Hierarchical Shrinkage: Improving the Accuracy and Interpretability of Tree-Based Methods

Abhineet Agarwal [*1] **Yan Shuo Tan** [*1] **Omer Ronen** [1] **Chandan Singh** [2] **Bin Yu** [1 2]

Abstract

Tree-based models such as decision trees and random forests (RF) are a cornerstone of modern machine-learning practice. To mitigate overfitting, trees are typically regularized by a variety of techniques that modify their structure (e.g. pruning). We introduce Hierarchical Shrinkage (HS), a post-hoc algorithm that does not modify the tree structure, and instead regularizes the tree by shrinking the prediction over each node towards the sample means of its ancestors. The amount of shrinkage is controlled by a single regularization parameter and the number of data points in each ancestor. Since HS is a post-hoc method, it is *extremely fast*, compatible with *any* tree-growing algorithm, and can be used synergistically with other regularization techniques. Extensive experiments over a wide variety of real-world datasets show that HS substantially increases the predictive performance of decision trees, even when used in conjunction with other regularization techniques. Moreover, we find that applying HS to each tree in an RF often improves accuracy, as well as its interpretability by simplifying and stabilizing its decision boundaries and SHAP values. We further explain the success of HS in improving prediction performance by showing its equivalence to ridge regression on a (supervised) basis constructed of decision stumps associated with the internal nodes of a tree. All code and models are released in a full-fledged package available on Github.[1]

[*]Equal contribution [1]Department of Statistics, UC Berkeley, Berkeley, California, USA [2]EECS Department, UC Berkeley, Berkeley, California, USA. Correspondence to: Bin Yu <binyu@berkeley.edu>.

Proceedings of the 39^{th} *International Conference on Machine Learning*, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

[1]HS is integrated into the `imodels` package github.com/csinva/imodels (Singh et al., 2021) with an sklearn-compatible API. Experiments for reproducing the results here can be found at github.com/Yu-Group/imodels-experiments.

1. Introduction

Decision tree models, used for supervised learning since the 1960s (Morgan & Sonquist, 1963; Messenger & Mandell, 1972; Quinlan, 1986), have recently attained renewed prominence because they embody key elements of interpretability: shallow trees are easily described and visualized, and can even be implemented by hand. While the precise definition and utility of interpretability have been a subject of much debate (Murdoch et al., 2019; Doshi-Velez & Kim, 2017; Rudin, 2019; Rudin et al., 2021), all agree that it is an important notion in high-stakes decision-making, such as medical-risk assessment and criminal justice. For this reason, decision trees have been widely applied in both areas (Steadman et al., 2000; Kuppermann et al., 2009; Letham et al., 2015; Angelino et al., 2017).

By far the most popular decision tree algorithm is Classification and Regression Trees (CART) (Breiman et al., 1984). These can be ensembled to form a Random Forest (RF) (Breiman, 2001) or used as weak learners in Gradient Boosting (GB) (Friedman, 2001); both algorithms have achieved state-of-the-art performance over a wide class of prediction problems (Caruana & Niculescu-Mizil, 2006; Caruana et al., 2008; Fernández-Delgado et al., 2014; Olson et al., 2018; Hooker & Mentch, 2021), and are implemented in popular machine learning packages such as `ranger` (Wright et al., 2017) and `scikit-learn` (Pedregosa et al., 2011). Variants of these algorithms, such as iterative random forest for finding stable interactions (Basu et al., 2018), have found use in scientific applications.

In view of the widespread use of tree-based methods, we seek to provide a new lens on their regularization. Decision trees are known to obey traditional statistical wisdom in that they need to be regularized to prevent overfitting. In practice, this is carried out by specifying an early stopping condition for tree growth, such as a maximum depth, or alternatively, pruning the tree after it is grown (Friedman et al., 2001). These procedures, however, only regularize tree models via their *tree structure*, and it is usually taken for granted that the prediction over each leaf should be the average response of the training samples it contains. We show that this can be very limiting: *shrinking these predictions in a hierarchical fashion can significantly reduce*

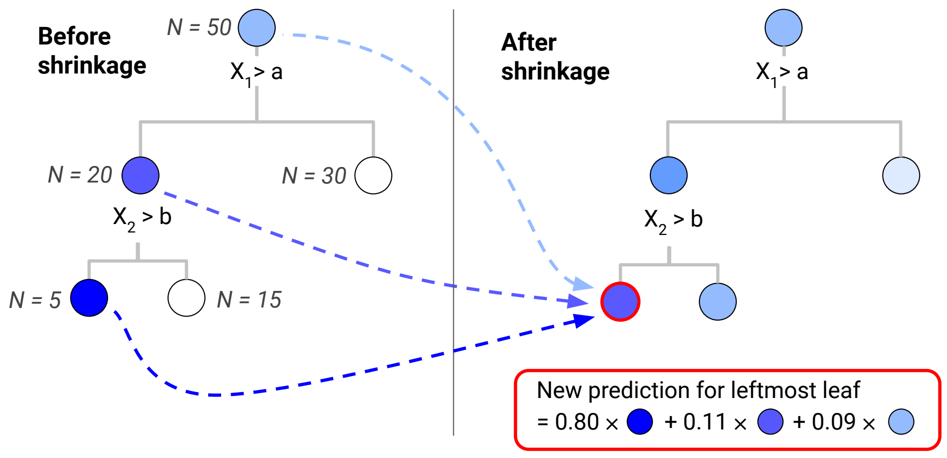

Figure 1. Diagram showing how HS works for a toy tree model. The predictions over each tree node (including internal nodes) gets shrunk toward the mean responses over each of its ancestors.

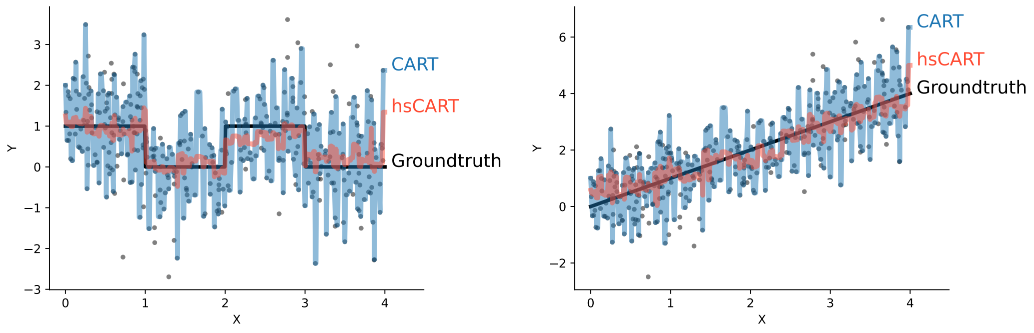

Figure 2. Example of HS for toy univariate regression problems. HS regularizes model predictions to improve estimates in noisy leaves that have few samples. CART is fit to the data in the blue dots and then HS is applied posthoc (hsCART).

generalization error in both regression and classification settings (e.g. see Fig 2).

Meanwhile, trees used in an RF are usually not explicitly regularized and interpolate the data by being grown to purity (e.g. see the default settings of `scikit-learn` and `ranger`). Instead, RF attempts to prevent overfitting by relying upon the randomness injected into the algorithm during tree growth, which acts as a form of implicit regularization (Breiman, 2001; Mentch & Zhou, 2019). We show that apart from this implicit regularization, more regularization, in the form of hierarchical shrinkage, does improve generalization, and allows us to use a smaller ensemble for many data sets.

Equally important, regularizing RFs also improves the quality of their post-hoc interpretations. RFs are usually interpreted via their feature and interaction importances, which have been used to provide scientific insight in areas such as remote sensing and genomics (Svetnik et al., 2003; Evans et al., 2011; Belgiu & Drăguţ, 2016; Díaz-Uriarte & De Andres, 2006; Boulesteix et al., 2012; Chen & Ishwaran, 2012; Basu et al., 2018; Behr et al., 2020). The reproducibility and scientific meaning of such interpretations become questionable when the underlying RF model has poor predictive performance (Murdoch et al., 2019), or when they are highly sensitive to data perturbations (Yu, 2013). We show that HS improves the interpretability of RF by both simplifying and stabilizing its decision boundaries and SHAP values (Lundberg & Lee, 2017) on a number of real-world data sets.

Our proposed method, which we call *Hierarchical Shrinkage* (HS), is an extremely fast and simple yet effective algorithm for the *post-hoc* regularization of *any* tree-based model. It does not alter the tree structure, and instead replaces the average response (or prediction) over a leaf in the tree with a *weighted average of the mean responses over the leaf and each of its ancestors* (see Fig 1). The weights depend on the number of samples in each leaf, and are controlled by a single regularization parameter λ that can be tuned efficiently via generalized cross validation. HS is agnostic to the way the tree is constructed and can be applied post-hoc to trees constructed with greedy methods such as CART and C4.5 (Quinlan, 2014), as well optimal decision trees grown via dynamic programming or integer optimization techniques (Lin et al., 2020; Aghaei et al., 2021).

A more naive form of shrinkage, which we call *leaf-based shrinkage* (LBS), appears as part of XGBoost (Chen & Guestrin, 2016): whenever a new tree is grown, the average response (prediction) over each leaf is shrunk directly towards the sample mean of the responses. LBS also occurs[2] in Bayesian Additive Regression Trees (BART) (Chipman et al., 2010), which grows an ensemble of trees via a backfitting MCMC algorithm. Comparing LBS to HS on several real-world datasets shows that HS has uniformly better predictive performance than LBS.

We explain the advantages of HS by building on recent work which uses decision stumps associated to each interior node of a tree to construct a new (supervised) feature representation (Klusowski, 2021). The original tree model is recovered as the linear model obtained by regressing the responses on the supervised features. We show that HS is exactly the *ridge regression* solution in this supervised feature space, while LBS can also be viewed as ridge regression, but with a different (supervised) feature space (of the same dimension) that relies only on the leaf nodes. This allows us to use ridge regression calculations heuristically to partially explain both the reasonableness of the shrinkage scaling in HS, as well as our empirical evidence that HS achieves consistently better predictive accuracy than LBS (see Sec 3).

While this paper was under review, we discovered two other relevant but somewhat obscure papers. Hastie & Pregibon (1990) described a method for shrinking the predictions of trees, but proposed to do so in a recursive fashion. This scheme is different from the one proposed in our paper. Chipman & McCulloch (2000) defined a hierarchical prior for a Bayesian CART model, which parameterizes the tree model in terms of "mean shifts" (differences between the means of child and parent nodes). This seems to have a closer connection to HS than to LBS, but it is unclear whether the resulting likelihood (see equation (14) in their paper) has the same form as HS.

The rest of the paper is organized as follows. Sec 2 gives a formal statement of HS, and discusses several computational considerations. Sec 3 discuss the interpretation of HS as ridge regression on the supervised features. Sec 4 presents the results of extensive numerical experiments on simulated and real world data sets that illustrate the gains in prediction accuracy from applying the method. Sec 5 shows how HS improves the interpretability of RFs.

[2]When conditioned on the structure of a given tree, as well as all other trees in the ensemble, the posterior distribution for the contribution of a leaf node is a product of Gaussian likelihood functions centered at the model residuals as well as a Gaussian prior. A simple calculation shows that the posterior mean can be obtained from the residual mean via LBS.

2. The Hierarchical Shrinkage (HS) Algorithm

Throughout this paper, we work in the supervised learning setting where we are given a training set $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, from which we learn a tree model $\hat{f}$ for the regression function. Given a query point $\mathbf{x}$, let $\mathfrak{t}_L \subset \mathfrak{t}_{L-1} \subset \cdots \subset \mathfrak{t}_0$ denote its leaf-to-root path, with $\mathfrak{t}_L$ and $\mathfrak{t}_0$ representing its leaf node and the root node respectively. For any node $\mathfrak{t}$, let $N(\mathfrak{t})$ denote the number of samples it contains, and $\hat{\mathbb{E}}_{\mathfrak{t}}\{y\}$ the mean response. The tree model prediction can be written as the telescoping sum:

$$\hat{f}(\mathbf{x}) = \hat{\mathbb{E}}_{\mathfrak{t}_0}\{y\} + \sum_{l=1}^{L}\Big(\hat{\mathbb{E}}_{\mathfrak{t}_l}\{y\} - \hat{\mathbb{E}}_{\mathfrak{t}_{l-1}}\{y\}\Big).$$

HS transforms $\hat{f}$ into a shrunk model $\hat{f}_\lambda$ via the formula:

$$\hat{f}_\lambda(\mathbf{x}) := \hat{\mathbb{E}}_{\mathfrak{t}_0}\{y\} + \sum_{l=1}^{L}\frac{\hat{\mathbb{E}}_{\mathfrak{t}_l}\{y\} - \hat{\mathbb{E}}_{\mathfrak{t}_{l-1}}\{y\}}{1 + \lambda/N(\mathfrak{t}_{l-1})}, \qquad (1)$$

where λ is a hyperparameter chosen by the user, for example by cross validation. We emphasize that HS maintains the tree structure, and *only* modifies the prediction over each leaf node (see Fig 1).

Since HS continues to make a constant prediction over each leaf node, our method thus comprises a one-off modification of these values. This can be computed in $O(m)$ time, where m is the total number of nodes in the tree. No other aspects of the underlying data structure are modified, with test time prediction occurring in exactly the same way as in the original tree. Moreover, our method HS does not even need to see the original training data, and only requires access to the fitted tree model. These features make it extremely lightweight and easy to implement, as we have done in the open-source package `imodels` (Singh et al., 2021). By applying HS to each tree in an ensemble, it can be generalized to methods such as RF and gradient boosting.

