

Masters' Degree in Informatics Engineering
Dissertation
Midterm Report

Personalization based on Grouping Strategies for Short-Term Cardiovascular Event Risk Assessment

Tânia Filipa Santos Marques
tmarques@student.dei.uc.pt

Supervisors:
Jorge Henriques
& Simão Paredes
July 2, 2013



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Here are the facts. Coronary artery disease is the leading killer of men and women in Western civilization. In the United States alone, more than half a million people die of it every single year. Three times that number suffer known heart attacks. And approximately three million more have “silent” heart attacks, experiencing minimal symptoms and having no idea, until well after the damage is done, that they are in mortal danger. In the course of a lifetime, one out of every two American men and one out of every three American women will have some form of the disease.

Caldwell B. Esselstyn Jr ¹

¹*Prevent and Reverse Heart Disease*, Avery, 2007.

Abstract

Cardiovascular Diseases (CVD) are the main cause of death in all Europe, representing 47% of all deaths. This is a major health problem which decreases the population life expectancy. It is also a financial and economic issue that costs to the European Union almost 196 billion euros every year. Studies indicate that some of those deaths are preventable if high risk patients (patients with a high probability of having a cardiovascular event in short-term, usually in less than 30 days), undergo early invasive strategies. This implies a need for a tool able to predict the patient's risk, which will help to determine the most adequate treatment option. The effectiveness of such a tool will be important in the treatment choice, because it can increase the probability of reducing the risk of cardiovascular events, and consequently, it can improve the patient's health, while decreasing the financial costs for the hospitals and governments.

There are several tools available, in the literature, which are able to predict that risk. They are called risk scores: risk prediction tools that calculate the patient's probability of having a cardiovascular event by using a fixed number of risk factors. However, it is not obvious which tool should be used in which situation, because there are several risk scores and their ability in terms of prediction is similar. This is one of the barriers to the integration of such a tool in the daily clinical practice, which is something that could potentially improve the life of millions of CVD patients.

The current risk scores also have some limitations: they work with a limited and static number of risk factors, they do not allow introduction of new clinical knowledge and they assume the population homogeneity. Considering that the patient population is in reality heterogeneous, building a unique system for all patients is rather difficult, which means that each risk score presents better results for patients with determined characteristics.

In this research, we combined well validated risk scores (TIMI, GRACE and PURSUIT) for short-term Coronary Artery Disease patients into a unique risk assessment tool. This combination was based on the idea that there is a heterogeneous population of patients. Therefore, assigning each patient to the most adequate risk score, may lead to a system with an overall better performance than using each risk score separately, while eliminating the need to choose between them. Furthermore, this would easily allow the incorporation of new knowledge, as risk scores, and increase the robustness to noise and missing data.

In order to determine the most adequate risk score for a patient, we used a personalization approach based on patients groups. Strictly speaking we divided the patients into groups, where each one has a risk score associated, and tried to find a way to assign a new patient to one of the groups and consequently to one of the risk scores. These groups can be determined using clustering algorithms. As an alternative, we can also consider that patients correctly classified with a risk score form a group, which would mean that we would have as many groups and the number of risk scores being used. After the groups are formed, we can generate rules to assign a new patient to a group, or use similarity

measures with the same purpose.

We validated this idea using a real dataset of Coronary Artery Disease (CAD) patients from Santa Cruz Hospital in Lisbon, which allowed us to verify that it could really improve the risk assessment when compared to the original risk scores. Using a subtractive clustering with a mixed distance (allows both interval-based and nominal data) to create the groups and generate the rules, we were able to obtain a sensitivity as good as the one obtained by GRACE (highest of the three risk scores used), while increasing its specificity by 11%. There was even a further improvement using similarity measures, when assigning weights to every risk factor. Its best solution was able to also preserve GRACE's sensitivity, and increase its specificity by 19%, which is 13% more than the one in PURSUIT (highest specificity of the three risk scores used).

Keywords: Risk Assessment, Risk Scores, Cardiovascular Diseases (CVD), Coronary Artery Disease (CAD), Group Personalization, Clustering, Similarity, Data Mining.

Resumo

As doenças cardiovasculares são a principal causa de morte em toda a Europa, representando 47% da taxa de mortalidade. Isto é um problema grave que diminui a esperança de vida da população, e é também um problema financeiro e económico que custa à União Europeia 196 mil milhões de euros todos os anos. Vários estudos indicam que algumas destas mortes podem ser evitadas se os pacientes com elevado risco (pacientes com elevada probabilidade de ter um evento cardiovascular a curto prazo, normalmente em menos de 30 dias) forem submetidos atempadamente a intervenções invasivas. Isto implica que é necessário a existência de uma ferramenta capaz de prever o risco do paciente, o que poderá ajudar a determinar o tratamento mais adequado. A eficiência deste tipo de ferramentas é muito importante, porque pode aumentar a probabilidade de reduzir o risco do paciente ter um evento cardiovascular, e consequentemente, pode melhorar a saúde dos pacientes, diminuindo simultaneamente os custos financeiros para os hospitais e governos.

Existem várias ferramentas disponíveis na literatura, que permitem prever esse risco. Estas são ferramentas que calculam a probabilidade de um paciente ter um evento cardiovascular através de um número fixo de factores de risco. Contudo, nem sempre é óbvio qual a ferramenta que deve ser usada numa determinada situação, porque existem várias ferramentas e a sua capacidade de predição é semelhante. Isto é uma das barreiras à integração destas ferramentas na prática médica, o que é algo que podia melhorar a vida de milhões de doentes com doenças cardiovasculares.

As ferramentas de avaliação de risco utilizadas actualmente têm algumas limitações: funcionam com um número limitado e estático de factores de risco, não permitem a introdução de novo conhecimento clínico, e assumem que a população é homogénea. Tendo em consideração que a população de pacientes é realmente heterogénea, a construção de um único sistema para todos os pacientes é bastante difícil, o que significa que cada ferramenta irá apresentar melhores resultados em pacientes com determinadas características.

Neste trabalho, nós combinámos ferramentas bem validadas (TIMI, GRACE e PURSUIT), que permitem avaliar o risco de pacientes com doença arterial coronariana a curto prazo. Esta combinação é baseada na ideia da existência de uma população heterogénea de pacientes. Se for assim, atribuir cada paciente à ferramenta mais adequada, pode levar a termos um sistema com melhor desempenho do que usando cada uma das ferramentas em separado. Além disso, isto permite a introdução de novo conhecimento, e aumenta a robustez do sistema a ruído e informação desconhecida.

Para determinar qual a ferramenta mais adequada para cada paciente, nós usamos uma abordagem de personalização baseada em grupos. Em outras palavras, nós dividimos os pacientes em grupos, onde cada um tinha uma ferramenta associada, e tentámos encontrar uma maneira de atribuir novos pacientes a um dos grupos, e consequentemente a uma ferramenta. Estes grupos podem ser determinados usando um algoritmo de agrupa-

mento (*clustering algorithms*). Como alternativa, nós podemos também considerar que os pacientes classificados correctamente com uma ferramenta formam um grupo, o que significaria que teríamos tantos grupos como o número de ferramentas usadas. Depois dos grupos estarem formados, nós podemos gerar um conjunto de regras que atribuirá um novo paciente a um grupo, ou então usar medidas de similaridade com o mesmo intuito.

Nós validámos esta ideia através de uma base de dados real, que contém os pacientes com doença arterial coronariana que estiveram internados no hospital de Santa Cruz em Lisboa. Isto permitiu-nos verificar que é possível, com a nossa abordagem, melhorar a avaliação de risco quando comparada com as ferramentas originais. Usando o algoritmo *subtractive clustering* com uma distância que permite tanto valores nominais como contínuos, conseguimos obter uma sensibilidade tão boa como a obtida pelo GRACE (valor mais elevado obtido pelas ferramentas usadas), e também aumentar a especificidade em 11%. Conseguimos ainda melhorar mais os resultados usando medidas de similaridade, quando atribuímos pesos a todos os factores de risco. A melhor solução, obtida assim, foi também capaz de preservar a sensibilidade do GRACE, e aumentar a especificidade em 19%, o que é 13% mais do que a obtida pelo PURSUIT (valor mais elevado obtido pelas ferramentas usadas).

Palavras Chave: Ferramentas de Avaliação de Risco, Doenças Cardiovasculares, Doença Arterial Coronariana, Personalização por Grupos, Agrupamento, Similaridade, Análise de Dados.

Contents

1	Introduction	1
2	Risk Assessment	5
2.1	Risk Scores	5
2.1.1	Thrombolysis In Myocardial Infarction (TIMI)	6
2.1.2	Platelet glycoprotein IIb/IIIa in Unstable angina: Receptor Suppression Using Integrilin Therapy (PURSUIT)	7
2.1.3	Global Registry of Acute Coronary Events (GRACE)	7
2.1.4	Comparative Studies	9
2.1.5	Disadvantages of Current Risk Scores	10
2.2	Similar Research	10
2.2.1	Parameters Combination	11
2.2.2	Output Combination	11
2.2.3	Group Personalization	12
2.3	Data Mining in Risk Assessment	13
2.3.1	Specific Requirements	13
2.3.2	Algorithms	14
3	Background	17
3.1	Clustering	17
3.1.1	Measures of Similarity and Similarity Algorithms	18
3.1.2	Clustering Techniques	19
3.1.2.1	Partitioning Clustering	20
3.1.2.2	Hierarchical Clustering	21
3.1.2.3	Density-Based Clustering	21
3.1.2.4	Grid-Based Clustering	22
3.1.2.5	Model-Based Clustering	22
3.1.3	Comparing the Techniques	23
3.2	Dimensionality Reduction	23
3.2.1	Linear Techniques	24
3.2.2	Non-Linear Techniques	25
3.2.2.1	Global Techniques	25
3.2.2.2	Local Techniques	26
3.2.2.3	Global Alignment of Linear Models	27
3.2.3	Comparing Techniques	27
3.3	Features Selection	27
3.3.1	Wrapper Model	28
3.3.2	Filter Model	29
3.3.3	Comparing Algorithms	30
3.4	Imbalanced Data	30

3.5	Validation	31
4	Methodology	35
4.1	Personalization based on Patients Groups	36
4.1.1	Clustering Patients	36
4.1.2	Dividing by Scores	37
4.1.3	Similarity Measures	39
4.2	Algorithms and Tools	40
4.2.1	Pre-Processing	40
4.2.1.1	Dealing with Missing Data	41
4.2.1.2	Normalizing	41
4.2.1.3	Discretization	41
4.2.1.4	Balancing Data	42
4.2.1.5	Dimensionality Reduction	42
4.2.1.6	Features Selection	43
4.2.2	Clustering Algorithms	43
4.2.3	Grouping by Scores	44
4.2.4	Similarity Measures	44
4.2.5	Selecting Scores	44
4.2.6	Generating rules	45
4.3	Tests Performed	45
4.4	Validation	46
4.5	Dataset	47
5	Results and Discussion	49
5.1	Clustering Patients	50
5.1.1	Subtractive Clustering	50
5.1.2	Fuzzy C-means	53
5.1.3	Decision Tree	55
5.2	Dividing by Scores	56
5.2.1	Subtractive Clustering	56
5.2.2	Fuzzy C-means	59
5.2.3	Decision Tree	60
5.3	Similarity Measures	61
5.4	Statistical Analysis	65
5.4.1	Sensitivity	65
5.4.2	Specificity	66
5.4.3	Gmean	68
5.5	Discuss Results	69
6	Conclusions	71
6.1	Future Work	72
7	References	73
	Appendices	81
A	Clustering Patients Results	83
B	Dividing by Scores Results	91
C	Similarity Measures Results	97

Abbreviations and Acronyms

AGNES	Agglomerative Nesting
AUC	Area Under the Curve
CAA	Cardiac Arrest at Admission
CAD	Coronary Artery Disease
CART	Classification and Regression Tree
CHD	Coronary Heart Disease
CVD	Cardiovascular Diseases
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DIANA	Divisive Analysis
EM	Expectation-Maximization algorithm
FCBF	Fast Correlation Based Filter
FCM	Fuzzy C-means
GMM	Gaussian Mixture Model
GRACE	Global Registry of Acute Coronary Events
HF	Heart Failure
LLC	Locally Linear Coordination
LLE	Local Linear Embedding
MI	Myocardial Infarction
OSS	One-Sided Selection
PCA	Principal Component Analysis
PURSUIT	Platelet glycoprotein IIb/IIIa in Unstable angina: Receptor Suppression Using Integrilin Therapy
ROS	Random Minority Oversampling
RS	Risk Scores
RUS	Random Majority Undersampling
SE	Sensitivity
SMOTE	Synthetic Minority Oversampling Technique

SOM Self-Organizing Maps

STE-ACS ST-segment elevation acute coronary syndromes

STING STatistical INformation Grid-based method

TIMI Thrombolysis In Myocardial Infarction

TIMI NSTEMI TIMI for Unstable Angina/Non-ST Elevation MI

TIMI STEMI TIMI ST-segment elevation MI

List of Figures

2.1	PURSUIT - Probability of Risk	8
3.1	Confusion Matrix	32
3.2	Statistic Methods	33
4.1	Clustering Patients Approach	37
4.2	Dividing by Scores Approach	38
4.3	Similarity Measures Approach	40
5.1	Statistical Analysis - Sensitivity	66
5.2	Statistical Analysis - Specificity	67
5.3	Statistical Analysis - Geometric Mean	68

List of Tables

2.1	TIMI Risk Score	6
2.2	PURSUIT Risk Score	7
2.3	GRACE Risk Score	7
3.1	Measures of Similarity	19
4.1	Clustering Patients / Dividing by Scores - Tests	45
4.2	Similarity Measures - Tests	46
4.3	Santa Cruz Dataset	47
4.4	Endpoints - Santa Cruz Dataset	47
5.1	Results - Risk Scores	49
5.2	Results - Perfect Division	50
5.3	Clustering Patients - Subtractive Clustering (Distances)	51
5.4	Clustering Patients - Subtractive Clustering (ROS)	51
5.5	Clustering Patients - Subtractive Clustering (RUS)	51
5.6	Clustering Patients - Subtractive Clustering (Dimensionality Reduction)	52
5.7	Clustering Patients - Subtractive Clustering (FCBF)	52
5.8	Clustering Patients - Subtractive Clustering (Gini index)	52
5.9	Clustering Patients - Subtractive Clustering (Relief-F)	53
5.10	Clustering Patients - Subtractive Clustering (Balancing + Dimensionality Reduction / Features Selection)	53
5.11	Clustering Patients - Fuzzy C-means (Balancing)	54
5.12	Clustering Patients - Fuzzy C-means (Dimensionality Reduction)	54
5.13	Clustering Patients - Fuzzy C-means (Feature Selection)	54
5.14	Clustering Patients - Fuzzy C-means (Balancing + FCBF)	55
5.15	Clustering Patients - Decision Tree	55
5.16	Results - Perfect Training	56
5.17	Dividing by Scores - Subtractive Clustering (Distances)	56
5.18	Dividing By Scores - Subtractive Clustering (ROS)	57
5.19	Dividing By Scores - Subtractive Clustering (RUS)	57
5.20	Dividing by Scores - Subtractive Clustering (Dimensionality Reduction)	57
5.21	Dividing by Scores - Subtractive Clustering (FCBF)	57
5.22	Dividing by Scores - Subtractive Clustering (Gini index)	58
5.23	Dividing by Scores - Subtractive Clustering (Relief-F)	59
5.24	Dividing By Scores - Subtractive Clustering (Balancing / PCA)	59
5.25	Dividing by Scores - Fuzzy C-means (Balancing)	59
5.26	Dividing by Scores - Fuzzy C-means (Dimensionality Reduction)	60
5.27	Dividing by Scores - Fuzzy C-means (Feature Selection)	60
5.28	Dividing by Scores - Fuzzy C-means (Balancing + PCA)	60
5.29	Dividing by Scores - Decision Tree	61

5.30	Similarity Measures - Distances	62
5.31	Similarity Measures - Dimensionality Reduction	62
5.32	Similarity Measures - FCBF	62
5.33	Similarity Measures - Gini index	63
5.34	Similarity Measures - Relief-F	63
5.35	Similarity Measures - Euclidean	63
5.36	Similarity Measures - Mixed	63
5.37	Similarity Measures - Hamming	64
5.38	Similarity Measures - Jaccard	64
5.39	Similarity Measures - Weights	64
5.40	Statistical Tests	65
A.1	Clustering Patients - Subtractive Clustering (Euclidean Distance)	83
A.2	Clustering Patients - Subtractive Clustering (Mixed Distance)	83
A.3	Clustering Patients - Subtractive Clustering (Hamming Distance)	84
A.4	Clustering Patients - Subtractive Clustering (Jaccard Distance)	84
A.5	Clustering Patients - Subtractive Clustering (PCA)	84
A.6	Clustering Patients - Subtractive Clustering (KPCA)	85
A.7	Clustering Patients - Subtractive Clustering (Relief-F / Euclidean Distance)	86
A.8	Clustering Patients - Subtractive Clustering (Relief-F / Mixed Distance)	86
A.9	Clustering Patients - Subtractive Clustering (Relief-F / Hamming Distance)	87
A.10	Clustering Patients - Subtractive Clustering (Relief-F / Jaccard Distance)	87
A.11	Clustering Patients - Fuzzy C-means (Clusters' number)	88
A.12	Clustering Patients - Fuzzy C-means (PCA)	88
A.13	Clustering Patients - Fuzzy C-means (KPCA)	88
A.14	Clustering Patients - Fuzzy C-means (Gini Index)	89
A.15	Clustering Patients - Fuzzy C-means (Relief-F)	89
A.16	Clustering Patients - Decision Tree	90
B.1	Dividing by Scores - Subtractive Clustering (Euclidean Distance)	91
B.2	Dividing by Scores - Subtractive Clustering (Mixed Distance)	91
B.3	Dividing by Scores - Subtractive Clustering (Hamming Distance)	92
B.4	Dividing by Scores - Subtractive Clustering (Jaccard Distance)	92
B.5	Dividing by Scores - Subtractive Clustering (PCA)	92
B.6	Dividing By Scores - Subtractive Clustering (KPCA)	93
B.7	Dividing By Scores - Subtractive Clustering (Relief-F)	93
B.8	Dividing by Scores - Fuzzy C-means (Clusters' number)	94
B.9	Dividing by Scores - Fuzzy C-means (PCA)	94
B.10	Dividing by Scores - Fuzzy C-means (KPCA)	94
B.11	Dividing By Scores - Fuzzy C-means (Gini Index)	95
B.12	Dividing by Scores - Fuzzy C-means (Relief-F)	95
B.13	Dividing by Scores - Decision Tree	96
C.1	Similarity Measures - PCA	97
C.2	Similarity Measures - KPCA	97
C.3	Similarity Measures - Relief-F	98

Chapter 1

Introduction

Cardiovascular Diseases (CVD), which include Coronary Heart Disease (CHD), stroke (brain attack), and Heart Failure (HF), represent the main cause of death in Europe, with 4 million deaths each year (approximately 47% of all deaths). In Portugal, for instance, this accounts for 12% of potential years of life lost due to preventable early death [1]. Moreover, this is more than a major health problem for the EU. It is also an economical and financial issue that costs approximately 196 billion euros every year.

In this work, we will focus on a specific cardiovascular disease: CHD, but it can also be generalized for other diseases. CHD, also called Coronary Artery Disease (CAD) is caused by the buildup of plaque in the arteries to the heart, which leads to their narrowing and consequently to the shortage of blood and oxygen supply [2]. CAD patients may suffer from cardiovascular events, particularly myocardial infarction (heart attack), and eventually death. The use of predictive and preventive medicine may help avoid at least a small percentage of such cases, which may still represent thousands of lives saved, and reductions on the health costs.

In preventive medicine, prevention can be classified as primary, secondary or tertiary [3]. Primary prevention consists in protecting peoples' health in order to prevent them from developing a disease. This usually includes educating people about risk factors and monitoring their health using tests and examinations. Secondary prevention only occurs after the disease is diagnosed. Its goal is to halt or slow the disease progress by using medication or early intervention, for instance, in order to decrease the number or intensity of cardiovascular events. Tertiary prevention is mainly about maximizing the patient's quality of life in spite of the long-term disease consequences and complications, as for example, cardiac or stroke rehabilitation programs.

On this research, we will focus on secondary prevention of CAD patients, which means that once a patient is diagnosed with this disease, an appropriate treatment should be determined to slow down its progression, and prevent a cardiovascular event. This can be done by assessing the risk of having such an event in short-term. This is necessary to help determine the correct treatment for each patient, in order to avoid a potential cardiovascular event, or decrease its consequences.

There are two main categories of treatments for CAD patients: pharmacological, which includes antithrombotic therapies, such as antiplatelet and anticoagulant agents; and invasive strategies like coronary angiography and revascularization [4]. However, evidence suggests that the effectiveness of these methods depends of the patient's risk. If the CVD patient has a high or intermediate risk of having a cardiovascular event, then using

invasive strategy decreases more the chance of a fatal event than using a pharmacological one. In contrast, low risk patients do not benefit of using such an aggressive treatment as the former, obtaining similar results with both methods [5].

Deciding which treatment should be used for which patient, may not be a trivial task. Invasive strategies are usually more effective. However, it is preferable to avoid using it in patients unless completely necessary, because it is a complicated procedure that sometimes has undesirable consequences like related heart attacks and increased risk of bleeding. Therefore, it should only be used when the patient has a risk high enough that this method's effectiveness overcomes its risks. In financial terms, invasive strategies are also impossible to use in every patient, because it is expensive, and the governments cannot afford such an expense. Consequently, it is important to divide the patients into categories: high risk patients, who need invasive intervention for their survival and low risk patients, who may be treated with pharmacological therapies without endangering their health. This can be done by correctly assessing the patients' risk.

However, risk assessment should not be done using only clinical judgment, because this leads to substantial difference in treatments depending on the physician [6]. This is due to clinical judgment being dependent of the physician's individual experience and knowledge, which may contain incongruences. This hampers the decision making, and leads to different outcomes [7]. On the other hand, modern hospitals have large information systems that contain all the data needed for a data mining algorithm to learn the rules necessary to help physicians. This kind of thought lead to the creation of several approaches that can do risk assessment in a more homogeneous way.

Some of those approaches were small studies, with a very limited number of patients, that were never used in clinical practice, for example [8] and [9], which the sole intention was to explore new ways to predict the risk. On the other hand, big studies were also conducted in order to develop well validated tools that can help the physicians to do a better cardiovascular risk assessment. These tools are called Risk Scores (RS) and can be divided in long term prediction (one or more years) and short term prediction (some months). Basically, RS are risk assessment tools that given a number of risk factors can determine the probability of an undesired event or disease. They can also be classified according to the kind of prevention, particularly primary and secondary prevention.

In the case of secondary prevention for Coronary Artery Disease, usually the RS purpose is short term prediction. Among those RS, some of the most known are: Thrombolysis In Myocardial Infarction (TIMI) [10], Global Registry of Acute Coronary Events (GRACE) [11], and to a smaller extent, Platelet glycoprotein IIb/IIIa in Unstable angina: Receptor Suppression Using Integrilin Therapy (PURSUIT) [12]. Their use is even recommended by the European Society of Cardiology [13].

However, it is not always easy to choose among RS, because comparative studies don't have enough evidence to justify that one is preferable to the other. This is further aggravated by the fact that building and validating a risk score requires a lot of time and money. Therefore, it would be useful to find a way to improve their accuracy, without creating a new one. This could be achieved by combining the current RS, because, as stated by Tsymbal et al. [27] and Bauer et al. [28], an ensemble of classifiers can be more accurate than a single classifier.

Combining the RS would allow us to obtain a more effective risk assessment tool than any one of the risk scores alone, and would also bring other advantages such as removing the need to choose a *standard* among the RS and increase its robustness towards missing values

and noise. Additionally, new well validated information could be easily incorporated to the system, as a risk scores, without further costs.

We opted to combine the risk scores using a personalization approach¹ inspired on the work developed by Paredes [32], which selects the best classifier to a group of patients. This is based on the idea that there are similarities between the patients that are correctly classified by a risk score, and if we were able to find them, then we could use the most adequate RS for each patient, which would improve the risk assessment.

In our research we propose three different approaches to achieve that:

- **Clustering Patients approach:** This is the most similar to the one proposed by Paredes [32]. We build the groups using clustering algorithms, and then assign a risk scores to each one of the groups. Afterwards, a new patient is classified in one of the groups and consequently in one of the risk scores.
- **Dividing by Scores approach:** As groups we consider the patients that are correctly classified with each one of the risk scores. Thus, we only need to find a set of rules to assign a new patient to one of the groups.
- **Similarity Measures approach:** In this approach we also divide the training patients using the risk scores, but instead of using a classifier to assign a new patient to one of the groups, we use similarity measures to find the group that contains the closest patient to the new one.

In each approach we also tested different data mining algorithms which could improve the results such as balancing methods, dimensionality reduction, and feature selection techniques. However, during that process we always kept in mind that our main goal was to confirm that combining risk scores using personalization by groups could lead to a more effective system.

In the Risk Assessment (section 2), we present in detail the different risk scores and the approaches that exist in the state of the art to improve their results. Furthermore, a brief review of data mining and its use in risk assessment is made. This is complemented by the Background (section 3), where the algorithms and techniques used in this work are presented. In the Methodology (section 4) we describe the approaches proposed and the work done, and in the Results and Discussion (section 5) the results obtained and respective discussion are depicted.

¹Personalization as defined in this work is not individual personalization, but group personalization, expressly, we are trying to adequate more the risk assessment to the individual, by using groups of patients with similarities.

Chapter 2

Risk Assessment

Our main goal is to combine risk scores to obtain a personalized risk assessment tool applicable to different patients group, so it is necessary to be aware of the risk scores that are currently applied, and what was already done by other researchers in this area. To achieve that we review in this section how data mining was used in the specific problem of risk assessment.

2.1 Risk Scores

When patients go to an hospital, they expect the diagnosis and treatment to be the same independently of the physician that will treat them. However, this may not always be the case. There is a lot of variance in physician practice, that will depend of his/her experience, knowledge, previous decisions and outcomes. A more detailed explanation of why this happen can be found in [7]. It seems obvious by this account, that medical judgment, by itself, will not always correspond to the best outcome. This lead medical societies to create clinical practice guidelines which indicate how to diagnose and treat certain diseases. This is also true for CVD. In Europe, the European Society of Cardiology has several guidelines to help physicians making medical judgments [14]. This is an attempt to homogenize the clinical practices, while increasing their reliability and trustfulness.

The European Society of Cardiology suggests in its guidelines the use of risk scores, which can, according to them, avoid under and overtreatment [14]. However, often which one to use is not clear. Even when guidelines are specific, it is not certain that they will be followed. Manfrini and Bugiardini [6] give an example where physicians were asked to follow the guidelines and assess the risk of the CVD patients. Those guidelines suggested that catheterization should be used in high risk patients. However, physicians only treated 70% of the patients considered high risk with catheterization. This points to the fact that physicians do not always trust the guidelines and risk scores, choosing to use their judgment most of the time. This, maybe, can be changed by creating more effective and reliable risk scores.

Independently of the barriers to the risk scores adoption, they may contribute to a more reliable clinical practice. Risk scores can be divided in two main groups according to the type of prevention [15]: primary prevention (assess the risk of having a certain disease, in this case CAD) and secondary prevention (knowing that a patient has a disease, we want

to predict the risk of having an event associated with it, like Myocardial Infarction (MI or death). In this work, we will only focus in secondary prevention of CAD.

