Assessing the Stability of Interpretable Models

Riccardo Guidotti and Salvatore Ruggieri

KDDLab, University of Pisa and ISTI-CNR, Pisa, Italy
{guidotti,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 KDD process, which, in particular, comprises data collection and filtering. Selection bias in data collection or in data pre-processing may affect the model learned. Although model induction algorithms are designed to learn to generalize, they pursue optimization of predictive accuracy. It remains unclear how interpretability is instead impacted. We conduct an experimental analysis to investigate whether interpretable models are able to cope with data selection bias as far as interpretability is concerned.

1 Introduction

Interpretable machine learning models aim at trading-off predictive performance with human-comprehensibility and verifiability. They are also used to explain the global logic of inscrutable black-box machine learning models [11]. This is achieved by a form of reverse engineering, where interpretable models are trained on a (typically, random) sample of the population. If the interpretable model can accurately reproduce the black-box decisions, it can be used as a surrogate model of the black-box. The KDD process of learning an interpretable model includes a number of design choices:

- on the set of features to use (*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 data to use (*instance selection*). Instance generation in black-box explanation can be purely random [26], or adopt refined approaches, e.g., genetic algorithms [10].
- on the machine learning model to use (*model selection*), on the specific learning algorithm, and on its parameters. An experimental phase is typically part of the design, with the purpose of selecting the most accurate model.

Such a process must be accountable, namely the interpretable (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

[1] IEEE Glossary of Ethics of Autonomous and Intelligent Systems: ethicsinaction.ieee.org.

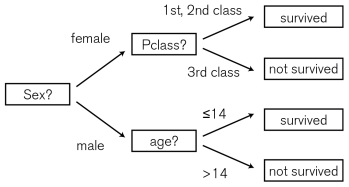
training data and/or design choices may lead to different interpretable models and decision explanations. Stability of interpretable models is then a key property towards accountability of machine learning (black-box) decision making.

We present 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 [9,14]. We conclude that, in order to pursue accountability, interpretable model's learning processes should comprise a *stability impact assessment* which is currently missing in guidelines and best-practices.

2 Related Work

Stability is a property of the output of a learning process. The representation of the output can be an intensional (a classifier) or extensional (its predictions). *Extensional stability* of classifier predictions was modelled by [31] through a measure of agreement among predictions. He proposed a $m \times 2$-fold cross-validation approach. 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 [13]. Bias 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, [2] proved that generalization error can be bound by (expectation of) stability.

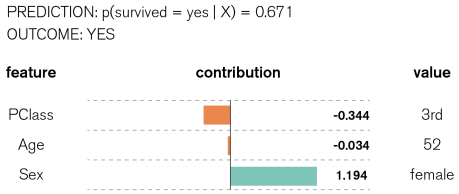
Measures of interpretability of classifiers must, however, be necessarily syntactic, since this is the level at which humans interface with models. This paper concentrates then on *intensional stability* of a learning process. One of the early studies regards the impact of training set size on the accuracy of decision trees [22], showing that the best performance can be achieved with sufficiently many data, after which there is no convenience to add more. Coping with variability of classifiers due to random noise in data has been tackled by adopting statistical tests for validating split tests at decision nodes [16], or by adopting split methods that account for almost equal split attributes (sources of instability) [20]. Finally, decision tree simplification is another class of approaches that trade-off accuracy with simplicity [4]. Intensional stability of feature selection method considered variability in the set of features selected [15,21]. 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 [15], 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 [1].



(a) decision tree

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

(b) rule-based classifier



(c) linear model

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

3 Setting the Stage

Interpretable models. Interpretability is the ability to explain or to provide meaning in terms understandable to a human [11]. 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., Figure. 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 learning algorithms: **CART** (Classification and Regression Trees) [3] as implemented by the *scikit-learn* Python library[2], and **C4.5** [24] as implemented by the computationally efficient *YaDT* (Yet another Decision Tree) system[3] [27]. **C4.5** performs multi-way univariate splits and it includes tree simplification (error-based pruning). We do not consider instead the split condition of [20], 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 Figure. 1 (b)). In this work, we consider the **FOIL** (First Order Inductive Learner) [25] and **CPAR** (Classification based on Predictive Association Rules) [33] algorithms, as implemented by the *LUCS-KDD* library[4]. The former generates a very small number of rules, but has lower accuracy than the latter. Similarly to DTs, and for space reasons, we restrict to sets of conjunctive classification rules. Another natural choice would have been RIPPER [5], which unfortunately produces ordered sequences of conjunctive rules. I.e., we compare DT and RB classifiers with the same expressivity.

