we sort the values of the remaining pairs by descending order of count, since the higher the value presented in the count, the more relevant the feature will be. This way, we can determine the most relevant features in a dataset and by doing so, we accomplish phase (f) of the pipeline.

4.2.1 Data Representation

Phase (b) was accomplished by applying the feature normalization and FD, data representation techniques as described in Subsection 2.2.1. In addition, Chapter 3 which provides a thorough analysis of each dataset, contains alternative data representations used for analysis purposes: these include tables, charts, summary statistics, statistical analysis, and others.

The feature normalization procedure consisted in scaling each feature between zero and one, which was achieved using the Scikit-learn's MinMaxScaler method. This allows us to balance all features in terms of the range of values they are represented with, and avoid crushing small values in the presence of large ones when performing floating-point computations.

Regarding FD, we noticed that this technique has had little attention in the literature, since it was harder to find algorithms to apply on the pipeline. However, this phase may cause an impact on the data and improve the results as irrelevant fluctuations are removed. That being the case, the storage space is reduced, given that the integer representation can occupy less space than the original floating-point values (e.g. when using short integers vs float64 values). This procedure, was carried out through four techniques:

- Unsupervised EFB, a method (KBinsDiscretizer) from Scikit-learn used as described in Algorithm 2, without the class labels;
- Supervised MDLP, a publicly available FD algorithm implementation from [32] (under BSD 3-Clause License) used as described in Algorithm 2;
- Unsupervised U-LBG1, a FD algorithm proposed in [3], which was implemented as described in Algorithm 3;
- Supervised R-LBG, a FD algorithm proposed in [5], which was implemented as described in Algorithm 4.

Algorithm 2 Data representation, phase (b) of the pipeline

```
Input:   Training set.
         Actual class labels.
         Test set.
Output:  Discretized training set.
         Discretized test set.

1: Instantiate a discretization algorithm.
2: Learn the discretization interval for each feature, on the training set.
3: Discretize the training and test sets, with the learned quantizer.
```

We have chosen the EFB method because even though it's a simple approach, it is known to produce adequate results in the literature when applied to microarray data (in addition to other types of data and machine learning problems). We have also chosen MDLP because it is a different discretization approach compared to EFB. In addition, the U-LBG1 and R-LBG are reported in the literature to achieve good results on multi-class high-dimensional datasets.

Algorithm 3 U-LBG1 - unsupervised Linde-Buzo-Gray 1

Input: Training set.
Test set.
Maximum number of bits per feature (q).
Minimum allowed distortion (Δ).

Output: Discretized training set.
Discretized test set.

```
for $i = 0$ to $d$ do                                    ▷ $d$ is the number of features
    for $j = 1$ to $q$ do
        Apply the kmeans method to the current feature from the training set, yielding the centroids.
        if $MSE < \Delta$ or $j = q$ then
            Apply the vq method to discretize the training and test sets.
            break                                        ▷ Move on to the next feature.
        end if
    end for
end for
```

Algorithm 4 R-LBG - relevance-based Linde-Buzo-Gray, with FiR relevance

Input: Training set.
Actual class labels.
Test set.
Maximum number of bits per feature (q).
Minimum increase in relevance (Δ).

Output: Discretized training set.
Discretized test set.

```
for $i = 0$ to $d$ do                                    ▷ $d$ is the number of features
    Previous feature relevance set to zero.
    for $j = 1$ to $q$ do
        Apply the kmeans method to the current feature from the training set, yielding the centroids.
        For each feature, compute current relevance using FiR.
        if $(current\ relevance - previous\ relevance) > \Delta$ or $j = q$ then
            Apply the vq method to discretize the training and test sets.
            break                                        ▷ Move on to the next feature.
        end if
        Set previous relevance feature value to the current relevance value.
    end for
end for
```

For the EFB technique, we tuned the number of discretization bins, the (n_bins) parameter, since its optimal value may vary for each dataset. Since the MDLP results in a random number of intervals, for its remaining parameters we proceeded with the default values.

As for the U-LBG1 discretizer, we decided not to tune its parameter in this work, because in the literature Δ set to 0.05 and q to 4 has proven adequate for many problems. Thus, we used these values. In R-LBG, we set Δ to 0.1 and q to 4 for the same reason. Additionally, instead of using the Linde-Buzo-Gray algorithm as in [3] and [5], we decided to use the kmeans and vq methods from SciPy. The kmeans algorithm generates a codebook (mapped centroids to codes) with minimum distortion (MSE) between features and centroids. The vq method discretizes the features, assigning to each feature value a code from the codebook.

4.2.2 Dimensionality Reduction

Phase (c) was accomplished by applying the dimensionality reduction techniques described in Subsection 2.2.2, specifically the FS procedure.

We chose to use FS instead of FR techniques, because it has advantages when it comes to performance, processing time and result interpretability. FS selects a subset of the original features rather than generating new ones, which allowed us to remove irrelevant and redundant features, improving the results and time performance. In addition, in FS the feature point of reference in the dimensional space remains the same as in the original, whereas with FR we lose this information, which makes results interpretation much more difficult.

This procedure, was carried out through four techniques:

- Unsupervised LS, a publicly available FS algorithm implementation from [28, 29] (under GPL-2.0 License) used as described in Algorithm 5;
- Unsupervised SPEC, a publicly available FS algorithm implementation from [28, 29] (under GPL-2.0 License) used as described in Algorithm 5;
- Unsupervised and supervised RRFS (with MM and FiR as a relevance metric), a FS algorithm proposed in [1], which was implemented as described in Algorithm 6;
- Supervised FiR, a publicly available FS algorithm implementation from [28, 29] (under GPL-2.0 License) used as described in Algorithm 5.

We have chosen the LS, SPEC, and FiR methods, because they are well-known techniques in the literature. We have also chosen RRFS because unlike other approaches, it keeps only features with high relevance and low similarity among themselves, thus also removing redundant features.

With the exception of the RRFS technique, we decided to not fine tune these technique's parameters, and have used their default values. For the RRFS technique, we set the maximum similarity (*ms*) parameter to 0.7, since it was reported to produce adequate results, in the literature

Algorithm 5 Dimensionality reduction, phase (c) of the pipeline

```
Input:   Training set.
         Actual class labels.
Output:  Indexes of selected features.

Obtain the scores of all the features by using a FS algorithm.
Sort these scores by descending order.
Compute the cumulative sum of these scores.
Select features with the most accumulated relevance (90%).
```

Algorithm 6 RRFS - relevance-redundancy feature selection

```
Input:   Training set.
         Actual class labels.
         Maximum similarity between features ($ms$).
Output:  Indexes of selected features.

Compute the relevance using a metric (MM or FiR).
Sort these relevance by descending order.
Keep the most relevant feature (the first one).
for $i = 1$ to $d$ do                              ▷ $d$ is the number of features
    Compute the similarity between the current and next feature.
    if $similarity < ms$ then
        Keep feature.
    end if
end for
```

By using the LOOCV procedure, we compute the number of remaining features for each fold (*m*). Using *m*, we can calculate the percentage of selected features (*m*′), which is given by

$$m' = \frac{\overline{\mathrm{m}}}{d}, \tag{4.1}$$

where $\overline{\mathrm{m}}$ is the average value for all *m* and *d* is the number of features in the dataset.

4.2.3 Data Classification

Phase (d) was accomplished by applying the data classification techniques described in Subsection 2.2.3:

- SVM, a method (SVC) from Scikit-learn used as described in Algorithm 7;
- DT, a method (DecisionTreeClassifier) from Scikit-learn used as described in Algorithm 7.

Algorithm 7 Data classification, phase (d) of the pipeline

```
Input:   Training set.
         Actual class labels.
         Test set.
Output:  Classifier prediction.

Instantiate a learning algorithm (classifier).
Train the classifier using training set and the actual class labels.
Classify the test set.
```

We have chosen the SVM classifier because it is frequently reported in the literature as the data classification technique that yields the best results in similar contexts. We have also chosen DT because it is a different classification approach, which is seldom applied to this type of data.

For the SVM classifier, we tuned the kernel and C parameters and evaluated its performance with several values on each dataset. By changing the kernel, we aimed to find a suitable kernel function for each dataset/problem, and by changing the C parameter, we aimed to understand the impact that the regularization has in the performance for this type of data.

As for the DT classifier, we tuned the criterion, random_state, and max_depth parameters. By changing the criterion, we aimed to find, for each dataset, a suitable function to split a node on the tree. As for the other two, we aimed to assess its performance for this type of data.

We chose to tune these classifiers based on the parameters mentioned above, because they are the most used when fine tuning these classifiers [19, 68]. It would have been impossible to explore every possible combination of every parameter since then the number of combinations would have grown exponentially and the time to produce results/analyze the data would have been prohibitively large.

4.2.4 Evaluation Metrics

Phase (e) was accomplished by applying a subset of the evaluation metrics described in Subsection 2.2.4. We considered the average test error rate for all the datasets. For the five datasets that have a non-cancer class label, we also considered the FNR and FPR, which are especially relevant metrics for the cancer detection problem. For the other six datasets, we don't report the FNR and FPR metrics, because these datasets focus on distinguishing between cancer types and they do not have a non-cancer class label. Please refer to Appendix A for more information about each dataset class labels.

The selected evaluation metrics can be calculated from a confusion matrix. Therefore, for all datasets and all combinations of techniques applied in this work, we first compute the confusion matrix, as described in Algorithm 8. Then, we calculated the error rate, FNR, and FPR as described in the equations presented in Section 2.2.4.

Algorithm 8 Compute confusion matrix

```
Input:  n predictions.
        Actual class labels.
Output: A cxc confusion matrix (rows: actual class, columns: predicted class).

Build a cxc array filled with zeros.                      ▷ c is the number of classes.
Build a dictionary to map the class label and the position this label will have on the
confusion matrix.
for $i = 0 \to n$ do                                      ▷ n is number of instances
    Get in the dictionary the position corresponding to the current actual class label
    and the predicted one, (row, column) respectively.
    Increment the value in the position (row, column) of the confusion matrix by 1.
end for
```

For consistency and simplicity in interpretability, we ensure that the first position in the confusion matrix is a non-cancer label (TN), where applicable.

4.2.5 Thesis Repository

The source code of this thesis is publicly available at [38].

The requirements.txt file lists all but one Python packages this work depends on. They can be installed using the following command: *pip install -r requirements.txt*.

The additional package [28] was not shared on *pip installs packages* (PIP), therefore it has to be installed individually (by cloning or downloading the zip file).

4.3 Comparison With Other Approaches

This work focuses on the same algorithms and most often used classifiers as those described in Section 2.3. However, due to its broader scope, this work doesn't focus on only one specific technique, but on exploring and evaluating different techniques, and the combination of those techniques. We also apply these techniques to a larger number of microarray datasets.

In addition, we also address different data representation and dimensionality reduction techniques, combining FD and FS techniques, before classification.

Our aim is not solely the correct classification (as expressed by the error rate, FNR, and FPR), but also to find the subset of features that are more relevant for the classification task.

5 Experimental Results

This Chapter describes the experimental evaluation of the proposed approach on the eleven microarray datasets described in Table 1.1 and Table 3.1. In Section 5.1, we describe the baseline classification results without FD or FS techniques, using the SVM and DT classifiers (phases (a), (d), and (e) of the pipeline depicted in Figure 4.1). Section 5.2 addresses the use of FD techniques (phases (a), (b), (d), and (e) from the pipeline), whereas Section 5.3 reports the experimental results of FS techniques (phases (a), (c), (d), and (e) of the pipeline). Section 5.4 shows the results of the entire pipeline (all phases), using the best configurations found in the previous experiments. Section 5.5 presents experimental results that support the explainability of the classification, by identifying the best subsets of features for some dataset. In addition, Section 5.6, compares the results attained in this work with those described in Section 2.3. Finally, Section 5.7 concludes the chapter with a summary of these results.

For the tables that follow in this Chapter:

- $\overline{\text{A}}$ denotes the average value of a given metric for the eleven datasets in this work (where applicable), which allows us to identify the best overall techniques;
- σ indicates the standard deviation, which allows us to understand the stability/-consistency of each technique (the closest to zero the better);
- the criteria used to decide the best overall technique as well as the best suited one for each dataset, consists in comparing three estimated rates between many

applied techniques: namely the error rate, FNR, and FPR where applicable. We always analyse the error rate first because it's universally applicable across all datasets.

5.1 Baseline Classification Results

First, we evaluate the data classification phase of the pipeline. We check the performance of the selected classifiers, SVM and DT, to establish the baseline results.

Table 5.1 presents the estimates of the error rate, FNR, and FPR of the LOOCV procedure for the SVM classifier with the C parameter set to 1 (its default value) and four different values for the kernel parameter: linear, polynomial (poly), rbf, and sigmoid. The best overall result was achieved with kernel set to linear (with an Err=0.11, FNR=0.24, and FPR=0.16). However, this is not the case for each dataset individually, because in some datasets (such as Lymphoma, MLL, and SRBCT) we achieved the best result with different kernel values (in addition to linear). Furthermore, in the Colon dataset only the rbf kernel yielded a good result. We can also observe that, in this phase of the pipeline, we obtained an ideal error estimate for three datasets: Lymphoma (Err=0), Ovarian (Err=0, FNR=0, and FPR=0), and SRBCT (Err=0). In these cases, the remaining challenge consisted in applying other phases of the pipeline and retain these results.

Table 5.1: Test error rate (Err), FNR, and FPR of LOOCV for the SVM classifier (C=1), with original features (no transformations). Different kernel types are considered. Best results are presented in bold face.

	linear			poly			rbf			sigmoid		
Dataset	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR
Breast	**0.33**	**0.35**	**0.31**	0.46	0.98	0.00	0.47	1.00	0.00	0.48	1.00	0.02
CNS	**0.32**	**0.52**	**0.21**	0.35	1.00	0.00	0.35	1.00	0.00	0.35	1.00	0.00
Colon	0.19	0.32	0.12	0.23	0.59	0.02	**0.15**	**0.27**	**0.08**	0.39	1.00	0.05
Leukemia	**0.03**	–	–	0.06	–	–	0.08	–	–	0.35	–	–
Leukemia_3c	**0.07**	–	–	0.08	–	–	0.17	–	–	0.47	–	–
Leukemia_4c	**0.12**	–	–	0.14	–	–	0.28	–	–	0.47	–	–
Lung	**0.06**	**0.01**	**0.18**	0.09	0.01	0.24	0.10	0.01	0.24	0.22	0.00	1.00
Lymphoma	**0.00**	–	–	0.12	–	–	**0.00**	–	–	**0.00**	–	–
MLL	**0.04**	–	–	**0.04**	–	–	0.06	–	–	0.29	–	–
Ovarian	**0.00**	**0.00**	**0.00**	0.004	0.00	0.01	0.02	0.01	0.02	0.36	0.00	1.00
SRBCT	**0.00**	–	–	**0.00**	–	–	0.01	–	–	0.41	–	–
$\overline{A}$	**0.11**	**0.24**	**0.16**	0.14	0.52	0.05	0.15	0.46	0.07	0.34	0.60	0.41
σ	**0.12**	**0.20**	**0.10**	0.14	0.44	0.09	0.15	0.45	0.09	0.13	0.49	0.48

Table 5.2 presents the experimental results for the same setup as in Table 5.1, but we

apply phase (b) of the pipeline to normalize all feature values to the 0 to 1 range. We can observe that, setting the kernel to linear yielded the best results for all datasets. We also managed to retain the ideal error estimate (Err=0) for the Lymphoma, Ovarian, and SRBCT datasets. By comparing the results of Table 5.1 and Table 5.2, we conclude that SVM with a linear kernel is the best choice and that the use of normalized features leads to a slight improvement on the results ($\overline{\text{A}}$ having reduced from Err=0.11 to Err=0.09). On the other hand, taking into account each dataset, we can observe that on the CNS and Colon datasets, better results were obtained with original features rather than normalized ones.

