Expert Systems With Applications 137 (2019) 191–201

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Fostering interpretability of data mining models through data perturbation

Seddik Belkoura [a,*], Massimiliano Zanin [b,c], Antonio LaTorre [a,d]

[a] Center for Computational Simulation, Universidad Politécnica de Madrid, Madrid, Spain
[b] Centro de Tecnología Biomédica, Universidad Politécnica de Madrid, Madrid, Spain
[c] Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, Portugal
[d] DATSI, ETSIINF, Universidad Politécnica de Madrid, Madrid, Spain

ARTICLE INFO

Article history:
Received 19 February 2019
Revised 1 July 2019
Accepted 1 July 2019
Available online 2 July 2019

Keywords:
Interpretability
Data mining
Random forest
Artificial neural networks

ABSTRACT

With the widespread adoption of data mining models to solve real-world problems, the scientific community is facing the need of increasing their interpretability and comprehensibility. This is especially relevant in the case of black box models, in which inputs and outputs are usually connected by highly complex and nonlinear functions; in applications requiring an interaction between the user and the model; and when the machine's solution disagrees with the human experience. In this contribution we present a new methodology that allows to simplify the process of understanding the rules behind a classification model, even in the case of black box ones. It is based on the perturbation of the features describing one instance, and on finding the minimal variation required to change the forecasted class. It thus yields simplified rules describing under which circumstances would the solution have been different, and allows to compare these with the human expectation. We show how such methodology is well defined, model-agnostic, easy to implement and modular; and demonstrate its usefulness with several synthetic and real-world data sets.

1. Introduction

If the concepts of interpretability and comprehensibility originally appeared alongside the first real-world data mining applications (Kononenko, 1993; Lavrač, Džeroski, Pirnat, & Križman, 1993), it has only been in the last years, with the rise of complex nonlinear models such as those produced by *e.g.* neural networks (Montavon, Samek, & Müller, 2018), that their relevance has substantially increased (Freitas, 2014; Gerretzen et al., 2016; Lipton, 2016). If one has to select one case exemplifying the need for an increased interpretability of data mining models, this can be easily found in medicine, *e.g.* in cancer diagnosis and treatment. Nowadays, physicians have to process a large number of inputs coming from different analyses, such as x-rays, biopsies, or genetic tests, to build an optimal treatment combining together different therapies (Urruticoechea et al., 2010). As this choice cannot be formalised using a simple set of rules, the future points towards a widespread adoption of Medical Diagnostic Decision Support systems (MDDS) to support physician activity (Berner, 2007; Miller, 1994; Musen, Middleton, & Greenes, 2014; Scott et al., 2019). In this context, the need for interpretability comes from different sides. From a scientific point of view, one is clearly interested in the mechanisms detected by the system: *e.g.* not just which treatment should be administered, but also why it is considered the most appropriate; this knowledge further fosters trust and reliance on the system (Bussone, Stumpf, & O'Sullivan, 2015). Additionally, it may be necessary to resolve conflicting solutions, *e.g.* when the MDDS and the clinician have different views on the same problem, thus requiring a reasoned disambiguation; and to support training, *i.e.* when a trainee is exposed to different situations and he/she compares his/her responses with those of the system, to learn from them (Lagro et al., 2014; Yoon, Velasquez, Partridge, & Nof, 2008).

In order to tackle this problem, several alternatives have been proposed in the literature, of which the most promising are based on *model-agnostic explanation* (Ribeiro, Singh, & Guestrin, 2016a). Roughly speaking, such solutions are based on an *a posteriori* description of a black-box model through a set of (simpler) rules - see Section 2 for further details. Nevertheless, it has to be noted that this approach does not truly tackle the problem, but just rephrases

* Corresponding author.
E-mail addresses: seddik.belkoura@gmail.com (S. Belkoura), massimiliano.zanin@gmail.com (M. Zanin), a.latorre@upm.es (A. LaTorre).

https://doi.org/10.1016/j.eswa.2019.07.001
0957-4174/

it, by mapping a complex model to a simpler, albeit not necessary interpretable, one.

In this contribution we propose a novel methodology for enhancing interpretability, which builds on top of the model-agnostic explanation concept, but does not rely on creating alternative classification models. Given an already trained model, for instance for the classification of instances between n classes, and a new instance to be analysed, we propose an algorithm yielding the smallest variation needed to change the class of the latter instance to match the one expected/desired by the user. The user is then able to understand under which conditions would the solution of the model have been different; or equivalently, why that specific solution was yielded. On a more abstract level, the user can leverage on this information to create his/her own representation of the black-box model, without being conditioned by any *a priori* assumption on the structure of that representation. Beyond yielding the smallest variation needed to swap classes, the method can also be tuned to minimise the number of features involved in this change - as, in many contexts, simpler solutions (that is, minimising the number of changes) can be preferred for being easier to understand and/or implement. We show how such methodology is well defined, model-agnostic, easy to implement and modular, as it does not impose constraints on the complexity of the classification algorithm, which is treated as a black-box. Above and beyond this, we demonstrate that the proposed approach allows to improve the interpretability of data mining models, and to help tackling problems as the previously described ones, *e.g.* the disambiguation of conflicting solutions and the improvement of medical training.

Beyond this introduction, the remainder of the paper is organised as follows. Section 2 presents a brief overview on the interpretability concept, and on how it has historically been dealt with in data mining. The proposed methodology is presented in Section 3, for then applying it to four case studies, constructed upon data sets respectively representing different real world problems and data characteristics (Section 4). Afterwards, Section 4.4 presents an analysis on the optimality and computational cost of the method. Finally some conclusions are drawn in Section 5.

2. Related work

In spite of past attempts, there seems to be no clear definition of what the interpretability of a model is and how it can be measured (Bibal & Frenay, 2016; Doshi-Velez & Kim, 2017; Narayanan et al., 2018).

There are a number of heuristics, guidelines and rules of thumb that the community has been using for years to both define and assess interpretability (Biran & Cotton, 2017; Gilpin et al., 2018), albeit without a clear formalisation nor empirical evaluation. In this sense, recent work (Bibal & Frenay, 2016; Freitas, 2014) proposes that comprehensibility can be assessed at two different levels: by examining models (what they call a heuristic approach) or representations (mainly with user-based surveys). In the former case, simple measures can be used to compare several models of the same type, such as the number of rules and terms in decision rules (Letham, Rudin, McCormick, & Madigan, 2013; Schwabacher, Langley, & Norvig, 2001) or the number of nodes in decision trees (Van Assche & Blockeel, 2008). If models differ, then this comparison is not that obvious and other heuristics have been proposed. Among others, it is worth citing the ranked taxonomy (Backhaus & Seiffert, 2014) that assigns a category to each model, where models within the same category are competitive among them in terms of interpretability, whereas models in upper levels are more interpretable than those in lower levels. With respect to representation comprehensibility, user-based surveys normally consider general abstract questions about the models, *e.g.* "is this model more understandable than the other one?" (Allahyari & Lavesson, 2011), which allow comparisons of models of different nature. Freitas (2014) provides a good overview on multiple studies on the interpretability of different models.

Closely related to interpretability, the research community has also proposed the concept of *interestingness, i.e.* the assessment of the potential interest to the user of the patterns generated by a data mining model. Clearly, interpretability is a prerequisite to interestingness, as one should be able to clearly extract patterns before assessing their usefulness. Broadly speaking, interestingness can be measured through three families of metrics: objective measures, based on the probability and format of the patterns; subjective measures, including surprisingness and novelty; and semantic measures, *e.g.* utility and actionability, thus based on the domain knowledge. For a complete review on the topic, the interested reader can refer to Geng and Hamilton (2006) and Potes Ruiz, Kamsu Foguem, and Grabot (2014).