[2] http://scikit-learn.org.
[3] http://pages.di.unipi.it/ruggieri/software.
[4] https://cgi.csc.liv.ac.uk/∼frans/KDD/Software.

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 Figure. 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**) [32], and its regularized forms **RIDGE** [30] and **LASSO**, [29] as implemented by the *scikit-learn* library. They are commonly used in black-box explanation approaches [19,26].

Measuring interpretability and stability. Several syntactic measures of interpretability were considered in the literature. Structural measures (SM) look at models in isolation, and quantify the degree of syntactic (intensional) interpretability of a model 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 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[5]: 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 [21]: 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.

Feature and instance selection. Feature selection (FS) [12] and instance selection (IS) [23] 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

[5] 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.

by preventing overfitting. Both techniques are widely used in reverse engineering of black-box models. We consider the following standard methods, as provided by the *scikit-learn*[6] 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 (we consider 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: $pct{=}10$).

For instance selection, we consider the following methods, as provided by the *imbalanced-learn*[7] 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 [6,10].

4 Evaluation Framework

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

First, any observational research project must account for variability/bias in data collection [7]. Following standard methodology for estimating accuracy of classifiers [18,17], we adopt a 5-repetition of 10-fold stratified cross-validation as a methodology 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 $\widehat{\cdot}$). 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 preprocessing, 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 6–16 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

[6] http://scikit-learn.org/stable/modules/feature_selection.
[7] http://contrib.scikit-learn.org/imbalanced-learn.

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 $i \in \{1, \ldots, 5\}$ **do**
  $F \leftarrow stratified10Fold(X, y)$ // 10 fold partitioning
  for $k \in \{1, \ldots, 10\}$ **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 $\mathcal{E}$

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

Table 1. Experimental datasets.

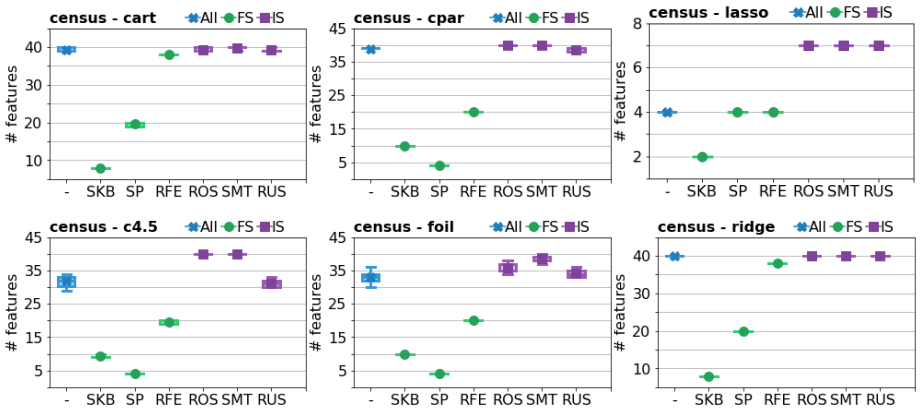


Fig. 2. Dataset: *census*. Measure: number of features.

the processed data. In Algorithm 1, models are stored in the set $\mathcal{M}$. 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 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 framework are intended to support a number of accountability questions that the data analyst should answer before deploying a classification model, namely, how sensitive is the interpretability of a classification model to changes: in data collection? in feature selection? in instance selection? in model selection?

5 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. Table 1 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[8] by integrating external libraries (*YaDT* and *LUCS-KDD*)

[8] Source code and datasets available at *url hidden for blind review*.

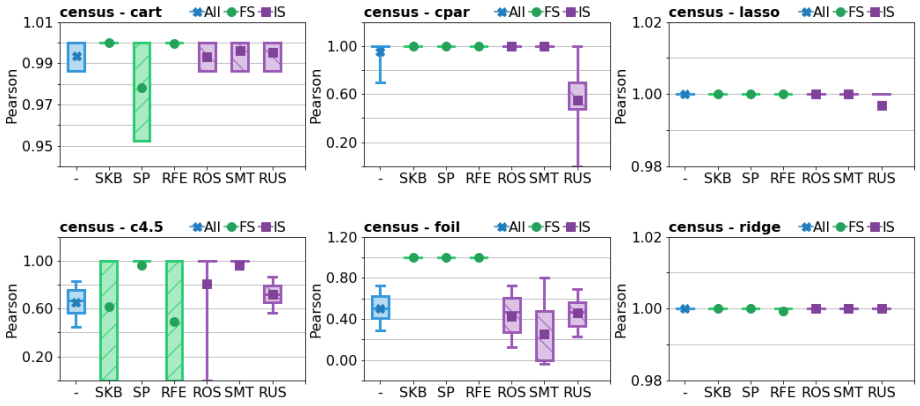


Fig. 3. Dataset: *census*. Measure: Pearson's correlation.

