Machine Learning manuscript No.
(will be inserted by the editor)

Optimal Survival Trees

Dimitris Bertsimas · Jack Dunn · Emma Gibson · Agni Orfanoudaki

Received: date / Accepted: date

Abstract Tree-based models are increasingly popular due to their ability to identify complex relationships that are beyond the scope of parametric models. Survival tree methods adapt these models to allow for the analysis of censored outcomes, which often appear in medical data. We present a new Optimal Survival Trees algorithm that leverages mixed-integer optimization (MIO) and local search techniques to generate globally optimized survival tree models. We demonstrate that the OST algorithm improves on the accuracy of existing survival tree methods, particularly in large datasets.

Keywords survival analysis, non-parametric models, recursive partitioning, censored data

1 Introduction

Survival analysis is a cornerstone of healthcare research and is widely used in the analysis of clinical trials as well as large-scale medical datasets such as Electronic Health Records and insurance claims. Survival analysis methods are required for censored data in which the outcome of interest is generally the time until an event (onset of disease, death, etc.), but the exact time of the event is unknown (censored) for some individuals. When a lower bound for these missing values is known (for example, a patient is known to be alive until at least time t) the data is said to be right-censored.

A common survival analysis technique is Cox proportional hazards regression (Cox, 1972) which models the hazard rate for an event as a linear combination of covariate effects. Although this model is widely used and easily interpreted, its parametric nature makes it unable to identify non-linear effects or interactions between covariates (Bou-Hamad et al., 2011).

Recursive partitioning techniques (also referred to as *trees*) are a popular alternative to parametric models. When applied to survival data, survival tree algorithms partition the covariate space into smaller and smaller regions (*nodes*) containing observations with homogeneous survival outcomes. The survival distribution in the final partitions (*leaves*) can be analyzed using a variety of statistical techniques such as Kaplan-Meier curve estimates (Kaplan and Meier, 1958).

Most recursive partitioning algorithms generate trees in a top-down, greedy manner, which means that each split is selected in isolation without considering its effect on subsequent splits in the tree. However, Bertsimas and Dunn (Bertsimas and Dunn, 2017, 2019) have proposed a new algorithm which uses modern

Dimitris Bertsimas
Operations Research Center, E40-111, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. E-mail: dbertim@mit.edu

Jack Dunn
Operations Research Center, E40-111, Massachusetts Institute of Technology Cambridge, MA 02139, USA. E-mail: jack@interpretable.ai

Emma Gibson
Operations Research Center, E40-111, Massachusetts Institute of Technology Cambridge, MA 02139, USA. E-mail: emgibson@mit.edu

Agni Orfanoudaki
Operations Research Center, E40-111, Massachusetts Institute of Technology Cambridge, MA 02139, USA. E-mail: agniorf@mit.edu

mixed-integer optimization (MIO) techniques to form the entire decision tree in a single step, allowing each split to be determined with full knowledge of all other splits. This *Optimal Trees* algorithm allows the construction of single decision trees for classification and regression that have performance comparable with state-of-the-art methods such as random forests and gradient boosted trees, without sacrificing the interpretability offered by a single-tree model.

The key contributions of this paper are:

1. We present *Optimal Survival Trees* (OST), a new survival trees algorithm that utilizes the *Optimal Trees* framework to generate interpretable trees for censored data.
2. We propose a new accuracy metric that evaluates the fit of Kaplan-Meier curve estimates relative to known survival distributions in simulated datasets. We also demonstrate that this metric is consistent with the Integrated Brier Score (Graf et al., 1999), which can be used to evaluate the fit of Kaplan-Meier curves when the true distributions are unknown.
3. We evaluate the performance of our method in both simulated and real-world datasets and demonstrate improved accuracy relative to two existing algorithms.
4. Finally, we provide an example of how the algorithm can be used to predict the risk of adverse events associated with cardiovascular health in the Framingham Heart Study (FHS) dataset.

The structure of this paper is as follows. We review existing survival tree algorithms in Section 2 and discuss some of the technical challenges associated with building trees for censored data. In Section 3, we give an overview of the Optimal Trees algorithm proposed by Bertsimas and Dunn and we adapt this algorithm for Optimal Survival Trees in Section 4. Section 5 begins with a discussion of existing survival tree accuracy metrics, followed by the new accuracy metrics that we have introduced to evaluate survival tree models in simulated datasets. Simulation results are presented in Section 6 and results for real-world datasets are presented in Sections 7–8. We conclude in Section 9 with a brief summary of our contributions.

2 Review of Survival Trees

Recursive partitioning methods have received a great deal of attention in the literature, the most prominent method being the Classification and Regression Tree algorithm (CART) (Breiman et al., 1984). Tree-based models are appealing due to their logical, interpretable structure as well as their ability to detect complex interactions between covariates. However, traditional tree algorithms require complete observations of the dependent variable in training data, making them unsuitable for censored data.

Tree algorithms incorporate a splitting rule which selects partitions to add to the tree, and a pruning rule determines when to stop adding further partitions. Since the 1980s, many authors have proposed splitting and pruning rules for censored data. Splitting rules in survival trees are generally based on either (a) node distance measures that seek to maximize the difference between observations in separate nodes or (b) node purity measures that seek to group similar observation in a single node (Zhou and McArdle, 2015; Molinaro et al., 2004).

Algorithms based on node distance measures compare the two adjacent child nodes that are generated when a parent node is split, retaining the split that produces the greatest difference in the child nodes. Proposed measures of node distance include the two-sample logrank test (Ciampi et al., 1986), the likelihood ratio statistic (Ciampi et al., 1987) and conditional inference permutation tests (Hothorn et al., 2006). We note that the score function used in Cox regression models also falls into the class of node distance measures, as the partial likelihood statistic is based on a comparison of the relative risk coefficient predicted for each observation.

Dissimilarity-based splitting rules are unsuitable for certain applications (such as the Optimal Trees algorithm) because they do not allow for the assessment of a single node in isolation. We will therefore focus on node purity splitting rules for developing the OST algorithm.

Gordon and Olshen (1985) published the first survival tree algorithm with a node purity splitting rule based on Kaplan-Meier estimates. Davis and Anderson (1989) used a splitting rule based on the negative log-likelihood of an exponential model, while Therneau et al. (1990) proposed using martingale residuals as an estimate of node error. LeBlanc and Crowley (1992) suggested comparing the log-likelihood of a saturated model to the first step of a full likelihood estimation procedure for the proportional hazards model and showed that both the full likelihood and martingale residuals can be calculated efficiently from the Nelson-Aalen cumulative hazard estimator (Nelson, 1972; Aalen, 1978). More recently, Molinaro et al. (2004) proposed a new approach to adjust loss functions for uncensored data based on inverse probability of censoring weights (IPCW).

Most survival tree algorithms make use of cost-complexity pruning to determine the correct tree size, particularly when node purity splitting is used. Cost-complexity pruning selects a tree that minimizes a weighted combination of the total tree error (i.e., the sum of each leaf node error) and tree complexity (the number of leaf nodes), with relative weights determined by cross-validation. A similar split-complexity pruning method was suggested by LeBlanc and Crowley (1993) for node distance measures, using the sum of the split test statistics and the number of splits in the tree. Other proposals include using the Akaike Information Criterion (AIC) (Ciampi et al., 1986) or using a p-value stopping criterion to stop growing the tree when no further significant splits are found (Hothorn et al., 2006).

Survival tree methods have been extended to include "survival forest" algorithms which aggregate the results of multiple trees. Breiman (2002) adapted the CART-based random forest algorithm to survival data, while both Hothorn et al. (2004) and Ishwaran et al. (2008) proposed more general methods that generate survival forests from any survival tree algorithm. The aim of survival forest models is to produce more accurate predictions by avoiding the instability of single-tree models. However, this approach leads to "black-box" models which are not interpretable and therefore lack one of the primary advantages of single-tree models.

Relatively few survival tree algorithms have been implemented in publicly available, well-documented software. Two user-friendly options are available in **R** (R Core Team, 2017) packages: Therneau's algorithm based on martingale residuals is implemented in the **rpart** package (Therneau et al., 2010) and Hothorn's conditional inference (**ctree**) algorithm in the **party** package (Hothorn et al., 2010).

3 Review of Optimal Predictive Trees

In this section, we briefly review approaches to constructing decision trees, and in particular, we outline the Optimal Trees algorithm. The purpose of this section is to provide a high-level overview of the Optimal Trees framework; interested readers are encouraged to refer to Bertsimas and Dunn (2019) and Dunn (2018) for more detailed technical information. The Optimal Trees algorithm and is implemented in `Julia` (Bezanson et al., 2017) and is available to academic researchers under a free academic license.[1]

Traditionally, decision trees are trained using a greedy heuristic that recursively partitions the feature space using a sequence of locally-optimal splits to construct a tree. This approach is used by methods like CART (Breiman et al., 1984) to find classification and regression trees. The greediness of this approach is also its main drawback—each split in the tree is determined independently without considering the possible impact of future splits in the tree on the quality of the here-and-now decision. This can create difficulties in learning the true underlying patterns in the data and lead to trees that generalize poorly. The most natural way to address this limitation is to consider forming the decision tree in a single step, where each split in the tree is decided with full knowledge of all other splits.

Optimal Trees is a novel approach for decision tree construction that significantly outperforms existing decision tree methods (Bertsimas and Dunn, 2019). It formulates the decision tree construction problem from the perspective of global optimality using mixed-integer optimization (MIO), and solves this problem with coordinate descent to find optimal or near-optimal solutions in practical run times. These Optimal Trees are often as powerful as state-of-the-art methods like random forests or boosted trees, yet they are just a single decision tree and hence are readily interpretable. This obviates the need to trade off between interpretability and state-of-the-art accuracy when choosing a predictive method.

The Optimal Trees framework is a generic approach that tractably and efficiently trains decision trees according to a loss function of the form

$$\min_{T} \quad \texttt{error}(T, D) + \alpha \cdot \texttt{complexity}(T), \tag{1}$$

where T is the decision tree being optimized, D is the training data, $\texttt{error}(T, D)$ is a function measuring how well the tree T fits the training data D, $\texttt{complexity}(T)$ is a function penalizing the complexity of the tree (for a tree with splits parallel to the axis, this is simply the number of splits in the tree), and α is the *complexity parameter* that controls the tradeoff between the quality of the fit and the size of the tree.

There have been many attempts in the literature to construct globally optimal predictive trees (Bennett and Blue, 1996; Son, 1998; Grubinger et al., 2014). However, these methods could not scale to datasets of the sizes required by practical applications, and therefore did not displace greedy heuristics as the approach used in practice. Unlike the others, Optimal Trees is able scale to large datasets (n in the millions, p in the thousands) by using coordinate descent to train the decision trees towards global optimality. When

[1] Please email survival-trees@mit.edu to request an academic license for the Optimal Survival Trees package.

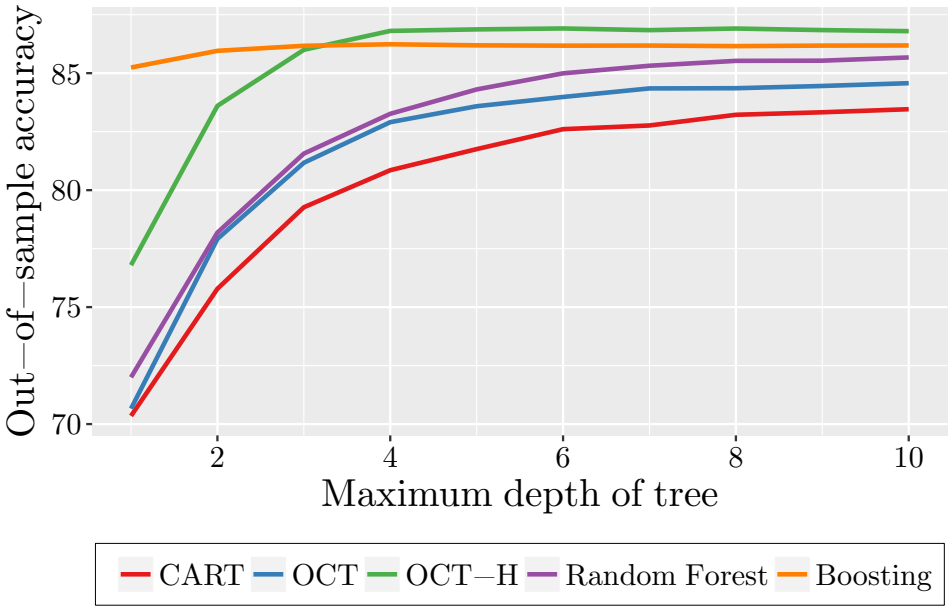

Fig. 1: Performance of classification methods averaged across 60 real-world datasets. OCT and OCT-H refer to Optimal Classification Trees without and with hyperplane splits, respectively.

training a tree, the splits in the tree are repeatedly optimized one-at-a-time, finding changes that improve the global objective value in Problem (1). To give a high-level overview, the nodes of the tree are visited in a random order and at each node we consider the following modifications:

- If the node is not a leaf, delete the split at that node;
- If the node is not a leaf, find the optimal split to use at that node and update the current split;
- If the node is a leaf, create a new split at that node.

For each of the changes, we calculate the objective value of the modified tree with respect to Problem (1). If any of these changes result in an improved objective value, then the modification is accepted. When a modification is accepted or all potential modifications have been dismissed, the algorithm proceeds to visit the nodes of the tree in a random order until no further improvements are found, meaning that this tree is a locally optimal for Problem (1). The problem is non-convex, so we repeat the coordinate descent process from various randomly-generated starting decision trees, before selecting the final locally-optimal tree with the lowest overall objective value as the best solution. For a more comprehensive guide to the coordinate descent process, we refer the reader to Bertsimas and Dunn (2019).

Although only one tree model is ultimately selected, information from multiple trees generated during the training process is also used to improve the performance of the algorithm. For example, the Optimal Trees algorithm combines the result of multiple trees to automatically calibrate the complexity parameter (α) and to calculate variable importance scores in the same way as random forests or boosted trees. More detailed explanations of these procedures can be found in Dunn (2018).