In simple terms, a risk score is a risk assessment tool where a number of risk factors are considered, and according to their value, or existence, a mathematical formula is used to calculate a score that will give the probability of a particular outcome to occur during an established period of time. There are basically short term (months) and long term (years). In the case of CAD, the disease is very severe, which means that prevention is usually in terms of days or months, such as ≤ 30 days (short term risk scores). These scores can also be classified into quantitative (use a formula that produces continuous results) and semiquantitative (define a score for the absence or presence of a risk factor and the risk is given by their sum) [4]. Usually, the latter are easier to apply in practice, but theoretically are less precise than the former.

In this work, we chose to use and present the most popular short term risk scores used for CAD prevention. These turned out to be: TIMI [10], GRACE [11], and to a smaller extent, PURSUIT [12], which were referred in several papers in the literature, where [4], [16], [17], [18] and [19] are just some examples of such. Below, we briefly explain each one of those risk scores.

2.1.1 Thrombolysis In Myocardial Infarction (TIMI)

TIMI for Unstable Angina/Non-ST Elevation MI (TIMI NSTEMI) was created by Antman et al. in 2000 [10]. It was developed using 1957 patients assigned to receive unfractionated heparin from TIMI 11B. In the development, a total of 12 baseline characteristics were calculated, and their importance was assessed by use of multivariate logistic regression, which led to the creation of a risk score with 7 risk factors of equal magnitude as input. Then it was validated using 3 separate groups of patients (total of 5124 patients): the enoxaparin group from TIMI 11B (1953), the unfractionated heparin group from ESSENCE (1564) and the enoxaparin group from ESSENCE (1607).

TIMI is a semi quantitative tool, where the score is given by the sum of the number of the risk factors a patient presents, in other words, every one of the 7 risk factor/input has a score of 1. This makes it a very simple score to apply without the need of a calculator. Nevertheless, it is possible to find the TIMI risk score calculator in <http://www.timi.org/>. TIMI is more accurate than a qualitative assessment of risk, but theoretically may not be as accurate as a quantitative risk score.

It was initially, developed for predicting the risk of death, MI and urgent revascularization within a period of 14 days, but has been proved to have good results for both 30 days and 1 year [17]. In table 2.1 is possible to find a representation of this risk score.

TIMI Risk Score (0-7)						
Age ≥ 65 years	1					
≥ 3 Risk Factors for CAD	1					
Known CAD (stenosis $\geq 50\%$)	1					
ASA Use in Past 7 days	1					
Severe angina (≥ 2 episodes w/in 24 hrs)	1					
ST changes ≥ 0.5 mm	1					
Elevated Cardiac Marker	1					
Score	0/1	2	3	4	5	6/7
Risk (%)	4.7	8.3	13.2	19.9	26.2	40.9

Table 2.1: Contains the risk factors of TIMI and their respective scores, and the risk associated to each score.

There is also the TIMI ST-segment elevation MI (TIMI STEMI) [20]. However, we did not use it in this research, because our focus was non-ST segment elevation patients.

2.1.2 Platelet glycoprotein IIb/IIIa in Unstable angina: Receptor Suppression Using Integrilin Therapy (PURSUIT)

PURSUIT was created by Boersma et al. in 2000 [12] using the 9461 patients enrolled in the PURSUIT trials. Employing a univariate and multivariate logistic regression analyses, they established the relation between the baseline characteristics and the occurrence of death or non-fatal MI and just death over the period of 30 days. It resulted in a rather complex quantitative risk score, which is the reason why they presented a more simplified model with the most important risk factors.

In table 2.2 is possible to see the PURSUIT risk score for mortality and MI, which is the one that will be used in this work. The graph in figure 2.1 help us convert from the score to a probability or risk of happening death or MI in the next 30 days.

PURSUIT Risk Score (0-18/20)	
Age decade	
50	8(11)
60	9(12)
70	11(13)
80	12(14)
Sex	
male	1
female	0
Worst CCS-class in previous 6 weeks	
No angina or CCS I/II	0
CCS III/IV	2
Signs of heart failure	2
ST-depression on presenting ECG	1

Table 2.2: Risk factors of PURSUIT for mortality and MI and their respective scores. There are separate points for enrollment diagnosis of UAP and MI (between parentheses).

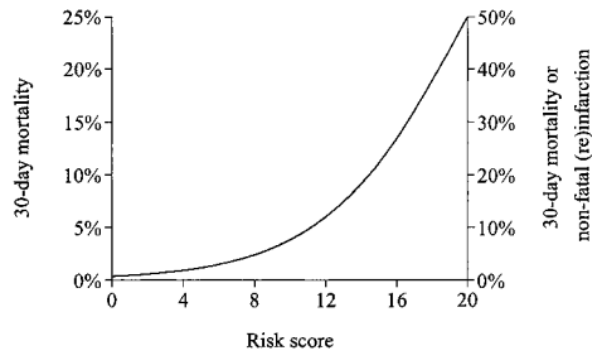


Figure 2.1: This graph can help convert the PURSUIT score into a probability of a cardiovascular event.

2.1.3 Global Registry of Acute Coronary Events (GRACE)

GRACE is a risk score created by Granger et al. in 2003 [11]. It was developed using a multivariate logistic regression in 11389 patients with and without ST-segment. In contrast to TIMI and PURSUIT, those patients were not part of trials, but registries of 94 hospitals located in 14 countries. Its validation was done using 16114 patients both from GRACE and GUSTO.

Initially it was developed to predict all-cause mortality in hospital over a period of 6 months. Later, additional work has been developed to validate it for other purposes. In [21] Keith et al. validated the risk score to be used from hospital admission to six months after discharge. In [22] the scores were updated to allow prediction within six months following discharge. Actually, there is an updated version, still unpublished, that can be found in [23], and that predicts death or death/MI at admission both in hospital and to 6 months. There is also a tool for discharge. GRACE was originally used for long-term prediction (6 months), however, several studies ([16],[18]) proved that it is also a good predictor for short-term, and therefore it can be used instead of TIMI or PURSUIT.

GRACE is not as simple as TIMI and PURSUIT, requiring specific tools to its calculation, which can be found in [23]. In table 2.3 it is possible to find the scores from GRACE risk score for predicting death or MI in hospital at admission, that are available in its site. A patient is considered to be of high risk if has non ST-segment elevation acute coronary syndromes (STE-ACS) and a score above 141, or if it has STE-ACS and a score above 155.

GRACE Risk Score (0-528)	
Age (years)	
< 40	0
40-49	15
50-59	29
60-69	44
70-79	59
80-89	73
> 90	80
Heart Rate (bpm)	
< 70	0
70-89	6
90-109	12
110-149	21
150-199	32
≥ 200	41
Systolic Blood Pressure (mmHg)	
< 80	57
80-99	53
100-119	43
120-139	34
140-159	24
160-199	10
≥ 200	0
Creatinine (mg/dL)	
0-0.39	2
0.4-0.79	5
0.8-1.19	8
1.2-1.59	11
1.6-1.99	14
2.0-3.99	23
≥ 4	31
Killip class	
I	0
II	33
III	67
IV	100
Cardiac arrest at admission	98
ST-segment deviation	67
Elevated cardiac enzymes/markers	54

Table 2.3: Risk factors of GRACE for predicting death or MI in hospital at admission.

2.1.4 Comparative Studies

There are some obvious differences between these scores. TIMI and PURSUIT only used clinical trials in their development, which theoretically may not represent the reality, while GRACE used data from hospitals registries, which has usually more patients and is closer to clinical practice. On the other hand, GRACE may be harder to adopt in real world than TIMI which can be calculated without use of any tool. But, in terms of performance it is not easy to say which one is the best.

Several comparative studies were conducted in different countries and with different patients, in order to evaluate the performance of the different scores. Here are some of the results:

- A study conducted in Portugal [17] concluded that GRACE had better predictive accuracy for predicting death or MI at 1 year with Area Under the Curve (AUC) of 0.715 (CI: 0.672 - 0.756), followed by PURSUIT with AUC: 0.630 (CI: 0.584 - 0.674), and TIMI with AUC of 0.585 (CI: 0.539 - 0.631). They also concluded that by using GRACE and PURSUIT to stratifying patients, it was possible to see a statistical significant benefit of myocardial revascularization for high risk patients, which did not happen with TIMI. The authors also give emphasis to the fact that all variables of GRACE are objective, which can be seen as advantageous;
- For the English and Welsh population [16], both PURSUIT and GRACE (in hospital and 6-month) presented similar performance for predicting 30-days mortality, but had difficulties dealing with higher-risk subgroups;
- A study in Scotland [18] compared TIMI and GRACE for predicting MI and all-cause mortality at 30 days, and concluded that there was not statistically significant differences in accuracy. Therefore, since GRACE is more complex, then TIMI may be seen as a better approach;
- A Canadian study [19] compared GRACE, PURSUIT and TIMI for in hospital death and 1 year, and concluded that GRACE and PURSUIT performed significantly better than TIMI, but all had good discriminatory potential;
- Another study in Canada [24] compared PURSUIT and GRACE, concluding that both had similar good discrimination for in hospital mortality, in spite of the fact that PURSUIT presented an overly poorer calibration, with an overestimation of the risk.

Analyzing the above results, it seems that all the scores have a similar performance, with TIMI obtaining slightly worse results in some cases. Furthermore, there is no clear candidate as a tool to really use in practice. This explains why in its guidelines, American College of Cardiology [25] recommends the use of one of the scores without specifying which. On the other hand, the Royal College of Physicians in UK [26] recommends GRACE and PURSUIT over TIMI, but affirm that there is insufficient evidence to allow a strong recommendation about which score would be most appropriate.

2.1.5 Disadvantages of Current Risk Scores

Risk scores have a number of advantages: they increase the accuracy of prediction, they homogenize the clinical practice and they help prevent over and under treatments. However, they also have a number of disadvantages. Here we present some of those:

- It is necessary to choose only one risk score, and there are not studies that support a well informed choice.
- Each risk score has a limited and static number of risk factors, and does not allow the introduction of new clinical knowledge.
- All the risk scores assume the existence of all the data needed to calculate the score. However, when a patient is admitted to a hospital and a clinician wants to assess the risk, this usually is not true.
- They were created from a group of patients, sometimes selected for trials or from registries, and will be applied to all kinds of patients without regarding specific differences between them.
- When a risk score is developed, the knowledge obtained from the previous scores is ignored and not incorporated in this new study.

All these factors could be improved in order to create better risk score. In this work we try to decrease somehow these drawbacks. We opted for a personalization based on groups approach where the patients are divided in clusters and only the score with best performance for this cluster is applied. This is basically a way to combine the risk scores, while trying to create a more patient directed model, instead of considering the population as an undefined mass. This will eliminate the need to choose a sole risk score, will allow new clinical knowledge to be introduced to the model, while maintaining the previous knowledge. Furthermore, it will improve the capacity to deal with missing values, because different scores have different risk factors.

In order to create this personalized scheme is important to analyze the data from patients and create the clusters needed to choose the best risk score for each one. This analysis should take in consideration the data mining algorithms that exist in the literature and the techniques that can improve its use.

2.2 Similar Research

In the above section (2.1), we briefly explained the most used risk scores for risk assessment of patients with known CAD disease. Each risk score can be seen as a classifier that assigns to each patient a score representing his/her risk of having a cardiovascular event. There are various authors ([27], [28]) who defend that with an ensemble of classifiers for the same problem is possible to obtain a more accurate result than with a single classifier. Based on this concept many works were developed by the scientific community, which proposed methods to combine several classifiers into a more accurate single classifier. Some of those works were dedicated to the combination of risk scores.

Combining risk scores presents several advantages in comparison with using a single risk score: it removes the need to choose a *standard*, it improves the results without creating new classifiers, previously well validated information is incorporated as a risk score, and it increases the robustness of the model towards missing values and noise.

In the available literature, there are two different approaches to the combination of classifiers: parameter combination and output combination. In this section we describe some of the works done for each approach.

2.2.1 Parameters Combination

Classifiers can be combined using their parameters. Although this is an approach less used in the literature than the output combination, it is possible to find several examples of it.

Samsa et al. [29] proposed a general linear regression strategy to combine risk factors from different datasets, in order to merge individual models into a multivariate risk model. They affirm that many diseases have numerous risk factors, and most of studies create models limited to only a small number. Using this strategy it is possible to obtain a more complete and robust model for identifying high-risk group of individuals.

Steyerberg et al. [30] used meta-analysis to combine univariate information from the literature (publicly available knowledge) with univariate and multivariate results from individual patient data (a dataset available to the researcher). This strategy explicitly incorporates known literature data, into the developed model, which improves the prediction of individual patients.

In order to create a medical expert system for estimating risk of CHD within 10 years, Twardy et al. [31] used Bayesian networks as common approach to combine published epidemiology models (Busselton, PROCAM) with clinical expert knowledge. The latter was only used when the interpretation was dubious. This approach has a graphical structure easy to understand, allows missing data, and can easily incorporate additional data or expert knowledge.

Paredes [32] also presented a methodology based on parameters combination using Naïve Bayes as a common representation. As in [31] this choice was deemed appropriate because of the algorithm's interpretability and ability to deal with missing values, while presenting a competitive performance when compared with other similar algorithms. Since the structure of this Bayesian classifier is completely defined, in order to represent a risk score in this form, the algorithm only needs to train it with a training dataset and the labels obtained by running the original risk score. Then the classifiers obtained can be combined using parameters/data fusion. Because the outputs of individual models are the same, it becomes easy for Paredes to just create a global naïve bayes representation whose parameters were derived based on the individual Bayesian models. To improve the model even further, a genetic algorithm's optimization is used to adjust the probabilities that are parameters to the global model.

2.2.2 Output Combination

One way of combining several classifiers is output combination. This can be done by voting methods (final result is based on the votes of individual models) or by selection methods (one of the classifiers is selected as the most adequate).

Voting methods may be the simplest approach for classifiers combination. All classifiers present a result to an instance, and the final results could be, for example, the class with more votes. More complex processes can also be found in the literature. Tsymbal et al. [27] used a weighted voting, where the weights were the reliability of each classifier. Cordela et al. [33], on the other side, proposed a method where the weights were dynamically adjusted to the particular characteristics of each instance.

In spite of its simplicity, voting methods may not be the most adequate for most of the

cases. They assume that each classifier has the same characteristics and generates the same range output. In some cases, like in [28], individual classifiers are even generated on purpose, in order to create a more accurate classification. In our case, there are several classifiers to solve the same problem. However, those classifiers are very distinct and so are their outputs. Combining their outputs using a weighted voting wouldn't be reasonable, because their ranges differ and changing scales could lead to information loss. Using a pure voting system (best result is the one with more votes) may not improve the result as much as we would like it to.

Inside the output combination approach there exist also the selection methods. Instead of using all the classifiers to produce the final result, they select the most adequate one (the one who will probably be the most accurate). The selection can be done using different methods. Zhang [34] in his thesis identified the best classifier using minimization of an information criterion. Todorovski et al. [35] preferred a dynamic selection, which assigned an individual classifier for each test instance. This was done using Meta-level decision trees, which were similar to regular decision trees but would choose the classifier for each instance, rather than classifying them.

The personalization based on groups of patients approach proposed by Paredes [32] can also be seen as an output combination using selection methods. Instead of selecting a classifier for each test instance, it groups the instances into clusters, and selects the best classifier for each group. Because, our approach was inspired by this methodology, we will briefly explain it in the next section.

2.2.3 Group Personalization

In Paredes's [32] work, the parameter combination methodology did not achieve good results in terms of specificity, which lead him to propose an alternative approach based on group personalization.

This methodology as proposed by Paredes has two steps: dimensionality reduction and clustering. In the first step, in order to reduce the dimensionality, he represents each patient data with an array containing the risk probabilities obtained from each risk scores used in this approach. After, reducing the data, subtractive clustering is applied to it, forming clusters/groups of patients with similar characteristics. For each cluster, the most suitable risk score is going to be selected using G_{mean} ¹.

In the results obtained by Paredes [32], this methodology was shown to have higher sensitivity than individual models, without reducing the specificity, which was a problem of the first methodology. This supports our idea that a personalization by groups may have higher performance than the individual risk scores.

We believe that it is possible to explore more this idea than was already done, with that purpose we chose different methods of personalization, and different techniques for treating the data, in order to improve even more this methodology.

¹ $G_{mean} = \sqrt{SE \times SP}$, where SE is the percentage of positive labeled instances correctly predicted and SP is the percentage of negative labeled instances correctly predicted.

2.3 Data Mining in Risk Assessment

Nowadays, hospitals are equipped with information systems that gather large quantities of data every day. These data contains lots of useful information ready to be discovered, which can be done using data mining algorithms. “Data mining is concerned with the analysis and extraction (discovery) of medical knowledge from data, aimed at supporting diagnostic, screening, prognostic, monitoring, therapy support or overall patient management tasks” [36]. However, most of the studies are still focused on diagnosis, since the medicine in hospitals is mostly reactive, in other words, patients are only treated for diseases after the symptoms begin. Nevertheless, data mining has a stronger potential than that, it can be used for predictive and preventive medicine.

Preventive medicine has been supported mainly by recent discoveries that lead researchers to believe that using information systems for processing genetic material can create a new form of medicine, where diseases can be individually predicted a long time before they occur. However, further research must be done, before this happen. On the other hand, there is another form of preventive medicine that has become more prominently lately that consists on the continuous monitoring of patients during their daily life. This is especially important for diseases where the first noticeable symptoms, may come too late, which is the case of cardiovascular diseases. One example of such a preventive system, is MyHeart, in Europe, which develops smart electronic and textile systems and services that allows a continuously monitoring of patients with CVD [37].

Our work has a focus on preventive medicine by creating a risk assessment tool and personalizing it by group of patients. In this section, we present some works also dedicated to preventive medicine, in particular to risk assessment in CVD patients.

2.3.1 Specific Requirements

Data mining can be very important to create a better health system based on predictive and preventive medicine solutions. However, in order for that to happen it is important to understand what some of the most important requirements of this field are, because they can improve the likelihood of those systems being accepted by the physicians. Below, we state some of the requirements found on several studies ([38], [39] and [40]).

- **Dealing with missing data:** In medicine is usual to encounter missing data. It can be unknown information from the patients, or else it can be from tests that were not done. This is a hard problem, when the number of such cases is very large, and they contain information needed to differentiate the classes. Risk scores also suffer from this problem, because missing data reduces their performance, since assume the presence of all risk factors. However, when it is urgent to assess the risk of patient, it may not always be possible to obtain all the information needed. Our approach decreases the severity of this problem;
- **Dealing with noise:** Data that comes from medical devices may contain error and uncertainty. It is necessary to pre-process it to decrease its effect on the data mining algorithm.
- **Interpretability:** Physicians are more likely to accept a system that they can understand how it works, and how it gives its results, in other words, a system

defined in such a way that is possible to interpret it. For example, a decision tree is easier to interpret than a neural network;

- **Integrate clinical knowledge:** A lot of data mining systems fail to integrate clinical knowledge. It will be very difficult for these systems to be accepted in real practice, even if they have a significant increase in accuracy;
- **Report probabilities and confidence interval:** Whenever possible, present the probability and confidence interval of a result. This will increase acceptance from physicians.

Besides, the requirements above, there is one particular characteristic of medical data, that while may not always be a problem, it is important to be aware of: medical data is often imbalanced. In other words, usually we have a lot of persons (instances) without a disease and only few with it. In section 3.4 we explain in more detail this problem and the solutions found in literature.

2.3.2 Algorithms

For using a data mining in an effective way, it is necessary to understand the algorithms available in the literature and what are their different advantages and disadvantages. In this section we will briefly review those algorithms.

There are two main classes of algorithms. There are the supervised algorithms, which receive data labeled into different classes, that is, they receive a training dataset with the input and the respective output. This means that these kinds of algorithms know when they are classifying correctly or not and its learning method may use evaluation measures like accuracy or entropy. On the other hand, there is the unsupervised algorithms, which deal with unlabeled data and have to make associations and learn the classes from the training data without knowing if it is correct or not. This is usually done by some kind of measure of similarity between the instances. For example, the risk scores are obtained using a supervised algorithm, because they use a set of patients with risk factors (input) and the time it took them to have a cardiovascular event (represents the risk of an event, the label). On the other hand, our approach pretends to divide patients in different groups and apply the best risk score for them, but we do not know the number of groups or what groups exist. Therefore, this will require an unsupervised algorithm that can create group using some similarity measure.

In medicine, it may be useful to do a further classification of the algorithms in “black box” and “white box”. A “black box” algorithm is an algorithm, which can be viewed as an opaque device or system to which we give some input and it gives back an output. However, there is no clear explanation how that works, and how the results are generated, and mostly important how to justify the outputs from the inputs given. This goes clearly against one of the requirements of data mining in medicine: interpretability. Physicians want to be able to interpret the results, which means that whenever possible, we should use “white box” algorithms, that unlike “black box” ones, can be interpreted and generate tools that justify the outputs from the inputs given.

In order to understand the different algorithms in the literature and their uses, we will briefly explain the most representative, while indicating how they can be classified and how they have been used in medicine, particularly in risk assessment and in cardiovascular diseases.

- **Decision Trees:** It is the classical example of a white box algorithm and a supervised one. It receives a dataset of labeled training data and uses a divide and conquer approach to create a tree, which will give the classification of a new instance. Basically, in each iteration, the algorithm looks at data, and if it all belongs to the same class, creates a leaf with that class, otherwise uses a measure, like entropy, to choose one attribute that will branch the data according to that measure. The result is a decision tree where each branch is a decision based on one of the input attributes, and the leafs are the final classes, or classifications. The most popular approach is C4.5 [41] which uses the information gain and gain ratio to branch the data. The main advantage of this algorithm, it is the interpretability and the generation of a tree that can be used without support system. On the other hand, it does not deal well with noise and missing data. In terms of practical use in cardiovascular diseases: Tsien et al. [42] compared the performance of decision trees and logistic regression for diagnosis of MI, and concluded that they performed equally well, and Karaolis et al. [9] used decision trees to extract the most important risk factors for some cardiovascular events, like MI.
- **Bayesian Networks:** It is also a “white box” and a supervised algorithm. In essence, a Bayesian network is a statistical model represented by a graph with the different attributes and output as nodes and their interdependencies as connections. The output is given by the probability of being of a certain class, knowing the conditional probabilities of the different values of the attributes, which are calculated using the data. Each Bayesian classifier implements a different configuration. The Naïve Bayes classifier, in specific, is a simple configuration that assumes the attributes independence. Regardless, of its simplification, it is a very powerful algorithm, that can have performances comparable to more complex algorithms [39]. Besides, it can easily deal with missing data, due to its statistical nature and independence assumption. This algorithm was used as the base algorithm to conjugate the risk assessment tools for CAD in [32], because of its interpretability, simplicity and capacity to deal with missing data.
- **Neural Networks:** A very popular algorithm, due to its accuracy, but also a typical example of a “black box” approach. A simple neural network is usually composed by 3 layers: the input, the output and one or more hidden layers. In each iteration, the output and error (supervised algorithm) are calculated using the input and expected value, respectively. This error is then back-propagated, and the weights updated using, for example, the gradient descent. There are also some unsupervised neural networks, like Self-Organizing Maps (SOM). Despite, the difficulty of interpretation of neural networks, they seem to be quite popular in medicine. Evidence indicates that 70% of reported studies for cancer prediction use neural networks as the main predictor [43]. Voss et al. [44], for example, compared the prediction power of neural networks and logistic regression for predicting MI or death, and concluded that neural networks perform better. In contrast, the study by Combolet et al. [45] indicate that the performances of both types of algorithms are similar for CAD assessment risk. The unsupervised neural networks are less popular, but it is possible to find some works for risk assessment, like the study by Churilov et al. [46], where they improve the risk grouping rules for prostate cancer using the visualization properties of SOM.
- **Logistic Regression:** Logistic regression is the classic algorithm when it comes to assess the risk of a determined disease, and CAD is no exception. All the risk scores presented before used logistic regression. It is basically a mathematical model

that can be used to describe the relationship of several independent variables to a dichotomous dependent variable, by estimating a set of unknown parameters using the maximum likelihood method [47]. In a few words, we could say it is a regression analysis using a logistic function. One of its main advantages, it is being a “white box” supervised algorithm, which generates an equation where we can understand how the inputs justify the outputs obtained. However, it does not handle well missing values, and may be outperformed by other algorithms: Voss et al. study [44] suggest that neural networks have better performance for predicting MI and death for CAD than logistic regression.

- **Clustering:** It is a class of unsupervised algorithms. These algorithms receive a dataset of unlabeled data and try to discover similarities between the instances, in order to divide the dataset in different clusters (groups), where the instances share more similarities between the ones belonging to the same cluster, than to the other ones. Usually, they use some kind of similarity distance, for example Euclidean distance, that allows the algorithms to identify similarities and dissimilarities between instances. Clustering algorithms can be more or less interpretable, depending on the algorithm. Nevertheless, it was shown to be possible to increase its interpretability by creating rules for classifying an instance into groups following the approach proposed by Chiu [48]. Lately clustering has become popular for grouping patients, in order to improve the resource allocation in hospitals ([49] and [50]). It has also been used for personalizing the use of risk scores in [32].
- **Fuzzy Systems:** Fuzzy systems are not really a data mining algorithm, but it is included in this section because it can help improve their performance and increase their interpretability. It is based on the fact that humans beings do not talk about precise and crisp values, but in contrast, talk about imprecise concepts like “high risk” and “low risk”, which means that most of the human knowledge is codified in a vague or fuzzy form. The idea of the fuzzy systems is to codify this knowledge in a way that computers can understand. It can be used, for example, to integrate an worker knowledge (fuzzy knowledge) in the construction of an automatic controller for a machine, or translating a crisp risk score into a fuzzy system to determine the risk of CHD [51]. In the work of Tsipouras et al. [52], they create an automated diagnosis system of CAD, using decision trees, in which the rules were transformed into a fuzzy system and optimized, which lead to better final results. In this work, we are more interested in creating fuzzy rules from the clusters obtained [48], which will increase interpretability, and also facilitate the process of classifying new patients.

In our research, we use data mining to personalize the use of risk scores by patients groups, which requires finding those groups and creating rules to assign new patients to them. From the algorithms reviewed above clustering algorithms seem the natural choice to find the groups, which could then be complemented by using fuzzy systems to create rules. Therefore, a further review of clustering algorithms is done in section 3.1.

Chapter 3

Background

The implementation of a risk assessment tool depends upon the algorithms and techniques adopted. For that reason, it is important to survey the different possible choices that could be used in our research. In this section we present several algorithms for clustering, dimensionality reduction, feature selection, balancing data and validation. After each review, we also present the algorithms that we chose to use.

3.1 Clustering

Clustering may be mathematical defined as stated in [53] as the partitioning of a set X of D data items x_i into N groups C_r such that data items that belong to the same group are more alike than data items that belong to different groups. The result of the algorithm is thus an injective mapping $X \rightarrow C$ of data items x_i to clusters C_r . In essence, this means that clustering is like classifying each item into a different class, which is unknown for us at the beginning, since clustering is unsupervised.

Clustering can be useful for several purposes. Halkidi et al. [54] suggest four main applications of clustering:

- **Data reduction:** Real data has often high dimensionality. Using clustering we can reduce the number of dimensionalities to the number of clusters, by choosing a representative value for each one, for example the cluster center;
- **Hypothesis Generation:** Clustering can makes us visualize the data differently, consequently giving us new perspectives, which leads to the generation of hypothesis to explain the clusters;
- **Hypothesis Testing:** On the other hand, we can begin by having hypothesis and test to see if the results of clustering correspond to our expectations;
- **Prediction based on groups:** The items on the same cluster are similar, which means that the group can be characterized by the items features, which by their turn, can be used to classify new items into one of the clusters. For example, the clustering can be used to attribute the best risk score to each cluster, and then new patients will use the risk score that has best performance for patients in the same cluster this patient would belong to.

