Goethals *et al. Journal of Big Data* (2022) 9:30
https://doi.org/10.1186/s40537-022-00579-2

Journal of Big Data

RESEARCH Open Access

The non-linear nature of the cost of comprehensibility

Sofie Goethals[1*], David Martens[1] and Theodoros Evgeniou[2]

*Correspondence:
sofie.goethals@uantwerpen.be
[1] Department of Engineering Management, University of Antwerp, Antwerp, Belgium
Full list of author information is available at the end of the article

Abstract

A key challenge in Artificial Intelligence (AI) has been the potential trade-off between the accuracy and comprehensibility of machine learning models, as that also relates to their safe and trusted adoption. While there has been a lot of talk about this trade-off, there is no systematic study that assesses to what extent it exists, how often it occurs, and for what types of datasets. Based on the analysis of 90 benchmark classification datasets, we find that this trade-off exists for most (69%) of the datasets, but that somewhat surprisingly for the majority of cases it is rather small while for only a few it is very large. Comprehensibility can be enhanced by adding yet another algorithmic step, that of surrogate modelling using so-called 'explainable' models. Such models can improve the accuracy-comprehensibility trade-off, especially in cases where the black box was initially better. Finally, we find that dataset characteristics related to the complexity required to model the dataset, and the level of noise, can significantly explain this trade-off and thus the cost of comprehensibility. These insights lead to specific guidelines on how and when to apply AI algorithms when comprehensibility is required.

Keywords: Explainable Artificial Intelligence, Accuracy-comprehensibility trade-off, Cost of comprehensibility

Introduction

In 2019, a series of tweets went viral where a tech entrepreneur was complaining about the fact that Apple Card offered him twenty times the credit limit that it offered to his wife, although they had shared assets. After complaining to Apple representatives, he got the reply: "I don't know why, but I swear we're not discriminating, IT'S JUST THE ALGORITHM" [1, 2]. Apple co-founder Steve Wozniak replied that the same thing happened to him and his wife and added [3]: "Hard to get to a human for a correction though. It's big tech in 2019." These complaints led to a formal investigation into the potential sexist credit scoring by Apple Card [1, 2]. This example shows how predictive modelling is facing major challenges due to its inability to explain its decisions, which often stems from the use of complicated models. But why is everyone using these kinds

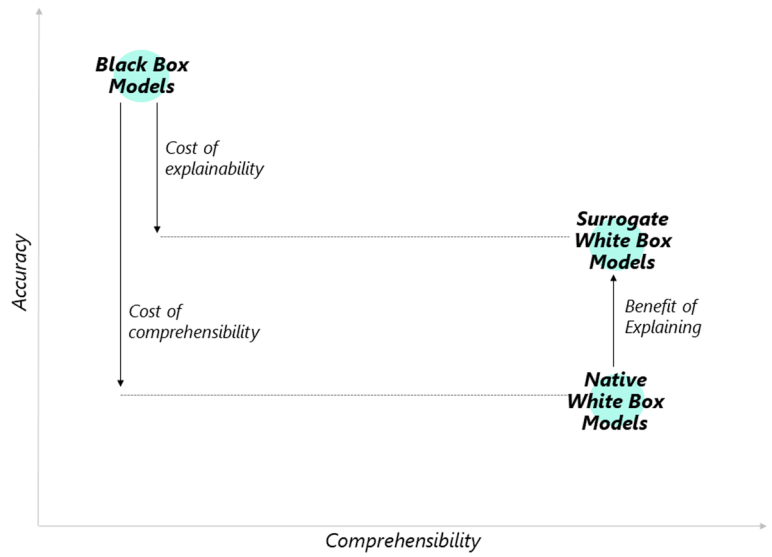

Fig. 1 Definitions of the Cost of Comprehensibility, the Cost of Explainability and the Benefit of Explaining

of models? It is often claimed that they have a higher performance than more simple models, but is this always true? How often is it the case and to what extent?

This trade-off between accuracy and comprehensibility is argubaly one of the important debates in Artificial Intelligence (AI)[1] [4, 5]. This trade-off can either limit the performance of AI, if accuracy is lost due to comprehensibility restrictions (for example imposed by regulators) [6, 7], or hurt AI adoption, if user trust is lost due to opaqueness [8]. The Apple Card example shows that companies may use black box models to achieve higher predictive performance, but with the risk of being unable to explain their AI decisions to users or regulators. However, while there has been a lot of research mentioning this trade-off, with most claiming there is one [5, 8–10] and others contradicting this [11, 12], there is no systematic study that assesses to what extent there indeed exists a trade-off and for what types of datasets.

The goal of this paper is to provide such a systematic study. We focus on tabular datasets as we believe that for these datasets the trade-off would be less clear - and possibly smaller than expected. Deep learning models, which are models composed of multiple layers to learn representations of data with multiple levels of abstraction [13] and can thus be considered as black box models, perform very well for classification on homogenous data such as image, audio or text but they not necessarily outperform other machine learning techniques on tabular datasets [14–16].

Based on the analysis of 90 benchmark datasets across different domains, we study the nature of the differences between the accuracies among a number of widely used a) opaque ("black box") models, b) comprehensible ("white box") models, and c) surrogate models used to develop a comprehensible surrogate of the opaque ones. We call the difference between (a) and (b) "Cost of Comprehensibility", that between (a) and (c) "Cost

[1] We focus on prediction models trained on data using machine learning algorithms.

of Explainability", and that between (b) and (c) the "Benefit of Explaining" (Fig. 1).[2] Our main findings are: first, there is indeed a trade-off but somewhat surprisingly it appears to be highly non-linear across datasets. Both costs are relatively small for most datasets, but very large for a few. Second, there are datasets for which the comprehensible models perform as well or better than the black box models, supporting that one should not forgo trying comprehensible models [17]. We call these datasets "comprehensible datasets", as opposed to datasets where the black box is strictly better which we call "opaque datasets". Understanding what makes a dataset "opaque" vs "comprehensible" and more so, given the non-linearities observed, what makes the costs very high (positive or negative) is a challenging question as it relates to understanding the data generation processes themselves (e.g., the "nature" of the data and problem at hand). We discuss initial results indicating that some of the main differences between opaque and comprehensible datasets are about their inherent complexity as well as the level of noise in the data. The results indicate that reporting some simple characteristics of a dataset can provide clues, for example to users or regulators, about the potential accuracy and comprehensibility trade-off. To summarize, the contributions of our paper are threefold:

- A benchmark study comparing state-of-the-art white box and black box algorithms on 90 tabular datasets, and assessing their difference in performance;
- An analysis of whether surrogate modelling could improve any trade-off between comprehensibility and accuracy;
- Insights in how dataset properties could predict the nature/size of the trade-off we study.

Background and setup of the study

What is comprehensibility?

Comprehensibility refers to the ability to represent a machine learning model and explain its outcomes in terms that are understandable to a human [18]. The lack of comprehensibility in black box models is one of their main pitfalls, as their inner working is hidden to the users preventing them from verifying whether the reasoning of the system is, for example, aligned with restrictions or preferences of how decisions are made [19–21]. Furthermore, it is easier to debug comprehensible models or to detect bias in them, and it also increases social acceptance [22]. In general, there are two ways to provide comprehensibility in machine learning [22, 23]: intrinsic comprehensibility is acquired when using models that are comprehensible by nature due to their simple structure, which are the so-called "white box" models [23], while post-hoc comprehensibility aims to explain the predictions without accessing the model's inner structure [23], as provided by LIME [24], SHAP [25] or counterfactual explanations [26]. Another distinction that can be made is between global comprehensibility and local comprehensibility. Global comprehensibility allows to understand the whole logic of a model and follow the reasoning that leads to every possible outcome, where for local comprehensibility it is possible to understand the reasons for a specific decision [22, 27]. Comprehensibility is very

[2] We note that the terms "interpretability", "comprehensibility" and "explainability" have also been used in different ways in the literature.

