






Universitat Autònoma de Barcelona

ADVERTIMENT. L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons:  http://cat.creativecommons.org/?page_id=184

ADVERTENCIA. El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons:  <http://es.creativecommons.org/blog/licencias/>

WARNING. The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license:  <https://creativecommons.org/licenses/?lang=en>

AUTONOMOUS UNIVERSITY OF BARCELONA

SCIENCE FACULTY

SUPERVISED MACHINE LEARNING:
A THEORETICAL STUDY WITH
APPLICATIONS

BY JOSÉ DAVID NÚÑEZ GONZÁLEZ

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY IN MATHEMATICS

Advisor: Rosario Delgado de la Torre

2022

Department of Mathematics

Acknowledgements

I open this chapter of acknowledgments by thanking my thesis director Rosario Delgado. I would need to write another thesis report to thank all her dedication and performance to successfully complete this Thesis.

I would also like to thank Juan Carlos Yébenes and Angel Lavado (Hospital de Mataró) for their contributions and their invitation to participate in the project financed by Marató TV3, key in this Thesis.

As in previous events, I want to thank my parents for all the support they have given me, whose help has been essential for everything in general and for the thesis in particular.

I do not want to forget to thank my great friend and colleague Joseba Santisteban, for his unconditional availability and for being the alma mater of this Thesis, when in May 2017, waiting to enter a graduation ceremony, he told me: “Why don’t you get a Ph.D. in Mathematics?”

Abstract

This Thesis is framed in the topic of Supervised Machine Learning, where we present a theoretical study with applications. Specifically, contributions have been made at the different moments of the Machine Learning life cycle from an integral point of view, focusing our attention on the three fundamental stages of the cycle: preprocessing of the dataset, construction of the predictive model (classifier), and validation of the model using performance metrics.

The first work focuses on the preprocessing phase. We have proposed a novel oversampling method that uses a Bayesian network constructed as the probabilistic model for the relationships of dependence between the features in the minority class setting, to generate artificial instances of the minority class for a dataset with both categorical and/or continuous variables. It relies on the fact that the likelihood is a measure of the goodness of fit of a model to a set of instances, which is a paradigm different from that in which the existing oversampling methods are based: the idea of distance between the features, which turns out to be a weakness when applied to datasets with non-continuous variables.

The second paper is related to the construction of a predictive model, specifically, a classifier. We have implemented an expert system based on an ensemble of Bayesian classifiers to help in decision making in the Intensive Care Unit of the Hospital of Mataró. The system predicts the vital outcome of the patient admitted to the ICU (live/die) as well as the destination upon discharge from the ICU, if the prediction is “live”, or the cause of death if it is “die”. The combination rule to decide the prediction provided by the ensemble, from the predictions given by the base classifiers, is a Weighted Average with specific weights based on the Area Under the Precision-Recall curve (AUPR), suitable for deal with unbalanced datasets, which is compatible with the MAP criterion.

The last contribution attends to the validation phase. We have introduced an improvement of the original definition of the Confusion ENtropy (CEN) metric, which is based on the Shannon’s entropy from the field of Information Theory, as a measure of the uncertainty entailed by the result of a classification process. This modification allows to avoid the undesired behaviour showed by CEN, which in some cases is

“out-of-range”, and in some others shows a lack of monotonicity when the situation monotonically goes from perfect to completely wrong classification.

Contents

Acknowledgements	2
Abstract	3
1 Introduction	7
1.1 Brief definitions and basic notations	7
1.2 General objectives	11
1.3 Research methodology	11
1.4 Specific objectives	12
1.4.1 Preprocessing	13
1.4.2 Building the model	16
1.4.3 Validation	24
2 Overall Results	34
2.1 Preprocessing (BOSME)	34
2.1.1 BOSME as oversampling method	35
2.1.2 The wrapper: cost-sensitive approach	36
2.1.3 Results	37
2.2 Building the model (Ensemble Weighted Average)	41
2.2.1 Comparing the models for the variable <i>Result</i>	42
2.2.2 Centrality measures	45
2.2.3 Odds Ratio	46
2.2.4 Feature strength	47
2.3 Validation (MCEN)	49
2.3.1 The perfectly symmetric and balanced case	49
2.3.2 The binary case	51
2.3.3 The multiclass Z_A family	55
2.4 Relevance of the results	58
3 Final conclusions	63

A	Articles	69
A.1	Bayesian Network-based Over-Sampling MMethod	69
A.2	Ensemble of Bayesian Classifiers	88
A.3	Modified Confusion Entropy	115

Chapter 1

Introduction

This introductory chapter aims to give an overall overview of the thesis, providing some of the motivations for the research work carried out, and some ideas on each of the topics addressed in it. The chapter is organized as follows: Section 1.1 provides brief definitions and basic notations on Supervised Machine Learning in general, and Classification in particular. Section 1.2 explains the general objectives of the investigation, and in Section 1.3 the research methodology is briefly introduced, while in Section 1.4 the general objectives are concretized in specific objectives framed in three moments of the Machine Learning cycle: Preprocessing, Building the model (classifier), and Validation.

The rest of the thesis is organized as follows: Chapter 2 presents overall results, Chapter 3 concludes this thesis, and the Appendix includes the published papers that constitute the thesis. Since the papers were written and published at different times, months apart, keep in mind that the notation has varied from one to another since it was adapted to the usual one in the specific field in which the work carried out falls. In this thesis we have combined notations to avoid inconsistencies and facilitate reading, but these notations do not always coincide with those of the corresponding papers.

1.1 Brief definitions and basic notations

Let start giving some brief definitions related to the topic of the thesis where contributions are given in **Machine Learning** (ML). In the summer of 1956, John McCarthy presented his definition of Artificial Intelligence (AI) at the Dartmouth College convention, as “the science and engineering to make intelligent machines”. ML was introduced in 1959 (see [1]) as a subset of AI in the field of Computer Science that often uses Mathematics and Statistics, to give computers the ability to learn. That is, ML is a part of the AI in the intersection between Computer Science on the one side, and Mathematics and Statistics on the other, as can be seen in Figure 1.1.

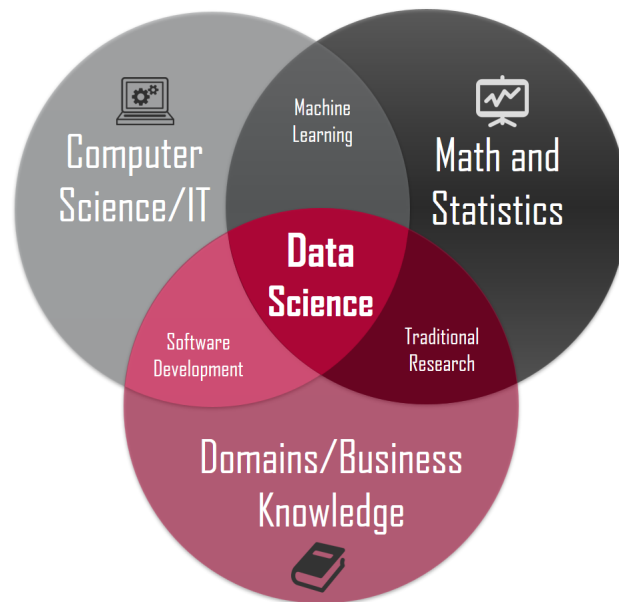


Fig 1.1. Diagram of the relationship between the different disciplines related to Machine Learning. From Kharkovyna, O. “A Gentle Intro to Probability and Statistics for Data Science”. 2020, March 09; Available from: <https://www.linkedin.com/pulse/gentle-intro-probability-statistics-data-science-oleksii-kharkovyna>

Machine Learning is subdivided in two main categories:

- **Supervised Learning** is defined by [2] as the ML task of learning a function that maps an input to an output based on example input-output pairs. In other words, it is the part of the ML devoted to the task of predict one or several output variables from the input variables, through an algorithm learned from a training dataset of instances for which both input and output variables are known. The name comes from the fact that the output variables allow us to know if the algorithm that predicts them has a good behavior or not and, therefore, they “supervise” it.
- **Unsupervised Learning** is defined by [3] as the part of ML who deals with the algorithms used for the task of discover relations between the variables in a dataset, capturing patterns and allowing the instances to group in clusters, for example, but in which there is no an output variable to predict that could, therefore, act as a “supervisor”.

In the Thesis we focus in the Supervised ML. On the one hand, it covers **Regression** tasks, that is, a linear approach to modelling the relationship between a quantitative (continuous or discrete taking a large number of different values) response (which is the output, or dependent variable) and one or more explanatory variables (which are the input, or independent variables).

On the other hand, Supervised ML also includes **Classification**, which is the task of identify to which of a set of categories (sub-populations) a new instance belongs, on the basis of a training set of data containing instances whose category membership is known. That is, classification tries to predict a categorical output variable, or what is the same, to assign labels to new instances. We use “class”, “label” or “category” interchangeably. This task is performed by a **classifier**, which is an algorithm implemented in a programming language, learned from a dataset formed of instances with known values of the input and output variables. What characterizes **classification** in the context of ML is that the output variables are categorical (or convertible to categorical), while in the **regression** task, the output or predicted variable is continuous or discrete, with a large number of possible values. Classification is the field within the Supervised ML in which the research of the Thesis has been developed.

We are especially interested in studying the **probabilistic classifiers**, which do not only predict the class of a new instance from its values for the input variables, but assigns it a (discrete) probability distribution over the classes. Usually, the class with the highest probability is the chosen one as the prediction, when following the *maximum a posteriori probability* (MAP) criterion, being this probability the *Confidence Level* associated to the prediction or classification.

As we have already mentioned, in the construction of the classifiers we follow a *data driven* approach, that is, we learn the classifiers from a dataset. To get an idea of how good the predictive behaviour of the classifier learned from a dataset is, as well as to compare with other classifiers, a validation process must be carried out. Let assume a dataset which instances have X_1, \dots, X_n features and are labeled according to the category or class to which they belong, corresponding to an output (or “class”) variable, V , which is, or can be thought of as, categoric, with $r \geq 2$ different categories: $\Omega_V = \{v_1, \dots, v_r\}$. Input variables X_1, \dots, X_n can be categoric or numeric (integer or continuous).

To carry out the validation procedure of the classifier, first we need to prepare a training set to learn the classifier, and a disjoint test set to validate it. As validation technique we use *k-fold cross-validation*, which consists of dividing the database into k disjoint folds at random, to construct the training dataset merging $k - 1$ folds and the test set as the remaining fold. This process is repeated k times changing each time the fold used as test set. In order to avoid bias, we select different seeds to make the random partition of the dataset into folds. Each time we take a different seed we are doing a different “run” of the procedure, learning the classifier from a different training dataset, and therefore obtaining a different predictive model. In this way, we avoid overfitting by generating a bit of “noise” to bypass obtaining a model excessively adapted to specific data. Once the classifier is learned from a training dataset, we

validate it using the test set. For that, we use the classifier to make the prediction of the output variable of the test set instances, using the MAP criterion mentioned before. In fact, for any instance of the test set, given the values of the input variables, this criterion assigns to the instance, as a label, the class \tilde{v} with the maximum a posteriori probability, that is:

$$\begin{aligned}\tilde{v} &= \arg \max_{v \in \Omega_V} P(V = v / X_1, \dots, X_n) = \arg \max_{v \in \Omega_V} \frac{P(X_1, \dots, X_n / V = v) P(V = v)}{P(X_1, \dots, X_n)} \\ &= \arg \max_{v \in \Omega_V} P(X_1, \dots, X_n / V = v) P(V = v)\end{aligned}\quad (1.1)$$

where we have used Bayes' Theorem. Note that in order to obtain \tilde{v} , by (1.1), we only need to know the marginal distribution of V (that is, $P(V = v)$ for $v \in \Omega_V$) and the conditional probabilities of the input variables to V : $P(X_1, \dots, X_n / V = v)$ for $v \in \Omega_V$, which can be obtained from the parameters of the classifier.

For each instance of the test set, we then can compare the predicted and the observed labels. Results of the prediction are summarized in the $r \times r$ confusion matrix $C = (C_{ij})_{i,j=1,\dots,r}$, which has the form

$$\begin{array}{c} \text{observed} \end{array} C = \begin{array}{c} \text{predicted} \\ \begin{pmatrix} C_{11} & \dots & C_{1r} \\ \dots & \dots & \dots \\ C_{r1} & \dots & C_{rr} \end{pmatrix} \end{array}\quad (1.2)$$

where C_{ij} is the number of instances in the test set that being of the class v_i have been predicted as belonging to the class v_j . Note that if the test set has N instances, then $N = \sum_{i=1}^r \sum_{j=1}^r C_{ij}$. From the confusion matrix, different metrics can be calculated. The golden standard in the general multiclass ($r \geq 2$) classification is *Accuracy* and, associated with it, the other side of the coin, which is the *Error rate*. Although we will not introduce them in this section, other specific measures have been introduced in the binary case $r = 2$ (see Section 1.4.2).

- *Accuracy* is the ratio between the number of correct predictions and the total number of predictions, that is, the proportion of correct predictions:

$$\text{Accuracy} = \frac{\sum_{i=1}^r C_{ii}}{N}$$

- *Error rate* is the the proportion of wrong predictions:

$$\text{Error rate} = 1 - \text{Accuracy} = \frac{N - \sum_{i=1}^r C_{ii}}{N}$$

From the metric values obtained using *k-fold cross-validation* (in each run we get k values) we can get an idea of how good the behavior of the classifier is in terms of its mean and standard deviation, or even a confidence interval. In addition, using the values of the metric it is possible to compare with other models (classifiers) by performing suitable statistical hypothesis tests.

1.2 General objectives

The general objectives of the present project, framed into the Supervised Machine Learning area, follow three lines of action.

1. **Basic research.** Starting from an analysis of the state-of-the-art, we make some theoretical contributions in the area, mainly related to, although not limited to, different aspects of the study of probabilistic classifiers, which is a subject of great importance, both theoretical and applied. In particular, the question of the extension to multi-class classification of the classifier behavior measures used for binary classification, not always immediate or obvious, will be addressed.
2. **Algorithmic/computing.** In parallel with the basic research, we study in deep some of the algorithms used in Supervised Machine Learning and introduce new ones that are variants or alternatives, comparing them with those already known in a heuristic way, following an adequate experimental methodology.
3. **Applications.** Finally, we also put into practice what was developed in the two previous points. We work both with artificial families of confusion matrices generated on purpose, and with real dataset, and within these, both with some obtained from well known and contrasted repositories of public access, as well as a real dataset provided by the Hospital de Mataró (Consorci Sanitari del Maresme) in the context of a research project of the Fundació Marató de TV3.

1.3 Research methodology

The methodological procedure of investigation in all the papers that make up this thesis has been the following:

- (i) detect a deficiency, problem or failure (in short, something that could be improved) in some aspect of Supervised Machine Learning,
- (ii) propose an alternative or solution to it, and
- (iii) test this proposal from a theoretical and/or empirical perspective.

The computational/algorithmic part of the thesis has been programmed through scripts in R ([4]), which has also been used to conduct analysis and procedures, with the help of different available libraries. In particular, R has been used to implement the algorithms that generate the families of confusion matrices we work with in paper [5]

Databases from UCI ¹ and KAGGLE ² repositories have been used in the experimental phase of our works.

High computational cost experiments have been carried out on the servers of the Autonomous University of Barcelona (UAB) and the University of the Basque Country (UPV/EHU), while the rest of procedures have been performed in a laptop whose processor is an Intel Core i7 10th generation (2.3 GHz) and 16 GB RAM.

1.4 Specific objectives

The specific objectives in which the aforementioned general objectives are concreted fit into the three phases of Supervised Machine Learning: **Preprocessing**, **Building the model**, and **Validation**. This is the context in which the three works that make up the thesis show to have a thematic unity.

For each specific objective we will introduce the works addressed in this thesis and after a brief introduction, the objective will be structured in the following steps: (i) deficiency or problem, (ii) proposed alternative, and (iii) testing the proposed alternative, as mentioned in Section 1.3. These objectives are structured around three topics of great interest in current research, that we will explain along this document:

- (a) The **class unbalanced** problem.
- (b) The **cost-sensitive** paradigm.
- (c) The **entropy** as a measure of the disorder.

We have addressed topic (a) both 1) from the cost-sensitive perspective of topic (b), by proposing an oversampling method that modifies conveniently the training set before learning the classifier, in [6], and 2) by proposing an ensemble of classifiers, that is, modifying the classification algorithm itself, in [7]. Finally, in [5] we have worked on the redefinition of a metric based on the concept of **entropy** of the area of the Information Theory, that gives a measure of the misclassification represented by a confusion matrix through topic (c).

¹<https://archive.ics.uci.edu>

²<https://www.kaggle.com>

1.4.1 Preprocessing

In the lifecycle of Machine Learning, during the *preprocessing phase* the original dataset is transformed in order to prepare it for the subsequent phases. There are several things we can do during the first stage of this phase, in which transformations are carried out whose need derives from the very nature of the dataset or from its necessary adaptation to the characteristics of the Supervised Machine Learning methodology that it is intended to apply. Some examples are: cleaning, normalization, discretization, transformation of variables, and feature selection.

Sometimes, however, it is necessary to carry out a second stage in this phase but only for the training dataset. Care must be taken since the training set changes with each change of fold, if we use the *k-fold cross-validation* procedure. This preprocessing modifies the part of the original data used for training and must be executed before the learning process of the predictive model (classifier). The reason, in this case, has to do with trying to learn a better predictive model from the (enlarged) training set, in the sense that its behavior, evaluated in the validation procedure, when assigning labels to the instances of the test set, be better. It is very important to remember that the dataset used as test should never be modified in this second stage of the *preprocessing phase*, since it is about handling the training set to learn a better classifier for the original test set. The opposite would be like “cheating playing solitaire”.

We focus on the binary case $r = 2$ and in the scenario in which the dataset is **unbalanced** (topic (a)). A dataset is balanced if the two classes are approximately equally represented, otherwise it is said that the dataset is **unbalanced** [8]. This is a typical situation in medical data domains, for instance: sick/healthy or alive/dead, to cite two examples. As usual, in the binary case the two classes are denoted by $+$ (positive) and $-$ (negative), and if the data set is unbalanced, typically the minority is the $+$ class.

(i) Deficiency or problem

If the dataset is unbalanced and if no preprocessing is done, after training a “usual” classifier that does not take into account the unbalanced character, the classifier will be biased to the majority class. For example, assume a dataset whose instances are distributed as follows: 5% for the positive class (minority) and 95% for the negative class (majority). Then, if the proportion 5-95% is maintained in the test set and it contains 100 instances, we could obtain a confusion matrix like the following, where classification clearly is in favor of the majority class:

$$\begin{array}{c} \text{observed} \\ \text{observed} \end{array} C_1 = \begin{array}{c} \text{predicted} \\ + \quad - \\ \left(\begin{array}{cc} 1 & 4 \\ 5 & 90 \end{array} \right) \end{array}$$

occurring the *accuracy paradox* since globally the classifier has a good *Accuracy* of 91% but while the instances of the majority class are well predicted (94.74% success), those of the minority class are poorly predicted (20% success).

The bias towards the majority class is detrimental if we assume the existence of different costs of misclassification (topic (b)). Considering the *Accuracy* as performance metric, or equivalently the *Error rate*, reflects a cost insensitive approximation, while a cost-sensitive approximation leads us to another metric, namely, the *Total Cost* (TC), which definition is (see [9] for instance):

$$\text{TC} = c_+ \times C_{12} + c_- \times C_{21}$$

where c_+ and c_- denote, respectively, the cost associated with misclassifying instances belonging to the positive class (*false negative error*) and to the negative class (*false positive error*).

Coming back to the example, let assume that $c_+ = 10$ and $c_- = 1$, then, the *Total Cost* associated to confusion matrix C_1 would be $\text{TC}(C_1) = 10 \times 4 + 1 \times 5 = 45$. However, if we managed to classify correctly one more element of the minority class, in exchange of misclassifying three elements of the majority class (previously well classified), the confusion matrix would be now:

$$\begin{array}{c} \text{observed} \\ \text{observed} \end{array} C_2 = \begin{array}{c} \text{predicted} \\ + \quad - \\ \left(\begin{array}{cc} 2 & 3 \\ 8 & 87 \end{array} \right) \end{array}$$

and the *Total Cost* would drop to $\text{TC}(C_2) = 10 \times 3 + 1 \times 8 = 38$, which means a reduction of more than 15%. This would imply a better classification in terms of *Total Cost* despite the fact that the *Accuracy* has worsened (going down from 91% to 89%).

As said before, there are two main possibilities to deal with the problem of bias of the classifier due to the unbalanced character of the dataset from which it is learned, and the cost of the misclassification: handling in a convenient way the training set from which the classifier is learned, or modify the classifier algorithm itself. In this section we are focused in the first option, which is the use of a method that enlarges the training dataset by creating new synthetic instances of the minority class (*oversampling*). The classic method of oversampling that we find in the literature and is taken as a baseline is

the Synthetic Minority Oversampling TEchnique (SMOTE) proposed by [8]. SMOTE generates the new artificial instances of the minority class by interpolation between existing minority instances that are close to each other. Then, it is based on the concept of distance between instances, from their features values, which makes no sense if we are working with categorical variables, being a weakness of the method. There are other similar oversampling methods that have been proposed after SMOTE, such as ROSE [10], which has been introduced for continuous variables and is also based on the concept of distance between instances. Both have been adapted to deal with discrete and/or categorical features as well, but continue to rely on the concept of distance, which is counterintuitive given the nature of the features.

(ii) Proposed alternative

The alternative oversampling method to the existing ones that we propose relies in a different paradigm that overcomes their weakness. We name it Bayesian OverSampling Method (BOSME), a method to generate new artificial instances of the minority class for a dataset with both categorical and/or continuous features. It generates artificial instances randomly from the joint probability distribution entailed by a Bayesian network that is constructed as the probabilistic model for the dependency relationships between the features in the minority class setting.

The structure of this Bayesian Network is learned with a score-based structure learning algorithm with the logarithm of the likelihood function ($\log\text{Lik}$) as score, and parameters learning is carried out following the Maximum Likelihood Estimation, as usual. BOSME is based on the fact that the likelihood is a measure of the goodness of fit of a model to a set of instances, specifically, in this case, those instances corresponding to the minority class.

(iii) Testing the proposed alternative

BOSME has been tested with several datasets from the UCI and KAGGLE open repositories. Those datasets contain both categorical and/or continuous features, and most of them are unbalanced.

Before we start creating synthetic instances we need to fix in advance how many instances we want to generate. This is usually done indicating the proportion of instances of the minority class we want to achieve in the final enlarged (training) dataset. This proportion can be decided attending to other criteria, or following the cost-sensitive paradigm approach (topic (b)). Let give more detailed explanation of this second option. **Cost-sensitive learning** is a subfield of machine learning that takes into account misclassification costs when learning a classifier, with the aim of minimize the expected cost of (mis)classification. We can differentiate two alternatives:

- *Direct cost-sensitive*: which includes methods that modify the original learning algorithm to take costs into account, without modifying the dataset.
- *Indirect cost-sensitive*: which comprises methods that handle the training dataset, from which the classifier will be learned, without modifying the classifier algorithm itself. The aim is to obtain a desired class distribution in the enlarged dataset, based on the misclassification costs.

We follow the indirect cost sensitive approach, proposing a wrapper, which using given costs, calculates the proportion that the minority class must represent in the enlarged training set, say q , and then uses the oversampling method to generate the corresponding number of artificial instances of the minority class, in such a way that the proportion is achieved. The wrapper is based on the application of the **Folk Theorem**, which states that “the classifier that minimizes the *Expected Error* when it is learned from the enlarged training dataset is the same that minimizes the *Expected Cost* when learned from the original training dataset, if the proportion q is the appropriate one”. That is, the **Folk Theorem** determines the value that q must have, based on the misclassification costs.

Once we fix the proportion of synthetic instances we have to achieve, we proceed with the experimental work that is divided into two stages. In the first stage, each dataset is divided into k folds to then apply *k-fold cross validation*, with $k = 10$. Each time, we preprocess the training set to generate artificial instances of the minority class with our method (BOSME), in the context of the wrapper which allows us to follow a cost-sensitive approach, as well as with the other oversampling methods (SMOTE and ROSE). We train three classifiers (Support Vector Machine, Random Forest and Logistic Regression) and we validate them with the corresponding test sets to obtain the confusion matrices from which to get the *Accuracy* values (see Figure 1.2).

In the second stage we make statistical tests in order to determine which oversampling method has obtained better (statistically) significant results, if any.

1.4.2 Building the model

Once the data set has been explored, and after the *preprocessing phase*, which includes data cleaning and transformation, we split the data set into training and validation set, depending on the procedure to be used, in our case the *k-fold cross-validation*. Then, from any training set, a model will be learned, which, in our case, will be a classifier, which will later be validated and deployed in reality.

We can select different types of classifiers from multiple libraries oriented to Machine Learning, or we can even implement and/or modify some classifier learning algorithm ourselves. This, of course, can and should depend on the nature of the situation to

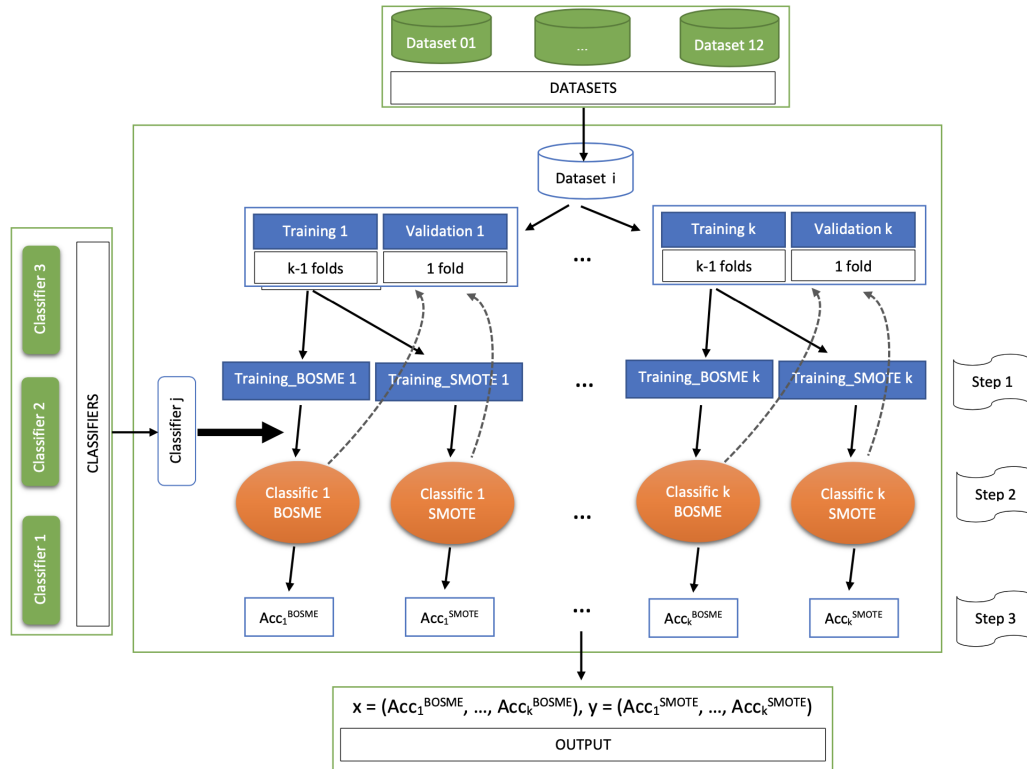


Fig 1.2. Testing proposed alternative: stage 1 out of 2, to compare BOSME with SMOTE (analogously for ROSE).

which the classifier is intended to be applied, that is, the purpose for which it is created and, in particular, on the nature of the data set on which it is based. We opted for the second possibility, in such a way that the second specific objective of this thesis consists in the creation/modification of a learning algorithm for a classifier, in our case oriented to the construction of a predictive model to assist medical professionals in decision-making at the Intensive Care Unit (ICU) of an hospital, from a data set provided by the Hospital de Mataró.

This data set contains information relating to 1,354 critical patients who had passed through the ICU of the hospital (that is, they had entered and left the ICU) from years 2016 and 2017, and has been used in this study to associate patients' syndromic evaluation result and the APACHE II score, among other features of predictive relevance chosen according to medical experts, with ICU mortality, as well as with destination at ICU discharge, or cause of death, as appropriate, using a supervised ML model that is built specifically for this purpose.

(i) Deficiency or problem

The ICU of the Hospital de Mataró maintains a register of admissions and discharges of all patients who have been treated in the unit. The volume of records (and the information associated with each record) justifies the need for an Expert System as a decision-making support tool. The implementation of the Expert System aims to cover three needs of the unit:

- improvement of early mortality prediction.
- help to make more efficient medical decision on patients at higher risk.
- evaluation of the efficacy of new treatments or detection of changes in clinical practice.

The traditional approach to mortality prediction, which is still the most widespread today, is based on APACHE (the Acute Physiology And Chronic Health Evaluation score), and its successive versions, from which APACHE II is the most used (see [11]). Specifically, the individual risk of death (probability of die) of a patient in the ICU is usually calculated by *Logistic Regression* as

$$\frac{e^{\text{logit}}}{1 + e^{\text{logit}}},$$

where *logit* is obtained from the following equation:

$$\begin{aligned} \text{logit} = & -3.517 + 0.146 \times \text{APACHE II} + 0.113 \text{ (only if sepsis at admission is present)} \\ & + 0.603 \text{ (only if the generic syndrome causing admission is urgent surgical)} \\ & + \text{coefficient } \beta \text{ of the main cause of admission,} \end{aligned} \tag{1.3}$$

where the main cause of admission and the corresponding coefficients β are given in Table 1.1 below, taking into account that only one cause of admission is considered to be the main one, and that the value of the β coefficients depends on whether the patient suffered from sepsis on admission, and whether the generic syndrome causing admission was from a surgical category (elective or urgent surgical) or not.

First, this approach, that we denote as the “APACHE II model”, presents significant limitations derived from the fact that the parameters of the Logistic Regression are not learned from the data at hand, but borrowed from a paper published in 1985:

- it does not incorporate variations between units or regions,
- it does not have good predictive behaviour in small populations,

Main cause of admission	Sepsis or non surgical category	No sepsis and surgical category
F ₄ : Acute Coronary Syndrome	-0.191	-0.797
F ₅ : Respiratory Failure	-0.890	-0.610
F ₆ : Shock	0.493	-0.797
F ₇ : Coma	-0.759	-1.150
F ₈ : Renal Failure	-0.885	-0.196
F ₉ : Hepatic Failure	0.501	-0.613
F ₁₀ : Cardio Respiratory Arrest	0.393	0.393
F ₁₁ : Elective Surgical		-0.248
F ₁₂ : Arrhythmia	-1.368	-0.797
F ₁₃ : Cranial Trauma	-0.517	-0.955
F ₁₄ : Other Trauma	-1.228	-1.684
F ₁₅ : Intoxication	-0.142	-0.196

Table 1.1. Main causes of admission to the ICU and the corresponding commonly used β coefficients. This table has been adapted to our setting from [11].

- it presents high stiffness as predictive model, since if the value of some of the key variables for the patient is not known, it cannot provide a prediction,

which make it highly unsatisfactory.

Second, from a Supervised Machine Learning point of view and focusing on the target output variable “Result”, which refers to ICU mortality, we observe that it is a binary class variable with $r = 2$ classes: “live” (85.3%) and “die” (14.7%), corresponding to how the patient leaves the ICU, with a notable unbalance which places us in the context of topic (a), with the problem of the bias towards the majority class that classifiers learned from an unbalanced data set will suffer. As we have already mentioned in Subsection 1.4.1, one possibility to address this problem consists in modifying the training data set previously to learning, by oversampling it with new instances artificially generated of the minority class (see [6]), and another, the one that concern us now, consists of modifying the classification algorithm itself, without modifying the training data set.

Following this way, of the possible classifiers whose learning algorithm we can modify to address the problem of bias towards the majority class we have chosen an **ensemble of classifiers**, which is a meta-algorithm that is constructed following one of the different possible *combination rules* to encompass two or more base classifiers. Indeed, ensemble learning of classifiers is a ML paradigm which consists of multiple classifiers that are trained to solve the same problem (in contrast to ordinary ML approaches which try to learn a single classifier from training data), and they are merged to build a classifier, which is the ensemble, by combining their predictions according to a combination rule, and in this way, we obtain a joint prediction that, in principle, should be better than that of the individual base classifiers that make up the ensemble.

Of the combination rules for ensembling, the best known is the Majority Vote (MV),

which is clearly the most used, with or without weights. However, any use of this rule, even if adapted to take into account the problem of unbalance in the distribution of the class variable, for example by choosing appropriate weights, presents a significant problem: it could well happen that the *Confidence Level* associated with the label (prediction) assigned by the ensemble, that is, the probability with which the ensemble assigns the label, be less than 0.5, which is completely counterintuitive.

We can see this better with some illustrative examples. We consider an ensemble formed by 5 base (probabilistic) classifiers, combined by means of the MV rule, that is, the final decision of the ensemble is the one that corresponds to the majority of votes. As each classifier has two possible votes (“live” and “die”), the label that the ensemble will assign to a patient will be the one assigned by 3 or more of the 5 base classifiers. Let denote by p_ℓ the probability assigned to “die” by the ℓ -th base classifier. Then, the probability with which the ensemble assigns the label “die” to the patient is given by the formula:

$$CL_{MV} = \prod_{\ell=1}^5 p_\ell + \sum_{j=1}^5 ((1 - p_j) \prod_{\substack{\ell=1 \\ \ell \neq j}}^5 p_\ell) + \sum_{j=1}^5 \sum_{\substack{k=1 \\ k \neq j}}^5 ((1 - p_j) (1 - p_k) \prod_{\substack{\ell=1 \\ \ell \neq j, k}}^5 p_\ell). \quad (1.4)$$

It is also very common to use the majority of votes with weights (Weighted Majority Vote, WMV) as a combination rule, that is, the ensemble assigns the label whose sum of weights of the base classifiers that assign that label is greater. To formalize this combination rule, we introduce for any class (label) j , the **discriminant** function $D_j = \sum_{i=1}^5 w_i d_{i,j}$ where $d_{i,j} = 1$ if classifier i assigns class j to the patient, and 0 otherwise, and w_i , $i = 1, \dots, 5$ are the weights of the five base classifiers, that is, D_j is the sum of weights corresponding to the base classifiers that assign the patient to class j . The inferred class for the given patient by the WMV classifier is taken to be the one that maximizes the discriminant function. (Note that with $w_i = w_j$ for all $i, j = 1, \dots, 5$, this rule corresponds to MV). In [7] we found the analogous of formula (1.4) for the WMV ensemble, and apply both to two particular examples, given in Tables 1.2 and 1.3.

Classifier	Weigths	Prob. of “die”	Prediction	Pred. MV	Pred. WMV
BC ₁	$w_1 = 0.25$	$p_1 = 0.55$	die	die (0.24731 < 0.5)	live (0.66236 > 0.5)
BC ₂	$w_2 = 0.10$	$p_2 = 0.55$	die		
BC ₃	$w_3 = 0.05$	$p_3 = 0.55$	die		
BC ₄	$w_4 = 0.30$	$p_4 = 0.10$	live		
BC ₅	$w_5 = 0.30$	$p_5 = 0.10$	live		

Table 1.2. Toy example 1. *Confidence level* for the prediction given by MV is < 0.5.

Classifier	Weights	Prob. of “die”	Prediction	Pred. MV	Pred. WMV
BC ₁	$w_1 = 0.25$	$p_1 = 0.95$	die	die (0.95324 > 0.5)	live (0.32725 < 0.5)
BC ₂	$w_2 = 0.10$	$p_2 = 0.95$	die		
BC ₃	$w_3 = 0.05$	$p_3 = 0.95$	die		
BC ₄	$w_4 = 0.30$	$p_4 = 0.45$	live		
BC ₅	$w_5 = 0.30$	$p_5 = 0.45$	live		

Table 1.3. Toy example 2. *Confidence level* for the prediction given by WMV is < 0.5 .

Note that in Table 1.2 the prediction for the MV ensemble would be “die” with a *Confidence Level* of $0.24731 < 0.5$. The same happens in Table 1.3 with the WMV ensemble, where the prediction of the ensemble is “live” with a *Confidence Level* of $0.32725 < 0.5$.

In short, neither the classical approach based on the use of the Logistic Regression from the APACHE II score with fixed coefficients, nor the approach based on the Supervised Machine Learning using an ensemble of classifiers as a meta-algorithm, with MV or WMV decision rules, have result to be completely satisfactory, since both present weaknesses.

(ii) Proposed alternative

Due to what was mentioned in the previous section, it seems necessary to develop an alternative to the known algorithms that

1. we introduce and justify from a theoretical point of view (*basic research*), and
2. is implemented with the dataset and shows to produce good results, improving the performance of the algorithms known from the state-of-the-art with which it is compared (*algorithmic/computing*).

The alternative that we propose is an **ensemble of classifiers** based on 5 base classifiers which are different types of Bayesian networks. It is necessary to define a strategy or combination rule to decide the prediction of the target output provided by the ensemble of classifiers, from the predictions given by the base classifiers, that take into account the unbalanced character of the data set.

The two examples of typical strategies already mentioned, that are Majority Vote (MV) and Weighted Majority Vote (WMV), are combination rules of the “fusion of labels” type, since they only need to know which are the predictions of any of the base classifiers. Other type of combination rules are that of “fusion of continuous-valued outputs”, which are not based on the predictions themselves but on the probabilities assigned to the classes by the base classifiers, as for example Ensemble Average (EA), that assigns the class that maximizes the average of the probabilities assigned to each class by the base classifiers, or its weighted version, named EWA (Ensemble Weighted

Average) on which we will focus specifically. Both have the advantage of being compatible with the MAP criterion, assigning a *Confidence Level* not less than 0.5 to the prediction (in the binary case, which is what concerns us), thus avoiding the problem posed by the MV and WMV ensembles.

Indeed, we can formalize both EA and EWA ensembles, similarly to how it was done with MV and WMV: fixed class (label) j , we introduce now the **discriminant** function $\tilde{D}_j = \sum_{i=1}^5 w_i \tilde{d}_{i,j}$ where $\tilde{d}_{i,j}$ is the probability with which classifier i assigns the class v_j to the patient, and w_i are the weights of the classifiers.

The inferred class by the EWA classifier is taken to be the one that maximizes the discriminant function \tilde{D} . Therefore, with this criterion, the *Confidence Level* associated to class v_j is \tilde{D}_j since $\tilde{D}_1 + \tilde{D}_2 = 1$, which implies compatibility with the **MAP criterion**. (Note that with $w_i = w_j = 1/5$ for all $i, j = 1, \dots, 5$, this rule corresponds to the simple mean combiner EA.)

For the examples in Tables 1.2 and 1.3, we can apply both the EA and the EWA, and obtain the respective predictions and *Confidence Level*. Indeed, for the example in Table 1.2,

$$\begin{aligned} \text{EA : } \quad \tilde{D}_1 &= \sum_{i=1}^5 \frac{p_i}{5} = 0.37 < \tilde{D}_2 = \sum_{i=1}^5 \frac{1-p_i}{5} = 0.63 \\ \text{EWA : } \quad \tilde{D}_1 &= \sum_{i=1}^5 \omega_i p_i = 0.28 < \tilde{D}_2 = \sum_{i=1}^5 \omega_i (1-p_i) = 0.72 \end{aligned}$$

which bring us to the “live” prediction with both classifiers, and respective *Confidence Level* of 0.63 and 0.72, both > 0.5 . Analogously, for the toy example in Table 1.3, both ensembles give as prediction “die”, with respective *Confidence Level* of 0.75 and 0.65, both > 0.5 . Although in these two examples the predictions of EA and EWA have coincided with each other, this does not necessarily have to happen in general.

In this way, with the ensembles EA or its weighted version EWA, we can avoid the problem that we have seen that both MV and WMV present of being counterintuitive in terms of the *Confidence Level* of the prediction they provide. As we have said before, our proposal is an EWA with adequate weights which, chosen appropriately, allow us to address the problem of topic (a) (that is to say, the bias towards the majority class caused by an unbalanced data set).

For the assignment of weights to the base classifiers, and bearing in mind that the combination of unbalanced data (14.7% “die” in variable *Result*) and a small sample size, we use the *Area Under the Precision-Recall curve* (AUPR), being the *Precision-Recall* (PR) curve that obtained by plotting *Precision* over *Recall*. The PR curve provides a more informative picture of the performance of the classifier than the Receiver Operator Characteristic (ROC) curve when dealing with highly skewed datasets,

as is our case. Let us explain this a bit more.

Both, *Precision* and *Recall* (or *Sensitivity*) are measures of the quality of a positive (minority) prediction. Instead, *Specificity* measures the quality of a negative (majority) prediction. Their definitions are:

- *Precision* is the number of true positives divided by the total number of positive predictions.

$$Precision = \frac{C_{11}}{C_{11} + C_{21}}$$

- *Recall* or *Sensitivity* is the number of true positives divided by the total number of positive observations.

$$Recall = \frac{C_{11}}{C_{11} + C_{12}}$$

- *False Positive Rate* (FPR) is the number of false positives (true negatives classified as positives) divided by the total number of negative observations.

$$FPR = \frac{C_{21}}{C_{21} + C_{22}}$$

Each one of the *Receiver Operator Characteristic* (ROC) curve and the *Precision-Recall* (PR) curve allow to visualize two metrics at the same time. The area under these curves is called, respectively, AUC (Area Under the ROC Curve) and AUPR (Area Under the Precision-Recall curve), and ranges between 0 and 1, being used for classifier validation as a behavioral metric. Their definitions are:

- ROC curve is the plot of *Sensitivity* (y-axis) versus FPR (x-axis).
- PR curve is the plot of *Precision* (y-axis) versus *Recall* (x-axis).

Finally, another performance metric that combines two other metrics and is commonly used is the F-score.

- *F-score* (also known as *F1-score*) is defined as the harmonic mean of *Precision* and *Recall*.

$$F - score = \frac{1}{\frac{1}{2} \left(\frac{1}{Precision} + \frac{1}{Recall} \right)} = \frac{2C_{11}}{2C_{11} + C_{12} + C_{21}}$$

Considering the above, we assign a weight w_i to the base classifier i , which is obtained from its estimated AUPR, denoted by $A_i \in [0, 1]$, in the following way:

$$w_i = \frac{h_i}{\sum_{j=1}^5 h_j}, \text{ where } h_i = \log \left(\frac{\frac{1}{2}(A_i + 1)}{1 - \frac{1}{2}(A_i + 1)} \right). \quad (1.5)$$

Note that

$$\frac{1}{2}(A_i + 1) \in [0.5, 1], \quad \text{and therefore, } \frac{\frac{1}{2}(A_i + 1)}{1 - \frac{1}{2}(A_i + 1)} \geq 1$$

and consequently $h_i \geq 0$. This transformation of the A_i 's is a dilatation since if $A_i < A_j$, therefore $h_j - h_i > A_j - A_i > 0$. With this assignment of weights, we magnify the relevance of the base classifiers using weights based on the AUPR metric.

A part from this, we propose a recalibration for the logistic regression of the ‘‘APACHE II method’’, where the coefficients are learned from data and not borrowed from [11] (see Table 1.1). We name this model ‘‘LR.APACHEII’’.

(iii) Testing the proposed alternative

Our EWA model, with the weights given by (1.5), has been tested against both the five Bayesian networks used as base classifiers, against three well known state-of-the-art classifiers (Neural Network NN, Random Forest RF, and Support Vector Machine SVM), and against three other ensembles constructed with different strategies as combination rules: the Majority Vote (MV), the Weighted Majority Vote (WMV) using the same weights as EWA, and the Ensemble Average (EA). In addition, we also compare it with the two models based on the APACHE II score that use Logistic Regression, the ‘‘APACHE II model’’, with fixed coefficients, and with the ‘‘LR.APACHEII’’, whose coefficients learned from the data set.

AUPR, *F-Score* and AUC are the different performance measures that have been used to evaluate the models and to compare them with each other. They are preferred for the performance assessment in the binary case to the most commonly used measure *Accuracy*. Specifically, AUPR is particularly well suited to address the situation of an unbalanced data set. For statistical significance studies, contrasts such as the paired t-test or Wilcoxon signed-rank test, both making Holm-Bonferroni adjustments for multiple comparisons, have been implemented, depending whether the samples of the performance measures obtained through the validation procedure (*k-fold cross-validation*), can be assumed to come from a Gaussian distribution or not.

1.4.3 Validation

The last step in the machine learning life cycle is the validation of the built model. With the performance measures we can evaluate different aspects of the predictions emitted by the classifiers, from the confusion matrices generated with the validation procedure. Once a classifier is built from a training data set (previously preprocessed, if needed), we use a performance measure to assess its behaviour and compare with other classifiers. In the binary case there are several classical measures, in addition to *Accuracy*, that are commonly used (see Section 1.4.2). Not of all these measures

can be extended to the multi-class case, how does *Accuracy* do it. Another well known performance measure, formerly introduced in the binary case but that extends without problems to the multi-class, is Matthew's Correlation Coefficient (MCC).

Matthews Correlation Coefficient (MCC) measures the correlation between the observed and the predicted classes. It was introduced for the binary case in [12] as the ϕ -coefficient, which is a measure of association obtained by discretization of the Pearson's correlation coefficient for two binary vectors. In [13] an extended correlation coefficient that applies to any number of categories is used to generalize the binary MCC to classification with $r > 2$ classes (see more details in [14]). MCC ranges in $[-1, +1]$, where $+1$ indicates perfect agreement (all the instances are well classified, and therefore matrix C is diagonal), -1 perfect disagreement (all the instances are misclassified, and then all the elements of the diagonal of C are zero), and 0 indicates no relationship (similar to a random classification). Its definition in the general multi-class setting is:

$$MCC = \frac{\sum_{k=1}^r \sum_{\ell=1}^r \sum_{m=1}^r (C_{kk} C_{\ell m} - C_{k\ell} C_{mk})}{\sqrt{\sum_{k=1}^r \left(\left(\sum_{\ell=1}^r C_{k\ell} \right) \left(\sum_{k'=1:k' \neq k}^r \sum_{\ell'=1}^r C_{k'\ell'} \right) \right)} \sqrt{\sum_{k=1}^r \left(\left(\sum_{\ell=1}^r C_{\ell k} \right) \left(\sum_{k'=1:k' \neq k}^r \sum_{\ell'=1}^r C_{\ell'k'} \right) \right)}}$$

In this section we are going to introduce the last work included in the thesis, which is related to topic (c). The idea is to concentrate on the study/improvement of a behavioral metric based on the Shannon's concept of **entropy** from the field of Information Theory. Entropy is originally defined as a measure of the molecular disorder, or randomness, of a physical system. That is, it is a measure of the uncertainty entailed by a random phenomenon. In our case, the random phenomenon will be the result of the classification process, reflected in a given confusion matrix.

Let X be a (discrete) random variable on a probability space (Ω, \mathcal{A}, P) , whose support is $S(X)$ and for any $x \in S(X)$, $p(x)$ denotes the probability assigned by X to x , that is, $p(x) = P(X = x)$. In Information Theory, the **(Shannon's) entropy** H associated to X , $H(X)$, is defined as the expected value of the self-information carried by X , which is: $I(X) = -\log_b(p(X))$, where usually the base of the logarithm is $b = 2$, although not necessarily. That is:

$$H(X) = E(I(X)) = \sum_{x \in S(X)} p(x) I(x) = - \sum_{x \in S(X)} p(x) \log_b(p(x)) .$$

In other words, $H(X)$ is the average level of uncertainty inherent to the possible outcomes of the variable. Given a set of non-negative numbers, say $\{n_1, \dots, n_s\}$, the

(Shannon's) entropy generated by the set is defined as $-\sum_{i=1}^s p_i \log_b(p_i)$, with $p_i = \frac{n_i}{n}$ if $n = \sum_{i=1}^s n_i$. That is, the **(Shannon's) entropy** generated by the set is $H(X)$ with X defined as the discrete random variable with support $S(X) = \{1, \dots, s\}$ and $p_i = P(X = i) = n_i/n$. Note that the minimum entropy (minimum uncertainty) corresponds to all the probability accumulated in a single point, that is, $p_\ell = 1$ for some $\ell = 1, \dots, s$ and the rest equal to 0, and in this case, entropy is 0. Maximum entropy (maximum uncertainty) corresponds to the uniform distribution of probability, with $p_1 = \dots = p_s = 1/s$, where entropy achieves the value $\log_b(s)$, which is:

$$\log_b(s) = \begin{cases} < 1 & \text{if } b > s \\ = 1 & \text{if } b = s \\ > 1 & \text{if } b < s \end{cases}$$

In terms of Machine Learning, and more specifically applied to a confusion matrix, we can use the entropy to measure the degree of disorder in the classification.

Now we introduce the Confusion Entropy (CEN), which is a performance metric in the general multi-class setting based in the concept of Shannon's entropy that has been introduced in [15]. Given a general confusion $r \times r$ matrix C as in (1.2), in [15] the misclassification probability of classifying class- v_i cases as being of class v_j "subject to class v_j ", denoted by $P_{i,j}^j$, is introduced as:

$$P_{i,j}^j = \frac{C_{i,j}}{\sum_{k=1}^r (C_{j,k} + C_{k,j})}, \quad i, j = 1, \dots, r, i \neq j, \quad (1.6)$$

that is, $P_{i,j}^j$ is introduced as the relative frequency of class- v_i cases that are classified as being of class v_j among all cases that are of class v_j or that have been classified as being of class v_j . But it is not really, not exactly. The reason is that class- v_j cases that have been correctly classified, whose number is $C_{j,j}$, are counted twice in the denominator.

Analogously, the misclassification probability of classifying class- v_i cases as being of class v_j "subject to class v_i ", with analogous interpretation, denoted by $P_{i,j}^i$, is defined in the same paper by:

$$P_{i,j}^i = \frac{C_{i,j}}{\sum_{k=1}^r (C_{i,k} + C_{k,i})}, \quad i, j = 1, \dots, r, i \neq j. \quad (1.7)$$

The Confusion Entropy associated to class v_j is defined by:

$$\text{CEN}_j = - \sum_{k=1, k \neq j}^r \left(P_{j,k}^j \log_{2(N-1)}(P_{j,k}^j) + P_{k,j}^j \log_{2(N-1)}(P_{k,j}^j) \right) \quad (1.8)$$

with the convention $a \log_b(a) = 0$ if $a = 0$. Finally, the overall Confusion Entropy associated to the confusion matrix C is defined as a convex combination of the Confusion Entropy of the classes as follows:

$$\text{CEN} = \sum_{j=1}^r P_j \text{CEN}_j, \quad (1.9)$$

where the non-negative weights P_j , summing up to 1, are defined by:

$$P_j = \frac{\sum_{k=1}^r (C_{j,k} + C_{k,j})}{2 \sum_{k,\ell=1}^r C_{k,\ell}}. \quad (1.10)$$

(i) Deficiency or problem

This definition of CEN presents kind of problems. On the one hand, from a technical perspective, while for $r > 2$ (multi-class setting), CEN ranges between 0 and 1 (indeed, 0 is attained with perfect classification, that is, the off-diagonal elements of matrix C being zero, and 1 under complete misclassification, symmetry and balance in C , that is, if all diagonal elements in C are zero, and the off-diagonal elements take all the same value), in the binary case ($r = 2$), although CEN remains to be 0 with perfect classification, and is 1 under complete misclassification with symmetry, in intermediate scenarios we can also obtain $\text{CEN} = 1$ and even higher values. That is, in some cases CEN is “out-of-range”, if what is intended is to introduce a measure ranging from 0 to 1. See, for example, the confusion matrices in Table 1.4.

	$\begin{pmatrix} 6 & 0 \\ 0 & 6 \end{pmatrix}$	$\begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix}$	$\begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix}$	$\begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$	$\begin{pmatrix} 2 & 4 \\ 4 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 5 \\ 5 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 6 \\ 6 & 0 \end{pmatrix}$
CEN =	0.0000	0.5975	0.8617	1.0000	1.0566	1.0525	1.0000

Table 1.4. CEN: examples in the symmetric and balanced binary setting.

In addition, as can be seen in Table 1.4, there is no monotonicity in the behaviour of CEN when the situation monotonously goes from perfect classification to completely imperfect classification in the symmetric and balanced binary setting. That is, the

value of CEN first increases, as expected, but after a certain moment it decreases, which is totally counterintuitive. This fact is clearly a weakness of this metric.

On the other hand, from the point of view of its meaning, some of the quantities that appear in the CEN definition, $P_{i,j}^i$ (1.6) and $P_{i,j}^j$ (1.7), are interpreted by their authors as relative frequencies that in reality are not, as we have already mentioned.

Now we will explain the modifications in the definition of CEN that we have introduced (denoted by MCEN) in [5], which allows us to address both types of problems at the same time: on the one hand, we avoid its undesired behaviour, and on the other hand, we further fine-tune the formal definition of the terms that make it up to the meaning that they should have.

(ii) Proposed alternative

Instead of (1.6), we propose to introduce the probability of classifying class- v_i cases as being of class v_j “subject to class v_j ”, as

$$\tilde{P}_{i,j}^j = \frac{C_{i,j}}{\sum_{k=1}^r (C_{j,k} + C_{k,j}) - C_{j,j}}, \quad i, j = 1, \dots, r, i \neq j. \quad (1.11)$$

that is, we overcome the fact that in (1.6) correctly classified class- v_j cases are counted twice in the denominator. With this definition, $\tilde{P}_{i,j}^j$ is really the relative frequency of class- v_i cases classified as belonging to class v_j among all cases that are of class v_j or that have been classified as of class v_j . Analogously, we modify definition (1.7) in the same sense:

$$\tilde{P}_{i,j}^i = \frac{C_{i,j}}{\sum_{k=1}^r (C_{i,k} + C_{k,i}) - C_{i,i}}, \quad i, j = 1, \dots, r, i \neq j, \quad (1.12)$$

and $\tilde{P}_{i,j}^i$ is really the relative frequency of class- v_i cases classified in class v_j among all cases that are of class v_i or that have been classified as being of class v_i . With definitions (1.11) and (1.12) we solve the problem of meaning in the definition posed by CEN in (1.6) and (1.7), respectively. Next, with the aim of solving the problem of the inadequate behaviour of the CEN metric mentioned above, we modify definition of the weights in (1.10) in the following way:

$$\tilde{P}_j = \frac{\sum_{k=1}^r (C_{j,k} + C_{k,j}) - C_{j,j}}{2 \sum_{k,\ell=1}^r C_{k,\ell} - \alpha \sum_{k=1}^r C_{k,k}}, \quad \text{where } \alpha = \begin{cases} 1/2 & \text{if } r = 2 \\ 1 & \text{if } r > 2. \end{cases}$$

With respect to (1.10), \tilde{P}_j avoids the repetition of the diagonal elements, which in P_j are counted twice, both in the numerator and in the denominator, if $r > 2$. In the binary case $r = 2$, an adjustment must be made so that the measure we define below meets the technical conditions required to it. Note that when $r > 2$, $\sum_{j=1}^r \tilde{P}_j = 1$, so the modified overall Confusion Entropy is also defined as a convex combination of the modified Confusion Entropy corresponding to the classes, while in the binary case ($r = 2$), it is just defined as a conical combination since although the weights \tilde{P}_j are non-negative, they do not necessarily sum up to 1 (indeed, their sum is 1 if and only if all the diagonal elements of the confusion matrix C are zero, that is, if all cases have been misclassified).

We define the (modified) Confusion Entropy associated to class v_j as in (1.8) by

$$\text{MCEN}_j = - \sum_{k=1, k \neq j}^r \left(\tilde{P}_{j,k}^j \log_{2(N-1)}(\tilde{P}_{j,k}^j) + \tilde{P}_{k,j}^j \log_{2(N-1)}(\tilde{P}_{k,j}^j) \right),$$

and the Modified Confusion Entropy (MCEN) as in formula (1.9), that is,

$$\text{MCEN} = \sum_{j=1}^r \tilde{P}_j \text{MCEN}_j. \quad (1.13)$$

(iii) Testing the proposed alternative

The proposed measure MCEN has been compared with the metric CEN as well as with other metrics using different families of confusion matrices, among which there are *Accuracy* (ACC) and *Matthews Correlation Coefficient* (MCC). Although the *Accuracy* is between 0 (null hit rate) and 1 (total hit rate) and the MCC is between -1 (total disagreement or total misclassification) and +1 (perfect prediction), for this work we have made a modification (indeed, a change of scale) in order to facilitate the illustration of the results in tables and graphs. Since MCEN indicates with the value 0 perfect classification and with 1 perfect misclassification, we are going to modify *Accuracy* and MCC to fit that range of values $[0, 1]$, in such a way that $\text{ACC}^* = 1 - \text{Accuracy}$ will indicate with a 0 a hit rate total and with a 1 a zero success rate. Similarly, $\text{MCC}^* = \frac{1 - \text{MCC}}{2}$ will indicate 0 as perfect prediction and 1 as total misclassification.

As we can see in the examples in Table 1.5, unlike what happens with CEN, MCEN (and also ACC^* and MCC^* , which coincide in this case) shows the desired behaviour. Indeed, MCEN does not go above the desired maximum of 1, and it also distinguishes between uniform distribution of instances (4th matrix) and perfect misclassification (last matrix), which is something that CEN does not do.

	$\begin{pmatrix} 6 & 0 \\ 0 & 6 \end{pmatrix}$	$\begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix}$	$\begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix}$	$\begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$	$\begin{pmatrix} 2 & 4 \\ 4 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 5 \\ 5 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 6 \\ 6 & 0 \end{pmatrix}$
ACC* = MCC* =	0.0000	0.1667	0.3333	0.5000	0.6667	0.8333	1.0000
CEN =	0.0000	0.5975	0.8617	1.0000	1.0566	1.0525	1.0000
MCEN =	0.0000	0.5910	0.8000	0.9057	0.9614	0.9891	1.0000

Table 1.5. CEN vs MCEN, ACC* and MCC*: examples in the symmetric and balanced binary setting.

- From a theoretical perspective (basic research).

We have made a comparative study with different families of confusion matrices starting with the perfectly symmetric and balanced case, with $r \geq 2$. We consider the case in which $C_{i,j} = F$ for all $i, j = 1, \dots, r, i \neq j$ and $C_{i,i} = T$, with

$$T \geq 0, F > 0, \text{ that is, } C = \begin{pmatrix} T & F & \dots & F & F \\ F & T & \dots & F & F \\ \vdots & \vdots & \dots & \vdots & \vdots \\ F & F & \dots & T & F \\ F & F & \dots & F & T \end{pmatrix}.$$

The comparative study collects the results of the metrics ACC*, MCC*, CEN, and MCEN according to $\gamma = T/F$.

We proceed in the same way with other families. Table 1.6 shows studied families, including the family mentioned above.

$\begin{pmatrix} T & F & \dots & F & F \\ F & T & \dots & F & F \\ \vdots & \vdots & \dots & \vdots & \vdots \\ F & F & \dots & T & F \\ F & F & \dots & F & T \end{pmatrix}$	$\begin{pmatrix} 1 & A \\ A & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & A \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} A & rA \\ rA & 1 \end{pmatrix}$
Symmetric case $N \geq 2$	U_A family	V_A family	$X_{A,r}$ family
$\begin{pmatrix} rA & rA \\ A & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & \dots & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & \dots & \dots & \dots \\ A & 1 & \dots & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 50 \\ A & 1 \end{pmatrix}$	$\begin{pmatrix} 50 & 1 \\ 1 & A \end{pmatrix}$
$Y_{A,r}$ family	Z_A family for $N \geq 2$	M_A family	W_A family

Table 1.6. Families of confusion matrices used to test the proposed alternative MCEN, from a theoretical point of view.

- From an experimental perspective (algorithmic/computing).

To help clarify the utility of MCEN in the evaluation of improvements in classification of the minority class while maintaining the same amount of imbalance, we consider two different examples.

Example 1: Family of confusion matrices $X_{50,2}^\alpha = \begin{pmatrix} 50 & 100 \\ 101 - \alpha & \alpha \end{pmatrix}$, with $\alpha = 1, 2, \dots, 101$. Note that when $\alpha = 1$, the corresponding matrix belongs to the family $X_{A,r}$ (Table 1.6) with $A = 50$ and $r = 2$. Imbalance in classes stays fix, with 150 cases of the majority class, and 101 of the minority. When $\alpha = 1$, the minority class is classified very badly, improving classification as α increases and reaching the perfect classification when $\alpha = 101$. MCEN is able to detect this behaviour. Unlike what happens with CEN, MCEN (as well as ACC* and MCC*) monotonically decreases when classification of the minority class improves (α increases). CEN incongruously first increases up to $\alpha = 18$ and then starts to decrease and behave like the other performance measures.