Table 5.2: Test error rate (Err), FNR, and FPR of LOOCV for the SVM classifier (C=1), with normalized features. Different kernel types are considered. Best results are presented in bold face.

	linear			poly			rbf			sigmoid		
Dataset	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR
Breast	**0.31**	**0.30**	**0.31**	0.33	0.28	0.37	0.37	0.46	0.29	0.47	1.00	0.00
CNS	**0.33**	**0.62**	**0.18**	0.37	0.62	0.23	0.35	1.00	0.00	0.35	1.00	0.00
Colon	**0.18**	**0.27**	**0.12**	0.27	0.55	0.12	0.21	0.50	0.05	0.39	0.82	0.15
Leukemia	**0.01**	–	–	0.03	–	–	0.15	–	–	0.35	–	–
Leukemia_3c	**0.04**	–	–	0.06	–	–	0.26	–	–	0.47	–	–
Leukemia_4c	**0.07**	–	–	0.10	–	–	0.32	–	–	0.47	–	–
Lung	**0.05**	**0.01**	**0.12**	0.05	0.01	0.18	0.09	0.01	0.24	0.32	0.00	1.00
Lymphoma	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–	0.30	–	–
MLL	**0.03**	–	–	0.06	–	–	0.10	–	–	0.61	–	–
Ovarian	**0.00**	**0.00**	**0.00**	0.004	0.00	0.01	0.02	0.01	0.02	0.36	0.00	1.00
SRBCT	**0.00**	–	–	0.01	–	–	0.07	–	–	0.65	–	–
$\overline{\text{A}}$	**0.09**	**0.24**	**0.15**	0.12	0.29	0.18	0.18	0.40	0.12	0.43	0.56	0.43
σ	**0.12**	**0.23**	**0.10**	0.13	0.26	0.12	0.13	0.37	0.12	0.11	0.47	0.47

For each dataset with normalized features, we also assessed the impact of the C parameter in the SVM classifier (with kernel set to linear). We observed that, even though we used different values for C (specifically 1.5, 2, 10, or 100), we obtained the same results as those reported in Table 5.2 (see column named linear). These results are the same as those achieved with C set to 1. Therefore, we concluded that for these datasets (with this setup), the C parameter does not have an impact on the classification results.

Table 5.3 identifies the best SVM parameter configuration found for each dataset during the data classification phase. In this Table, "O" stands for original features while "N" means normalized features.

Table 5.4 shows the results of the DT classifier with the criterion parameter set to gini

Table 5.3: Summary of the best results and respective configurations, for each dataset, obtained during the data classification phase, with the SVM classifier - a, b, and c denote combinations of techniques.

		Configurations				
Dataset	Processing	C	kernel	Err	FNR	FPR
Breast	N	1	linear	0.31	0.30	0.31
CNS	O	1	linear	0.32	0.52	0.21
Colon	O	1	rbf	0.15	0.27	0.08
Leukemia	N	1	linear	0.01	–	–
Leukemia_3c	N	1	linear	0.04	–	–
Leukemia_4c	N	1	linear	0.07	–	–
Lung	N	1	linear	0.05	0.01	0.12
Lymphoma	O^a, N^b	1	linear, poly^b, rbf, sigmoid^a	0.00	–	–
MLL	N	1	linear	0.03	–	–
Ovarian	O, N	1	linear	0.00	0.00	0.00
SRBCT	O, N^c	1	linear^c, poly	0.00	–	–
$\overline{A}$	–	–	–	0.09	0.22	0.14
σ	–	–	–	0.11	0.20	0.11

and max_depth (the learned tree maximum allowed depth) set to None, both their default values. Table 5.5 shows a similar evaluation as in Table 5.4, but now considering entropy as a criterion. With these experiments, we have found that using entropy as a criterion to build the tree yielded the best results for all datasets. Even though for four datasets (namely Leukemia, Lymphoma, MLL, and Ovarian) we attained the same results using gini, we conclude that setting the criterion to entropy is the best choice, since the best overall result was also achieved using the same value for this parameter. In addition, we can also observe that, since the random_state parameter controls the randomness of the tree and each dataset has its own characteristics, the best choice for each one may vary. Nevertheless, setting the random_state to 42 yielded the best overall results in both Table 5.4 and Table 5.5. Moreover, we obtained an ideal error estimate only for the Lymphoma dataset, which was acquired with random_state set to 42, as we can see in these tables. We also assessed the impact of feature normalization on the same setup as in Table 5.4 and Table 5.5. However, the results obtained were the same as those in these tables. Therefore, for the sake of consistency in the pipeline, we decided to keep normalizing the features since it doesn't decrease the quality of the results.

Table 5.6 assesses the impact of the learned tree maximum allowed depth (max_depth) for these datasets. As we can see, setting the max_depth to 5, 7 or 10 yielded the

Table 5.4: Test error rate (Err), FNR, and FPR of LOOCV for the DT classifier (criterion=gini and max_depth=None), with original features. Different values for the random_state parameter are evaluated. Best results are presented in bold face.

	5			13			29			42		
Dataset	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR
Breast	0.36	0.37	0.35	0.43	0.46	0.41	0.35	0.35	0.35	**0.35**	**0.30**	**0.39**
CNS	**0.27**	**0.48**	**0.15**	0.43	0.48	0.41	0.32	0.43	0.26	0.33	0.43	0.28
Colon	0.19	0.36	0.10	0.19	0.32	0.12	0.21	0.41	0.10	**0.16**	**0.23**	**0.12**
Leukemia	**0.19**	–	–	0.21	–	–	0.22	–	–	0.28	–	–
Leukemia_3c	**0.15**	–	–	0.22	–	–	0.18	–	–	0.18	–	–
Leukemia_4c	0.26	–	–	0.17	–	–	**0.14**	–	–	0.22	–	–
Lung	0.12	0.02	0.35	0.10	0.02	0.12	0.11	0.02	0.35	**0.07**	**0.03**	**0.18**
Lymphoma	0.06	–	–	0.09	–	–	0.05	–	–	**0.00**	–	–
MLL	0.12	–	–	**0.07**	–	–	0.11	–	–	0.08	–	–
Ovarian	0.03	0.02	0.04	0.02	0.01	0.04	**0.02**	**0.01**	**0.03**	0.03	0.01	0.07
SRBCT	**0.18**	–	–	0.23	–	–	0.29	–	–	0.20	–	–
$\overline{A}$	0.18	0.25	0.20	0.20	0.26	0.22	0.18	0.24	0.22	**0.17**	**0.20**	**0.21**
σ	0.09	0.19	0.13	0.13	0.21	0.16	0.10	0.19	0.13	**0.11**	**0.16**	**0.11**

Table 5.5: Test error rate (Err), FNR, and FPR of LOOCV for the DT classifier (criterion=entropy and max_depth=None), with original features. Different values for the random_state parameter are evaluated. Best results are presented in bold face.

	5			13			29			42		
Dataset	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR
Breast	0.34	0.30	0.37	**0.31**	**0.35**	**0.27**	0.39	0.33	0.45	0.33	0.30	0.35
CNS	0.32	0.33	0.31	0.25	0.38	0.18	0.32	0.43	0.26	**0.25**	**0.33**	**0.21**
Colon	**0.13**	**0.23**	**0.08**	0.21	0.32	0.15	0.23	0.36	0.15	0.19	0.23	0.18
Leukemia	0.24	–	–	**0.19**	–	–	0.21	–	–	0.26	–	–
Leukemia_3c	**0.14**	–	–	0.19	–	–	0.18	–	–	0.17	–	–
Leukemia_4c	0.19	–	–	0.15	–	–	**0.10**	–	–	0.15	–	–
Lung	**0.06**	**0.01**	**0.18**	0.09	0.01	0.12	0.07	0.01	0.29	0.07	0.01	0.12
Lymphoma	0.06	–	–	0.09	–	–	0.05	–	–	**0.00**	–	–
MLL	0.12	–	–	**0.07**	–	–	0.11	–	–	0.08	–	–
Ovarian	0.03	0.02	0.04	0.02	0.01	0.04	**0.02**	**0.01**	**0.03**	0.03	0.01	0.07
SRBCT	**0.16**	–	–	0.19	–	–	0.27	–	–	0.17	–	–
$\overline{A}$	0.16	0.18	0.20	0.16	0.21	0.15	0.18	0.23	0.24	**0.15**	**0.18**	**0.19**
σ	0.10	0.14	0.13	0.08	0.17	0.08	0.11	0.18	0.14	**0.10**	**0.14**	**0.10**

same results, being these also the same as those attained with max_depth set to None (see column entitled 42 in Table 5.5). These results show that, the trees generated by the configurations described in Table 5.6 are equivalent. That's why, a max_depth beyond 5 didn't yield improvements. Thus, in general, setting it to 5 is the best way to decrease complexity in the learning algorithm and computational cost. However, as stated before, the best choice for each dataset may vary. For instance, the CNS dataset shows a good improvement when used with normalized features and max_depth set to 2. In addition, for the Lymphoma dataset we managed to keep a 0 error rate (result observed for all max_depth values explored in this phase).

Table 5.6: Test error rate (Err), FNR, and FPR of LOOCV for the DT classifier (criterion=entropy and random_state=42), with normalized features. Different values for the max_depth parameter are evaluated. Best results are presented in bold face.

	2			5			7			10		
Dataset	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR
Breast	0.40	0.35	0.45	**0.33**	**0.30**	**0.35**	**0.33**	**0.30**	**0.35**	**0.33**	**0.30**	**0.35**
CNS	**0.18**	**0.48**	**0.03**	0.25	0.33	0.21	0.25	0.33	0.21	0.25	0.33	0.21
Colon	**0.18**	**0.36**	**0.08**	0.19	0.23	0.18	0.19	0.23	0.18	0.19	0.23	0.18
Leukemia	**0.26**	–	–	**0.26**	–	–	**0.26**	–	–	**0.26**	–	–
Leukemia_3c	**0.15**	–	–	0.17	–	–	0.17	–	–	0.17	–	–
Leukemia_4c	**0.11**	–	–	0.15	–	–	0.15	–	–	0.15	–	–
Lung	0.13	0.01	0.06	**0.07**	**0.01**	**0.12**	**0.07**	**0.01**	**0.12**	**0.07**	**0.01**	**0.12**
Lymphoma	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–
MLL	**0.08**	–	–	**0.08**	–	–	**0.08**	–	–	**0.08**	–	–
Ovarian	**0.03**	**0.01**	**0.07**	**0.03**	**0.01**	**0.07**	**0.03**	**0.01**	**0.07**	**0.03**	**0.01**	**0.07**
SRBCT	0.27	–	–	**0.17**	–	–	**0.17**	–	–	**0.17**	–	–
$\overline{A}$	0.16	0.24	0.14	**0.15**	**0.18**	**0.19**	**0.15**	**0.18**	**0.19**	**0.15**	**0.18**	**0.19**
σ	0.11	0.19	0.16	**0.10**	**0.14**	**0.10**	**0.10**	**0.14**	**0.10**	**0.10**	**0.14**	**0.10**

Table 5.7 identifies the best DT parameter configuration found for each dataset during the data classification phase.

Table 5.7: Summary of the best results and respective configurations, for each dataset, obtained during the data classification phase with the DT classifier.

		Configurations					
Dataset	Processing	criterion	max_depth	random_state	Err	FNR	FPR
Breast	O, N	entropy	None	13	0.31	0.35	0.27
CNS	N	entropy	2	42	0.18	0.48	0.03
Colon	O, N	entropy	None	5	0.13	0.23	0.08
Leukemia	O, N	gini[a], entropy[b]	None	5[a], 13[b]	0.19	–	–
Leukemia_3c	O, N	entropy	None	5	0.14	–	–
Leukemia_4c	O, N	entropy	None	29	0.10	–	–
Lung	O, N	entropy	None	5	0.06	0.01	0.18
Lymphoma	O[c], N	gini[d], entropy	None[c,d], 2, 5, 7, 10	42	0.00	–	–
MLL	O, N	gini, entropy	None	13	0.07	–	–
Ovarian	O, N	gini, entropy	None	29	0.02	0.01	0.03
SRBCT	O, N	entropy	None	5	0.16	–	–
$\overline{A}$	–	–	–	–	0.12	0.22	0.12
σ	–	–	–	–	0.08	0.19	0.09

For the next pipeline phases (FD in Section 5.2 and FS in Section 5.3), we decided to normalize all feature values for all datasets. Since normalized features lead to an improvement on the results for all but two datasets (CNS and Colon for the SVM classifier, as we can see in Table 5.3), for the sake of consistency in the pipeline, we proceeded as such. In addition, for the sake of simplicity, the configurations considered for the SVM and DT classifiers in the next phases of the pipeline, are those reported with the best overall results in this data classification phase. As follows:

- SVM, with kernel set to linear and C to 1;
- DT, with criterion set to entropy, max_depth to 5, and random_state to 42.

5.2 Feature Discretization Assessment

We now address the use of FD on the pipeline. We assess the performance of the EFB and MDLP discretization techniques, for all experiments in this phase.

Table 5.8 reports the results of the SVM classifier on data discretized by EFB, with different number of discretization bins (n_bins). Analyzing these results for all datasets, we conclude that EFB discretization yields a small improvement as the number of bins increases (lower standard deviation in all datasets). However, once we find an optimal value, this improvement tends to stop and the performance decreases. In this case, we consider setting n_bins to 6 the best choice, since it yielded the best results for all but two datasets (CNS and Leukemia_3c) and it's also the value which we obtained the best overall result. In spite of that, we can also observe that, for some datasets (namely Leukemia, Lymphoma, and SRBCT) the best outcome for each one was obtained regardless the number of bins. We also managed to keep a 0 error rate for the Lymphoma, Ovarian, and SRBCT datasets in this phase with the SVM classifier.

Table 5.9 shows a similar experiment as in Table 5.8, but now considering the DT classifier. As we can observe, the DT classifier results do not seem to benefit from the use of discretization as a data representation technique. However, for some datasets (namely CNS and Leukemia), we have an improvement on the results as comparable to Table 5.6.

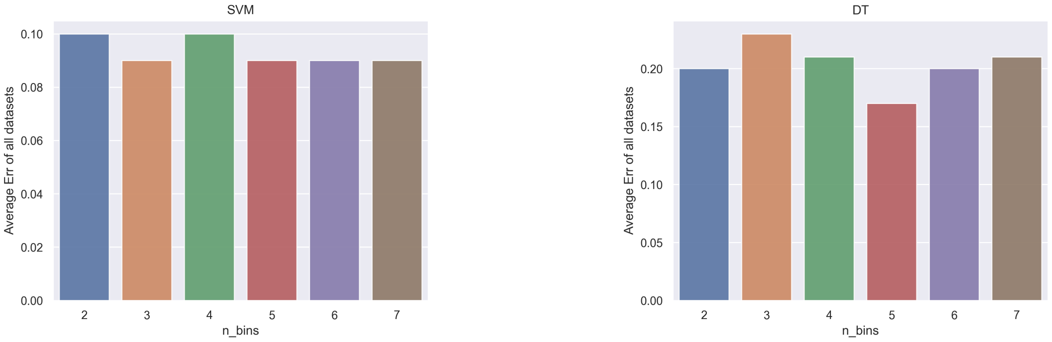
Figure 5.1 shows the relation between the average error rate of all datasets and the number of discretization bins, for the SVM and DT classifiers. We can observe the increasing and decreasing effect regarding the improvements on the classifiers performance. For the SVM classifier, the optimal n_bins is 6 (lower error and lower standard deviation). On the DT classifier, the optimal value is n_bins = 5.

Table 5.8: Test error rate (Err), FNR, and FPR of LOOCV for the SVM classifier (C=1 and kernel=linear) with EFB discretization. Different values for the n_bins parameter were evaluated. Best results are presented in bold face.

	2			3			4			5			6			7		
Dataset	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR
Breast	0.32	0.30	0.33	0.33	0.33	0.33	0.32	0.33	0.31	0.32	0.33	0.31	**0.30**	**0.30**	**0.29**	0.31	0.33	0.29
CNS	0.35	0.71	0.15	**0.30**	**0.62**	**0.13**	0.38	0.71	0.21	0.32	0.62	0.15	0.32	0.62	0.15	0.37	0.67	0.21
Colon	0.18	0.27	0.12	0.18	0.27	0.12	0.16	0.27	0.10	**0.15**	**0.23**	**0.10**	**0.15**	**0.23**	**0.10**	0.16	0.27	0.10
Leukemia	**0.01**	–	–	**0.01**	–	–	**0.01**	–	–	**0.01**	–	–	**0.01**	–	–	**0.01**	–	–
Leukemia_3c	**0.03**	–	–	**0.03**	–	–	**0.03**	–	–	**0.03**	–	–	0.04	–	–	0.04	–	–
Leukemia_4c	0.08	–	–	**0.07**	–	–	**0.07**	–	–	**0.07**	–	–	**0.07**	–	–	**0.07**	–	–
Lung	0.05	0.01	0.18	0.05	0.01	0.18	0.05	0.01	0.18	**0.04**	**0.01**	**0.18**	**0.04**	**0.01**	**0.18**	**0.04**	**0.01**	**0.18**
Lymphoma	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–
MLL	0.04	–	–	**0.03**	–	–	**0.03**	–	–	**0.03**	–	–	**0.03**	–	–	**0.03**	–	–
Ovarian	0.004	0.00	0.01	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
SRBCT	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–
$\overline{A}$	0.10	0.26	0.16	0.09	0.25	0.15	0.10	0.26	0.16	0.09	0.24	0.15	**0.09**	**0.23**	**0.14**	0.09	0.26	0.16
σ	0.12	0.26	0.10	0.12	0.23	0.11	0.13	0.26	0.10	0.12	0.23	0.10	**0.11**	**0.23**	**0.10**	0.12	0.25	0.10

Table 5.9: Test error rate (Err), FNR, and FPR of LOOCV for the DT classifier (criterion=entropy, max_depth=5, and random_state=42) with EFB discretization. Different values for the n_bins parameter were evaluated. Best results are presented in bold face.

