UNIVERSIDADE FEDERAL FLUMINENSE

LUIZ ANTONIO DA PONTE JUNIOR

APPLYING DATA MINING TO PREDICT POSTTRAUMATIC STRESS SYMPTOMS USING PHYSIOLOGICAL SIGNALS

NITERÓI

2020

UNIVERSIDADE FEDERAL FLUMINENSE

LUIZ ANTONIO DA PONTE JUNIOR

APPLYING DATA MINING TO PREDICT POSTTRAUMATIC STRESS SYMPTOMS USING PHYSIOLOGICAL SIGNALS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Sistemas de Computação

Orientadora:
Profa. Débora Christina Muchaluat Saade, D.Sc.

Coorientador:
Prof. Alexandre Plastino de Carvalho, D.Sc.

NITERÓI

2020

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

P813a Ponte junior, Luiz Antonio da
Applying Data Mining to Predict Posttraumatic Stress Symptoms Using Physiological Signals / Luiz Antonio da Ponte junior ; Débora Christina Muchaluat Saade, orientadora ; Alexandre Plastino de Carvalho, coorientador. Niterói, 2020.
109 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2020.

DOI: http://dx.doi.org/10.22409/PGC.2020.m.14854613750

1. Mineração de dados (Computação). 2. Transtorno de estresse pós-traumático. 3. Frequência cardíaca. 4. Produção intelectual. I. Saade, Débora Christina Muchaluat, orientadora. II. Carvalho, Alexandre Plastino de, coorientador. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

CDD -

Bibliotecário responsável: Sandra Lopes Coelho - CRB7/3389

Luiz Antonio da Ponte Junior

APPLYING DATA MINING TO PREDICT POSTTRAUMATIC STRESS SYMPTOMS USING PHYSIOLOGICAL SIGNALS

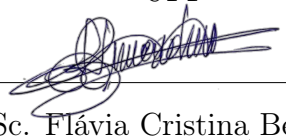
Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Sistemas de Computação

Aprovada em Agosto de 2020.

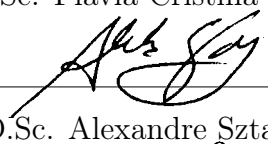
BANCA EXAMINADORA

Profa. D.Sc. Débora Christina Muchaluat Saade - Orientadora, UFF

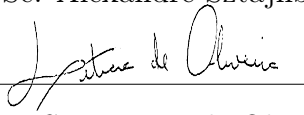
Prof. D.Sc. Alexandre Plastino de Carvalho - Coorientador, UFF



Profa. D.Sc. Flávia Cristina Bernardini, UFF



Prof. D.Sc. Alexandre Sztajnberg, UERJ



Profa. D.Sc. Leticia de Oliveira, UFF

Niterói

2020

Es ist nicht genug zu wissen - man muss auch anwenden. Es ist nicht genug zu wollen - man muss auch tun (Johann Wolfgang von Goethe).

Acknowledgment

Maybe this is my favorite part of a dissertation because here we can be more personal, letting the formalism aside. There are many people to thank. Probably I will write some names and unfortunately I will forget some others. For those I forgot, my sincere apologies.

First of all, I thank God for all I have and for being able to accomplish this work. I also thank my parents and my family for all support, love, care and teachings.

I thank the professors of the Institute of Computing, especially professors Flávio Seixas, Diego Passos (you are the most punctual professor I have ever met) and my advisers professor Débora Saade and professor Alexandre Plastino for their patience, lessons, help, criticisms and suggestions.

I am very grateful for all the help of the Biomedical Institute, especially Orlando Fernandes and the super women of the Laboratory of Neurophysiology of Behavior: professors Leticia Oliveira, Mirtes Pereira, Rita de Cássia Alves and Liana Lima Portugal (you are absolutely the funniest person I met during the master's degree).

I also want to thank my dear teachers Vinícius Leal, Márcia Monteiro, Verônica Andrade, Laerte Theobald, Daniela Porto, Greta Schwankhaus, Kai Cohrs, Helga Küster, Sabine Goertz, Jessica Remane (du bist eine der besten Personen, die ich kenne) and Laura Olbrich. Your classes, lessons and advices brought me new perspectives and made the learning more pleasant and enjoyable.

I must thank my friends for their support and for understanding my absence. I focused and dedicated my time to the research and development of this work but even so our talks were very important to reduce the stress and to maintain and strengthen our friendship. Bruno Furtado (you are like an older brother to me), Mauro Amim, Márcia Amim, Eliane Guilarducci, Christian Guilarducci, Alice Vieira, Mylla Coffaro, Julia Polzer, Екатерина Маханькова, Mona Gehrke, Mandy Lindemann, Christina Sophia Matl, Roksana Długosz and Katrin Rizzello (du bist eine unglaubliche Person), thank you very much for everything. You are the best!

Lastly but not least, I want to thank my colleagues from MídiaCom Lab for the hints and enjoyable time and conversations. And I also want to thank my dear professors Renato Mauro, Glauco Amorim and Myrna Amorim for motivating and encouraging me to apply for the master's degree.

Our relatives, friends, professors and colleagues can teach us a lot of things. We can daily learn with every person. Basically we learn how to do and also how not to do something. That said, I thank every person that I met so far for teaching me always something new. I also sincerely want to thank you for investing a little bit of your time reading this work.

Thanks to CAPES, INCT-MACC, CNPq and FAPERJ for the financial support partially provided during the research.

Resumo

O número de indivíduos portadores de algum transtorno psiquiátrico tem aumentado. Muitos desses transtornos possuem alguns sintomas em comum e identificar tais sutilezas não é uma tarefa rápida e trivial. Um diagnóstico equivocado faz com que indivíduos despendam, muitas vezes, tempo e dinheiro com exames e remédios até que o correto diagnóstico seja efetuado e o tratamento comece a surtir efeito. Visando auxiliar médicos e especialistas a prescreverem diagnósticos mais eficientes e eficazes, diversos estudos e trabalhos propõem a aplicação da computação às áreas da saúde. Através de análises estatísticas, as áreas de Inteligência Artificial, Mineração de Dados, Aprendizagem de Máquina e Reconhecimento de Padrões permitem identificar portadores de transtornos e até mesmo candidatos a desenvolver futuramente tais transtornos. Uma vez identificados, tais indivíduos podem ser direcionados a exames e tratamentos mais específicos. A aplicação da computação auxilia, portanto, no diagnóstico e também na prevenção do desenvolvimento de transtornos. Este trabalho é direcionado ao Transtorno de Estresse Pós-traumático (TEPT), cujos portadores vivenciaram algum evento traumático. Muitos indivíduos vivenciam eventos traumáticos, contudo não desenvolvem o TEPT. O impacto que estes eventos apresentam ao indivíduo influencia o desenvolvimento do transtorno. O TEPT afeta tanto a vida pessoal quanto profissional de seus portadores, pois estes podem evitar determinadas situações com receio de que possam vivenciar novamente o trauma. A recordação de um evento traumático ocasiona alterações neurofisiológicas como taquicardia, bradicardia e sudorese no corpo do indivíduo. Para auxiliar no diagnóstico e no monitoramento do tratamento do TEPT uma das ferramentas utilizadas por médicos consiste do uso de escalas. Uma destas escalas é o Posttraumatic Stress Disorder Checklist (PCL), que avalia o grau de sintomas de TEPT. Este trabalho estuda a aplicação de técnicas de Mineração de Dados, como classificação e regressão, a sinais fisiológicos (frequência cardíaca e condutância da pele) de indivíduos sobreviventes a eventos traumáticos para a predição de valores da escala PCL. O melhor resultado obtido na classificação utilizou o algoritmo SMO (com os valores de seus hiperparâmetros sugeridos pelo plugin Auto-WEKA), aplicando a técnica SMOTE de balanceamento de classe para aumentar a classe minoritária em 100%. Este resultado apresentou acurácia de 85,45% (p-valor = 0,001) e as seguintes medidas para a classe minoritária: *precision* de 0,8 (p-valor = 0,001), *recall* de 0,5714 (p-valor = 0,271) e *F-Measure* de 0,6667 (p-valor = 0,001). Na regressão, o melhor resultado foi obtido com o algoritmo IBk (com k = 4) e apresentou coeficiente de correlação igual a 0,4164 (p-valor = 0,001).

Palavras-chave: Mineração de Dados, Transtorno de Estresse Pós-traumático, Escala PCL, Frequência Cardíaca e Condutância da Pele.

Abstract

The number of individuals diagnosed with some psychiatric disorder has increased. Many of these disorders have common symptoms, and identifying such subtleties is neither a quick nor a trivial task. Misdiagnosis often causes individuals to spend time and money on tests and medicines until receiving the correct diagnosis and the treatment starts to present some progress. Aiming at helping doctors and specialists to prescribe more efficient and effective diagnoses, several scientific studies propose the application of computing to healthcare. Through statistical analysis, the Artificial Intelligence (AI), Data Mining (DM), Machine Learning (ML) and Pattern Recognition (PR) areas enable the identification of disorder patients and even candidates to develop these disorders in the future. Once identified, these individuals can be directed to more specific examinations and treatments. The application of computing therefore helps in the diagnosis and also in the prevention of disorder development. This work is directed to Posttraumatic Stress Disorder (PTSD), in which its patients experienced some traumatic event. Many individuals experience traumatic events, but they do not develop PTSD. The impact that these events have on the individual influences the disorder development. PTSD affects both personal and professional lives of its patients, as they may avoid certain situations for fear that they may experience the trauma again. Recalling a traumatic event causes neurophysiological changes such as tachycardia, bradycardia and sweating in the individual's body. To aid in the diagnosis and monitoring of PTSD treatment, one of the tools used by physicians is the use of scales. One of them is the Posttraumatic Stress Disorder Checklist (PCL), which assesses the degree of PTSD symptoms. This work studies the application of DM techniques, such as classification and regression, to physiological signals (heart rate and skin conductance) of survivors of traumatic events to predict the PCL scale values. The best result obtained in the classification was obtained by SMO algorithm (with its hyperparameters values suggested by the Auto-WEKA plugin), applying the class balancing technique SMOTE to increase the minority class by 100%. This result presented accuracy of 85.45% (p-value = 0.001) and the following measures for the minority class: precision of 0.8 (p-value = 0.001), recall of 0.5714 (p-value of 0.271) and F-Measure of 0.6667 (p-value = 0.001). In the regression, the best result was obtained by IBk (with k = 4), presenting correlation coefficient equal to 0.4164 (p-value = 0.001).

Keywords: Data Mining, Posttraumatic Stress Disorder, PCL Scale, Heart Rate and Skin Conductance.

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence . . . 2
APA	American Psychological Association . . . 6
API	Application Programming Interface . . . 52
ASDS	Acute Stress Disorder Scale . . . 41
BDI	Beck Depression Inventory . . . 41
CADSS	Clinician-Administered Dissociative State Scale . . . 41
CAPS	Clinician-Administered PTSD Scale . . . 6
CFS	Correlation Based-Feature Selection . . . 36
CGI	Clinical Global Impression . . . 41
CV	Cross-Validation . . . 31
DES	Dissociative Experience Scale . . . 41
DM	Data Mining . . . 2
DSM	Diagnostic and Statistical Manual of Mental Disorders . . . 5
DTS	Davidson Trauma Scale . . . 6
ECG	Eletrocardiogram . . . 37
EEG	Electroencephalography . . . 3
epsilon-SVR	epsilon-Support Vector Regression . . . 79
FFT	Fast Fourier Transform . . . 45
fMRI	Functional Magnetic Resonance Imaging . . . 36
FM	F-Measure . . . 27
FN	False Negative . . . 26
FP	False Positive . . . 25
GPR	Gaussian Process Regression . . . 24
HAM-A	Hamilton Anxiety Rating Scale . . . 41

HAM-D Hamilton Depression Rating Scale . . . 41

HRV Heart Rate Variability . . . 3

HR Heart Rate . . . 3

HUAP *Hospital Universitário Antônio Pedro* . . . 43

IES Impact of Events Scale . . . 6

k-NN k-Nearest-Neighbor . . . 22

K6 Kessler Psychological Distress Scale 6 . . . 41

KDD Knowledge Discovery from Data . . . 9

KRR Kernel Ridge Regression . . . 24

LR Linear Regression . . . 24

MAE Mean Absolute Error . . . 27

MISS or M-PTSD Mississippi Scale for Combat-Related PTSD . . . 6

ML Machine Learning . . . 5

MPSS-SR Modified PTSD Symptom Scale . . . 6

MRI Magnetic Resonance Imaging . . . 3

MSE Mean Squared Error . . . 27

NB Naïve Bayes . . . 19

NR Number of Responses . . . 43

NaN Not a Number . . . 55

NeutralTDA Neutral Threat Directed Away . . . 44

NeutralTDT Neutral Threat Directed Towards . . . 44

PANAS Positive and Negative Affect Schedule . . . 41

PANSS Positive and Negative Syndrome Scale . . . 41

PC-PTSD Primary Care PTSD Screen . . . 39

PCA Principal Component Analysis . . . 36

PCL Posttraumatic Stress Disorder Checklist . . . 2

PDEQ Peritraumatic Dissociative Experiences Questionnaire . . . 41

PRoNTo Pattern Recognition for Neuroimaging Toolbox . . . 4

PR Pattern Recognition . . . 2

PSS PTSD Symptom Scale . . . 6

PTSD Posttraumatic Stress Disorder . . . 1

Puk Pearson Universal Kernel . . . 53

RF Random Forest . . . 19

RMSE Root Mean Squared Error . . . 27

RVR Relevance Vector Regression . . . 25

SCID-5 Structured Clinical Interview for DSM-5 . . . 6

SCID Structured Clinical Interview for DSM-IV . . . 41

SCL Symptom Checklist . . . 41

SC Skin Conductance . . . 3

SIP or SI-PTSD Structured Interview for PTSD . . . 6

SMOTE Synthetic Minority Oversampling Technique . . . 15

SMO Sequential Minimal Optimization . . . 36

SPECT Single Photon Emission Computerized Tomography . . . 37

SPRINT Short PTSD Rating Interview . . . 6

STAI State-Trait Anxiety Inventory . . . 41

SVM Support Vector Machine . . . 20

TDA Threat Directed Away . . . 44

TDT Threat Directed Towards . . . 44

THQ Trauma History Questionnaire . . . 41

TLEQ Traumatic Life Events Questionnaire . . . 41

TN True Negative . . . 25

TOP-8 Treatment-Outcome Posttraumatic Stress Disorder Scale . . . 6

TP True Positive . . . 25

UFF *Universidade Federal Fluminense* . . . 3

VA Veterans Affairs . . . 7

Contents

Chapter 1

Introduction

The incidence of psychiatric disorders in our society has been increasing in recent years[1]. During his lifetime, an individual may develop this kind of disorders. The loss of a loved one, the stress experienced at work, being a victim of a robbery or witnessing it, are examples of situations and factors that contribute to develop a psychiatric disorder.

Unlike a mere disease, where a simple consultation or examination is enough for a patient to be diagnosed and his or her treatment be started, psychiatric disorders are more complex to diagnose. Many disorders have symptoms in common, which requires the patient to perform several tests until the proper diagnosis is reached. In contrast, not all individuals are able to perform some exams, as these exams are often expensive or not performed in the proximity where the individual lives.

One of the psychiatric disorders, and focus of study of this work, is Posttraumatic Stress Disorder (PTSD). This disorder affects individuals who have experienced some traumatic event in their lives. The impact of this event, that is, the way the individual perceives and reacts to the event, is what contributes to the disorder development.

As with other disorders, PTSD triggers neurophysiological reactions in the individual's body, such as blood pressure increase, skin conductance variability, tachycardia and bradycardia – acceleration and deceleration of heart rate, respectively. When experiencing events that refer to some past trauma, the individual is affected by such neurophysiological reactions.

As one of its consequences, PTSD interferes with the daily lives of its patients, both in their personal and professional lives, as they end up avoiding some situations for fear that they may relive past trauma.

[1] `https://www.who.int/news-room/fact-sheets/detail/mental-disorders`

Correct diagnosis of PTSD, and any disorder, is critical to the efficiency and effectiveness of treatment. However, the patient may spend a lot of time and money on examinations and consultations until reaching a diagnosis. In addition, there are various treatments, both in traditional medicine with the use of medicines and psychotherapies, as well as in alternative medicine with acupuncture and animal-assisted therapy, targeted to each specific type of disorder.

One of the tools employed by doctors and specialists to help diagnose PTSD is the use of questionnaires that assess the degree of PTSD symptoms according to a predefined scale. One of these scales is the Posttraumatic Stress Disorder Checklist (PCL), which in its fourth version has three variants: PCL-M (military), developed for soldiers, PCL-C (civilian), developed for civilians and PCL-S (specific), assigned to a specific traumatic event [58]. The PCL scale selects possible PTSD patients, providing a provisional diagnosis, able to assist physicians and specialists. It can be used in follow-up treatment as a way of assessing change and development of symptoms.

Like the diagnostic stage, finding the right treatment that produces positive results is a task that requires time and commitment from the patient. Several researches and works have proposed the application of computing to healthcare.

Computer science is present in many areas because of its capacity of cooperating with them. For example, its application contributes to task automation and agility, error and operating cost reduction, intelligent analysis and decision making, information discovery and knowledge extraction, and performance improvement.

Computing areas such as Artificial Intelligence (AI), Data Mining (DM) and Pattern Recognition (PR) have been shown to be highly effective in helping physicians and specialists to diagnose diseases and disorders, increasing the effectiveness of diagnostics, and collaborating to discover indicators (biomarkers) that trigger the diseases and disorders development, thereby helping their prevention.

Some studies, for example, apply AI techniques to identify the existence of a correlation between health data from individuals diagnosed with PTSD, or who have experienced a traumatic event, and scale values that assess PTSD symptoms [29, 30, 48].

Thus, such studies not only help the diagnosis and prevention of diseases and disorders, but also propose new mechanisms capable of identifying their symptoms. These mechanisms are usually simpler and more economical, which makes diagnosis possible for individuals who cannot afford the costs of certain tests.

As mentioned, two of the neurophysiological reactions triggered by PTSD (tachycardia and bradycardia) are related to the Heart Rate Variability (HRV) of the individual. Skin conductance (sweating) is strongly related to heart rate, so altering the latter implies increasing or decreasing the former.

In addition to therapies, there are several exams to diagnose PTSD. However, some of these exams are not accessible to many PTSD patients. Two factors related to accessibility are the cost and the distance between the place where the individual resides and the clinic or laboratory where the exam is performed. Therefore, the use of physiological signals in the prediction of PTSD symptom scales can be a viable alternative to more complex and expensive exams (e.g., Magnetic Resonance Imaging (MRI) and Electroencephalography (EEG)), considering that such signals can be collected non-invasively and with low-cost sensors. In addition, using this kind of signals also allows remote patient monitoring that sometimes can be more convenient to patients or healthcare teams.

An experiment conducted at the Biomedical Institute of *Universidade Federal Fluminense* (UFF) collected the physiological signals of Heart Rate (HR) and Skin Conductance (SC) of civilian volunteers who experienced traumatic events related to violence [2, 3]. The signals were collected during the visualization of emotional and neutral stimuli images and, at the end, the volunteers were asked to complete the PCL-C scale questionnaire. The correlation between those collected signals and the PCL-C score was analyzed to evaluate the possibility of finding future PTSD biomarkers.

1.1 Goals

This work studies the PCL-C scale values prediction through physiological data collected in the Biomedical Institute's experiment with civilian volunteers and aims at evaluating the existence of a correlation between physiological signals and PCL-C scale values. Because these signals are easy to collect, that is, the sensors used are inexpensive, small in size and non-invasive, their use can be an affordable and viable alternative to existing PTSD examinations.

Hence, the possibility of predicting PTSD traits (measured by the score of the PCL-C scale), using physiological signals and applying DM techniques (e.g., classification and regression algorithms), is analyzed in this work. In addition, DM techniques such as supervised discretization, attribute selection and class balancing were applied in order to improve the performance of the prediction models employed. In order to support our

analysis, two algorithms were implemented for the WEKA software [63]: the permutation test algorithm and a brute force algorithm for supervised discretization.

The permutation test algorithm evaluates the statistical significance of the results obtained in classification and regression analyses. To calculate the significance of a result, the algorithm performs n iterations, randomly shuffling the dependent attribute (i.e., the attribute that will be predicted) values among the training set instances before executing classification and regression algorithms. Comparing how many results obtained with shuffled data are better than those obtained with original data (not shuffled data), this concept evaluates if a result was not obtained by chance.

The brute force algorithm for supervised discretization (see Section 4.3) is presented as another way to discretize numeric attributes, solving the problem found with WEKA supervised discretization algorithm [19], where all the attribute values of the dataset used in this work were placed into a single bin.

1.2 Dissertation Structure

The remaining of the text is structured as follows. Chapter 2 discusses the main concepts of health and computing areas used in this work, covering DM techniques and their algorithms, as well as metrics, outcome measures, validation and statistical significance (permutation test and its implementation made) methods.

Chapter 3 deals with a review of the literature, presenting the main techniques and scales employed. Studies are grouped by the type of used data.

Chapter 4 describes the dataset used in this work and explains how data acquisition was performed in the Biomedical Institute experiment. This chapter also discusses the problem found with existing discretization algorithms and presents the implementation of a brute force algorithm for supervised discretization.

Chapter 5 discusses the results obtained with the WEKA [63] classification algorithms employed, along with the applied DM techniques: supervised discretization, attribute selection and class balancing.

Chapter 6 discusses the results obtained with the regression algorithms using WEKA [63] and Pattern Recognition for Neuroimaging Toolbox (PRoNTo) [49] softwares.

Chapter 7 presents the conclusions obtained, along with the contributions of this work. In this chapter, future work is also discussed.

Chapter 2

Background

The main concepts used in this work are presented in this chapter. Sections 2.1 and 2.2 respectively contextualize Posttraumatic Stress Disorder and the PCL scale used to measure the degree of symptoms of this disorder. Section 2.3 covers the DM area, followed by Section 2.4 that discusses Machine Learning (ML). Section 2.5 presents the data preprocessing techniques. Sections 2.6 and 2.7 explain the classification and regression concepts respectively, together with the algorithms used in this work. Section 2.8 presents the metrics and measures employed in this work to evaluate the prediction capacity of the generated models. Sections 2.9 and 2.10 respectively explain how the validation and statistical significance assessment methods are made.

2.1 Posttraumatic Stress Disorder (PTSD)

PTSD is included in the Diagnostic and Statistical Manual of Mental Disorders (DSM) [4], published by the *American Psychiatric Association*. According to the DSM, PTSD is a psychiatric disorder that can develop in people who have experienced or witnessed a traumatic event.

Although PTSD became better known during the First and Second World Wars, it does not only affect military and war veterans. Approximately 3.5% of United States (US) adults have PTSD and it is estimated that one in 11 people will be diagnosed with PTSD throughout their lifetime. Also, women are two times more likely to be affected by this disorder[1].

It is important to notice that not all the people who experience a traumatic event

[1] `https://www.psychiatry.org/patients-families/ptsd/what-is-ptsd`

develop PTSD. The impact and the number of occurrences of the event experienced are linked to the disorder's development. For a person to be diagnosed with PTSD, its symptoms must last for more than one month and persist for a long time (months and even years). In addition, PTSD often occurs in conjunction with other conditions such as depression, anxiety and stress disorders, chemical and alcohol abuse, and memory problems [4].

To diagnose PTSD, physicians and specialists make use not only of interviews with patients in order to obtain reports about the event, but also of specific scales developed to evaluate and measure the symptoms presented by the patient. Table 2.1 presents the main scales applied according to American Psychological Association (APA)[2] and based on the PTSD symptoms described by DSM.

Table 2.1: Main scales related to PTSD symptoms described by DSM, according to APA

Type	Scale Name
Interviews	Clinician-Administered PTSD Scale (CAPS) for DSM-5
Interviews	PTSD Symptom Scale (PSS) Interview for DSM-5
Interviews	Structured Clinical Interview for DSM-5 (SCID-5)
Interviews	Structured Interview for PTSD (SIP or SI-PTSD)
Interviews	Treatment-Outcome Posttraumatic Stress Disorder Scale (TOP-8)
Self-Report Instruments	Davidson Trauma Scale (DTS)
Self-Report Instruments	Impact of Events Scale (IES) – Revised
Self-Report Instruments	Mississippi Scale for Combat-Related PTSD (MISS or M-PTSD)
Self-Report Instruments	Modified PTSD Symptom Scale (MPSS-SR)
Self-Report Instruments	PCL for DSM-5
Self-Report Instruments	PSS Self-Report Version
Self-Report Instruments	Short PTSD Rating Interview (SPRINT)

PTSD patients often have intrusive thoughts and feelings related to the event experienced. By reliving the event through memories or even similar situations, several neurophysiological reactions are triggered, causing variations in heart rate, skin conductance, and blood pressure, for example. Because of this, PTSD patients can avoid situations or people that remind them of their experiences during the traumatic event [4].

