On The Stability of Interpretable Models

Riccardo Guidotti
KDDLab, ISTI-CNR, Pisa, Italy
riccardo.guidotti@isti.cnr.it

Salvatore Ruggieri
KDDLab, University of Pisa, Italy
ruggieri@di.unipi.it

***Abstract*—Interpretable classification models are built with the purpose of providing a comprehensible description of the decision logic to an external oversight agent. When considered in isolation, a decision tree, a set of classification rules, or a linear model, are widely recognized as human-interpretable. However, such models are generated as part of a larger analytical process. Bias in data collection and preparation, or in model's construction may severely affect the accountability of the design process. We conduct an experimental study of the stability of interpretable models with respect to feature selection, instance selection, and model selection. Our conclusions should raise awareness and attention of the scientific community on the need of a *stability impact assessment* of interpretable models.**

***Index Terms*—Classifiers, Interpretability, Model Stability**

I. Introduction

Interpretable machine learning models aim at trading-off predictive accuracy with human-comprehensibility and verifiability. They are also used to explain the global logic of inscrutable black-box machine learning models, such as neural networks, or the local logic of specific decisions taken by such black-boxes [1], [2]. This is achieved by a form of reverse engineering, where interpretable models are trained on a sample of the population or on a random sample in the neighborhood of the instance whose decision has to be explained. If the interpretable model can accurately reproduce the black-box decisions, then it can be used as a surrogate model of the black-box.

The data science process of learning an interpretable model, either directly for decision-making or by reverse engineering a black-box, includes a number of design choices.

- On the set of features (*feature selection*): a black-box uses a set of features which may be not completely known, hence reverse engineering it must consider which features to use for the surrogate model.
- On the subset of instances (*instance selection*): instance generation/perturbation in black-box explanation can be purely random [3], [4], or adopt refined approaches, e.g., the genetic generation process described in [5].
- On the interpretable model (*model selection*): a growing number of variants and specific learning algorithms are being proposed [1].

Such a process must be accountable [6], namely the interpretable (possibly, surrogate) model must be able to provide "a satisfactory answer [about black-box decisions] to an external oversight agent"[1]. However, since the above design choices include a number of elements subject to randomness, it may end up with unstable results, i.e., variations in training data collection and selection, or apparently neutral design choices may lead to different interpretable models and decision explanations. Stability of interpretable models is then a key property towards accountability of machine learning (black-box) decision making.

The contribution of this paper consists of an *experimental study* of the stability of interpretable classification models with respect to the three design choices above. We will consider *decision trees*, *rule-based classifiers*, and *linear models*, which are widely agreed to provide explanations of their decisions that are easily interpretable by humans [7], [8]. We conclude that, to pursue accountability, interpretable model's learning processes should comprise a *stability impact assessment* which is currently missing in guidelines and best-practices[2].

The rest of the paper is organized as follows. In Section II, we survey related work on the impact of training data variations. Section III introduces interpretable models and measures, and feature/instance selection algorithms that will be considered in experiments. The evaluation framework is motivated and formalized in Section IV. Section V reports and discusses experimental results. Finally, Section VI summarizes the contribution of the paper, its limitations, and future work.

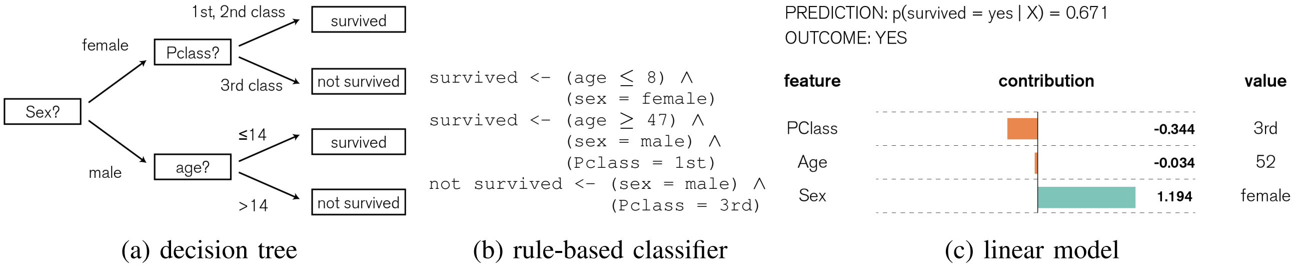
II. Related Work

Stability is a property of the output of a learning process. The representation of the output can be *intensional* (a classifier) or *extensional* (its predictions).

Extensional stability of classifier predictions was modelled by [9] through a measure of agreement among predictions. The proposed approach consists of a $m \times 2$-fold cross-validation. At each of the m steps, two classifiers are built on the two folds, and tested on artificially generated instances from a population distribution. The agreement measure is the percentage of instances whose predictions of the two classifiers coincide. The average agreement over the m runs is the final estimate of stability of the learning process. Agreement is a semantic measure, and it has the advantage of being classifier-agnostic. Related to measurement of extensional stability is the bias-variance decomposition of the error of classifiers [10]. Bias

[1] IEEE Glossary for Discussion of Ethics of Autonomous and Intelligent Systems: *https://ethicsinaction.ieee.org*https://ethicsinaction.ieee.org.

[2] See, e.g., the AINow report 2018 on *Algorithmic impact assessments*: *https://ainowinstitute.org/aiareport2018.pdf*. Stability impact assessment differs from sensitivity analysis as the the first one evaluates how much vary the model when different features or records are used to train it, while the last one estimates the impact of varying certain features on the final outcome.



```
survived <- (age ≤ 8) ∧
            (sex = female)
survived <- (age ≥ 47) ∧
            (sex = male) ∧
            (Pclass = 1st)
not survived <- (sex = male) ∧
                (Pclass = 3rd)
```

(a) decision tree (b) rule-based classifier (c) linear model

Fig. 1. Interpretable models learned on the Titanic dataset *https://www.kaggle.com/c/titanic*

is reduced and variance is increased with increasing model complexity at the risk of overfitting. This would suggest that less interpretable models are also more unstable and overfitted. On the theoretical side, [11] proved that generalization error can be bound by (expectation of) stability.