Example 2: A similar phenomenon can be observed with family $Y_{100,1}^\beta = \begin{pmatrix} 100 & 100 \\ 101 - \beta & \beta \end{pmatrix}$, with $\beta = 1, 2, \dots, 101$. With $\beta = 1$ the corresponding matrix belongs to the family $Y_{A,r}$ with $A = 100$ and $r = 1$. As in Example 1, imbalance in classes is constant and when $\beta = 1$, the minority class is classified very badly, improving classification as β increases up to 101, when perfect classification is reached. MCEN as well as ACC* and MCC*, monotonically decrease when β increases, while CEN increases up to $\beta = 14$ and then starts to decrease and behave like the other performance measures.

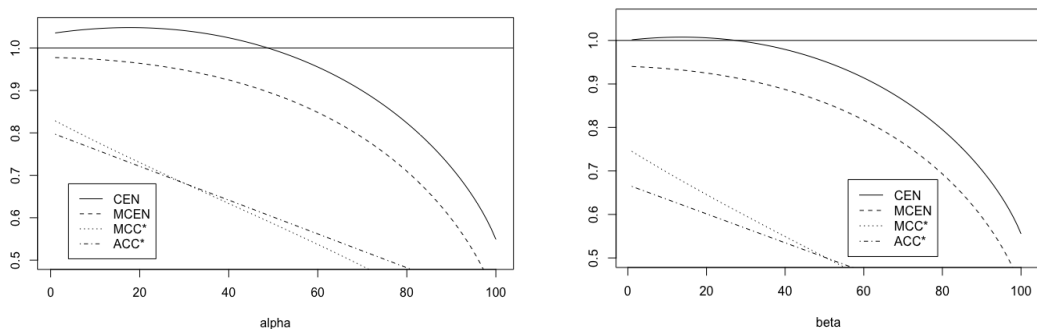


Fig 1.3. CEN, MCEN, ACC* and MCC* for families $X_{50,2}^\alpha$ (left) and $Y_{50,2}^\beta$ (right).

Different comparative works have been carried out between different performance measures, some of them based on the definition of Shannon's entropy. As examples, we have considered the metrics PACC (Probabilistic Accuracy) and NIT (Normalized Information Transfer) introduced by [16] and [17], respectively. Those two metrics have been rescaled to facilitate comparison with the rest of the metrics considered, and redefined as $\text{PACC}^* = 1 - \text{PACC}$ and $\text{NIT}^* = 1/\text{NIT}$. As we can see in the examples in Tables 1.7 and 1.8, MCEN is a better measure of the entropy associated with a confusion matrix than these two metrics.

	Baseline	(a)			(b)		
	$\begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$	$\begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}$	$\begin{pmatrix} 1 & 3 \\ 3 & 5 \end{pmatrix}$	$\begin{pmatrix} 0 & 3 \\ 3 & 6 \end{pmatrix}$	$\begin{pmatrix} 3 & 2 \\ 4 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 1 \\ 5 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 0 \\ 6 & 3 \end{pmatrix}$
ACC*=	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
MCC*=	0.5000	0.5130	0.5625	0.6667	0.4881	0.4375	0.3333
CEN =	1.0000	0.9898	0.9575	0.8962	0.9591	0.8250	0.5000
MCEN =	0.9057	0.9006	0.8848	0.8571	0.8590	0.7057	0.3343
PACC*=	0.5000	0.5071	0.5312	0.5833	0.4929	0.4687	0.4167
NIT* =	2.0000	1.9992	1.9840	1.8371	1.9992	1.9840	1.8371

Table 1.7. Toy example adding PACC and the NIT factor.

	$A = \begin{pmatrix} 10 & 0 \\ 10 & 10 \end{pmatrix}$	$B = \begin{pmatrix} 0 & 10 \\ 10 & 10 \end{pmatrix}$	$C = \begin{pmatrix} 10 & 0 & 0 \\ 10 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix}$	$D = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 10 & 10 \\ 10 & 0 & 0 \end{pmatrix}$		
ACC*=	0.3333	<	0.6667	0.2500	<	0.5000
MCC*=	0.2500	<	0.7500	0.1500	<	0.3500
CEN =	0.5283	<	1.0000	0.1981	<	0.3231
MCEN =	0.4000	<	0.9400	0.2000	<	0.3333
PACC*=	0.2917	<	0.7083	0.1944	<	0.5000
NIT* =	1.6799	=	1.6799	1.5000	=	1.5000

Table 1.8. Two toy examples. With $N = 30$ for $r = 2$, and with $N = 40$ for $r = 3$.

Finally, we introduce notations $\text{OUT}(C)$ and $\text{IN}(C)$, respectively, to denote the Shannon's entropy generated by the elements of outside and inside the main diagonal of a confusion matrix C . While IN is the entropy generated by the number of well classified cases of any of the classes, OUT is generated by the number of misclassified cases for each combination of observed and predicted classes. The objective of introducing both is to better understand the different behaviour of CEN and MCEN, since the two measures are defined from Shannon's entropy. We compare them in the examples of Table 1.9.

In Table 1.9 the baseline confusion matrix is constant with all its entries equal to 3. First, maintaining the total sum equal to $N = 12$ and the out-diagonal invariant, we reduce the entropy IN in (a). In the baseline case, the diagonal elements are the

	Baseline	(a)			(b)		
	$\begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$	$\begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}$	$\begin{pmatrix} 1 & 3 \\ 3 & 5 \end{pmatrix}$	$\begin{pmatrix} 0 & 3 \\ 3 & 6 \end{pmatrix}$	$\begin{pmatrix} 3 & 2 \\ 4 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 1 \\ 5 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 0 \\ 6 & 3 \end{pmatrix}$
Entropy=	1.0000	0.9183 (8.17%)	0.6500 (35.00%)	0.0000 (100.00%)	0.9183 (8.17%)	0.6500 (35.00%)	0.0000 (100.00%)
ACC*=	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
MCC*=	0.5000	0.5130	0.5625	0.6667	0.4881	0.4375	0.3333
CEN =	1.0000	0.9898 (1.02%)	0.9575 (4.25%)	0.8962 (10.38%)	0.9591 (4.09%)	0.8250 (17.50%)	0.5000 (50.00%)
MCEN =	0.9057	0.9006 (0.56%)	0.8848 (2.31%)	0.8571 (5.37%)	0.8590 (5.16%)	0.7057 (22.08%)	0.3343 (63.09%)

Table 1.9. Example: binary case with $N = 12$. (a): Entropy reduction within the main diagonal, IN. (b) Entropy reduction outside the main diagonal, OUT. In brackets the relative reduction with respect to the baseline case. Entropy refers to IN in (a) and to OUT in (b).

set $\{3, 3\}$, whose entropy is 1 (maximum value). The corresponding values of IN in case (a) are in Table 1.9, in a decreasing order. Analogously for (b) with changes introduced outside the main diagonal. While ACC* remains insensitive to changes in the arrangement of the elements of the matrix, since the sum of the main diagonal is constant, MCC* only decreases with decreasing entropy OUT, while when IN decreases, its value increases. As far as their interpretation is concerned, both CEN and MCEN measure the overall entropy of the confusion matrix, giving less weight to the IN entropy (the generated by the well classified cases) than to OUT entropy, corresponding to misclassification. In this example we observe how their values are reduced when IN decreases, maintaining its constant sum, or when the one that is reduced is OUT, but in this second case the reduction is much more drastic, both for CEN and MCEN, and more sharply for the second. The main difference between CEN and MCEN in this sense is that the former is more sensitive to changes of IN entropy than MCEN, while less than CEN to that of OUT (observe the percentages in brackets in Table 1.9, which are the relative reduction in the measure with respect to that of the baseline case).

Chapter 2

Overall Results

Along this chapter we will present a global vision of the results obtained, focusing on those that best represent our main contribution to the state-of-the-art. We leave the deepening and details of the theoretical and experimental work of each of the parts that make up this thesis for reading in the papers themselves. The chapter is organized as follows: Section 2.1 is devoted to the work done in the framework of the **Preprocessing** phase of the life cycle of Supervised Machine Learning, specifically regarding a new method of oversampling (BOSME). Section 2.2 treats the work done on the **Building-the-model** phase, with the introduction of the Ensemble Weighted Average (EWA) constructed from the data set of ICU patients. Finally, Section 2.3 treats the research carried out during the completion of the thesis on the **Validation** phase, studying the performance metrics defined from the Shannon’s Entropy as measure of the disorder, specifically by introducing the MCEN metric, which is a modification of the well-known CEN measure.

2.1 Preprocessing (BOSME)

In this section we present the paper *“Bayesian network-based over-sampling method (BOSME) with application to indirect cost-sensitive learning”* [6].

The fundamental contribution of this work is the proposal of the novel oversampling method BOSME based on a new paradigm consisting in building a model (Bayesian network) for the joint probability distribution of the input variables X_1, \dots, X_n in the binary case, when the class is the minority class, that is, learned from the data subset corresponding to $V = “+”$. To get a model as adjusted as possible to the data (of the minority class), the Bayesian network is built by learning the structure in such a way that the likelihood of the model associated to the data subset is maximized. The new artificial instances are generated randomly from the model. Using this paradigm we avoid the use of the concept of distance, which is key for the usual oversampling

methods SMOTE and ROSE. We have empirically shown that our method is especially suitable for datasets with categorical features, although it also can be used with discrete and/or continuous features.

In addition to the BOSME method, we have built a form of **wrapper** based on the costs of misclassification, such that through the wrapper any oversampling method, in particular BOSME, can serve to the purpose of the indirect cost-sensitive learning, as explained in Section 1.4.1. With the wrapper we can calculate the desired final proportion of the minority class in the enlarged dataset, once the synthetic instances generated using the oversampling method have been added.

2.1.1 BOSME as oversampling method

Denote by S the original (unbalanced) training data set, with M instances. Let m_+ be the number of instances corresponding to the minority (positive) class in S , and m_- that of the majority class. The original distribution of the class variable V in S is therefore (p_+, p_-) , where $p_+ = \frac{m_+}{M}$ and $p_- = \frac{m_-}{M}$. BOSME is designed to generate a number of artificial new instances, say n , of the minority class, such that in the enlarged training data set augmented with the synthetically generated instances, denoted by \tilde{S} , the minority class represents a desired proportion q of the total. We can see that in order to achieve the desired proportion q , n must be:

$$n = \text{round}\left(\frac{qM - m_+}{1 - q}\right) \quad (2.1)$$

The steps of the BOSME over-sampling method can be seen schematically in Figure 2.1.

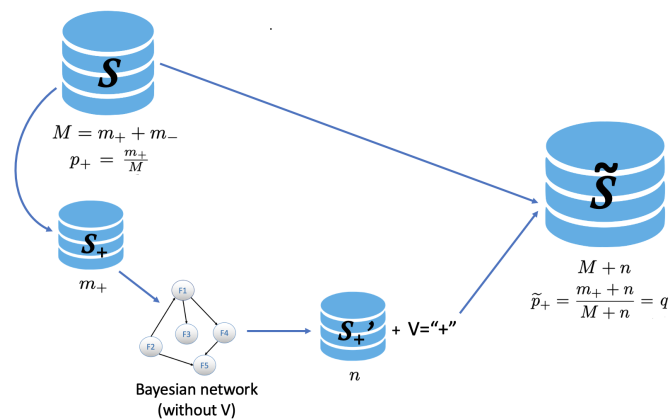


Fig 2.1. Graphical scheme of the BOSME algorithm.

The algorithm used to simulate the new instances generated randomly following the joint probability distribution entailed by the Bayesian network, is (probabilistic)

`logic sampling` (from `rbn` function implemented in the `bnlearn` package). This algorithm is introduced in [18] and is the first one applying stochastic sampling to simulation with Bayesian networks. The algorithm consists in generate values for the root nodes sampling from their (unconditional) distribution, then those of their children using their conditional distribution conditioned to the values already generated for their respective parent, and so on, this is done iteratively until values have been sampled for all nodes.

2.1.2 The wrapper: cost-sensitive approach

We also design a wrapper to determine the desired proportion q that must represent the minority class in the enlarged data set \tilde{S} . For that, associated to the classification of a generic instance, in the context of cost-sensitive approach, we can define the random variable `Cost`:

$$\text{Cost} = \begin{cases} c_+ & \text{if instance is positive but classified as negative} \\ c_- & \text{if instance is negative but classified as positive} \\ 0 & \text{otherwise} \end{cases}$$

whose expectation is

$$\begin{aligned} \text{Expected Cost} &= c_+ P(\text{instance is positive but classified as negative}) \\ &\quad + c_- P(\text{instance is negative but classified as positive}) \end{aligned}$$

If approximation were cost insensitive, the variable `Cost` and the Expected Cost would be respectively the variable `Error` and the Expected Error, just replacing c_+ and c_- by 1. We assume that $c_+ > c_-$, and denote by $\gamma = \frac{c_+}{c_-} > 1$ the **cost rate**.

We use a **Folk Theorem** (*Translation Theorem 2.1* [19]) to determine the proper proportion q . In fact, this result indicates how to modify the data set to reflect the misclassification costs optimally: if we modify the distribution of the class variable V on the data set to a new one, say $(\tilde{p}_+, \tilde{p}_-)$, multiplying any of the components of the original distribution (p_+, p_-) by a constant proportional to the associated misclassification costs, the resulting distribution has the following property: choosing the classifier that minimizes the Expected Error

$$\begin{aligned} \text{Expected Error} &= P(\text{instance is positive but classified as negative}) \\ &\quad + P(\text{instance is negative but classified as positive}) \end{aligned}$$

under the new distribution is equivalent to choosing the classifier that minimizes the Expected Cost under the original distribution.

The cited Theorem is formally stated and proved in [19] in a more general context than ours. We give an intuitive idea of the proof of the theorem in our context and try to find the transformation of the original distribution to the new distribution that suits us: consider a probabilistic classifier learned from \tilde{S} . Given a new instance, if the classifier assigns it to the positive class, the Expected Cost (with respect to the distribution of the class variable in S) is:

$$\begin{aligned} \text{Expected Cost}^+ &= c_+ P^+(\text{instance is positive but classified as negative}) \\ &+ c_- P^+(\text{instance is negative but classified as positive}) \\ &= c_+ \times 0 + c_- P(\text{instance is negative in } S) = c_- p_- \end{aligned}$$

(where the superscript $+$ indicates we are conditioning to the fact that the instance is assigned to the positive class). Similarly, if the classifier assigns the new instance to the negative class, the expected cost is $\text{Expected Cost}^- = c_+ p_+$. Analogously, the corresponding values for the Expected Error, with respect to the distribution of the class variable in \tilde{S} , are (by replacing c_+ and c_- by 1), are:

$$\text{Expected Error}^+ = \tilde{p}_-, \quad \text{Expected Error}^- = \tilde{p}_+$$

Then, minimizing the Expected Cost is equivalent to minimize the Expected Error provided that

$$\tilde{p}_+ = C p_+ c_+ \quad \text{and} \quad \tilde{p}_- = C p_- c_-$$

for some constant $C > 0$. Since \tilde{p}_+ and \tilde{p}_- must add up to 1, we obtain that the constant necessarily has to be $C = 1/(p_+ c_+ + p_- c_-)$. Therefore,

$$\tilde{p}_+ = \frac{p_+ c_+}{p_+ c_+ + p_- c_-} \quad \text{and} \quad \tilde{p}_- = \frac{p_- c_-}{p_+ c_+ + p_- c_-}$$

That is, the value for the proportion q must be

$$q = \tilde{p}_+ = \frac{m_+ \gamma}{m_+ \gamma + m_-} \quad (2.2)$$

with $\gamma = \frac{c_+}{c_-}$ the *cost rate*, showing the functional dependence of q on the initial number of instances of each class and on the misclassification costs through γ .

2.1.3 Results

Once the first stage of the experimental work has been completed (see Figure 1.2), we continue with the second stage to analyze the results through different statistical hypothesis tests. First, we test whether we can assume normality. Next, we compare the means (or medians, as appropriate) of paired samples to assess whether there are

statistically significant differences (BOSME vs. SMOTE or BOSME vs. ROSE) using Student's t-test if we have not rejected the null hypothesis in the normality test, or a Wilcoxon test otherwise. Provided that the contrast indicates that the difference of means, or medians, is statistically significant, we will add a mark in favor of BOSME when BOSME wins to SMOTE (or ROSE), while we will add a mark in favor of its opponent if just the opposite happens. We repeat the described process 10 runs as explained in Figure 2.2.

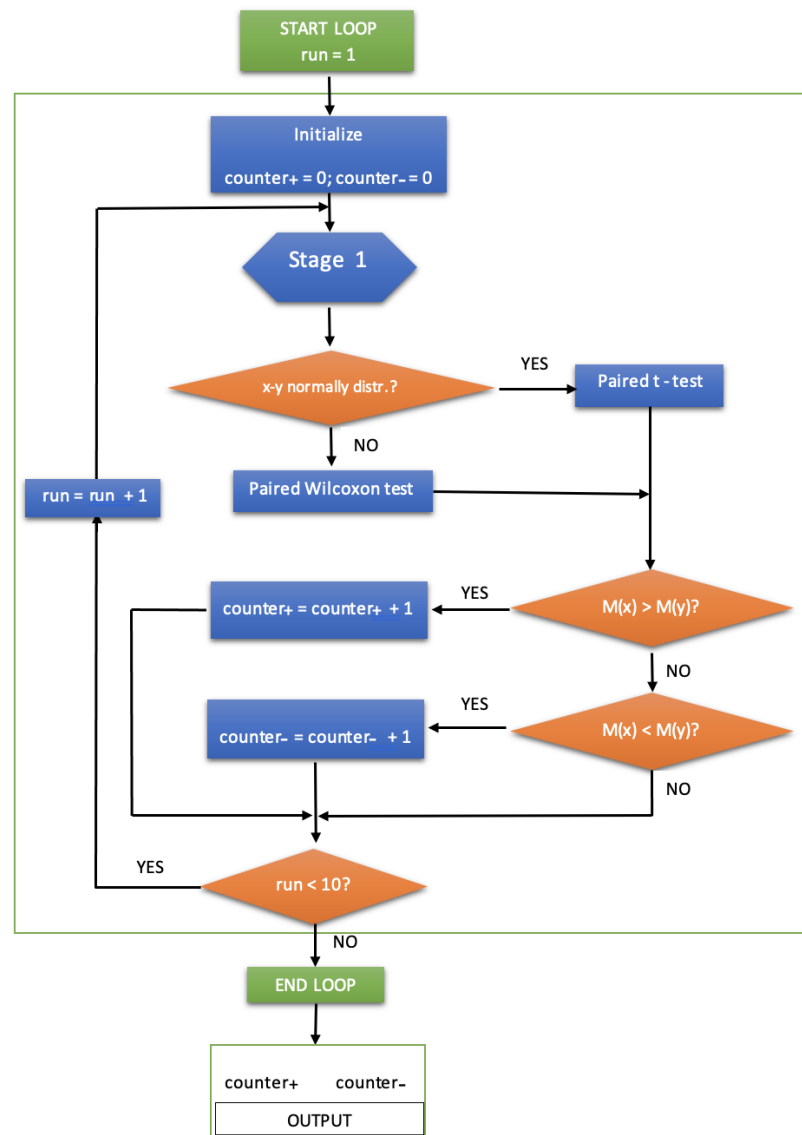


Fig 2.2. Comparing BOSME with the others oversampling methods. $M(\cdot)$ denotes the mean or median, as appropriate.

As mentioned in the Subsection 2.1.2, $\gamma = \frac{c_+}{c_-}$ denotes the **cost rate**. In the experimental phase we set γ values from 5 to 50 with incremental step of 5. Table

2.1 below shows the obtained results for $\gamma = 10$, for each database that has returned significant results, and for each classifier, in the comparison BOSME vs SMOTE.

$\gamma = 10$	SVM	RF	LR
Car eval.	+10 0.00098***		+7 0.00781
Spect heart	+6 0.01563*		
Balance	+10 0.00098***	+10 0.00098***	
Monks		+6 0.01563*	
Post-oper.	+9 0.00195**		
Tic-tac-toe	+10 0.00098***	+10 0.00098***	+10 0.00098***
Solar flare	+8/-1 0.01758	+7 0.00781**	
Breast	+10 0.00098***	+10 0.00098***	
Pizza	+10 0.00098***		
Haberman	/ -9 0.00195**	+7 0.00781**	
Saheart	/ -7 0.00781**		
Happiness			

Table 2.1. Number of runs (of the possible 10) for which there is statistical evidence in favor of BOSME (positive number, *counter*₊) or SMOTE (negative in red, *counter*₋), and the corresponding exact Binomial p-value, for any data set with significant differences, and classifier: Support Vector Machine (SVM), Random Forest (RF) and Logistic Regression (LR). $\gamma = 10$.

If we take as an example the results in Table 2.1, the Support Vector Machine classifier and `Spect heart` database, +6 represents that BOSME has been better than SMOTE a total of 6 times out of 10 possible. The value 0.01563* is the p-value obtained from the exact Binomial test which give us the probability of observe that BOSME wins SMOTE a total of 6 times of the 6 times we observe statistically significant differences, if indeed, there are no differences between the two oversampling methods. That is,

$$\text{p-value} = P(B(n = 6, p = 0.5) = 6) = \binom{6}{6} \left(\frac{1}{2}\right)^6 \left(\frac{1}{2}\right)^0 = 0.01563***$$

In Table 2.2 we summarize the information given in Table 2.1, showing for each dataset with how many classifiers BOSME has been significantly better than SMOTE (in black) and how many classifiers SMOTE has been significantly better than BOSME (in red). Since there are only three classifiers, the range of possible values varies from -3 to +3. We also introduce the measure β -score which is the difference. We observe that BOSME has been significantly better 15 times while SMOTE only has been significantly better 2 times. The difference is +13, in favor of BOSME.

γ	5	10	15	20	25	30	35	40	45	50
Car eval.		+2	+2	+3	+3	+3	+3	+3	+3	+3
Spect heart	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
Balance	+1	+2	+2 /-1	+2 /-1	+2	+2	+2	+2	+2	+2
Monks		+1	+1	+2	+2	+2	+2	+2	+2	+2
Post-oper.	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
Tic-tac-toe	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3
Solar flare		+2	+2	+1	+1	+1	+1	+2	+2	+2
Breast	+2	+2	+2	+2	+2	+2	+3	+2	+2	+3
Pizza				+1	+2	+2	+2	+2	+2	+2
Haberman	+1 /-1	+1 /-1	+1 /-1	+1 /-1	+1 /-1	+1 /-1	+1 /-1	+1 /-1	+1 /-1	+1 /-1
Saheart		/-1	/-1	/-1	/-3	/-2	/-3	/-3	/-3	/-3
Happiness					+1	+1	+1	+1	+1	+1
β -score	+8	+13	+12	+14	+15	+16	+16	+16	+16	+17

Table 2.2. Summary of the results, by data set. The numbers in the boxes indicate for how many classifiers, of the possible 3, there is statistical significance in favor of BOSME (positive) or in favor of SMOTE (negative, in red). For each γ , we take count of the β -score.

Analogously, Table 2.3 also summarizes the information for each γ and classifier, recording with how many datasets (of the 12 considered) BOSME has been significantly better than SMOTE (in black) and with how many datasets SMOTE has been significantly better than BOSME (in red). Since there are 12 data sets, the range of possible values varies from -12 to +12. For example, for $\gamma = 10$ and Support Vector Machine, there have been significant results in favor of BOSME in 7 databases and in favor of SMOTE in 2 databases.

γ	5	10	15	20	25	30	35	40	45	50
SVM	+5	+7/-2	+7/-2	+8/-2	+8/-2	+8/-2	+8/-2	+9/-2	+9/-2	+9/-2
RF	+3	+6	+6	+7	+8/-1	+8/-1	+8/-1	+8/-1	+8/-1	+8/-1
LR	+1/-1	+2	+2/-1	+2/-1	+3 /-1	+3	+4/-1	+3/-1	+3/-1	+4/-1

Table 2.3. Summary of the results, by classifier. The numbers in the boxes indicate for how many data sets, of the possible 12, there is statistical significance in favor of BOSME (positive) or in favor of SMOTE (negative, in red).

Figure 2.3 shows the evolution of the β -score as γ varies. We observe that the β -score is always a positive value, which means that BOSME beats SMOTE for all the considered γ . The corresponding p-value for the exact Binomial test for this event (0.0009765625^{***}) implies a statistical significance in favor of BOSME. Furthermore, the β -score tends to grow when the γ value increases. If there were no differences, it would take positive and negative values randomly. We use Mann-Kendall test to check the statistical significance of the trend monotonicity and Sen's slope to take the magnitude of the trend. The results of the Mann-Kendall test, Sen's slope and also Spearman's rank correlation test, are given in Table 2.4 below.

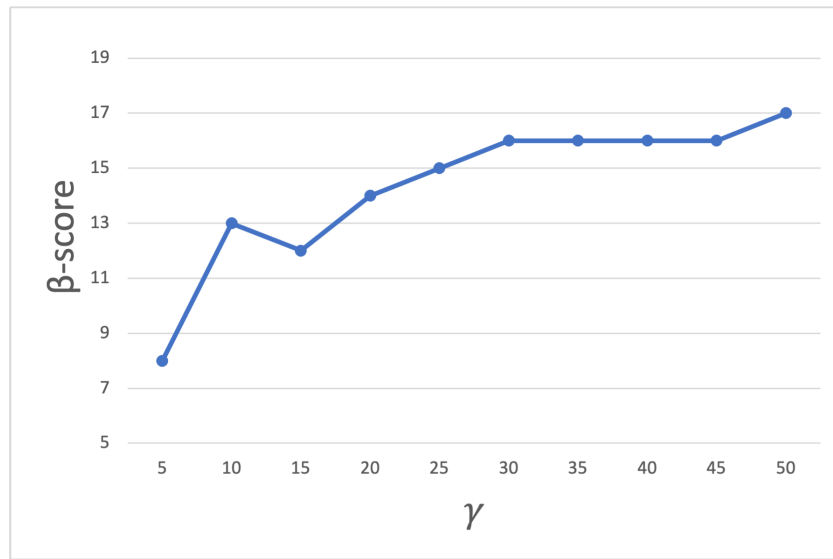


Fig 2.3. Graphic representation of the β -score, where the evolution as the cost ratio γ increases can be observed.

Mann-Kendall			Spearman's rank correlation	
τ	two-sided p-value	Sen's slope	ρ	p-value
0.649	0.000091***	1.34286 , 95% CI: (0.4, 2.5)	0.95672	0.000007***

Table 2.4. β -score. Mann-Kendall test: τ statistic, two-sided p-value and Sen's slope with a confidence interval (CI) of 95%. Spearman's rank correlation test: rho statistic and one-sided p-value for the alternative hypothesis that $\rho > 0$.

Table 2.4 shows that there is indeed a significant monotonically increasing trend in the β -score, as the cost ratio γ increases. The empirical evidence is in the sense that: BOSME outperforms SMOTE for all the values tested in the experimental phase, but it also does so more the higher the value of the cost ratio γ .

2.2 Building the model (Ensemble Weighted Average)

In this section we present the paper “*Survival in the Intensive Care Unit: A prognosis model based on Bayesian classifiers*” [7].

In this paper we address the problem of imbalance dataset (topic (a)), giving a proposal from a different paradigm (ensemble of classifiers) than the one presented in the previous section, in addition to giving a solution to a need from another discipline (medicine): our EWA model. As we have explained in the previous chapter, our model is an *ensemble* of classifiers constructed using the Weighted Average combination rule, with conveniently chosen weights to handle the imbalance problem, that assigns labels maintaining consistency with the *Confidence Level* associated to the prediction.

In addition, our EWA model allows to generate knowledge, and to identify protective and risk factors for death in the hospital ICU. We have measured the importance of the features (their “influence”) using measures of *centrality* and *betweenness*, among other methodologies that we have applied. We also compute the Odds Ratio to quantify the effect on the risk of death of the different features.

2.2.1 Comparing the models for the variable *Result*

In [7] we have introduced a hierarchical predictive model in which the variable *Result* (live/die) is first predicted and then the Destination upon discharge from the ICU, in the event that the prediction is “live”, or the Cause of death, in case the prediction is “die”. However, in this chapter we are going to restrict ourselves to presenting the results relative only to the *Result* variable, which is, in fact, the most important of the three that are predicted.

In Table 2.5 we record the average over the runs of the averages and the standard deviations, \bar{x} and s , respectively, over the folds, for the performance metrics AUPR, F-score and AUC.

<i>Result</i>	AUPR		F-score		AUC	
	\bar{x}	s	\bar{x}	s	\bar{x}	s
BC ₁	0.52058 (5)	0.08914	0.54445 (1)	0.06056	0.87230 (3)	0.02858
BC ₂	0.50671	0.08514	0.52763 (3)	0.06642	0.85987	0.03192
BC ₃	0.35161	0.13386	0.06805	0.02496	0.82825	0.04087
BC ₄	0.48450	0.07944	0.49974	0.06621	0.83958	0.03605
BC ₅	0.46424	0.07927	0.47569	0.06794	0.83277	0.03456
NN	0.27294	0.23893	0.43670	0.07689	0.70228	0.18079
SVM	0.43432	0.08309	0.32713	0.08628	0.79698	0.04014
RF	0.37071	0.08110	0.37567	0.08028	0.76864	0.04096
APACHEII	0.37899	0.09393			0.77518	0.04837
LR.APACHEII	0.42621	0.09342	0.30706	0.09075	0.83154	0.03744
MV	0.52467 (3)	0.08276	0.50274	0.06744	0.86440 (4)	0.03027
WMV	0.52317 (4)	0.08316	0.51137 (5)	0.06791	0.86377 (5)	0.03098
EA	0.53829 (2)	0.08510	0.52354 (4)	0.06666	0.87913 (2)	0.02538
EWA	0.54131 (1)	0.08423	0.53270 (2)	0.06766	0.88026 (1)	0.02522

Table 2.5. Average over the runs of the averages (\bar{x}) and the standard deviations (s) over the folds, for the metrics AUPR, F-score and AUC, with the different classifiers. In boldface, the top five for each metric.

Note that we record the values for any of the classifiers considered: the five base classifiers BC₁, . . . , BC₅, the state-of-the-art Neural Network (NN), Support Vector Machine (SVM) and Random Forest (RF), the predictive model based on the APACHE II score

using the fixed coefficients (“APACHEII”), the corresponding but learning the coefficients from the training data (“LR.APACHEII”), and the ensembles considered with the different combination rules: MV, WMV, EA and EWA, with the weights obtained from the AUPR (see (1.5)). The blank cells indicate that the F-score can not be calculated by the arrangement of the zeros in the confusion matrices generated by the APACHEII model.

From the experiment, we can see that there is a clear advantage for the ensembles, especially EWA and EA, over the rest of the classifiers, with AUPR and AUC, while for the F-score, the best classifiers are BC_1 , EWA, BC_2 and EA. That is why we focus on the comparison between the ensembles EWA, EA, WMV and MV, to each other, in addition to in their comparison with the rest. In this chapter we are only going to collect the results for the AUPR metric, as an example. Indeed, Table 2.6 reports for each run if there is a statistically significant (p -value < 0.1) improvement of either EWA or EA, with respect to WMV and/or MV (“2” means that there is an improvement over WMV and MV, “1” means that there is only over one of them, and “0” that there is none for either), and we observe that in no case are WMV or MV better than EWA or EA.

AUPR	Run																			
Result	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
EWA	0	0	1	2	2	0	2	0	2	1	1	2	2	0	1	1	1	1	2	2
EA	0	0	0	2	2	0	2	0	2	1	1	2	1	0	0	1	1	0	2	1

Table 2.6. AUPR: comparison between EWA/EA and WMV/MV.

The p -values of the comparisons are reported in Table 2.7 and have been adjusted for multiple comparisons between the four ensembles by using the method of Holm-Bonferroni, with the pairwise Wilcoxon signed-rank test to compare matched pairs of samples corresponding to the same run. This statistical test is used as an alternative to the Student’s t -test when the population cannot be assumed to be normally distributed (according to the Shapiro-Wilk test, which has been previously performed).

From these tables we see that EWA and EA outperform WMV and MV, and that in 5 runs, there are significant differences among EWA and EA and, in all the cases, EWA shows to be better. This is confirmed in Table 2.8, where we observe that EWA is significantly better than EA in 8 runs, when we compare only the two and, therefore, the p -values have not been adjusted, and in all the cases, EWA shows to be better.

What significance does this fact have? We compute the p -value for the exact Binomial test in order to compare the proportions of cases in which EWA outperforms EA and vice versa, instead of use McNemar test, because the sample is small. The one-sided p -value for the exact Binomial test is $P(B(n = 5, p = 0.5) = 5) = 0.5^5 = \mathbf{0.03125^*}$ when we compare EWA and EA but adjust for the comparison of the four ensembles,

AUPR (<i>Result</i>)	EWA	EA	WMV	MV
EWA>		0.0059**	0.029*	0.0342*
		0.029*	0.018*	0.098·
		(5) 0.0059**	0.024*	0.018*
		0.0342*	0.0117*	0.018*
		0.049*	(8) 0.039*	0.0098**
			0.029*	0.082·
			0.012*	0.012*
			0.055·	(15) 0.056·
				0.074·
				0.0059**
				0.029*
				0.093·
				0.059·
				0.012*
				0.018*
EA>			0.049*	0.098·
			0.029*	0.027*
			0.056·	0.039*
			0.0645·	0.0146*
		(8)	0.024*	0.093·
			0.049*	0.068·
			0.012*	0.012*
			0.018*	0.034*
				0.029*
				0.021*
		(10)		

Table 2.7. Adjusted p-values for the comparisons between the four ensembles, corresponding to the statistical significances in Table 2.6 when we compare EWA/EA against WMV and MV. Also, in boldface, the adjusted p-values corresponding to the comparison between the two EWA and EA, that were not reported there.

AUPR (<i>Result</i>)	Run	3	5	9	14	15	18	19	20
EWA>EA	p-value	0.00098***	0.014*	0.00098***	0.042*	0.0068**	0.08·	0.024*	0.042*

Table 2.8. Non-adjusted p-values corresponding to the comparisons between EWA and EA in Table 2.7, but only between them two (so the p-values are not adjusted). In boldface the 5 runs corresponding to the adjusted p-values that have been reported in Table 2.7.

which decreases to $P(B(n = 8, p = 0.5) = 8) = 0.5^8 = \mathbf{0.00391}^{**}$ if we consider the non-adjust corresponding to comparison of EWA against EA alone. In both cases there is a statistically significant evidence in favour of EWA as opposed to EA for prediction of variable *Result*, with AUPR as performance measure.

As regards APACHEII and LR.APACHEII, both are clearly worse than any of the ensembles, and we observe significant differences among them, in favour of the latter. Indeed, for 18 runs there are differences between the two models for the AUPR metric, and in all cases LR.APACHEII turns out to be better than the standard based on APACHE II, with a one-sided p-value for the exact Binomial test: $0.5^{18} = \mathbf{3.81470} \times 10^{-6}^{***}$.

2.2.2 Centrality measures

Centrality and betweenness: Each BC_2 , BC_3 , BC_4 and BC_5 model has a graphical representation through a Directed Acyclic Graph (DAG) from which centrality and betweenness measures defined in Graph Theory and network analysis can be extracted. We can establish which features play the main role in the model by using centrality and/or betweenness measures borrowed from the Network Analysis area applied to the DAGs. In Graph Theory and Network Analysis, indicators of centrality identify the most important (influential) nodes within a graph, where “importance” is conceived as involvement in the cohesiveness of the network. For the features we compute four different of these indicators:

- a) *Freeman’s degree of centrality*, which counts paths which pass through each node, that is, directed arcs which arrive at or depart from it.
- b) *Basic standard betweenness measure*, which quantifies the number of times a node acts as a “bridge” along the shortest path between two other nodes (called “geodesic”). Nodes that have a high probability to occur on a randomly chosen geodesic between two randomly chosen nodes, have a high betweenness. Fixed a node v , this measure is defined by $\sum_{i,j, i \neq j, i \neq v, j \neq v} (g_{ivj}/g_{ij})$ (with the convention $0/0 = 0$), where g_{ij} is the number of geodesics from i to j in the graph, and g_{ivj} is the number of geodesics in the subset of those that pass through v .
- c) *Borgatti’s proximal source betweenness* is a variant of basic standard betweenness to accumulate only for the last intermediating vertex in each incoming geodesic. This expresses the notion that, by serving as the “proximal source” for the target, this particular intermediary node will in some settings have greater influence than the rest. Fixed a node v , this measure is defined by $\sum_{i,j, i \neq j, j \neq v, i \rightarrow v} (g_{ivj}/g_{ij})$.
- d) *Borgatti’s proximal target betweenness* is the counterpart to proximal source betweenness that allows betweenness to accumulate only for the first intermediating vertex in each outgoing geodesic. This expresses the notion that, by serving as the “proximal target” for the source, this particular intermediary node will in some settings have greater influence or control than others. Fixed a node v , this measure is defined by $\sum_{i,j, i \neq j, i \neq v, j \leftarrow v} (g_{ivj}/g_{ij})$.

Table 2.9 summarizes the most influential features because of their values of centrality and betweenness. They act as *gateways*, and the arcs that connect them as *bridges* through which information flows from one cluster of variables in the model to another.

Demographic characteristics	F ₂ : Age
Main cause of admission	F ₄ : Acute Coronary Syndrome F ₅ : Respiratory Failure F ₆ : Shock
Admission	F ₁₈ : Generic Syndrome F ₁₉ : Sepsis
Severity (on first 24 hours of admission)	F ₂₀ : ICU Workload F ₂₁ : APACHE II

Table 2.9. The most influential features attending to centrality and betweenness.

2.2.3 Odds Ratio

Once we have chosen the predictive model with the best performance among those we have compared, our EWA, we can learn it from the entire database, not just the training set, and use it to evaluate the effect of the features in the assessment of the risk of death in the ICU. An odds ratio (OR) is a measure of association between a feature and the outcome (variable *Result*, in this case), which represents the odds in favour of “die” given a particular value of a feature, compared to the odds in favour of “die” given another value, fixed the other features. For instance, consider a critically ill patient with the characteristics in Table 2.10.

Charlson	Origin	Generic syndrome	Sepsis	ICU Workload	APACHE II
2	Emergency Room	Medical	yes	M. unstable coma/shock	5–9

Table 2.10. Example of characteristics of a critically ill patient.

In Table 2.11 we record the OR, disaggregated by sex, in favour of the event “die”, for a critically patient whose characteristics are given in Table 2.10, according to what of the “Main cause of admission” has been reported for the patient (from F₄ to F₁₆).

F ₁ : Sex F ₂ : Age	Male			Female		
	75–84	> 84	OR	75–84	> 84	OR
F ₄	0.07878	0.09715	1.25837	0.08084	0.10693	1.36138
F ₅	0.19627	0.30552	1.80145	0.22859	0.36471	1.93737
F ₆	0.20421	0.31516	1.79328	0.23714	0.37257	1.91022
F ₇	0.20070	0.31982	1.87257	0.23325	0.37825	1.99987
F ₈	0.46216	0.55110	1.42871	0.50267	0.57898	1.36059
F ₉	0.16010	0.16010	1.00000	0.16010	0.16010	1.00000
F ₁₀	0.49500	0.66774	2.05030	0.53169	0.72091	2.27516
F ₁₁	0.20996	0.26928	1.38660	0.23044	0.31191	1.51378
F ₁₂	0.07956	0.13082	1.74141	0.09336	0.16242	1.88317
F ₁₃	0.30486	0.38341	1.41787	0.35674	0.44058	1.42009
F ₁₄	0.07100	0.07003	0.98518	0.07433	0.07305	0.98137
F ₁₅	0.10251	0.20942	2.31921	0.11517	0.22107	2.18048
F ₁₆	0.08263	0.10638	1.32156	0.08962	0.12278	1.42184

Table 2.11. Example of Table 2.10: probabilities of “die” and OR in favour of “die”, for each of the possible “Main cause of admission”. In boldface those probabilities > 0.5, which carry a prediction of “die” for the patient.

In Table 2.11, OR is defined as the odds of event “die” occurring in the group of age >84 divided by the odds of it occurring in the group of age 75–84. For example, continuing with the patient whose characteristics are given in Table 2.10, if the patient is a male between 75 and 84 years old with renal failure ($F_8=1$), thus, his risk of death (probability of “die”) assigned by the model is 0.46216. This probability increases up to 0.55110 if the age increases to be > 84 . Therefore, the OR in favour of “die” is:

$$\text{OR}_{>84/75-84} = \frac{0.55110/(1 - 0.55110)}{0.46216/(1 - 0.46216)} = 1.42871$$

We also see that for patients having the characteristics recorded in Table 2.10 and having cardio respiratory arrest ($F_{10} = 1$) or intoxication ($F_{15} = 1$), both for men and women, the increase in age is an important risk factor (OR greater than 2 in Table 2.11).

On the other hand, we can study which of the “Main cause of admission” are risk factors for a male who is more than 85 years old, and with the features in Table 2.10, for example, and consider the question: “What is the Odds Ratio between F_{10} and F_5 in favor of *die*?”, which is answered by computing the ratio between the odds in favour of “die” when $F_{10} = 1$ and when $F_5 = 1$, which is:

$$\text{OR}_{F_{10}/F_5} = \frac{0.66774/(1 - 0.66774)}{0.30552/(1 - 0.30552)} = 4.56836$$

(see Table 2.11) that is, the odds in favour of “die” for a male who is more than 85 and has the features in Table 2.10 is approximately 4.6 times greater if his main cause of admission is a cardio respiratory arrest than if it is a respiratory failure.

2.2.4 Feature strength

We want to measure the strength of any of the features to predict the output variable *Result*. We follow [20] and introduce a measure based on the conditional probability tables of *Result* with respect to each feature, obtained with EWA, which uses the Kolmogorov-Smirnov statistical distance and a correction parameter: for each feature F , we introduce the *Strength Distance* (SD), as:

$$\text{SD}(F) = \max_{a, b \in \mathcal{F}} d_{a, b}^F$$

where \mathcal{F} is the set of the possible outcomes of variable F , and $d_{a, b}^F$ denotes the Kolmogorov-Smirnov statistical distance between the *a posteriori* conditional probability distributions of *Result* given the evidence $F = a$, and given the evidence $F = b$.

To take into account if different instantiations of a feature produce different predic-

tions for *Result*, we introduce the correction term $\delta(F) = \gamma(F)/2 \in (0, 1]$, where $\gamma(F)$ is the number of different predictions obtained from the classifier for *Result* given the evidences of the form $F = a$, with a varying in \mathcal{F} . Then, $\delta(F)$ is the proportion of different predictions actually obtained by the classifier for *Result* among the possible we could obtain from an evidence on F , which is 2, and we use it to correct strength measure SD by introducing the *Corrected Strength Distance* (CSD) by $CSD(F) = SD(F) \times \delta(F)$. Note that $CSD(F) \geq 0$, and that $CSD(F) = 0$ if and only if F and *Result* are independent variables. In Table 2.12 we have recorded for each feature the correction term δ and the feature strength measure SD and CSD as well.

Feature	SD	δ	CSD(= SD \times δ)
F ₁ : Sex	0.03432	1/2	0.01716
F ₂ : Age	0.09623	1/2	0.04812
F ₃ : Charlson	0.14104	1/2	0.07052
F ₄ : Acute Coronary Syndrome	0.17367	1/2	0.08684
F ₅ : Respiratory Failure	0.03301	1/2	0.01651
F ₆ : Shock	0.08698	1/2	0.04349
F ₇ : Coma	0.02511	1/2	0.01256
F ₈ : Renal Failure	0.00865	1/2	0.00433
F ₉ : Hepatic Failure	0.13364	1/2	0.06682
F ₁₀ : Cardio Respiratory Arrest	0.49694	1	0.49694
F ₁₁ : Elective Surgical	0.15203	1/2	0.07602
F ₁₂ : Arrhythmia	0.09849	1/2	0.04925
F ₁₃ : Cranial Trauma	0.15729	1/2	0.07865
F ₁₄ : Other Trauma	0.12650	1/2	0.06325
F ₁₅ : Intoxication	0.07601	1/2	0.03801
F ₁₆ : Other syndromes	0.12105	1/2	0.06053
F ₁₇ : Origin	0.35811	1/2	0.17906
F ₁₈ : Generic syndrome	0.17512	1/2	0.08756
F ₁₉ : Sepsis	0.11333	1/2	0.05667
F ₂₀ : ICU workload	0.43213	1/2	0.21607
F ₂₁ : APACHE II	0.63546	1	0.63546

Table 2.12. SD, the correction term δ and CSD for the 21 features, with the EWA model

Attending to CSD as feature strength measure, we can rank the features as follows, from stronger to weaker:

$$F_{21}, F_{10}, F_{20}, F_{17}, F_{18}, F_4, F_{13}, F_{11}, F_3, F_9, \\ F_{14}, F_{16}, F_{19}, F_{12}, F_2, F_6, F_{15}, F_1, F_5, F_7, F_8.$$

2.3 Validation (MCEN)

In this section we present the last paper, entitled “*Enhancing Confusion Entropy (CEN) for binary and multiclass classification*” [5].

In this work we have introduced a new measure, that we name MCEN, by modifying conveniently the measure CEN (Confusion Entropy), which was inspired by Shannon’s entropy. MCEN is introduced to avoid CEN’s unwanted behaviour in the binary case. Now we summarize the main results in which MCEN and CEN (and other measures) are compared, and it is verified that MCEN avoids the problem posed by CEN, which has been the leitmotif to introduce it.

2.3.1 The perfectly symmetric and balanced case

We denote by “perfectly symmetric and balanced case” the case in which $C_{i,j} = F$ for all $i, j = 1, \dots, N$, $i \neq j$ and $C_{i,i} = T$, with $T \geq 0$, $F > 0$, that is, the confusion matrix is of the form

$$C = \begin{pmatrix} T & F & \dots & F & F \\ F & T & \dots & F & F \\ \vdots & \vdots & \dots & \vdots & \vdots \\ F & F & \dots & T & F \\ F & F & \dots & F & T \end{pmatrix}.$$

Proposition 1. *In the perfectly symmetric and balanced case,*

$$\begin{aligned} \text{If } N > 2, \quad \text{CEN} &= \frac{2(N-1)}{\delta} \log_2(\delta), \quad \text{MCEN} = \frac{2(N-1)}{\tilde{\delta}} \log_2(\tilde{\delta}), \quad (2.3) \\ \text{If } N = 2, \quad \text{CEN} &= \frac{1}{1+\gamma} \log_2(\delta), \quad \text{MCEN} = \frac{1}{1+\frac{3}{4}\gamma} \log_2(\tilde{\delta}), \end{aligned}$$

where $\gamma = \frac{T}{F} \geq 0$, $\delta = 2(N-1) + 2\gamma > 0$ and $\tilde{\delta} = 2(N-1) + \gamma > 0$,

$$\text{ACC}^* = \frac{N-1}{\gamma + (N-1)} \quad \text{and} \quad \text{MCC}^* = \frac{N}{2(\gamma + (N-1))} = \frac{N}{2(N-1)} \text{ACC}^*.$$

Note that ACC^* , MCC^* , CEN and MCEN depend on the matrix values T and F only through its ratio γ .

Corollary 1. *In the perfectly symmetric and balanced case, we have that:*

- For any $N > 2$, CEN, MCEN, ACC* and MCC* are monotonically decreasing functions of $\gamma \geq 0$, with

$$\lim_{\gamma \rightarrow +\infty} \text{CEN}(\gamma) = \lim_{\gamma \rightarrow +\infty} \text{MCEN}(\gamma) = \lim_{\gamma \rightarrow +\infty} \text{ACC}^*(\gamma) = \lim_{\gamma \rightarrow +\infty} \text{MCC}^*(\gamma) = 0,$$

$$\text{CEN}(0) = \text{MCEN}(0) = \text{ACC}^*(0) = 1, \text{MCC}^*(0) = \frac{N}{2(N-1)} \rightarrow \frac{1}{2} \text{ as } N \rightarrow +\infty,$$

and if $\gamma > 0$, $\text{MCC}^* < \text{ACC}^* < \text{CEN} < \text{MCEN}$.

- Nevertheless, when $N = 2$, we have that although MCEN and $\text{ACC}^* = \text{MCC}^*$ remain to be monotonically decreasing as functions of $\gamma \geq 0$, CEN does not. Indeed, CEN achieves its global maximum when $\gamma = \frac{e}{2} - 1$, which is $\text{CEN}(\frac{e}{2} - 1) \approx 1.06148 > 1$. More specifically,

$$\text{CEN}(0) = \text{CEN}(1) = 1, \text{CEN}(\gamma) > 1, \text{ for all } 0 < \gamma < 1, \lim_{\gamma \rightarrow +\infty} \text{CEN}(\gamma) = 0,$$

$$\text{MCEN}(0) = 1, \lim_{\gamma \rightarrow +\infty} \text{MCEN}(\gamma) = 0,$$

$$\text{ACC}^*(0) = \text{MCC}^*(0) = 1, \lim_{\gamma \rightarrow +\infty} \text{ACC}^*(\gamma) = \lim_{\gamma \rightarrow +\infty} \text{MCC}^*(\gamma) = 0.$$

Moreover, there exists $\gamma_0 \approx 5.78$ such that

$$\begin{aligned} \text{MCC}^* = \text{ACC}^* < \text{MCEN} < \text{CEN} & \text{ if } 0 < \gamma < \gamma_0, \\ \text{MCC}^* = \text{ACC}^* < \text{MCEN} = \text{CEN} & \text{ if } \gamma = \gamma_0, \text{ and} \\ \text{MCC}^* = \text{ACC}^* < \text{CEN} < \text{MCEN} & \text{ if } \gamma > \gamma_0. \end{aligned}$$

Remark 1. Note that if $N = 2$, CEN exhibits the unwanted behaviour, not showed by MCEN, of being out-of-range $[0, 1]$, which despairs for $N > 2$ (see Figura 2.4).

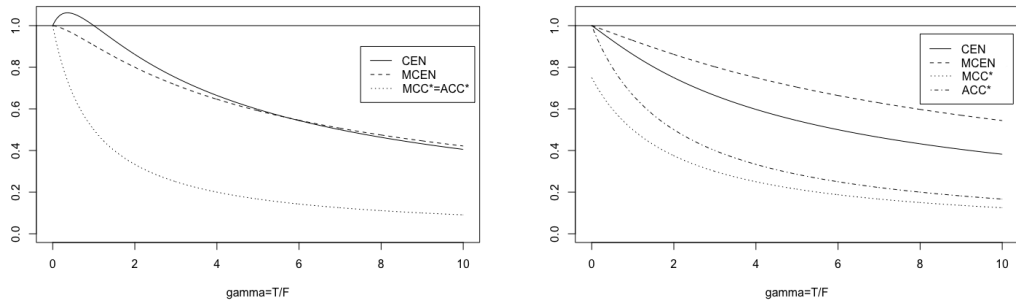


Fig 2.4. The symmetric case. CEN, MCEN, ACC* and MCC* for $\gamma \in [0, 10]$, with $N = 2$ (left) and $N = 3$ (right).

Remark 2. Consider the particular case in which $T = F$, that is, $\gamma = 1$. In other words, the confusion matrix is constant, say $\begin{pmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}$. Then, $\text{ACC}^* = \frac{N-1}{N}$ and $\text{MCC}^* = \frac{1}{2}$. Moreover, $\delta = 2N$ and $\tilde{\delta} = 2N - 1$.

If $N > 2$, $\text{CEN} = (1 - \frac{1}{N}) \log_{2(N-1)}(2N)$ and $\text{MCEN} = (1 - \frac{1}{2N-1}) \log_{2(N-1)}(2N - 1)$. If $N = 2$, $\text{CEN} = 1$ and $\text{MCEN} = \frac{4}{7} \log_2(3) < 1$.

As a consequence, we can easily check that if $N > 2$, $\text{MCC}^* < \text{ACC}^* < \text{CEN} < \text{MCEN}$, with $\lim_{N \rightarrow +\infty} \text{ACC}^* = \lim_{N \rightarrow +\infty} \text{CEN} = \lim_{N \rightarrow +\infty} \text{MCEN} = 1$, while if $N = 2$, $\text{MCC}^* = \text{ACC}^* < \text{MCEN} < \text{CEN}$.

2.3.2 The binary case

Symmetric but unbalanced U_A family

Consider the particular case of a confusion matrix of type $U_A = \begin{pmatrix} 1 & A \\ A & 0 \end{pmatrix}$, with $A > 0$. Both class-1 and class-2 cases are mainly misclassified if $A > 1$. Entropy out of the main diagonal is 1 and within the diagonal is 0, regardless of the value of A . When $0 < A < 1$, say for example that $A = 1/B$ with $B > 1$, then matrix U_A is equivalent to $\begin{pmatrix} B & 1 \\ 1 & 0 \end{pmatrix}$, that is, corresponds to an unbalanced scenario in which class 2 is underrepresented and class-1 cases are mainly well classified. We can observe some properties of CEN , MCEN , ACC^* and MCC^* (see Fig. 2.5) in Proposition 2.

Proposition 2. For confusion matrix U_A with $A > 0$, we have:

$$\begin{aligned} \text{CEN}(A) &= \frac{A \log_2((2A+1)^2 - 1) - 2A \log_2(A)}{2A+1}, \\ \text{MCEN}(A) &= \frac{4A \log_2(2A(2A+1)) - 8A \log_2(A)}{3(2A+1) + 2A}, \\ \text{ACC}^*(A) &= \frac{2A}{2A+1}, \quad \text{MCC}^*(A) = \frac{2A+1}{2(A+1)}. \end{aligned}$$

As a consequence:

$$\begin{aligned} \text{CEN}(A) &< 1 \text{ if } A < 1, \text{ CEN}(1) = 1, \text{ CEN}(A) > 1 \text{ if } A > 1, \\ \text{MCEN}(A) &< 1 \quad \text{and} \quad \text{ACC}^*(A) < \text{MCC}^*(A) < 1, \quad \text{for all } A > 0, \\ \text{MCEN}, \text{ACC}^* \quad \text{and} \quad \text{MCC}^* &\text{ are monotonically increasing functions of } A > 0, \end{aligned}$$

CEN is not, and achieves its global maximum when $A \approx 2.54$, which is > 1 ,

$$\lim_{A \rightarrow 0} \text{CEN}(A) = \lim_{A \rightarrow 0} \text{MCEN}(A) = \lim_{A \rightarrow 0} \text{ACC}^*(A) = 0 < \lim_{A \rightarrow 0} \text{MCC}^*(A) = 0.5,$$

$$\lim_{A \rightarrow +\infty} \text{CEN}(A) = \lim_{A \rightarrow +\infty} \text{MCEN}(A) = \lim_{A \rightarrow +\infty} \text{ACC}^*(A) = \lim_{A \rightarrow +\infty} \text{MCC}^*(A) = 1.$$

Moreover, there exists $A_0 \in (0, 1)$ (indeed, $A_0 \approx 0.24$) such that

$$\begin{cases} \text{MCEN}(A) < \text{CEN}(A) & \text{if } A > A_0, \\ \text{MCEN}(A_0) = \text{CEN}(A_0), \\ \text{MCEN}(A) > \text{CEN}(A) & \text{if } 0 < A < A_0. \end{cases}$$

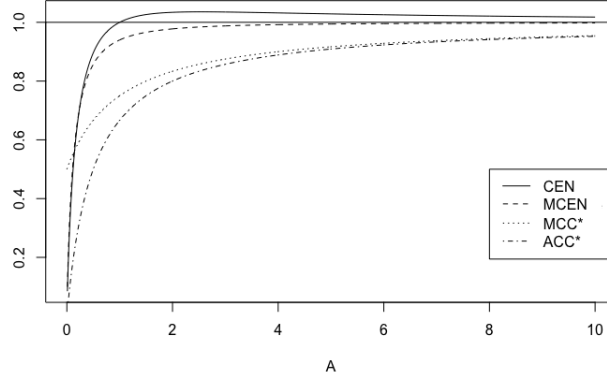


Fig 2.5. Family U_A . CEN, MCEN, ACC* and MCC* for $A \in (0, 10]$.

Symmetric V_A family

Consider the particular case of confusion matrices of type $V_A = \begin{pmatrix} 1 & A \\ 1 & 0 \end{pmatrix}$, with $A > 0$.

This is an asymmetric and unbalanced case in which class 2 is systematically misclassified and is underrepresented if $A > 1$. Class 1 is also mainly misclassified if $A > 1$. As $A \rightarrow +\infty$, entropy out the diagonal, which is $-\frac{A}{A+1} \log(\frac{A}{A+1})$, decreases to zero. Entropy within diagonal is zero, while the overall entropy of the elements of matrix V_A is $\log(A+2) - \frac{A}{A+2} \log(A)$, which tends to 0 as $A \rightarrow +\infty$. When $0 < A < 1$ with $A = 1/B$, $B > 1$, matrix V_A is equivalent to $\begin{pmatrix} B & 1 \\ B & 0 \end{pmatrix}$, which corresponds to an almost balanced but asymmetric scenario in which class 1 is mainly well classified but class 2 is not. As B increases ($A \rightarrow 0$), entropy out the diagonal also drops to zero. Some properties of CEN, MCEN, ACC* and MCC* are given in Proposition 3 (see also Fig. 2.6).

Proposition 3. For confusion matrix V_A with $A > 0$, we have:

$$\begin{aligned} \text{CEN}(A) &= \frac{(A+1) \log_2((A+2)^2 - 1) - 2A \log_2(A)}{2(A+2)}, \\ \text{MCEN}(A) &= \frac{2(A+1) \log_2((A+1)(A+2)) - 4A \log_2(A)}{3(A+2) + (A+1)}, \\ \text{ACC}^*(A) &= \frac{A+1}{A+2}, \quad \text{MCC}^*(A) = \frac{1 + \sqrt{\frac{A}{2(A+1)}}}{2}. \end{aligned}$$

As a consequence, there exists $A_1 \in (1, 2)$ ($A_1 \approx 1.414$) such that:

$\text{CEN}(A) > 1$ if $1 < A < A_1$, $\text{CEN}(1) = \text{CEN}(A_1) = 1$, $\text{CEN}(A) < 1$ if $A \notin [1, A_1]$,
 $\text{MCEN}(A) < 1$, $\text{ACC}^*(A) < 1$, $\text{MCC}^*(A) < 1$ and $\text{MCEN}(A) < \text{CEN}(A)$ for all $A > 0$,

$$\begin{aligned} \lim_{A \rightarrow 0} \text{MCC}^*(A) = \lim_{A \rightarrow 0} \text{ACC}^*(A) &= \frac{1}{2} > \lim_{A \rightarrow 0} \text{CEN}(A) = \frac{\log_2(3)}{4} > \lim_{A \rightarrow 0} \text{MCEN}(A) = \frac{2}{7}, \\ \lim_{A \rightarrow +\infty} \text{ACC}^*(A) = 1 > \lim_{A \rightarrow +\infty} \text{MCC}^*(A) &= \frac{2 + \sqrt{2}}{4} > \lim_{A \rightarrow +\infty} \text{CEN}(A) = \lim_{A \rightarrow +\infty} \text{MCEN}(A) = 0. \end{aligned}$$

Note that as in previous cases, $\text{CEN}(A)$ does not stay always (that is, for any $A > 0$) restricted to $[0, 1]$, while MCEN does.

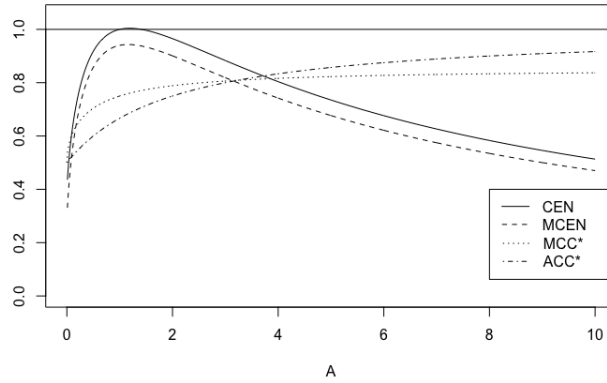


Fig 2.6. Family V_A . CEN, MCEN, ACC* and MCC* for $A \in (0, 10]$.