According to the fith revision of DSM (DSM-5), PTSD symptoms fall into four categories and may vary in severity. Those four categories are[1]:

1. Intrusive thoughts like repetitive and involuntary memories, nightmares and flashbacks;

[2] https://www.apa.org/ptsd-guideline/assessment/

2. Avoidance of people, places, activities, objects or situations related to the trauma experienced;

3. Negative thoughts and feelings about others and themselves;

4. Reactive symptoms such as anger, irritability and concentration problems.

Just as not all people who have experienced traumatic events develop PTSD, not all PTSD patients require psychiatric treatment, as for some people the symptoms subside or disappear over time. However, many patients still need professional treatments such as psychotherapies and medications to recover from the disorder.

2.2 Posttraumatic Stress Disorder Checklist (PCL)

The PCL scale is an instrument developed by the Veterans Affairs (VA) National Center for PTSD[3] that screens potential PTSD patients and provides a provisional diagnosis to assist physicians and specialists. The PCL scale can also be used to monitor the change and development of symptoms during and after treatment [58].

The PCL scale is based on the PTSD symptoms described by DSM [4]. As DSM is revised, the PCL scale is also often updated. Currently the latest version of the PCL scale is the PCL-5 [59], resultant of the fifth DSM revision. In this work, however, the PCL-4 scale, referring to the fourth revision of DSM (DSM-IV), is used.

As explained in Chapter 1, the PCL-4 scale has three variations: PCL-C for civilians, PCL-M for soldiers, and PCL-S for a specific event. The current scale of PCL (PCL-5) does not have these three variations. Due to the reformulation of the number of items in the PCL-5 scale, the scores are not compatible with the previous version (PCL-4) and cannot be used alternately.

In this work, the PCL-C version 4 is used. This scale is composed of a questionnaire of 17 questions related to problems and complaints that people usually present in response to an experienced trauma. For each question, the participant evaluates a problem that occurred in the month prior to the questionnaire application, defining a score from 1 to 5, where 1 corresponds to the lowest degree of discomfort caused by the problem and 5 corresponds to the highest degree. Hence, the minimum score that can be obtained is 17, for the scenario where an individual assigns the value 1 to all the 17 questions and,

[3] https://www.ptsd.va.gov/professional/assessment/adult-sr/ptsd-checklist.asp

the maximum score that can be obtained is 85, when assigning the value 5 to all the 17 questions.

As explained, the scale values range from 17 to 85, and according to the VA National Center for PTSD, there are studies that suggest certain cut-off points to meet PTSD classification or identification criteria for provisional diagnoses [5]. Lower cut-off points are indicated for criteria that wish to maximize detection of possible PTSD cases. Higher cut-offs are indicated to minimize false positives.

Table 2.2 informs the suggested range of cut-off points for each criterion[4]. In this work, value 36 is used as the cut-off point. That is, values greater than or equal to 36 indicate possible PTSD patients (i.e., belonging to class *high*).

Table 2.2: Used cut-off points to determine the PCL classes

Typical Setting	Suggested PCL Cut-off Point Scores
Department of Defense screening, civilian primary care, general population samples	30-35
Specialized medical clinics, VA primary care	36-44
civilian specialty mental health clinics, VA primary care	45-50

To monitor patient progress in their treatment, evidence suggests that a 5-10 point change in the PCL-4 score result indicates a response to treatment and a 10-20 point change indicates that patient progress is significant [38]. Therefore, VA National Center for PTSD recommends using value 5 for change as the minimum threshold to determine if the individual has responded to treatment and value 10 as the minimum threshold to determine if progress has been significant.

2.3 Data Mining

A huge amount of data is produced and collected daily. Analyzing this data is an important process for extracting and discovering information. The rapid growth of data production and storage exceeds the human capacity to understand and analyze data. As a result, decisions are made, often by intuition and previous experience, regardless of the information that this large amount of data carries. It is therefore necessary to use more powerful and appropriate tools and techniques to extract information from data [24].

[4] `https://www.ptsd.va.gov/professional/assessment/documents/PCL_handoutDSM4.pdf`

A commonly used term for data information acquisition is Knowledge Discovery from Data (KDD). Many people treat DM and KDD as synonyms, while others treat DM as an essential step in KDD. Figure 2.1 illustrates the KDD process. KDD consists of the iterative sequence of the following steps:

1. Data cleaning: removes noise and inconsistencies;
2. Data integration: multiple data sources (e.g., datasets) can be combined;
3. Data selection: data relevant to the analysis are selected from datasets;
4. Data transformation: transforms and consolidates data into suitable forms for the mining process;
5. Data mining: smart methods are applied to extract information and patterns from data;
6. Pattern evaluation: identifies relevant and interesting patterns;
7. Knowledge presentation: the extracted knowledge is presented through visualization and data representation techniques.

DM can be applied to various types of data, such as relational datasets, data warehouses, transactional datasets, temporal and sequential data, media data (e.g., text, audio, video and image) and spatial data, for instance. There are some features of DM that specify the types of patterns to be found in the analysis:

- Characterization and Discrimination: data can be associated with classes or concepts that define them;
- Discovery of frequent patterns, associations and correlations: patterns that occur frequently in a dataset and that may contribute to the increase or reduction of the occurrence of certain data;
- Classification: process to find a model that can describe and distinguish data classes;
- Regression: statistical process often used for numerical data prediction;
- Clustering: data is grouped into clusters based on the principle of similarity;
- Outlier Detection Analyses: find data that does not match the overall behavior of the data.

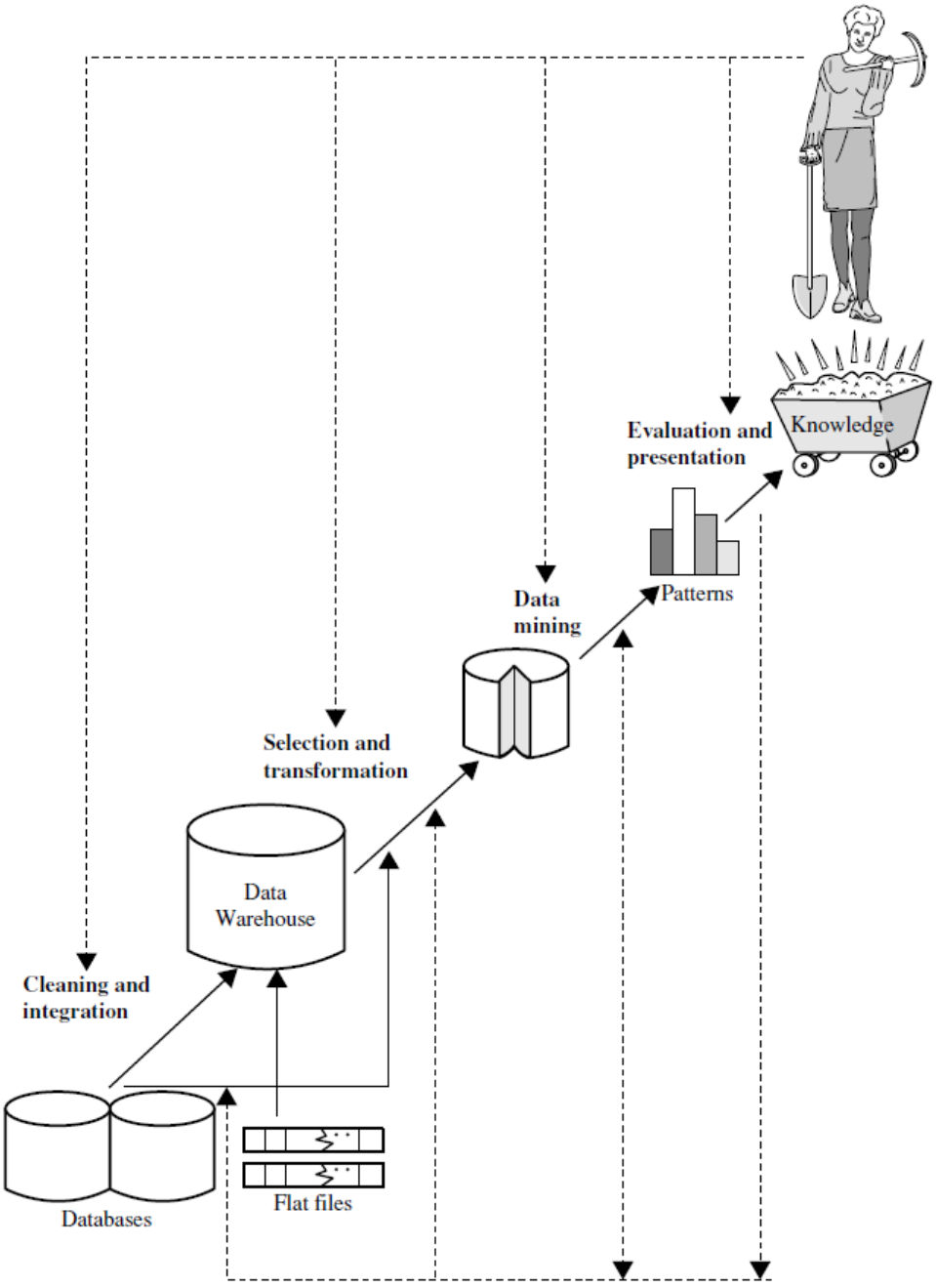


Figure 2.1: The Knowledge Discovery from Data (KDD) process [24]

Data mining analyses can be divided into two categories: descriptive and predictive. Descriptive analyses are concerned with the characterization of data properties. Predictive analyses perform inductions in order to make predictions. In this work, predictive analyses are used. Section 2.6 discusses the *classification* approach and Section 2.7 discusses the *regression* approach.

DM, as an interdisciplinary area, uses techniques from other domains, as illustrated by Figure 2.2. Two of these strongly connected domains with DM are Statistics and ML. A statistical model consists of mathematical functions that describe the behavior of objects and their associated probabilistic distributions. Statistical models can be used to perform data predictions and validate the significance of the results (e.g., permutation test, discussed in Section 2.10). ML will be covered in Section 2.4.

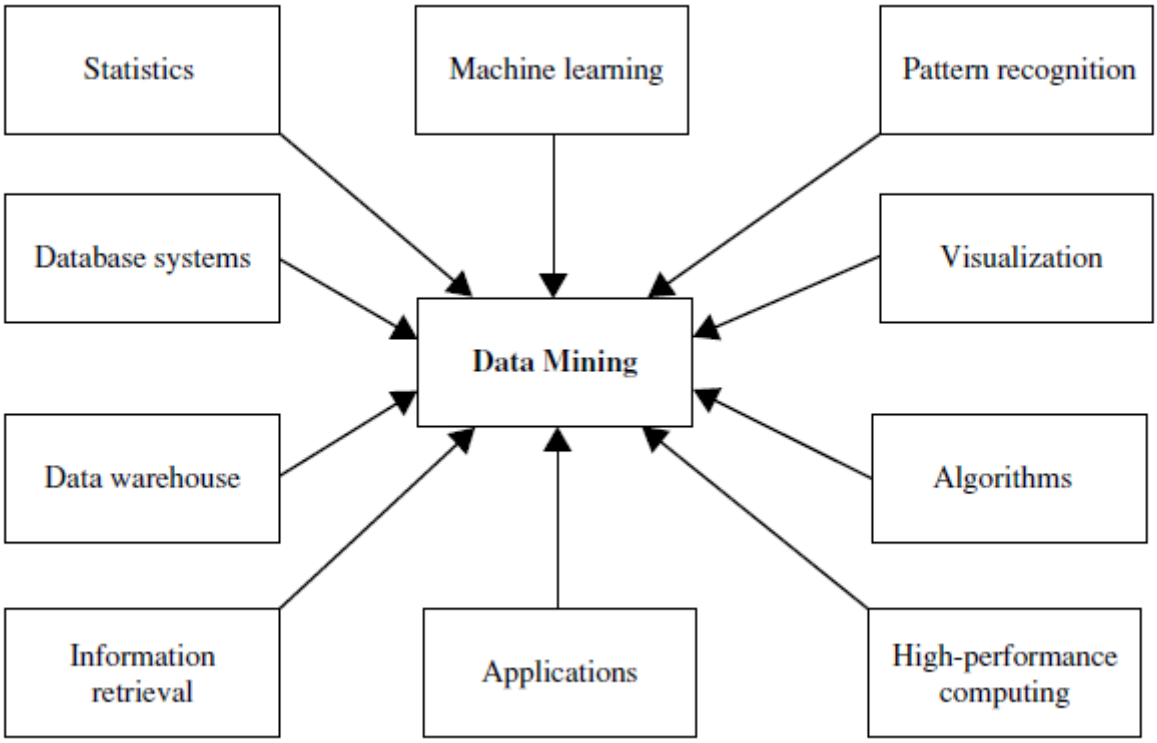


Figure 2.2: Interdisciplinarity of data mining [24]

2.4 Machine Learning

ML uses intelligent algorithms and techniques to generate models that enable the computer to learn complex patterns and make intelligent decisions based on data. ML has the following learning approaches [24]:

- Supervised learning: during learning, data is divided into training and testing partitions. This type of approach uses labeled data in its training partition;
- Unsupervised learning: often treated as a synonym for clustering, this approach uses unlabeled data;
- Semi-supervised learning: this approach uses both data types from previous approaches (labeled and unlabeled). Labeled data is used to learn class models, while unlabeled data is used to refine the class boundaries learned by these models;
- Active learning: is an approach in which the user actively participates in the learning process. The system may require the user to label some data to improve model quality.

2.5 Data Preprocessing

Data preprocessing consists of KDD steps 1 through 4. Often data need to be preprocessed before applying DM. However, there are times when preprocessing is performed as a way to optimize the performance of models created by DM. As explained in Section 2.3, KDD steps are iterative, and it may be necessary to perform them more than once before getting the final result. This section will cover three preprocessing techniques used in this work: attribute selection (or feature selection), discretization and class balancing.

2.5.1 Attribute Selection

A dataset can contain hundreds of attributes, many of these attributes may be irrelevant or redundant for DM analysis. For example, in a prediction analysis of diabetes patients, the attributes regarding the patient's address and telephone number are irrelevant to the diabetes prediction. However, the attributes related to the blood glucose and cholesterol level are relevant to this example of analysis.

While it is possible for an application domain expert to select important attributes and exclude attributes that are irrelevant to analysis, this is not always a trivial task because of the complexity and amount of attributes that a dataset may have.

Excluding relevant attributes or maintaining irrelevant attributes may impair the algorithm employed in the analysis, generating poor quality results. Additionally, a large number of irrelevant or redundant attributes can slow down the process.

Attribute selection aims at finding a minimum subset of attributes that have a probability distribution of classes as close as possible to the original distribution when using all attributes.

For a dataset of n attributes there are 2^n possible subsets. In order not to calculate all possible subsets, there are heuristic methods that exploit a narrow search space whose strategy is to make the optimal local choice, hoping it will lead to the optimal overall solution. Attributes that will be added or removed from the subset are determined through metrics and evaluation measures such as *entropy* and *information gain*, explained in Section 2.8.

One of the attribute selection methods available in WEKA [63] is the ranking search method called *Ranker*, which can use, for example, the *information gain* measure to evaluate the attributes. A ranking is obtained by sorting the attributes according to their evaluation.

In this work, the *Ranker* search method uses the *information gain* metric and is applied in each training partition, generating a ranking for each one of them.

To determine which independent attribute subset will be selected for each training and its corresponding test partition, successive classification analyses with the algorithms described in Chaper 5 are performed. These classification analyses are made as follows: using only the first independent attribute of each ranking; using only the first and second independent attributes of each ranking; and so on until using the first $n-1$ independent attributes of each ranking, where n is the number of dataset independent attributes.

After performing the classification analyses, the attribute subset, which presents the highest accuracy, is selected. If there are two or more subsets presenting the highest accuracy, then the subset, which contains the fewest amount of attributes, will be selected. For example, in a dataset with ten independent attributes, the highest accuracy in average could be obtained using the first seven independent attributes of each ranking. In this example, applying the attribute selection in each training and its corresponding test partition would keep the first seven independent attributes of their corresponding ranking and remove the other independent attributes.

2.5.2 Discretization

Discretization is a form of data transformation, corresponding to the fourth KDD step, discussed in Section 2.3. One of the discretization goals is to make the mining process

more efficient. In addition, the patterns found are more intelligible.

The applied discretization technique is categorized according to the use of attribute class information. If the discretization process uses class information, then it is a supervised discretization. Otherwise it is an unsupervised discretization.

In supervised discretization, the class distribution information is used to calculate and determine the split-points for delimiting attribute ranges. The idea is to determine split-points so that a resulting range contains as many instances of the same class as possible. One of the commonly used measures in supervised discretization is the *entropy*. A supervised discretization algorithm selects a value as a split-point for an attribute that generates a class distribution with the minimum *entropy*. Then the algorithm recursively repeats this process of minimum *entropy* evaluation to the resulting intervals, until matching a stopping criterion.

Discretization, besides being a form of data transformation, is also a form of data reduction, because the raw values of a numeric attribute are replaced by a smaller set of intervals. The original data is transformed into a smaller number of ranges, simplifying the dataset and making the process more efficient and its found patterns easier to understand [24].

One of the forms of discretizing an attribute is the binning discretization. In this form, the values of an attribute are sorted and distributed into bins. The distribution into bins is done by consulting the neighboring (adjacent) values of each value. There are some binning techniques, like [24]:

- Equal-frequency binning: each bin has the same amount of values;
- Equal-width binning: all bins have the same size, i.e., the difference between the minimum and maximum values of each bin is constant.

These binning discretization techniques perform an unsupervised discretization because they do not use attribute class information. The binning discretization allows to specify the desired number of bins. Figure 2.3 illustrates the binning techniques explained.

Sorted data: 4, 8, 15, 20, 20, 23, 26, 28, 33

Equal-frequency binning		all the bins have the same size
Bin 1:	4, 8, 15	
Bin 2	20, 20, 23	
Bin 3:	26, 28, 33	
Equal-width binning		length = [(max - min) + 1] / number of bins = [(33 - 4) + 1] / 3 = 10
Bin 1:	4, 8	from 4 to 13
Bin 2:	15, 20, 20, 23	from 14 to 23
Bin 3:	26, 28, 33	from 24 to 33

Figure 2.3: Discretizing by binning

2.5.3 Class Balancing

Data imbalance is observed when the number of instances per class is not well distributed. Two techniques used to address the data imbalance problem are oversampling and undersampling. The Synthetic Minority Oversampling Technique (SMOTE) algorithm addresses the problem of data imbalance by using the oversampling technique.

In the oversampling technique applied by SMOTE, synthetic (artificial) instances belonging to the minority class are created and added to the dataset. The attributes of these instances have values close to the values of the actual instances belonging to the minority class [10].

In the undersampling technique, some instances of the majority class are removed from the dataset in order to reduce the difference in the number of instances of each existing class.

Since the total of instances in the dataset is small, the oversampling technique applied by the SMOTE algorithm was used in this work to increase the number of minority class instances.

2.6 Classification

Classification is a type of data analysis that creates models capable of describing data classes, consisting of categorical (nominal) attributes. These models are called classifiers and can predict the instance class (also called dependent attribute) of a dataset. Classification models therefore predict categorical values (classes). Problems involving numerical value prediction are addressed by regression analysis, which will be explained in Section 2.7.

The classification analysis consists of two steps: the learning step and the prediction step. During the learning step, a classifier is constructed by analyzing a subset of the original dataset, called a training set or training partition. Each instance (tuple) of the training partition has, in addition to its attributes, a predefined class. The classifier is responsible for assigning the class value to the instance. This learning step can also be viewed as a discovery of a rule set or a $y = f(x)$ function that maps an instance x to a class y, resulting in its prediction. By knowing the class information (values) during this step, the classification analysis is also part of supervised learning.

During the prediction step, the classifier constructed and trained in the learning step is used to perform class value predictions of new instances. In this step, the classifier will make predictions on a test set or test partition, made up of the original dataset instances that were not in the training partition. Thus, it is possible to correctly evaluate the prediction performance of the classifier, since it has not previously used these instances in its learning step.

Figure 2.4 illustrates the process of classification analysis, covering the two steps. In the first part (a) of this figure, a classification algorithm analyzes the training data and the classifier (learned model) is represented in the form of classification rules. In the second part (b), the test data are used to evaluate the predictive power of the classifier. If this predictive capacity is considered admissible, the learned rules can be applied to classify new data (i.e., future data instances for which the class is unknown).

The algorithms employed in the classification analyses presented in Chapter 5 will be covered in this section. The purpose of this section is to provide an overview of how each algorithm works.

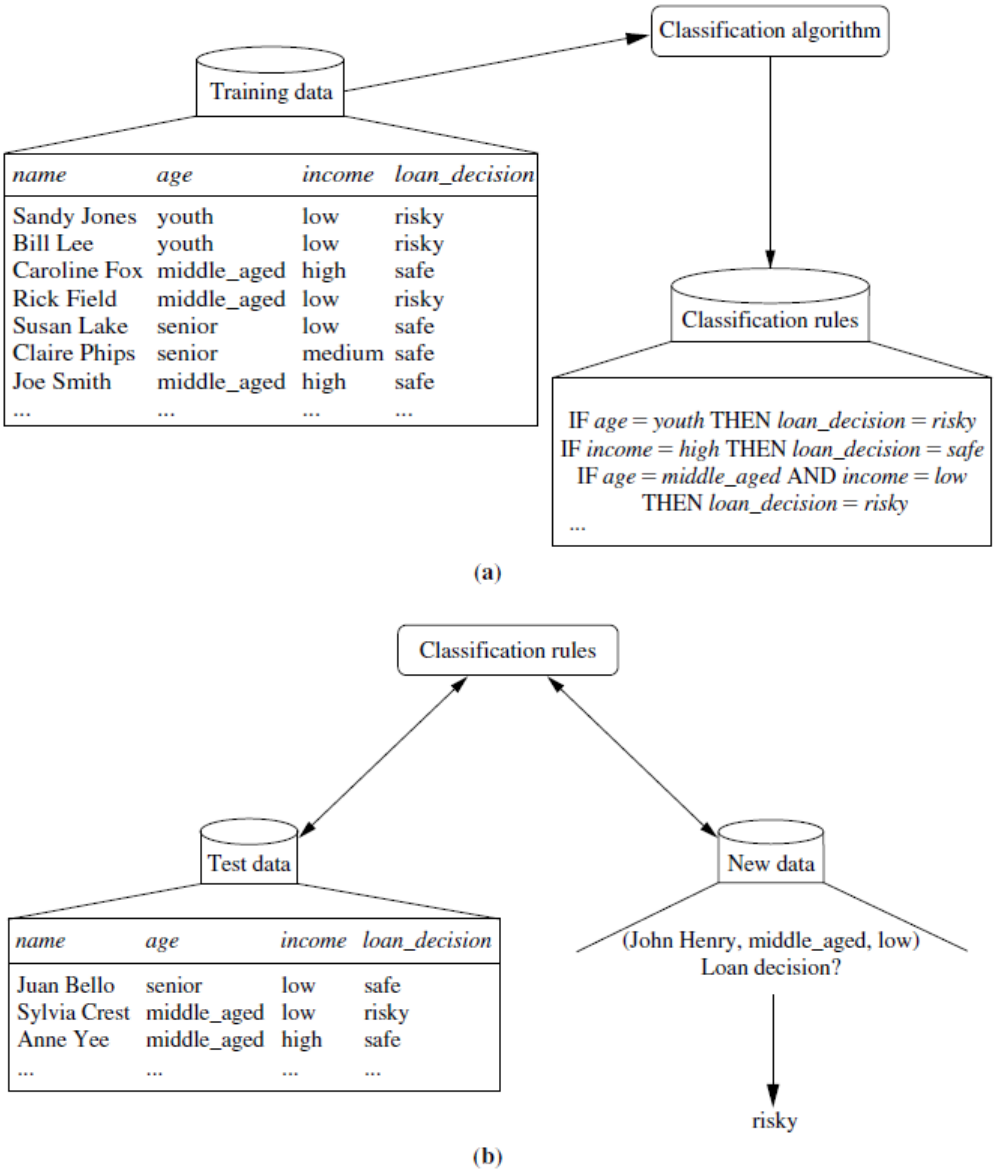


Figure 2.4: Classification's process [24]

2.6.1 Decision Tree

The Decision Tree algorithm generates a model based on the known instances of the training partition. This model has a flowchart with a tree structure that is intuitive and easy to understand. In this structure, each inner node represents a test on an attribute, each branch represents a test result, and each final node (leaf node) represents a class. To predict the class value of an instance I of the test partition, a path is traced from the root to one of the tree leaves, which contains the predicted class of instance I. Figure 2.5 illustrates a decision tree [24].