The coordinate descent approach used by Optimal Trees is generic and can be applied to optimize a decision tree under any objective function. For example, the Optimal Trees framework can train Optimal Classification Trees (OCT) by setting $\texttt{error}(T, D)$ to be the misclassification error associated with the tree predictions made on the training data. We provide a comparison of performance between various classification methods from Bertsimas and Dunn (2019) in Figure 1. This comparison shows the performance of two versions of Optimal Classification Trees: OCT with parallel splits (using one variable in each split); and OCT with hyperplane splits (using a linear combination of variables in each split). These results demonstrate that not only do the Optimal Tree methods significantly outperform CART in producing a single predictive tree, but also that these trees have performance comparable with some of the best classification methods.

In Section 4, we will extend the Optimal Trees framework to work with censored data and generate Optimal Survival Trees.

4 Survival tree algorithm

In this section, we adapt the Optimal Trees algorithm described in Section 3 for the analysis of censored data. For simplicity, we will use terminology from survival analysis and assume that the outcome of interest is the time until death. We begin with a set of observations $(t_i, \delta_i)_{i=1}^n$ where t_i indicates the time of last observation and δ_i indicates whether the observation was a death ($\delta_i = 1$) or a censoring ($\delta_i = 0$).

Like other tree algorithms, the OST model requires a target function that determines which splits should be added to the tree. Computational efficiency is an important factor in the choice of target function, since it must be re-evaluated for every potential change to the tree during the optimization procedures. A key requirement for the target function is that the "fit" or error of each node should be evaluated independently of the rest of the tree. In this case, changing a particular split in the tree will only require re-evaluation of the subtree directly below that split, rather than the entire tree. This requirement restricts the choice of target function to the node purity approaches described in Section 2.

The splitting rule implemented in the OST algorithm is based on the likelihood method proposed by LeBlanc and Crowley (1992). This splitting rule is derived from a proportional hazards model which assumes that the underlying survival distribution for each observation is given by

$$\mathrm{P}(S_i \leq t) = 1 - e^{-\theta_i \Lambda(t)}, \tag{2}$$

where $\Lambda(t)$ is the baseline cumulative hazard function and the coefficients θ_i are the adjustments to the baseline cumulative hazard for each observation.

In a survival tree model we replace $\Lambda(t)$ with an empirical estimate for the cumulative probability of death at each of the observation times. This is known as the Nelson-Aalen estimator (Nelson, 1972; Aalen, 1978),

$$\hat{\Lambda}(t) = \sum_{i:t_i \leq t} \frac{\delta_i}{\sum_{j:t_j \geq t_i} 1}. \tag{3}$$

Assuming this baseline hazard, the objective of the survival tree model is to optimize the hazard coefficients θ_i. We impose that the tree model uses the same coefficient for all observations contained in a given leaf node in the tree, i.e. $\theta_i = \hat{\theta}_{T(i)}$. These coefficients are determined by maximizing the within-leaf sample likelihood

$$L = \prod_{i=1}^{n} \left(\theta_i \frac{d}{dt} \Lambda(t_i) \right)^{\delta_i} e^{-\theta_i \Lambda(t_i)}, \tag{4}$$

to obtain the node coefficients

$$\hat{\theta}_k = \frac{\sum_i \delta_i I_{\{T_i = k\}}}{\sum_i \hat{\Lambda}(t_i) I_{\{T_i = k\}}}. \tag{5}$$

To evaluate how well different splits fit the available data we compare the current tree model to a tree with a single coefficient for each observation. We will refer to this as a fully saturated tree, since it has a unique parameter for every observation. The maximum likelihood estimates for these saturated model coefficients are

$$\hat{\theta}_i^{sat} = \frac{\delta_i}{\hat{\Lambda}(t_i)}, \quad i = 1, \ldots, n. \tag{6}$$

We calculate the prediction error at each node as the difference between the log-likelihood for the fitted node coefficient and the saturated model coefficients at that node:

$$\texttt{error}_k = \sum_{i:T(i)=k} \left(\delta_i \log \left(\frac{\delta_i}{\hat{\Lambda}(t_i)} \right) - \delta_i \log(\hat{\theta}_k) - \delta_i + \hat{\Lambda}(t_i) \hat{\theta}_k \right). \tag{7}$$

The overall error function used to optimize the tree is simply the sum of the errors across the leaf nodes of the tree T given the training data D:

$$\texttt{error}(T, D) = \sum_{k \in \text{leaves}(T)} \texttt{error}_k(D). \tag{8}$$

We can then apply the Optimal Trees approach to train a tree according to this error function by substituting this expression into the overall loss function (1). At each step of the coordinate descent process, we determine new estimates for $\hat{\theta}_k$ for each leaf node k in the tree using (B.2). We then calculate and sum the errors at each node using (7) to obtain the total error of the current solution, which is used to guide the coordinate descent and generate trees that minimize the error (8).

5 Survival tree accuracy metrics

In order to assess the performance of the OST algorithm, we now introduce a number of accuracy metrics for survival tree models. We will use the notation T to represent a tree model, where $T_i = T(X_i)$ is the leaf node classification of observation i with covariates X_i in the tree T. We will use the notation T^0 to represent a null model (a tree with no splits and a single node).

5.1 Review of survival model metrics

We begin by reviewing existing accuracy metrics for survival models that are commonly used in both the literature as well as practical applications.

1. **Cox Partial Likelihood Score**
   The Cox proportional hazards model (Cox, 1972) is a semi-parametric model that is widely used in survival analysis. The Cox hazard function estimate is

$$\lambda(t|X_i) = \lambda_0(t) \exp(\beta_1 X_{i1} + \cdots + \beta_p X_{ip}) = \lambda_0(t) \exp(\beta^T X_i), \tag{9}$$

   where $\lambda_0(t)$ is the baseline hazard function and β is a vector of fitted coefficients. This proportional hazards model does not make any assumptions about the form of $\lambda_0(t)$, and its parameters can be estimated even when the baseline is completely unknown (Cox, 1975). The coefficients β are estimated by maximizing the partial likelihood function[2],

$$L(\beta) = \prod_{\text{t}_\text{i}\text{uncensored}} \frac{\exp(X_i\beta)}{\sum_{t_j \geq t_i} \exp(X_j\beta)} = \prod_{\text{t}_\text{i}\text{uncensored}} \frac{\theta_i}{\sum_{t_j \geq t_i} \theta_j}. \tag{10}$$

   For computational convenience, the Cox model is generally implemented using the log partial likelihood,

$$l(\beta) = \log L(\beta) = \sum_{\text{t}_\text{i}\text{uncensored}} X_i\beta - \log(\sum_{t_j \geq t_i} \exp(X_j\beta)). \tag{11}$$

   In the context of survival trees, we can find the Cox hazard function associated with a particular tree model by assigning one coefficient to each leaf node in the tree, i.e.,

$$\lambda_T(t) = \lambda_0(t) \exp(\sum_{k \in T} \beta_k \mathbb{1}(T_i = k)) = \lambda_0(t) \exp(\beta_{T_i}). \tag{12}$$

   We define the Cox Score for a tree model as the maximized log partial likelihood for the associated Cox model, $\max_\beta l(\beta|T)$. To assist with interpretation, we also define the Cox Score Ratio (CSR) as the percentage reduction in the Cox Score for tree T relative to a null model,

$$CSR(T) = 1 - \frac{\max_\beta l(\beta|T)}{\max_\beta l(\beta|T^0)}. \tag{13}$$

2. **The Concordance Statistic**
   Applying a ranking approach to survival analysis is an effective way to deal with the skewed distributions of survival times as well as censored of the data. The Concordance Statistic, which is most familiar from logistic regression, is another popular metric that has been adapted to measure goodness-of-fit in survival models (Harrell et al., 1982). The concordance index is defined as the proportion of all *comparable* pairs of observations in which the model's predictions are *concordant* with the observed outcomes.
   Two observations are *comparable* if it is know with certainty that one individual died before the other. This occurs when the actual time of death is observed for both individuals (neither is censored) or when the one individual's death is observed before the other is censored. A comparable pair is *concordant* if the predicted risk is higher for the individual that died first, and the pair is discordant if the predicted risk is lower for the individual that died first. Thus, the number of concordant pairs in a sample is given by

$$CC = \sum_{i,j} \mathbb{1}(t_i > t_j)\mathbb{1}(\theta_i < \theta_j)\delta_j, \tag{14}$$

[2] This definition of the partial likelihood assumes that there are no ties in the data set (i.e., no two subjects have the same event time).

and the number of discordant pairs is

$$DC = \sum_{i,j} \mathbb{1}(t_i > t_j)\mathbb{1}(\theta_i > \theta_j)\delta_j, \tag{15}$$

where the indices i and j refer to pairs of observations in the sample. Multiplication by the factor δ_j discards pairs of observations that are not comparable because the smaller survival time is censored, i.e., $\delta_j = 0$. These definitions do not include comparable pairs with tied risk predictions, so we denote these pairs as

$$TR = \sum_{i,j} \mathbb{1}(t_i > t_j)\mathbb{1}(\theta_i = \theta_j)\delta_j. \tag{16}$$

The number of concordant and discordant pairs is commonly summarized using Harrell's C-index (Harrell et al., 1982),

$$H_C = \frac{CC + 0.5 \times TR}{CC + DC + TR}. \tag{17}$$

Harrell's C takes values between 0 and 1, with higher values indicating a better fit. Note that randomly assigned predictions have an expected score of $H_C = 0.5$.
More recently, Uno et al. (2011) introduced a non-parametric C-Statistic,

$$U_{C_\tau} = \frac{\sum_{i,j}(\hat{G}(t_j))^{-2}\mathbb{1}(t_i > t_j, t_j < \tau)\mathbb{1}(\theta_i < \theta_j)\delta_j}{\sum_{i,j}\hat{(}(G)(t_j))^{-2}\mathbb{1}(t_i > t_j, t_j < \tau)(\theta_i > \theta_j)(\delta_j) + (t_i > t_j, t_j < \tau)(\theta_i < \theta_j)(\delta_j)}, \tag{18}$$

where $\hat{G}(\cdot)$ is the Kaplan-Meier estimate for the censoring distribution. Due to these coefficients, U_C converges to a quantity that is independent of the censoring distribution. U_C takes values between 0 and 1, with higher values indicating a better fit.
It is important to note that the metrics described above are not specifically designed for survival trees, and therefore have certain limitations when applied in this context. The Cox partial likelihood score and the C-statistics become less informative when a large number of observations have the same predicted risk coefficient, which is generally the case in tree models. Increasing the number of nodes in the tree may inflate these scores even if the overall quality of the model does not improve.

3. **Integrated Brier score**
The Brier score metric is commonly used to evaluate classification trees (Brier, 1950). It was originally developed to verify the accuracy of a probability forecast, primarily purposed for weather forecasting. The most common formula calculates the mean squared prediction error:

$$B = \frac{1}{n}\sum_{i}^{n}(\hat{p}(y_i) - y_i)^2, \tag{19}$$

where n is the sample size, $y_i \in \{0, 1\}$ is the outcome of observation i, and $\hat{p}(y_i)$ is the forecast probability of this observed outcome. In the context of survival analysis, the Brier score may be used to evaluate the accuracy of survival predictions at a particular point in time relative to the observed deaths at that time. We will refer to this as the Brier Point Score:

$$BP_\tau = \frac{1}{|\mathcal{I}_\tau|}\sum_{i \in \mathcal{I}_\tau}(\hat{S}_i(\tau) - \mathbb{1}(t_i > \tau))^2, \tag{20}$$

$$\text{where } \mathcal{I}_\tau = \{i \in \{1, \ldots, n\}, |t_i \geq \tau \text{ or } \delta_i = 1\}. \tag{21}$$

In this case, $\hat{S}_i(\tau)$ is the predicted survival probability for observation i at time τ and $\mathcal{I}_\tau$ is the set of observations that are known to be alive/dead at time τ. Observations censored before time τ are excluded from this score, as their survival status is unknown.
Applying this version of the Brier score may be useful in applications where the main outcome of interest is survival at a particular time, such as the 1-year survival rates after the onset of a disease. In the experiments that follow, the point-wise Brier Score will be evaluated at the median observation time in each dataset. For easy interpretation, the reported scores are normalized relative to the score for a null model, i.e.

$$BPR_\tau = 1 - \frac{BP_\tau(T)}{BP_\tau(T^0)}. \tag{22}$$

The Brier Point score has two significant disadvantages in survival analysis. First, it assess predictive accuracy of survival models a single point in time rather than over the entire observation period, which is not well-suited to applications where survival distributions are the outcome of interest. Second, it becomes less informative as the number of censored observations increases, because a greater number of observations are discarded when calculating the score.

Graf et al. (1999) have addressed these challenges by proposing an adjusted version of the Brier Score for survival datasets with censored outcomes. Rather than measuring the accuracy of survival predictions at a single point, this measure aggregates the Brier score over the entire time interval observed in the data. This modified measure is commonly used in the survival literature and has been interchangeably called the Brier Score or the Integrated Brier Score by various authors (Reddy and Kronek, 2008). In this paper, we will refer to the metric specific to survival analysis as the Integrated Brier score (IB), defined as

$$IB = \frac{1}{n}\frac{1}{t_{max}}\sum_{i=1}^{n}\int_0^{t_i}\frac{(1-\hat{S}_i(t))^2}{\hat{G}_i(t)}dt + \delta_i\int_{t_i}^{t_{max}}\frac{(\hat{S}_i(t))^2}{\hat{G}_i(t_i)}dt. \tag{23}$$

The IB score uses Kaplan-Meier estimates for both the survival distribution, $\hat{S}(t)$, and the censoring distribution, $\hat{G}(t)$. In a survival tree model, these estimates are obtained by pooling observations in each node in the tree, i.e., $\hat{S}_i(t) = \hat{S}_{T(i)}(t)$. The IB score is a weighted version of the original Brier Score, with the weights being $1/\hat{G}_i(t_i)$ if an event occurs before time t_i, and $1/\hat{G}_i(t)$ if the event occurs after time t.

We report the Integrated Brier score ratio (IBR), which compares the sum of the Integrated Brier scores in a given tree to the corresponding Integrated Brier scores in a null tree[3]:

$$IBR = 1 - \frac{IB(T)}{IB(T^0)}. \tag{24}$$

We note that all of the above metrics have some limitations and do not provide definitive evidence that one model is better than another. In practice, these metrics often provide contradictory assessments when comparing different tree models. For example, our empirical experiments comparing three candidate models were only able to identify a non-dominated model for about 30% of the instances. In the other 70% of our test cases, none of the three candidate models scored at least as high as the other models on all metrics.