Table 1 Models that are used in other benchmark studies

ML algorithms	Count	Olson et al. [43]	Fernandez et al. [44]	Zhang et al. [45]	Lessman et al. [46]	Mayr et al. [47]	Lorena et al. [48]	Macia et al. [49]
Random Forest	7	✓	✓ (1)	✓ (1)	✓	✓ (3)	✓ (1)	✓
Bayesian	7	✓	✓	✓	✓	✓	✓	✓
SVM	7	✓	✓ (2)	✓(2)	✓	✓ (2)	✓ (2)	✓
LR	6	✓	✓	✓	✓		✓	✓
Nearest Neighbor	6	✓	✓	✓	✓	✓	✓	
Neural networks	5		✓ (3)	✓(3)	✓	✓(1)	✓	
Decision tree	4	✓	✓ (4)	✓				✓
Boosting	3	✓	✓	✓(1)				
Discriminant analysis	2		✓		✓			
Bagging	1		✓					
Rule-based	1				✓			

Symbol ✓ indicates that this kind of model was used in the study, and the numbers between brackets indicate the rank of the model (if this was included in the study)

difficult to measure due to its subjective nature. Some compare the comprehensibility of models using user-based surveys [28, 29] while others based on mathematical heuristics [9], typically the size of the model (e.g., number of rules for a rule learner, number of nodes for a decision tree, or number of variables for a linear model) [30–33]. Very deep decision trees, for example, can be considered as less comprehensible than a compact neural network [34]. We use the latter, heuristic approach to measure comprehensibility due to its objectivity and scalability.

What are intrinsically comprehensible models?

In line with the literature, we consider small decision trees, rule sets and linear models as comprehensible or "white box" models [8, 22, 27, 35]. We limit the size of these models during training in order for them to be comprehensible. We opted for seven as the size limit for comprehensibility, based on cognitive load theory [36]. According to this theory, the span of absolute judgement and the span of short-term memory pose severe limitations on the amount of information that humans can receive and process correctly, with seven being the typically considered maximum size in both cases [36]. We consider larger decision trees[3], rule sets and linear models as "black box" ones. We also consider three other machine learning methods in the list of black boxes we test: neural networks, random forests and nonlinear support vector machines. It is generally agreed upon that these algorithms are not comprehensible as their line of reasoning cannot be followed by human users. We base this choice of black box models on the results of benchmark studies in the literature, where these often are among the best performing ones, as can be

[3] A decision tree of eight nodes is arguably not a black box model, and may be in a "grey zone" of comprehensibility. For this reason, in our experiments we focus on the very large and small trees, rule sets and linear models, defined as those with size larger than 50 or smaller than 8 (in number of nodes/rules/coefficients) as it is a general assumption in the literature that smaller decision trees are more comprehensible than larger ones due to the cognitive size limit [9, 28, 37, 38]. This focus ensures that our findings are applicable to all applications and end users, because of the arbitrariness to consider models with size between 8 and 50 as black box, which actually depends on the application and end user.

seen in Table 1.[4] Comparing all possible models available is of course infeasible, which is a practical limitation of such a study. All the papers mentioned in Table 1 compare different machine learning models but none investigate the difference in performance between the best black box model and the best white box model, nor whether this can be linked to any dataset properties. Many papers claim that black box models will always have a better performance, or on the contrary that simpler models work equally well [11, 12], but a large-scale study about the difference of performance is missing.[5]

Surrogate modelling

A common practice is to mimic the predictions of a black box with a global white box surrogate model, in order to improve the accuracy while remaining comprehensible [50, 51]. The typical process is to first build a black box model using the available training data, and then build a comprehensible model by training a white box model using the predictions of the black box instead of the original training data. This process is called surrogate modelling [22], oracle coaching [52, 53], or rule extraction in case the white box model is a decision tree or rule set [6, 54]. A key metric of the quality of the surrogate model is *fidelity*, which measures how well the predictions of the surrogate model match those of the black box [55]. The most common goal of this kind of modelling is to use the surrogate model to explain the black box model, while still using the black box to make predictions. This requires of course that the surrogate model is (1) more comprehensible than the black box model and (2) sufficiently explains the predictions made (high fidelity).

One can also use the surrogate model instead of the black box to make predictions, in order to improve the performance one could achieve using only comprehensible models. A possible reason why this approach can work, instead of just training a white box model directly using the training data, can be that the black box model may filter out noise or anomalies that are present in the original training data [53, 56]. In this case, a comprehensible model mimicking a black box may be more accurate than a comprehensible model trained on the original data, as shown in some previous work [51–53]. Therefore, we also investigate whether surrogate modelling can lead to better performing comprehensible models and, as such, improve the trade-off we study. Specifically, for each dataset we train a white box on the predictions of the *best* performing black box for that dataset. We call this a *surrogate white box model* as opposed to a comprehensible model trained on the training dataset which we call a *native white box model*—see Fig. 1.

Dataset properties

Finally, we study whether there are simple (standard) properties of a dataset that may determine whether it is opaque (the best black box model outperforms the best white

[4] We do not include k-nearest neighbors and Bayesian networks, which are also used frequently in other benchmark studies, as it is debatable whether they can be considered as comprehensible models. K-nearest neighbors lacks global model comprehensibility as there are is no global model structure learned [22] and in Bayesian networks, it is not easy to interpret the mapping implicit in the network or do other data inference tasks, as the reasoning method is not necessarily aligned with human reasoning [40, 41].

[5] Besides white box and black box models, some researchers also mention the existence of "grey box" models, which are defined as aiming to develop an ensemble of black and white box models and acquire the benefits of both by being nearly as accurate as black box models but more comprehensible [23]. As the literature is not conclusive on whether grey boxes are always as comprehensible as white box models [23, 32, 42], we will focus only on the trade-off between black box and white box models in this study.

box) or comprehensible (the reverse happens). We use a standard toolbox, Alcobaba [57], which automatically extracts numerous characteristics ("meta-features") for any given dataset. We consider four types of dataset characteristics from this toolbox: general ones, which capture basic information such as the number of instances or the number of attributes [58]; statistical ones, which capture information about the data distribution such as the number of outliers, variance, skewness, etc. [58]; information-theoretic ones, which capture characteristics such as the joint entropy, class entropy, class concentration, etc. [58]; and so-called complexity related ones, which, for example in the case of a classification problem estimate the difficulty in separating the data into their classes [59].[6] We opt for using a standard toolbox and set of dataset characteristics to make this analysis general, easily reproducible and simple to use in practice.

Materials and methods

Materials

We use a large benchmark study to compare the algorithms on different tabular datasets. Benchmark comparisons are usually developed over a few, typically standard data sets, as a machine learning method might perform well on some of the datasets but not generalize to a broader range of problems [43].

To perform our experiments, we use all the binary classification datasets from the Penn Machine Learning Benchmark (PMLB) suite [43]. This is a dataset suite that is publicly available on Github,[7] which consists both of real-world and simulated benchmark datasets to evaluate supervised classification methods. It is compiled from a wide range of existing ML benchmark suites such as KEEL, Kaggle, the UCI ML repository and the meta-learning benchmark. At this moment, PMLB consists of 162 classification datasets and 122 regression datasets. We focus on the binary classification datasets which amount to 90 datasets in total.

Some preprocessing was already done by the compilers of this benchmark suite. All the datasets were preprocessed to follow a standard row-column format and all the categorical and features with non-numerical encodings were replaced with numerical equivalents. All datasets with missing data were excluded, to avoid the impact of imposing a specific data imputation method. The used datasets are shown in Table 3.

Methods

Our methodology is shown in Fig. 2. For each dataset we create a training and test set, using 75% of the data for training and 25% for testing. Both the training and the test set are scaled according to the parameters of the training set with Sklearn's MinMaxScaler.[8] This estimator scales each feature individually so that it is between zero and one on the training set. We also use a stratified split to make sure that enough labels are present for the training phase. GridSearchCV from *Sklearn*[9] is used with its default 5-fold cross validation to tune the hyperparameters of every model. The dataset is divided in five

[6] See Supplementary Information material.

[7] https://github.com/EpistasisLab/pmlb.

[8] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html.