Obviously, in our case we want to use clustering in order to be able to do prediction based on groups, or in other words, obtain a personalized risk score for group of patients. However there are different types of clustering algorithms and measures of similarity from where to choose. Here we will do a brief review showing their advantages and disadvantages, according to the literature.

3.1.1 Measures of Similarity and Similarity Algorithms

Before choosing the measure of similarity to use on the clustering algorithm, it is important to understand that it depends on the nature of the values of the attributes on the dataset. According to Anderberg [55] there are four classes:

- *Nominal scale*: In this class the values are categorical, in other words, we can say that they are different from one another, but cannot order them or know the number of units they differ from each other. An example of this class could be a scale of colors: Red, Green, Blue;
- *Ordinal scale*: The values in this scale are also categorical, but have an additional property that distinguishes them from the nominal-scaled values: they can be ordered (we know if an value is greater or smaller than the other). However, we still do not know by how much they differ. For example, grades: Poor, Good, Excellent;
- *Interval scale*: In this scale, the values can be represented in an interval scale, like a ruler, that tells us how much units of difference the values have from each other. An example could be the numbers from one to five: 1,2,3,4,5;
- *Ratio scale*: It is an interval scale that has a zero point, for instance [0..5].

Additionally, there are also binary attributes, which in some sense are part of the nominal-scale, because it usually represents two mutually exclusive categories.

Considering the instances x and y of a dataset with n attributes, and that q is a positive integer, α the number of ones in x and y , β the number of ones in x and zeros in y , γ the number of ones in y and zeros in x , δ the number of zeros in both, and m the number of matches, then we can find in table 3.1 the measures of similarity for each type of values, according to Andritsos [56].

However, sometimes the attributes in real life do not belong all to the same class. For these kind of situations Andritsos [56] and Maimon et al. [57] suggest equation (2.1), which accepts mixed attributes and missing values.

$$d(x, y) = \frac{\sum_{i=1}^n \delta_i d_i}{\sum_{i=1}^n \delta_i} \quad (2.1)$$

If the value in x_i or y_i is missing, then δ_i is 0, otherwise it is 1. And d_i , in case the attribute is binary or nominal, is 0 if $x_i = y_i$, else it is 1. If the attribute is interval-based then $d_i = \frac{|x_i - y_i|}{\max_i - \min_i}$, where \max_i is the maximum value of the attribute i and \min_i the minimum. And in the last case, where the attributes is ordinal or ratio-based, they need to be converted to interval-based, with the equations found in table 3.1.

Measures of similarity		
Interval-Scaled Attributes	Minkowski	$d(x, y) = (\sum_{i=1}^n x_i - y_i ^q)^{\frac{1}{q}}$
	Euclidean	$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
	Manhattan	$d(x, y) = \sum_{i=1}^n x_i - y_i $
	Chebychev	$d(x, y) = \max_{i=1}^n x_i - y_i $
Binary Attributes	Simple Matching Coefficient	$d(x, y) = \frac{\alpha + \delta}{n}$
	Jaccard Coefficient	$d(x, y) = \frac{\alpha}{\alpha + \beta + \gamma}$
Nominal-Scaled Attributes	Dissimilarity Coefficient (Hamming Distance)	$d(x, y) = \frac{n - m}{n}$
Ordinal/Ratio-Scaled Attributes	To Interval-Scaled Attributes	$z_i = \frac{x_i - 1}{\max_{i=1}^n x_i - 1}$

Table 3.1: Possible clustering similarity measures depending of the values' class of the attributes.

These measures aren't exclusively employed in clustering. They can also be used to find the most similar known instances to a new one, like for example, in image retrieval or document search. These are called: similarity algorithms. In our particular case it would make sense to divide the known data into groups that are best classified with each risk score. Therefore, each new instance could be tested for similarity and the group containing the most similar instances, would define the risk score to be used.

Regardless of using similarity measures for similar search or clustering, it is important to choose an adequate similarity measure that adapts well to the features. In our case, the features contain both binary and interval-scaled values, which mean that a Euclidean distance may not be able to generate the best results. One solution could be, the use of a mixed distance with Euclidean for the interval-scaled values and binary distance for these kind of values. Another solution, could be the discretization of the non-binary values. In this way, it would be possible to apply a nominal-scaled distance, like the Hamming distance, or the Jaccard distance (1 - Jaccard coefficient). The latter is very similar to the former, with the exception that it gives the percentage of non-zero similarity for binary values, which implies a focus on the existence of an attribute. For nominal values, it is equal to the Hamming distance.

3.1.2 Clustering Techniques

Over the years thousands of clustering techniques were created, and it is not easy to keep track of them all, which implies a necessity to categorize them according to their similarities, advantages and disadvantages. However, depending on the researcher we can find different categories in the literature. Here, we will just focus on the five most common clustering approaches found in [58]:

- **Partitioning Clustering:** In this approach, the algorithms divide the dataset in clusters and iteratively evaluate and update them, until a certain criterion is fulfilled. The classical examples are k-means and k-medoids.

- **Hierarchical Clustering:** These algorithms build a hierarchy of clusters. There are two main approaches: agglomerative and divisive. The agglomerative approach assumes that all elements belong to a different cluster and merge them until obtaining a hierarchy with the desired levels. The divisive approach puts every element in the same cluster and successively splits it in smaller ones. The classical examples are AGNES and DIANA algorithms respectively.
- **Density-Based Clustering:** Density is defined as the number of instances in the neighborhood, and it is used in this approach as the grouping function that leads to the clusters formation. Known examples are DBSCAN and Subtractive Clustering.
- **Grid-Based Clustering:** These algorithms are specifically used for spatial data. They divide the space into a grid of cells with a certain granularity and group the objects that belong to the same cell. Typical examples are STING and WaveCluster.
- **Model-Based Clustering:** In this approach, a model is proposed for the clusters and the algorithms try to find the best approximation possible, assuming the model is correct. Examples of these approach are Gaussian Mixture Model and Self-Organizing Maps.

In the following sections, we will describe these approaches in more detail, in order to understand how the different techniques work and their advantages and disadvantages.

3.1.2.1 Partitioning Clustering

The algorithm k-means [59] is maybe the most well-known and commonly used of all clustering algorithms, and a classical example of partitioning clustering. Before using it, it has to be decided the number of clusters (k) that the data will be divided into. Then it randomly assigns each instance to a different cluster, and iteratively assigns each instance to the cluster which has the closest center (mean of the points belonging to a cluster) to it, until the squared sum of the distances between the center and the instances stabilizes. This is an offline minimization of the distance between the points inside the cluster. However, it is very probably that it will fall into a local minimum, instead of the global [58]. Additionally, this is going to be very dependent of the initial randomly created partitions. Furthermore, k-means can be highly influenced by outliers, which may deteriorate its performance. This lead to the creation of k-medoids [60], which is in all equal to k-means, with the exception that instead of using the mean, it uses the data point belonging to the cluster that is more in the center (medoid). This creates a more robust algorithm, when it comes to outliers.

Inside the partitioning clustering algorithms, there are also the fuzzy algorithms, in which a point does not belong to exactly one cluster, but instead has a degree of belonging to the different clusters. One of the most known algorithms of this type is the Fuzzy C-means (FCM) [61], which is basically an adaptation of k-means, in order to introduce fuzzy logic to its behavior. The main difference is that the center of cluster is calculated by a mean of all points weighted by the degree of belonging, and not just the mean of the points belonging to that cluster. This algorithm produces a membership function for each point, which facilitates the creation of fuzzy rules, to assign a new point to a cluster. However, it has the same problems as every partitioning algorithm of being very dependent of the initial partitions, and falling sometimes into a local minimum. Additionally, it can be slower than k-means, because it requires more calculations.

3.1.2.2 Hierarchical Clustering

The hierarchical clustering can be further divided into agglomerative, like the algorithm Agglomerative Nesting (AGNES), and divisive, like the algorithm Divisive Analysis (DIANA) [62]. In both cases the number of clusters does not need to be specified, but a stop criterion must be defined. AGNES begins by considering that every instance belongs to its own cluster and iteratively joins clusters that are the most similar, constructing a tree of clusters (dendrogram), where the leaves are all the initial clusters, and the root, the sole cluster containing all instances. That is why a stop criterion is needed, otherwise only one cluster will be obtained, containing all the initial data. Basically, in AGNES the dendrogram is cut at the desired level, obtaining the clusters. DIANA, is a very similar algorithm, but done in the reverse. It begins by considering that every instance belongs to the same cluster, and divides it into clusters that present more dissimilarity, constructing the dendrogram into the other way around. Eventually, if not stopped by a criterion, all instances will belong to its own cluster. The main weaknesses of this approach is that what was previously done cannot be undone, and that it does not scale well, having at least quadratic complexity [58].

One possible alternative to the classical hierarchical clustering techniques are decision trees. As pointed out by Kavsek et al [63], decision trees and hierarchical clustering are very similar, with the difference that one is supervised and the other one is unsupervised. This is because, a decision tree may be seen as induction of concept hierarchies [64], where a concept is associated to each node. On the other hand, the hierarchical cluster may also be seen as tree (dendrogram), representing the concept hierarchies. Based on this knowledge, many studies were made in order to develop an unsupervised decision tree, like in the study conducted by Basak et al [65], where they introduce unsupervised splitting methods for creating the tree, and consider that each leaf node is a different cluster. Therefore, if there is some supervised information, even if not the one need to cluster the data, then it can be possible to create a decision tree, and consider each leaf node a cluster. The main advantage of this approach consists on its interpretability, and automatic generation of rules to classify a new instance in one of the clusters. Inside the decision trees algorithms, one of the most used one is possibly Classification and Regression Tree (CART) [66], which instead of using training data measures of relevance uses cross-validation for splitting the data. Besides, it accepts both categorical and numerical variables, and it is able to handle missing values.

3.1.2.3 Density-Based Clustering

One example of density-based clustering is the subtractive clustering [67]. In this algorithm all points are a potential cluster center, whose potential is given by equation (2.2), where r_a is a positive constant, defining the radius of the neighborhood. This equation will give a higher potential to the points with more neighbors (more density), and the one with the highest will be chosen as the center of the new cluster. Then, the potential of the points is revised using equation (2.3), by removing the influence of the new center (P_c), and establishing the neighborhood radius, which will have reduction of potential (r_b). This process is repeated until all points have a potential inferior to usually 0.15 of the first center created.

$$P_i = \sum_{j=1}^n e^{\frac{-4\|x_i - x_j\|^2}{r_a^2}} \quad (2.2)$$

$$P_i = P_i - P_c e^{\frac{-4\|x_i - x_c\|^2}{r_b^2}} \quad (2.3)$$

One the main disadvantages of this algorithm it is its quadratic complexity, due the fact that we need to compute the distance between all the points in equation (2.2). Density-Based Spatial Clustering of Applications with Noise (DBSCAN), on the other hand, has a complexity of $O(n \log(n))$, which is fairly better. DBSCAN [68] introduces two parameters: Eps (maximum radius of the neighborhood) and MinPts (minimum number of points in Eps-neighborhood of a point). Additionally, it defines that a point p is density-reachable from a point q , if $p \in N_{eps}(q)$ ¹ and $\|N_{eps}(q)\| \geq MinPts$. This is a central concept of the algorithm that is used in every iteration. Basically, in each iteration it selects an arbitrarily point p , and retrieves all points density-reachable, if there is at least one point retrieved, then p becomes the center of a cluster. The problem with this algorithm is that requires the user to define two parameters (Eps and MinPts), and its performance is very sensitive to the values chosen for it.

3.1.2.4 Grid-Based Clustering

The main idea behind grid-based clustering is to divide the space into cells. SStatistical INformation Grid-based method (STING) [69], for example, divides the space into rectangular cells, and creates an hierarchical structure with different levels, where each cell in a higher level is divided into smaller cells, and for each one, statistical information is calculated, like mean and standard deviation. In essence, each level has several independent clusters, which facilitates the addition of new objects, and creates efficient parallel queries in the space. WaveCluster [70], also divides the space into a grid, but uses the frequency domain (by using wavelet transformations), to discover the dense region that appears naturally using this new transformed domain, and consequently identifies the clusters, and its cells. An advantage of this approach is that the user does not need to define the number of clusters. However, it is necessary to define the grid characteristics.

3.1.2.5 Model-Based Clustering

Gaussian Mixture Model (GMM) usually is implemented over a Expectation-Maximization algorithm (EM) [71] algorithm using a gaussian function. The algorithm begins with an initial model and iteratively updates the function parameters (weights, means and variances), increasing the likelihood of the model, until the error is inferior to a certain threshold [72]. It can be considered similar to k-means, in the sense that the means (centers) are updated in each iteration but following a probabilistic approach, instead of using just the real mean. Self-organizing maps (SOM) [73], by contrast, follows a complete different model: not probabilistic, but topological. Initially we create a model or a codebook of vectors in the original data space, but associated to nodes in 2D space. The topology of these vectors is given by a neighborhood function that will also be used in each update of the vectors, until the convergence with the data is achieved [74]. This algorithm can also be used for dimensionality reduction and data visualization.

¹ $N_{eps}(q) = \{\text{point } p \mid \text{dist}(p, q) \leq Eps\}$

3.1.3 Comparing the Techniques

All of the algorithms presented have their advantages and disadvantages. Furthermore, its results are dependent on the data and parameters' choice. However, it is important to understand the main problems and features each algorithm presents.

Inside the partitioning algorithms, k-means is a very simple algorithm to implement and can run efficiently in linear time. Its downside is the sensitivity to outliers as well as to the initial randomly generated centers, as seen in the research done by Fung [72]. In regard to some of its extensions: K-medoids is more robust, since it is not so sensitive to outliers; and fuzzy c-means is able to deal with fuzzy logic and create rules to put new instances on a cluster. However, they still have k-means' disadvantages: they cannot discover clusters with non-convex shapes, cannot handle noise, and the number of clusters has to be decided *a priori*. Besides, they do not scale well for large datasets [58].

The classical hierarchical approaches (DIANA and AGNES) may also not scale well, but they are very robust in terms of outliers [54]. On the other hand, they do not handle well missing values, and there are a lot of possible measures for them, which means that choosing a good measure may not be trivial [58]. One possible solution is the use of CART as hierarchical clustering, which eliminates the need to choose a good measure, and is able to deal with missing values. It is also a more interpretable option.

The density-based algorithms can handle noise easily, and discover clusters of arbitrary shape. However, they also have their drawbacks. Subtractive clustering, for instance, is not very efficient in terms of temporal complexity. DBSCAN, on the other side, needs two density parameters to be defined, and is very sensitive to their values [58].

Grid-based algorithms, in contrast, have a low temporal complexity, which makes them fast. Besides, they are also able to discover arbitrarily shaped clusters, but because they expect spatial data, instead of numerical [54], they may not be the most adequate algorithms for most situations.

In our specific case, temporal complexity is not a major problem. Probably, the most relevant aspects are: if the algorithms can discover clusters of arbitrary shape, and how robust they are in terms of dealing with noise, outliers and missing values. Besides, it would be important the possibility of constructing rules to assign a new instance to a cluster. Thus, subtractive clustering was our choice, because it has the advantages of a density-based clustering technique, and a system of rules can be created by using the method proposed by Chiu [48]. We also chose Fuzzy c-means and CART, mainly because they can easily be used to cluster data and create rules, which can increase our system interpretability.

3.2 Dimensionality Reduction

The real world datasets have a great number of features (variables), and consequently high dimensionality. This may decrease the efficiency of data mining algorithms, because with the increase of features, it becomes exponentially difficult or even impossible to compute the data (*Curse of Dimensionality*), and because it may deteriorate the algorithm's performance and accuracy. This is also true for clustering algorithms. Kriegel et al. [75] point out four problems with using clustering for high dimensionality data spaces:

- The functional dependencies in the data become more complex as more attributes contribute to the actual relationships;
- Concepts like proximity, distance, or neighborhood become less meaningful with the increasing dimensionality of a dataset;
- Many irrelevant attributes may interfere with the efforts to find the clusters;
- In a dataset containing many attributes, there may be some correlations among subsets of attributes.

One way to deal with it, it is dimensionality reduction, or strictly speaking, a technique that transforms the dataset X with dimensionality D into a new dataset Y with dimensionality d , where $d < D$, and often $d \ll D$ [76]. This is possible because of two data characteristics: many variables have smaller variation than the measurement noise, and thus will not contain relevant information, which means they can be discarded; some variables are correlated, thus a set of representative uncorrelated variables can be found, while discarding the remaining ones without losing information [74]. In essence, there is a minimum number of variables that can define the data space without losing its relevant information and properties. This is called *intrinsic dimensionality* [77]. Finding this dimensionality, can facilitate the visualization, compression and classification of the data, while avoiding the *Curse of Dimensionality*.

Dimensionality reduction techniques can be divided in two main groups: linear and non-linear techniques. Traditionally, the methods used were linear, such as PCA. However, in reality, most data is not linear. Therefore, it cannot be completely represented using a linear approach, which lead to the development of non-linear techniques that have the potential to handle more complex data, without simplifying it to a linear model. Maaten et al. [76] classified the non-linear techniques further into three categories: global techniques, local techniques, and global alignment of linear models.

In this section, we will explore some of the dimensionality reduction techniques algorithms proposed by the scientific community.

3.2.1 Linear Techniques

Principal Component Analysis (PCA) [78] is the most representative unsupervised linear technique for dimensionality reduction. It reduces the dimensionality of a large number of correlated variables to a set of new uncorrelated variables, while preserving the maximum variance. This is performed by calculating of the covariance matrix between all dimensions ($D \times D$) and determining the d principal eigenvalues (principal components).

PCA has some major drawbacks: the computation of eigenvalues is proportional to data dimensionality, which means it may not doable for very high dimensional data spaces; the number of principal components we need to preserve the intrinsic dimensionality is unknown; it is as a linear technique that can only find linear subspaces, and thus it cannot represent accurately non-linear manifolds [74]. The last drawback suggests that, in theory, a non-linear technique is more adequate to deal with non-linear data. Nevertheless, that is not always true. In the comparative review done by Maaten et al. [76] most of the algorithms were outperformed by PCA in real world databases. Besides, another advantage of PCA versus the non-linear algorithms is that it easily accepts new data points into an existing low-dimensional space (out-of-sample extension), which requires approximation schemes for the latter [79].

3.2.2 Non-Linear Techniques

Non-linear dimensionality reduction techniques were created in order to deal more accurately with real data that is in reality non-linear. In this section, we will briefly review some of the non-linear techniques that can be found in the literature.

3.2.2.1 Global Techniques

Global techniques are a category inside the non-linear techniques that tries to preserve the global properties of the original data in the low-dimensional representation that will be obtained. In this section, we will present three of such algorithms: Isomap, Kernel PCA and Multilayer Autoencoders.

- **Isomap:** Traditionally, the dimensionality reduction algorithms, for instance PCA, measure the variance using Euclidean distance. However, if the points are in a non-linear, curved manifold, this kind of distance may be deceptively close (straight-line), while in reality the distance in the curve is considerable. Isomap [80] tries to overcome this problem by using an estimation of the geodesic distance (curvilinear distance, which will give the real distances between points on the manifold). This is attained by constructing a neighborhood graph (k neighbors), and finding the shortest path between two points (pairwise geodesic distance). Then, it maps the high dimensional data into a low dimensional representation, while preserving the geodesic distances. Some of Isomap weaknesses are [76]: it may construct erroneous connections in the graph; it is sensitive to 'holes' in the manifold; it has trouble finding a nonconvex manifold, and it is necessary to give the number of neighbors to construct the graph. In spite of that, Isomap can outperform PCA in highly dimensional spaces [76]. In Essence, Isomap is similar to PCA, but using geodesic distances.
- **Kernel PCA:** Kernel PCA [81], like the name indicates, is a reformulation of the PCA algorithm using a kernel function. Instead of computing the covariance matrix, it calculates the eigenvectors of the kernel matrix, and projects the data into them, obtaining a low dimensional representation of the data. Since it works in the kernel space it can use non-linear mappings. However, this will mainly depend of the kernel function chosen. If this function is linear, then this algorithm will turn into PCA. The possible non-linear functions include polynomial, sigmoidal and gaussian kernels [82]. Kernel PCA has the same temporal complexity as Isomap (proportional to the square of the number of instances in the dataset). We also must keep in mind, that the choice of the kernel, while it gives a wide range of options, it also introduces an additional obstacle of implementation. The kernel function will decide the linearity, locality and globality of the approach. In the review by Maaten et al. [76] Kernel PCA performed strongly in almost all databases, and they also put strong emphasis on the fact that it uses an objective function that can be optimized. Kernel PCA, like PCA, also accepts out-of-sample extension with some additional calculations [83].
- **Multilayer Autoencoders:** It was first introduced by Rumerlhart et al. as an unsupervised neural network using backpropagation [84]. More recently, though, it has been used for dimensionality reduction purposes [85]. Basically, it is a multilayer neural network with input and output layer of D neurons and an odd number of hidden layers with d neurons, in other words, we can see it as two neural networks:

an encoder network that converts the high dimensional data to a low dimensional data, and a decoder network that does the inverse. Because those two networks are part of the same backpropagation, it is possible to teach it how to reduce the dimensionality by using the error between the input (original) and output (reconstruction) that in theory should be the same. Then, we can obtain the low dimensional representation of a instance by using the trained network and extracting the values of the middle layer. Backpropagation approaches are usually slow and sensitive to local minima, which can be overcome by using “Restricted Boltzmann machine” (RBM) as in [85]. In order for Autoencoders to work for non-linear manifold, the activation function must be non-linear, otherwise it will be similar to PCA . This approach has the advantage of being able to handle nonconvex data, which is not possible in the former techniques. In the review by Maaten et al. [76] Autoencoders also performed strongly in almost all databases. Since this algorithm is basically a neural network the out-of-sample extension is very natural obtained by running it using new data points.

3.2.2.2 Local Techniques

While global techniques try to preserve the global properties of the original data, local techniques try to preserve only the properties around a small neighborhood of the data points (local properties). However, studies indicate that they seem to perform poorly in real world databases [76]. In this section we will present two of them: LLE and Laplacian Eigenmaps.

- **LLE:** Local Linear Embedding (LLE) [86], like Isomap, constructs a neighborhood graph with the k -nearest neighbors, but it attempts only to keep the local properties . It does that by assigning weights to the edges and reconstructing an instance from its k neighbors, using linear combination. This means that, it assumes local linearity, which allows it to preserve the local geometry of manifold. In essence, it is an optimization problem to minimize the reconstruction error. Because it preserves only local properties , it is less sensitive to errors in the graph and allows embedding of nonconvex manifolds [76]. On the other side, it has difficulties to deal with “holes” in the manifold and tends to collapse large portion of data onto a single point when the target dimensionality is too low [76]. Additionally, it is sensitive to the choice of the number of neighbors, just like Isomap.
- **Laplacian Eigenmaps:** This algorithm is based on the fact that the eigenmaps of the Laplace Beltrami operator are able to define the entire manifold [87]. It also constructs a graph with the k nearest neighbors, and chooses the weights using what they call a heat kernel, where closest data points will have a larger weight, which means they will contribute more to the cost function that will be minimized and will be kept closer in the low-dimensional representation. Hence, it will also keep the local pairwise distance between neighbors (local property). The formulation of the minimization problem can be viewed as an eigenproblem, where the eigenvalues and eigenvectors must be computed, and the low-dimensional values will be the smallest k eigenvalues calculated. Due to the locality preserving, this algorithm is insensitive to errors on the graph, noise and outliers. Additionally, it implicitly emphasizes the natural clusters in the data [88].

3.2.2.3 Global Alignment of Linear Models

This is a combination of local techniques and global techniques. It computes several locally linear models and globally aligns them. This kind of methods are able to optimize nonconvex spaces. However, they suffer from local optima. Besides, they do not seem to perform that well in real world databases [76]. In this section, we will present one of those techniques: LLC.

- **LLC:** Locally Linear Coordination (LLC) [89] was proposed to overcome the main weakness of local techniques. While such methods provide an accurate representation of curved manifolds, they only perform local dimensionality reduction, because there is not a coherent low-dimensional coordinates for all data space, and the different local components do not agree with each other. Therefore, it is necessary to aggregate these components and globally align them. In essence, this algorithm computes local linear models using Expectation Maximization [71] and aligning them using a variant of LLE [76]. This algorithm is more temporally efficient than LLE, because the eigenvalue system to be solved is smaller, and it is also more accurate, because it creates a complete coordination system from the high dimensional data space to low dimensional data space. However, it requires two parameters (number of mixture factors in EM and number of neighbors in LLE), instead of one.

3.2.3 Comparing Techniques

In terms of performance PCA, albeit being a linear algorithm, it seems to deal well with real world datasets. However, there were also others algorithms which show a good performance, like Kernel PCA, Autoencoders and Isomap [76]. In this particular study, performance is a desirable characteristic, alongside the possibility of computing the out-of-sample extension of the algorithm.

Out-of-sample extension is the capacity of adding new data points to the low-dimensional space created before using a dimensionality reduction technique, without the necessity of running it again. This is necessary, because dimensional techniques may create a different low-dimensional space with the introduction of new data points, which would invalidate the rules previously created.

Considering the review done in this section, we chose to use PCA and Kernel PCA, since they fulfill the following requirements: all of them have a natural out-of-sample extension, and seem to have a good performance based on [76].

3.3 Features Selection

Dimensionality reduction techniques are not the only possible solution for overcoming the high dimensionality problem. Usually high dimensional data contains features that are irrelevant or redundant for the data mining algorithm's performance. Therefore, a straightforward approach would be to eliminate some of the features or give them different weights. This is the idea in which the feature selection algorithms are based on. A more formal definition of feature selection could be: "Let X be the original set of features, with cardinality $\#X = n$. The continuous feature selection problem refers

to the assignment of weights w_i to each feature $x_i \in X$ in such a way that the order corresponding to its theoretical relevance is preserved” [90]. By this definition, the use of binary weights represents the elimination of irrelevant features, and consequently it is the same as selection of a subset of the initial features.

Deciding if a feature is relevant depends on the chosen evaluation measures and approaches. There are two main approaches for features selection: the filter model and the wrapper model [91]. The filter model is basically the preprocessing of the training data, using its characteristics to choose a subset of features, without taking into consideration which classifier is going to be used. It can use evaluation measures like dependence, correlation or mutual information between features. The focus of wrapper model, on the other hand, it is not the training data, but the classifier. It performs a search in the space of possible weights for features, evaluating its quality by using the classifier as reference. For instance, it could use the classification error as an evaluation measure.

Theoretically, the wrapper model should give better results, because the classifier itself should provide a better estimate of accuracy than a completely different evaluation measure [92]. However, it is also the most computationally expensive option, because it requires an iterative search on the space of possible solutions, in which the classification algorithm is used at least once in each iteration. Even in a binary problem, the number of possible subsets is 2^n , which is an exponential complexity.

In this section, some of the most common wrapper and filter algorithms for feature selection will be reviewed, emphasizing the characteristics that would be most useful in our study.

3.3.1 Wrapper Model

The wrapper model is a search algorithm that tries to find the best weighting for the features using the classification algorithm itself in order to estimate the weighting’s accuracy. However, this search can be done in several different ways. The most commonly used are: exponential search, sequential search and random search [90].

Exponential Search: This approach is always complete, which means that it guarantees the optimal solution. It can be an exhaustive search on all the space, use heuristic information, or use bounds to decrease the running times. Examples include A* and Branch and Bound algorithms. However, these algorithms are not very practical, because their exponential complexity make them a very computationally expensive choice. In our case there is 16 features, and even a binary search becomes impractical, because of the time required by the classifier and its cross-validation.