Measures of interpretability of classifiers must, however, be necessarily syntactic, since this is the level at which humans interface with models. This paper concentrates then on *intensional stability* of a learning process. One of the early studies regards the impact of training set size on the accuracy of decision trees [12], showing that the best performance can be achieved with sufficiently many instances, after which there is no convenience to add more data. Coping with variability of classifiers due to random noise has been tackled by adopting statistical tests for validating split tests at decision nodes [13], or by adopting split methods that account for almost equal split attributes (sources of instability) [14]. Finally, decision tree simplification is another class of approaches that trade-off accuracy with simplicity [15]. Intensional stability of feature selection methods considered variability in the set of features selected [16], [17]. Measures of stability include average Jaccard similarity and Pearson's correlation among all pairs of feature subsets selected from different training sets generated using cross-validation, jacknife or bootstrap. As pointed out by [16], intensional instability of feature selection does not necessarily implies extensional instability of the final classifier, due to redundant features. In summary, an experimental study of the intensional instability of interpretable models at the variation of the learning process design choices is missing in the literature. This is becoming relevant in the context of black-box explanation, where an early attempt at studying robustness of single explanations is [18], [19].

III. Setting the Stage

A. Interpretable Models

Interpretability is the ability to explain or to provide meaning in terms understandable to a human [1]. Decision trees, rule-based classifiers, and linear models are acknowledged as being interpretable classification models.

Decision trees (DT) consist of a tree graph with internal nodes representing tests on predictive features, and leaf nodes assigning a class label to instances reaching the leaf (see e.g., Fig. 1(a)). A path from the root to a leaf represents an explanation of the decision at the leaf in terms of a conjunction of test conditions. We consider the two mostly adopted greedy learning algorithms: **CART** (Classification and Regression Trees) [20] as implemented by the *scikit-learn* Python library[3], and **C4.5** [21] as implemented by the computationally efficient *YaDT* system[4] [22]. **C4.5** performs multi-way univariate splits and it includes tree simplification (error-based pruning). We do not consider instead the split condition of [14], designed for stability, since it produces disjunctive test conditions, thus leading to a higher expressivity language.

Rule-Based (RB) classifiers consist of a set of classification rules, typically in the form of *if-then* rules stating the class label for a given conjunctive condition on the predictive feature values (see Fig. 1(b)). In this work, we consider the **FOIL** (First Order Inductive Learner) [23] and **CPAR** (Classification based on Predictive Association Rules) [24] algorithms, as implemented by the *LUCS-KDD* library[5]. The former generates a very small number of rules, but has lower accuracy than the latter. Similarly to DTs, we restrict to sets of conjunctive classification rules. Another natural choice would have been **RIPPER** [25], which however produces *ordered* sequences of conjunctive rules. Instead, we compare DT and RB classifiers with the same expressivity.

Linear Models (LM) classifiers consist of the sign and the magnitude of the contribution of feature values (or ranges) to a class label (encoded as an integer) as stated by coefficients in a linear formula (see Fig. 1(c)). If the contribution is positive (resp., negative), the value of the feature increases (resp., decreases) the probability of the model's decision. We focus on three algorithms for linear models: Linear Regression (**LINREG**) [26], and its regularized forms **RIDGE** [27] and **LASSO** [28], as implemented by the *scikit-learn* library. They are commonly used in black-box explanation approaches [3], [29].

B. Measuring Interpretability and Stability

Several syntactic measures of interpretability are considered in the literature. Structural measures (SM) look at models in isolation, and quantify the degree of syntactic (intensional) interpretability of a model by resorting to model complexity. Stability is quantified through the deviation of the measure distribution over models learned from different samples of the population. Comparative measures (CM) look at pairs of models, and quantify the syntactic similarity between the two

[3] *http://scikit-learn.org*.

[4] *http://pages.di.unipi.it/ruggieri/software*.

[5] *https://cgi.csc.liv.ac.uk/∼frans/KDD/Software*.

models. Stability is quantified by the mean value over all pairs of models learned from different samples of the population.

Measures common to decision trees, rule-based classifiers and linear models include:

- *number of features* (SM) used[6]: for DT the features used in at least one split node, for RB those used in at least one rule, for LM the features with non-zero coefficient;
- *Jaccard coefficient* (CM): the ratio of the number of shared features of two models over the total number of features used by at least one such models.
- sample *Pearson's* (CM) correlation coefficient [17]: the Pearson's coefficient over the 0/1 vector of features used by two models.

Measures specific of a model type include:

- for decision trees: *number of nodes* (SM).
- for rule-based classifiers: *number of rules* (SM) and *size of rules* (SM), namely the total number of conjuncts in the *if*-part of rules.
- for linear models: *Kendall's* τ (CM) rank correlation of coefficients.

In summary, for structural measures, one aims at low mean values (interpretability) and low deviation (stability). For comparative measures, one aims at high mean values (stability) and low deviation (extreme outlier models).

Finally, in order to investigate the relationship between model stability, prediction accuracy, and overfitting, we will also compute the F1-scores of models on the training set ($F1_{train}$) and on the test set ($F1_{test}$), and their relative difference ($(F1_{train} - F1_{test})/F1_{train}$), which represents a measure of overfitting.

C. Feature and Instance Selection

Feature selection (FS) [30] and instance selection (IS) [31] are beneficial in removing noise and redundancies, in reducing the data collection effort, in balancing the data distribution, in speeding up model learning. They are supposed to enhance model interpretability by reducing the number of features and by preventing overfitting. Both techniques are widely used in reverse engineering of black-box models. In this paper, we consider the following standard methods, as provided by the *scikit-learn*[7] library for feature selection:

- **RFE** (Recursive Feature Elimination): given an external estimator that assigns weights to features (a decision tree by default), it greedily removes the least important feature until a given number of features is left (default: half of the total number of features);
- **SKB** (Select K Best) removes all but the k top scoring features according to the ANOVA F-value function of the features (default: K=10);
- **SP** (Select Percentile) removes all but a user-specified top scoring percentage of features with respect to the ANOVA F-value (default: pct=10).