Asymmetric but unbalanced $Y_{A,r}$ family

Finally, we consider a particular doubly indexed family of confusion matrices in the binary case, denoted by $Y_{A,r}$, with $A, r > 0$. We define this family by $Y_{A,r} = \begin{pmatrix} rA & rA \\ A & 1 \end{pmatrix}$. Class-2 is underrepresented and mainly misclassified if $A, r > 1$, while class-1 cases are classified “at random”, that is, a class-1 case has the same probability to be classified into any of the two classes.

When $0 < A < 1$, $A = 1/B$ with $B > 1$, then matrix $Y_{A,r}$ is equivalent to $\begin{pmatrix} r & r \\ 1 & B \end{pmatrix}$. In Proposition 4 we give some properties of CEN, MCEN, ACC* and MCC*. See Fig. 2.7 for $r = 0.1$, Fig. 2.8 for $r = 0.8$, and Fig. 2.9 for a plot of them as function of r , fixed $A = 10$.

Proposition 4. *For confusion matrix $Y_{A,r}$ with $A, r > 0$ we have:*

$$\begin{aligned} \text{CEN}(A) &= \frac{(r+1)A \log_2 \left(((r+1)A+2)(3r+1) \right) + (r-1)A \log_2(A) - 2rA \log_2(rA)}{2((2r+1)A+1)}, \\ \text{MCEN}(A) &= \frac{2 \left((r+1)A \log_2 \left(((r+1)A+1)(2r+1) \right) + (r-1)A \log_2(A) - 2rA \log_2(rA) \right)}{3((2r+1)A+1) + (r+1)A}, \\ \text{ACC}^*(A) &= \frac{(r+1)A}{(2r+1)A+1}, \quad \text{MCC}^*(A) = \frac{1 - \frac{r(1-A)}{\sqrt{2r(A+1)(r+1)(rA+1)}}}{2}. \end{aligned}$$

As a consequence, $L_{\text{CEN}}(r) = \lim_{A \rightarrow +\infty} \text{CEN}(A) = \frac{1}{2(2r+1)} \log_2 \left(\frac{((3r+1)(r+1))^{r+1}}{r^{2r}} \right) > 0$,

and there exists $R_0 < 1$ ($R_0 \approx 0.71$) such that $L_{\text{CEN}}(r) \begin{cases} > 1 & \text{if } R_0 < r < 1, \\ = 1 & \text{if } r = R_0, 1, \\ < 1 & \text{if } r < R_0 \text{ or } r > 1. \end{cases}$

Moreover, there exist $0 < R_1 < R_0 < 1 < R_2$ ($R_1 \approx 0.5$, $R_2 \approx 1.4$) such that

$$\left\{ \begin{array}{ll} \text{if } r \in [R_0, 1], & \text{there exists } A_r > 0 \text{ such that } \text{CEN}(A) < 1 \text{ if } A < A_r, \\ & \text{CEN}(A_r) = 1, \text{ CEN}(A) > 1 \text{ if } A > A_r, \\ \text{if } r \in (R_1, R_0) \cup (1, R_2), & \text{there exist } 0 < A_r < B_r \text{ such that } \text{CEN}(A) < 1 \text{ if } A < A_r \\ & \text{or } A > B_r, \text{ CEN}(A_r) = \text{CEN}(B_r) = 1, \\ & \text{CEN}(A) > 1 \text{ if } A \in (A_r, B_r), \\ \text{if } r \notin (R_1, R_2), & \text{CEN}(A) \leq 1 \text{ for any } A > 0. \end{array} \right.$$

On the other hand, for any $r > 0$,

$\text{MCEN}(A) < 1$, $\text{ACC}^*(A) < 1$ and $\text{MCC}^*(A) < 1$, for all $A > 0$,
 ACC^* and MCC^* are monotonically increasing functions of A ,
 CEN is not, and MCEN is or not, depending on the value of r ,

$$\lim_{A \rightarrow 0} \text{CEN}(A) = \lim_{A \rightarrow 0} \text{MCEN}(A) = \lim_{A \rightarrow 0} \text{ACC}^*(A) = 0, \quad \lim_{A \rightarrow 0} \text{MCC}^*(A) = \frac{1 - \sqrt{\frac{r}{2(r+1)}}}{2},$$

$$\lim_{A \rightarrow +\infty} \text{ACC}^*(A) = \frac{r+1}{2r+1} = L_{\text{ACC}^*}(r), \quad \lim_{A \rightarrow +\infty} \text{MCC}^* = \frac{1 + \frac{1}{\sqrt{2(r+1)}}}{2} = L_{\text{MCC}^*}(r),$$

$$L_{\text{MCEN}}(r) = \lim_{A \rightarrow +\infty} \text{MCEN}(A) = \frac{2}{3(2r+1) + (r+1)} \log_2 \left(\frac{((2r+1)(r+1))^{r+1}}{r^{2r}} \right) < 1,$$

$$L_{\text{MCEN}}(r) < L_{\text{CEN}}(r) \text{ for all } r > 0.$$

Note that $L_{\text{ACC}^*}(r) < L_{\text{MCC}^*}(r)$ if and only if $r > \frac{-1+\sqrt{5}}{4} > 0$.

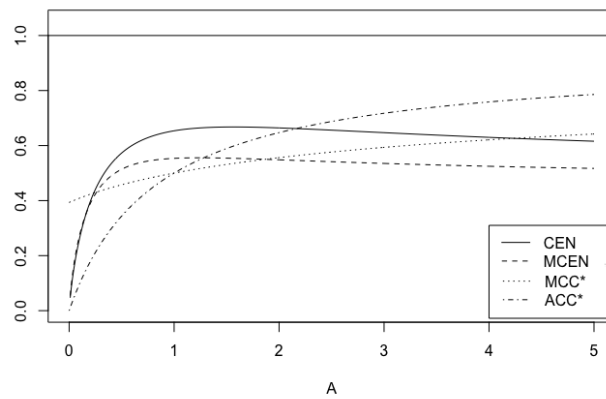


Fig 2.7. Family $Y_{A,r}$. CEN, MCEN, ACC* and MCC* as function of $A > 0$ for $r = 0.1$.

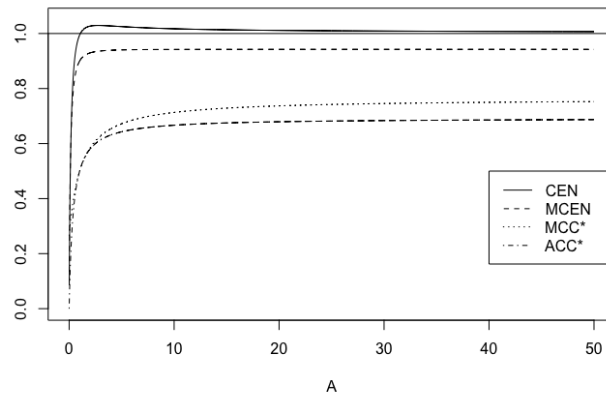


Fig 2.8. Family $Y_{A,r}$. CEN, MCEN, ACC* and MCC* as function of $A > 0$ for $r = 0.8$.

2.3.3 The multiclass Z_A family

As noted in [21], the behaviour of the Confusion Entropy CEN is rather diverse from that of MCC* and ACC* for the pathological case of the family of confusion matrices

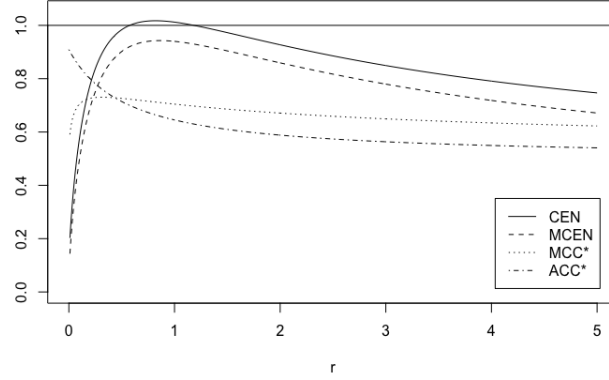


Fig 2.9. Family $Y_{A,r}$. CEN, MCEN, ACC* and MCC* as function of r for $A = 10$.

$Z_A = (a_{i,j})_{i,j=1,\dots,N}$, defined by $a_{i,j} = \begin{cases} A & \text{if } i = N, j = 1 \\ 1 & \text{otherwise,} \end{cases}$, with $A > 0$. That is,

$$Z_A = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots \\ 1 & 1 & \dots & 1 \\ A & 1 & \dots & 1 \end{pmatrix}. \text{ We want to study how MCEN behaves when applied to}$$

elements of this family, and the results are in Proposition 5. See also Fig. 2.10.

Proposition 5.

$$\begin{aligned} \text{If } N > 2, \text{ CEN}(Z_A) &= \frac{1}{N^2 + A - 1} \left((N-1)(N-2) \log_{2(N-1)}(2N) \right. \\ &\quad \left. + (2N + A - 3) \log_{2(N-1)}(2N + A - 1) - A \log_{2(N-1)}(A) \right), \\ \text{MCEN} &= \frac{2}{2(N^2 + A - 1) - N} \left((N-1)(N-2) \log_{2(N-1)}(2N-1) \right. \\ &\quad \left. + (2N + A - 3) \log_{2(N-1)}(2N + A - 2) - A \log_{2(N-1)}(A) \right), \\ \text{if } N = 2, \text{ CEN}(Z_A) &= \frac{1}{A+3} \left((A+1) \log_2(A+3) - A \log_2(A) \right), \\ \text{MCEN} &= \frac{2}{2A+5} \left((A+1) \log_2(A+2) - A \log_2(A) \right). \end{aligned}$$

In general ($N \geq 2$),

$$\text{MCC}^*(Z_A) = \frac{N(N^2 + 2(A-1)) - (N^2 + (A-1))}{2(N-1)(N^2 + 2(A-1))}, \quad \text{ACC}^*(Z_A) = \frac{N^2 - N + (A-1)}{N^2 + (A-1)}$$

As a consequence,

- If $N = 2$,

$\text{MCEN} < \text{CEN}(Z_A)$ for all $A > 0$,

$\text{MCEN} < 1$ for all $A > 0$, and there exists $A_3 \in (1, 2)$ ($A_3 \approx 1.85$) such that

$$\text{CEN}(Z_1) = \text{CEN}(Z_{A_3}) = 1,$$

$$\text{CEN}(Z_A) > 1 \text{ if } A \in (1, A_3) \text{ and } \text{CEN}(Z_A) < 1 \text{ if } A \notin [1, A_3],$$

$$\lim_{A \rightarrow 0} \text{MCC}^*(A) = \frac{1}{4} < \lim_{A \rightarrow 0} \text{ACC}^* = \frac{1}{3} < \lim_{A \rightarrow 0} \text{MCEN}(A) = \frac{2}{5} < \lim_{A \rightarrow 0} \text{CEN}(A) = \frac{\log_2(3)}{3},$$

$$\lim_{A \rightarrow +\infty} \text{CEN}(A) = \lim_{A \rightarrow +\infty} \text{MCEN}(A) = 0 < \lim_{A \rightarrow +\infty} \text{MCC}^* = \frac{3}{4} < \lim_{A \rightarrow +\infty} \text{ACC}^* = 1.$$

- If $N = 3$ (we take this case as example of what happens with $N > 2$),

$$\lim_{A \rightarrow 0} \text{MCC}^*(A) = \frac{13}{28} < \lim_{A \rightarrow 0} \text{ACC}^* = \frac{5}{8} <$$

$$< \lim_{A \rightarrow 0} \text{CEN}(A) = \frac{2 \log_4(6) + 3 \log_4(5)}{8} < \lim_{A \rightarrow 0} \text{MCEN}(A) = \frac{2}{13}(2 \log_4(5) + 3) < 1,$$

$$\lim_{A \rightarrow +\infty} \text{CEN}(A) = \lim_{A \rightarrow +\infty} \text{MCEN}(A) = 0 < \lim_{A \rightarrow +\infty} \text{MCC}^* = \frac{5}{8} < \lim_{A \rightarrow +\infty} \text{ACC}^* = 1.$$

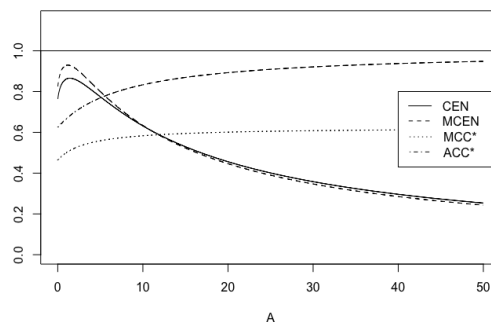
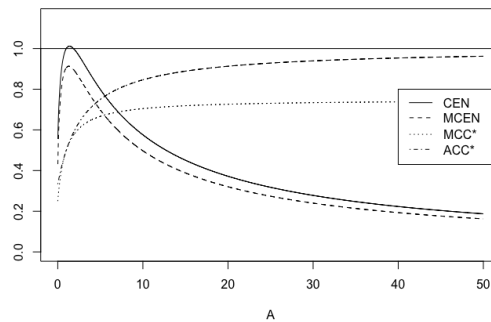


Fig 2.10. Family Z_A . CEN , MCEN , MCC^* and ACC^* as function of $A > 0$ for $N = 2$ (up) and $N = 3$ (down).

2.4 Relevance of the results

In this Section we describe the contribution and impacts of this Thesis. Although the works considered in this Thesis report are sorted according to the life cycle of Machine Learning, the chronological order of the works has been the following:

- The first contribution on “*Enhancing confusion entropy as measure for evaluating classifiers*” was presented in SOCO 2018 (International Conference on Soft Computing Models in Industrial and Environmental Applications) and published in “Advances in Intelligent Systems and Computing” as conference paper [22] in 2019. In this year we contributed with an extended version in PlosOne journal under the title “*Enhancing Confusion Entropy (CEN) for binary and multiclass classification*” [5].

Journal ranking:

PLOS ONE (2019). Category: Multidisciplinary Sciences.

2019 Journal Impact Factor: 2.740 (27/171, Q2). JIF percentile 62.68

2019 Journal Citation Indicator: 0.57 (29/126, Q1). JCI percentile 77.38

2019 Total Citations: 688,786

Cited by:

So far, and as far as we know, the published works that have referenced [22] or [5] (we do not record self-references) are:

Preprints

- Wang Z.; Belecciu T.; Eaves J.; Reimers M.; Bachmann M.; Woldring D.: “Phytochemical Drug Discovery for COVID-19 Using High-resolution Computational Docking and Machine Learning Assisted Binder Prediction”. (2022), ChemRxiv. Cambridge: Cambridge Open Engage.

PhD Thesis

- Xu, H.: “Three-dimensional quantitative analysis of bone microvasculature in synchrotron micro-CT imaging. Signal and Image processing.” (2021), Université de Lyon. English.
- Ouzounov, A. P.: “Speech Detection in Speaker Recognition Systems”. (2020), Bulgarian Academy of Sciences. Institute of Information and Communication Technologies.

Conference papers

- Gosgens, M.; Zhiyanov, A.; Tikhonov, A.; Prokhorenkova, L.: “Good Classification Measures and How to Find Them” (2021) *Advances in Neural Information Processing Systems*, 21, pp. 17136-17147.
- Huk, M.; Shin, K.; Kuboyama, T.; Hashimoto, T.: “Random Number Generators in Training of Contextual Neural Networks” (2021) *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12672 LNAI, pp. 717-730.
- Boyadjian, Q.; Vanderesse, N.; Toews, M.; Bocher, P.: “Detecting Defects in Materials Using Deep Convolutional Neural Networks”. (2020), In: Campilho, A.; Karray, F.; Wang, Z.: (eds) “Image Analysis and Recognition”. ICIAR 2020. *Lecture Notes in Computer Science*, vol 12131. Springer, Cham.

Journal papers

- Acuña-Rello, L.; Spavento, E.; Casado-Sanz, M.; Basterra, L.A.; López-Rodríguez, G.; Ramón-Cueto, G.; Relea-Gangas, E.; Morillas-Romero, L.; Escolano-Margarit, D.; Martínez, R.D.; Balmori, J.A.: “Assessment of machine learning algorithm-based grading of *Populus x euramericana* I-214 structural sawn timber”. (2022), *Engineering Structures*, Volume 254, 113826.
- Valencia, O.; Ortiz, M.C.; Sánchez, M.S.; Sarabia, L.A.: “A modified entropy-based performance criterion for class-modelling with multiple classes”. (2021), *Chemometrics and Intelligent Laboratory Systems*, 217, art. no. 104423.
- Stapor, K.; Ksieniewicz, P.; García, S.; Wozniak, M.: “How to design the fair experimental classifier evaluation”. (2021), *Applied Soft Computing*, 104, art. no. 107219.
- Shah, D.A.; de Wolf, E.D.; Paul, P.A.; Madden, L.V.: “Accuracy in the prediction of disease epidemics when ensembling simple but highly correlated models”. (2021), *PLoS Computational Biology*, 17 (3), art. no. e1008831.
- Chicco, D.; Starovoitov, V.; Jurman, G.: “The Benefits of the Matthews Correlation Coefficient (MCC) over the Diagnostic Odds Ratio (DOR) in Binary Classification Assessment”. (2021), *IEEE Access*, 9, art. no. 9385097, pp. 47112-47124.
- Chicco, D.; Jurman, G.: “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation”. (2020), *BMC Genomics*, 21 (1), art. no. 6.

- Zirui Wang; Theodore Belecciu; Joelle Eaves; Mark Reimers; Michael Bachmann; Daniel Woldring: “Phytochemical drug discovery for COVID-19 using high-resolution computational docking and machine learning assisted binder prediction”, (2022), Journal of Biomolecular Structure and Dynamics.

Other evidences of relevance:

As impact, the Modified Confusion Entropy (MCEN) metric has been incorporated in 2018 to the `pymc` Python library <https://pypi.org/project/pymc/>

- Subsequently, we presented a contribution in LOD 2019 (International Conference on Machine Learning, Optimization, and Data Science) under the title “*Vital Prognosis of Patients in Intensive Care Units Using an Ensemble of Bayesian Classifiers*” [23] published in Lecture Notes and Computer Science. Afterwards, in 2021, a extended version “*Survival in the Intensive Care Unit: A prognosis model based on Bayesian classifiers*” [7] has been published in Artificial Intelligence in Medicine.

Journal ranking:

Artificial Intelligence in Medicine (2021). Category: Computer Science, Artificial Intelligence.

2021 Journal Impact Factor: 7.011 (32/144, Q1). JIF percentile 78.13

2021 Journal Citation Indicator: 1.26 (34/189, Q1). JCI percentile 82.28

2021 Total Citations: 5,314

Cited by:

So far, and as far as we know, the published works that have referenced [7] (we do not record self-references) are:

Journal papers

- Amador, T., Saturnino, S., Veloso, A., Ziviani, N. Early identification of ICU patients at risk of complications: Regularization based on robustness and stability of explanations (2022) Artificial Intelligence in Medicine, 128, art. no. 102283.
- Karim, M.R., Islam, T., Lange, C., Rebholz-Schuhmann, D., Decker, S. Adversary-Aware Multimodal Neural Networks for Cancer Susceptibility Prediction from Multiomics Data (2022) IEEE Access, 10, pp. 54386-54409.
- Li, X., Zheng, R., Zhang, T., Zeng, Z., Li, H., Liu, J., Association between blood urea nitrogen and 30-day mortality in patients with sepsis: a retrospective analysis, (2021), Annals of Palliative Medicine, vol 10 (11)

- Garcia D.k Garcia K.; ‘‘Artificial Intelligence for Medica Diagnosis, Prognosis, and Treatment: A Brief Overview’’. (2022) *Industrial Vision* 8 (1): 53-54.

Other evidences of relevance:

The paper [7] has received the mention of ‘‘position paper’’ by the journal *Artificial Intelligence in Medicine*.

As impact, an expert system has been implemented in the Hospital de Mataró based on this work. It is expected to develop the expert system in the whole public health system of Catalonia in the context of the project of the Marató de TV3.

- Finally, in 2022, we contributed in *Scientific Reports* journal with the paper titled ‘‘*Bayesian network-based over-sampling method (BOSME) with application to indirect cost-sensitive learning*’’ [6].

Journal ranking:

Scientific Reports (2022). Category: Multidisciplinary Sciences.

2021 Journal Impact Factor: 4.996 (19/73, Q2). JIF percentile 74.66

2021 Journal Citation Indicator: 1.05 (19/134, Q1). JCI percentile 86.19

2021 Total Citations: 699,320

Cited by:

Due to the recent publication, the publication has not yet received citations.

Other evidences of relevance:

As impact, it is expected to develop an R library based on the method proposed in the paper.

Tables 2.13 and 2.14 summarize this information, including quartile of journals and the number of cites which each contribution has achieved.

	Previous conference?	JCR quartile	Scopus cites	Google Scholar cites
BOSME	no	Q2	0	0
ENSEMBLE	yes	Q1	1(c.)+2(JCR)=2	1(c.)+4(JCR)=5
MCEN	yes	Q2	3(c.)+8(JCR)=11	10(c.)+14(JCR)=24

Table 2.13. Summary of contributions. "c." refers to conference paper. As example, in Scopus the conference paper of the third work (MCEN) has 3 references and the journal paper has 8 references. Self-references have not been counted. Last update on September 1st, 2022

	Other impacts
BOSME	R library (under construction)
ENSEMBLE	Position paper, Expert System in Hospital de Mataró
MCEN	Included in pycm3.5 library

Table 2.14. Summary of impacts

Chapter 3

Final conclusions

A few words by way of conclusion

In this Thesis we have made contributions to different moments of the life cycle of Machine Learning from an integral point of view, focusing our attention on the three fundamental stages of the cycle: **preprocessing** of the dataset, **building the predictive model (classifier)**, and **validation** of the model using performance metrics.

Since it is framed in a PhD. Program in Mathematics, the theoretical justification has gained special importance in this Thesis, always trying to highlight the correct conceptual use of the novel proposal over the simple improvement of results with respect to other state-of-the-art works. Throughout the development of the Thesis, the research methodology has been followed using a quantitative analysis, through a case study in one of the works (the application to the ICU patient database).

In this sense, throughout the three works presented in this Thesis we have tried to cover the three general objectives: *basic research*, *algorithmic/computing*, and *applications*. These objectives are concreted into some specific objectives, structured around three topics, the first two with interaction between them: (a) the problem of having a **class unbalanced** database, (b) the **cost-sensitive** approach to classification, and (c) the use of **Shannon's entropy** to measure the disorder of the elements of the confusion matrix.

For each specific objective, we have approached the research following the same three steps: (i) a methodology or procedure that can be improved, or problem, has been identified, (ii) a solution or alternative has been proposed, and (iii) that solution has been tested from a theoretical and/or empirical perspective.

Finally, we emphasize that **Probability**, and specifically its application to Supervised Machine Learning, has been a backbone around which the entire body of the thesis has been built. Indeed, the introduced **BOSME** oversampling method as an alternative to SMOTE is based on the construction of a probabilistic model, which

is a Bayesian network that maximizes its **likelihood** associated with a suitable data subset, which is nothing more than the probability of observing the instances that have actually been observed, calculated with the probability distribution that the Bayesian network represents. And the ensemble of classifiers **EWA** that we have introduced to construct an expert system to help the corps of physicians to make decisions in their clinical practice, is also based on the concept of probability, since the base classifiers on which this meta-classifier is built are probabilistic, and the combination rule is a weighted average based on the probabilities assigned by any of the base classifiers to the classes, presenting the advantage over other combination rules of make predictions consistent with their *Confidence Level*. Finally, the **MCEN** metric that we introduce as a modification of CEN to overcome its unreliability in a twofold sense: the departure of the range where it should be (the interval $[0, 1]$), and the lack of monotonicity when predictive ability monotonically gets better/worse, is defined from probabilities and uses the Shannon's entropy, which is introduced for a discrete probability distribution.

“It is a remarkable fact that a science who started analyzing games of chance end up becoming the most important object of human knowledge”

P.S. Laplace (1749-1827)

Some limitations

Like any research work, this Thesis is not without limits. We describe in the following the limitations we are referring to:

- The experiments carried out in the different works presented in the Thesis have been designed according to the available infrastructure and according to the established timeline. Due to the computational cost of the experiments, they have been prepared to be processed in a reasonable time.
- The pandemic suffered by COVID-19 has prevented further development of the work initiated in the application of the Machine Learning methodology to medical data. Regardless of the paralysis suffered by the project of collaboration with the Hospital de Mataró during the (long) hardest period of the pandemic, for obvious reasons, a hypothetical use of the data from the year 2020 would have biased the proposed models due to the volume of patients infected by the SARS-COV-2 virus.

Future work

From the presented papers, new lines of research arise which are proposed as future work.

- According to oversampling techniques, we will try to deepen the study of the effect of the type of features and the distribution of the class variable in the data set, on the behavior of BOSME, and we will compare it with other methods of oversampling using more data sets as well as modifications on the very proposal introducing tree-width constraints on the learning structure that would lead to less complex structures.
- In regards to the ICU database, the prediction of the variable *length-of-stay in the ICU* is of interest for the optimization of hospital resources, and it would be compelling to develop predictive models for it as an output variable, based on the characteristics of the patients as input variables. Another aspect of interest would be the study of how the incorporation of COVID patients to the database affects the predictive models we learn from it, and the predictions we obtain with them, highlighting the differences that could be observed in behaviour in the ICU between COVID and non-COVID patients, for example.
- With respect to performance measures, in a stage after the Thesis we would like to return to basic research. Specifically, we want to address the problem of ordinal classification within multiclass classification, a particular situation in which the performance metrics have to take into account not only if a classification error is made, but between which classes that error occurs, since the error must penalize more the further apart they are (according to the order given in the classes).

Final disposition

Despite the adversities that the preparation of this Thesis has gone through, such as the fact of having done it remotely, part-time and with a pandemic in between, the thesis has finally been concluded.

I would like to highlight that as far as I know, this is the first Thesis deposited in the department that deals with issues related to Machine Learning, being, in my opinion, relevant in a Mathematics doctorate program, giving rise to new lines of research, adapting so to the new times.

Bibliography

- [1] Samuel, A. L.: “Some Studies in Machine Learning Using the Game of Checkers”. (1988), *I. Computer Games I*. Springer New York. 335–365
- [2] Russell, S. J. ; Norvig, P.: “Artificial intelligence: a modern approach”. (1995), Pearson Education Limited, Malaysia.
- [3] Hinton, Geoffrey; Sejnowski, Terrence: “Unsupervised Learning: Foundations of Neural Computation”. (1999), MIT Press.
- [4] R Core Team. R: “A language and environment for statistical computing”. (2018), R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>
- [5] Delgado, R.; Núñez-González, J. David: “Enhancing Confusion Entropy (CEN) for binary and multiclass classification” (2019), *PLoS ONE*, 14 (1), art. no. e0210264,
- [6] Delgado, R.; Núñez-González, J. David: “Bayesian network-based over-sampling method (BOSME) with application to indirect cost-sensitive learning”. (2022), *Scientific Reports*. 12. 1. pp 1-18
- [7] Delgado, R.; Núñez-González, J.D.; Yébenes, J.C.; Lavado, A: “Survival in the Intensive Care Unit: A prognosis model based on Bayesian classifiers” (2021), *Artificial Intelligence in Medicine*, 115, art. no. 102054
- [8] Chawla, N.; Bowyer, K.; Hall, L.; Kegelmeyer, W.P: “SMOTE: Synthetic Minority Over-sampling Technique”. (2002), *J. Artif. Intell. Res.*, 16, 321-357.
- [9] George, N.I.; Lu, T-P.; Chang, Ch-W: “Cost-sensitive Performance Metric for Comparing Multiple Ordinal Classifiers”. (2016), *Artif. Intell Res.* vol 5(1), pp. 135-143.
- [10] Lunardon, N.; Menardi, G.; Torelli, N.: “ROSE a Package for Binary Imbalanced Learning” (2014), *R Journal*, 6:82-92

-
- [11] Knaus, W.A.; Draper, E.A.; Wagner, D.P.; Zimmerman, J.E.: "APACHE II: a severity of disease classification system". (1985), *Crit Care Med*, 13(10):818–829.
- [12] Matthews, B. W.: "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". (1975), *Biochimica et Biophysica Acta (BBA) - Protein Structure*. 405 (2): 442-451
- [13] Gorodkin, J.: "Comparing two K-category assignments by a K-category correlation coefficient". (2004), *Comput Biol Chem*. Dec;28(5-6):367-74
- [14] Delgado, R.; Tibau, X.A.: "Why Cohen's Kappa should be avoided as performance measure in classification". (2019), *PLoS ONE* 14(9): e0222916
- [15] Wei, J.-M.; Yuan, X.-Y.; Hu, Q.-H.; Wang, S.-Q.: "A novel measure for evaluating classifiers". (2010), *Expert Systems with Applications*, Vol 37, 3799–3809.
- [16] Sigdel M.; Aygun R.: "Pacc - A Discriminative and Accuracy Correlated Measure for Assessment of Classification Results". (2013), *Machine Learning and Data Mining in Pattern Recognition*. Vol 7988. LNCS. pp 281-295
- [17] Valverde-Albacete F.J.; Peláez-Moreno C.: "100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox". (2014), *Plos One*. Vol 9, Num 1, 1-10.
- [18] Henrion, M.: "Propagating Uncertainty in Bayesian Networks by Probabilistic Logic Sampling." (2019), Editor(s): John F. Lemmer, Laveen N. Kanal, *Machine Intelligence and Pattern Recognition*, North-Holland, vol 5, pp. 149–163,
- [19] Zadrozny, B.; Langford, J.; Abe, N.: "Cost-sensitive learning by costproportionate example weighting." (2003), *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)*.
- [20] Delgado R.; Tibau X.A.: "Measuring Features Strength in Probabilistic Classification" (2019). In: Medina J. et al. (eds) "Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations." (2018), *Communications in Computer and Information Science*, vol 853. Springer, Cham.
- [21] Jurman, G.; Riccadonna, S.; Furlanello, C.: "A Comparison of MCC and CEN Error Measures in Multi-Class Prediction". (2012), *Plos One*. Vol 7, Num 8, 1–8.
- [22] Delgado, R.; Núñez-González, J.D.: "Enhancing confusion entropy (CEN) as measure for evaluating classifiers" (2019), *Advances in Intelligent Systems and Computing*, 771, pp. 79-89.

- [23] Delgado, R.; Núñez-González, J.D.; Yébenes, J.C.; Lavado, A.: “Vital Prognosis of Patients in Intensive Care Units Using an Ensemble of Bayesian Classifiers” (2019), *Lecture Notes in Computer Science* (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*), 11943 LNCS, pp. 619-630.

Appendix A

Articles

A.1 Bayesian Network-based Over-Sampling Method



OPEN

Bayesian network-based over-sampling method (BOSME) with application to indirect cost-sensitive learning

Rosario Delgado¹✉ & J. David Núñez-González^{1,2}

Traditional supervised learning algorithms do not satisfactorily solve the classification problem on imbalanced data sets, since they tend to assign the majority class, to the detriment of the minority class classification. In this paper, we introduce the Bayesian network-based over-sampling method (BOSME), which is a new over-sampling methodology based on Bayesian networks. Over-sampling methods handle imbalanced data by generating synthetic minority instances, with the benefit that classifiers learned from a more balanced data set have a better ability to predict the minority class. What makes BOSME different is that it relies on a new approach, generating artificial instances of the minority class following the probability distribution of a Bayesian network that is learned from the original minority classes by likelihood maximization. We compare BOSME with the benchmark synthetic minority over-sampling technique (SMOTE) through a series of experiments in the context of *indirect cost-sensitive learning*, with some state-of-the-art classifiers and various data sets, showing statistical evidence in favor of BOSME, with respect to the expected (misclassification) cost.

Abbreviations

BOSME	Bayesian network-based over-sampling method
SMOTE	Synthetic minority over-sampling technique
SMOTE-NC	Synthetic minority over-sampling technique-nominal continuous
SVDD	Support vector data description
G-SMOTE	Variant of SMOTE that allows the generation of synthetic instances in a geometric region around the selected instances
BN	Bayesian network
DAG	Directed acyclic graph (the graphical part of a BN)
PA	Set of nodes that are parents, in the DAG, of a given node
MLE	Maximum likelihood estimation method for parameters estimation
LS	Logic Sampling algorithm
ROSE	Random over-sampling examples
LR	Logistic regression
RF	Random forest
SVM	Support vector machine
RBF	Radial basis function kernel

In classification, an imbalanced data set is one with a skewed class distribution. We can assume we mean binary class data sets (otherwise non-minority classes can be merged into a single *majority* class), with a majority class (*negative*), and the minority class (*positive*) being generally the one we are most interested in predicting.

Imbalanced data sets are pervasive across a multitude of fields, making it difficult for machine learning algorithms to identify the minority cases. In fact, detecting instances belonging to the minority class is generally difficult, and the cost associated with misclassifying them (*false negative*) is often much higher than that of misclassifying an instance of the majority class (*false positive*). There are many real-world situations, such as spam detection, fraud identification, disease diagnosis, or vital prognosis, where misclassifying a positive class is clearly

¹Department of Mathematics, Universitat Autònoma de Barcelona, Campus de la UAB, 08193 Cerdanyola del Vallès, Spain. ²Department of Applied Mathematics, University of the Basque Country (UPV/EHU), 29 Otaola Av., 20600 Eibar, Spain. ✉email: delgado@mat.uab.cat

worse than misclassifying a negative class. For example, in¹ the minority class is the death of the patient in the ICU, and the cost of a false negative error, corresponding to classifying a patient who is going to die as a survivor, implies failing to recognize the severity of the situation and includes postponing or ruling out treatments that could actually improve the patient's life expectancy, revealing the seriousness of this error.

This simple example shows the practical inadequacy of classical cost-insensitive classification, which focuses on maximizing accuracy but does not take into account the costs associated with different types of classification errors. This is because, due to the disparity of the class distribution, the algorithms learned from the data set tend to assign the majority class, misclassifying the minority cases, but at the same time giving the false impression of high accuracy. That is, algorithms learned from an unbalanced data set are biased towards the majority class and fail to learn the underlying patterns that distinguish between classes, so they are prone to overfit the majority class.

To address this issue, we focus on probably the most common approach, which is *over-sampling*.

Over-sampling. *Over-sampling* is a suitable methodology to modify the class variable distribution at a data-level stage (pre-processing), before the learning process, to address the problem of learning classifiers from an imbalanced data set. In fact, it consists of creating new synthetic cases of the minority class based on the available data, and then learning the classification algorithm from the enlarged and more balanced data set, instead of using the original one.

The most widely used over-sampling algorithm is SMOTE (synthetic minority over-sampling technique) which was proposed in 2002² as an alternative to the standard random over-sampling, based on interpolation between neighboring cases of the minority class, and became a pioneer for the research community in imbalance classification. Since then, it has become a benchmark for preprocessing imbalanced data for the purpose of learning classifiers from it, and has proven successful in a variety of applications from several different domains. Due to its popularity, SMOTE is the most influential over-sampling algorithm.

SMOTE is designed to deal with continuous features, since it over-samples the minority class by taking each minority class instance and introducing artificial cases by choosing points along line segments connecting it with one of its (typically 5) nearest minority class neighbors in the feature space, and translates the same methodology to the categorical scenario, a methodology that makes no sense in this case, although it may (or may not) give good practical results. In fact, it generates the synthetic instances along the line segments joining neighbors of the k nearest neighbors in the minority class, where k is a hyper-parameter to be specified. More specifically, to generate a new synthetic instance, randomly selects one of two values of any categorical feature: the one corresponding to an instance and one of its neighbors (see details in³). Even works that generalize SMOTE to handle mixed data sets of categorical and continuous features have the same drawback. For example², introduces SMOTE-NC (Synthetic Minority Over-sampling TEchnique-Nominal Continuous) which, as described there, uses the median of the standard deviations of the continuous features of the minority class to define a "distance" between instances that differ in categorical features. Aside from the fact that this makes it impossible for this method to work with categorical data sets that do not contain continuous features (which BOSME can, however), it clearly lacks theoretical justification for this technique, regardless of whether experimentally it can experimentally give good results, since it requires working with the concept of "distance" between values taken by categorical variables. That is, the idea behind SMOTE lacks justification, in our opinion, for categorical features, and this method in no way approximates the distribution of minority instances.

In spite of this, until 2018, the date of the publication of³, a large number of SMOTE-based extensions have been proposed in the specialized literature. And nowadays, SMOTE is still used as the main method of over-sampling. See for example⁴, where it is used in combination with a support vector data description SVDD model, or⁵, where the G-SMOTE algorithm used in classification is extended to regression tasks, being G-SMOTE a variant of SMOTE that allows the generation of synthetic instances in a geometric region around the selected instances instead of in the line segment that joins them. And work continues to find variants that somehow compensate for SMOTE's weaknesses focusing, for example, on the definition of the neighborhood to generate new minority samples using the Euclidean distance (see⁶). However, some works critical of SMOTE have begun to appear recently in the same vein as ours. An example is⁷, where two imbalanced binary data classification methods based on diversity over-sampling by generative methods are proposed as an alternative, just as we propose BOSME.

Another approach to dealing with imbalanced data sets is *under-sampling*, which is just the opposite of *over-sampling*, meaning a removal of instances of the majority class. There is even an intermediate approach, called *hybrid-sampling*, which uses a combination of both. See, for example, the wrapper framework for applying under-sampling and over-sampling using SMOTE in⁸. Nevertheless, in this paper we will focus on *over-sampling*, since it avoids the loss of information that comes with deleting instances.

The objective of our work has been the introduction of a new general methodology of *over-sampling*, which represents a new paradigm, called Bayesian network-based over-sampling method (BOSME), which pre-processes any set of imbalanced data by augmenting it with new cases of the minority class, so that any type of classifier can be learned from the enlarged data set. More specifically, BOSME consists of randomly generating new instances of the minority class using a Bayesian network. This Bayesian network is a model for the probabilistic relationships between the features that is learned from the subset of instances in the original data set that belong to the minority class, with the criterion of maximizing the likelihood.

Bayesian networks. Bayesian networks (BN) are graphical models representing the probabilistic relationships among variables affecting a phenomenon, which can be (and usually are) used for probabilistic inference. For a set of random variables $V = \{X_1, \dots, X_n\}$, a BN is a model that represents their joint probability distribution P , the graphical part of the model consisting of a *directed acyclic graph* (DAG), whose n nodes represent

the random variables. The directed arcs among the nodes represent conditional dependencies (not necessarily causal) governed by the *Markov condition*, which we explain below.

Node X is a “parent” of node Y (and Y is a “child” of X) if there is a directed arc in the DAG from X to Y . We denote by $PA(Y)$ the set of parents of Y . If $PA(Y) = \emptyset$ we say that Y is a *root* node. If there is a *path* from node Z to node T (that is, a concatenation of directed arcs connecting them), then we say that T is a “descendant” of Z ; if a node has no descendants, we say that it is a “leaf”. What characterizes the BN is the **Markov condition**, which can be expressed as follows: *each variable in V is conditionally independent of any of its non-descendants conditioning to the state of all its parents*. Moreover, P can be expressed as the product of the conditional distributions of all nodes given the values of their parents, whenever these conditional distributions exist. This is what is known as **chain rule** and is formally expressed for discrete/categorical variables as follows:

$$P(X_1 = x_1, \dots, X_n = x_n) = \prod_{i=1}^n P(X_i = x_i / PA(X_i))$$

for all the possible values (*instantiations*) x_i for $X_i, i = 1, \dots, n$ (see Neapolitan⁹). The chain rule is very useful because it allows to obtain the joint distribution of the variables from the conditional distributions of each node to its parents, and from the marginal distributions of the root nodes. The probability values of these conditional and marginal distributions are the parameters of the BN to be learned from data.

We adopt the *hill climbing greedy search-and-score-based* structure learning algorithm to learn the DAG, which is the structure of the BN. This algorithm explores the space of the directed acyclic graphs by single-arc addition, removal and reversals, to find the structure that maximizes the score function, taking advantage of the score decomposability to decrease its complexity and make it computationally feasible. For our purpose we choose the logarithm of the likelihood function (**logLik**) as score function to be maximized, since it is a measure of how well the model fits the actually observed data when the parameters are estimated by using the *maximum likelihood estimation* (MLE) method.

Once we have learned from data the BN that represents the probabilistic dependency relations between the variables of V , both the structure and the parameters, we can obtain samples of instances following the probability distribution P entailed by the BN. For that, we will use the **logic sampling** (LS) algorithm¹⁰, that generates instances from the network distribution by randomly selecting values for each node, weighted by the probability of that value occurring. Indeed, LS generates the values of a new instance starting from the root nodes, which are sampled from their marginal probability distributions. The nodes are traversed from the “roots” down to the “leaves”, so at each step the weighting probability is either the marginal or the conditional probability distribution entry for the sampled parent values: once the values for the root nodes have been generated, the values of their children in the DAG are sampled from their conditional distributions (conditional on the values already sampled from the parents), and so on, iteratively, until that all nodes have been visited and the values of the “leaf” nodes have been sampled, and with them, those of all the nodes, finishing the process. That the instances generated in this way follow the distribution of P is a consequence of the *chain rule*.

Its character as a graphic model given by the DAG, together with the Markov condition and the Chain rule, which allow obtaining the joint probability distribution of the model variables (and, therefore, any other probability) from the conditional probabilities of each node to its parents in the DAG, make this probabilistic model a really versatile, useful and unique model in the current landscape of machine learning models.

BOSME is original and different from the other *over-sampling* methods in that it generates the new cases from a model chosen using the likelihood criterion: the more likely the model is for cases of the minority class, the more representative of this class the cases artificially generated from the model will be, and thus allow a classifier learned from the enlarged data set to better discriminate the classes.

But how many cases of the minority class must be artificially generated? It depends on what is intended with it. We will make sense of this question and answer it in the context of *cost-sensitive learning*.

Cost-sensitive approach to classification. *Cost-sensitive learning* is a subfield of *machine learning* that takes into account misclassification costs when learning a classifier. It is closely related to the study of classification in the scenario of imbalanced data sets, so they share techniques and procedures (see¹¹). The aim of a cost-sensitive classifier is to minimize the expected cost of (mis)classification.

Cost-sensitive learning techniques can be categorized into two groups: *black box* and *transparent box* (see¹²), which coincide, respectively, with the data-level and the algorithm-level approaches referred in¹³. The second category includes methods that modify the original learning algorithm to take cost into account, which makes it necessary to have a deep understanding of the algorithm itself, and thus the methods are algorithm-dependent. In contrast, the first category (also known as *indirect methods*¹⁴) uses techniques as *sampling*, *relabeling* and *weighting* before the learning of the classifier, to modify the training data set in a pre-processing phase, with the aim of obtaining a desired class distribution based on the misclassification costs. In this paper we focus on the *sampling indirect methods* for cost-sensitive learning, what are shared with the imbalanced classification problem. SMOTE has been chosen as the over-sampling method for the preprocessing of imbalanced data sets in relation to cost-sensitive learning by different authors^{8,15}. In this paper we propose to use BOSME alternatively.

Much effort has been devoted so far to the development of cost-sensitive decision tree learners, but much less to the development of cost-sensitive Bayesian networks. See the recent paper¹⁶, in which direct and indirect approaches to cost-sensitive learning of Bayesian networks were followed, and experimentally compared with a cost-sensitive decision tree learning algorithm, showing that they are better in terms of misclassification costs and accuracy. In¹⁷, the indirect approach has been applied to some state-of-the-art Bayesian network classifiers,

which perform better than when derived from the original training data set, and in¹⁸ cost-sensitive Bayesian networks apply to rock burst prediction.

In¹², the authors used *resampling* and *reject sampling* as cost-sensitive basic indirect methods, the former presenting the risk of severe overfitting, and the latter requiring the averaging of different classifiers to improve predictive performance, which might not be very good since *reject sampling* implies a reduction of cases in the data set. Compared to them, the approach we focus on, based on the use of an *over-sampling* method, is a methodology that avoids these two sources of poor performance: overfitting and data set reduction.

When dealing with an imbalanced data set in the *cost-sensitive approach* scenario, considering *over-sampling* as a *sampling indirect method* and the expected cost as the performance metric, we see as application of the **Folk Theorem** how from the misclassification costs, we can determine a priori how many cases we should artificially generate from the minority class, so that the classifier that maximizes accuracy with the enlarged data set is the same that minimizes the expected cost with the original. In fact, the **Folk Theorem** states that for that, the distribution of the class variable must be modified with a factor proportional to the costs of misclassification. In this way, we can transform any supervised learning problem with costs into a costless one suitable for applying any cost-insensitive classifier learning algorithm, simply by conveniently extending the data entering the learning process with an appropriate number of new artificial instances of the minority class.

The layout of the paper is as follows. In “Section **Bayesian network-based over-sampling method (BOSME)**” we introduce and describe BOSME, including the pseudo-algorithm that implements it. “Section **Application: sampling indirect method for cost-sensitive learning**” explains the use of the **Folk Theorem** to determine the number of new artificial cases that will be generated from the minority class by applying BOSME. “Section **Experiments**” describes the experimental phase to evaluate BOSME and compare it with other methods of *over-sampling* such as SMOTE, considered as benchmark, and ROSE (Random Over-Sampling Examples). The results of these experiments are given in “Section **Results**”, and we conclude with a few words in “Section **Conclusion**”. To lighten “Section **Results**”, we moved to the “**Appendix**” some auxiliary tables.

Bayesian network-based over-sampling method (BOSME)

We introduce BOSME as a theoretically well-motivated over-sampling preprocessing technique that can be used for general data sets. That is, it can be applied both when the features are (or can be transformed into) categorical, when they are mixed (categorical and continuous), and when they are all continuous. The goal is to generate new artificial instances of the minority class, and this method consists of randomly generate them from the joint probability distribution entailed by a Bayesian network that is constructed as the probabilistic model for the dependency relationships between the features in the minority class setting, with the highest likelihood. This makes BOSME a new paradigm for *over-sampling* methods.

The Bayesian network is learned from the subset of the data set corresponding to the instances belonging to the minority class. While parameters learning is carried out following the *Maximum Likelihood Estimation*, structure learning is performed following a score-based structure learning algorithm with the logarithm of the likelihood function (*logLik*) as the score function. In the case of mixed categorical and continuous features, we assume that

- categorical nodes can only have other categorical nodes as parents,
- the distribution of continuous nodes is a conditional linear Gaussian, that is, conditional on any combination of values of the categorical parents, and on any value of the continuous parents, the distribution of a continuous node is Gaussian with a linear function of the values of the continuous parents as mean value.

If all the features are continuous, we assume that they follow a joint Gaussian distribution and each variable is normally distributed, being its mean a linear function of its parents, and having a common standard deviation.

The learned Bayesian network is a pseudo-optimal probabilistic model for the relationship between the features of the minority class, since it reaches a local maximum of the likelihood function. Since the likelihood function is a measure of the goodness of fit of a model to the set of instances for given unknown parameters, it sounds quite natural to randomly generate a sample of new synthetic instances for the minority class following the joint probability distribution of the features entailed by the learned Bayesian network. The intuition of this method is clear, leaving aside its efficacy as an over-sampling method, which we will evaluate in the experimentation section, contrary to what happens with the benchmark over-sampling method SMOTE.

We introduce some notations: denote by S the original (imbalanced) data set, with M instances and with binary class variable V . Let m_+ be the number of instances corresponding to the minority (positive) class in S , and m_- be the number of instances in the majority class, that is, $m_- = M - m_+$. We assume that $\frac{m_+}{M} < 0.5$ (usually, but not necessarily, $<< 0.5$). The original distribution of the class variable V in the data set S is therefore (p_+, p_-) , where $p_+ = \frac{m_+}{M}$ and $p_- = \frac{m_-}{M}$. Therefore, BOSME's goal is to generate a number of new instances, say n , of the minority class, such that in the enlarged data set augmented with the synthetically generated instances, denoted by \tilde{S} , the minority class represents a desired proportion q of the total.

For the sake of explanation, we want to determine the number of instances that the over-sampling method will generate, n , such that

$$\frac{m_+ + n}{M + n} = q.$$

Isolating from this equation, we get that

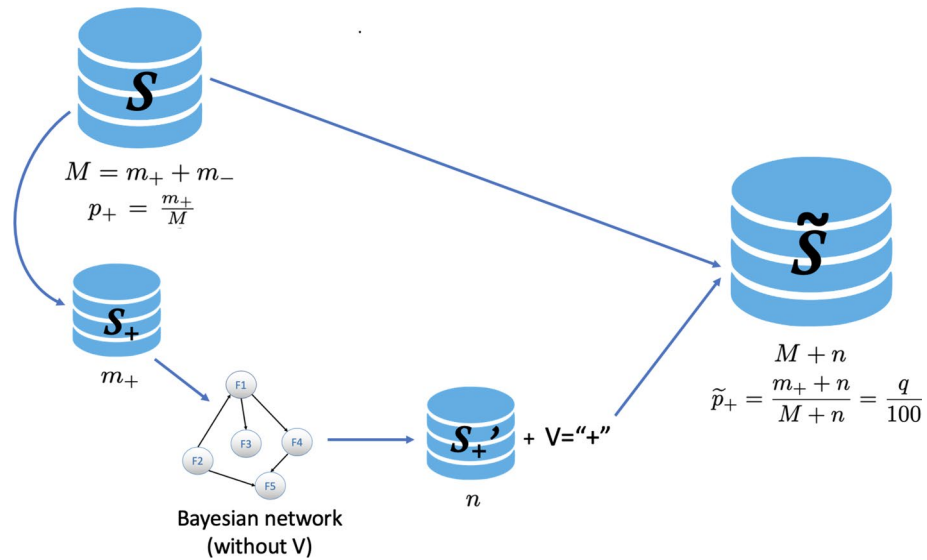


Figure 1. Graphical scheme of the BOSME algorithm.

$$n = \text{round}\left(\frac{qM - m_+}{1 - q}\right) \tag{1}$$

rounding to the nearest integer, since n must be a (positive) integer. In this way, the proportion actually achieved will be approximately q .

The steps of the BOSME over-sampling method are detailed below (see Fig. 1 and the pseudo-code in Algorithm 1):

- Step 1: Extract from the data set S the subset of the minority class, that is, the cases for which V is the positive class “+”, and denote it with S_+ , which is composed of m_+ instances. Only if it makes sense, that is, if the proportion we would like for the minority class is greater than what it initially represents and less than 1 ($\frac{m_+}{M} < q < 1$), we can continue.
- Step 2: Construct a Bayesian network named BN as a model for the relationship between the model features (all variables except the class variable V) from the data set S_+ , using a score-based structure learning algorithm with score the log-likelihood function ($\log Lik$). In this sense, BN is a pseudo-optimal model that (locally) maximizes the probability of the observed instances of the minority class.
- Step 3: Simulate from BN as many new instances as needed by using the LS algorithm, to reach the desired n given by (1), with no missing values, forming a set of complete instances indicated by S'_+ . Note that the class variable V does not appear in the generated synthetic instances, and must be added manually, taking the value of the minority (positive) class.
- Step 4: Bind the synthetically generated instances corresponding to the minority class, S'_+ , and the original S , to obtain the new enlarged data set $\tilde{S} = S \cup S'_+$.

Algorithm 1 BOSME algorithm

Input data set S , variable name V , minority class “+”, proportion q
Output enlarged data set \tilde{S}

- 1: Separate S into S_+ (with $V = “+”$) and S_- (with $V = “-”$).
- 2: Get
 - m_+ , the number of instances in S_+ ,
 - m_- , the number of instances in S_- , and
 - $M = m_+ + m_-$
- 3: **if** $\frac{m_+}{M} < q < 1$ **then**
- 4: Build a Bayesian network BN using a score-based structure learning algorithm with score=log-Likelihood, from $D = S_+$ without V .
- 5: Compute the number of new instances to be generated
 - $n = \text{round}\left(\frac{qM - m_+}{1 - q}\right)$
- 6: Initialize $S'_+ = \emptyset$ and Count = 0
- 7: **while** Count < n **do**
- 8: Randomly generate a new instance \mathbf{x} , following the joint probability distribution entailed by BN
- 9: Update $S'_+ = S'_+ \cup \mathbf{x}$
- 10: Update Count = Count+1
- 11: **end while**
- 12: **end if**
- 13: Tack on S'_+ a new variable called V with all values “+”
- 14: Enlarge the data set by $\tilde{S} = S \cup S'_+$

Note that the amount of over-sampling, n , is a parameter of the algorithm that is deduced by (1) from the input $q \in (\frac{m_+}{M}, 1)$, which is the desired proportion for the over-sampled minority class in the final data set, including the new synthetic instances.

Application: sampling indirect method for cost-sensitive learning

We denote by c_+ and c_- , respectively, the cost associated with misclassifying instances belonging to the positive (*false negative*) and the negative (*false positive*) classes. We assume that $c_+ > c_-$. So, if $\gamma = \frac{c_+}{c_-}$ denotes the **cost rate**, we assume that $\gamma > 1$.

We use a **Folk Theorem** (*Translation Theorem 2.1*¹⁹) to determine the proper proportion q . In fact, this result indicates how to modify the data set to reflect the misclassification costs optimally: if we modify the distribution of the class variable V on the data set to a new one, say $(\tilde{p}_+, \tilde{p}_-)$, multiplying any of the components of the original distribution (p_+, p_-) by a constant proportional to the associated misclassification costs, the resulting distribution has the following property: *choose the classifier that minimizes misclassification error rates (maximizes accuracy) under the new distribution is equivalent to choosing the classifier that minimizes the expected cost under the original distribution.*

The rationale behind this theorem is as follows: consider a probabilistic classifier learned from the modified data set. Given a new instance, if the classifier assigns it to the positive class, the expected cost (with respect to the class distribution of the original data set) is: $0 \times p_+ + c_- p_- = c_- p_-$. Similarly, if the classifier assigns it to the negative class, the expected cost is $0 \times p_- + c_+ p_+ = c_+ p_+$. Then, the assigned class that minimizes the expected cost (with respect to the class distribution of the original data set) is

$$\begin{cases} + & \text{if } c_- p_- < c_+ p_+ \\ - & \text{if } c_- p_- > c_+ p_+, \end{cases}$$

which matches the class that minimizes misclassification error rates under the new distribution, which are \tilde{p}_- if the predicted class is “+”, and \tilde{p}_+ if the predicted class is “-”, provided that

$$\tilde{p}_+ = C p_+ c_+ \quad \text{and} \quad \tilde{p}_- = C p_- c_-$$

for some constant $C > 0$. Since \tilde{p}_+ and \tilde{p}_- must add up to 1, we obtain that the constant necessarily has to be

Data set	Repository	Instances	Minority class	Majority class	Categorical features	Continuous features
Car evaluation	UCI	1728	134 (7.75%)	1594 (92.25%)	6	0
Spect heart	UCI	267	55 (20.6%)	212 (79.4%)	22	0
Balance scale	UCI	625	49 (7.84%)	576 (92.16%)	4	0
Monks	UCI	415	186 (44.82%)	229 (55.18%)	6	0
Post-operative patient	UCI	88	24 (27.27%)	64 (72.73%)	8	0
Tic-tac-toe endgame	UCI	958	332 (34.66%)	626 (65.34%)	9	0
Solar Flare	UCI	1066	43 (4.03%)	1023 (95.97%)	11	0
Breast cancer	UCI	286	85 (29.72%)	201 (70.28%)	9	0
Pizza price	KAGGLE	129	43 (33.33%)	86 (66.67%)	4	2
Haberman	KEEL	306	81 (26.47%)	225 (73.53%)	0	3
Saheart	KEEL	462	160 (34.63%)	302 (65.37%)	1	8
Happiness	UCI	143	66 (46.15%)	77 (53.85%)	6	0

Table 1. Summary of data sets. For the data set **Car evaluation**, the categories *good* and *v-good* on one side, and *acc* and *unacc* on the other, were merged. For **Solar flare**, we have taken the second data section `flare.data2` as the data set as it seems to be more reliable. In **Pizza price**, *extra-cheese* has been taken as output class variable.

$$C = \frac{1}{p_+ c_+ + p_- c_-}.$$

Therefore, to account for misclassification costs in the sampling indirect method for cost-sensitive learning, we will enlarge the original data set by over-sampling the minority class, and by the Folk Theorem we will choose the proportion q that the minority class will represent in the enlarged data set such that the modified distribution of the class variable is

$$\tilde{p}_+ = \frac{p_+ c_+}{p_+ c_+ + p_- c_-} \quad \text{and} \quad \tilde{p}_- = \frac{p_- c_-}{p_+ c_+ + p_- c_-}$$

That is,

$$q = \tilde{p}_+ = \frac{p_+ c_+}{p_+ c_+ + p_- c_-} = \frac{p_+ \gamma}{p_+ \gamma + p_-} = \frac{m_+ \gamma}{m_+ \gamma + m_-} \tag{2}$$

with $\gamma = \frac{c_+}{c_-}$, showing the functional dependence of q on the initial number of instances of each class and on the misclassification costs rate γ .

Remark 1: Note that by (2), $q \in (\frac{m_+}{M}, 1)$. Indeed, since $m_+ \gamma + m_- > m_+ \gamma$ we have that $q < 1$. On the other hand, using $M = m_+ + m_-$,

$$q = \frac{m_+ \gamma}{m_+ \gamma + m_-} > \frac{m_+}{M} \iff \gamma (m_+ + m_-) > m_+ \gamma + m_- \iff \gamma > 1,$$

which is true by assumption.

Experiments

We have performed some experiments to evaluate BOSME and compare it to the benchmark SMOTE. For that, we consider different open access data sets and some state-of-the-art classifiers. In addition to SMOTE, we also compare BOSME with the over-sampling method ROSE²⁰, which is based on a smoothed bootstrap form of re-sampling from data, used to draw artificial samples from the feature space neighborhood around the minority class using a probability distribution centered at a randomly selected case and based on a smoothing matrix of scale parameters.

Data sets. In the experimentation phase, the data sets summarized in Tables 1 and 2 were considered, some with only categorical features, others with mixed features (categorical and continuous), and the rest with all the features continuous. In the data preprocessing phase, the missing cases of the categorical variables have been consigned as a new category different from the others, while for the continuous variables they have been eliminated. Also, when the class variable originally had more than two categories, it has been made binary by category merging.

Classifiers. In the experiments, we used the following three supervised machine learning algorithms for classification to compare BOSME with SMOTE and ROSE.

Data set	Repository	Instances	Minority class	Majority class	Categorical features	Continuous features
Congressional Voting	UCI	435	168 (38.62%)	267 (61.38%)	16	0
Lymphography	KEEL	148	6 (4.05%)	142 (95.95%)	18	0
Diabetes	UCI	520	200 (38.46%)	320 (61.54%)	15	0
Zoo	UCI	101	5 (4.95%)	96 (95.05%)	16	0
Qualitative Bankruptcy	UCI	250	107 (42.8%)	143 (57.2%)	6	0
Dishonest	UCI	322	97 (30.12%)	225 (69.88%)	4	0
Lung	KAGGLE	309	39 (12.62%)	270 (87.38%)	14	1
Indian	KAGGLE	399	195 (48.87%)	204 (51.13%)	5	0
TAE	UCI	151	29 (19.21%)	122 (80.79%)	2	3
Bupa	KEEL	345	145 (42.03%)	200 (57.97%)	0	6

Table 2. Summary of data sets whose results are statistically insignificant to compare BOSME with SMOTE.

1. *Logistic Regression* (LR) is a Supervised Machine Learning method dedicated to classification tasks that has gained popularity during the last two decades, especially in the financial sector. This method uses a linear regression equation to produce discrete binary outputs. We implement it through the R function `stats::glm`, using the argument “family=binomial” (see²¹). Note that `stats` package is a part of R (R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.)
2. *Random Forest* (RF)²² is an ensemble learning method for classification, based on a series of decision trees as basic classifiers (we use the default 500). The output is the mode of the classes of the individual trees (according to the majority vote criterion). We use its implementation in the `mlearning` R package (see “R packages references” at end of paper).
3. *Support Vector Machine* (SVM)²³ uses a representation of the instances of the data set by mapping them as points in a space, in such a way that they are separated in the two categories by a gap as wide as possible. A new instance is then mapped into this space; depending on which side of the gap its point representation falls on, the instance class is predicted to be one or the other. We use the radial basis function (RBF) kernel to define the map, and the implementation of the algorithm in the `mlearning` R package (see “R packages references” at end of paper).

Implementation. The implementation of the experimental phase has been carried out in two stages. In Stage 1, for any of the data sets described in “Section Data sets”, since we will be using k -fold cross-validation (with $k = 10$), we first randomly divide the data set into k folds of roughly the same dimension, and for any fold, we reserve it for later use as a validation set, and use the rest as a training set. Then, for any pair of training/validation sets, we follow the steps below (see the architecture of the proposed implementation in Fig. 2).