Sequential Search: Sequential search tries to find a good solution in polynomial time. In order to do this, it selects only one successor from the possible successors. This means that it may not find the best solution, because it can be a state that was not visited. There are several ways to choose a successor. For instance a forward approach: in each iteration it adds a new feature to the subset of chosen features, but only if it improves the subset in terms of the system’s accuracy. While this approach is clearly faster, it does not guarantee the best solution. Furthermore, for a non-binary search is still very computationally expensive.

Random Search: When the search space is too large, a random approach may be the best choice. It allows searching it, while avoiding local minima, and in much less

time than an exhaustive search. Usually, the number of chosen iterations will give its complexity, which mean it can even be a linear time. However, it does not guarantee that the optimal solution is going to be found. Dash et al. [93] in their work concluded that, for the binary problem, an specific subset has $\frac{1}{2^n}$ probability of being generated, which means that a random approach has to have at least a number of iterations quadratic to the number of features, in order to find the optimal solution. Anyway, for a non-binary search it can give better results in much less time.

3.3.2 Filter Model

This model is more practical and faster than the wrapper model. Although, it has theoretically a poorer results than the latter, it can in some cases overcome it in both results and performance. Here, we present some of the most common filters used by the scientific community.

Information Gain: Information Gain or Kullback-Leibler divergence is a measure that comes from information theory, and computes the divergence between probabilistic distributions, which can be a good measure of a feature relevance. It is commonly used in Decision Trees to determine the most relevant attribute in each iteration. It can also be used to attribute weights to the different features, or to choose the n most relevant features.

Gini Index: It is a measure of inequality created by Corrado Gini [94], which is also used as splitting criteria in decision trees, most specifically in the CART algorithm. However, studies indicate that it disagrees only 2% with the information gain, which explains why most previously published empirical results concluded that it is not possible to decide which one of the two is better [95]. Just like information gain, it can be used to determine weights for the features, or the most relevant ones.

Relief-F Algorithm: Relief [96] is a filter algorithm that chooses the relevant features as the ones which separate the most random instances from the near miss (closest instances from the other class), and the least separate them from the near hit (closest instance from the same class). However, this algorithm can only deal with continuous or discrete features, and two-class problem. Relief-F is an extension of the algorithm created by Kononenko [97], which can deal with multi-classes, noise and missing values in the dataset. It has two parameters that need to be defined: the size of neighborhood and the number of random instances.

Fast Correlation Based Filter: Fast Correlation Based Filter (FCBF) is a widely used method for feature selection, maybe because it removes both irrelevant and redundant features. It has two stage [98]: in the first stage, it calculates the symmetrical uncertainty, in order to choose the most relevant features based on a threshold; in the second stage, it does a redundancy analysis, by iteratively comparing the reference features (in the beginning the most relevant feature) to the remaining features. If they are not redundant, adds them to the reference features. This process continues, until no more features remain to be added.

3.3.3 Comparing Algorithms

Wrapper algorithms are usually intractable, because the search space is too large. However, a random search may give good results in a short time, and because wrapper algorithms theoretically have better results than filter algorithms, it is worthwhile to try to use this approach, both to discover weights for the features, or to just select features to use. Therefore, we decided to use a random search to find a good subset of weights for our features.

Concerning the filter algorithms, both Gini index and information gain are well known measures, that can be easily applied to choose the most relevant features or attribute weights to them. On the other hand, they tend to overestimate the importance of the multi valued features [97]. The Relief-F algorithm can also attribute weights based on its relevance, and does not overestimate so much the importance of the features. However, it does not have into account the correlation between features, which means that it can overestimate the importance of a redundant feature. FCBF has both into account relevance and redundancy, but can only indicate the features that should be used, and not their weights.

Considering the characteristics of the filter algorithms and our problem, we decided to use Gini index and Relief-F to attribute weights to the features, and FCBF to choose the most relevant features.

3.4 Imbalanced Data

Most data mining algorithms assume that they are given a dataset with the same number of samples for each class, in other words, they assume the data is balanced. Unfortunately, the data acquired in real life most of the times is not balanced, further from that, it can be really imbalanced. It is considered that a dataset has *between-class imbalance* when the class imbalances are on the order 100:1, 1000:1 and 10000:1 [99]. This is especially true in medicine where there are *intrinsic* imbalance from the fact that the number of patients with disease is fairly smaller than the patients without it, but also *extrinsic* imbalance that arises from the time and storage of the data.

The data containing patients with CAD is also imbalanced, since the number of patients with high risk (minority class) is usually smaller than the number of patients with low risk (majority class). This means that applying a data mining algorithm to this kind of data may theoretically lead to a very high accuracy for the majority class, but very low accuracy for the minority class. This is particularly problematic because it is important to classify correctly high risk patients for them to receive the appropriate treatment, which may help avoid a cardiovascular vascular event in short time. Perhaps even a fatal one.

A quite intuitive way of solving the problem would be to rebalance the dataset artificially, using *up-sampling* (creating new instances of the minority class) or *down-sampling* (discarding instances from the majority class) [100]. These kind of solutions are called sampling methods. The most common are Random Minority Oversampling (ROS) and Random Majority Undersampling (RUS). In the former, instances of the minority class are randomly duplicated, while in the latter instances of the majority class are randomly discarded. There are also more complex techniques, like Synthetic Minority Oversampling Technique (SMOTE) [101], which creates synthetic new instances of the minority

class, and One-Sided Selection (OSS) [102], which removes instances of majority class that are in the borderline, redundant or noisy.

Another possible solution is the use of cost-sensitive methods that modify the algorithm to take into consideration the cost of misclassifying an instance. Unfortunately, it is difficult to estimate the cost of a misclassification, and those costs are rarely available in most of the problems.

In our work we chose to balance the data using the ROS and RUS methods. This choice was based on a review done by Husel et al. [103]. In their study, they reviewed several balancing methods and concluded that creating or removing samples in a more intelligent way, like SMOTE and OSS, does not seem to compensate in practical terms, since ROS and RUS usually have better results, although it depends on the classifier used.

3.5 Validation

It is important to validate a system in order to assess its reliability. First, the training data should not be used for validation, because the results are biased towards those instances, since the system was created from them. This means that a validation set should also be used to choose the best model. Additionally, it is also reasonable to use a test set to assess the reliability of the final model with a less biased dataset [104]. According to Steyerberg [105] there are four main groups of validation techniques:

- *Apparent Validation:* This approach uses the same dataset for training and validation. As stated above, these results are not very reliable, because it can simply be result of overfitting of the system to the training data, which means that the results cannot be generalized to new and unknown data;
- *Split-Sample Validation:* The data is randomly divided into two or three datasets, which will be used to perform the training, validation and test. This division is static, and may not be the best option, because it may not test how well the system perform for rare events;
- *Cross-Validation:* It is similar to the previous method, but instead of splitting the data only once, it divides it in k sub datasets, keeping one or two out for validation and testing, while the rest is used for training. This is repeated k times. The performance is given by the mean of all the runs. This method is more reliable than any of the previous ones;
- *Bootstrapping:* It is based on the statistical principle that a sample can represent the population. Therefore, it creates a sample of the size of original dataset, by randomly choosing instances with replacement (the instances chosen are introduced again into the dataset). It can be used to obtain statistical information of the system performance. It is also more reliable than the first two methods.

From the above techniques the Cross-Validation and Bootstrapping are the most used, because they are dynamic approaches and consequently give a more reliable validation. Between them, we chose to use cross-validation, because it is less biased than Bootstrapping because it does not use repeated instances.

Another important concept in validation consists on the metrics used. In a classifier with two classes (positive and negative), the performance can be demonstrated by a simple confusion matrix such as in figure 3.1. This matrix allows us to define two evaluation

		True Class	
		P	N
Output	P	TP (True Positive)	FP (False Positive)
	N	FN (False Negative)	TN (True Negative)

Figure 3.1: Confusion matrix for evaluating the performance of a classifier with two classes.

metrics that represent the accuracy by class: the sensitivity or recall (Sensitivity (SE)), which gives the accuracy for the positive class, and specificity (SP), which gives the accuracy for the negative class. The equations for those metrics are respectively (3.2), and (3.3).

$$SE(Recall) = \frac{TP}{TP + FN} \quad (3.2)$$

$$SP = \frac{TN}{TN + FP} \quad (3.3)$$

The overall accuracy is given by equation (3.4). However, this can be deceiving since it does not express how well a classifier performed for each class, and if it has a good performance for both, or if it can only correctly determine instances from one class. For example, we can obtain 95% of accuracy, where 95% is the number of correctly predicted instances for the negative class, but 0% for the positive class. Another possible metric is the precision (3.5), which gives the exactness of the prediction, or in other words, inside the ones considered positive, how many were really predicted correctly.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.4)$$

$$Precision = \frac{TP}{TP + FP} \quad (3.5)$$

Both the measures presented above (accuracy and precision) cannot be used to represent the completely effectiveness of the classifier, because they can hide a high rate of misclassification in one of the classes. In order to express the total behavior and performance of a classifier is better to use the $F_{measure}$ (3.6), or G_{mean} (3.7), which take in consideration both classes. However, $F_{measure}$ is still a bit sensitive to the data distributions, which make us prefer to use G_{mean} in our tests.

$$F_{measure} = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (3.6)$$

$$G_{mean} = \sqrt{SE \times SP} \quad (3.7)$$

Validation measures may seem to indicate that a model is better than another one, but after statistical analysis, it can be concluded that there is not a statistically significant

difference between them. That is why it is important to use statistical analysis to support the results obtained.

The first step that needs to be done, in order to have statistical valid results is to run the model several times (usually 30 or more) to obtain a large sample in which its means tend to approach a normal distribution (Central Limit Theorem). However, this does not guarantee that the data follows a normal distribution. Hence, before choosing a statistical test, it is important to decide if the data is normal, that is the main assumption of parametric tests. Using a parametric test is usually preferred because it allows a flexible modeling and estimation of parameters and confidence intervals. The normality of the data can be tested by Kolmogorov-Smirnov statistical test, which detects deviations from the distribution. Considering 95% as confidence level, if the test gives $p < 0.05$ then the population is unlikely normal and a non-parametric test should be used. In case $p > 0.05$, then Levene's test must be used to assess the variance homogeneity which is a pre-condition for parametric tests.

Using figure 3.2 it is possible to decide the correct statistical test for continuous data. The decision depends on the number of categories (or algorithms being compared), if the samples are matched (were tested on the same dataset and in the same conditions), and if the data follows a normal distribution or not.

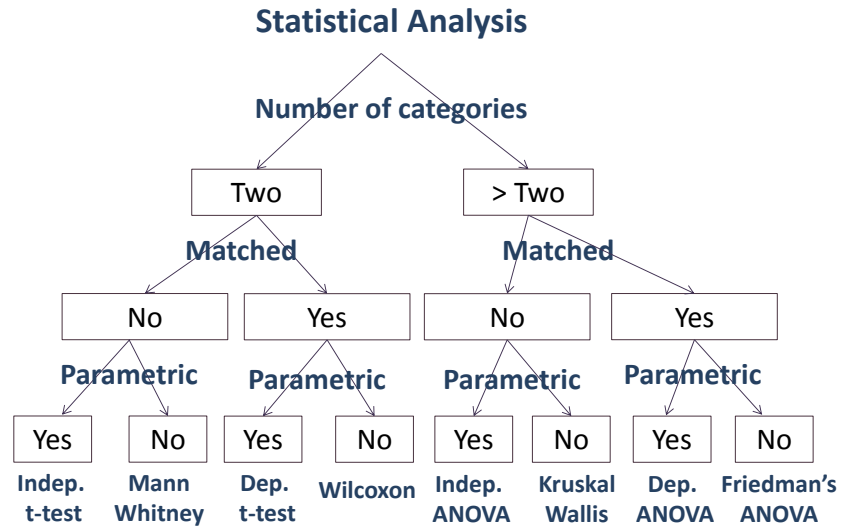


Figure 3.2: Statistical test that can applied on data.

When multiple comparisons are performed, the test only indicates if there are significant differences or not, but it does not say where those differences are. For that we need to run post-hoc methods: apply a statistical test of 2 categories for all comparisons. This test must have exactly the same characteristics and assumptions as the one used before. For instance, if the first test is conducted with dependent ANOVA, the post-hoc methods must be done using dependent t-test, which also works with matched samples and parametric data.

Increasing the multiplicity of tests also increases the likelihood of witnessing a rare event and the probability of rejecting the null hypotheses when it is true. This means that the level of confidence does not represent correctly the post-hoc methods significance. A way to address this problem is the Bonferroni correction, which simply divides the level of confidence by the number of comparisons done. For example, if in the beginning the test was using $p = 0.05$, and in post-hoc methods three comparisons were performed, then the significance level is going to be $p = \frac{0.05}{3} = 0.017$.

Chapter 4

Methodology

The main goal of our research is the creation of a short-term risk assessment tool for CAD patients. However, creating a new risk score requires time and a large dataset. In section 2.1, we briefly reviewed well known risk scores: TIMI [10], GRACE [11] and PURSUIT [12]. All those can be used for determining short-term risk for CAD patients, and comparative reviews show that they present good results for several different datasets. Based on this, we opted to create a new risk assessment tool that is a combination of those three. This removes the need to choose a standard tool, allows the addition of new well validated information in the form of new risk scores and improves the risk assessment without further costs.

There are several methods of combining classifiers in the available literature. In section 2.2, we divided them in parameter fusion and output combination. The latter can be further divided into voting methods and selection methods. Among the output combination using selection methods, there was a personalization based on groups approach proposed by Paredes [32], which selects the best risk score for similar groups of patients. This makes sense because each risk score has patients that it classifies correctly, and they are not necessarily the same. Considering this, it is possible to combine the risk scores and obtain more accurate results than using a single one. In his work, Paredes does a personalization using dimensionality reduction and clustering. In our work, we further explored this using other algorithms and personalization approaches.

Our first personalization approach was to find the natural clusters in the data, assigning to each one the risk score with the best results for that cluster (Clustering Patients, section 4.1.1). Additionally, rules were generated to assign a new patient into one of the groups. The results were not satisfactory. Therefore, we developed a second approach (Dividing by Scores, section 4.1.2). In this approach the data was divided in 4 groups: patients there were correctly classified by each score and patients that were always incorrectly classified (the GRACE score was used in this group, because it presented more sensitivity in general). Then, rules were generated to assign each patient to one of the groups, consequently choosing the risk score to use. This approach was not satisfactory either, which lead to the creation of a third one (Similarity Measures, section 4.1.3). In the same way, the patients were divided in 4 groups, but instead of using rules to assign a patient to a score, similarity measures were used.

In this section, we describe in more detail the methodology used, including the approaches, algorithms, the dataset and validation process used.

4.1 Personalization based on Patients Groups

During our research we developed three different approaches for combining risk scores using personalization based on patients groups. For simplification purposes, names were assigned to each approach. Their designations are: clustering patients, dividing by scores and similarity measures.

All the approaches consist of two phases: training and validation. In the training phase, part of the database was used to create a set of rules. Then, in the validation phase the rules were used to assign a new patient to a risk score. This is done on a different group of patients of the ones used in the training phase. This way, we will know if the algorithm really works for new data, and not only for the training data, which could be due to overfitting. In the schematic representation of our proposed approaches, the first line represents the training process and the other one represents the validation process. In order to calculate the classifier effectiveness in both phases, we calculate the risk using a score. If the score indicates a patient with high risk, then its probability of having a cardiovascular event in short-term is supposed to be high. We can verify if that happened by looking at the number of days a patient took to have a cardiovascular event. If it took less than 30 days, then obviously the patient was indeed a high risk patient.

For simplification purposes, we consider two classes: high risk (patients that had an event within 30 days) and low risk (patients that had not an event within 30 days), instead of the three purposed by the risk scores (high, intermediate and low risk). This is done by assuming that low risk, includes low and intermediate risk. This is clinically correct, as stated in [32]: “The reduction of output categories (low risk/high risk) is correct. In fact, the aim of cardiologist in clinical practice is to discriminate between high risk patients and low risk patients. In a clinical perspective, the identification of intermediate risk patients is not so significant.”

In all the approaches a pre-processing stage is effectuated. This stage includes different kinds of algorithms that differ from test to test. For example, in the results section for clustering patients the pre-processing phase may contain in one of the tests normalizing and feature selection and the other may contain discretization and dimensionality reduction. The detailed description of the algorithms it may contain is presented on section 4.2.1. Additionally, in each test we explain what algorithms were used when presenting the results in section 5.

In the following sections we present the three approaches and their differences in more detail.

4.1.1 Clustering Patients

This approach is based on the idea that finding the natural clusters on data, will maximize the similarity of the patients inside the group and the dissimilarity between groups. As such, clustering patients creates robust rules for assigning new patients to an already formed group, because their similarity is well defined. These rules are, then, used to select the most adequate score for each patient. In figure 4.1 it is possible to see a graphical representation of this approach and its training and validation phases.

The training phase uses the dataset to construct the rules that will be used in new patients. First the data is pre-processed. After this step, a clustering algorithm is applied

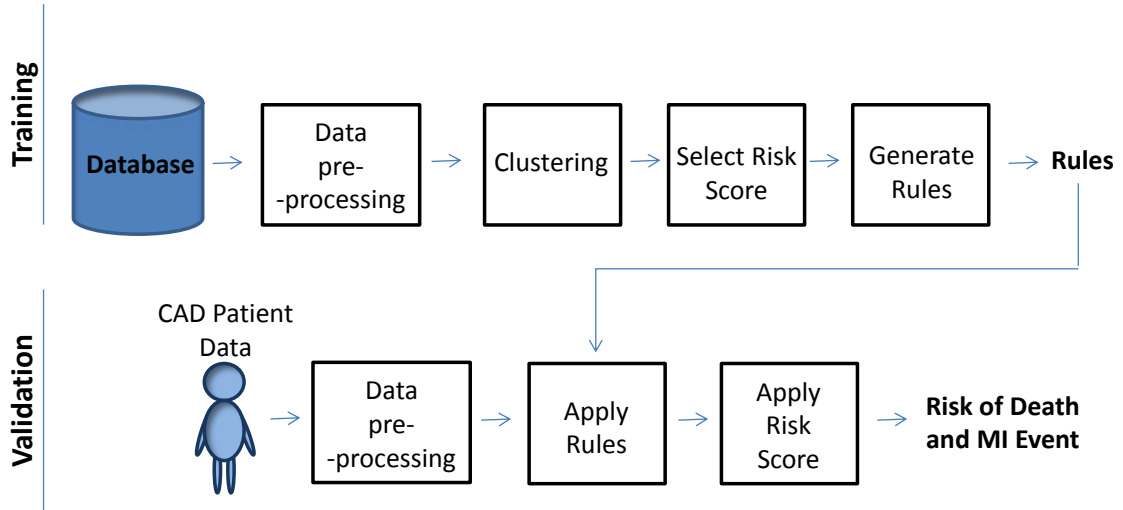


Figure 4.1: Visual scheme of the clustering patients approach.

to cluster patients in order to attain the personalization by groups. This allows us to select the most adequate risk score for each group and generate the rules that will assign a new patient to one of the risk scores.

In the validation phase, the new data must be once again pre-processed the same way it was done in the previous phase. This guarantees that the rules are applicable. After applying them, a risk score is attributed to each patient, and the risk of a cardiovascular event can be calculated using it.

4.1.2 Dividing by Scores

The patients that are correctly classified by a risk score (True Positives and True Negatives) may have similarities between themselves. This approach tries to find those similarities by dividing the data into the n groups, where n is the number of risk scores being used. Nevertheless, there are some patients that are incorrectly classified by all scores. It was decided to consider those instances as part of the GRACE group, because it presents higher sensitivity, and in our case it is preferable to have a good sensitivity rather than specificity, because the fast identification of the high risk patients is more important than identifying low risk patients. When more than one risk score correctly classifies that instance, it is assigned to one of them by giving preference to GRACE, then to PURSUIT and only then to TIMI.

This division can be seen as a label that can be used in a classification algorithm for determining which score should be used for each patient. This is different of what was done in the previous approach, where the data was clustered and then risk scores were selected for each one. Here, the patients are divided by the scores and a classifier is trained to assign each instance to a group.

As it can be seen in the training phase (figure 4.2), the data is initially divided into groups without pre-processing, because all the factors are needed for the risk scores to calculate the risk. Then, the data is finally pre-processed. In the next step a classification/clustering algorithm is used. It could be any supervised classification algorithm, but because it needs to permit the generation of rules, the same clustering algorithms were

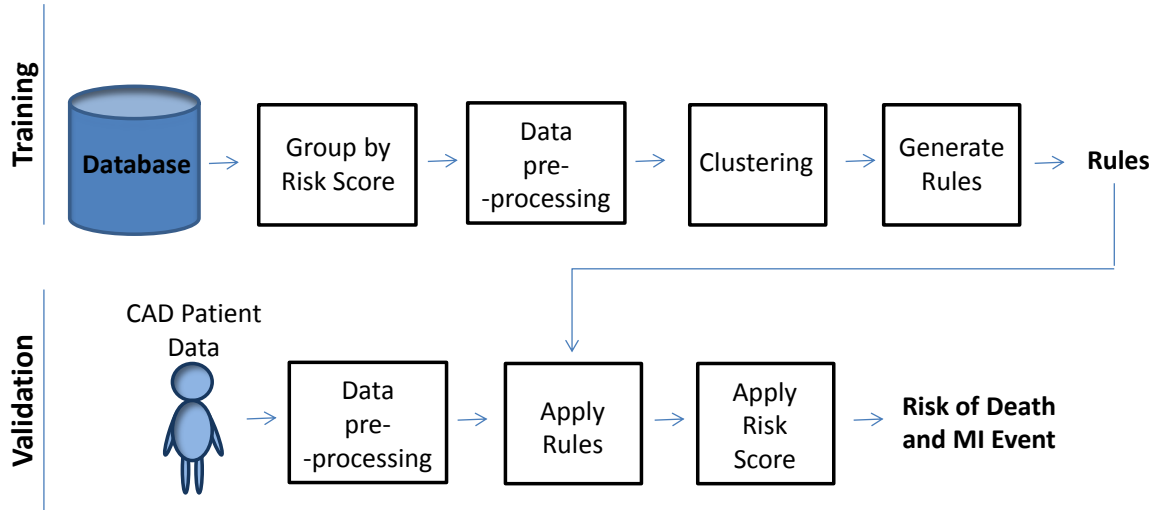


Figure 4.2: Visual scheme of the dividing by scores approach.

chosen. Despite being unsupervised algorithms, we can use them by adding the label as a new attribute. Afterwards, we discover the clusters using fuzzy versions of the clustering algorithms which give us the number of rules and the membership functions. This allows us to estimate the fuzzy rules that assign an instance to a cluster using the least square method. In one of the versions of this approach we also use a decision tree, which is a supervised algorithm, although it can also be seen as hierarchical clustering.

The validation phase is exactly the same as the clustering patients approach. All the new data is pre-processed, for the sake of creating instances compatible with the rules, which determine the risk score applicable to each patient, and consequently indicates the patient risk of suffering a cardiovascular event.

4.1.3 Similarity Measures

The process of clustering and generating rules is complex, because it involves finding similarity between instances, creating coherent groups, and generating rules that correctly assign a patient to one of the clusters. In all this complexity some relations and information may be lost. Besides, maybe there is not enough data on our dataset to correctly identify the groups that exist in the patients, and validate it thoroughly. Even so, there are similarities between patients that can be used to identify the best risk score for new patients. This approach finds these similarities without introducing unnecessary complexity.

A new patient can be more similar to one of the patients of the training set. This similarity can be measured using the distances described in section 3.1.1. Therefore, if, according to a similarity distance, a patient is more similar to one that is correctly classified by TIMI, it is probable that TIMI will also be able to classify it accurately. Based on this, we developed the similarity measures approach that can be seen in figure 4.3.

The training phase is only composed by two steps: group creation using risk scores, and pre-processing of the data. The groups' creation is equal to the previous approach, in which the instances that are correctly classified with a risk score are grouped together. Afterwards, the data is pre-processed. This may help the measures to identify the sim-

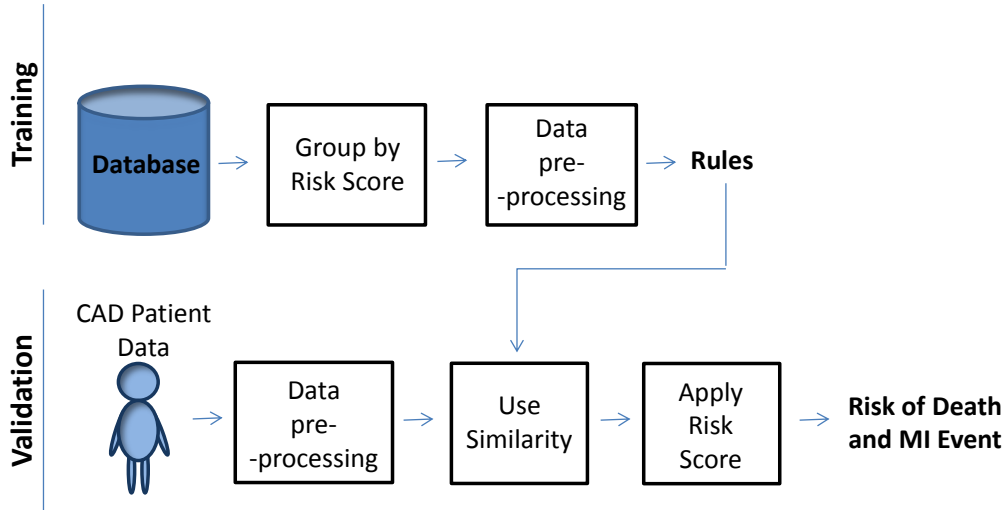


Figure 4.3: Visual scheme of the similarity measures approach.

ilarities between instances. The training validation measures indicate how well the risk scores perform together in the training set, considering that all instances are assigned to the risk score that correctly classifies it. One should keep in mind that the accuracy is not 100%, because there are instances incorrectly classified with all scores.

The groups created in the training phase are used in the validation phase to assign new patients to the most adequate risk score. Foremost, the data must be pre-processed as was the training data. Then, the similarity measures identify the most similar instance of the training set for each new instance. This is done by comparing all the training set instances to the new instance using a similarity measure. The closest one is considered the most similar. Basically, we are assuming that similar patients are well correctly classified with the same score, and that if the closest patient to the new one is well classified with that risk score, then belongs to the group that is well classified by it. Although this assumption may be wrong sometimes and can be sensitive to outliers, with a dataset representative of the patients it is possible to find a potential good risk score for the new instance, even if there are not enough patients to find similarities in the dataset that can be used in clustering. However, this idea could be further explored and confirmed using a larger dataset, since the ones used has a very small quantity of high risk patients (approximately 7.6%).

4.2 Algorithms and Tools

This research involved several algorithms and MATLAB tools for its implementation. For comprehension purposes, we present them in this section separated by categories with the names given to the steps presented in the general explanation of the proposed approaches.

4.2.1 Pre-Processing

Pre-processing is a common step to all the different approaches. It is basically the application of a set of algorithms and methods to a test. Some transform the data, adequating it to another specific algorithm, and some filter it, simplifying the data mining process.

Furthermore, the pre-processing methods used depend of the algorithms being employed in each step. In this section, the methods used during our research are presented.

4.2.1.1 Dealing with Missing Data

Missing data is part of almost all medical research. Our dataset also contains some instances with missing values. Fortunately, it is a small number, less than 1% of the patients, and they are all low risk patients. We opted to substitute every missing value in a feature by the median of its domain. Additionally, we also analyzed the dataset and concluded that the binary feature Cardiac Arrest at Admission (CAA) only presented one instance where its value was 1. Considering this, there is not enough information for this feature to be used to help clustering and classifying the patients, especially, because there is no way to validate its use. Based on this knowledge, we did not use CAA in the data mining process. The feature HF signs was also removed, because it was totally defined by the Killip class feature (HF is Killip class > 1), which means that a similar patients using information from the Killip class, will also be similar using HF signs. Consequently, this feature was unnecessary for finding similarity between patients.