[9] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

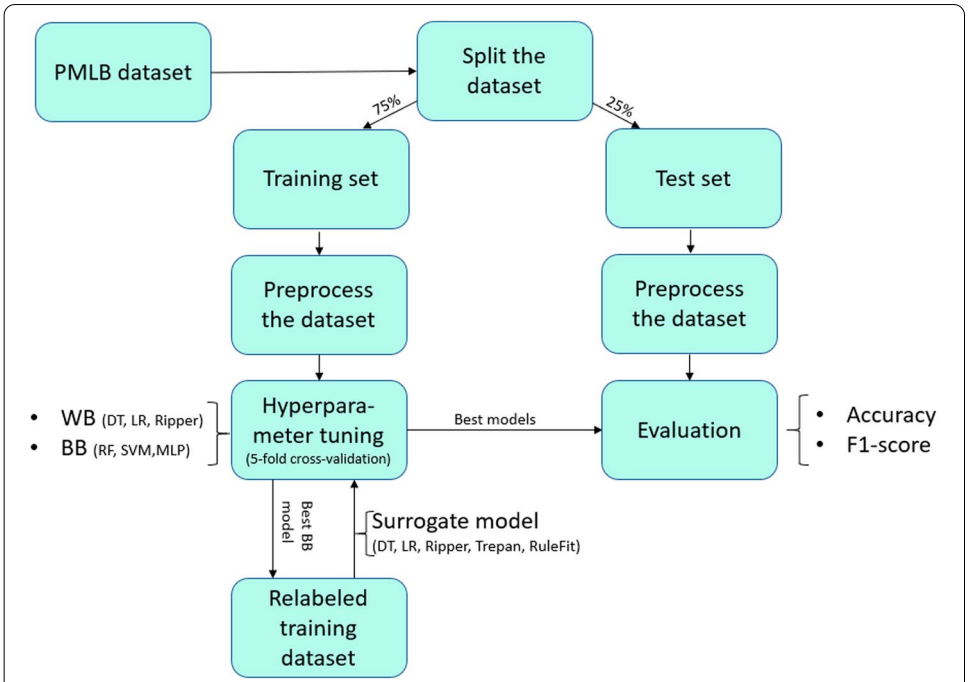

Fig. 2 Methodology

folds, where each time another fold is taken as the validation set. GridSearchCV then performs an exhaustive search over a specified hyperparameter grid, which is reported in the Sects. "Black Box Models", for each modelling technique, and then checks on the validation set which parameter settings performed best. By doing this five times, instead of just using one validation set, we get a more accurate representation of how the model behaves on unseen data, and we are not reliant on the data we used as the validation set. We select the best hyperparameter values for each modelling technique based on this tuning. Moreover, for each dataset we also select the best surrogate model. We do this by creating a new training set, which is a copy of the original training set but with as labels the predictions of the best black box model, based on the cross-validation performance. The surrogate model is trained on this relabeled training set and can be any of the original white box models, as well as Trepan or RuleFit. The final performance of all the models(black box, white box and surrogate) is evaluated on the test set based on two metrics: accuracy and f1-score. The difference in the test set performance among the different models is shown in Fig. 3. For each dataset we select the best black box, the best white box and the best surrogate, based on their performance on the test set.[10] In our aggregate analyses, we compare the test performances of these across all datasets.

[10] Note that using the test data to select the best black and white boxes and then reusing the same data to compare those two across all datasets adds some bias in the results. We opt for this approach (instead of also using, for example, a validation set) as some datasets do not have many observations and we only select among a few (in total six) black boxes and among a few (in total three) white ones, making the bias small. We also verified whether our results are robust when using cross validation to select the best model and note that our results indeed hold (e.g., still for 68.89% of the datasets, the best black box model outperforms the best white box model).

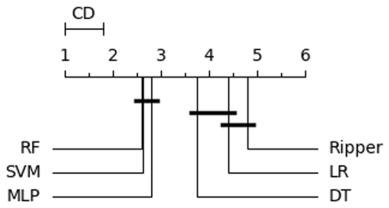

Fig. 3 Critical difference diagram of the comparison of classifiers. Models that are not connected with a bold line have a significant difference in performance (at a 5% level with the Nemenyi test)

Black box models

We use three state-of-the-art black box models: neural networks, random forests and nonlinear support vector machines [39, 60]. As noted below, we also include in the list of black boxes the three comprehensible models when their size - after training - is very large.

Random forest We use the RandomForestClassifier[11] from *Sklearn* and use a grid search to tune the number of trees in the forest with values between 10 and 2000 and the number of features to consider when looking for the best split with ('*sqrt*', '*none*').

Support vector machine We use the SVC[12] from *Sklearn* and use a grid search to tune the regularization hyperparameter with values between 0.1 and 1000 and the kernel coefficient with values between 0.0001 and 1. We use the default kernel type of *rbf*.

Neural network We use the MLPClassifier[13] from *Sklearn* and use a grid search to tune the size of the hidden layer. We only test neural networks with one hidden layer. We tune the hidden layer with sizes between 10 and 1000.

Comprehensible models

We use three models that are in general considered to be comprehensible, when their size is constrained. As discussed in the main article, we limit the size of these models to 7 (maximum number of nodes for trees, rules for rule based systems, coefficients for logistic regression). We also train these models without constraining their size. In this case, when their size after training is very large, with more than 50 elements,we consider them as part of the black boxes in our analysis.

Decision tree We use the DecisionTreeClassifier[14] from *Sklearn*. We use a grid search to tune the function to measure the quality of the split (gini, entropy), tune the maximal depth between 2 and 30 and tune the minimum number of samples in a leaf (2,4). We tune the maximal amount of leaf nodes between 2 and 7 for the constrained cases (white boxes) and between 2 and 1000 for the unconstrained ones (black boxes).

Logistic Regression We use the LogisticRegression[15] from *Sklearn*. We use $l2$ regularization and the liblinear solver. We use a grid search to tune the regularization parameter values between 0.0001 and 1000.

[11] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[12] https://scikit-learn.org/stable/modules/svm.html.

[13] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.

[14] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

[15] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

Ripper We use a rule learning algorithm, based on sequential covering. This method repeatedly learns a single rule to create a rule list that covers the entire dataset rule by rule [22]. RIPPER (Repeated Incremental Pruning to produce Error Reduction), which was introduced by Cohen in 1995 is a variant of this algorithm [61]. We use the Python implementation of Ripper hosted on Github.[16]

Surrogate models

We use the three comprehensible models above but this time we train them on the predictions of the best performing black box instead of using the training data. We also include Trepan [54], which is used for rule extraction based surrogate modeling, and RuleFit [62], which is based on an underlying Random Forest model. Again, we limit the size of the comprehensible models to 7.

Trepan We use the Python package Skater to implement TreeSurrogates,[17] which is based on [54]. The base estimator (oracle) can be any supervised learning model. The white box model has the form of a decision tree and can be trained on the decision boundaries learned by the oracle. We use the same hyperparameter settings to tune the decision trees from Trepan as for the DecisionTreeClassifier.

RuleFit The RuleFit algorithm learns sparse linear models that include automatically detected interaction effects in the form of decision rules [62]. The interpretation is the same as for normal linear models but now some of the features are derived from decision rules. We use the Python implementation of RuleFit hosted on Github.[18]

Results

First, we address the cost of comprehensibility, by testing whether native white and black box models have a significant difference in performance. To assess this cost, we use both the models' *f1-score* and *accuracy*.[19] The figures for the latter are reported in Fig. 6. We first compare all the classifiers using the Friedman test[20] [63] to identify whether there are any significant differences between the different models, and then the post-hoc Nemenyi test [64] to identify significant pairwise differences.[21] The null hypothesis of the Friedman test is rejected with a p-value of $2.43 \cdot e^{-25}$ (a value with the same order of magnitude when using accuracy instead of f1-scores). This means that there are significant differences among some groups of algorithms. We use the post-hoc Nemenyi test to perform all possible pairwise comparisons [65]. The results are shown in the critical difference diagram[22] in Fig. 3. The performance of the black box models (RF, MLP, SVM) is significantly better than the performance of the white box models (DT, LR, Ripper), already confirming that, overall, the cost of comprehensibility indeed exists.

[16] Imoscovitz. Ripper Python package. url: https://github.com/imoscovitz/wittgenstein.

[17] A. Kramer et al Skater Python package. url: https://github.com/oracle/Skater.

[18] Molnar. RuleFit Python package. url:https://github.com/christophM/rulefit.

[19] We include the results with f1-score to account for imbalance issues that could bias our results.

[20] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.friedmanchisquare.html.

[21] We cannot just use a pairwise comparison because this would inflate the probability of a type I error. The Friedman test is the non-parametric equivalent to the repeated-measures ANOVA [63].

[22] These diagrams were created with the Orange Data Mining Library [66].