Besides measuring it, many applications require improving the interpretability of a model. This can trivially be achieved by designing inherently interpretable models, *e.g.* by using white-box models (such as decision trees or classification rules) over black-box models (SVMs, ANNs, *etc.*) when interpretability is important. This can nevertheless introduce some important drawbacks, as accuracy must be sacrificed in favor of comprehensibility and some models might present design artifices that may induce errors (Freitas, 2014). Additionally, an untrained operator may find difficult to interpret even results yielded by simple algorithms, as previously shown with decision trees (Bratko, 1997), rule induction (Lavrač et al., 1993) or bayesian classifiers (Kononenko, 1993).

In order to solve the aforementioned problems, some authors proposed to enforce interpretability in a model-agnostic way (Ribeiro et al., 2016a). These methods lean on learning a posteriori explanations of the models taken as a black-box, which would locally (though not globally) be faithful to such models; in other words, this can be seen as the creation of a simplified and easily interpretable model, given the black box one. Note that this implies an equivalence between the concept of interpretability and the following question: "are humans able to make accurate predictions about a model's behaviour?". Examples include the use of sparse linear models (Ribeiro, Singh, & Guestrin, 2016c), gradients (Baehrens et al., 2010), and if-then rules (Ribeiro, Singh, & Guestrin, 2016b).

Finally, it is worth noting that the purpose of making comprehensible models is also frequently unclear (Lipton, 2016; Miller, 2019; Mittelstadt, Russell, & Wachter, 2019). Lipton (2016), in his study, tries to contextualise the definition of comprehensibility and he finds out that it appears often related to trust, normally in the context of medicine and health-care (Breiman, 2001b; Caruana et al., 2015), but also in other scenarios, such as criminal justice systems, financial markets (Lipton, 2016) or even education (Kim, Glassman, Johnson, & Shah, 2015). Apart from trust, there are other reasons for comprehensibility, such as causality, transferability or informativeness. In this sense, he argues that interpretable models may have different characteristics, mainly dealing with transparency (how the model works) and post-hoc interpretability (what else the model can tell me).

Our contribution leverages on the existing literature and on the concept of model agnostic explanation. Specifically, we start from the idea that model comprehensibility can be understood as the ability to identify the factors behind the model classification process; consequently, the methodology we propose leverages on identifying the features which are responsible for the yielded result, or alternatively, for reaching an alternative result. While this approach is consistent with the literature on the identification of important features (or combination of features) at the macroscopic level, it is - to the best of our knowledge - the first research to

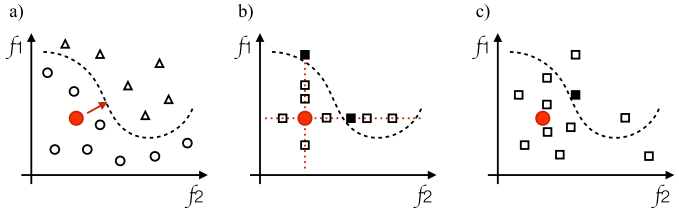

Fig. 1. Graphical representation of the process for creating rationales from a black-box classification model. Panel *a*) depicts the initial situation, while *b*) and *c*) respectively represent the search in one and two dimensions. See main text for details.

also focus on an individualised understanding of a single observation. We believe that while state-of-the-art techniques offer value in the understanding of models as a whole, and as such are useful to understand a set of observations, they lack granularity to interpret the model results at a microscopic level, *i.e.* for a single observation. In other words, we here shift the focus from increasing the intepretability of a whole model, to improving our understanding of the rationales behind the classification of a single instance.

3. Methodology

In order to simplify the explanation of our approach, and without loss of generality, we here suppose a simple two-classes classification. As depicted in Fig. 1 Left, several instances are described by just two features f_1 and f_2, such that the problem space is limited to a plane. The class of each instance is denoted by black circles and triangles, while a black dashed line depicts the inner separation of the classification model (*a priori* unknown to the user). Finally, a new instance is presented to the system (or user) and is classified by the model - as represented by the red solid circle. As previously introduced, the goal is to create a simple representation of the rationales of such new classification; or, in other words, an understandable abstraction of the black dashed line. Note that a simple estimation of the two features' importance in the classification would not suffice, as this is calculated for the whole model, and is not specific to the analysed instance. We here propose to achieve this by finding the minimal change in the instance features (here, f_1 and f_2) that would result in a different forecast label - see the red solid arrow pointing towards the classification frontier.

Finding such minimal change could be achieved by resorting to a *brute force* analysis, in which a large number of locations on the features' plane are evaluated by the model, for then finding the one closest to the analysed instance. The computational cost of such solution would nevertheless be of $O(n^d)$, n being the sampling resolution required for each feature and d the number of features. The brute force approach thus becomes computationally unfeasible even for a handful of features.

Instead of sampling the complete feature space, the methodology here proposed is based on restricting the search in the neighbourhood of the instance under analysis (the red circle of Fig. 1 Left). Additionally, the dimensionality of the search is progressively increased. In the first iteration each feature is changed in an independent way, *i.e.* by considering one dimension of the feature space at a time; afterwards, if no solution of interest has been detected, the algorithm moves to higher numbers of dimensions. The methodology steps have been summarised in a flowchart (Fig. 2) and are further described hereafter:

1. Initially set n_f, *i.e.* the number of features to be changed at the same time, to 1.
2. Create and classify new virtual instances:
   a Create a vector C containing all the possible combinations of n_f elements out of d (the latter being the total number of available features). Note that, when $n_f = 1$, C contains a list of all the features in the problem.
   b For every element c in C, generate n_p new virtual instances. The coordinates of these new instances are randomly drawn from (i) an *a priori* distribution specified by the user or (ii) an empirical distribution observed in similar datasets. All other features, *i.e.* those not included in c, are copied from the reference instance. Note that, for $n_f = 1$, this is equivalent to randomly changing one feature at a time, and save the result as a new instance.
   c Sort all new virtual instances as a function of their distance to the target instance.
   d Apply the classification model to predict the class of each new virtual instance, starting from the one closest to the target instance. When one of them is assigned to another possible class chosen by the user,[1] stop the process and jump to step (3).
3. If no element of the desired class is found, increment n_f by one and go back to step (2).
4. Select the closest virtual instance for each combination of features - note that there might be no solution for one specific combination.

Four important aspects of the method must be highlighted. Firstly, the algorithm always gives priority to low dimensional solutions. Following Occam's Razor principle, larger combinations of features are not explored unless a solution could not be found in the previous steps. The stop conditions for steps 2.d and 3 can nevertheless be customised. For instance, multiple feasible solutions can be saved for each set of modified features; likewise, even if a valid solution is found for a given n_f, the algorithm can still be executed for $n_f + 1$. In both cases, the obtained result would be a Pareto front, *i.e.* multiple solutions, all of them feasible, that have to be evaluated and prioritised by the user of the system. Going back to the previous biomedical example, an MDDS may suggest a different treatment would the patient had a very different age, or a different (yet close) age and a different physical condition, and so forth. The physician may then evaluate all options, and get a better view of the classification rationales.

Secondly, the sampling procedure in step 2.b is designed to optimise the search in the feature space. By drawing the new feature values from the corresponding empirical distributions, one minimises the number of virtual instances created in regions of the feature space that are sparsely populated, thus maximising the probability of finding a relevant result. This can nevertheless be substituted by more complex search strategies, as for instance a Simulated Annealing (Kirkpatrick, Gelatt, & Vecchi, 1983) or a Genetic Algorithm (Davis, 1991).