While not typically done, it is possible to regularize RFs via other hyperparameters such as maximum tree depth. Tuning these hyperparameters, however, requires refitting the RF at every value in a grid. This quickly becomes computationally expensive in a cross-validation (CV) set up, even for moderate dataset sizes.[3] In contrast, since HS is applied

[3]Many popular tree-building algorithms such as CART have a run time of $O(pn^2)$ for constructing a binary tree.

post-hoc, *we only need to fit the RF once per CV fold*, leading to potentially enormous time savings. In addition, due to the connection between our method and ridge regression, it is even possible to get away with fitting the RF only *once* by using generalized cross-validation (Golub et al., 1979)[4].

We also note the formula for LBS:

$$\hat{f}^l_\lambda(\mathbf{x}) := \hat{\mathbb{E}}_{\mathfrak{t}_0}\{y\} + \frac{\hat{\mathbb{E}}_{\mathfrak{t}_L}\{y\} - \hat{\mathbb{E}}_{\mathfrak{t}_0}\{y\}}{1 + \lambda/N(\mathfrak{t}_L)}. \tag{2}$$

Expanding this into a telescoping sum similar to (1), we see that the major difference between the two formulas is that whereas LBS shrinks each term by the same factor, HS shrinks each term by a different amount, with the amount of shrinkage controlled by the number of samples in the ancestor. This increased flexibility leads to better prediction performance for the final model, as evidenced by our results presented in the next section.

3. HS as Ridge Regression on Supervised Features

Recent work by Klusowski (2021) showed that decision tree algorithms can be viewed as a two-step process, where the first step comprises supervised feature learning, and the second step fits a linear models on the collection of learnt features.

To see this, consider a tree model $\hat{f}$, with a fixed indexing of its interior nodes $\{\mathfrak{t}_0, \mathfrak{t}_1, \ldots, \mathfrak{t}_{m-1}\}$. We first associate to each node $\mathfrak{t}$ the decision stump

$$\psi_{\mathfrak{t}}(\mathbf{x}) = \frac{N(\mathfrak{t}_R)\mathbf{1}\{\mathbf{x} \in \mathfrak{t}_L\} - N(\mathfrak{t}_L)\mathbf{1}\{\mathbf{x} \in \mathfrak{t}_R\}}{\sqrt{N(\mathfrak{t}_L)N(\mathfrak{t}_R)}}, \tag{3}$$

where $\mathfrak{t}_L$ and $\mathfrak{t}_R$ denote the left and right children of $\mathfrak{t}$ respectively. This is a tri-valued function that is positive on the left child, negative on the right child, and zero everywhere else. Concatenating the decision stumps together yields a supervised feature map via $\Psi(\mathbf{x}) = \big(\psi_{\mathfrak{t}_0}(\mathbf{x}), \ldots, \psi_{\mathfrak{t}_{m-1}}(\mathbf{x})\big)$ and a transformed training set $\Psi(\mathcal{D}_n) \in \mathbb{R}^{n\times m}$. One can easily check that these feature vectors are orthogonal in $\mathbb{R}^n$, and furthermore that their squared ℓ_2 norms are the number of samples contained in their corresponding nodes: $\|\psi_{\mathfrak{t}_i}\|^2 = N(\mathfrak{t}_i)$. Klusowski (2021) showed (see Lemma 3.2 therein) that we have functional equality between the tree model and the kernel regression model with respect to the supervised feature map Ψ, or in other words, $\hat{f}(\mathbf{x}) = \hat{\boldsymbol{\beta}}^T\Psi(\mathbf{x})$, where $\hat{\boldsymbol{\beta}} = \Psi(\mathcal{D}_n)^\dagger\mathbf{y}$. An easy extension of his proof yields the following result.

[4]This allows for efficient computation of leave-one-out cross-validation error, which can be used to select λ, without refitting the RF.

Theorem 1. *Let $\hat{\boldsymbol{\beta}}_\lambda$ be the solution to the ridge regression problem*

$$\min_{\boldsymbol{\beta}}\left\{\sum_{i=1}^n\Big(\boldsymbol{\beta}^T\Psi(\mathbf{x}_i) - y_i\Big)^2 + \lambda\|\boldsymbol{\beta}\|^2\right\}. \tag{4}$$

We have the functional equality $\hat{f}_\lambda(\mathbf{x}) = \hat{\boldsymbol{\beta}}_\lambda^T\Psi(\mathbf{x})$.

Proof. See Appendix S2. □

Since the decision stumps (3) are orthogonal, we can decompose (4) into m independent univariate ridge regression problems, one with respect to each node $\mathfrak{t}$:

$$\min_{\beta}\left\{\sum_{i=1}^n(\beta\psi_{\mathfrak{t}}(\mathbf{x}_i) - y_i)^2 + \lambda\beta^2\right\}. \tag{5}$$

Next, we use this connection of HS to ridge regression to argue heuristically that *the same λ works well for each regression subproblem* (5). This helps to justify our choice of denominator for each term in the HS formula (1) (a different choice would have led to a rescaling of the features $\psi_{\mathfrak{t}_i}$.)

Assume for the moment that the tree structure and hence the feature map is independent of the responses, which can be achieved via sample splitting. This is known in the literature as the "honesty condition", and has been widely used to simplify the analysis of tree-based methods (Athey & Imbens, 2016). Define $\boldsymbol{\beta}_* \in \mathbb{R}^m$ to have the value

$$\boldsymbol{\beta}_*(\mathfrak{t}) := \frac{\sqrt{N(\mathfrak{t}_L)N(\mathfrak{t}_R)}}{N(\mathfrak{t})}(\mathbb{E}\{y \mid \mathfrak{t}_L\} - \mathbb{E}\{y \mid \mathfrak{t}_R\}), \tag{6}$$

for the coordinate associated with each node $\mathfrak{t}$. For any query point $\mathbf{x}$, $\boldsymbol{\beta}_*^T\Psi(\mathbf{x})$ gives the mean response over the leaf $\mathfrak{t}(\mathbf{x})$ containing it. Furthermore, knowing $\Psi(\mathbf{x})$ is equivalent to knowing the leaf containing $\mathbf{x}$. Putting these two facts together show that the population residuals $r_i := y_i - \boldsymbol{\beta}_*^T\Psi(\mathbf{x})$ satisfy $\mathbb{E}\{r_i \mid \Psi(\mathbf{x}_i)\} = 0$, so that we have a generative linear model, in which we can calculate that the optimal regularization parameter for (5) is equal to $\lambda_{opt}(\mathfrak{t}) = \sigma^2(\mathfrak{t})/\boldsymbol{\beta}_*(\mathfrak{t})^2$, where $\sigma^2(\mathfrak{t})$ is roughly equal to the conditional variance[5] of the residual over $\mathfrak{t}$.

Given the connection between impurity and residual variance, if the tree model $\hat{f}$ considered in this section is grown using an impurity decrease stopping condition, we should expect the residual variance to be relatively similar over all leaves, so that $\sigma^2(\mathfrak{t})$ also does not vary too much over different nodes. Meanwhile (see Appendix S2.2)

$$(\mathbb{E}\{y \mid \mathfrak{t}_L\} - \mathbb{E}\{y \mid \mathfrak{t}_R\})^2 \approx \mathrm{diam}(\mathfrak{t})^2 \approx 2^{-2\mathrm{depth}(\mathfrak{t})/p} \tag{7}$$

[5]More precisely, it is equal to $\frac{N(\mathfrak{t}_L)}{N(\mathfrak{t})}\mathrm{Var}\{r \mid \mathfrak{t}_R\} + \frac{N(\mathfrak{t}_R)}{N(\mathfrak{t})}\mathrm{Var}\{r \mid \mathfrak{t}_L\}$.

where p is the dimension of the original feature space. Since the maximum depth is typically $O(\log n)$, $\lambda_{opt}(\mathfrak{t})$ also does not vary too much across different nodes, and we do not lose too much by using a common value of λ across all the univariate subproblems (5) corresponding to different decision stump features.

A more naive (supervised) tree-based feature map is the one-hot encoding of an original feature vector obtained by treating the leaf index as a categorical variable. We denote this using Ξ. While Ξ can be obtained from Ψ via an invertible linear transformation, the two maps result in different kernels, and thus different ridge regression problems. Indeed, the ridge regression solution with respect to Ξ is equivalent to performing LBS on the tree model. The leaf indicator features are also orthogonal, so we may similarly decompose this ridge regression problem into independent univariate subproblems, one for each leaf. However, in this case, the population regression vector $\boldsymbol{\beta}_*^l$, for which $\boldsymbol{\beta}_*^{lT}\Xi(\mathbf{x})$ gives the expected response over each leaf, has coordinates equal to the population expectation over the leaves: $\boldsymbol{\beta}_*^l(\mathfrak{t}) = \mathbb{E}\{y \mid \mathfrak{t}\}$. As such, the optimal regularization parameters for different leaf nodes could be very different, and we lose more by having to use a common value of λ.

In practice, sample splitting is rarely done, and the feature map depends on the responses. Nonetheless, we believe that the heuristics detailed above continue to hold to a certain extent as shown in our experimental results.

4. HS Improves Predictive Performance on Real-World Datasets

4.1. Data Overview

In this section, we study the performance of HS on a collection of classification and regression datasets selected as follows. For classification, we consider a number of datasets used in the classic Random Forest paper (Breiman, 2001; Asuncion & Newman, 2007), one (*Breast cancer* with id=13) from the openML repository, as well as two (*Juvenile* and *Recidivism*) that are commonly used to evaluate rule-based models (Wang, 2019). For regression, we consider all datasets used by Breiman (2001) with at least 200 samples, as well as a variety of data-sets from the PMLB benchmark (Romano et al., 2020) ranging from small to large sample sizes. Table 1 displays the number of samples and features present in each dataset, with more details provided in Appendix S1. In all cases, $2/3$ of the data is used for training (hyperparameters are selected via 3-fold CV on this set) and $1/3$ of the data is used for testing.

	Name	Samples	Features
Classification	Heart	270	15
	Breast cancer	286	9
	Haberman	306	3
	Ionosphere (Sigillito et al., 1989)	351	34
	Diabetes (Smith et al., 1988)	768	8
	German credit	1000	20
	Juvenile (Osofsky, 1997)	3640	286
	Recidivism	6172	20
Regression	Friedman1 (Friedman, 1991)	200	10
	Friedman3 (Friedman, 1991)	200	4
	Diabetes (Efron et al., 2004)	442	10
	Geographical music	1059	117
	Red wine	1599	11
	Abalone (Nash et al., 1994)	4177	8
	Satellite image (Romano et al., 2020)	6435	36
	CA housing (Pace & Barry, 1997)	20640	8

Table 1. Real-world datasets analyzed here for classification (top panel) and regression (bottom panel).

4.2. HS Improves Prediction Performance for Commonly Used Tree Methods

The prediction performance results for classification and regression are plotted in Fig 4A and Fig 4B respectively, with the number of leaves, a measure of model complexity, plotted on the x-axis. We consider trees grown using four different techniques: CART, CART with cost-complexity pruning (CCP), C4.5 (Quinlan, 2014), and GOSDT (Lin et al., 2020), a method that grows optimal trees in terms of the cost-complexity penalized misclassification loss. To reduce clutter, we only display the classification results for CART and CART with CCP in Fig 4A/B and defer the results for C4.5 (Fig S3) and GOSDT (Appendix S4.2) to the appendix.

For each of the four tree-growing methods, we grow a tree to a fixed number of leaves m,[6] for several different choices of $m \in \{2, 4, 8, 12, 15, 20, 24, 28, 30, 32\}$ (in practice, m would be pre-specified by a user or selected via cross-validation). For each tree, we compute its prediction performance before and after applying HS, where the regularization parameter for HS is selected from the set $\lambda \in \{0.1, 1.0, 10.0, 25.0, 50.0, 100.0\}$ via cross-validation. Results for each experiment are averaged over 10 random data splits. We observe that HS (solid lines in Fig 4A,B) *does not hurt prediction in any of our data sets, and often leads to substantial performance gains.* For example, taking $m = 15$, we observe an average increase in relative predictive performance (measured by AUC) of $6.2\%, 6.5\%, 8\%$ for HS applied to CART, CART with CCP, and C4.5 respectively for the classification data sets. For the regression data sets with $m = 15$, we observe an aver-

[6]For CART (CCP), we grow the tree to maximum depth, and tune the regularization parameter to yield m leaves.

age relative increase in R^2 performance of 9.8% and 10.1% for CART and CART with CCP respectively.

As expected, the improvements are more significant for smaller datasets, and tend to increase with the number of leaves (e.g. the top row of Fig 4A and Fig 4B). For larger datasets, we start to see substantial improvements using HS at even larger numbers of leaves (Appendix S4.4).

The fact that improvements hold for CART (CCP) shows that the effect of HS is not entirely replicated by tree structure regularization, and instead, *the two regularization methods can be used synergistically*. Indeed, applying HS can lead to the selection of a larger tree. Since tree models are sometimes used for subgroup search, larger trees from HS could allow for the discovery of otherwise undetected subgroups.

Fig 3 shows a simulation result analyzing the bias-variance tradeoff for CART with and without HS. Here, data is generated from a linear model with Gaussian noise added during training (see Appendix S3 for experimental details, and other simulations). While predictive performance curves are often U-shaped because of the bias-variance tradeoff, those for HS are monotonic since HS is able to effectively reduce variance. The optimal regularization parameter λ decreases with the total number of leaves; this is corroborated by our calculations in Sec 3.

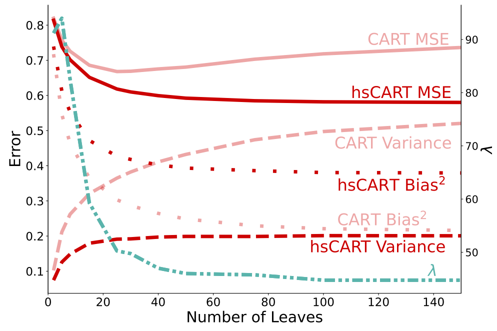

Figure 3. Test error for CART with HS (hsCART) stays low as the number of leaves increases, whereas CART test error increases due to overfitting. Data is simulated from a linear model with Gaussian noise.