Algorithm 1: $EvaluateStability(M, X, y)$

```
Input     : $M$ - classification model, $X$ - dataset, $y$ - outcome
Output    : $\mathcal{E}$ - evaluations
Variables: SM - structural measures, CM - comparative measures,
           $P$ - pre-processing methods

$\mathcal{M} \leftarrow \emptyset$; $\mathcal{E} \leftarrow \emptyset$                     // trained models and evaluations
for $i \in \{1, \dots, 5\}$ do
    $F \leftarrow stratified10Fold(X, y)$                 // 10 fold partitioning
    for $k \in \{1, \dots, 10\}$ do
        $X', y' \leftarrow F_{-k}(X, y)$                    // remove k-th fold
        $\widehat{X}', \widehat{y}' \leftarrow F_k(X, y)$                    // select k-th fold
        for $p \in P$ do
            $X'', y'' \leftarrow p(X', y')$                 // pre-processing
            $m^p_{i,k} \leftarrow fit(M, X'', y'')$               // learn model
            $\mathcal{M} \leftarrow \mathcal{M} \cup \{m^p_{i,k}\}$              // store the model
            $y^* \leftarrow predict(m^p_{i,k}, X'')$         // predict training
            $\widehat{y}^* \leftarrow predict(m^p_{i,k}, \widehat{X}')$             // predict test
            $f^p_{i,k} \leftarrow f1(\widehat{y}', \widehat{y}^*)$                // performance
            $\mathcal{P} \leftarrow \mathcal{P} \cup \{f^p_{i,k}\}$
            $o^p_{i,k} \leftarrow \frac{f1(y'', y^*) - f1(\widehat{y}', \widehat{y}^*)}{f1(y'', y^*)}$          // overfitting
            $\mathcal{O} \leftarrow \mathcal{O} \cup \{o^p_{i,k}\}$
for $p \in P$ do
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{\underset{m^p_{i,k} \in \mathcal{M}}{avg} f^p_{i,k}\}$             // aggr. performance
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{\underset{f^p_{i,k} \in \mathcal{M}}{avg} o^p_{i,k}\}$             // aggr. overfitting
    for $s \in SM$ do
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{\underset{m^p_{i,k} \in \mathcal{M}}{avg} s(m^p_{i,k})\}$              // aggr. s
    for $c \in CM$ do
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{\underset{m^p_{i,k} \neq \hat{m}^p_{i,k} \in \mathcal{M}}{avg} c(m^p_{i,k}, \hat{m}^p_{i,k})\}$   // aggr. c
return $\mathcal{E}$
```

Finally, we consider the following instance selection methods, as provided by the *imbalanced-learn*[8] library:

- **RUS** (Random Under Sampling) under-samples the majority class by randomly picking instances of the other classes;
- **ROS** (Random Over Sampling) over-samples the minority class by replicating instances of that class at random with replacement;
- **SMOTE** (Synthetic Minority Over-sampling Technique) over-samples minority class by generating instances along the linear segment between an instance of the minority class and one of its k nearest neighbors (default: k=5).

We restrict here to class balancing and random sampling methods, because they are widely adopted in black-box explanation approaches [5], [32].

IV. Experimental Framework

Interpretable classification models are the end products of an articulated analytic process. We will evaluate the impact of process design on their intensional stability. To this end, we consider the following steps, which motivate the experimental procedure of Algorithm 1.

First, any observational research project must account for variability/bias in data collection [33]. Following a standard methodology for estimating accuracy of classifiers[9], we adopt

[6] While *YaDT* and *LUCS-KDD* work directly on discrete features, algorithms of the *scikit-learn* require binarization of such features. Nevertheless, we count the number of original features, not of the binarized ones.

[7] *http://scikit-learn.org/stable/modules/feature_selection.*

[8] *http://contrib.scikit-learn.org/imbalanced-learn.*

[9] Cross-validation is an nearly unbiased estimator of accuracy [34]. Variability of the estimator is accounted for by adopting repetitions [35].

TABLE I
EXPERIMENTAL DATASETS.

dataset	adult	anneal	census	clean1	clean2	coil	cover	credit	sonar	soybean
instances	48,842	898	299,285	476	6,598	9,822	581,012	1,000	208	683
features	14	38	40	166	166	85	54	20	60	35
class values	2	6	2	2	2	2	7	2	2	19

a 5-repetition of 10-fold stratified cross-validation to account for variations in the data. At each iteration, all the available data is split in 10 folds. For each fold, the process described next is applied on 9 folds used as training data, and one fold as test (denoted by the hat $\hat{\cdot}$). This is formalized in the two outer loops at lines 2–16 of Algorithm 1. Second, the impact of pre-processing steps is evaluated by considering no pre-processing, feature selection, instance selection, and possibly combinations of them. Let P be the a set of pre-processing methods, including no modification at all. The inner loop at lines 7–16 of Algorithm 1 iterates over P for the current fold k at iteration i. A pre-processing $p \in P$ is applied to the training data, and then the model is learned from the processed data. In Algorithm 1, models are stored in the set $\mathcal{M}$. Since pre-processing is the inner loop, paired comparison of models built from different pre-processing methods can be conducted when analyzing the results of the evaluation. Moreover, lines 13–16 keep track of the predictive performance and of the degree of overfitting on the test data (the k^{th} fold).

Third, measures of interpretability, performance, and overfitting of the learned interpretable classification models must be aggregated over the 50 models (5 repetitions, 10 models each) of each pre-processing method. Performance, overfitting, and structural measures (SM) are aggregated using the mean value (lines 18–21). Comparative measures (CM) are aggregated by taking the all-pairs average (lines 22–23). Both loops are inside the loop at lines 17–23 that iterates over the set P of pre-processing algorithms.