Thirdly, the distance that has here been considered between instances, both real and virtual, is a simple Euclidean one; in other words, all features are considered as equal from a distance point of view and all features are supposed to be continuous. Some situations may nevertheless call for a different approach: for instance, when looking for ways of modifying the classification of an instance, some features may be more complicated to change (*e.g.*, some treatments cannot easily be changed, as essential for the well-being of the patient) and the sensitivity of their contribution to the best solution should be adapted accordingly. Additionally, when features are not equally normalised, the same absolute variation may have different relative importance. This can easily be accounted for in the proposed methodology, by adding a weight vector, or alternatively by normalising the input data set; it is yet to be noted that the choice of the weighting vector de-

[1] In a binary case, this would be equivalent to stop when a change of class has been detected.

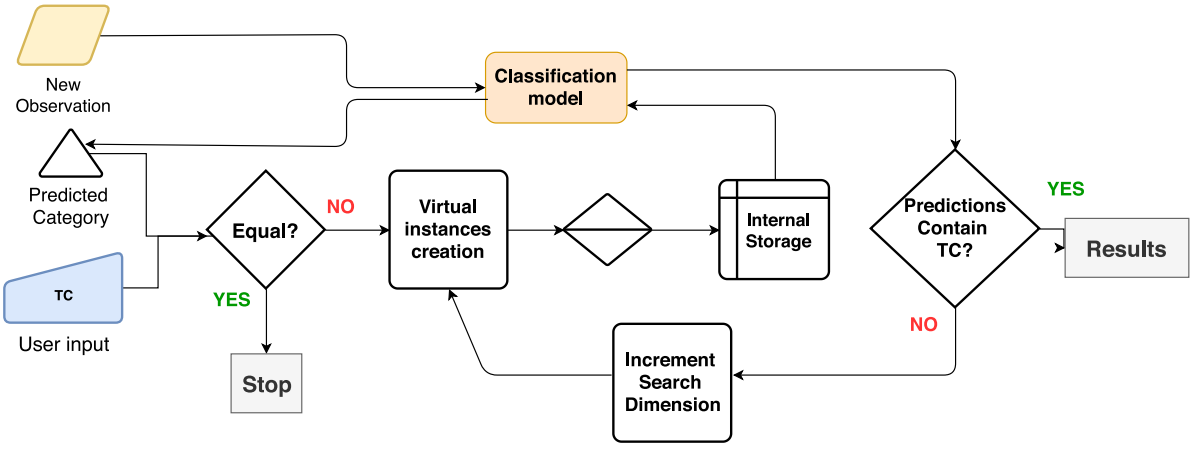

Fig. 2. Flowchart of the proposed methodology - See Section 3 for details.

pends on the user's understanding of the problem, and hence is an arbitrary choice. Finally, the very nature of the feature might influence the choice of the distance metric. It is evident that categorical features must be treated differently than continuous ones, as a Euclidean distance is ill-defined in the first case. While some categorical features, such as age, can be transformed back into a continuous value where the Euclidean distance is meaningful, we can imagine other scenarios where this is not possible, as *e.g.*, the type of a cell or the eye colour. In such cases, the solution may entail developing customised distances to take this into account (*e.g.*, refer to distance techniques like categorical embedding (Guo & Berkhahn, 2016; Levy & Goldberg, 2014) or similarity measurements (Boriah, Chandola, & Kumar, 2008)).

Fourthly, the computational cost of this approach, while still important, is substantially reduced with respect to the brute force analysis, as it is dominated by the maximum number of features explored at the same time - as opposed to the total number of features. This point will further be discussed in Section 4.5.

We believe that the proposed methodology can provide valuable information in numerous decision making problems; and we demonstrate this in the following two sections, by presenting four case studies based on real-world data-sets. In order to further ensure the usefulness to the scientific community, a Python implementation of the methodology is freely available at the following link: https://gitlab.com/SBINX/DataWhitening.

4. Results

4.1. Data set description

The proposed methodology has been tested using four public and well-known data sets, all of them available through the UCI repository. They have been selected in order to be representative of heterogeneous areas and classification problems - e.g. different types of input features and number of labels.

Breast cancer

As a first data set we considered the Breast Cancer Wisconsin Data Set, described in Street, Wolberg, and Mangasarian (1993) and Mangasarian, Street, and Wolberg (1995) and publicly available at [dataset]Wisconsin Breat Cancer data set (2017). It contains information about 569 patients with breast cancer, each one characterised by 30 features computed from a digitalised image of a fine needle aspirate (FNA) (Wu & Burstein, 2004) of the breast mass. Each cell nucleus is described by ten features: radius, texture, perimeter, area, smoothness, compactness, concavity, number of concave points, symmetry, and fractal dimension. The average, standard error and average of the three largest values are then calculated for each feature, thus yielding a final set of 30 values for each patient. The task aims at diagnosing the cancer as malign or benign.

Portuguese wines

The second data set includes characteristics of Portuguese white wines, as described in Cortez, Cerdeira, Almeida, Matos, and Reis (2009) and available at [dataset]Wine Quality data set (2019). Each one of the 4898 wines is described by 12 physiochemical features: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol. Additionally, a quality rating in 11 classes was assigned to each wine by experts, based on a sensory taste test: from 0, very bad, to 10, excellent. We simplified the case study by considering a bivariate output, such that wines of quality inferior to 6 have been grouped together in class 1, and all the others in class 2.

Mushrooms

As a third example we have considered the well-known mushrooms data set ([dataset] Mushroom data set, 2019; Schlimmer, 1987), in which 8124 mushrooms are described in terms of 22 categorical physical characteristics. As in previous cases, the task to be executed is a classification one, aimed at discriminating between poisonous and edible mushrooms.

Lung cancer

The fourth and final example includes a data set of lung cancer patients data ([dataset] Lung Cancer data set 2019; Hong & Yang, 1991). The problem consists in a classification task, in which the label of each instance is to be forecast among three pathological lung cancer types. The 32 instances are described by means of 56 nominal features, all of them assuming integer values in the range $[0-3]$. The interest of this data set resides in the fact that the problem is ill-defined, as more features than instances are provided. Finally, note that Ref. (Hong & Yang, 1991) gives no information on the meaning of the features, nor on the origin of the data.

4.2. Classification models creation

The classification models have been based on two well known algorithms. First, a Random Forest (Breiman, 2001a; Verikas, Gelzinis, & Bacauskiene, 2011), an evolution of the classical Decision Tree model (Quinlan, 1986), in which multiple trees are trained on subsets of the original data, for then merging their results into a

Table 1
Classification scores and average training time for the four considered data sets, for the RF and ANN classification algorithms.

Classification model	Accuracy No CV	Accuracy CV	Avg. training time (sec.)
Breast cancer			
Random Forest	99.29%	94.31%	1.54
ANN	97.89%	94.83%	2.15
Portuguese wines			
Random Forest	97.48%	76.77%	3.62
ANN	72.64%	72.30%	2.01
Mushrooms			
Random Forest	100.0%	100.0%	5.58
ANN	100.0%	100.0%	2.36
Lung cancer			
Random Forest	100.0%	46.87%	1.18
ANN	100.0%	53.12%	0.19

Table 2
Synthesis of the results obtained for the breast cancer, Portuguese wines and mushrooms data sets. In each case we report the feature (or set of features) most frequently responsible for a class change; the frequency of times such feature of combination appeared; and the average change suggested by the methodology.