- Step 1: Use the BOSME over-sampling technique (Algorithm 1), as well as SMOTE and ROSE, for cost-sensitive learning of the classifiers from the initial training data set, using misclassification costs and the initial distribution of the class variable. That is, determine the proportion q that the minority class “+” should represent in the enlarged training set by (2), and apply the over-sampling technique to obtain it. To learn the Bayesian network in Algorithm 1, we use the hill-climbing algorithm implemented in the R package `bnlearn` by means of the function `hc`. As score we use the option `loglik` when all features are categorical, `loglik-cg` in the conditionally Gaussian mixed Bayesian network case, and `loglik-g` in the Gaussian case with all the features continuous. For the simulation of the new instances corresponding to the minority class from the Bayesian network with the LS algorithm, we use the `rbn` function from the `bnlearn` package. We use the implementation of SMOTE given by the `smote` function of the R package `performanceEstimation`, and function `ROSE` of the R package of the same name, for the implementation of the ROSE oversampling method. See “R packages references” at the end of the paper.
- Step 2: Learn the classifiers introduced in “Section Classifiers” from the enlarged training data sets, obtained in Step 1, with BOSME, SMOTE and ROSE for comparison.
- Step 3: Evaluate the classifiers using the original validation set. As performance metric we use accuracy, as explained in “Section Application: sampling indirect method for cost-sensitive learning”.

Therefore, for each data set we get a 10-dimensional vector of values for any classifier and any of the over-sampling methods as output of Stage 1 (see the output in the scheme depicted in Fig. 2).

To avoid possible bias, in a second stage we repeat the described procedure 10 runs with different seeds for the random splitting of the data set into the k folds. Next, we analyze the results obtained to make comparisons between BOSME, SMOTE and ROSE as shown in the flowchart described in Fig. 3, which encompasses the architecture portrayed in Fig. 2.

For that, we have performed pairwise statistical tests of hypotheses to determine the significance of the results. More specifically, for any data set and for any run, given the classifier, we can perform a paired test to compare the mean (or median, as appropriate) accuracy of the two corresponding samples of size 10 obtained using the

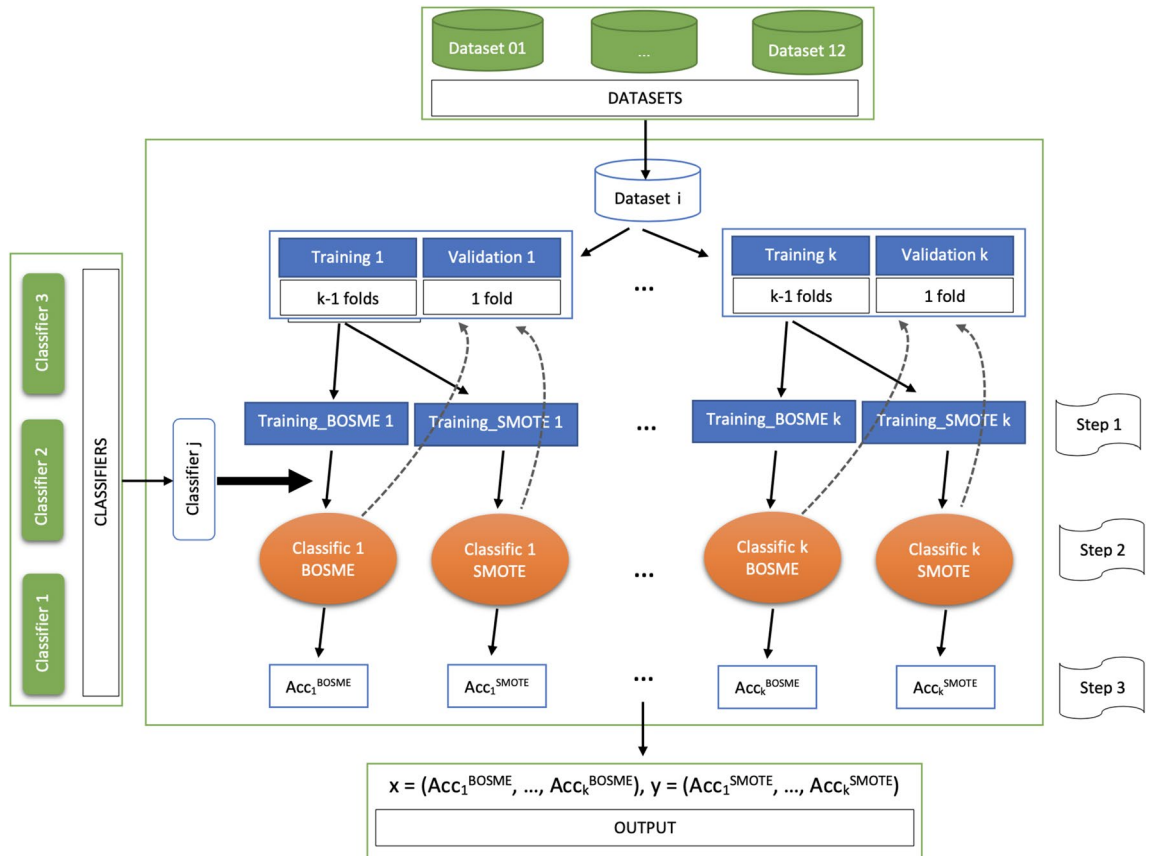


Figure 2. Implementation: Stage 1 architecture to compare BOSME with SMOTE (analogous with ROSE).

BOSME, SMOTE and ROSE over-sampling methods. We use the Shapiro-Wilk normality test to choose between the parametric t-test and the paired non-parametric Wilcoxon signed-rank test, with the criterium that if its p value is < 0.05 , normality cannot be assumed and therefore the last one must be carried out; otherwise, we can use the paired t-test.

In this way, for each data set, each classifier and each over-sampling method, we have two counters that collect the number of runs, of the 10, in which BOSME outperforms the other over-sampling method (counter₊) and the number of runs for which just the opposite happens (counter₋), which are shown as the output of the procedure described in Fig. 3. The above procedure is performed for different values of the cost ratio $\gamma = \frac{c_+}{c_-}$, varying between 5 and 50, from 5 to 5. The results obtained are explained in the next section.

Results

BOSME versus SMOTE. Tables 8, 9, 10, 11 and 12 in the "Appendix" summarize the results of the experimental process when comparing BOSME with SMOTE. They record the number of runs (out of 10 possible) for which there is statistical evidence in favor of BOSME (positive number, counter₊) and in favor of SMOTE (negative number, counter₋). If only a positive number appears in a box, it means that counter₋ = 0, and the same happens if only a negative number appears, which means that counter₊ = 0. As usual, * means statistical significance at the 0.05 level, ** at 0.01, and *** at 0.001.

The corresponding exact Binomial p values (used instead of the McNemar test, because the sample is small) have also been recorded in these tables, provided that they are significant (< 0.05), for any data set and classifier, for the different values of γ . For example, in Table 8 with $\gamma = 10$, for the SVM classifier and the *Post operative* data set, counter₊ = 9 and counter₋ = 0, that is, there are 9 of the 10 runs with statistical differences between BOSME and SMOTE, all in favor of BOSME, with a one-sided p value for the exact Binomial test equal to

$$P(B(n, 0.5) = \text{counter}_+) = P(B(9, 0.5) = 9) = \binom{9}{9} \left(\frac{1}{2}\right)^9 \left(\frac{1}{2}\right)^0 = 0.001953125^{***}$$

One more example: in the same table, for the data set *Flare* with the SVM classifier, counter₊ = 8 and counter₋ = 1, which means that out of 10 runs, there are 9 with statistical differences between BOSME and SMOTE, 8 in favor of BOSME and 1 in favor of SMOTE, giving a one-sided p value for the exact Binomial test in favor of BOSME equal to

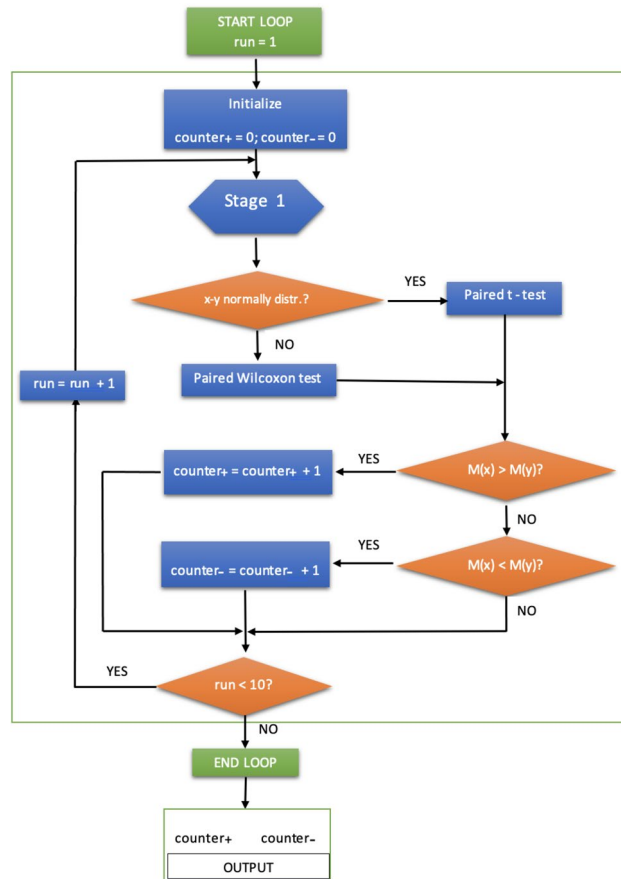


Figure 3. Implementation: Stage 2 procedure flowchart covering the Stage 1 architecture shown in Fig. 2. The letter “M” in the decision boxes denotes the mean/average or the median, depending on whether or not the normality of the distribution of the sample $x - y$ can be assumed, respectively, where x and y are the output of Stage 1 (Fig. 2).

$$P(B(n, 0.5) = \text{counter}_+) = P(B(9, 0.5) = 8) = \binom{9}{8} \left(\frac{1}{2}\right)^8 \left(\frac{1}{2}\right)^1 = 0.01757812^*$$

In Table 3 we summarize by data set the results given in Tables 8, 9, 10, 11 and 12, showing with a positive number for how many classifiers BOSME has been statistically successful against SMOTE (p value < 0.05 and $\text{counter}_+ > \text{counter}_-$). On the contrary, a negative number expresses the number of classifiers with which SMOTE has been significantly better than BOSME (p value < 0.05 and $\text{counter}_- > \text{counter}_+$). Since we have used 3 different classifiers, $+3/-3$ are the best and worst ratings, respectively, in favor of BOSME. The white boxes do not show significant results in any sense. The information from Table 3 is represented in Fig. 4, where we can observe the behavior of BOSME with respect to SMOTE for the different values of the cost ratio γ and any data set. As expected, although this behavior varies with the data set, in all the cases except the *Saheart* data set, BOSME outperforms SMOTE, especially for high values of γ . In Table 3 we also record the value of the β -score, which we enter as the sums per column. So β ranges from -36 to $+36$ ($36 = 12 \times 3$, with 12 data sets and 3 classifiers).

Both in Table 3 and in Fig. 4 the data sets “Haberman” and “Saheart” appear to behave differently of the rest. These data sets have a characteristic that, together with “Pizza”, differentiates them from the rest, and that is the fact that some of the features are continuous, so the Bayesian network that is learned in the BOSME method is no longer a standard but a Gaussian Bayesian network (“Haberman”) or a hybrid Bayesian network (“Saheart”, “Pizza”). Therefore, it is not surprising that with these datasets BOSME does not behave so well with respect to SMOTE, since this last method has been designed for datasets with continuous features, and although SMOTE can also be used with categorical features, seems that in this case BOSME outperforms it. For both “Haberman” and “Saheart”, all the features (in the first), or almost (8 of 9 in the second), are continuous. However, “Pizza” is a hybrid case in which of the 6 features, only 2 are continuous, behaving more in line with the rest of the datasets.

Table 4 below is complementary to Table 3 in summarizing by classifier the results given in Tables 8, 9, 10, 11 and 12. Since we have considered 12 different data sets for which there are significant results, $+12/-12$ are the best and worst ratings, respectively, in favor of BOSME. Figure 5 below represents the information in this table and allows comparing the behavior of BOSME with respect to SMOTE for the different values of the cost ratio γ and any type of classifier. We see in Fig. 5 that although with some classifiers the behavior of BOSME relative

γ	5	10	15	20	25	30	35	40	45	50
Car eval.		+2	+2	+3	+3	+3	+3	+3	+3	+3
Spect heart	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
Balance	+1	+2	+2/-1	+2/-1	+2	+2	+2	+2	+2	+2
Monks		+1	+1	+2	+2	+2	+2	+2	+2	+2
Post-oper.	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
Tic-tac-toe	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3
Solar flare		+2	+2	+1	+1	+1	+1	+2	+2	+2
Breast	+2	+2	+2	+2	+2	+2	+3	+2	+2	+3
Pizza				+1	+2	+2	+2	+2	+2	+2
Haberman	+1/-1	+1/-1	+1/-1	+1/-1	+1/-1	+1/-1	+1/-1	+1/-1	+1/-1	+1/-1
Saheart		/-1	/-1	/-1	/-3	/-2	/-3	/-3	/-3	/-3
Happiness					+1	+1	+1	+1	+1	+1
β -score	+8	+13	+12	+14	+15	+16	+16	+16	+16	+17

Table 3. Summary of the results, by data set, for different values of the cost ratio $\gamma = \frac{c_+}{c_-}$. The numbers in the boxes indicate for how many classifiers, of the possible 3, there is statistical significance in favor of BOSME (positive) or in favor of SMOTE (negative, in bold). For each γ , we take count of the β -score.

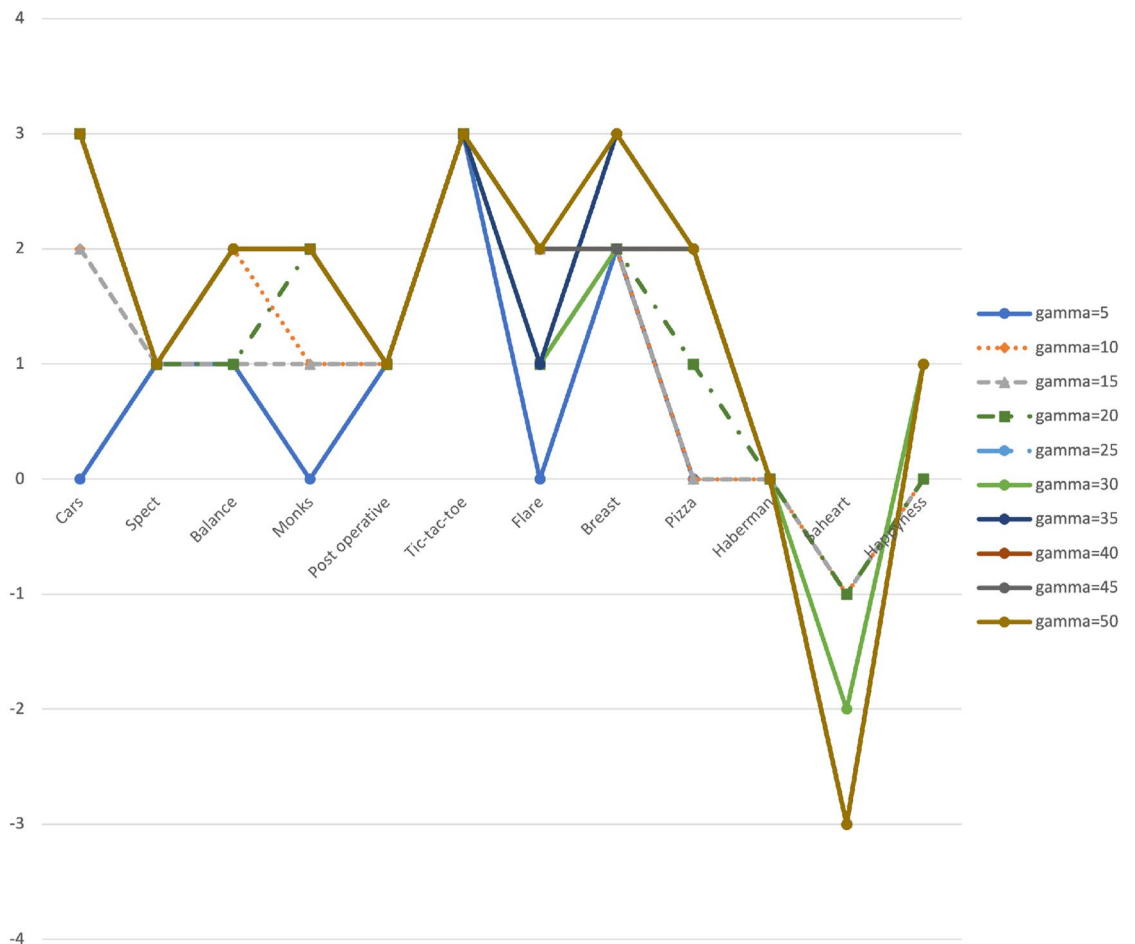


Figure 4. Representation of the information in the Table 3, by data set, for the different values of γ .

to that of SMOTE is better than with others (it seems that with the Logistic Regression it is clearly worse), in general it improves when γ increases, regardless of the chosen classifier.

We can represent the β -score provided by Table 3 with the help of the graph in Fig. 6, where we can observe two interesting results: (a) the β -score turns out to be always positive, and (b) it increases with the cost ratio γ .

We can perform some statistical tests of hypotheses to check the importance of these two observed phenomena.

γ	5	10	15	20	25	30	35	40	45	50
SVM	+5	+7/-2	+7/-2	+8/-2	+8/-2	+8/-2	+8/-2	+9/-2	+9/-2	+9/-2
RF	+3	+6	+6	+7	+8/-1	+8/-1	+8/-1	+8/-1	+8/-1	+8/-1
LR	+1/-1	+2	+2/-1	+2/-1	+3/-1	+3	+4/-1	+3/-1	+3/-1	+4/-1

Table 4. Summary of the results, by classifier, for different values of the cost ratio $\gamma = \frac{c_+}{c_-}$. The numbers in the boxes indicate for how many data sets, of the possible 12, there is statistical significance in favor of BOSME (positive) or in favor of SMOTE (negative, in bold).

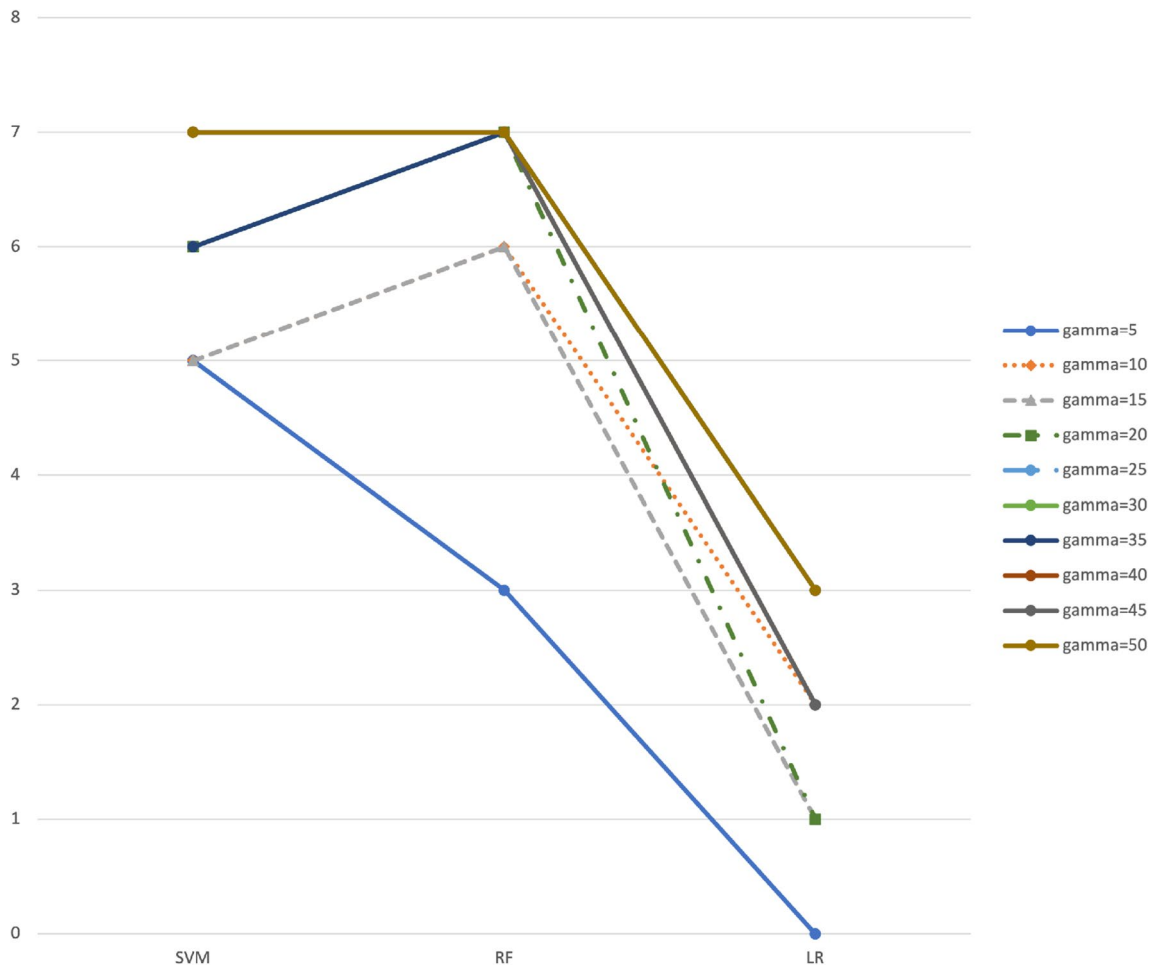


Figure 5. Graphic representation of the information in Table 4, by type of classifier, for the different values of γ .

- (a) **Positiveness of the β -score:** Indeed, in Table 3, of the 10 considered values of γ , the number of them for which the corresponding β -score is strictly positive is 10. The corresponding p value for the exact Binomial test is

$$P(B(n = 10, p = 0.5) = 10) = \binom{10}{10} \left(\frac{1}{2}\right)^{10} \left(\frac{1}{2}\right)^0 = 0.0009765625^{***},$$

which implies a statistical significance in favor of BOSME (the one associated with the positive value of the β -score).

- (b) **Trend monotonicity of the β -score with respect to γ :** We observe in Table 3 that, in general, the values of the β -score increase with γ (see Fig. 6). To check the statistical significance of this trend monotonicity, we use the Mann–Kendall test^{24,25}, which statistically evaluates whether there is a monotonic upward or downward trend of the variable of interest, which is the β -score, relative to an ordered variable like γ (which does not necessarily have to be temporary in nature). A monotone up (down) trend means that the variable consistently increases (decreases) as γ increases. If the Mann–Kendall test gives a significant positive or negative trend (p value < 0.05), which in this case will be positive, Sen’s slope captures the magnitude of

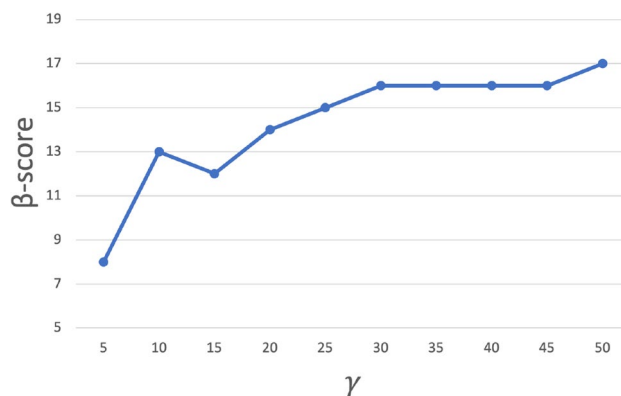


Figure 6. Graphic representation of the β -score for the results of Table 3, where the evolution as the cost ratio γ increases can be observed.

Mann–Kendall			Spearman’s rank correlation	
τ	Two-sided p value	Sen’s slope	ρ	p value
0.649	0.000091***	1.34286, 95% CI: (0.4, 2.5)	0.95672	0.000007***

Table 5. β -score. Mann–Kendall test: τ statistic, two-sided p value and Sen’s slope with a confidence interval (CI) of 95%. Spearman’s rank correlation test: rho statistic and one-sided p value for the alternative hypothesis that $\rho > 0$.

γ	5	10	15	20	25	30	35	40	45	50
Pizza		-1	-1	-1						
Haberman		+1	+1	+1						
Saheart	+2	+2	+2	+2	+2	+2	+2	+2	+3	+2
β -score	+2	+2	+2	+2	+2	+2	+2	+2	+3	+2

Table 6. Summary of the results, by data set, for different values of the cost ratio $\gamma = \frac{c_{+}}{c_{-}}$. The numbers in the boxes indicate for how many classifiers, of the possible 3, there is statistical significance in favor of BOSME (positive) or in favor of ROSE (negative, in bold). For each value of γ , we take count of the β -score.

that trend (that is, it provides an estimate of the average increase in the β -score per increase of a section of γ). The results of the Mann–Kendall test, and also Spearman’s rank correlation test, are given in Table 5 below.

Table 5 shows that there is indeed a significant monotonically increasing trend in the β -score, as the cost ratio γ increases, which is associated with a better behavior of BOSME with respect to SMOTE. The empirical evidence is in the sense that: BOSME outperforms SMOTE for all the values tested in the experimental phase, but it also does so more the higher the value of the cost ratio γ .

BOSME versus ROSE. Comparing BOSME with ROSE similarly to the comparison with SMOTE, we find that there are significant differences only for the 3 data sets: Pizza price, Haberman, and Saheart. The results are in Tables 6 and 7.

Positiveness of the β -score for BOSME versus ROSE: in Table 6 we observe that of the 10 values considered for γ , the number of them for which the corresponding β -score is strictly positive is 10. The corresponding p value for the exact Binomial test is the same as when compared to SMOTE: 0.0009765625*** in favor of BOSME. Since, except in one case, all values of the β -score are constant with γ , there is not statistical significance for trend monotonicity (two-sided Mann–Kendall p value 0.1616).

Conclusion

The introduced BOSME is an over-sampling method that has achieved moderate to good performance against the SMOTE and ROSE over-sampling methods, through a series of experiments, in the context of the indirect cost-sensitive learning approach. This approach consists of enlarge the original imbalanced data set with a number of artificially generated minority instances, which is determined from the misclassification costs. In

γ	5	10	15	20	25	30	35	40	45	50
SVM		+1/-1	+2/-1	+2/-1	+1	+1	+1	+1	+1	+1
RF	+1	+1							+1	
LR	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1

Table 7. Summary of the results, by classifier, for different values of the cost ratio $\gamma = \frac{c_+}{c_-}$. The numbers in the boxes indicate for how many data sets, of the possible 3, there is statistical significance in favor of BOSME (positive) or in favor of ROSE (negative, in bold).

	$\gamma = 5$			$\gamma = 10$		
	SVM	RF	LR	SVM	RF	LR
Car eval.				+10 0.00098***		+7 0.00781**
Spect heart	+6 0.01563*			+6 0.01563*		
Balance	+10 0.00098***			+10 0.00098***	+10 0.00098***	
Monks					+6 0.01563*	
Post-oper.	+7 0.00781**			+9 0.00195**		
Tic-tac-toe	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***
Solar flare				+8/-1 0.01758*	+7 0.00781**	
Breast	+9 0.00195**	+10 0.00098***		+10 0.00098***	+10 0.00098***	
Pizza				+10 0.00098***		
Haberman		+5 0.03125*	-6 0.01563*	-9 0.00195**	+7 0.00781**	
Saheart				-7 0.00781**		
Happiness						

Table 8. Number of runs (of the possible 10) for which there is statistical evidence in favor of BOSME (positive number, counter+) or SMOTE (negative in bold, counter-), and the corresponding exact Binomial p value, for any data set with significant differences, and classifier: Support Vector Machine (SVM), Random Forest (RF) and Logistic Regression (LR). $\gamma = 5, 10$.

this way, we use over-sampling methods and take misclassification costs into account, to extend the data used to feed cost-insensitive supervised learning algorithms.

In fact, the results empirically show that in the context of the *cost-sensitive* approach,

- (a) there is statistical evidence in favor of BOSME dominance over SMOTE,
- (b) this evidence is stronger as the cost ratio γ increases, and for data sets with all categorical features (above continuous or mixed type),
- (c) there is slight evidence in favor of BOSME’s dominance over ROSE, which remains constant as γ varies.

Other highlights of this new method that distinguish it from SMOTE are:

1. BOSME is a novel *over-sampling* method based on a new paradigm, using Bayesian networks.
2. The generation of the artificial instances of the minority class is carried out from a model for the relationship between the features, instead of using the idea of distance between instances, which is the paradigm followed by SMOTE and its derivatives.
3. Maximizing the likelihood function is the criterion for choosing the Bayesian network to use as a model. In this way, the model will be the most plausible given the minority instances, and approximates their probability distribution.
4. The Bayesian network is then a good model that captures the relationship between the features for the minority class, with which generate new instances of this class that are really representative, and from them, learn classifiers that can better differentiate between the two classes, improving their predictive power.
5. This method has wide applicability, for all kinds of features.

	$\gamma = 15$			$\gamma = 20$		
	SVM	RF	LR	SVM	RF	LR
Car eval.	+10 0.00098***		+10 0.00098***	+10 0.00098***	+7 0.00781**	+10 0.00098***
Spect heart	+6 0.01563*			+6 0.01563*		
Balance	+10 0.00098***	+10 0.00098***	-10 0.00098***	+10 0.00098***	+10 0.00098***	-9 0.00195**
Monks		+10 0.00098***		+5 0.03125*	+9 0.00195**	
Post-oper.	+9 0.00195**			+9 0.00195**		
Tic-tac-toe	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***
Solar flare	+6 0.01563*	+8 0.00391**			+8 0.00391**	
Breast	+10 0.00098***	+10 0.00098***		+10 0.00098***	+10 0.00098***	
Pizza				+8 0.00391**		
Haberman	-10 0.00098***	+10 0.00098***		-10 0.00098***	+10 0.00098***	
Saheart	-10 0.00098***			-10 0.00098***		
Happiness						

Table 9. Analogous to Table 8 but with $\gamma = 15, 20$.

	$\gamma = 25$			$\gamma = 30$		
	SVM	RF	LR	SVM	RF	LR
Car eval.	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+8 0.00391**	+10 0.00098***
Spect heart	+6 0.01563*			+6 0.01563*		
Balance	+10 0.00098***	+10 0.00098***		+10 0.00098***	+10 0.00098***	
Monks	+6 0.01563*	+10 0.00098***		+9 0.00195**	+10 0.00098***	
Post-oper.	+9 0.00195**			+9 0.00195**		
Tic-tac-toe	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***
Solar flare		+8 0.00391**			+9 0.00195**	
Breast	+10 0.00098***	+10 0.00098***		+10 0.00098***	+9 0.00195**	
Pizza	+7 0.00781**		+5 0.03125*	+10 0.00098***		+6 0.01563*
Haberman	-10 0.00098***	+10 0.00098***		-10 0.00098***	+10 0.00098***	
Saheart	-10 0.00098***	-5 0.03125*	-6 /0.01563*	-10 0.00098***	-7 0.00781**	
Happiness		+5 0.03125*			+6 0.01563*	

Table 10. Analogous to Table 8 but with $\gamma = 25, 30$.

As a consequence, we conclude that BOSME, which is the method presented in this paper, is a reasonable over-sampling method that has shown very promising results for implementing indirect cost-sensitive learning, in the duel against the benchmark SMOTE, especially in the case of having data sets with all features of categorical type, and for a moderate to high cost ratio. In the case of data sets with mixed features, BOSME does not perform better but it can withstand SMOTE’s onslaught. With respect to ROSE, significant differences are only observed, in favor of BOSME, in the case of mixed features. Therefore, given that its results in the experimental phase

	$\gamma = 35$			$\gamma = 40$		
	SVM	RF	LR	SVM	RF	LR
Car eval.	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***
Spect heart	+6 0.01563*			+6 0.01563*		
Balance	+10 0.00098***	+10 0.00098***		+10 0.00098***	+10 0.00098***	
Monks	+10 0.00098***	+10 0.00098***		+7 0.00781**	+10 0.00098***	
Post-oper.	+10 0.00098***			+10 0.00098***		
Tic-tac-toe	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***
Solar flare		+9 0.00195**		+5 0.03125*	+9 0.00195**	
Breast	+10 0.00098***	+10 0.00098***	+6 0.01563*	+10 0.00098***	+10 0.00098***	
Pizza	+9 0.00195**		+6 0.01563*	+8 0.00391**		+5 0.03125*
Haberman	-10 0.00098***	+10 0.00098***		-10 0.00098***	+10 0.00098***	
Saheart	-10 0.00098***	-9 0.00195**	-6 0.01563*	-10 0.00098***	-9 0.00195**	-8 0.00391**
Happiness		+7 0.00781**			+7 0.00781**	

Table 11. Analogous to Table 8 but with $\gamma = 35, 40$.

	$\gamma = 45$			$\gamma = 50$		
	SVM	RF	LR	SVM	RF	LR
Car eval.	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***
Spect heart	+6 0.01563*			+6 0.01563*		
Balance	+10 0.00098***	+10 0.00098***		+10 0.00098***	+10 0.00098***	
Monks	+9 0.00195**	+10 0.00098***		+10 0.00098***	+10 0.00098***	
Post-oper.	+10 0.00098***			+10 0.00098***		
Tic-tac-toe	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***	+10 0.00098***
Solar flare	+7 0.00781**	+9 0.00195**		+7 0.00781**	+9 0.00195**	
Breast	+10 0.00098***	+10 0.00098***		+10 0.00098***	+9 0.00195**	+5 0.03125*
Pizza	+10 0.00098***		+7 0.00781**	+10 0.00098***		+6 0.01563*
Haberman	-10 0.00098***	+10 0.00098***		-10 0.00098***	+10 0.00098***	
Saheart	-10 0.00098***	-8 0.00391**	-9 0.00195**	-10 0.00098***	-8 0.00391**	-9 0.00195**
Happiness		+8 0.00391**			+7 0.00781**	

Table 12. Analogous to Table 8 but with $\gamma = 45, 50$.

have been very promising, we promote the use of BOSME as an over-sampling methodology with completely general applicability.

In future research, we will try to deepen the study of the effect of the type of features and the distribution of the class variable in the data set, on the behavior of BOSME, and we will compare it with other methods of over-sampling using more data sets. We are also interested in considering extensions/modifications of the version of

BOSME that we present in this paper, for example by introducing tree-width constraints on the learning structure that would lead to less complex structures.

R packages references

- `performanceEstimation` (version 1.1.0). Function: `smote`. Reference: Torgo, L. An Infra-Structure for Performance Estimation and Experimental Comparison of Predictive Models in R (2014). [arXiv:1412.0436](https://arxiv.org/abs/1412.0436) [cs.MS]
- `ROSE` (version 0.0-3). Function: `ROSE`. Reference: Lunardon, N., Menardi, G., Torelli, N. ROSE: a Package for Binary Imbalanced Learning. *R Journal*, 6:82–92 (2014).
- `mlearning` (version 1.0-0). Functions: `mlSvm` and `mlRforest`. Reference: Grosjean, Ph., Denis, K. *mlearning: Machine learning algorithms with unified interface and confusion matrices* (2013). <https://CRAN.R-project.org/package=mlearning>
- `bnlearn` (version 4.7). Functions: `hc` and `rbn`. Reference: Scutari, M. Learning Bayesian Networks with the `bnlearn` R Package. *Journal of Statistical Software* vol. 35(3), pp. 1–22 (2010). <http://www.jstatsoft.org/v35/i03/>

Code availability

The data sets analyzed during the current study are available in the following repositories: (1) UCI: <https://archive.ics.uci.edu/>, (2) KAGGLE: <https://www.kaggle.com/>, (3) KEEL https://sci2s.ugr.es/keel/data_sets.php.

Data availability

The R source code for this project is available upon reasonable request.

Appendix

See Tables 8, 9, 10, 11 and 12.

Received: 24 February 2022; Accepted: 13 May 2022

Published online: 24 May 2022

References

1. Delgado, R., Núñez-González, J. D., Yébenes, J. C. & Lavado, A. Survival in the intensive care unit: A prognosis model based on Bayesian classifiers. *Artif. Intell. Med.* **115**, Article ID 102054, 26 pages (2021).
2. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002).
3. Chawla, N. V., Fernández, A., García, S. & Herrera, F. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **61**, 863–905 (2018).
4. Tao, X. *et al.* SVDD-based weighted over-sampling technique for imbalanced and overlapped data set learning. *Inf. Sci.* **588**, 13–51 (2022).
5. Camacho, L., Douzas, G. & Bacao, F. Geometric SMOTE for regression. *Expert Syst. Appl.* <https://doi.org/10.1016/j.eswa.2021.116387> (2022).
6. Maldonado, S., Vairetti, C., Fernandez, A. & Herrera, F. FW-SMOTE: A feature-weighted oversampling approach for imbalanced classification. *Pattern Recognit.* <https://doi.org/10.1016/j.patcog.2021.108511> (2022).
7. Zhai, J., Qi, J. & Shen, Ch. Binary imbalanced data classification based on diversity over-sampling by generative models. *Inf. Sci.* **585**, 313–343 (2022).
8. Chawla, N. V., Cieslak, D. A., Hall, L. O. & Joshi, A. Automatically countering imbalance and its empirical relationship to cost. *Data Min. Knowl. Disc.* **17**(2), 225–252 (2008).
9. Neapolitan, R.E. Learning Bayesian networks. Prentice Hall Series in Artificial Intelligence (2004).
10. Henrion, M. Propagation uncertainty in Bayesian networks by probabilistic Logic Sampling. In *Uncertainty in Artificial Intelligence 2* (eds Lemmer, J. F. & Kanal, L. N.) 149–163 (Elsevier Science Publishers B.V., 1988).
11. Liu, Z., Gao, Ch., Yang, H., He, Q. A cost-sensitive sparse representation based classification for class-imbalance problem. *Hidawi Publ. Corp. Sci. Programm.* 2016, Article ID 8035089, 9 pages (2016).
12. Zadrozny, B., Langford, J., Abe, N. A simple method for cost-sensitive learning. IBM Technical Report RC22666 (2003).
13. Krawczyk, B. Learning from imbalanced data: open challenges and future directions. *Prog. Artif. Intell.* **5**, 221–232 (2016).
14. Ling, C.X., Sheng, V.S. Cost-sensitive learning. In *Encyclopedia of Machine Learning*, pp. 231–235 (2010).
15. López, V., Fernández, A., Moreno-Torres, J. G. & Herrera, F. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. *Expert Syst. Appl.* **39**, 6585–6608 (2012).
16. Nashnush, E. & Vadera, S. Learning cost-sensitive Bayesian networks via direct and indirect methods. *Integr. Comput. Aided Eng.* **24**, 17–26 (2017).
17. Jiang, L., Li, Ch., Cai, Z., Zhang, H. Sampled Bayesian network classifiers for class-imbalance and cost-sensitive learning. In *IEEE 25th International Conference on Tools with Artificial Intelligence*, pp. 512–517 (2013).
18. Kong, G., Xia, Y., Qiu, Ch. Cost-sensitive Bayesian network classifiers and their applications in rock burst prediction. In *ICIC 2014, LNCS 8588*, pp. 101–112 (2014).
19. Zadrozny, B., Langford, J., Abe, N. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)* (2003).
20. Menardi, G. & Torelli, N. Training and assessing classification rules with imbalanced data. *Data Min. Knowl. Disc.* **28**, 92–122 (2014).
21. Cramer, J.S. The Origins of Logistic Regression, Tinbergen Institute Discussion Papers 02-119/4, Tinbergen Institute (2002).
22. Breiman, L. Random forests. *Mach. Learn.* **45**(1), 5–32 (2001).
23. Cortes, C. & Vapnik, V. N. Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995).
24. Mann, H. B. Non-parametric tests against trend. *Econometrica* **13**, 163–171 (1945).
25. Kendall, M. G. *Rank Correlation Methods* 4th edn. (Charles Griffin, 1975).

Acknowledgements

Technical and human support provided by IZO-SGI, SGIker (UPV/EHU, MICINN, GV/EJ, ERDF and ESF) for the use of powerful computing clusters in the experimental phase is gratefully acknowledged. We also thank the reviewers who, with their comments, helped us to make an improved final version of our work.

Author contributions

Both authors contributed equally to the manuscript.

Funding

The authors are supported by Ministerio de Ciencia, Innovación y Universidades, Gobierno de España, project ref. PGC2018-097848-B-I0.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to R.D.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022

A.2 Ensemble of Bayesian Classifiers



Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

Position Paper

Survival in the Intensive Care Unit: A prognosis model based on Bayesian classifiers

Rosario Delgado^a, J. David Núñez-González^{a,b,*}, J. Carlos Yébenes^c, Ángel Lavado^d^a Department of Mathematics, Universitat Autònoma de Barcelona, Campus de la UAB, 08193 Cerdanyola del Vallès, Spain^b Department of Applied Mathematics, Engineering School of Gipuzkoa – Eibar Section, University of Basque Country (UPV/EHU), Otaola Av. 29, 20600 Eibar, Gipuzkoa, Spain^c Sepsis, Inflammation and Critical Patient Safety Research Group, Critical Care Department, Hospital de Mataró, Mataró, Spain^d Information Management Unit, Maresme Health Consortium, Hospital de Mataró, Mataró, Spain

ARTICLE INFO

Keywords:

Intensive Care Unit

Mortality risk

Bayesian classifier ensemble

Area Under the Curve

F-score

APACHE II

ABSTRACT

We develop a predictive prognosis model to support medical experts in their clinical decision-making process in Intensive Care Units (ICUs) (a) to enhance early mortality prediction, (b) to make more efficient medical decisions about patients at higher risk, and (c) to evaluate the effectiveness of new treatments or detect changes in clinical practice. It is a *machine learning* hierarchical model based on Bayesian classifiers built from some recorded features of a real-world ICU cohort, to bring about the assessment of the risk of mortality, also predicting destination at ICU discharge if the patient survives, or the cause of death otherwise, constructed as an ensemble of five base Bayesian classifiers by using the average ensemble criterion with weights, and we name it the Ensemble Weighted Average (EWA).

We compare EWA against other state-of-the-art *machine learning* predictive models. Our results show that EWA outperforms its competitors, presenting in addition the advantage over the ensemble using the majority vote criterion of allowing to associate a confidence level to the provided predictions. We also prove the convenience of locally recalibrate from data the standard model used to predict the mortality risk based on the APACHE II score, although as a predictive model it is weaker than the other.

1. Introduction

Medical care is one of the most exciting frontiers in data mining and *machine learning*. Although the methodology of prognostic research, including the prediction rules and the approaches to validate them, is still relatively underdeveloped, accurate *prognosis*, which refers to a prediction of the course and outcomes of a patient based on the most likely trajectory of a disease or health problem, has become a key concept in patient care today.

Clinical decision-making for critically ill patients admitted in Intensive Care Units (ICU) is a costly and complex process, which suffers from excessive variability between the opinion of physicians, since it is largely driven by experience and instinct [1,2]. Apart from age, comorbidities or organ failures, there are other aspects related to the

death of patients in ICU, such as delay on attention or inadequate management, which are also linked to the length of stay and costs, as well as to the decrease in quality of life at ICU discharge in survivors [3–5]. In order to improve the quality of the attention, it is important to establish protocols for the management of the healthcare process [6,7].

The traditional approach to improve the performance of ICUs is founded on the development of scores which try to predict the likelihood of negative outcomes (e.g. risk of death). From them, the Acute Physiology And Chronic Health Evaluation (APACHE) is a commonly used scoring system to quantify the severity of illness and to group adult ICU patients by predicted risk of mortality, based on patient data corresponding to the first 24 h after admission to the ICU. This prediction is carried out by means of a logistic regression model in which APACHE is one of the regressors, validated on previous groups of ICU patients [8].

* Corresponding author at: Department of Mathematics, Universitat Autònoma de Barcelona, Campus de la UAB, 08193 Cerdanyola del Vallès, Spain and Department of Applied Mathematics, Engineering School of Gipuzkoa – Eibar Section, University of Basque Country (UPV/EHU), Otaola Av. 29, 20600 Eibar, Gipuzkoa, Spain.

E-mail addresses: delgado@mat.uab.cat (R. Delgado), josedavid.nunez@ehu.eus (J.D. Núñez-González), jyebenes@cscdm.cat (J.C. Yébenes), alavado@cscdm.cat (Á. Lavado).

<https://doi.org/10.1016/j.artmed.2021.102054>

Received 30 October 2020; Received in revised form 1 March 2021; Accepted 17 March 2021

Available online 23 March 2021

0933-3657/© 2021 Elsevier B.V. All rights reserved.

Although there are different versions of this score, APACHE II [9] is still the most used today (see for example the recent works [10], [11] and [12]). This traditional approach based on APACHE II and its successive versions presents some limitations, such as:

- (a) not incorporating variations between units or regions,
- (b) better behaviour in large populations than in small ones, and
- (c) stiffness as predictive model, since if the value of any of the key variables for the patient is not known (variables related with the kind of admission and the severity score APACHE itself), cannot provide a prediction,

which make it unsatisfactory. With its more than 40 years, in some sense APACHE score has become obsolete, taking into account the evolution of medical practice today. In addition, the impact of age on survival, one of the items that scores the most on it, has changed, as well as the life expectancy of neoplastic, coronary or HIV patients, for example. APACHE has not adapted to this new reality.

Thus, there is a need to find out new methods to address these shortcomings and to improve the predictions of the risk of mortality for ICU patients. The application of Artificial Intelligence to Medicine to build predictive prognostic models may represent an opportunity for improvement over scale-based models, such as the one that uses the APACHE score. It is our purpose in this work to present a machine learning methodology that will be validated with a real database. Experimentally we see that it provides good results and avoids the weak points of the traditional approach, so it proves to be a better alternative to the regression models based on the APACHE score.

1.1. Literature review

Different data-driven models have been considered in the literature to support medical experts in their clinical decision-making process in hospital ICUs, to improve quality and for benchmarking purposes on the one hand, and for a greater personalization of care on the other. These models could reduce the inter-clinical variability and are able of treating a large number of variables, finding complex relationships between them. In recent years we can find several works on the state-of-the-art of the use of quantitative methods to assist in medical prognosis and decision-making in health centers and hospitals [13–15]. Some of these methods use *machine learning* tools to face with different situations; in particular, there have been several attempts to apply *machine learning* to improve the management of ICUs (see [16–20]).

Examples of recent works are: [21], in which the authors introduce a predictive model for the survival probability via support vector machines (SVM), making comparisons with other based on logistic regression (LR), where the APACHE score is recalibrated, and they show that SVM outperforms others. Benchmark results for mortality and length of stay predictions using Deep Learning have been presented in [22], where an ensemble of *machine learning* models and some scores have also been considered, using the Medical Information Mart for Intensive Care III (MIMIC-III) publicly available dataset [23], and showing that Deep Learning models consistently outperform the other approaches. To predict the risk of death from some quantitative measures based on the heart rate signals of ICU patients suffering cardiovascular diseases, eight supervised classifiers have been introduced in [24] with the MIMIC-III dataset: decision tree, linear discriminant, logistic regression, SVM, random forest (RF), boosted trees, Gaussian SVM, and K-nearest neighbors (K-NN), showing that the former performs better than the others. A deep multi-scale convolutional architecture trained on the MIMIC-III dataset for mortality prediction has been introduced in [25], to address the problem that although deep neural networks are able to outperform the score approach, they suffer from lack of interpretability. The same problem has been considered in [26], proposing a different solution: an interpretable Bayesian neural network architecture which offers the flexibility of neural networks

without sacrificing interpretability in terms of the selected features, and that has been evaluated using two real-world ICU datasets, MIMIC-III and CENTER-TBI [27]. We finish this review on related works with [28] and [29], which use multiple *machine learning* methods to improve prediction performance.

1.2. Black versus white box models in pattern recognition

Pattern recognition consists of classifying objects or individuals, which are described by a set of characteristics or features, using a model built on the basis of some data, assigning them a class label. In the *supervised learning* pattern recognition problems, each object or individual in the data comes with an observed label, and all the information relating to an object or individual forms a “case”. Then, the task of pattern recognition is to construct (train) a model, that is, a *classifier*, to assign a class to each new case.

Neural networks (NN), which are designed to mimic the performance of the human brain, is by far the most widely used *machine learning* methodology for pattern recognition in the environment of an hospital ICU so far. Its major weakness, however, is that the procedure by which NN discovers relationships or patterns in the data is hidden or opaque (it is said that NN is a *black box*) and therefore it is not easily understood nor explained. That is, although the predictive capacity of this classifier usually shows its adequacy in practice, the classification knowledge learned by the NN turns out to be obscure.

Instead, using Bayesian Networks (BN) is a good way to model situations under uncertainty since, unlike what happens with the *black box* models, they are characterized by being *white box* models that show the relationships and patterns found between the variables in a completely understandable (transparent) way. For this reason BN have been gaining popularity as classifiers in health care applications, thanks to their versatility and power. Just to mention some examples, they have been used in public health evaluation [30], for risk assessment with emerging diseases [31] or for medical diagnosis [32]. BN have been used in the Intensive Care Unit to evaluate EEGs [33] and to establish prognosis in patients with head injuries [34]. To finish this brief summary, the application of Dynamic Bayesian network has been useful for the prediction of organic failure sequences in patients admitted to the ICU in [35].

The authors of [36] use Naive Bayes, jointly with other *machine learning* methodologies, as predictive tools for the inference of lactate level and mortality risk regarding sepsis. Just one comment on this: although Naive Bayes is a particular case of BN, it is not a *white box* model but a *black box*, since its structure (DAG) is fixed and not learned from the data, so it does not reflect the dependency relationships between the variables included in the model. Naive Bayes is based on a very strong independence assumption between the features conditioned to the class variable; despite that, it has shown to work quite well in many complex real-world situations.

1.3. The methodology

With the aim of helping in the vital prognosis of patients admitted to the ICU of a hospital, we propose a machine learning methodology for pattern recognition purpose, consisting in the use of ensembles of BN to build a hierarchical predictive prognosis model.

The idea behind the *ensembles of classifiers* is to combine several individual classifiers to get a new one that beats them all. It seems a natural strategy since we tend to seek input from different people before making our important decisions, and this is especially so in the field of clinical diagnosis, where the opinions of different experts can be taken into account to reach the final decision about a patient. Instead of putting the emphasis on choosing a good classifier, if there is one, we put it on the combination of various classifiers, in the hope that by combining them, the faults of some will be compensated by the others, and the joint result improves each of the parts.

The prediction process with the hierarchical model consists of two stages: (1) predicting the class variable *Result* (live/die), (2) predicting the class variables *Destination* (at ICU discharge) or *Cause* (of death), depending on the prediction in the previous stage. We consider the cause of death as an essential element in the hierarchical predictive prognostic model, whose prediction can help to improve the evaluation of the quality of the care process at the ICU level. Both stages lean on an ensemble of five base Bayesian classifiers, that we denote by EWA, constructed using the *weighted average* criterion with appropriate weights. This criterion is of the type “*fusion of continuous-valued outputs*”, that unlike what happens with the criteria based on the “*fusion of labels*”, such as the *majority vote*, is based not on the prediction, but on the probabilities assigned to the classes by each of the classifiers that make up the ensemble, and has the advantage of being compatible with the MAP (Maximum A Posteriori) criterion. About the weights, our proposal is to use an adequate transformation of the Area Under the Precision-Recall curve (AUPR), which is used instead of Accuracy as performance metric since our dataset is quite skewed for the class variable.

Up to our knowledge, this is a novel approach for vital prognosis of critically ill patients, both by the fact of using a hierarchical model, and by the use of an ensemble of BN as machine learning methodology for pattern recognition, which is based on the weighted average criterion with weights defined from the AUPR with an adequate transformation.

1.4. Evaluation

We apply the proposed methodology to a real-world dataset of critical patients admitted to the ICU of a hospital. To show its usefulness to aid in the vital prognosis, we carried out an experimental evaluation to make comparisons with other pattern recognition proposals, since the performance of the prediction models appears to be, usually, context dependent.

For that, we compare EWA with other state-of-the-art *machine learning* methodologies, such as NN, SVM and RF, without intending to be exhaustive, but rather to highlight its strengths and weaknesses. We also compare the standard mortality prediction based on the APACHE II score using a logistic regression against a locally recalibrated model that we also built using APACHE II, for which coefficients are estimated from the data by means of a logistic regression, and then compare them both with the EWA ensemble. In addition, we compare EWA with the ensemble but without weights, denoted by EA (or, to be more rigorous, with weights all the same), and also with the ensembles obtained from the same base Bayesian classifiers by using both the majority vote criterion (denoted by MV) and the weighted majority vote criterion (denoted by WMV), that have already been introduced in our proceedings paper [37].

Both from the medical and from the management of the ICU points of view, the hierarchical model based on the EWA ensemble gave interesting results from where we get relevant conclusions. Besides, it allows to associate a reasonable confidence level to predictions, issue on which the ensembles MV and WMV fail. Moreover, we have implemented this model as inference engine of an expert system that helps in the vital prognosis at the ICU level, and developed a computational tool intended to make easier the communication between the medical staff and the expert system.

1.5. A further utility

In each specific context it may happen that some of the features might be irrelevant for the prognosis of patients (they are only “noise”), while others show to be important for prediction, and even some features might be important but only in relation to others. That is, they are not all equally relevant.

As said in Section 1.2, one of the advantages of our predictive prognosis model is that it is a white box and, therefore, allows us to to

Table 1
List of variables (part I).

1. Demographic characteristics	% respect to non-missing values
F_1 : Sex	
Male	63.6%
Female	36.4%
F_2 : Age	Median: 70 Q ₁ , Q ₃ : 59, 7
Ranges:	
<45	8.8%
45–54	9.8%
55–64	19.4%
65–74	24.9%
75–84	26.2%
> 84	10.9%
2. Comorbidities	
F_3 : Charlson comorbidity index	
0	31.7%
1	24.2%
2	15.9%
3	10.5%
>3	17.7%
3. Admission	
F_{17} : Origin (location before ICU admission)	
Ward	20.2%
Operation Room	14.0%
Emergency Room	41.0%
Extra Hospital Emergency	1.7%
Other Hospital	23.1%
F_{18} : Generic syndrome (causing admission)	
Elective Surgical	6.5%
Urgent Surgical	9.8%
Coronary	17.5%
Medical	64.3%
Trauma	1.9%
F_{19} : Sepsis (at admission)	
Yes	35.7%
No	64.3%
Main cause of admission (yes/no)	% of yes
F_4 : ACS (Acute Coronary Syndrome)	18.7%
F_5 : RF (Respiratory Failure)	33.0%
F_6 : Shock	27.1%
F_7 : Coma	7.3%
F_8 : Renal F (Renal Failure)	4.1%
F_9 : Hepatic F (Hepatic Failure)	0.2%
F_{10} : CRA (Cardio Respiratory Arrest)	4.8%
F_{11} : ES (Elective Surgical)	6.7%
F_{12} : Arrhythmia	4.1%
F_{13} : CT (Cranial Trauma)	0.2%
F_{14} : OT (Other Trauma)	1.3%
F_{15} : Intoxication	1.0%
F_{16} : Other syndromes	6.3%

carry out a study (see Section 4) on the importance of the features that are included in the model based on the consideration of two different aspects: *centrality and betweenness* and *feature strength*. Centrality and betweenness, on the one hand, are concepts of the field of Graph Theory and Network Analysis that can be applied to BN to identify the most “influential” variables in the model, in a sense that will be specified. On the other hand, we introduce a measure of the feature strength based on a statistical distance between the *a posteriori* conditional probability distributions of the class variable given different values of a fixed feature. In Section 4 we also consider the odds ratio (OR), which is a well known quantification of the strength of the association between two events, that in our context will be “die” when a fixed feature is present, and when it is absent.

The organization of the rest of the paper is as follows: in Section 2 we present the data set and the hierarchical model that we will use to

Table 2

List of variables (part II). *Destination* = “Morgue” if *Result* = “die”. *Cause* = “Not Dead” if *Result* = “live”. We have merged classes “Septic Complications” and “Non-septic Complications” (1.57% and 1.53%, respectively) for variable *Cause*, into the single class “Complications”.

4. Severity (on first 24 h of admission)	
<i>F</i> ₂₀ : ICU workload (therapeutic requirements)	
Medical monitoring	25.4%
Medical unstable with coma or shock	22.4%
Medical unstable without coma neither shock	21.2%
Post-surgical monitoring	5.1%
Post-surgical unstable	25.9%
<i>F</i> ₂₁ : APACHE II	
	Median: 13
	Q1, Q3: 8, 18.25
Ranges:	
<5	9.0%
5–9	25.6%
10–14	23.6%
15–19	19.6%
20–24	11.5%
25–29	6.2%
30–34	2.7%
>34	1.8%
Outcomes	
<i>Result</i>	
live	85.3%
die	14.7%
<i>Destination</i> (at ICU discharge)	
First Attention Hospital	77.7%
Major Complexity Hospital	7.6%
Morgue	14.7%
<i>Cause</i> (of death)	
Cause of Admission	11.2%
Complications	3.1%
Not Dead	85.7%

predict the risk of mortality in the ICU, as well as the destination for those patients who are expected to survive, or the cause of death for the rest. Specifically, we introduce Bayesian networks, used as base models, and the ensembling of classifiers. We also explain the standard method used to predict the risk of mortality at the ICU level based on the APACHE II score, as well as the implementation and validation procedures. Section 3 shows the results we have obtained, Section 5 is devoted to the final comments and conclusions, and the appendices include some figures and tables.

2. Materials and methods

2.1. Dataset description

Our dataset is a cohort of 2510 critical patients admitted to the ICU of the Mataró Hospital (Mataró, Spain) from years 2016 (661 patients), 2017 (693), 2018 (663) and 2019 (493). With the aim of predicting mortality/survival at the ICU first, and then the destination at ICU discharge for patients who survive their stay, or the cause of death for patients who pass away, different features of the patients have been considered (see Tables 1 and 2). ICUs can be thematic (related to a specific kind of patient, as can be neuro-trauma ICU, Coronary unit, medical ICU, post-surgical ICU, ...) or polyvalent, as in our case. To clarify the syndromic classification of critically ill patients, as is usual in polyvalent ICUs, we use four categories:

1. Demographic characteristics
 - Sex (*F*₁)
 - Age (*F*₂)
2. Comorbidities (Charlson comorbidity index, *F*₃)

3. Admission
 - Origin (location of patient before ICU Admission, *F*₁₇)
 - Generic syndrome (that cause admission, *F*₁₈)
 - Sepsis (*F*₁₉)
 - Main cause of admission (*F*₄–*F*₁₆)
4. Severity (on first 24 h of admission)
 - ICU workload (therapeutic requirements, *F*₂₀)
 - APACHE II score (*F*₂₁)

In general, on admission we classify critically ill patients attending to the generic syndrome (*F*₁₈) into

- Surgical (a major invasive procedure is related to the cause or the treatment of admission). We distinguish between “elective” and “urgent”.
- Coronary (admission related to a coronary syndrome).
- Medical (no acute coronary syndrome neither major invasive procedures related to the cause or treatment of admission).
- Trauma (in case of physical external agent damage).

Related to severity on first 24 h of admission, the therapeutic requirements (ICU workload *F*₂₀) of medical (including coronary) or surgical (including trauma) patients, depending on the presence or not of organ failure, can be

	Medical patient	Surgical patient
Presence of organ failure	Medical unstable	Medical monitoring
Absence of organ failure	Post-surgical unstable	Post-surgical monitoring

Stable patients without organ failure just require monitoring to prevent complications, while unstable patients require specific organ failure support, and in this case, we distinguish if they are (or not) in coma or shock. Patients in coma (if Glasgow Coma Score is under 9) or shock (requirement of vasoactive agents to maintain organ perfusion) present the highest mortality and require the highest therapeutic effort.

The presence or absence of sepsis at admission (*F*₁₉) allows a better understanding of the patient’s characteristics. But this is a too nonspecific classification, so we can make an additional classification according to more specific syndromes grouped in the category of “Main cause of admission” (see Table 1). Note that despite that some syndromes can overlap in the same patient, we only identify the most severe condition that causes the ICU admission. For example, in case of a patient in coma, due to a shock secondary to a pancreatitis infarction, the primary specific syndrome is shock (*F*₆ = “yes”), the generic syndrome is *F*₁₈ = “Medical”, without sepsis (*F*₁₉ = “no”) and the ICU workload is *F*₂₀ = “Medical unstable with coma or shock”.