	2			3			4			5			6			7		
Dataset	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR
Breast	**0.30**	**0.35**	**0.25**	**0.30**	**0.35**	**0.25**	0.46	0.50	0.43	0.32	0.37	0.27	0.47	0.59	0.37	0.48	0.61	0.37
CNS	0.42	0.67	0.28	0.65	0.81	0.56	0.43	0.57	0.36	**0.18**	**0.33**	**0.10**	0.37	0.57	0.26	0.50	0.71	0.38
Colon	0.26	0.32	0.22	0.34	0.55	0.22	0.29	0.55	0.15	**0.16**	**0.18**	**0.15**	0.34	0.45	0.28	0.24	0.41	0.15
Leukemia	**0.01**	–	–	0.11	–	–	0.12	–	–	0.19	–	–	0.08	–	–	0.14	–	–
Leukemia_3c	0.19	–	–	0.22	–	–	0.12	–	–	0.14	–	–	**0.10**	–	–	0.19	–	–
Leukemia_4c	0.21	–	–	0.28	–	–	**0.10**	–	–	0.26	–	–	0.21	–	–	0.17	–	–
Lung	0.27	0.07	0.47	0.17	0.01	0.12	0.16	0.02	0.12	0.15	0.03	0.41	0.18	0.05	0.35	**0.15**	**0.03**	**0.29**
Lymphoma	**0.06**	–	–	**0.06**	–	–	0.11	–	–	0.09	–	–	0.09	–	–	0.11	–	–
MLL	0.19	–	–	0.17	–	–	0.25	–	–	0.15	–	–	**0.08**	–	–	0.18	–	–
Ovarian	0.06	0.07	0.03	**0.02**	**0.01**	**0.03**	0.04	0.04	0.03	0.03	0.01	0.05	0.02	0.01	0.04	0.03	0.01	0.07
SRBCT	0.18	–	–	0.25	–	–	0.22	–	–	**0.16**	–	–	0.23	–	–	0.17	–	–
$\overline{A}$	0.20	0.30	0.25	0.23	0.35	0.24	0.21	0.34	0.22	**0.17**	**0.18**	**0.20**	0.20	0.33	0.26	0.21	0.35	0.25
σ	0.11	0.22	0.14	0.16	0.31	0.18	0.13	0.25	0.15	**0.07**	**0.15**	**0.13**	0.14	0.25	0.12	0.14	0.29	0.12



(a) (b)

Figure 5.1: Analysis of the error rate (Err) and number of discretization bins (n_bins) for the SVM and DT classifiers.

Table 5.10 shows a summary of the results of the best configurations of EFB discretization and SVM/DT classifiers. For each dataset, we select the best configuration found in our experiments. The * symbol denotes an improvement over the baseline classification results of Table 5.3 and 5.7. As we can see, for the Breast and CNS datasets, the best results were achieved with a single combination: (Classifier= SVM and n_bins=6) for the former and (Classifier=DT and n_bins=5) for the latter. For the remaining datasets we were able to achieve the same best results with more than one combination of these configurations. For instance, for the Colon dataset, the configurations (Classifier=SVM and n_bins=5) and (Classifier=SVM and n_bins=6) yielded the same best results. For the Leukemia dataset, the best results were achieved with both the SVM and DT classifiers and different numbers of bins: (SVM, 2), (SVM, 3), (SVM, 4), (SVM, 5), (SVM, 6), (SVM, 7), and (DT, 2). For the remaining datasets, we observe a similar phenomenon. The best result was achieved with the SVM classifier, and with a wide variety on n_bins.

Table 5.10: Summary of the best results and respective configurations for each dataset, obtained during the data representation phase with the EFB discretizer.

	Configurations				
Dataset	Classifier	n_bins	Err	FNR	FPR
Breast	SVM	6	0.30*	0.30	0.29
CNS	DT	5	0.18*	0.33	0.10
Colon	SVM	5, 6	0.15*	0.23	0.10
Leukemia	SVM, DT^a	2^a, 3, 4, 5, 6, 7	0.01	–	–
Leukemia_3c	SVM	2, 3, 4, 5	0.03*	–	–
Leukemia_4c	SVM	3, 4, 5, 6, 7	0.07	–	–
Lung	SVM	5, 6, 7	0.04*	0.01	0.18
Lymphoma	SVM	2, 3, 4, 5, 6, 7	0.00	–	–
MLL	SVM	3, 4, 5, 6, 7	0.03	–	–
Ovarian	SVM	3, 4, 5, 6, 7	0.00	0.00	0.00
SRBCT	SVM	2, 3, 4, 5, 6, 7	0.00	–	–
$\overline{A}$	–	–	0.07	0.17	0.13
σ	–	–	0.09	0.14	0.10

Table 5.11 reports the results of the SVM and DT classifiers on data discretized by the MDLP technique. The * symbol denotes an improvement over the baseline classification results of Table 5.3 and 5.7, while ** denotes an improvement over the EFB results of Table 5.10. We can observe that, like for the EFB discretization, MDLP also generates better results for the SVM classifier, whereas the DT still does not seem to benefit from the use of discretization. We can also perceive that, regarding the baseline results, the

DT yields a slight improvement for the Leukemia_3c dataset (Err decreased by 0.01). In addition, we also obtained an improvement over the EFB discretization for the Colon dataset, on the FNR metric. Although the FPR was increased by 0.05, the decreasing of 0.09 on the FNR metric is a reasonable progress.

Table 5.11: Test error rate (Err), FNR, and FPR of LOOCV for the SVM (C=1 and kernel=linear) and DT (criterion=entropy, max_depth=5, and random_state=42) classifiers with MDLP. Best results are presented in bold face.

	SVM			DT		
Dataset	Err	FNR	FPR	Err	FNR	FPR
Breast	**0.37**	**0.41**	**0.33**	0.38	0.41	0.35
CNS	0.45	0.76	0.28	**0.30**	**0.43**	**0.23**
Colon	0.18	0.32	0.10	**0.15****	**0.14**	**0.15**
Leukemia	**0.01**	–	–	0.17	–	–
Leukemia_3c	**0.03***	–	–	0.12	–	–
Leukemia_4c	**0.07**	–	–	0.12	–	–
Lung	**0.05**	**0.01**	**0.24**	0.06	0.01	0.12
Lymphoma	**0.00**	–	–	0.09	–	–
MLL	**0.03**	–	–	0.08	–	–
Ovarian	**0.01**	**0.00**	**0.02**	0.02	0.01	0.03
SRBCT	**0.01**	–	–	0.17	–	–
$\overline{A}$	**0.11**	**0.30**	**0.19**	0.15	0.20	0.18
σ	**0.15**	**0.28**	**0.12**	0.10	0.19	0.11

5.3 Feature Selection Assessment

We now address the use of FS on the pipeline (without discretization). We consider the LS, Spectral (SPEC), FiR, and RRFS techniques.

Table 5.12 shows the experimental results of these methods for the SVM classifier. Table 5.13 reports the corresponding results of Table 5.12, but now considering the DT classifier. The * symbol denotes an improvement over the baseline classification results of Table 5.3 and Table 5.7. We can observe that, for the SVM classifier, using the RRFS(MM) and FiR methods we obtained an improvement on the Breast dataset (FNR reduced from 0.30 to 0.28). FiR also yielded an improvement on the Lung dataset (Err reduced from 0.05 to 0.04). In addition, RRFS(FiR) decreases the 0.04 error rate value to 0.03 for Leukemia_3c. We can also observe that, for six datasets (namely Leukemia, Leukemia_4c, Lymphoma, MLL, Ovarian, and SRBCT) we managed to keep the same values as seen in the baseline results (Table 5.3). In these cases, even though there's no improvement on the results, we consider applying FS to be beneficial, because we managed to reduce the number of features and still keep a good result. Regarding the DT classifier, we can observe that only one improvement was achieved, which was attained with the RRFS(FiR) method on the Breast dataset. In addition, only on the Leukemia_3c dataset we keep the same result as seen in the baseline (Table 5.7). We conclude that applying these FS techniques with the SVM classifier yielded better results than with DT.

Table 5.12: Test error rate (Err), FNR, and FPR of LOOCV for the SVM classifier (C=1 and kernel=linear) with LS, SPEC, FiR, and RRFS (with MM and FiR relevance and *ms*=0.7). Best results are presented in bold face.

	Unsupervised									Supervised					
	LS			SPEC			RRFS (MM)			FiR			RRFS (FiR)		
Dataset	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR
Breast	0.33	0.35	0.31	0.32	0.30	0.33	**0.31***	**0.28**	**0.33**	**0.31***	**0.28**	**0.33**	**0.31**	**0.28**	**0.33**
CNS	0.35	0.52	0.26	0.33	0.62	0.18	**0.27***	**0.48**	**0.15**	0.30	0.57	0.15	0.33	0.67	0.15
Colon	**0.16**	**0.27**	**0.10**	0.19	0.32	0.12	0.21	0.36	0.12	0.19	0.32	0.12	0.18	0.27	0.12
Leukemia	**0.01**	–	–	**0.01**	–	–	**0.01**	–	–	**0.01**	–	–	**0.01**	–	–
Leukemia_3c	0.04	–	–	0.06	–	–	0.04	–	–	0.04	–	–	**0.03***	–	–
Leukemia_4c	0.08	–	–	0.10	–	–	**0.07**	–	–	**0.07**	–	–	**0.07**	–	–
Lung	0.05	0.01	0.12	0.05	0.01	0.12	0.05	0.01	0.12	**0.04***	**0.01**	**0.12**	0.05	0.01	0.18
Lymphoma	**0.00**	–	–	**0.00**	–	–	0.03	–	–	**0.00**	–	–	0.02	–	–
MLL	0.04	–	–	0.06	–	–	**0.03**	–	–	**0.03**	–	–	0.04	–	–
Ovarian	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.004	0.00	0.01	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
SRBCT	0.02	–	–	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–	**0.00**	–	–
$\overline{A}$	0.10	0.23	0.16	0.10	0.25	0.15	**0.09**	**0.23**	**0.15**	0.09	0.24	0.14	0.09	0.25	0.16
σ	0.12	0.20	0.11	0.12	0.23	0.11	**0.11**	**0.19**	**0.10**	0.11	0.21	0.11	0.12	0.24	0.11

Table 5.13: Test error rate (Err), FNR, and FPR of LOOCV for the DT classifier (criterion=entropy, max_depth=5, and random_state=42) with LS, SPEC, FiR, and RRFS (with MM and FiR relevance and *ms*=0.7). Best results are presented in bold face.

	Unsupervised									Supervised					
	LS			SPEC			RRFS (MM)			FiR			RRFS (FiR)		
Dataset	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR	Err	FNR	FPR
Breast	0.43	0.50	0.37	0.29	0.30	0.27	0.41	0.50	0.33	0.37	0.33	0.41	**0.26***	**0.33**	**0.20**
CNS	0.38	0.67	0.23	**0.22**	**0.38**	**0.13**	0.28	0.43	0.21	0.32	0.38	0.28	0.32	0.38	0.28
Colon	0.32	0.50	0.22	0.34	0.55	0.22	**0.19**	**0.32**	**0.12**	0.26	0.27	0.25	0.21	0.27	0.18
Leukemia	**0.14**	–	–	0.21	–	–	0.21	–	–	0.25	–	–	0.15	–	–
Leukemia_3c	**0.07***	–	–	0.14	–	–	0.14	–	–	0.17	–	–	0.15	–	–
Leukemia_4c	0.12	–	–	0.25	–	–	0.14	–	–	**0.11**	–	–	0.22	–	–
Lung	0.15	0.03	0.41	0.16	0.01	0.18	0.09	0.01	0.12	0.09	0.01	0.18	**0.08**	**0.01**	**0.18**
Lymphoma	0.23	–	–	0.18	–	–	**0.08**	–	–	**0.08**	–	–	0.12	–	–
MLL	0.26	–	–	0.26	–	–	0.14	–	–	**0.07**	–	–	0.15	–	–
Ovarian	0.04	0.02	0.08	0.04	0.04	0.05	0.02	0.02	0.02	0.03	0.02	0.04	**0.02**	**0.01**	**0.04**
SRBCT	0.23	–	–	0.20	–	–	**0.17**	–	–	0.23	–	–	**0.17**	–	–
$\overline{A}$	0.22	0.34	0.26	0.21	0.26	0.17	0.17	0.26	0.16	0.18	0.20	0.23	**0.17**	**0.20**	**0.18**
σ	0.12	0.27	0.12	0.08	0.21	0.08	0.10	0.20	0.10	0.11	0.16	0.12	**0.08**	**0.16**	**0.08**

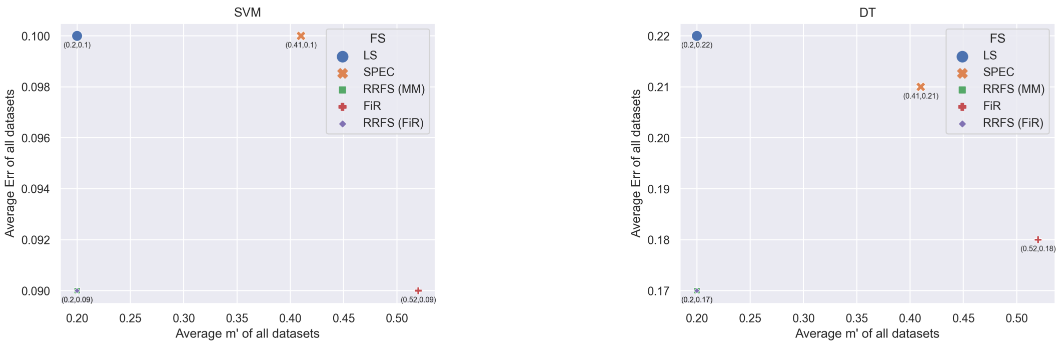
Table 5.14 shows the reduction of dimensionality in terms of the percentage of the selected features (m'), in which we have considerable reductions on the dimensionality.

Table 5.14: Percentage of selected features (m') for LS, SPEC, FiR, and RRFS (with MM and FiR relevance and ms=0.7). Best results are presented in bold face.

	Unsupervised			Supervised	
Dataset	LS	SPEC	RRFS (MM)	FiR	RRFS (FiR)
Breast	0.17	0.23	**0.14**	0.62	0.15
CNS	**0.12**	0.34	0.17	0.48	0.17
Colon	0.55	0.42	0.19	0.43	**0.18**
Leukemia	**0.13**	0.47	0.18	0.43	0.18
Leukemia_3c	**0.13**	0.47	0.18	0.53	0.18
Leukemia_4c	**0.13**	0.47	0.18	0.59	0.17
Lung	**0.13**	0.38	0.15	0.67	0.14
Lymphoma	0.22	0.40	**0.02**	0.52	**0.02**
MLL	**0.13**	0.31	0.23	0.54	0.24
Ovarian	0.12	0.54	**0.04**	0.36	**0.04**
SRBCT	**0.36**	0.49	0.68	0.60	0.68
$\overline{A}$	**0.20**	0.41	0.20	0.52	0.20
σ	**0.13**	0.09	0.16	0.09	0.16

As we can see, the LS method yielded the best overall m' (lower standard deviation than RRFS with FiR as a relevance metric), but not every technique that presented a smaller m' should be considered the best one. For instance, on the Ovarian dataset with the RRFS method, we get a reduction to 4% of the original dimensionality, which implies that the number of reduced features is about 606, from the original set of 15154 features. However, bearing in mind the SVM classification error results presented in Table 5.12 for this dataset, we conclude that the RRFS with a FiR relevance metric is a better choice than the RRFS with a MM relevance: Err=0.004 and m'=0.04 for the former, and Err=0.00 and m'=0.04 for the latter. Following the same procedure - comparison between Table 5.14, Table 5.12, and Table 5.13 - we also conclude that the RRFS(FiR) method (along with SVM) is also the best choice for the Breast, Leukemia_3c, and Leukemia_4c datasets. In addition, we consider the LS method (along with SVM) the best choice for the Colon, Leukemia, and Lymphoma datasets. In these cases, regarding the Colon dataset, we can see that the RRFS(FiR) technique presented a better m' than LS but the classification error is worse. A similar outcome was observed for the Lymphoma dataset, in which we keep 2% of the original features with the RRFS(MM) and 22% with the LS technique (80 features remaining for the former and about 885 for the latter, from the original set of 4026) but the SVM classification error rate is worse with RRFS(MM) as compared to the LS method. As for the CNS dataset, we consider the SPEC technique to be the best choice (along with DT), regardless of the smaller m' attained with the LS method, because SPEC yielded better results with regards to the classification error. For the same reason, we consider the SPEC method (along with SVM) the best choice for the SRBCT dataset. Finally, for the MLL dataset we consider the RRFS(MM) technique (along with SVM) the most appropriate one, since it yielded the best classification error and m'.