Feature	Frequency	Suggested change
Breast cancer (*malignant* → *benign*)		
Concave points	62.2%	Decrease below 0.04
Breast cancer (*benign* → *malignant*)		
Radius & area	34.5%	Increase radius above 17 and area above 105
Radius & perimeter	25.4%	Increase radius above 18 and area above 950
Portuguese wines (*bad* → *good*)		
Volatile acidity	46.1%	Decrease below 0.21
Density	23.2%	Decrease below 0.991
Alcohol	18.6%	Increase above 10. In a 18% of cases, an increase above 12 is required.
Mushrooms (*edible* → *poisonous*)		
Gill size	100.0%	Broad to narrow
Mushrooms (*poisonous* → *edible*)		
Odor	71.31%	Pungent to almond
Gill size	28.69%	Narrow to broad

single classification score. We used the implementation included in the Scikit-learn library (Pedregosa et al., 2011), with 1,000 estimators and a minimum number of samples in each split of 10. Second, the Artificial Neural Network (ANN) model, in the implementation known as Tensorflow (Abadi et al., 2016; Rampasek & Goldenberg, 2016). The model was composed of one hidden layer with three neurons, and has been trained during 1,000 epochs. Note that, in both cases, the algorithms' parameters have not been fine tuned, as the objective here is not to obtain the highest classification score, but instead to illustrate the capabilities of the methodology. Table 1 reports the classification score obtained by both models on the four data sets, both with and without a Leave-One-Out Cross-Validation (CV) (Kohavi, 1995), and the time needed to train them. As previously highlighted, the methodology here proposed is agnostic as to the algorithm used in the classification, which is treated as a black box; any other classification model, from the many available in the literature, could thus be used.

4.3. Classification models analysis

Once the classification models have been trained, we applied the proposed methodology, whose main results are synthesised in Table 2.

The breast cancer and Portuguese wines data sets yield expected results - see also Fig. 3 for a graphical representation. Regarding the first, the forecast class changes to benign when the number of concave points is reduced, *i.e.* the shape is more regular; on the other hand, the class becomes malignant with an increase in the radius, area and / or perimeter (note that these three features are related, thus a change on at least two of them is required). These results are in line with what previously reported in the literature (Abdalla et al., 2008; Narasimha, Vasavi, & Kumar, 2013; Wittekind & Schulte, 1987). In the case of the wines data set, the most important element allowing to increase the quality of a lesser wine is the reduction of its volatile acidity. This has extensively been studied in the literature: a high acidity can result in an acrid taste (Boulton, Singleton, Bisson, & Kunkee, 2013; Lonvaud-Funel, 1999), and can be addressed through techniques like microbial stabilisation, nanofiltration or refermentation (Vilela-Moura et al., 2011). Note that the values reported in Table 2 correspond to RF classification models; using ANNs yields qualitatively similar results.

The mushrooms data set presents an important difference with respect to the two previous cases, *i.e.* the fact that features can only assume discrete values. This can easily be addressed by changing the way virtual instances are created: instead of sampling the new feature values from a uniform distribution, these are forced to be the discrete values in the original data set. Additionally, if the classification model is not able to handle categorical variables (as is the case of ANNs), these new feature values can be converted to binary variables through a One Hot Encoder. Table 2 suggests that the two most important features for achieving a class change are the gill size and the odour - this is also in line with previous results (Dong & Li, 1999; Ishikawa, 2000).

The fourth and final data set is the most complex one. Firstly, as the mushrooms one, its features are categorical; as in the previous case, the generation of virtual instances should be modified accordingly. Secondly, it involves a classification task with three

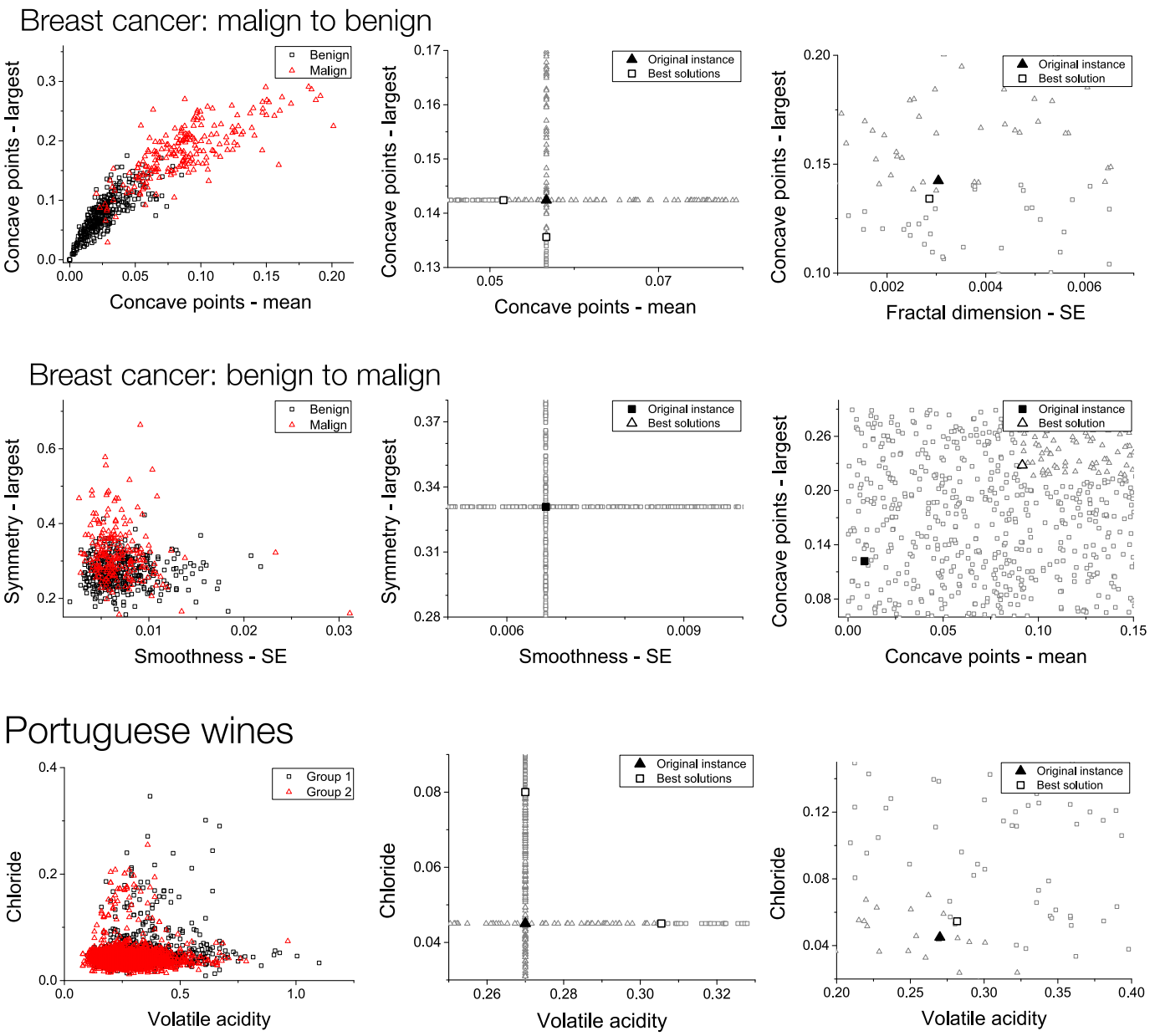

Fig. 3. Graphical representation of the proposed methodology applied to the breast cancer (first and second row) and the wine (third row) data sets. In the former case, the top row corresponds to a malign cancer, the central row a benign one. Left, centre and right panels respectively depict the original data set, the search by varying one single feature, and the full search in the plane. See main text for details.