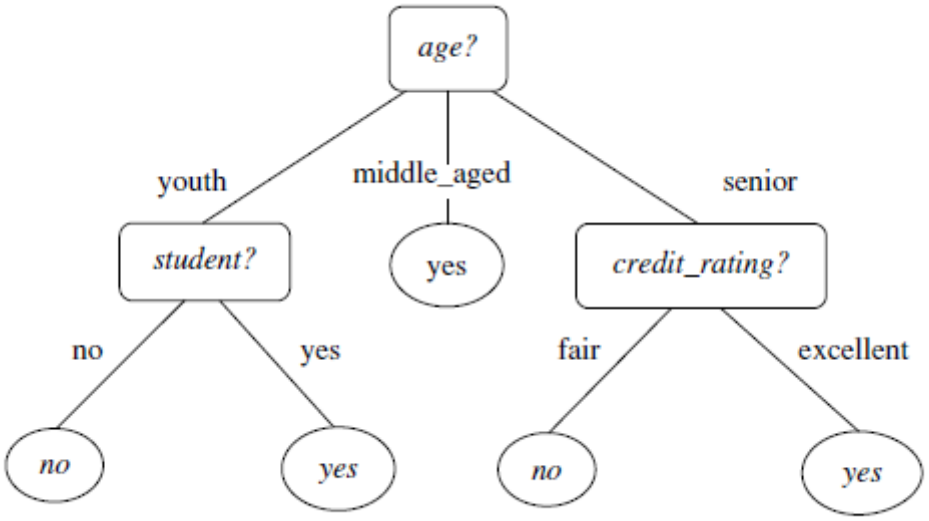


Figure 2.5: Example of a decision tree [24]

When a decision tree is built, many branches may reflect noise and outlier anomalies, making the tree unintelligible. Tree pruning methods use statistical measures to remove less reliable branches. These methods have a tendency to generate smaller and less complex trees, which are better for correctly classifying test partition instances. Figure 2.6 illustrates a decision tree before and after applying a *pruning* method.

In WEKA software, one of the decision tree implementations is the *J48* algorithm. The *J48* algorithm has the *confidenceFactor* (CF) and the *unpruned* parameters, related to the decision tree pruning. The variation of their values will be explained and discussed in Chapter 5.

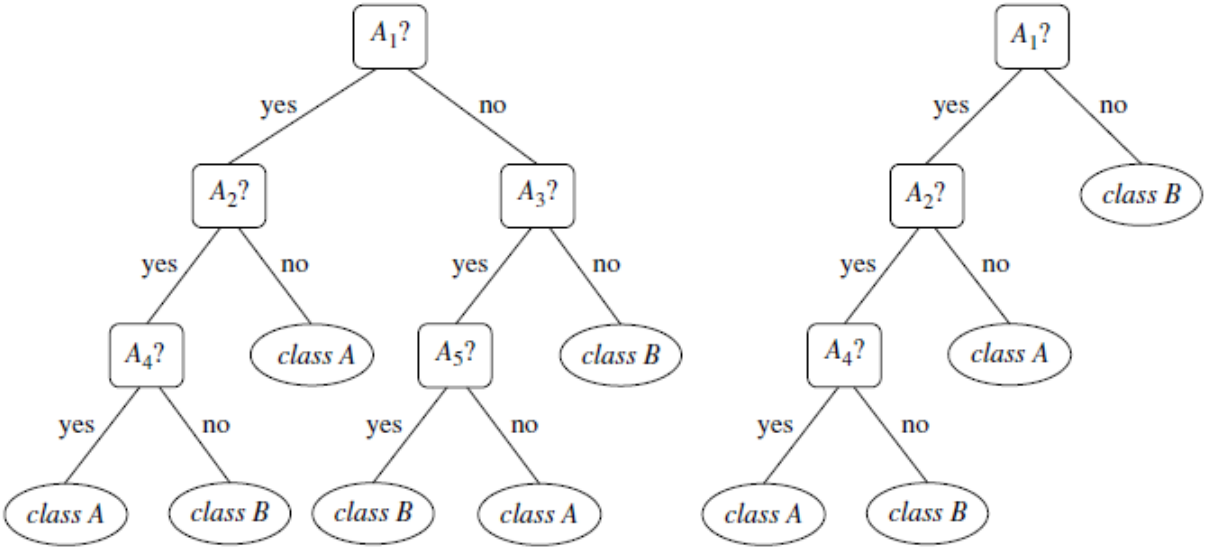


Figure 2.6: Example of pruning a decision tree [24]

2.6.2 Naïve Bayes

Bayesian classification algorithms such as Naïve Bayes (NB) are statistical classifiers that are based on the Thomas Bayes Theorem. These algorithms predict the probability of an instance I to belong to a class C. For this, the NB algorithm makes use of conditional class independence, in which it assumes that the effect of an attribute on a C class is independent of the values of the other attributes. The algorithm then calculates the probability that an instance I belongs to each of the dataset classes. It then assigns the instance I to the class with the highest probability [28].

2.6.3 Random Forest

The Random Forest (RF) algorithm belongs to the ensemble paradigm. This method consists of a composite model, constructed by combining classifiers. The prediction of an instance class is performed through the vote of each classifier. At the end of the vote, the most voted class is then assigned to the instance. The RF algorithm, as its name suggests, consists of a model composed of several decision trees. In this model, each decision tree is generated using a random selection of attributes on each node and a random selection of the training instances. During classification, each decision tree votes based on its set of rules from its tree structure and the most voted class is assigned to the instance by the model [9]. Figure 2.7 illustrates how the RF algorithm predicts the class of an instance.

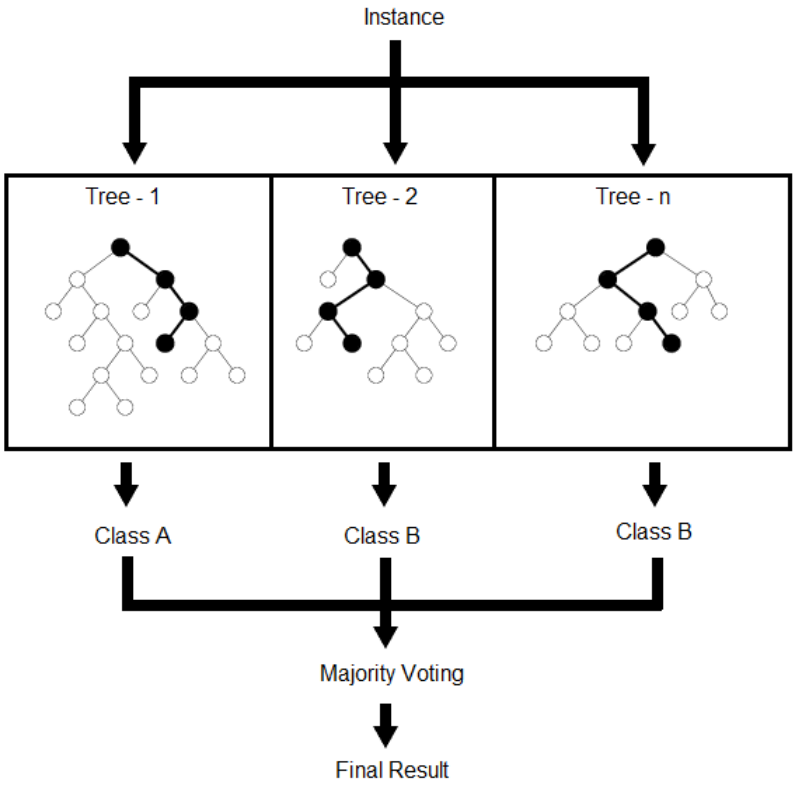


Figure 2.7: Class prediction performed by the Random Forest algorithm

In WEKA, the RF implementation has the *numIterations* (NI) parameter, related to the amount of decision tree models built that will vote. The variation of its values will be explained and discussed in Chapter 5.

2.6.4 Support Vector Machine

The Support Vector Machine (SVM) looks for the optimal linear hyperplane that separates instances belonging to one class from another, as illustrated by Figure 2.8. SVM finds the hyperplane using support vectors (i.e., training instances located on the edges of each cluster), calculating the margin width ("Large margin" in Figure 2.8) according to the distance between the support vectors. The midpoint between them is commonly used as the threshold.

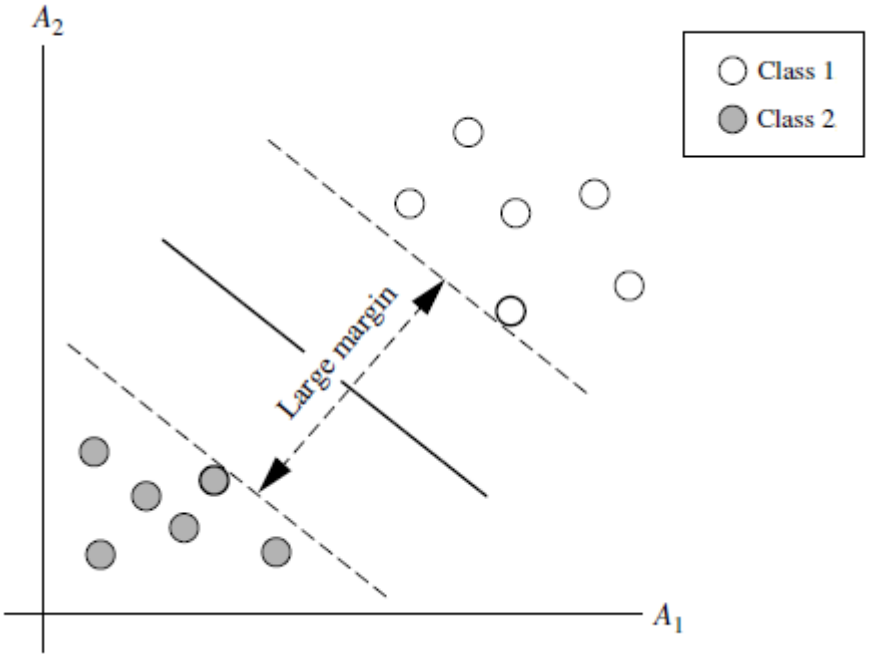


Figure 2.8: Hyperplane calculated by the SVM algorithm [24]

Sometimes it is not possible to separate classes using lines (in case of two-dimensional data) or planes (in case of three-dimensional data). By applying the SVM algorithm in situations where classes are not linearly separable, SVM applies a kernel function, which transforms the original data from the training partition into a higher dimensional space where a hyperplane can separate the classes. Figure 2.9 illustrates the transformation performed by a kernel function.

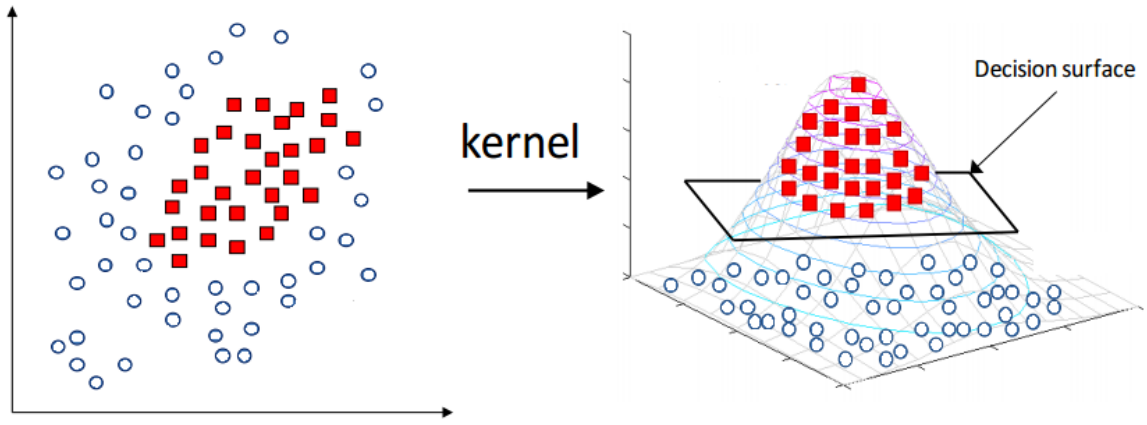


Figure 2.9: Transformation made by a kernel function[5]

The transformation performed by the kernel function is known as the kernel trick

[5] https://www.hackerearth.com/blog/developers/simple-tutorial-svm-parameter-tuning-python-r

and it also works as a tool to reduce hyperplane calculation complexity. Although the execution of this algorithm may be slow due to its complexity, it is usually highly accurate. In addition, SVM can be used for both numeric and categorical prediction [26, 31, 53, 54].

SVM algorithm has the *kernel* parameter, responsible to determine the kernel function that will be applied. Another parameter is the C parameter, known as the regularization parameter because it controls the trade-off between penalizing misclassifications and the margin width. Like the algorithms have their parameters, the kernels have also their own parameters, which were varied only by suggestion of Auto-WEKA plugin [32, 55]. The variation of their values will be explained and discussed in Chapter 5.

Furthermore, in numeric prediction, i.e., regression analysis, the SVM algorithm uses a ϵ-insensitive hinge loss function, which is a loss function that uses the ϵ value to define a margin of tolerance to penalize errors. The higher the value of ϵ, more errors will be admited, i.e., less penalty is given to errors [17, 54].

2.6.5 k-Nearest-Neighbor

The algorithms discussed so far are examples of the eager learning approach. In this approach, the algorithms build a classification model before receiving new instances to classify (test partition). The k-Nearest-Neighbor (k-NN) algorithm, on the other hand, belongs to the lazy learning approach. In this other approach, the algorithm stores the training partition instances and upon receiving a test partition instance, it predicts its class. The k-NN, for example, performs the prediction based on the similarity of previously stored training instances. Algorithms that use the lazy learning approach are more costly when making a prediction because they do not build models, and all calculations need to be performed each time a new instance needs to be predicted [24].

The k-NN algorithm stores training instances, described by n attributes. Thus, each instance represents a point in a pattern space of n dimensions. When receiving a test instance I, k-NN searches for the k training instances in the pattern space that are closest to the test instance. In classification, the prediction is made according to the most common class among the k-nearest neighbors (k training instances). In regression, the predicted value of the dependent attribute is obtained calculating the average of the k-nearest neighbors dependent attribute values. Thus, the k-NN algorithm can be used in both classification analysis and regression analysis [1].

The similarity or closeness is calculated through a distance metric, such as *Euclidean*

distance. The analyses performed with the k-NN algorithm (see Chapter 5) used the *Euclidean distance* and its formula is explained in Equation 2.1. In the formula, X_1 and X_2 represent two instances with their n attributes. For each numeric attribute, the difference between the corresponding values of that attribute in those two instances is calculated, then squared and accumulated. Lastly, the square root of the accumulated total is calculated. When the attribute is nominal, the difference between the corresponding values of that attribute in those two instances will be zero if the values are equal and 1 otherwise.

$$distance(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2} \tag{2.1}$$

Usually the values of each attribute are normalized before applying Equation 2.1. This normalization prevents large ranges attributes outweighing smaller ranges attributes. It transforms a numeric value v of an attribute A into a value v' in the range [0,1], as shown in Equation 2.2. min_A and max_A are respectively the minimum and maximum values of attribute A.

$$v' = \frac{v - min_A}{max_A - min_A} \tag{2.2}$$

In WEKA, the implementation of k-NN (called *IBk*) has the k parameter, which refers to the number of training instances, searched in a pattern space, that are closest to the test instance. The variation of its values will be explained and discussed in Chapter 5.

2.7 Regression

Regression is a type of statistical methodology also used in DM to perform numerical value predictions. Similar to classification analysis, regression analysis is also composed of two steps: learning step and prediction step, as described in Section 2.6. What differs between the two types of analysis is that in regression analysis, the dependent attribute to be predicted has a numeric value rather than a nominal or categorical value (class). And the constructed model consists, in general, of a mathematical function generated based on the known values of the training partition. In the prediction step, the model uses the generated mathematical function to perform the prediction of the new instance dependent attribute value.

The algorithms employed in the regression analysis of Chapter 6 will be discussed in

this section. The purpose of this section is to provide an overview of how each algorithm works. Importantly, the k-NN and SVM algorithms explained in Section 2.6 are also used in regression analyses because they are able to predict both nominal and numeric attributes.

2.7.1 Linear Regression

The Linear Regression (LR) algorithm models the variable to be predicted (dependent attribute) y as a linear function of the dataset x independent attributes using the following Equation 2.3:

$$y = wx + b \tag{2.3}$$

In this equation, y and x are respectively the dataset dependent and independent attributes, w is the regression coefficient and b is the stochastic component (often referred to noise), which represents possible errors and deviations [24].

2.7.2 Gaussian Process Regression

The Gaussian Process Regression (GPR) algorithm consists of a Bayesian approach that uses a collection of random variables, which have a Gaussian distribution, to infer the probability distribution of a regression function. Rather than calculating the probability distribution of the parameter values in a specific function, GPR infers the probability distribution over all applicable functions that fit the data. GPR identifies the relationships between data through Bayesian inference. This algorithm uses the lazy learning approach and a kernel function to identify patterns and perform prediction [61, 50, 62].

2.7.3 Kernel Ridge Regression

The Kernel Ridge Regression (KRR) algorithm learns a linear function inferred from the kernel used. Nonlinear kernels produce nonlinear functions. The way the prediction model is built by KRR resembles the SVM algorithm process. The difference is that KRR uses the error squared loss function [60].

2.7.4 Relevance Vector Regression

The Relevance Vector Regression (RVR) algorithm is proposed in [56] as an evolution of the SVM algorithm. RVR builds a model of the same functionality as the model generated by the SVM algorithm. However, it uses fewer support vectors and fewer kernel functions for faster execution.

2.8 Metrics and Measures

After creating a model, an evaluation of its predictive ability is desired. Moreover, more than one model is often created by choosing different algorithms and parameters, and it is desired to compare the performance of these models in order to select the best or best set of models.

The dataset used in this work will be described in detail in Chapter 4. However to explain the metrics and how to calculate them, the dataset class attribute can be used as example. The class attribute of the dataset is called PCL and contains two classes: *high* and *low*. Out of the 55 instances present, 14 belong to the *high* class and 41 belong to the *low* class. Here are six important terms that are used to calculate various metrics:

1. Positive tuples (P): refers to tuples (instances) of the class of interest. In this example, using the *high* class as the interest class, P corresponds to the 14 instances of the *high* class;

2. Negative tuples (N): refers to tuples (instances) that do not belong to the class of interest. In this example, using the *high* class as the interest class, N corresponds to the 41 instances of the *low* class;

3. True Positive (TP): refers to positive instances that have been correctly classified as belonging to the interest class. In the example, they correspond to instances of the *high* class that were correctly classified as belonging to the *high* class;

4. True Negative (TN): refers to negative instances that were correctly classified as not belonging to the interest class. In the example, they correspond to instances of the *low* class that were correctly classified as belonging to the *low* class;

5. False Positive (FP): refers to negative instances that have been classified as positive. In the example, they correspond to instances of the *low* class that were wrongly classified as belonging to the *high* class;

6. False Negative (FN): refers to positive instances that have been classified as negative. In the example, they correspond to instances of the *high* class that were wrongly classified as belonging to the *low* class.

These terms are presented in the confusion matrix, which is used as an easily understood tool to evaluate the prediction of different classes performed by the classifier. The main diagonal of the matrix (TP and TN) indicates that the classifier is correctly classifying the instances, while the secondary diagonal (FN and FP) indicates the opposite. Ideally, for good performance, the secondary diagonal values should be close to zero. Table 2.3 illustrates the confusion matrix with the respective terms covered.

Table 2.3: Confusion matrix

		Predicted		
		High	Low	Total
Actual	High	TP	FN	P
	Low	FP	TN	N

Once the meaning of each term was well understood, the following metrics can be easily calculated:

- Accuracy: corresponds to the percentage of test partition instances that were correctly classified. Accuracy is calculated using Equation 2.4.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{P + N} \tag{2.4}$$

- Precision: corresponds to the percentage of instances classified as belonging to the interest class and actually belonging to the interest class. In other words: from the instances classified as belonging to the interest class, how many of them belong to the interest class. Precision is calculated by Equation 2.5.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2.5}$$

- Recall: corresponds to the percentage of interest class instances that are classified as belonging to the interest class. In other words: from all interest class instances, how many of them were classified as belonging to the interest class. Recall is calculated by Equation 2.6.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = \frac{\mathrm{TP}}{P} \tag{2.6}$$

- F-Measure (FM): A classifier may have a high precision, indicating that all instances it classifies as belonging to the *high* class, for example, actually belong to this class. However, precision does not indicate how many instances belonging to the *high* class were incorrectly classified by the classifier as belonging to the *low* class. On the other hand, a classifier can present a high recall, indicating that all instances of the *high* class, for example, were classified as belonging to the *high* class. Similarly, the recall does not indicate how many instances of the *low* class were incorrectly classified as belonging to the *high* class. There is therefore an inversely proportional relationship between these two metrics, where it is possible to increase the value of one, with the cost of reducing the value of the other. To evaluate a classifier, an alternative is the use of F-Measure, which is a metric that combines precision and recall using a harmonic average calculated using Equation 2.7.

$$F\text{-}Measure = \frac{2 * precision * recall}{precision + recall} \tag{2.7}$$

The metrics explained so far correspond to classification analyses metrics. Regression analyses also have their own metrics, many of them derived from statistics. One of the widely used metrics is the *correlation coefficient*, also known as Pearson's product moment coefficient and often indicated by letters R and ρ. The correlation coefficient calculates the correlation between two numeric attributes A and B using Equation 2.8.

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} \tag{2.8}$$

In the formula, n is the number of instances, a_i and b_i are the respective values of the attributes A and B in the instance i, $\bar{A}$ and $\bar{B}$ correspond to the mean values of A and B, σ_A and σ_B are the respective standard deviations of A and B.

In statistics and regression problems involving numeric values, the predicted values do not always match the original data actual values. Knowing the difference between predicted and actual values is useful to refine future predictions making them more accurate. Three measures used in the regression analyses (see Chapter 6) are Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

MAE can be calculated using Equation 2.9. In the formula, X_{pi} corresponds to the predicted value for the X attribute of the instance i, X_{ai} corresponds to the actual value of the X attribute of the instance i and n corresponds to the number of instances.

$$MAE = \frac{\sum_{i=1}^{n}(|X_{pi} - X_{ai}|)}{n} \tag{2.9}$$

MSE can be calculated using Equation 2.10. In the formula, X_{ai} corresponds to the actual value of the X attribute of instance i, X_{pi} corresponds to the predicted value for the X attribute of instance i and n corresponds to the number of instances.

$$MSE = \frac{\sum_{i=1}^{n}(X_{ai} - X_{pi})^2}{n} \tag{2.10}$$

RMSE can be calculated using Equation 2.11. In the formula, RMSE is obtained by the square root of the MSE calculated with Equation 2.10.

$$RMSE = \sqrt{MSE} \tag{2.11}$$

Some classification algorithms use measurements in their operations. Such measures may serve as selection, cutting or stopping criteria, for example. Some of the measures used in this work and that are employed in the implementation of the brute force supervised discretization algorithm (see Section 4.3) are: *entropy*, *information gain*, *gain ratio* and *gini index*.

Information gain is one of the metrics used in attribute selection. By this measure, it is possible to select the best attribute A that minimizes the information needed to classify the instances of a resulting partition D, leading to the reduction of partition "impurities". In other words, reducing the number of instances from different classes within a partition to achieve a partitioning that contains only instances from a single class.

Entropy measures the degree of "impurity" of the resulting partition from the selection of an attribute. Let C be the class attribute containing m distinct values, C_i be one of the m class values and C_i, D be the set of instances of C_i class in D. Let also $|C_i, D|$ and $|D|$ be, respectively, the number of instances in C_i, D and in D. The expected information needed to classify an instance in a partition D (*Info(D)*), also known as the *entropy* of D, is calculated by Equation 2.12.

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{2.12}$$

In the *entropy* calculation formula, p_i is the nonzero probability that an arbitrary instance in partition D belongs to class C_i, and is estimated by $|C_i, D|/|D|$. $Info(D)$

then represents the average amount of information needed to identify the class of an instance in partition D.

It is also expected that a categorical attribute A has v distinct values, $\{a_1, a_2, ..., a_v\}$. When using A to split D into v partitions $\{D_1, D_2, ..., D_v\}$, where D_j contains the instances of D that have the value a_j of A, it is desirable to get pure partitions, leading to an exact classification of their instances. However, it is likely that the partitions will be impure.