These limitations make it difficult to obtain an unambiguous comparison between the performance of different survival tree algorithms. To address this challenge, we will now introduce a simulation procedure and associated accuracy metrics that are specifically designed to assess survival tree models.

5.2 Simulation accuracy metrics

A key difficulty in selecting performance metrics for survival tree models is that the definition of "accuracy" can depend on the context in which the model will be used. For example, consider a survival tree that models the relationship between lifestyle factors and age of death. A medical researcher may use such a model to *identify risk factors* associated with early death, while an insurance firm may use this model to *predict mortality risks* for individual clients in order to estimate the volume of life insurance policy pay-outs in the coming years. The medical researcher is primarily interested whether the model has identified important splits, while the insurer is more focused on whether the model can accurately estimate survival distributions.

In subsequent sections we refer to these two properties as *classification accuracy* and *prediction accuracy*. We develop metrics to measure these outcomes in simulated datasets with the following structure:

Let $i = 1, \ldots, n$ be a set of observations with independent, identically distributed covariates $\mathbf{X}_i = (X_{ij})_{j=1}^m$. Let C be a tree model that partitions observations based on these covariates such that $C_i = C(\mathbf{X}_i)$ is the index of the leaf node in C that contains individual i. Let S_i be a random variable representing the survival time of observation i, with distribution $S_i \sim F_{C_i}(t)$. The survival distribution of each individual is entirely determined by its location in the tree C, and so we refer to C as the "true" tree model.

This underlying tree structure provides an unambiguous target against which we can measure the performance of empirical survival tree models. In this context, an empirical survival tree model T has high accuracy if it achieves the following objectives:

1. Classification accuracy: the model recovers structure of the true tree (i.e., $T(\mathbf{X}_i) = C(\mathbf{X}_i)$).

[3] Radespiel-Tröger et al. (2003) calls this *explained residual variation*

2. Prediction accuracy: the model recovers the corresponding survival distributions of the true tree (i.e., $\hat{F}_{T_i}(t) = F_{C_i}(t)$).

It is important to recognize that these two objectives are not necessarily consistent, particularly in small samples. Trees with perfect classification accuracy may have a small number of observations in each leaf node, leading to noisy survival estimates with low prediction accuracy.

5.3 Classification accuracy metrics

We measure the classification accuracy of an empirical tree model (T) relative to the true tree (C) using the following metrics:

1. **Node homogeneity** The node homogeneity statistic measures the proportion of the observations in each node $k \in T$ that have the same true class in C. Let $p_{k,l}$ be the proportion of observations in node $k \in T$ that came from class $\ell \in C$ and let $n_{k,l}$ be the total number of observations at node $k \in T$ from class $\ell \in C$. Then,

$$NH = \frac{1}{n} \sum_{k \in T} \sum_{l \in C} n_{k,l} p_{k,l}. \tag{25}$$

   A score of $NH = 1$ indicates that each node in the new tree model contains observations from a single class in C. This does not necessarily mean that the structure of T is identical to C — For example, a saturated tree with a single observation in each node would have a perfect node homogeneity score (see Figure 2). The node homogeneity metric is therefore biased towards larger tree models with few observations in each node.
2. **Class recovery**
   Class recovery is a measure of how well a new tree model is able to keep similar observations together in the same node, thereby avoiding unnecessary splits. Class recovery is calculated by counting the proportion of observations from a true class $\ell \in C$ that are placed in the same node in T. Let $q_{k,l}$ be the proportion of observations from class $\ell \in C$ that are classified in node $k \in T$ and let $n_{k,l}$ be the total number of observations at node $k \in T$ from class $\ell \in C$. Then,

$$CR = \frac{1}{n} \sum_{\ell \in C} \sum_{k \in T} n_{k,l} q_{k,l}. \tag{26}$$

   This metric is biased towards smaller trees, since a null tree with a single node would have a perfect class recovery score. It is therefore useful to consider both the class recovery and node homogeneity scores simultaneously in order to assess the performance of a tree model (see Figure 2 for examples). When used together, these metrics indicate how well the model T reflects the structure of the true model C.

The node homogeneity and class recovery scores can also be used to compare any two tree models, T_1 and T_2. In this case, these metrics should be interpreted as a measure of structural similarity between the two tree models. Note that when T_1 and T_2 are applied to the same dataset, the node homogeneity for model T_1 relative to T_2 is equivalent to the class recovery for T_2 relative to T_1, and vice versa. The average node homogeneity score for T_1 and T_2 is therefore equal to the average class recovery score for T_1 and T_2. We will refer to this as the *similarity score* for models T_1 and T_2.

5.4 Prediction accuracy metric

Our prediction accuracy metric measures how well the non-parametric Kaplan-Meier curves at each leaf in T estimate true the survival distribution of each observation.

1. **Area between curves (ABC)**
   For an observation i with true survival distribution $F_{C_i}(t)$, suppose that $\hat{S}_{T_i}(t)$ is the Kaplan-Meier estimate at the corresponding node in tree T (see Figure 3). The area between the true survival curve and the tree estimate is given by

$$ABC_i^T = \frac{1}{t_{max}} \int_0^{t_{max}} |1 - F_{C_i}(t) - \hat{S}_{T_i}(t)| dt. \tag{27}$$

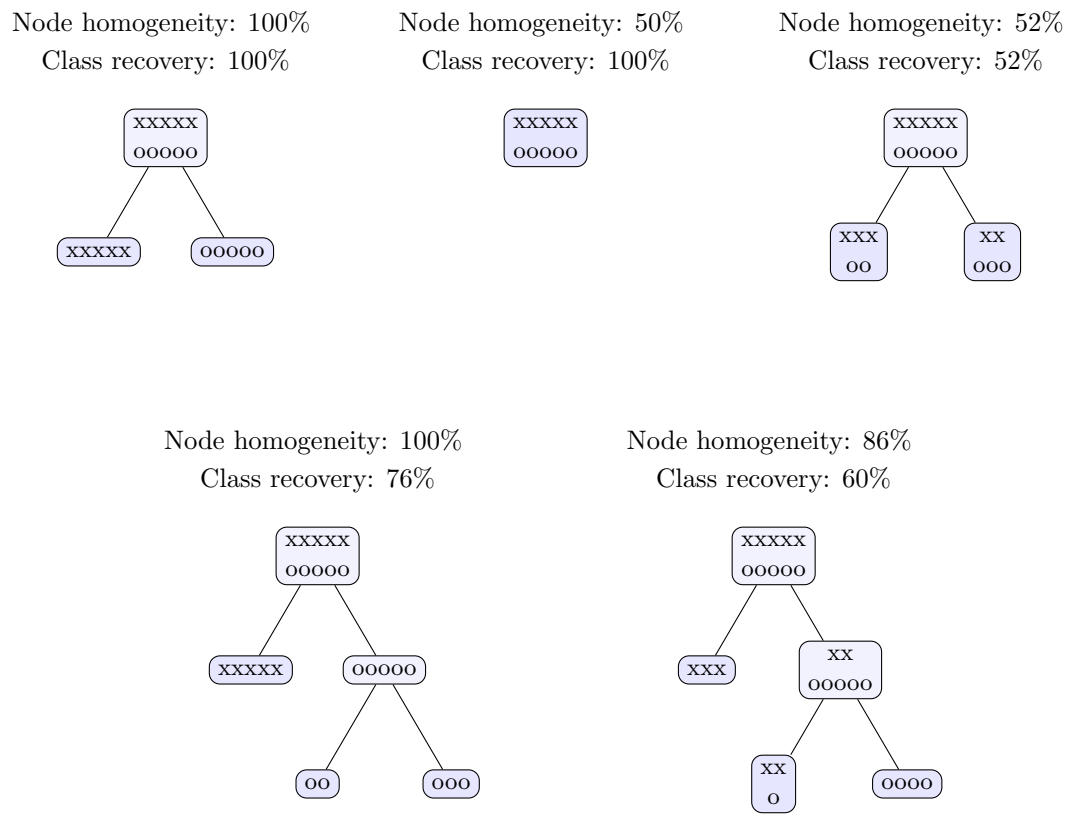

Fig. 2: Classification accuracy metrics for a survival tree with two classes of observations. The top left tree represents the true tree model.

To make this metric easier to interpret, we compare the area between curves in a given tree to the score of a null tree with a single node (T_0). The area ratio (AR) is given by

$$AR = 1 - \frac{\sum_i ABC_i^T}{\sum_i ABC_i^{T^0}}. \tag{28}$$

Similar to the popular R^2 metric for regression models, the AR indicates how much accuracy is gained by using the Kaplan-Meier estimates generated by the tree relative to the baseline accuracy obtained by using a single estimate for the whole population.

6 Simulation results

In this section we evaluate the performance of the Optimal Survival Trees (OST) algorithm and compare it to two existing survival tree models available in the **R** packages **rpart** and **ctree**. Our tests are performed on simulated datasets with the structure described in Section 5.2.

6.1 Simulation procedure

The procedure for generating simulated datasets in these experiments is as follows:

1. Randomly generate a sample of 20000 observations with six covariates. The first three covariates are uniformly distributed on the interval $[0, 1]$ and remaining three covariates are discrete uniform random variables with 2, 3 and 5 levels.
2. Generate a random "ground truth" tree model, C, that partitions the dataset based on these six covariates (see Algorithm 1 in the Appendix).

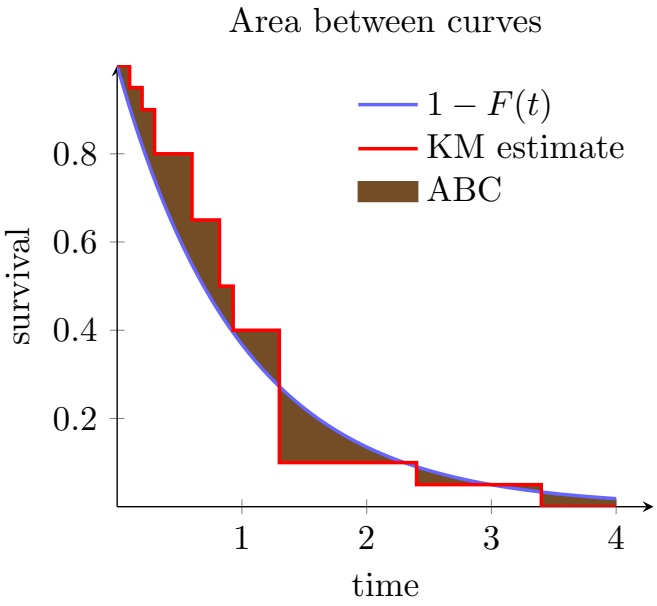

Fig. 3: An illustration of the area between the true survival distribution and the Kaplan-Meier curve.

3. Assign a survival distribution to each leaf node in the tree C (see Appendix for a list of distributions).
4. Classify observations into node classes $C_i = C(\mathbf{X}_i)$ according to the ground truth model. Generate a survival time, s_i, for each observation based the survival distribution of its node: $S_i \sim F_{C_i}(t)$.
5. Generate a censoring time for each observation, $c_i = \kappa(1 - u_i^2)$, where u_i follows a uniform distribution and κ is a non-negative parameter used to control the proportion of censored individuals.
6. Assign observation times $t_i = \min(s_i, c_i)$. Individuals are marked as censored ($\delta_i = 0$) if $t_i = c_i$.

We used this procedure to generate 1000 datasets based on ground truth trees with a minimum depth of 3 and a maximum depth of 4 (i.e., $2^4 = 16$ leaf nodes). In each instance, 10000 observations were set aside for testing the tree models. Training datasets of n observations were sampled from the remaining data for $n \in \{100, 200, 500, 1000, 2000, 5000, 10000\}$.

In addition to varying the size of the training dataset, we also varied the proportion of censored observations in the data by adjusting the parameter κ. Censoring was applied at nine different levels to generate examples with low censoring (0%, 10%, 20%), moderate censoring (30%, 40%, 50%) and high censoring (60%, 70%, 80%). In total, 63 OST models were trained for each dataset to test each of the seven training sample sizes at each of the nine censoring levels.

We evaluated the performance of the OST algorithm relative to two existing survival tree algorithms available in the **R** packages **rpart** (Therneau et al., 2010) and **ctree** (Hothorn et al., 2010). Each of the three algorithms was trained and tested on exactly the same data in each instance.

Each of the three algorithms tested require two input parameters that control the model size: a maximum tree depth and a complexity/significance parameter that determines which splits are worth keeping in the tree (the interpretation of the **ctree** significance parameter is different to the complexity parameters in the OST and **rpart** algorithms, but it serves a similar function).

Since neither **rpart** nor **ctree** have built-in methods for selecting tree parameters, we used a similar 5-fold cross-validation procedure to select the parameters for each algorithm. We considered tree depths up to three levels greater than the true tree depth and complexity parameter/significance values between 0.001 and 0.1 for the **rpart** and **ctree** algorithms (the OST complexity parameter is automatically selected during training). Equation (7) was used as the scoring metric to evaluate out-of-sample performance during cross-validation, and the minimum node size for all algorithms was fixed at 5 observations.

6.2 Results

To demonstrate the effect of this cross-validation procedure, we summarize the average size of the models produced by each algorithm in Figure 4. We see a clear link between tree size and the number of training observations, indicating the cross-validation procedure is selecting more conservative depth/complexity parameters when relatively little data is available. In larger datasets, the OST models grow to approximately the same size as the true tree models (6 nodes, on average), while the **rpart** and **ctree** models models are slightly larger.

6.2.1 Survival analysis metrics

Figure 5 summarizes the performance of each algorithm in our simulations using the four survival model metrics from Section 5. The values displayed in each chart are the average out-of-sample performance statistics across all test instances.

As expected, the average performance of all three algorithms consistently improves as the size of the training dataset increases. The performance statistics also increase as the proportion of censored observations increases, which seems counter-intuitive (we would expect more censoring to lead to less accurate models). In the case of the Cox partial likelihood and C-statistics, this trend is directly linked to the number of observed deaths, since only observations with observed deaths contribute to the partial likelihood and concordance scores. Similarly, censored observations do not contribute to the Integrated Brier Score after their censoring time.

Each chart also indicates the performance of the true tree model, C, as a point of comparison for the other algorithms. The true tree model performs significantly better than the empirical models trained on smaller datasets, but all three algorithms approach the performance of the true tree for very large sample sizes.

Based on these results, we conclude that the average performance of the OST algorithm in these simulations is consistently better than either of the other two algorithms. In order to understand why this algorithm is able to generate better models, we now analyse the results of the tree metrics introduced in Section 5.2.