classes; hence, two results have to be found for each instance, corresponding to the two classes different from the forecast one. Finally, the high number of features, compared to the limited number of instances, implies that many of them are strongly correlated; as a consequence, modifying a few features seldom results in a change of class, as all correlated features must be changed coherently. To illustrate this, only in a 12.5% of cases a new class was found by changing two features; this number increases to 17% and 30% for respectively three and four features. This can be solved by a feature selection process, in which the most correlated features are identified and dropped. To confirm this idea, Fig. 4 reports the evolution of the percentage of times a solution is found, as a function of the number of deleted features (using a χ^2 test to evaluate the correlation between them). Single-features solutions are started to be found when more than 30 redundant features are deleted; at the same time, the number of solutions with

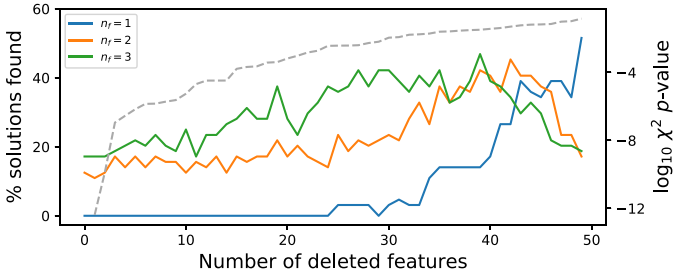

Fig. 4. Evolution of the percentage of instances for which the methodology is able to find a solution, as a function of the number of features deleted during the feature selection process. Solid lines depict the evolution for $n_f = 1$, 2 and 3. The dashed grey line (right Y axis) depicts the evolution of the $\log_{10}$ of the p-value yielded by a χ^2 test executed over the feature deleted at each iteration.

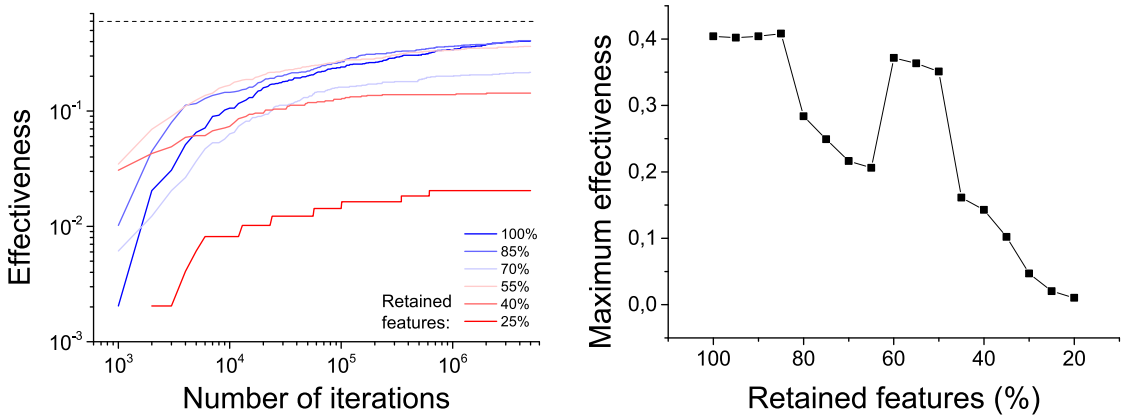

Fig. 5. Efficiency of a Simulated Annealing optimisation. The six curves in the left panel represent the evolution of the SA success (*i.e.* the fraction of times it finds a solution better than the proposed methodology) as a function both of the percentage of features included in the search, and of the number of iterations. Additionally, the right panel depicts the maximum achieved success as a function of the percentage of features included in the search.

two or three features starts to decline, as complex combinations of changes are no longer required. It can then be concluded that, while the proposed methodology is agnostic to the used classification model, the latter one should be a meaningful representation of the problem, as unreliable results may otherwise be obtained.

4.4. Solution optimality and computation cost

As previously discussed, the proposed methodology is based on the idea that a classification model can be better understood when one or few features are varied, both to make these features *actionable* and to reduce the computational cost associated to a full feature space search. It is nevertheless possible that a better solution, in terms of the sum of the variations applied to the features, could exist, provided the number of modified features is not constrained. In other words, the proposed methodology may be yielding a sub-optimal solution; while not a problem *per se*, this may introduce biases in the way the model is understood. If such optimal solutions could in principle be detected through other approaches, as *e.g.* a brute force search, one has to balance the enhanced precision with the increased computational cost of the analysis. This balance is studied in this section, by presenting the results obtained with Simulated Annealing and brute force search strategies.

4.4.1. Simulated annealing

As a first step, we here compare the results obtained for the wine data set with the solutions found by a Simulated Annealing (SA) algorithm (Hwang, 1988). SA is an optimisation technique aimed at finding an approximate global optimum in a fixed amount of time. It is inspired on annealing in metallurgy, a technique involving heating and controlled cooling of a material. Given a target function, in this case the Euclidean distance between the solution and the original instance, the SA algorithm stochastically samples new solutions around the previous best result, while slowly decreasing the search radius.

With respect to a standard Simulated Annealing model, we have here added the possibility of fixing the number of features composing the search space. Given the original instance, which is used as the starting point of the optimisation, the model creates multiple copies of it and randomly searches a solution in its neighbourhood; the latter is composed of a subset of randomly chosen features, which is fixed at the beginning of the process. This allows to study a full spectrum of results: from unrestricted searches, in which all features can be explored, thus corresponding to a standard SA implementation; to searches focusing on a few, randomly selected features, in a fashion similar to what proposed in this contribution.

Fig. 5 Left reports the evolution of the effectiveness of the SA search, defined as the fraction of times the SA solution is better (*i.e.* closer to the target instance) than what obtained by the proposed methodology. Such effectiveness is reported as a function of the number of iterations of the SA algorithm, and of the percentage of retained features. Additionally, Fig. 5 Right depicts the evolution of the maximum effectiveness achieved by the SA algorithm as a function of the percentage of retained features.

Results indicate that, on one hand, the SA algorithm is able to reach results as good as the proposed methodology - note that an effectiveness of 0.5 represents a situation in which both algorithms are virtually identical. Yet, this can only be achieved at the cost of executing a large number of optimisation iterations, implying a significant computational cost. On the other hand, the SA is ill-designed to handle situations in which few features are allowed to vary; this is to be expected, as the randomly selected features (*i.e.* those that are allowed to change) could be irrelevant for the problem at hand, thus lowering the effectiveness.

4.4.2. Brute force search

If the SA algorithm is in principle able to yield a solution close to the one proposed by our algorithm, while with some caveats when the number of free features is limited, the previous analyses still do not clarify how far away such solution is from the optimal one. As previously discussed, this optimum can be obtained through a brute force approach; yet this is feasible only when the problem under analysis is described by few features, as the computational cost increases exponentially. We here study this issue through a simplified model, in which the classification model is defined as an $(n-1)$-dimensional hyper-plane intersecting an n-dimensional feature space, and in which virtual instances are created according to a regular mesh. This allows to estimate the average error of a brute force search as half the distance between neighbouring points in the search mesh. Fig. 6 reports the evolution of the average error yielded by the proposed algorithm (blue line) and the corresponding standard deviation (blue whiskers), when 10,000 virtual instances per feature are considered, as a function of the number of dimensions of the problem - *i.e.* the number of features. It can be appreciated that the error is almost constant, and is thus independent from the feature space dimensionality. On the other hand, the red line represents the ratio between the number of instances in the brute force search, as required to reach the same average error, and that of our solution; in other words, this represents how the computational cost scales