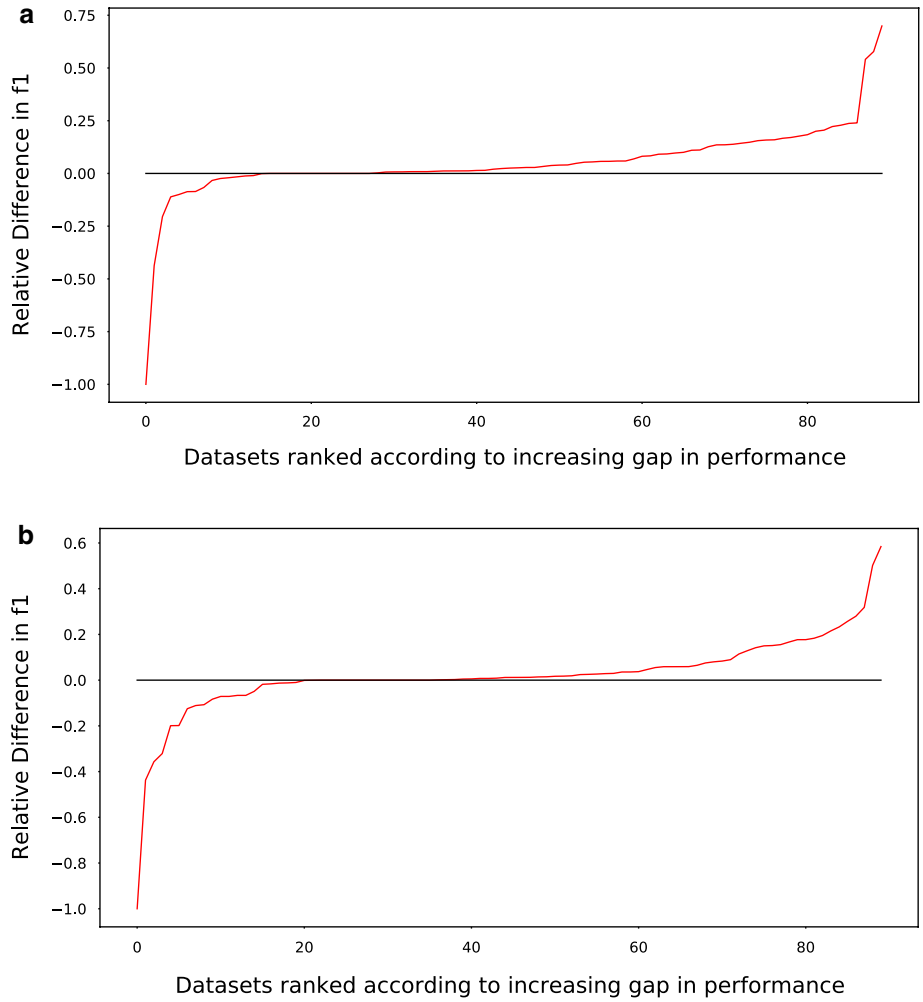

Fig. 4 Comparing black box and white box models. For both plots, the datasets are ordered according to the gap in f1-score between the best black box and the best native (left figure) or surrogate (right) white box model (right). The y-axis measures the relative difference in the f1-score, defined as the ratio of the difference between the black and white box f1-scores divided by that of the best model

The cost of comprehensibility

Having established that the cost of comprehensibility exists, we study how large it is across datasets. As discussed, for each dataset we select the best black and white boxes and measure their relative difference in performance - namely, the cost of comprehensibility. Figure 4a shows the results across all datasets when we order them according to this cost. This figure reveals a somewhat surprising result: this cost is highly non-linear (e.g., the plot is a sigmoid instead of being closer to a straight line). For most datasets the accuracy-comprehensibility trade-off is low, only for a few it is very high (right) and for a few it is very "negative" indicating that comprehensible models largely outperform the black box ones for these datasets (left). Yet, for 68.89% of the datasets the best black box model outperforms the best white box model, reconfirming the overall existence of the cost of comprehensibility. The results for accuracy can be seen in Fig. 7a.

Can surrogate modeling improve the accuracy-comprehensibility trade-off?

We next investigate whether surrogate modelling can improve the performance of the (native) comprehensible models. For all datasets we generate the best black box and the best (native) white box trained on the training data, and then we also train a surrogate model mimicking the best black box one - what we previously called a surrogate white box. We compare the performance of these three types of models across all datasets in Fig. 5. As indicated in Fig. 5a, surrogate modelling does improve accuracy slightly relative to native white box models, on average across all datasets. We term this improvement the "Benefit of Explaining", a benefit in terms of improved predictive accuracy. Based on the Wilcoxon Signed Rank test[23] [63], used to compare classifiers across several datasets, we can reject the hypothesis that the native and surrogate white boxes perform equally well (p-value 0.003) – the latter performing on average better. The result for accuracy can beseen in Fig. 8.

We perform the same analysis, but this time for two different types of datasets: those for which the best performing model is a black box, what we termed opaque datasets, and those for which white boxes perform at least as well as or better than black boxes, what we called comprehensible datasets. The results are shown in Fig. 5b, c. Interestingly, in this case the surrogate white box models outperform the native white box models on average across the opaque datasets (Wilcoxon test p-value of $7.72 \cdot e^{-5}$), while the two are not significantly different for the comprehensible datasets (Wilcoxon test p-value of 0.20). In the latter case there is no need to go through a black box if its performance is not better than that of a native white box [56, 67], as the latter would dominate both in terms of accuracy and comprehensibility. Hence, if one considers only opaque datasets, the use of surrogate modeling can indeed improve the accuracy-comprehensibility trade-off on average.

The cost of explainability

Next, we investigate the difference in performance between the best black box model for each dataset and the best surrogate white box model from that black box - what we call the cost of explainability. Fig. 4b shows the results when we sort all datasets based on this cost. The results are similar to what we observe for the cost of comprehensibility: the difference is small for most datasets, but very large for a few. The results are also in agreement with those in Fig. 7, where we see that the cost of explainability is a bit lower than the cost of comprehensibility (Fig. 7).

Opaque vs. comprehensible datasets

Finally, we study whether the cost of comprehensibility relates to some properties of the dataset. To do so, for each dataset we generate a number of standard dataset properties as discussed above (see also Supplementary Information material), and use them to explain the cost of comprehensibility. Specifically, we run a regression analysis using the generated dataset properties as independent variables with the dependent variable being the difference between the performance of the best black box model and the best native white box model. We used all 90 datasets, hence the number of observations used for the regression was also 90. The variables that are

[23] https://scikit-posthocs.readthedocs.io/en/latest/generated/scikit_posthocs.posthoc_wilcoxon/.

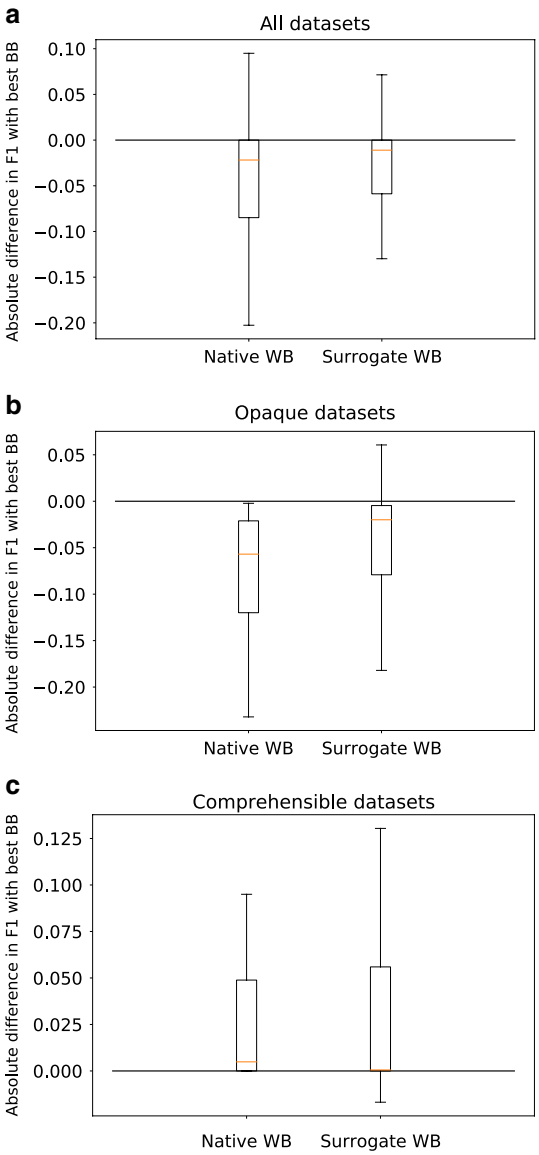

Fig. 5 Comparison across datasets of best black box model for each dataset, surrogate white box model mimicking this best black box, and best native white box model. BB stands for black box and WB for white box. The line at 0 indicates the performance of the best black box model. The y-axis indicates the absolute difference in f1-score from the best black box model