The results of the above experimental framework are intended to support accountability questions that a data analyst should answer before deploying a classification model, namely, how sensitive is the interpretability of a classification model to changes: *in feature selection? in instance selection? in model selection? And, questions such as: which pre-processing provide the lowest variability for a given model? at which price (e.g., accuracy loss)?*

Finally, it is worth noting that the experimental framework does not contain original parts *per se*. All in all, it consists of a repeated stratified cross-validation process plus statistical tests of collected measures of interpretability, stability, accuracy, and overfitting. However, it allows for assessing in a principled way the stability of combinations of old and new feature, instance, and model selection methods.

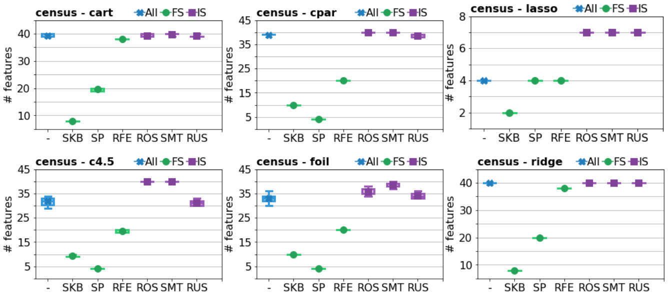


Fig. 2. Dataset: *census*. Measure: number of features.

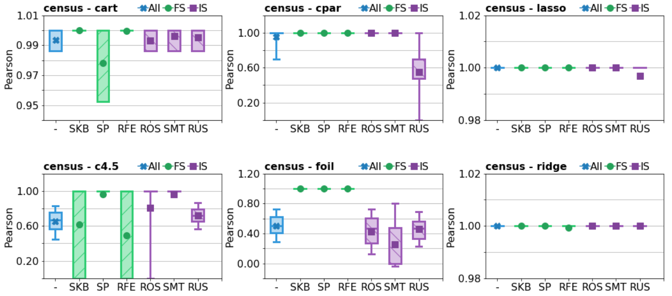


Fig. 3. Dataset: *census*. Measure: Pearson's correlation.

V. EXPERIMENTS

We run experiments on a selection of ten small and medium sized datasets widely referenced for classification tasks and publicly available from the UCI ML repository[10]. Table I shows summary statistics on the datasets: instances are in the range 208–581K, features in 14–166, and number of classes in 2–19. The framework of Algorithm 1 has been implemented in Python[11] by integrating external libraries (**YADT**, **FOIL**, and **CPAR**) through wrappers of inputs/outputs. The software has been designed to be extensible to additional models, pre-processing methods, and intepretability measures. Unless specified otherwise, parameters of algorithms are the defaults in their original systems[12].

A. Preliminary Analysis on the Census Dataset

Let us start focusing on the number of features used by a classification model. Fig. 2 considers the *census* dataset. Left plots report on DT models (**CART** and **C4.5**), middle plots on RB models (**CPAR** and **FOIL**), and right plots on LM models (**LASSO** and **RIDGE**[13]). Each plot shows the boxplots for

[10] *https://archive.ics.uci.edu/ml/datasets.html*

[11] Source code and datasets: *https://github.com/riccotti/InterpretableModels*.

[12] **C4.5**: split = Gain Ratio, stop criterion = -m 2, pruning = -ebp (error-based); **CART**: split = Gini, min_samples_split = 2, min_samples_leaf = 1, max_depth = None; **CPAR**: delta = 0.05, alpha = 0.3, gain_similarity_ratio = 0.99, min_gain_thr = 0.7; **FOIL**: min_gain_thr = 0.7; **LASSO**: alpha = 1.0; **RIDGE**: alpha = 1.0. FS: **SBK**: k=10, fun=ANOVA F-value, **SP**: percentile=10, fun=ANOVA F-value, **RFE**: n_features_to_select = half of the features, estimator=sklearn DecisionTree default parameters. IS: **SMT**: k=5.

[13] We omit **LINREG** for space reasons as it behaves as **RIDGE**.

TABLE II
NUMBER OF FEATURES, PEARSON'S CORRELATION, AND F1-SCORE (MEAN ± STDEV).

	number of features			**Pearson's correlation**			**F1-score**		
dataset	**C4.5**	**RFE+C4.5**	**C4.5 d $\leq$ 5**	**C4.5**	**RFE+C4.5**	**C4.5 d $\leq$ 5**	**C4.5**	**RFE+C4.5**	**C4.5 d $\leq$ 5**
adult	14.00 ± 0.00	7.00 ± 0.00	**6.24 ± 1.09**	**1.00 ± 0.00**	**1.00 ± 0.00**	0.76 ± 0.19	**0.85 ± 0.00**	**0.85 ± 0.00**	0.84 ± 0.00
anneal	11.20 ± 0.69	9.04 ± 1.06	**5.76 ± 0.43**	0.85 ± 0.09	0.66 ± 0.17	**0.88 ± 0.11**	**0.98 ± 0.01**	**0.98 ± 0.01**	0.85 ± 0.03
census	31.92 ± 1.92	19.66 ± 0.47	**7.78 ± 0.70**	0.66 ± 0.15	0.49 ± 0.50	**0.89 ± 0.07**	**0.95 ± 0.00**	**0.95 ± 0.00**	0.93 ± 0.00
clean1	23.70 ± 2.82	24.68 ± 2.73	**6.30 ± 2.16**	0.25 ± 0.15	0.03 ± 0.13	**0.48 ± 0.25**	**0.83 ± 0.05**	0.81 ± 0.05	0.68 ± 0.06
clean2	74.80 ± 4.78	60.06 ± 3.64	**13.26 ± 0.77**	0.37 ± 0.08	-0.45 ± 0.22	**0.59 ± 0.12**	**0.97 ± 0.01**	**0.97 ± 0.01**	0.89 ± 0.01
coil	8.12 ± 3.68	5.34 ± 3.85	**3.70 ± 3.53**	0.20 ± 0.18	0.18 ± 0.31	**0.27 ± 0.37**	**0.91 ± 0.00**	**0.91 ± 0.00**	**0.91 ± 0.00**
cover	52.78 ± 0.50	27.00 ± 0.00	**16.42 ± 0.70**	0.71 ± 0.36	1.00 ± 0.00	**0.92 ± 0.05**	**0.95 ± 0.00**	0.94 ± 0.00	0.45 ± 0.00
credit	17.56 ± 1.44	9.66 ± 0.51	**10.70 ± 1.76**	0.12 ± 0.29	0.32 ± 0.79	**0.43 ± 0.22**	0.70 ± 0.04	**0.71 ± 0.04**	**0.71 ± 0.04**
sonar	12.04 ± 1.44	11.50 ± 1.46	**9.32 ± 1.83**	0.23 ± 0.18	-0.06 ± 0.22	**0.28 ± 0.21**	**0.74 ± 0.08**	0.73 ± 0.08	0.73 ± 0.09
soybean	22.52 ± 1.70	15.42 ± 1.17	**14.82 ± 1.71**	0.70 ± 0.13	-0.80 ± 1.06	**0.79 ± 0.12**	**0.91 ± 0.03**	0.89 ± 0.03	0.68 ± 0.03