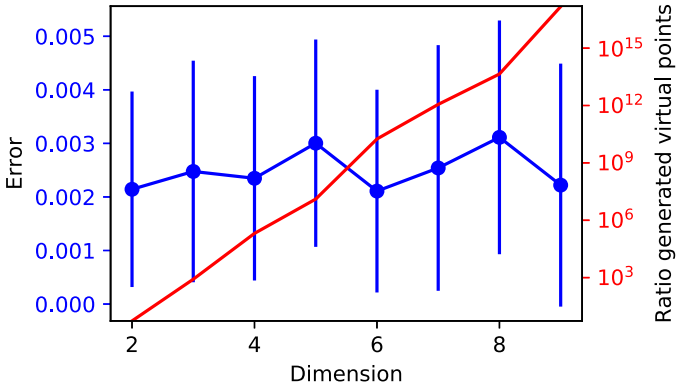

Fig. 6. Efficiency of a brute force search, as compared with the proposed algorithm. (Blue line, left scale) Average error yielded by the proposed algorithm, as a function of the number of dimensions of the feature space. (Red line, right scale) Ratio between the number of virtual instances needed by the brute force approach and that of the proposed algorithm, to achieve the same average error. See main text for a description of the classification model.

for the brute force search, when the same precision is to be obtained. Results indicate that a complete exploration of the feature space is unfeasible for more than four features (note the logarithmic Y scale).

4.5. Computational cost

The results presented in the two previous sections demonstrated that our approach is efficient, with respect to other more optimal alternatives, in terms of the number of virtual instances analysed. Specifically, as introduced in Section 3, the complexity of the methodology is dominated by the maximum number of features to be changed at the same time. One last question to be answered is how this translates into the computational cost; or, in other words, what is the execution time corresponding to a medium-sized problem in a standard desktop machine - an important aspect for real-world implementations.

Accordingly, Fig. 7 reports an histogram of the time required by the proposed methodology to compute the best solution, in a MacBook Pro-with an Intel Core i5 at 2.7GHz, both with one (left panel) and two (right panel) features, in the case of the wine data set. It can be appreciated that almost all solutions are found within 10 seconds, thus enabling real-time applications.

While this methodology does not depend on the nature of the classification algorithm and presents several advantages in operational decision making, as for instance in the previously discussed medical example, two interconnected limitations should here be discussed.

Firstly, while the computational complexity of the methodology does not depend on the output dimensionality, it does depend on the input one. The analysis has a complexity of $O(n_f^{\max} \cdot n_p \cdot c)$, $n_f^{\max}$ being the maximum number of features to be varied at the same time; n_p the number of virtual instances to be created; $c = \binom{n_f}{d}$; and d being the number of input features. This implies that applying the proposed methodology to some models, especially those expecting a large number of input features, may not be feasible. This may be the case of, for instance, convolutional neural networks (CNN) models, commonly used in medical imaging classification problems (Li et al., 2014; Lo et al., 1995; Milletari, Navab, & Ahmadi, 2016). However, a potential solution may involve the use of the inner layers of a highly dimensional network to reduce the dimensionality of the problem, coupled with a back-engineering for the extraction of the closest solution. Even without resorting to these solutions, tens of input features can be dealt with in reasonable time.

Secondly, the dimensionality of the input also depends on the number of features that should be perturbed to alter the forecast of an observation. If n_f is forced to be 1, then no limit on d is necessary, otherwise a trade-off must be encountered. If more features need to be perturbed at the same time, d must be controlled in order for the problem to be tractable. As this depends on the characteristics of the problem under analysis, the limit should be estimated in every specific case.

Both limitations can further be tackled by weighting the features to be perturbed, one can improve the probability of finding a solution of low dimension. But, highly dimensional solutions are seldom interesting in an operational context as more efforts needs to be implemented.

4.6. Feature relevance

One may intuitively expect that the features more relevant for the methodology here presented should correspond to those that are relevant in the construction of the original classification model. To illustrate, the opposite situation is easy to visualise. Let us suppose that, for example, one feature is irrelevant for defining the classification function $\tilde{F}$; given one instance, varying such feature will never change the predicted class of the instance, and therefore its analysis will be useless. While the opposite may not always be true, we explore this issue in Fig. 8, which depicts the importance of features in the classification as measured through

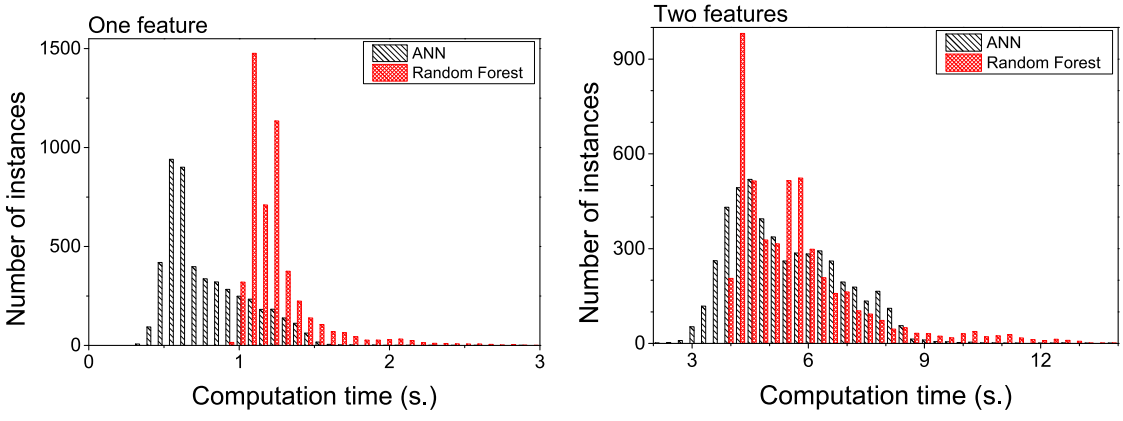

Fig. 7. Histogram of the computation time associated with the proposed methodology, for one (left panel) and two (right panel) features searches.

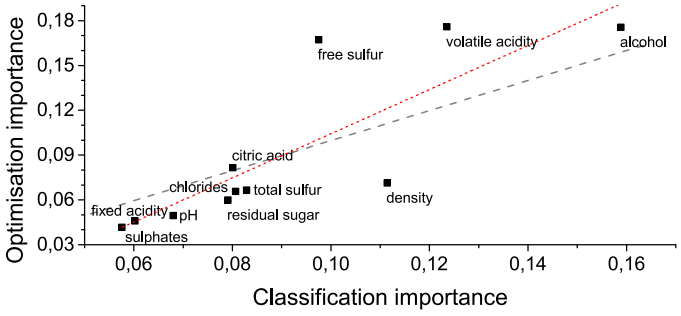

Fig. 8. Importance of features in the classification as a function of their importance in the proposed methodology, for the wine dataset - see main text for definitions. The grey dashed line represents the identity function, while the red one the best linear fit. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

their Gini impurity / information gain (Breiman, Friedman, Olshen, & Stone, 1984), as a function of their importance in the proposed methodology (fraction of times they are chosen in the best solution). Note that the Gini impurity measure is recognised to be a biased approach (Strobl, Boulesteix, Zeileis, & Hothorn, 2007), as it has a tendency to inflate the importance of continuous features or high-cardinality categorical variables. However, in case of continuous independent variables, results are consistent with other more robust methods, *e.g.* permutation importance (Strobl, Boulesteix, Kneib, Augustin, & Zeileis, 2008). The wine data-set presents continuous and independent input variables, as so the comparison of the most frequent variables used for class swap and the results of the Gini impurity is expected to be a relevant one. A linear relationship can be appreciated, as represented by the dashed red line - slope of 1.478 ± 0.33, $R^2 = 0,830$. While such relationship is not perfect, it nevertheless suggests that the initial model can be used as a way of optimising the whole process, by focusing initially on those features more relevant for the classification. Also, this serves as a proof of concept of the methodology. Indeed, by scanning the complete set of observations and extracting for each the individual characteristics that trigger a forecast change, we have been able to rebuild - or at least approximate - the distribution of the more important features used by the classification algorithm. This shows how our methodology results are indeed able to approximate the decision making rules of the algorithm.