4.3. HS Outperforms LBS

We next compare the performance of HS to that of leaf-based shrinkage (LBS), which is used in XGBoost. Fig 4C shows that hsCART tends to outperform CART (LBS), when repeating the same experiments as in Fig 4A); regression results are pushed to Appendix S4.3 due to space constraints.

4.4. HS Improves Prediction Performance for RF

As mentioned earlier, trees in an RF are typically grown to purity without any constraints on depth or node size. Nonetheless, Mentch & Zhou (2019) argues that the `mtry` parameter[7], which controls the degree of feature subselection, "serves much the same purpose as the shrinkage penalty in explicitly regularized regression procedures like lasso and ridge regression." This parameter is typically set to a default value[8], and is not tuned. We compare the performance of regularizing RF via HS against that obtained by controlling maximum tree depth or `mtry`, tuning the hyperparameter for each method via cross validation. We repeat this for several different choices of B, the number of trees in the RF, and average the results over 10 random data splits.

The results, displayed in Fig 4D, show that *HS significantly improves the prediction accuracy of RF* across the datasets we considered. Moreover, HS clearly outperforms the two other RF regularization methods (using `depth` and `mtry`) in all datasets. This is especially promising because HS is also the fastest and easiest method to implement, as it does not require refitting the RF. Moreover, *hsRF tends to achieve its maximum performance with fewer trees* than RF without regularization; as a consequence, RF with HS is often able to achieve the same performance with an ensemble that is five times smaller, allowing us to achieve large savings in computational resources.

We also compare hsRF to the predictive performance of Bayesian Additive Regression Trees (BART) (Chipman et al., 2010), and observe that hsRF and BART are comparable in terms of prediction performance. However, hsRF is much faster to fit than BART (typically 10-15 times faster) and we can also apply HS to BART (see Appendix S4.5).

Additionally, we also investigate the effect of applying HS to gradient boosted trees. The results, displayed in Fig S8, indicate that applying HS slightly improves the performance of GBTs for smaller datasets, while performing similarly for larger datasets. This is unsurprising since the benefit of HS is most pronounced for smaller datasets and when growing deeper trees.

5. HS Improves RF Interpretations by Simplifying and Stabilizing Them

In addition to improving predictive performance, HS reduces variance and removes sampling artifacts, resulting in (i) simplified boundaries, and (ii) stabilized feature impor-

[7]This parameter is denoted `mtry` in `ranger` and `max_features` in `scikit-learn`.

[8]Typically $\sqrt{p}$ for classification and $p/3$ for regression, where p is the number of features.

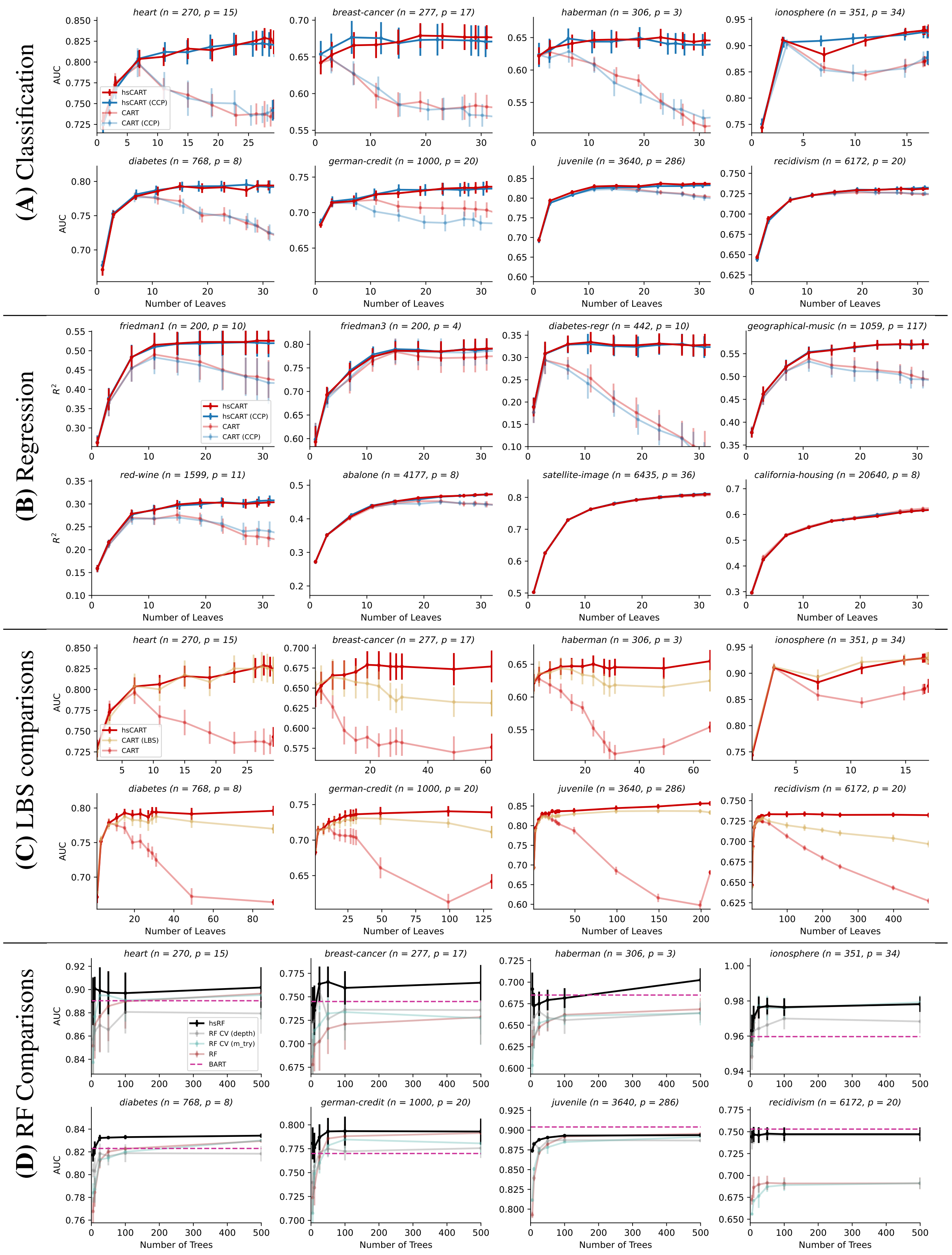

Figure 4. Hierarchical Shrinkage (solid lines) often improves predictive performance across various datasets, particularly for small datasets. **(A)** Top two rows show results for classification datasets (measured by AUC of the ROC curve) and **(B)** the next two rows show results for regression datasets (measured by R^2). HS often significantly improves the performance over CART, CART with CCP, and **(C)** leaf-based shrinkage. **(D)** HS even improves results for Random Forests as a function of the number of trees. Across all panels, errors bars show standard error of the mean computed over 10 random data splits. Note that the y-axis scales differ across plots.

tance scores, (iii) making it easier to interpret interactions in the model.

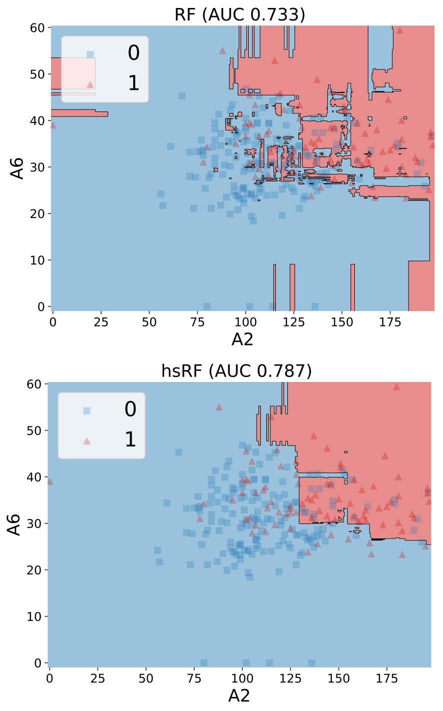

Figure 5. Comparison of decision boundary learned by RF vs hsRF on the Diabetes dataset, when fitted using only two features. HS prevents overfitting by creating a smoother, simpler decision boundary, resulting in improved performance and interpretability.

Fig 5 shows an example of simplified decision boundaries. On the diabetes dataset (Smith et al., 1988), RF can achieve strong performance (AUC 0.733) even when fitted to only two features. When HS is applied to this RF, the performance increases (to an AUC of 0.787), but the decision boundary also becomes considerably smoother and less fragmented. Since these two features are the only inputs to the model, these smooth boundaries enable a user to identify much clearer regions for high-risk predictions. Appendix S5.1 plots decision boundaries for all 8 classification datasets, showing that HS consistently makes boundaries smoother.[9]

In models with many features, post-hoc interpretations such as SHAP values (Lundberg & Lee, 2017) can be used to understand how the model makes its predictions. Fig 6 shows that HS improves the stability of SHAP values with respect to resampling of the dataset. In this experiment, we randomly choose 50 samples in the breast-cancer dataset to hold out, and for each of 100 iterations, we randomly select two thirds of the remaining samples and train an RF on this reduced dataset. For each held-out sample, we measure the variance of its SHAP values per feature across the 100 iterations. We then average the variance per feature across all 50 held-out samples, with these values plotted in Fig 6 for RF with HS and without. We observe that the variances of the SHAP values for RF with HS are substantially smaller than those for RF without HS. Moreover, these improvements in stability persist even for datasets such as Heart, Diabetes, and Ionosphere, for which HS does not greatly improve prediction performance (see SHAP stability plots for all datasets in Fig S13 and Fig S14). When SHAP values are more stable, we can have more faith that they reflect true rather than spurious patterns in the data generating process.

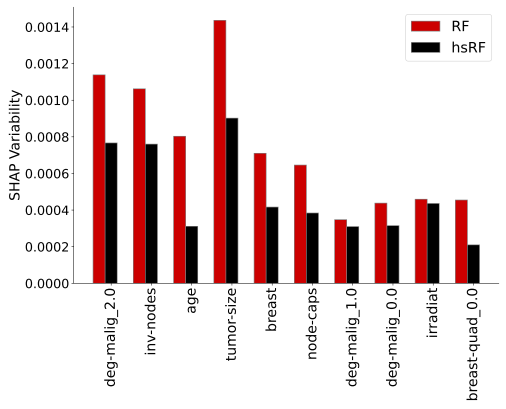

Figure 6. Comparison of SHAP plots learnt by Random Forests on the breast-cancer dataset before / after applying HS. HS displays lower variability across different data perturbations (without dropping in predictive performance), indicating enhanced stability.

Fig 7 investigates the SHAP values for an RF trained on the breast-cancer dataset. After applying HS, the SHAP values for each feature have tighter clusters. Each cluster comprises a group of samples for which the feature contributes a similar amount to the predicted response, and hence can be interpreted without taking into account feature interactions, thereby reducing cognitive burden. More globally, the clustering effect suggests that HS improves prediction performance by regularizing some unnecessary interactions in the model, making the fitted function closer to being additive. Appendix S5.2 shows the SHAP plots for all 8 classification datasets, showing that HS consistently leads to more clustered SHAP values.

6. Discussion

HS is a fast yet powerful regularization procedure that can be applied to any tree-based model without changing its

[9]To obtain these plots, we fit an RF on each dataset using only the two most important features, as measured using mean-decrease-impurity (MDI).

structure. In our experiments, HS never hurts prediction performance, and often leads to substantial gains in both predictive performance and interpretability. HS is partly motivated by previous non-minimax-optimal generalization lower bounds for decision trees that predict using average responses over each leaf (Tan et al., 2021), pointing to a possible limitation of averaging. HS allows us to break this inferential barrier by pooling information from multiple leaves.

The work here naturally suggests many exciting future directions regarding the regularization of trees and RFs. First, replacing ridge regression with more sophisticated linear methods such as lasso or elastic net can result in diverse, promising regularization methods. In addition, HS could improve other structured rule-based models, such as rule lists and tree sums (Tan et al., 2022; Nasseri et al., 2022).

Meanwhile, we have only scratched the surface of the relationships between regularization, robustness, and interpretability. Indeed, recent work showed that ridge regression helps with robust generalization (Donhauser et al., 2021). The connection between HS and ridge regression suggests that HS could also operate similarly, and this is supported by our observation that HS results in simpler and smoother decision boundaries. Hence, we conjecture that HS could improve the predictive performance of RF with respect to covariate perturbations such as adversarial examples or covariate shift. Moreover, the improved clustering and stability of SHAP values after applying HS suggest that regularization via HS could stabilize other popular feature importance measures, and thus support better feature selection.

Acknowledgements

We gratefully acknowledge partial support from NSF TRIPODS Grant 1740855, DMS-1613002, 1953191, 2015341, IIS 1741340, ONR grant N00014-17-1-2176, the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370, NSF grant 2023505 on Collaborative Research: Foundations of Data Science Institute (FODSI), the NSF and the Simons Foundation for the Collaboration on the Theoretical Foundations of Deep Learning through awards DMS-2031883 and 814639, and a grant from the Weill Neurohub.

References

Aghaei, S., Gómez, A., and Vayanos, P. Strong optimal classification trees. *arXiv preprint arXiv:2103.15965*, 2021.

Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., and Rudin, C. Learning certifiably optimal rule lists for categorical data. *arXiv preprint arXiv:1704.01701*, 2017.

Asuncion, A. and Newman, D. Uci machine learning repository, 2007.

Athey, S. and Imbens, G. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

Basu, S., Kumbier, K., Brown, J. B., and Yu, B. iterative random forests to discover predictive and stable high-order interactions. *Proceedings of the National Academy of Sciences*, pp. 201711236, 2018.