To calculate how much more information is still needed to obtain an exact classification, Equation 2.13 is applied. In this equation, $\frac{|D_j|}{|D|}$ corresponds to the weight of the j^{th} partition and $Info_A(D)$ is the expected information needed to classify an instance from D, based on the partitioning by A.

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \tag{2.13}$$

Information gain is defined by the difference between the original information needed ($Info(D)$) and the new amount of required information obtained after the partitioning used by the selected attribute A ($Info_A(D)$). That is, the *information gain* indicates how much information would be gained by partitioning through attribute A. The attribute with the highest *information gain* is chosen as the splitting attribute. Its calculation is performed using Equation 2.14.

$$Gain(A) = Info(D) - Info_A(D) \tag{2.14}$$

The *information gain* measure is biased. It prefers to select attributes having a large number of distinct values. For example, in a dataset with an attribute *ID* that stores a unique identifier, a split on *ID* would result in a large number of partitions (as many as the number of values in *ID*). The information required to classify instances based on this partitioning is zero, because each partition is pure. Therefore the information gained by splitting on *ID* attribute is maximal, although this partitioning is useless for classification.

Thus, *gain ratio* measure is an extension to *information gain* aiming at overcoming its bias. *Gain ratio* applies a normalization to *information gain* using a split information value, defined by Equation 2.15.

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2 \left(\frac{|D_j|}{|D|} \right) \tag{2.15}$$

The $SplitInfo_A(D)$ value represents the potential information generated by splitting D into v partitions, corresponding to the v values of attribute A. The *gain ratio* is then calculated as shown in Equation 2.16. The attribute with the maximum *gain ratio* is selected as the splitting attribute.

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)} \tag{2.16}$$

The *gini index*, also used in discretization, measures the impurity of D as shown in Equation 2.17.

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2 \tag{2.17}$$

In Equation 2.17, p_i is the nonzero probability that an instance in D belongs to class Ci and is calculated by $|Ci, D|/|D|$. The sum is calculated over the m class values of C.

The *gini index* considers a binary split for each attribute. Let A be an attribute with v distinct values, $\{a_1, a_2, ..., a_v\}$. To determine the best binary split on A, all the possible subsets formed with the known values of A are examined. Each subset S_A is a binary test for attribute A of the form $A \in S_A$. Given an instance, this test is satisfied if the value of A for the instance is among the values listed in S_A. If A has v possible values, then there are $2^v - 2$ possible subsets, because the empty subset and the subset containing all values are excluded, since they do not represent a split.

In a binary split the weighted sum of each resulting partition impurity is calculated. If a binary split on A partitions D into D_1 and D_2, the *gini index* of D is calculated by Equation 2.18.

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \tag{2.18}$$

For each attribute, each of the possible binary splits is considered. The subset that gives the minimum *gini index* for an attribute is selected as its splitting subset. The reduction in impurity incurred by a binary split on attribute A is calculated by Equation 2.19.

$$\Delta Gini(A) = Gini(D) - Gini_A(D) \tag{2.19}$$

The attribute that maximizes the reduction in impurity or, equivalently, has the minimum *gini index* is selected as the splitting attribute.

2.9 Validation Method

As explained in Sections 2.6 and 2.7, the classification and regression analyses are each composed of two steps: the learning step and the prediction step. During the learning step, training and test partitions are generated. The training partition will be used in the first step (learning step), while the test partition will be used in the second step (prediction step). This partitioning of the dataset is fundamental to evaluate the prediction estimate of the model. The way partitions are generated may differ according to the employed method. This section deals with the k-Fold Cross-Validation (CV) method employed in the analyses of this work.

In the k-Fold Cross-Validation method, the initial dataset is randomly partitioned into k mutually exclusive partitions or folds $\{F_1, F_2, F_3, ..., F_k\}$, where each fold has about the same size. Training and testing, corresponding to the learning and prediction steps respectively, are performed k times. In the *i-th* execution, fold F_i is intended for testing and the other $k-1$ folds are used for model training.

Figure 2.10 illustrates the k-Fold Cross-Validation partitioning method. In this method, each instance is used the same number of times $(k-1)$ for training and once for testing. In classification analysis, the estimated accuracy is calculated by the total instances correctly classified in k runs, divided by the total instances of the initial dataset.

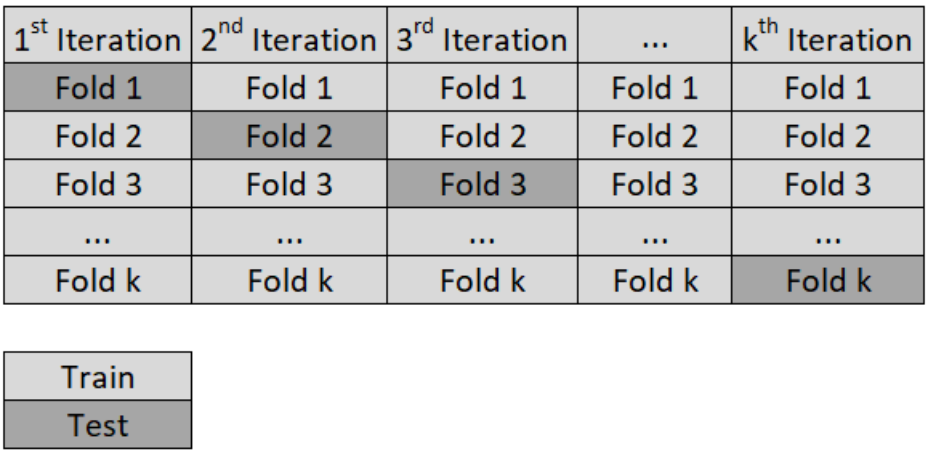


Figure 2.10: k-Fold Cross-Validation method

In this work, we used the Stratified Cross-Validation variation of the k-Fold Cross-Validation method. In this variation, folds are partitioned so that the class distribution of the instances in each fold is approximately the same and similiar to the class distribution of the original dataset. This technique aims at ensuring that each fold contains approximately the same number of instances of each class.

2.10 Statistical Significance Test

After constructing a model and obtaining its estimates of accuracy and other metrics, it may be questioned whether such results are statistically significant, i.e., if they did not occur by chance. One of the statistical tests of significance is the permutation test, also called randomization test. In this test, n datasets are generated, where the values of the attribute to be predicted (the class attribute to be predicted in the classification analysis or the dependent numeric attribute to be predicted in the regression analysis) for each instance are randomly permuted (shuffled) with each other [20, 42, 27, 39].

For each of the permuted datasets, the learning and prediction steps are performed, obtaining the estimates of the metrics. Then, the metrics obtained from the initial dataset are compared with the metrics obtained from each permuted dataset, calculating the percentage of results better or equal to the results from the initial dataset. This calculated percentage is the *p-value* of the permutation test. Finally, an α significance level is set as the threshold for assessing the significance of the results [20, 42].

In practice, $\alpha = 0.05$ or $\alpha = 0.01$ is used. The value of α indicates that the results obtained are significant if $p\text{-}value \leq \alpha$. In other words, the purpose of estimating the

p-value is to decide *p-value* $\leq \alpha$, thus rejecting the null hypothesis (often denoted by H_0 or *H-null*).

In inferential statistics, the null hypothesis defines that there is no relationship or no association between two measured phenomena or groups. The statistics area provides precise criteria for rejecting or accepting the null hypothesis whithin a confidence level.

Usually the null hypothesis is assumed to be true until there is an evidence that indicates the opposite. If there is no relationship between the measured phenomena, then the H_0 is true and it will be accepted, otherwise, if the measured phenomena are associated, then H_0 is false and it will be rejected, proving that the result is significant.

Modern statistical hypothesis tests have also an alternate hypothesis (often denoted by H_a), which is just the negation of the null hypothesis.

When there are too many possible data permutations to perform, a perfectly valid alternative is the *Monte Carlo permutation test* or *approximate permutation test*, which performs a randomic amount n of the total number of permutations [21]. After the n randomic permutations, it is possible to obtain the confidence level for the *p-value*.

The WEKA software [63] used in classification and regression analyses does not have an implementation of the permutation test allowing the user to evaluates the statistical significance of the results obtained. That said, as a contribution of this work, the permutation test was implemented for the WEKA (available in `https://github.com/luizponte/WEKA`).

In the implementation, a number of iterations is defined by the user, where in each iteration, the class values of all instances in training partition are randomly permutated with each other. After permuting the class values in an iteration, a classification or regression algorithm is applied, using the permutated training partition and the original (i.e., not permutated) test partition.

Depending on the analyses performed (classification or regression), the corresponding available metrics (e.g., accuracy and F-Measure for classification and correlation coefficient for regression) are calculated and compared with the results of these metrics obtained with the original training and test partitions. This comparison serves to calculate how many results with permutated training partitions were better than or equal to the results with the original training partition.

The n iterations performed generate n permutated training partitions that produce n results for each calculated metric. The *p-value* of a metric corresponds to the ratio

between the number of its results obtained that are better than or equal to the result obtained with the original training partition and the number of iterations performed.

An important thing to mention in the implementation is the *p-value* correction. After performing the n iterations defined by the user, if there is no result better than or equal to the result obtained with the original training partition, then the minimum *p-value* assumed will be $\frac{1}{n}$ instead of zero. This correction is made because, even with a bigger number of iterations, it is not possible to ensure that no iteration could obtain results better than or equal to the results obtained with the original training partition.

2.11 Final Remarks

This chapter has covered the main concepts used in this dissertation. In the health area, the PTSD was discussed, along with one of its scales (PCL). In the computer science area, the main DM techniques were addressed, including data preprocessing, classification and regression analyses, along with their algorithms. The main metrics and evaluation measures employed, along with validation techniques and significance tests of results, were also explained. The next chapter presents some studies related to identification and prediction of PTSD patients through statistical and AI analyses.

Chapter 3

Literature Review

As mentioned in Chapter 1, the diagnosis of psychiatric disorders is neither a quick nor a trivial task, because many disorders have symptoms in common. Thus, the correct diagnosis is fundamental for the effectiveness of the treatment.

Many studies apply computing to assist doctors and healthcare specialists in diagnosing and preventing disease and disorders. The results of these studies prove the importance of computing and present their contributions to healthcare.

This chapter deals with a review of the literature that analyzes data from individuals with psychiatric disorders or candidates to develop one of these disorders in the future. The studies are divided into sections according to the type of data collected from individuals.

Because they deal with psychiatric disorders such as PTSD, most of these studies use several scales that measure the degree of symptoms. At the end of the chapter, Section 3.5 discusses studies that evaluate the efficiency of some of these scales through statistical analyses and Table 3.1 shows the scales used by the studies presented in this chapter.

3.1 Demographic Data

Many studies apply data mining techniques or perform statistical analyses to demographic data of individuals with a disorder or disease. These data refer to individual characteristics (e.g., age, gender, weight), behavioral habits (e.g., smoking, drinking, physical activity) or to questionnaire responses and symptom measurement scales, for example.

The work in [40] proposes an intelligent hybrid system that combines classification and feature selection algorithms to classify individuals at risk of developing PTSD. In

the analyses, a dataset of 391 individuals was used, 321 of which are at risk of developing PTSD and 70 are not. The authors used the Sequential Minimal Optimization (SMO), Multilayer Perceptron and NB classifiers, as well as the feature extraction and selection techniques: Principal Component Analysis (PCA) and Correlation Based-Feature Selection (CFS). The classifiers presented accuracy between 74 and 79%.

PTSD does not only affect adult individuals. According to [12] and [13], in the United States (US), more than 20% of children under 16 have experienced at least one traumatic event and may eventually develop PTSD. In [48], SVM and RF algorithms, along with feature selection, were applied to a dataset of 163 hospitalized injured children, containing 105 features collected during the hospitalization period. PTSD was determined three months after leaving hospital. The feature selection technique was able to identify the 58 most relevant features.

In [34], the authors applied classification algorithms to a dataset of 13690 United Kingdom militaries. In addition to containing demographic data, the dataset contains the score of the PCL scale questionnaire. Although the individuals are military, the civil version of the PCL scale was applied, using the value 50 as cut-off point. Accoding to the authors, the PCL-C scale was selected because it is less restrictive in populations that may have experienced traumatic events unrelated to military deployment. The authors applied SVM, Random Forest, Artificial Neural Networks and Bagging algorithms, presenting accuracy equals to 91%, 97%, 89% and 95%, respectively. One of the limitations pointed out is the dataset imbalance because only 3.95% of the individuals correspond to possible PTSD patients.

3.2 Image Data

Imaging tests such as MRI and EEG are also widely used in research that seeks to predict and identify biomarkers or risk factors in individuals with a psychiatric disorder.

The work published in [33] analyzes, through Functional Magnetic Resonance Imaging (fMRI), the areas of brain activation in patients with PTSD who present dissociative responses by recalling the traumatic events experienced. The dissociative response consists in the loss of memory, awareness or perception of the environment when trying to remember the traumatic event, being considered as an involuntary defense mechanism of the organism in order to preserve the individual. Individuals presenting dissociative responses do not accurately remember the details of the event. The results of this study re-

veal that individuals with PTSD had greater brain activation in certain regions compared to individuals in the control group (i.e., healthy individuals).

Authors in [35] conducted an experiment with 14 PTSD war veterans, 11 non-PTSD fighters, and 14 control subjects. The authors investigated the role of certain brain regions in PTSD patients by measuring blood flow in these regions using Single Photon Emission Computerized Tomography (SPECT) images. During image collection, subjects were exposed to white noise and combat noise (e.g., battlefield sounds).

In [43], authors applied regression techniques such as the RVR algorithm and pattern recognition in fMRI images of 57 young people to determine the participants' behavioral and emotional dysregulation. The algorithm was able to identify patterns of neural activity associated with dysregulation, indicating the brain regions with the highest contribution.

3.3 Physiological Data

Heart rate, skin conductance, and blood pressure are some of the types of physiological data that are commonly collected when conducting experiments with individuals with psychiatric disorders. These data, compared to image data, are generally more accessible and easier to collect due to the simplicity of commercially available equipment and sensors.

The work in [52] studies the relationship between heart rate and blood pressure collected immediately after a traumatic event and the subsequent development of PTSD. In that study, 86 survivors of traumatic events were monitored for four months, having their data and symptoms collected and measured upon arrival at the hospital, on first week, first month and fourth month after arrival. Out of the 86 individuals, 20 were diagnosed with PTSD in the fourth month. The results of the study show that the 20 individuals diagnosed with PTSD had high heart rates upon arrival at the hospital and at the first week, compared to the other individuals. In contrast, blood pressure did not differ.

HRV consists of changes in the time interval between consecutive beats. HRV in a healthy heart is complex and constantly occurring so that the cardiovascular system can quickly react to physical and physiological changes [51]. In [11], the authors analyze HRV by providing a dynamic map of sympathetic and parasympathetic interactions. Through an Eletrocardiogram (ECG), they collected the heart rate of 18 individuals, nine of them had PTSD and nine belonged to the control group. The experiment was divided into two stages: in the first stage the individuals remained at rest and in the second stage the

individuals remembered traumatic or stressful situations experienced. The results show that individuals with PTSD had sympathetic hyperactivity and reduced parasympathetic activity at rest.

The work in [18] analyzed the physiological responses of individuals while viewing trauma-related images. The study gathered 86 participants, 37 of them were victims of recent traumatic events, 18 were PTSD patients and 31 were healthy individuals (control group). The results showed that victims of recent traumatic events and those with PTSD presented acceleration (tachycardia) during image visualization.

In [36], by measuring skin conductance and blood oxygenation level using fMRI images, it was seen that the patterns of brain activity are different between individuals with anxiety disorders and the control group.

3.4 Molecular Data

Molecular data, unlike the other types of data mentioned, are used less frequently for studies related to the prediction of psychiatric disorders. However, this type of data contributes to a large amount of features, proving interesting for analyses of identification of biomarkers of such disorders.

Unlike the studies mentioned using demographic, physiological and imaging data, the authors in [16] used molecular data from blood samples of 165 war-exposed soldiers, 83 of them were PTSD patients. By applying SVM, RF, and Decision Tree algorithms, along with feature selection, the authors were able to identify 28 biomarkers from 343 candidate PTSD biomarkers.

3.5 Scales Analyses

Many psychiatric disorders such as depression, PTSD, and anxiety disorders have their own scales that measure the degree of their symptoms. Doctors and specialists use the scales as a diagnostic tool and also use them for treatment as a way to monitor the development of symptoms.

A research has found that a woman with PTSD is 1.4 times more likely to become alcoholic than a woman without PTSD [46]. The work in [8] suggests that the co-occurrence of PTSD and alcohol abuse result in symptomatology increase and poorer treatment out-

comes.

In [25], authors analyze the Penn Inventory and PCL-C scales to find a cut-off point that maximizes the prediction of substance-dependent women who met the PTSD diagnostic criteria. This study aims at determining an optimal cut-off point that presents a balancing between sensitivity, specificity and accuracy. Demographic data (e.g., age, marital status, income, education, number of children and use of Alcohol Anonymous and Narcotics Anonymous resources) of 44 women were collected. The value 38 used as cut-off point in the PCL-C scale maximized the number of women identified with PTSD and minimized the false positives and false negatives.

In [6], authors evaluate the diagnostic efficiency of the Primary Care PTSD Screen (PC-PTSD) and PCL scales as a clinical screening tool applied to 352 active soldiers recently returned from a combat deployment. This study evaluated the item-level characteristics of both scales. The overall diagnostic efficiency measured by the area under the curve (AUC) was virtually the same for both scales.

According to the results, the best PCL cut-off point values for primary care settings were between 30 and 34 and presented specificity equals to 0.9 and sensitivity greater than 0.7. The analyses also identified that the most relevant scale item belongs to avoidance symptoms. Although lower values (< 50) are not recommended as cut-off points for military, the evaluated cut-off point presented desired results.

The authors raised a question regarding the way the questionnaire is applied to the samples of individuals. Higher values are known to indicate a high probability of having PTSD. Therefore, when applying the questionnaire anonymously to a sample, the individuals tend to be more sincere. On the other hand, when gathering personal data that can identify an individual, he does not tend to be sincere when assessing the degree of symptoms, as he fears being referred to some treatment that may result in losing his job.

Another point raised concerns individuals who are already seeking treatment and individuals who do not admit to have a problem. Individuals who openly assume to be seeking treatment tend to be more sincere in assessing the degree of symptoms when completing the questionnaire.

The work in [38] investigated the association between PTSD chronic patients and clinicians ratings of PTSD symptoms over the course of treatment and follow-up, using the PCL and CAPS scales. The investigation was made with two randomized clinical trials of 360 veterans with chronic PTSD, using data analytic methods. The results presented

a significant association between PTSD patients and clinicians ratings over the course of treatment, showing that chronic PTSD patients do self-report changes in their symptoms across treatment and time.

3.6 Miscellaneous

Although many works use one data type, it is not unusual to find works that use two or more data types. The studies in [23] and [29] used a dataset of 957 survivors of recent traumatic events at the time of admission to the hospital's emergency department. The used dataset contains the following types of data: demographic, trauma type, loss of consciousness during the traumatic incident, head injury, whiplash injury, blood pressure, pulse, perceived pain, prescribed analgesics and duration of emergency department admissions. Survivors were monitored for 15 months and data were collected on admission to the hospital and on the 7^{th} and 15^{th} month after admission. The authors applied SVM, RF, and KRR algorithms, along with feature selection, to find the minimum subset of features that maximize the PTSD prediction. The results support the hypothesis of the existence of multiple sets of risk factors associated with PTSD.

Similar to [23] and [29], the work in [30] applied the SVM algorithm along with feature selection to a dataset of 561 soldiers before and after missions, in order to identify risk indicators and predict PTSD responses. The study in [37] predicts individuals with PTSD by applying the RF, SVM and Decision Tree algorithms to a dataset with demographic and molecular data of 51 individuals with PTSD and 51 with other disorders.

3.7 Final Remarks

According to the literature related to prediction of PTSD patients or traumatic events victims through computer science techniques using physiological data and scales (e.g., AI, DM and PR techniques), it was possible to verify that the PCL scale has not been widely explored yet. Most of the studies found use other scales related to PTSD symptoms, as illustrated in Table 3.1.

Table 3.1: Scales used in reviewed work

Scale	Reviewed Work
Acute Stress Disorder Scale (ASDS)	[29]
Beck Depression Inventory (BDI)	[18, 30]
Clinician-Administered Dissociative State Scale (CADSS)	[33]
Clinician-Administered PTSD Scale (CAPS)	[11, 16, 25, 33, 37, 38]
Clinical Global Impression (CGI)	[29]
Dissociative Experience Scale (DES)	[33]
Hamilton Anxiety Rating Scale (HAM-A)	[37]
Hamilton Depression Rating Scale (HAM-D)	[11, 37]
Impact of Events Scale (IES)	[18, 52]
Kessler Psychological Distress Scale 6 (K6)	[29]
MISS or M-PTSD	[52]
Positive and Negative Affect Schedule (PANAS)	[30]
Positive and Negative Syndrome Scale (PANSS)	[37]
Primary Care PTSD Screen (PC-PTSD)	[6]
Posttraumatic Stress Disorder Checklist (PCL)	[6, 25, 30, 34, 38]
Peritraumatic Dissociative Experiences Questionnaire (PDEQ)	[52]
PTSD Symptom Scale (PSS)	[29]
Penn Inventory	[25]
Structured Clinical Interview for DSM-IV (SCID)	[33, 35]
Symptom Checklist (SCL)	[30]
State-Trait Anxiety Inventory (STAI)	[18, 36, 52]
Trauma History Questionnaire (THQ)	[52]
Traumatic Life Events Questionnaire (TLEQ)	[30]
UCLA PTSD Reaction Index	[48]

As mentioned, the study [34] uses demographic data to predict whether a military individual is a PTSD patient or not. The authors used the PCL-C version and selected the value 50 as cut-off point. The study [30] also uses demographic data with the PCL-C version and applies ML methods to assess the potential for pre- and early post-deployment prediction of PTSD development in soldiers.

While studies [6], [25] and [38] evaluate the PCL scale, comparing and measuring its results with other PTSD scales, this work explores the use of the PCL scale along with physiological signals of heart rate and skin conductance. As illustrated in Table 3.2, the studies with the PCL scale found do not use physiological signals. By applying classification and regression algorithms to predict PCL scale scores, it is possible to evaluate a correlation between the scale and physiological data.

Table 3.2: Data type and applied techniques of reviewed work using PCL scale

Reviewed Work	Data	Applied Techniques
[6]	Demographic and Scale	Generalized Additive Models (Statistical Model)
[25]	Demographic and Scale	Statistical Analysis
[30]	Demographic	Attribute Selection and Classification (SVM)
[34]	Demographic	Classification (Artificial Neural Networks, Bagging, SVM, RF)
[38]	Scale	Regression Analysis (Longitudinal Data Analysis)
Our Work	Physiological Signals	Classification, Supervised Discretization, Attribute Selection, Class Balancing, and Regression

Chapter 4

Dataset

The dataset used in this work was obtained through an experiment conducted at the Biomedical Institute at UFF [2, 3] and approved by the Research Ethics Committee of the *Hospital Universitário Antônio Pedro* (HUAP), UFF, under statement *203/09* of December 11, 2009.

The experiment was conducted in a special room, with sound attenuation and indirected lighting, located at the Physiology and Pharmacology Department. The physiological signals of heart rate and skin conductance of volunteers who suffered some traumatic event related to violence were collected. The skin conductance signal measures the electrical variation response of the skin. This electrical variation is measured through the Number of Responses (NR) and its unity is micro Siemens (μS).

In addition, version IV of the PCL-C scale was applied to assess the degree of PTSD symptoms. This chapter therefore describes the dataset used, explaining how the data was acquired and describing the characteristics and information of each attribute, in Sections 4.1 and 4.2 respectively. In Section 4.3, the supervised discretization is discussed, explaining the problem faced with the existing algorithms and the techniques applied to minimize it.

4.1 Data Acquisition

As mentioned, the dataset used in this work was provided by UFF Biomedical Institute. The authors performed an experiment with 83 volunteers who were victims of some traumatic event [2, 3]. Due to data acquisition problems, volunteers using drugs acting on the central nervous system and presenting cardiac arrhythmias were removed, thus the

dataset provided contains data from 55 volunteers.

The original goal of the experiment was to investigate the autonomic response of heart rate and skin conductance caused by the visualization of images related to violence and to explore how stimulus directionality and the experience of traumatic events influence the cardiac reactivity of the individual.