6.2.2 Classification accuracy

The out-of-sample classification accuracy metrics for all three algorithms are summarized in Table 1 and Figure 6. The average node homogeneity/class recovery scores are given side-by-side to allow for a comprehensive assessment of each algorithm's performance. These results confirm that the OST models perform significantly better than the other two models across all censoring levels.

The node homogeneity scores for all three algorithms increase with larger sample sizes, indicating that the availability of additional data leads to better detection of relevant splits. In large populations, the OST algorithm selects more efficient splits than the other models and is able to achieve better node homogeneity with fewer splits (recall Figure 4 — the OST models trained on large data sets have fewer leaf nodes than the other models, on average).

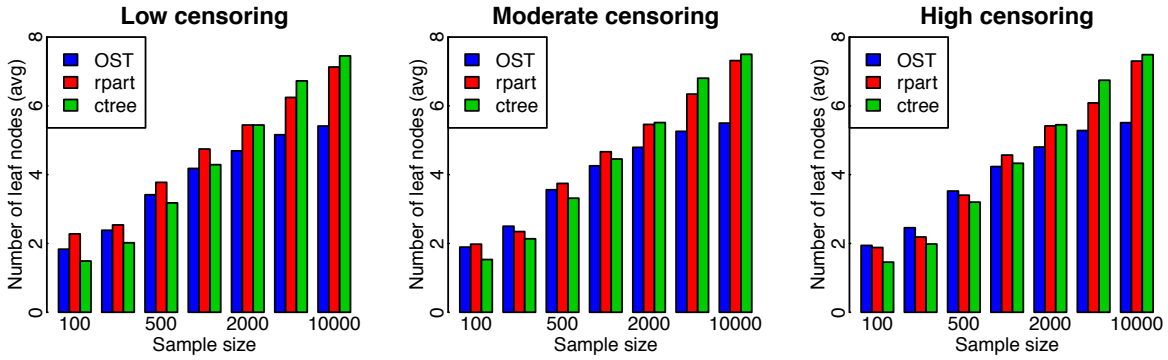

Fig. 4: The average tree size for models trained on various sample sizes.

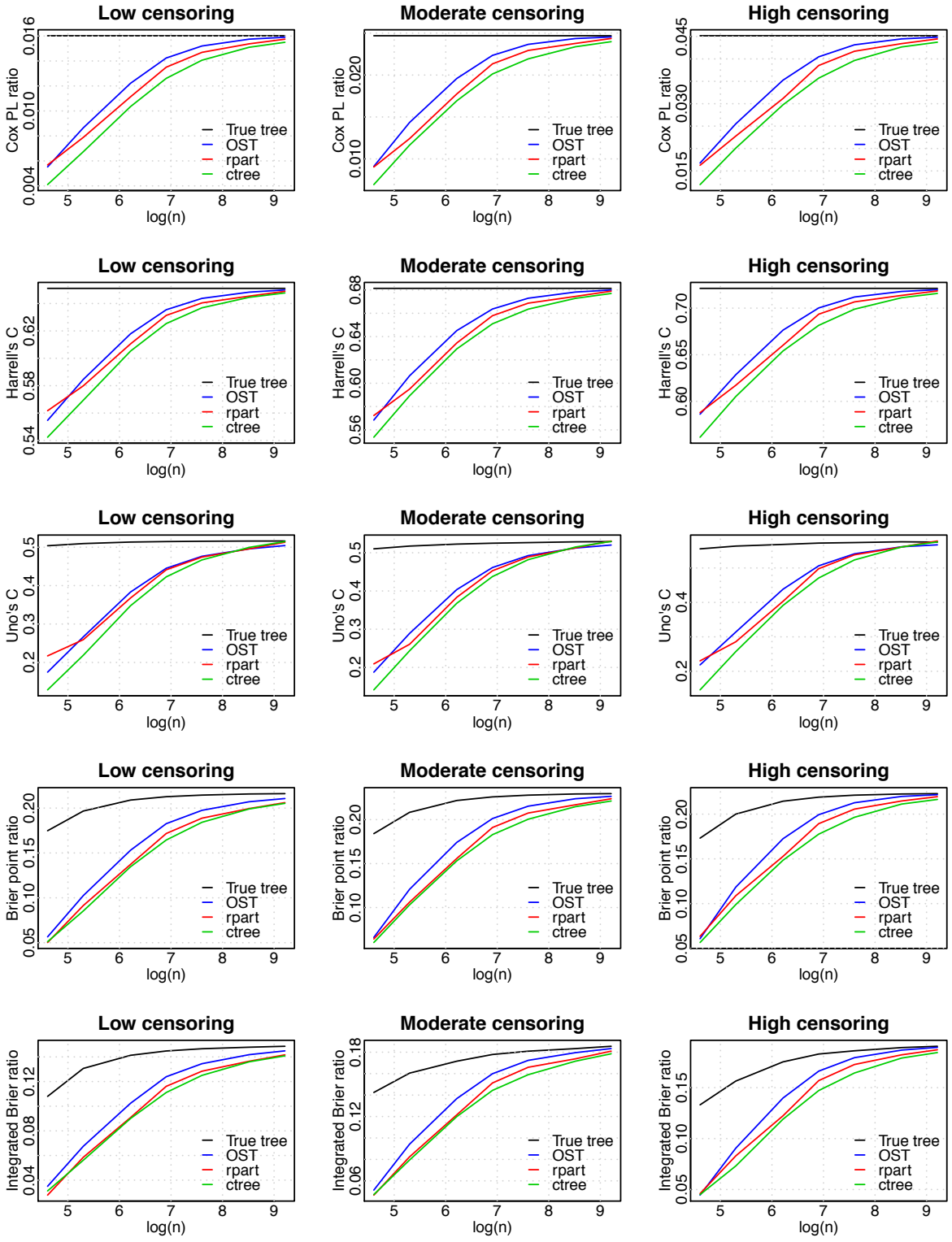

Fig. 5: A summary of the survival model metrics from simulation experiments. The average out-of-sample outcomes for each algorithm are shown in color, while the performance of the true tree model, C, in indicated in black.

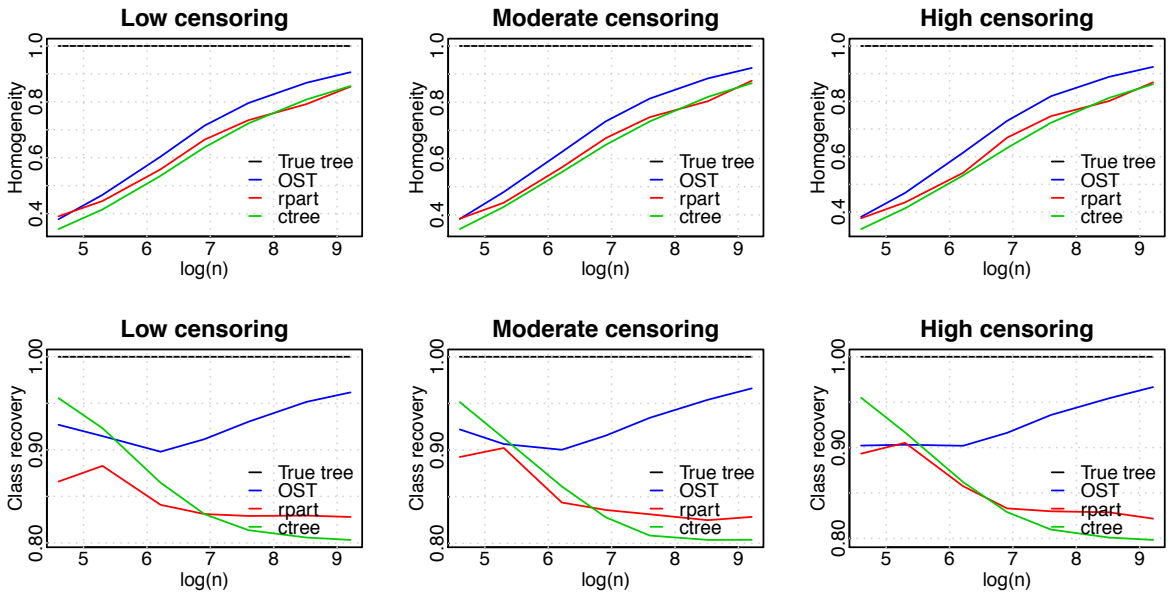

Fig. 6: A summary of the classification accuracy metrics for survival tree algorithms.

	Low censoring			Moderate censoring			High censoring		
n	rpart	ctree	OST	rpart	ctree	OST	rpart	ctree	OST
100	38/87	**40**/77	37/**93**	38/90	**40**/78	37/**92**	37/89	**40**/78	37/**90**
200	42/89	**45**/76	43/**91**	42/**90**	**46**/77	45/90	42/**91**	45/78	**45**/90
500	53/84	56/71	**57**/**88**	55/84	57/70	**59**/**88**	53/85	56/72	**59**/**88**
1000	63/82	66/63	**68**/**89**	65/82	67/63	**70**/**89**	64/82	66/64	**70**/**89**
2000	70/81	73/57	**76**/**89**	72/81	75/57	**78**/**90**	72/81	74/58	**78**/**90**
5000	76/80	82/53	**84**/**91**	77/80	83/53	**85**/**92**	77/80	82/53	**85**/**91**
10000	82/79	85/50	**87**/**91**	84/79	86/51	**89**/**92**	84/78	86/51	**88**/**91**

Table 1: A summary of the average node homogeneity/class recovery scores for synthetic experiments.

The relationship between tree size and class recovery rates is somewhat more complicated. In datasets smaller than 500 observations the class recovery rates seem to be closely linked to the tree size: the **ctree** models have the highest average class recovery for models trained on 100 and 200 observations, and also the smallest number of nodes (see Figure 4). However, this trend does not hold in datasets with 500 observations, where OST models are larger than the **ctree** models on average, but also have slightly better class recovery. This suggests that tree size is no longer a dominant factor in larger datasets ($n \geq 500$).

In these larger datasets we observe distinct trends in class recovery scores. The OST class recovery rate increases consistently despite the increases in model size, which means that the OST models are able to produce more complex trees without overfitting in the training data. By contrast, both of the other algorithms have consistently worse class recovery rates as sample size increases and their models become larger. Based on this trend, neither of these algorithms will reliably converge to the true tree.

6.2.3 Prediction accuracy

The out-of-sample prediction accuracy metric for each of the three algorithms is summarized in Table 2 and Figure 7. Overall, the results indicate that sample size plays the most significant role in out-of-sample accuracy across all three algorithms. There is also a small increase in accuracy when censoring is increased, which is due to the reduction in the maximum observed time, t_{max}. The OST results are generally better than the other algorithms across all sample sizes, although the performance gap is relatively small in smaller datasets.

To illustrate the effect of sample size on the accuracy of the Kaplan-Meier estimates, Figure 7 also shows the curve accuracy metrics for the true tree, C. It is immediately apparent that even the true tree models produce poor survival curve estimates in small datasets. Based on these results, it may be necessary

	Low censoring			Moderate censoring			High censoring		
n	rpart	ctree	OST	rpart	ctree	OST	rpart	ctree	OST
100	6.87	4.79	**9.30**	10.61	7.74	**11.01**	**10.79**	7.76	9.99
200	18.69	16.82	**20.99**	21.93	21.09	**25.25**	24.20	21.24	**26.13**
500	35.03	32.56	**41.17**	40.14	37.12	**47.16**	40.84	38.34	**48.21**
1000	51.27	44.29	**56.44**	57.28	49.68	**61.99**	58.86	51.30	**63.95**
2000	62.76	55.04	**67.97**	68.71	60.30	**73.53**	70.35	61.67	**75.31**
5000	72.62	66.94	**79.45**	77.26	71.63	**83.50**	79.22	72.38	**84.68**
10000	80.06	73.57	**84.41**	84.84	77.44	**87.77**	85.80	77.94	**88.72**

Table 2: A summary of the average Kaplan-Meier area ratio (AR) scores for simulation experiments.

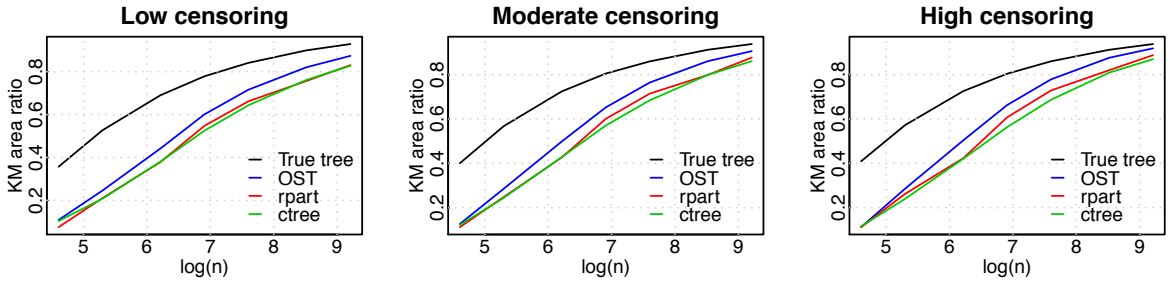

Fig. 7: A summary of the average Kaplan-Meier Area Ratio results for simulation experiments. The performance of the true tree model is indicated in black.

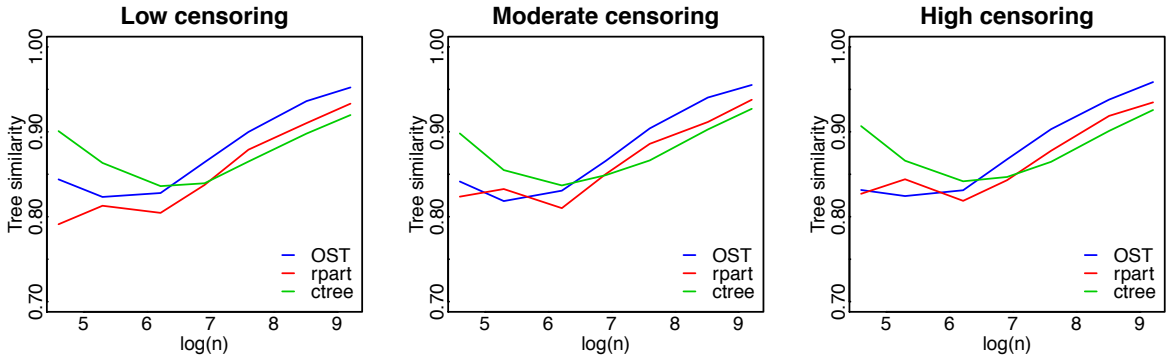

Fig. 8: A summary of the average similarity scores between pairs of trees trained on mutually exclusive sets of observations.

to increase the minimum node size to at least 50 observations in applications where Kaplan-Meier curves will be used to summarize survival tree nodes.