5. Conclusions and discussion

In this contribution we have proposed a new method for increasing the interpretability and comprehensibility of data mining classification models. Starting from an instance that has been classified as belonging to one class, the solution entails identifying the minimum variation in the features' values required to change the output class. This methodology presents four major advantages worth discussing. First, it is model-agnostic, as it does not depend on the considered classification model, which is treated as a black-box - a beneficial feature in the case of complex models. Second, it presents a very low computational cost, especially when compared with alternative solutions like Simulated Annealing or brute force methods. Third, it has a flexible nature, as several steps and assumptions can be adapted to the problem under analysis. Finally, it is also worth noting that, while in this contribution we have focused on classification problems, the same approach can be applied to unsupervised scenarios, *e.g.* for improving the interpretability of clustering solutions. In this final section, we are going to present an additional consideration of relevance in real-world applications: the behaviour of the methodology for noisy or wrong classification models.

First of all, throughout the text we have considered a simplified situation in which the classification model is assumed as correct - *i.e.* if the classification of the instances is defined, in the real world, by an unknown function F, the trained data mining model reaches a function $\tilde{F}$ such that $\tilde{F} = F$. This is nevertheless an atypical situation: most data mining models only reach an approximation (at best) of the reality, due both to their intrinsic limitations and underlying hypotheses, to limited training data, and to the presence of noise in the training labels. This latter problem, known in data mining as learning with noisy labels (Natarajan, Dhillon, Ravikumar, & Tewari, 2013), is especially common in biomedicine: for instance, it is known that a 10% of the Alzheimer's diagnostics are wrong, as a reliable diagnosis requires a post-mortem analysis (Chang & Silverman, 2004; Eastley, Wilcock, Ames, Burns, & OBrien, 2005; Ryan, 1994). This problem is not specific to the proposed methodology, but is on the contrary common to all data mining and machine learning algorithms; even the best DSS system can be wrong, and the user should be aware of this. The methodology here discussed can nevertheless help tackling such situations: having knowledge about how the underlying classification algorithm works can help the user to be better aware of its limitations, enabling the confrontation of his or her own knowledge with the machines one. This has here been illustrated through the fourth real-world data set, which suffered from an overfitting in the training phase. Conversely, the user should be aware that results may depend on the distance metric used, which may favour one dimension over another and thus give biased results, if an appropriate weighting is not applied by the user.

To conclude, while this contribution has only explored low dimensional problems, the proposed methodology can in principle be applied to higher numbers of dimensions. This would require further inquiries, for instance on how to conceptually combine changes coming from a large number of features. On the other hand, this will open the door to the study of large black-box models, as for instance deep learning ones.

Conflict of Interest

The authors declare no conflict of interest.

Credit authorship contribution statement

Seddik Belkoura: Formal analysis, Writing - original draft, Writing - review & editing. **Massimiliano Zanin:** Conceptualization, Writing - review & editing. **Antonio LaTorre:** Formal analysis, Writing - original draft, Writing - review & editing.

Acknowledgement

A.L. acknowledges funding from the Spanish Ministry of Science and Innovation (TIN2017-83132-C2-2-R) and Universidad Politécnica de Madrid (PINV-18-XEOGHQ-19-4QTEBP).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M. et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint: 1603.04467

Abdalla, F., Boder, J., Buhmeida, A., Hashmi, H., Elzagheid, A., & COLLAN, Y. (2008). Nuclear morphometry in fnabs of breast disease in libyans. *Anticancer Research, 28*(6B), 3985–3989.

Allahyari, H., & Lavesson, N. (2011). User-oriented assessment of classification model understandability. In *Proceedings of the 11th scandinavian conference on artificial intelligence* (pp. 11–19). IOS Press.

Backhaus, A., & Seiffert, U. (2014). Classification in high-dimensional spectral data: Accuracy vs. interpretability vs. model size. *Neurocomputing, 131*, 15–22. doi:10.1016/j.neucom.2013.09.048.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., & MÃžller, K.-R. (2010). How to explain individual classification decisions. *Journal of Machine Learning Research, 11*(Jun), 1803–1831.

Berner, E. S. (2007). *Clinical decision support systems*. Springer.

Bibal, A., & Frenay, B. (2016). Interpretability of machine learning models and representations: an introduction. In *24th european symposium on artificial neural networks, computational intelligence and machine learning* (pp. 77–82).

Biran, O., & Cotton, C. V. (2017). Explanation and justification in machine learning : A survey. *IJCAI-17 workshop on explainable AI (XAI)*.

Boriah, S., Chandola, V., & Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 siam international conference on data mining* (pp. 243–254). SIAM.

Boulton, R. B., Singleton, V. L., Bisson, L. F., & Kunkee, R. E. (2013). *Principles and practices of winemaking*. Springer Science & Business Media.

Bratko, I. (1997). Machine learning: Between accuracy and interpretability. In *Learning, networks and statistics* (pp. 163–177). Springer.

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

Breiman, L. (2001b). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science, 16*(3), 199–231. doi:10.1214/ss/1009213726.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth & Brooks.

Bussone, A., Stumpf, S., & O'Sullivan, D. (2015). The role of explanations on trust and reliance in clinical decision support systems. In *2015 international conference on healthcare informatics* (pp. 160–169). IEEE.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. In *KDD '15* (pp. 1721–1730). New York, NY, USA: ACM. doi:10.1145/2783258.2788613.

Chang, C. Y., & Silverman, D. H. (2004). Accuracy of early diagnosis and its impact on the management and course of alzheimers disease. *Expert Review of Molecular Diagnostics, 4*(1), 63–69.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems, 47*(4), 547–553.

[dataset] Lung Cancer data set (2019). UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/Lung+Cancer.

[dataset] Mushroom data set (2019). UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/Mushroom.

[dataset] Wine Quality data set (2019). UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/Wine+Quality.

[dataset] Wisconsin Breat Cancer data set (2017). UCI machine learning repository. http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29.

Davis, L. (1991). *Handbook of genetic algorithms*. Van Nostrand Reinhold, New York.

Dong, G., & Li, J. (1999). Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 43–52). ACM.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv,.

Eastley, R., Wilcock, G. K., Ames, D., Burns, A., & OBrien, J. (2005). Assessment of dementia. *Dementia*, 38–44.

Freitas, A. A. (2014). Comprehensible classification models: A position paper. *SIGKDD Explor. Newsl., 15*(1), 1–10. doi:10.1145/2594473.2594475.

Geng, L., & Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR), 38*(3), 9.

Gerretzen, J., Szymaska, E., Bart, J., Davies, A. N., van Manen, H.-J., van den Heuvel, E. R., … Buydens, L. M. (2016). Boosting model performance and interpretation by entangling preprocessing selection and variable selection. *Analytica Chimica Acta, 938*, 44–52. doi:10.1016/j.aca.2016.08.022.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th international conference on data science and advanced analytics (DSAA)* (pp. 80–89). doi:10.1109/DSAA.2018.00018.

Guo, C., & Berkhahn, F. (2016). Entity embeddings of categorical variables. arXiv preprint: 1604.06737

Hong, Z.-Q., & Yang, J.-Y. (1991). Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recognition, 24*(4), 317–324.