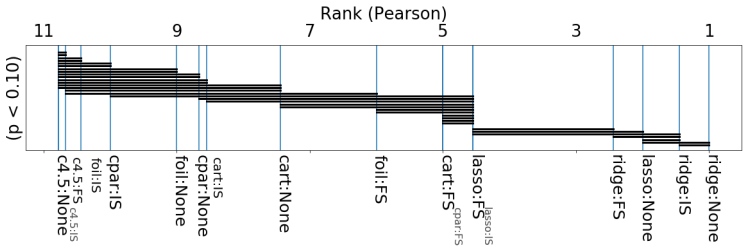


Fig. 4. Comparison of all model's rank w.r.t. Pearson's correlation against each other with the Nemenyi test. Groups of classifiers that are not significantly different at 90% significance level are connected. Best ranks on the right.

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[9].

Common measures. Let us start focusing on the number of features used by a classification model. Figure 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**[10]). Each plot shows the boxplots for no preprocessing ("-"), 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 sta-

[9] **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.

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

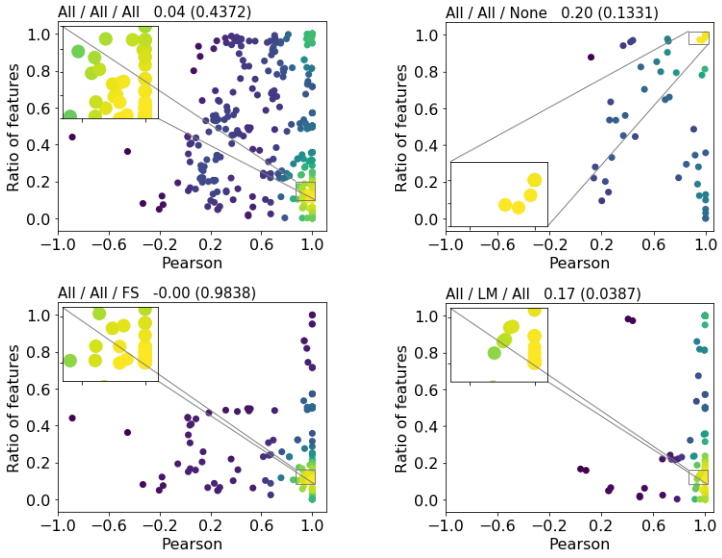


Fig. 5. Scatter density plots of Pearson's correlation vs Ratio of features used. Each point's coordinates are the mean values over the 50 experimental folds of some dataset for the following conditions (left to right, top to bottom): experiments for all datasets/classifiers/pre-processing; experiments 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.

bility. 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 of training data can better measure variability of the set of features used by a classifier. Figure 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. Finally, 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.

Statistical comparison of models' 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 Figure 4 (see [8] 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. *In-*

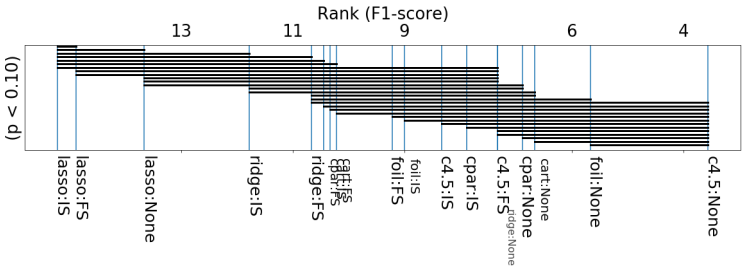


Fig. 6. Same as Figure 4 but w.r.t. the F1-score.

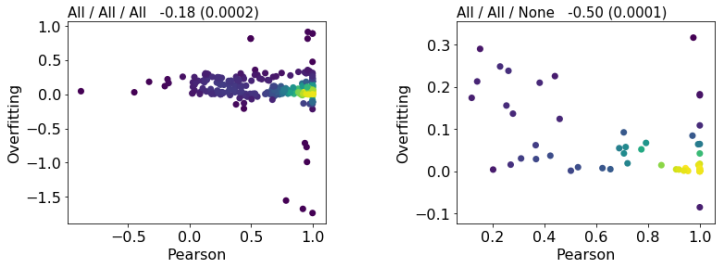


Fig. 7. Pearson vs Overfitting. Left: all exp.; right: no pre-proc. On top: correlation and p-value.

stance 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.