6.2.4 Stability

A frequent criticism of single-tree models is their sensitivity to small changes in the training data. This may be apparent when a tree algorithm produces very different models for different training datasets sampled from the same population. This type of instability is often an indication that the model will not perform well on unseen data.

Given the challenges associated with measuring out-of-sample accuracy for survival tree algorithms, it may be tempting to use stability as a performance metric for these models. Stability is a necessary condition for accuracy in tree models (provided that a tree structure is suitable for the data) but stable models are not necessarily accurate. For example, greedy tree models with depth 1 may select the same split for all permutations of the training data, but these models will not be accurate if the data requires a tree of depth 3.

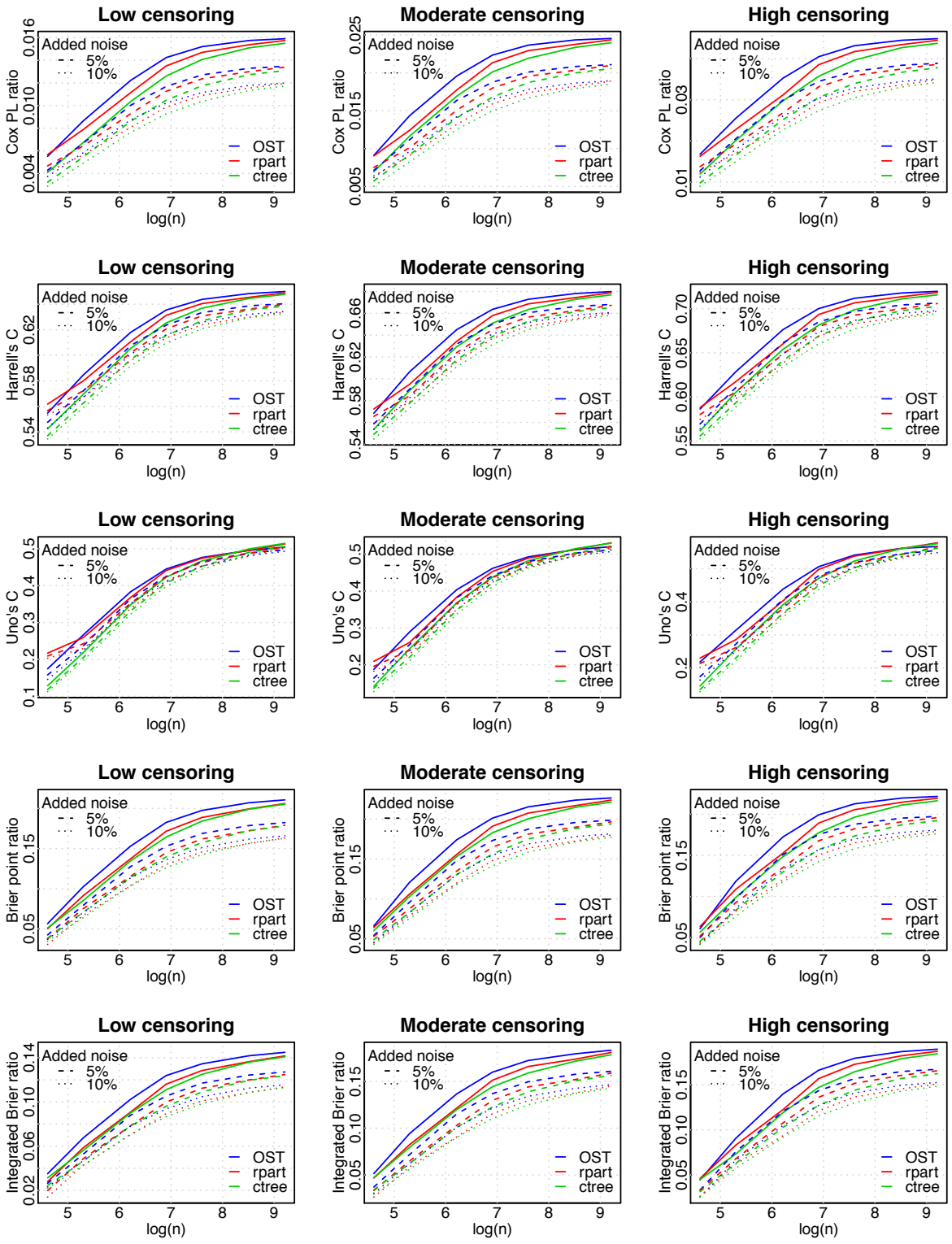

Fig. 9: A summary of survival tree accuracy metrics for datasets with added noise.

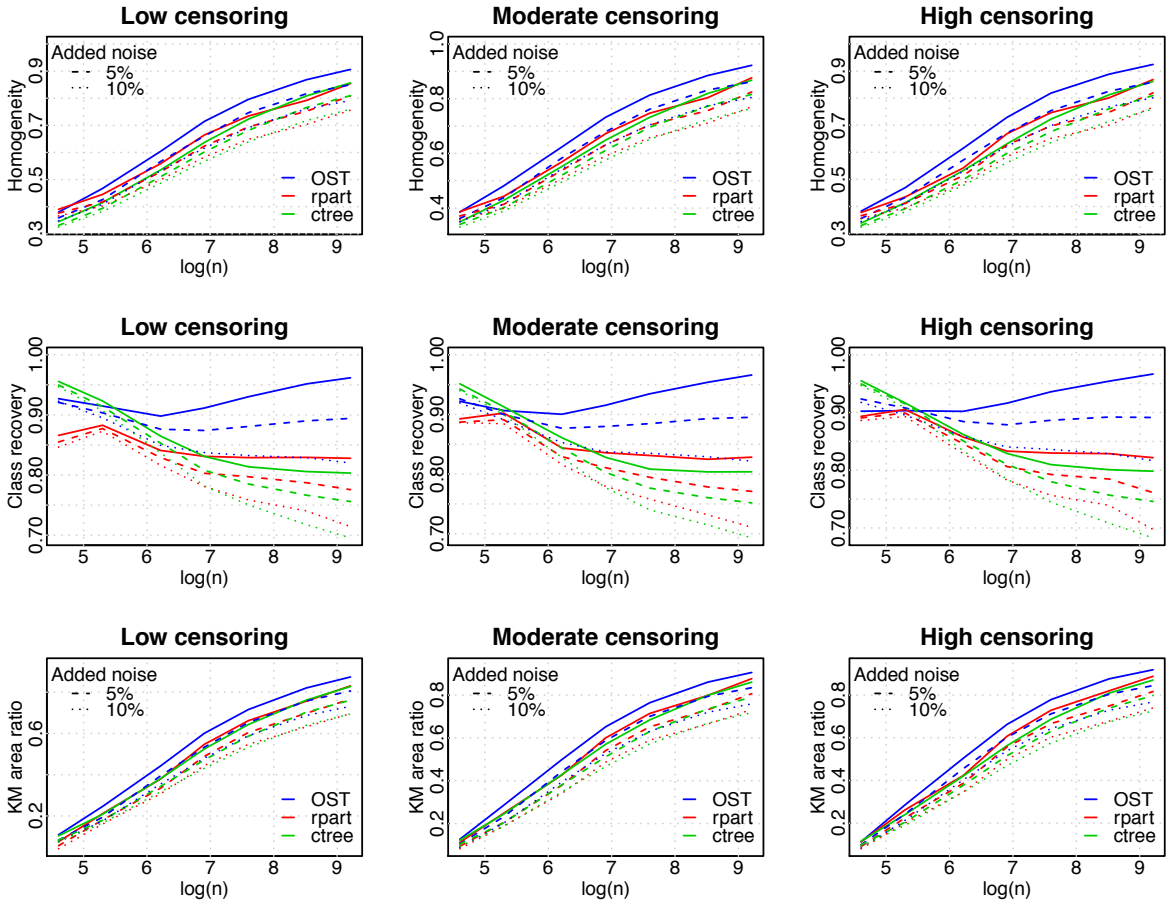

Fig. 10: A summary of simulation accuracy metrics for datasets with added noise.

Although stability is not necessarily a good indicator of the quality of a model, it is nevertheless interesting to consider how the stability of globally optimized trees may differ to the stability of greedy trees. Globally optimized trees are theoretically capable of greater stability because they may include splits that are not necessarily locally optimal for a particular training dataset. However, globally optimized trees also consider a significantly larger number of possible tree configurations and therefore have many more opportunities for overfitting on features of a particular training dataset.

We ran two sets of experiments to investigate the stability of the survival tree models in our simulations. In the first set of experiments we used each algorithm to train two models, T_1 and T_2, on completely separate training datasets of equal size. We then applied each model to the entire dataset (20000 observations) and used the tree similarity score described in Section 5.3 to assess the structural similarity between the two models. The average similarity scores for each algorithm are illustrated in Figure 8.

These results demonstrate that stability across different training datasets is not a sufficient condition for accuracy: models trained on 100 and 200 observations are both more stable and less accurate than models trained on 500 observations. The **ctree** algorithm produced the most stable results in smaller datasets due to the smaller model sizes selected during cross-validation. For example, 33.1% of **ctree** models trained on 100 observations had fewer than 2 splits, compared to 29.5% of the **rpart** models and 26.5% of the OST models.

The stability results for larger training datasets ($n > 1000$) are reasonably consistent with the accuracy metrics discussed above, and both stability and accuracy increase with sample size across all three algorithms. The OST models have the highest average similarity scores in large datasets and the **rpart** models are slightly more stable than the **ctree** models.

In the second set of stability experiments we investigated how small perturbations to the covariate values in the training dataset affect the out-of-sample accuracy of each model. We added noise to the training data by replacing the original continuous covariate values, x_{ij}, with "noisy" values $\tilde{x}_{ij} = x_{ij} + \epsilon_{ij}$. The

initial covariates were uniformly distributed between 0 and 1 and the added noise terms were generated from the following two distributions:

$$\epsilon_{ij} \sim U(-0.05, 0.05) \quad \text{(5\% noise), and}$$
$$\epsilon_{ij} \sim U(-0.1, 0.1) \quad \text{(10\% noise).}$$

A similar approach was applied to the categorical variables, which were generated by rounding off continuous values (x_{ij} or $\tilde{x}_{ij}$) to the appropriate thresholds. Note that noise was only added to the observations used for training data; the testing data was unchanged.

The results of these experiments are contrasted with the initial outcomes (without added noise) in Figures 9-10. The effects of additional noise in the training data are visible in the results of all three algorithms and the drop in accuracy appears to be fairly consistent. Overall, the OST models maintain the highest scores regardless of noise.

These results indicate that perturbations in the training data affect the OST and greedy tree algorithms in similar ways. The OST algorithm's performance is diminished by adding noise to the training data, but its ability to consider a wider range of split configurations does not make it more sensitive to these perturbations. In fact, the OST algorithm is generally slightly more stable than the greedy algorithms across permutations of the training data because it tends to produce models that are consistently closer to the true tree.

7 Computational results with real-world datasets

In this section, we compare the performance of the OST, **rpart** and **ctree** algorithms on 44 real-world datasets. The datasets used for this analysis were sourced from the UCI repository (Dua and Graff, 2017) and contained continuous outcome measures. The selected datasets[4] had sample sizes ranging from 63 observations to 100000, and the maximum number of features considered was 383.

For each observation in these datasets, we generated censoring values $c_i = \kappa(1 - u_i^2)$, where u_i follows a uniform distribution. We adjusted the parameter κ to generate different censoring levels (0%,10%,...,80%) within each dataset. We then split each dataset into training and testing sets (50%) and compared the performance of the three tree algorithms on each instance.

We applied the 5-fold cross-validation procedure described in Section 6.1 to select the depth and complexity of each tree, allowing tree depths of up to 7 (128 leaf nodes). Both the OST and **ctree** algorithms produced trees with over 100 leaf nodes in some of the largest datasets, while the largest **rpart** trees had only 77 nodes. The smaller size of the **rpart** trees indicates that larger models performed poorly in the cross-validation step.

On average, the OST models outperformed the other two algorithms across all 5 accuracy metrics. A summary of each algorithm's performance is given in Tables 3–4 and Figure 11, and aggregated results for each dataset are displayed in Table 5. The difference in performance was not statistically significant for the Cox ratios and Harrell's C scores, where all three algorithms had very similar average outcomes, but OST models did score significantly better than the other algorithms on the remaining three metrics. OST models achieved the best score in 48-60% of the instances tested, while the other algorithms each had undominated scores in 27-39% of instances.

	Mean score			Paired T-Test H_1:	
	OST	**rpart**	**ctree**	$S_{OST} > S_{rpart}$	$S_{OST} > S_{ctree}$
Cox Ratio	**0.1118**	0.1091	0.1090	p=0.2288	p=0.2222
Harrell's C	**0.7873**	0.7866	0.7818	p=0.4355	p=0.1045
Uno's C	**0.6650**	0.6523	0.6441	**p=0.0288**	**p=0.0013**
Brier Point Ratio	**0.3841**	0.3627	0.3516	**p=0.0001**	**p< 10^{-5}**
Intg. Brier Ratio	**0.4451**	0.4262	0.4231	**p=0.0135**	**p=0.0055**

Table 3: Average scores for OST, **rpart** and **ctree** models on real-world datasets. The final columns show the one-sided p-values for paired t-tests comparing the outcome metrics on each instance.

[4] We excluded the following types of datasets from our analysis: (1) datasets used for time series predictions (multiple observations of each individual); (2) datasets with unclear variable definitions; (3) datasets which required significant cleaning, pre-processing, or recoding; (4) datasets with too many variables (p) to cross-validate all three algorithms in reasonable times. Dataset selection was independent of the analysis of model accuracy.

	OST	rpart	ctree
Cox Ratio	48.7	32.8	36.4
Harrell's C	57.3	30.8	33.6
Uno's C	59.3	27.3	34.1
Brier Point Ratio	56.6	33.3	38.4
Intg. Brier Ratio	57.6	30.6	33.6

Table 4: The percentage of instances for which each algorithm was undominated by the other algorithms. Note that rows do not sum to 100, as several instances were tied.