The volunteers sat in front of a computer monitor, using a support for positioning their forehead and chin, so that the distance between their eyes and the monitor was 57 centimeters. The E-prime® software[1] was used to generate the stimuli presented on the monitor. The heart rate and skin conductance signals were recorded using the BIOPAC Acqknowledge 3.9.0 software.[2]

Physiological signals of heart rate and conductance of the volunteers were collected while viewing images related to violence. The images were divided into blocks of Threat Directed Towards (TDT) and Threat Directed Away (TDA). Each block contained 16 emotional stimulus and 16 neutral stimulus images, totaling 32 images per block.

TDT images consist of a person holding a gun directed at the viewer, belonging to the emotional stimulus images of the directed towards block. Neutral Threat Directed Towards (NeutralTDT) images consist of a person holding an object (e.g., camera, microphone, umbrella and binoculars) directed at the viewer, belonging to the neutral stimulus images of the directed towards block.

TDA images consist of a person holding a gun directed at a third party, belonging to the emotionally stimulus images of the directed away block. Neutral Threat Directed Away (NeutralTDA) images consist of a person holding an object (e.g., camera, microphone, umbrella and binoculars) directed at a third party belonging to the neutral stimulus images of the directed away block.

The emotional and neutral stimuli images selection used the following criteria:

- Ethnicity: the ethnicity of the people present in the photos was balanced and it was sought to select individuals who had common physical characteristics to the Brazilian population;
- Number of people: each image contains only one person to isolate the main features of the scene such as facial and gesture expression;

[1] `https://pstnet.com/products/e-prime/`
[2] `https://www.biopac.com/manual/acqknowledge-3-9-software-guide/`

- Gender: the images were composed of male people only;
- Physical parameters of the image: dimension, brightness, contrast and spatial frequency.

After gathering the images for the experiment, it was necessary to perform a preprocessing, so that all elements and artifacts that interfere in the emotional content could be removed from the image. The images had to present the same dimensions of 1024 x 768 pixels.

Emotional and neutral images had also to be paired in relation to the following criterion: brightness, contrast and spatial frequency (through the Fast Fourier Transform (FFT)), according to the methodology stablished in [7]. The goal doing that is to create homogeneous samples of images, so that they could be allocated into blocks having emotional and neutral stimuli with equivalent complexity. Controlling that, it is expected to minimize the physical characteristics influence on the emotional results.

Each image was displayed for 6 seconds, interspersed by a fixation cross located in the center of the screen and displayed for 6 to 8 seconds. During the 6 seconds of image visualization, heart rate and electrical variation response (number of responses) of the skin were gathered. These signals collected were sampled in 12 points of 0.5 second.

Neutral stimulus images (NeutralTDT and NeutralTDA) were pseudorandomly displayed within their blocks to prevent many images of the same stimulus from being presented in sequence. The visualization of each images block lasted approximately 7 minutes and between the blocks there was a pause, when the experimenter entered the room to verify if everything was correct with the volunteer and to warn him about the beginning of the next block.

Figure 4.1 illustrates the image visualization step of the experiment. At the end of the image visualization, the volunteers were asked to complete the PCL-C scale questionnaire and were released from the experiment.

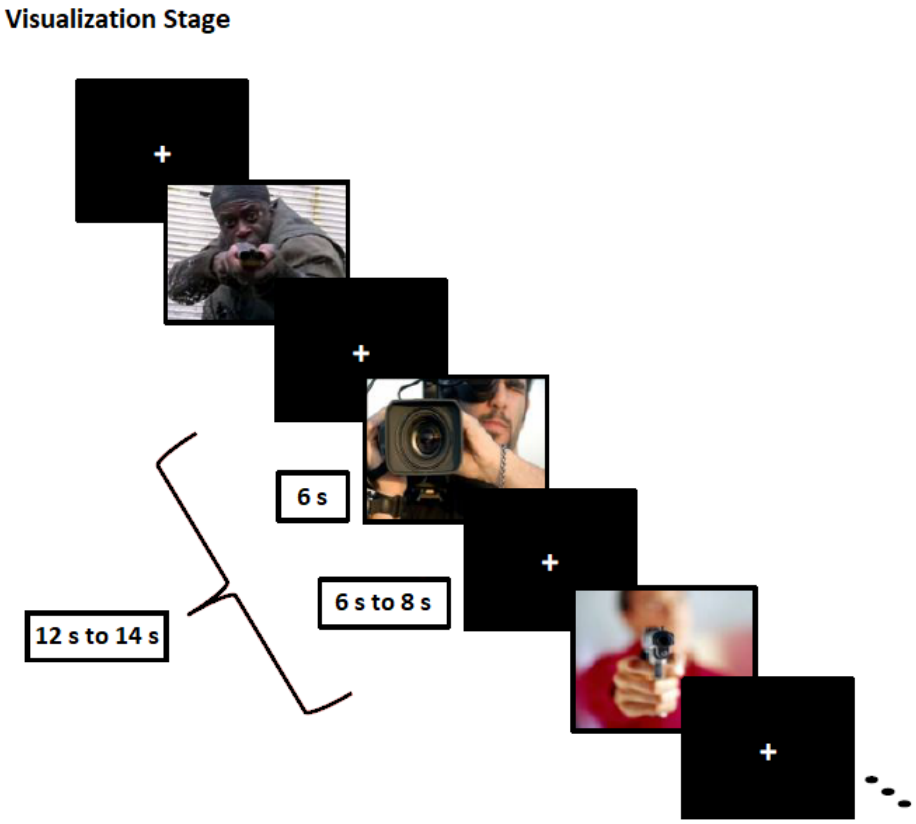


Figure 4.1: Experiment visualization stage [2]

4.2 Data Description

The dataset of 55 volunteers used in this work contains five attributes referring to physiological heart rate signals, five attributes referring to skin conductance signals and one attribute referring to the score of the PCL-C scale, totaling 11 attributes.

Originally, the attribute referring to the PCL-C scale score is numeric and can assume values from 17 to 85 (in the dataset, the minimum value found is 17 and the maximum is 77). In the classification analyses, which will be explained in Chapter 5, the PCL-C scale scores were converted into two classes (*high* and *low*), following the range of cut-off points suggested by the scale, which is 36. In the regression analyses, which will be explained in Chapter 6, the original PCL-C scale scores were used. Table 4.1 contains the dataset independent attributes with their types and minimum and maximum values.

Table 4.1: Dataset independent attributes

Attribute	Signal Type	Minimum Value	Maximum Value
TDT	Heart Rate	-4.81	2.465
TDA	Heart Rate	-5.179	3.674
NeutralTDT	Heart Rate	-2.91	3.436
NeutralTDA	Heart Rate	-3.167	2.048
NR_TDT	Skin Conductance	0	8
NR_TDA	Skin Conductance	0	11
NR_NeutralTDT	Skin Conductance	0	5
NR_NeutralTDA	Skin Conductance	0	8
HR_Threat-Neutral	Heart Rate	-3.517	2.622
NR_Threat-Neutral	Skin Conductance	-3	7

The meaning of each independent attribute is explained as follows:

- TDT: contains the average of the HR signal values collected during the six seconds (12 sample points of 0.5 second each) of visualization of all 16 TDT images. In other words, firstly an average of the 12 sample points of one image is calculated, resulting in the HR average value of that image. Then this process is applied to the rest of the TDT images, calculating their HR average value. At this moment, there are 16 HR average values (one for each image). Lastly, it is calculated the final average of the 16 HR average values, obtained in the calculation before;

- TDA, NeutralTDT and NeutralTDA: the same calculation described in the TDT attribute is performed, using the HR signal values collected during the visualization of the corresponding images;

- NR_TDT: contains the average of the NR values (i.e., electrical variation response of the skin) collected during the six seconds (12 sample points of 0.5 second each) of visualization of all 16 TDT images. In other words, firstly an average of the 12 sample points of one image is calculated, resulting in the NR average value of that image. Then this process is applied to the rest of the TDT images, calculating their NR average value. At this moment, there are 16 NR average values (one for each image). Lastly, it is calculated the final average of the 16 NR average values, obtained in the calculation before;

- NR_TDA, NR_NeutralTDT and NR_NeutralTDA: the same calculation described in the NR_TDT attribute is performed, using the NR values collected during the visualization of the corresponding images;

- HR_Threat-Neutral: contains the subtraction of neutral stimulus (NeutralTDT and NeutralTDA) from emotional stimulus (TDT and TDA) of HR signal values. The calculation is performed using Equation 4.1;

$$\text{HR\_}\textit{Threat-Neutral} = \frac{(\text{TDT} + \text{TDA})}{2} - \frac{(\text{NeutralTDT} + \text{NeutralTDA})}{2} \quad (4.1)$$

- NR_Threat-Neutral: contains the subtraction of neutral stimulus (NR_NeutralTDT and NR_NeutralTDA) from emotional stimulus (NR_TDT and NR_TDA) of NR values. The calculation is performed using Equation 4.2.

$$\text{NR\_Threat-Neutral} = (\text{NR\_TDT} + \text{NR\_TDA}) - (\text{NR\_NeutralTDT} + \text{NR\_NeutralTDA}) \quad (4.2)$$

4.3 Supervised Discretization

As explained in Section 2.5.2, the discretization is a form of data transformation, in which numerical values are transformed into nominal values. Using discretized values can make the mining process more efficient and the patterns found more intelligible. There are two types of discretization: supervised and unsupervised. When the class attribute is used to determine the split-points for delimiting the attribute ranges (named bins), it is called supervised discretization. In this work, the independent attributes were discretized through supervised discretization.

The data mining analyses of this work were performed using WEKA [63] software. This software, in addition to having several algorithms for data mining analysis (e.g., classification, regression and clustering), also has several algorithms for data preprocessing, including the supervised discretization, which implements the discretization proposed in [19]. However, the supervised discretization algorithm of WEKA could not determine more than just one bin for the attributes used in this work. That means, when applying the supervised discrezation algorithm for each attribute, its values were placed into a single bin (called "All" in WEKA).

There are other softwares that can be used in DM analysis besides WEKA. Two of these softwares are the Python library *scikit-learn* (version 0.22) [41] and the *R* programming language [45], which is more focused on statistical analysis.

Currently the *scikit-learn* library has only one discretization algorithm, the *KBinsDiscretizer*, from the *sklearn.preprocessing* package. This algorithm corresponds to un-

supervised discretization and calculates the bins using the *equal-width* or *equal-frequency* technique, according to its parameter setting.

The *R* programming language has a function (algorithm) named *discretizeDF.supervised*, located in package *arulesCBA* (version 1.1.5). This function also implements the discretization proposed in [19], presenting the same result obtained with the WEKA supervised discretization algorithm, i.e., a single bin for each attribute.

To address the problem of getting a single bin, a brute force discretization was implemented in this work (available in `https://github.com/luizponte/WEKA`), using the following metrics explained in Section 2.8: *information gain*, *gain ratio* and *gini index*. During the execution, the brute force evaluated all the possible split-points to select those that maximize each metric. It is important to note that the brute force is not often feasible and recommended as it can be computationally costly. However, as the number of instances and attribute in our dataset is small, the brute force was feasible.

The brute force implementation was applied to each fold obtained in the 5-Fold Stratified Cross-Validation and was executed four times for each metric. The first execution determined two bins, the second execution determined three bins, and the fourth execution determined five bins. After determining the bins, we evaluated which amount of bins (from two to five) produced the best value for each metric *information gain*, *gain ratio* and *gini index*.

As a result, the supervised discretization produced three discretized dataset, one for each metric used. For those cases where there were two or more amounts of bins that produced the best metric value, the minimum amount was chosen. Tables 4.2, 4.3 and 4.4 contain the number of bins determined for each attribute using the *information gain*, *gain ratio* and *gini index* metrics, respectively. The results of the classification analyses performed with this supervised discretization will be presented in Chapter 5.

Table 4.2: Number of bins for each attribute per fold determined by information gain

Attribute with Information Gain	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
TDT	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
TDA	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NeutralTDT	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NeutralTDA	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NR_TDT	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NR_TDA	5 Bins	5 Bins	5 Bins	5 Bins	4 Bins
NR_NeutralTDT	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NR_NeutralTDA	4 Bins	5 Bins	5 Bins	5 Bins	4 Bins
HR_Threat-Neutral	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NR_Threat-Neutral	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins

Table 4.3: Number of bins for each attribute per fold determined by gain ratio

Attribute with Gain Ratio	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
TDT	2 Bins	5 Bins	2 Bins	2 Bins	2 Bins
TDA	3 Bins	2 Bins	4 Bins	2 Bins	2 Bins
NeutralTDT	4 Bins	5 Bins	5 Bins	5 Bins	2 Bins
NeutralTDA	2 Bins	3 Bins	3 Bins	3 Bins	3 Bins
NR_TDT	2 Bins	3 Bins	2 Bins	3 Bins	3 Bins
NR_TDA	2 Bins	2 Bins	2 Bins	2 Bins	2 Bins
NR_NeutralTDT	2 Bins	4 Bins	2 Bins	2 Bins	2 Bins
NR_NeutralTDA	2 Bins	2 Bins	2 Bins	2 Bins	2 Bins
HR_Threat-Neutral	2 Bins	2 Bins	2 Bins	2 Bins	5 Bins
NR_Threat-Neutral	3 Bins	5 Bins	3 Bins	3 Bins	3 Bins

Table 4.4: Number of bins for each attribute per fold determined by gini index

Attribute with Gini Index	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
TDT	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
TDA	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NeutralTDT	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NeutralTDA	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NR_TDT	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NR_TDA	5 Bins	5 Bins	5 Bins	5 Bins	4 Bins
NR_NeutralTDT	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NR_NeutralTDA	4 Bins	5 Bins	5 Bins	5 Bins	4 Bins
HR_Threat-Neutral	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins
NR_Threat-Neutral	5 Bins	5 Bins	5 Bins	5 Bins	5 Bins

4.4 Final Remarks

This chapter presented the dataset used in this work. We explained how the experiment conducted at the Biomedical Institute at UFF collected trauma victims volunteers' physiological data (heart rate and skin conductance) during violence stimuli images visualization. The meaning of each dataset attribute was also explained.

This chapter also discussed the supervised discretization we applied, approaching the existing algorithms, the problem faced and the implementation developed.

Chapter 5 presents the results obtained in the classification analyses, applying the supervised discretization, class balancing and attribute selection techniques.

Chapter 5

Classification

This chapter discusses the classification analyses performed with the *WEKA* software (version 3.8.3) and its Application Programming Interface (API) [63], presenting the results obtained. As explained in Section 2.2, the PCL scale consists of a questionnaire with 17 questions, each being evaluated with values from 1 to 5. Thus, the value of the PCL scale is numeric, resulting from the sum of each value of the 17 questions.

As discussed in Section 2.6, classification analyses predict categorical (i.e., nominal) values of an attribute (called class). As also explained in Chapter 4, the PCL attribute has been transformed into a nominal attribute for classification analyses. To do that, value 36 was chosen as cut-off point, according to the suggested range of cut-off points by the PCL scale (see Table 2.2). Therefore the *high* class value was assigned to instances where PCL score ≥ 36 and the *low* class value was assigned to instances where PCL score < 36.

The analyses were performed using the 5-Fold Stratified Cross-Validation method, obtaining the results for accuracy, precision, recall and F-Measure measures. These last three measures correspond to the results of the minority class (*high* PCL). To evaluate the statistical significance of the obtained results, the permutation test algorithm, which we implemented for WEKA software, was applied, performing 1000 permutations and using $\alpha = 0.05$.

An important concept to consider in classification analyses is the baseline. The baseline used in our analyses corresponds to the percentage of instances belonging to the majority class in relation to the total of instances in dataset. The dataset used in this work has 55 instances, 41 of them belong to the *low* class (majority class). The baseline for this case is 74.55% and it means that a classifier can get 74.55% of accuracy if it "blindly" classifies any instance of the dataset as belonging to the *low* PCL class.

The imbalance of the dataset used directly affects accuracy. That is, a relatively high accuracy such as 74.55% may seem attractive, however, when evaluating how many instances of the minority class (*high* PCL) were correctly classified, it can be possible to check if the classifier is just voting for the majority class, meaning in this case that the classifier could not identify any pattern during the learning step.

The baseline serves as an evaluation criterion, where higher accuracy is desired. That is, an accuracy higher than the baseline indicates that the classifier is beginning to correctly classify instances belonging to the minority class.

As discussed in Section 2.6, the following five WEKA software algorithms were applied to the classification analyses:

1. IBk: this is the implementation of the k-NN algorithm in WEKA [1];
2. J48: this is the implementation of the *C4.5* Decision Tree algorithm in WEKA [44];
3. Naïve Bayes (NB) [28];
4. Random Forest (RF) [9];
5. SMO: this is the implementation of the SVM algorithm for classification in WEKA [26, 31].

As explained in Section 2.6.1, the WEKA J48 algorithm has the *confidenceFactor* (CF) and the *unpruned* parameters, related to the decision tree pruning. The classification analyses performed used the values 0.25, 0.5 and 0.75 for the *confidenceFactor* parameter. Although it is not possible to assign the value 1 to this parameter (representing no pruning), another *J48*'s parameter, called *unpruned*, behaves the same way, when it receives the value *true* (meaning that there is no pruning). Therefore, this last parameter received the value *true* to represent a *confidenceFactor* equal to 1.

As Section 2.6.3 explains, the RF implementation has the *numIterations* (NI) parameter, related to the amount of decision tree models built that will vote. The classification analyses perfomed used the values 100, 500, 1000 and 10000 for this parameter.

SVM algorithm has the *kernel* and the *C* parameters (see Section 2.6.4). In the classification analyses performed, the kernels *PolyKernel*, *Pearson Universal Kernel (Puk)* and *RBFKernel* were used by SVM implementation in WEKA (called SMO). The *C* parameter, however, was not varied (the default value 1 was used).

The implementation of k-NN (called *IBk* in WEKA) has the k parameter, which refers to the number of training instances, searched in a pattern space, that are closest to the test instance. In the classification analyses performed, the k parameter was varied with values from 1 to 15 and the *Euclidean distance* was used as distance metric, when searching the k training instances (see Section 2.6.5).

The five selected algorithms are widely used in classification analyses. The J48 and the Random Forest algorithms belong to algorithms that build decision tree models. The Naïve Bayes algorithm belongs to the probabilistic algorithms. The SMO is the implementation of the SVM algorithm, widely used in classification and also in regression analyses. The IBk algorithm, unlike the previous four algorithms, uses the lazy learning approach and can also be used in regression analyses.

This chapter has six main sections. Section 5.1 explains the methodology used. Section 5.2 deals with analyses performed with the unbalanced dataset (i.e., original dataset). Section 5.3 deals with the analyses performed using the class balancing technique, explained in Section 2.5.3. Section 5.4 explains the Auto-WEKA plugin used. Section 5.5 evaluates the results obtained using a higher PCL cut-off point value (44). Finally, Section 5.6 discusses the main points observed with the analyses.

5.1 Methodology

As explained, five classification algorithms were selected and applied to the dataset. Aiming at increasing the classifiers performance, their hyperparameters values were varied and those with the highest accuracy were selected. To validate the results obtained, the 5-Fold Stratified Cross-Validation method along with the implementation of the permutation test (performing 1000 permutations) were used.

To increase the results obtained by the classifiers, DM techniques such as supervised discretization, attribute selecion and class balancing were applied to the dataset and compared with the previous results. Moreover, the Auto-WEKA plugin (see 5.4) was used to obtain a suggested algorithm and its hyperparameters values. This suggestion consists of the algorithm that obtained the best result for a selected measure (the accuracy measure was selected for the analyses performed).

Lastly, using another cut-off point value could be interesting to the domain specialists. Thus, the cut-off point value 44 was selected and the previous classification analyses (i.e., those with the cut-off point value 36) were performed once more.

5.2 Unbalanced Dataset

This section deals with the analyses performed with the unbalanced dataset. In an attempt to improve the accuracy obtained by each classifier, supervised discretization and attribute selection techniques were also applied. The application of these techniques will be discussed in the Sections 5.2.2 and 5.2.3.

5.2.1 Non-discretized Dataset

This section deals with classification analyses performed with the non-discretized unbalanced dataset (i.e., original dataset). Table 5.1 contains the results obtained with the IBk algorithm (k-NN), varying its k parameter with values from 1 to 15. The best accuracy was obtained with $k = 4$ and $k = 7$, however, comparing their results, $k = 4$ was selected because it also obtained the best F-Measure. It presented accuracy of 76.36% (p-value of 0.005), precision of 0.5294 (p-value of 0.005), recall of 0.6429 (p-value of 0.002) and F-Measure of 0.5806 (p-value of 0.001).

Table 5.1: IBk results with non-discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure
IBk k = 1	74.55%	0.5	0.6429	0.5625
IBk k = 2	61.82%	0.3704	0.7143	0.4878
IBk k = 3	74.55%	0.5	0.5	0.5
IBk k = 4	**76.36%**	**0.5294**	**0.6429**	**0.5806**
IBk k = 5	74.55%	0.5	0.2857	0.3636
IBk k = 6	70.91%	0.4444	0.5714	0.5
IBk k = 7	76.36%	0.6667	0.1429	0.2353
IBk k = 8	70.91%	0.3333	0.1429	0.2
IBk k = 9	74.55%	0.5	0.0714	0.125
IBk k = 10	70.91%	0.25	0.0714	0.1111
IBk k = 11	72.73%	0	0	NaN
IBk k = 12	72.73%	0	0	NaN
IBk k = 13	72.73%	0	0	NaN
IBk k = 14	72.73%	0	0	NaN
IBk k = 15	72.73%	0	0	NaN

It is important to highlight that in some results it was not possible to calculate precision, recall or F-Measure, being indicated by Not a Number (NaN) value in the tables. This occurs when the denominator of the corresponding measure equation is equal to zero. Consequently, when a measure can not be calculated, its *p-value* can not be calculated either.

Table 5.2 contains the results obtained with the J48 algorithm, varying its parameters *confidenceFactor* and *unpruned*. The higher the *confidenceFactor* value, the lower the pruning performed on the decision tree. The *unpruned* parameter equal to *true* is equivalent to a *confidenceFactor* equal to 1. Although the results with *confidenceFactor* equal to 0.5, 0.75 and *unpruned* presented the best measures values, the *confidenceFactor* = 0.5 presented the best results, since lower values of *confidenceFactor* represents more pruning, helping to the decision tree's performance and readability. The best accuracy obtained was 70.91% (p-value of 0.206), precision of 0.4375 (p-value of 0.085), recall of 0.5 (p-value of 0.008) and F-Measure of 0.4667 (p-value of 0.01).

Table 5.2: J48 results with non-discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure
J48 Confidence Factor 0.25	69.09%	0.3636	0.2857	0.32
J48 Confidence Factor 0.5	**70.91%**	**0.4375**	**0.5**	**0.4667**
J48 Confidence Factor 0.75	70.91%	0.4375	0.5	0.4667
J48 Unpruned	70.91%	0.4375	0.5	0.4667

Table 5.3 contains the results obtained with the Naïve Bayes algorithm. Since there is no parameter variation for Naïve Bayes algorithm, the best result presented accuracy of 67.27% (p-value of 0.25), precision of 0.375 (p-value of 0.12), recall of 0.4286 (p-value of 0.132) and F-Measure of 0.4 (p-value of 0.08).

Table 5.3: Naïve Bayes results with non-discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	**67.27%**	**0.375**	**0.4286**	**0.4**

Table 5.4 contains the results obtained with the Random Forest algorithm, varying its *numIterations* parameter with the values 100, 500, 1000 and 10000. Since all parameter variations presented the same measures values, the best result was obtained with *numIterations* = 100, because less iterations were needed. The best result presented accuracy of 70.91% (p-value of 0.433), precision of 0.4 (p-value of 0.186), recall of 0.2857 (p-value of 0.042) and F-Measure of 0.3333 (p-value of 0.037).

Table 5.4: Random Forest results with non-discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure
Random Forest Iterations 100	**70.91%**	**0.4**	**0.2857**	**0.3333**
Random Forest Iterations 500	70.91%	0.4	0.2857	0.3333
Random Forest Iterations 1000	70.91%	0.4	0.2857	0.3333
Random Forest Iterations 10000	70.91%	0.4	0.2857	0.3333

Table 5.5 contains the results obtained with the SMO algorithm using three kernel types (*PolyKernel*, *Puk* and *RBFKernel*) in their *kernel* parameter. Since the three kernel types presented the same measures values, it was not possible to select a best result for them. The three kernel types *PolyKernel*, *Puk* and *RBFKernel* presented accuracy p-values of 0.889, 0.917 and 1, respectively.

Table 5.5: SMO results with non-discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure
SMO PolyKernel	74.55%	NaN	0	NaN
SMO Puk	74.55%	NaN	0	NaN
SMO RBFKernel	74.55%	NaN	0	NaN

As the results showed so far, the IBk algorithm presented the highest accuracy using the non-discretized unbalanced dataset. Although the obtained accuracy of 76.36% is higher than the baseline, it is not satisfactory yet. Thus, the supervised discretization technique will be applied, aiming at increasing the classifiers performance.