Figure 5.2 shows the graphical representation of the error rate (for the SVM and DT classifiers) and the corresponding percentage of the selected features (m'), for the methods of FS considered in this work. We report the average error rate (from Table 5.12 and Table 5.13) and the average number of selected features (from Table 5.14).



(a)

(b)

Figure 5.2: Average error rate (Err) and average percentage of selected features (m') for LS, SPEC, FiR, and RRFS (with MM and FiR; ms=0.7).

5.4 The Machine Learning Pipeline

We now address the joint effect of all the pipeline phases, depicted in Figure 4.1.

Table 5.15 presents the best suited configurations for each phase and each dataset. Bearing in mind the previous analysis, the criteria we considered to select these configurations are as follows:

1. When confronted with the same results while applying the SVM and DT classifiers, we choose SVM over DT, because, according to the literature, the former typically yields better results when compared to the latter. Otherwise, we opted for the one that yielded the best result.

2. When confronted with the same results while applying EFB discretization with a wide variety on n_bins, we considered the smaller one, thereby increasing the results interpretability by simplifying the information (range of values) the feature in question conveys. In addition, when MDLP and EFB produce the same results, we opted for the EFB technique, because it's faster and is known to have produced adequate results in the literature.

3. When confronted with the same results while applying FS techniques, we choose the one with the smaller m'. Otherwise, we selected unsupervised over supervised, since from a computational point of view, it's less costly.

Table 5.15: Pipeline's configuration for each dataset. More details about this and other possible combinations can be found in Appendix B.

	Configurations		
Dataset	Discretization	Selection	Classification
Breast	EFB (n_bins=6)	RRFS (FiR, *ms*=0.7)	SVM (C=1, kernel=linear)
CNS	EFB (n_bins=5)	SPEC	DT (criterion=entropy, max_depth=5, random_state=42)
Colon	MDLP	LS	DT (criterion=entropy, max_depth=None, random_state=5)
Leukemia	EFB (n_bins=2)	LS	SVM (C=1, kernel=linear)
Leukemia_3c	EFB (n_bins=2)	RRFS (FiR, *ms*=0.7)	SVM (C=1, kernel=linear)
Leukemia_4c	EFB (n_bins=3)	RRFS (FiR, *ms*=0.7)	SVM (C=1, kernel=linear)
Lung	EFB (n_bins=5)	FiR	SVM (C=1, kernel=linear)
Lymphoma	EFB (n_bins=2)	LS	SVM (C=1, kernel=linear)
MLL	EFB (n_bins=3)	RRFS (MM, *ms*=0.7)	SVM (C=1, kernel=linear)
Ovarian	EFB (n_bins=3)	RRFS (FiR, *ms*=0.7)	SVM (C=1, kernel=linear)
SRBCT	EFB (n_bins=2)	SPEC	SVM (C=1, kernel=linear)

Table 5.16 shows the results obtained with the pipeline configurations presented in Table 5.15. For insight, we reported the comparison between the pipeline built with all phases enabled and the pipeline without the discretization phase. As we can see across all datasets in this work, combining FD and FS techniques didn't yield as good results as those seen for the pipeline with FD disabled. For this reason, we considered two other discretization techniques (U-LBG1 and R-LBG) to verify if the previous conclusion holds for different discretization techniques. Table 5.17 reports the results attained with the U-LBG1 and R-LBG discretization techniques. By comparing Table 5.16 and Table 5.17, we conclude that there's no relevant improvement. Therefore, based on these results, we can observe that the discretization seems to have a notable negative impact when jointly used with the FS process, increasing the classification error.

Based on the assessment presented above, we conclude that the best combination of techniques for each dataset are as follows:

- Breast - EFB (n_bins=6) and SVM (C=1 and kernel=linear);

- CNS - EFB (n_bins=5) and DT (criterion=entropy, max_depth=5, and random_state=42);
- Colon - DT (criterion=entropy, max_depth=None, and random_state=5);
- Leukemia - LS and SVM (C=1 and kernel=linear);
- Leukemia_3c - RRFS(FiR) and SVM (C=1 and kernel=linear);
- Leukemia_4c - RRFS(FiR) and SVM (C=1 and kernel=linear);
- Lung - FiR and SVM (C=1 and kernel=linear);
- Lymphoma - LS and SVM (C=1 and kernel=linear);
- MLL - RRFS(MM) and SVM (C=1 and kernel=linear);
- Ovarian - RRFS(FiR) and SVM (C=1 and kernel=linear);
- SRBCT - SPEC and SVM (C=1 and kernel=linear).

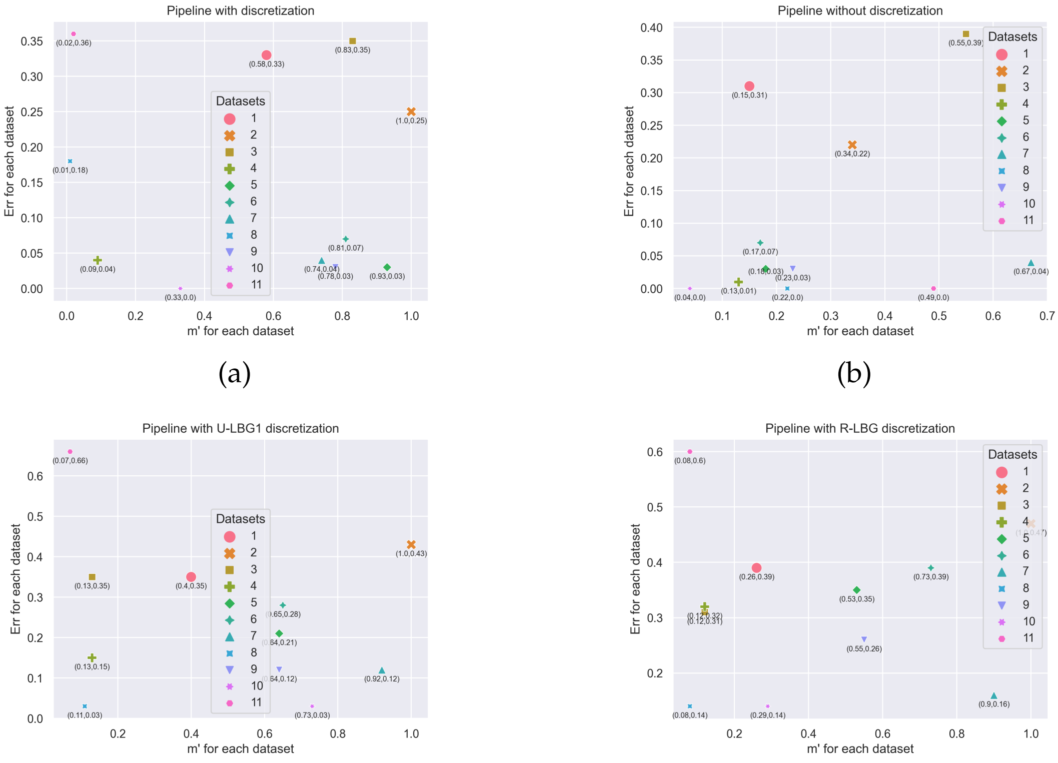
Table 5.16: Test error rate (Err), FNR, and FPR of LOOCV for the pipeline with discretization (EFB or MDLP) and without discretization. Best results are presented in bold face.

	With Discretization				No Discretization			
Dataset	Err	FNR	FPR	m'	Err	FNR	FPR	m'
Breast	0.33	0.35	0.31	0.58	**0.31**	0.28	0.33	**0.15**
CNS	0.25	0.43	0.15	1.00	**0.22**	0.38	0.13	**0.34**
Colon	**0.35**	1.00	0.00	0.83	0.39	0.59	0.28	**0.55**
Leukemia	0.04	–	–	**0.09**	**0.01**	–	–	0.13
Leukemia_3c	**0.03**	–	–	0.93	**0.03**	–	–	**0.18**
Leukemia_4c	**0.07**	–	–	0.81	**0.07**	–	–	**0.17**
Lung	**0.04**	0.01	0.18	0.74	**0.04**	0.01	0.12	**0.67**
Lymphoma	0.18	–	–	**0.01**	**0.00**	–	–	0.22
MLL	**0.03**	–	–	0.78	**0.03**	–	–	**0.23**
Ovarian	**0.00**	0.00	0.00	0.33	**0.00**	0.00	0.00	**0.04**
SRBCT	0.36	–	–	**0.02**	**0.00**	–	–	0.49
$\overline{A}$	0.15	0.36	0.13	0.56	**0.10**	0.25	0.17	**0.29**
σ	0.14	0.37	0.12	0.36	**0.13**	0.23	0.12	**0.19**

In Figure 5.3, we show a scatter plot of the error rate and the corresponding m', for all the datasets, as stated in Table 5.16 and Table 5.17.

Table 5.17: Test error rate (Err), FNR, and FPR of LOOCV for the pipeline with U-LBG1(q=4, Δ=0.05) and R-LBG(q=4, Δ=0.1) discretization techniques. Best results are presented in bold face.

	U-LBG1				R-LBG			
Dataset	Err	FNR	FPR	m'	Err	FNR	FPR	m'
Breast	**0.35**	0.37	0.33	0.40	0.39	0.48	0.31	**0.26**
CNS	**0.43**	0.62	0.33	1.00	0.47	0.62	0.38	1.00
Colon	0.35	0.45	0.30	0.13	**0.31**	0.41	0.25	**0.12**
Leukemia	**0.15**	–	–	0.13	0.32	–	–	**0.12**
Leukemia_3c	**0.21**	–	–	0.64	0.35	–	–	**0.53**
Leukemia_4c	**0.28**	–	–	**0.65**	0.39	–	–	0.73
Lung	**0.12**	0.01	0.47	0.92	0.16	0.00	0.65	**0.90**
Lymphoma	**0.03**	–	–	0.11	0.14	–	–	**0.08**
MLL	**0.12**	–	–	0.64	0.26	–	–	**0.55**
Ovarian	**0.03**	0.02	0.05	0.73	0.14	0.08	0.25	**0.29**
SRBCT	0.66	–	–	0.07	**0.60**	–	–	**0.08**
$\overline{\text{A}}$	**0.25**	0.29	0.30	0.49	0.32	0.32	0.37	**0.42**
σ	**0.18**	0.24	0.14	0.32	0.14	0.24	0.15	**0.32**



(a)

(b)

(c)

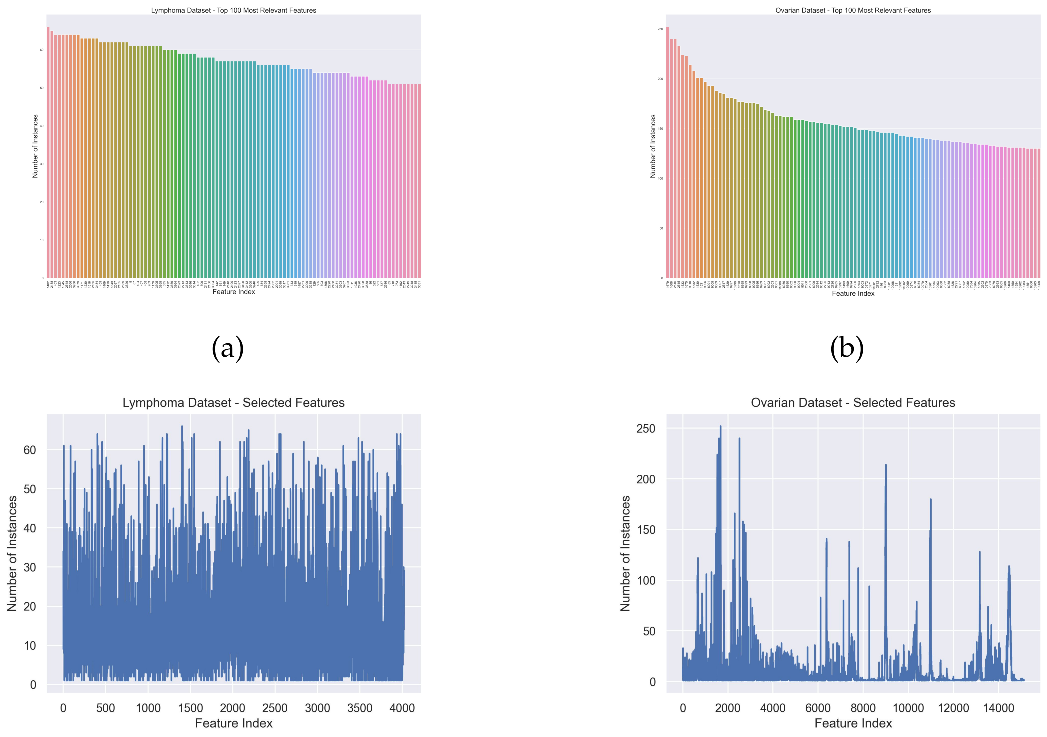
(d)

Figure 5.3: Error rate (Err) and percentage of selected features (m') for the pipeline.

5.5 Explainability of the Data

For all but three datasets (Breast, CNS, and Colon), the FS techniques applied were able to select a subset of features that could maintain or improve the classifiers' results. We now identify the most relevant features that better explain these datasets.

Figure 5.4 (top) shows the feature indices that are chosen more often on the LOOCV procedure for the Lymphoma and Ovarian datasets. For a dataset with n instances, each feature can be chosen up to n times. The importance of a feature to (accurately classify) a dataset and to explain the classification results is proportional to the number of times that feature is chosen in this procedure. We show the top 100 features. For both datasets, we can observe that only one single feature is chosen n times (on the LOOCV folds), thus it is identified as the most relevant feature (gene) for cancer detection. Afterwards, we observe a decreasing trend that shows the relative importance of the features to perform classification. The top features in this can be studied further by clinical researchers.



(a)

(b)

(c)

(d)

Figure 5.4: The number of times each feature is chosen/selected on the FS step on the LOOCV procedure for the Lymphoma (n=66, d=4026) and Ovarian (n=253, d=15154) datasets. The FS techniques are: LS for the Lymphoma dataset and RRFS(FiR) for the Ovarian dataset.

In the bottom of Figure 5.4, we show a similar plot, but now we consider all the features in the dataset, displaying the number of times each feature is chosen. We observe that only a few features are chosen approximately *n* times, being the most relevant in clinical terms.

For the remaining datasets, please refer to Appendix C which provides further insights, with similar plot results, for the most relevant features.

5.6 Comparison With Other Approaches

We now address the comparison between the experimental results attained in this work with other approaches.

Table 5.18 reports experimental results from [5] (about the R-LBG technique) and [1] (regarding the RRFS technique). These evaluations are performed on publicly available datasets, of which four of them (namely Colon, Leukemia, Lung, and SRBCT) are also used in this work.

Table 5.18: Test error rate of a 10-fold CV, for the linear SVM classifier.

	FD technique	FS technique	
Name	R-LBG(q=3, Δ=0.05)	RRFS(MM, ms=0.8)	RRFS(FiR, ms=0.8)
Colon	–	0.242	0.226
Leukemia	–	0.028	0.125
Lung	–	0.059	0.064
SRBCT	0.000	0.000	0.000

Comparing these results with ours, we can perceive that they yield roughly the same error rate. However, for the FS technique with the configurations from our work, we can see slightly better results.

Regarding the FD technique, we conclude that combining it with a FS technique, as in our work, is not a good choice. Applying it individually yields better results.

5.7 Discussion

We now summarize the conclusions of the experimental results presented across this Chapter.

Based on the combination of techniques that yielded the best results (presented in Section 5.4) we observe that applying both FD and FS techniques did not improve the results in any dataset.

For the Breast and CNS datasets, applying FS did not improve the results, but applying FD techniques did (the best result was achieved by applying the FD technique only). In addition, for the Colon dataset in particular, the best results were achieved by applying the baseline classifier to the original feature set (without FD and FS techniques). For the remaining datasets, applying FS techniques did improve the results. In some cases it improved the Err/FPR/FNR metrics, in other cases it was able to produce the same results. In either case, the reduction of the number of features improved the explainability of the results and the time to compute them.

From these experimental results we can also observe that DT does not achieve better results than the SVM classifier (DT only performs better than SVM on the CNS and Colon datasets). In addition, the EFB discretization also proved to be a better choice when compared to the MDLP technique. As for the FS techniques explored in this work, the RRFS technique is in general the best choice, taking into account the classification error and m'.