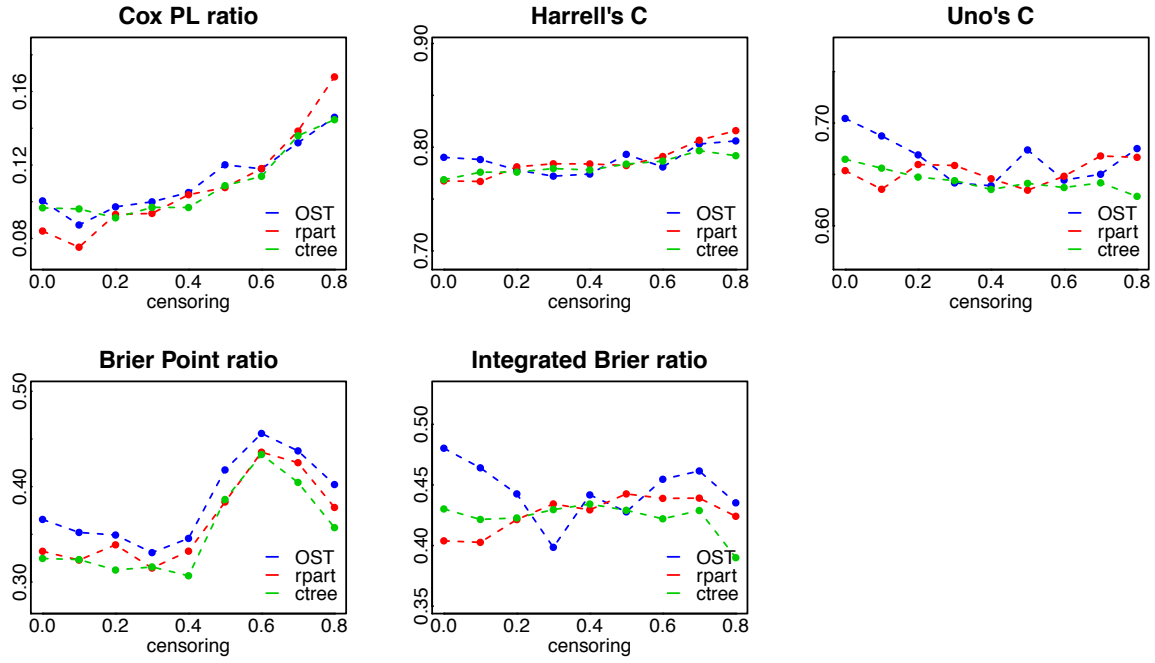

Fig. 11: Average performance of survival tree models on real datasets with different levels of censoring.

8 An Application to Heart Disease

In this section, we provide an example of a practical application of the OST algorithm to a real-world dataset from the Framingham Heart Study. Analysis of the FHS successfully identified the common factors or characteristics that contribute to CHD using the Cox regression model (Cox, 1972). In our survival tree model, we include all participants in the study from the original cohort (1948-2014) and the offspring cohort (1971-2014) who were diagnosed with Coronary Heart Disease (CHD). The event of interest in this model is the occurrence of a myocardial infarction or stroke. All patients were followed for a period of at least 10 years after their first diagnosis of CHD and observations are marked as censored if no event was observed while the patient was under observation.

We applied our algorithm to the primary variables that have been used in the established 10-year Hard Coronary Heart Disease (HCHD) Risk Calculator and the Cardiovascular Risk Calculator (Expert Panel on Detection and Evaluation and Treatment of High Blood Cholesterol in Adults, 2001; D'Agostino et al., 2008). For each participant who was diagnosed with CHD, we include the following covariates in our training dataset: gender, smoking status (smoke), Systolic Blood Pressure (SBP), Diastolic Blood Pressure (DBP), use of anti-hypertensive medication (AHT), Body Mass Index (BMI), diabetic status (diabetes). We did not include cholesterol levels in our analysis because these variables are highly correlated with the use of lipid lowering treatment and a high proportion of the sample population did not have sufficient data to account for this interaction.

In Figure 12 we illustrate the output of our algorithm on the FHS dataset. Every node of the tree provides the following information:

– The node number.

			Integrated Brier Score			Harrell's C Score			Uno's C Score			Cox Partial Likelihood			Brier Point Ratio		
Dataset	**n**	**p**	**OST**	**rpart**	**ctree**	**OST**	**rpart**	**ctree**	**OST**	**rpart**	**ctree**	**OST**	**rpart**	**ctree**	**OST**	**rpart**	**ctree**
3D Spatial Network (Kaul et al., 2013)	100000	1	**0.44**	0.33	0.39	**0.82**	0.77	0.79	**0.79**	0.73	0.76	**0.05**	**0.05**	**0.05**	**0.48**	0.35	0.41
Airfoil Self Noise (Dua and Graff, 2017)	1503	4	**0.39**	0.33	0.35	**0.83**	0.78	0.78	**0.77**	0.7	0.71	**0.09**	**0.09**	0.08	**0.53**	0.42	0.46
Appliances Energy Prediction (Candanedo et al., 2017)	19735	25	**0.19**	0.18	0.18	**0.74**	0.73	**0.74**	**0.7**	0.69	**0.7**	**0.03**	**0.03**	**0.03**	**0.14**	0.13	0.12
Automobile (Candanedo et al., 2017)	164	23	0.03	**0.07**	0.06	0.53	**0.65**	0.61	0.08	**0.41**	0.27	0.01	**0.05**	0.03	0	**0.11**	**0.11**
Auto MPG (Dua and Graff, 2017)	398	7	0.55	**0.56**	0.55	0.85	**0.87**	**0.87**	**0.79**	0.78	0.77	0.19	0.2	**0.21**	0.58	**0.6**	0.58
Behavior Urban Traffic	135	16	0.18	**0.2**	0.18	0.66	**0.67**	0.64	0.37	**0.41**	0.33	0.08	**0.09**	0.08	0.13	**0.16**	0.14
Bike Sharing	17379	13	0.92	0.88	**0.93**	**0.98**	0.96	**0.98**	**0.96**	0.93	**0.96**	0.15	0.09	**0.2**	0.94	0.91	**0.95**
Blog Feedback (Fanaee-T and Gama, 2014)	52397	279	**0.39**	**0.39**	0.38	0.84	**0.85**	**0.85**	0.79	0.8	**0.82**	**0.03**	**0.03**	**0.03**	0.17	**0.18**	0.17
Buzz in Social Media (Kawala et al., 2013)	100000	76	**0.77**	0.75	**0.77**	**0.92**	0.91	**0.92**	**0.91**	0.88	0.9	**0.13**	0.12	0.12	**0.76**	0.74	0.75
Cargo2000 (Metzger et al., 2015)	3943	95	**1**	**1**	0.84	**1**	**1**	0.95	**1**	**1**	0.9	**0.21**	**0.21**	0.16	0.22	**0.23**	0.17
Communities Crime (Redmond and Baveja, 2002)	2215	145	0.64	0.65	**0.69**	0.89	0.89	**0.91**	0.81	0.83	**0.85**	0.17	0.17	**0.19**	0.68	0.7	**0.75**
Computer Hardware (Dua and Graff, 2017)	209	8	**0.69**	0.61	0.65	**0.86**	0.83	0.85	**0.74**	0.67	0.7	0.24	0.27	**0.29**	**0.73**	0.68	0.62
Concrete Slump (Yeh, 2007)	103	6	0.07	**0.14**	0.03	0.62	**0.66**	0.56	0.27	**0.35**	0.14	0.04	**0.07**	0.02	0.11	**0.13**	0.05
Concrete Strength (Yeh, 1998)	1030	7	**0.42**	0.41	0.4	**0.84**	0.83	0.82	0.74	0.74	**0.75**	0.11	**0.13**	0.12	0.5	0.47	**0.51**
CSM (Ahmed et al., 2015)	232	11	0.25	**0.32**	0.25	0.71	**0.76**	0.73	0.48	0.56	**0.57**	0.08	**0.11**	0.09	0.34	**0.42**	0.26
Cycle Power	9568	3	**0.73**	0.71	**0.73**	**0.92**	0.91	**0.92**	**0.89**	0.86	**0.89**	0.16	0.17	**0.18**	**0.75**	0.72	**0.75**
Electrical Stability (Dua and Graff, 2017)	10000	11	**0.4**	0.34	0.39	**0.82**	0.79	**0.82**	**0.79**	0.75	**0.79**	**0.08**	0.06	**0.08**	**0.44**	0.37	**0.44**
Energy efficiency 1 (Tsanas and Xifara, 2012)	1296	7	**0.95**	0.9	0.9	**0.99**	0.97	0.98	**0.98**	0.95	0.93	**0.35**	0.3	0.31	-0.11	**-0.04**	-0.14
Energy efficiency 2 (Tsanas and Xifara, 2012)	1296	7	**0.94**	0.9	0.9	**0.99**	0.97	0.97	**0.97**	0.95	0.96	**0.27**	0.13	0.21	-0.14	**-0.01**	-0.16
Facebook Comments (Kamaljot et al., 2015)	40949	52	**0.56**	**0.56**	0.55	0.88	0.88	**0.89**	0.84	0.84	**0.86**	**0.06**	**0.06**	**0.06**	-0.1	**-0.09**	-0.11
Facebook Metrics (Moro et al., 2016)	500	6	**0.03**	0.02	0.02	0.55	**0.56**	0.53	0.1	**0.14**	0.05	**0.01**	**0.01**	**0.01**	**0.05**	**0.05**	0.02
Fires	517	11	**0**	**0**	**0**	**0.5**	**0.5**	**0.5**	**0**	**0**	**0**	**0**	**0**	**0**	**0.11**	**0.11**	**0.11**
GeoMusic (Zhou et al., 2014)	1059	115	0.03	**0.06**	0.03	0.58	**0.61**	0.59	0.32	0.37	**0.38**	0.01	**0.02**	0.01	0.02	**0.06**	0.03
Insurance Company Benchmark (van der Putten and van Someren, 2000)	5822	84	0.02	0.02	**0.03**	0.59	0.6	**0.62**	0.24	0.25	**0.27**	**0**	**0**	**0**	**0.33**	**0.33**	**0.33**
KEGG Directed (Dua and Graff, 2017)	53413	22	**0.81**	0.78	0.79	**0.96**	0.95	0.95	**0.94**	0.93	0.93	**0.11**	**0.11**	**0.11**	0.06	**0.07**	**0.07**
KEGG Undirected (Dua and Graff, 2017)	65554	25	**0.87**	0.81	0.86	0.96	0.95	**0.97**	**0.97**	0.94	0.96	0.14	0.16	**0.17**	**0.87**	0.81	0.85
Kernel Performance (Ballester-Ripoll et al., 2017)	100000	13	**0.73**	0.67	0.69	**0.84**	0.8	0.82	**0.84**	0.79	0.81	**0.09**	0.07	0.08	**0.53**	0.43	0.47
Las Vegas Strip (Moro et al., 2017)	504	18	**0.02**	0	0.01	**0.54**	0.51	0.52	**0.12**	0.05	0.05	**0**	**0**	**0**	**0.02**	-0.01	0
Online News Popularity	39644	58	**0.05**	**0.05**	**0.05**	0.62	0.62	**0.63**	0.56	0.57	**0.58**	**0.01**	**0.01**	**0.01**	0.16	0.16	**0.17**
Online Video Characteristics (Dua and Graff, 2017)	68784	19	0.75	0.7	**0.76**	**0.92**	0.91	**0.92**	**0.91**	0.89	**0.91**	0.06	**0.15**	**0.15**	**0.76**	0.73	**0.76**
Optical Interconnection Network (Aci and Akay, 2015)	640	8	-0.08	**0.32**	0.27	**0.81**	0.79	**0.81**	0.72	0.67	**0.73**	**0.12**	**0.12**	0.11	**0.7**	0.68	0.69
Parkinson Telemonitoring (Tsanas et al., 2010)	5875	18	**0.59**	0.49	0.42	**0.85**	0.82	0.8	**0.8**	0.77	0.73	**0.14**	0.1	0.08	**0.67**	0.55	0.48
PM2.5-Beijing (Liang et al., 2016)	50387	12	**0.35**	0.32	0.34	**0.8**	0.79	**0.8**	**0.77**	0.75	0.76	0.05	0.05	**0.06**	**0.43**	0.4	0.42
Propulsion Plant (Coraddu et al., 2016)	11934	15	**0.64**	0.43	0.28	**0.88**	0.8	0.72	**0.87**	0.74	0.57	**0.11**	0.08	0.04	**0.7**	0.46	0.32
Protein (Dua and Graff, 2017)	45730	8	**0.3**	0.26	0.26	**0.75**	0.73	0.72	**0.72**	0.69	0.69	**0.04**	0.03	0.03	**0.32**	0.27	0.27
Real Estate 1 (Yeh and Hsu, 2018)	414	5	**0.44**	0.4	0.42	**0.83**	0.79	0.8	**0.67**	0.58	0.64	**0.18**	0.15	0.15	**0.59**	0.56	0.56
Real Estate 2 (Yeh and Hsu, 2018)	53500	383	**0.8**	0.75	0.76	0.55	**0.92**	0.88	**0.9**	0.87	0.88	0.02	**0.1**	0.05	**0.83**	0.76	0.78
Residential Building (Rafiei and Adeli, 2016)	372	107	0.63	**0.64**	0.62	0.86	**0.87**	**0.87**	0.73	**0.76**	0.74	0.24	**0.27**	0.24	0.63	**0.68**	**0.68**
Servo (Dua and Graff, 2017)	167	3	**0.51**	0.4	0.39	**0.85**	0.73	0.69	**0.75**	0.5	0.41	**0.26**	0.16	0.14	**0.48**	0.29	0.24
Stock Market Istanbul (Akbilgic et al., 2013)	536	6	0.11	0.12	**0.14**	0.68	**0.69**	**0.69**	0.47	**0.5**	**0.5**	0.04	**0.05**	**0.05**	**0.18**	0.17	0.2
Stock Portfolio (Liu and Yeh, 2015)	63	11	**0.42**	0.26	0.29	**0.78**	0.73	0.74	**0.53**	0.46	0.43	**0.34**	0.26	0.27	**0.49**	0.4	0.35
Student Performance (Mohamed et al., 2016)	395	29	0.05	**0.08**	**0.08**	0.58	**0.61**	**0.61**	0.2	0.21	**0.26**	**0.02**	**0.02**	**0.02**	0.07	**0.11**	0.1
Wine Quality (Cortez et al., 2009)	6497	10	0.16	0.16	**0.18**	0.73	0.74	**0.76**	0.63	0.65	**0.71**	0.02	0.02	**0.03**	**-0.04**	**-0.04**	-0.07
Yacht (Dua and Graff, 2017)	308	5	**0.84**	0.8	0.82	**0.94**	0.91	0.9	**0.85**	0.81	0.77	0.37	**0.41**	0.4	0.8	0.76	**0.82**

Table 5: Average scores for OST, **rpart**, **ctree** for each dataset across all levels of censoring.