Behr, M., Kumbier, K., Cordova-Palomera, A., Aguirre, M., Ashley, E., Butte, A., Arnaout, R., Brown, J. B., Preist, J., and Yu, B. Learning epistatic polygenic phenotypes with boolean interactions. *bioRxiv*, 2020.

Belgiu, M. and Drăguţ, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31, 2016.

Boulesteix, A.-L., Janitza, S., Kruppa, J., and König, I. R. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):493–507, 2012.

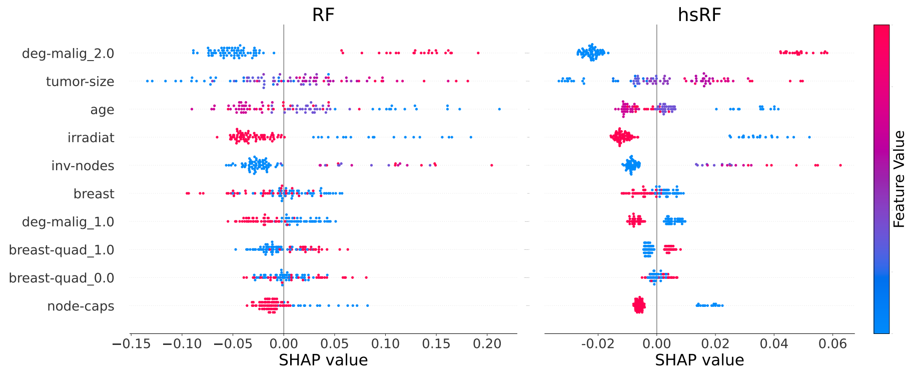

Figure 7. Comparison of SHAP values for RF on the breast-cancer dataset before/after applying HS with features arranged by importance. Each point represents a single sample in the dataset. HS leads to more clustered SHAP values for each feature, reflecting less heterogentity in the SHAP values of each feature.

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

Breiman, L., Friedman, J., Olshen, R., and Stone, C. J. *Classification and regression trees*. Chapman and Hall/CRC, 1984.

Caruana, R. and Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine learning*, pp. 161–168, 2006.

Caruana, R., Karampatziakis, N., and Yessenalina, A. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th International Conference on Machine learning*, pp. 96–103, 2008.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Chen, X. and Ishwaran, H. Random forests for genomic data analysis. *Genomics*, 99(6):323–329, 2012.

Chipman, H. and McCulloch, R. E. Hierarchical priors for bayesian cart shrinkage. *Statistics and Computing*, 10(1):17–24, 2000.

Chipman, H., George, E., and McCulloch, R. Bayesian ensemble learning. *Advances in neural information processing systems*, 19, 2006.

Chipman, H. A., George, E. I., and McCulloch, R. E. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

Díaz-Uriarte, R. and De Andres, S. A. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

Donhauser, K., Tifrea, A., Aerni, M., Heckel, R., and Yang, F. Interpolation can hurt robust generalization even when there is no noise. *Advances in Neural Information Processing Systems*, 34, 2021.

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

Evans, J. S., Murphy, M. A., Holden, Z. A., and Cushman, S. A. Modeling species distribution and change using random forest. In *Predictive Species and Habitat Modeling in Landscape Ecology*, pp. 139–159. Springer, 2011.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

Friedman, J., Hastie, T., Tibshirani, R., et al. *The Elements of Statistical Learning*, volume 1. Springer Series in Statistics New York, 2001.

Friedman, J. H. Multivariate adaptive regression splines. *The annals of statistics*, pp. 1–67, 1991.

Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232, 2001.

Golub, G. H., Heath, M., and Wahba, G. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979. doi: 10.1080/00401706.1979.10489751. URL `https://www.tandfonline.com/doi/abs/10.1080/00401706.1979.10489751`.

Hastie, T. and Pregibon, D. *Shrinking trees*. AT & T Bell Laboratories, 1990.

Hill, J. L. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Hooker, G. and Mentch, L. Bridging Breiman's brook: From algorithmic modeling to statistical learning. *Observational Studies*, 7(1):107–125, 2021.

Klusowski, J. M. Universal consistency of decision trees in high dimensions. *arXiv preprint arXiv:2104.13881*, 2021.

Kuppermann, N., Holmes, J. F., Dayan, P. S., Hoyle, J. D., Atabaki, S. M., Holubkov, R., Nadel, F. M., Monroe, D., Stanley, R. M., Borgialli, D. A., et al. Identification of children at very low risk of clinically-important brain injuries after head trauma: a prospective cohort study. *The Lancet*, 374(9696): 1160–1170, 2009.

Letham, B., Rudin, C., McCormick, T. H., Madigan, D., et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*, 9(3):1350–1371, 2015. doi: 10.1214/15-aoas848.

Lin, J., Zhong, C., Hu, D., Rudin, C., and Seltzer, M. Generalized and scalable optimal sparse decision trees. In *International Conference on Machine Learning*, pp. 6150–6160. PMLR, 2020.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pp. 4768–4777, 2017.

Mentch, L. and Zhou, S. Randomization as regularization: a degrees of freedom explanation for random forest success. *arXiv preprint arXiv:1911.00190*, 2019.

Messenger, R. and Mandell, L. A modal search technique for predictive nominal scale multivariate analysis. *Journal of the American Statistical Association*, 67(340):768–772, 1972.

Morgan, J. N. and Sonquist, J. A. Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58(302):415–434, 1963.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. doi: 10.1073/pnas.1900654116.

Nash, W. J., Sellers, T. L., Talbot, S. R., Cawthorn, A. J., and Ford, W. B. The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. *Sea Fisheries Division, Technical Report*, 48:p411, 1994.

Nasseri, K., Singh, C., Duncan, J., Kornblith, A., and Yu, B. Group probability-weighted tree sums for interpretable modeling of heterogeneous data. *arXiv preprint arXiv:2205.15135*, 2022.

Olson, R. S., Cava, W. L., Mustahsan, Z., Varik, A., and Moore, J. H. Data-driven advice for applying machine learning to bioinformatics problems. In *Biocomputing 2018: Proceedings of the Pacific Symposium*, pp. 192–203. World Scientific, 2018.

Osofsky, J. D. The effects of exposure to violence on young children (1995). *Carnegie Corporation of New York Task Force on the Needs of Young Children; An earlier version of this article was presented as a position paper for the aforementioned corporation.*, 1997.

Pace, R. K. and Barry, R. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

Pedregosa, F., Varoquaux, G. ë. l., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011. URL `http://jmlr.org/papers/v12/pedregosa11a.html`.

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

Quinlan, J. R. *C4. 5: programs for machine learning*. Elsevier, 2014.

Romano, J. D., Le, T. T., La Cava, W., Gregg, J. T., Goldberg, D. J., Ray, N. L., Chakraborty, P., Himmelstein, D., Fu, W., and Moore, J. H. Pmlb v1. 0: an open source dataset collection for benchmarking machine learning methods. *arXiv preprint arXiv:2012.00058*, 2020.

Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. doi: 10.1038/s42256-019-0048-x.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., and Zhong, C. Interpretable machine learning: Fundamental principles and 10 grand challenges. *arXiv preprint arXiv:2103.11251*, 2021.

Sigillito, V. G., Wing, S. P., Hutton, L. V., and Baker, K. B. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3):262–266, 1989.

Singh, C., Nasseri, K., Tan, Y. S., Tang, T., and Yu, B. imodels: a python package for fitting interpretable models. *Journal of Open Source Software*, 6(61):3192, 2021. doi: 10.21105/joss.03192. URL `https://doi.org/10.21105/joss.03192`.

Smith, J. W., Everhart, J. E., Dickson, W., Knowler, W. C., and Johannes, R. S. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the annual symposium on computer application in medical care*, pp. 261. American Medical Informatics Association, 1988.

Steadman, H. J., Silver, E., Monahan, J., Appelbaum, P., Robbins, P. C., Mulvey, E. P., Grisso, T., Roth, L. H., and Banks, S. A classification tree approach to the development of actuarial violence risk assessment tools. *Law and Human Behavior*, 24 (1):83–100, 2000.

Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., and Feuston, B. P. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of Chemical Information and Computer Sciences*, 43(6): 1947–1958, 2003.

Tan, Y. S., Agarwal, A., and Yu, B. A cautionary tale on fitting decision trees to data from additive models: generalization lower bounds. *arXiv preprint arXiv:2110.09626*, 2021.

Tan, Y. S., Singh, C., Nasseri, K., Agarwal, A., and Yu, B. Fast interpretable greedy-tree sums (figs). *arXiv preprint arXiv:2201.11931*, 2022.

Wang, T. Gaining free or low-cost interpretability with interpretable partial substitute. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6505–6514. PMLR, 09–15 Jun 2019. URL `http://proceedings.mlr.press/v97/wang19a.html`.

Wright, M. N., Ziegler, A., et al. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(i01), 2017.

Yu, B. Stability. *Bernoulli*, 19(4):1484–1500, 2013.

Supplement

S1. Further Details on Datasets Used

Name	Samples	Features	Class 0	Class 1	Majority class %
Heart	270	15	150	120	55.6
Breast cancer	286	9	196	81	70.8
Haberman	306	3	81	225	73.5
Ionosphere (Sigillito et al., 1989)	351	34	126	225	64.1
Diabetes (Smith et al., 1988)	768	8	500	268	65.1
German credit	1000	20	300	700	70.0
Juvenile (Osofsky, 1997)	3640	286	3153	487	86.6
Recidivism	6172	20	3182	2990	51.6

Table S2. Classification datasets (extended).

Name	Samples	Features	Mean	Std	Min	Max
Friedman1 (Friedman, 1991)	200	10	14.3	4.6	2.1	25.2
Friedman3 (Friedman, 1991)	200	4	1.3	0.4	0.0	1.6
Diabetes (Efron et al., 2004)	442	10	152.1	77.0	25.0	346.0
Geographical music	1059	117	0.0	1.0	-1.5	5.9
Red wine	1599	11	5.6	0.8	3.0	8.0
Abalone (Nash et al., 1994)	4177	8	9.9	3.2	1.0	29.0
Satellite image (Romano et al., 2020)	6435	36	3.7	2.2	1.0	7.0
California housing (Pace & Barry, 1997)	20640	8	2.1	1.2	0.1	5.0

Table S3. Regression datasets (extended).

S2. Theory

In this section, we provide a proof of Theorem 1 and the heuristic arguments discussed in Sec 3.

S2.1. Proof of Theorem 1

Throughout this proof, we denote the left and right children of a node $\mathfrak{t}_i$ by $\mathfrak{t}_{i,L}$ and $\mathfrak{t}_{i,R}$ respectively. Further, we assume WLOG that the sample mean satisfies $\hat{\mathbb{E}}_{\mathfrak{t}_0}\{y\} = 0$. Then using the well known solution to ridge regression, we have that

$$\hat{\boldsymbol{\beta}}_\lambda = (\Psi(\mathcal{D}_n)^T \Psi(\mathcal{D}_n) + \lambda I)^{-1} \Psi(\mathcal{D}_n)^T \mathbf{y}.$$

Because the feature vectors are orthogonal, the kernel matrix $\Psi(\mathcal{D}_n)^T\Psi(\mathcal{D}_n)$ is diagonal with entries $(\Psi(\mathcal{D}_n)^T\Psi(\mathcal{D}_n))_{ii} = N(\mathfrak{t}_i)$. Therefore, we have the following expansion for the i-th coordinate of $\hat{\boldsymbol{\beta}}_\lambda$

$$\begin{aligned}
\hat{\beta}_{\lambda,i} &= \frac{\langle \psi_{t_i}(\mathcal{D}_n), \mathbf{y}\rangle}{N(\mathfrak{t}_i) + \lambda} \\
&= \frac{N(\mathfrak{t}_{i,R})\sum_{\mathbf{x}_i \in \mathfrak{t}_{i,L}} y_i - N(\mathfrak{t}_{i,L})\sum_{\mathbf{x}_i \in \mathfrak{t}_{i,R}} y_i}{\sqrt{N(\mathfrak{t}_{i,L})N(\mathfrak{t}_{i,R})}(N(\mathfrak{t}_i)+\lambda)} \\
&= \frac{\sqrt{N(\mathfrak{t}_{i,L})N(\mathfrak{t}_{i,R})}}{N(\mathfrak{t}_i)+\lambda}\Big(\hat{\mathbb{E}}_{\mathfrak{t}_{i,L}}\{y\} - \hat{\mathbb{E}}_{\mathfrak{t}_{i,R}}\{y\}\Big) \\
&= \frac{\sqrt{N(\mathfrak{t}_{i,L})N(\mathfrak{t}_{i,R})}}{N(\mathfrak{t}_i)+\lambda}\left(\hat{\mathbb{E}}_{\mathfrak{t}_{i,L}}\{y\} - \frac{N(\mathfrak{t}_i)\hat{\mathbb{E}}_{\mathfrak{t}_i}\{y\}}{N(\mathfrak{t}_{i,R})} + \frac{\hat{\mathbb{E}}_{\mathfrak{t}_{i,L}}\{y\}N(\mathfrak{t}_{i,L})}{N(\mathfrak{t}_{i,R})}\right) \\
&= \sqrt{\frac{N(\mathfrak{t}_i)^2 N(\mathfrak{t}_{i,L})}{N(\mathfrak{t}_{i,R})(N(\mathfrak{t}_i)+\lambda)^2}}\Big(\hat{\mathbb{E}}_{\mathfrak{t}_{i,L}}\{y\} - \hat{\mathbb{E}}_{\mathfrak{t}_i}\{y\}\Big). \qquad (8)
\end{aligned}$$

Note that a similar formula holds with $\mathfrak{t}_{i,L}$ and $\mathfrak{t}_{i,R}$ exchanged.