Although some variables (*F*₉, *F*₁₃, *F*₁₄, *F*₁₅) or categories (*F*₁₇ = “Extra Hospital Emergency”, *F*₁₈ = “Trauma”) have very little presence in the current cohort, they have been kept in the study to be able to explore all the possibilities of patient flow, when the database grows.

All variables were, or have been transformed into type factor through a discretization procedure. Age variable has been categorized as well as is done with APACHE II score. In the same way, variable “Charlson comorbidity index”, that can show integer values from 0 to 29, has been discretized into 5 categories: 0, 1, 2, 3 and >3. Missing values are infrequent, and only appear in 9 of the 24 variables, none of them of the “Main cause of admission” category. As expected, patients with missing values in variable *Result* also have missing values in *Destination* (at ICU discharge) if the variable *Result* is *live* and in *Cause* (of death) for patients for which the variable *Result* is *die*.

We observe that 63.6% of patients are male, with a median age of 70 years (average of 67.34) and that mortality at the ICU is 14.7%. “First attention hospital” is the destination at ICU discharge for 91% of patients that survive, and among the patients that did not survive, the cause of death of the 78.4% was the cause of admission, and only for a 21.6% it was a complication suffered at the ICU, of which, half are of a

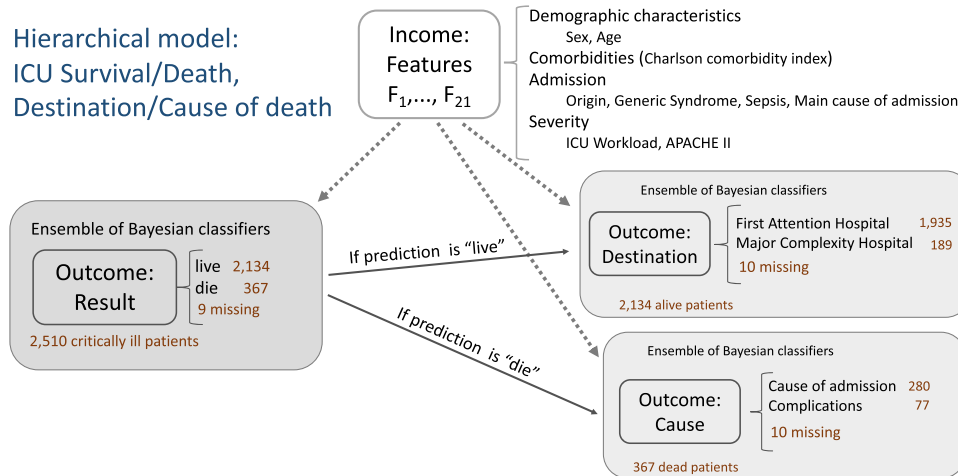


Fig. 1. Processing pipeline of vital prognosis (survival/mortality) prediction for patients in Intensive Care Units.

septic nature.

In Fig. 2 in Appendix A we observe the temporal evolution of the rate of mortality at the ICU, both for the overall population and disaggregated by sex. Instead, in Fig. 3 we see the evolution of the mortality rate with age, also disaggregated by sex and for the overall population. We can observe that the mortality rate is greater for females (except for the youngest patients), and that it increases from young to older people but decreases for the oldest. Finally, in Figs. 4 and 5 we show the distribution of the missing values. The colors in Fig. 5, from white to black in the gray scale, correspond to the different categories that each of the variables takes, while the red color is reserved to indicate missing values. We do not observe any pattern in the distribution of missing values among cases, and as for the variables, those that present missing values are those of the “Admission” and “Severity (on first 24 h of admission)” categories, especially F₂₁: APACHE II, with a 12.67%.

2.2. Building the hierarchical model

The hierarchical model consists of three parts, namely, the classifier for predicting the variable *Result* (live/die), at the first stage, and two more classifiers at the second stage, one for predicting the variable *Destination*, and the other for predicting *Cause*, depending on whether the prediction in the first stage was “live” or “die”, respectively, as can be seen in Fig. 1.

Each of the parts consists of an ensemble of BN built using the *weighted average* criterion with appropriate weights, denote by EWA. Next we explain what Bayesian networks are and how ensembles are constructed.

2.2.1. Bayesian networks

Bayesian networks (BN) are graphical models representing the probabilistic relationships among variables affecting a phenomenon, which are used for probabilistic inference. For a set of random variables, a BN is a model that represents their joint probability distribution *P*, the graphical part of the model consisting of a *directed acyclic graph (DAG)*, whose nodes represent the random variables. The directed arcs among the nodes represent conditional dependencies (not necessarily causal) governed by the **Markov condition**, which establishes that each node in the DAG is independent of those who are not its descendants given its parents are known. When a Bayesian network is used to classify cases into a set of categories or classes, we term it *Bayesian classifier*.

(*Bayesian*) *inference* is the term used to refer to the update of probabilities of the network from a given evidence: we compute a *posteriori* probabilities from evidences and *a priori* probabilities. Prediction of a query variable *X* given the evidence *E* is the instantiation of *X* with the largest *a posteriori* probability, and this probability is said to be the

Table 3

Traits of the five base Bayesian classifiers used to construct the ensembles.

Classifier	Score	Restriction on the directed arcs
BC ₁ (Naive)		Whitelist: from class to each feature. Blacklist: among features
BC ₂	BIC	Whitelist: from class to each feature
BC ₃	AIC	Blacklist: from each feature to class
BC ₄	AIC	Whitelist: from class to each feature
BC ₅ (TAN)		Whitelist: from class to each feature Each feature has an extra incoming arc from other feature

confidence level of the prediction.

To predict the risk of death of critically ill patients, we learn five different base Bayesian classifiers from data, say BC₁, ..., BC₅, by considering the features and the class variable *Result*. This allows further enquiry into the relationships between the features and the vital prognosis, being this an advantage over typically *black box machine learning* methods, such as Neural Networks, which are unable to provide explanations for their predictions. In Table 3 we report the traits of construction of these classifiers, including the score function for the structure learning (learning of the DAG) and the restrictions on the allowed directed arcs, in the form of *whitelist/blacklist* of forced/forbidden arcs. Maximum Likelihood Estimation is used to estimate the parameters.

Naive Bayes has a fixed structure (DAG) which is not learned from the data, and assumes that features are independent of each other given the class, which can be unrealistic in many applications. The other four classifiers in Table 3 are different attempts to improve classification by relaxing this assumption and trying, at the same time, to maintain simplicity and efficiency as much as possible. In particular, TAN (Tree Augmented Naive) relaxes the feature independence assumption of the Naive Bayes through a tree structure, in which each feature only depends on the class and one other feature. Note that both, BC₂ and BC₄ are Augmented Naive Bayes classifiers [39] since the class variable is assumed to be a root node parent of every feature, and the subgraph of the features is an unrestricted Bayesian network.

2.2.2. Ensembles based on the fusion of labels outputs

In [37] we built an ensemble of classifiers to predict the class variable *Result*, say WMV, acronym for Weighted Majority Vote (denoted by EBC there), from the five base classifiers BC₁, ..., BC₅, with the *weighted majority vote* criterion, which is a single-winner voting system but in which more power is given to more “competent” base classifiers. This criterion, as the *majority vote*, falls into the *fusion of labels outputs* ensemble methods. Concretely, fixed a critical patient and a class *j*, we

Table 4
Toy example 1. Confidence level for the prediction given by MV is <0.5.

Classifier	Weights	Prob. of “die”	Prediction	Pred. MV	Pred. WMV
BC ₁	w ₁ = 0.25	p ₁ = 0.55	die		
BC ₂	w ₂ = 0.10	p ₂ = 0.55	die		
BC ₃	w ₃ = 0.05	p ₃ = 0.55	die	die (0.24731 < 0.5)	live (0.66236 > 0.5)
BC ₄	w ₄ = 0.30	p ₄ = 0.10	live		
BC ₅	w ₅ = 0.30	p ₅ = 0.10	live		

Table 5
Toy example 2. Confidence level for the prediction given by WMV is <0.5.

Classifier	Weights	Prob. of “die”	Prediction	Pred. MV	Pred. WMV
BC ₁	w ₁ = 0.25	p ₁ = 0.95	die		
BC ₂	w ₂ = 0.10	p ₂ = 0.95	die		
BC ₃	w ₃ = 0.05	p ₃ = 0.95	die	die (0.95324 > 0.5)	live (0.32725 < 0.5)
BC ₄	w ₄ = 0.30	p ₄ = 0.45	live		
BC ₅	w ₅ = 0.30	p ₅ = 0.45	live		

consider the **discriminant** function $D_j = \sum_{i=1}^5 w_i d_{ij}$ where $d_{ij} = 1$ if classifier i assigns class j to the patient, and 0 otherwise, and $w_i, i = 1, \dots, 5$ are the weights of the five base classifiers, that is, D_j is the sum of weights corresponding to classifiers that assign the patient to class j . The inferred class for the given patient by the WMV classifier is taken to be the one that maximizes the discriminant function. (Note that with $w_i = w_j$ for all $i, j = 1, \dots, 5$, this rule corresponds to the mere criterion of the *majority vote* and we denote by the acronym MV the corresponding ensemble.)

For the assignment of weights to the base classifiers, and bearing in mind that the combination of unbalanced data (14.7% “die” in variable *Result*) and a small sample size (2, 510 patients) prevents the use of *Accuracy* as an evaluation metric in classification, we followed [40] and [41] when considering a measure based on the *Recall* (also called *Sensitivity*) and the *Precision*, with “positive class” the minority class *die*, which provides a good representation of performance assessment in the binary classification: the *Area Under the Precision-Recall curve* (AUPR), being the *Precision-Recall* (PR) curve that obtained by plotting *Precision* over *Recall*. The PR curve provides a more informative picture of the performance of the classifier than the Receiver Operator Characteristic (ROC) curve when dealing with highly skewed datasets, as is our case. For example, in [42] AUPR has been used for mortality and decompensation tasks since the MIMIC-III dataset, which is the one used by the authors for experimentation purposes, suffers from class imbalance. Considering the above, we assign a weight w_i to the base classifier i , which is obtained from its estimated AUPR, denoted by $A_i \in [0, 1]$, in the following way:

$$w_i = \frac{h_i}{\sum_{j=1}^5 h_j}, \quad \text{where } h_i = \log \left(\frac{\frac{1}{2}(A_i + 1)}{1 - \frac{1}{2}(A_i + 1)} \right) \quad (1)$$

Note that

$$\frac{1}{2}(A_i + 1) \in [0.5, 1],$$

and that therefore,

$$\frac{\frac{1}{2}(A_i + 1)}{1 - \frac{1}{2}(A_i + 1)} \geq 1$$

and consequently $h_i \geq 0$. This transformation of the A_i ’s is a dilatation since if $A_i < A_j$, therefore $h_j - h_i > A_j - A_i > 0$. With this assignment of weights, we magnify the relevance of the base classifiers using weights

based on the AUPR metric.

The WMV classifier proved to have a good performance in [37] but it presents the problem of not following the criterion of **maximum a posteriori probability (MAP)**, which is optimal in the sense of minimizing the expected 0–1 loss. For this reason, it is perfectly possible that the predicted class in the binary case may have an associated confidence level less than 0.5, what is counterintuitive and difficult to justify. Let us show two toy examples in Tables 4 and 5, exemplifying this paradoxical situation, with classes “die” ($j = 1$) and “live” ($j = 2$), and the confidence levels in brackets. In both examples, weights are the same and the prediction of any base classifier as well, so the MV classifier gives 3 votes to “die” class and 2 votes for “live” class, that is,

$$D_1 = 3 > D_2 = 2,$$

bringing us to “die” as prediction, while for the WMV,

$$D_1 = w_1 + w_2 + w_3 = 0.4 < D_2 = w_4 + w_5 = 0.6,$$

resulting from this that the prediction is “live”, which is the opposite of the class predicted with the MV. What changes from one example to the other is the probability of “die” of the base classifiers. We can observe that in the first example, the confidence level associated to the prediction provided by the MV is <0.5, while the same happens for the WMV in the second example.

The confidence levels in Tables 4 and 5 have been computed in the following way: for the MV, the confidence level is the probability that the majority of the votes (3, 4 or 5) be for “die”, that is:

$$CL_{MV} = \prod_{\ell=1}^5 p_\ell + \sum_{j=1}^5 ((1 - p_j) \prod_{\substack{\ell=1 \\ \ell \neq j}}^5 p_\ell) + \sum_{j=1}^5 \sum_{\substack{k=1 \\ k \neq j}}^5 ((1 - p_j)(1 - p_k) \prod_{\substack{\ell=1 \\ \ell \neq j, k}}^5 p_\ell),$$

with p_ℓ being the probability of “die” for the ℓ th base classifier. For the WMV, the confidence levels have been computed as the probability that the sum of the weights of the base classifiers that vote “die” be >0.5, which is dependent on weights, by means of the following formula (note that in this case ties are not possible since there is no combination of weights whose sum is exactly 0.5):

$$CL_{WMV} = \prod_{\ell=1}^5 p_\ell + \sum_{r=1}^4 \sum_{\substack{i_1, \dots, i_r=1 \\ (i_1, \dots, i_r) \in \Delta_{w_1, \dots, w_5}^r}} \left(\prod_{h=1}^r (1 - p_{i_h}) \prod_{\substack{\ell=1 \\ \ell \neq i_1, \dots, i_r}}^5 p_\ell \right)$$

Table 6
Toy example 3. Predictions with EA and EWA are different (confidence levels are in brackets).

Classifier	Weights	Prob. of “die”	Prob. of “live”	Prediction	Pred. EA	Pred. EWA
BC ₁	w ₁ = 0.25	p ₁ = 0.90	1 - p ₁ = 0.10	die		
BC ₂	w ₂ = 0.10	p ₂ = 0.90	1 - p ₂ = 0.10	die		
BC ₃	w ₃ = 0.05	p ₃ = 0.90	1 - p ₃ = 0.10	die	die (0.60>0.5)	live (0.55>0.5)
BC ₄	w ₄ = 0.30	p ₄ = 0.15	1 - p ₄ = 0.85	live		
BC ₅	w ₅ = 0.30	p ₅ = 0.15	1 - p ₅ = 0.85	live		
					$\tilde{D}_1 = 0.60$	$\tilde{D}_1 = 0.45$
					$\tilde{D}_2 = 0.40$	$\tilde{D}_2 = 0.55$

where

$$\Delta_{w_1, \dots, w_5}^r = \{(i_1, \dots, i_r) : 1 \leq i_1, \dots, i_r \leq 5, i_1 \neq \dots \neq i_r, \sum_{\ell=1}^r w_{i_\ell} < 0.5\}.$$

2.2.3. Ensembles based on the fusion of continuous-valued outputs

In this paper we consider, as the main novelty regarding [37], two ensembles based on combiners that fall into the fusion of continuous-valued outputs, which are the simple mean (average), denoted by EA hereinafter, and the weighted average with the weights given by (1), denoted by EWA from now on. More specifically, fixed a critical patient and a class *j*, let us introduce the discriminant function $\tilde{D}_j = \sum_{i=1}^5 w_i \tilde{d}_{i,j}$ where $\tilde{d}_{i,j}$ is the probability that classifier *i* assigns the class *j* to the patient, and weights *w_i* are given by (1).

The inferred class for the given critical patient by the EWA classifier is taken to be the one that maximizes the discriminant function \tilde{D} . Therefore, with this criterion, the confidence level associated to class *j* is \tilde{D}_j since $\tilde{D}_1 + \tilde{D}_2 = 1$, which implies compatibility with the MAP criterion. (Note that with $w_i = w_j = 1/5$ for all *i, j* = 1, ..., 5, this rule corresponds to the simple mean combiner EA.) In the toy examples 1 and 2 (Tables 4 and 5, respectively) we can apply both the EA and the EWA, and obtain the respective predictions and confidence levels. Indeed, for the EA in Table 4,

$$\tilde{D}_1 = \sum_{i=1}^5 \frac{p_i}{5} = 0.37 < \tilde{D}_2 = \sum_{i=1}^5 \frac{1-p_i}{5} = 0.63$$

while for the EWA,

$$\tilde{D}_1 = \sum_{i=1}^5 \omega_i p_i = 0.28 < \tilde{D}_2 = \sum_{i=1}^5 \omega_i (1-p_i) = 0.72$$

which bring us to the “live” prediction with both classifiers, and respective confidence levels of 0.63 and 0.72, both >0.5. Analogously, for the toy example in Table 5, both ensembles give as prediction “die”, with respective confidence levels of 0.75 and 0.65, both >0.5. Although in these two toy examples the predictions of EA and EWA have coincided with each other, this does not necessarily have to happen in general. For instance, consider the third toy example in Table 6.

2.3. Implementation

The processing pipeline of the vital prognosis is summarized in Fig. 1. After survival/mortality prediction, outcome variable *Destination* will be predicted in a second step, if the prediction for *Result* is “live”, by using an appropriate classifier which will be an ensemble similar to that used for predicting *Result* but substituting the response variable *Result* for *Dest*. Otherwise, if the prediction for *Result* is “die”, another ensemble similar to that used for *Result* but substituting *Result* as outcome for *Cause*, will be used to predict the cause of death.

Learning and prediction algorithms have been implemented in R

language. For structure learning of the base Bayesian networks BC₂, BC₃ and BC₄, hill-climbing score-based structure learning algorithm has been used, implemented by function **hc** of the *bnlearn* package [43], whereas for BC₅ the **tree.bayes** function has been used which implements the Tree-Augmented naive Bayes classifier. BC₁ represents classic Naive Bayes algorithm, whose structure (DAG) is fixed and must not be learned from the data. The estimation of the other parameters, for the other classifiers, are got using the *maximum likelihood estimation* (MLE) method. We used *gRain* package [44] to carry on the Bayesian inferences.

We make some comparisons among the ensemble we construct from the five Bayesian classifiers and other classifiers from state-of-the-art: Neural Network (NN), Support Vector Machine (SVM) and Random Forest (RF), which have been constructed, respectively, with the functions **mlNnet**, **mlSvm** and **mlRforest** of the *mlearning* package of R,¹ by using the default values in the first two (maximum number of iterations = 1000 for NN and radial kernel for SVM), and 5 trees to generate for RF.

In the literature on predicting the risk of mortality for hospital patients, it is common to use APACHE II, which is a severity of disease classification system that uses basic physiologic principles, to stratify acutely ill patients prognostically by risk of death. The standard approach (see for example [9]) is to compute the individual risk of death (probability of “die” for the variable *Result*) as

$$\frac{e^{\text{logit}}}{1 + e^{\text{logit}}}$$

where *logit* is obtained from the following equation:

$$\begin{aligned} \text{logit} = & -3.517 + 0.146 \times \text{APACHEII} \text{ (the numeric value of } F_{21} \text{)} \\ & + 0.603 \text{ (only if post-emergency surgical, that is, } F_{18} = \text{'UrgentSurgical'})} \\ & + \text{coefficient } \beta \text{ (weight) of the diagnostic categories } F_4 \text{ to } F_{15} \text{ and } F_{19}. \end{aligned} \tag{2}$$

(coefficients β are fixed and have been recorded in Appendix D). When we follow this approach the data is just used for validation, not for training. With the intention of improving this classifier, we construct a locally recalibrated model based on APACHE II using the same training sets used for the base classifiers BC₁ to BC₅, NN, SVM, RF, and the ensembles EWA, EA, WMV and MV, on which the coefficients in Eq. (2) are learned on each training dataset, instead to be fixed. We name this model by LR.APACHEII, and it is built using *logistic regression*, implemented by the function **glm** of R (with argument *family* equal to “binomial”), from the same data used to construct the other classifiers, with regressors: *F₄* to *F₁₅*, *F₁₈* and *F₁₉*. The individual risk of death (probability of “die” for the variable *Result*) with the model LR.APACHEII has been computed as

¹ Grosjean, Ph., Denis, K.; (2013) *mlearning: Machine learning algorithms with unified interface and confusion matrices*. R package version 1.0-0. <https://CRAN.R-project.org/package=mlearning>.

Table 7

Confusion matrices, with the corresponding Accuracy (Acc) and F-score (F), for the fourteen classifiers predicting output variable *Result*, for the first run and the first fold. Predictions are given by row, and observed classes by column, with the order of the classes, +“die”, –“live”. NaN means “not a number”.

$BC_1 : \begin{pmatrix} 22 & 28 \\ 4 & 196 \end{pmatrix}$ Acc=0.872, F=0.2895	$BC_2 : \begin{pmatrix} 14 & 17 \\ 12 & 207 \end{pmatrix}$ Acc=0.844, F=0.491	$BC_3 : \begin{pmatrix} 0 & 0 \\ 26 & 217 \end{pmatrix}$ Acc=0.893, F=NaN	$BC_4 : \begin{pmatrix} 11 & 9 \\ 15 & 211 \end{pmatrix}$ Acc=0.902, F=0.478	$BC_5 : \begin{pmatrix} 10 & 12 \\ 16 & 208 \end{pmatrix}$ Acc=0.886, F=0.417
$NN : \begin{pmatrix} 0 & 0 \\ 26 & 224 \end{pmatrix}$ Acc=0.896, F=NaN	$SVM : \begin{pmatrix} 7 & 5 \\ 19 & 219 \end{pmatrix}$ Acc=0.904, F=0.368	$RF : \begin{pmatrix} 8 & 13 \\ 18 & 211 \end{pmatrix}$ Acc=0.876, F=0.340		
$APACHEII : \begin{pmatrix} 0 & 0 \\ 19 & 200 \end{pmatrix}$ Acc=0.913, F=NaN	$LR.APACHEII : \begin{pmatrix} 2 & 1 \\ 17 & 199 \end{pmatrix}$ Acc=0.918, F=0.091			
$MV : \begin{pmatrix} 11 & 11 \\ 15 & 213 \end{pmatrix}$ Acc=0.896, F=0.458	$WMV : \begin{pmatrix} 11 & 10 \\ 15 & 214 \end{pmatrix}$ Acc=0.900, F=0.468	$EA : \begin{pmatrix} 15 & 11 \\ 11 & 213 \end{pmatrix}$ Acc=0.912, F=0.577	$EWA : \begin{pmatrix} 16 & 11 \\ 10 & 213 \end{pmatrix}$ Acc=0.916, F=0.604	

$$\frac{e^{LR.logit}}{1 + e^{LR.logit}}$$

where *LR.logit* is obtained from the following equation, learned from data:

$$LR.logit = \alpha_0 + \alpha_1 \times APACHE\ II \text{ (the numeric value of } F_{21}) + \alpha_2 \times F_{18} + \alpha_3 \times F_{19} + \sum_{j=4}^{15} \alpha_j \times F_j. \tag{3}$$

That is, the coefficients α of the diagnostic categories are learned from data. In our case, when we learn the model from the complete dataset, the corresponding coefficients α have been recorded in Table 34 in Appendix D, with the *p*-values for statistical significance. Referring to mortality, in Table 34 we can see what is the only protection factor, which is F_4 (in boldface), and what are the risk factors (the remaining in the table).

2.4. Validation and comparison with other classifiers

We choose to carry out the process of *k*-fold cross-validation with *k* = 10 folds to validate our proposed hierarchical model. We use four different performance metrics to make comparisons between the EWA classifier and its single component base classifiers BC_1, \dots, BC_5 , as well as with the other *pattern recognition* methods: NN, SVM and RF, and with the ensembles EA and that based on the fusion of labels outputs introduced in [37], MV and WMV. If a tie takes place with the latter, what happens when the evidence consisting of the patient’s features has an estimated probability equal to zero with any of the five base classifiers, the tiebreaker rule will assign one of the categories at random, with equal probabilities. A further comparison is against the classifiers based on the APACHE II score following the traditional approach, the usual one and the enhancement we proposed, which is the locally recalibrated LR.APACHEII.

We randomize in order to reduce the possible bias due to the (random) choice of the folds in the validation process. Moreover, we repeat the process 20 times, using a different seed (randomly selected) in each case to carry out the partition of the database into the *k* = 10 folds. The metrics used to make comparisons between the classifiers are:

–**AUPR**: as we have already commented, this is considered a good measure when the database is unbalanced with respect to the class variable, as is our case.

–**F-score**: our goal has been to enhance the prediction of the minority class (identified with the “positive” class). For that, our interest is focused on the improvement of *sensitivity (recall)*, which together with *precision* are the two measures that make up the F-score, defined as their harmonic mean.

–**AUC (Area Under the ROC² Curve)**: very popular in the medical literature. An advantage of incorporating it as a metric in the validation process is that the results obtained in our study may be compared with those of others made with different populations and methodologies.

We must highlight the imbalance of the class distribution in the case of the output variable *Result*, with minority class “die” (14.7%), and also in the case of output variable *Destination*, with minority class “Major Complexity Hospital” representing a 8.9% (of the cases with known destination and different from “Morgue”). In the case of the output variable *Cause*, the minority class “Complications” represents 21.6% of the cases with known cause of death, so the imbalance is not so extreme.

Although it is the most common of the metrics, we do not include the *Accuracy* in this study because it is not very suitable in cases of imbalance by the *accuracy paradox*. In Table 7 below we report as illustrative example the confusion matrices obtained in the validation procedure for each the fourteen classifiers predicting output variable *Result*, for the first run and the first fold, jointly with the corresponding Accuracy (Acc) and F-score (F) values. In the matrices, predicted classes are given by row, while observed by column, in order: +“die”, –“live”. Note that the number of observed cases for APACHEII and LR.APACHEII models is 219 while for the rest is 250; the reason is that the first cannot provide any prediction if F_{18}, F_{19} or F_{21} are missing. Although it is only an example and the matrices are subject to variability, from them we can get an idea of what is happening with the different models: APACHEII always predicts “live” (BC_3 and NN only do it sometimes), having exactly the Accuracy given by the proportion of the majority class in the validation set, and F-score cannot be computed. The rest of classifiers sacrifice the correct prediction of all the majority class in order to be able to correctly predict some patients of the minority class, that is, patients that died, which is what we are interested in from a clinical point of view. However, the Accuracy of the ensembles EA and EWA is comparable to that of APACHEII and LR.APACHEII, having a higher F-score value. This idea is confirmed with the statistical comparison among them explained in Section 3.

² The ROC, Receiver Operating Characteristic curve serves to illustrate the capacity of diagnostic of a binary classifier as the discrimination threshold varies; it plots the *sensitivity* (or True Positive Rate) against the False Positive Rate (1-*specificity*).

Table 8

Average over the runs of the averages (\bar{x}) and the standard deviations (s) over the folds, for the metrics AUPR, F-score and AUC, with the different classifiers. Output variable *Result*. In boldface, the top five for each metric.

Result	AUPR		F-score		AUC	
	\bar{x}	s	\bar{x}	s	\bar{x}	s
BC ₁	0.52058 (5)	0.08914	0.54445 (1)	0.06056	0.87230 (3)	0.02858
BC ₂	0.50671	0.08514	0.52763 (3)	0.06642	0.85987	0.03192
BC ₃	0.35161	0.13386	0.06805	0.02496	0.82825	0.04087
BC ₄	0.48450	0.07944	0.49974	0.06621	0.83958	0.03605
BC ₅	0.46424	0.07927	0.47569	0.06794	0.83277	0.03456
NN	0.27294	0.23893	0.43670	0.07689	0.70228	0.18079
SVM	0.43432	0.08309	0.32713	0.08628	0.79698	0.04014
RF	0.37071	0.08110	0.37567	0.08028	0.76864	0.04096
APACHEII	0.37899	0.09393			0.77518	0.04837
LR.APACHEII	0.42621	0.09342	0.30706	0.09075	0.83154	0.03744
MV	0.52467 (3)	0.08276	0.50274	0.06744	0.86440 (4)	0.03027
WMV	0.52317 (4)	0.08316	0.51137 (5)	0.06791	0.86377 (5)	0.03098
EA	0.53829 (2)	0.08510	0.52354 (4)	0.06666	0.87913 (2)	0.02538
EWA	0.54131 (1)	0.08423	0.53270 (2)	0.06766	0.88026 (1)	0.02522

3. Results

3.1. For the variable *Result*

The boxplots in Fig. 6 (Appendix A) correspond to the values of AUPR, F-score and AUC obtained by using k -fold cross validation with $k = 10$, for prediction of the output variable *Result* with positive class “die”, for the first run and the fourteen classifiers considered in the study (including the two based on APACHE II for predicting mortality at the ICU level, which are the standard one and the locally recalibrated LR.APACHEII).

We record the average over the 20 runs of the averages and the standard deviations, \bar{x} and s , respectively, over the folds, for AUPR, F-score and AUC when considering the output variable *Result*, in Table 8. The blank cells indicate that the F-score could not be calculated by the arrangement of the zeros in the confusion matrices generated by the APACHEII model.

From the experiment, we can see that there is a clear advantage for the ensembles, especially EWA and EA, over the rest of the classifiers that have been considered, using AUPR and AUC as performance measures while for the F-score, the best classifiers are BC₁, EWA, BC₂ and EA. That is why we will focus on the comparison between the ensembles EWA, EA, WMV and MV, to each other, in addition to in their comparison with the rest. For each of the metrics, below we detail some of the results.

AUPR: Table 17 (Appendix B) reports for each run if there is a statistically significant (p -value < 0.1) improvement of either EWA or EA, with respect to WMV and/or MV. Here “2” means that there is an improvement over WMV and MV, “1” means that there is only an improvement over one of them, and “0” that there is none for either. In no case are WMV or MV better than EWA or EA. p -values³ are reported in Table 18 and have been adjusted for multiple comparisons between the four ensembles by using the method of Holm-Bonferroni, with the pairwise Wilcoxon signed-rank test [45] to compare matched pairs of samples corresponding to the same run. This statistical test is used as an alternative to the Student’s t-test when the population cannot be assumed to be normally distributed (according to the Shapiro-Wilk test [46], which has been previously performed).

From these tables we see that EWA and EA outperform WMV and MV, and that in 5 runs, there are significant differences among EWA and EA

³ As usual, throughout the paper · denotes significance at 10%, superscript * denotes statistical significance at 5%, ** at 1% and *** at 1%, for all the p -values.

and, in all the cases, EWA shows to be better. This is confirmed in Table 19, where we observe that EWA is significantly better than EA in 8 runs, when we compare only the two and, therefore, the p -values have not been adjusted, and in all the cases, EWA shows to be better. What significance does this fact have? We compute the p -value for the exact Binomial test in order to compare the proportions of cases in which EWA outperforms EA and vice versa, instead of use McNemar test, because the sample is small. The one-sided p -value for the exact Binomial test is $P(B(n = 5, p = 0.5) = 5) = 0.5^5 = 0.03125^*$ when we compare EWA and EA but adjust for the comparison of the four ensembles, which decreases to $P(B(n = 8, p = 0.5) = 8) = 0.5^8 = 0.00391^{**}$ if we consider the non-adjust corresponding to comparison of EWA against EA alone. In both cases there is a statistically significant evidence in favour of EWA as opposed to EA for prediction of variable *Result*, with AUPR as performance measure.

As regards APACHEII and LR.APACHEII, both are clearly worse than any of the ensembles, and we observe significant differences among them, in favour of the latter. Indeed, for 18 runs (see details in Table 20 in Appendix B) there are differences between the medians for the AUPR metric, and in all cases LR.APACHEII turns out to be better than the standard based on APACHE II, with a one-sided p -value for the exact Binomial test: $0.5^{18} = 3.81470 \times 10^{-6***}$.

F-score: We repeat the procedure with the F-score and obtain Tables 21–23 in Appendix B. We observe that EWA and EA outperform WMV and MV (and also that WMV behaves better than MV), and that EWA is better than EA. From Table 23 we have that in 14 runs there are significant differences among EWA and EA, when comparing the only two, and in all the cases, EWA is better, which has the following one-sided p -value for the exact Binomial test: $0.5^{14} = 6.10352 \times 10^{-5***}$, which gives a clear statistical significance in favour of EWA against EA.

If we compare the best of the ensembles, EWA, against the base classifiers (with the adjusted p -values for multiple comparisons) we see that the median of the F-score of both EWA and BC₁ is significantly greater than that of BC₄ and BC₅, and that the median of BC₂ is significantly greater than that of BC₅. We can no obtain statistical significance with respect to BC₃, which has the lowest F-score values due to the large number of missing values. They are not observed either significant differences between EWA, BC₁ and BC₂ as far as the F-score is concerned, if we compare the three with each other as a single block, and even if we compare independently in pairs.

The locally recalibrated LR.APACHEII is clearly worse than the four ensembles when using the F-score as performance metric. Note that it is not possible to calculate the F-score for the standard based on APACHE II since in all cases the prediction with this classifier for the output variable *Result* was the majority class “live”, resulting in degenerate confusion

Table 9

Average over the runs of the averages (\bar{x}) and the standard deviations (s) over the folds, for the metrics AUPR, F-score and AUC, with the different classifiers. Output variable *Destination* In boldface, the top five for each metric.

Destination	AUPR		F-score		AUC	
	\bar{x}	s	\bar{x}	s	\bar{x}	s
BC ₁	0.26687	0.10371	0.28345	0.11281	0.75532 (4)	0.07043
BC ₂	0.27571	0.09357	0.26376	0.10863	0.74604 (5)	0.07077
BC ₃	0.36120 (1)	0.11843	0.33922 (1)	0.12737	0.75596 (3)	0.06692
BC ₄	0.26142	0.09548	0.31516 (2)	0.11396	0.70311	0.07645
BC ₅	0.24706	0.09042	0.26310	0.11213	0.70424	0.07592
NN	0.16834	0.11205			0.67487	0.10562
SVM	0.22107	0.08253	0.13317	0.04624	0.66455	0.07480
RF	0.21222	0.07203	0.21634	0.09616	0.65715	0.066328
MV	0.29333 (4)	0.10329	0.30076 (4)	0.12116	0.73914	0.07266
WMV	0.29598 (3)	0.09814	0.30463 (3)	0.12205	0.73493	0.07158
EA	0.28726 (5)	0.10200	0.28635	0.11838	0.76802 (2)	0.057239
EWA	0.29766 (2)	0.10470	0.29444 (5)	0.11912	0.77201 (1)	0.05715

matrices with a row equal to zero.

AUC: Comparing the fourteen classifiers all at one, we see that the ensembles jointly with BC₁ and BC₂ are the best, the rest being far below. Then, first we compare the ensembles between them. In Table 24 (Appendix B), which is analogous to Table 17, we report for each run if there is a statistically significant (p -value < 0.1) improvement of either EWA or EA, with respect to WMV and/or MV, and the corresponding p -values are in Table 26 (note that they have been adjusted for multiple comparisons between the four ensembles). We see clearly that EWA and EA outperform WMV and MV, and that EWA does with respect to EA in 6 runs. Previously, in Table 25 we record the results of the comparison between only EWA and EA, and in 12 runs EWA shows to be better (one-sided p -value for the exact Binomial test: $0.5^{12} = 0.00024^{***}$). Consequently, EWA is the best of the ensembles.

It only remains for us to compare it with BC₁ and BC₂. In comparing the three at one, with the corresponding adjustment of the p -values, we observe that BC₁ is significantly better than BC₂ in 8 runs, and both are worst than EWA (in 5 and 18 of the runs, respectively, with one-sided p -values for the exact Binomial test: $0.5^5 = 0.03125^*$ and $0.5^{18} = 3.8 \times 10^{-6}^{***}$). Definitely, then, EWA is the best of the classifiers for output variable *Result* attending to AUC metric.

As regards comparison between APACHEII and LR.APACHEII, the latter turns out to be better in all the runs, with a one-sided p -value for the exact Binomial test: $0.5^{20} = 9.5367 \times 10^{-7}^{***}$.

3.2. For the variable *Destination*

Fig. 7 in Appendix A shows the boxplots for AUPR, F-score and AUC for first run and the twelve classifiers, for prediction of the output variable *Destination* with positive class “Major Complexity Hospital”. Analogous results to that of Table 8 are in Table 9. Below we specify a little more the results by metrics.

AUPR: Regarding the comparison between ensembles, Table 28 in Appendix C shows the adjusted p -values for multiple comparisons between the four ensembles MV, WMV, EA and EWA. We can see that EWA clearly outperforms EA, and WMV also outperforms EA in two runs. Table 29 refers to the comparison between EWA and EA alone (non-adjusted p -values), showing again that the former beats the last one in 19 of the runs. The corresponding one-sided p -value for the exact Binomial test in favour of EWA is $0.5^{19} = 1.90735 \times 10^{-6}^{***}$.

However, the classifier that performs the best in general is not one of the ensembles but BC₃. See Table 30 in Appendix C where the comparison of BC₃ with two different groups of classifiers is shown: the base classifiers on the one hand, and the ensembles on the other. It is

Table 10

Average over the runs of the averages (\bar{x}) and the standard deviations (s) over the folds, for the metrics AUPR, F-score and AUC with the different classifiers. Output variable *Cause*. In boldface, the top five for each metric.

Cause	AUPR		F-score		AUC	
	\bar{x}	s	\bar{x}	s	\bar{x}	s
BC ₁	0.28843 (2)	0.13782	0.35156	0.13504	0.66947	0.13908
BC ₂	0.26481 (5)	0.14900	0.40100	0.15465	0.68449 (3)	0.15798
BC ₄	0.23687	0.14077	0.41818	0.14623	0.64460	0.21162
BC ₅	0.19745	0.14208	0.49238 (1)	0.17201	0.53813	0.21570
NN	0.19639	0.14891			0.55821	0.10866
SVM	0.30422 (1)	0.12329			0.62733	0.11411
RF	0.18800	0.09173	0.42469	0.11681	0.52595	0.12109
MV	0.27028 (4)	0.15662	0.48086 (2)	0.16753	0.70198 (1)	0.17819
WMV	0.27764 (3)	0.16227	0.46805 (3)	0.15084	0.69832 (2)	0.17886
EA	0.26473	0.15841	0.43952 (5)	0.13424	0.68046 (4)	0.19690
EWA	0.26479	0.15690	0.44878 (4)	0.13594	0.67563 (5)	0.19324

observed that BC₃ surpasses them all.

F-score: With regard to the F-score metric, there is no significant differences considering all the classifiers at once (although for some runs RF shows to have a lower F-score than classifiers BC₃, BC₄ or the ensembles), nor considering the four ensembles together, but there are if we consider EWA and EA alone, showing that EWA is the best of both. The corresponding p -values are in Table 31, showing that in 7 runs, EWA outperforms EA; the one-sided p -value for the exact Binomial test in favor of EWA is $0.5^7 = 0.00781^{**}$. Regarding BC₃, there are no significant differences between this classifier and the ensembles, when we compare the five together (using adjusted p -values), although there are if we compare in pairs (BC₃ vs. each ensemble).

AUC: If we compare the twelve classifiers all at one, we see that the ensembles EWA and EA are significantly better than NN, SVM and RF, slightly better than BC₄ and BC₅, and no differences have been observed with BC₁, BC₂ and BC₃. For that, we decide to compare as a block these last with EWA and EA, since the adjustment of the p -values when making a large number of multiple comparisons can mask differences that are

really significant. There is only a slight evidence in favour of EWA, since it proves to have a significantly higher AUC median than BC₂ and EA in 3 of the runs, while none of the other classifiers beat it in any run, and if we compare in pairs (without adjusting the *p*-values), EWA outperforms each of EA, BC₁, BC₂ and BC₃ (see Table 32) so it turns out to be the favorite, although relative to BC₃, it is only slightly higher.

3.3. For the variable Cause

The boxplots for AUPR, F-score and AUC for first run and the twelve classifiers, for prediction of the output variable *Cause* with *positive* class “Complications”, are in Fig. 8 (Appendix A). The average over the 20 runs of the averages and the standard deviations over the folds, for the output variable *Cause*, are recorded in Table 10 below.

There are few significant differences between the classifiers for any of the metrics (AUPR, F-score and AUC), so the results do not seem conclusive, and we will have to wait for more data. The problem with the output variable *Cause* is that, due to the scarcity of cases in which the patient died in ICU (367 patients in our cohort), there are a large number of zeros in the confusion matrices, and consequently, a large number of missing values in the metrics. For example, if we compare AUPR for the classifiers SVM, EWA and EA with the pairwise Wilcoxon signed-rank test and making Holm-Bonferroni adjustments for multiple comparisons, we cannot see significant differences. We have to make comparisons in pairs to find something. Specifically, in 4 of the runs the median of SVM is statistically significantly greater than that of EWA (one-sided *p*-value for the exact Binomial test: $0.5^4 = 0.0625$), while this number increases to 6 for the comparison with EA instead of EWA (*p*-value $0.5^6 = 0.015625$).

4. Centrality, odds ratio and feature strength

Centrality and betweenness: Directed Acyclic Graphs (DAGs) from Figs. 9 and 10 represent the relationships of conditional independence entailed by the Bayesian networks BC₂, BC₃, BC₄ and BC₅, which are helpful to interpret the EWA ensemble, defined as an ensemble of them, jointly with BC₁, which is a Naive Bayes. We can establish which features play the main role in the model by using centrality and/or betweenness measures borrowed from the Network Analysis area applied to the DAGs, as it was done in [47]. In Graph Theory and Network Analysis, indicators of centrality identify the most important (influential) nodes within a graph, where “importance” is conceived as involvement in the cohesiveness of the network. For each feature we compute four different of these indicators (see [48]), which are shown in Tables 35 and 36 (Appendix E), normalized in order to sum up 100:

- (a) *Freeman’s degree of centrality*, which counts paths which pass through each node, that is, directed arcs which arrive at or depart from it.
- (b) *Basic standard betweenness measure*, which quantifies the number of times a node acts as a “bridge” along the shortest path between two other nodes (which we will call “geodesic” from now on). Nodes that have a high probability to occur on a randomly chosen geodesic between two randomly chosen nodes, have a high betweenness. Fixed a node *v*, this measure is defined by $\sum_{i,j,i \neq j, i \neq v, j \neq v} (g_{ivj} / g_{ij})$ (with the convention $0/0 = 0$), where g_{ij} is the number of geodesics from *i* to *j* in the graph, and g_{ivj} is the number of geodesics in the subset of those that pass through *v*.
- (c) *Borgatti’s proximal source betweenness* is a variant of basic standard betweenness to accumulate only for the last intermediating vertex in each incoming geodesic; this expresses the notion that,

Table 11
The most influential features attending to centrality and betweenness.

Demographic characteristics	F ₂ : Age
Main cause of admission	F ₄ : ACS F ₅ : RF F ₆ : Shock
Admission	F ₁₈ : Generic Syndrome F ₁₉ : Sepsis
Severity (on first 24 h of admission)	F ₂₀ : ICU Workload F ₂₁ : APACHE II

Table 12
Example of characteristics of a critically ill patient.

F ₃ : Charlson	F ₁₇ : Origin	F ₁₈ : Generic syndr.	F ₁₉ : Sepsis	F ₂₀ : ICU Workload	F ₂₁ : APACHE II
2	Emergency Room	Medical	Yes	M. unstable coma/shock	5–9

by serving as the “proximal source” for the target, this particular intermediary node will in some settings have greater influence than the rest. Fixed a node *v*, this measure is defined by $\sum_{i,j,i \neq j, j \neq v, i \rightarrow v} (g_{ivj} / g_{ij})$

- (d) *Borgatti’s proximal target betweenness* is the counterpart to proximal source betweenness that allows betweenness to accumulate only for the first intermediating vertex in each outgoing geodesic; this expresses the notion that, by serving as the “proximal target” for the source, this particular intermediary node will in some settings have greater influence or control than others. Fixed a node *v*, this measure is defined by $\sum_{i,j,i \neq j, i \neq v, j \rightarrow v} (g_{ivj} / g_{ij})$

Features F₉ (Hepatic F), F₁₃ (CT), F₁₄ (OT) and F₁₅ (Intoxication), all of them corresponding to the category of “Main cause of admission”, do not appear in Tables 35 and 36 because their value of the three betweenness variants is 0 for BC₂, BC₃, BC₄ and BC₅, and at the same time, their Freeman’s centrality value is very small. They are, therefore, the least important for the cohesiveness of the network, which is logical since none of them exceeds 1.5% of prevalence in the cohort. At the other extreme, there are the most important features in this regard, with higher values of centrality and betweenness (see Table 11): they are the most influential for the predictive model but in relation with others. Note that those that are in the “Main cause of admission” category, are the three most prevalent.

Features in Table 11 act as gateways, and the arcs that connect them as bridges, through which information flows from one cluster of variables in the model to another. We can see from the DAG of BC₃ in Fig. 9 (we consider this DAG because it is built without forcing any directed arcs, see Table 3) that

–The link between F₄ and F₁₈ is a bridge, and while F₄ is a gateway to F₂₁ and to the demographic characteristics, F₁₈ is to F₂₀, to the main features of the “Main cause of admission” category, and to that of “Admission”.

–The information between the two clusters mentioned in the above item, “Demographic Characteristics” and “Comorbidities” on one hand, and “Admission” on the other, also flows through the concatenation of two bridges: between F₂₀ and F₂₁, the features of “Severity

Table 13

Example of **Table 12**: probabilities of “die” and odds ratio in favour of “die”, for each of the possible “Main cause of admission”. In boldface those probabilities >0.5, which carry a prediction of “die” for the patient.

F ₁ : Sex	Male			Female			
	F ₂ : Age	75–84	>84	OR	75–84	>84	OR
F ₄		0.07878	0.09715	1.25837	0.08084	0.10693	1.36138
F ₅		0.19627	0.30552	1.80145	0.22859	0.36471	1.93737
F ₆		0.20421	0.31516	1.79328	0.23714	0.37257	1.91022
F ₇		0.20070	0.31982	1.87257	0.23325	0.37825	1.99987
F ₈		0.46216	0.55110	1.42871	0.50267	0.57898	1.36059
F ₉		0.16010	0.16010	1.00000	0.16010	0.16010	1.00000
F ₁₀		0.49500	0.66774	2.05030	0.53169	0.72091	2.27516
F ₁₁		0.20996	0.26928	1.38660	0.23044	0.31191	1.51378
F ₁₂		0.07956	0.13082	1.74141	0.09336	0.16242	1.88317
F ₁₃		0.30486	0.38341	1.41787	0.35674	0.44058	1.42009
F ₁₄		0.07100	0.07003	0.98518	0.07433	0.07305	0.98137
F ₁₅		0.10251	0.20942	2.31921	0.11517	0.22107	2.18048
F ₁₆		0.08263	0.10638	1.32156	0.08962	0.12278	1.42184

(on first 24 h of admission)”, and between F₂₁ and F₂. The latter is very natural since the APACHE II score is calculated based on age.

–Within the cluster of features of “Admission”, there is a bridge between F₅ and F₆, connecting sub-clusters; F₅ is a gateway to F₁₇: Origin and F₁₉: Sepsis, while F₆ is to F₁₀: CRA and F₁₆: Other syndromes.

Odds Ratio: Besides, we can use EWA, which has proven to be the best of those we have considered, to evaluate the effect of the features in the evaluation of the risk of death. For example, for each of the “Main cause of admission” we can compute the Odds Ratio (OR) in favour of “die” when the feature is present compared to when it is not, being the other absents. An odds ratio (OR) is a measure of association between a feature and the outcome (variable *Result*, in this case), which represents the odds in favour of “die” given a particular value of a feature, compared to the odds in favour of “die” occurring given another value. For that, we fix the other features.

Just as an example of this, consider a critically ill patient with the characteristics in **Table 12**, in the year 2018.

In **Table 13** we record the odds ratio, disaggregated by sex, in favour of the event “die”, for a critically patient whose characteristics are given in **Table 12**, according to what of the “Main cause of admission” has been reported for the patient (from F₄ to F₁₆). The odds ratio is defined as the ratio of the odds of event “die” occurring in the group of age >84 to the odds of it occurring in the group of age 75–84. Let continue with the example of the patient whose characteristics are given in **Table 12**: a male between 75 and 84 years old and with renal failure (F₈ = 1). Thus, the risk of death (probability of “die”) is 0.46216. This probability increases up to 0.55110 if the age increases to be > 84. Therefore, the Odds Ratio in favour of “die” is:

$$OR_{>84/75-84} = \frac{0.55110/(1 - 0.55110)}{0.46216/(1 - 0.46216)} = 1.42871$$

With respect to the risk of death, we observe the following, which is consistent with what is observed in **Figs. 2 and 3** in **Appendix A**:

–it is greater for women than for men, for both intervals of age 75–84 and >84 and for all of the “Main cause of admission” features except for F₉, in which case no variation in risk is observed (F₉, Hepatic Failure, is one of the features less important from the perspective of centrality, we have seen).

Table 14

SD, the correction term δ and CSD for the 21 features.

Feature	SD	δ	CSD(= SD \times δ)
F ₁	0.03432	1/2	0.01716
F ₂	0.09623	1/2	0.04812
F ₃	0.14104	1/2	0.07052
F ₄	0.17367	1/2	0.08684
F ₅	0.03301	1/2	0.01651
F ₆	0.08698	1/2	0.04349
F ₇	0.02511	1/2	0.01256
F ₈	0.00865	1/2	0.00433
F ₉	0.13364	1/2	0.06682
F ₁₀	0.49694	1	0.49694
F ₁₁	0.15203	1/2	0.07602
F ₁₂	0.09849	1/2	0.04925
F ₁₃	0.15729	1/2	0.07865
F ₁₄	0.12650	1/2	0.06325
F ₁₅	0.07601	1/2	0.03801
F ₁₆	0.12105	1/2	0.06053
F ₁₇	0.35811	1/2	0.17906
F ₁₈	0.17512	1/2	0.08756
F ₁₉	0.11333	1/2	0.05667
F ₂₀	0.43213	1/2	0.21607
F ₂₁	0.63546	1	0.63546

–within each sex, it is greater for older people, except when Other Trauma (F₁₄) is present, case in which the variation is very small (the same thing happens than with F₉).

We also see that for patients having the characteristics recorded in **Table 12** and having cardio respiratory arrest (F₁₀: CRA) or intoxication (F₁₅: Intoxication), both for men and women, the increase in age is an important risk factor (OR greater than 2 in **Table 13**).

On the other hand, we can study which of the “Main cause of admission” are risk factors for a male who is more than 85 years old, and with the features in **Table 12**, for example, and consider the question: “What is the Odds Ratio between F₁₀: CRA and F₅: RF in favor of *die*?”, which is answered by computing the ratio between the odds in favour of “die” when F₁₀ = 1 and when F₅ = 1, which is:

$$OR_{F_{10}/F_5} = \frac{0.66774/(1 - 0.66774)}{0.30552/(1 - 0.30552)} = 4.56836$$

(see **Table 13**) that is, the odds in favour of “die” if F₁₀ = 1 (Cardio Respiratory Arrest) is approximately 4.6 times greater than if F₅ = 1 (Respiratory Failure), for a patient with the mentioned characteristics, on which this result may depend, obviously.

Feature strength: Finally, we compute a measure of the feature strength to predict the output *Result*. For that, we follow [49] and introduce a measure based on the conditional probability tables of *Result* with respect to each feature, obtained with EWA (see **Appendix F**). This measure uses the Kolmogorov-Smirnov statistical distance and a correction parameter. Indeed, we first introduce a strength measure for each feature, say F, named *Strength Distance* (SD), in this way:

$$SD(F) = \max_{a,b \in \mathcal{F}} d_{a,b}^F$$

where \mathcal{F} is the set of the possible outcomes of variable F, and $d_{a,b}^F$ denotes the Kolmogorov-Smirnov statistical distance between the *a posteriori* conditional probability distributions of *Result* given the evidence F = a, and given the evidence F = b. The values of SD have been recorded in **Table 14** below. To take into account if different instantiations of a feature produce different predictions for *Result*, we introduce the correction term $\delta(F) = \gamma(F)/2 \in (0, 1]$, where $\gamma(F)$ is the number of different predictions obtained from the classifier for *Result* given the

Table 15

Best classifier(s) for the output variables *Result* and *Destination* and performance metrics, according to our experimental evaluation. If the classifiers are not in boldface, it means that only slightly exceeds its competitors. In each scenario, it is indicated by Yes/No if LR.APACHEII is significantly better than APACHEII, when the comparison makes sense.

		Performance metric		
		AUPR	F-score	AUC
Output	Result	EWA Yes	BC ₁ ,EWA,BC ₂	EWA Yes
	Destination	BC₃	BC₃	EWA

evidences of the form $F = a$, with a varying in \mathcal{F} . Then, $\delta(F)$ is the proportion of different predictions actually obtained by the classifier for *Result* among the possible we could obtain from an evidence on F , which is 2, and we use it to correct strength measure SD by introducing the *Corrected Strength Distance* (CSD) by $CSD(F) = SD(F) \times \delta(F)$. Note that $CSD(F) \geq 0$, and that $CSD(F) = 0$ if and only if F and *Result* are independent variables. In Table 14 we have recorded for each feature the correction term δ and the feature strength measure CSD as well.

Attending to CSD as feature strength measure, we can rank the features as follows, from stronger to weaker, attending to their capacity to modify prediction of the output *Result*:

- $F_{21}, F_{10}, F_{20}, F_{17}, F_{18}, F_4, F_{13}, F_{11}, F_3, F_9,$
- $F_{14}, F_{16}, F_{19}, F_{12}, F_2, F_6, F_{15}, F_1, F_5, F_7, F_8.$

5. Conclusions

It is unlikely that intelligent software will replace the clinician in medical diagnosis and prognosis for patients care. *Machine learning* expert systems are more likely to act as intelligent agents for specialized, complicated problems, and are intended to enhance the performance of the expert physician, given place to smart Intensive Care Units in the future. Our primary goal in this work has been to demonstrate the feasibility and benefits of routinely collecting information from critically ill patients admitted to ICU. The development of automatic tools to assist in clinical decision-making remains a challenge, although steps have already been taken in this direction. Our research is directed towards the construction of a *machine learning* hierarchical classifier to predict the risk of death in the ICU, as well as destination, for those who survive their stay in the ICU, or the cause of death for the rest, from multiple data streams (“Demographic Characteristics”, “Comorbidities”, “Admission” and “Severity (on first 24 h of admission)”).

In a first step, an ensemble of Bayesian classifiers (EWA) is developed to predict the risk of death (output variable *Result*), while another is used to predict the destination of the patient at ICU discharge if the predicted value for the variable *Result* is *live*, or his/her cause of death if the prediction is *die*. EWA is constructed as an ensemble of five different base Bayesian networks, with the weighted average rule with appropriate weights, which are obtained from the estimations of the AUPR values of the base classifiers to give more power to more “competent” base classifiers in the average criterion. When dealing with highly unbalanced and sparse datasets, AUPR, F-score and AUC are preferred as representation of performance assessment in the binary classification to the most commonly used measure, the *Accuracy*.

We compare the performance of EWA with that of the base Bayesian classifiers from which it has been constructed, and with some state-of-the-art *machine learning* methodologies (Neural Networks, Support Vector Machine and Random Forest), as well as with the ensembles of the same

Table 16

Top five features, with the category that maximizes risk of death for each, and the associated risk.

Feature	Category that maximizes risk	Risk
F_{21} : APACHE II	>34	64%
F_{10} : CRA (Cardio Respiratory Arrest)	Yes	62%
F_{20} : ICU Workload	Medical unstable with coma or shock	44%
F_{17} : Origin	Extra Hospital Emergency	46%
F_{18} : Generic Syndrome	Medical	19.5%

base classifiers obtained using the average rule without weights, the majority vote and the weighted majority vote criteria, finding that EWA has best overall performance. We also see as expected, that EWA improves the models based on scales, such as the traditional approach based on the APACHE II score, which suffers from not incorporating elements that clinical practice reveals to be of great value, such as the origin of the patient (collected in our model in variable F_{17} , that has been ranked fourth in importance, according to CSD as measure of the feature strength), since depending on the origin, the patient may have a very different evolution in the ICU, presenting fragility to a greater or lesser degree. EWA even outperforms a local recalibration of this model obtained from the dataset, LR.APACHEII, both with the AUPR and AUC metrics, and between them, LR.APACHEII behaves better from a predictive point of view than APACHEII. When we consider the F-score metric, what happens is that, on the one hand, all the classifiers show better than LR.APACHEII while, on the other, it is not possible to calculate the F-score for the standard based on the APACHE II (*accuracy paradox*).

Table 15 shows the best classifier for each output variable and performance metric. Note that *Cause* does not appear since for this output variable, there are no significant differences among the classifiers.

The conclusion from the statistical point of view of LR.APACHEII’s superiority compared to APACHEII is that the logistic regression model based on the score APACHE II is a better predictor when estimates the parameters from the current database, which seems quite logical, since in this way the model better reflects the characteristics of the patients for whom the mortality risk prediction is intended and, in particular, it catches the changes and improvements in medical practice. Indeed, APACHE II score was described and validated by means of a logistic regression model based on the management and results of critically ill patients in 1985, and since then, some obvious advances, in preventive and primary medicine or in the control of chronic diseases, have changed the relevance of age and comorbidities in the prognosis of critically ill patients, and in our century there has been a dramatic reduction in mortality due to sepsis, coronary syndromes and trauma. Therefore, healthcare professionals must take this into account when faced with the need to use scoring systems in their daily practice.

We also delve into interpretability of the EWA ensemble, both from the DAGs of the base classifier from which EWA has been constructed, using centrality and betweenness measures, and from the conditional probability tables of the outcome *Result* conditioned to any of the features, which allow us to rank the features attending to a measure of their strength. While this last approach discovers which features are important for prediction, features that are important but in relation with others, that is, the most “influential”, stand out using centrality and betweenness, the rest of features being irrelevant for prognosis purpose.

The top five features are, in order of strength, in Table 16 below, with the categories that maximize the mortality risk, obtained from tables in Appendix F. Of these, F_{18} , F_{20} and F_{21} are also influential from the point of view of centrality and betweenness, so they appear as clearly

highlighted as basic characteristics to take into account to predict the risk of death in patients admitted to a hospital ICU. Note that a feature can be influential from the point of view of centrality and betweenness but have little predictive importance; for example, F_2 : Sex has a relatively low value of CSD (meaning that the risk of death is similar for men and women, but it is influential since acts as connecting node between the features of the “Demographic Characteristics” and “Comorbidities” categories, which also have weak importance, and the rest of features).

Top five features are followed by F_4 : ACS (Acute Coronary Syndrome), which is the only one characteristic of the “Main cause of admission” category that acts as factor of protection (its presence reduces the risk of death). From Table 38 we compute the OR in favour of “die” corresponding to the presence of ACS ($F_4 = 1$) with respect to its absence ($F_4 = 0$), regardless of the other features:

$$OR_{F_4=1/F_4=0} = \frac{0.00390/(1 - 0.00390)}{0.11757/(1 - 0.11757)} = 0.02939$$

That is, the odds in favour of “die” divides by approximately 34 when ACS is present with respect to when it is absent, in general, without taking into account the other characteristics of the patient. This fact may seem counterintuitive, but it must be taken into account that clinical practice indicates that among the patients admitted to the ICU of a hospital, those who do so with “Generic Syndrome” $F_{18} = \text{“Coronary”}$, which is clearly associated with ACS ($F_4 = \text{“yes”}$), are the ones with the best prognosis. Keep in mind, that if it is not due to that reason, the admission will be for another with a worse prognosis. For example, if they present a respiratory failure RF ($F_5 = \text{“yes”}$), which is associated with “Generic Syndrome” $F_{18} = \text{“Medical”}$, their prognosis worsens (increases the risk of death), what is consistent with the information in Table 16. Note that 95.2% of the patients with $F_{18} = \text{“Coronary”}$ present ACS, while only 0.7% of them present RF; from the patients with $F_{18} = \text{“Medical”}$, 47.8% present RF but only 2.8% ACS.

Among the top five features ranked by strength, there is only one of the “Main cause of admission” category which seems to be the most important risk factor, F_{10} : CRA. From Table 40 we obtain similarly that the OR in favour of “die” corresponding to the presence of CRA ($F_{10} = 1$) with respect to its absence ($F_{10} = 0$) is

$$OR_{F_{10}=1/F_{10}=0} = \frac{0.62298/(1 - 0.62298)}{0.12604/(1 - 0.12604)} = 11.45758$$

That is, the odds in favour of “die” multiplies by approx. 11.5 when CRA is present with respect to when it is absent. For a specific patient, based on its known characteristics, these result can be adjusted, as we have done in Section 4 in some cases to evaluate the effect of the features in the evaluation of the risk of death by means of the OR.

It would be very interesting to extrapolate our model to a database with a case mix of different ICUs, which would make it possible to compare the performance of different units; to do this, the characteristics of a typical patient would be introduced into the model and the risk of death would be predicted with each ICU. This tool would also make it possible to carry out a longitudinal study and analyze the improvement over time of the healthcare processes of a specific ICU, as well as adapting the model to the different types of ICU, from the trauma center to the thematic respiratory or cardiovascular ICU, if we learn it from data collected in these more specific scenarios.