no pre-processing ("-"), for 3 Feature Selection (FS) methods (**SKB**, **SP**, and **RFE**), and for 3 Instance Selection (IS) methods (**ROS**, **SMT**, and **RUS**). Feature selection methods reduce the total number of features used by the classification model, as one would expect, thus improving the interpretability measure. Moreover, since redundant/noisy features are removed as well, this also reduces deviation over the 50 folds, thus improving stability. Instance selection has a similar beneficial effect on deviation, but in some cases (**LASSO** and **C4.5**) it increases the number of features. However, for a gross-grained measure such as the number of features, the low variability provides a distorted indication of stability. In fact, two models may still largely differ in the set of features used while the number of such features is the same for both models. Jaccard similarity or Pearson's correlation among all pairs of feature sets across the 50 folds can better measure variability of the set of features (rather than the number of them) used by a classifier. Fig. 3 reports Pearson's correlation for the *census* dataset. We omit the Jaccard measure for lack of space and because it yields similar patterns. Linear models are stable, independently from the pre-processing method. In fact, Pearson's correlation is always very close to 1. For rule-based models, FS also leads to stable models. IS increases deviation of Pearson's correlation for rule-based and decision trees classifiers. This means that extreme outlier models (in terms of feature's vector) become more frequent.

B. Preliminary Analysis on Decision Trees

It is worth looking into details to the case of decision trees. Table II shows the number of features and Pearson's correlation for three **C4.5** classifiers: the one built on all available features, the one built on features selected by **RFE**, and the one built on all features but restricted to a maximum depth of 5 (**d** $\leq$ **5**). Setting a maximum depth is a valid approach to enhance the interpretability of decision trees. Such a method performs the best both w.r.t. the number of features and the Pearson's correlation. This is achieved at the expenses of loss in accuracy, as shown by the F1 score columns in Table II. Contrasting **C4.5** alone with **RFE+C4.5**, we observe that addition of the **RFE** method reduces the number of used features (interpretability), at the expenses of Pearson's correlation (stability), while the F1-score (accuracy) remains almost the same.

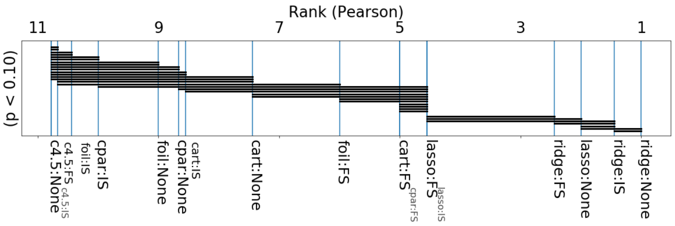


Fig. 4. Comparison of model's rank w.r.t. Pearson's correlation against each other with the Nemenyi test. Groups of classifiers that are not significantly different at 90% significance level are connected. Best ranks on the right.

C. Statistical Comparison of Models' Stability

The previous two subsections demonstrate how the exploration of the experimental results can provide insights when considering a specific dataset (Section V-A) or a specific model (Section V-B). The rest of the experiments will be devoted, instead, to understand whether there are general patterns among all possible usages of IS, FS, and models.

First, we compare the various methods among them with reference to stability. The non-parametric Friedman test compares the average ranks of learning methods over multiple datasets w.r.t. an evaluation measure, in our case Pearson's correlation. The null hypothesis that all methods are equivalent is rejected ($p < .001$). The comparison of the ranks of all methods against each other can be visually represented as shown in Fig. 4 (see [36] for details). The post-hoc Nemenyi test is used to connect methods that are not significantly different among each other. Linear models have the best ranks. For a fixed classifier, models obtained using feature selection pre-processing rank better than methods without. *Instance selection methods and decision trees* have the lowest ranks, i.e., they *are the most unstable with respect to the set of features* used by the learned model.

D. Stability-Interpretability

Let us now investigate patterns of correlation between interpretability and stability through the scatter density plots in Fig. 5, where Pearson's index (stability) is plotted against

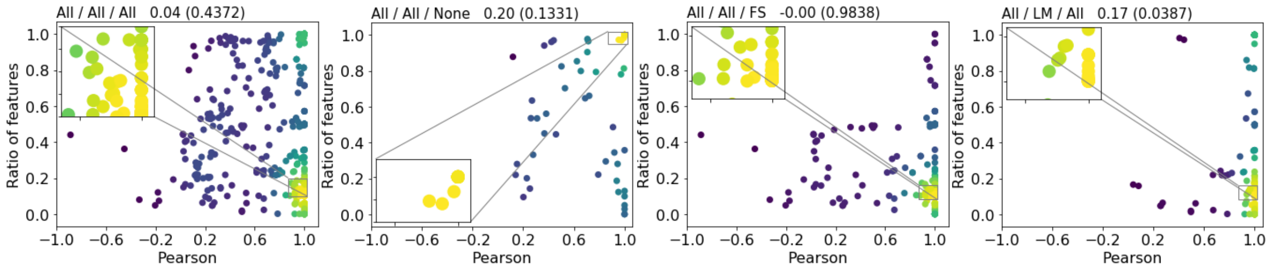


Fig. 5. Scatter density plots of Pearson's correlation vs Ratio of features used. Each point's coordinates are the mean values over the 50 experimental folds of some dataset for the following conditions (left to right): experiments for all datasets/classifiers/pre-processing; experiments with no pre-processing; experiments with FS; experiments with LM. On top: correlation and p-value. Colors: yellow = high density, green = medium density, purple = low density.