Stability-interpretability. We summarize the relation between interpretability and stability through the scatter density plots in Figure 5, where Pearson's correlation (stability) is plotted against 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-preprocessing methods increase the correlation (higher stability means lower interpretability). Feature selection does not impact on the correlation. Finally, the right bottom plot shows some positive correlation for linear models at 95% significance level.

Stability-accuracy. Figure 6 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 Figure 4. Decision trees and rule-based classifiers are the best performing. Linear models are at the bottom of the ranking. The adoption of instance selection does not improve ranks of classifiers. In summary, for the interpretable models considered here, *stability and accuracy are contrasting objectives, which then require a trade-off analysis*.

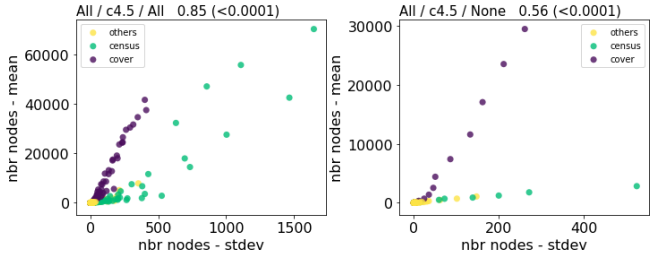


Fig. 8. Scatter plot of stability (deviation) vs interpretability (mean) w.r.t. number of nodes, left: all **C4.5** experiments; right: **C4.5** with no pre-processing.

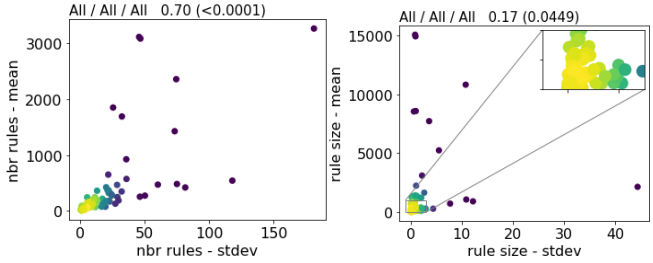


Fig. 9. Scatter density plot of stability (deviation) vs interpretability (mean) w.r.t. number of rules (left) and size of rules (right) in RB models. All experiments.

Stability-overfitting. Let us now contrast stability with overfitting. Figure 7 reports scatter plots of stability vs overfitting, defined as the relative difference of F1 accuracy between training and 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. 7). This is somehow expected, due to the bias-variance decomposition [13]. In summary, *stability and overfitting appear to be contrasting objectives*.

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 exponentially varying the stopping parameter in tree construction from m=2 (default value) to m=half of the size of the dataset. 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. Figure 8 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, Figure 9 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

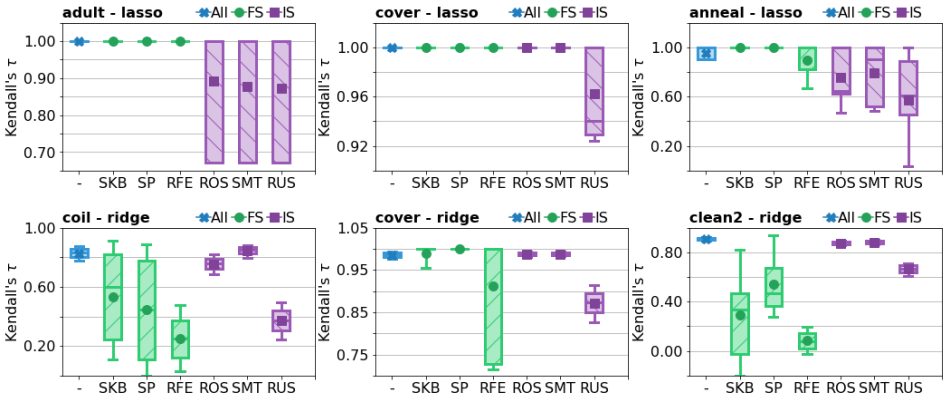


Fig. 10. Measure: Kendall's τ. Models: LM.

scatter plots in Figure 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.

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. Figure 10 reports the boxplots of τ's values over the 50 experimental folds for a few datasets and methods. **LASSO** is generally more stable than **RIDGE** (high values of τ), due to the fact it uses less features. 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**.

Discussion. Experimental results highlight 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 difficult to interpret. Thus, 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.

6 Conclusion

Our main contributions consist of a framework for intensional stability impact assessment, and experiments parametric to 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.

Several extensions of the approach 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 [28] is even more fine-grained than decision tree size. Second, with the exception of Figure 8, 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.