To the extent that it can help physicians in undertaking patient-tailored therapeutic decisions, and to the health authorities to manage more optimally the available resources, the local data-driven *machine learning* methodology introduced in this work for estimating the risk of death and predicting the destination at ICU discharge or the cause of death, using an ensemble of Bayesian classifiers, seems to be a useful and promising tool with important clinical applicability.

Funding

R. Delgado and J.D. Núñez-González are supported by Ministerio de Ciencia, Innovación y Universidades, Gobierno de España, project ref. PGC2018-097848-B-I0.

R. Delgado, J.D. Núñez-González, Juan Carlos Yébenes and Angel Lavado are partially supported by TV3 Fundació Marató (Sepsis Training, Audit and Feedback (STAF) Project; Codi Projecte 201836).

Conflict of interest

The authors declare no conflict of interest.

Acknowledgments

We want to acknowledge received contributions from reviewers during the review process and, especially Thanks to Editor-in-Chief Carlo Combi for the chance of having the article in consideration.

Appendix A. Plots

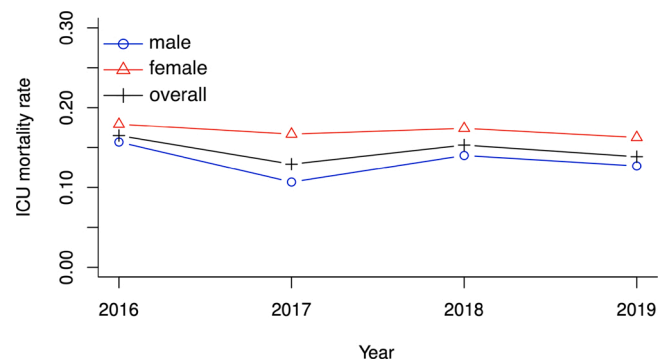


Fig. 2. Evolution of mortality rate at the ICU with year, disaggregated by sex, and for the overall population.

Appendix B. Tables for output Result

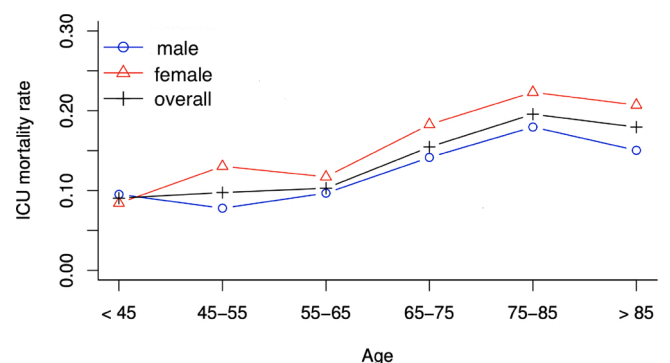


Fig. 3. Evolution of mortality rate at the ICU with age, disaggregated by sex, and for the overall population.

Appendix C. Tables for output *Destination*

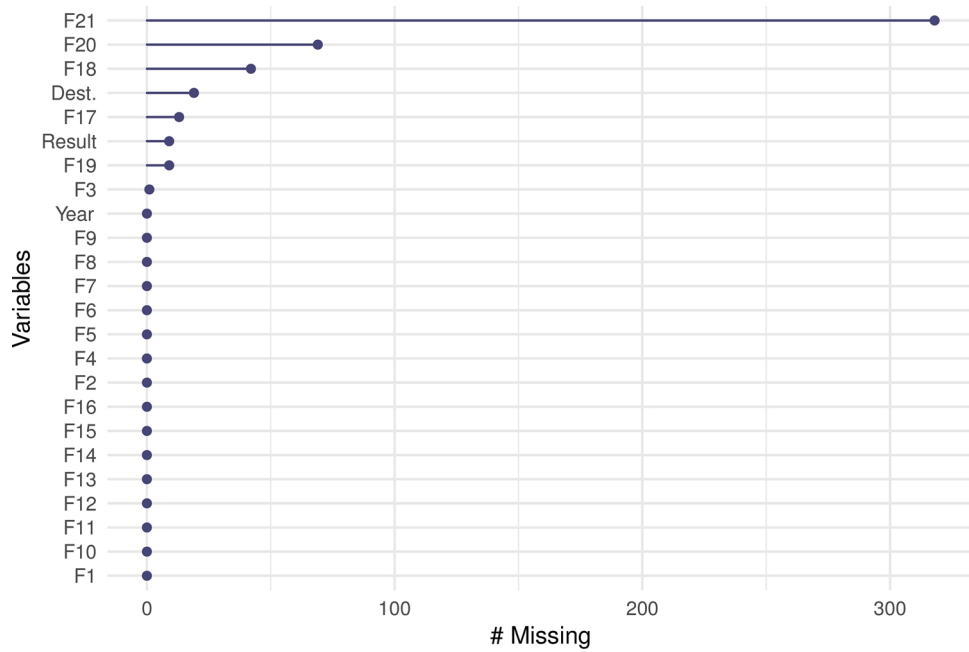


Fig. 4. Variables ordered by the number of missing values. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

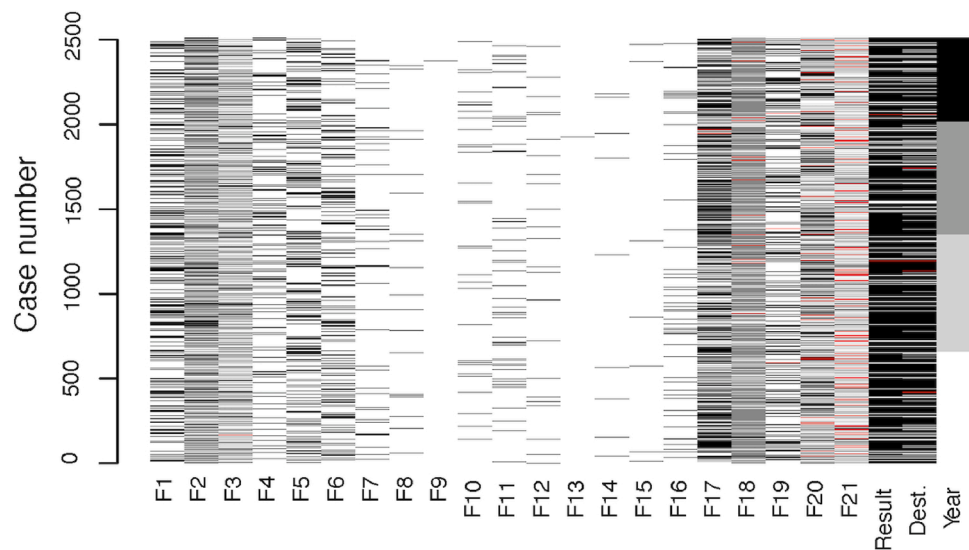


Fig. 5. Distribution of missing values (in red), where cases have been ordered by year. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

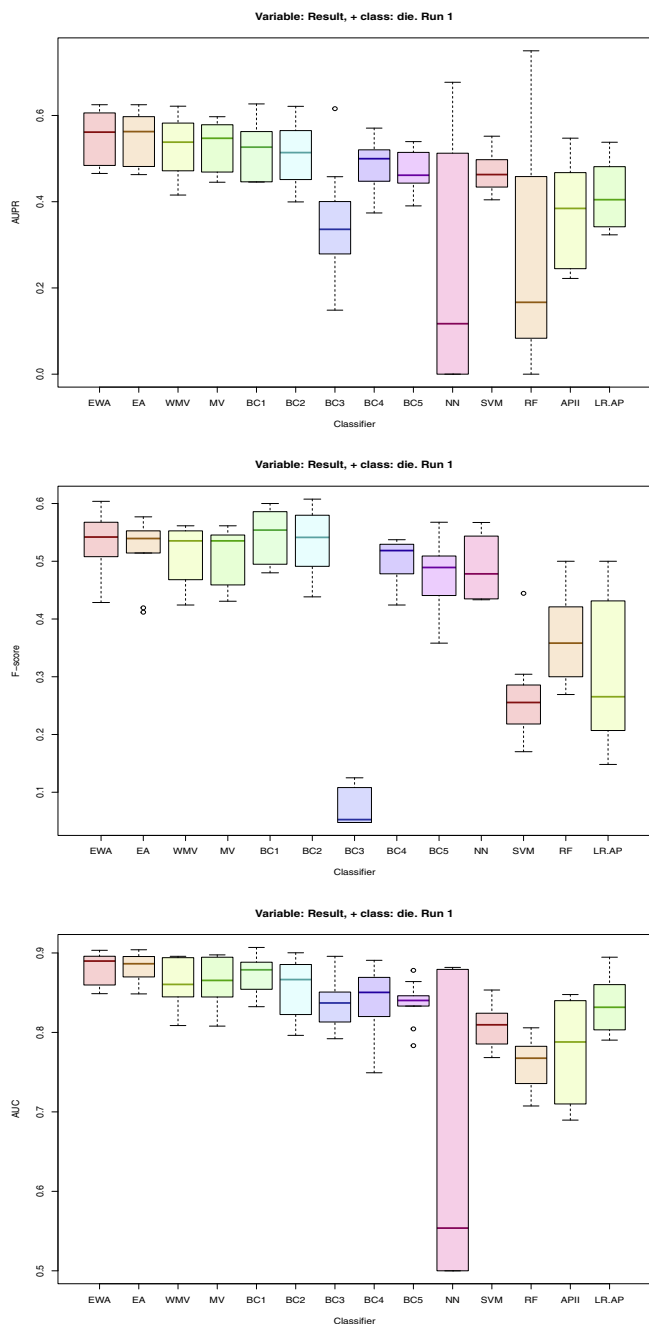


Fig. 6. Boxplots for output variable *Result* in the first run. AUPR, F-score and AUC. F-score cannot be computed for the standard approach based on APACHE II (“APII” as x-axis label), but it can for the locally recalibrated LR.APACHEII (“LR.AP” as x-axis label).

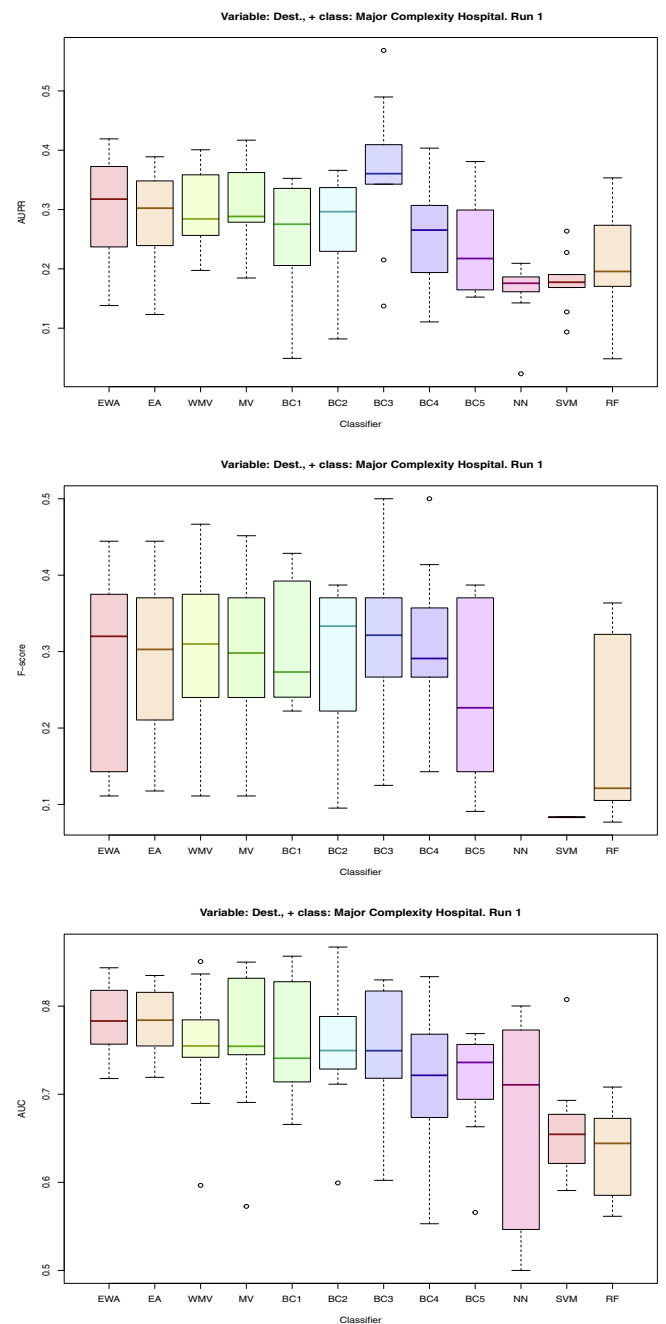


Fig. 7. Boxplots for output variable *Destination* in the first run, for AUPR, F-score and AUC.

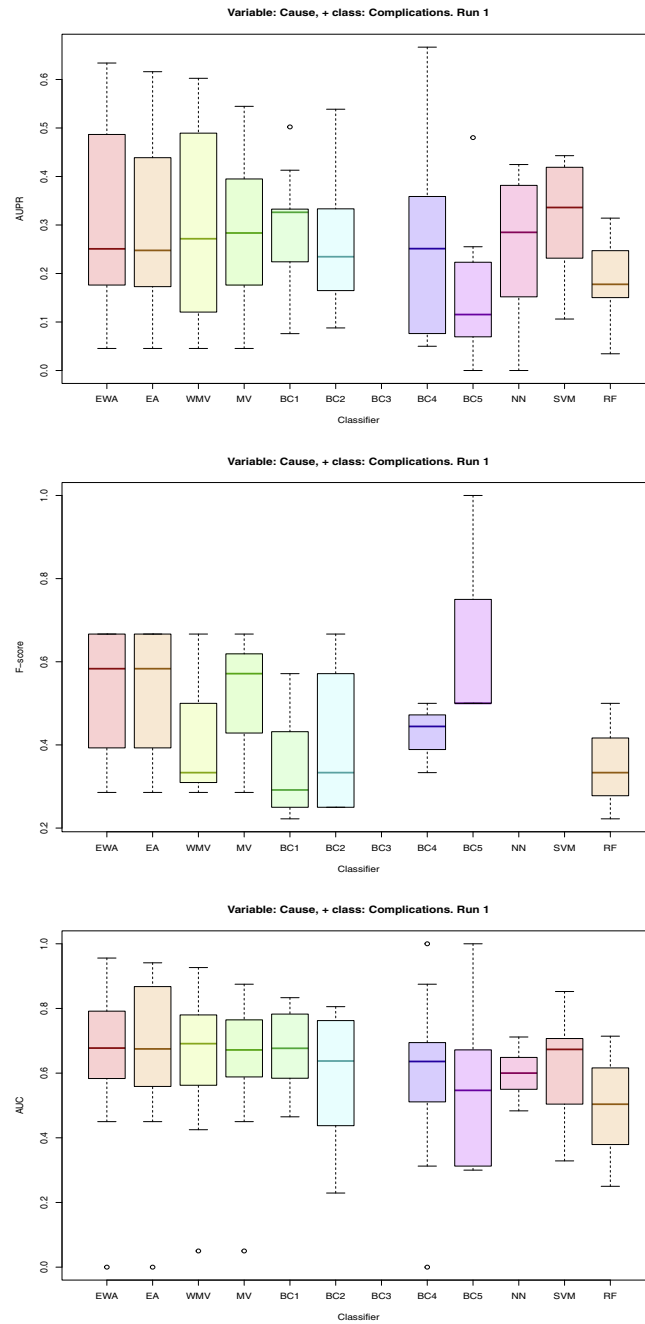


Fig. 8. Boxplots for output variable *Cause* in the first run, for AUPR, F-score and AUC.

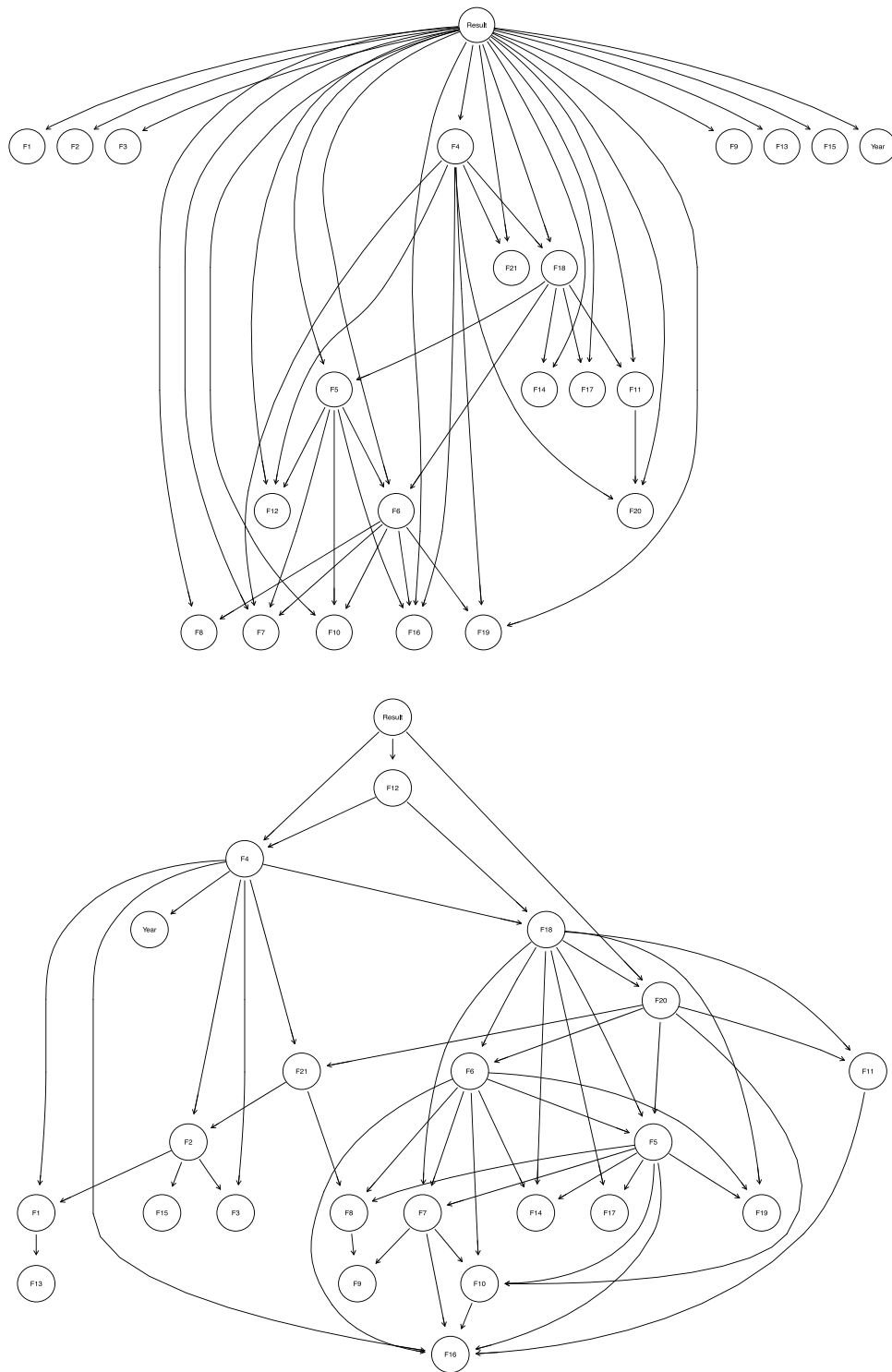


Fig. 9. DAGs for the base classifiers BC₂ (top) and BC₃ (bottom), learned from the whole database set.

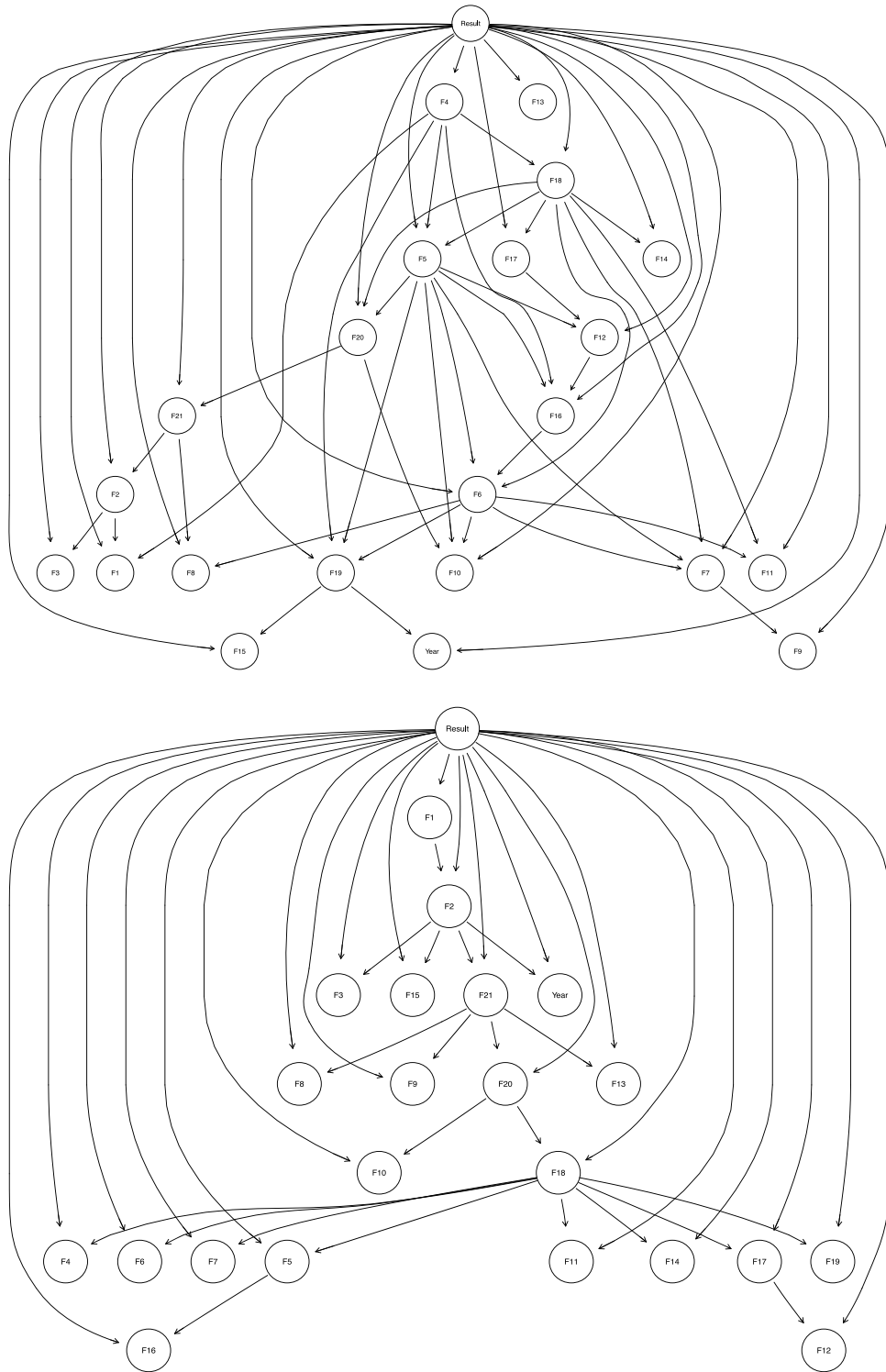


Fig. 10. DAGs for the base classifiers BC₄ (top) to BC₅ (bottom), learned from the whole database set.

Table 17

AUPR median for EWA/EA is significantly greater (p -value <0.1) than that of WMV and/or MV, for output variable *Result* and for each run? (“2” if it is greater for both, WMV and MV, “1” if it is greater for only one, “0” if it is not greater for either of them).

AUPR <i>Result</i>	Run																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
EWA	0	0	1	2	2	0	2	0	2	1	1	2	2	0	1	1	1	1	2	2
EA	0	0	0	2	2	0	2	0	2	1	1	2	1	0	0	1	1	0	2	1

Table 18

Adjusted p -values for the comparisons between the four ensembles, corresponding to the statistical significances in Table 17 when we compare EWA/EA against WMV and MV. Also, in boldface, the adjusted p -values corresponding to the comparison between the two EWA and EA, that were not reported there.

AUPR (Result)	EWA	EA	WMV	MV
		0.0059**	0.029*	0.0342*
		0.029 *	0.018*	0.098.
	(5)	0.0059**	0.024*	0.018*
		0.0342 *	0.0117*	0.018*
		0.049 *	(8) 0.039*	0.0098**
			0.029*	0.082.
			0.012*	0.012*
EWA>			0.055.	(15) 0.074.
				0.0059**
				0.029*
				0.093.
				0.059.
				0.012*
				0.018*
			0.049*	0.098.
			0.029*	0.027*
			0.056.	0.039*
			(8) 0.0645.	0.0146*
			0.024*	0.093.
EA>			0.049*	(10) 0.068.
			0.012*	0.012*
			0.018*	0.034*
				0.029*
				0.021*

Table 19

Non-adjusted p -values corresponding to the comparisons between EWA and EA in Table 18, but only between them two (so the p -values are not adjusted). In boldface the 5 runs corresponding to the adjusted p -values that have been reported in Table 18.

AUPR (Result)	Run	3	5	9	14	15	18	19	20
EWA>EA	p -Value	0.00098***	0.014*	0.00098***	0.042*	0.0068**	0.08.	0.024*	0.042*

Table 20

(Non-adjusted) p -values corresponding to the comparison between LR.APACHEII and the standard based on the APACHE II score, for the alternative hypothesis that the former has greater AUPR median.

AUPR	Run	2	3	4	5	7	8	9	10	11
	p -Value	0.042*	0.0049**	0.014*	0.032*	0.0068**	0.0098**	0.0049**	0.0098**	0.032*
(Result)	Run	12	13	14	15	16	17	18	19	20
	p -Value	0.042*	0.065.	0.053.	0.0068**	0.08.	0.0049**	0.024*	0.065.	0.014*

Table 21

F-score median for EWA/EA is significantly greater (p -value <0.1) than that of WMV and/or MV, for output variable *Result* and for each run? (“2” if it is greater for both, WMV and MV, “1” if it is greater for only one, “0” if it is not greater for either of them).

F-score <i>Result</i>	Run																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
EWA	0	0	0	0	2	0	2	0	0	1	2	2	0	1	0	0	2	1	0	1
EA	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	2	0	0	0

Table 22

Adjusted p -values for the comparisons between the four ensembles, corresponding to the statistical significances in Table 21 when we compare EWA/EA against WMV and MV. Also, in boldface, the adjusted p -values corresponding to the comparison between the two EWA and EA, that were not reported there.

F-score (Result)	EWA	EA	WMV	MV
		(2)	0.054	0.027*
			0.0391 *	0.027*
				(5) 0.0977.
				0.066.
EWA>				0.0391* (9) 0.027*
				0.059.
				0.0059**
				0.082.
				0.018*
			(1)	0.0645.
				0.027*
				0.049*
EA>				(5) 0.0146*
				0.032*
				0.0146*
WMV>				(2) 0.0907
				0.090

Table 23

Non-adjusted *p*-values corresponding to the comparisons between EWA and EA in Table 22, but only between them two (so the *p*-values are not adjusted). In boldface the 2 runs corresponding to the adjusted *p*-values that have been reported in Table 22.

F-score (Result)	Run <i>p</i> -Value	1	2	3	5	8	9	10
		0.062	0.078	0.026*	0.018*	0.029*	0.071	0.038*
EWA>EA	Run <i>p</i> -Value	11	13	14	17	18	19	20
		0.062	0.054	0.022*	0.012*	0.029*	0.030*	0.029*

Table 24

AUC median for EWA/EA is significantly greater (*p*-value<0.1) than that of WMV and/or MV, for output variable *Result* and for each run? (“2” if it is greater for both, WMV and MV, “1” if it is greater for only one, “0” if it is not greater for either of them).

AUC	Run																				
<i>Result</i>		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
EWA		2	0	2	2	2	0	2	2	2	2	0	2	2	0	2	2	0	0	2	2
EA		2	0	2	2	2	0	2	2	2	2	0	2	2	0	2	2	0	0	2	2

Table 25

Non-adjusted *p*-values corresponding to the comparisons between EWA and EA when comparing them alone (non-adjusted *p*-values), for the AUC metric and the output variable *Result*.

AUC (Result)	Run <i>p</i> -Value	1	2	3	4	5	9	10	14	15	18	19	20
EWA>EA		0.08	0.024*	0.0029**	0.096	0.002**	0.0068**	0.0098**	0.08	0.012*	0.065	0.02*	0.065

Table 26

Adjusted *p*-values for the comparisons between the four ensembles, corresponding to the statistical significances in Table 24 when we compare EWA/EA against WMV and MV. Also, in boldface, the adjusted *p*-values corresponding to the comparison between the two EWA and EA, that were not reported there, indicating to which run they correspond.

AUC (Result)	EWA	EA	WMV	MV
			0.018 * (run 3)	0.029*
			0.012 * (run 5)	0.041*
			0.034 * (run 9)	0.018*
		(6)	0.059 (run 10)	0.049*
			0.039 * (run 15)	0.012*
			0.012 * (run 19)	0.074
EWA>			(14)	0.029*
				0.074
				(14)
				0.029*
				0.059
				0.029*
				0.012*
				0.039*
				0.018*
				0.012*
				0.015*
				0.082
				0.082
				0.039*
				0.049*
				0.023*
				0.020*
				0.049*
				0.049*
EA>				0.012*
				0.029*
				0.097
			(14)	0.034*
				(14)
				0.074
				0.084
				0.029*
				0.029*
				0.012*
				0.039*
				0.029*
				0.012*
				0.041*
				0.015*
				0.082
				0.082

Table 27

(Non-adjusted) *p*-values corresponding to the comparison between LR.APACHEII and APACHEII, for the alternative hypothesis that the former has greater AUPR median.

AUPR	Run	2	3	4	5	7	8	9	10	11
	<i>p</i> -Value	0.042*	0.0049**	0.014*	0.032*	0.0068**	0.0098**	0.0049**	0.0098**	0.032*
(Result)	Run	12	13	14	15	16	17	18	19	20
	<i>p</i> -Value	0.042*	0.065	0.053	0.0068**	0.08	0.0049**	0.024*	0.065	0.014*

Table 28

Adjusted *p*-values for the comparisons between the four ensembles. Only significant *p*-values (<0.1) have been recorded.

AUPR (Destination)	EA
EWA>	(10)
	0.041*
	0.0059**
	0.018*
	0.012*
	0.012*
	0.018*
	0.012*
	0.082
	0.082
	0.082
WMV>	(2)
	0.082
	0.059

Table 32

Number of runs for which there are statistically significant differences between EWA and each of the classifiers EA, BC₁, BC₂ and BC₃, for the output variable *Destination* and the metric AUC. These differences are always in the sense that the median of EWA is greater than that of the others. The one-sided *p*-values for the exact Binomial test for the statistical significance of the number of runs is also given in the second column.

AUC (Destination)	Number of runs	<i>p</i> -Value
EWA>EA	15	$0.5^{15} = 3.05 \times 10^{-5***}$
EWA>BC ₁	7	$0.5^7 = 0.0078**$
EWA>BC ₂	9	$0.5^9 = 0.0020**$
EWA>BC ₃	4	$0.5^4 = 0.0625$

Table 29

Non-adjusted *p*-values corresponding to the comparisons between EWA and EA in Table 28, for output variable *Destination* and AUPR. In boldface the 10 runs for which there is statistical significance when considering adjusted *p*-values for comparison of the four ensembles in Table 28, for comparison between EWA and EA.

AUPR (Destination)	Run	1	2	3	4	5	6	7	8	9	10
	<i>p</i> -Value	0.019*	0.0068**	0.042*	0.00098***	0.0029**	0.002**	0.002**	0.0029**	0.053	0.024*
EWA>EA	Run	11	12	13	14	15	16	17	18	20	
	<i>p</i> -Value	0.002*	0.014*	0.042*	0.019*	0.014*	0.032*	0.019*	0.042*	0.014*	

Table 30

Number of runs, say *n*, for which AUPR of BC₃ model (for the output variable *Destination*) is statistically greater than that of the other classifiers, when considering adjusted *p*-values for comparison of five classifiers at the same time (BC₁ to BC₅ first, and secondly BC₃ and the ensembles). Below appear the corresponding *p*-values for the exact Binomial test in favour of BC₃, which are 0.5^n .

AUPR (Destination)	BC ₁	BC ₂	BC ₄	BC ₅	EWA	EA	WMV	MV
	9	10	16	20	6	10	8	7
BC ₃ >	0.002**	0.0009***	$1.53 \times 10^{-5***}$	$9.54 \times 10^{-7***}$	0.016*	0.0009***	0.004**	0.008**

Table 31

(Non-adjusted) *p*-values corresponding to the comparison between EWA and EA for output variable *Destination* and F-score.

F-score (Destination)	Run	3	8	10	11	13	14	15
EWA>EA	<i>p</i> -Value	0.03*	0.05	0.09	0.05	0.05	0.02*	0.08

Appendix D

D.1 Coefficients (weights) β for Eq. (2)

Table 33

Sepsis means $F_{19} = 1$. Non surgical category means $F_{18} =$ Coronary, Medical or Trauma, while surgical category means $F_{18} =$ Elective or Urgent Surgical. Blanc spaces mean excluding category. This table has been adapted to our setting from [9].

Features	Sepsis or non surgical category	No sepsis and surgical category
F ₄	-0.191	-0.797
F ₅	-0.890	-0.610
F ₆	0.493	-0.797
F ₇	-0.759	-1.150
F ₈	-0.885	-0.196
F ₉	0.501	-0.613
F ₁₀	0.393	0.393
F ₁₁		-0.248
F ₁₂	-1.368	-0.797
F ₁₃	-0.517	-0.955
F ₁₄	-1.228	-1.684
F ₁₅	-0.142	-0.196
F ₁₉	0.113	

D.2 Coefficients (weights) α for Eq. (3)

Table 34

Coefficients α for Eq. (3) (only those with significant p -values, that is >0.10 , have been recorded). (a): the odds in favour of “die” where the regressors are at their reference value (all equal to “0”, including APACHE II). (b): increase in odds in favour of “die” for a one-unit increase in APACHE II score, holding the other regressors at a fixed value. (c): increase in odds in favour of “die” for the regressor taken the value “1”, with respect to value “0”, holding the other regressors at a fixed value. (d): decrease in odds in favour of “die” for F₄ taken the value “1”, with respect to value “0”, holding the other regressors at a fixed value.

Features	α estimated	p -Value	interpretation
Intercept	$\alpha_0 = -4.85711$	$5.72 \times 10^{-14***}$	0.00777 ^(a)
APACHE II	$\alpha_1 = 0.11544$	$<2 \times 10^{-16***}$	12.2% ^(b)
F ₁₉	$\alpha_3 = 0.48604$	0.00212**	62.6% ^(c)
F ₄	$\alpha_4 = -1.83024$	0.03768*	84.1% ^(d)
F ₅	$\alpha_5 = 0.62866$	0.00126**	87.5% ^(c)
F ₆	$\alpha_6 = 0.52522$	0.00896**	69.1% ^(c)
F ₇	$\alpha_7 = 0.49130$	0.05651-	63.4% ^(c)
F ₁₀	$\alpha_{10} = 1.82376$	$4.38 \times 10^{-9***}$	519.5% ^(c)

Appendix E. Centrality and betweenness measures

Table 35

(Normalized to sum up 100) Freeman’s degree of centrality and Basic standard betweenness measure of the features. In boldface the highest 5 values of each column.

Feature	Freeman’s centrality (%)				Basic standard betw. (%)			
	BC ₂	BC ₃	BC ₄	BC ₅	BC ₂	BC ₃	BC ₄	BC ₅
F ₁	1.5	2.0	2.0	1.5	0.0	6.1	0.0	0.0
F ₂	1.5	3.0	4.5	10.0	0.0	13.0	7.0	0.0
F ₃	1.5	2.0	2.0	3.5	0.0	0.0	0.0	14.5
F ₄	12.0	9.5	6.5	3.5	0.0	15.6	0.0	0.0
F ₅	10.5	10.5	11.0	5.0	16.7	6.9	7.0	0.0
F ₆	12.0	9.5	10.0	3.5	38.9	4.8	23.3	3.6
F ₇	6.0	6.5	5.5	3.5	0.0	4.7	5.5	0.0
F ₈	3.0	4.0	3.5	3.5	0.0	3.1	0.0	0.0
F ₁₀	4.5	5.5	4.5	3.5	0.0	0.2	0.0	0.0
F ₁₁	4.5	3.0	3.5	3.5	5.6	0.4	0.0	0.0
F ₁₂	4.5	3.0	4.5	3.5	0.0	1.5	7.8	0.0
F ₁₆	6.0	6.5	5.5	3.5	0.0	0.0	14.6	0.0
F ₁₇	3.0	2.0	3.5	5.0	0.0	0.0	0.4	0.0
F ₁₈	10.5	10.5	10.0	16.5	38.9	19.6	5.2	3.6
F ₁₉	4.5	3.0	6.5	3.5	0.0	0.0	10.9	29.0
F ₂₀	4.5	7.5	5.5	5.0	0.0	12.6	8.9	0.0
F ₂₁	3.0	4.0	4.5	10.0	0.0	11.4	9.4	26.1

Table 36

(Normalized to sum up 100) Borgatti’s proximal source and proximal target betweenness measures of the features. In boldface the highest 5 values of each column.

Feature	Borgatti’s proximal source (%)				Borgatti’s proximal target (%)			
	BC ₂	BC ₃	BC ₄	BC ₅	BC ₂	BC ₃	BC ₄	BC ₅
F ₁	0.0	8.5	0.0	0.0	0.0	2.4	0.0	0.0
F ₂	0.0	14.6	12.5	0.0	0.0	6.1	2.8	0.0
F ₃	0.0	0.0	0.0	6.7	0.0	0.0	0.0	33.3
F ₄	0.0	14.6	0.0	0.0	0.0	21.8	0.0	0.0
F ₅	18.75	8.8	7.9	0.0	15.6	6.7	12.5	0.0
F ₆	43.75	6.0	27.5	8.3	34.4	4.2	17.8	1.7
F ₇	0.0	6.6	9.7	0.0	0.0	3.0	4.2	0.0
F ₈	0.0	4.4	0.0	0.0	0.0	2.4	0.0	0.0
F ₁₀	0.0	0.3	0.0	0.0	0.0	0.3	0.0	0.0
F ₁₁	6.25	0.6	0.0	0.0	6.25	0.6	0.0	0.0
F ₁₂	0.0	0.6	1.4	0.0	0.0	2.1	13.9	0.0
F ₁₆	0.0	0.0	3.2	0.0	0.0	0.0	13.4	0.0
F ₁₇	0.0	0.0	0.7	0.0	0.0	0.0	0.7	0.0
F ₁₈	31.25	22.2	6.0	8.3	43.75	24.6	9.3	1.7
F ₁₉	0.0	0.0	19.4	53.3	0.0	0.0	8.3	16.7
F ₂₀	0.0	7.6	4.6	0.0	0.0	17.0	11.6	0.0
F ₂₁	0.0	5.1	6.9	10.0	0.0	8.6	5.6	20.0

Appendix F. CPT tables for the output *Result*

Each table is obtained computing the *a posteriori* conditional probability of the output *Result* to the feature, assuming the rest of features are not observed, except in the case of the “Main cause of admission”

category, for which when one is present, the others are necessarily absent, since they are mutually exclusive⁴; instead, if it is absent, we assume the others are not observed.

Table 37

CPT of variable *Result* conditioned to F_1 and to F_2 .

	F_1 : Sex		F_2 : Age					
	Male	Female	<45	45–54	55–64	65–74	75–84	>84
live	0.86575	0.83143	0.90517	0.89967	0.89313	0.84506	0.80894	0.82340
die	0.13425	0.16857	0.09483	0.10033	0.10687	0.15494	0.19106	0.17660

Table 38

CPT of variable *Result* conditioned to F_3 and to F_4 .

	F_3 : Charlson comorbidity index					F_4 : ACS	
	0	1	2	3	>3	0	1
live	0.90627	0.86975	0.84657	0.81614	0.76523	0.82243	0.99610
die	0.09373	0.13025	0.15343	0.18386	0.23477	0.17757	0.00390

Table 39

CPT of variable *Result* conditioned to F_5 , to F_6 , to F_7 and to F_8 .

	F_5 : RF		F_6 : Shock		F_7 : Coma		F_8 : Renal F	
	0	1	0	1	0	1	0	1
live	0.87220	0.83919	0.88342	0.79644	0.86008	0.83497	0.85822	0.84957
die	0.12780	0.16081	0.11658	0.20356	0.13992	0.16503	0.14178	0.15043

Table 40

CPT of variable *Result* conditioned to F_9 , to F_{10} , to F_{11} and to F_{12} .

	F_9 : Hepatic F		F_{10} : CRA		F_{11} : ES		F_{12} : Arrhythmia	
	0	1	0	1	0	1	0	1
live	0.85302	0.98666	0.87396	0.37702	0.84377	0.99580	0.85084	0.94933
die	0.14698	0.01334	0.12604	0.62298	0.15623	0.00420	0.14916	0.05067

Table 41

CPT of variable *Result* conditioned to F_{13} , to F_{14} , to F_{15} and to F_{16} .

	F_{13} : CT		F_{14} : OT		F_{15} : Intoxication		F_{16} : Other syndromes	
	0	1	0	1	0	1	0	1
live	0.85369	0.69640	0.85177	0.97827	0.85271	0.92872	0.84672	0.96777
die	0.14631	0.30360	0.14823	0.02173	0.14729	0.07128	0.15328	0.03223

Table 42

CPT of variable *Result* conditioned to F_{17} .

	F_{17} : Origin					
	Ward	Operation Room	Extra Hospital Emergency	Other Hospital	Emergency Room	unknown
live	0.77059	0.89578	0.53767	0.87632	0.88156	0.71747
die	0.22941	0.10422	0.46233	0.12368	0.11844	0.28253

⁴ Although it is possible for a patient to present more than one of the features of the “Main cause of admission” category, in practice, there were cases in which several were recorded, in principle only the most significant had to be reported and, in fact, this is so for almost 89% of patients.

Table 43
CPT of variable *Result* conditioned to F_{18} .

	F_{18} : Generic Syndrome					
	Elective surgical	Urgent surgical	Coronary	Medical	Trauma	unknown
live	0.94701	0.85312	0.97963	0.80451	0.93671	0.93971
die	0.05299	0.14688	0.02037	0.19549	0.06329	0.06029

Table 44
CPT of variable *Result* conditioned to F_{19} .

	F_{19} : Sepsis	
	No	Yes
live	0.89398	0.78065
die	0.10602	0.21935

Table 45
CPT of variable *Result* conditioned to F_{20} .

	F_{20} : ICU Workload					
	M. monitoring	M. unstable without coma/shock	M. unstable coma/shock	Post-surg. monitor.	Post-surg. unstable	Unknown
live	0.95564	0.86603	0.55982	0.99195	0.98728	0.85176
die	0.04436	0.13397	0.44018	0.00805	0.01272	0.14824

Table 46
CPT of variable *Result* conditioned to F_{21} .

	F_{21} : APACHE II (discretized)								
	<5	5–9	10–14	15–19	20–24	25–29	30–34	>34	Unknown
live	0.99579	0.96505	0.92690	0.86155	0.71854	0.59862	0.58245	0.36033	0.76492
die	0.00421	0.03495	0.07310	0.13845	0.28146	0.40138	0.41755	0.63967	0.23508

References

[1] Kerlin MP, Cooke CR. Understanding costs when seeking value in critical care. *Ann Am Thorac Soc* 2015;12(12):1743–4.

[2] Lone NI, Gillies MA, Haddow C, Dobbie R, Rowan KM, Wild SH, et al. Five-year mortality and hospital costs associated with surviving intensive care. *Am J Respir Crit Care Med* 2016;194(2):198–208.

[3] Detsky ME, Harhay MO, Bayard DF, Delman AM, Buehler AE, Kent SA, et al. Six-month morbidity and mortality among intensive care unit patients receiving life-sustaining therapy. A prospective cohort study. *Ann Am Thorac Soc* 2017;14(10):1562–70.

[4] Granholm A, Miller MH, Krag M, Perner A, Hjortrup PB. Predictive performance of the Simplified Acute Physiology Score (SAPS) II and the initial Sequential Organ Failure Assessment (SOFA) score in acutely ill intensive care patients: post-hoc analyses of the SUP-ICU inception cohort study. *PLOS ONE* 2016;11(12):e0168948. 10.1371/journal.pone.0168948.

[5] Li Z, Cheng B, Wang J, Xie G, Yu X, Huang M, et al. A multifactor model for predicting mortality in critically ill patients: a multicenter prospective cohort study. *J Crit Care* 2017;42:18–24.

[6] McGlynn EA, Asch SM, Adams J, Keesey J, Hicks J, DeCristofaro A, et al. The quality of health care delivered to adults in the United States. *N Engl J Med* 2003;348(26):2635–45.

[7] Steinberg EP. Improving the quality of care. Can we practice what we preach? *N Engl J Med* 2003;348(26):2681–3.

[8] Niewiński G, Starczewska M, Kański A. Prognostic scoring systems for mortality in intensive care units. The APACHE model. *Anaesthesiol Intensive Ther* 2014;46(1):46–9.

[9] Knaus WA, Draper EA, Wagner DP, Zimmerman JE. APACHE II: a severity of disease classification system. *Crit Care Med* 1985;13(10):818–29.

[10] Theresa SJ, Latheef F. Evaluation of acute physiology and chronic health evaluation (APACHE) II in predicting ICU mortality among critically ill. *Int J Adv Med* 2017;4(6):1566–72.

[11] Sekuli AD, Trpkovic SV, Pavlovic AP, Marinkovic OM, Llic AN. Scoring systems in assessing survival of critically ill ICU patients. *Med Sci Monit* 2015;21:2621–9.

[12] Godinjal A, Igllica A, Rama A, Tancica I, Jusufovic S, Ajanovic A, et al. Predictive value of SAPS II and APACHE II scoring systems for patient outcome in a medical intensive care unit. *Acta Med Acad* 2016;45(2):97–103.

[13] Barado Barado J, Guergué JM, Esparza L, Azcarate C, Mallor F, Ochoa S. A mathematical model for simulating daily bed occupancy in an intensive care unit. *Crit Care Med* 2012;40(4):1098–104.

[14] Garg AX. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review. *JAMA* 2005;293:1223–38.

[15] Chaudhry B, Wang J, Wu S, Maglione M, Mojica W, Roth E, et al. Systematic review: impact of health information technology on quality, efficiency, and costs of medical care. *Ann Intern Med* 2006;144:742–52.

[16] Tu JV, Guerriere MR. Use of a neural network as a predictive instrument for length of stay in the intensive care unit following cardiac surgery. *Comp Biomed Res* 1993;26:220–9.

[17] Doig G, Inman K, Sibbald W, Martin C, Robertson J. Modeling mortality in the intensive care unit: comparing the performance of a back-propagation, associative-learning neural network with multivariate logistic regression. In: Proceedings of the annual symposium on computer application in medical care, 1993; 1993. p. 361–5.

[18] Buchman TG, Kubos KL, Seidler AJ, et al. A comparison of statistical and connectionist models for the prediction of chronicity in a surgical intensive care unit. *Crit Care Med* 1994;22:750–62.

[19] Dybowski R, Gant V, Weller P, Chang R. Prediction of outcome in critically ill patients using artificial neural networks synthesised by genetic algorithm. *Lancet* 1996;347:1146–50.

[20] Hanson III CW, Marshall BE. Artificial intelligence applications in the intensive care unit. *Crit Care Med* 2001;29:427–35.

[21] Luaces O, Taboada F, Albaiceta GM, Domínguez LA, Enríquez P, Behamonde A. Predicting the probability of survival in intensive care unit patients from a small number of variables and training examples. *Artif Intell Med* 2009;45:63–76.

[22] Purushotham S, Meng C, Che Z, Liu Y. Benchmarking deep learning models on large healthcare datasets. *J Biomed Inform* 2018;83:112–34.

[23] Johnson AE, Pollard TJ, Shen L, Lehman LW, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. *Sci Data* 2016;3:160035. <https://doi.org/10.1038/sdata.2016.35>.

[24] Sadeghi R, Banerjee T, Romine W. Early hospital mortality prediction using vital signals. Submitted to smart health. 2019 [cs.LG] 9 Feb 2019, arXiv:1803.06589v2.

[25] Caicedo-Torres W, Gutiérrez J. ISeeU: Visually interpretable deep learning for mortality prediction inside the ICU. 2019 [cs.LG] 24 Jan 2019, arXiv:1901.08201v1.

- [26] Overweg H, Popkes AL, Ecole A, Li Y, Hernández-Lobato JM, Zaykov Y, et al. Interpretable outcome prediction with sparse bayesian neural networks in intensive care. 2019 [cs.LG] 9 Sep 2019, arXiv:1905.02599v2.
- [27] Maas AIR, Menon DK, Steyerberg EW, Citerio G, Lecky F, Manley GT, et al. Collaborative european neurotrauma effectiveness research in traumatic brain injury (center-tbi): a prospective longitudinal observational study. *Neurosurgery* 2014;76(1):67–80.
- [28] Pirracchio R, Petersen ML, Carone M, Rigon MR, Chevret S, van der Laan MJ. Mortality prediction in intensive care units with the Super ICU Learner Algorithm (SICULA): a population-based study. *Lancet Respir Med* 2015;3:42–52.
- [29] Aczon M, Ledbetter D, Ho L, Gunny A, Flynn A, Williams J, et al. Dynamic mortality risk predictions in pediatric critical care using recurrent neural networks. 2017. *Cs Math Q-Bio Stat* 2017 Jan 23, arXiv:170106675.
- [30] Spiegelhalter DJ. Incorporating Bayesian ideas into healthcare evaluation. *Stat Sci* 2004;19:156–74.
- [31] Walshe T, Burgman M. A framework for assessing and managing risks posed by emerging diseases. *Risk Anal* 2010;30(2):236–49.
- [32] Cruz-Ramírez N, Acosta-Mesa HG, Carrillo-Calvet H, Alonso Nava-Fernández L, Barrientos-Martínez RE. Diagnosis of breast cancer using BN: a case study. *Comput Biol Med* 2007;37:1553–64.
- [33] Gade J, Rosenfalck A, van Gils M, et al. Modelling techniques and their application for monitoring in high dependency environments-learning models. *Comput Methods Programs Biomed* 1996;51:75–84.
- [34] Nikiforidis GC, Sakellaropoulos GC. Expert system support using Bayesian belief networks in the prognosis of head-injured patients of the ICU. *Med Inf* 1998;23: 1–18.
- [35] Sandri M, Berchiolla P, Baldi I, Gregori D, De Blasi RA. Dynamic Bayesian Networks to predict sequences of organ failures in patients admitted to ICU. *J Biomed Inform* 2014;48:106–13.
- [36] Gultepe E, Green JP, Nguyen H, Adams J, Albertson T, Tagkopoulos I. From vital signs to clinical outcomes for patients with sepsis: a machine learning basis for a clinical decision support system. *J Am Med Inform Assoc* 2014;21:315–25.
- [37] Delgado R, Núñez-González JD, Yébenes JC, Lavado A. Vital prognosis of patients in intensive care units using an Ensemble of Bayesian Classifiers. In: *Proceedings of the LOD 2019*; 2019 [to appear in].
- [39] Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. *Mach Learn* 1997;29(2–3):131–63.
- [40] Davis J, Goadrich M. The relationship between precision-recall and roc curves. In: *Proceedings of the 23rd international conference on machine learning*; 2006. p. 233–40.
- [41] He H, Garcia EA. Learning from Imbalanced Data. *IEEE Trans Knowl Data Eng* 2009;21(9):1263–84.
- [42] Khadanga S, Aggarwal K, Joty S, Srivastava J. Using clinical notes with time series data for ICU management. 2019 [cs.CL] 12 Sep 2019, arXiv:1909.09702v1.
- [43] Scutari M. Learning Bayesian Networks with the bnlearn R Package. *J Stat Softw* 2010;35(3):1–22.
- [44] Hojsgaard S. Graphical independence networks with the gRain package for R. *J Stat Softw* 2012;46(10):1–26.
- [45] Wilcoxon F. Individual comparisons by ranking methods. *Biomet Bull* 1945;1(6): 80–3.
- [46] Shapiro SS, Wilk MB. An analysis of variance test for normality (complete samples). *Biometrika* 1965;52(3–4):591–611.
- [47] Delgado R, Gonzalez JL, Sotoca A, Tibau XA. Archetypes of wildfire arsonists: an approach by using bayesian networks. *Forest Fire Cap* 2018;2:25–50. Janusz Szmyt, IntechOpen.
- [48] Freeman LC. A set of measures of centrality based upon betweenness. *Sociometry* 1977;40:35–41.
- [49] Delgado R, Tibau XA, et al. Measuring features strength in probabilistic classification. In: Medina J, editor. *Information processing and management of uncertainty in knowledge-based systems. Theory and foundations. IPMU 2018. Communications in computer and information science, vol. 853. Cham: Springer; 2018.*

A.3 Modified Confusion Entropy

RESEARCH ARTICLE

Enhancing Confusion Entropy (CEN) for binary and multiclass classification

Rosario Delgado¹, J. David Núñez-González^{2*}

1 Department of Mathematics, Universitat Autònoma de Barcelona, Campus de la UAB, Cerdanyola del Vallès, Spain, **2** Department of Mathematics, University of the Basque Country (UPV/EHU), Leioa, Spain

* josedavid.nunez@ehu.eus



Abstract

Different performance measures are used to assess the behaviour, and to carry out the comparison, of classifiers in Machine Learning. Many measures have been defined on the literature, and among them, a measure inspired by Shannon's entropy named the Confusion Entropy (CEN). In this work we introduce a new measure, MCEN, by modifying CEN to avoid its unwanted behaviour in the binary case, that disables it as a suitable performance measure in classification. We compare MCEN with CEN and other performance measures, presenting analytical results in some particularly interesting cases, as well as some heuristic computational experimentation.

OPEN ACCESS

Citation: Delgado R, Núñez-González JD (2019) Enhancing Confusion Entropy (CEN) for binary and multiclass classification. PLoS ONE 14(1): e0210264. <https://doi.org/10.1371/journal.pone.0210264>

Editor: Francesco Ciccarello, Università degli Studi di Palermo Dipartimento di Fisica e Chimica, ITALY

Received: April 19, 2018

Accepted: December 15, 2018

Published: January 14, 2019

Copyright: © 2019 Delgado, Núñez-González. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work was supported by Ministerio de Economía y Competitividad, Gobierno de España, MTM2015 67802-P to R.D. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Introduction

Machine Learning is the subfield of Computer Science, as well as the branch of Artificial Intelligence, whose objective is to develop techniques that allow computers to learn. It has a wide range of applications, such as search engines or pattern recognition. Examples are: medical diagnosis, fraud detection, stock market analysis, classification of DNA sequences, recognition of speech and written language, images, games and robotics.

Machine learning tasks are typically grouped into two broad categories: Supervised and Unsupervised Learning. Classification falls in the former, since it deals with some input variables (features or characteristics) and an output variable (the class), and uses an algorithm to infer the class of (that is, to classify) a new case from its known features. Different models are used to build classifiers. Decision Trees (J48, Random Forest), Rules (Decision Table, JRip, ZeroR), Neural Networks (Multilayer Perceptron, Extreme Learning Machines, RBFN), Support Vector Machines, and Bayesian Networks (Naive Bayes, TAN) are some, although not the only ones, approximations to supervised classification.

Once a classifier is built, a performance measure is needed in order to assess its behaviour and to compare it with other classifiers. In the binary case, in which the class variable has only two labels or classes, there are several classical measures that have been widely used: Accuracy, Sensitivity, Specificity and F-score, only to mention some of the most commonly used. Not of all them allow a natural extension to the multi-class case (more than two labels), and only few measures have been specially designed for multi-class classification, which is a more complex scenario. Accuracy, by far the simplest and widespread performance measure

in classification, extends seamlessly its definition in the binary case to multi-class classification. Another well known performance measure, formerly introduced in the binary case but that extends without problems, is Matthew's Correlation Coefficient (MCC), introduced by Matthews in [1].

In this work, whose seed is [2], we focus on a different performance measure, named Confusion Entropy (CEN), which measures the uncertainty generated by classification, and has been recently introduced by Wang et al. in [3] as a novel measure for evaluating classifiers based on the concept of Shannon's entropy. CEN measures generated entropy from misclassified cases considering not only how the cases of each fixed class have been misclassified into other classes, but also how the cases of the other classes have been misclassified as belonging to this class, as well as entropy inside well-classified cases. Given a set of non-negative numbers, say $\{n_1, \dots, n_r\}$, the Shannon's entropy generated by the set can be defined as the sum $\sum_{i=1}^r -p_i \log(p_i)$, with $p_i = \frac{n_i}{n}$ if $n = \sum_{i=1}^r n_i$, where \log can be, as usual, the logarithm in base 2.

CEN is compared in [3] with Accuracy and other measures, showing a relative consistency with them: higher Accuracy tends to result in lower Confusion Entropy. This performance measure, which is more discriminating for evaluating classifiers than Accuracy, specially when the number of cases grows, has also been studied in [4], where the authors show the strong monotone relation between CEN and MCC, and that both, MCC and CEN, improve over Accuracy.

There are some works in the recent literature using Confusion Entropy. For example, in [5] the authors propose a novel splitting criterion based on CEN for learning decision trees with higher performance; experimental results on some data sets show that this criterion leads to trees with better CEN value without reducing accuracy. The authors of [6] and [7] use CEN, among other performance measures, to compare several common data mining methods used with highly imbalanced data sets where the class of interest is rare. Other works propose modifications of this measure, as [8], in which a Confusion Entropy measure based on a probabilistic confusion matrix is introduced, measuring if cases are classified into true classes and separated from others with high probabilities. A similar approach to that of [8] is followed in [9] to analyze the probability sensitivity of the Gaussian processes in a bankruptcy prediction context, by means of a probabilistic confusion entropy matrix based on the model estimated probabilities. In the context of horizontal collaboration, the system global entropy is introduced in [10] analogously to CEN (see also [11] and [12]), and it is used in the collaborative part of a clustering algorithm, which is iterative with the optimization process continuing as long as the system global entropy is not stable.

It is remarkable that CEN shows to have a weakness in the binary case that invalidates it as a suitable performance measure: in some situations CEN gets values larger than one, unlike what happens in the multi-class case, in which CEN ranges between zero and one. CEN is a measure of the "overall" entropy associated to the confusion matrix, that can be thought as generated by two sources: entropy within the main diagonal, and the one generated by the values outside it, corresponding to misclassification. We will show that CEN is more sensible to the later. A second but not least important point in the weakness of the behaviour of CEN is its lack of monotonicity when the overall entropy does increase (or decrease) monotonously. Along the paper we will show different situations to stand out these items.

Our aim is to introduce an enhanced CEN measure, that we denote by MCEN, and compare it with CEN, MCC and Accuracy. This new measure will show to be highly correlated with CEN. Two aspects deserve to be highlighted:

1. definitions of probabilities involved in the construction of CEN have been modified in MCEN to improve interpretability as real probabilities,
2. weakness of CEN in the binary case (out-of-range and lack of monotonicity) are overcome with MCEN.

The paper is structured as follows: first we introduce the Modified Confusion Entropy MCEN and deal with the multi-dimensional perfectly symmetric and balanced case, which is deeply studied, performing a cross comparison between CEN, MCEN, Accuracy and MCC. The general binary case is treated next, focusing on different families of matrices and carrying out the corresponding cross comparisons. Next part is devoted to study the Z_A family of confusion matrices. Then, we compare CEN, MCEN, Accuracy and MCC with two recently introduced measures: the Probabilistic Accuracy PACC ([13]) and the Entropy-Modulated Accuracy EMA ([14]). Finally, some experiments performed in the binary setting to compare CEN with MCEN through four real database sets are included in the Supporting Information file. These experiments show that their behaviour is mostly analog, but when it is not the case, MCEN is the one that behaves more according to entropy generated by misclassification. The paper finishes with a conclusion section.

Methods

Given a multi-class classifier learned from a training dataset, with $N \geq 2$ classes labelled $\{1, 2, \dots, N\}$, we apply it in order to classify cases from a testing dataset, that is, to infer the class of the cases from their known features or characteristics. Since for the cases in the testing dataset we actually know the class to which they belong, we can construct the $N \times N$ confusion matrix $C = (C_{ij})_{i,j=1, \dots, N}$, which collects the results issued by the classifier over the testing dataset. C_{ij} is the number of cases of class i that have been classified as belonging to class j . We denote by S the sum of values of the matrix, that is, the total number of cases in the testing dataset,

$$S = \sum_{i=1}^N \sum_{j=1}^N C_{ij}.$$

We introduce notations $OUT(C)$ and $IN(C)$, respectively, to denote the Shannon's entropy generated by the elements of outside (respectively, inside) the main diagonal of matrix C . That is, while IN is the entropy generated by the well classified cases, OUT is generated by misclassification.