5.2.2 Discretized Dataset

In order to improve the performance of the classifiers, increasing accuracy, the supervised discretization technique was applied to the ten numerical independent attributes of the original dataset used. Supervised discretization was performed applying the brute force algorithm for the *information gain*, *gain ratio* and *gini index* metrics, as explained in Section 4.3. For this, the five training and test folds of the 5-Fold Stratified Cross-Validation method were obtained and the supervised discretization algorithm was performed in each of the five training folds. The test folds were discretized using the bins obtained in each of the corresponding training folds.

Table 5.6 contains the results obtained with the IBk algorithm (k-NN), varying its parameter k with values from 1 to 15. Since the variations of each metric (i.e., *gain ratio*, *gini index* and *information gain*) presented the same measures values, the $k = 9$ was selected for the best results because less neighbors were needed and it used *gain ratio* metric. The best result presented accuracy of 74.55% (p-value of 0.898), recall of 0 (p-value of 1) and it was not possible to calculate the precision and neither the F-Measure.

Table 5.6: IBk results with discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric
IBk k = 9	**74.55%**	**NaN**	**0**	**NaN**	**Gain Ratio**
IBk k = 10	74.55%	NaN	0	NaN	Gini Index
IBk k = 12	74.55%	NaN	0	NaN	Information Gain

Table 5.7 contains the results obtained with the J48 algorithm, varying its parameters *confidenceFactor* and *unpruned*. The best result was obtained with *confidenceFactor* = 0.25, using the *gain ratio* metric and presented accuracy of 70.91% (p-value of 0.976), precision of 0.25 (p-value of 0.148), recall of 0.0714 (p-value of 0.154) and F-Measure of 0.1111 (p-value of 0.148).

Table 5.7: J48 results with discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric
J48 Confidence Factor 0.25	**70.91%**	**0.25**	**0.0714**	**0.1111**	**Gain Ratio**
J48 Confidence Factor 0.25	65.45%	0	0	NaN	Gini Index
J48 Confidence Factor 0.25	69.09%	0	0	NaN	Information Gain

Table 5.8 contains the results obtained with the Naïve Bayes algorithm. The best result was obtained using the *gini index* metric, presenting accuracy of 72.73% (p-value of 0.139), precision of 0.4444 (p-value of 0.099), recall of 0.2857 (p-value of 0.213) and F-Measure of 0.3478 (p-value of 0.12).

Table 5.8: Naïves Bayes results with discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric
Naïve Bayes	72.73%	0.4286	0.2143	0.2857	Gain Ratio
Naïve Bayes	**72.73%**	**0.4444**	**0.2857**	**0.3478**	**Gini Index**
Naïve Bayes	69.09%	0.3333	0.2143	0.2609	Information Gain

Table 5.9 contains the results obtained with the Random Forest algorithm, varying its parameter *numIterations* with the values 100, 500, 1000 and 10000. The best result was obtained with *numIterations* = 500, using *gain ratio* metric, presenting accuracy of 74.55% (p-value of 0.187), precision of 0.5 (p-value of 0.183), recall of 0.1429 (p-value of 0.385) and F-Measure of 0.2222 (p-value of 0.201).

Table 5.9: Random Forest results with discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric
Random Forest Iterations 500	**74.55%**	**0.5**	**0.1429**	**0.2222**	**Gain Ratio**
Random Forest Iterations 1000	69.09%	0.2857	0.1429	0.1905	Gini Index
Random Forest Iterations 100	63.64%	0.2	0.1429	0.1667	Information Gain

Table 5.10 contains the results obtained with the SMO algorithm using three kernel types (*PolyKernel, Puk and RBFKernel*) in its *kernel* parameter. The best result was obtained with *kernel* = Puk, using *gain ratio* metric, presenting accuracy of 74.55% (p-value of 0.489), precision of 0.5 (p-value of 0.381), recall of 0.0714 (p-value of 0.583) and F-Measure of 0.125 (p-value of 0.408).

Table 5.10: SMO results with discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric
SMO Puk	**74.55%**	**0.5**	**0.0714**	**0.125**	**Gain Ratio**
SMO Puk	74.55%	NaN	0	NaN	Gini Index
SMO Puk	74.55%	NaN	0	NaN	Information Gain

Although the supervised discretization technique increased the accuracy of the Naïve Bayes (increased from 67.27% to 72.73%) and Random Forest (increased from 70.91% to 74.55%) algorithms, it did not presented an accuracy better than that obtained by applying the IBk algorithm to the non-discretized unbalanced dataset. Thus, the attribute selection technique will be applied, aiming at increasing the classifiers performance.

5.2.3 Attribute Selection

In order to try to improve classifier accuracy, the attribute selection technique, explained in Section 2.5.1, was applied. This technique was applied after supervised discretization, commented in Section 5.2.2, attempting to improve the best results obtained with the discretization.

The *Ranker* search method of the WEKA software was used, along with the attribute evaluator *InfoGainAttributeEval*. This attribute evaluator uses the *information gain* metric in the evaluations, while the *Ranker* method is a single-attribute evaluator method, i.e., it evaluates the degree of correlation of each attribute to the class independently. Because it is single-attribute (independently evaluates each attribute), it can eliminate irrelevant but not redundant attributes (e.g., duplicate attributes). A good subset of attributes is the one where attributes are strongly correlated to the class and poorly correlated with each other.

Table 5.11 contains the ranking of all independent attributes of the dataset discretized with the *information gain* metric. This ranking, obtained through the *Ranker* method, was generated using the 5-Fold Cross-Validation method, which evaluates each attribute in each of the five folds.

Table 5.11: Attributes ranking for the *information gain* metric

Ranking	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
1	TDA	NeutralTDA	NeutralTDA	TDA	NR_Threat-Neutral
2	TDT	NR_Threat-Neutral	TDA	TDT	TDT
3	HR_Threat-Neutral	TDT	TDT	NeutralTDA	NeutralTDA
4	NeutralTDA	TDA	HR_Threat-Neutral	NR_Threat-Neutral	HR_Threat-Neutral
5	NR_Threat-Neutral	HR_Threat-Neutral	NR_Threat-Neutral	HR_Threat-Neutral	TDA
6	NeutralTDT	NeutralTDT	NeutralTDT	NeutralTDT	NeutralTDT
7	NR_TDT	NR_TDT	NR_TDA	NR_TDT	NR_TDA
8	NR_TDA	NR_TDA	NR_NeutralTDA	NR_TDA	NR_TDT
9	NR_NeutralTDA	NR_NeutralTDA	NR_TDT	NR_NeutralTDA	NR_NeutralTDA
10	NR_NeutralTDT	NR_NeutralTDT	NR_NeutralTDT	NR_NeutralTDT	NR_NeutralTDT

It is important to mention that the others two datasets discretized with *gain ratio* and *gini index* metrics also have their own ranking, obtained through the *Ranker* method. Although the attribute evaluator *InfoGainAttributeEval* applied by *Ranker* uses the *information gain* metric, it is not necessary to perform a supervised discretization using this metric. That means, a dataset can be discretized following any criteria or metric and the attribute evaluator *InfoGainAttributeEval* can still be applied to generate the ranking.

To determine which subset obtained the best performance among the others, classification analyses were performed using the same algorithms described in the beginning of this chapter. The analyses were performed as follows: using only the first ranking attribute, using only the first and second attributes, and so on until using the first nine attributes, because all analyses performed before, without attribute selection, used all attributes in dataset.

Table 5.12 contains the results obtained with the IBk algorithm (k-NN), using the same values of the supervised discretization of Table 5.6 for parameter k. Since the results with $k = 10$ and $k = 12$ presented the same measures values, $k = 12$ was selected as the best results, because it used a smaller subset of attributes (six attributes). The best result used the *information gain* metric, presenting accuracy of 74.55% (p-value of 0.924), precision of 0.5 (p-value of 0.091), recall of 0.0714 (p-value of 0.092) and F-Measure of 0.125 (p-value of 0.091).

Table 5.12: IBk results of attribute selection with discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Subset
IBk k = 9	74.55%	NaN	0	NaN	Gain Ratio	Subset 9
IBk k = 10	74.55%	0.5	0.0714	0.125	Gini Index	Subset 7
IBk k = 12	**74.55%**	**0.5**	**0.0714**	**0.125**	**Information Gain**	**Subset 6**

Table 5.13 contains the results obtained with the J48 algorithm, using the same values of the supervised discretization of Table 5.7 for parameter *confidenceFactor*. The best result was obtained with *confidenceFactor* = 0.25, using the *gain ratio* metric and a subset of nine attributes, presenting accuracy of 70.91% (p-value of 0.976), precision of 0.25 (p-value of 0.104), recall of 0.0714 (p-value of 0.109) and F-Measure of 0.1111 (p-value of 0.104).

Table 5.13: J48 results of attribute selection with discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Subset
J48 Confidence Factor 0.25	**70.91%**	**0.25**	**0.0714**	**0.1111**	**Gain Ratio**	**Subset 9**
J48 Confidence Factor 0.75	65.45%	0.2727	0.2143	0.24	Gini Index	Subset 2
J48 Confidence Factor 0.5	65.45%	0	0	NaN	Information Gain	Subset 1

Table 5.14 contains the results obtained with the Naïve Bayes algorithm. The best result was obtained using the *gini index* metric and using a subset of nine attributes, presenting accuracy of 74.55% (p-value of 0.063), precion of 0.5 (p-value of 0.063), recall of 0.3571 (p-value of 0.049) and F-Measure of 0.4167 (p-value of 0.022).

Table 5.14: Naïve Bayes results of attribute selection with discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Subset
Naïve Bayes	72.73%	0.4286	0.2143	0.2857	Gain Ratio	Subset 9
Naïve Bayes	**74.55%**	**0.5**	**0.3571**	**0.4167**	**Gini Index**	**Subset 9**
Naïve Bayes	72.73%	0.4444	0.2857	0.3478	Information Gain	Subset 9

Table 5.15 contains the results obtained with the Random Forest algorithm, using the same values of the supervised discretization of Table 5.9 for the parameter *numIterations*. The best result was obtained with *numIterations* = 500, using the *gain ratio* metric and a subset of three attributes, presenting accuracy of 74.55% (p-value of 0.231), precision of 0.6 (p-value of 0.231), recall of 0.2 (p-value of 0.275), and F-Measure of 0.3 (p-value of 0.115).

Table 5.15: Random Forest results of attribute selection with discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Subset
Random Forest Iterations 500	**74.55%**	**0.6**	**0.2**	**0.3**	**Gain Ratio**	**Subset 3**
Random Forest Iterations 1000	67.27%	0.3	0.2143	0.25	Gini Index	Subset 2
Random Forest Iterations 100	69.09%	0.3333	0.2143	0.2609	Information Gain	Subset 6

Table 5.16 contains the results obtained with the SMO algorithm using the same value of the supervised discretization of Table 5.10 for parameter *kernel*. The best result was obtained with *kernel* = Puk, using the *gain ratio* metric and a subset of six attributes, presenting accuracy of 74.55% (p-value of 0.323), precision of 0.5714 (p-value of 0.323), recall of 0.2667 (p-value of 0.004) and F-Measure of 0.3636 (p-value of 0.002).

Table 5.16: SMO results of attribute selection with discretized unbalanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Subset
SMO Puk	**74.55%**	**0.5714**	**0.2667**	**0.3636**	**Gain Ratio**	**Subset 6**
SMO Puk	74.55%	0.5	0.1429	0.2222	Gini Index	Subset 3
SMO Puk	74.55%	0.5	0.1429	0.2222	Information Gain	Subset 4

Although the attribute selection technique increased the accuracy of the Naïve Bayes algorithm (increased from 72.73% to 74.55%), it did not presented an accuracy better than that obtained by applying the IBk algorithm to the non-discretized unbalanced dataset. Thus, the class balancing technique will be applied, aiming at increasing the classifiers performance.

5.3 Balanced Dataset

As explained at the beginning of this chapter, the class imbalance problem strongly influences the classifier result. To address this problem, class balancing was performed using the SMOTE algorithm [10], which adds artificial instances of the minority class to the training fold, as explained in Section 2.5.3. The goal of this technique is to reduce the imbalance between the number of instances belonging to the majority and minority classes.

This section deals with the analyses performed using the class balancing technique, gradually increasing the number of instances belonging to the *high* PCL class (minority class). The minority class was increased by 25, 50, 75, 100 and 150%. Section 5.3.1 deals with balancing performed with the non-discretized dataset, while Section 5.3.2 deals with balancing performed after supervised discretization of the dataset attributes.

5.3.1 Non-Discretized Dataset

This subsection deals with the results obtained by applying class balancing to the non-discretized dataset. Table 5.17 contains the results obtained with the IBk algorithm, varying its k parameter with values from 1 to 15. Although the variations of $k = 1$, $k = 7$ and $k = 12$ presented the best accuracy, $k = 1$, increasing the minority class in 50% also presented the best F-Measure. So, the best result presented accuracy of 76.36% (p-value of 0.004), precision of 0.5238 (p-value of 0.004), recall of 0.7857 (p-value of 0.001) and F-Measure of 0.6286 (p-value of 0.001).

Table 5.17: IBk results with non-discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Increase
IBk k = 1	**76.36%**	**0.5238**	**0.7857**	**0.6286**	**50%**
IBk k = 2	61.82%	0.3793	0.7857	0.5116	50%
IBk k = 3	70.91%	0.4583	0.7857	0.5789	150%
IBk k = 4	74.55%	0.5	0.7143	0.5882	25%
IBk k = 5	74.55%	0.5	0.5714	0.5333	50%
IBk k = 6	72.73%	0.4762	0.7143	0.5714	25%
IBk k = 7	76.36%	0.5294	0.6429	0.5806	50%
IBk k = 8	74.55%	0.5	0.7143	0.5882	50%
IBk k = 9	74.55%	0.5	0.4286	0.4615	50%
IBk k = 10	74.55%	0.5	0.4286	0.4615	50%
IBk k = 11	74.55%	0.5	0.2857	0.3636	50%
IBk k = 12	76.36%	0.5385	0.5	0.5185	50%
IBk k = 13	72.73%	0.3333	0.0714	0.1176	25%
IBk k = 14	72.73%	0.4	0.1429	0.2105	25%
IBk k = 15	72.73%	0	0	NaN	25%

Table 5.18 contains the results obtained with the J48 algorithm, varying its parameters *confidenceFactor* and *unpruned*. The best result was obtained with *confidenceFactor* = 0.25, increasing the minority class in 25% and presenting accuracy of 70.91% (p-value of 0.237), precision of 0.4 (p-value of 0.119), recall of 0.2857 (p-value of 0.22) and F-Measure of 0.3333 (p-value of 0.107).

Table 5.18: J48 results with non-discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Increase
J48 Confidence Factor 0.25	**70.91%**	**0.4**	**0.2857**	**0.3333**	**25%**
J48 Confidence Factor 0.5	69.09%	0.4118	0.5	0.4516	75%
J48 Confidence Factor 0.75	69.09%	0.4118	0.5	0.4516	75%
J48 Unpruned	69.09%	0.4118	0.5	0.4516	75%

Table 5.19 contains the results obtained with the Naïve Bayes algorithm. The best

result was obtained increasing the minority class in 50%, presenting accuracy of 65.45% (p-value of 0.148), precision of 0.3684 (p-value of 0.065), recall of 0.5 (p-value of 0.132) and F-Measure of 0.4242 (p-value of 0.059).

Table 5.19: Naïves Bayes results with non-discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Increase
Naïve Bayes	**65.45%**	**0.3684**	**0.5**	**0.4242**	**50%**

Table 5.20 contains the results obtained with the Random Forest algorithm, varying its parameter *numIterations* with the values 100, 500, 1000 and 10000. Since the results with *numIterations* = 500 and *numIterations* = 1000 presented the same measures values with the same minority class increase, *numIterations* = 500 was selected as the best result, because less iterations were needed. So, this best result presented accuracy of 74.55% (p-value of 0.008), precision of 0.5 (p-value of 0.008), recall of 0.4286 (p-value of 0.228) and F-Measure of 0.4615 (p-value of 0.026).

Table 5.20: Random Forest results with non-discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Increase
Random Forest Iterations 100	74.55%	0.5	0.3571	0.4167	100%
Random Forest Iterations 500	**74.55%**	**0.5**	**0.4286**	**0.4615**	**100%**
Random Forest Iterations 1000	74.55%	0.5	0.4286	0.4615	100%
Random Forest Iterations 10000	72.73%	0.4545	0.3571	0.4	75%

Table 5.21 contains the results obtained with the SMO algorithm using three kernel types (*PolyKernel, Puk and RBFKernel*) in its *kernel* parameter. The best result was obtained with *kernel* = Puk, increasing the minority class in 100% and presenting accuracy of 81.82% (p-value of 0.001), precision of 0.75 (p-value of 0.037), recall of 0.4286 (p-value of 0.002) and F-Measure of 0.5455 (p-value of 0.001).

Table 5.21: SMO results with non-discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Increase
SMO PolyKernel	74.55%	0.5	0.2143	0.3	75%
SMO Puk	**81.82%**	**0.75**	**0.4286**	**0.5455**	**100%**
SMO RBFKernel	74.55%	NaN	0	NaN	25%

As the results showed, the class balancing technique increased the accuracy of Random Forest (increased from 70.91% to 74.55%) and SMO (increased from 74.55% to 81.82%) algorithm. Increasing the minority class in non-discretized dataset by 100% resulted in the highest accuracy obtained so far. The supervised discretization technique will be applied to the balanced dataset, aiming at increasing the classifiers performance.

5.3.2 Discretized Dataset

This subsection deals with the results obtained by applying class balancing to the discretized dataset. Table 5.22 contains the results obtained with the IBk algorithm, varying its k parameter with values from 1 to 15. The best result was obtained with $k = 15$, using the *gini index* metric and increasing the minority class in 50%. It presented accuracy of 74.55% (p-value of 0.43), precision of 0.5 (p-value of 0.223), recall of 0.1429 (p-value of 0.085) and F-Measure of 0.2222 (p-value of 0.059).

Table 5.22: IBk results with discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Increase
IBk k = 4	74.55%	0.5	0.0714	0.125	Gain Ratio	25%
IBk k = 15	**74.55%**	**0.5**	**0.1429**	**0.2222**	**Gini Index**	**50%**
IBk k = 9	74.55%	0.5	0.0714	0.125	Information Gain	25%

Table 5.23 contains the results obtained with the J48 algorithm, varying its parameters *confidenceFactor* and *unpruned*. The best result was obtained with *confidenceFactor* = 0.5, using the *gain ratio* metric and increasing the minority class in 75%. It presented accuracy of 70.91% (p-value of 0.51), precision of 0.4 (p-value of 0.232), recall of 0.2857 (p-value of 0.072) and F-Measure of 0.3333 (p-value of 0.052).

Table 5.23: J48 results with discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Increase
J48 Confidence Factor 0.5	**70.91%**	**0.4**	**0.2857**	**0.3333**	**Gain Ratio**	**75%**
J48 Confidence Factor 0.25	65.45%	0.2222	0.1429	0.1739	Gini Index	50%
J48 Confidence Factor 0.25	67.27%	0.25	0.1429	0.1818	Information Gain	50%

Table 5.24 contains the results obtained with the Naïve Bayes algorithm. The best result was obtained with the *gini index* metric and increasing the minority class in 75%. It presented accuracy of 74.55% (p-value of 0.005), precision of 0.5 (p-value of 0.005), recall of 0.3571 (p-value of 0.487) and F-Measure of 0.4167 (p-value of 0.083).

Table 5.24: Naïve Bayes results with discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Increase
Naïve Bayes	72.73%	0.4545	0.3571	0.4	Gain Ratio	100%
Naïve Bayes	**74.55%**	**0.5**	**0.3571**	**0.4167**	**Gini Index**	**75%**
Naïve Bayes	69.09%	0.3333	0.2143	0.2609	Information Gain	150%

Table 5.25 contains the results obtained with the Random Forest algorithm, varying its parameter *numIterations* with the values 100, 500, 1000 and 10000. The best result was obtained with *numIterations* = 500, using the *gain ratio* metric and increasing the

minority class in 50%. It presented accuracy of 74.55% (p-value of 0.081), precision of 0.5 (p-value of 0.081), recall of 0.2143 (p-value of 0.35) and F-Measure of 0.3 (p-value of 0.146).

Table 5.25: Random Forest results with discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Increase
Random Forest Iterations 500	**74.55%**	**0.5**	**0.2143**	**0.3**	**Gain Ratio**	**50%**
Random Forest Iterations 100	69.09%	0.2857	0.1429	0.1905	Gini Index	25%
Random Forest Iterations 100	61.82%	0.1818	0.1429	0.16	Information Gain	100%

Table 5.26 contains the results obtained with the SMO algorithm using three kernel types (*PolyKernel, Puk and RBFKernel*) in its *kernel* parameter. The best result was obtained with *kernel* = RBFKernel, using the *information gain* metric and increasing the minority class in 150%. It presented accuracy of 74.55% (p-value of 0.649), precision of 0.5 (p-value of 0.052), recall of 0.0714 (p-value of 0.289) and F-Measure of 0.125 (p-value of 0.121).

Table 5.26: SMO results with discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Increase
SMO RBFKernel	74.55%	NaN	0	NaN	Gain Ratio	150%
SMO Puk	74.55%	NaN	0	NaN	Gini Index	75%
SMO RBFKernel	**74.55%**	**0.5**	**0.0714**	**0.125**	**Information Gain**	**150%**

Although the supervised discretization technique increased the accuracy of the Naïve Bayes algorithm (increased from 72.73% to 74.55%), it did not presented an accuracy better than that obtained by applying the SMO algorithm to the non-discretized balanced dataset. Thus, the attribute selection technique will be applied, aiming at increasing the classifiers performance.

5.3.3 Attribute Selection

This subsection deals with the attribute selection applied to the discretized balanced datasets, in order to try to improve the classifier accuracy. This time, the attribute selection was applied after the class balancing of the discretized datasets. This differs from Section 5.2.3 because here each minority class increase will result in new rankings of attributes. The *Ranker* search method of WEKA, along with its attribute evaluator *InfoGainAttributeEval*, was also used in the analyses discussed in this section. After each gradual increase of the minority class, the *Ranker* was applied, generating a ranking of attributes for each fold. Because five gradual increases (i.e., 25, 50, 75, 100 and 150%) were performed, five rankings were generated for each fold.

Table 5.27 contains the results obtained with the IBk algorithm (k-NN), using the same values of the supervised discretization of Table 5.22 for parameter k. The best result was obtained with $k = 15$, using the *gini index* metric, increasing the minority class in 50% and using a subset of five attributes. It presented accuracy of 76.36% (p-value of 0.525), precision of 0.6667 (p-value of 0.151), recall of 0.1429 (p-value of 0.021) and F-Measure of 0.2353 (p-value of 0.018).

Table 5.27: IBk results of attribute selection with discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Increase	Subset
IBk k = 4	74.55%	0.5	0.0714	0.125	Gain Ratio	25%	Subset 9
IBk k = 15	**76.36%**	**0.6667**	**0.1429**	**0.2353**	**Gini Index**	**50%**	**Subset 5**
IBk k = 9	74.55%	0.5	0.0714	0.125	Information Gain	25%	Subset 9

Table 5.28 contains the results obtained with the J48 algorithm, using the same values of the supervised discretization of Table 5.23 for parameter *confidenceFactor*. The best result was obtained with *confidenceFactor* = 0.5, using the *gain ratio* metric, increasing the minority class in 75% and using a subset of one attribute. It presented accuracy of 70.91% (p-value of 0.945), precision of 0.3333 (p-value of 0.159), recall of 0.1429 (p-value of 0.029) and F-Measure of 0.2 (p-value of 0.023).

Table 5.28: J48 results of attribute selection with discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Increase	Subset
J48 Confidence Factor 0.5	**70.91%**	**0.3333**	**0.1429**	**0.2**	**Gain Ratio**	**75%**	**Subset 1**
J48 Confidence Factor 0.25	65.45%	0.2222	0.1429	0.1739	Gini Index	50%	Subset 6
J48 Confidence Factor 0.25	61.82%	0.1818	0.1429	0.16	Information Gain	50%	Subset 6

Table 5.29 contains the results obtained with the Naïve Bayes algorithm. The best result was obtained using the *gini index* metric, increasing the minority class in 75% and using a subset of eight attributes. It presented accuracy of 76.36% (p-value of 0.003), precision of 0.5556 (p-value of 0.002), recall of 0.3571 (p-value of 0.409) and F-Measure of 0.4348 (p-value of 0.05).