Table 5.19 summarizes the best results obtained in this work for each dataset.

Table 5.19: Summary of the best combination of techniques and their respective results for each dataset. Techniques: EFB(n_bins), RRFS(relevance metric, *ms*), SVM(C, kernel), and DT(criterion, max_depth, random_state).

	Configurations			Results			
Dataset	Discretization	Selection	Classification	Err	FNR	FPR	m'
Breast	EFB (6)	–	SVM (1, linear)	0.30	0.30	0.29	–
CNS	EFB (5)	–	DT (entropy, 5, 42)	0.18	0.33	0.10	–
Colon	–	–	DT (entropy, None, 5)	0.13	0.23	0.08	–
Leukemia	–	LS	SVM (1, linear)	0.01	–	–	0.13
Leukemia_3c	–	RRFS (FiR, 0.7)	SVM (1, linear)	0.03	–	–	0.18
Leukemia_4c	–	RRFS (FiR, 0.7)	SVM (1, linear)	0.07	–	–	0.17
Lung	–	FiR	SVM (1, linear)	0.04	0.01	0.12	0.67
Lymphoma	–	LS	SVM (1, linear)	0.00	–	–	0.22
MLL	–	RRFS (MM, 0.7)	SVM (1, linear)	0.03	–	–	0.23
Ovarian	–	RRFS (FiR, 0.7)	SVM (1, linear)	0.00	0.00	0.00	0.04
SRBCT	–	SPEC	SVM (1, linear)	0.00	–	–	0.49

6

Conclusions

This Chapter describes the key conclusions reached in this thesis. In Section 6.1, we start with an overview of what was achieved, followed by the directions of future work in Section 6.2.

6.1 Overview

Cancer detection and classification from high-dimensional DNA microarray data is an important problem, with many techniques having been successfully applied to these problems. However, more than just classifying the data, it is also important to identify the most relevant genes for the classification task, allowing for human interpretability of the classification results.

In this work, we have proposed an approach using FS and FD techniques, able to identify small subsets of relevant genes for the subsequent classifier. To evaluate this approach, we have built a machine learning pipeline consisting of a set of techniques to be applied in the eleven datasets considered in this work. This pipeline allowed us to compare and evaluate the performance of each applied technique, and thus, draw our conclusions.

The proposed approach is based on standard machine learning procedures. Even though applying both FD and FS techniques did not improve the classification performance, we observed that using these techniques individually yielded good results.

Especially FS techniques that achieve large degrees of dimensionality reduction on several public domain datasets. By using the LOOCV procedure, we also identified the features (genes) that are often more relevant for the classifier decision, providing explainability and allowing human analysis. This subset of features is clinically relevant and thus should be the focus of further investigation by clinical practitioners and or medical researchers. In addition, as often stated in the literature, we confirmed that SVM with a linear kernel seems to be an adequate classifier for these datasets. It is also important to normalize the data before performing all of these machine learning tasks.

6.2 Future Work

Given the accomplishments presented in this thesis, the directions for future work are as follows:

- We will explore more FD, FS, and classification techniques;
- We will address joint FS and FD techniques, in an attempt to improve the results;
- We will also fine tune the maximum similarity parameter of the RRFS algorithm to further reduce the size of the subsets, allowing medical experts to focus on fewer features.

References

[1] A. Ferreira and M. Figueiredo, "Efficient feature selection filters for high-dimensional data", *Pattern Recognition Letters*, vol. 33, no. 13, pages 1794 –1804, 2012, ISSN: 0167-8655. DOI: http://dx.doi.org/10.1016/j.patrec.2012.05.019. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2012.05.019.

[2] A. Ferreira and M. Figueiredo, "Exploiting the bin-class histograms for feature selection on discrete data", in *Iberian Conference on Pattern Recognition and Image Analysis*, Springer, 2015, pages 345–353.

[3] A. Ferreira and M. Figueiredo, "Unsupervised joint feature discretization and selection", in *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), LNCS 6669, Springer-Verlag Berlin Heidelberg*, Las Palmas de Gran Canaria, Spain, Jun. 2011, pages 200–207.

[4] A. Statnikov, C. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis", *Bioinformatics*, vol. 21, no. 5, pages 631–643, 2005.

[5] A.Ferreira and M.Figueiredo, "Relevance and mutual information-based feature discretization", in *Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, Barcelona, Spain, Feb. 2013.

[6] Amparo Alonso-Betanzos, Verónica Bolón-Canedo, Laura Morán-Fernández, and Noelia Sánchez-Maroño, "A review of microarray datasets: Where to find them and specific characteristics", in *Microarray Bioinformatics*, Springer, 2019, pages 65–85.

[7] Arianna Consiglio, Gabriella Casalino, Giovanna Castellano, Giorgio Grillo, Elda Perlino, Gennaro Vessio, and Flavio Licciulli, *Explaining ovarian cancer gene expression profiles with fuzzy rules and genetic algorithms. electronics 2021, 10, 375*, 2021.

[8] Arindam Bhattacharjee, William G. Richards, Jane Staunton, Cheng Li, Stefano Monti, Priya Vasa, Christine Ladd, Javad Beheshti, Raphael Bueno, Michael Gillette, *et al.*, "Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses", *Proceedings of the National Academy of Sciences*, vol. 98, no. 24, pages 13 790–13 795, 2001.

[9] Ash A. Alizadeh, Michael B. Eisen, R. Eric Davis, Chi Ma, Izidore S. Lossos, Andreas Rosenwald, Jennifer C. Boldrick, Hajeer Sabet, Truc Tran, Xin Yu, *et al.*, "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling", *Nature*, vol. 403, no. 6769, pages 503–511, 2000.

[10] *Cancer Council NSW*. [Online]. Available: https://www.cancercouncil.com.au/, Accessed on January 31st, 2022.

[11] *Cancer.Net - Doctor Approved Patient Information from ASCO*. [Online]. Available: https://www.cancer.net/, Accessed on January 31st, 2022.

[12] Christopher M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995, ISBN: 0198538642.

[13] Claude Sammut and Geoffrey I. Webb, *Encyclopedia of Machine Learning*. Springer, 2011, ISBN: 9780387307688.

[14] David M. W. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation", *arXiv preprint arXiv:2010.16061*, 2020.

[15] Diego A. Forero and George P. Patrinos, *Genome Plasticity in Health and Disease*, ser. Translational and Applied Genomics. Elsevier Science, 2020, ISBN: 9780128178201.

[16] *DNA Microarray Experiment*. [Online]. Available: https://learn.genetics.utah.edu/content/labs/microarray/, Accessed on January 31st, 2022.

[17] Emanuel F. Petricoin III, Ali M. Ardekani, Ben A. Hitt, Peter J. Levine, Vincent A. Fusaro, Seth M. Steinberg, Gordon B. Mills, Charles Simone, David A. Fishman, Elise C. Kohn, *et al.*, "Use of proteomic patterns in serum to identify ovarian cancer", *The lancet*, vol. 359, no. 9306, pages 572–577, 2002.

[18] Ethem Alpaydin, *Introduction to Machine Learning*, 3rd ed. The MIT Press, 2014, ISBN: 0-262-02818-9.

[19] Frauke Friedrichs and Christian Igel, "Evolutionary tuning of multiple svm parameters", *Neurocomputing*, vol. 64, pages 107–117, 2005.

[20] Gordon Hughes, "On the mean accuracy of statistical pattern recognizers", *IEEE Transactions on Information Theory*, vol. 14, no. 1, pages 55–63, 1968.

[21] Harold Hotelling, "Analysis of a complex of statistical variables into principal components.", *Journal of educational psychology*, vol. 24, no. 6, page 417, 1933.

[22] I. Witten, E. Frank, M. Hall, and C. Pal, *Data mining: practical machine learning tools and techniques*, fourth. Morgan Kauffmann, 2016, ISBN: 978-0128042915.

[23] Igor Kononenko, "Estimating attributes: Analysis and extensions of relief", in *European conference on machine learning*, Springer, 1994, pages 171–182.

[24] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A. Zadeh, *Feature Extraction: Foundations and Applications*. Springer, 2006, ISBN: 9783540354871.

[25] Ivan Miguel Pires, Faisal Hussain, Nuno M. Garcia, Petre Lameski, and Eftim Zdravevski, "Homogeneous data normalization and deep learning: A case study in human activity classification", *Future Internet*, vol. 12, no. 11, page 194, 2020.

[26] James Dougherty, Ron Kohavi, and Mehran Sahami, "Supervised and unsupervised discretization of continuous features", in *Machine learning proceedings 1995*, Elsevier, 1995, pages 194–202.

[27] Javed Khan, Jun S. Wei, Markus Ringnér, Lao H. Saal, Marc Ladanyi, Frank Westermann, Frank Berthold, Manfred Schwab, Cristina R. Antonescu, Carsten Peterson, *et al.*, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks", *Nature Medicine*, vol. 7, no. 6, pages 673–679, 2001.

[28] Jundong Li, Kewei Cheng, and Suhang Wang, *Open-source feature selection repository in python*, version 1.0.0. [Online]. Available: `https://github.com/jundongl/scikit-feature`, Accessed on January 31st, 2022.

[29] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu, "Feature selection: A data perspective", *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, page 94, 2018.

[30] Laura J. Van't Veer, Hongyue Dai, Marc J. Van De Vijver, Yudong D. He, Augustinus A. M. Hart, Mao Mao, Hans L. Peterse, Karin Van Der Kooy, Matthew J. Marton, Anke T. Witteveen, *et al.*, "Gene expression profiling predicts clinical outcome of breast cancer", *Nature*, vol. 415, no. 6871, pages 530–536, 2002.

[31] Leo Breiman, "Random forests", *Machine learning*, vol. 45, no. 1, pages 5–32, 2001.

[32] Henry Lin, *Minimum Description Length Binning*, version 0.3.3, Nov. 2017. [Online]. Available: `https://github.com/hlin117/mdlp-discretization`, Accessed on January 31st, 2022.

[33] Luying Liu, Jianchu Kang, Jing Yu, and Zhongliang Wang, “A comparative study on unsupervised feature selection methods for text clustering”, in *2005 International Conference on Natural Language Processing and Knowledge Engineering*, IEEE, 2005, pages 597–601. DOI: 10.1109/NLPKE.2005.1598807.

[34] *MATLAB for Artificial Intelligence*. [Online]. Available: https://www.mathworks.com/, Accessed on January 31st, 2022.

[35] *Matplotlib: Visualization with Python*. [Online]. Available: https://matplotlib.org/, Accessed on January 31st, 2022.

[36] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, *Foundations of Machine Learning*. MIT press, 2018.

[37] Mikhail Belkin and Partha Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation”, *Neural computation*, vol. 15, no. 6, pages 1373–1396, 2003.

[38] Adara Nogueira, *My Research Software - Clinical Data Mining and Classification*, Jan. 2022. [Online]. Available: https://github.com/adaranogueira/cancer-diagnosis-ml, Accessed on January 31st, 2022.

[39] *Numpy - The fundamental package for scientific computing with Python*. [Online]. Available: https://numpy.org/, Accessed on January 31st, 2022.

[40] *Oracle Java*. [Online]. Available: https://www.oracle.com/java/, Accessed on January 31st, 2022.

[41] P. Meyer, C. Schretter, and G. Bontempi, “Information-theoretic feature selection in microarray data using variable complementarity”, *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 3, pages 261–274, 2008.

[42] *Pandas - Data analysis and manipulation tool*. [Online]. Available: https://pandas.pydata.org/, Accessed on January 31st, 2022.

[43] *Python Programming language*. [Online]. Available: https://www.python.org/, Accessed on January 31st, 2022.

[44] *Pytorch - From research to production*. [Online]. Available: https://pytorch.org/, Accessed on January 31st, 2022.

[45] Ramón Díaz-Uriarte and Sara Alvarez De Andrés, “Gene selection and classification of microarray data using random forest”, *BMC bioinformatics*, vol. 7, no. 1, pages 1–13, 2006.

[46] Richard E. Bellman, *Adaptive Control Processes - A Guided Tour*. Princeton University Press, 2015, ISBN: 9780070428072.

[47] Richard M. Simon, Edward L. Korn, Lisa M. McShane, Michael D. Radmacher, George W. Wright, and Yingdong Zhao, *Design and analysis of DNA microarray investigations*. Springer Science & Business Media, 2003.

[48] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Wiley, 2001, ISBN: 9780471056690.

[49] Robert A. Weinberg, *The Biology of Cancer*, 2nd ed. Garland Science, 2014, ISBN: 9780815342205.

[50] Robert C. Holte, “Very simple classification rules perform well on most commonly used datasets”, *Machine learning*, vol. 11, no. 1, pages 63–90, 1993.

[51] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu, “Diagnosis of multiple cancer types by shrunken centroids of gene expression”, *Proceedings of the National Academy of Sciences*, vol. 99, no. 10, pages 6567–6572, 2002.

[52] Ronald A. Fisher, “The use of multiple measurements in taxonomic problems”, *Annals of eugenics*, vol. 7, no. 2, pages 179–188, 1936.

[53] Salvador Garcia, Julian Luengo, José Antonio Sáez, Victoria Lopez, and Francisco Herrera, “A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pages 734–750, 2012.

[54] *Scikit-learn - Machine learning in Python*. [Online]. Available: https://scikit-learn.org/, Accessed on January 31st, 2022.

[55] *SciPy - Fundamental library for scientific computing*. [Online]. Available: https://www.scipy.org/, Accessed on January 31st, 2022.

[56] Scott A. Armstrong, Jane E. Staunton, Lewis B. Silverman, Rob Pieters, Monique L. den Boer, Mark D. Minden, Stephen E. Sallan, Eric S. Lander, Todd R. Golub, and Stanley J. Korsmeyer, “Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia”, *Nature Genetics*, vol. 30, no. 1, pages 41–47, 2002.

[57] Scott L. Pomeroy, Pablo Tamayo, Michelle Gaasenbeek, Lisa M. Sturla, Michael Angelo, Margaret E. McLaughlin, John Y. H. Kim, Liliana C. Goumnerova, Peter M. Black, Ching Lau, *et al.*, “Prediction of central nervous system embryonal tumour outcome based on gene expression”, *Nature*, vol. 415, no. 6870, pages 436–442, 2002.

[58] *Seaborn: statistical data visualization*. [Online]. Available: https://seaborn.pydata.org/, Accessed on January 31st, 2022.

[59] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. 2020, ISBN: 9780134610993.

[60] *Tensorflow - An end-to-end open source machine learning platform*. [Online]. Available: https://www.tensorflow.org/, Accessed on January 31st, 2022.

[61] *The R Project for Statistical Computing*. [Online]. Available: https://www.r-project.org/, Accessed on January 31st, 2022.

[62] Todd R. Golub, Donna K. Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P. Mesirov, Hilary Coller, Mignon L. Loh, James R. Downing, Mark A. Caligiuri, *et al.*, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring", *Science*, vol. 286, no. 5439, pages 531–537, 1999.

[63] Tom M. Mitchell, *Machine Learning*. McGraw-hill New York, 1997, ISBN: 9780070428072.

[64] Uri Alon, Naama Barkai, Daniel A. Notterman, Kurt Gish, Suzanne Ybarra, Daniel Mack, and Arnold J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays", *Proceedings of the National Academy of Sciences*, vol. 96, no. 12, pages 6745–6750, 1999.

[65] Usama Fayyad and Keki Irani, "Multi-interval discretization of continuous-valued attributes for classification learning", in *Proceedings of the International Joint Conference on Uncertainty in AI*, 1993, pages 1022–1027.

[66] *Visual Studio Code*. [Online]. Available: https://code.visualstudio.com/, Accessed on January 31st, 2022.

[67] Wai-Ki Yip, Samir B. Amin, and Cheng Li, "A survey of classification techniques for microarray data analysis", in *Handbook of Statistical Bioinformatics*. Springer Berlin Heidelberg, 2011, pages 193–223, ISBN: 978-3-642-16345-6. DOI: 10.1007/978-3-642-16345-6_10. [Online]. Available: https://doi.org/10.1007/978-3-642-16345-6_10.

[68] Wedad Alawad, Mohamed Zohdy, and Debatosh Debnath, "Tuning hyperparameters of decision tree classifiers using computationally efficient schemes", in *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, IEEE, 2018, pages 168–169.

[69] *WEKA - The workbench for machine learning*. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/, Accessed on January 31st, 2022.

[70] *World Health Organization*. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cancer, Accessed on January 31st, 2022.

[71] Xiaofei He, Deng Cai, and Partha Niyogi, "Laplacian score for feature selection", *Advances in neural information processing systems*, vol. 18, 2005.

[72] Zexuan Zhu, Yew-Soon Ong, and Manoranjan Dash, "Markov blanket-embedded genetic algorithm for gene selection", *Pattern Recognition*, vol. 40, no. 11, pages 3236–3248, 2007. [Online]. Available: http://csse.szu.edu.cn/staff/zhuzx/Datasets.html.