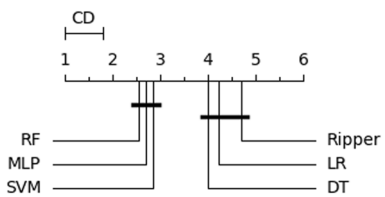

Fig. 6 Critical difference diagram of the comparison of classifiers. Models that are not connected by the bold line have a significant difference in performance (at a 5% level with the Nemenyi test)

Table 2 The dataset properties that are significant when explaining the cost of comprehensibility using a number of standard dataset properties as independent variables in a regression model where the cost is the dependent variable

Variable	MSE	P-value	Coef
EqNumAttr	0.508	$4.57 \cdot e^{-10}$	−0.72e
NsRatio	0.508	$4.57 \cdot e^{-10}$	−0.72e
N3	0.148	0.00191	−0.23e
F1v	0.139	0.00267	0.15
L1	0.089	0.0170	0.12

significant are shown in Table 2. Overall, these results indicate that properties related to the complexity required to model a dataset and the level of noise in a dataset significantly explain the cost. While this is a relatively simple analysis, the results suggest that one may be able to identify or communicate whether there is a potential cost of comprehensibility by simply reporting specific dataset properties.

Specifically, the following five properties are found to be significant. *F1v*, which is the directional-vector Maximum Fisher's discriminant ratio that indicates whether a linear hyperplane can separate most of the data, where lower values means that more data can be separated this way [59]. *L1*, which is a linearity measure that quantifies whether the classes can be linearly separated [58]. Higher values of this attribute indicate more complex problems as they require a non-linear classifier [59]. These properties have a positive coefficient in the regression analysis, which means that all these factors increase the gap between the best black box model and the best white box model. The sign of these coefficients is as expected, namely that for datasets that are more complex to separate linearly, the performance of black box models compared to simple models is on average better.

Two other features, *EqNumAttr* and *NsRatio*, capture information related to the minimum number of attributes necessary to represent the target attribute and the proportion of data that is irrelevant to the problem (level of noise) [58, 68]. We see that these dataset properties have a negative relationship with the size of the cost. Note that when we analyze this result at the level of each individual prediction model, we see that these properties negatively affect both the performance of the black box models and the white box models, but more so for the black box ones. This could be because black box models may pick up more of the noise or use a lot of irrelevant features. Finally, *N3* [59] is a neighbor-based measure that refers to the error rate of the nearest neighbor classifier. Low values of this dataset property indicate that there is a large gap in the class boundary [69]. We see again that this property negatively affects both the performance of the black box models and the white box models [69], and that the effect on the gap depends on how much it affects the performance of each model.

Discussion

Understanding the trade-off between comprehensibility and accuracy can have important implications for regulators as well as companies [70]. Our results indicate that most of the time the trade-off is relatively small, indicating that one should consider native

white box algorithms as a key benchmark. Indeed, given the non-linearities we observe, one would expect that black boxes are used relatively infrequently, even if for the majority of cases they outperform white boxes, as our study indicates that this outperformance is typically relatively small. Some papers in the literature also indicate that for certain datasets simple models work as well as complex ones [11, 12] or that for most datasets the out-performance by black box models will be very small [71], despite the popular belief that more complex models are always better. Of course it depends on the use case and application domain whether this small difference in performance is worth the loss in comprehensibility. Due to social and ethical pressure, insight in when one should opt for a comprehensible model could be a competitive differentiator and drive real business value [70]. Insights in this trade-off could lead to specific guidelines from regulators on how and when to apply AI algorithms when comprehensibility is required.

Our results also show that using surrogate modelling could reduce the cost of comprehensibility, especially for opaque datasets. As we discussed, this may be the case because the black box model in between can filter out noise and anomalies [53, 56]. We also see that simple properties of a dataset could provide insights (for example to a third party such as a user or regulator) in the nature of the trade-off without requiring knowledge of the algorithms tested or the data used. For example, attributes that measure how difficult it is to linearly separate the data are significantly correlated with the size of the gap. Indeed, one would expect that for these datasets black box models might be better in capturing the non-linearities. This can lead to practical tests of the feasibility of using a native white box – and the potential accuracy loss – in a given use case.

Our general findings suggest the following guidelines:

1. Start with white box models.
2. Train additional black box models if: (a) the application allows for a (possibly small) increase in performance at a cost of comprehensibility, and, (b) the level of noise is high and the data requires complex modeling, as indicated by the listed, easy to calculate dataset metrics.
3. If there is a practically important cost of comprehensibility (hence you are dealing with an opaque dataset), apply additional surrogate modeling algorithms.

Finally, we note that in this study we focused on tabular datasets. For other kinds of datasets, the trade-off we study may be different. For example, for image or text data, more flexible models are needed to handle the data complexity [9, 13] and the difference in performance between comprehensible models compared to black box ones such as deep learning is often considered unbridgeable [8].

Appendix

Materials

Datasets

See Table 3.

Table 3 Used datasets

Dataset	# observations	# features	Imbalance
Adult	48842	14	0.27
Agaricus_lepiota	8145	22	0
Analcatdata_aids	50	4	0
Analcatdata_asbestos	83	3	0.01
Analcatdata_bankruptcy	50	6	0
Analcatdata_boxing1	120	3	0.09
Analcatdata_boxing2	132	3	0.01
Analcatdata_creditscore	100	6	0.21
Analcatdata_cyyoung8092	97	10	0.26
Analcatdata_cyyoung9302	92	10	0.34
Analcatdata_fraud	42	11	0.15
Analcatdata_japansolvent	52	9	0
Analcatdata_lawsuit	264	4	0.73
Appendicitis	106	7	0.36
Australian	690	14	0.01
Backache	180	32	0.52
Biomed	209	8	0.08
Breast_cancer_wisconsin	569	30	0.06
Breast_cancer	286	9	0.16
Breast_w	699	9	0.1
Breast	699	10	0.1
BuggyCrx	690	15	0.01
Bupa	345	5	0
Chess	3196	36	0
Churn	5000	20	0.51
Clean1	476	168	0.02
Clean2	6598	168	0.48
Cleve	303	13	0.01
Coil2000	9822	85	0.78
Colic	368	22	0.07
Corral	160	6	0.02
Credit_a	690	15	0.01
Credit_g	1000	20	0.16
crx	690	15	0.01
Diabetes	768	8	0.09
Dis	3772	29	0.94
Flare	1066	10	0.43
GAMETES_Epistasis_2_Way_1000atts_0.4H_EDM_1_EDM_1_1	1600	1000	0
GAMETES_Epistasis_2_Way_20atts_0.1H_EDM_1_1	1600	20	0
GAMETES_Epistasis_2_Way_20atts_0.4H_EDM_1_1	1600	20	0
GAMETES_Epistasis_3_Way_20atts_0.2H_EDM_1_1	1600	20	0
GAMETES_Heterogeneity_20atts_1600_Het_0.4_0.2_50_EDM_2_001	1600	20	0
GAMETES_Heterogeneity_20atts_1600_Het_0.4_0.2_75_EDM_2_001	1600	20	0
German	1000	20	0.16
Glass2	163	9	0

Table 3 (continued)