- Number of observations classified into the node.
- Proportion of the node population which has been censored.
- A plot of survival probability vs. time. In this example, the x-axis represents age and the y-axis gives the Kaplan-Meier estimate for the probability of experiencing no adverse events.
- Color-coded survival curves to describe the different sub-populations. In each node, the blue curves describe the individuals classified into that node.
- In internal (parent) nodes, the orange/green curves describe the sub-populations that are split into the left/right child node. After each split, the sub-population with higher likelihood of survival goes into the left node.
- In leaf nodes, the red curve shows the average survival curve for the entire tree. This facilitates easy comparisons between the survival of a specific node and the rest of the population.

The splits illustrated in Figure 12 include known risk factors for heart disease and are consistent with well-established medical guidelines. The algorithm identified a BMI threshold of 25 as the first split (node 1), which is in accordance with the NIH BMI ranges that classify an individual as overweight if his/her BMI is greater than or equal to 25. Multiple splits indicated a higher risk of heart attack or stroke in patients who smoke (nodes 2, 6). The group with the highest risk of an adverse event was overweight patients with diabetes (node 5).

Figures 13 and 14 illustrate the output of the **ctree** and **rpart** algorithms applied to the same FHS population. The **rpart** model has a single split (BMI), while the **ctree** model contains the same variables as the OST output. The Brier scores for each model are 0.0486 (OST), 0.0249 (**rpart**) and 0.0467 (**ctree**).

The discrepancy in the Brier scores for the OST and **ctree** models is due to slight differences in the threshold and position of certain splits. For example, both methods identify that BMI is the most appropriate variable for the first split, but the BMI threshold differs. The **ctree** model sets the splitting threshold to 24.117, which is the locally optimal value for the split when building the tree greedily (the same threshold is used in the **rpart** model). By contrast, the OST algorithm selects a threshold of 25.031. This example demonstrates how the OST algorithm's efforts to find a globally optimal solution differ from the results of locally optimal splits.

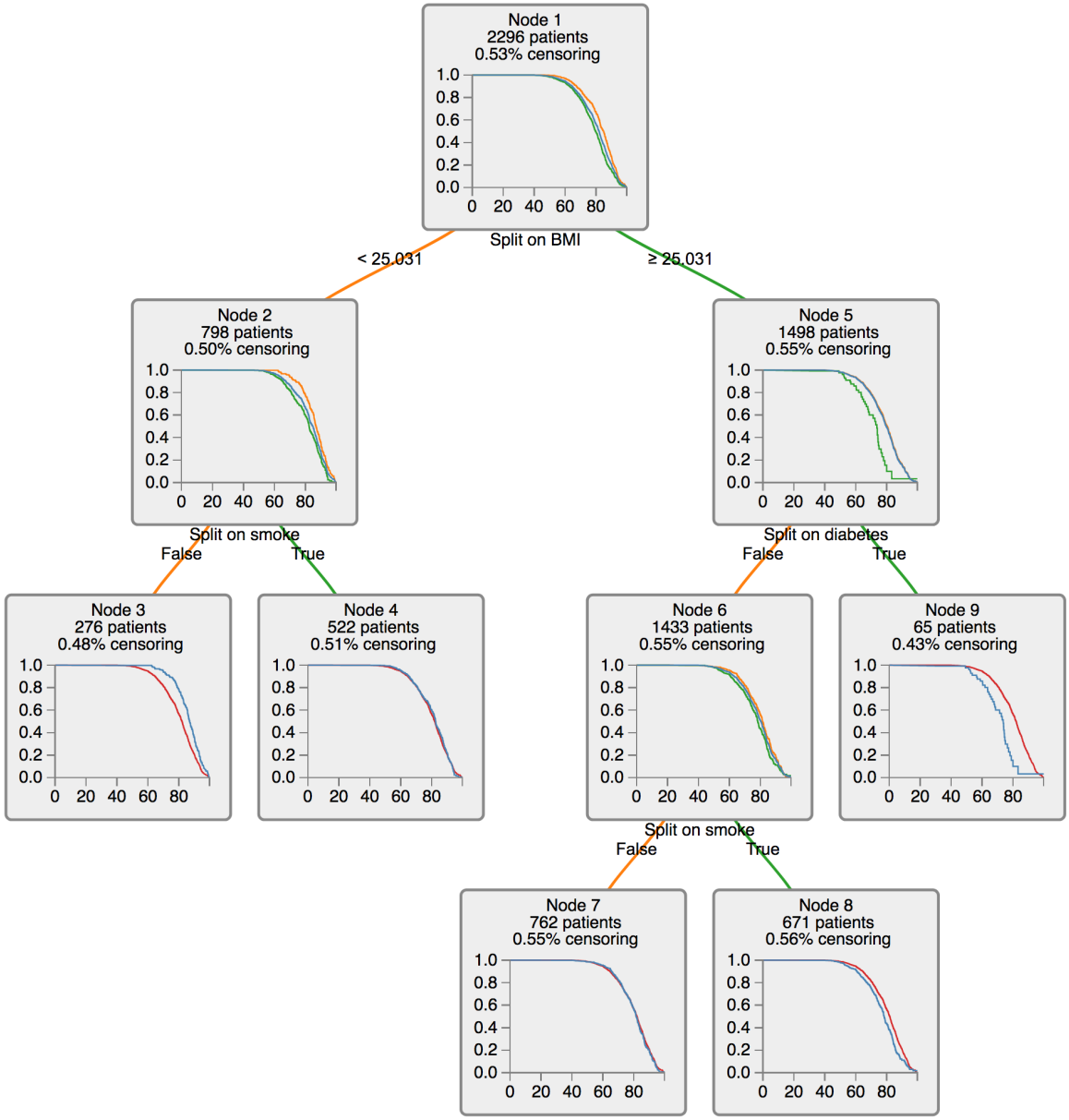

Fig. 12: An illustration of Optimal Survival Trees for chd patients in the FHS.

A second difference between the tree models is the order of the smoking and diabetes splits within the overweight population. The **ctree** model splits on smoking first, since this split has the most significant p-value of the variables at node 5 in the **ctree** tree. The algorithm also recognizes that diabetes is a risk factor and incorporates this in the subsequent split. Since greedy approaches like **ctree** do not reevaluate the spits once they have been decided, the algorithm does not recognize that the overall quality of the tree can be improved by reversing the order of these splits. This discrepancy in two otherwise similar trees highlights the advantages of the more sophisticated optimization conducted by OST.

9 Conclusion

In this paper, we have extended the state-of-the-art Optimal Trees framework to generate interpretable models for censored data. We have also introduced a new accuracy metric, the Kaplan-Meier Area Ratio, which provides an effective way to measure the predictive power of survival tree models in simulations.

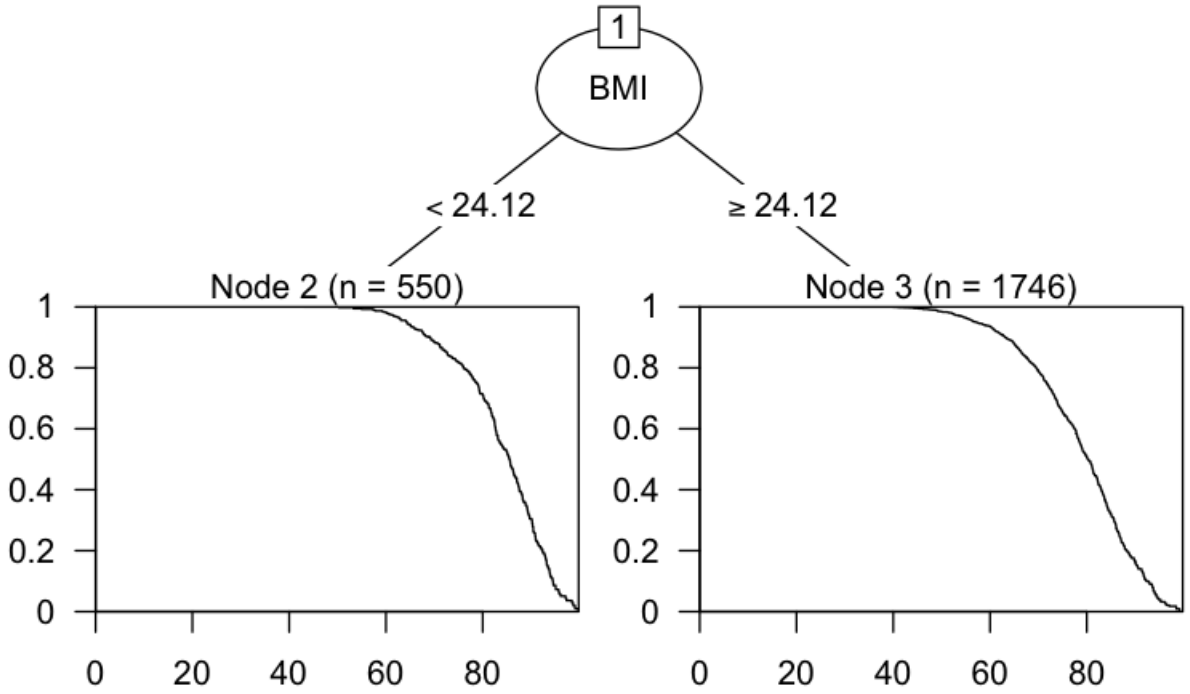

Fig. 13: Illustration of the **rpart** output for chd patients in the FHS.

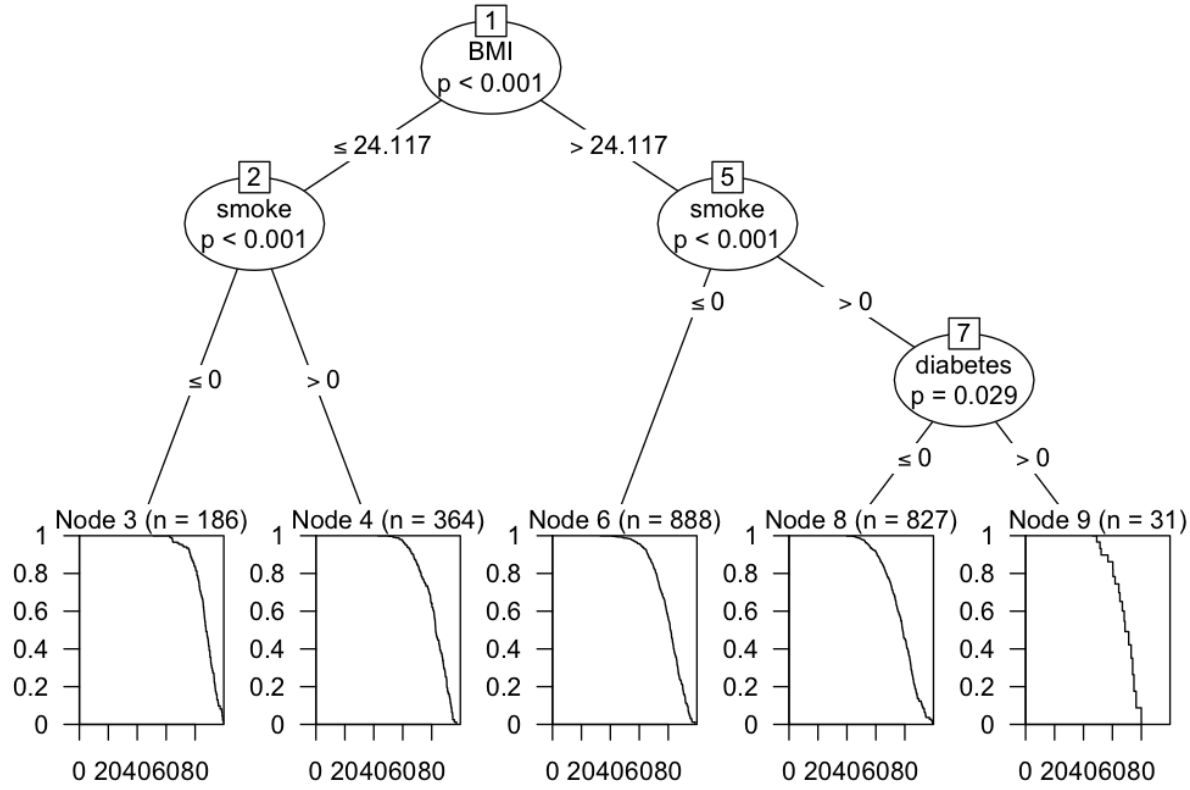

Fig. 14: Illustration of the **ctree** output for chd patients in the FHS.

The Optimal Survival Trees algorithm improves on the performance of existing algorithms in terms of both classification and predictive accuracy. Our results in simulations indicate that the OST models improve consistently with increasing sample size, whereas existing algorithms are prone to overfitting in larger datasets. This is particularly important, given that the volume of medical data available for research is likely to increase significantly over the coming years.