[73] Zheng Zhao and Huan Liu, "Spectral feature selection for supervised and unsupervised learning", in *Proceedings of the 24th international conference on Machine learning*, 2007, pages 1151–1157.

[74] Zifa Li, Weibo Xie, and Tao Liu, "Efficient feature selection and classification for microarray data", *PloS one*, vol. 13, no. 8, e0202167, 2018.

A

Dataset Description

This Appendix provides a brief description of all DNA microarray datasets used in this work. In addition, it presents the summarized characteristics of each one.

More information on cancer and their causes can be found in [10, 11].

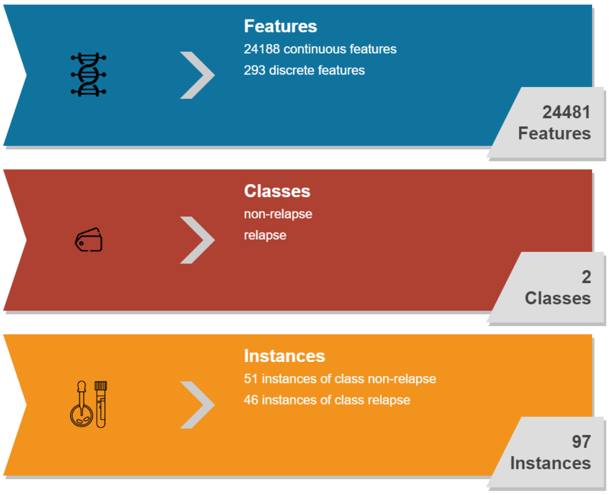
A.1 Breast

The Breast dataset [30] allows us to improve the quality in diagnosing the recurrence of breast cancer [49].

This dataset represents a binary classification task that distinguishes between relapsed patients from non-relapsed patients.

It may help healthcare professionals to make new diagnosis by identifying in their patients if a particular cancer has manifested itself again (relapse) and those in which it didn't (non-relapse). Relapsing may occur since it's possible for a few of the original cancer cells to survive the previous treatment [49].

Figure A.1 shows the summarized properties of the Breast dataset.



Features
24188 continuous features
293 discrete features
24481 Features

Classes
non-relapse
relapse
2 Classes

Instances
51 instances of class non-relapse
46 instances of class relapse
97 Instances

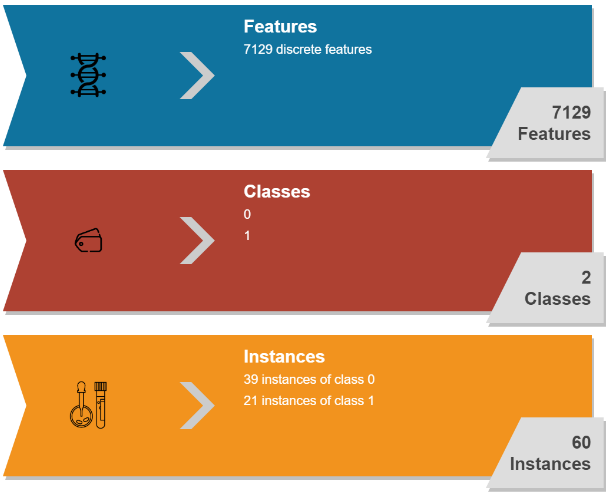
Figure A.1: Breast Dataset Properties

A.2 CNS

The CNS dataset [57] allows us to improve quality in diagnosing central nervous system tumor, i.e. tumors in the brain or spinal cord [49].

This dataset represents a binary classification task that distinguishes between the presence or absence of tumors in the central nervous system.

Figure A.2 shows the summarized properties of the CNS dataset.



Features
7129 discrete features
7129 Features

Classes
0
1
2 Classes

Instances
39 instances of class 0
21 instances of class 1
60 Instances

Figure A.2: CNS Dataset Properties

A.3 Colon

The Colon dataset [64] was collected in order to improve the quality in diagnosing the state of colon tissues (the longest part of the large intestine) [49].

This dataset represents a binary classification task that allows us to distinguish between the presence or absence of tumors in the colon's tissues.

Figure A.3 shows the summarized properties of the Colon dataset.

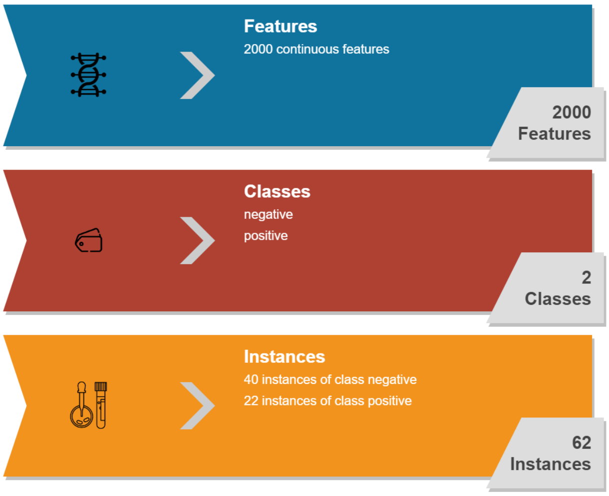


Figure A.3: Colon Dataset Properties

A.4 Leukemia

The Leukemia dataset [62] allows us to improve the quality in diagnosing acute leukemia, a blood cancer [49].

This dataset represents a binary classification task that allows us to distinguish between acute lymphocytic leukemia and acute myelogenous leukemia.

The difference between acute lymphocytic leukemia and acute myelogenous leukemia can be found in the type of white blood cell affected. For instance, acute lymphocytic leukemia can develop from different types of lymphocytes cells, such as B-cells or T-cells, and acute myelogenous leukemia develops from myeloid cells [49].

Figure A.4 shows the summarized properties of the Leukemia dataset.

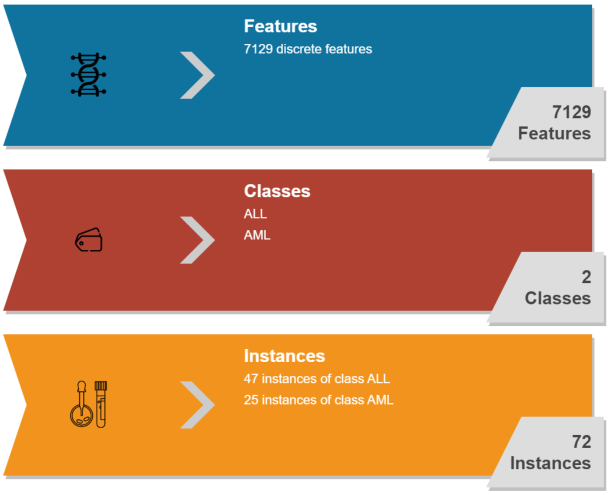


Figure A.4: Leukemia Dataset Properties

A.5 Leukemia_3c

The Leukemia_3c dataset [62] allows us to improve quality in diagnosing blood cells that have become cancerous.

This dataset represents a multi-class classification task that allows us to distinguish between three types of cells which became cancerous: myeloid white blood cell, B-cell and T-cell.

Once they identify the affected cell, healthcare professionals can determine which acute leukemia cancer a patient has (see "The blood" section, at [10]).

Thus, if the result is a myeloid white blood cell the diagnosis is acute myelogenous leukemia cancer, in which case the class value is AML. On the other hand, if the result is one of the white blood cell called lymphocyte (which belong to the immune system), i.e. B-cell or T-cell, the diagnosis is acute lymphocytic leukemia.

Figure A.5 shows the summarized properties of the Leukemia_3c dataset.

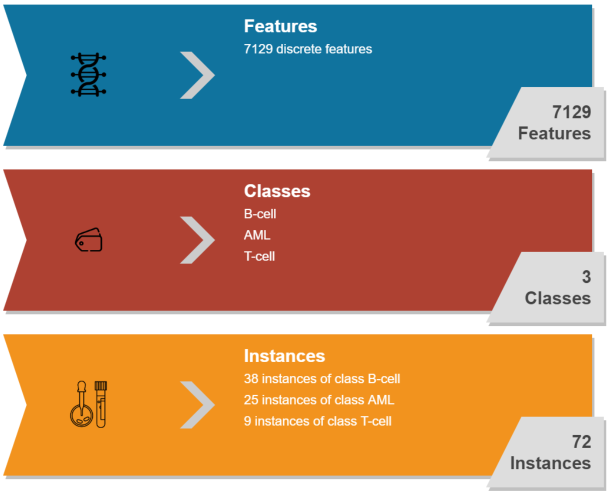


Figure A.5: Leukemia_3c Dataset Properties

A.6 Leukemia_4c

The Leukemia_4c dataset [62] also allows us to improve quality in diagnosing blood cells that have become cancerous.

This dataset represents a multi-class classification task that distinguishes between four types of cells which became cancerous: B-cell, T-cell, bone marrow cell (BM) and peripheral blood cell (PB).

Since acute leukemia cancer can originate from both blood cells and the cells in the bone marrow, it may help healthcare professionals to identify the type of cancer and its location.

Figure A.6 shows the summarized properties of the Leukemia_4c dataset.

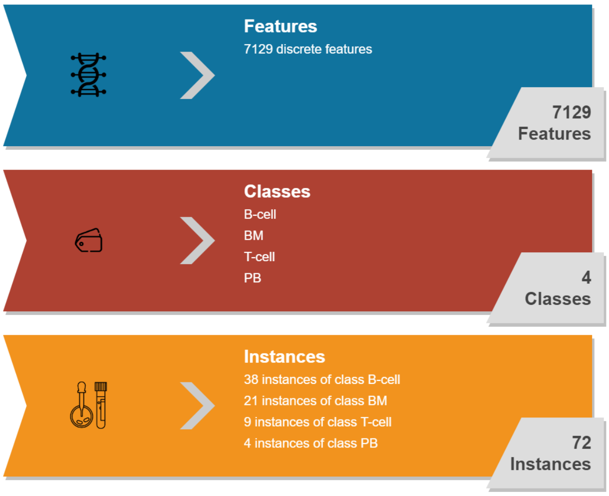


Figure A.6: Leukemia_4c Dataset Properties

A.7 Lung

The Lung dataset [8] allows us to improve quality in diagnosing lung cancer [49].

This dataset represents a multi-class classification task that allows us to distinguish between four types of lung cancer and normal tissues: adenocarcinoma (label 1), normal (label 2), small cell lung carcinoma (label 3), squamous cell lung carcinoma (label 4), and lung carcinoid tumor (label 5).

Figure A.7 shows the summarized properties of the Lung dataset.

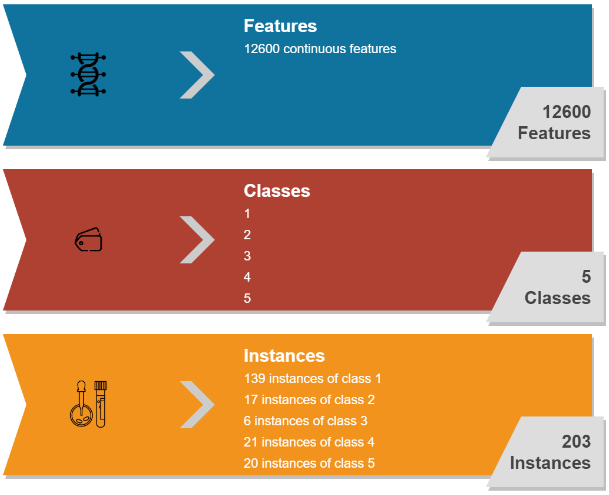


Figure A.7: Lung Dataset Properties

A.8 Lymphoma

The Lymphoma dataset [9] allows us to improve quality in diagnosing subtypes of non-Hodgkin's lymphoma, a lymphatic system cancer.

This dataset represents a multi-class classification task that distinguishes between three subtypes of B-cell lymphoma: follicular lymphoma, diffuse large B-cell lymphoma, small lymphocytic lymphoma[1].

Different types and subtypes of non-Hodgkin's lymphoma can be determined according to which blood cell was affected, for instance, B-cell (such as in this work), T-cell, or natural killer cell [49].

Figure A.8 shows the summarized properties of the Lymphoma dataset.

[1] Which is the same as B-cell chronic lymphocytic leukemia, reason why the class label is CLL.

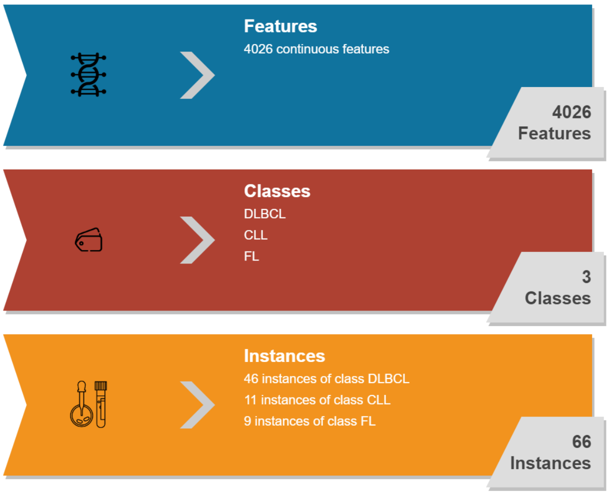


Figure A.8: Lymphoma Dataset Properties

A.9 MLL

The MLL dataset [56] allows us to improve quality in diagnosing different types of acute leukemia.

This dataset represents a multi-class classification task that allows us to distinguish between three types of acute leukemia: acute myelogenous leukemia, acute lymphocytic leukemia, and mixed lineage leukemia.

Patients with mixed lineage leukemia manifest characteristics of both, acute myelogenous leukemia and acute lymphocytic leukemia [49].

Figure A.9 shows the summarized properties of the MLL dataset.

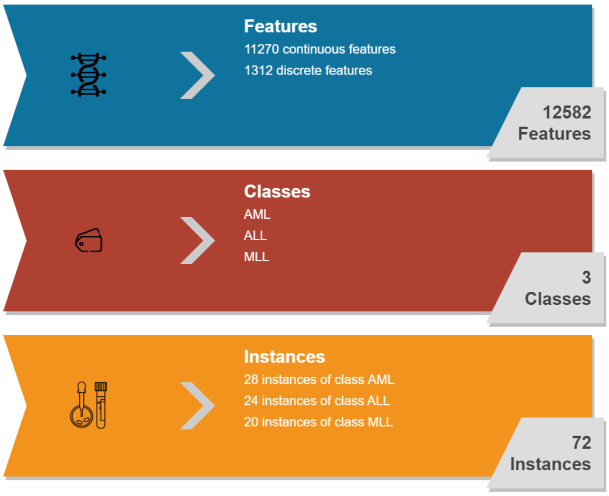

Features
11270 continuous features
1312 discrete features
12582
Features
Classes
AML
ALL
MLL
3
Classes
Instances
28 instances of class AML
24 instances of class ALL
20 instances of class MLL
72
Instances

Figure A.9: MLL Dataset Properties

A.10 Ovarian

The Ovarian dataset [17] allows us to improve quality in diagnosing cancer in the ovaries (part of a woman's reproductive system) [49].

This dataset represents a binary classification task that allows us to distinguish between the presence or absence of ovarian cancer.

Figure A.10 shows the summarized properties of the Ovarian dataset.

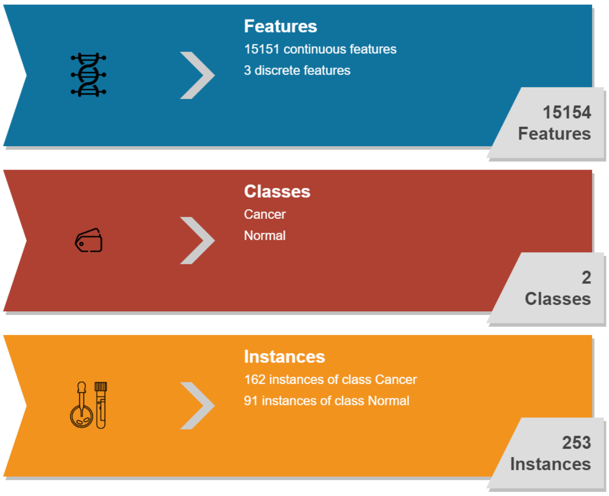


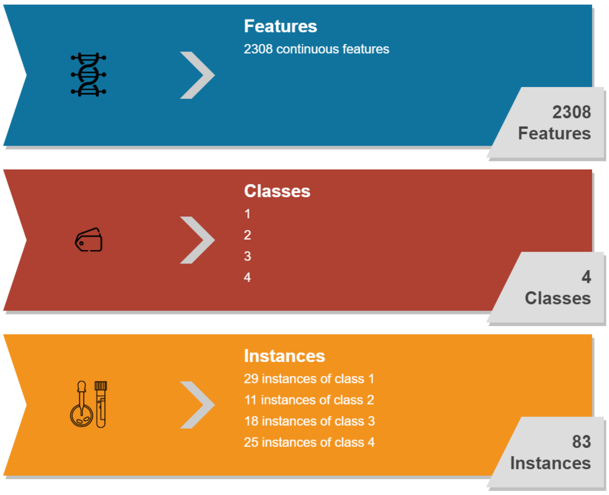
Figure A.10: Ovarian Dataset Properties

A.11 SRBCT

The SRBCT dataset [27] allows us to improve quality in diagnosing small round blue cell tumors [49].