In [3] the misclassification probability of classifying class- i cases as being of class j "subject to class j ", denoted by P_{ij}^j , is introduced as:

$$P_{ij}^j = \frac{C_{ij}}{\sum_{k=1}^N (C_{i,k} + C_{k,j})}, \quad i, j = 1, \dots, N, \quad i \neq j, \tag{1}$$

that is, P_{ij}^j is "almost" the relative frequency class- i cases that are classified as being of class j among all cases that are of class j or that have been classified as being of class j . But not exactly. The reason is that class- j cases that have been correctly classified, whose number is $C_{j,j}$, are counted twice in the denominator.

Analogously, the misclassification probability of classifying class- i cases as being of class- j "subject to class i ", with analogous interpretation, denoted by P_{ij}^i , is defined in the same paper by:

$$P_{ij}^i = \frac{C_{ij}}{\sum_{k=1}^N (C_{i,k} + C_{k,i})}, \quad i, j = 1, \dots, N, \quad i \neq j. \tag{2}$$

Then, the Confusion Entropy associated to class j is defined in [3] by:

$$\text{CEN}_j = - \sum_{k=1, k \neq j}^N (P_{j,k}^j \log_{2(N-1)}(P_{j,k}^j) + P_{k,j}^j \log_{2(N-1)}(P_{k,j}^j)) \tag{3}$$

with the convention $a \log_b(a) = 0$ if $a = 0$. Finally, the overall Confusion Entropy associated to the confusion matrix C is defined as a convex combination of the Confusion Entropy of the classes as follows:

$$\text{CEN} = \sum_{j=1}^N P_j \text{CEN}_j, \tag{4}$$

where the non-negative weights P_j , summing 1, are

$$P_j = \frac{\sum_{k=1}^N (C_{j,k} + C_{k,j})}{2 \sum_{k,\ell=1}^N C_{k,\ell}}. \tag{5}$$

Note that CEN is an invariant measure; if we multiply all elements of the confusion matrix by a constant we obtain the same result. The same convenient and useful property holds with Accuracy, MCC and the modified Confusion Entropy measure MCEN, that we will introduce below. As MCC lives in $[-1, 1]$ while Accuracy, CEN and MCEN range in $[0, 1]$, we scale MCC and introduce $\text{MCC}^* = \frac{1-\text{MCC}}{2} \in [0, 1]$. Besides, since Accuracy usually has an inverse relationship with both CEN and MCEN, we choose to consider $\text{ACC}^* = 1 - \text{Accuracy}$ instead of Accuracy itself.

For $N > 2$, CEN ranges between 0 and 1, 0 is attained with perfect classification (the off-diagonal elements of matrix C being zero), while 1 under complete misclassification, symmetry and balance in C , that is, if all diagonal elements in C are zero, and the off-diagonal elements take all the same value. In the binary case ($N = 2$), although CEN remains to be 0 with perfect classification, and is 1 under complete misclassification with symmetry, in intermediate scenarios we can also obtain $\text{CEN} = 1$ and even higher values. That is, in some cases CEN is out-of-range. See, for example, the confusion matrices in Table 1, which have already been considered in [4]. The lack of monotonicity when the situation monotonously goes from perfect classification to completely symmetric and balanced misclassification, as showed by the sequence of matrices in Table 1, represents a great inconvenience of CEN in the binary case, and is our main motivation for introducing a modified version of it.

Definition

Instead of (1), we propose to introduce the probability of classifying class- i cases in class j “subject to class j ”, as

$$\tilde{P}_{ij}^j = \frac{C_{ij}}{\sum_{k=1}^N (C_{j,k} + C_{k,j}) - C_{j,j}}, \quad i, j = 1, \dots, N, \quad i \neq j.$$

Table 1. Examples in the perfectly symmetric and balanced binary case with $S = 12$. Only CEN values.

	$\begin{pmatrix} 6 & 0 \\ 0 & 6 \end{pmatrix}$	$\begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix}$	$\begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix}$	$\begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$	$\begin{pmatrix} 2 & 4 \\ 4 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 5 \\ 5 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 6 \\ 6 & 0 \end{pmatrix}$
CEN =	0.0000	0.5975	0.8617	1.0000	1.0566	1.0525	1.0000

<https://doi.org/10.1371/journal.pone.0210264.t001>

that is, we overcome the fact that in (1) correctly classified class- j cases are counted twice in the denominator. With this definition, \tilde{P}_{ij}^j is really the relative frequency of class- i cases classified as belonging to class j among all cases that are of class j or that have been classified as being of class j . Analogously, we modify definition (2) in the same sense:

$$\tilde{P}_{ij}^i = \frac{C_{ij}}{\sum_{k=1}^N (C_{i,k} + C_{k,i}) - C_{i,i}}, \quad i, j = 1, \dots, N, \quad i \neq j,$$

and \tilde{P}_{ij}^i is really the relative frequency of class- i cases classified in class j among all cases that are of class i or that have been classified as being of class i .

Next, we modify definition of the weights in (5) in the following way:

$$\tilde{P}_j = \frac{\sum_{k=1}^N (C_{j,k} + C_{k,j}) - C_{j,j}}{2 \sum_{k,\ell=1}^N C_{k,\ell} - \alpha \sum_{k=1}^N C_{k,k}},$$

where

$$\alpha = \begin{cases} 1/2 & \text{if } N = 2 \\ 1 & \text{if } N > 2. \end{cases}$$

Then, we define the Confusion Entropy associated to class j as in (3) by

$$\text{MCEN}_j = - \sum_{k=1, k \neq j}^N (\tilde{P}_{j,k}^j \log_{2(N-1)}(\tilde{P}_{j,k}^j) + \tilde{P}_{k,j}^j \log_{2(N-1)}(\tilde{P}_{k,j}^j)),$$

and the modified Confusion Entropy as in formula (4), that is,

$$\text{MCEN} = \sum_{j=1}^N \tilde{P}_j \text{MCEN}_j. \tag{6}$$

Note that when $N > 2$, $\sum_{j=1}^N \tilde{P}_j = 1$, so the modified overall Confusion Entropy is also defined as a convex combination of the modified Confusion Entropy corresponding to the classes, while in the binary case ($N = 2$), it is just defined as a conical combination since although the weights \tilde{P}_j are non-negative, they do not necessarily sum up to 1 (indeed, their sum is 1 if and only if all the diagonal elements of the confusion matrix C are zero, that is, if all cases have been misclassified).

We see from (4) and (6) that both measures CEN and MCEN, are decomposable along classes, which makes it easy to assess the effect on the behaviour of the classifier of a simple modification affecting just one class.

We can start performing a preliminary comparison of the behaviour of ACC^* , MCC^* , CEN and MCEN in the toy example in dimension 2 of Table 2. In this example, the baseline confusion matrix is constant with all its entries equal to 3. First, maintaining the total sum equal to $S = 12$ and the out-diagonal invariant, we reduce the entropy IN in Table 2(a). In the baseline case, the diagonal elements are the set $\{3, 3\}$, whose entropy is 1 (maximum value). The corresponding values of IN in case (a) are consigned in Table 2, in a decreasing order. Analogously for Table 2(b) but in this case changes have been introduced outside the main diagonal. We observe that while ACC^* remains insensitive to changes in the arrangement of the elements of the matrix, since the sum of the main diagonal remains constant, MCC^* only decreases with decreasing entropy OUT, while when IN decreases, its value increases. As far as their interpretation is concerned, both CEN and MCEN measure the overall entropy of the confusion

Table 2. Toy example: Binary case with S = 12. (a): Entropy reduction within the main diagonal, IN. (b) Entropy reduction outside the main diagonal, OUT. In brackets the relative reduction in each measure with respect to the baseline case. Entropy refers to IN in (a) and to OUT in (b).

	Baseline	(a)			(b)		
	$\begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$	$\begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}$	$\begin{pmatrix} 1 & 3 \\ 3 & 5 \end{pmatrix}$	$\begin{pmatrix} 0 & 3 \\ 3 & 6 \end{pmatrix}$	$\begin{pmatrix} 3 & 2 \\ 4 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 1 \\ 5 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 0 \\ 6 & 3 \end{pmatrix}$
Entropy =	1.0000	0.9183	0.6500	0.0000	0.9183	0.6500	0.0000
		(8.17%)	(35.00%)	(100.00%)	(8.17%)	(35.00%)	(100.00%)
ACC* =	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
MCC* =	0.5000	0.5130	0.5625	0.6667	0.4881	0.4375	0.3333
CEN =	1.0000	0.9898	0.9575	0.8962	0.9591	0.8250	0.5000
		(1.02%)	(4.25%)	(10.38%)	(4.09%)	(17.50%)	(50.00%)
MCEN =	0.9057	0.9006	0.8848	0.8571	0.8590	0.7057	0.3343
		(0.56%)	(2.31%)	(5.37%)	(5.16%)	(22.08%)	(63.09%)

<https://doi.org/10.1371/journal.pone.0210264.t002>

matrix, giving less weight to the IN entropy, that is, that generated by the well classified cases, than to OUT entropy, corresponding to misclassification. In this example we observe how their values are reduced when IN decreases, maintaining its constant sum, or when the one that is reduced is OUT, but in this second case the reduction is much more drastic, both for CEN and MCEN, and more sharply for the second. The main difference between CEN and MCEN in this sense is that the former is more sensitive to changes of IN entropy than MCEN, while less than CEN to that of OUT (observe the percentages in brackets in Table 2, which are the relative reduction in the measure with respect to that of the baseline case).

We can extend this comparison to matrices of type $M_A = \begin{pmatrix} 1 & 50 \\ A & 1 \end{pmatrix}$, with $A = 1, \dots, 100$, for example. Their main diagonal stays constant. Fig 1 shows the behaviour of CEN, MCEN, ACC* and MCC* as OUT increases. We can observe that indeed, CEN is less correlated with this entropy than MCEN. The same can be observed from the correlations matrix given in Table 3.

Instead, if we consider matrices $W_A = \begin{pmatrix} 50 & 1 \\ 1 & A \end{pmatrix}$, with $A = 1, \dots, 100$, the values outside the main diagonal stay constant. Fig 2 shows the behaviour of CEN, MCEN, ACC* and MCC* as IN increases. CEN shows more correlation with this entropy than MCEN (see Table 4), although IN is less correlated (and in an inverse sense that could not be appreciated in the toy example of Table 2) than OUT, both with CEN and MCEN.

The perfectly symmetric and balanced case

In this section we consider the case in which $C_{i,j} = F$ for all $i, j = 1, \dots, N, i \neq j$ and $C_{i,i} = T$,

$$\text{with } T \geq 0, F > 0, \text{ that is, } C = \begin{pmatrix} T & F & \dots & F & F \\ F & T & \dots & F & F \\ \vdots & \vdots & \dots & \vdots & \vdots \\ F & F & \dots & T & F \\ F & F & \dots & F & T \end{pmatrix}.$$

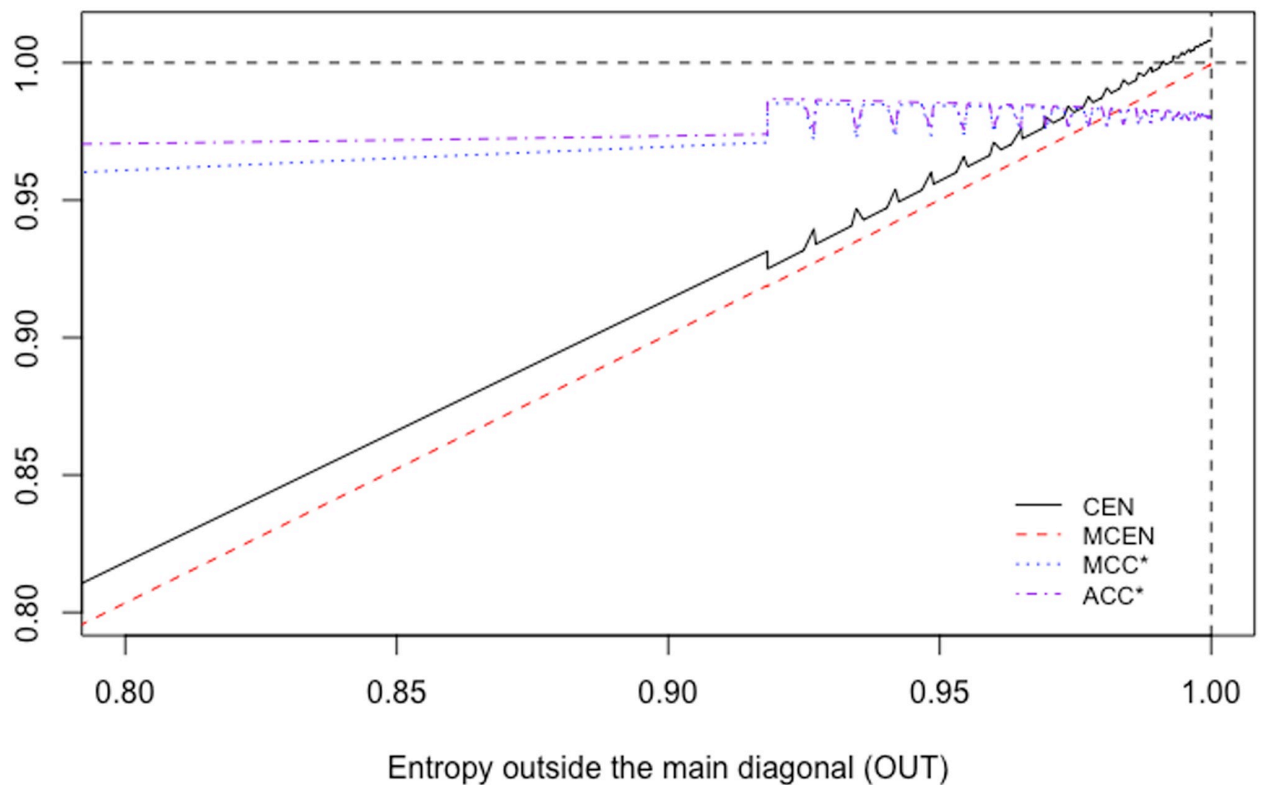


Fig 1. CEN, MCEN, ACC* and MCC* for matrix M_A , as function of entropy outside the diagonal.

<https://doi.org/10.1371/journal.pone.0210264.g001>

Proposition 1 In the perfectly symmetric and balanced case,

$$\begin{aligned} \text{If } N > 2, \quad \text{CEN} &= \frac{2(N-1)}{\delta} \log_{2(N-1)}(\delta), \quad \text{MCEN} = 2(N-1)\tilde{\delta} \log_{2(N-1)}(\tilde{\delta}), \\ \text{If } N = 2, \quad \text{CEN} &= \frac{1}{1+\gamma} \log_2(\delta), \quad \text{MCEN} = \frac{1}{1+\frac{3}{4}\gamma} \log_2(\tilde{\delta}), \end{aligned} \quad (7)$$

where

$$\gamma = \frac{T}{F} \geq 0, \quad \delta = 2(N-1) + 2\gamma > 0 \quad \text{and} \quad \tilde{\delta} = 2(N-1) + \gamma > 0,$$

$$\text{ACC}^* = \frac{N-1}{\gamma + (N-1)} \quad \text{and} \quad \text{MCC}^* = \frac{N}{2(\gamma + (N-1))} = \frac{N}{2(N-1)} \text{ACC}^*.$$

Table 3. Correlation matrix (Pearson) for the measures of the family of matrices M_A , $A = 1, \dots, 100$.

	CEN	MCEN	MCC*	ACC*	OUT
CEN	1.0000000	0.9999334	0.9229026	0.7783573	0.9999320
MCEN		1.0000000	0.9233945	0.7855300	0.9999963
MCC*			1.0000000	0.7340543	0.9241870
ACC*				1.0000000	0.7852756
OUT					1.0000000

<https://doi.org/10.1371/journal.pone.0210264.t003>

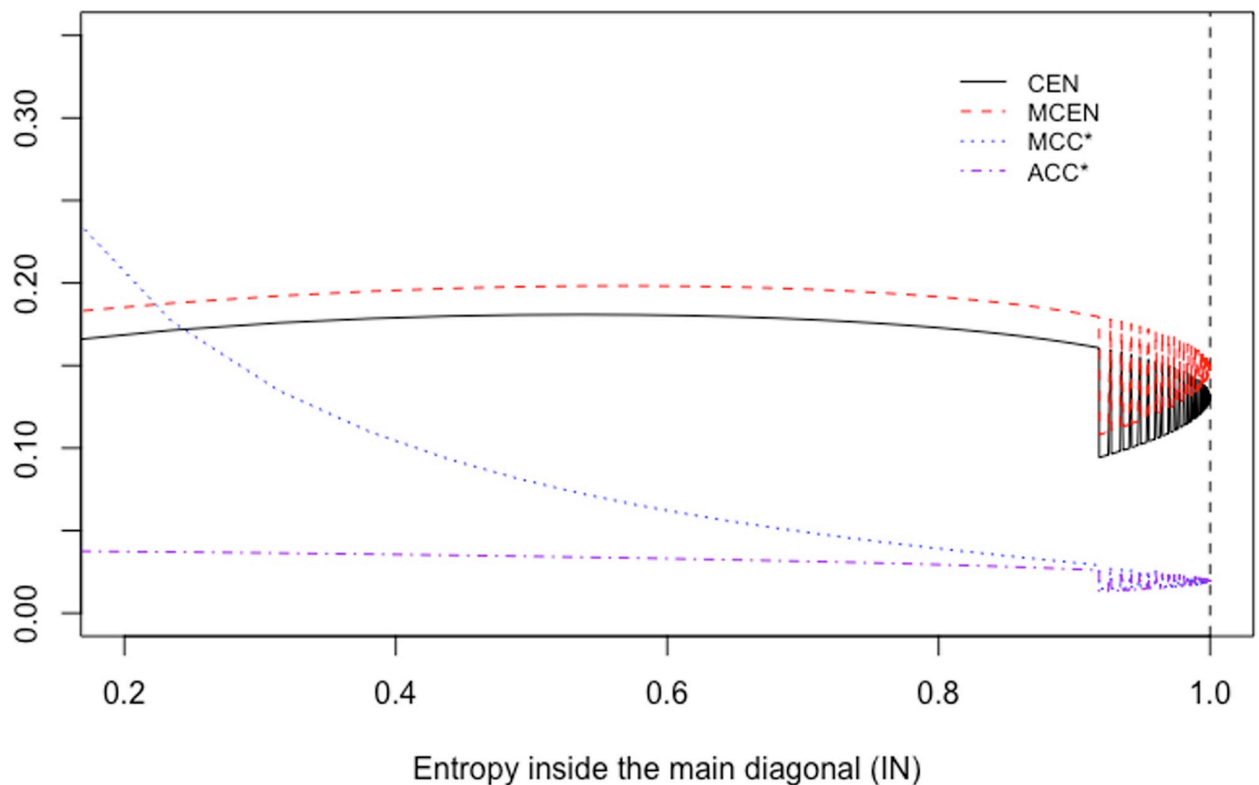


Fig 2. CEN, MCEN, ACC* and MCC* for matrix W_A , as function of entropy inside the diagonal.

<https://doi.org/10.1371/journal.pone.0210264.g002>

Note that ACC*, MCC*, CEN and MCEN depend on the matrix values T and F only through its ratio γ . In (7) (case $N > 2$), CEN and MCEN have the same expression except that CEN depends on δ , which is function of 2γ , while MCEN does on $\tilde{\delta} = \delta - \gamma$, which is the same function but of γ . Therefore,

$$\text{if } N > 2, \quad \text{MCEN}(2\gamma) = \text{CEN}(\gamma),$$

where in the notation we highlight the dependency of CEN and MCEN on γ .

Corollary 1 *In the perfectly symmetric and balanced case, we have that:*

Table 4. Correlation matrix (Pearson) for the measures of the family of matrices $W_A, A = 1, \dots, 100$.

	CEN	MCEN	MCC*	ACC*	IN
CEN	1.0000000	0.9995962	0.5499231	0.9672182	-0.6062876
MCEN		1.0000000	0.5355098	0.9609698	-0.5857654
MCC*			1.0000000	0.7340543	-0.9241870
ACC*				1.0000000	-0.7852756
IN					1.0000000

<https://doi.org/10.1371/journal.pone.0210264.t004>

- For any $N > 2$, CEN, MCEN, ACC* and MCC* are monotonically decreasing functions of $\gamma \geq 0$, with

$$\lim_{\gamma \rightarrow +\infty} \text{CEN}(\gamma) = \lim_{\gamma \rightarrow +\infty} \text{MCEN}(\gamma) = \lim_{\gamma \rightarrow +\infty} \text{ACC}^*(\gamma) = \lim_{\gamma \rightarrow +\infty} \text{MCC}^*(\gamma) = 0,$$

$$\text{CEN}(0) = \text{MCEN}(0) = \text{ACC}^*(0) = 1, \text{MCC}^*(0) = \frac{N}{2(N-1)} \rightarrow \frac{1}{2} \text{ as } N \rightarrow +\infty,$$

and if $\gamma > 0$, $\text{MCC}^* < \text{ACC}^* < \text{CEN} < \text{MCEN}$.

- Nevertheless, when $N = 2$, we have that although MCEN and $\text{ACC}^* = \text{MCC}^*$ remain to be monotonically decreasing as functions of $\gamma \geq 0$, CEN does not. Indeed, CEN achieves its global maximum when $\gamma = \frac{\xi}{2} - 1$, which is $\text{CEN}(\frac{\xi}{2} - 1) \approx 1.06148 > 1$. More specifically,

$$\text{CEN}(0) = \text{CEN}(1) = 1, \text{CEN}(\gamma) > 1, \text{ for all } 0 < \gamma < 1, \lim_{\gamma \rightarrow +\infty} \text{CEN}(\gamma) = 0,$$

$$\text{MCEN}(0) = 1, \lim_{\gamma \rightarrow +\infty} \text{MCEN}(\gamma) = 0,$$

$$\text{ACC}^*(0) = \text{MCC}^*(0) = 1, \lim_{\gamma \rightarrow +\infty} \text{ACC}^*(\gamma) = \lim_{\gamma \rightarrow +\infty} \text{MCC}^*(\gamma) = 0.$$

Moreover, there exists $\gamma_0 \approx 5.78$ such that

$$\begin{aligned} \text{MCC}^* &= \text{ACC}^* < \text{MCEN} < \text{CEN} \text{ if } 0 < \gamma < \gamma_0, \\ \text{MCC}^* &= \text{ACC}^* < \text{MCEN} = \text{CEN} \text{ if } \gamma = \gamma_0, \text{ and} \\ \text{MCC}^* &= \text{ACC}^* < \text{CEN} < \text{MCEN} \text{ if } \gamma > \gamma_0. \end{aligned}$$

Proof 1 The proofs of both Proposition 1 and Corollary 1 are straightforward, and then omitted. However, it is worth mentioning that in order to prove $\text{CEN} < \text{MCEN}$ in case $N > 2$ we use that function $f(x) = \frac{1}{x} \log_b(x)$ is strictly decreasing for any base $b > 1$ (in our case, $b = 2(N-1) \geq 4$), and $x > e$. We apply that fact to see that $f(x_0) > f(x_1)$ with $x_0 = 2(N-1) + \gamma < x_1 = 2(N-1) + 2\gamma$, since $x_0 \geq 4 > e$.

The same property of function f allows to prove that both CEN and MCEN are monotonically decreasing as functions of γ , with $x = \delta = 2(N-1) + 2\gamma$ and $x = \tilde{\delta} = 2(N-1) + \gamma$, respectively, being both $> e$ for any $\gamma \geq 0$. Note that since for $N = 2$ the expression of CEN as function of δ is as in case $N > 2$, the monotonous decrease fails since $x = \delta = 2 + 2\gamma < e$ for $\gamma < \frac{\xi}{2} - 1$.

The rest of proofs are also omitted.

Remark 1 Note that if $N = 2$, CEN exhibits the unwanted behaviour, not showed by MCEN, of being out-of-range $[0, 1]$, which despairs for $N > 2$ (see Figs 3 and 4).

Remark 2 Consider the particular case in which $T = F$, that is, $\gamma = 1$. In other words, the con-

fusion matrix is constant, say $\begin{pmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}$. Then, $\text{ACC}^* = \frac{N-1}{N}$ and $\text{MCC}^* = \frac{1}{2}$. More-

over, $\delta = 2N$ and $\tilde{\delta} = 2N - 1$.

If $N > 2$, $\text{CEN} = (1 - \frac{1}{N}) \log_{2(N-1)}(2N)$ and $\text{MCEN} = (1 - \frac{1}{2N-1}) \log_{2(N-1)}(2N - 1)$.

If $N = 2$, $\text{CEN} = 1$ and $\text{MCEN} = \frac{4}{7} \log_2(3) < 1$.

As a consequence, we can easily check that if $N > 2$, $\text{MCC}^* < \text{ACC}^* < \text{CEN} < \text{MCEN}$, with $\lim_{N \rightarrow +\infty} \text{ACC}^* = \lim_{N \rightarrow +\infty} \text{CEN} = \lim_{N \rightarrow +\infty} \text{MCEN} = 1$, while if $N = 2$, $\text{MCC}^* = \text{ACC}^* < \text{MCEN} < \text{CEN}$.

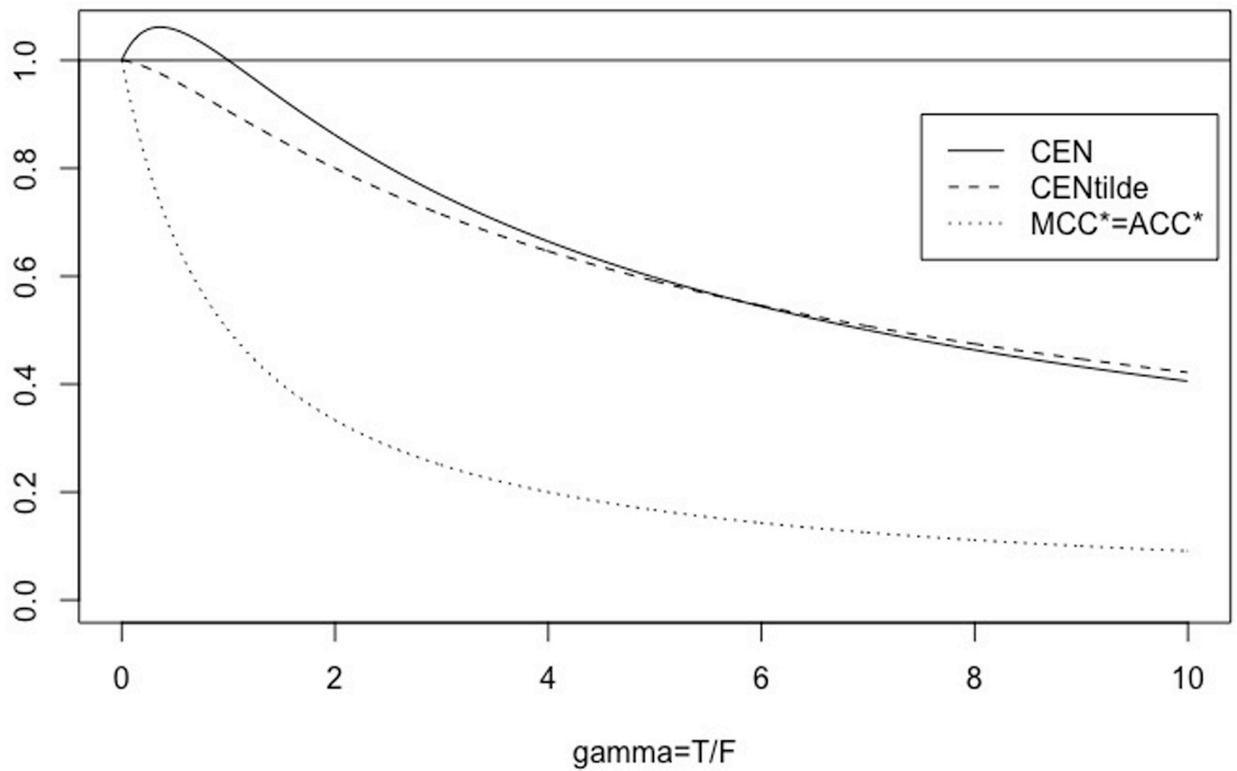


Fig 3. The symmetric case. CEN, MCEN, ACC* and MCC* for $\gamma \in [0, 10]$, with $N = 2$.

<https://doi.org/10.1371/journal.pone.0210264.g003>

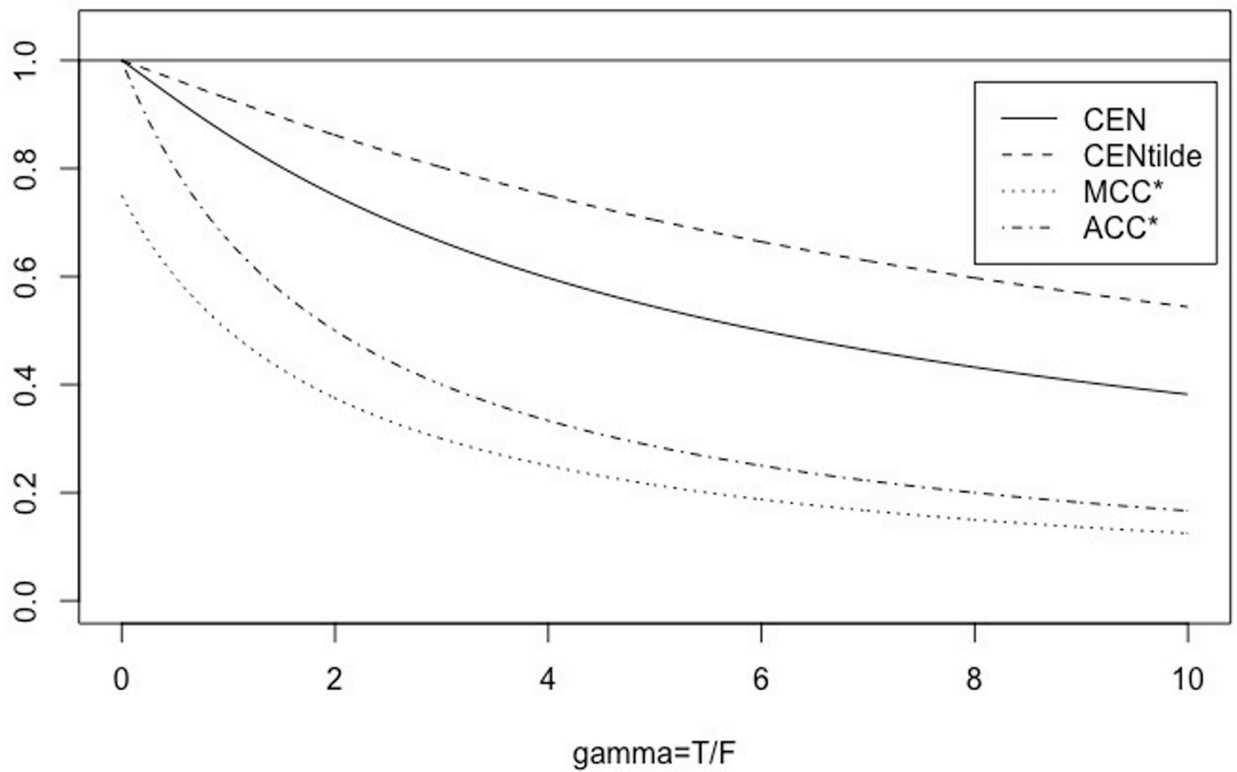


Fig 4. The symmetric case. CEN, MCEN, ACC* and MCC* for $\gamma \in [0, 10]$, with $N = 3$.

<https://doi.org/10.1371/journal.pone.0210264.g004>

The particular pathological case of matrices Z_A will be studied in the multi-class setting, but before we consider in some detail the binary case.

The general binary case

The binary case ($N = 2$) can be studied in more detail. We will use the following notation for the confusion matrix in the most general setting, taking class 1 as reference:

$$C = \begin{pmatrix} TP & FN \\ FP & TN \end{pmatrix}, \tag{8}$$

where TP is the true positive or number of class-1 cases that have been correctly classified, and the same for the true negative number of cases TN with class 2. On the other hand, FP denotes false positives or number of class-2 cases that have been misclassified, and FN false negatives.

Proposition 2 *If the confusion matrix C is given by (8), we have that with $S = TP + TN + FP + FN$,*

$$\begin{aligned} \text{CEN} &= \frac{(FN + FP) \log_2(S^2 - (TP - TN)^2)}{2S} - \frac{FN \log_2(FN) + FP \log_2(FP)}{S}, \\ \text{MCEN} &= \frac{2(FN + FP) \log_2((S - TN)(S - TP))}{3S + (FN + FP)} - \frac{4(FN \log_2(FN) + FP \log_2(FP))}{3S + (FN + FP)}, \\ \text{ACC}^* &= \frac{FP + FN}{S} \text{ and } \text{MCC}^* = \frac{1 - \text{MCC}}{2}, \\ \text{with } \text{MCC} &= \frac{TP \, TN - FP \, FN}{\sqrt{(TP + FN)(FP + TN)(TP + FP)(TN + FN)}}. \end{aligned} \tag{9}$$

To carry out a deeper study, we have to consider particular situations; is what we do in the subsections below, where different particular scenarios have been introduced and developed.

The perfectly symmetric and balanced case. Table 5 below shows some examples of

2×2 confusion matrices of type $\begin{pmatrix} T & F \\ F & T \end{pmatrix}$, that is, in which $TP = TN = T$ and $FP = FN = F$.

All of them correspond to $S = 12$ and have already been considered in [4]. This is a particular case of the previously considered setting, and Proposition 1 and Corollary 1 apply here. We can observe again the anomalous behaviour of CEN, in contrast with the other measures.

The symmetric but unbalanced family U_A . Consider the particular case of a confusion

matrix of type $U_A = \begin{pmatrix} 1 & A \\ A & 0 \end{pmatrix}$, with $A > 0$. Both class-1 and class-2 cases are mainly misclassified if $A > 1$. Entropy out of the main diagonal is 1 and within the diagonal is 0, regardless of the value of A . When $0 < A < 1$, say for example that $A = 1/B$ with $B > 1$, then matrix U_A is equivalent to $\begin{pmatrix} B & 1 \\ 1 & 0 \end{pmatrix}$, that is, corresponds to an unbalanced scenario in which class 2 is

Table 5. Examples in the perfectly symmetric and balanced binary case with $S = 12$.

	$\begin{pmatrix} 6 & 0 \\ 0 & 6 \end{pmatrix}$	$\begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix}$	$\begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix}$	$\begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$	$\begin{pmatrix} 2 & 4 \\ 4 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 5 \\ 5 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 6 \\ 6 & 0 \end{pmatrix}$
ACC* = MCC* =	0.0000	0.1667	0.3333	0.5000	0.6667	0.8333	1.0000
CEN =	0.0000	0.5975	0.8617	1.0000	1.0566	1.0525	1.0000
MCEN =	0.0000	0.5910	0.8000	0.9057	0.9614	0.9891	1.0000

<https://doi.org/10.1371/journal.pone.0210264.t005>

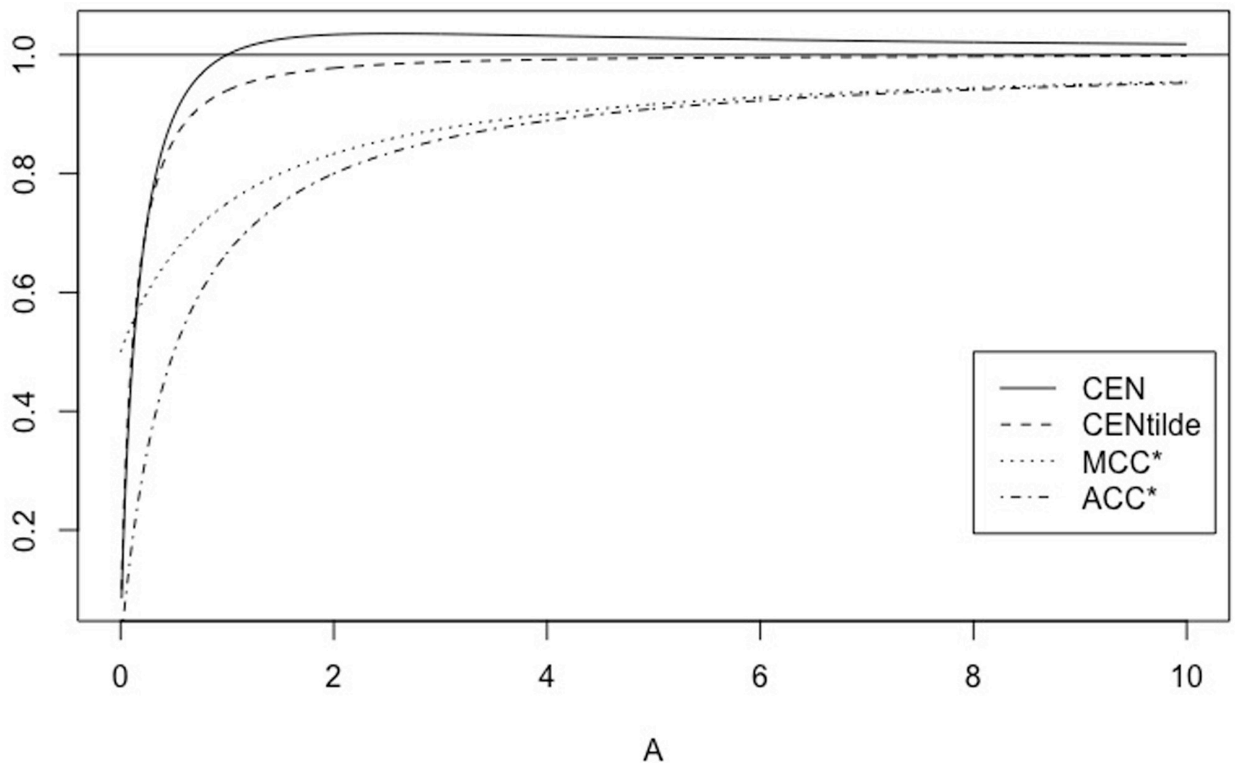


Fig 5. Family U_A . CEN, MCEN, ACC* and MCC* for $A \in (0, 10]$.

<https://doi.org/10.1371/journal.pone.0210264.g005>

underrepresented and class-1 cases are mainly well classified. We can observe some properties of CEN, MCEN, ACC* and MCC* (see Fig 5) in Proposition 3, which is derived from Proposition 2.

Proposition 3 For confusion matrix U_A with $A > 0$, we have:

$$\begin{aligned} \text{CEN}(A) &= \frac{A \log_2((2A + 1)^2 - 1) - 2A \log_2(A)}{2A + 1}, \\ \text{MCEN}(A) &= \frac{4A \log_2(2A(2A + 1)) - 8A \log_2(A)}{3(2A + 1) + 2A}, \\ \text{ACC}^*(A) &= \frac{2A}{2A + 1}, \quad \text{MCC}^*(A) = \frac{2A + 1}{2(A + 1)}. \end{aligned}$$

As a consequence:

$\text{CEN}(A) < 1$ if $A < 1$, $\text{CEN}(1) = 1$, $\text{CEN}(A) > 1$ if $A > 1$, $\text{MCEN}(A) < 1$ and $\text{ACC}^*(A) < \text{MCC}^*(A) < 1$, for all $A > 0$, MCEN, ACC* and MCC* are monotonically increasing functions of $A > 0$, CEN is not, and achieves its global maximum when $A \approx 2.54$, which is > 1 ,

$$\lim_{A \rightarrow 0} \text{CEN}(A) = \lim_{A \rightarrow 0} \text{MCEN}(A) = \lim_{A \rightarrow 0} \text{ACC}^*(A) = 0 < \lim_{A \rightarrow 0} \text{MCC}^*(A) = 0.5,$$

$$\lim_{A \rightarrow +\infty} \text{CEN}(A) = \lim_{A \rightarrow +\infty} \text{MCEN}(A) = \lim_{A \rightarrow +\infty} \text{ACC}^*(A) = \lim_{A \rightarrow +\infty} \text{MCC}^*(A) = 1.$$

Moreover, there exists $A_0 \in (0, 1)$ (indeed, $A_0 \approx 0.24$) such that

$$\begin{aligned} \text{MCEN}(A) &< \text{CEN}(A) \text{ if } A > A_0, \\ \text{MCEN}(A_0) &= \text{CEN}(A_0), \\ \text{MCEN}(A) &> \text{CEN}(A) \text{ if } 0 < A < A_0. \end{aligned}$$

The overall entropy associated to the four elements of the confusion matrix, which results to be $-\frac{2A}{2A+1} \log\left(\frac{A}{2A+1}\right)$, increases to 1 when $A \rightarrow +\infty$ and decreases to 0 when $A \rightarrow 0$, and both CEN and MCEN, are sensible to this fact. Note that the lack of monotonicity of CEN(A) as A (and then, as the overall entropy) monotonically increases, is an anomalous behaviour that MCEN has managed to overcome. Moreover, MCEN ranges between 0 and 1. We can also observe this phenomenon in the examples in Table 6.

The asymmetric family V_A . Consider the particular case of confusion matrices of type $V_A = \begin{pmatrix} 1 & A \\ 1 & 0 \end{pmatrix}$, with $A > 0$. This is an asymmetric and unbalanced case in which class 2 is systematically misclassified and is underrepresented if $A > 1$. Class 1 is also mainly misclassified if $A > 1$. As $A \rightarrow +\infty$, entropy out the diagonal, which is $-\frac{A}{A+1} \log\left(\frac{A}{A+1}\right)$, decreases to zero. Entropy within diagonal is zero, while the overall entropy of the elements of matrix V_A is $\log(A+2) - \frac{A}{A+2} \log(A)$, which tends to 0 as $A \rightarrow +\infty$. When $0 < A < 1$ with $A = 1/B$, $B > 1$, matrix V_A is equivalent to $\begin{pmatrix} B & 1 \\ B & 0 \end{pmatrix}$, which corresponds to an almost balanced but asymmetric scenario in which class 1 is mainly well classified but class 2 is not. As B increases ($A \rightarrow 0$), entropy out the diagonal also drops to zero. Some properties of CEN, MCEN, ACC* and MCC* are given in Proposition 4 (see also Fig 6).

Proposition 4 For confusion matrix V_A with $A > 0$, we have:

$$\begin{aligned} \text{CEN}(A) &= \frac{(A+1) \log_2((A+2)^2 - 1) - 2A \log_2(A)}{2(A+2)}, \\ \text{MCEN}(A) &= \frac{2(A+1) \log_2((A+1)(A+2)) - 4A \log_2(A)}{3(A+2) + (A+1)}, \\ \text{ACC}^*(A) &= \frac{A+1}{A+2}, \quad \text{MCC}^*(A) = \frac{1 + \sqrt{\frac{A}{2(A+1)}}}{2}. \end{aligned}$$

As a consequence, there exists $A_1 \in (1, 2)$ ($A_1 \approx 1.414$) such that:

$\text{CEN}(A) > 1$ if $1 < A < A_1$, $\text{CEN}(1) = \text{CEN}(A_1) = 1$, $\text{CEN}(A) < 1$ if $A \notin [1, A_1]$, $\text{MCEN}(A) < 1$, $\text{ACC}^*(A) < 1$, $\text{MCC}^*(A) < 1$ and $\text{MCEN}(A) < \text{CEN}(A)$ for all $A > 0$,

Table 6. Examples in the binary case for family U_A .

	$\begin{pmatrix} 10^3 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 10^2 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 10 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 10 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 10^2 \\ 10^2 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 10^3 \\ 10^3 & 0 \end{pmatrix}$
A =	1/10 ³	1/10 ²	1/10	1	10	10 ²	10 ³
ACC* =	0.00200	0.01961	0.16667	0.66667	0.952381	0.995025	0.9995002
MCC* =	0.50050	0.50495	0.54545	0.75000	0.954545	0.995050	0.9995005
CEN =	0.01194	0.08488	0.45495	1.00000	1.017859	1.002167	1.0002210
MCEN =	0.01459	0.09964	0.48263	0.93999	0.997778	0.9998483	0.9999856

<https://doi.org/10.1371/journal.pone.0210264.t006>

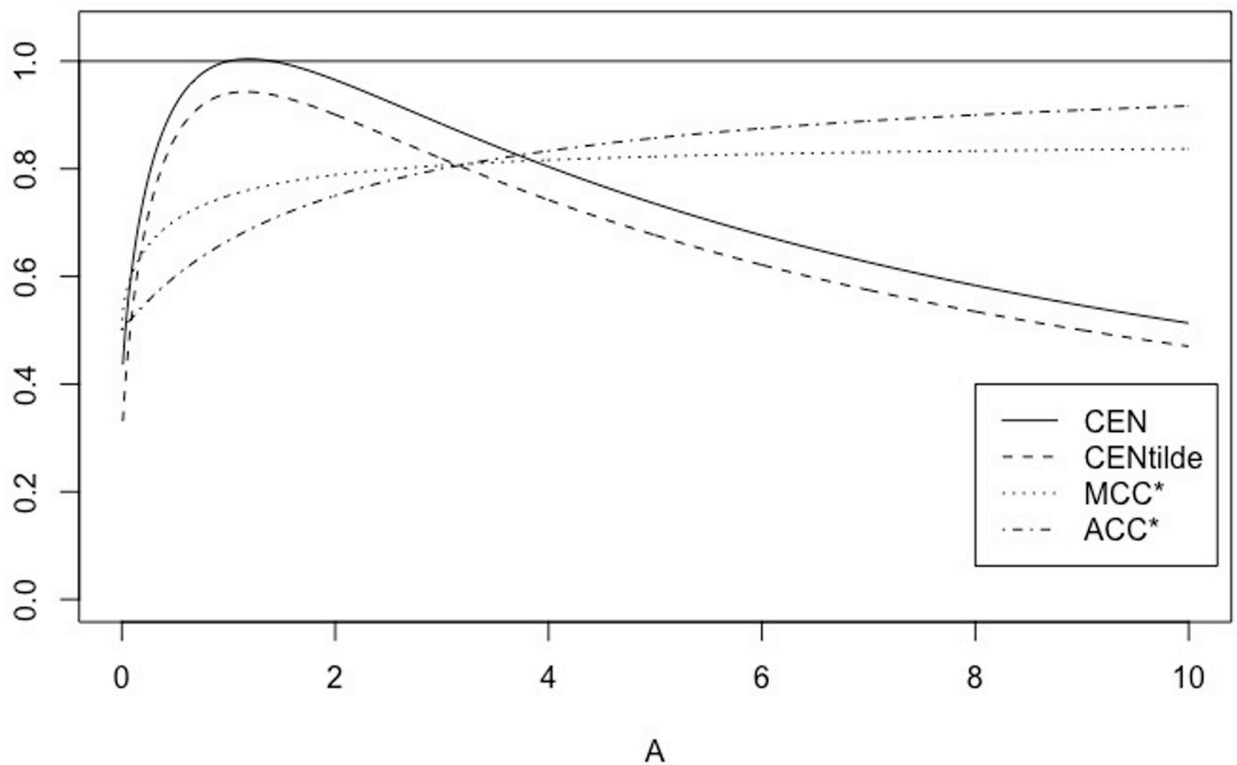


Fig 6. Family V_A . CEN, MCEN, ACC* and MCC* for $A \in (0, 10]$.

<https://doi.org/10.1371/journal.pone.0210264.g006>

$$\lim_{A \rightarrow 0} MCC^*(A) = \lim_{A \rightarrow 0} ACC^*(A) = \frac{1}{2} > \lim_{A \rightarrow 0} CEN(A) = \frac{\log_2(3)}{4} > \lim_{A \rightarrow 0} MCEN(A) = \frac{2}{7}$$

$$\lim_{A \rightarrow +\infty} ACC^*(A) = 1 > \lim_{A \rightarrow +\infty} MCC^*(A) = \frac{2 + \sqrt{2}}{4} > \lim_{A \rightarrow +\infty} CEN(A) = \lim_{A \rightarrow +\infty} MCEN(A) = 0.$$

Note that as in previous cases, CEN(A) does not stay always (that is, for any $A > 0$) restricted to $[0, 1]$, while MCEN does. See Fig 6 and some examples in Table 7.

Apart from the fact that CEN is out-of-range for some values of A, its behaviour is similar to that of MCEN, both decreasing with entropy, while nor ACC* nor MCC* are sensitive to the decrease of entropy when $A \rightarrow +\infty$.

Table 7. Examples in the binary case for family V_A .

	$\begin{pmatrix} 10^3 & 1 \\ 10^3 & 0 \end{pmatrix}$	$\begin{pmatrix} 10^2 & 1 \\ 10^2 & 0 \end{pmatrix}$	$\begin{pmatrix} 10 & 1 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 5 & 6 \\ 5 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 10 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 10^2 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 10^3 \\ 1 & 0 \end{pmatrix}$
A =	$1/10^3$	$1/10^2$	$1/10$	1	1.2	10	10^2	10^3
ACC* =	0.5002	0.5025	0.5238	0.6667	0.6875	0.9167	0.9902	0.9990
MCC* =	0.5112	0.5352	0.6066	0.7500	0.7611	0.8371	0.8518	0.8535
CEN =	0.4019	0.4361	0.6217	1.0000	1.0041	0.5133	0.0934	0.0128
MCEN =	0.2921	0.3309	0.5387	0.9400	0.9429	0.4702	0.0866	0.0121

<https://doi.org/10.1371/journal.pone.0210264.t007>

The symmetric but unbalanced family $X_{A,r}$. Now we introduce the family of confusion matrices $X_{A,r} = \begin{pmatrix} A & rA \\ rA & 1 \end{pmatrix}$, with $A, r > 0$. Both class-1 and class-2 cases are mainly misclassified if $A, r > 1$. Overall entropy of $X_{A,r}$ is $-\frac{A}{(2r+1)A+1} \log\left(\frac{A}{(2r+1)A+1}\right) - \frac{2rA}{(2r+1)A+1} \log\left(\frac{rA}{(2r+1)A+1}\right)$, which drops to 0 when $A \rightarrow 0$, and when $A \rightarrow +\infty$ converges to $\log(2r+1) - \frac{2r}{2r+1} \log(r)$, which in turn converges to 1 as $r \rightarrow +\infty$. Fixed $A > 0$, overall entropy converges to 1 as $r \rightarrow +\infty$, and as $r \rightarrow 0$, it converges to $-\frac{A}{A+1} \log\left(\frac{A}{A+1}\right)$, which in turn converges to 0 both when $A \rightarrow 0$ and when $A \rightarrow +\infty$.

When $0 < A < 1, A = 1/B$ with $B > 1$, matrix $X_{A,r}$ is equivalent to $\begin{pmatrix} 1 & r \\ r & B \end{pmatrix}$. We have some properties of CEN, MCEN, ACC* and MCC* in Proposition 5 below. Moreover, for $r = 0.5$, 5 Figs 7 and 8 show how the measures evolve as function of A , while Figs 9 and 10 show their plots as function of r , fixed $A = 0.5, 10$.

Proposition 5 For confusion matrix $X_{A,r}$ with $A, r > 0$ we have:

$$\begin{aligned} \text{CEN}(A) &= -\frac{rA}{(2r+1)A+1} \log_2\left(\frac{r^2A}{4(r+1)(rA+1)}\right), \\ \text{MCEN}(A) &= -\frac{4rA}{(8r+3)A+3} \log_2\left(\frac{r^2A}{(2r+1)(2rA+1)}\right), \\ \text{ACC}^*(A) &= \frac{2rA}{(2r+1)A+1}, \quad \text{MCC}^*(A) = \frac{2r^2A+rA+r}{2(r+1)(rA+1)}. \end{aligned}$$

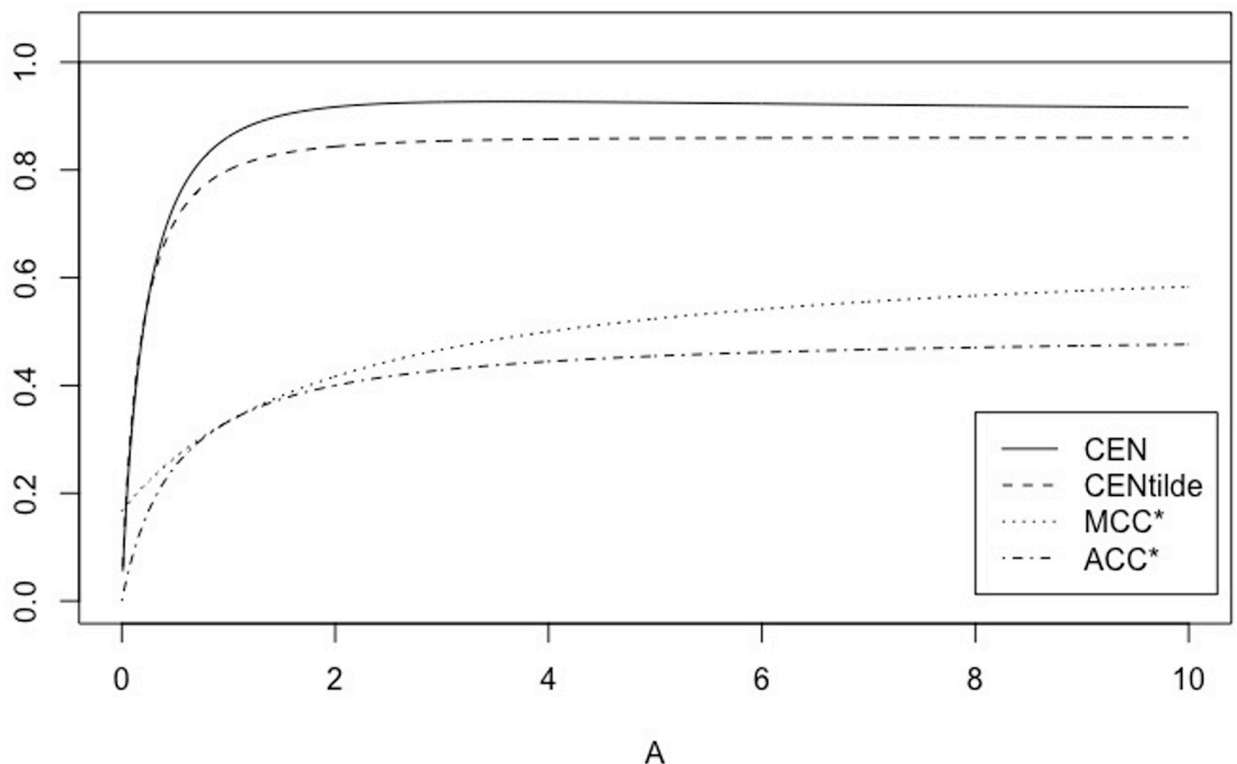


Fig 7. Family $X_{A,r}$ CEN, MCEN, ACC* and MCC* as function of $A > 0$ for $r = 0.5$.

<https://doi.org/10.1371/journal.pone.0210264.g007>

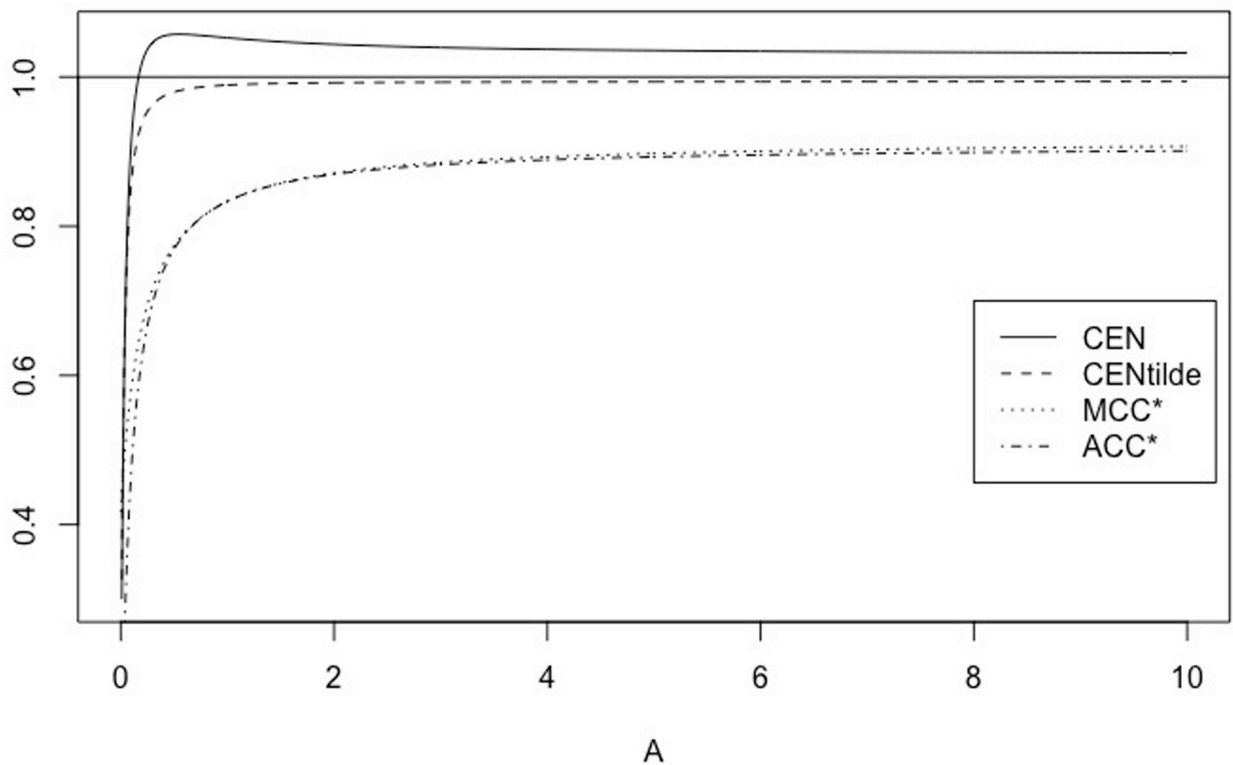


Fig 8. Family $X_{A,r}$, CEN, MCEN, ACC* and MCC* as function of $A > 0$ for $r = 5$.

<https://doi.org/10.1371/journal.pone.0210264.g008>

As a consequence, $\ell_{CEN}(r) = \lim_{A \rightarrow +\infty} CEN(A) = \frac{r}{2r+1} \log_2\left(\frac{4(r+1)}{r}\right) > 0$, and there exists $r_0 < 1$ ($r_0 \approx 0.8$) such that for any $r > r_0$, there exists $A_r > 0$ such that $CEN(A) < 1$ if $A < A_r$, $CEN(A_r) = 1$, $CEN(A) > 1$ if $A > A_r$ and

$$\ell_{CEN}(r) \begin{cases} > 1 & \text{if } r > 1, \\ = 1 & \text{if } r = 1, \\ < 1 & \text{if } r_0 < r < 1. \end{cases}$$

If $r \leq r_0$, $CEN(A) \leq 1$ for any $A > 0$ and $\ell_{CEN}(r) < 1$.