Dataset	# observations	# features	Imbalance
Haberman	306	3	0.22
Heart_c	303	13	0.01
Heart_h	294	13	0.08
Heart_statlog	270	13	0.01
Hepatitis	155	19	0.34
Hill_Valley_with_noise	1212	100	0
Hill_Valley_without_noise	1212	100	0
Horse_colic	368	22	0.07
House_votes_84	435	16	0.05
Hungarian	294	13	0.08
Hypothyroid	3163	25	0.82
Ionosphere	351	34	0.08
Irish	500	5	0.01
kr_vs_kp	3196	36	0
Labor	57	16	0.09
Lupus	87	3	0.04
Magic	19020	10	0.09
Mofn_3_7_10	1324	10	0.31
Molecular_biology_promoters	106	57	0
Monk1	556	6	0
Monk2	601	6	0.1
Monk3	554	6	0
Mushroom	8124	22	0
Mux6	128	6	0
Parity5	32	5	0
Parity5+5	1124	10	0
Phoneme	5404	5	0.17
Pima	768	8	0.09
Postoperative_patient_data	88	8	0.21
Prnn_crabs	200	7	0
Prnn_synth	250	2	0
Profb	672	9	0.11
Ring	7400	20	0
Saheart	462	9	0.09
Sonar	208	60	0
Spambase	4601	57	0.04
Spect	267	22	0.35
Spectf	349	44	0.21
ThreeOf9	512	9	0
Tic_tac_toe	958	9	0.09
Tokyo1	959	44	0.08
Twonorm	7400	20	0
Vote	435	16	0.05
Wdbc	569	30	0.06
Xd6	973	9	0.11

Dataset properties

For the analysis of the dataset properties, we use the metafeature toolbox of Alcobaba [57], that automatically extracts metafeatures out of the dataset. The metafeatures of

this toolbox are based on those described in [58]. We select the metafeatures out of the groups: general, statistical, info-theory and complexity. The general metafeatures represent the basic information about the dataset. They capture metrics such as the number of instances, attributes, or other information about the predictive attribute [58]. The statistical measures represent information about the data distribution like the number of outliers, the variance, the skewness or the correlation in the data, and others [58]. The information-theoretic measures capture the amount of information present in the data such as the joint entropy, class entropy, class concentration, and others [58]. The last group of measures we include is the group of information-complexity based on [59]. We do not include the clustering, landmarking or model-based metafeatures because they already fit a model to the dataset and extract information from this model. The used dataset properties can be seen in Table 4.[24]

Table 4 Dataset properties used in the analysis

Metafeature name	Description
AttrConc (mean)	Concentration coef. of each pair of distinct attributes
AttrEnt (mean)	Shannon's entropy for each predictive attribute
AttrToInst	The ratio between the number of attributes
C1	The entropy of class proportions
C2	The imbalance ratio
CanCor (mean)	Canonical correlations of data
CatToNum	The ratio between the number of categoric and numeric features
ClassConc (mean)	Concentration coefficient between each attribute and class
ClassEnt	Target attribute Shannon's entropy
ClsCoef	Clustering coefficient
Cor (mean)	The absolute value of the correlation of distinct dataset column pairs
Cov (mean)	The absolute value of the covariance of distinct dataset attribute pairs
Density	Average density of the network
Eigenvalues (mean)	Eigenvalues of covariance matrix from dataset
EqNumAttr	Number of attributes equivalent for a predictive task
F1 (mean)	Maximum Fisher's discriminant ratio
F1v (mean)	Directional-vector maximum Fisher's discriminant ratio
F2 (mean)	Volume of the overlapping region
F3 (mean)	Feature maximum individual efficiency
F4 (mean)	Collective feature efficiency
FreqClass (mean)	Relative frequency of each distinct class
Gmean (mean)	Geometric mean of each attribute
Gravity	Distance between minority and majority classes center of mass
Hmean (mean)	Harmonic mean of each attribute
Hubs (mean)	Hub score
InstToAttr	Ratio between the number of instances and attributes
IqRange (mean)	Interquartile range (IQR) of each attribute
JointEnt (mean)	Joint entropy between each attribute and class
Kurtosis (mean)	Kurtosis of each attribute
L1 (mean)	Sum of error distance by linear programming
L2 (mean)	OVO subsets error rate of linear classifier
L3 (mean)	Non-Linearity of a linear classifier

[24] Based on the list: https://pymfe.readthedocs.io/en/latest/auto_pages/meta_features_description.html.

Table 4 (continued)

Metafeature name	Description
LhTrace	Lawley-Hotelling trace
Lsc	Local set average cardinality
Mad (mean)	Median Absolute Deviation (MAD) adjusted by a factor
Max (mean)	Maximum value from each attribute
Mean (mean)	Mean value of each attribute
Median (mean)	Median value from each attribute
Min (mean)	Minimum value from each attribute
MutInf (mean)	Mutual information between each attribute and target
N1	Fraction of borderline points
N2 (mean)	Ratio of intra and extra class nearest neighbor distance
N3 (mean)	Error rate of the nearest neighbor classifier
N4 (mean)	Non-linearity of the k-NN Classifier
NrAttr	Total number of attributes
NrBin	Number of binary attributes
NrCat	Number of categorical attributes
NrClass	Number of distinct classes
NrCorAttr	Number of distinct highly correlated pair of attributes
NrDisc	Number of canonical correlation between each attribute and class
NrInst	Number of instances (rows) in the dataset
NrNorm	Number of attributes normally distributed based in a given method
NrNum	Number of numeric features
NrOutliers	Number of attributes with at least one outlier value
NsRatio	Noisiness of attributes
NumToCat	Number of numerical and categorical features
Ptrace	Pillai's trace
Range (mean)	Range (max - min) of each attribute
RoyRoot	Roy's largest root
Sd (mean)	Standard deviation of each attribute
SdRatio	Statistical test for homogeneity of covariances
Skewness (mean)	Skewness for each attribute
Sparsity (mean)	(Possibly normalized) sparsity metric for each attribute
T1 (mean)	Fraction of hyperspheres covering data
T2	Average number of features per dimension
T3	Average number of PCA dimensions per points
T4	Ratio of the PCA dimension to the original dimension
TMean (mean)	Trimmed mean of each attribute
Var (mean)	Variance of each attribute
WLambda	Wilks' Lambda value

Extra results on accuracy

We report the empirical results as in the main article, this time using the accuracy of the models as our metric instead of the f1-score. All results are in line with the results for the f1-score. The hypothesis of the Friedman test is rejected with a value of $2.09 \cdot e^{-23}$. In Fig. 6, we show that the black box models are significantly better than the white box models but not significantly different from each other. The same can be said for the white box models. We see a non-linear nature of the cost of comprehensibility and explainability in Fig. 7a and b. Finally, from the boxplots in Fig. 8 we see again that for the opaque

datasets the surrogate white box models are better on average than the native ones. We also reject the hypothesis that the native and surrogate white boxes perform equally well (p-value $9.63 \cdot e^{-6}$) on average across all datasets. When we perform the same analysis for the two different types of datasets, we see again that the surrogate white box models outperform the native white box ones for the opaque datasets (Wilcoxon test p-value of $2.71 \cdot e^{-6}$), while the two are not significantly different for the comprehensible datasets (Wilcoxon test p-value of 0.53). All these results are comparable with the results obtained when using f1-score as a metric.

finally, we also compare the dataset properties that predict whether a dataset is *opaque* or *comprehensible* and see if they are the same for both metrics. We see in Table 5 that the same dataset properties are important in predicting the gap in *accuracy* as in predicting the gap in *f1-score*, but that now some more attributes are significant. *F1v*, *L1*, *EqNumAttr* and *NsRatio* were already significant in predicting the gap in *f1-score*. The linearity measures *L2* and *L3* are now also significant but they have a similar meaning as *L1*, namely they are linearity measures that quantify whether the data is linearly