This dataset represents a multi-class classification task that distinguishes between four types of tumor: Ewing's sarcoma (label 1), Burkitt's lymphoma (also know as non-Hodgkin lymphoma, label 2), neuroblastoma (label 3), and rhabdomyosarcoma (label 5).

Figure A.11 shows the summarized properties of the SRBCT dataset.



Features

2308 continuous features

2308 Features

Classes

1

2

3

4

4 Classes

Instances

29 instances of class 1

11 instances of class 2

18 instances of class 3

25 instances of class 4

83 Instances

Figure A.11: SRBCT Dataset Properties

B

Pipeline’s Configurations

This Appendix includes a detailed analysis of the best suited techniques for each dataset used in the machine learning pipeline.

The following procedure was used to evaluate which techniques are best suited for each phase of the pipeline. This procedure is comprised of two steps. The first, consists of identifying the techniques that give the best results for each phase of the pipeline and for each dataset individually. Given the results of the first step, the second consists of identifying the best combination of the selected techniques for the whole pipeline.

- **Dataset:** Breast
  - **Data classification (baseline)**

    Processing: normalized

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.31; FNR=0.30; FPR=0.31
  - **Data representation**

    Processing: normalized

    Discretizer: EFB (n_bins=6)

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.30; FNR=0.30; FPR=0.29

  - **Dimensionality reduction**

    Processing: normalized

    Feature Selector: RRFS (relevance measure=FiR; *ms*=0.7)

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.26; FNR=0.33; FPR=0.20

  - **Pipeline**

    Processing: normalized

    Discretizer: EFB (n_bins=6)

    Feature Selector: RRFS (relevance measure=FiR; *ms*=0.7)

    Classifier: SVM (C=1; kernel=linear)

- **Dataset:** CNS

  - **Data classification (baseline)**

    Processing: normalized

    Classifier: DT (criterion=entropy; max_depth=2; random_state=42)

    Evaluations: Err=0.18; FNR=0.48; FPR=0.03

  - **Data representation**

    Processing: normalized

    Discretizer: EFB (n_bins=5)

    Classifier: DT (criterion=entropy; max_depth=5; random_state=42)

    Evaluations: Err=0.18; FNR=0.33; FPR=0.10

  - **Dimensionality reduction**

    Processing: normalized

    Feature Selector: SPEC

    Classifier: DT (criterion=entropy; max_depth=5; random_state=42)

    Evaluations: Err=0.22; FNR=0.38; FPR=0.13

  - **Pipeline**

    Processing: normalized

    Discretizer: EFB (n_bins=5)

    Feature Selector: SPEC

    Classifier: DT (criterion=entropy; max_depth=5; random_state=42)

- **Dataset:** Colon

- **Data classification (baseline)**

  Processing: original/normalized

  Classifier: DT (criterion=entropy; max_depth=None; random_state=5)

  Evaluations: Err=0.13; FNR=0.23; FPR=0.08

- **Data representation**

  Processing: normalized

  Discretizer: MDLP

  Classifier: DT (criterion=entropy; max_depth=5; random_state=42)

  Evaluations: Err=0.15; FNR=0.14; FPR=0.15

- **Dimensionality reduction**

  Processing: normalized

  Feature Selector: LS

  Classifier: SVM (C=1; kernel=linear)

  Evaluations: Err=0.16; FNR=0.27; FPR=0.10

- **Pipeline**

  Processing: original/normalized

  Discretizer: MDLP

  Feature Selector: LS

  Classifier: DT (criterion=entropy; max_depth=None; random_state=5)

- **Dataset:** Leukemia

  - **Data classification (baseline)**

    Processing: normalized

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.01

  - **Data representation**

    Processing: normalized

    Discretizer:

    · EFB (n_bins=2^{a}; n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)

    · MDLP

    Classifier:

    · SVM (C=1; kernel=linear)

    - DT[a] (criterion=entropy; max_depth=5; random_state=42)

  Evaluations: Err=0.01

- **Dimensionality reduction**

  Processing: normalized

  Feature Selector:

    - LS
    - SPEC
    - RRFS (relevance measure=MM; *ms*=0.7)
    - FiR
    - RRFS (relevance measure=FiR; *ms*=0.7)

  Classifier: SVM (C=1; kernel=linear)

  Evaluations: Err=0.01

- **Pipeline 1**

  Processing: normalized

  Discretizer: EFB (n_bins=2)

  Feature Selector:

    - LS
    - SPEC
    - RRFS (relevance measure=MM; *ms*=0.7)
    - FiR
    - RRFS (relevance measure=FiR; *ms*=0.7)

  Classifier: DT (criterion=entropy; max_depth=5; random_state=42)

- **Pipeline 2**

  Processing: normalized

  Discretizer:

    - EFB (n_bins=2; n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)
    - MDLP

  Feature Selector:

    - LS
    - SPEC
    - RRFS (relevance measure=MM; *ms*=0.7)
    - FiR
    - RRFS (relevance measure=FiR; *ms*=0.7)

Classifier: SVM (C=1; kernel=linear)

- **Dataset:** Leukemia_3c
  - **Data classification (baseline)**

    Processing: normalized

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.04
  - **Data representation**

    Processing: normalized

    Discretizer:
    - EFB (n_bins=2; n_bins=3; n_bins=4; n_bins=5)
    - MDLP

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.03
  - **Dimensionality reduction**

    Processing: normalized

    Feature Selector: RRFS (relevance measure=FiR; *ms*=0.7)

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.03
  - **Pipeline**

    Processing: normalized

    Discretizer:
    - EFB (n_bins=2; n_bins=3; n_bins=4; n_bins=5)
    - MDLP

    Feature Selector: RRFS (relevance measure=FiR; *ms*=0.7)

    Classifier: SVM (C=1; kernel=linear)
- **Dataset:** Leukemia_4c
  - **Data classification (baseline)**

    Processing: normalized

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.07
  - **Data representation**

Processing: normalized

Discretizer:

· EFB (n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)

· MDLP

Classifier: SVM (C=1; kernel=linear)

Evaluations: Err=0.07

- **Dimensionality reduction**

Processing: normalized

Feature Selector:

· RRFS (relevance measure=MM; *ms*=0.7)

· FiR

· RRFS (relevance measure=FiR; *ms*=0.7)

Classifier: SVM (C=1; kernel=linear)

Evaluations: Err=0.07

- **Pipeline**

Processing: normalized

Discretizer:

· EFB (n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)

· MDLP

Feature Selector:

· RRFS (relevance measure=MM; *ms*=0.7)

· FiR

· RRFS (relevance measure=FiR; *ms*=0.7)

Classifier: SVM (C=1; kernel=linear)

- **Dataset:** Lung

- **Data classification (baseline)**

Processing: normalized

Classifier: SVM (C=1; kernel=linear)

Evaluations: Err=0.05; FNR=0.01; FPR=0.12

- **Data representation**

Processing: normalized

Discretizer: EFB (n_bins=5; n_bins=6; n_bins=7)

Classifier: SVM (C=1; kernel=linear)

Evaluations: Err=0.04; FNR=0.01; FPR=0.18

- **Dimensionality reduction**

  Processing: normalized

  Feature Selector: FiR

  Classifier: SVM (C=1; kernel=linear)

  Evaluations: Err=0.04; FNR=0.01; FPR=0.12

- **Pipeline**

  Processing: normalized

  Discretizer: EFB (n_bins=5; n_bins=6; n_bins=7)

  Feature Selector: FiR

  Classifier: SVM (C=1; kernel=linear)

- **Dataset:** Lymphoma

  - **Data classification (baseline)**

    Processing: original[a]/normalized[b]

    Classifier:

    - SVM (C=1; kernel=linear; kernel=poly[b]; kernel=rbf; kernel=sigmoid[a])
    - DT (criterion=gini[c]; criterion=entropy; max_depth=None[a,c]; max_depth=2; max_depth=5; max_depth=7; max_depth=10; random_state=42)

    Evaluations: Err=0.00

  - **Data representation**

    Processing: normalized

    Discretizer:

    - EFB (n_bins=2; n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)
    - MDLP

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.00

  - **Dimensionality reduction**

    Processing: normalized

    Feature Selector:

    - LS
    - SPEC

- FiR

Classifier: SVM (C=1; kernel=linear)

Evaluations: Err=0.00

- **Pipeline 1**

  Processing: original

  Discretizer:

  - EFB (n_bins=2; n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)
  - MDLP

  Feature Selector:

  - LS
  - SPEC
  - FiR

  Classifier:

  - SVM (C=1; kernel=linear; kernel=rbf; kernel=sigmoid)
  - DT (criterion=gini; criterion=entropy; max_depth=None; random_state=42)

- **Pipeline 2**

  Processing: normalized

  Discretizer:

  - EFB (n_bins=2; n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)
  - MDLP

  Feature Selector:

  - LS
  - SPEC
  - FiR

  Classifier:

  - SVM (C=1; kernel=linear; kernel=poly; kernel=rbf)
  - DT (criterion=gini; max_depth=None; random_state=42)
  - DT (criterion=entropy; max_depth=None; max_depth=2; max_depth=5; max_depth=7; max_depth=10; random_state=42)

- **Dataset:** MLL

  - **Data classification (baseline)**

    Processing: normalized

Classifier: SVM (C=1; kernel=linear)

Evaluations: Err=0.03

- **Data representation**

  Processing: normalized

  Discretizer:

  · EFB (n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)

  · MDLP

  Classifier: SVM (C=1; kernel=linear)

  Evaluations: Err=0.03

- **Dimensionality reduction**

  Processing: normalized

  Feature Selector:

  · RRFS (relevance measure=MM; ms=0.7)

  · FiR

  Classifier: SVM (C=1; kernel=linear)

  Evaluations: Err=0.03

- **Pipeline**

  Processing: normalized

  Discretizer:

  · EFB (n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)

  · MDLP

  Feature Selector:

  · RRFS (relevance measure=MM; ms=0.7)

  · FiR

  Classifier: SVM (C=1; kernel=linear)

- **Dataset:** Ovarian

  - **Data classification (baseline)**

    Processing: original/normalized

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.00; FNR=0.00; FPR=0.00

  - **Data representation**

    Processing: normalized

    Discretizer: EFB (n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.00; FNR=0.00; FPR=0.00

  - **Dimensionality reduction**

    Processing: normalized

    Feature Selector:

    - LS
    - SPEC
    - FiR
    - RRFS (relevance measure=FiR; *ms*=0.7)

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.00; FNR=0.00; FPR=0.00

  - **Pipeline**

    Processing: original/normalized

    Discretizer: EFB (n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)

    Feature Selector:

    - LS
    - SPEC
    - FiR
    - RRFS (relevance measure=FiR; *ms*=0.7)

    Classifier: SVM (C=1; kernel=linear)

- **Dataset:** SRBCT

  - **Data classification (baseline)**

    Processing: original/normalized[a]

    Classifier: SVM (C=1; kernel=linear[a]; kernel=poly)

    Evaluations: Err=0.00

  - **Data representation**

    Processing: normalized

    Discretizer: EFB (n_bins=2; n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)

    Classifier: SVM (C=1; kernel=linear)

    Evaluations: Err=0.00

- **Dimensionality reduction**

  Processing: normalized

  Feature Selector:

  - SPEC
  - RRFS (relevance measure=MM; *ms*=0.7)
  - FiR
  - RRFS (relevance measure=FiR; *ms*=0.7)

  Classifier: SVM (C=1; kernel=linear)

  Evaluations: Err=0.00

- **Pipeline 1**

  Processing: original

  Discretizer: EFB (n_bins=2; n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)

  Feature Selector:

  - SPEC
  - RRFS (relevance measure=MM; *ms*=0.7)
  - FiR
  - RRFS (relevance measure=FiR; *ms*=0.7)

  Classifier: SVM (C=1; kernel=linear; kernel=poly)

- **Pipeline 2**

  Processing: normalized

  Discretizer: EFB (n_bins=2; n_bins=3; n_bins=4; n_bins=5; n_bins=6; n_bins=7)

  Feature Selector:

  - SPEC
  - RRFS (relevance measure=MM; *ms*=0.7)
  - FiR
  - RRFS (relevance measure=FiR; *ms*=0.7)

  Classifier: SVM (C=1; kernel=linear)

C

Additional Experimental Results

This Appendix contains additional experimental results that provide further insights on the most relevant features of the DNA microarray datasets explored in this work. These experimental results complement the ones reported in Section 5.5.

C.1 Selected Features

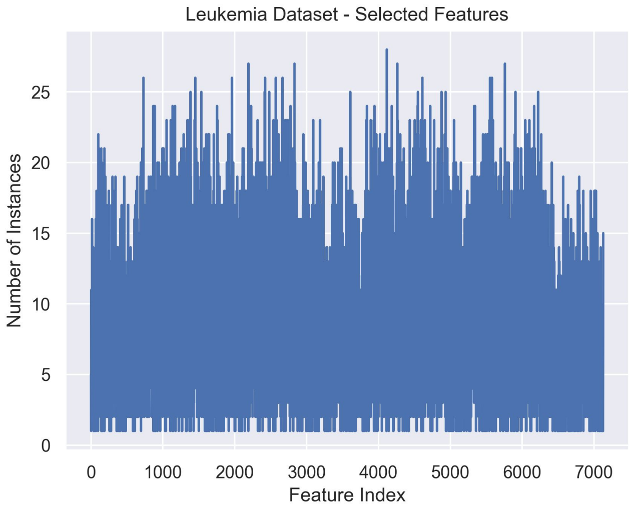


Figure C.1: The number of times each feature is chosen/selected using the LS technique on the FS step on the LOOCV procedure for the Leukemia dataset (n=72, d=7129).

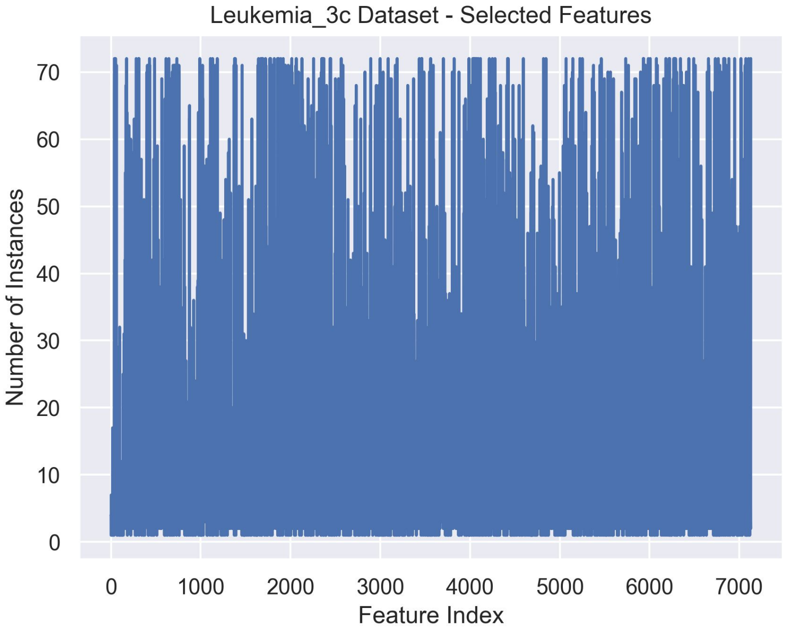


Figure C.2: The number of times each feature is chosen/selected using the RRFS(FiR) technique on the FS step on the LOOCV procedure for the Leukemia_3c dataset (n=72, d=7129).