On the other hand, for any $r > 0$,

$MCEN(A) < 1$, $ACC^*(A) < 1$ and $MCC^*(A) < 1$, for all $A > 0$, $MCEN$, ACC^* and MCC^* are monotonically increasing functions of A , CEN is not, and has a global maximum, which is > 1 if $r > r_0$, $\lim_{A \rightarrow 0} CEN(A) = \lim_{A \rightarrow 0} MCEN(A) = \lim_{A \rightarrow 0} ACC^*(A) = 0$, $\lim_{A \rightarrow 0} MCC^*(A) = \frac{r}{2(r+1)}$,

$$0 < \lim_{A \rightarrow +\infty} ACC^*(A) = \frac{2r}{2r+1} < \lim_{A \rightarrow +\infty} MCC^* = \frac{2r+1}{2(r+1)} = \ell_{MCC^*}(r) < 1,$$

$$0 < \lim_{A \rightarrow +\infty} MCEN(A) = \frac{4r}{8r+3} \log_2\left(\frac{2(2r+1)}{r}\right) = \ell_{MCEN}(r) < 1, \lim_{r \rightarrow +\infty} \ell_{MCEN}(r) = 1.$$

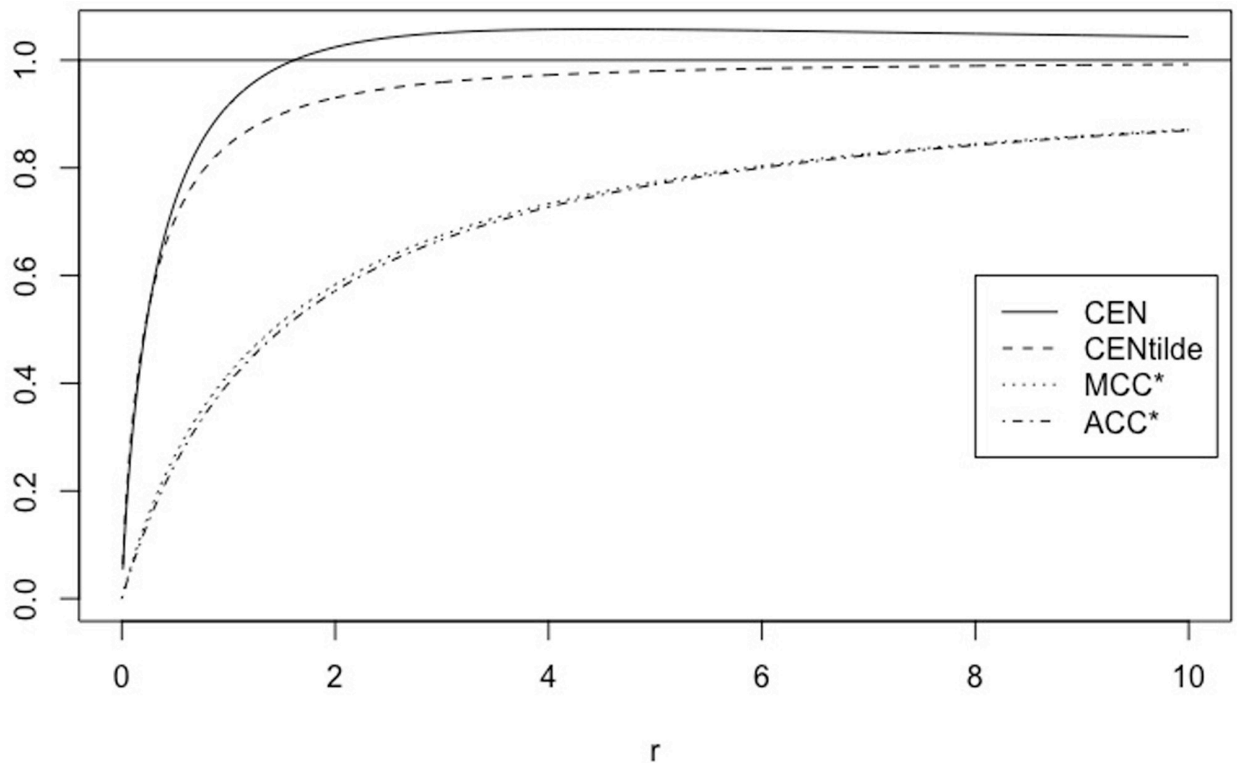


Fig 9. Family $X_{A,r}$. CEN, MCEN, ACC* and MCC* as function of $r > 0$ for $A = 0.5$.

<https://doi.org/10.1371/journal.pone.0210264.g009>

Moreover, there exist $0 < r_3 < r_2 < r_1 < r_0 < 1$ ($r_3 \approx 0.13, r_2 \approx 0.15, r_1 \approx 0.23$) such that:

$$\left\{ \begin{array}{ll} \ell_{MCC}(r) > \ell_{MCEN}(r) > \ell_{CEN}(r) & \text{if } 0 < r < r_3, \\ \ell_{MCC}(r) = \ell_{MCEN}(r) > \ell_{CEN}(r) & \text{if } r = r_3, \\ \ell_{MCEN}(r) > \ell_{MCC}(r) > \ell_{CEN}(r) & \text{if } r_3 < r < r_2, \\ \ell_{MCEN}(r) > \ell_{MCC}(r) = \ell_{CEN}(r) & \text{if } r = r_2, \\ \ell_{MCEN}(r) > \ell_{CEN}(r) > \ell_{MCC}(r) & \text{if } r_2 < r < r_1, \\ \ell_{MCEN}(r) = \ell_{CEN}(r) > \ell_{MCC}(r) & \text{if } r = r_1, \\ \ell_{CEN}(r) > \ell_{MCEN}(r) > \ell_{MCC}(r) & \text{if } r > r_1. \end{array} \right.$$

Finally, for any fixed, $A > 0$, while MCEN, ACC* and MCC* are monotonically increasing functions of r , CEN is not, as can be seen in Figs 9 and 10, for two values of A . Given $A > 0$, there exists $r_A > r_0$ such that $CEN(A) > 1$ for all $r > r_A$.

Note that although we do not specify it in the notations so as not to complicate them, the performance measures depend on both A and r in the case of this doubly indexed family $X_{A,r}$.

The asymmetric family $Y_{A,r}$. Finally, we consider another particular doubly indexed family of confusion matrices in the binary case, with the same overall entropy as $X_{A,r}$

denoted by $Y_{A,r}$ with $A, r > 0$. We define this family by $Y_{A,r} = \begin{pmatrix} rA & rA \\ A & 1 \end{pmatrix}$. Class-2 is

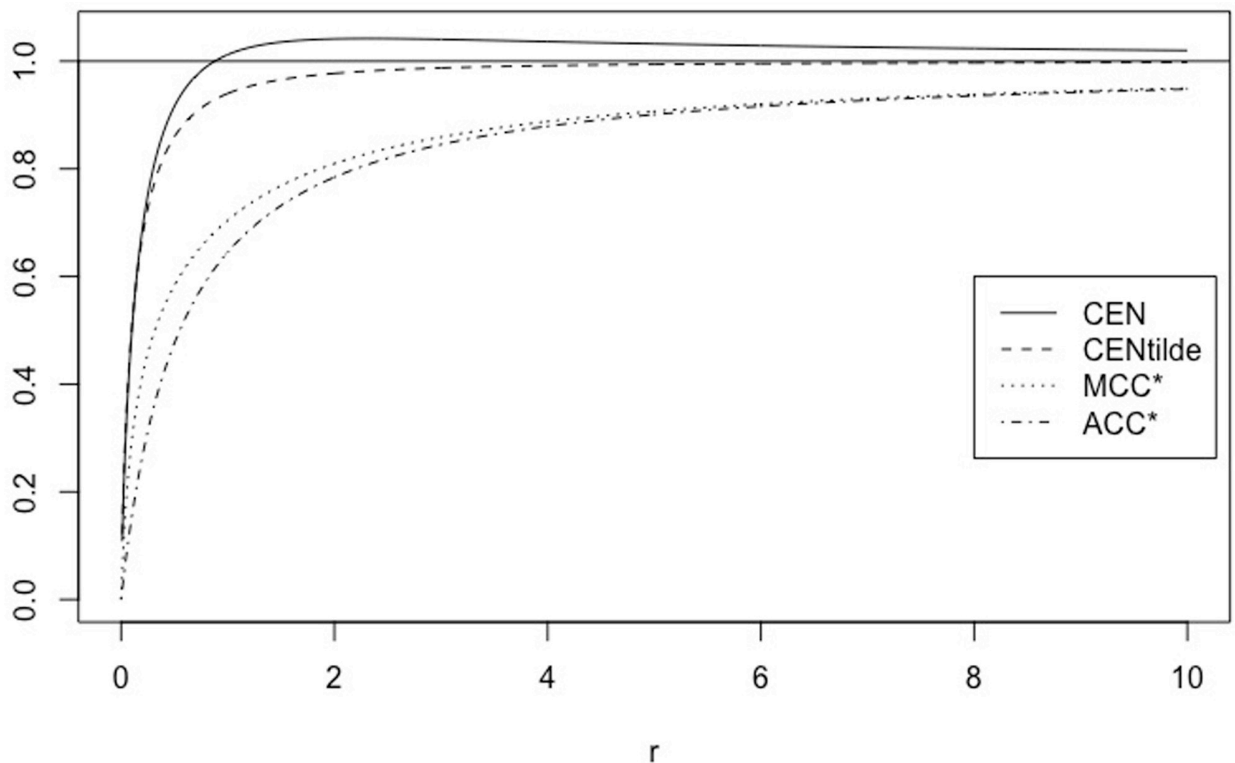


Fig 10. Family $X_{A,r}$ CEN, MCEN, ACC* and MCC* as function of $r > 0$ for $A = 10$.

<https://doi.org/10.1371/journal.pone.0210264.g010>

underrepresented and mainly misclassified if $A, r > 1$, while class-1 cases are classified “at random”, that is, a class-1 case has the same probability to be classified into any of the two classes. Although entropy is as for $X_{A,r}$ we will see that performance measures behave in a different way for this family of confusion matrices. When $0 < A < 1, A = 1/B$ with $B > 1$, then matrix

$Y_{A,r}$ is equivalent to $\begin{pmatrix} r & r \\ 1 & B \end{pmatrix}$. In Proposition 6 we give some properties of CEN, MCEN, ACC* and MCC*. See in Fig 11 for $r = 0.1$, in Fig 12 for $r = 0.8$, and see Fig 13 for a plot of them as function of r , fixed $A = 10$.

Proposition 6 For confusion matrix $Y_{A,r}$ with $A, r > 0$ we have:

$$\begin{aligned} \text{CEN}(A) &= \frac{(r+1)A \log_2(((r+1)A+2)(3r+1)) + (r-1)A \log_2(A) - 2rA \log_2(rA)}{2((2r+1)A+1)}, \\ \text{MCEN}(A) &= \frac{2((r+1)A \log_2(((r+1)A+1)(2r+1)) + (r-1)A \log_2(A) - 2rA \log_2(rA))}{3((2r+1)A+1) + (r+1)A}, \\ \text{ACC}^*(A) &= \frac{(r+1)A}{(2r+1)A+1}, \quad \text{MCC}^*(A) = \frac{1 - \frac{r^{(1-A)}}{\sqrt{2r^{(A+1)}(r+1)^{(rA+1)}}}}{2}. \end{aligned}$$

As a consequence, $L_{\text{CEN}}(r) = \lim_{A \rightarrow +\infty} \text{CEN}(A) = \frac{1}{2(2r+1)} \log_2\left(\frac{((3r+1)(r+1))^{r+1}}{r^{2r}}\right) > 0$, and there

exists $R_0 < 1 (R_0 \approx 0.71)$ such that $L_{\text{CEN}}(r) \begin{cases} > 1 & \text{if } R_0 < r < 1, \\ = 1 & \text{if } r = R_0, 1, \\ < 1 & \text{if } r < R_0 \text{ or } r > 1. \end{cases}$ Moreover, there exist 0

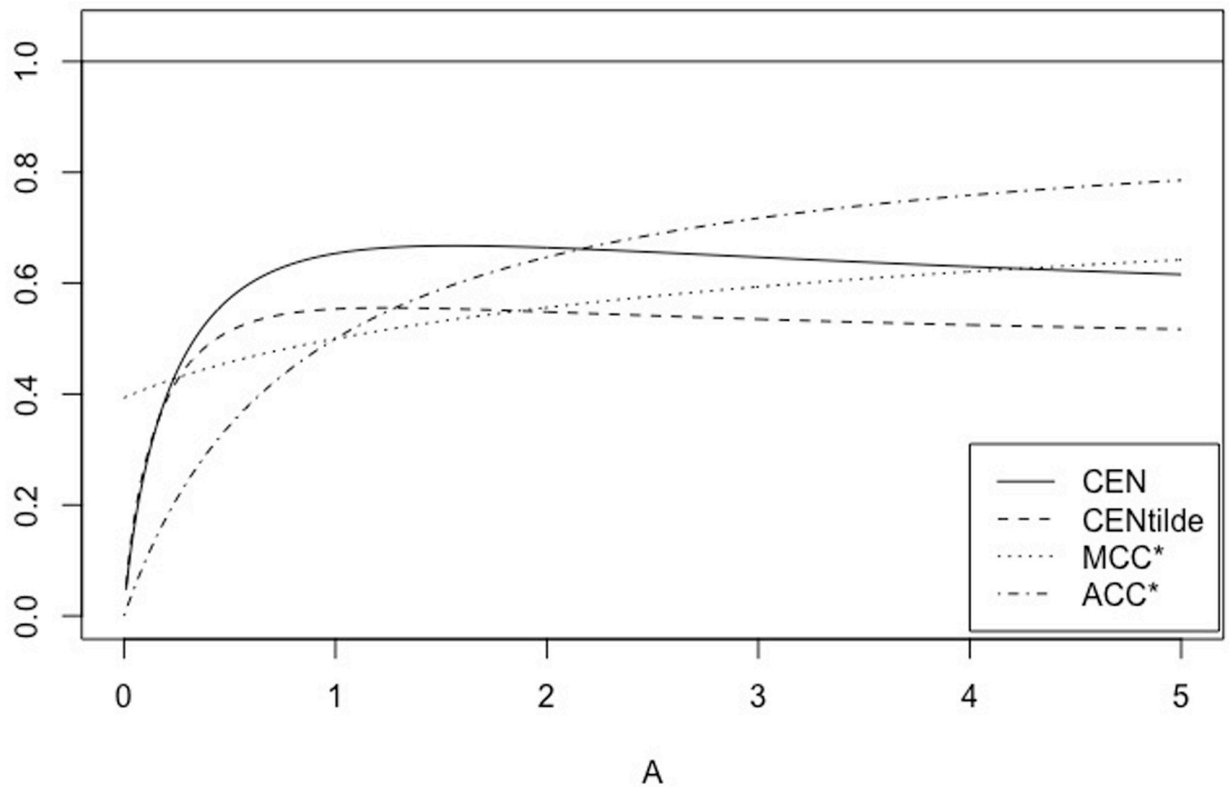


Fig 11. Family $Y_{A,r}$, CEN, MCEN, ACC* and MCC* as function of $A > 0$ for $r = 0.1$.

<https://doi.org/10.1371/journal.pone.0210264.g011>

$< R_1 < R_0 < 1 < R_2 (R_1 \approx 0.5, R_2 \approx 1.4)$ such that

$$\left\{ \begin{array}{ll} \text{if } r \in [R_0, 1], & \text{there exists } A_r > 0 \text{ such that } \text{CEN}(A) < 1 \text{ if } A < A_r, \\ & \text{CEN}(A_r) = 1, \text{ CEN}(A) > 1 \text{ if } A > A_r, \\ \text{if } r \in (R_1, R_0) \cup (1, R_2), & \text{there exist } 0 < A_r < B_r \text{ such that } \text{CEN}(A) < 1 \text{ if } A < A_r, \\ & \text{or } A > B_r, \text{ CEN}(A_r) = \text{CEN}(B_r) = 1, \\ & \text{CEN}(A) > 1 \text{ if } A \in (A_r, B_r), \\ \text{if } r \notin (R_1, R_2), & \text{CEN}(A) \leq 1 \text{ for any } A > 0. \end{array} \right.$$

On the other hand, for any $r > 0$,

$\text{MCEN}(A) < 1, \text{ACC}^*(A) < 1$ and $\text{MCC}^*(A) < 1$, for all $A > 0$, ACC* and MCC* are monotonically increasing functions of A, CEN is not, and MCEN is or not, depending on the value of r,

$$\begin{aligned} \lim_{A \rightarrow 0} \text{CEN}(A) = \lim_{A \rightarrow 0} \text{MCEN}(A) = \lim_{A \rightarrow 0} \text{ACC}^*(A) = 0, \lim_{A \rightarrow 0} \text{MCC}^*(A) &= \frac{1 - \sqrt{\frac{r}{2(r+1)}}}{2}, \\ \lim_{A \rightarrow +\infty} \text{ACC}^*(A) = \frac{r+1}{2r+1} = L_{\text{ACC}^*}(r), \lim_{A \rightarrow +\infty} \text{MCC}^* &= \frac{1 + \frac{1}{\sqrt{2(r+1)}}}{2} = L_{\text{MCC}^*}(r), L_{\text{MCEN}}(r) = \\ \lim_{A \rightarrow +\infty} \text{MCEN}(A) = \frac{2}{3(2r+1) + (r+1)} \log_2 \left(\frac{((2r+1)(r+1))^{r+1}}{r^{2r}} \right) < 1, L_{\text{MCEN}}(r) < L_{\text{CEN}}(r) & \text{for all } r > 0. \end{aligned}$$

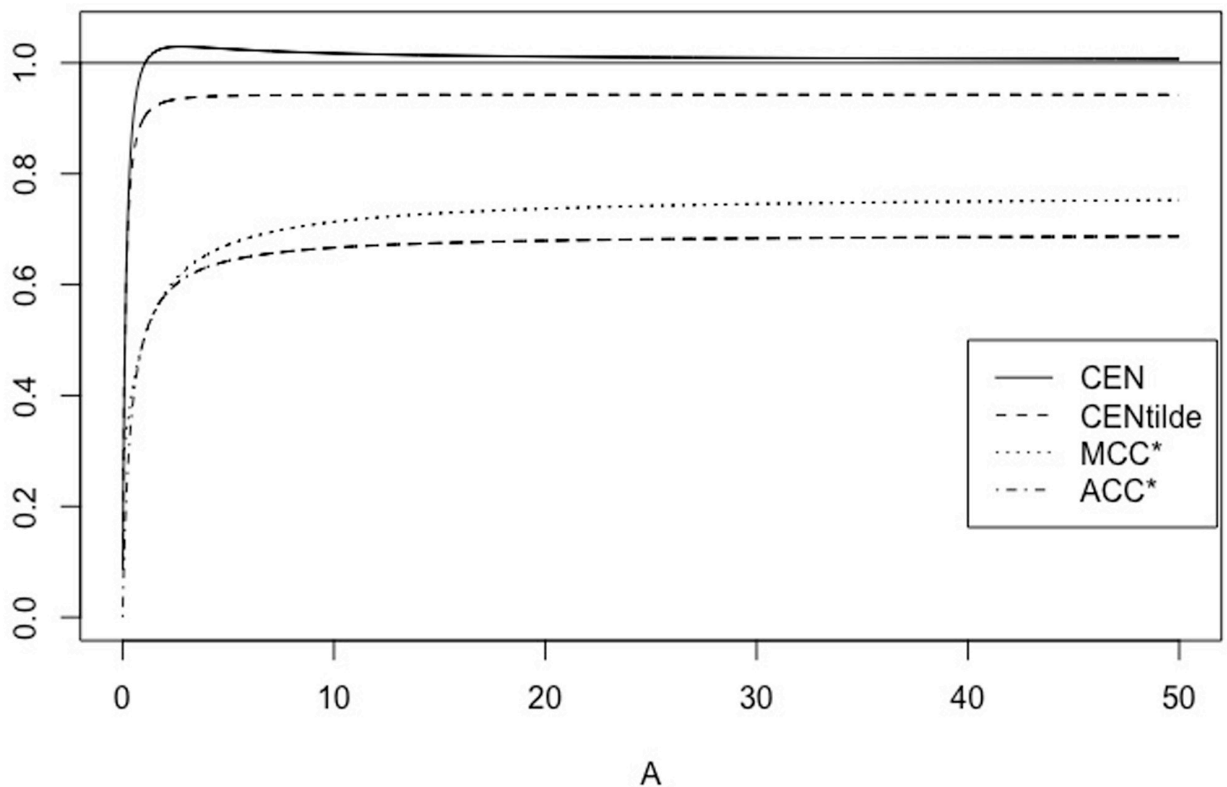


Fig 12. Family $Y_{A,r}$, CEN, MCEN, ACC* and MCC* as function of $A > 0$ for $r = 0.8$.

<https://doi.org/10.1371/journal.pone.0210264.g012>

Note that $L_{ACC^*}(r) < L_{MCC^*}(r)$ if and only if $r > \frac{-1+\sqrt{5}}{4} > 0$.

Improving classification of the minority class while maintaining the imbalance between the classes. Up to now, we have evaluated binary confusion matrices with different balances of the two classes but not different classification results. Now let's do just the opposite. To help clarify the utility of MCEN in the evaluation of improvements in classification of the minority class while maintaining the same amount of imbalance, we consider two different examples.

Example 1: We introduce the family of confusion matrices $X_{50,2}^\alpha = \begin{pmatrix} 50 & 100 \\ 101 - \alpha & \alpha \end{pmatrix}$,

with $\alpha = 1, 2, \dots, 101$. Note that when $\alpha = 1$, the corresponding matrix belongs to the family $\{X_{A,r}\}$ with $A = 50$ and $r = 2$. Imbalance in classes stays fix. When $\alpha = 1$, the minority class is classified very badly, improving classification as α increases and reaching the perfect classification when $\alpha = 101$. Is MCEN able to detect this behaviour? Yes, it is. Unlike what happens with CEN, MCEN (as well as ACC* and MCC*) monotonically decreases when classification of the minority class improves (α increases). CEN incongruously first increases up to $\alpha = 18$ and then starts to decrease and behave like the other performance measures (see Fig 14).

Example 2: A similar phenomenon can be observed with family $Y_{100,1}^\beta = \begin{pmatrix} 100 & 100 \\ 101 - \beta & \beta \end{pmatrix}$,

with $\beta = 1, 2, \dots, 101$ (with $\beta = 1$ the corresponding matrix belongs to the family $\{Y_{A,r}\}$ with $A = 100$ and $r = 1$). As in Example 1, imbalance in classes is constant and when $\beta = 1$, the

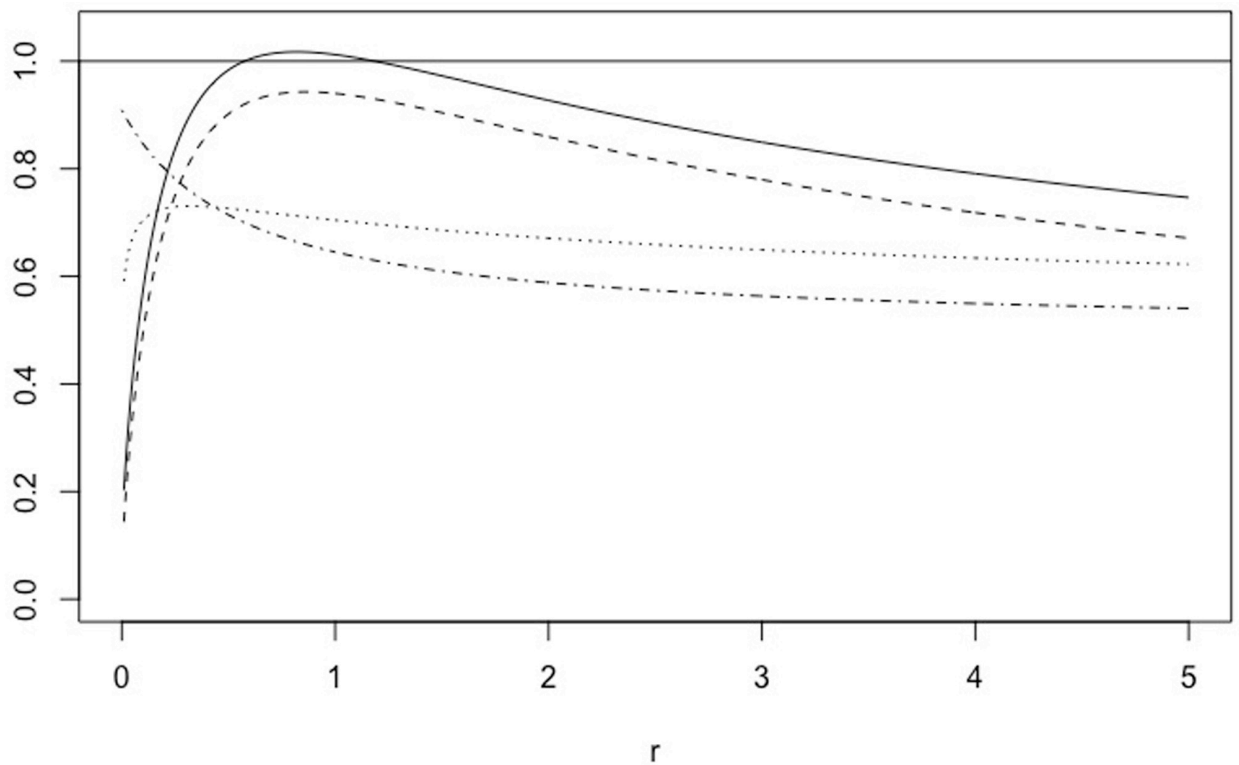


Fig 13. Family $Y_{A,r}$. CEN, MCEN, ACC* and MCC* as function of r for $A = 10$.

<https://doi.org/10.1371/journal.pone.0210264.g013>

minority class is classified very badly, improving classification as β increases up to 101, when perfect classification is reached. MCEN as well as ACC* and MCC*, monotonically decrease when β increases, while CEN increases up to $\beta = 14$ and then starts to decrease and behave like the other performance measures (see Fig 15).

The Z_A family

As noted in [4], the behaviour of the Confusion Entropy CEN is rather diverse from that of MCC* and ACC* for the pathological case of the family of confusion matrices

$Z_A = (a_{ij})_{i,j=1,\dots,N}$, defined by $a_{ij} = \begin{cases} A & \text{if } i = N, j = 1 \\ 1 & \text{otherwise,} \end{cases}$ with $A > 0$. That is,

$$Z_A = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots \\ 1 & 1 & \dots & 1 \\ A & 1 & \dots & 1 \end{pmatrix}. \text{ We want to study how MCEN behaves when applied to elements of}$$

this family.

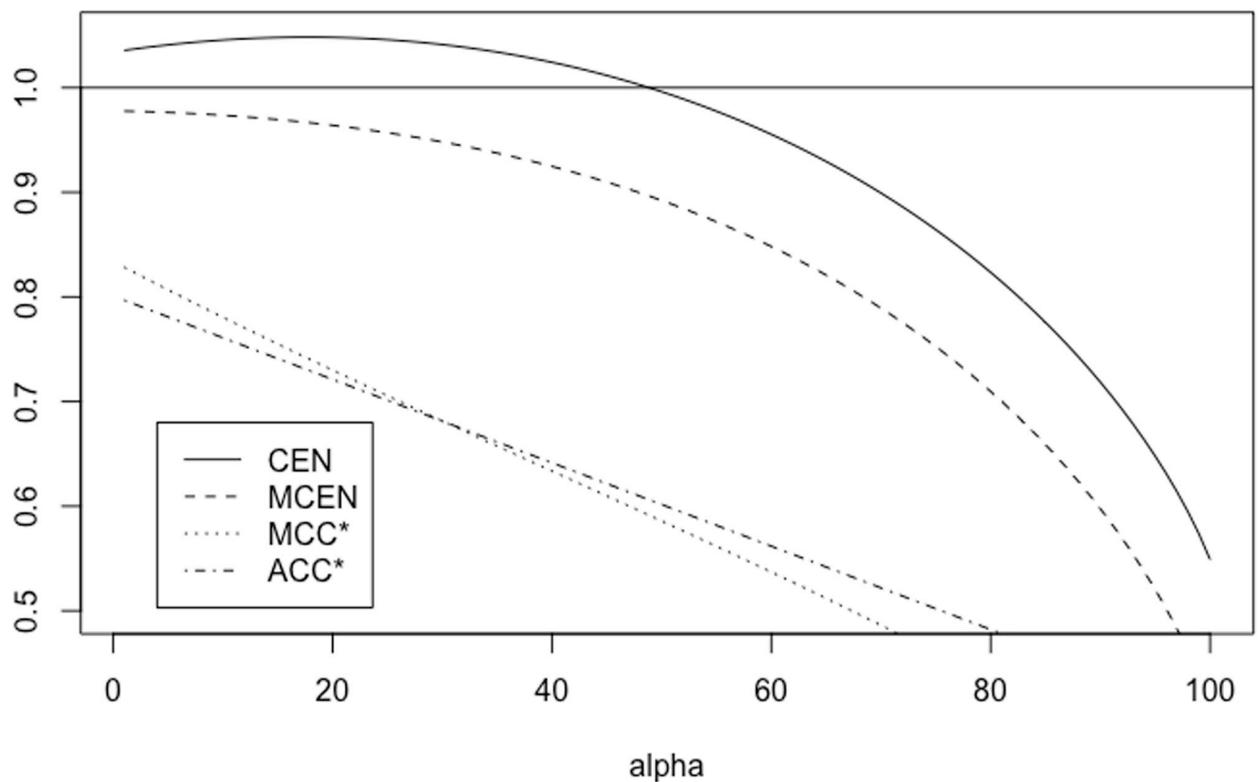


Fig 14. Family $X_{50,2}^{\alpha}$ with $\alpha = 1, 2, \dots, 101$. CEN, MCEN, ACC* and MCC* as function of α .

<https://doi.org/10.1371/journal.pone.0210264.g014>

Proposition 7

$$\text{If } N > 2, \text{CEN}(Z_A) = \frac{1}{N^2 + A - 1} \left((N - 1)(N - 2) \log_{2(N-1)}(2N) + (2N + A - 3) \log_{2(N-1)}(2N + A - 1) - A \log_{2(N-1)}(A) \right),$$

$$\text{MCEN} = \frac{2}{2(N^2 + A - 1) - N} \left((N - 1)(N - 2) \log_{2(N-1)}(2N - 1) + (2N + A - 3) \log_{2(N-1)}(2N + A - 2) - A \log_{2(N-1)}(A) \right),$$

$$\text{if } N = 2, \text{CEN}(Z_A) = \frac{1}{A + 3} \left((A + 1) \log_2(A + 3) - A \log_2(A) \right),$$

$$\text{MCEN} = \frac{2}{2A + 5} \left((A + 1) \log_2(A + 2) - A \log_2(A) \right).$$

In general ($N \geq 2$),

$$\text{MCC}^*(Z_A) = \frac{N(N^2 + 2(A - 1)) - (N^2 + (A - 1))}{2(N - 1)(N^2 + 2(A - 1))},$$

$$\text{ACC}^*(Z_A) = \frac{N^2 - N + (A - 1)}{N^2 + (A - 1)}$$

As a consequence,

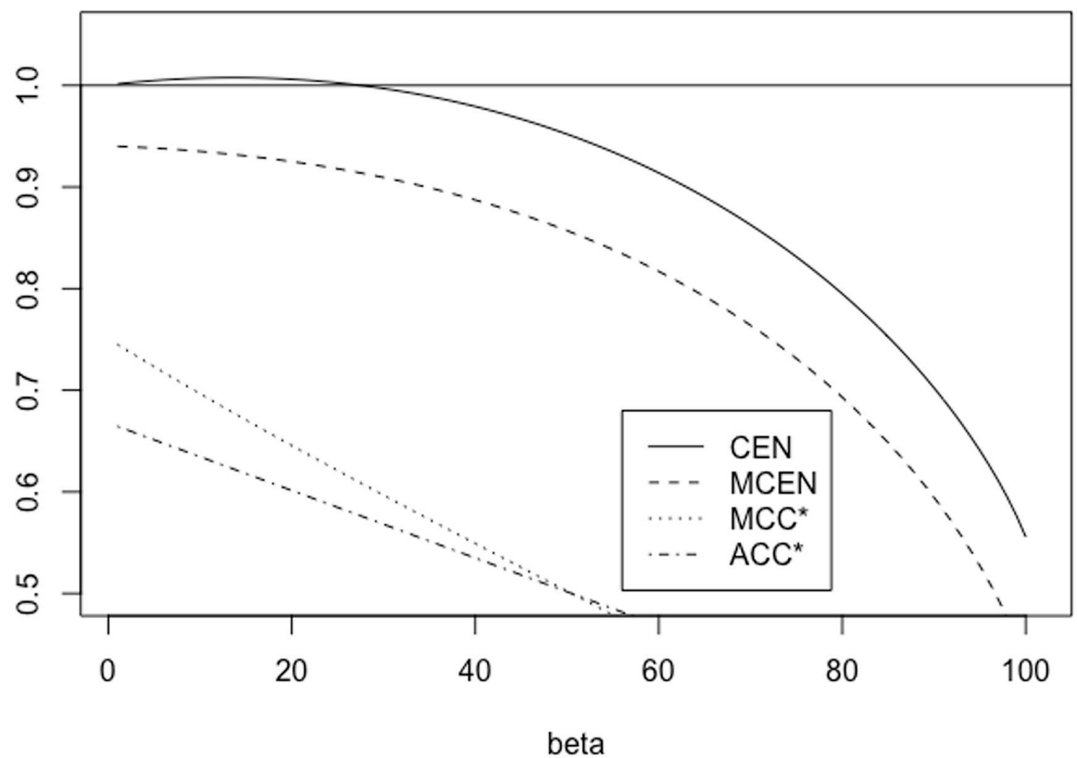


Fig 15. Family $Y_{50,2}^\beta$ with $\beta = 1, 2, \dots, 101$. CEN, MCEN, ACC* and MCC* as function of β .

<https://doi.org/10.1371/journal.pone.0210264.g015>

- If $N = 2$,
 $MCEN < CEN(Z_A)$ for all $A > 0$,
 $MCEN < 1$ for all $A > 0$, and there exists $A_3 \in (1, 2)$ ($A_3 \approx 1.85$) such that

$$CEN(Z_1) = CEN(Z_{A_3}) = 1,$$

$$CEN(Z_A) > 1 \text{ if } A \in (1, A_3) \text{ and } CEN(Z_A) < 1 \text{ if } A \notin [1, A_3],$$

$$\lim_{A \rightarrow 0} MCC^*(A) = \frac{1}{4} < \lim_{A \rightarrow 0} ACC^* = \frac{1}{3} < \lim_{A \rightarrow 0} MCEN(A) = \frac{2}{5} < \lim_{A \rightarrow 0} CEN(A) = \frac{\log_2(3)}{3},$$

$$\lim_{A \rightarrow +\infty} CEN(A) = \lim_{A \rightarrow +\infty} MCEN(A) = 0 < \lim_{A \rightarrow +\infty} MCC^* = \frac{3}{4} < \lim_{A \rightarrow +\infty} ACC^* = 1.$$

- If $N = 3$ (we take this case as example of what happens with $N > 2$),

$$\lim_{A \rightarrow 0} MCC^*(A) = \frac{13}{28} < \lim_{A \rightarrow 0} ACC^* = \frac{5}{8} <$$

$$< \lim_{A \rightarrow 0} CEN(A) = \frac{2 \log_4(6) + 3 \log_4(5)}{8} < \lim_{A \rightarrow 0} MCEN(A) = \frac{2}{13} (2 \log_4(5) + 3) < 1,$$

$$\lim_{A \rightarrow +\infty} CEN(A) = \lim_{A \rightarrow +\infty} MCEN(A) = 0 < \lim_{A \rightarrow +\infty} MCC^* = \frac{5}{8} < \lim_{A \rightarrow +\infty} ACC^* = 1.$$

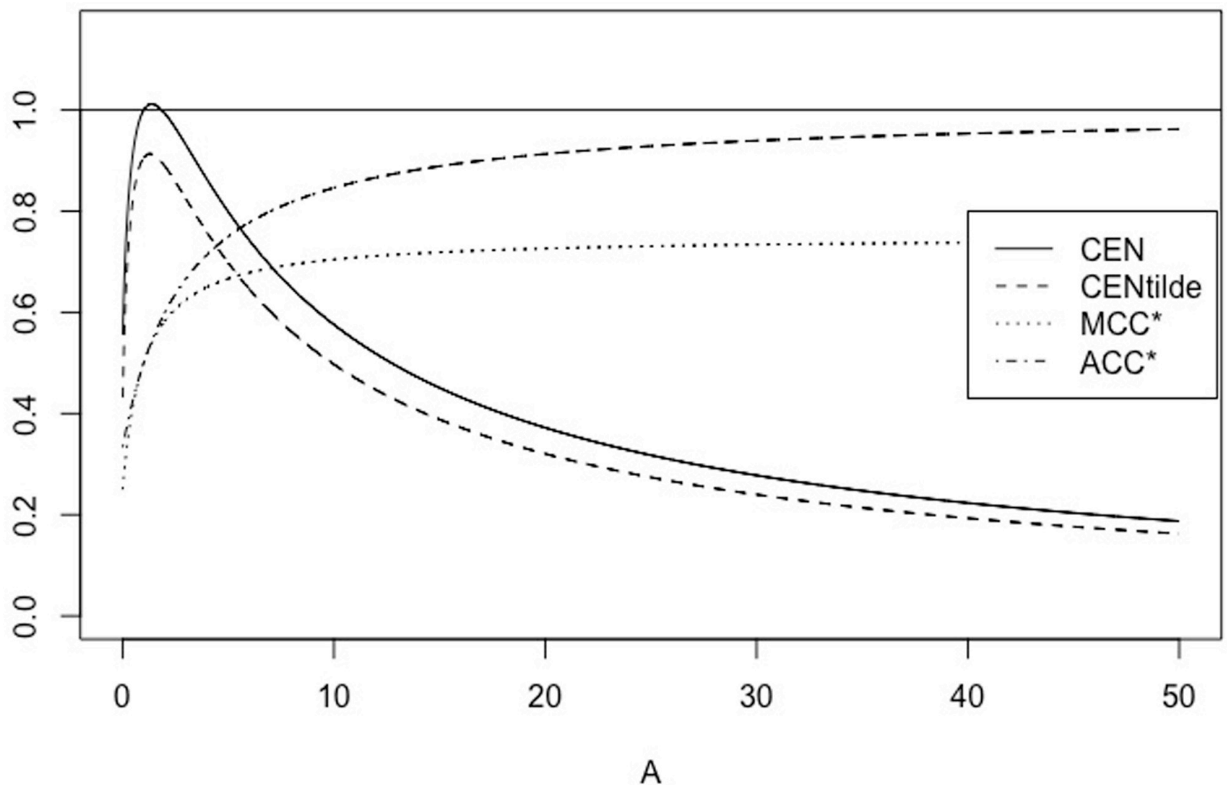


Fig 16. Family Z_A . CEN, MCEN, MCC* and ACC* as function of $A > 0$ for $N = 2$.

<https://doi.org/10.1371/journal.pone.0210264.g016>

In Figs 16 and 17 we can observe this behaviour when $N = 2$ and $N = 3$, respectively.

Table 8 shows some examples of confusion matrices of the family Z_A , first with $N = 2$, and secondly with $N = 4$.

Note that CEN and MCEN exhibit a very different behaviour comparing with ACC* and MCC*, since the former are sensitive to the overall entropy associated to the elements of the matrix, which is $\log(N^2 + A - 1) - \frac{A}{N^2 + A - 1} \log(A)$. Entropy decreases to $\log(N^2 - 1)$ when $A \rightarrow 0$, and drops to 0 when $A \rightarrow +\infty$.

Comparing with other performance measures

Several works have considered the question of the introduction and comparison of different performance measures for classification, inspired, in one way or another, by Shannon's entropy. For example, in [13] the authors introduce a novel measure called PACC (Probabilistic Accuracy) in the multi-class setting, making a comparative study of it with other measures as Accuracy, MCC and CEN, among others.

Besides, Entropy-Modulated Accuracy (EMA), introduced in [14], is a performance measure of classification tasks based on the concept of *perplexity*, the latter being defined as the effective number of classes a classifier sees. The authors also introduce NIT (Normalized Information Transfer) factor, which is a correction of EMA. They compare both EMA and NIT factor with Accuracy and CEN, rejecting rankings of classifiers based in Accuracy and choosing more meaningful and interpretable classifiers. They show in some examples that MCC is

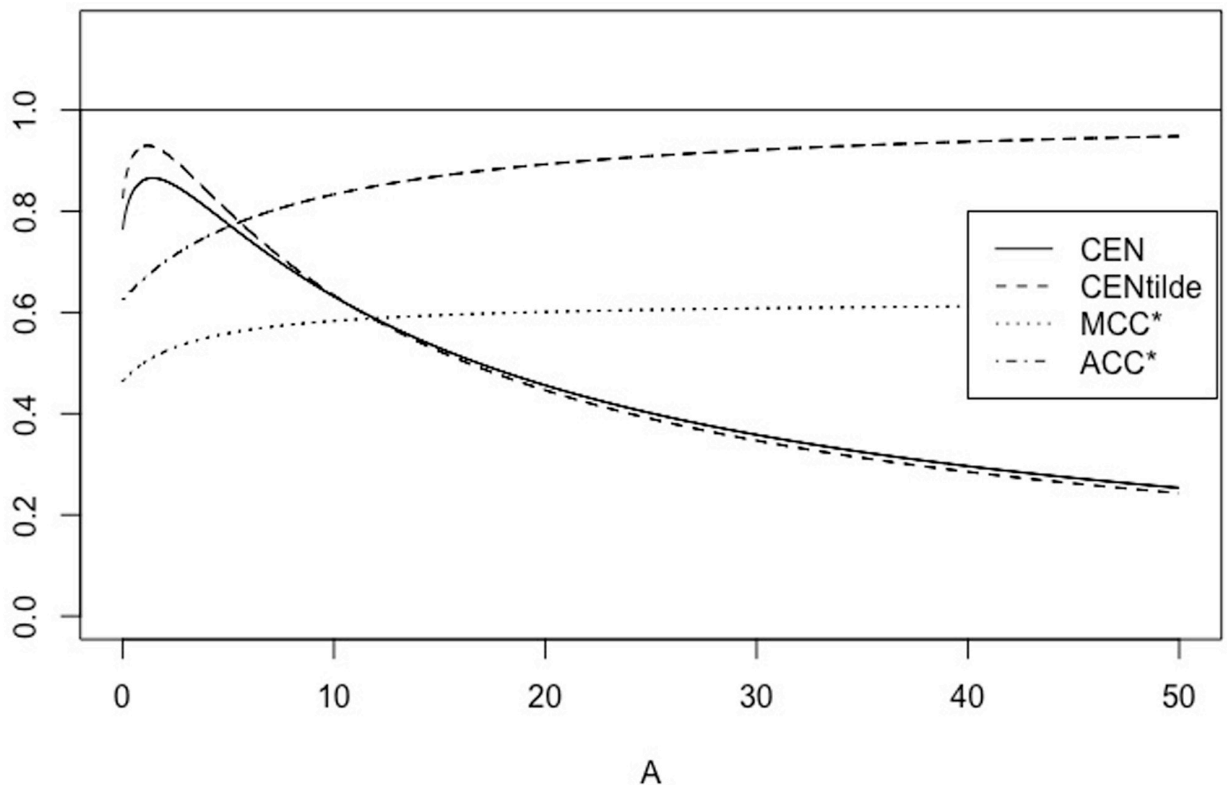


Fig 17. Family Z_A . CEN, MCEN, MCC* and ACC* as function of $A > 0$ for $N = 3$.

<https://doi.org/10.1371/journal.pone.0210264.g017>

highly correlated with Accuracy, while rankings obtained with CEN, EMA and NIT factor are comparable in some cases but disagree in others.

Although PACC, EMA and NIT factor are useful measures to assess classifiers, in our opinion none of them is completely satisfactory in grading the effectiveness of the classifier learning process, since all reflect some concrete feature of the classification process, being insufficient for covering all the aspects of this complex task, so they should be used cautiously and in a complementary way. That is, all the measures suffer from certain weaknesses that are evident in specific, more or less gimmicky situations. This comment extends also to both CEN and MCEN, although it should be noted that the latter solves the problems showed by CEN in the binary setting, as well as to MCC and Accuracy, the last one having been widely treated (see, for example, the Introduction section in [14]).

Let us exemplify this fact by going back to the toy example in Table 2. In Table 9 we add the calculated values of $PACC^* = 1 - PACC$ and $1/NIT$ to that of Table 2. We use NIT factor (inverted to make it comparable with the other measures) instead of EMA since the probability distribution of classes in the validation set is not uniform. Note that our confusion matrices are transposed with respect to that in [14], and also that for the NIT factor we use formula (4). We have used the corrected definition provided by the authors, which had already acknowledged an erratum in Eq (4) in the comments of https://www.researchgate.net/publication/259743406_100_Classification_Accuracy_Considered_Harmful_The_Normalized_Information_Transfer_Factor_Explains_the_Accuracy_Paradox/.

The behaviour of $PACC^*$ showed in Table 9 is consistent with that of MCC^* , increasing when IN entropy decreases (a) and decreasing when OUT decreases (b). However, the

Table 8. Examples with different matrices Z_A in cases $N = 2$ and $N = 4$.

	$\begin{pmatrix} 10 & 10 \\ 1 & 10 \end{pmatrix}$	$\begin{pmatrix} 2 & 2 \\ 1 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 10 & 1 \end{pmatrix}$
A =	1/10	1/2	1	2	10
ACC* =	0.3548	0.4286	0.5000	0.6000	0.8462
MCC* =	0.2955	0.4167	0.5000	0.5833	0.7045
CEN =	0.6864	0.9174	1.0000	0.9932	0.5758
MCEN =	0.5806	0.8276	0.9057	0.8889	0.4972
$Z_A =$	$\begin{pmatrix} 10^2 & 10^2 & 10^2 & 10^2 \\ 10^2 & 10^2 & 10^2 & 10^2 \\ 10^2 & 10^2 & 10^2 & 10^2 \\ 1 & 10^2 & 10^2 & 10^2 \end{pmatrix}$	$\begin{pmatrix} 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 \\ 1 & 10 & 10 & 10 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 10 & 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 10^2 & 1 & 1 & 1 \end{pmatrix}$
A =	10^{-2}	10^{-1}	1	10	10^2
ACC* =	0.7335	0.7351	0.7500	0.8400	0.9652
MCC* =	0.4882	0.4894	0.5000	0.5441	0.5771
CEN =	0.8284	0.8391	0.8704	0.7132	0.2068
MCEN =	0.8883	0.9001	0.9309	0.7338	0.2016

<https://doi.org/10.1371/journal.pone.0210264.t008>

behaviour of 1/NIT is consistent with that of CEN and MCEN, decreasing in both cases. Nevertheless, unlike what happens with CEN and MCEN, NIT factor does not distinguish among scenarios (a) and (b). This is because both EMA and NIT factor are invariants to permutations of the columns.

Another example is that of the *MEG mind reading challenge* organized by the PASCAL (Pattern Analysis, Statistical modeling and Computational Learning) network in [15], already considered in [14]. We restrict our comparison to the group of the four most outstanding systems, denoted C_1 (Huttunen et al.), C_2 (Santana et al.), C_3 (Jylänki et al.) and C_4 (Tu & Sun), since for them, unlike what happens with the rest, we could access to the confusion matrices in [15]. The results are in Table 10, and from them we see that the most comparable rankings are that given by the NIT factor, CEN and MCEN, showing clusters $\{C_4, C_2\}$ and $\{C_1, C_3\}$, with very small differences inside the clusters, specially the second. The authors of the report [15] were specially interested in comparison C_1 vs. C_4 , and 1/NIT factor, as well as CEN and MCEN, give the same ordering: C_4 is better (lower value) than C_1 , in concordance with interpretability given in [14].

Table 9. Toy example of Table 2 revisited, adding PACC and the NIT factor.

	Baseline	(a)			(b)		
	$\begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$	$\begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}$	$\begin{pmatrix} 1 & 3 \\ 3 & 5 \end{pmatrix}$	$\begin{pmatrix} 0 & 3 \\ 3 & 6 \end{pmatrix}$	$\begin{pmatrix} 3 & 2 \\ 4 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 1 \\ 5 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 0 \\ 6 & 3 \end{pmatrix}$
ACC* =	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
MCC* =	0.5000	0.5130	0.5625	0.6667	0.4881	0.4375	0.3333
CEN =	1.0000	0.9898	0.9575	0.8962	0.9591	0.8250	0.5000
MCEN =	0.9057	0.9006	0.8848	0.8571	0.8590	0.7057	0.3343
PACC* =	0.5000	0.5071	0.5312	0.5833	0.4929	0.4687	0.4167
1/NIT =	2.0000	1.9992	1.9840	1.8371	1.9992	1.9840	1.8371

<https://doi.org/10.1371/journal.pone.0210264.t009>

Table 10. Results for the first four systems of the MEG mind reading challenge. Confusion matrices have been obtained from [15].

System	ACC*	MCC*	CEN	MCEN	PACC*	1/NIT
C ₁	0.3201	0.2010	0.4360	0.5694	0.3230	2.5877
C ₂	0.3675	0.2286	0.4043	0.4981	0.3668	2.4715
C ₃	0.3721	0.2319	0.4483	0.5645	0.3667	2.6151
C ₄	0.3783	0.2369	0.4213	0.5279	0.3737	2.4545

<https://doi.org/10.1371/journal.pone.0210264.t010>

Table 11. Two toy examples. With $S = 30$ for $N = 2$, and with $S = 40$ for $N = 3$.

	$A = \begin{pmatrix} 10 & 0 \\ 10 & 10 \end{pmatrix}$		$B = \begin{pmatrix} 0 & 10 \\ 10 & 10 \end{pmatrix}$		$C = \begin{pmatrix} 10 & 0 & 0 \\ 10 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix}$		$D = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 10 & 10 \\ 10 & 0 & 0 \end{pmatrix}$
ACC* =	0.3333	<	0.6667		0.2500	<	0.5000
MCC* =	0.2500	<	0.7500		0.1500	<	0.3500
CEN =	0.5283	<	1.0000		0.1981	<	0.3231
MCEN =	0.4000	<	0.9400		0.2000	<	0.3333
PACC* =	0.2917	<	0.7083		0.1944	<	0.5000
1/NIT =	1.6799	=	1.6799		1.5000	=	1.5000

<https://doi.org/10.1371/journal.pone.0210264.t011>

One more example to show the variability when performance measures are compared: in Table 11 we see that the NIT factor (equivalently, EMA), unlike the other measures, is not able to distinguish between classifiers whose confusion matrices are A and B in the binary case, nor between C and D in multi-class classification.

Supporting information file: Experiments and results

The advantages of using Modified Confusion Entropy MCEN measure against CEN have been tested on different binary classifiers, constructed from four available datasets from the UCI ML Repository (<https://archive.ics.uci.edu>). From each dataset we construct and assess eight different classifiers, five of which are Bayesian networks, while the rest are other standard machine learning procedures used in supervised classification problems.

Because of the comparisons carried out previously with different examples, we have to recognize the impossibility of deciding what measure of behaviour, of the considered ones, can allow to decide in the case that the rankings of classifiers obtained with CEN and MCEN were different. We decided, then, to use OUT entropy as such a reference when there is disparity; in case of a tie, we will use IN entropy to break it. This is what we will call “the criterion of entropy”.

To compare rankings obtained from CEN and MCEN and that obtained by the criterion of entropy, we use both the Hamming distance and the degree of consistency indicator c (see [16]).

The results obtained with all the considered datasets heuristically reinforce that MCEN is more correlated with entropy than CEN. (see S1 File and Tables A-F in S1 File).

Conclusion

We introduced MCEN as a modification of the original Confusion Entropy performance measure CEN introduced in [3], both for binary and multi-class classification, proving some

properties. We compared this measure with CEN, MCC and Accuracy, showing that in the binary case, MCEN overcomes the unreliability of CEN in a twofold sense: the departure of the range where it should be (the interval $[0, 1]$), and the lack of monotonicity when the entropy increases or decreases. These features made CEN an inappropriate measure in the binary case, proving MCEN to be a good alternative, and we study different scenarios to highlight this fact. Moreover, while neither Accuracy nor MCC can distinguish among different misclassification distributions of cases in the confusion matrix, MCEN and CEN have a high level of discrimination.

First, we show that in the binary case (see Table 2), both CEN and MCEN are sensitive to the decreasing in the entropy within the main diagonal IN, and also to that outside the diagonal OUT, but while CEN is more sensitive than MCEN to IN, the opposite occurs with OUT. By contrast, ACC is insensitive as long as the sum of the diagonal and the total sum remain constant. Secondly, we consider the multi-class perfectly symmetric and balanced case in which the main diagonal elements are equal to T and the elements outside the diagonal are equal to F , which is analytically studied in detail, showing the output-of-range of CEN in the binary case when $\gamma = T/F \in (0, 1)$.

After that, we consider different particular situations in the binary setting, through the study of some families of confusion matrices. Family U_A is symmetric and unbalanced, showing the out-of-range of CEN for any $A > 1$, and in addition a lack of monotonicity that contrasts with the behaviour of the overall entropy associated to the elements of the matrix. Family V_A is asymmetric and unbalanced, and also shows the out-of-range of CEN but only for A in the interval $(1, A_1)$, where $A_1 \approx 1.4$.

Two doubly indexed families have been considered in the binary case. CEN has an anomalous behaviour for family $X_{A,r}$, which is symmetric but unbalanced, for $r > r_0$ (with $r_0 \approx 0.8$) since it is not only out-of-range from a certain value of A , but its limit when $A \rightarrow +\infty$ is > 1 if $r > 1$, showing lack of monotonicity. The same happens from a certain value of r , fixed A . Family $Y_{A,r}$ is also unbalanced but asymmetric. When r is in the interval $(R_0, 1)$ with $R_0 \approx 0.71$, CEN is not only out-of-range from a certain value of A , but its limit when $A \rightarrow +\infty$ is > 1 if $r > 1$, showing lack of monotonicity. But there are other two intervals of values for r in which $CEN > 1$ for A living in a certain bounded interval.

Besides evaluating binary confusion matrices with the same classification results for the minority class but different balances of the two classes, we compare through two examples the behaviour of MCEN with that of CEN, ACC^* and MCC^* , in evaluating improvements in classification of the minority class while maintaining the same amount of imbalance. We show that CEN is the only one that does not show a monotonous decrease as the classification improves, for which MCEN proves, also in this sense, that it outperforms CEN.

Finally, we also consider the multi-class family Z_A , which is asymmetric and unbalanced, and observe that in the binary case, CEN is out-of-range for $A \in (1, A_3)$, with $A_3 \approx 1.85$.

In all of these examples, MCEN behaves appropriately. Comparing with the overall Shannon's entropy associated to the set of elements of the confusion matrix, both CEN and MCEN are sensitive to it but CEN sometimes does not show the same behaviour in terms of monotonicity than entropy. With respect to Accuracy and MCC, conveniently scaled, they show sometimes a behaviour in contradiction with Shannon's entropy, as for families V_A and Z_A .

A further comparison has been carried out with the Probabilistic Accuracy (PACC) introduced in [13], and the Entropy-Modulated-Accuracy EMA and the Normalized Information Transfer (NIT) factor, both introduced in [15]. We consider different examples in which sometimes $PACC^* = 1 - PACC$ behaves consistently with MCC^* , increasing when IN entropy decreases and decreasing when OUT decreases, while $1/NIT$ behaves in accordance with CEN and MCEN, decreasing in both cases, but with the handicap that unlike what happens with

CEN and MCEN, NIT factor does not distinguish between IN and OUT. But not always. Actually, no measure seems to be completely satisfactory since each one reflects a specific characteristic of the classification process, so they should be used in a complementary way and none can be taken as a gold standard to compare the others.

Finally, to make clear the improvement of MCEN over CEN, we carry out experimentation consisting in the comparison of the rankings of some classifiers obtained from four different real datasets by using both measures. Mostly the classifiers orderings match, but when they do not, it is the MCEN that most agrees with the criterion of entropy. To see that, we use both the Hamming distance and the degree of consistency indicator c . These results heuristically support the use of MCEN as a better alternative to CEN in the binary case, when a performance measure based in entropy is required.

Supporting information

S1 File. Supporting information: Experiments and results. Table A. Datasets used in the experiments. Table B. Classifiers used in the experiments. Table C. Results for the Breast cancer dataset. Table D. Results for the SPECT heart dataset. Table E. Results for the Congressional voting dataset. Table F in S1 File. Results for the MONK's Problems.
(PDF)

S2 File. Breast cancer dataset.
(CSV)

S3 File. Breast cancer description.
(PDF)

S4 File. SPECT dataset.
(CSV)

S5 File. SPECT description.
(PDF)

S6 File. UCB admissions dataset.
(CSV)

S7 File. UCB admissions description.
(PDF)

Acknowledgments

This work have been supported by Ministerio de Economía y Competitividad, Gobierno de España, project ref. MTM2015 67802-P.

The authors wish to thank the anonymous referees for careful reading and helpful comments that resulted in an overall improvement of the paper, and more especially for drawing their attention on the paper [14]. They also are grateful to the Center for Machine Learning and Intelligent Systems of the Bren School of Information and Computer Science (University of California, Irvine, U.S.A.) for creating and maintaining the UCI Machine Learning repository.

Author Contributions

Formal analysis: Rosario Delgado, J. David Núñez-González.

Investigation: Rosario Delgado, J. David Núñez-González.

Writing – original draft: Rosario Delgado, J. David Núñez-González.

Writing – review & editing: Rosario Delgado, J. David Núñez-González.

References

1. Matthews B.: Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et biophysica acta*. Vol 405, Num 2, 442–451 (1975). [https://doi.org/10.1016/0005-2795\(75\)90109-9](https://doi.org/10.1016/0005-2795(75)90109-9)
2. Delgado, R., Núñez-González, D.: Enhancing Confusion Entropy (CEN) as measure for evaluating classifiers. In: Graña M. et al. (eds) *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18*. *Advances in Intelligent Systems and Computing*, vol 771. Springer, Cham (2019).
3. Wei J.-M., Yuan X.-Y., Hu Q.-H., Wang S.-Q.: A novel measure for evaluating classifiers. *Expert Systems with Applications*, Vol 37, 3799–3809 (2010). <https://doi.org/10.1016/j.eswa.2009.11.040>
4. Jurman G., Riccadonna S., Furlanello C.: A Comparison of MCC and CEN Error Measures in Multi-Class Prediction. *Plos One*. Vol 7, Num 8, 1–8 (2012).
5. Jin, H., Wang, X.-N., Gao, F., Li, J., Wei, J.-M.: Learning Decision Trees using Confusion Entropy. *Proceedings of the 2013 International Conference on Machine Learning and Cybernetics*, Tianjin, 14-17 July (2013).
6. Roumani Y.-F., May J.-H., Strum D.-P.: Classifying highly imbalanced ICU data. *Health Care Manag. Sci.* Vol 16, 119–128 (2013). <https://doi.org/10.1007/s10729-012-9216-9>
7. Roumani Y.-F., Rouman Y., Nwankpa J.-K., Tanniru M.: Classifying readmissions to a cardiac intensive care unit. *Annals of Operations Research*, vol. 263 (1-2), 429–451 (2018). <https://doi.org/10.1007/s10479-016-2350-x>
8. Wang X.-N., Wei J.-M., Jin H., Yu G., Zhang H.-W.: Probabilistic Confusion Entropy for Evaluating Classifiers. *Entropy*, Vol 15, 4969–4992 (2013). <https://doi.org/10.3390/e15114969>
9. Antunes F., Ribeiro B., Pereira F.: Probabilistic modeling and visualization for bankruptcy prediction. *Applied Soft Computing* vol. 60, 831–843 (2017). <https://doi.org/10.1016/j.asoc.2017.06.043>
10. Sublime J., Grozavu N., Cabanes G., Bennani Y., Cornuéjols A.: From Horizontal to Vertical Collaborative Clustering using Generative Topographic Maps. *International Journal of Hybrid Intelligent Systems*, vol. 12(4), 245–256 (2015). <https://doi.org/10.3233/HIS-160219>
11. Sublime, J., Matei, B., Murena, P.-A.: Analysis of the influence of diversity in collaborative and multi-view clustering. *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, 4126–4133 (2017).
12. Sublime J., Matei B., Cabanes G., Grozavu N., Bennani Y., Cornuéjols A.: Entropy based probabilistic collaborative clustering. *Pattern Recognition*, vol. 72, 144–157 (2017). <https://doi.org/10.1016/j.patcog.2017.07.014>
13. Sigdel M., Aygun R.: Pacc—A Discriminative and Accuracy Correlated Measure for Assessment of Classification Results. *Machine Learning and Data Mining in Pattern Recognition*. Vol 7988. LNCS. pp 281–295 (2013). https://doi.org/10.1007/978-3-642-39712-7_22
14. Valverde-Albacete F.J., Peláez-Moreno C.: 100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox. *Plos One*. Vol 9, Num 1, 1–10 (2014).
15. Klami, A., Ramkumar, P., Virtanen, S., Parkkonen, L., Hari, R., Kaski, S.: ICANN / PASCAL 2 Challenge: MEG Mind Reading –Overview and Results. In: Klami, A., editor, *Proceedings of ICANN/PASCAL2 Challenge: MEG Mind Reading*. Espoo, Aalto University Publication series SCIENCE + TECHNOLOGY 29/2011, pp. 3–19. <http://urn.fi/URN:ISBN:978-952-60-4456-9>
16. Huang J., Ling C.: Using AUC and Accuracy in Evaluating Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, 299–310 (2005). <https://doi.org/10.1109/TKDE.2005.50>