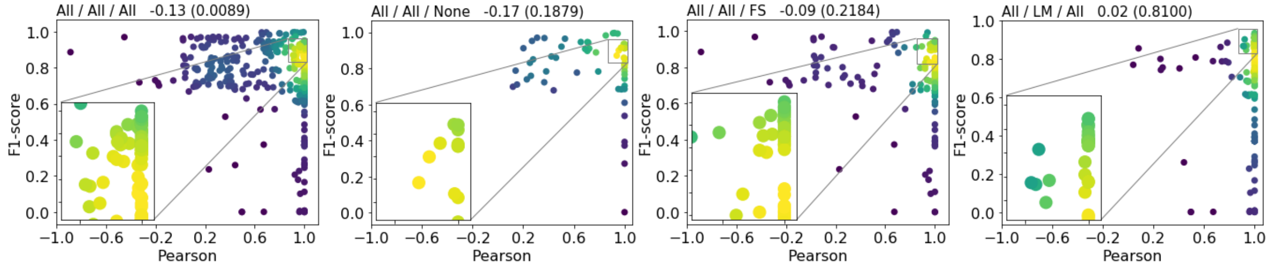


Fig. 6. Scatter plots of Pearson vs F1-score. See caption of Fig. 5.

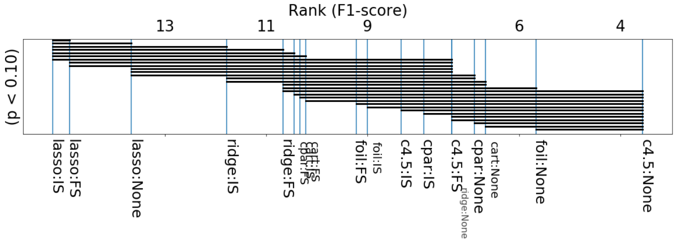


Fig. 7. Same as Fig. 4 but w.r.t. the F1-score.

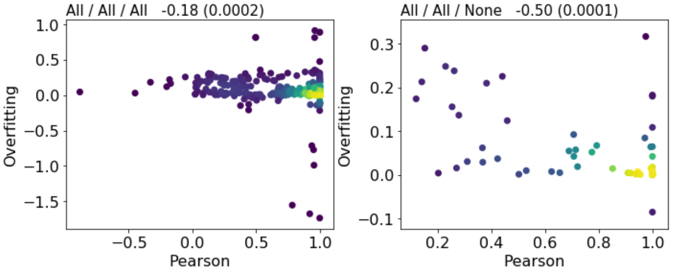


Fig. 8. Pearson vs Overfitting. Left: all experiments; right: no pre-processing. On top: correlation and p-value.

the ratio of the number of used features over the total number of features (interpretability). There are 4 scatter plots. Each point represents an experiment (50 folds). From left to right and top to bottom: experiments for all datasets/classifiers/pre-processing, experiments for all datasets and classifiers but only those with no pre-processing, experiments for all datasets and classifiers but only those with feature selection pre-processing, and experiments for only linear model classifiers. Numbers on top of scatter plots are linear correlation and, in parenthesis, p-values of such correlation. The top left plot does not highlight correlation between the measures of stability and interpretability, in general. Using no pre-processing methods increase the correlation (higher stability means lower interpretability). Feature selection does not impact on the correlation. Finally, the rightmost plot shows some positive correlation for linear models at 95% significance level.

E. Stability-Accuracy

We now investigate the relation between stability and predictive accuracy. Fig. 6 reports scatter density plots of Pearson's correlation against F1 score (averaged over the 50 test folds) for the same conditions as in Fig. 5. There is a small negative correlation for the three leftmost plots: the higher the stability (high values of Pearson's correlation) the lower the accuracy (low values of F1-score). Such a correlation is statistically significant at 99% confidence level only for the leftmost plot, which includes all experiments. For linear models, the two measures are uncorrelated. Let us look more in detail to the case of all experiments. Fig. 7 compares the ranks of the various models w.r.t. the F1 measure averaged over the 50 experimental folds. Ranks are approximately symmetric to the ones of Pearson's correlation shown in Fig. 4. Decision trees and rule-based classifiers are the best performing. Linear models are at the bottom of the ranking. Instance selection does not improve ranks of classifiers. In summary, for the interpretable models considered here, *stability and accuracy are contrasting objectives*. This demands for a trade-off analysis.

F. Stability-Overfitting

Fig. 8 reports scatter plots of stability vs overfitting, defined as the relative difference of F1 accuracy between training and

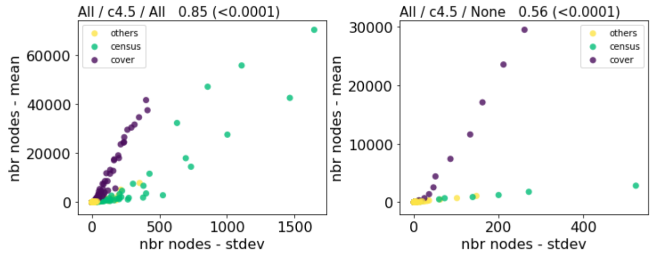


Fig. 9. Scatter plot of stability (deviation) vs interpretability (mean) w.r.t. number of nodes, left: all **C4.5** exp.'s; right: **C4.5** with no pre-processing.