Hwang, C.-R. (1988). Simulated annealing: theory and applications. *Acta Applicandae Mathematicae, 12*(1), 108–111.

Ishikawa, M. (2000). Rule extraction by successive regularization. *Neural Networks, 13*(10), 1171–1183.

Kim, B., Glassman, E., Johnson, B., & Shah, J. (2015). iBCM: Interactive Bayesian case model empowering humans via intuitive interaction. *Technical Report*. MIT.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on artificial intelligence-volume 2* (pp. 1137–1143). Morgan Kaufmann Publishers Inc..

Kononenko, I. (1993). Inductive and bayesian learning in medical diagnosis. *Applied Artificial Intelligence an International Journal, 7*(4), 317–337.

Lagro, J., van de Pol, M. H., Laan, A., Huijbregts-Verheyden, F. J., Fluit, L. C., & Rikkert, M. G. O. (2014). A randomized controlled trial on teaching geriatric medical decision making and cost consciousness with the serious game geriatrix. *Journal of the American Medical Directors Association, 15*(12), 957–e1.

Lavrač, N., Džeroski, S., Pirnat, V., & Križman, V. (1993). The utility of background knowledge in learning medical diagnostic rules. *Applied Artificial Intelligence an International Journal, 7*(3), 273–293.

Letham, B., Rudin, C., McCormick, T., & Madigan, D. (2013). An interpretable stroke prediction model using rules and bayesian analysis. In *27th AAAI conference on artificial intelligence: WS-13-17* (pp. 65–67). AI Access Foundation.

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems (NIPS)* (pp. 2177–2185).

Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., & Chen, M. (2014). Medical image classification with convolutional neural network. In *13th international conference on control automation robotics & vision (ICARCV)* (pp. 844–848). IEEE.

Lipton, Z. C. (2016). The mythos of model interpretability. In *2016 ICML workshop on human interpretability in machine learning (WHI 2016)* (pp. 96–100).

Lo, S.-C. B., Chan, H.-P., Lin, J.-S., Li, H., Freedman, M. T., & Mun, S. K. (1995). Artificial convolution neural network for medical image pattern recognition. *Neural Networks, 8*(7–8), 1201–1214.

Lonvaud-Funel, A. (1999). Lactic acid bacteria in the quality improvement and depreciation of wine. In *Lactic acid bacteria: Genetics, metabolism and applications* (pp. 317–331). Springer, Dordrecht.

Mangasarian, O. L., Street, W. N., & Wolberg, W. H. (1995). Breast cancer diagnosis and prognosis via linear programming. *Operations Research, 43*(4), 570–577.

Miller, R. A. (1994). Medical diagnostic decision support systemspast, present, and future. *Journal of the American Medical Informatics Association, 1*(1), 8–27.

Miller, T. (2019). Explanation in artificial intelligence: insights from the social sciences. *Artificial Intelligence, 267*, 1–38. doi:10.1016/j.artint.2018.07.007.

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Fourth international conference on 3d vision (3DV)* (pp. 565–571). IEEE.

Mittelstadt, B., Russell, C., & Wachter, S. (2019). Explaining explanations in AI. In *Proceedings of the conference on fairness, accountability, and transparency*. In *FAT* '19* (pp. 279–288). New York, NY, USA: ACM. doi:10.1145/3287560.3287574.

Montavon, G., Samek, W., & Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing, 73*, 1–15. doi:10.1016/j.dsp.2017.10.011.

Musen, M. A., Middleton, B., & Greenes, R. A. (2014). Clinical decision-support systems. In *Biomedical informatics* (pp. 643–674). Springer.

Narasimha, A., Vasavi, B., & Kumar, H. M. (2013). Significance of nuclear morphometry in benign and malignant breast aspirates. *International Journal of Applied and Basic Medical Research, 3*(1), 22.

Narayanan, M., Chen, E., He, J., Kim, B., Gershman, S., & Doshi-Velez, F. (2018). An evaluation of the human-interpretability of explanation. arXiv abs/1902.00006.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., & Tewari, A. (2013). Learning with noisy labels. In *Advances in neural information processing systems* (pp. 1196–1204).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*(Oct), 2825–2830.

Potes Ruiz, P., Kamsu Foguem, B., & Grabot, B. (2014). Generating knowledge in maintenance from experience feedback. *Knowledge-Based Systems, 68*, 4–20.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106.

Rampasek, L., & Goldenberg, A. (2016). Tensorflow: Biologys gateway to deep learning? *Cell Systems, 2*(1), 12–14.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016a). Model-agnostic interpretability of machine learning. In *2016 ICML workshop on human interpretability in machine learning (WHI)* (pp. 91–95).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016b). Nothing else matters: model-agnostic explanations by identifying prediction invariance. arXiv preprint: 1611.05817

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016c). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. In *KDD '16* (pp. 1135–1144). New York, NY, USA: ACM. doi:10.1145/2939672.2939778.

Ryan, D. (1994). Misdiagnosis in dementia: comparisons of diagnostic error rate and range of hospital investigation according to medical speciality. *International Journal of Geriatric Psychiatry, 9*(2), 141–147.

Schlimmer, J. C. (1987). Concept acquisition through representational adjustment,.

Schwabacher, M., Langley, P., & Norvig, P. (2001). Discovering communicable scientific knowledge from spatio-temporal data. In *Proc. 18th int. conf. on machine learning (ICML-2001)* (pp. 489–496). Morgan Kaufmann.

Scott, P. J., Brown, A. W., Adedeji, T., Wyatt, J. C., Georgiou, A., Eisenstein, E. L., & Friedman, C. P. (2019). A review of measurement practice in studies of clinical decision support systems 1998–2017. *Journal of American Medical Informatics Association*.

Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumor diagnosis. In *Is&t/SPIE's symposium on electronic imaging: Science and technology* (pp. 861–870). International Society for Optics and Photonics.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics, 9*(1), 307.

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics, 8*(1), 25.

Urruticoechea, A., Alemany, R., Balart, J., Villanueva, A., Vinals, F., & Capella, G. (2010). Recent advances in cancer therapy: An overview. *Current Pharmaceutical Design, 16*(1), 3–10.

Van Assche, A., & Blockeel, H. (2008). Seeing the forest through the trees: Learning a comprehensible model from a first order ensemble. In *Proceedings of the 17th international conference on inductive logic programming*. In *ILP'07* (pp. 269–279). Berlin, Heidelberg: Springer-Verlag.

Verikas, A., Gelzinis, A., & Bacauskiene, M. (2011). Mining data with random forests: A survey and results of new tests. *Pattern Recognition, 44*(2), 330–349.

Vilela-Moura, A., Schuller, D., Mendes-Faia, A., Silva, R. D., Chaves, S. R., Sousa, M. J., & Côrte-Real, M. (2011). The impact of acetate metabolism on yeast fermentative performance and wine quality: Reduction of volatile acidity of grape musts and wines. *Applied Microbiology and Biotechnology, 89*(2), 271–280.

Wittekind, C., & Schulte, E. (1987). Computerized morphometric image analysis of cytologic nuclear parameters in breast cancer.. *Analytical and Quantitative Cytology and Histology, 9*(6), 480–484.

Wu, M., & Burstein, D. E. (2004). Fine needle aspiration. *Cancer Investigation, 22*(4), 620–628.

Yoon, S. W., Velasquez, J. D., Partridge, B., & Nof, S. Y. (2008). Transportation security decision support system for emergency response: A training prototype. *Decision Support Systems, 46*(1), 139–148.