Consider a query point $\mathbf{x}$ with leaf-to-root path $\mathfrak{t}_L \subset \mathfrak{t}_{L-1} \subset \cdots \subset \mathfrak{t}_0$, assuming WLOG that each descendant is always the left child of its ancestor. Using (8), the prediction at $\mathbf{x}$ can then be expanded as follows

$$\begin{aligned}\hat{f}_\lambda(\mathbf{x}) &= \Psi(\mathbf{x})^T \hat{\boldsymbol{\beta}}_\lambda \\ &= \sum_{l=0}^{L-1} \frac{N(\mathfrak{t}_{l,R})\langle \psi_{\mathfrak{t}_l}(\mathcal{D}_n), \mathbf{y}\rangle}{\sqrt{N(\mathfrak{t}_{l,L})N(\mathfrak{t}_{l,R})}(N(\mathfrak{t}_l)+\lambda)} \\ &= \sum_{l=0}^{L-1} \frac{\hat{\mathbb{E}}_{\mathfrak{t}_{l+1}}\{y\} - \hat{\mathbb{E}}_{\mathfrak{t}_l}\{y\}}{1+\lambda/N(\mathfrak{t}_l)}\end{aligned}$$

We see that the equation above is precisely that proposed for HS in (1).

S2.2. Heuristics for Equation (7)

Consider the generative model

$$y = f(\mathbf{x}) + \epsilon. \tag{9}$$

where $\mathbb{E}\{\epsilon \mid \mathbf{x}\} = 0$. Suppose for the moment that $f(\mathbf{x}) = \boldsymbol{\beta}^T\mathbf{x}$ is linear, so that

$$\mathbb{E}\{y \mid \mathfrak{t}_L\} - \mathbb{E}\{y \mid \mathfrak{t}_R\} = \boldsymbol{\beta}^T(\mathbb{E}\{\mathbf{x} \mid \mathfrak{t}_L\} - \mathbb{E}\{\mathbf{x} \mid \mathfrak{t}_R\}). \tag{10}$$

Now further assume that for some β_{min} and β_{max}, we have

$$\beta_{min} \leq |\beta_i| \leq \beta_{min}$$

for all i. Plugging these into (10) allows us to compute

$$\beta_{min}^2 \text{diam}(\mathfrak{t})^2 \leq (\mathbb{E}\{y \mid \mathfrak{t}_L\} - \mathbb{E}\{y \mid \mathfrak{t}_R\})^2 \leq \beta_{max}^2 \text{diam}(\mathfrak{t})^2. \tag{11}$$

Next, for a given node $\mathfrak{t}$, let its side lengths be denoted by l_i, with corresponding measure $\mu(\mathfrak{t}) = \prod_{i=1}^p l_j$. Assuming that all the side lengths of $\mathfrak{t}$ are similar, we have that the diameter of the node is roughly $\text{diam}(\mathfrak{t}) \approx \mu(\mathfrak{t})^{1/p}$. Furthermore, assuming that each node $\mathfrak{t}$ is split fairly evenly so that its left and right children have roughly equivalent measure, then the measure of the node $\mathfrak{t}$ is approximately $2^{-depth(\mathfrak{t})}$, which implies that $\text{diam}(\mathfrak{t}) \approx 2^{-depth(\mathfrak{t})/p}$. Substituting this into (11) gives us the heuristic claimed in (7).

For a more general C^1 regression function f, note that the C^1 assumption implies that f is approximately linear locally, i.e. when the nodes are small enough.

S3. Additional Simulations for Investigating the Bias-Variance Trade-off

In this section, we provide experimental details for our bias-variance trade-off simulation in Fig 3 as well as other simulations settings that we display below. Note that we reproduce Fig 3 in Fig S1 for ease of the reader.

Experimental Design: In Fig S1, we simulate data via a linear model $y = \sum_{i=1}^{10} x_i + \epsilon$ with $\mathbf{x} \sim \text{Unif}[0,1]^{50}$ and ϵ being drawn from a Gaussian or a Laplacian distribution for the left and right panel respectively, with noise variance $\sigma^2 = 0.01$ in both cases. In Fig S2, we simulate responses from a linear model with pairwise interactions $y = \sum_{i=1}^{10} x_i + x_1x_2 + x_5x_6 + x_{11}x_{12} + \epsilon$, and use the same noise models described above. In both experiments, we used a training set of 500 samples to fit CART and hsCART models with a prescribed number of leaves, varying this number across a grid. For each hsCART model, the regularization parameter λ was chosen on the training set via 3-fold cross-validation. Finally, we repeat this entire process 100 times with resampled datasets.

Evaluation: Denote the training set using $\mathcal{D}_n$ and the query point using $\mathbf{x}$. We define the MSE of a model $\hat{f}$ as

$$\mathbb{E}_{\mathbf{x},\mathcal{D}_n}\left\{\left(\hat{f}(\mathbf{x};\mathcal{D}_n) - f(\mathbf{x})\right)^2\right\}$$

and notice that it can be decomposed pointwise into squared bias and variance terms as follows:

$$\mathbb{E}_{\mathbf{x},\mathcal{D}_n}\left\{\left(\hat{f}(\mathbf{x};\mathcal{D}_n)-f(\mathbf{x})\right)^2\right\}=\mathbb{E}_{\mathbf{x}}\left\{\left(\mathbb{E}_{\mathcal{D}_n}\left\{\hat{f}(\mathbf{x};\mathcal{D}_n)\right\}-f(\mathbf{x})\right)^2\right\}+\mathbb{E}_{\mathbf{x}}\left\{\mathrm{Var}_{\mathcal{D}_n}\left\{\hat{f}(\mathbf{x})\right\}\right\}.$$

Our goal is to plot the approximate values of all three quantities for the CART and hsCART models under the settings described in the experimental design. To do this, we use a noiseless test set of 500 samples to compute an approximate expectation with respect to the query point $\mathbf{x}$. For each test set sample $\mathbf{x}$, we approximate $\mathbb{E}_{\mathcal{D}_n}\left\{\hat{f}(\mathbf{x};\mathcal{D}_n)\right\}$ using the mean prediction of the 100 models obtained from resampled training sets, and approximate $\mathrm{Var}_{\mathcal{D}_n}\left\{\hat{f}(\mathbf{x})\right\}$ by taking the variance of the predictions across the 100 models.

Results: We observe similar behavior in all four experiments: The the MSE curve for CART is U-shaped, whereas that for hsCART is monotonic. Furthermore, we see that this is due to hsCART being better at controlling its variance term. In all cases, the optimal regularization parameter λ decreases with the total number of leaves.

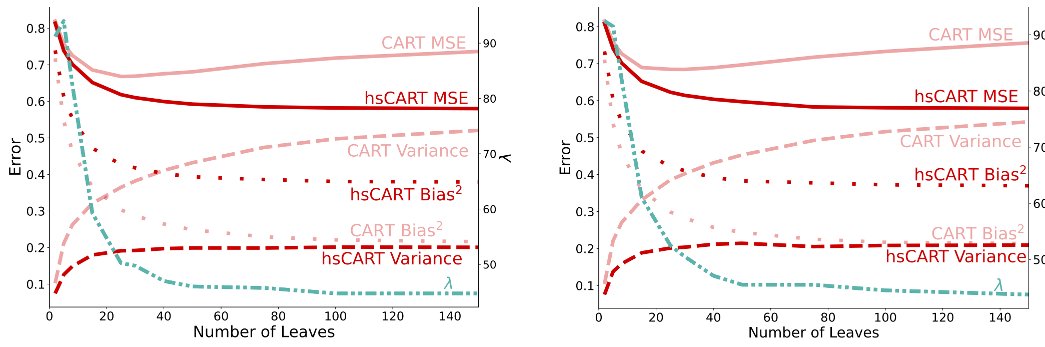

Figure S1. CART (HTS) test error rate for a linear model with gaussian (left)/laplacian (right) noise stays low as the number of leaves increases, whereas CART test error displays U-shaped bias-variance tradeoff as model complexity increases.

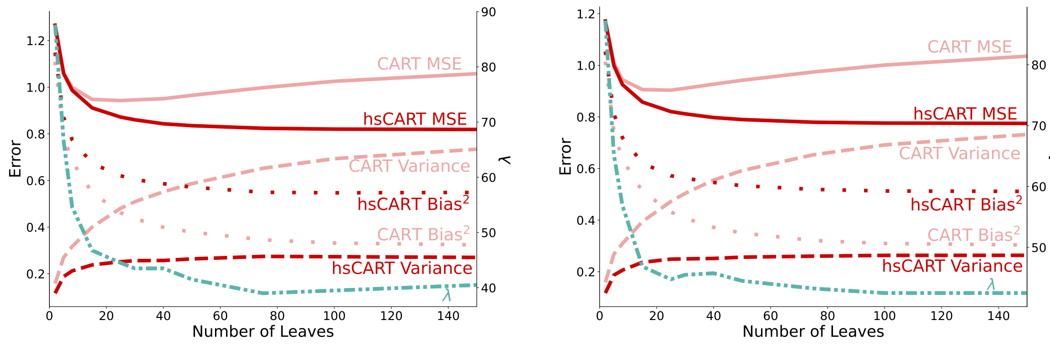

Figure S2. CART (HTS) test error rate for a linear model with pairwise interactions and gaussian (left)/laplacian (right) noise stays low as the number of leaves increases, whereas CART test error displays U-shaped bias-variance tradeoff as model complexity increases.

S4. Further Experimental Results for Predictive Performance

S4.1. C4.5 trees

In this section, we display prediction accuracy results on the classification datasets in Table 1 for tree models obtained via C4.5 (Quinlan, 2014) before/after applying HS post hoc. The experimental details are provided in Sec 4.2. We see that HS significantly improves prediction performance even with very few leaves.

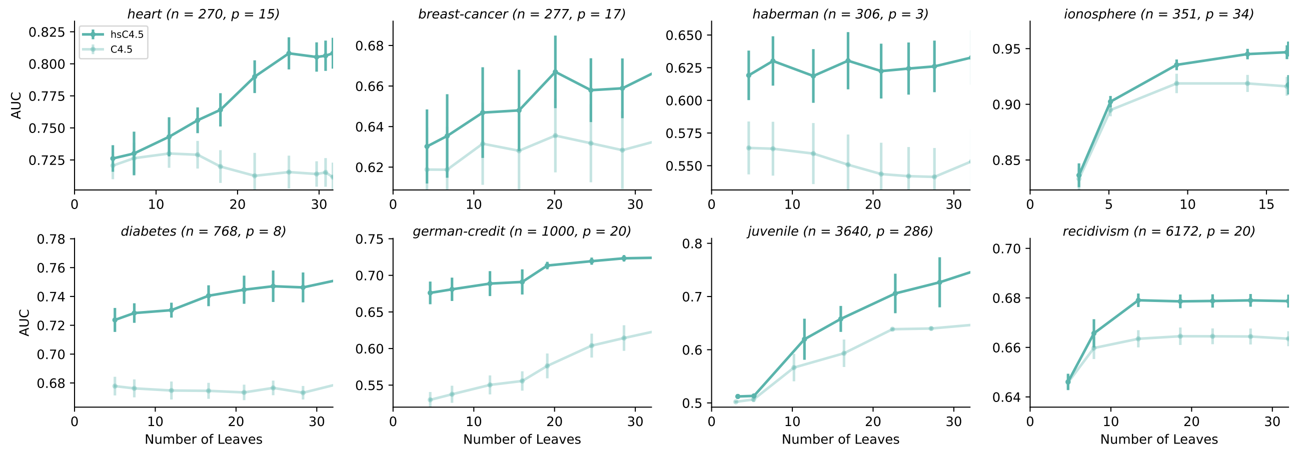

Figure S3. Classification results for C4.5. HS significantly improves prediction performance even with very few leaves. Errors bars show standard error of the mean computed over 10 random data splits

S4.2. Generalized Optimal Sparse Decision Trees (GOSDT)

In this section, we display prediction accuracy results on the classification datasets in Table 1 for tree models obtained via GOSDT (Lin et al., 2020) before/after applying HS post hoc. We performed the simulation to show that HS can be beneficial even for tree models that are obtained using global optimization (rather than greedy) techniques. We use the best tree found by GOSDT within one hour of running time for the algorithm, since the implementation of GOSDT we used[10] fails to converge for many datasets within a reasonable timeframe (24 hours). To make the running time more tractable, we use preprocessed each dataset by selecting the 5 most important features (using RF feature importance).[11] For GOSDT, we are unable to tune m exactly, and instead tune its cost-complexity regularization parameter. Fig S4 shows the results, and we again see that HS does not hurt prediction performance, although it also does not offer much improvement, possibly because the fitted trees are often very shallow.

[10]We used code from github.com/Jimmy-Lin/GeneralizedOptimalSparseDecisionTrees.

[11]If this step is omitted then the fitted tree is almost always extremely shallow (with 3 leaves or less) or GOSDT may fail to due to a memory error. In fact, even with this step, GOSDT only outputs the root node for many datasets when the regularization parameter is set to anything larger than 0. We omit these datasets from our results.