4.2.1.2 Normalizing

Our features have both interval-scaled, nominal and binary data. A possible solution to deal with all this types of values is normalizing the attributes to adjust their scales and use an interval-scaled distance such as the Euclidean distance. This can be done by calculating the z-score of every value for each attribute.

The z-score is a statistically standardization method, which converts all values to a common scale of average of zero and standard deviation of one. It can be calculated by subtracting the mean of the population and dividing by the standard deviation, expressly by using equation (3.1).

$$z = \frac{x - \mu}{\sigma} \quad (3.1)$$

Another option is a mixed distance as the one proposed by Andritsos [56] and Maimon et al. [57] that can be found in section 3.1.1, in which the interval-based attributes use Euclidean distance and the nominal/binary ones use Hamming distance. Even in this case it is important to use the z-score in the interval-based attributes to adjust the scale.

4.2.1.3 Discretization

From the 16 features available on the dataset only 4 are interval-scaled data. Normalizing the data and applying a Euclidean distance or a mixed one may not be the best option, because it can indirectly give more importance to the interval-scaled features than it is due. For minimizing this problem, we compared the use of normalization and interval-based distances, with the use of discretization and nominal/binary distances such as Hamming and Jaccard.

The continuous data was discretized by rounding the values to the nearest 10. For example, the age was reduced to categories representing the decades (20, 30, 40), and a

patient with 2.2 of Creatinine levels belongs to the second category. This allows us to apply nominal distances, which give more importance to the nominal/binary features, and do not find similarities in such small differences as there are in the interval-based data.

4.2.1.4 Balancing Data

As explained before, we assume that the risk scores classify the patients in two classes: low and high risk. A high risk instance was defined as a patient who suffered a cardiovascular event within 30 days. All other are considered low risk patients. However, the data is not well balanced. We only have 7% of high risk instances in a dataset of 460 patients, which corresponds to approximately a *between-class imbalance* on the order 13:1. Furthermore, misclassification of the minority class outweighs the cost of misclassifying the other, because it is more important to assess correctly high risk patients, in order for them to undergo an early invasive strategy, which may prevent an avoidable cardiovascular event.

An imbalanced dataset can potentially produce a classifier with worse accuracy for the minority class. Therefore, balancing data is an option that may improve its classification. In section 3.4, we made a review on the balancing methods, which demonstrated that their performance depends on the data mining algorithm being used. Nevertheless, it seems that ROS and RUS usually have better results. Based on this, we decided to implement and apply these two methods to the data and see if they improved our classifier. In order to maintain a robust validation using only new data, this was applied only to the training dataset.

4.2.1.5 Dimensionality Reduction

Dimensionality reduction transforms high dimensional data into low dimensional, eliminating irrelevant and correlated attributes, which may improve classification quality by facilitating the process of identifying the correct information needed for classifying the different instances.

The desirable characteristics of dimensionality techniques, in our particular case, are: a good performance in real data, eventually, non-linear data, and acceptance of new data points in existing low-dimensional space (out-of-sample extension). This is necessary, because dimensional techniques may create a different low-dimensional space with the introduction of new data points, which would invalidate the rules created previously. Considering the review done in section 3.2 PCA and Kernel PCA seem to be the most adequate, since they fulfill all the requirements: all of them have a natural out-of-sample extension, and have shown good performance in [76]. Kernel PCA depends on the kernel used, if a linear one is used then it would be equal to PCA. Therefore, we used a non-linear kernel, namely a gaussian one.

In this work, we used the Matlab toolbox implemented by Maaten that can be found at [106]. It has already implemented the different dimensionality reduction techniques and the corresponding out-of-sample extension. It also has the possibility of estimating the intrinsic dimensionality of the data using eigenvalue-based estimator [107].

4.2.1.6 Features Selection

Instead of using dimensionality reduction techniques it is possible to use feature selection methods that assign a weight to each variable giving more importance to the relevant features, or even eliminating the irrelevant ones. These methods are divided into wrapper and filter algorithms, and some of them were reviewed in section 3.3.

Assigning weights to the features preserves more information and may give better results. We chose to use a random search (wrapper algorithm) attributing weights in the interval $[0.1, 0.2, \dots, 1.0]$. In a training dataset, iteratively, we assigned random weights to the features, which were then used to our approach and then validated by applying cross-validation. The weights with better results for the training dataset were saved, and then tested in the validation set. The best weights on this new data, were posteriorly validated with all data and several runs.

In terms of filter algorithms, we did both: assign weights and eliminate irrelevant features. The former was done with Gini index and Relief-F, and the latter with Relief-F and FCBF. For using these algorithms, we adopted the feature selection package implemented at Arizona State University available at [108], and described by their technical article [109].

4.2.2 Clustering Algorithms

Both the clustering patients and dividing by scores approaches include clustering algorithms. In our research, the most desirable characteristic in this kind of algorithms, it is the ability of producing a system of rules from the clusters obtained.

In the review done in section 3.1, we presented several algorithms. However, the most adequate are: the subtractive clustering that can find arbitrarily shaped clusters and fuzzy rules can be created by the method proposed by Chiu [48]; the Fuzzy c-means that uses uncertainty and can easily produce fuzzy rules [61]; and CART, which is a decision tree, but has common characteristics with hierarchical clustering and can be applied as such. As a decision tree it creates a set of rules as part of the algorithm.

Regarding their implementation, we chose to use the methods available in Matlab in the fuzzy logic toolbox and the statistics toolbox. More specifically, we used the following functions:

- *subclus*: It implements the subtractive clustering algorithm. We used this algorithm in the beginning, but because it did not allow different distances, we implemented our version of subtractive clustering. We used *subclus* to compare the clusters, and confirm that our version was obtaining the same results when using a Euclidean distance.
- *fcm*: The Fuzzy-c means is implemented by it. We did not try different distances in it, because it calculates the center as the mean of the points, and consequently its values will be interval-based, and it would not be reasonable to use a nominal distance.
- *classregtree*: It implements the CART algorithm. To convert it to a clustering algorithm, we considered that each leaf corresponds to a different cluster, and the rule to get to this leaf as the rule for assigning a patient to a cluster.

4.2.3 Grouping by Scores

This is used in the division by scores and similarity measures approaches. It consists in grouping the patients using the risk scores. We consider that there are $n + 1$ groups, where n is the number of risk scores. If a patient is correctly classified by a score then it belongs to its group. If it is wrongly classified with every score then it belongs to the extra group. In order to implement this, for each patient we calculate the risk using all scores. If there is score that gives a true positive or a true negative, then it attributes the instance to that score, else it assigns it to wrongly classified group.

This process creates labels that can be used by a classification algorithms or similarity measures to assign new patients in one of the groups, and consequently determine the risk score that should be used. Every time a patient is assigned to the wrongly classified group, we decided to apply the GRACE, because it presents in overall more sensitivity than the other two and some of the cases may be slightly different and be well classified by it. In the case where a patient is correctly classified by more than one risk scores, even if it is not really important which score we choose, we decided to assign it by the following preferences: GRACE, PURSUIT and TIMI.

4.2.4 Similarity Measures

Similarity measures give how close two instances are to each other. If they are close it means that they are similar, and potentially belong to the same group. This is usually the idea in which the clustering algorithms are based on. Nevertheless, the similarity measures can also be used without clustering. With known groups, we can calculate the similarity between the new instance and all the ones in the groups, and assign it to the same group as the closest one. In our case, we divided the patients in groups that are well classified with each risk score, and then found the closest patient to each new patient, which will give the risk score that should be applied to it.

Our data has mixed types of attributes. Based on the review done in section 3.1.1, the best options between the similarity measures is either normalizing the data and using a Euclidean or a mixed distance, or discretizing the data and using Hamming or Jaccard distance. The Euclidean distance has the advantage of being a well-known and validated measure, although it may not be adequate to mixed types. The mixed distance seems appropriate, but because most of our attributes are binary is important to test nominal distance because they may give better results.

4.2.5 Selecting Scores

In the clustering patients approach, after the clusters creation, it is necessary to select the most adequate risk score for each one. For this to happen, a metric has to be used to decide which one has a better result for each cluster. One of the most common metrics is accuracy, which is a percentage of all the correctly classified instances. However, it can be fairly deceiving, because it may hide the incapacity of the classifier of correctly identifying one of the classes. Therefore, we chose to use the geometric mean ($G_{mean} = \sqrt{SE \times SP}$), that tries to maximize the percentage of correct classification of both classes.

In some cases it is possible to obtain a cluster that only has patients belonging to one class, which means that the G_{mean} may not be calculated correctly. To prevent this, we

select the risk score using the G_{mean} only if there is a score that has $G_{mean} > 0.5$ for that cluster, because this means that it is being well calculated and has a good percentage. If it is less than 0.5, then the value is too low or cannot be calculated. In this case, we choose the risk score by sensitivity, because it is the most important measure for us after the G_{mean} . If this value is also inferior to 0.5, it means that it wasn't possible to calculate because there were not any instances of that class, or the classification was worse than random. If this happens the score selected is the one with the best specificity.

4.2.6 Generating rules

The generation of rules is important because it will increase the interpretability, and usability of the system for new instances, because they classify each new patient into a group, and consequently into the most adequate risk score. The methods used to construct these rules depend on the clustering algorithm used.

For subtractive clustering, the rules were constructed using the approach proposed by Chiu in [48], where he constructs them by assuming that a data point belongs to a cluster if it is near its center (translated by a mathematical function). This creates a crisp rule that is, then, transformed to a fuzzy rule, and optimized to decrease the error rate. This is implemented in Matlab through the function *genfis2*, which can already receive the clusters, or may find them using labels. It uses sugeno fuzzy inference.

Matlab also has a similar function for Fuzzy C-means called *genfis3*. Since the algorithm already calculates the membership of each patient into the cluster, it is easy to compute the fuzzy rules using the method proposed by Bezdek [61], which is implemented in Matlab. For compatibility reasons we also used sugeno fuzzy inference in this method.

The decision trees generate in a natural way rules to assign each instance to a leaf/class. In Matlab, CART also generates a tree of rules that may be seen using the function *view*.

4.3 Tests Performed

There are a lot of possible tests that can be done with all the approaches and algorithms presented. However, only some of them were effectively performed. Tables 4.1 and 4.2 indicate the combinations of algorithms tested.

Clustering Patients / Dividing by Scores	
Pre-Processing	
Balancing	ROS, RUS
Normalizing Discretization	z-score round continuous values
Dimensionality Features	PCA (dim), KPCA (dim, arg) Gini Index, Relief-F (k, m), FCBF
Clustering	Subtractive Clustering (distances, r_a), FCM (clusters' number), CART (pruning level)

Table 4.1: Tests that will be performed on the first two approaches.

Examining the tables, it is possible to conclude that some hypotheses were never used together. This is true for normalization and discretization, but also for dimensionality

Similarity Measures	
Pre-Processing	
Normalizing Discretization	z-score round continuous values
Dimensionality Features	PCA (dim), KPCA (dim, arg) Gini Index, Relief-F (k, m), FCBF, Random Search
Similarity Measures	Euclidean, Mixed, Hamming, Jaccard

Table 4.2: Tests that will be performed on the similarity measures approach.

reduction and features selection. It is not relevant the use of normalization with discretization because the latter creates categories, which means that it is not important to normalize the scale of the attributes. In the case of dimensionality reduction and features selection, they are both solutions for dealing with high dimensions. Therefore, the use of both in simultaneous could lead to a high loss of information.

There are also some combinations that are not possible due to the algorithms characteristics. For instance, using feature selection like Relief-F and Gini index that assign weights to features is not used with CART, because the algorithm does not receive weights in its implementation.

In the table is possible to see that some algorithms also have parameters in parenthesis which need to be tested in order to find the most adequate model.

4.4 Validation

Validation is an essential part of research, seeing as it allows us to obtain more robust and reliable results. In section 3.5 we made a brief review of the validation techniques and metrics available in the literature.

From the metrics reviewed we decided to calculate Gmean, sensitivity and specificity, which gives us a more complete idea how our approaches are classifying each class, than simply using accuracy or precision. Additionally, whenever we need to select the best risk score, or the best classifier, we give priority to the Gmean, because if it is higher, it means the classifier is handling well both classes. In order to calculate these measures, we defined high risk as the positive class and low risk as the negative class.

In terms of validation techniques, we opted to use the 10-fold cross-validation, which divides the data into 10 test datasets and 10 training datasets. The system is trained with each training dataset and then it is tested in the new data of the test dataset. The training and test results are a mean of results obtained for each dataset. This is repeated 30 times, in order to improve its statistical significance. For choosing the best model we only use a run, and then validate its choice by running 30 times and calculating the mean and standard deviation. Those are the results that are presented in the results section.

We calculated the statistical significance with a confidence interval of 95%. Since we performed a lot of tests, we decided to make a statistical analysis only comparing the best results obtained and the results of the risk scores. Because this imply a multiplicity of test, we decided to use first a statistical analysis to see if there was any differences between the solutions and only then if there was any difference, we would compare them with each other.

With the purpose of increasing the comparability between results, we used the same seed in every test, which means that they were run using the same characteristics and are matched samples. For deciding if the data was parametric, the normality test was computed. Unfortunately, the test gave negative to most of the results.

Using all this information, we consulted the scheme in figure 3.2, and concluded that we should use Friedman’s ANOVA statistical analysis to see if there was significant differences between the categories. Whenever that happened, we used Wilcoxon analysis with Bonferroni correction to identify those differences.

4.5 Dataset

The dataset that is going to be used for training and validation is a real dataset of CAD patients with non-ST segment elevation, that were admitted into Santa Cruz Hospital (Lisbon, Portugal) from March 1999 to July 2001. It contains 460 patients and 16 risk factors. In table 4.3, it is possible to find the baseline characteristics of this dataset and in table 4.4 the percentage of endpoints (patients who died or suffered a cardiovascular event in less than 30 days or 1 year) may be consulted.

Santa Cruz Dataset	
Age	63.4 ± 10.8
Sex (Male/Female)	78.5%/21.5%
Risk Factors (%)	
Diabetes mellitus	23.5
Hypercholesterolemia	60.9
Systemic hypertension	61.7
Smoking	21.3
Known CAD	64.6 %
Sbp (mmHg)	142.4 ± 26.9
Hr (bpm)	75.3 ± 18.1
Creatinine (mg/dl)	1.37 ± 1.26
Enrolment [0 UA, 1 MI]	39.1%/60.9%
Killip 1/2/3/4	85.9%/6.8%/7.3%/0%
CCS [0 I/II; 1 CSS III/IV]	24.0%/76.0%
ST Segment Deviation	53.0%
Signs of Heart Failure	14.1%
Tn I > 0.1 ng/ml	32.0%
Cardiac Arrest Admission	0%
Aspirin (0/1)	60.0%
Angina (0/1)	96.0%

Table 4.3: Baseline characteristics of the Santa Cruz dataset.

Time	Event	N	(%)
30 days	MI/Death	35	7.6
1 year	MI/Death	76	16.5

Table 4.4: Endpoints of the Santa Cruz dataset.

Chapter 5

Results and Discussion

In this section we present the results obtained for the tests performed. Each test was effectuated using 10-folds cross-validation. In order to create more robust results, we run each test 30 times. The results presented in this section are the means and standard deviation of those 30 runs.

Each time an algorithm needed to be parameterized, we run the test with different parameters, and choose the parameters that gave the best results in terms of Gmean. Then, we tested the parameters chosen with 30 runs. The parameterization tests may be found on the appendices.

Our main goal is the creation of a risk assessment tool that surpasses the use of a single risk score. Therefore, it is important to understand how the risk scores perform in the test dataset, which is going to be used to test our approaches. In table 5.1, it is possible to find the results' mean for the risk scores, calculated using 10-fold validation and 30 runs.

GRACE presents the best sensitivity and Gmean, while PURSUIT presents the best specificity. It is possible to verify by the standard deviation that the sensitivity results are more unstable than the specificity ones. However, the specificities presented by the risk scores are very close to 50%, which means that the classification of low risk patients is almost random.

Risk Score	Gmean	SE	SP
GRACE	0.6442 ± 0.0030	0.7856 ± 0.0082	0.5318 ± 0.0001
PURSUIT	0.5864 ± 0.0135	0.6391 ± 0.0129	0.5601 ± 0.0003
TIMI	0.5398 ± 0.0100	0.6925 ± 0.0138	0.4379 ± 0.0004

Table 5.1: These are the results initially obtained on the test data using each single risk scores.

Before running tests, it is useful to verify the validity of our hypothesis, which states that choosing the most adequate risk score for a patient can lead to better results. To prove this was possible, we assumed that we could correctly assign each patient, of the test dataset, to the most adequate risk score. This would mean that a patient was either correctly classified by it, or there was no risk score which could classify it. The results obtained using this assumption can be found on table 5.2.

From this table we can see that the results can be improved just by selecting the correct risk score. Although sensitivity increased, specificity improved a lot from the best result obtained by PURSUIT, 0.5601, to 0.7530, as a consequence Gmean also improved. It is

obvious that to obtain such good results, it would be needed a perfect method to assign a new patient to the correct risk score. At any rate, this demonstrates that our hypothesis has a chance of improving the results. Our work is to find the best method to obtain the results as close to this as possible.

Gmean	SE	SP
0.7786 ± 0.0034	0.8100 ± 0.0066	0.7530 ± 0.0000

Table 5.2: Results given if all the patients were assigned to the correct risk score.

In the next sections, we present the results obtained for each test and each approach, and discuss them. For the best results obtained a statistical analysis was also performed to verify their statistical significance.

5.1 Clustering Patients

The clustering patients approach relies heavily on clustering. Therefore, it is essential that we choose the most adequate clustering algorithm. With this purpose, we tested 3 different ones: subtractive clustering, FCM and CART. The results for each one of them can be found in the subsections of this section.

All algorithms have parameters that need to be tested in order to find the most appropriate model for a problem. In our research, we run each algorithm with different parameters and considered that the best model was the one with higher Gmean in the test dataset. Because parameterization required several runs and results, we decided to present them only in appendix. In the case of this approach the results are in the appendix A.

In this section, we present only comparison of the results using different methods and algorithms, which were already parameterized.

5.1.1 Subtractive Clustering

Matlab requires the definition of r_a for using subtractive clustering, which is the cluster center's range of influence in each of the data dimensions, assuming the data falls within a unit hyperbox. This basically means that it gives the radius of the neighborhood that will form the cluster. Tests had to be run for determining its most appropriate value.

Usually subtractive clustering uses a Euclidean distance. However, due to the specific characteristics of our dataset that contain different types of attributes, we decided to test other distances as well.

In the appendix A the tables A.1, A.2, A.3 and A.4 contain runs that were effectuated to determine which was the best r_a for each distance. As expected those values were less than 0.5, which usually gives the best results. As a matter of fact, with the exception of the Euclidean, which performed better with 0.1, the other distances had better results with 0.5. For simplification purposes, we will fix r_a in all the following tests, this will increase the comparability of the results, although testing different values could lead to a better outcomes.

After determining the r_a , we used 30 runs to have a more reliable idea how this approach works with the different distances, using only clustering. These results are presented in table 5.3.

Distance Comparison

The Euclidean distance is the one which presents the best training results, but it is not the one that performs better in the test dataset. The best result is given by the mixed distance, which does not decrease the sensitivity so much as the others, and the specificity is slightly better than the risk score with highest specificity: PURSUIT had 0.5601 and our approach with mixed distance is 0.5815. This improvement is still too small, but it shows that it may be possible to obtain better results with combination of risk scores than using a single score. Interesting to note that the other distances create a large number of clusters where some of them contained only a patient, while this approach usually creates two clusters with several patients, which may be the reason for its success.

Dist (r_a)	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc (0.1)	0.7756 \pm 0.0001	0.8001 \pm 0.0001	0.7519 \pm 0.0002	0.6593 \pm 0.0117	0.7607 \pm 0.0230	0.5783 \pm 0.0084
Mixed (0.5)	0.6691 \pm 0.0037	0.7710 \pm 0.0012	0.5810 \pm 0.0060	0.6715 \pm 0.0077	0.7814 \pm 0.0135	0.5815 \pm 0.0083
Ham (0.5)	0.6219 \pm 0.0045	0.7288 \pm 0.0093	0.5313 \pm 0.0044	0.6206 \pm 0.0173	0.7364 \pm 0.0301	0.5322 \pm 0.0133
Jac (0.5)	0.6393 \pm 0.0016	0.7642 \pm 0.0058	0.5352 \pm 0.0033	0.6345 \pm 0.0149	0.7619 \pm 0.0307	0.5352 \pm 0.0045

Table 5.3: Comparing the use of different distances in subtractive clustering.

Balancing Data

Balancing the data can in theory improve the classification, and consequently the results. For this reason, we used ROS (table 5.4) and RUS (table 5.5) on the training data to balance it. However, the results did not improve. While it is true that the training results were better in some cases, the test results for all distances were worse than without using balancing techniques, both in terms of sensitivity and specificity. Furthermore, the Gmean was always inferior to GRACE's.

Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.6118 \pm 0.0141	0.7213 \pm 0.0206	0.5210 \pm 0.0145	0.6068 \pm 0.0255	0.7233 \pm 0.0482	0.5203 \pm 0.0152
Mixed	0.6492 \pm 0.0037	0.7605 \pm 0.0068	0.5548 \pm 0.0072	0.6437 \pm 0.0129	0.7578 \pm 0.0274	0.5534 \pm 0.0109
Ham	0.6302 \pm 0.0042	0.7493 \pm 0.0104	0.5312 \pm 0.0078	0.6148 \pm 0.0210	0.7348 \pm 0.0360	0.5261 \pm 0.0123
Jac	0.6249 \pm 0.0056	0.7524 \pm 0.0108	0.5201 \pm 0.0079	0.6099 \pm 0.0148	0.7329 \pm 0.0253	0.5166 \pm 0.0157

Table 5.4: Comparing the ROS on different distances in subtractive clustering.

Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.7813 \pm 0.0098	0.8001 \pm 0.0001	0.7649 \pm 0.0188	0.6359 \pm 0.0127	0.7568 \pm 0.0237	0.5411 \pm 0.0116
Mixed	0.6633 \pm 0.0110	0.7447 \pm 0.0112	0.5950 \pm 0.0172	0.6233 \pm 0.0211	0.7382 \pm 0.0349	0.5372 \pm 0.0095
Ham	0.6820 \pm 0.0121	0.7422 \pm 0.0123	0.6304 \pm 0.0230	0.6187 \pm 0.0198	0.7373 \pm 0.0345	0.5277 \pm 0.0145
Jac	0.6916 \pm 0.0134	0.7462 \pm 0.0116	0.6445 \pm 0.0261	0.6133 \pm 0.0241	0.7311 \pm 0.0380	0.5273 \pm 0.0131

Table 5.5: Comparing the RUS on different distances in subtractive clustering.

Dimensionality Reduction

Theoretically, another way that may improve the results is using dimensionality reduction, in order to eliminate the irrelevant and duplicate information, which can difficult the clustering process. Dimensionality reduction techniques create a new low-dimensional space, where every attribute is continuous. With this in mind, we opted to use only the Euclidean distance in these tests.

The algorithms chosen were PCA and gaussian KPCA. One of their limitations is that they do not know how many of the most relevant dimensions should be chosen. In tables

A.5 and A.6 in the appendix A we tried different numbers of dimensions and even an eigenvalue-based estimator which gave an estimation of the number of dimensions according to the data obtained in the low-dimensional space. For KPCA we also parameterized the gaussian variance.

In terms of dimensions, the best results were obtained with a fixed dimension and never with the estimator. PCA had better results with 4 dimensions and KPCA with 9 dimensions and variance of 10. Table 5.6 presents the validation of those two parameterized techniques. It can be seen that the results did not improve, since they are all inferior to the GRACE's Gmean. KPCA had slightly better results, but it does not seem significant.

Subtractive Clustering - Dimensionality Reduction						
Alg (dim)	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
PCA (4)	0.7470 \pm 0.0068	0.7904 \pm 0.0071	0.7065 \pm 0.0085	0.6245 \pm 0.0213	0.7375 \pm 0.0359	0.5392 \pm 0.0116
KPCA (9)	0.7750 \pm 0.0001	0.8001 \pm 0.0001	0.7507 \pm 0.0002	0.6399 \pm 0.0166	0.7047 \pm 0.0290	0.5946 \pm 0.0105

Table 5.6: Comparing the dimensionality reduction techniques in subtractive clustering.

Features Selection

An alternative to dimensionality reduction techniques are the feature selection methods, which assign weights to every feature, according to its importance and relevance. The data is not transformed, which means that it makes sense to test the different distances. In terms of algorithms we chose to test FCBF, Gini index and Relief-F. The last one has two parameters: size of the neighborhood (k) and number of samples used (m). Their parameterization can be found in tables A.7, A.8, A.9 and A.10.

In table 5.7 we can see the results using the algorithm FCBF, which chooses the most relevant features. However, it seems that its selection was rather bad, because with the exception of the Euclidean distance, all distances presented similar results to that obtained only with GRACE. This indicates that the clusters being formed assign almost all the patients to that risk scores, which is not how our approach is supposed to work.

Subtractive Clustering - FCBF						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.6204 \pm 0.0059	0.7273 \pm 0.0114	0.5303 \pm 0.0028	0.6235 \pm 0.0193	0.7446 \pm 0.0337	0.5309 \pm 0.0086
Mixed	0.6405 \pm 0.0000	0.7716 \pm 0.0001	0.5318 \pm 0.0000	0.6442 \pm 0.0030	0.7856 \pm 0.0082	0.5318 \pm 0.0000
Ham	0.6405 \pm 0.0000	0.7716 \pm 0.0001	0.5318 \pm 0.0000	0.6442 \pm 0.0030	0.7856 \pm 0.0082	0.5318 \pm 0.0000
Jac	0.6406 \pm 0.0002	0.7716 \pm 0.0001	0.5319 \pm 0.0004	0.6442 \pm 0.0030	0.7856 \pm 0.0082	0.5318 \pm 0.0011

Table 5.7: Comparing distances using FCBF in subtractive clustering.

Both Gini index and Relief-F assign weights to the features, opposed to the binary selection done by FCBF. Tables 5.8 and 5.9 show that both had better results than the latter. Yet, only Euclidean distance presents slightly better results than GRACE in terms of Gmean, which are worse in terms of sensitivity.

Subtractive Clustering - Gini Index						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.6523 \pm 0.0105	0.7740 \pm 0.0028	0.5509 \pm 0.0168	0.6468 \pm 0.0061	0.7831 \pm 0.0116	0.5385 \pm 0.0073
Mixed	0.6461 \pm 0.0012	0.7537 \pm 0.0037	0.5541 \pm 0.0025	0.6450 \pm 0.0078	0.7656 \pm 0.0165	0.5489 \pm 0.0057
Ham	0.6379 \pm 0.0019	0.7559 \pm 0.0092	0.5392 \pm 0.0056	0.6345 \pm 0.0092	0.7577 \pm 0.0209	0.5397 \pm 0.0076
Jac	0.6353 \pm 0.0036	0.7518 \pm 0.0109	0.5377 \pm 0.0059	0.6316 \pm 0.0164	0.7552 \pm 0.0324	0.5354 \pm 0.0101

Table 5.8: Comparing distances using Gini index in subtractive clustering.