Table 5.29: Naïve Bayes results of attribute selection with discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Increase	Subset
Naïve Bayes	74.55%	0.5	0.2857	0.3636	Gain Ratio	100%	Subset 5
Naïve Bayes	**76.36%**	**0.5556**	**0.3571**	**0.4348**	**Gini Index**	**75%**	**Subset 8**
Naïve Bayes	70.91%	0.4	0.2857	0.3333	Information Gain	150%	Subset 8

Table 5.30 contains the results obtained with the Random Forest algorithm, using the same values of the supervised discretization of Table 5.25 for the parameter *numIterations*. The best result was obtained with *numIterations* = 500, using the *gain ratio* metric,

increasing the minority class in 50% and using a subset of five attributes. It presented accuracy of 74.55% (p-value of 0.201), precision of 0.5 (p-value of 0.199), recall of 0.2143 (p-value of 0.298) and F-Measure of 0.3 (p-value of 0.159).

Table 5.30: Random Forest results of attribute selection with discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Increase	Subset
Random Forest Iterations 500	**74.55%**	**0.5**	**0.2143**	**0.3**	**Gain Ratio**	**50%**	**Subset 5**
Random Forest Iterations 100	69.09%	0.2857	0.1429	0.1905	Gini Index	25%	Subset 9
Random Forest Iterations 100	63.64%	0.25	0.2143	0.2308	Information Gain	100%	Subset 5

Table 5.31 contains the results obtained with the SMO algorithm using the same value of the supervised discretization of Table 5.26 for parameter *kernel*. Since the results with *kernel* = Puk and *kernel* = RBFKernel presented the same measures values, *kernel* = Puk was selected as the best result because it used a smaller increase (i.e., 75% instead of 150% increase). So, the best result used the *gini index* metric, increasing the minority class in 75% and using a subset of five attributes. It presented accuracy of 74.55% (p-value of 0.17), precision of 0.5 (p-value of 0.148), recall of 0.0714 (p-value of 0.538) and F-Measure of 0.125 (p-value of 0.224).

Table 5.31: SMO results of attribute selection with discretized balanced dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Metric	Increase	Subset
SMO RBFKernel	74.55%	NaN	0	NaN	Gain Ratio	150%	Subset 1
SMO Puk	**74.55%**	**0.5**	**0.0714**	**0.125**	**Gini Index**	**75%**	**Subset 5**
SMO RBFKernel	74.55%	0.5	0.0714	0.125	Information Gain	150%	Subset 5

Although the supervised discretization technique increased the accuracy of the IBk (increased from 74.55% to 76.36%) and Naïve Bayes algorithms (increased from 74.55% to 76.36%), it did not presented an accuracy better than that obtained by applying the SMO algorithm to the non-discretized balanced dataset.

5.4 Auto-WEKA

WEKA software, besides being a free and open-source software, has several plugins and algorithms that can be installed through its package manager. One such plugin is *Auto-WEKA* (version 2.6.1) [32, 55]. This plugin aims at assisting the user in choosing a classification algorithm (classifier) and defining its parameters. The user tells the plugin the measure he wants to maximize and also the execution time (the *timeLimit* hyperparameter) at which the plugin must execute to find the best recommended algorithm.

The Auto-WEKA plugin tries all WEKA algorithms, varying their parameter values.

At the end of its execution (defined by the user in the *timeLimit* parameter), it returns the algorithm that obtained the best value for the given measure, along with the parameter values used. One of the limitations of this plugin is that it does not allow the user to configure the validation method used. That is, the user is required to supply the entire dataset to the plugin and it uses internally a 10-Fold Cross-Validation.

When running the Auto-WEKA plugin for 1440 minutes (one day) with the original dataset, to maximize the accuracy measure, the suggested algorithm was SMO, with the following parameter values:

- parameter $C = 1.235172507048868$. Its *default* value is 1;
- parameter *buildCalibrationModels* = *true*. Its *default* value is *false*. According to WEKA documentation, this parameter defines whether to fit calibration models to the SMO outputs for proper probability estimates;
- parameter *kernel* = Puk;
- *kernel* parameter *sigma* (S) = 1.0698699932037348. Its *default* value is 1;
- *kernel* parameter *omega* (O) = 0.8329205951746678. Its *default* value is 1.

The Auto-WEKA plugin suggestion was applied to each analysis performed so far. The goal was to evaluate if the plugin could obtain better results, improving the accuracy measure. Table 5.32 contains the results obtained applying the Auto-WEKA suggested algorithm with its parameters values to each analysis performed previously.

Table 5.32: Results of Auto-WEKA suggested algorithm applied to each analysis

Accuracy	Precision	Recall	F-Measure	Increase	Metric	Subset
78.18%	0.5833	0.5	0.5385	Unbalanced	Non-discretized	All attributes
85.45%	**0.8**	**0.5714**	**0.6667**	**100%**	**Non-discretized**	**All attributes**
74.55%	NaN	0	NaN	Unbalanced	Information Gain	All attributes
72.73%	0	0	NaN	150%	Gini Index	All attributes
74.55%	NaN	0	NaN	Unbalanced	Gini Index	Subset 8
76.36%	0.6667	0.1429	0.2353	75%	Gini Index	Subset 7

The best result was obtained using the non-discretized dataset, performing a class balancing inceasing the minority class in 100% and using all attributes (i.e., it did not perform the attribute selection). It presented accuracy of 85.45% (p-value of 0.001), precision of 0.8 (p-value of 0.001), recall of 0.5714 (p-value of 0.271) and F-Measure of 0.6667 (p-value of 0.001). Thus, using the Auto-WEKA plugin presented the best result of all analyses performed with the value 36 as cut-off point.

5.5 Using a Higher Cut-off Point

As explained in Section 2.2, there are studies that suggest cut-off points to meet PTSD classification or identification criteria for provisional diagnoses [5]. Lower cut-off points are indicated when whishing to maximize detection of possible PTSD cases. On the other hand, higher cut-offs are indicated to minimize false positives. Table 2.2 showed suggested ranges of cut-off points for each criterion.

The classification analyses discussed so far used value 36 as PCL cut-off point. Because the selected range goes from 36 to 44, an evaluation of the range's maximum value (44) was also considered interesting. Since the number of dataset instances is small and the number of majority class instances is almost three times bigger than the minority class, using a higher cut-off will increase the dataset imbalance.

Another important point to notice is that the cut-off point value 44 is already accepted and used by the literature. Some studies [5, 47, 57, 22] propose and use value 44. On the other hand, value 36 has not been widely explored yet, suggesting another gap to investigate.

When using value 44 as cut-off point for PCL, the number of majority class instances (class *low*) is 47, while the number of minority class decreased to 8 (in this scenario, the number of majority class instances is almost six times bigger than the minority class). For this new cut-off point, the baseline increased to 85.45%. Like the previous classification analyses performed with value 36, this section summarizes the best result of each analysis obtained with value 44. The same five algorithms (IBk, J48, NB, RF and SMO) with their parameters variations used so far were also applied in the analysys of this section. The 5-Fold Stratified Cross-Validation method and the permutation test performing 1000 iterations were also used.

Table 5.33 contains the best results of each analysis performed using the non-discretized dataset. Since algorithms *IBk* with $k = 5$ and SMO with class balancing and *kernel* = Puk presented the same measures values, the *IBk* algorithm with $k = 5$ was selected as the best result, because it used the unbalanced dataset insted of increasing the minority class in 100% like the SMO algorithm. So, the best result with $k = 5$ presented accuracy of 89.09% (p-value of 0.001), precision of 1 (p-value of 0.048), recall of 0.25 (p-value of 0.009) and F-Measure of 0.4 (p-value of 0.001).

Table 5.33: Best results obtained with the cut-off point 44 using the non-discretized dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Increase
IBk k = 5	**89.09%**	**1**	**0.25**	**0.4**	**Unbalanced**
IBk k = 11	85.45%	NaN	0	NaN	50%
J48 Confidence Factor 0.25	85.45%	NaN	0	NaN	Unbalanced
J48 Confidence Factor 0.25	83.64%	0.3333	0.125	0.1818	50%
Naïve Bayes	70.91%	0.1	0.125	0.1111	Unbalanced
Naïve Bayes	72.73%	0.2308	0.375	0.2857	100%
Random Forest Iterations 100	85.45%	NaN	0	NaN	Unbalanced
Random Forest Iterations 500	85.45%	0.5	0.125	0.2	50%
SMO Puk	85.45%	NaN	0	NaN	Unbalanced
SMO Puk	89.09%	1	0.25	0.4	100%

Like the analyses perfomed with cut-off point 36, the supervised discretization and the attribute selection were applied to cut-off point 44, obtaining new rankings for each metric (*information gain*, *gain ratio* and *gini index*). Table 5.34 contains the best results of each analysis performed using the discretized datasets, obtained after applying the brute force supervised discretization implementation with the *information gain*, *gain ratio* and *gini index* metrics.

The best result was obtained with *IBk* with $k = 14$ using the discretized dataset by the *gain ratio* metric, increasing the minority class in 50% and using a subset of five attributes. It presented accuracy of 87.27% (p-value of 0.001), precision of 1 (p-value of 0.001), recall of 0.125 (p-value of 0.001) and F-Measure of 0.2222 (p-value of 0.001).

Table 5.34: Best results obtained with cut-off point 44 using the discretized dataset

Algorithm	Accuracy	Precision	Recall	F-Measure	Increase	Metric	Subset
IBk k = 4	85.45%	NaN	0	NaN	Unbalanced	Gain Ratio	Subset 1
IBk k = 14	**87.27%**	**1**	**0.125**	**0.2222**	**50%**	**Gain Ratio**	**Subset 5**
J48 Confidence Factor 0.25	85.45%	NaN	0	NaN	Unbalanced	Gain Ratio	Subset 1
J48 Confidence Factor 0.25	85.45%	NaN	0	NaN	50%	Gain Ratio	Subset 1
Naive Bayes	83.64%	0.3333	0.125	0.1818	Unbalanced	Gain Ratio	Subset 6
Naïve Bayes	81.82%	0.3333	0.25	0.2857	50%	Gain Ratio	Subset 8
Random Forest Iterations 100	81.82%	0	0	NaN	Unbalanced	Gain Ratio	Subset 3
Random Forest Iterations 100	81.82%	0	0	NaN	50%	Gain Ratio	Subset 1
SMO RBFKernel	85.45%	NaN	0	NaN	Unbalanced	Gain Ratio	Subset 1
SMO RBFKernel	85.45%	NaN	0	NaN	50%	Gain Ratio	Subset 1
IBk k = 5	85.45%	NaN	0	NaN	Unbalanced	Gini Index	Subset 1
IBk k = 8	85.45%	0.5	0.125	0.2	50%	Gini Index	All attributes
J48 Confidence Factor 0.25	85.45%	NaN	0	NaN	Unbalanced	Gini Index	Subset 1
J48 Confidence Factor 0.25	80%	0	0	NaN	50%	Gini Index	Subset 1
Naive Bayes	85.45%	0.5	0.125	0.2	Unbalanced	Gini Index	Subset 2
Naïves Bayes	83.63%	0.3333	0.125	0.1818	50%	Gini Index	Subset 5
Random Forest Iterations 100	85.45%	NaN	0	NaN	Unbalanced	Gini Index	Subset 6
Random Forest Iterations 100	81.82%	0	0	NaN	50%	Gini Index	All attributes
SMO RBFKernel	85.45%	NaN	0	NaN	Unbalanced	Gini Index	Subset 1
SMO RBFKernel	85.45%	NaN	0	NaN	50%	Gini Index	Subset 1
IBk k = 11	85.45%	NaN	0	NaN	Unbalanced	Information Gain	Subset 1
IBk k = 14	85.45%	NaN	0	NaN	50%	Information Gain	Subset 1
J48 Confidence Factor 0.25	85.45%	NaN	0	NaN	Unbalanced	Information Gain	Subset 1
J48 Confidence Factor 0.25	81.82%	0	0	NaN	50%	Information Gain	Subset 1
Naive Bayes	85.45%	0.5	0.125	0.2	Unbalanced	Information Gain	Subset 3
Naïve Bayes	81.82%	0.3333	0.25	0.2857	50%	Information Gain	Subset 1
Random Forest Iterations 100	83.64%	0	0	NaN	Unbalanced	Information Gain	Subset 5
Random Forest Iterations 500	81.82%	0.3333	0.25	0.2857	50%	Information Gain	Subset 1
SMO RBFKernel	85.45%	NaN	0	NaN	Unbalanced	Information Gain	Subset 1
SMO RBFKernel	85.45%	NaN	0	NaN	50%	Information Gain	Subset 1

The Auto-WEKA plugin was also applied to the non-discretized unbalanced dataset (i.e., original dataset), running for 1440 minutes (one day), to maximize the accuracy measure. The suggested algorithm was IBk, with the following parameter values:

- parameter $k = 14$ neighbors;
- parameter *distanceWeighting* = weight by $1/distance$. Its *default* value is *no distance weighting*. The Auto-WEKA suggested parameter value weights neighbors by the inverse of their distance;
- parameter *crossValidate* = *true*. Its *default* value is *false*. This parameter defines whether hold-one-out cross-validation will be used to select the best k value between 1 and the value specified as the k parameter.

The Auto-WEKA plugin suggestion was applied to each analysis performed. The goal was to evaluate if the plugin could obtain better results, improving the accuracy measure. Table 5.35 contains the results obtained applying the Auto-WEKA suggested algorithm with its parameters values to each analysis performed with cut-off point 44. The best result

was obtained with the original dataset (i.e., non-discretized and unbalanced dataset), not performing the attribute selection (i.e., using all the ten independent attributes). It presented accuracy of 90.91% (p-value of 0.001), precision of 0.8 (p-value of 0.02), recall of 0.5 (p-value of 0.001) and F-Measure of 0.6154 (p-value of 0.001).

Table 5.35: Results of Auto-WEKA suggested algorithm applied to each analysis with cut-off point 44

Accuracy	Precision	Recall	F-Measure	Increase	Metric	Subset
90.91%	**0.8**	**0.5**	**0.6154**	**Unbalanced**	**Non-discretized**	**All attributes**
81.82%	0.3333	0.25	0.2857	50%	Non-discretized	All attributes
81.82%	0	0	NaN	Unbalanced	Gini Index	All attributes
78.18%	0.1667	0.125	0.1429	50%	Gini Index	All attributes
81.82%	0.25	0.125	0.1667	Unbalanced	Gini Index	Subset 2
81.82%	0.25	0.125	0.1667	50%	Gini Index	Subset 2

5.6 Final Remarks

The supervised discretization, class balancing and attribute selection techniques applied improved the accuracy of some classifiers. When comparing the unbalanced datasets, the supervised discretization improved the accuracy of Naïve Bayes and Random Forest with *numIterations* = 500. When comparing the attribute selection for unbalanced discretized dataset, there was an improve of Naïve Bayes accuracy, while the accuracies of IBk with $k = 9$, J48 with *confidenceFactor* = 0.25, Random Forest with *numIterations* = 500 and SMO with *kernel* = Puk remained the same.

When comparing the use of class balancing in non-discretized datasets, there was an increase in accuracy of Random Forest with *numIterations* = 500 and SMO with *kernel* = Puk. When comparing the use of class balancing in discretized datasets, there was an increase in accuracy of Naïve Bayes.

When comparing the attribute selection performed with the balanced discretized datasets, there was an increase in accuracy of IBk with $k = 15$ and Naïve Bayes, while accuracy of J48 with *confidenceFactor* = 0.5, Random Forest with *numIterations* = 500 and SMO with *kernel* = Puk remained the same.

The best result with cut-off point value 36 was obtained with the SMO algorithm with *kernel* = Puk using the non-discretized dataset and increasing the minority class in 100%. It correctly classified six of 14 minority class instances and 39 of 41 majority class instances, presenting accuracy of 81.82% (p-value of 0.001), precision of 0.75 (p-value of

0.037), recall of 0.4286 (p-value of 0.002) and F-Measure of 0.5455 (p-value of 0.001). Table 5.36 contains its confusion matrix.

Table 5.36: Confusion matrix of the best result using the cut-off point 36

		Predicted	
		High	Low
Actual	High	6	8
	Low	2	39

The SMO classifier, along with the parameter values suggested by the Auto-WEKA plugin, showed better results than the other classifiers used. As explained in Section 5.4, due to Auto-WEKA plugin limitations (i.e., it is not possible to configure the validation method used and to provide separate training and test partitions), the plugin was executed only with the original dataset and its suggestion (the SMO classifier with its parameters values) was maintained and used when the dataset was discretized and balanced.

The best result with the Auto-WEKA suggestion correctly classified eight of 14 minority class instances and 39 of 41 majority class instances, presenting accuracy of 85.45% (p-value of 0.001), precision of 0.8 (p-value of 0.001), recall of 0.5714 (p-value of 0.271) and F-Measure of 0.6667 (p-value of 0.001), using the non-discretized dataset, without appling the attribute selection technique, and increasing the minority class in 100%. Table 5.37 contains its confusion matrix.

Table 5.37: Confusion matrix of the best result using the Auto-WEKA suggestion and the cut-off point 36

		Predicted	
		High	Low
Actual	High	8	6
	Low	2	39

Another factor that influenced the results of the analyses in this chapter was the small number of dataset instances and their class distribution, that is, the dataset imbalance. As much as the SMOTE algorithm tries to minimize the imbalance problem by inserting artificial instances, it is ideal to balance the dataset by inserting real (not artificial) instances (e.g., running the Biomedical Institute experiment again with new individuals). However, this is not always feasible or capable of being accomplished over a small period of time.

An article containing the results obtained in the classification analyses performed in this chapter, using the value 36 as cut-off point and explaining the techniques applied was

published [15].

Aiming at exploring even further the PCL scale, the classification analyses were performed using a higher cut-off point value, following the suggested range found in the literature. The value selected was 44 because it was the maximum value of the range and it would prove more challenging, since the class imbalance would increase, as explained in Section 5.5.

Although the best results obtained with cut-off point value 44 presented higher accuracy then those best results obtained with cut-off point value 36, it is important to observe that the number of minority class instances has also decreased, increasing the baseline for this new scenario (i.e., the baseline was 74.55% and it increased to 85.45%).

The best result obtained with the non-discretized dataset and performing no class balancing presented accuracy of 89.09% (p-value of 0.001), precision of 1 (p-value of 0.048), recall of 0.25 (p-value of 0.009), F-Measure of 0.4 (p-value of 0.001), appling the IBk $k = 5$ algorithm. It correctly classified only two of eight minority class instances and all majority class instances. Table 5.38 contains its confusion matrix.

Table 5.38: Confusion matrix of the best result using the non-discretized unbalanced dataset and the cut-off point 44

		Predicted	
		High	Low
Actual	High	2	6
	Low	0	47

The best result obtained with the discretized dataset, using the *gain ratio* metric and increasing the minority class by 50%, presented accuracy of 87.27% (p-value of 0.001), precision of 1 (p-value of 0.001), recall of 0.125 (p-value of 0.001) and F-Measure of 0.2222 (p-value of 0.001). It applied the IBk algorithm with $k = 14$, using the first five attributes of the ranking. Even performing a class balancing, i.e., increasing the minority class by 50%, the classifier could correctly classify only one minority class instance and all majority class instances. Table 5.39 contains its confusion matrix.

Table 5.39: Confusion matrix of the best result using the discretized balanced dataset and the cut-off point 44

		Predicted	
		High	Low
Actual	High	1	7
	Low	0	47

The IBk classifier, along with the parameter values suggested by the Auto-WEKA plugin, showed better results than the other classifiers used. It correctly classified four of eight minority class instances and 46 of 47 majority class instances, presenting accuracy of 90.91% (p-value of 0.001), precision of 0.8 (p-value of 0.02), recall of 0.5 (p-value of 0.001) and F-Measure of 0.6154 (p-value of 0.001), using the unbalanced non-discretized dataset, without appling the attribute selection technique. Table 5.40 contains its confusion matrix.

Table 5.40: Confusion matrix of the best result using the Auto-WEKA suggestion and the cut-off point 44

		Predicted	
		High	Low
Actual	High	4	4
	Low	1	46

As already explained, although cut-off point value 44 increases the baseline to 85.45%, it becomes more challenging, since the number of minority class instances is not big. A new execution of the Biomedical Institute experiment with new individuals would be very interesting and helpful to minimize, and perhaps, solve this problem.

Table 5.41 summarizes the best results obtained with the cut-off points 36 and 44, applying the class balancing, supervised discretization and attribute selection techniques. As already explained, when changing the cut-off point value, the number of minority class instances and the baseline also change. Thus, the measures precision, recall and F-Measure must be taken into account, when comparing the results. Since the goal is to identify individuals with higher PCL scores, indicating possible PTSD patients, the correct classification of minority class instances is more relevant to the addressed problem and was better performed using the value 36 as cut-off point.

Table 5.41: Best results using the cut-off points 36 and 44

Algorithm	Accuracy	Precision	Recall	F-Measure	Increase	Metric	Subset	Cut-off
IBk k = 4	76,36%	0,5294	0,6429	0,5806	Unbalanced	Non-Discretized	All Attributes	36
IBk k = 5	89,09%	1	0,25	0,4	Unbalanced	Non-Discretized	All Attributes	44
Naïve Bayes	74,55%	0,5	0,3571	0,4167	Unbalanced	Gini Index	Subset 9	36
Naïve Bayes	85,45%	0,5	0,125	0,2	Unbalanced	Gini Index	Subset 2	44
SMO Puk	81,82%	0,75	0,4286	0,5455	100%	Non-Discretized	All Attributes	36
SMO Puk	89,09%	1	0,25	0,4	100%	Non-Discretized	All Attributes	44
Naïve Bayes	76,36%	0,5556	0,3571	0,4348	75%	Gini Index	Subset 8	36
IBk k = 14	87,27%	1	0,125	0,2222	50%	Gain Ratio	Subset 5	44

Chapter 6

Regression

This chapter discusses the regression analyses performed, presenting the results obtained. In the regression analyses, the original dataset, i.e., the dataset with ten independent numeric attributes and the PCL attribute was used. Unlike the classification analyses of Chapter 5, where the PCL attribute has nominal values (*high* and *low* classes), for regression analyses, this attribute has the numerical values of the PCL scale.

In addition to the WEKA software [63] used for classification analyses, regression analyses used PRoNTo (version 3) [49] software. This software was initially developed to perform DM tasks on neuroimaging data (e.g., fMRI). However, in its latest version (version 3) it is already possible to apply its DM algorithms to flat file data (e.g., csv data format).

This chapter has four sections. Section 6.1 explains the methodology used. Sections 6.2 and 6.3 deal with regression analyses and their results obtained with WEKA and PRoNTo software respectively. Section 6.4 discusses the main points observed with the analyses.

6.1 Methodology

The regression analyses were performed using two softwares: WEKA and PRoNTo. Each one of these softwares has its own regression algorithms implementations. To validate the results obtained, the 5-Fold Stratified Cross-Validation method along with the implementation of the permutation test (performing 1000 permutations) were used. Since there is no class attribute in the regression, the supervised discretization, attribute selection and class balancing DM techniques were not applied to the dataset used.

6.2 WEKA

This section deals with the regression analyses performed with the WEKA software [63]. As briefly explained in Section 2.7, the three following WEKA algorithms were used:

1. IBk: this is the implementation of the k-NN algorithm in WEKA. As explained in Section 2.6.5, this algorithm can be applied to both classification analysis and regression analysis [1];
2. Linear Regression;
3. SMOreg: this is the implementation of the SVM algorithm for regression in WEKA [53, 54].

The SVM algorithm, as explained in the Chapter 2, is also widely used for regression. Because it is widely used, algorithms such as KRR and RVR were implemented in PRoNTo as adaptations of the SVM algorithm. Like the SVM, which can be used in both classification and regression analyses, the IBk algorithm was also selected for the regression analyses in this section. The Linear Regression algorithm, also widely used for regression, is an algorithm whose output (mathematical function) is simple to understand.

The analyses were performed using the 5-Fold Stratified Cross-Validation method, obtaining the results for the correlation coefficient (R), MAE and RMSE metrics. To evaluate the statistical significance of the obtained results, the permutation test algorithm implemented for WEKA software was applied, performing 1000 permutations and using $\alpha = 0.05$.

The IBk algorithm has the k parameter, which refers to the number of training instances, searched in a pattern space, that are closest to the test instance (see Section 2.6.5). In the analyses performed, the k parameter was varied with values from 1 to 15. Among the values used for the k parameter, the one with the highest correlation coefficient and the lowest mean absolute error and root mean squared error was selected.