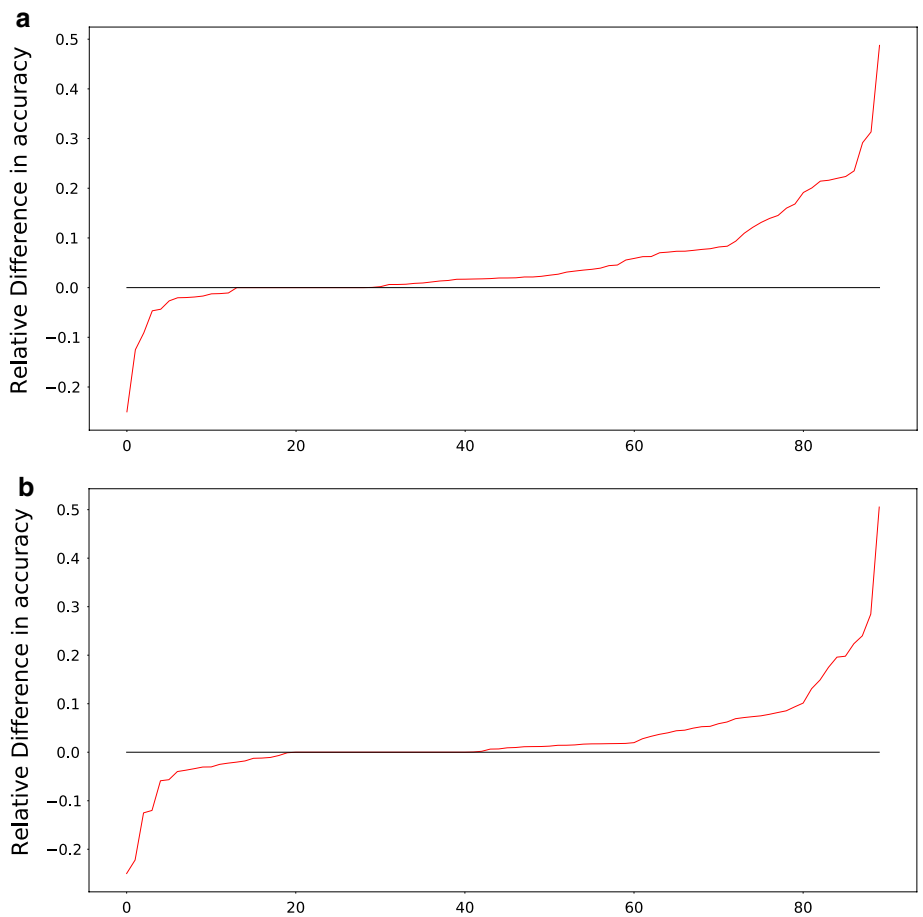

Fig. 7 Comparing black box and white box models. For both plots, the datasets are ordered according to the gap in accuracy between the best black box and the best native (left figure) or surrogate (right) white box model (right).The y-axis measures the relative difference in the accuracy, defined as the ratio of the difference between the black and white box accuracy divided by that of the best model

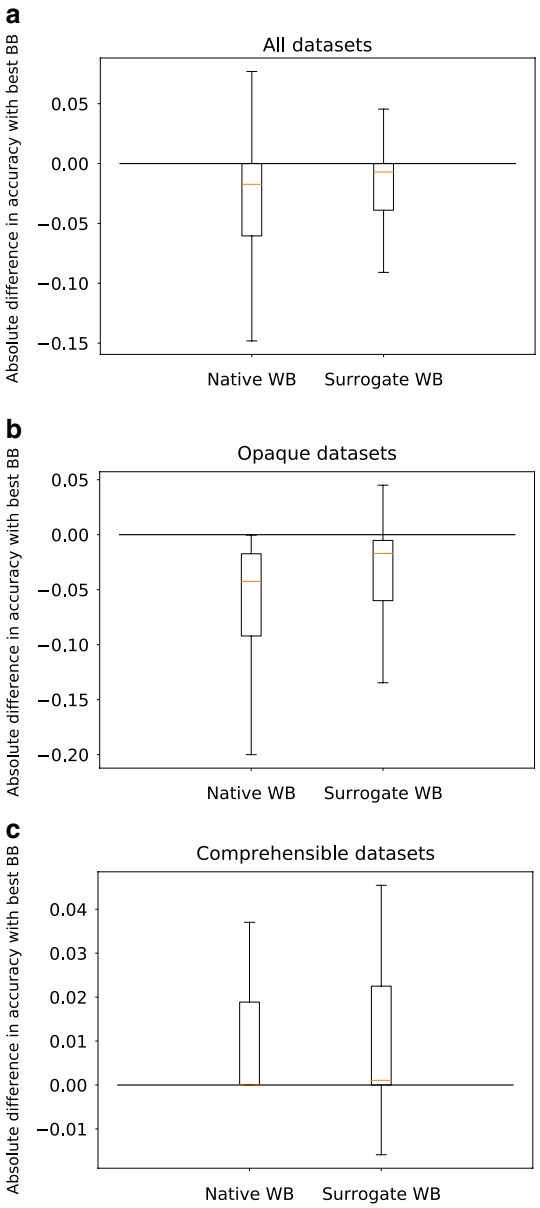

Fig. 8 Comparison across datasets of best black box model for each dataset, surrogate white box model mimicking this best black box, and best native white box model. BB stands for black box and WB for white box. The line at 0 indicates the performance of the best black box model. The y-axis indicates the absolute difference in accuracy from the best black box model

separable, which means higher values of these attributes point to more complex problems [59]. *N4* signifies the non-linearity of the nearest neighbor classifier and higher values are also indicative of problems of greater complexity [59]. *F3* signifies the Maximum Individual Feature Efficiency where lower values indicate simpler problems [59]. *JointEnt* computes the relationship of each attribute with the target variable, capturing the relative importance of the predictive attributes [58]. *CanCor* measures the canonical correlation between the predictive attribute and the target [58].

Table 5 The dataset properties that are significant when explaining the cost of comprehensibility using a number of standard dataset properties as independent variables in a regression model where the cost is the dependent variable

Variable	MSE	Pr(>F)	Coef
F1v	0.089	0.0002	0.116
L3	0.059	0.003	0.096
T4	0.045	0.012	0.063
L2	0.044	0.012	0.089
L1	0.042	0.014	0.082
N4	0.038	0.020	0.094
CanCor	0.032	0.034	−0.075e
F3	0.031	0.036	0.087
F3	0.031	0.036	0.087
EqNumAttr	0.029	0.044	−0.171e
NsRatio	0.029	0.044	−0.171e

Abbreviations
AI: Artificial Intelligence; ML: Machine Learning; RF: Random Forest; MLP: Multi-Layer Perceptron; SVM: Support Vector Machine; DT: Decision Tree; LR: Logistic Regression; RIPPER: Repeated Incremental Pruning to produce Error Reduction; PMLB: Penn Machine Learning Benchmark; LIME: Local Interpretable Model-Agnostic Explanations; SHAP: SHapley Additive exPlanations.

Acknowledgements
Not applicable.

Authors' contributions
All authors contributed equally on the text. SG performed all analyses. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The datasets generated and/or analysed during the current study are available in the PMLB repository, https://github.com/EpistasisLab/pmlb [43].

Declarations

Ethics approval and consent to participate
Not applicable

Consent for publication
Not applicable

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Department of Engineering Management, University of Antwerp, Antwerp, Belgium. [2]Decision Sciences and Technology Management, INSEAD, Fontainebleau, France.

Received: 7 October 2021 Accepted: 15 February 2022
Published online: 07 March 2022

References

1. Agrawal A. New York regulator orders probe into Goldman Sachs' credit card practices over Apple Card and sexism; November 12, 2019. Medianama, Online, https://www.medianama.com/2019/11/223-apple-card-sexism-goldman-sachs/. Accessed 1 Feb 2022.
2. Martens D. Data Science ethics: concepts, Techniques and Cautionary Tales. Oxford: Clarendon Press; 2022.
3. Wozniak S. Tweet; November 10, 2019. Twitter, Online, accessed February 1, 2022. https://twitter.com/stevewoz/status/1193330241478901760.
4. Breiman L. Statistical modeling: the two cultures (with comments and a rejoinder by the author). Stat Sci. 2001;16(3):199–231.