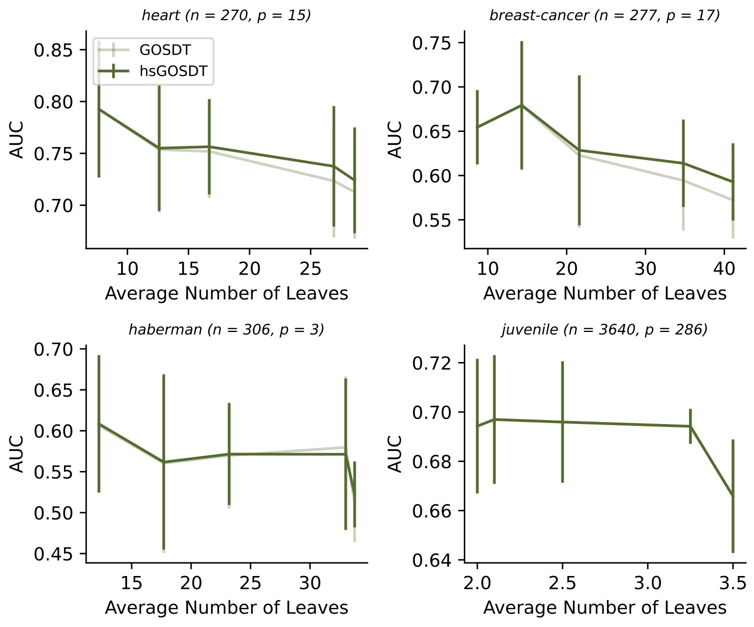

Figure S4. HS performance across selected datasets. HS is as good as and sometimes slightly improves GOSDT.

S4.3. Comparison between HS and LBS for Regression Datasets

In this section, we investigate the prediction performance of CART with the alternate shrinkage scheme LBS to CART with HS on the regression datasets in Table 1 with experimental details provided in Sec 4.3. We see that, just like in classification, HS outperforms LBS.

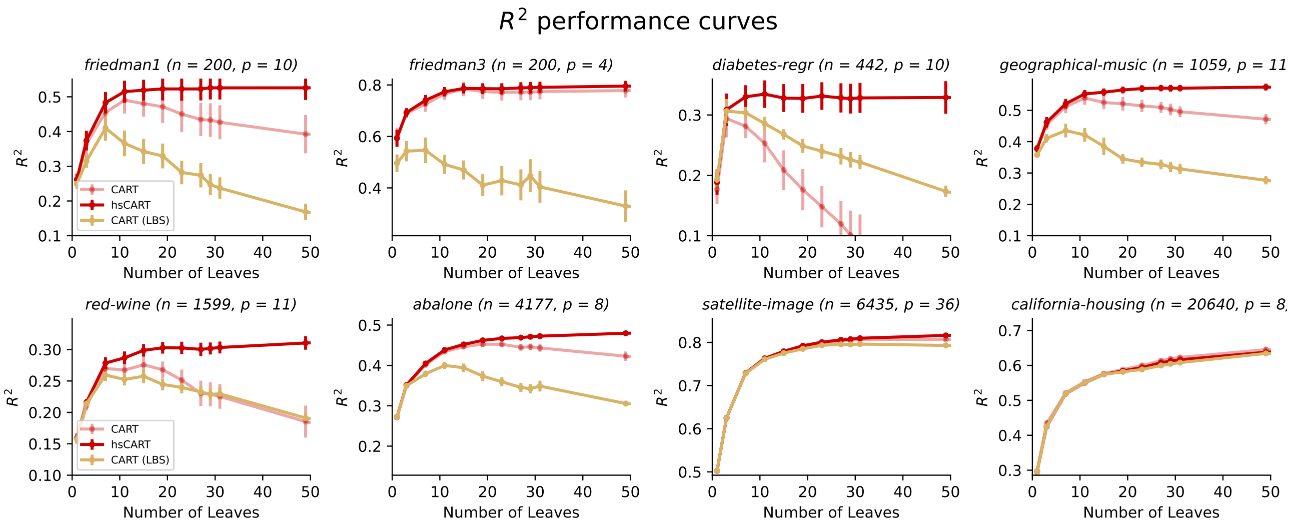

Figure S5. HS performs better than LBS for regression datasets in Table 1. Error bars show the standard error of the mean computed over 10 random data splits.

S4.4. Deeper CART trees

In this section, we investigate the prediction performance of decision tree algorithms: (i) CART, (ii) C4.5, (iii) CART with CCP before/after applying HS while growing deeper trees for all the datasets in Table 1 and Table 1. We observe that the performance of the baselines drop dramatically while the performance of HS continuously improves as the ratio of number of leaves to samples increases. This is explained by the connection between HS and ridge regression on a supervised basis, as discussed in Sec 3.

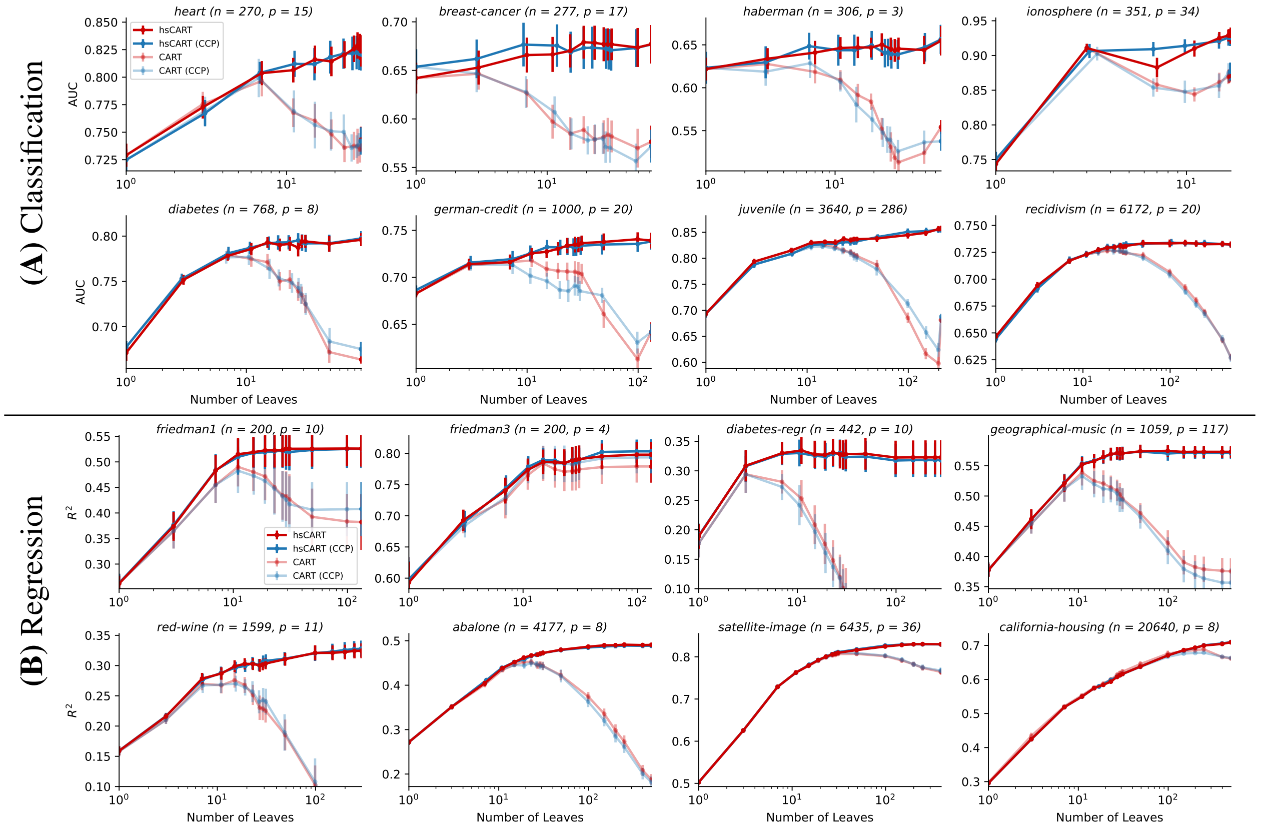

Figure S6. HS is able to improve **(A)** Classification and **(B)** regression predictive performance for both baselines: CART and CART with CCP for larger data-sets when using deeper trees by preventing overfitting. As shown by our connection to ridge regression in Sec 3, HS is beneficial when the ratio of number of leaves to samples is large. Error bars show standard error of the mean computer over 10 random data splits.

S4.5. Bayesian Additive Regression Trees (BART)

Background on BART: BART is a Bayesian tree ensemble model (Chipman et al., 2010), with the tree structure and leaf values are modeled as random variables. Conditioned on the tree structure, the leaf values, and a query point $\mathbf{x}$, the response variable y is assumed to follow a Gaussian distribution. The posterior distribution of the tree structures and leaf values is used for inference, with samples generated via a backfitting MCMC algorithm. Each posterior sample is a sum of trees function, and the final prediction is made using the posterior mean, i.e. an average prediction over many such samples.

BART has become popular in applied statistics research, especially in the field of causal inference (Hill, 2011). It has been claimed to obtain state-of-the-art prediction performance on many datasets, once its hyperparameters have been appropriately tuned (Chipman et al., 2006). As such, it offers both a baseline comparison for RFs regularized using HS (shown in Fig 4D), as well as an alternate tree ensemble growing method to which we can apply HS. Furtheremore, BART implicitly applies a form of shrinkage, as discussed in Sec 1.

In this section, we investigate applying HS to a tree ensemble grown using BART.

Experimental design: We used the implementation of BART in the `bartpy` package[12]. First, we fit a BART model with a prescribed number of trees by running a single MCMC chain for a standard number of burn-in iterations, and all other

[12]Code can be found at github.com/JakeColtman/bartpy.

hyperparameters set to defaults. We extract a tree ensemble from the BART model by drawing a single sample from the posterior distribution. Note that this gives us both the tree structures as well as the leaf values. We then apply HS to this model, tuning the regularization parameter via CV as before, to obtain the hsBART model.

Results: The results show that HS does not significantly alter prediction performance of BART on both regression and classification datasets. We believe that this can be explained by the tendency of BART to construct relatively small trees when the recommended parameters are used (Chipman et al., 2010). Furthermore, the raw BART predictions already come with leaf-based shrinkage, which is somewhat similar to HS for shallow trees.

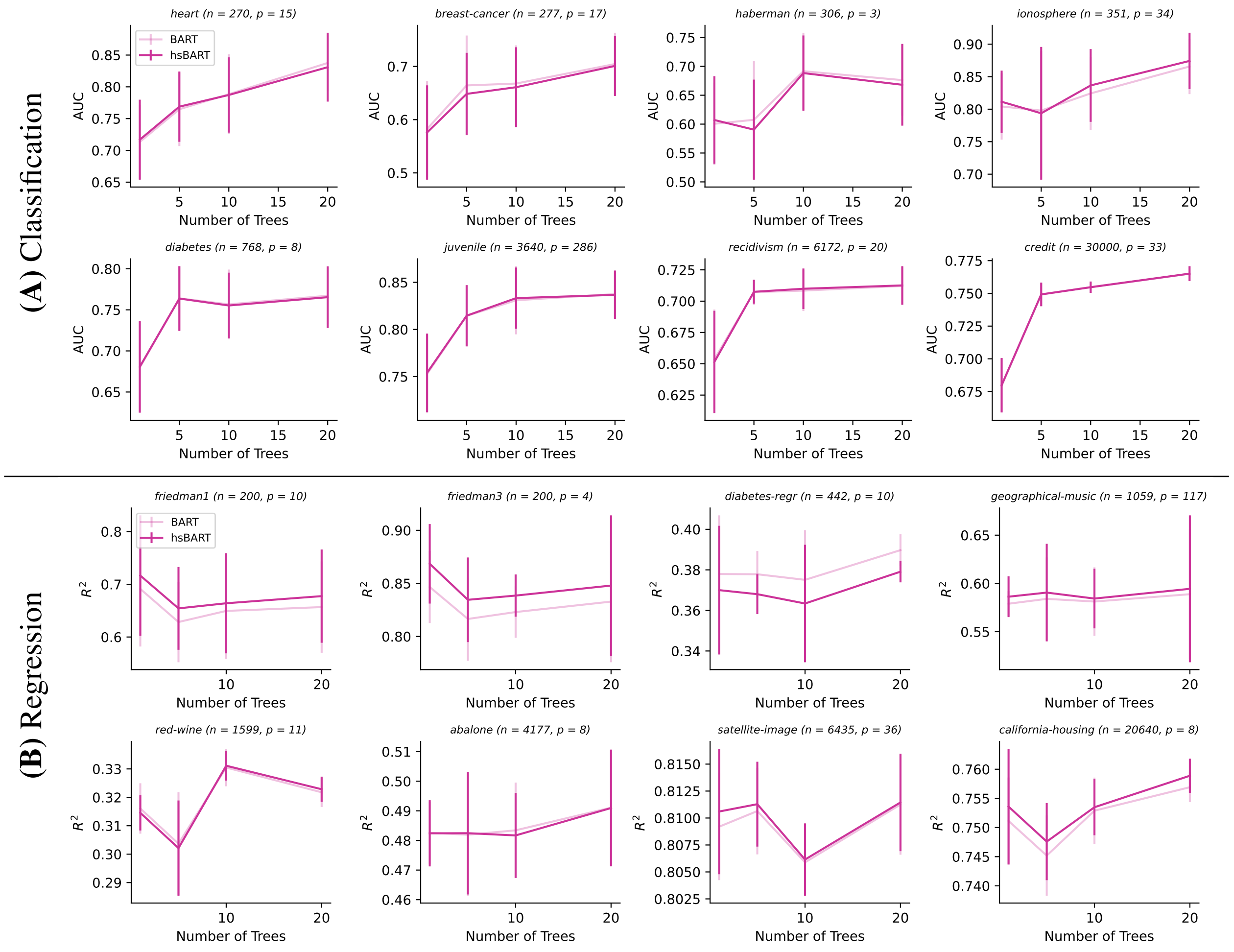

Figure S7. Hierarchical Shrinkage (solid lines) applied post-hoc to BART trees, shows similar performance for classification **(A)**, and regression **(B)** datasets in terms of AUC and R^2 respectively

S4.6. Gradient Boosted Trees

In this section, we investigate the effect of applying HS to gradient boosted trees (GBTs). We vary the number of trees B in the ensemble, constraining each tree to have maximum depth 3. All other experimental details are identical to those described in Sec 4.4.