References

1. Alvarez-Melis, D., Jaakkola, T.S.: On the robustness of interpretability methods. CoRR **abs/1806.08049** (2018)
2. Bousquet, O., Elisseeff, A.: Stability and generalization. Journal of Machine Learning Research **2**, 499–526 (2002)
3. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. CRC press (1984)
4. Breslow, L.A., Aha, D.W.: Simplifying decision trees: A survey. The Knowledge Engineering Review **12**, 1–40 (1997)
5. Cohen, W.W.: Fast effective rule induction. In: ICML. pp. 115–123. Morgan Kaufmann (1995)
6. Craven, M.W., Shavlik, J.W.: Using sampling and queries to extract rules from trained neural networks. In: Machine Learning Proceedings 1994, pp. 37–45. Elsevier (1994)
7. Danks, D., London, A.J.: Algorithmic bias in autonomous systems. In: IJCAI. pp. 4691–4697. ijcai.org (2017)
8. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7**, 1–30 (2006)
9. Freitas, A.A.: Comprehensible classification models: A position paper. ACM SIGKDD explorations newsletter **15**(1), 1–10 (2014)
10. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local rule-based explanations of black box decision systems. CoRR **abs/1805.10820** (2018)
11. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Pedreschi, D., Giannotti, F.: A survey of methods for explaining black box models. ACM CSUR **51**(5), 93:1–93:42 (Aug 2018)
12. Guyon, I., Nikravesh, M., Gunn, S., Zadeh, L.A. (eds.): Feature Extraction: Foundations and Applications, Studies in Fuzziness and Soft Computing, vol. 207. Springer (2006)
13. Hastie, T., Tibshirani, R., Friedman, J.H.: The elements of statistical learning: data mining, inference, and prediction, 2nd Edition. Springer series in statistics, Springer (2009)
14. Huysmans, J., et al.: An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. Decision Support Systems **51**(1), 141–154 (2011)
15. Kalousis, A., Prados, J., Hilario, M.: Stability of feature selection algorithms: A study on high-dimensional spaces. Knowl. Inf. Syst. **12**(1), 95–116 (2007)

16. Katz, G., Shabtai, A., Rokach, L., Ofek, N.: ConfDTree: A statistical method for improving decision trees. J. Comput. Sci. Technol. **29**(3), 392–407 (2014)
17. Kim, J.: Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. Computational Statistics & Data Analysis **53**(11), 3735–3745 (2009)
18. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI. pp. 1137–1145. Morgan Kaufmann (1995)
19. Kononenko, I., et al.: An efficient explanation of individual classifications using game theory. Journal of Machine Learning Research **11**, 1–18 (2010)
20. Li, R., Belford, G.G.: Instability of decision tree classification algorithms. In: KDD. pp. 570–575. ACM (2002)
21. Nogueira, S., Brown, G.: Measuring the stability of feature selection. In: ECML/PKDD (2). Lecture Notes in Computer Science, vol. 9852, pp. 442–457. Springer (2016)
22. Oates, T., Jensen, D.: The effects of training set size on decision tree complexity. In: Proc. of Int. Conf. on Machine Learning (ICML 1997). pp. 254–262. Morgan Kaufmann (1997)
23. Olvera-López, J.A., Carrasco-Ochoa, J.A., Martínez Trinidad, J.F., Kittler, J.: A review of instance selection methods. Artif. Intell. Rev. **34**(2), 133–143 (2010)
24. Quinlan, J.R.: C4. 5: Programs for Machine Learning. Elsevier (1993)
25. Quinlan, J.R., Cameron-Jones, R.M.: FOIL: A midterm report. In: ECML. Lecture Notes in Computer Science, vol. 667, pp. 3–20. Springer (1993)
26. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": Explaining the predictions of any classifier. In: KDD. pp. 1135–1144. ACM (2016)
27. Ruggieri, S.: YaDT: Yet another decision tree builder. In: ICTAI. pp. 260–265. IEEE Computer Society (2004)
28. Schwarz, S., Pawlik, M., Augsten, N.: A new perspective on the tree edit distance. In: SISAP. Lecture Notes in Computer Science, vol. 10609, pp. 156–170. Springer (2017)
29. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288 (1996)
30. Tikhonov, A.: Solution of incorrectly formulated problems and the regularization method. Soviet Meth. Dokl. **4**, 1035–1038 (1963)
31. Turney, P.D.: Technical note: Bias and the quantification of stability. Machine Learning **20**(1-2), 23–33 (1995)
32. Yan, X., Su, X.: Linear regression analysis: theory and computing. World Scientific (2009)
33. Yin, X., Han, J.: CPAR: classification based on predictive association rules. In: SDM. pp. 331–335. SIAM (2003)