Subtractive Clustering - Relief-F						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.6966 \pm 0.0224	0.7822 \pm 0.0076	0.6238 \pm 0.0353	0.6554 \pm 0.0089	0.7743 \pm 0.0181	0.5598 \pm 0.0133
Mixed	0.6431 \pm 0.0017	0.7711 \pm 0.0011	0.5364 \pm 0.0028	0.6426 \pm 0.0061	0.7774 \pm 0.0163	0.5357 \pm 0.0047
Ham	0.6406 \pm 0.0004	0.7712 \pm 0.0011	0.5323 \pm 0.0014	0.6412 \pm 0.0123	0.7806 \pm 0.0188	0.5326 \pm 0.0021
Jac	0.6355 \pm 0.0042	0.7451 \pm 0.0136	0.5430 \pm 0.0085	0.6278 \pm 0.0168	0.7398 \pm 0.0381	0.5422 \pm 0.0097

Table 5.9: Comparing distances using Relief-F in subtractive clustering.

Combining Tests

Despite the poor results obtained in both dimensionality reduction and feature selection techniques, we decided to combine the tests with better results (KPCA and Relief-F with Euclidean distance) with ROS and RUS to try to improve the results. This can be found in table 5.10. As it can be seen, the results were not improved at all. In Relief-F case all the metrics were worse than the ones obtained without balancing methods, and in KPCA only the sensitivity increased.

Subtractive Clustering - Balancing + Dimensionality Reduction / Features Selection						
	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
ROS + KPCA	0.6797 \pm 0.0037	0.7969 \pm 0.0079	0.5802 \pm 0.0066	0.6028 \pm 0.0243	0.7122 \pm 0.0429	0.5229 \pm 0.0158
RUS + KPCA	0.7813 \pm 0.0098	0.8001 \pm 0.0001	0.7649 \pm 0.0188	0.6338 \pm 0.0110	0.7495 \pm 0.0257	0.5438 \pm 0.0086
ROS + Relief-F	0.6078 \pm 0.0180	0.7145 \pm 0.0228	0.5189 \pm 0.0173	0.6107 \pm 0.0258	0.7313 \pm 0.0461	0.5209 \pm 0.0203
RUS + Relief-F	0.7786 \pm 0.0127	0.7990 \pm 0.0031	0.7608 \pm 0.0227	0.6278 \pm 0.0187	0.7507 \pm 0.0319	0.5337 \pm 0.0120

Table 5.10: Best tests of dimensionality reduction and features selection using balancing algorithms (ROS and RUS).

From the parameterization and testing of the clustering patients approach with subtractive clustering, it seems that it is better to simply normalize the data and use a mixed distance, because all the other algorithms could not improve the results. For instance, there was no situation in which the balancing techniques were better than the test without them. In terms of dimensionality reduction and feature selection methods, the latter worked better, but was still not enough. Overall, this approach using a mixed distance and normalizing increases slightly Gmean and specificity but it does not seem significant when compared with the GRACE performance.

5.1.2 Fuzzy C-means

The main algorithm in this approach (Clustering Patients) is supported on clustering strategies. Therefore, it is important to test different clustering techniques. Fuzzy C-means was one of the algorithms tested, since it creates membership functions for each point in a cluster that permits the creation of fuzzy rules. Basically, it is a fuzzy version of the classic k-means, which iteratively improves the centers of the clusters. A center is calculated as the mean of the points that belong to a cluster. Taking this into consideration, we decided that in this algorithm it makes no sense to use a different distances other than the Euclidean, because the centers, as means, are not binary or nominal and the distance that is calculated is between a point and the center. This means that it would not make sense to use a mixed or nominal distance.

Balancing Data

As in k-means, this clustering algorithm requires that the number of clusters is given for it to work. In table A.11 we tested different numbers, and 5 clusters seem to be the one

with the best results. Using that number, we tested the algorithm using 30 runs, without and with balancing techniques (ROS and RUS). These results can be found on table 5.11.

The initial results obtained with fuzzy c-means were very poor, both in the training and testing stage. The sensitivity was reduced, while the specificity increased only in the version without balancing. However, in general all the results were very lower compared to the ones obtained with GRACE, even the one with RUS, which presented the best results.

Fuzzy C-means - Balancing						
	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
-	0.6303 \pm 0.0063	0.6576 \pm 0.0080	0.6049 \pm 0.0094	0.6195 \pm 0.0198	0.6641 \pm 0.0329	0.6023 \pm 0.0137
ROS	0.6270 \pm 0.0075	0.6874 \pm 0.0171	0.5740 \pm 0.0120	0.6159 \pm 0.0183	0.6881 \pm 0.0311	0.5698 \pm 0.0165
RUS	0.6499 \pm 0.0187	0.7093 \pm 0.0116	0.6010 \pm 0.0340	0.6267 \pm 0.0236	0.7256 \pm 0.0396	0.5564 \pm 0.0142

Table 5.11: Comparing the Fuzzy C-means algorithm without and with balancing techniques (ROS and RUS).

Dimensionality Reduction

Fuzzy c-means was also tested with dimensionality reduction techniques. We opted to use 5 dimensions for PCA, and 6 dimensions and 3 of variance for KPCA according to the results obtained in tables A.12 and A.13. This parameterization validation is in table 5.12. Both algorithms had very similar results, with KPCA with slightly better Gmean, and PCA with slightly higher sensitivity. In spite of that, the results are still lower than GRACE, especially the sensitivity.

Fuzzy C-means - Dimensionality Reduction						
Alg (dim)	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
PCA (5)	0.6146 \pm 0.0107	0.6817 \pm 0.0112	0.5570 \pm 0.0205	0.6020 \pm 0.0339	0.6827 \pm 0.0456	0.5528 \pm 0.0221
KPCA (6)	0.6164 \pm 0.0045	0.6509 \pm 0.0089	0.5848 \pm 0.0093	0.6027 \pm 0.0202	0.6461 \pm 0.0307	0.5834 \pm 0.0168

Table 5.12: Comparing the dimensionality reduction techniques in Fuzzy C-means.

Features Selection

The Fuzzy C-means implementation in Matlab does not allow weights in its distance. Because of that we decided to simply select the most relevant features by their weights. FCBF already does this automatically, but both Gini index and Relief-F must be parameterized to decide the most adequate number of features which is considered relevant. This parameterization (tables A.14 and A.15) led us to use 3 features in Gini index and 7 features, a neighborhood of 5 and 50 samples in Relief-F.

Table 5.13 contains the mean of 30 runs using each one of the feature selection algorithms. FCBF clearly presents the best results, both in terms of Gmean and sensitivity. This is also the best performance obtained using Fuzzy C-means. However, it is still very poor compared with subtractive clustering or GRACE.

Fuzzy C-means - Feature Selection						
Alg (Feat)	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
FCBF	0.6450 \pm 0.0065	0.7167 \pm 0.0082	0.5832 \pm 0.0103	0.6499 \pm 0.0193	0.7386 \pm 0.0385	0.5837 \pm 0.0119
Gini(3)	0.6542 \pm 0.0069	0.7207 \pm 0.0088	0.5956 \pm 0.0123	0.6335 \pm 0.0227	0.6943 \pm 0.0337	0.5956 \pm 0.0176
Relief-F (7)	0.6450 \pm 0.0047	0.6811 \pm 0.0087	0.6122 \pm 0.0095	0.6293 \pm 0.0278	0.6753 \pm 0.0425	0.6124 \pm 0.0137

Table 5.13: Comparing the feature selection techniques in Fuzzy C-means.

Combining Tests

We tried to improve the best result obtained with FCBF using balancing techniques

(table 5.14). RUS once again performed better, but was a result lower than the one using only FCBF.

Fuzzy C-means - Balancing / FCBF						
Alg	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
ROS	0.6262 ± 0.0072	0.7338 ± 0.0140	0.5359 ± 0.0101	0.6192 ± 0.0191	0.7283 ± 0.0349	0.5366 ± 0.0133
RUS	0.6506 ± 0.0169	0.7431 ± 0.0116	0.5739 ± 0.0289	0.6307 ± 0.0218	0.7417 ± 0.0382	0.5481 ± 0.0133

Table 5.14: Combining FCBF balancing techniques (ROS and RUS) in Fuzzy C-means.

Using Fuzzy C-means produced unsatisfactory results. Not only where they clearly worse than using only GRACE, but were also inferior to the ones using subtractive clustering. In general they seem to improve the specificity, while decreasing the sensitivity, which lead to worse results.

5.1.3 Decision Tree

Decision trees are very similar to hierarchical clustering, and very interpretable, because a tree is like a system of rules that classify an object into a category. Therefore, it seemed reasonable to use decisions tree as a third clustering algorithm. For this, we discretized the data using rounding the numbers. This eliminates some detail on the data, which may help reduce the overfitting possibility. Besides, we also tested several levels of pruning (see table A.16).

In table 5.15 we present the results of CART without and with balancing techniques. As it can be seen the results are very unsatisfactory, with high levels of specificity, but very low sensitivity. This is even worse with ROS, which has a moderate rate of sensitivity for the training dataset, but the worst sensitivity in the test dataset seen so far. This seems to indicate that the tree is overfitting. RUS eliminates some of the problem, but the results are still very low compared with the ones obtained with the other clustering algorithms. Considering this, we decided to forsake the use of decision trees in this approach.

Decision Tree						
Alg (level)	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
- (0)	0.6794 ± 0.0085	0.6304 ± 0.0189	0.7360 ± 0.0211	0.5102 ± 0.0562	0.4286 ± 0.0561	0.7216 ± 0.0238
ROS (2)	0.7759 ± 0.0106	0.6574 ± 0.0171	0.9177 ± 0.0036	0.3801 ± 0.0493	0.2614 ± 0.0335	0.8630 ± 0.0122
RUS (1)	0.6764 ± 0.0165	0.6052 ± 0.0236	0.7646 ± 0.0334	0.5577 ± 0.0423	0.5072 ± 0.0489	0.6771 ± 0.0205

Table 5.15: Comparing the Decision Tree algorithm without and with balancing techniques (ROS and RUS).

Considering the results obtained it has become clear that, for this approach, the subtractive clustering is the most appropriate clustering algorithm. Not only it preserved better the highest sensitivity achieved by a risk score (GRACE) than the other clustering algorithms, but also produced a solution that seems to surpass the use of a single risk score. This solution was obtained using normalization plus a mixed distance, and was able to keep GRACE's sensitivity, while obtaining a higher specificity than any of the risk scores can obtain. Unfortunately, this result could not be improved by using balancing methods, dimensionality reduction or features selection techniques.

5.2 Dividing by Scores

The dividing by scores approach is very similar to the clustering patients approach. The main difference is the fact that it considers that the groups are determined by how well they are classified by the risk scores. Then, the rules to assign new patients to these groups are found using clustering algorithms. Supposing that those rules were perfect, then all the patients in the training dataset would be assigned to the group where it belongs, obtaining the results found in table 5.16.

The algorithms used were the same as in the previous approach: subtractive clustering, FCM and CART. In this section we present the results obtained for each one of them. Additional parameterization can be found in the appendix B.

Gmean	SE	SP
0.7761 ± 0.0000	0.8001 ± 0.0001	0.7529 ± 0.0000

Table 5.16: Results given if all the patients were assigned to the correct risk score in the training dataset.

5.2.1 Subtractive Clustering

In order to apply subtractive clustering in this approach, we tested different values of r_a for each distance (see tables B.1, B.2, B.3 and B.4). The Euclidean, mixed and Hamming distances seemed to perform better with $r_a = 0.1$, while Jaccard distance showed better results with $r_a = 0.6$.

Distance Comparison

In table 5.17 the mean of 30 runs using different distances can be found. Once again, like in the clustering patients approach, the mixed distance obtained the best results. However, only the Euclidean distance seems to classify correctly all the patients in the training dataset, since its results are similar to the ones in table 5.16. Regardless, this did not translate in a better performance. This implies that those rules are not totally capable of assigning new patients to the correct group.

Subtractive Clustering - Comparing Distances						
Dist (r_a)	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc (0.1)	0.7761 ± 0.0000	0.8001 ± 0.0001	0.7529 ± 0.0000	0.6406 ± 0.0136	0.6954 ± 0.0181	0.6061 ± 0.0056
Mixed (0.1)	0.6372 ± 0.0014	0.7375 ± 0.0027	0.5507 ± 0.0015	0.6423 ± 0.0075	0.7516 ± 0.0156	0.5562 ± 0.0045
Ham (0.1)	0.7750 ± 0.0000	0.8001 ± 0.0001	0.7508 ± 0.0001	0.6384 ± 0.0155	0.6969 ± 0.0234	0.6002 ± 0.0063
Jac (0.6)	0.6333 ± 0.0020	0.6301 ± 0.0038	0.6368 ± 0.0019	0.6133 ± 0.0224	0.6388 ± 0.0307	0.6194 ± 0.0074

Table 5.17: Comparing the use of different distances in subtractive clustering.

Balancing Data

Applying balancing techniques (see tables 5.18 and 5.19) did not improve the results at all. Both in RUS and ROS the best distance was the mixed, but ROS had slightly better results in all distances. As in the clustering patients approach, these techniques produced worse results than the ones obtained without them.

Dimensionality Reduction

In this approach, we also only used Euclidean distance with the dimensionality reduction methods. By their parameterization (tables B.5 and B.6), we chose to apply 9 dimensions

Subtractive Clustering - ROS						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.6429 \pm 0.0029	0.7718 \pm 0.0065	0.5358 \pm 0.0013	0.6198 \pm 0.0058	0.7286 \pm 0.0118	0.5363 \pm 0.0046
Mixed	0.6508 \pm 0.0031	0.7671 \pm 0.0070	0.5524 \pm 0.0020	0.6306 \pm 0.0064	0.7279 \pm 0.0107	0.5557 \pm 0.0063
Ham	0.6435 \pm 0.0030	0.7718 \pm 0.0065	0.5367 \pm 0.0017	0.6206 \pm 0.0057	0.7286 \pm 0.0118	0.5376 \pm 0.0051
Jac	0.6494 \pm 0.0055	0.7217 \pm 0.0112	0.5848 \pm 0.0037	0.6216 \pm 0.0161	0.6913 \pm 0.0243	0.5728 \pm 0.0113

Table 5.18: Comparing the ROS on different distances in subtractive clustering.

Subtractive Clustering - RUS						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.7791 \pm 0.0119	0.7994 \pm 0.0013	0.7614 \pm 0.0220	0.6097 \pm 0.0201	0.6926 \pm 0.0345	0.5536 \pm 0.0111
Mixed	0.7034 \pm 0.0233	0.7790 \pm 0.0064	0.6413 \pm 0.0384	0.6279 \pm 0.0166	0.7273 \pm 0.0288	0.5527 \pm 0.0110
Ham	0.7790 \pm 0.0120	0.7994 \pm 0.0013	0.7613 \pm 0.0221	0.6205 \pm 0.0218	0.7091 \pm 0.0325	0.5591 \pm 0.0119
Jac	0.7380 \pm 0.0149	0.7724 \pm 0.0096	0.7081 \pm 0.0234	0.6276 \pm 0.0201	0.7554 \pm 0.0343	0.5293 \pm 0.0120

Table 5.19: Comparing the RUS on different distances in subtractive clustering.

to PCA and 5 dimensions and 5 of variance to KPCA. When compared with the original results using only Euclidean distance, it is noticeable that the sensitivity improved, especially in PCA, in which the Gmean increased from 0.6423 to 0.6538. Although this represents a Gmean superior to that of any risk score alone, the sensitivity is too lower when compared to GRACE, or even to the best solution obtained in the clustering patients approach.

Subtractive Clustering - Dimensionality Reduction						
Alg (dim)	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
PCA (9)	0.7761 \pm 0.0000	0.8001 \pm 0.0001	0.7529 \pm 0.0000	0.6538 \pm 0.0150	0.7284 \pm 0.0285	0.5959 \pm 0.0094
KPCA (5)	0.7761 \pm 0.0000	0.8001 \pm 0.0001	0.7529 \pm 0.0000	0.6371 \pm 0.0200	0.7134 \pm 0.0308	0.5856 \pm 0.0105

Table 5.20: Comparing the dimensionality reduction techniques in subtractive clustering.

Features Selection

In the subtractive clustering in the clustering patients approach, the selection features methods improved some of the results obtained. Therefore, it is worthy to try them also in this approach. We tested it with the same algorithms: FCBF, Gini index and Relief-F (parameterized in table B.7). However, as it can be seen in tables 5.21, 5.22, and 5.23, it did not improve any result, when compared with the results obtained without selection features. The FCBF results were particularly poor, with very low specificity rates.

Subtractive Clustering - FCBF						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.5840 \pm 0.0042	0.7089 \pm 0.0038	0.4822 \pm 0.0067	0.5804 \pm 0.0124	0.7189 \pm 0.0146	0.4850 \pm 0.0108
Mixed	0.5479 \pm 0.0032	0.6973 \pm 0.0035	0.4318 \pm 0.0067	0.5457 \pm 0.0092	0.7172 \pm 0.0138	0.4317 \pm 0.0092
Ham	0.5809 \pm 0.0040	0.6951 \pm 0.0041	0.4864 \pm 0.0067	0.5747 \pm 0.0144	0.7025 \pm 0.0203	0.4858 \pm 0.0100
Jac	0.6334 \pm 0.0034	0.6582 \pm 0.0054	0.6098 \pm 0.0046	0.6290 \pm 0.0165	0.6850 \pm 0.0291	0.5934 \pm 0.0080

Table 5.21: Comparing distances using FCBF in subtractive clustering.

Subtractive Clustering - Gini Index						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.6918 ± 0.0165	0.7633 ± 0.0080	0.6293 ± 0.0237	0.6356 ± 0.0184	0.7270 ± 0.0298	0.5683 ± 0.0109
Mixed	0.6337 ± 0.0019	0.7319 ± 0.0040	0.5488 ± 0.0020	0.6306 ± 0.0084	0.7346 ± 0.0172	0.5504 ± 0.0051
Ham	0.7417 ± 0.0132	0.7866 ± 0.0059	0.7013 ± 0.0197	0.6368 ± 0.0185	0.7104 ± 0.0287	0.5858 ± 0.0083
Jac	0.6369 ± 0.0030	0.6469 ± 0.0058	0.6275 ± 0.0035	0.6249 ± 0.0189	0.6644 ± 0.0260	0.6088 ± 0.0068

Table 5.22: Comparing distances using Gini index in subtractive clustering.

Subtractive Clustering - Relief-F						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.6992 \pm 0.0228	0.7674 \pm 0.0106	0.6397 \pm 0.0333	0.6397 \pm 0.0167	0.7303 \pm 0.0298	0.5707 \pm 0.0125
Mixed	0.6279 \pm 0.0022	0.7157 \pm 0.0023	0.5511 \pm 0.0038	0.6265 \pm 0.0054	0.7291 \pm 0.0103	0.5477 \pm 0.0074
Ham	0.6217 \pm 0.0022	0.7138 \pm 0.0016	0.5417 \pm 0.0037	0.6209 \pm 0.0065	0.7271 \pm 0.0119	0.5400 \pm 0.0076
Jac	0.6335 \pm 0.0035	0.6364 \pm 0.0051	0.6310 \pm 0.0048	0.6159 \pm 0.0202	0.6516 \pm 0.0302	0.6102 \pm 0.0067

Table 5.23: Comparing distances using Relief-F in subtractive clustering.

Combining Tests

The best result in this approach using subtractive clustering was the one that employed PCA with 9 dimensions and Euclidean distance. This was the best solution in terms of Gmean, but presented only a sensitivity of 0.7284, which could be improved using of ROS or RUS (table 5.24). However, that improvement would decrease the specificity. Overall, these results were worse than the ones obtained in the previous approach, and while they are better than the risk scores in terms of specificity, their sensitivity is too low to be considered good solutions.

Subtractive Clustering - Balancing / PCA						
Alg	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
ROS	0.7326 \pm 0.0041	0.7914 \pm 0.0074	0.6784 \pm 0.0028	0.6281 \pm 0.0118	0.7653 \pm 0.0204	0.5211 \pm 0.0121
RUS	0.7813 \pm 0.0098	0.8001 \pm 0.0001	0.7649 \pm 0.0188	0.6337 \pm 0.0148	0.7563 \pm 0.0265	0.5381 \pm 0.0120

Table 5.24: Best test (PCA) combined with balancing algorithms (ROS and RUS).

5.2.2 Fuzzy C-means

Fuzzy C-means was also tested in the dividing by scores approach. We started by parameterizing the number of clusters (table B.8), and then validated that number using 30 runs without and with ROS or RUS.

Balancing Data

It is possible to find on table 5.25 the results obtained with and without balancing techniques for this algorithm. Overall the results were similar to the ones obtained before by this algorithm in clustering patients approach: neither sensitivity nor specificity was high, and the general results were lower than the ones obtained by subtractive clustering. The difference is that the test without balancing techniques performed better this time than both RUS and ROS, which had very low specificity.

Fuzzy C-means - Balancing						
	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
-	0.6227 \pm 0.0010	0.6580 \pm 0.0013	0.5895 \pm 0.0017	0.6213 \pm 0.0158	0.6803 \pm 0.0213	0.5842 \pm 0.0063
ROS	0.6143 \pm 0.0042	0.6519 \pm 0.0077	0.5792 \pm 0.0024	0.5950 \pm 0.0184	0.6393 \pm 0.0262	0.5765 \pm 0.0074
RUS	0.6296 \pm 0.0205	0.6678 \pm 0.0102	0.5974 \pm 0.0356	0.5965 \pm 0.0212	0.6681 \pm 0.0364	0.5517 \pm 0.0164

Table 5.25: Comparing the Fuzzy C-means algorithm without and with balancing techniques (ROS and RUS).

Dimensionality Reduction

We used dimensionality reduction methods, in order to try to improve the results. The number of dimensions used in both PCA and KPCA were estimated by the eigenvalue-based estimator, and the variance used on KPCA was 8. These values were defined

according to the parameterization done in tables B.9 and B.10. Table 5.26 shows the results obtained.

Both algorithms improved the Gmean, especially PCA, but it was due to an increase of specificity, while the sensitivity remained relatively the same, and consequently it is still low.

Fuzzy C-means - Dimensionality Reduction						
Alg (dim)	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
PCA (int)	0.6389 ± 0.0018	0.6668 ± 0.0032	0.6124 ± 0.0015	0.6418 ± 0.0103	0.6953 ± 0.0158	0.6069 ± 0.0057
KPCA (int)	0.6381 ± 0.0015	0.6595 ± 0.0028	0.6176 ± 0.0013	0.6392 ± 0.0164	0.6875 ± 0.0218	0.6113 ± 0.0065

Table 5.26: Comparing the dimensionality reduction techniques in Fuzzy C-means.

Features Selection

Features selection, especially FCBF improved fuzzy c-means in the clustering patients approach. Therefore, we also tested it in this approach. We began by choosing the most adequate parameters (tables B.11 and B.12). Then, 30 runs were performed with those parameters. These results can be found in table 5.27.

Fuzzy C-means - Feature Selection						
Alg (Feat)	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
FCBF	0.6083 ± 0.0030	0.6609 ± 0.0067	0.5606 ± 0.0020	0.6052 ± 0.0176	0.6746 ± 0.0273	0.5599 ± 0.0160
Gini (9)	0.6222 ± 0.0052	0.6857 ± 0.0067	0.5653 ± 0.0063	0.6081 ± 0.0182	0.6776 ± 0.0304	0.5612 ± 0.0135
Relief-F (9)	0.6386 ± 0.0025	0.6829 ± 0.0044	0.5974 ± 0.0026	0.6341 ± 0.0142	0.6951 ± 0.0185	0.5921 ± 0.0062

Table 5.27: Comparing the feature selection techniques in Fuzzy C-means.

Gini index and FCBF did not produce any additional improvement to the results. Relief-F increased slightly both the sensitivity and specificity, but was not really significant or comparable to the improvement obtained with PCA.

Combining Tests

This algorithm once again did not produce satisfactory results. Even the test with PCA, which presented the best results, was very poor both in terms of sensitivity as specificity, and could not be improved using balancing techniques (table 5.24). Furthermore, in this approach the sensitivity was lower than 0.70, which is very low when compared with the almost 0.80 that GRACE can reach.

Fuzzy C-means - Balancing / PCA						
Alg	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
ROS	0.6137 ± 0.0038	0.6531 ± 0.0072	0.5771 ± 0.0020	0.5972 ± 0.0171	0.6485 ± 0.0204	0.5722 ± 0.0064
RUS	0.6298 ± 0.0175	0.6671 ± 0.0090	0.5985 ± 0.0330	0.6015 ± 0.0219	0.6741 ± 0.0372	0.5558 ± 0.0171

Table 5.28: Combining PCA with balancing techniques (ROS and RUS) in Fuzzy C-means.

5.2.3 Decision Tree

In this approach, we tested decision trees using different levels of pruning (see table B.13), and with the level that obtained the best results, we run 30 runs to verify the algorithm general behavior. These results are presented in table 5.29.

The results are still unsatisfactory and very poor when compared with the other algorithms. In this case the test without balancing methods has a bit worse performance

than the one in the clustering patients approach, while the other tests had better results. Nevertheless, the sensitivity obtained by RUS (best result) is only of 0.5689, which is far lower than the sensitivity of GRACE which is almost 0.80. Because of that, we did not pursue further this algorithm in this approach.

Decision Tree						
Alg (level)	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
- (2)	0.6116 \pm 0.0096	0.4223 \pm 0.0131	0.8902 \pm 0.0044	0.4891 \pm 0.0459	0.3469 \pm 0.0391	0.8292 \pm 0.0088
ROS (2)	0.7998 \pm 0.0042	0.7454 \pm 0.0069	0.8584 \pm 0.0050	0.4839 \pm 0.0580	0.3714 \pm 0.0518	0.7639 \pm 0.0105
RUS (1)	0.6840 \pm 0.0149	0.6330 \pm 0.0193	0.7424 \pm 0.0225	0.6090 \pm 0.0194	0.5689 \pm 0.0333	0.6664 \pm 0.0138

Table 5.29: Comparing the Decision Tree algorithm without and with balancing techniques (ROS and RUS).

The subtractive clustering was the most adequate algorithm for dividing by scores approach in an analogous way to what happened with the clustering patients approach. However, even those results were not as satisfactory as the ones obtained in the previous approach, because there was only a test with a bit higher Gmean than the ones found on the risk scores (subtractive clustering plus PCA), but considerably lower than the best result obtained so far.

5.3 Similarity Measures

The dividing by scores approach produced worse results than the clustering patients approach, which seems to indicate that the similarities between the patients correctly classified with the same risk score are not easy to find. However, this does not mean that there are no similarities between such patients. In this approach, instead of assigning the new patient to the most similar group, we assign it to the same risk score as the one the closest training instance belongs to. Even if it is difficult to obtain a set of rules using clustering, this approach using only similarity measures may be able to give an estimation of the most adequate risk score to use.

Distance Comparison

The distances used were the same as the ones used in the clustering algorithms: normalization and Euclidean or a mixed distance; or discretization and a nominal distance (Hamming and Jaccard). No balancing techniques were applied, because this approach does not employ a data mining algorithm, and balancing would not help the process and could even deteriorate it. ROS would introduce unnecessary repetitions, since we always chose the closest one, which means it only needs to appear once, and RUS eliminates instances that could be important for finding the closest instance.

Table 5.30 presents the results for the different distances using only this approach without additional processing. As expected the training results are always the same. This indicates that every distance assigns the same patients to the same risk score, which is supposed, since the training dataset is used to determine the most adequate risk scores, and it is also used to compare new instances. In terms of the testing dataset, the best result is obtained by the Hamming distance, with better specificity than any risk score but lower sensitivity than GRACE. This is still a very low result when compared with the best obtained by the clustering patients approach.