The results, displayed in Fig S8, indicate that applying HS slightly improves the performance of GBTs for smaller datasets, while performing similarly for larger datasets in terms of AUC. This is not entirely unsurprising since the benefits of HS is most pronounced for smaller datasets and when growing deeper trees.

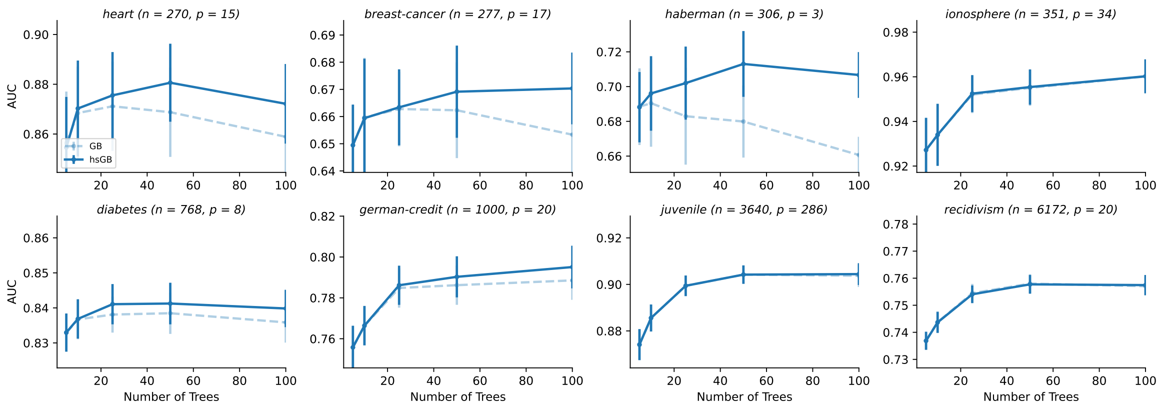

Figure S8. HS (solid lines) applied post-hoc to Gradient Boosted Trees. HS slightly improves performance for smaller datasets, and provides similar performance for larger datasets in terms of AUC.

S5. Further Experimental Results for Interpretability

S5.1. Decision Boundaries

In this section, we further investigate how HS can help to simplify decision boundaries. For each of our classification datasets, we fit an RF with 50 trees using only the two most important features, as measured using Mean Decrease in Impurity (MDI) feature importance. We also apply HS post hoc to the fitted RF model to obtain a second model. In Fig S9 and Fig S10, we plot the different decision boundaries of both models for each dataset.

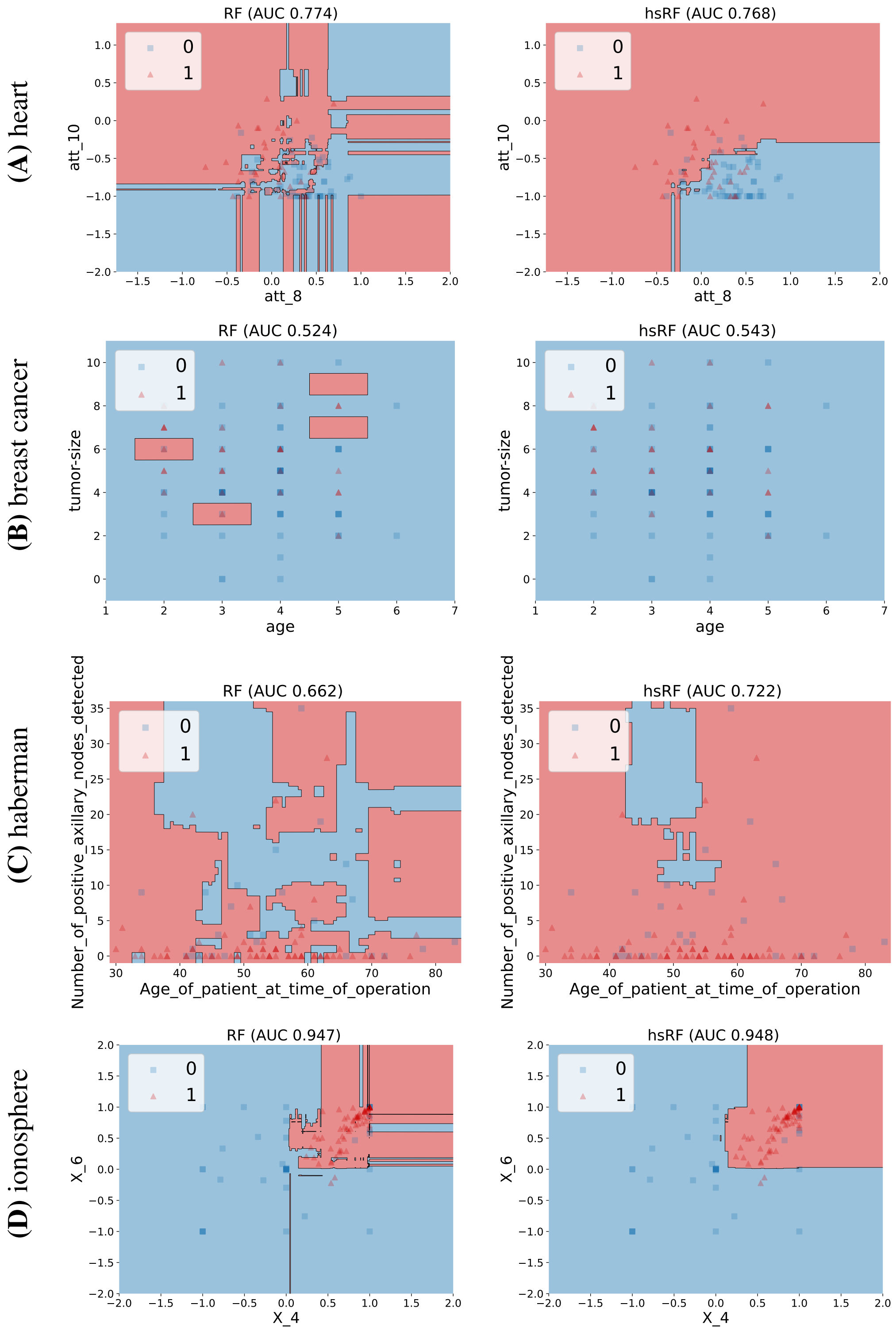

Figure S9. Comparison of decision boundary learnt by RF on the smaller datasets before / after applying HS. HS results in a smoother, simpler decision boundary, resulting in improved interpretability and simplicity.

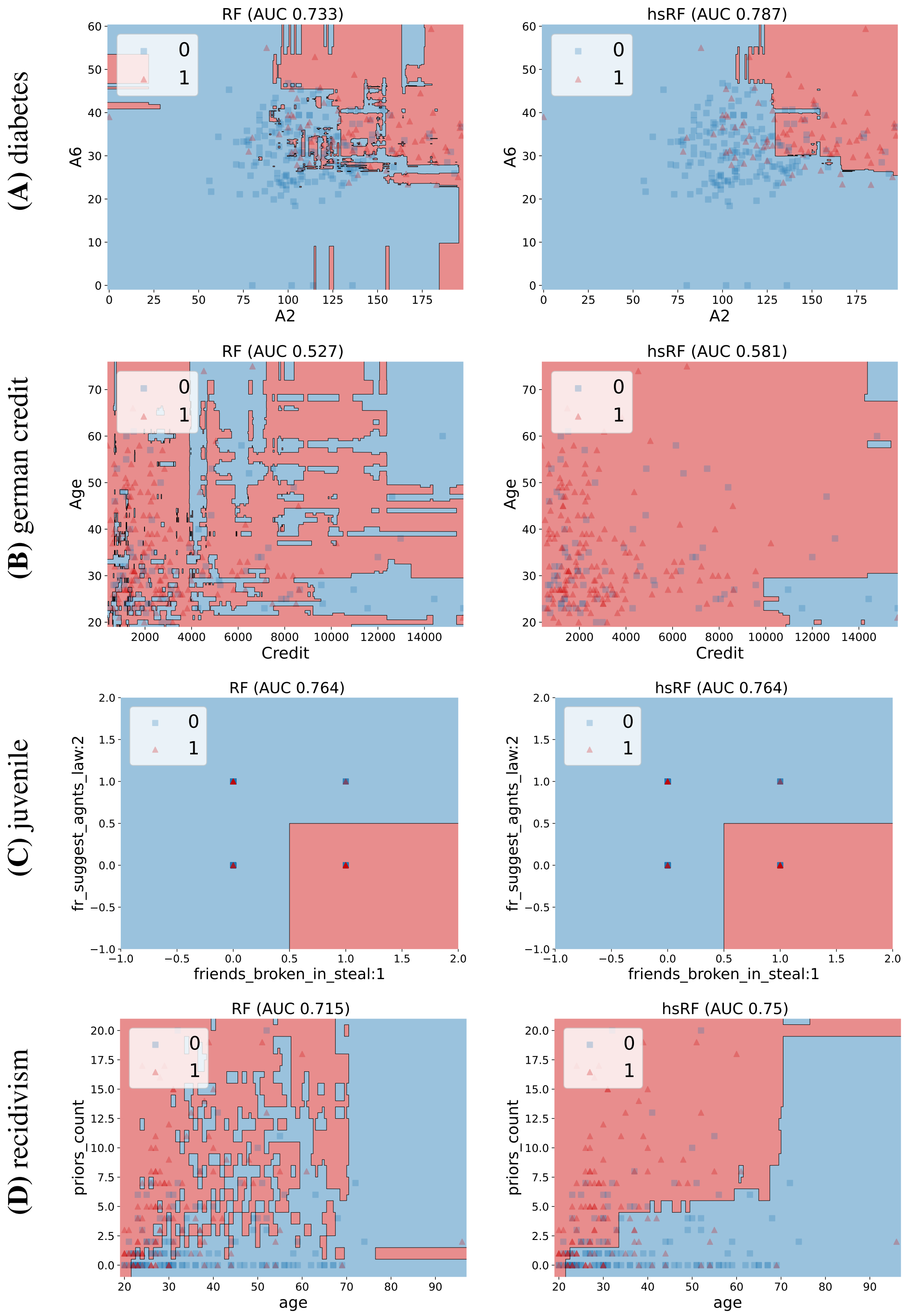

Figure S10. Comparison of decision boundary learnt by RF on the larger datasets before / after applying HS. HS results in a smoother, simpler decision boundary, resulting in improved interpretability and simplicity.

S5.2. SHAP Plots

In this section, we investigate SHAP plots (Lundberg & Lee, 2017) for RF with and without HS. SHAP plots provide a visualization of SHAP values, which are feature importance measures that are computed for each sample in the dataset. For each feature, a horizontal swarm plot of the SHAP values is provided, with the original feature value indicated using color.

Such plots have become a popular tool to explain black-box models such as RFs. We provide SHAP plots for RF with and without HS for every dataset in Table 1. After applying HS, the SHAP values for each feature have tighter clusters.

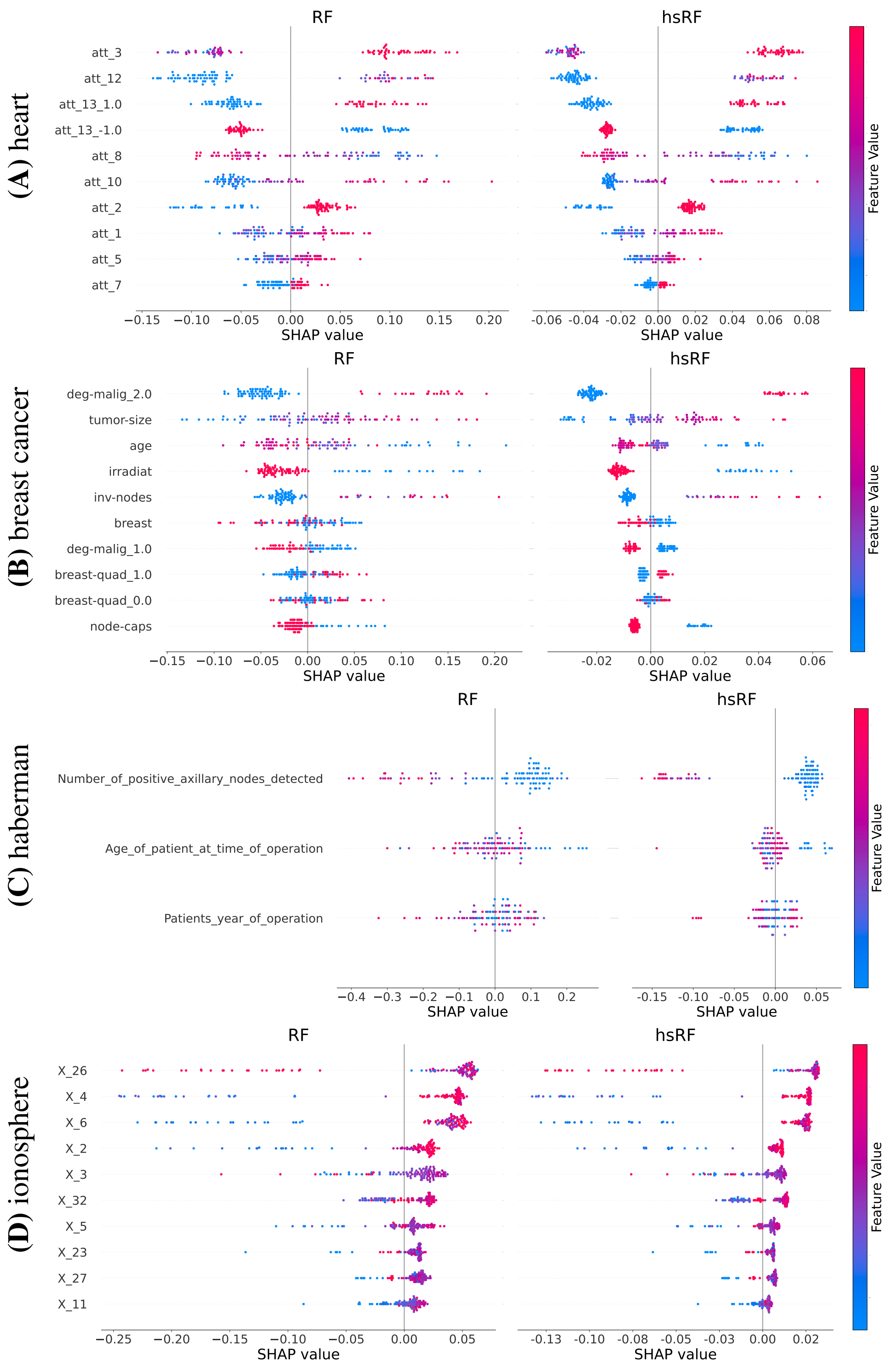

Figure S11. Comparison of SHAP plots learnt by Random Forests on the smaller datasets before / after applying HS. HS displays more clustered SHAP values for each feature, reflecting less heterogeneity in the effect of each feature on predicted response.