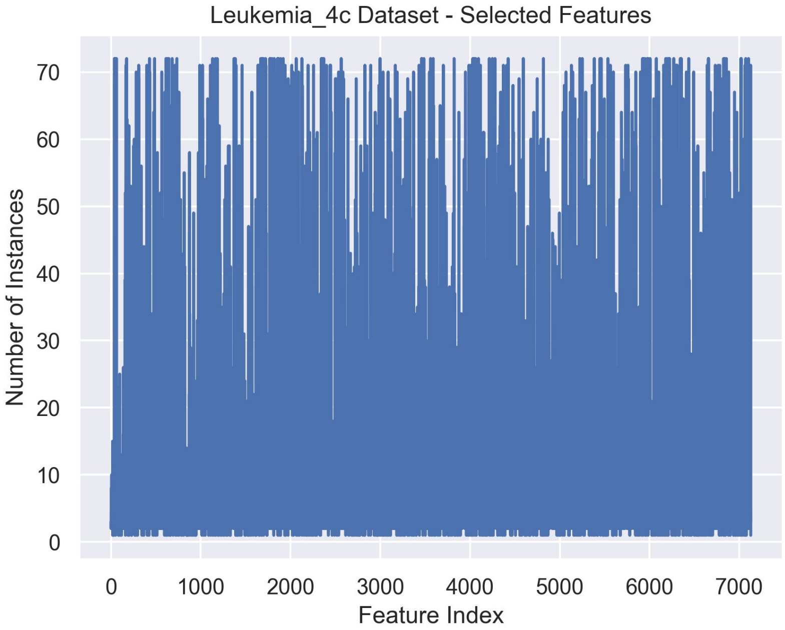


Figure C.3: The number of times each feature is chosen/selected using the RRFS(FiR) technique on the FS step on the LOOCV procedure for the Leukemia_4c dataset (n=72, d=7129).

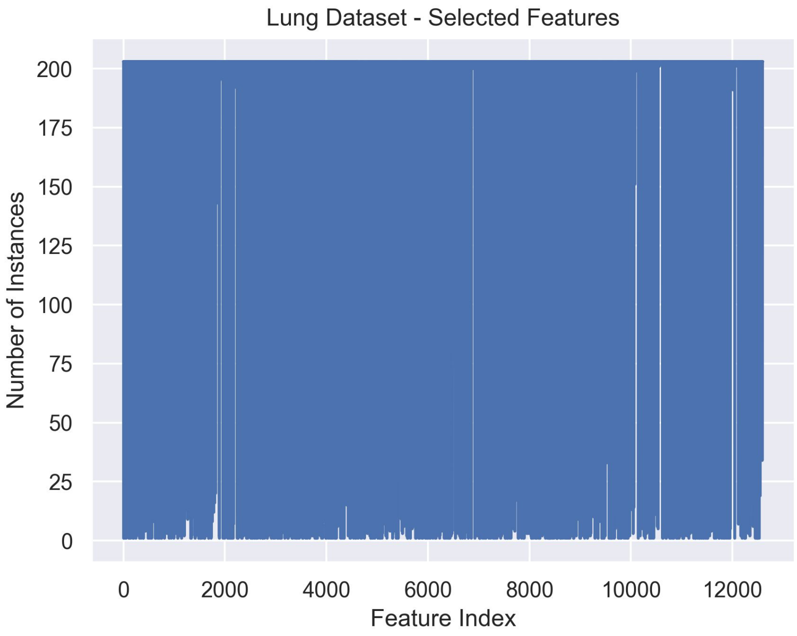


Figure C.4: The number of times each feature is chosen/selected using the FiR technique on the FS step on the LOOCV procedure for the Lung dataset (n=203, d=12600).

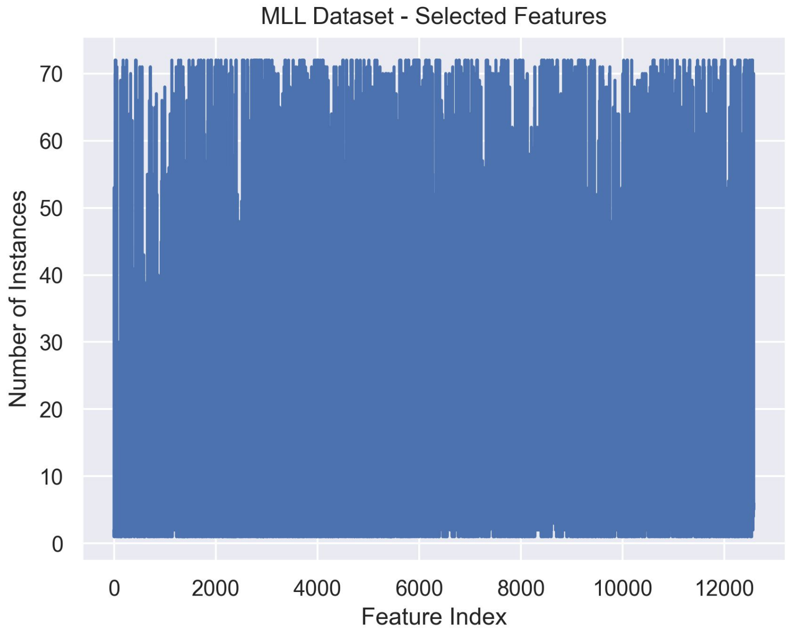


Figure C.5: The number of times each feature is chosen/selected using the RRFS(MM) technique on the FS step on the LOOCV procedure for the MLL dataset (n=72, d=12582).

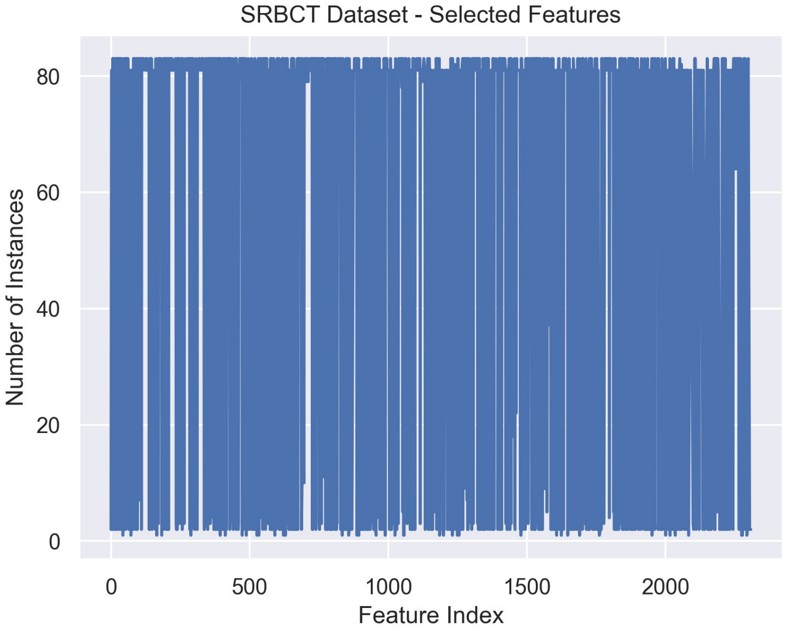


Figure C.6: The number of times each feature is chosen/selected using the SPEC technique on the FS step on the LOOCV procedure for the SRBCT dataset (n=83, d=2308).

C.2 Top-100 Most Relevant Features

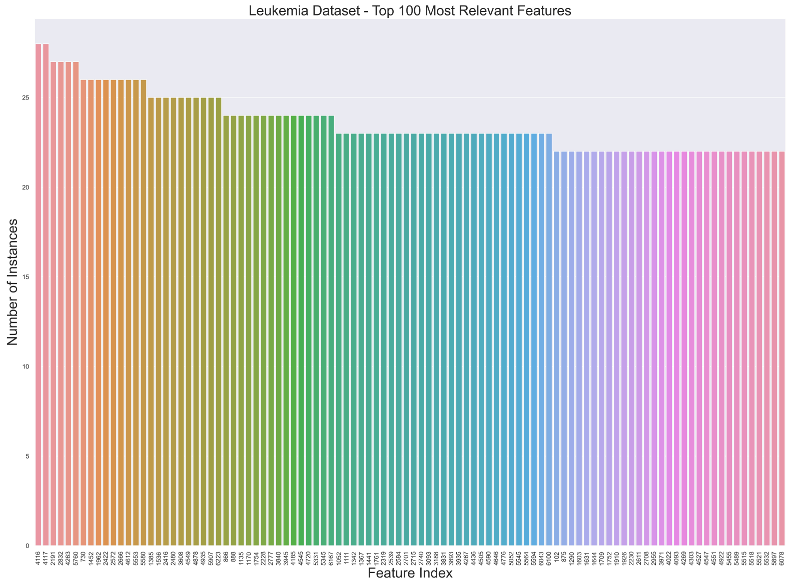


Figure C.7: The number of times each feature is chosen/selected using the LS technique on the FS step on the LOOCV procedure for the Leukemia dataset (n=72, d=7129). Showing the top-100 entries only.

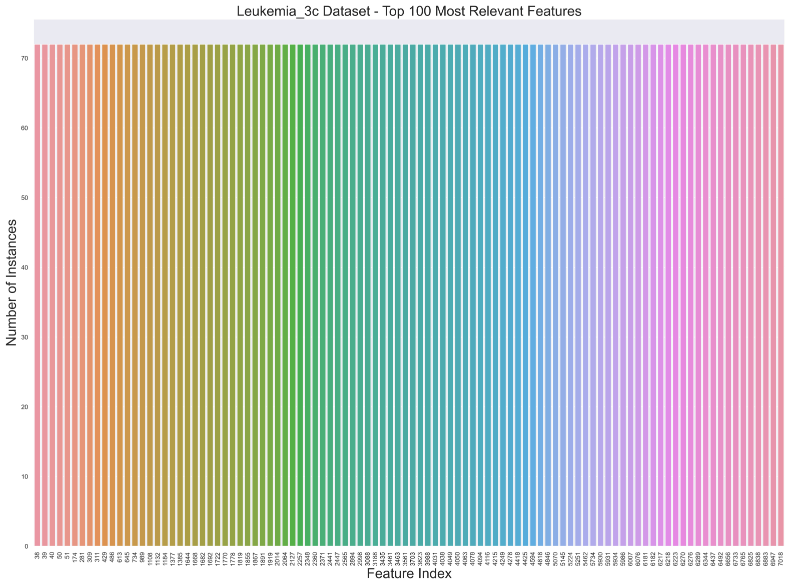


Figure C.8: The number of times each feature is chosen/selected using the RRFS(FiR) technique on the FS step on the LOOCV procedure for the Leukemia_3c dataset (*n*=72, *d*=7129). Showing the top-100 entries only.

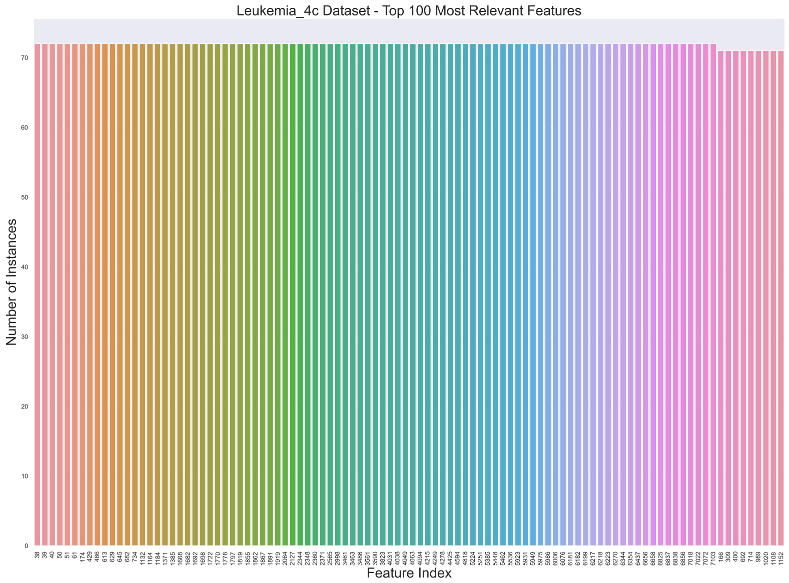


Figure C.9: The number of times each feature is chosen/selected using the RRFS(FiR) technique on the FS step on the LOOCV procedure for the Leukemia_4c dataset (*n*=72, *d*=7129). Showing the top-100 entries only.

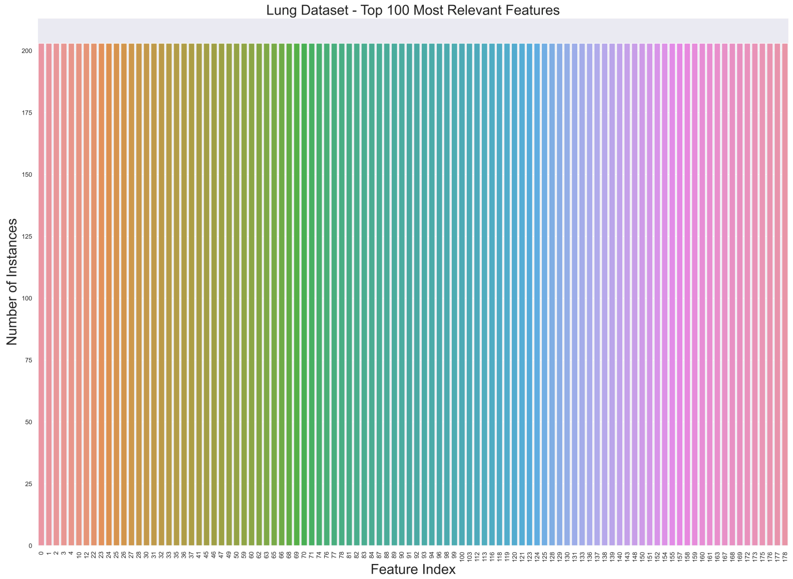


Figure C.10: The number of times each feature is chosen/selected using the FiR technique on the FS step on the LOOCV procedure for the Lung dataset (n=203, d=12600). Showing the top-100 entries only.

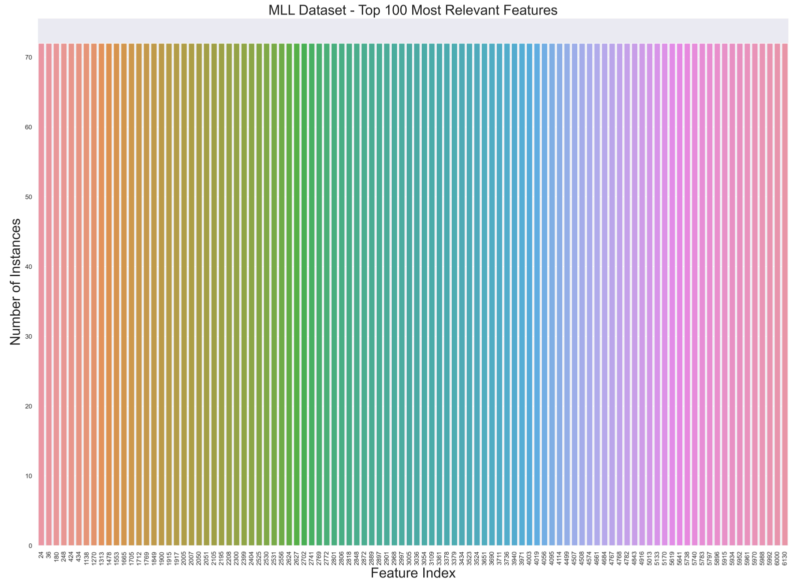


Figure C.11: The number of times each feature is chosen/selected using the RRFS(MM) technique on the FS step on the LOOCV procedure for the MLL dataset (n=72, d=12582). Showing the top-100 entries only.

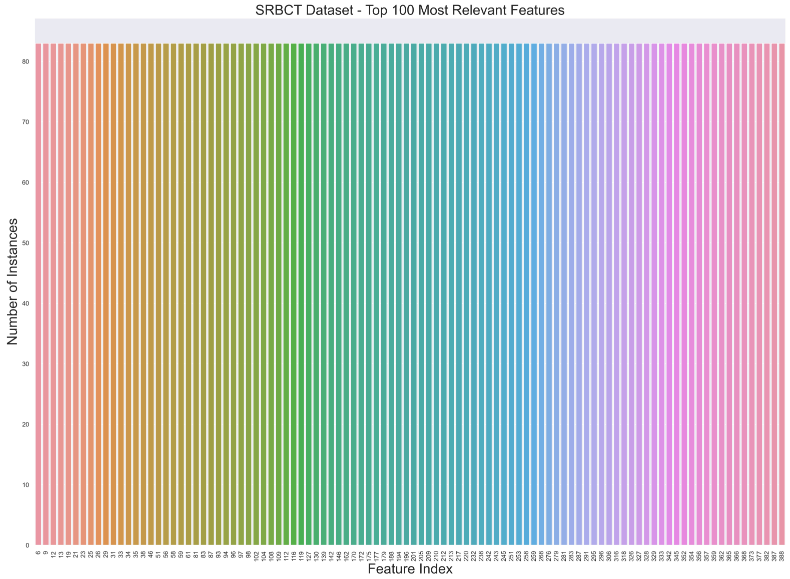


Figure C.12: The number of times each feature is chosen/selected using the SPEC technique on the FS step on the LOOCV procedure for the SRBCT dataset (*n*=83, *d*=2308). Showing the top-100 entries only.