Similarity Measures - Comparing Distances						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.7761 \pm 0.0000	0.8001 \pm 0.0001	0.7529 \pm 0.0000	0.6468 \pm 0.0140	0.6676 \pm 0.0174	0.6464 \pm 0.0051
Mixed	0.7761 \pm 0.0000	0.8001 \pm 0.0001	0.7529 \pm 0.0000	0.6508 \pm 0.0130	0.6619 \pm 0.0163	0.6613 \pm 0.0037
Ham	0.7761 \pm 0.0000	0.8001 \pm 0.0001	0.7529 \pm 0.0000	0.6552 \pm 0.0121	0.7233 \pm 0.0180	0.6054 \pm 0.0041
Jac	0.7761 \pm 0.0000	0.8001 \pm 0.0001	0.7529 \pm 0.0000	0.6494 \pm 0.0133	0.6954 \pm 0.0157	0.6224 \pm 0.0051

Table 5.30: Comparing the use of different distances in the similarity measures approach.

Dimensionality Reduction

In this approach, we also tested dimensionality reduction and feature selection techniques. The parameterization of PCA and KPCA using a Euclidean distance may be found in appendix C, tables C.1 and C.2. As for Relief-F, its parameterization is on table C.3.

Using dimensionality reduction methods, the sensitivity of this approach was improved while compared with the one using Euclidean distance, which is the distance that we used in these tests. However, the specificity decreased. The most balanced result is the one obtained by PCA, which presented a Gmean of 0.6627. However, its sensitivity is low when compared with GRACE.

Similarity Measures - Dimensionality Reduction						
Alg (dim)	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
PCA(4)	0.7761 \pm 0.0000	0.8001 \pm 0.0001	0.7529 \pm 0.0000	0.6627 \pm 0.0142	0.7255 \pm 0.0206	0.6171 \pm 0.0085
KPCA(8)	0.7761 \pm 0.0000	0.8001 \pm 0.0001	0.7529 \pm 0.0000	0.6533 \pm 0.0099	0.7357 \pm 0.0195	0.5891 \pm 0.0058

Table 5.31: Comparing the dimensionality reduction techniques in similarity measures approach.

Features Selection

The selection of features done by FCBF was not very reasonable (table 5.32), because most of the distances present results similar to GRACE, which means that almost all the patients are being assigned to this risk scores. It is also interesting to note that using feature selection algorithms, leads to the fluctuation of the results for the training dataset. This means that the capacity of assigning a patient to the right group is lost. This is not necessary bad, and may help improve the results for the test dataset.

Similarity Measures - FCBF						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.6407 \pm 0.0005	0.7716 \pm 0.0001	0.5320 \pm 0.0008	0.6442 \pm 0.0030	0.7856 \pm 0.0082	0.5318 \pm 0.0000
Mixed	0.6407 \pm 0.0005	0.7716 \pm 0.0001	0.5320 \pm 0.0008	0.6442 \pm 0.0030	0.7856 \pm 0.0082	0.5318 \pm 0.0000
Ham	0.6407 \pm 0.0002	0.7716 \pm 0.0001	0.5320 \pm 0.0004	0.6439 \pm 0.0039	0.7847 \pm 0.0103	0.5319 \pm 0.0004
Jac	0.6406 \pm 0.0002	0.7716 \pm 0.0001	0.5319 \pm 0.0003	0.6439 \pm 0.0039	0.7847 \pm 0.0103	0.5319 \pm 0.0004

Table 5.32: Comparing distances using FCBF in the similarity measures approach.

There is a noticeable improvement of the results using Gini index, with the exception of the mixed distance. In particular the result obtained for the test dataset by Hamming distance is very similar to the best result obtained in clustering patients approach using mixed distance (Gmean = 0.6715 \pm 0.0077, SE = 0.7814 \pm 0.0135 SP = 0.5815 \pm 0.0083). However, the latter is still better in terms of sensitivity, and it is also more constant.

Relief-F was able to improve the results, but was less successful than Gini index. The best result using Hamming distance had only a Gmean of 0.6673. Nevertheless, it is interesting to note that this algorithm had the highest results in parameterization with a Gmean of approximately 0.68 (C.3). This and the good results obtained by Gini index,

Similarity Measures - Gini Index						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.7736 \pm 0.0028	0.7980 \pm 0.0026	0.7500 \pm 0.0036	0.6637 \pm 0.0177	0.6947 \pm 0.0262	0.6489 \pm 0.0075
Mixed	0.7736 \pm 0.0028	0.7980 \pm 0.0026	0.7500 \pm 0.0036	0.6481 \pm 0.0169	0.6778 \pm 0.0263	0.6390 \pm 0.0086
Ham	0.7720 \pm 0.0027	0.7971 \pm 0.0029	0.7478 \pm 0.0027	0.6718 \pm 0.0148	0.7196 \pm 0.0285	0.6386 \pm 0.0071
Jac	0.7720 \pm 0.0027	0.7971 \pm 0.0029	0.7478 \pm 0.0027	0.6671 \pm 0.0171	0.7126 \pm 0.0253	0.6387 \pm 0.0065

Table 5.33: Comparing distances using Gini index in the similarity measures approach.

made us believe that features selection could help improve our results further in this approach. Therefore, we decided to use a random search to find the most appropriate weights.

Similarity Measures - Relief-F						
Dist	Train Gmean	Train SE	Train SP	Test Gmean	Test SE	Test SP
Euc	0.7761 \pm 0.0000	0.8001 \pm 0.0001	0.7529 \pm 0.0000	0.6365 \pm 0.0202	0.6464 \pm 0.0271	0.6520 \pm 0.0082
Mixed	0.7761 \pm 0.0000	0.8001 \pm 0.0001	0.7529 \pm 0.0000	0.6555 \pm 0.0143	0.6725 \pm 0.0223	0.6547 \pm 0.0076
Ham	0.7758 \pm 0.0004	0.8001 \pm 0.0001	0.7523 \pm 0.0007	0.6673 \pm 0.0128	0.7111 \pm 0.0248	0.6372 \pm 0.0054
Jac	0.7756 \pm 0.0004	0.8001 \pm 0.0001	0.7520 \pm 0.0008	0.6661 \pm 0.0118	0.7078 \pm 0.0226	0.6382 \pm 0.0059

Table 5.34: Comparing distances using Relief-F in the similarity measures approach.

In order to find the most adequate weights, we randomly generated those using values in the range $[0.1, 0.2, \dots, 1.0]$. Then, we used cross-validation and choose the ones with Gmean superior to ≥ 0.68 , since the best result so far is Gmean = 0.67. From those we selected the ones that obtained the best results for 30 runs using the different distances. We present only the weights and the results for the test dataset, because the training results were always similar to the ones obtained without weights.

Both Euclidean and mixed distances (see tables 5.35 and 5.36), improved mainly in terms of specificity, while the sensitivity remained low (approximately 0.72). Therefore, these results were not very satisfactory, because GRACE obtains sensitivities around 0.78.

Similarity Measures - Euclidean			
Weights	Test Gmean	Test SE	Test SP
[0.8,0.1,1.0,0.1,0.8,1.0,0.5,0.4,0.7,0.2,0.2,0.5,0.4,0.5]	0.6815 \pm 0.0093	0.7187 \pm 0.0183	0.6570 \pm 0.0045
[0.3,0.3,0.8,0.9,0.5,1.0,0.1,0.4,0.8,0.1,0.3,0.3,0.4,0.3]	0.6797 \pm 0.0083	0.7223 \pm 0.0166	0.6498 \pm 0.0048
[1.0,0.2,0.6,0.5,0.8,0.8,0.4,0.6,0.5,0.2,0.9,0.1,0.4,0.8]	0.6820 \pm 0.0098	0.7179 \pm 0.0178	0.6586 \pm 0.0054

Table 5.35: Best weights found for Euclidean distance.

Similarity Measures - Mixed			
Weights	Test Gmean	Test SE	Test SP
[1.0,0.9,0.3,0.6,0.8,0.9,0.9,0.5,0.9,0.7,0.3,0.2,0.2,0.8]	0.6886 \pm 0.0107	0.7148 \pm 0.0191	0.6747 \pm 0.0052
[0.5,0.5,0.3,0.2,1.0,1.0,0.4,0.3,0.9,0.2,0.1,0.5,0.2,0.2]	0.6861 \pm 0.0097	0.7159 \pm 0.0182	0.6690 \pm 0.0056
[0.3,0.8,0.2,0.9,0.5,0.1,1.0,1.0,0.9,0.7,0.8,0.9,1.0,0.6]	0.6752 \pm 0.0098	0.7196 \pm 0.0176	0.6437 \pm 0.0048

Table 5.36: Best weights found for mixed distance.

The results obtained with Hamming and Jaccard distances are more interesting in the sense that there was a clear improvement of both sensitivity and specificity. especially, in the first and third line of tables 5.37 and 5.38, have rates of sensitivity around 0.78, which are very similar to the one in the GRACE risk score (0.7856), and specificity around 0.60, which is very superior to any one of the risk scores, that have almost random specificity near 0.50.

From these results, the best are probably the first and third using Jaccard distance, because they are more balanced (Gmean is higher). It is possible that the first weights are going to lead to best performances, since they maintain almost intact the sensitivity of GRACE, while improving the specificity, which is clearly better than any obtained by a single risk score.

Similarity Measures - Hamming				
Weights	Test Gmean	Test SE	Test SP	
[0.8,0.3,0.8,0.1,0.4,0.1,0.3,0.1,0.2,0.2,0.3,0.2,0.2,0.6]	0.6857 ± 0.0051	0.7822 ± 0.0104	0.6055 ± 0.0046	
[0.7,0.4,0.2,0.9,0.8,0.4,1.0,0.6,0.5,0.3,0.2,0.1,0.9,0.3]	0.6715 ± 0.0071	0.7593 ± 0.0134	0.6011 ± 0.0039	
[0.5,0.1,0.8,0.5,1.0,0.4,0.5,0.4,0.9,0.3,0.2,1.0,0.3,0.8]	0.6909 ± 0.0074	0.7783 ± 0.0138	0.6186 ± 0.0044	

Table 5.37: Best weights found for Hamming distance.

Similarity Measures - Jaccard				
Weights	Test Gmean	Test SE	Test SP	
[0.8,0.3,0.8,0.1,0.4,0.1,0.3,0.1,0.2,0.2,0.3,0.2,0.2,0.6]	0.6910 ± 0.0052	0.7822 ± 0.0104	0.6152 ± 0.0053	
[1.0,0.1,0.9,0.8,1.0,0.6,0.2,0.2,0.3,0.5,1.0,0.6,0.2,0.4]	0.6855 ± 0.0080	0.7511 ± 0.0157	0.6330 ± 0.0043	
[0.5,0.1,0.8,0.5,1.0,0.4,0.5,0.4,0.9,0.3,0.2,1.0,0.3,0.8]	0.6974 ± 0.0087	0.7767 ± 0.0172	0.6316 ± 0.0038	

Table 5.38: Best weights found for Jaccard distance.

It is important to mention that the weights do not give the relevance of the risk factors for assessing the risk of having a cardiovascular event. They give the how relevant they are to separate the patients into the different risk scores, by using similarities. Maybe one risk factor is really important for risk assessment, but it is not relevant in terms of separating patients into risk scores. Nevertheless, looking at the most relevant factors, can give us more insight into these results.

Table 5.39 presents the best weights obtained in Jaccard and the factors associated. It seems that they all give a fair amount of importance to the sex, if there was a myocardial infarction on the patient enrolment or not, the class of CSS, systolic blood pressure and know CAD disease. On the other hand, factors like age and ST depression which are present in all risk scores seem to be less relevant to the risk scores choice, maybe because they are well taken in consideration in all the risk scores.

Similarity Measures - Weights														
Sex	Age	EAM	RF	CCS>II	DEP ST	SBP	HR	KILLIP	TN	Creat	AAS	Angina	CAD	
0.8	0.3	0.8	0.1	0.4	0.1	0.3	0.1	0.2	0.2	0.3	0.2	0.2	0.6	
0.5	0.1	0.8	0.5	1.0	0.4	0.5	0.4	0.9	0.3	0.2	1.0	0.3	0.8	

Table 5.39: Analysis of the weights meaning.

This approach in the beginning did not show a great potential in terms of results, when compared with the results obtained in the clustering patients approach. However, it was possible to see by using the same feature selection methods as before, that they had more potential to be further improved. That is why, we decided to use a random search to try to improve the results. While this does not guarantee the best solution, it allowed us to explore this concept and to obtain two solutions, which seem to have better results than using a single risk score and seem even better than the best result obtained in the clustering patients approach. However, a statistical analysis is needed to confirm that.

5.4 Statistical Analysis

In this statistical analysis we compared the Gmean, sensitivity and specificity obtained in the test dataset by the risk scores and our three best solutions in the same conditions. Their means and standard deviations can be found in table 5.40.

A test that permits multiplicity of comparisons is needed, because we are comparing 5 distinct solutions. This test can be parametric or not, depending if all the data follows a normal distribution or not. For assessing it, we used a normality test (Kolmogorov-Smirnov) for each one of the variables. Regarding Gmean data, TIMI, PURSUIT and our third solution presented a $p < 0.05$, 0.004, 0.000 and 0.002 respectively, which means that we cannot say that their data is normal. In SE and SP, this also happened. In the former, GRACE ($p = 0.023$), our first solution ($p = 0.007$) and third solutions ($p = 0.000$) had $p < 0.05$, and in the latter GRACE presented $p = 0.000$ and TIMI $p = 0.027$. Therefore, it was very unlikely that they followed a normal distribution, and we opted to use a non-parametric test, the Friedman's ANOVA with significance level of $p < 0.05$, which was complemented with the post-hoc methods with Bonferroni correction (significance level was set at $p < \frac{0.05}{15} < 0.0033$).

Statistical Tests				
	Solution	Test Gmean	Test SE	Test SP
	GRACE	0.6442 ± 0.0030	0.7856 ± 0.0082	0.5318 ± 0.0001
	PURSUIT	0.5864 ± 0.0135	0.6391 ± 0.0129	0.5601 ± 0.0003
	TIMI	0.5398 ± 0.0100	0.6925 ± 0.0138	0.4379 ± 0.0004
	Solution 1: Mixed distance (Clustering patients approach)	0.6715 ± 0.0077	0.7814 ± 0.0135	0.5815 ± 0.0083
	Solution 2: [0.8,0.3,0.8,0.1,0.4,0.1,0.3,0.1,0.2,0.2,0.3,0.2,0.2,0.6]	0.6910 ± 0.0052	0.7822 ± 0.0104	0.6152 ± 0.0053
	Solution 3: [0.5,0.1,0.8,0.5,1.0,0.4,0.5,0.4,0.9,0.3,0.2,1.0,0.3,0.8]	0.6974 ± 0.0087	0.7767 ± 0.0172	0.6316 ± 0.0038

Table 5.40: Solutions that were tested in the statistical tests: all the risk scores, the mixed distance of the clustering patients approach and the best weights obtained using Jaccard distance in similarity measures approach.

5.4.1 Sensitivity

Sensitivity is very important because it expresses the ability to correctly identify the high risk patients, which can suffer a cardiovascular event in short term. Thus, a decrease on the sensitivity, when compared with the risk scores, it is not acceptable.

In the figure 5.1, the sensitivities of the risk scores and our solutions are compared. There is a clear difference between the risk scores, in which GRACE presents the highest sensitivity and TIMI the lowest. Considering this, one of our goals is to have a solution with at least the same sensitivity as GRACE. Both solution 1 and solution 2 seem to fulfill that goal, while solution 3 seems to have to some extent a lower sensitivity.

Friedman test indicates that there are really statistically significant differences in the sensitivities of the solutions and risk scores, $\chi^2(5) = 137.748$, $p = 0.000$. A post-hoc analysis with Wilcoxon signed-rank tests was conducted with a Bonferroni correction applied to identify those differences.

Comparing the risk scores between themselves there is a statistically significant difference in the sensitivities between GRACE ($Mdn^1 = 0.7833$) and TIMI ($Mdn = 0.6925$, $z^2 =$

¹Median.

²z-score.

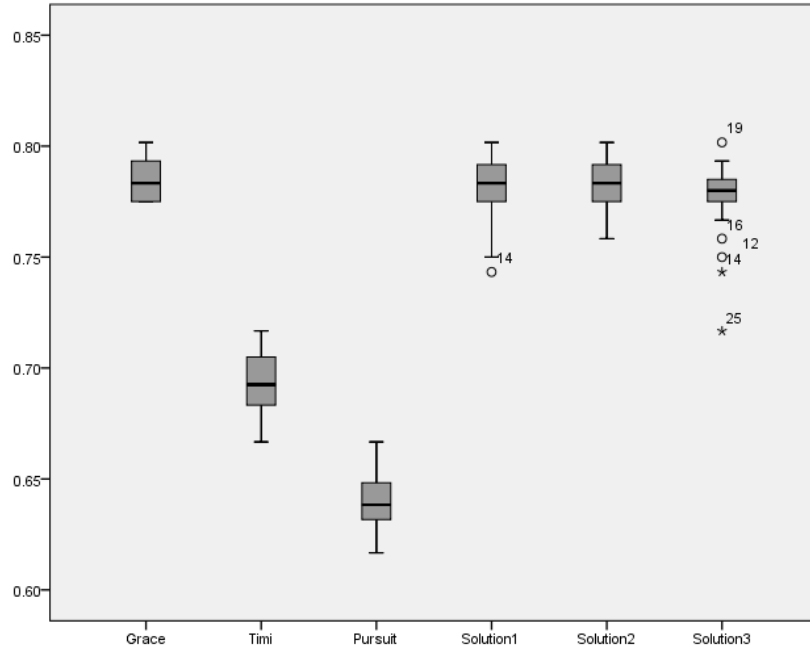


Figure 5.1: Sensitivity of the different solutions and risk scores.

-4.789 , $p = 0.000 < 0.0033$), and GRACE and PURSUIT ($Mdn = 0.6384$, $z = -4.793$, $p = 0.000 < 0.0033$). Additionally, it is statistically unlikely that TIMI sensitivity is not higher than the one from PURSUIT ($z = -4.787$, $p = 0.000 < 0.0033$).

No significant differences were found, when looking statistically at our solutions and GRACE ($Mdn = 0.7833$): solution 1 ($Mdn = 0.7833$, $z = -1.890$, $p = 0.059 > 0.0033$), solution 2 ($Mdn = 0.7833$, $z = -2.000$, $p = 0.046 > 0.0033$) and solution 3 ($Mdn = 0.7800$, $z = -2.585$, $p = 0.010 > 0.0033$); or when comparing them between themselves: solution 1 - solution 2 ($z = -0.302$, $p = 0.763 > 0.0033$), solution 1 - solution 3 ($z = -1.512$, $p = 0.131 > 0.0033$), and solution 2 - solution 3 ($z = -1.826$, $p = 0.068 > 0.0033$).

This analysis allow us to conclude that all our solutions have probably a sensitivity at least as good as GRACE, because there are no significant differences between them and GRACE or between themselves. Furthermore, this means that they present a sensitivity that cannot be said not higher than the ones presented by TIMI or PURSUIT, since GRACE's sensitivity is significantly different from the one of those risk scores.

5.4.2 Specificity

While sensitivity shows the ability to correctly classify high risk patients, specificity shows the ability to classify low risk patients. This is also important because there is no need for this kind of patient to undergo an invasive strategy. However, the risk scores presented a rather low specificity, around 0.50, which is almost random, as can be seen in figure 5.2. In terms of specificity our solutions seem to have better results, especially when compared with TIMI.

Once again we used Friedman test to verify if there are really significant differences as it appear in figure 5.2. This test indicated that this is true, $\chi^2(5) = 150.000$, $p = 0.000$. Hence, we applied post-hoc analysis with Wilcoxon signed-rank tests and Bonferroni

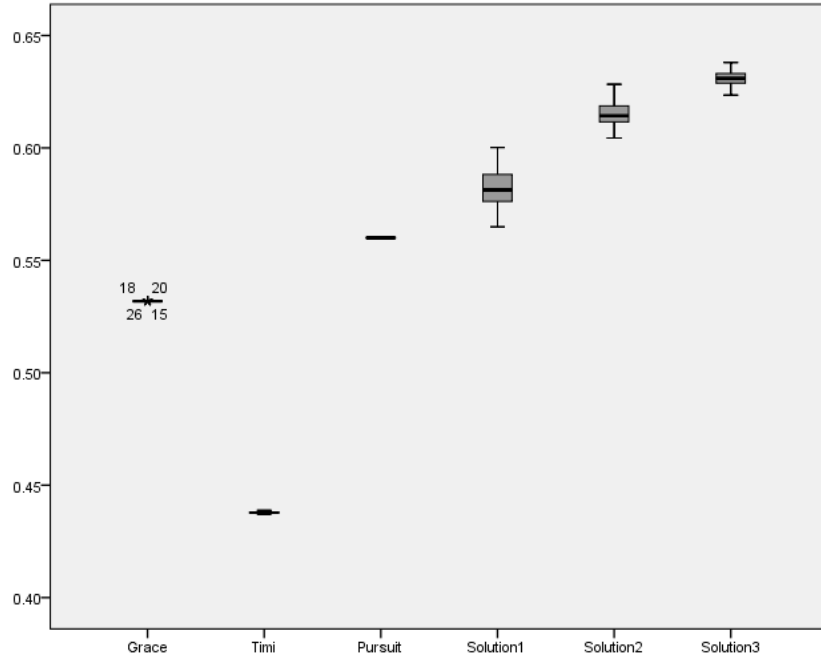


Figure 5.2: Specificity of the different solutions and risk scores.

correction.

Statistically speaking PURSUIT ($Mdn = 0.5600$) has a specificity significantly better than GRACE ($Mdn = 0.5318$, $z = -4.796$, $p = 0.000 < 0.0033$) or TIMI ($Mdn = 0.4378$, $z = -4.788$, $p = 0.000 < 0.0033$). In its turn GRACE has a higher specificity than TIMI ($z = -4.788$, $p = 0.000 < 0.0033$).

Because PURSUIT is obviously the best risk score in terms of specificity, we compared it to our solutions, and found out that all our solutions are statistically better than PURSUIT: solution 1 ($Mdn = 0.5813$, $z = -4.783$, $p = 0.000 < 0.0033$), solution 2 ($Mdn = 0.6143$, $z = -4.783$, $p = 0.000 < 0.0033$) and solution 3 ($Mdn = 0.6310$, $z = -4.783$, $p = 0.000 < 0.0033$).

There are also differences between our solutions: solution 2 is better than solution 1 ($z = -4.783$, $p = 0.000 < 0.0033$), solution 3 is better than solution 1 ($z = -4.782$, $p = 0.000 < 0.0033$) and solution 3 is also better than solution 2 ($z = -4.783$, $p = 0.000 < 0.0033$).

By this analysis, we may conclude that solution 3 is significantly better than all the risk scores and all our solutions.

5.4.3 Gmean

Gmean is the geometric mean between the specificity and sensitivity of the classifier, and expresses how well balanced its classification is, when looking at both classes. Figure 5.3 presents the boxplot with the Gmeans of the different risk scores and our solutions. It seems that GRACE is the risk score with higher Gmean, which is surpassed by all our solutions, particularly by solution 3.

Friedman test also indicate that there are significant differences, $\chi^2(5) = 146.248$, $p =$

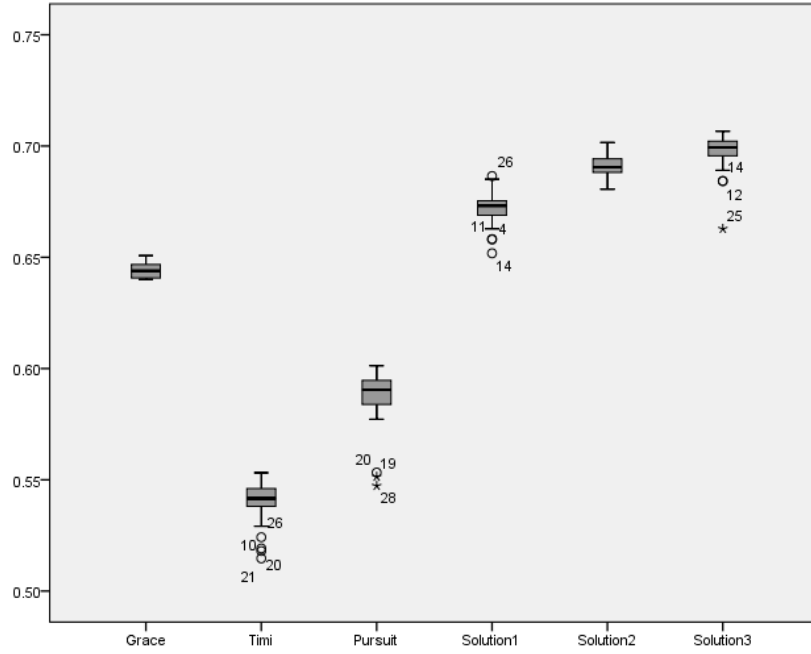


Figure 5.3: Geometric mean of the different solutions and risk scores.

0.000. Those differences were analyzed using Wilcoxon signed-rank tests with Bonferroni correction.

GRACE ($Mdn = 0.6439$) presents a significant superior Gmean than TIMI ($Mdn = 0.5416$, $z = -4.782$, $p = 0.000 < 0.0033$), and PURSUIT ($Mdn = 0.5905$, $z = -4.782$, $p = 0.000 < 0.0033$). The difference between PURSUIT ($Mdn = 0.5905$) and TIMI ($Mdn = 0.5416$), although smaller, it is also significant ($z = -4.782$, $p = 0.000 < 0.0033$).

Our solutions Gmeans are all significantly better than GRACE's: solution 1 ($Mdn = 0.6732$, $z = -4.782$, $p = 0.000 < 0.0033$), solution 2 ($Mdn = 0.6905$, $z = -4.783$, $p = 0.000 < 0.0033$) and solution 3 ($Mdn = 0.6994$, $z = -4.783$, $p = 0.000 < 0.0033$). This also means that they are significantly better than TIMI and PURSUIT.

Considerable differences between our solutions were also found: solution 2 is better than solution 1 ($z = -4.721$, $p = 0.000 < 0.0033$), but solution 3 is better than solution 1 ($z = -4.700$, $p = 0.000 < 0.0033$) and solution 3 ($z = -3.507$, $p = 0.000 < 0.0033$).

As expected the statistical analysis demonstrated that solution 3 has a significantly higher Gmean than any other of our solutions or risk scores. This is normal, because there were no significant differences in the sensitivity, and solution 3 specificity is the highest.

5.5 Discuss Results

The risk scores used in this research (GRACE, TIMI and PURSUIT) present already high results of sensitivity, but their specificity is close to 0.50, which means that the low risk patients' classification is almost random. Our idea was to find a way to select the most adequate risk score for each patient, leading to a better balanced risk assessment tool. With this goal in mind, we proposed three different approaches.

Approaches Proposed

Among the approaches, the clustering patients approach presented one of the best results without any need of pre-processing, using only a mixed distance and subtractive clustering. All the other approaches without pre-processing had worse results than it. Inclusively, the dividing by scores approach could only produce one satisfactory result when compared with GRACE (subtractive clustering plus PCA), and still had a very low sensitivity by comparison.

Our results seems to indicate that the groups of patients which are correctly classified with the same risk score present no coherent similarities that can be found in order to call them clusters. However, there are similarities between patients, because the similarity approach has some of the best results, obtained with feature selection using random search. It is possible that one cluster of the dividing by scores approach makes no sense, and it would be important to divide them in several clusters with more similar patients on it. If this is true, the first approach had more potential, but was probable that it could not produce the correct clusters due to the lack of patients. Therefore, it is normal that the similarity approach was able to find similarities even if there are not enough patients to form a cluster. especially, because it does not loose information in complex processes like clustering algorithm is.