Table 6.1 contains the results obtained with the three algorithms (IBk, Linear Regression and SMOreg). As can be seen, the best result was obtained with the IBk k = 4 algorithm, with a correlation coefficient (R) of 0.4164 and a p-value equal to 0.001.

Table 6.1: Regression analysis results in WEKA

Algorithm	R	R p-value	MAE	MAE p-value	RMSE	RMSE p-value
IBk k = 1	0.3808	0.005	10.2364	0.021	14.2184	0.023
IBk k = 2	0.4031	0.001	9.6636	0.029	12.583	0.014
IBk k = 3	0.4055	0.001	9.1394	0.012	11.9894	0.004
IBk k = 4	**0.4164**	**0.001**	**8.9636**	**0.013**	**11.5186**	**0.001**
IBk k = 5	0.3874	0.001	8.8691	0.007	11.6159	0.001
Ibk k = 6	0.3232	0.005	9.3455	0.05	11.9413	0.006
IBk k = 7	0.2616	0.025	9.561	0.103	12.1385	0.021
IBk k = 8	0.1513	0.095	9.7795	0.189	12.5402	0.077
IBk k = 9	0.014	0.307	10.1636	0.445	12.9528	0.253
IBk k = 10	-0.0249	0.384	10.1782	0.493	12.9678	0.303
IBk k = 11	-0.1288	0.613	10.4149	0.728	13.1805	0.514
IBk k = 12	-0.1187	0.585	10.3697	0.712	13.1396	0.524
IBk k = 13	-0.0739	0.459	10.1469	0.58	12.9559	0.381
IBk k = 14	-0.0309	0.355	9.9351	0.393	12.8605	0.322
IBk k = 15	0.0096	0.258	9.8848	0.373	12.7532	0.242
LR	0.0396	0.242	10.2669	0.375	13.4548	0.436
SMOreg	0.3318	0.015	8.442	0.005	12.0681	0.013

6.3 PRoNTo

This section deals with regression analyses performed with PRoNTo software [49]. PRoNTo is also a free and open-source software, having four algorithms for regression analysis. Currently, in its third version, PRoNTo does not have algorithms for preprocessing and its algorithms do not have user-defined parameters. As briefly explained in Section 2.7, the following PRoNTo algorithms were used:

1. Gaussian Process Regression (GPR) [61, 50, 62];
2. Kernel Ridge Regression (KRR) [60];
3. Relevance Vector Regression (RVR) [56];
4. epsilon-Support Vector Regression (epsilon-SVR): this is the implementation of the SVM algorithm for PRoNTo regression, using epsilon to define the margin of tolerance, where penalties are applied to errors.

All the regression algorithms of PRoNTo were selected for the analyses of this section. As explained in Section 6.2, the KRR and RVR algorithms are adaptations of the widely used SVM algorithm, while the epsilon-SVR algorithm is the SVM algorithm implementation, using the ϵ-insensitive hinge loss function (see Section 2.6.4). Like the IBk algorithm,

the GPR uses the lazy learning approach. The GPR also uses a Bayesian approach to calculate the probability distribution of the calculated mapping functions.

The analyses were performed using the 5-Fold Stratified Cross-Validation method, obtaining the results for the correlation coefficient (R) and MSE metrics. To evaluate the statistical significance of the results obtained, the permutation test was applied performing 1000 permutations and using $\alpha = 0.05$. Unlike WEKA, PRoNTo already provides a permutation test algorithm.

An important observation to make is that PRoNTo does not offer a *seed* value in order to enable the replicability of the calculated *p-values*. As explained in Section 2.10, the permutation test applied in this work, randomly shuffles the values of the PCL dependent attribute. In order to enable a replicability of the calculate *p-value* for each results measure, the implementation for WEKA was made so that the user can choose a *seed*, which will allow to obtain the same random shuffle of the attribute values in a future run of the same analysis. Therefore, the use of the same *seed* guarantees the replicability of the calculated *p-values*.

Table 6.2 contains the results obtained with the four algorithms. As mentioned, PRoNTo does not allow the user to define parameter values for the algorithms used. Therefore, the results were obtained with the default settings of each algorithm. As can be seen, the highest correlation coefficient (0.28) was obtained with the epsilon-SVR algorithm, presenting a p-value = 0.0559. Due to the limitation of PRoNTo, it was not possible to know which value was assigned to ϵ. Even performing 1000 iterations, PRoNTo informs the calculated *p-values* using four decimal places.

Table 6.2: Regression analysis results in PRoNTo

Algorithm	R	R p-value	MSE	MSE p-value
GPR	0.2	0.1598	155.64	0.1758
KRR	0.19	0.1598	182.17	0.2488
RVR	0.19	0.1518	158.89	0.1449
epsilon-SVR	**0.28**	**0.0559**	**160.04**	**0.05**

6.4 Final Remarks

Although most of the applied regression algorithms are based on SVM (i.e., KRR, RVR and epsilon-SVR in PRoNTo and SMOreg in WEKA), the best correlation coefficient (0.4164) obtained was using algorithm IBk with $k = 4$ in WEKA. Also when analyzing

the results of SVM-based algorithms of both softwares, the WEKA SMOreg algorithm obtained a better correlation coefficient (0.3318) compared to 0.28 of epsilon-SVR algorithm and 0.19 of KRR and RVR algorithms. Moreover, an article with the results obtained in the regression analyses using WEKA performed in this chapter was published [14].

The fact that PRoNTo does not allow the definition of parameter values becomes a disadvantage compared to WEKA. In addition, WEKA has several algorithms to fulfill various DM tasks (e.g., preprocessing).

Chapter 7

Conclusion

The application of computing for healthcare to identify patients with psychiatric disorders and even candidates to develop such disorders in the future is the focus of many studies and researches. This kind of application can assist physicians and specialists in prescribing more efficient and effective diagnoses.

DM techniques and their classification and regression analyses applied in this work used a dataset with 55 instances and 11 attributes (ten independent attributes and one dependent attribute). Out of the ten independent attributes, two attributes (HR_Threat-Neutral and NR_Threat-Neutral) are derived, i.e., they are calculated using other attributes.

This chapter highlights our main results and contributions. Section 7.1 summarizes the best results obtained in classification and regression analyses discussing the limitations faced and contributions achieved. Section 7.2 discusses future work.

7.1 Best Results and Contributions

The best result obtained in the classification analysis using 36 as cut-off point and without applying the Auto-WEKA plugin suggestion presented accuracy of 81.82% (p-value of 0.001), precision of 0.75 (p-value of 0.037), recall of 0.4286 (p-value of 0.002) and F-Measure of 0.5455 (p-value of 0.001) using the SMO algorithm with its *kernel* = Puk in the non-discretized dataset and increasing the minority class by 100% (see Table 5.21).

In most classification analyses performed, the supervised discretization, attribute selection, and class balancing techniques increased the accuracy of the classifiers. The use of Auto-WEKA plugin, as showed in Section 5.4, has also increased the accuracy

when applying the suggested SMO algorithm with its suggested parameters values to the non-discretized dataset and increasing the minority class by 100%. Its result presented accuracy of 85.45% (p-value of 0.001), precision of 0.8 (p-value of 0.001), recall of 0.5714 (p-value of 0.271) and F-Measure of 0.6667 (p-value of 0.001).

As explained in Section 2.2, the suggested range of cut-off points values goes from 36 to 44. Thus, an evaluation of the range's maximum value (44) was also considered interesting. It is important to keep in mind that using a higher cut-off will increase the dataset imbalance in this case.

When using 44 as cut-off point, the best result presented accuracy of 89.09% (p-value of 0.001), precision of 1 (p-value of 0.048), recall of 0.25 (p-value of 0.009) and F-Measure of 0.4 (p-value of 0.001) applying the k-NN algorithm with $k = 5$ to the unbalanced non-discretized dataset (see Table 5.33).

Using the Auto-WEKA plugin with 44 as cut-off point has also increased the accuracy when applying the suggested k-NN algorithm with its suggested parameters values to the unbalanced non-discretized dataset. Its result (see Table 5.35) presented accuracy of 90.91% (p-value of 0.001), precision of 0.8 (p-value of 0.02), recall of 0.5 (p-value of 0.001) and F-Measure of 0.6154 (p-value of 0.001).

In the regression version of the problem, the best result obtained with the WEKA software [63] presented correlation coefficient of 0.4164 (p-value of 0.001), also using the IBk algorithm with its $k = 4$ parameter (see Table 6.1). Using the PRoNTo software [49], the best result presented correlation coefficient of 0.28 (p-value of 0.559), using the epsilon-SVR algorithm (see Table 6.2).

The main limitations faced with the dataset were its small number of instances, along with its imbalance (the number of majority class instances is almost three times bigger than minority class, when using 36 as cut-off point, and almost six times bigger when using 44 as cut-off point). They certainly affected the classification and regression analysis results.

Despite the limitations, this work has analyzed different classification and regression techniques, proving to be possible to apply DM in order to predict PTSD traits in individuals who suffered a traumatic event, using the HR and SC physiological signals collected during the visualization of emotional and neutral stimuli images. The results achieved satisfactory accuracies (85.45% and 90.91%, using the cut-off points 36 and 44, respectively) and correlation coefficients (0.4164 and 0.28, using the WEKA [63] and PRoNTo

[49] softwares, respectively). In the future, this kind of analysis could be used for defining new biomarkers for this type of psychiatric disorder, enabling clinicians to identify PTSD traits using an affordable setup.

An additional technical contribution of this work was the implementation of the permutation test algorithm, available in `https://github.com/luizponte/WEKA` (see Section 2.10), aiming at adding a statistical significance validation of the results of both classification and regression analyses to the WEKA software.

Section 4.3 explained the implemented brute force algorithm for supervised discretization, using the *gain ratio*, *gini index* and *information gain* metrics, as an attempt to solve the problem faced by the supervised discretization algorithms of WEKA, *scikit-learn* and the *R* programming language. The implementation is also available in `https://github.com/luizponte/WEKA`. This could serve as inspiration to implement a new supervised discretization algorithm, which allows the user to delimit how many bins are desired, preventing just a single bin.

As mentioned in sections 5.6 and 6.4, two articles were published, reporting the results obtained in classification [15] and in regression [14] analyses, presenting the techniques applied.

7.2 Future Work

As explained in Section 2.2, the PCL scale used in this work already has a new version, the PCL-5 [59]. In order to use a more updated version and also to increase the dataset size, a new execution of the experiment would be interesting. This execution could tackle the imbalance problem in parallel, seeking to recruit volunteer candidates to present high PCL scale values (i.e., values above the cut-off point used). Thus, by inserting real instances into the dataset, the imbalance between the number of instances of the classes would be minimized.

In addition, other DM techniques can be explored, such as outlier detection, to identify anomalies in individuals data which significantly differ from the majority of data and can lead to a biased result; association rule discovery, to find interesting informations about how much one or more attributes increase or decrease the chances of an individual to present a high or a low PCL value; and the use of other classification and regression algorithms.

Still as future work, the collection and use of other physiological signals or types of data (e.g., molecular and image data) would also be interesting. As indicated in Chapter 3, many studies explored other types of physiological signals and data, but they did so using other scales different from PCL.

References

[1] Aha, D. W.; Kibler, D.; Albert, M. K. Instance-based learning algorithms. *Machine Learning 6*, 1 (1991), 37–66.

[2] Alves, R. d. C. S. Modulação da Frequência Cardíaca pela Visualização de Estímulos de Ameaça. Master's thesis, Biomedical Institute, Federal Fluminense University, Niterói, RJ, Brazil, 2012.

[3] Alves, R. d. C. S.; Portugal, L. C.; Fernandes Jr, O.; Mocaiber, I.; Souza, G. G.; David, I. d. P. A.; Volchan, E.; de Oliveira, L.; Pereira, M. G. Exposure to trauma-relevant pictures is associated with tachycardia in victims who had experienced an intense peritraumatic defensive response: the tonic immobility. *Frontiers in Psychology 5* (2014), 1514.

[4] Association, A. P., et al. *Diagnostic and statistical manual of mental disorders (DSM-5)*, 5 ed. American Psychiatric Pub, 2013.

[5] Blanchard, E. B.; Jones-Alexander, J.; Buckley, T. C.; Forneris, C. A. Psychometric properties of the PTSD Checklist (PCL). *Behaviour Research and Therapy 34*, 8 (1996), 669–673.

[6] Bliese, P. D.; Wright, K. M.; Adler, A. B.; Cabrera, O.; Castro, C. A.; Hoge, C. W. Validating the primary care posttraumatic stress disorder screen and the posttraumatic stress disorder checklist with soldiers returning from combat. *Journal of Consulting and Clinical Psychology 76*, 2 (2008), 272.

[7] Bradley, M. M.; Hamby, S.; Löw, A.; Lang, P. J. Brain potentials in perception: picture complexity and emotional arousal. *Psychophysiology 44*, 3 (2007), 364–373.

[8] Brady, K. T.; Killeen, T.; Saladln, M. E.; Dansky, B.; Becker, S. Comorbid substance abuse and posttraumatic stress disorder: Characteristics of women in treatment. *American Journal on Addictions 3*, 2 (1994), 160–164.

[9] Breiman, L. Random forests. *Machine Learning 45*, 1 (2001), 5–32.

[10] Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16* (2002), 321–357.

[11] Cohen, H.; Kotler, M.; Matar, M. A.; Kaplan, Z.; Loewenthal, U.; Miodownik, H.; Cassuto, Y. Analysis of heart rate variability in posttraumatic stress disorder patients in response to a trauma-related reminder. *Biological Psychiatry 44*, 10 (1998), 1054–1059.

[12] Copeland, W. E.; Keeler, G.; Angold, A.; Costello, E. J. Traumatic events and posttraumatic stress in childhood. *Archives of General Psychiatry 64*, 5 (2007), 577–584.

[13] Costello, E. J.; Erkanli, A.; Fairbank, J. A.; Angold, A. The prevalence of potentially traumatic events in childhood and adolescence. *Journal of Traumatic Stress: Official Publication of The International Society for Traumatic Stress Studies 15*, 2 (2002), 99–112.

[14] da Ponte Junior, L. A.; Muchaluat-Saade, D. C.; Plastino, A.; Alves, R. d. C.; Lima Portugal, L. C.; de Oliveira, L.; Pereira, M. G. Identificando Sinais de Estresse Pós-traumático Utilizando Dados Fisiológicos e Técnicas de Regressão. In *Anais Principais do XX Simpósio Brasileiro de Computação Aplicada à Saúde* (2020), SBC.

[15] da Ponte Junior, L. A.; Muchaluat-Saade, D. C.; Plastino, A.; Alves, R. d. C.; Lima Portugal, L. C.; de Oliveira, L.; Pereira, M. G. Identifying Post-Traumatic Stress Symptoms Using Physiological Signals and Data Mining. In *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)* (2020).

[16] Dean, K. R.; Hammamieh, R.; Mellon, S. H.; Abu-Amara, D.; Flory, J. D.; Guffanti, G.; Wang, K.; Daigle, B. J.; Gautam, A.; Lee, I., et al. Multi-omic biomarker identification and validation for diagnosing warzone-related post-traumatic stress disorder. *Molecular Psychiatry* (2019), 1–13.

[17] Dekel, O.; Shalev-Shwartz, S.; Singer, Y. Smooth ε-insensitive regression by loss symmetrization. *Journal of Machine Learning Research 6*, May (2005), 711–741.

[18] Elsesser, K.; Sartory, G.; Tackenberg, A. Attention, heart rate, and startle response during exposure to trauma-relevant pictures: a comparison of recent trauma victims and patients with posttraumatic stress disorder. *Journal of Abnormal Psychology 113*, 2 (2004), 289.

[19] Fayyad, U.; Irani, K. Multi-interval discretization of continuous-valued attributes for classification learning. In *The Thirteenth International Joint Conference on Artificial Intelligence* (1993), pp. 1022–1027.

[20] Fisher, R. A. Design of experiments. *Br Med J 1*, 3923 (1936), 554–554.

[21] Fishman, G. *Monte Carlo: concepts, algorithms, and applications*. Springer Science & Business Media, 2013.

[22] Freedy, J. R.; Steenkamp, M. M.; Magruder, K. M.; Yeager, D. E.; Zoller, J. S.; Hueston, W. J.; Carek, P. J. Post-traumatic stress disorder screening test performance in civilian primary care. *Family practice 27*, 6 (2010), 615–624.

[23] Galatzer-Levy, I. R.; Karstoft, K.-I.; Statnikov, A.; Shalev, A. Y. Quantitative forecasting of PTSD from early trauma responses: A machine learning application. *Journal of Psychiatric Research 59* (2014), 68–76.

[24] Han, J.; Pei, J.; Kamber, M. *Data mining: concepts and techniques*, 3 ed. Elsevier, 2011.

[25] Harrington, T.; Newman, E. The psychometric utility of two self-report measures of PTSD among women substance users. *Addictive Behaviors 32*, 12 (2007), 2788–2798.

[26] Hastie, T.; Tibshirani, R. Classification by pairwise coupling. In *Advances in Neural Information Processing Systems* (1998), pp. 507–513.

[27] Hsing, T.; Attoor, S.; Dougherty, E. Relation between permutation-test P values and classifier error estimates. *Machine Learning 52*, 1-2 (2003), 11–30.

[28] John, G. H.; Langley, P. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (1995), Morgan Kaufmann Publishers Inc., pp. 338–345.

[29] Karstoft, K.-I.; Galatzer-Levy, I. R.; Statnikov, A.; Li, Z.; Shalev, A. Y. Bridging a translational gap: using machine learning to improve the prediction of PTSD. *BMC Psychiatry 15*, 1 (2015), 30.

[30] Karstoft, K.-I.; Statnikov, A.; Andersen, S. B.; Madsen, T.; Galatzer-Levy, I. R. Early identification of posttraumatic stress following military deployment: Application of machine learning methods to a prospective study of Danish soldiers. *Journal of Affective Disorders 184* (2015), 170–175.

[31] Keerthi, S. S.; Shevade, S. K.; Bhattacharyya, C.; Murthy, K. R. K. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation 13*, 3 (2001), 637–649.

[32] Kotthoff, L.; Thornton, C.; Hoos, H. H.; Hutter, F.; Leyton-Brown, K. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *Journal of Machine Learning Research 18*, 1 (2017), 826–830.

[33] Lanius, R. A.; Williamson, P. C.; Boksman, K.; Densmore, M.; Gupta, M.; Neufeld, R. W.; Gati, J. S.; Menon, R. S. Brain activation during script-driven imagery induced dissociative responses in PTSD: a functional magnetic resonance imaging investigation. *Biological Psychiatry 52*, 4 (2002), 305–311.

[34] Leightley, D.; Williamson, V.; Darby, J.; Fear, N. T. Identifying probable post-traumatic stress disorder: applying supervised machine learning to data from a UK military cohort. *Journal of Mental Health 28*, 1 (2019), 34–41.

[35] Liberzon, I.; Taylor, S. F.; Amdur, R.; Jung, T. D.; Chamberlain, K. R.; Minoshima, S.; Koeppe, R. A.; Fig, L. M. Brain activation in PTSD in response to trauma-related stimuli. *Biological Psychiatry 45*, 7 (1999), 817–826.

[36] Marin, M.-F.; Zsido, R. G.; Song, H.; Lasko, N. B.; Killgore, W. D.; Rauch, S. L.; Simon, N. M.; Milad, M. R. Skin conductance responses and neural activations during fear conditioning and extinction recall across anxiety disorders. *JAMA Psychiatry 74*, 6 (2017), 622–631.

[37] Marinić, I.; Supek, F.; Kovačić, Z.; Rukavina, L.; Jendričko, T.; Kozarić-Kovačić, D. Posttraumatic stress disorder: diagnostic data analysis by data mining methodology. *Croatian Medical Journal 48*, 2. (2007), 185–197.

[38] Monson, C. M.; Gradus, J. L.; Young-Xu, Y.; Schnurr, P. P.; Price, J. L.; Schumm, J. A. Change in posttraumatic stress disorder symptoms: do clinicians and patients agree? *Psychological Assessment 20*, 2 (2008), 131.

[39] Ojala, M.; Garriga, G. C. Permutation tests for studying classifier performance. *Journal of Machine Learning Research 11*, Jun (2010), 1833–1863.

[40] Omurca, S. İ.; Ekinci, E. An alternative evaluation of post traumatic stress disorder with machine learning methods. In *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)* (2015), IEEE, pp. 1–7.

[41] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V., et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, Oct (2011), 2825–2830.

[42] Pitman, E. J. Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society 4*, 1 (1937), 119–130.

[43] Portugal, L. C.; Rosa, M. J.; Rao, A.; Bebko, G.; Bertocci, M. A.; Hinze, A. K.; Bonar, L.; Almeida, J. R.; Perlman, S. B.; Versace, A., et al. Can emotional and behavioral dysregulation in youth be decoded from functional neuroimaging? *PLOS ONE 11*, 1 (2016), e0117603.

[44] Quinlan, J. R. *C4.5: programs for machine learning*. Elsevier, 2014.

[45] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[46] Regier, D. A.; Farmer, M. E.; Rae, D. S.; Locke, B. Z.; Keith, S. J.; Judd, L. L.; Goodwin, F. K. Comorbidity of mental disorders with alcohol and other drug abuse: results from the Epidemiologic Catchment Area (ECA) study. *JAMA 264*, 19 (1990), 2511–2518.

[47] Ruggiero, K. J.; Del Ben, K.; Scotti, J. R.; Rabalais, A. E. Psychometric properties of the PTSD Checklist—Civilian version. *Journal of traumatic stress 16*, 5 (2003), 495–502.

[48] Saxe, G. N.; Ma, S.; Ren, J.; Aliferis, C. Machine learning methods to predict child posttraumatic stress: a proof of concept study. *BMC Psychiatry 17*, 1 (2017), 223.

[49] Schrouff, J.; Rosa, M. J.; Rondina, J. M.; Marquand, A. F.; Chu, C.; Ashburner, J.; Phillips, C.; Richiardi, J.; Mourao-Miranda, J. PRoNTo: pattern recognition for neuroimaging toolbox. *Neuroinformatics 11*, 3 (2013), 319–337.

[50] Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology 85* (2018), 1–16.

[51] Shaffer, F.; Ginsberg, J. An overview of heart rate variability metrics and norms. *Frontiers in Public Health 5* (2017), 258.

[52] Shalev, A. Y.; Sahar, T.; Freedman, S.; Peri, T.; Glick, N.; Brandes, D.; Orr, S. P.; Pitman, R. K. A prospective study of heart rate response following trauma and the subsequent development of posttraumatic stress disorder. *Archives of General Psychiatry 55*, 6 (1998), 553–559.

[53] Shevade, S. K.; Keerthi, S. S.; Bhattacharyya, C.; Murthy, K. R. K. Improvements to the SMO algorithm for SVM regression. *IEEE Transactions on Neural Networks 11*, 5 (2000), 1188–1193.

[54] Smola, A. J.; Schölkopf, B. A tutorial on support vector regression. *Statistics and Computing 14*, 3 (2004), 199–222.

[55] Thornton, C.; Hutter, F.; Hoos, H. H.; Leyton-Brown, K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2013), ACM, pp. 847–855.

[56] Tipping, M. E. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research 1*, Jun (2001), 211–244.

[57] Vázquez, C.; Pérez-Sales, P.; Matt, G. Post-traumatic stress reactions following the March 11, 2004 terrorist attacks in a Madrid community sample: A cautionary note about the measurement of psychological trauma. *The Spanish Journal of Psychology 9*, 1 (2006), 61–74.

[58] Weathers, F. W.; Litz, B. T.; Herman, D. S.; Huska, J. A.; Keane, T. M., et al. The PTSD Checklist (PCL): Reliability, validity, and diagnostic utility. In *Annual Convention of The International Society for Traumatic Stress Studies, San Antonio, TX* (1993), vol. 462, San Antonio, TX.

[59] Weathers, F. W.; Litz, B. T.; Keane, T. M.; Palmieri, P. A.; Marx, B. P.; Schnurr, P. P. The PTSD checklist for DSM-5 (PCL-5). *Scale available from the National Center for PTSD at* `www.ptsd.va.gov` (2013).

[60] Welling, M. Kernel ridge regression. *Max Welling's Classnotes in Machine Learning* (2013), 1–3.

[61] Williams, C. K.; Rasmussen, C. E. *Gaussian processes for machine learning*, vol. 2. MIT press Cambridge, MA, 2006.

[62] Wilson, A.; Adams, R. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning* (2013), pp. 1067–1075.

[63] Witten, I. H.; Frank, E.; Hall, M. A.; Pal, C. J. *Data Mining: Practical machine learning tools and techniques*, 4 ed. Morgan Kaufmann, 2016.