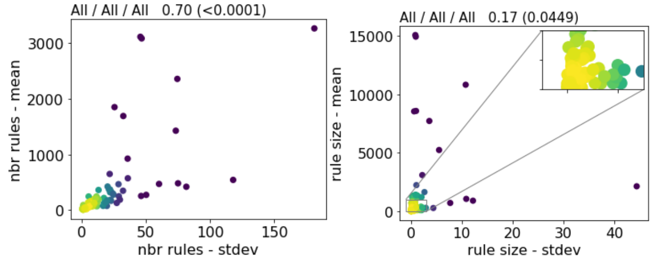


Fig. 10. Scatter density plot of stability (stddev) vs interpretability (mean) w.r.t. nbr of rules (left) and size of rules (right) in RB models. All experiments.

test set averaged over 50 folds. A negative correlation is clearly observed and statistically significant: higher Pearson's correlation (stability) leads to smaller overfitting (generalizability). This is more apparent in experiments with no pre-processing (right in Fig. 8). Such a result is somehow expected, due to the bias-variance decomposition mentioned earlier [10]. In summary, *stability and overfitting are contrasting objectives.*

G. Model-Specific Measures

When restricting to specific classifiers, finer-grained measures of interpretability can be adopted. Let us start considering the number of nodes in decision trees (for the tree depth measure, we obtain similar findings). We study the relation between interpretability and stability by varying the stopping parameter in tree construction from m=2 (default value) to m=half of the size of the dataset using a geometric progression. Such parameter stops node splitting during tree construction if the number of cases at the node is below the threshold m. Thus, we can control the maximum size of a decision tree. Fig. 9 shows the scatter plot of mean number of nodes vs standard deviation of the number of nodes over the 50 experimental folds. A statistically significant positive correlation is clearly visible, especially when restricting to a dataset in isolation (experiments with the two largest datasets are shown in different colors).

For rule-based classifiers, Fig. 10 shows the stability-interpretability relation in terms of number of rules (left) and size of rules (right). Each point has coordinates the standard deviation (x-axis) and the mean (y-axis) number/size of rules over the 50 experimental folds. Basically, the two plots are RB-specific versions of the density scatter plots in Fig. 5. Contrasting the two figures, there is now a larger statistically significant positive correlation between stability and interpretability. The correlation for the finer grained measure of sizes of rules is smaller than for the gross grained measure of number of rules, which is somehow expected.

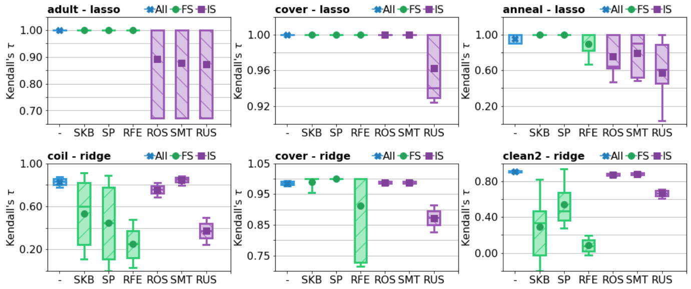


Fig. 11. Measure: Kendall's τ. Models: LM.

Finally, let us consider linear models. Kendall's τ measures the rank correlation of two sets of features, where the rank of a feature is calculated w.r.t. the descending absolute value of its coefficient. Fig. 11 reports the boxplots of τ's values over the 50 experimental folds. **LASSO** is generally more stable than **RIDGE** (high values of τ). Feature selection increases variability of the measure (extreme outlier models) for **RIDGE**, but not for **LASSO**. Vice-versa, IS increases variability for **LASSO**, but not for **RIDGE**.

H. Discussion

Experimental results highlight how the data science process of learning classifiers suffers from some variability in the measures of interpretability of produced models. There is a tension between optimizing predictive accuracy from one side, and intensional stability of interpretable classifiers on the other side. Stability and generalizability appear to be common goals, or, stated otherwise, stability and overfitting appear contrasting objectives. Also, stability and interpretability appear to be slightly positively correlated. Existing approaches for improving generalizability of classifiers, however, cannot be always applied to interpretable models. Aggregation methods (e.g., bagging, boosting, random forests) produce models that are widely agreed to be difficult to interpret. We claim that the data analyst should conduct a *stability impact assessment* together with predictive performance analysis in order to alleviate the tension between the two objectives. Such a stability impact assessment amounts at analysing the empirical distribution of the relevant interpretability measures at the variation of the design choices. Which measure is the most relevant is another design aspect that must be accounted for and that may affect the stability impact assessment. E.g., the two scatter density plots in Fig. 10 show different correlation between interpretability and stability for different measures. In summary, by exploiting the experimental framework proposed in this paper, a trade-off assessment allows for evaluating the impact of design choices made over a collection of candidate models and pre-processing methods. Overall, it provides evidence that the data analyst has been accountable for one's conduct.

VI. Conclusion

The intended objective of this paper is to raise awareness by the machine learning community on the issue of being accountable in the design of classification models, particularly those used for socially sensitive decision making. Accountability is more than interpretability, and it requires, in our opinion, an impact assessment of the whole learning process. The case of interpretable models is challenging; being them comprehensible "by definition", there is the risk that data analysts overlook the issue of accountability of the extraction process as a whole. In concrete, explanations of automated decisions made by black box models, such as neural networks, may suffer from data selection/processing bias, which makes it hard argumenting on the decision, e.g., in a trial before a court.

Our main contributions consist of a framework for intensional stability impact assessment, and an experimental analysis on several pre-processing methods and classification algorithms. The approach is implemented, released as open source, and extensible to new classifiers, methods, and measures. Experimental results show that the studied interpretable models exhibit considerable variability in terms of structural and comparative measures. Interpretability of linear models appears to be more stable than for other models, but at the expenses of lower accuracy. Decision trees, on the other hand, exhibit more variability, but they are more accurate. Stability is clearly negatively correlated to accuracy and to overfitting. However, no other generally valid pattern can be drawn. Thus, in parallel to optimizing predictive accuracy, practitioners have to look at the impact on intensional stability as well.

Several extensions of our framework are possible. First, for sake of space, we considered only a limited number of interpretable models, pre-processing methods, datasets, and measures. E.g., the comparative measure of tree edit distance [37] is even more fine-grained than decision tree size. Second, with the exception of Table II and Fig. 9, we did not consider parameters of the learning algorithms and pre-processing methods. This would add a further loop to Algorithm 1, where parameters are optimized from a parameter space (uniformly, greedly, etc.). Third, we considered only objective measures of interpretability and stability. A lab experiment can test subjective measures (legibility, understantability) on a pool of actual users. Fourth, we concentrated on interpretability of models as a whole, possibly surrogate models of black-boxes. Interpretability of explanations of individual black-box decisions is also worth considering. We could extend Algorithm 1 to collect explanations for the instances in the test fold data, and then to compute measures of interpretability of such explanations.