The best solution produced was the solution 3 using the similarity approach and Jaccard distance combined by weights found by random search. While this result is not so good as the ideal result that could be theoretically reached, it maintains the sensitivity of GRACE which is the best among risk scores and, improves the specificity, which was 0.5318 on GRACE and 0.5601 on PURSUIT and increases to 0.6316 with low variation. These results are statistically significant.

Algorithms Used

Besides the approaches it is also important to discuss how the different algorithms tested performed in overall, which includes clustering algorithms, balancing techniques, dimensionality reduction and features selection methods.

In terms of clustering algorithms, subtractive clustering presented undeniably better results than FCM or CART. The clusters found by FCM led to results with lower sensitivity than the ones obtained by subtractive clustering, regardless of the data pre-processing. This may be because it does not allow the creation of randomly shaped clusters. CART, which seemed a good idea due to its interpretability, showed results with very high levels of specificity, but extremely low sensitivity when compared with the other algorithms. This seems to imply that the algorithm is only focused on the low risk patients' classification, and assigns to the new instances the risk scores that incorrectly classify high risk patients similar to them.

Both balancing techniques (RUS and ROS), decreased the performance of the classifier more often than not. Unexpectedly, RUS, which decreases the number of instances, could in some cases improve the classifier, as could be seen in the first approach using fuzzy-c means and CART. This was probably due to the fact that the elimination of some instances of the low risk class, allowed the classifier to focus more on the high risk class, and increased the sensitivity.

Dimensionality reduction methods most of the times increased the results obtained with the Euclidean distance. However, they had the disadvantage of converting all the data into a continuous subspace, with no distinction of variable types as it was present in the original data. This information seems to be relevant, because the best results were obtained with mixed or nominal distances, which could not be used in a continuous space.

Feature selection techniques were able to produce more satisfactory results than the dimensionality reduction, especially when using attribution of weights, instead of the mere selection. This may happen, because they maintain the nature of the features, giving them only different relevance. This way it is possible to test all the distances and keep most of the variables information intact.

Chapter 6

Conclusions

Short-term risk assessment for cardiovascular diseases, such as risk scores, can improve the patients' treatment, and consequently lead to fewer or weaker cardiovascular events. The most popular risk scores for CAD, among the current ones, are probably TIMI [10], GRACE [11], and PURSUIT [12]. However, it is not easy to choose between them, because there is no clear evidence suggesting that one is superior to another one. Besides, they all assume that they work equally well for all the patient population, in spite of being created using only a small sample of it. This led us to believe that maybe one risk score is better for one type of patients than the other. This idea supports the proposed approach in our research, which is based on group personalization by dividing the patients into different groups and assigning to each one the risk score with the best performance for it. This eliminates the need to choose between risk scores, allows the incorporation of new knowledge as a risk score, and leads to a more effective and robust risk assessment tool.

In order to put our idea into practice we created three approaches: one that finds the groups using clustering algorithms, selects the most adequate risk score for each cluster, and then creates the rules to assign each patient to one of the groups (Clustering Patients approach); one that creates three groups that correspond to the patients that are correctly classified with each risk score, then finds rules to put new patients in one of the groups (Dividing by Scores approach); and one that also divides the patients by scores, but considers that a new patient belongs to one group using similarity measures (Similarity Measures approach). Additionally, we also implemented several data mining algorithms with those approaches, including balancing methods, dimensionality reduction and feature selection techniques.

In our research we compared all the approaches proposed with the performance of each individual risk score, which performed relatively well in terms of sensitivity, but presented specificities very close to 0.50 (almost a random classification of the low risk patients). The comparison allowed us to conclude that it was possible to improve the results using a combination of the risk scores by using personalization based on groups of patients, because both clustering patients and similarity measures approach were able to generate solutions that preserved the best sensitivity among the risk scores, while increasing the specificity. This permitted us to obtain a classification for the low risk patients higher than the initial 0.50%, which is not a random classification anymore.

The best solution obtained for the clustering patients approach used subtractive clustering, normalization and a mixed distance, which applied a Euclidean distance to the interval-based risk factors and a Hamming distance to the nominal ones. Its sensitivity is

not significantly different from the one obtained by GRACE (highest sensitivity among the risk scores), but its specificity increased until 11% more than the one obtained by GRACE (from 0.5318 ± 0.0001 to 0.5815 ± 0.0053). Comparing to PURSUIT, which was the highest specificity this is an increase of 5% (from 0.5601 ± 0.0003 to 0.5815 ± 0.0053).

In terms of the best solution obtained by the similarity measures approach, it was implemented rounding the continuous values to discretize them, which allowed us to apply a Jaccard distance. Furthermore, we searched for the most adequate weights for each risk factor using a random search. This solution was even better than the one obtained by the clustering patients approach, since it also preserved the GRACE's sensitivity, but increased more its specificity. Comparing to GRACE the specificity was increased until 19% (from 0.5318 ± 0.0001 to 0.6316 ± 0.0038), which represents 13% when compared to PURSUIT (from 0.5601 ± 0.0003 to 0.6316 ± 0.0038).

All those results were also analyzed statistically, which confirmed their significance. Considering this, it seems that the combination of risk scores, using similarity based measures and an adequate selection of weights for the risk factors, is the most favorable method to improve the risk assessment without the need to create a new risk score.

6.1 Future Work

In our research, we explored several approaches and algorithms. However, our time was limited and there are still a number of ideas that could be further explored to complement and perhaps improve the results obtained. Some of those ideas are depicted here.

In every study, the results obtained are very dependent of the data available. Idealistically, we would like to use a representative and well balanced dataset. Nevertheless, obtaining such a dataset is not an easy task. The Santa Cruz dataset, which was used by us, has only 7.6% of high risk patients. Therefore, it would be important to test our idea on additional new datasets, if possible.

The similarity measures approach presented the best results when compared with other ones. Thus, it would make sense to explore it further. This could be done by testing new similarity concepts, like considering that a new patient belongs to a group if its distance to the group's mean is the smallest, which is a concept similar to the one used on clustering. Another option would be to explore new methods to find the weights for the risk factors. In our research we simply used a random search, which is completely random and has no guarantee of finding a good solution. While it is impractical to search all the space, there are other methods that have the potential to find good solutions, such as genetic algorithms. These algorithms could evolve arrays of weights, where its fitness could be the Gmean obtained in a test dataset.

The dividing by scores approach did not present good results. However, it could also be improved in future work. We assigned to GRACE, the new patients that belong to the group of patients which are not correctly classified with any one of the risk scores. This was a justifiable solution in the sense that GRACE is the risk score with higher sensitivity, and there may be instances that could be well classified by it. Nevertheless, maybe those instances are few, since they are similar to the group of wrongly classified patients. Considering this, perhaps it would be more effective to create an additional risk assessment tool for classifying them.

Chapter 7

References

- [1] M. Nichols, N. Townsend, R. Luengo-Fernandez, J. Leal, A. Gray, P. Scarborough, and M. Rayner, *European Cardiovascular Disease Statistics 2012*. European Heart Network, Brussels, European Society of Cardiology, Sophia Antipolis, 2012.
- [2] A.D.A.M. Medical Encyclopedia [Internet]. Atlanta (GA): A.D.A.M., Inc.; ©2012. Coronary heart disease [updated 2012 Jun 22; cited 2012 December 20]; [about 4 p.]. Available from: <http://www.nlm.nih.gov/medlineplus/ency/article/007115.htm>.
- [3] National Public Health Partnership (2006). The Language of Prevention. Melbourne: NPHP.
- [4] H. Bueno and F. Fernández-Avilés, “Use of risk scores in acute coronary syndromes,” *Heart*, vol. 98, no. 2, pp. 162–168, 2012.
- [5] C. P. Cannon, W. S. Weintraub, L. A. Demopoulos, R. Vicari, M. J. Frey, N. Lakkis, F.-J. Neumann, D. H. Robertson, P. T. DeLucca, P. M. DiBattiste, C. M. Gibson, and E. Braunwald, “Comparison of Early Invasive and Conservative Strategies in Patients with Unstable Coronary Syndromes Treated with the Glycoprotein IIb/IIIa Inhibitor Tirofiban,” *New England Journal of Medicine*, vol. 344, no. 25, pp. 1879–1887, 2001.
- [6] O. Manfrini and R. Bugiardini, “Barriers to clinical risk scores adoption,” *European Heart Journal*, vol. 28, no. 9, pp. 1045–1046, 2007.
- [7] D. M. Eddy, “Variations in physician practice: the role of uncertainty,” *Health affairs*, vol. 3, no. 2, pp. 74–89, 1984.
- [8] D. Gamberger, N. Lavrac, and G. Krstacic, “Active subgroup mining: a case study in coronary heart disease risk group detection,” *Artificial Intelligence in Medicine*, vol. 28, no. 1, pp. 27–57, 2003.
- [9] M. Karaolis, J. Moutiris, D. Hadjipanayi, and C. Pattichis, “Assessment of the Risk Factors of Coronary Heart Events Based on Data Mining With Decision Trees,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 3, pp. 559–566, 2010.
- [10] A. Elliott, C. Marc, B. Peter, *et al.*, “The TIMI risk score for unstable angina/non-ST elevation MI: A method for prognostication and therapeutic decision making,” *JAMA: The Journal of the American Medical Association*, vol. 284, no. 7, pp. 835–842, 2000.

- [11] C. B. Granger, R. J. Goldberg, O. Dabbous, K. S. Pieper, K. A. Eagle, C. P. Cannon, F. Van de Werf, A. Avezum, S. G. Goodman, M. D. Flather, *et al.*, “Predictors of hospital mortality in the global registry of acute coronary events,” *Archives of Internal Medicine*, vol. 163, no. 19, pp. 2345–2353, 2003.
- [12] E. Boersma, K. S. Pieper, E. W. Steyerberg, R. G. Wilcox, W.-C. Chang, K. L. Lee, K. M. Akkerhuis, R. A. Harrington, J. W. Deckers, P. W. Armstrong, A. M. Lincoff, R. M. Califf, E. J. Topol, M. L. Simoons, and for the PURSUIT Investigators, “Predictors of Outcome in Patients With Acute Coronary Syndromes Without Persistent ST-Segment Elevation: Results From an International Trial of 9461 Patients,” vol. 101, no. 22, pp. 2557–2567, 2000.
- [13] C. Hamm, J.-P. Bassand, S. Agewall, J. Bax, E. Boersma, *et al.*, “ESC Guidelines for the management of acute coronary syndromes in patients presenting without persistent ST-segment elevation: The Task Force for the management of acute coronary syndromes (ACS) in patients presenting without persistent ST-segment elevation of the European Society of Cardiology (ESC),” *European Heart Journal*, vol. 32, no. 23, 2011.
- [14] J. Perk, G. De Backer, H. Gohlke, I. Graham, Z. Reiner, *et al.*, “European Guidelines on cardiovascular disease prevention in clinical practice (version 2012): The Fifth Joint Task Force of the European Society of Cardiology and Other Societies on Cardiovascular Disease Prevention in Clinical Practice (constituted by representatives of nine societies and by invited experts) developed with the special contribution of the European Association for Cardiovascular Prevention & Rehabilitation (EACPR),” *European Heart Journal*, vol. 33, no. 13, 2012.
- [15] F. Hobbs, “Cardiovascular disease: different strategies for primary and secondary prevention?,” *Heart*, vol. 90, no. 10, pp. 1217–1223, 2004.
- [16] C. P. Gale, S. O. M. Manda, C. F. Weston, J. S. Birkhead, P. D. Batin, and A. S. Hall, “Evaluation of risk scores for risk stratification of acute coronary syndromes in the Myocardial Infarction National Audit Project (MINAP) database,” *Heart*, vol. 95, no. 3, pp. 221–227, 2009.
- [17] P. d. A. Gonçalves, J. Ferreira, C. Aguiar, and R. Seabra-Gomes, “TIMI, PURSUIT, and GRACE risk scores: sustained prognostic value and interaction with revascularization in NSTEMI-ACS,” *Eur Heart J.*, vol. 26, no. 9, pp. 865–72, 2005.
- [18] R. Lyon, A. Morris, D. Caesar, S. Gray, and A. Gray, “Chest pain presenting to the emergency department—to stratify risk with grace or timi?,” *Resuscitation*, vol. 74, no. 1, pp. 90–93, 2007.
- [19] A. T. Yan, R. T. Yan, M. Tan, A. Casanova, M. Labinaz, K. Sridhar, D. H. Fitchett, A. Langer, and S. G. Goodman, “Risk scores for risk stratification in acute coronary syndromes: useful but simpler is not necessarily better,” *European Heart Journal*, vol. 28, no. 9, pp. 1072–1078, 2007.
- [20] D. A. Morrow, E. M. Antman, A. Charlesworth, R. Cairns, S. A. Murphy, J. A. de Lemos, R. P. Giugliano, C. H. McCabe, and E. Braunwald, “TIMI Risk Score for ST-Elevation Myocardial Infarction: A Convenient, Bedside, Clinical Score for Risk Assessment at Presentation: An Intravenous nPA for Treatment of Infarcting Myocardium Early II Trial Substudy,” *Circulation*, vol. 102, no. 17, pp. 2031–2037, 2000.

- [21] K. A. Fox, O. H. Dabbous, R. J. Goldberg, K. S. Pieper, K. A. Eagle, F. V. de Werf, Á. Avezum, S. G. Goodman, M. D. Flather, J. Frederick A Anderson, and C. B. Granger, “Prediction of risk of death and myocardial infarction in the six months after presentation with acute coronary syndrome: prospective multinational observational study (GRACE),” *BMJ*, vol. 333, no. 7578, pp. 1091–1094, 2006.
- [22] K. A. Eagle, M. J. Lim, O. H. Dabbous, *et al.*, “A validated prediction model for all forms of acute coronary syndrome: Estimating the risk of 6-month postdischarge death in an international registry,” *JAMA: The Journal of the American Medical Association*, vol. 291, no. 22, pp. 2727–2733, 2004.
- [23] Center of Outcomes Research [Internet]. Global Registry of Acute Coronary Events (GRACE) [cited 2012 December 21]; [about 5 p.]. Available from: <http://www.outcomes-umassmed.org/GRACE/>.
- [24] A. T. Yan, P. Jong, R. T. Yan, M. Tan, D. Fitchett, C.-M. Chow, M. T. Roe, K. S. Pieper, A. Langer, and S. G. Goodman, “Clinical trial derived risk model may not generalize to real-world patients with acute coronary syndrome,” *American heart journal*, vol. 148, no. 6, pp. 1020–1027, 2004.
- [25] R. S. Wright, J. L. Anderson, C. D. Adams, C. R. Bridges, D. E. C. Jr, S. M. Ettinger, F. M. Fesmire, T. G. Ganiats, H. Jneid, A. M. Lincoff, E. D. Peterson, G. J. Philippides, P. Theroux, N. K. Wenger, and J. P. Zidar, “2011 ACCF/AHA Focused Update Incorporated Into the ACC/AHA 2007 Guidelines for the Management of Patients With Unstable Angina/Non ST Elevation Myocardial Infarction: A Report of the American College of Cardiology Foundation/American Heart Association Task Force on Practice Guidelines”, journal = "Journal of the American College of Cardiology," vol. 57, no. 19, pp. 215–367, 2011.
- [26] National Clinical Guideline Centre for Acute and Chronic Conditions. Unstable angina and NSTEMI: the early management of unstable angina and non-ST-segment-elevation myocardial infarction. London (UK): National Institute for Health and Clinical Excellence (NICE); 2010 Mar. 26 p. (Clinical guideline; no. 94).
- [27] A. Tsymbal, S. Puuronen, and D. W. Patterson, “Ensemble feature selection with the simple bayesian classification,” *Information Fusion*, vol. 4, no. 2, pp. 87–100, 2003.
- [28] E. Bauer and R. Kohavi, “An empirical comparison of voting classification algorithms: Bagging, boosting, and variants,” *Machine Learning*, vol. 36, pp. 105–139, 1999.
- [29] G. Samsa, G. Hu, and M. Root, “Combining information from multiple data sources to create multivariable risk models: illustration and preliminary assessment of a new method,” *BioMed Research International*, vol. 2005, no. 2, pp. 113–123, 2005.
- [30] E. W. Steyerberg, *Clinical prediction models: a practical approach to development, validation, and updating*. Springer, 2009.
- [31] C. R. Twardy, A. E. Nicholson, K. B. Korb, J. McNeil, *et al.*, “Epidemiological data mining of cardiovascular bayesian networks,” *Electronic Journal of Health Informatics*, vol. 1, no. 1, pp. 1–13, 2006.
- [32] S. Paredes, *Integration of different risk assessment tools to improve the event risk assessment in cardiovascular disease patients*. PhD thesis, University of Coimbra, 2012.

- [33] L. Cordella, P. Foggia, C. Sansone, F. Tortorella, and M. Vento, “Reliability parameters to improve combination strategies in multi-expert systems,” *Pattern Analysis & Applications*, vol. 2, no. 3, pp. 205–214, 1999.
- [34] B. Zhang, *Adaptive Model Selection in Linear Mixed Models*. PhD thesis, UNIVERSITY OF MINNESOTA, 2009.
- [35] L. Todorovski and S. Džeroski, “Combining classifiers with meta decision trees,” *Machine learning*, vol. 50, no. 3, pp. 223–249, 2003.
- [36] N. Lavrac, E. Keravnou, and B. Zupan, “Intelligent data analysis in medicine,” *Encyclopedia of computer science and technology*, vol. 42, no. 9, pp. 113–157, 2000.
- [37] J. Habetha *et al.*, “The myheart project—fighting cardiovascular diseases by prevention and early diagnosis,” in *Conf Proc IEEE Eng Med Biol Soc*, pp. 6746–6749, 2006.
- [38] R. Bellazzi and B. Zupan, “Predictive data mining in clinical medicine: current issues and guidelines,” *international journal of medical informatics*, vol. 77, no. 2, pp. 81–97, 2008.
- [39] I. Kononenko, “Machine learning for medical diagnosis: history, state of the art and perspective,” *Artificial Intelligence in Medicine*, vol. 23, pp. 89–109, 2001.
- [40] M. J. Pazzani, S. Mani, and W. R. Shankle, “Acceptance by medical experts of rules generated by machine learning,” in *Methods of Information in Medicine*, vol. 40, pp. 380–385, 2001.
- [41] J. Quinlan, *C4.5: programs for machine learning*, vol. 1. Morgan kaufmann, 1993.
- [42] C. Tsien, H. Fraser, W. Long, R. Kennedy, *et al.*, “Using classification tree and logistic regression methods to diagnose myocardial infarction,” *Studies in health technology and informatics*, no. 1, pp. 493–497, 1998.
- [43] J. Cruz and D. Wishart, “Applications of machine learning in cancer prediction and prognosis,” *Cancer Informatics*, vol. 2, pp. 59–77, 2006.
- [44] R. Voss, P. Cullen, H. Schulte, and G. Assmann, “Prediction of risk of coronary events in middle-aged men in the Prospective Cardiovascular Münster Study (PRO-CAM) using neural networks,” *International journal of epidemiology*, vol. 31, no. 6, pp. 1253–1262, 2002.
- [45] I. Colombet, A. Ruelland, G. Chatellier, F. Gueyffier, P. Degoulet, and M. Jaulent, “Models to predict cardiovascular risk: comparison of CART, multilayer perceptron and logistic regression,” in *Proceedings of the AMIA Symposium*, p. 156, American Medical Informatics Association, 2000.
- [46] L. Churilov, A. Bagirov, D. Schwartz, K. Smith, and M. Dally, “Data Mining with Combined Use of Optimization Techniques and Self-Organizing Maps for Improving Risk Grouping Rules: Application to Prostate Cancer Patients,” *Journal of Management Information Systems*, vol. 21, no. 4, pp. 85–100, 2005.
- [47] D. Kleinbaum, M. Klein, and E. Pryor, *Logistic Regression: A Self-Learning Text*. Statistics for Biology and Health Series, Springer-Verlag, 2002.
- [48] S. Chiu, “Method and software for extracting fuzzy classification rules by subtractive clustering,” in *Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American*, pp. 461–465, 1996.

- [49] A. Bagirov and L. Churilov, “An optimization-based approach to patient grouping for acute healthcare in Australia,” *Computational Science-ICCS 2003*, pp. 694–694, 2003.
- [50] E. El-Darzi, R. Abbi, C. Vasilakis, F. Gorunescu, M. Gorunescu, and P. Millard, “Length of stay-based clustering methods for patient grouping,” *Intelligent Patient Management*, pp. 39–56, 2009.
- [51] N. Allahverdi, S. Torun, and I. Saritas, “Design of a fuzzy expert system for determination of coronary heart disease risk,” in *Proceedings of the 2007 international conference on Computer systems and technologies*, CompSysTech ’07, pp. 36–36, ACM, 2007.
- [52] M. Tsipouras, T. Exarchos, D. Fotiadis, A. Kotsia, K. Vakalis, K. Naka, and L. Michalis, “Automated diagnosis of coronary artery disease based on data mining and fuzzy modeling,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, no. 4, pp. 447–458, 2008.
- [53] T. Graepel, “Statistical physics of clustering algorithms,” *Técnica port*, vol. 171822, 1998.
- [54] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “Clustering algorithms and validity measures,” in *Scientific and Statistical Database Management, 2001. SSDBM 2001. Proceedings. Thirteenth International Conference on*, pp. 3–22, 2001.
- [55] M. Anderberg, “Cluster analysis for applications,” tech. rep., DTIC Document, 1973.
- [56] P. Andritsos *et al.*, “Data clustering techniques,” *Toronto, University of Toronto, Dep. of Computer Science*, vol. 1, no. 1, pp. 3–2, 2002.
- [57] O. Maimon and L. Rokach, *Decomposition methodology for knowledge discovery and data mining: theory and data applications*. Series in machine perception and artificial intelligence, World Scientific Publishing Company Incorporated, 2005.
- [58] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems Series, Elsevier Science & Tech, 2006.
- [59] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, vol. 1, pp. 281–297, Univ. of Calif. Press, 1967.
- [60] L. Kaufman and P. Rousseeuw, “Clustering by means of medoids,” *Statistical data analysis based on the L1-norm and related methods*, vol. 405, 1987.
- [61] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1981.
- [62] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 9th ed., 1990.
- [63] B. Kavsek, N. Lavrac, and A. Ferligoj, “Consensus decision trees: Using consensus hierarchical clustering for data relabelling and reduction,” in *Proceedings of the 12th European Conference on Machine Learning*, EMCL ’01, (London, UK, UK), pp. 251–262, Springer-Verlag, 2001.
- [64] P. Langley, *Elements of machine learning*. Morgan Kaufmann Pub, 1996.

- [65] J. Basak and R. Krishnapuram, “Interpretable hierarchical clustering by constructing an unsupervised decision tree,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 1, pp. 121–132, 2005.
- [66] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Statistics/Probability Series, Belmont, California, U.S.A.: Wadsworth Publishing Company, 1984.
- [67] S. Chiu, “Fuzzy model identification based on cluster estimation,” *Journal of intelligent and Fuzzy systems*, vol. 2, no. 3, pp. 267–278, 1994.
- [68] M. Ester, H. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, vol. 1996, pp. 226–231, AAAI Press, 1996.
- [69] W. Wang, J. Yang, R. Muntz, *et al.*, “STING: A statistical information grid approach to spatial data mining,” in *Proceedings of the International Conference on Very Large Data Bases*, pp. 186–195, IEEE, 1997.
- [70] G. Sheikholeslami, S. Chatterjee, and A. Zhang, “Wavecluster: A multi-resolution clustering approach for very large spatial databases,” pp. 428–439, 1998.
- [71] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.
- [72] G. Fung, “A comprehensive overview of basic clustering algorithms,” 2001.
- [73] T. Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
- [74] M. Carreira-Perpinán, “A review of dimension reduction techniques,” *Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09*, pp. 1–69, 1997.
- [75] H. Kriegel, P. Kröger, and A. Zimek, “Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 1, pp. 1–58, 2009.
- [76] L. Van der Maaten, E. Postma, and H. Van Den Herik, “Dimensionality reduction: A comparative review,” *Journal of Machine Learning Research*, vol. 10, pp. 1–41, 2009.
- [77] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic Pr, 1990.
- [78] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.
- [79] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, “Out-of-sample extensions for LLE, ISOMAP, MDS, EigenMaps, and Spectral Clustering,” *Advances in neural information processing systems*, vol. 16, pp. 177–184, 2004.
- [80] J. Tenenbaum, V. De Silva, and J. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

- [81] B. Schölkopf, A. Smola, and K. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [82] B. Schölkopf, C. Burges, and V. Vapnik, “Extracting support data for a given task,” in *Proceedings, First International Conference on Knowledge Discovery & Data Mining. AAAI Press, Menlo Park, CA*, pp. 252–257, AAAI Press, 1995.
- [83] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- [84] D. Rumelhart, G. Hinton, and R. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [85] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [86] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [87] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” *Advances in neural information processing systems*, vol. 14, pp. 585–591, 2001.
- [88] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [89] Y. Teh and S. Roweis, “Automatic alignment of local representations,” *Advances in neural information processing systems*, vol. 15, pp. 841–848, 2002.
- [90] L. C. Molina, L. Belanche, and À. Nebot, “Feature selection algorithms: A survey and experimental evaluation,” in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pp. 306–313, IEEE, 2002.
- [91] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.
- [92] P. Langley, “Selection of relevant features in machine learning,” in *In Proceedings of the AAAI Fall symposium on relevance*, pp. 140–144, AAAI Press, 1994.
- [93] M. Dash, H. Liu, and H. Motoda, “Consistency based feature selection,” in *Knowledge Discovery and Data Mining. Current Issues and New Applications*, pp. 98–109, Springer, 2000.
- [94] C. Gini, *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche*. No. pt. 1, Tipogr. di P. Cuppini, 1912.
- [95] L. E. Raileanu and K. Stoffel, “Theoretical comparison between the gini index and information gain criteria,” *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 77–93, 2004.
- [96] K. Kira and L. A. Rendell, “A practical approach to feature selection,” in *Proceedings of the ninth international workshop on Machine learning*, pp. 249–256, Morgan Kaufmann Publishers Inc., 1992.
- [97] I. Kononenko, “Estimating attributes: analysis and extensions of relief,” in *Machine Learning: ECML-94*, pp. 171–182, Springer, 1994.

- [98] L. Yu and H. Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution,” in *Machine Learning - International Workshop then Conference*, vol. 20, pp. 856–863, 2003.
- [99] H. He and E. A. Garcia, “Learning from imbalanced data,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [100] F. Provost, “Machine learning from imbalanced data sets 101,” in *Proceedings of the AAAI’2000 Workshop on Imbalanced Data Sets*, 2000.
- [101] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [102] M. Kubat and S. Matwin, “Addressing the Curse of Imbalanced Training Sets: One-Sided Selection,” in *In Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 179–186, Morgan Kaufmann, 1997.
- [103] J. Van Hulse, T. Khoshgoftaar, and A. Napolitano, “Experimental perspectives on learning from imbalanced data,” in *Proceedings of the 24th international conference on Machine learning*, pp. 935–942, ACM, 2007.
- [104] B. Ripley, *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.
- [105] E. Steyerberg, *Clinical prediction models: a practical approach to development, validation, and updating*. Springer, 2008.
- [106] Matlab Toolbox for Dimensionality Reduction (v0.8 - April 2012) [cited 2013 February 21]; [about 6 p.]. Available from: http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html.
- [107] L. Van der Maaten, “An introduction to dimensionality reduction using matlab,” *Report*, vol. 1201, pp. 07–07, 2007.
- [108] Feature Selection at Arizona State University In conjunction with the DMML [cited 2013 March 12]; [about 4 p.]. Available from: <http://featureselection.asu.edu/software.php>.
- [109] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand, and H. Liu, “Advancing feature selection research,” *ASU Feature Selection Repository*, 2010.