5. Broad Agency Announcement, Explainable Artifcial Intelligence (XAI). https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf. Accessed 12 Nov 2020.
6. Martens D, Baesens B, Van Gestel T, Vanthienen J. Comprehensible credit scoring models using rule extraction from support vector machines. Eur J Oper Res. 2007;183(3):1466–76.
7. Wachter S, Mittelstadt B, Floridi L. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. Int Data Priv Law. 2017;7(2):76–99.
8. Linardatos P, Papastefanopoulos V, Kotsiantis S. Explainable ai: a review of machine learning interpretability methods. Entropy. 2021;23(1):18.
9. Freitas AA. Comprehensible classification models: a position paper. ACM SIGKDD Explorat Newslet. 2014;15(1):1–10.
10. Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B. Definitions, methods, and applications in interpretable machine learning. Proc Natl Acad Sci. 2019;116(44):22071–80.
11. Rudin C, Radin J. Why are we using black box models in AI when we don't need to? A lesson from an explainable AI competition. Harvard Data Sci Rev. 2019;1:2.
12. Makridakis S, Hibon M. The M3-Competition: results, conclusions and implications. Int J Forecast. 2000;16(4):451–76.
13. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44.
14. Borisov V, Leemann T, Seßler K, Haug J, Pawelczyk M, Kasneci G. Deep neural networks and tabular data: A survey. arXiv preprint arXiv:211001889. 2021.
15. Popov S, Morozov S, Babenko A. Neural oblivious decision ensembles for deep learning on tabular data. arXiv preprint arXiv:190906312. 2019.
16. Arık SO, Pfister T. Tabnet: Attentive interpretable tabular learning. arXiv. 2020.
17. Zeng J, Ustun B, Rudin C. Interpretable classification models for recidivism prediction. arXiv preprint arXiv:150307810. 2015.
18. Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inform Fusion. 2020;58:82–115.
19. Carvalho DV, Pereira EM, Cardoso JS. Machine learning interpretability: a survey on methods and metrics. Electronics. 2019;8(8):832.
20. Shorten C, Khoshgoftaar TM, Furht B. Deep Learning applications for COVID-19. J Big Data. 2021;8(1):1–54.
21. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data. 2021;8(1):1–74.
22. Molnar C. Interpretable machine learning. Lulu. com; 2020.
23. Pintelas E, Livieris IE, Pintelas P. A grey-box ensemble model exploiting black-box accuracy and white-box intrinsic interpretability. Algorithms. 2020;13(1):17.
24. Ribeiro MT, Singh S, Guestrin C. "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; 2016. p. 1135–1144.
25. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems; 2017. p. 4768–4777.
26. Martens D, Provost F. Explaining data-driven document classifications. MIS Quart. 2014;38(1):73–100.
27. Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. ACM Comput Surv. 2018;51(5):1–42.
28. Huysmans J, Dejaeger K, Mues C, Vanthienen J, Baesens B. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. Decision Support Syst. 2011;51(1):141–54.
29. Allahyari H, Lavesson N. User-oriented assessment of classification model understandability. In: 11th scandinavian conference on Artificial intelligence. IOS Press; 2011.
30. Askira-Gelman I. Knowledge discovery: comprehensibility of the results. In: Proceedings of the thirty-first Hawaii international conference on system sciences. vol. 5. IEEE; 1998. p. 247–255.
31. Bibal A, Frénay B. Interpretability of machine learning models and representations: an introduction. In: ESANN; 2016. .
32. Freitas AA. Automated machine learning for studying the trade-off between predictive accuracy and interpretability. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. Springer; 2019. p. 48–66.
33. Rüping S, et al. Learning interpretable models. Universität Dortmund. 2006.
34. Lipton ZC. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue. 2018;16(3):31–57.
35. Stiglic G, Kocbek P, Fijacko N, Zitnik M, Verbert K, Cilar L. Interpretability of machine learning-based prediction models in healthcare. Wiley Interdiscipl Rev. 2020;10(5):e1379.
36. Miller GA. The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol Rev. 1956;63(2):81.
37. Confalonieri R, Weyde T, Besold TR, Martín FMdP. Trepan Reloaded: A Knowledge-driven Approach to Explaining Artificial Neural Networks. arXiv preprint arXiv:190608362. 2019.
38. Ramon Y, Martens D, Evgeniou T, Praet S. Can metafeatures help improve explanations of prediction models when using behavioral and textual data? Machine Learning. 2021;p. 1–40.
39. Baesens B, Van Gestel T, Viaene S, Stepanova M, Suykens J, Vanthienen J. Benchmarking state-of-the-art classification algorithms for credit scoring. J Oper Res Soc. 2003;54(6):627–35.
40. Lacave C, Díez FJ. A review of explanation methods for Bayesian networks. Knowl Eng Rev. 2002;17(2):107–27.
41. Chubarian K, Turán G. Interpretability of Bayesian Network Classifiers: OBDD Approximation and Polynomial Threshold Functions. In: ISAIM; 2020.
42. García IdCG. Self-labeling Grey-box Model: An Interpretable Semi-supervised Classifier [Ph.D. thesis]. Queens University Belfast, United Kingdom; 2020.
43. Olson RS, La Cava W, Orzechowski P, Urbanowicz RJ, Moore JH. PMLB: a large benchmark suite for machine learning evaluation and comparison. BioData Mining. 2017;10(1):1–13.
44. Fernández-Delgado M, Cernadas E, Barro S, Amorim D. Do we need hundreds of classifiers to solve real world classification problems? J Mach Learn Res. 2014;15(1):3133–81.

45. Zhang C, Liu C, Zhang X, Almpanidis G. An up-to-date comparison of state-of-the-art classification algorithms. Expert Syst Appl. 2017;82:128–50.
46. Lessmann S, Baesens B, Seow HV, Thomas LC. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. Eur J Oper Res. 2015;247(1):124–36.
47. Mayr A, Klambauer G, Unterthiner T, Steijaert M, Wegner JK, Ceulemans H, et al. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. Chem Sci. 2018;9(24):5441–51.
48. Lorena AC, Jacintho LF, Siqueira MF, De Giovanni R, Lohmann LG, De Carvalho AC, et al. Comparing machine learning classifiers in potential distribution modelling. Exp Syst Appl. 2011;38(5):5268–75.
49. Macià N, Bernadó-Mansilla E. Towards UCI+: a mindful repository design. Inform Sci. 2014;261:237–62.
50. Fung G, Sandilya S, Rao RB. Rule extraction from linear support vector machines. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining; 2005. p. 32–40.
51. Martens D, Huysmans J, Setiono R, Vanthienen J, Baesens B. Rule extraction from support vector machines: an overview of issues and application in credit scoring. Rule extraction from support vector machines. 2008;p. 33–63.
52. Johansson U, Sönströd C, Löfström T, Boström H. Obtaining accurate and comprehensible classifiers using oracle coaching. Intell Data Analy. 2012;16(2):247–63.
53. Johansson U, Sönströd C, Accurate König R. Interpretable regression trees using oracle coaching. In: IEEE Symposium on Computational Intelligence and Data Mining (CIDM). IEEE. 2014;2014:194–201.
54. Craven M, Shavlik J. Extracting tree-structured representations of trained networks. Adv Neural Inform Process Syst. 1995;8:24–30.
55. Zhou ZH. Rule extraction: using neural networks or for neural networks? J Comput Sci Technol. 2004;19(2):249–53.
56. Martens D, Baesens B, Van Gestel T. Decompositional rule extraction from support vector machines by active learning. IEEE Trans Knowl Data Eng. 2008;21(2):178–91.
57. Alcobaça E, Siqueira F, Rivolli A, Garcia LPF, Oliva JT, de Carvalho AC, et al. MFE: Towards reproducible meta-feature extraction. J Mach Learn Res. 2020;21:111–1.
58. Rivolli A, Garcia LP, Soares C, Vanschoren J, de Carvalho AC. Characterizing classification datasets: a study of meta-features for meta-learning. arXiv preprint arXiv:180810406. 2018.
59. Lorena AC, Garcia LP, Lehmann J, Souto MC, Ho TK. How Complex is your classification problem? A survey on measuring classification complexity. ACM Comput Surv. 2019;52(5):1–34.
60. Singh A, Thakur N, Sharma A. A review of supervised machine learning algorithms. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE; 2016. p. 1310–1315.
61. Cohen WW. Fast effective rule induction. In: Machine learning proceedings 1995. Elsevier; 1995. p. 115–123.
62. Friedman JH, Popescu BE. Predictive learning via rule ensembles. Annal Appl Stat. 2008;2(3):916–54.
63. Demšar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res. 2006;7:1–30.
64. Nemenyi PB. Distribution-free multiple comparisons. Princeton University; 1963.
65. Trawiński B, Smętek M, Telec Z, Lasota T. Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. Int J Appl Math Comput Sci. 2012;22:867–81.
66. Demšar J, Curk T, Erjavec A, Gorup Č, Hočevar T, Milutinovič M, et al. Orange: data mining toolbox in Python. J Mach Learn Res. 2013;14(1):2349–53.
67. de Fortuny EJ, Martens D. Active learning-based pedagogical rule extraction. IEEE Trans Neural Netw Learn Syst. 2015;26(11):2664–77.
68. Michie D, Spiegelhalter DJ, Taylor CC. Machine learning, neural and statistical classification. Citeseer; 1994.
69. Luengo J, Herrera F. An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. Knowl Inform Syst. 2015;42(1):147–80.
70. Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access. 2018;6:52138–60.
71. Schwartzenberg C, van Engers T, Li Y. The fidelity of global surrogates in interpretable Machine Learning. BNAIC/BeneLearn. 2020;2020:269.

Publisher's Note