Acknowledgment

Work partially supported by the European Community H2020 programme under the funding schemes: "INFRAIA-1-2014-2015: Research Infrastructures" G.A. 654024 *"SoBigData"* (*www.sobigdata.eu*), G.A. 78835 *"Pro-Res"* (*prores-project.eu*), and G.A. 825619 *"AI4EU"* (*www.ai4eu.eu*).

References

[1] R. Guidotti *et al.*, "A survey of methods for explaining black box models," *ACM CSUR*, vol. 51, no. 5, pp. 93:1–93:42, Jan. 2019.

[2] D. Pedreschi, F. Giannotti, R. Guidotti *et al.*, "Meaningful explanations of black box AI decision systems," in *AAAI*, 2019.

[3] M. T. Ribeiro *et al.*, ""Why should I trust you?": Explaining the predictions of any classifier," in *KDD*. ACM, 2016, pp. 1135–1144.

[4] S. Haufe *et al.*, "On the interpretation of weight vectors of linear models in multivariate neuroimaging," *Neuroimage*, vol. 87, pp. 96–110, 2014.

[5] R. Guidotti *et al.*, "Local rule-based explanations of black box decision systems," *CoRR*, vol. abs/1805.10820, 2018.

[6] J. A. Kroll *et al.*, "Accountable algorithms," *University of Pennsylvania Law Review*, vol. 165, pp. 633–633, 2017.

[7] A. A. Freitas, "Comprehensible classification models: A position paper," *ACM SIGKDD explorations newsletter*, vol. 15, no. 1, pp. 1–10, 2014.

[8] J. Huysmans *et al.*, "An empirical evaluation of the comprehensibility of decision tree," *DSS*, vol. 51, no. 1, pp. 141–154, 2011.

[9] P. Turney, "Bias and the quantification of stability," *Machine Learning*, vol. 20, no. 1-2, pp. 23–33, 1995.

[10] H. Trevor *et al.*, *The elements of statistical learning: data mining, inference, and prediction*, ser. Springer Statistics. Springer, 2009.

[11] O. Bousquet and A. Elisseeff, "Stability and generalization," *Journal of machine learning research*, vol. 2, no. Mar, pp. 499–526, 2002.

[12] D. Jensen and T. Oates, "The effects of training set size on decision tree complexity," in *ICML*, 1999, pp. 254–262.

[13] G. Katz *et al.*, "Confdtree: A statistical method for improving decision trees," *JCST*, vol. 29, no. 3, pp. 392–407, 2014.

[14] R.-H. Li and G. G. Belford, "Instability of decision tree classification algorithms," in *KDD*. ACM, 2002, pp. 570–575.

[15] L. A. Breslow and D. W. Aha, "Simplifying decision trees: A survey," *The Knowledge Engineering Review*, vol. 12, no. 1, pp. 1–40, 1997.

[16] A. Kalousis *et al.*, "Stability of feature selection algorithms: a study on high-dimensional spaces," *KAIS*, vol. 12, no. 1, pp. 95–116, 2007.

[17] S. Nogueira and G. Brown, "Measuring the stability of feature selection," in *ECML-PKDD*. Springer, 2016, pp. 442–457.

[18] D. Alvarez-Melis and T. S. Jaakkola, "On the robustness of interpretability methods," *arXiv preprint arXiv:1806.08049*, 2018.

[19] D. A. Melis and T. Jaakkola, "Towards robust interpretability with self-explaining neural networks," in *NIPS*, 2018, pp. 7786–7795.

[20] L. Breiman *et al.*, *Classification and regression trees*. CRC press, 1984.

[21] J. R. Quinlan, *C4. 5: Programs for Machine Learning*. Elsevier, 1993.

[22] S. Ruggieri, "YaDT: Yet another decision tree builder," in *ICTAI*. IEEE Computer Society, 2004, pp. 260–265.

[23] J. R. Quinlan and R. M. Cameron-Jones, "FOIL: A midterm report," in *ECML*, ser. LNCS, vol. 667. Springer, 1993, pp. 3–20.

[24] X. Yin and J. Han, "CPAR: classification based on predictive association rules," in *SDM*. SIAM, 2003, pp. 331–335.

[25] W. W. Cohen, "Fast effective rule induction," in *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 115–123.

[26] X. Yan and X. Su, *Linear regression analysis: theory and computing*. World Scientific, 2009.

[27] A. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Soviet Meth. Dokl.*, vol. 4, pp. 1035–1038, 1963.

[28] R. Tibshirani, "Regression shrinkage and selection via the lasso," *JRSS Series B (Methodological)*, pp. 267–288, 1996.

[29] I. Kononenko *et al.*, "An efficient explanation of individual classifications using game theory," *JMLR*, vol. 11, pp. 1–18, 2010.

[30] I. Guyon *et al.*, Eds., *Feature Extraction: Foundations and Applications*, ser. Studies in Fuzziness and Soft Computing. Springer, 2006, vol. 207.

[31] J. A. Olvera-López *et al.*, "A review of instance selection methods," *Artificial Intelligence Review*, vol. 34, no. 2, pp. 133–143, 2010.

[32] M. W. Craven *et al.*, "Using sampling and queries to extract rules from trained neural networks," in *JMLR*. Elsevier, 1994, pp. 37–45.

[33] D. Danks and A. J. London, "Algorithmic bias in autonomous systems," in *IJCAI*, 2017, pp. 4691–4697.

[34] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI*. MK, 1995, pp. 1137–1145.

[35] J. Kim, "Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap," *CSDA*, vol. 53, no. 11, p. 3735, 2009.

[36] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[37] S. Schwarz, M. Pawlik, and N. Augsten, "A new perspective on the tree edit distance," in *SISAP*. Springer, 2017, pp. 156–170.