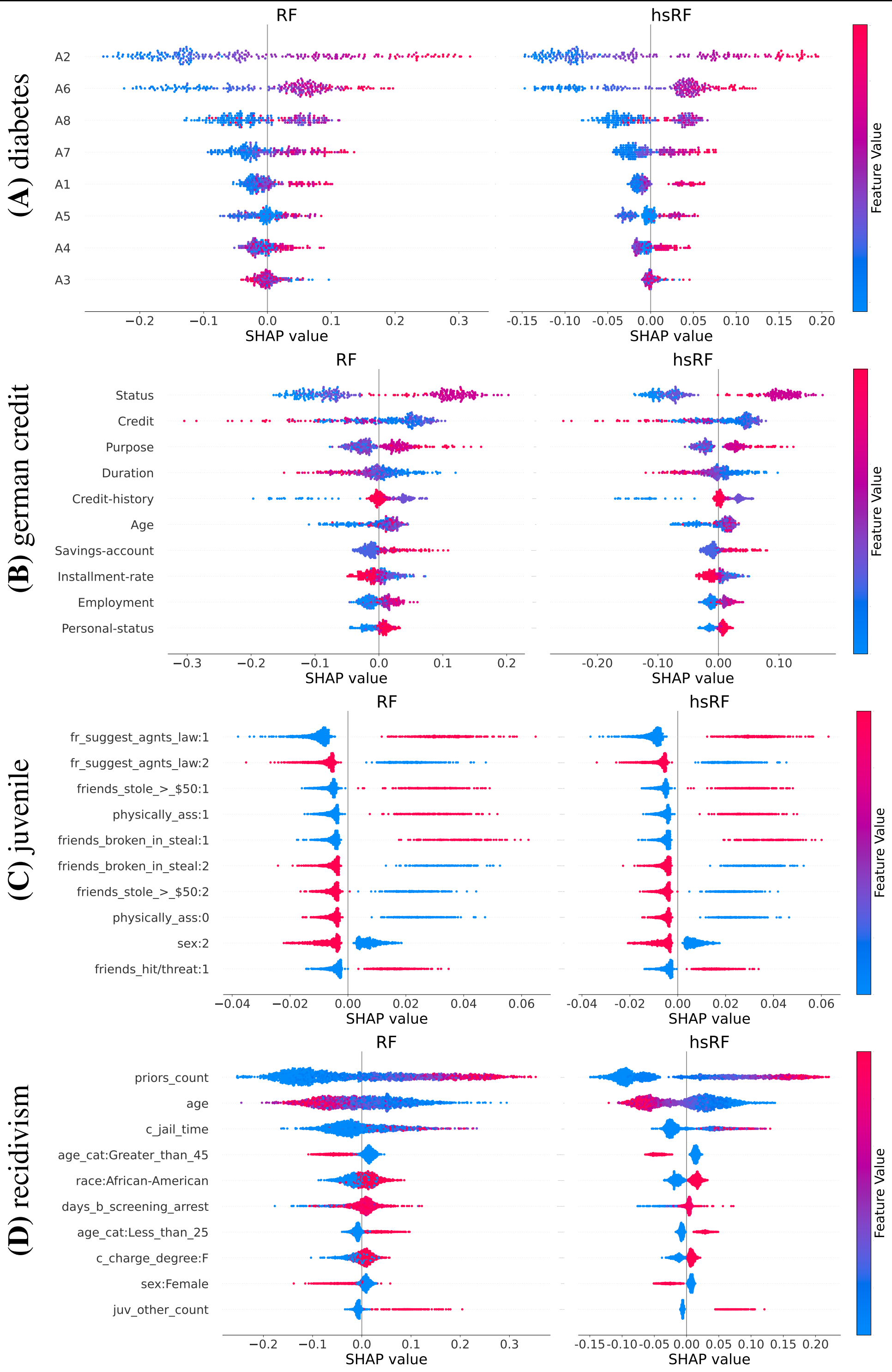

Figure S12. Comparison of SHAP plots learnt by Random Forests on the larger datasets before / after applying HS. HS displays more clustered SHAP values for each feature, reflecting less heterogeneity in the effect of each feature on predicted response.

S5.3. Stability of SHAP Values to Data Resampling

In this experiment, for each dataset, we randomly choose 50 samples to hold out, and for each of 100 iterations, we randomly select two thirds of the remaining samples and train an RF on the reduced dataset. For each held-out sample, we measure the variance of its SHAP values per feature across the 100 iterations. We then average the variance per feature across all 50 held-out samples, with these values plotted in Fig S13 and Fig S14 for RF with HS and without. We observe that the variances of the SHAP values for RF with HS are substantially smaller than those for RF without HS.

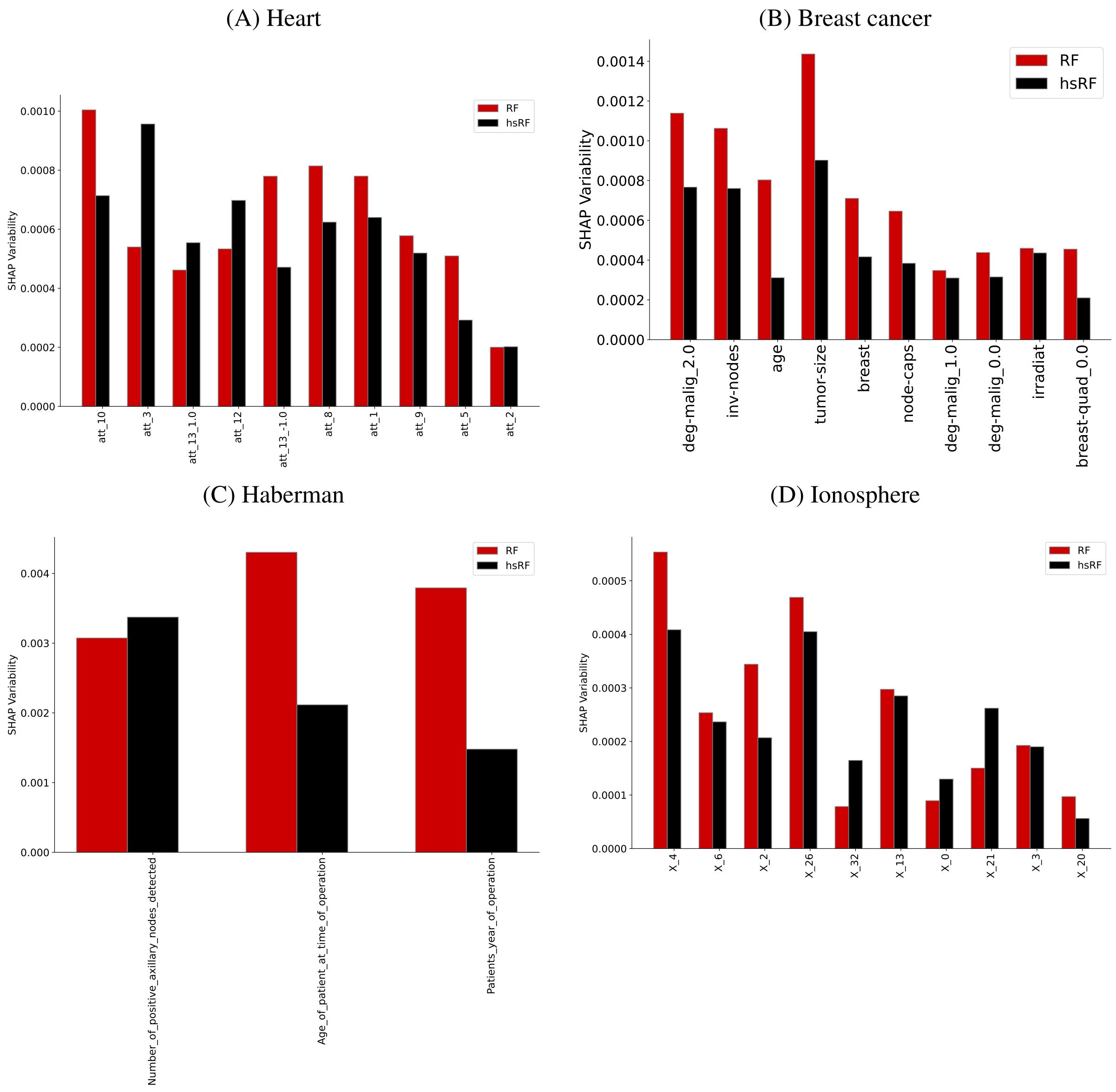

Figure S13. Comparison of SHAP plots learnt by Random Forests on different (small) datasets before / after applying HS. HS displays lower variability across different data perturbations, indicating enhanced stability.

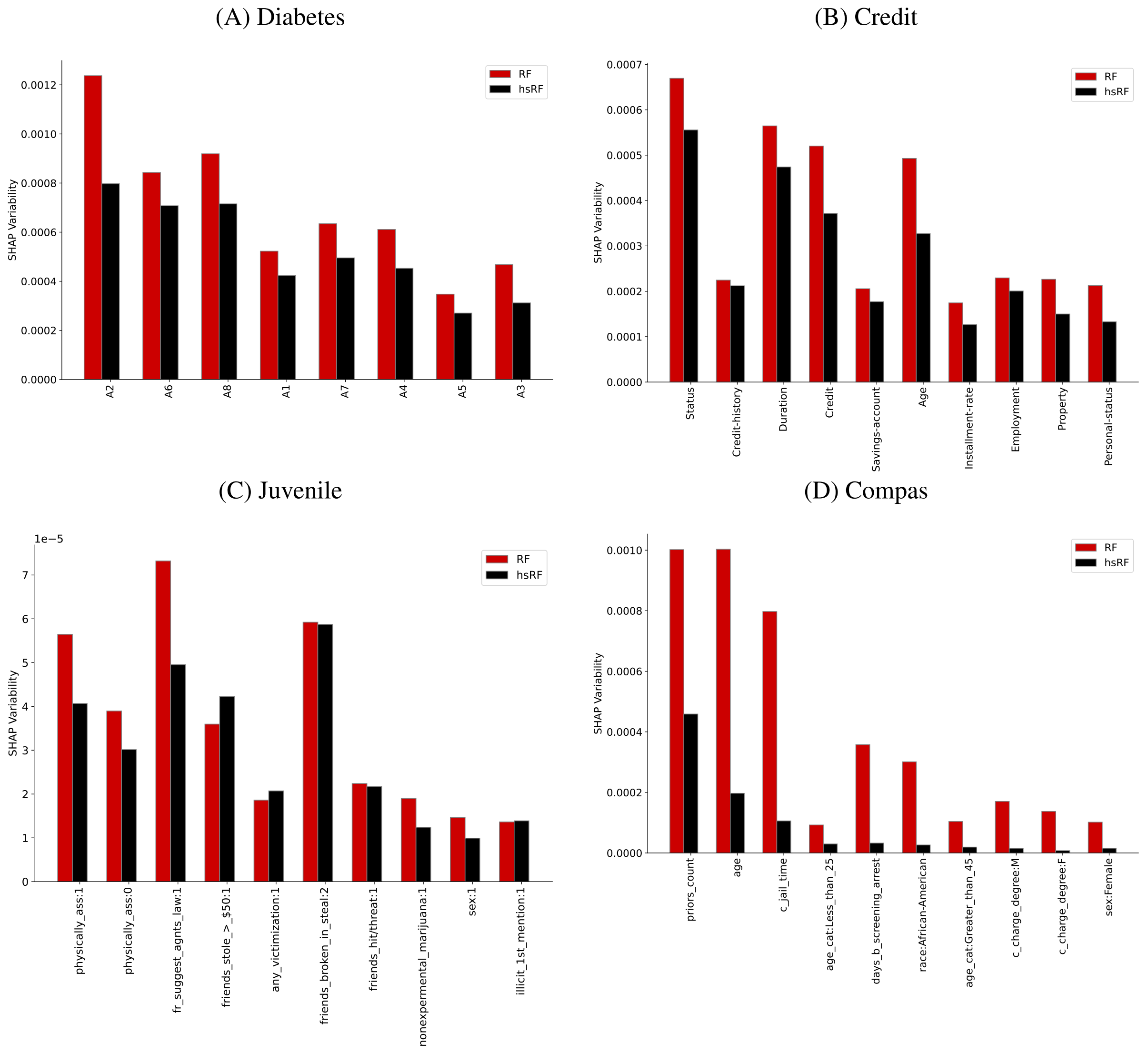

Figure S14. Comparison of SHAP plots learnt by Random Forests on different (large) datasets before / after applying HS. HS displays lower variability across different data perturbations, indicating enhanced stability.