References

Aalen, O. (1978). Nonparametric inference for a family of counting processes. *The Annals of Statistics* pp 701–726

Aci, C. I., Akay, M. F. (2015). A hybrid congestion control algorithm for broadcast-based architectures with multiple input queues. *The Journal of Supercomputing* 71(5):1907–1931, DOI 10.1007/s11227-015-1384-1, URL https://doi.org/10.1007/s11227-015-1384-1

Ahmed, M., Jahangir, M., Afzal, H., Majeed, A., Siddiqi, I. (2015). Using crowd-source based features from social media and conventional features to predict the movies popularity. In: 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), pp 273–278, DOI 10.1109/SmartCity.2015.83

Akbilgic, O., Bozdogan, H., Balaban, M. E. (2013). A novel hybrid RBF neural networks model as a forecaster. *Statistics and Computing* 24, DOI 10.1007/s11222-013-9375-7

Ballester-Ripoll, R., G. Paredes, E., Pajarola, R. (2017). Sobol tensor trains for global sensitivity analysis. *Reliability Engineering and System Safety* 183, DOI 10.1016/j.ress.2018.11.007

Bennett, K., Blue, J. (1996). Optimal decision trees. *Rensselaer Polytechnic Institute Math Report* 214

Bertsimas, D., Dunn, J. (2017). Optimal classification trees. *Machine Learning* pp 1–44

Bertsimas, D., Dunn, J. (2019). *Machine Learning under a Modern Optimization Lens*. Dynamic Ideas, Belmont, to appear

Bezanson, J., Edelman, A., Karpinski, S., Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review* 59(1):65–98, URL https://doi.org/10.1137/141000671

Bou-Hamad, I., Larocque, D., Ben-Ameur, H. (2011). A review of survival trees. *Statistics Surveys* 5:44–71

Breiman, L. (2002). Software for the masses. URL www.stat.berkeley.edu/ breiman/wald2002-3.pdf

Breiman, L., Friedman, J., Stone, C., Olshen, R. (1984). *Classification and regression trees*. CRC press

Brier, G. (1950). Verification of forecasts expressed in terms of probability. *Monthey Weather Review* 78(1):1–3

Candanedo, L. M., Feldheim, V., Deramaix, D. (2017). Data driven prediction models of energy use of appliances in a low-energy house. *Energy and buildings* 140:81–97

Ciampi, A., Thiffault, J., Nakache, J., Asselain, B. (1986). Stratification by stepwise regression, correspondence analysis and recursive partition: a comparison of three methods of analysis for survival data with covariates. *Computational statistics & data analysis* 4(3):185–204

Ciampi, A., Chang, C.-H., Hogg, S., McKinney, S. (1987). Recursive partition: A versatile method for exploratory data analysis in biostatistics. *Biostatistics* pp 23–50

Coraddu, A., Oneto, L., Ghio, A., Savio, S., Anguita, D., Figari, M. (2016). Machine learning approaches for improving condition-based maintenance of naval propulsion plants. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment* 230(1):136–153, DOI 10.1177/1475090214540874, URL https://doi.org/10.1177/1475090214540874, https://doi.org/10.1177/1475090214540874

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

Cox, D. (1972). Regression models and life tables. *Journal of the Royal Statistical Society* 34:187–220

Cox, D. R. (1975). Partial likelihood. *Biometrika* 62(2):269–276

Davis, R., Anderson, J. (1989). Exponential survival trees. *Statistics in medicine* 8(8):947–961

Dua, D., Graff, C. (2017). UCI machine learning repository. URL http://archive.ics.uci.edu/ml

Dunn, J. (2018). Optimal Trees for prediction and prescription. PhD thesis, Massachusetts Institute of Technology

D'Agostino, R., Vasan, R., Pencina, M., Wolf, P., Cobain, M., Massaro, J., Kannel, W. (2008). General cardiovascular risk profile for use in primary care. *Circulation* 117

Expert Panel on Detection and Evaluation and Treatment of High Blood Cholesterol in Adults, (2001). Executive summary of the third report of the national cholesterol education program (ncep) expert panel on detection, evaluation, and treatment of high blood cholesterol in adults (adult treatment panel III). *JAMA* 285

Fanaee-T, H., Gama, J. (2014). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence* 2(2-3):113–127

Gordon, L., Olshen, R. (1985). Tree-structured survival analysis. *Cancer treatment reports* 69(10):1065–1069

Graf, E., Schmoor, C., Sauerbrei, W., Schumacher, M. (1999). Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine* 18(17-18):2529–2545

Grubinger, T., Zeileis, A., Pfeiffer, K.-P. (2014). evtree: Evolutionary learning of globally optimal classification and regression trees in R. *Journal of statistical software* 61(1):1–29, DOI 10.18637/jss.v061.i01, URL https://www.jstatsoft.org/v061/i01

Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., Rosati, R. A. (1982). Evaluating the yield of medical tests. *Jama* 247(18):2543–2546

Hothorn, T., Lausen, B., Benner, A., Radespiel-Tröger, M. (2004). Bagging survival trees. *Statistics in medicine* 23(1):77–91

Hothorn, T., Hornik, K., Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics* 15(3):651–674

Hothorn, T., Hornik, K., Strobl, C., Zeileis, A. (2010). Party: A laboratory for recursive partitioning

Ishwaran, H., Kogalur, U., Blackstone, E., Lauer, M. (2008). Random survival forests. *The annals of applied statistics* pp 841–860

Kamaljot, S., Ranjeet Kaur, S., Kumar, D. (2015). Comment volume prediction using neural networks and decision trees. In: IEEE UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim2015 (UKSim2015), Cambridge, United Kingdom

Kaplan, E., Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American statistical association* 53(282):457–481

Kaul, M., Yang, B., Jensen, C. S. (2013). Building accurate 3d spatial networks to enable next generation intelligent transportation systems. In: 2013 IEEE 14th International Conference on Mobile Data Management, IEEE, vol 1, pp 137–146

Kawala, F., Douzal-Chouakria, A., Gaussier, E., Dimert, E. (2013). Prédictions d'activité dans les réseaux sociaux en ligne. In: 4ième conférence sur les modèles et l'analyse des réseaux : Approches mathématiques et informatiques, France, p 16, URL https://hal.archives-ouvertes.fr/hal-00881395

LeBlanc, M., Crowley, J. (1992). Relative risk trees for censored survival data. *Biometrics* pp 411–425

LeBlanc, M., Crowley, J. (1993). Survival trees by goodness of split. *Journal of the American Statistical Association* 88(422):457–467

Liang, X., Li, S., Zhang, S., Huang, H., Chen, S. X. (2016). Pm2.5 data reliability, consistency, and air quality assessment in five chinese cities. *Journal of Geophysical Research: Atmospheres* 121(17):10,220–10,236, DOI 10.1002/2016JD024877, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016JD024877, https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2016JD024877

Liu, Y.-c., Yeh, I.-C. (2015). Using mixture design and neural networks to build stock selection decision support systems. *Neural Computing and Applications* 28, DOI 10.1007/s00521-015-2090-x

Metzger, A., Leitner, P., Ivanović, D., Schmieders, E., Franklin, R., Carro, M., Dustdar, S., Pohl, K. (2015). Comparing and combining predictive business process monitoring techniques. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45(2):276–290, DOI 10.1109/TSMC.2014.2347265

Mohamed, A., Rizaner, A., Ulusoy, A. H. (2016). Using data mining to predict instructor performance. *Procedia Computer Science* 102:137 – 142, DOI https://doi.org/10.1016/j.procs.2016.09.380, URL http://www.sciencedirect.com/science/article/pii/S1877050916325601, 12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016, 29-30 August 2016, Vienna, Austria

Molinaro, A., Dudoit, S., Van der Laan, M. (2004). Tree-based multivariate regression and density estimation with right-censored data. *Journal of Multivariate Analysis* 90(1):154–177

Moro, S., Rita, P., Vala, B. (2016). Predicting social media performance metrics and evaluation of the impact on brand building: A data mining approach. *Journal of Business Research* 69(9):3341 – 3351, DOI https://doi.org/10.1016/j.jbusres.2016.02.010, URL http://www.sciencedirect.com/science/article/pii/S0148296316000813

Moro, S., Rita, P., Coelho, J. (2017). Stripping customers' feedback on hotels through data mining: The case of Las Vegas Strip. *Tourism Management Perspectives* 23:41 – 52, DOI https://doi.org/10.1016/j.tmp.2017.04.003, URL http://www.sciencedirect.com/science/article/pii/S2211973617300387

Nelson, W. (1972). Theory and applications of hazard plotting for censored failure data. *Technometrics* 14(4):945–966

R Core Team, (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org/

Radespiel-Tröger, M., Rabenstein, T., Schneider, H., Lausen, B. (2003). Comparison of tree-based methods for prognostic stratification of survival data. *Artificial Intelligence in Medicine* 28(3):323–341

Rafiei, M. H., Adeli, H. (2016). A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineer-*

ing and Management 142(2):04015066, DOI 10.1061/(ASCE)CO.1943-7862.0001047, URL https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29CO.1943-7862.0001047, https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29CO.1943-7862.0001047

Reddy, A., Kronek, L.-P. (2008). Logical analysis of survival data: prognostic survival models by detecting high-degree interactions in right-censored data. *Bioinformatics* 24(16):i248–i253, DOI 10.1093/bioinformatics/btn265, URL https://doi.org/10.1093/bioinformatics/btn265, http://oup.prod.sis.lan/bioinformatics/article-pdf/24/16/i248/507165/btn265.pdf

Redmond, M., Baveja, A. (2002). A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research* 141(3):660–678

Son, N. (1998). From optimal hyperplanes to optimal decision trees. *Fundamenta Informaticae* 34(1, 2):145–174

Therneau, T., Grambsch, P., Fleming, T. (1990). Martingale-based residuals for survival models. *Biometrika* 77(1):147–160

Therneau, T., Atkinson, B., Ripley, B. (2010). The rpart package

Tsanas, A., Xifara, A. (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings* 49:560 – 567, DOI https://doi.org/10.1016/j.enbuild.2012.03.003, URL http://www.sciencedirect.com/science/article/pii/S037877881200151X

Tsanas, A., Little, M. A., McSharry, P. E., Ramig, L. O. (2010). Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *IEEE Transactions on Biomedical Engineering* 57(4):884–893, DOI 10.1109/TBME.2009.2036000

Uno, H., Cai, T., Pencina, M. J., D'Agostino, R. B., Wei, L. (2011). On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine* 30(10):1105–1117

van der Putten, P., van Someren, M. (2000). CoIL Challenge 2000: The insurance company case. https://kdd.ics.uci.edu/databases/tic/tic.data.html, accessed: 2019-04-25

Yeh, I.-C. (1998). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research* 28(12):1797–1808

Yeh, I.-C. (2007). Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cement and Concrete Composites* 29(6):474–480

Yeh, I.-C., Hsu, T.-K. (2018). Building real estate valuation models with comparative approach through case-based reasoning. *Applied Soft Computing* 65, DOI 10.1016/j.asoc.2018.01.029

Zhou, F., Claire, Q., King, R. D. (2014). Predicting the geographical origin of music. In: 2014 IEEE International Conference on Data Mining, pp 1115–1120, DOI 10.1109/ICDM.2014.73

Zhou, Y., McArdle, J. (2015). Rationale and applications of survival tree and survival ensemble methods. *Psychometrika* 80(3):811–833

A Tree simulations

A.1 Tree generation algorithm

Algorithm 1 was used to generate ground truth models for simulated datasets.

Algorithm 1 Tree generation algorithm

Inputs:
  X ▷ $n \times p$ data matrix
  min_bucket ▷ minimum node population
  max_depth ▷ maximum tree depth

INITIALIZE:
  $T \leftarrow \{1\}$ ▷ list of tree nodes, node 1 is the root node
  status(1) $\leftarrow$ open ▷ node status: open nodes may be split, closed nodes are leaf nodes
  population(1) $\leftarrow$ {1,2,...,n} ▷ observations in each node
  depth(1) $\leftarrow$ 1 ▷ depth of the node in the tree

GROW TREE:
  while status(k) = open for any $k \in T$ **do**
    current_node = select($k \mid k \in T$ and status(k) = open) ▷ Select an open node to split
    feature_list = permute(1:p+1) ▷ Randomly order features
    for $j \in$ feature_list **do**
      if $j = p + 1$ or depth(current_node) = max_depth **then**
        status(current_node) $\leftarrow$ closed ▷ Close node without splitting
        break and go to (A)
      feature_values = permute(unique($\{X_{ij} \mid i \in$ population(current_node)$\}$))
      for $b \in$ feature_values **do** ▷ Attempt to split on feature j with threshold b
        L_1 = length($\{i \mid i \in$ population(current_node), $X_{ij} \leq b$ $\}$)
        L_2 = length($\{i \mid i \in$ population(current_node), $X_{ij} > b$ $\}$)
        if $L_1 \geq$ min_bucket and $L_2 \geq$ min_bucket **then** ▷ If split is feasible, create new nodes
          $T = T \bigcup$ {total_nodes+1,total_nodes+2}
          status(total_nodes+1) = open
          depth(total_nodes+1) = depth(current_node)+1
          population(total_nodes+1) = $\{i \mid i \in$ population(current_node), $X_{ij} \leq b\}$
          status(total_nodes+2) = open
          depth(total_nodes+2) = depth(current_node)+1
          population(total_nodes+2) = $\{i \mid i \in$ population(current_node), $X_{ij} > b\}$
          status(current_node) $\leftarrow$ closed ▷ Current node is closed
          break and go to GROW TREE ▷ Select another open node to split
  Return T

A.2 Survival distributions

Nodes in simulated trees were randomly assigned one of the following survival distributions:

- Exponential(θ):
  Parameters: 0.3, 0.4, 0.6, 0.8, 0.9, 1.15, 1.5, 1.8
- Weibull(k,λ):
  Parameters: (0.8,0.4), (0.9,0.5), (0.9,0.7), (0.9,1.1), (0.9,1.5), (1,1.1),(1,1.9), (1.3,0.5)
- Lognormal(μ,σ^2):
  Parameters: (0.1,1.0), (0.2,.75), (0.3,0.3), (0.3,0.5), (0.3,0.8), (0.4,0.32), (0.5,0.3), (0.5,0.7)
- Gamma(k,θ):
  Parameters: (0.2,.75), (0.3,1.3), (0.3,2), (0.5,1.5), (0.8,1.0), (0.9,1.3), (1.4,0.9), (1.5,0.7)

A.3 FHS dataset

FHS patients inclusion criteria

- Participation in the Original and Offspring cohort of the FHS.
- Formal diagnosis with chd (as indicated by the records of FHS).
- Participants outcomes were followed for 10 consecutive years after diagnosis.

B Real data tests

B.1 Detailed results for UCI datasets

Table 6: Dataset specific Brier Score Results for each level of censoring.

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
3D Spatial Network	OST	0.40	0.42	0.39	0.43	0.43	0.47	0.50	0.48	0.48
	rpart	0.31	0.31	0.31	0.33	0.31	0.34	0.34	0.34	0.37
	ctree	0.36	0.41	0.39	0.39	0.39	0.39	0.41	0.40	0.35
Airfoil Self Noise	OST	0.47	0.49	0.41	0.40	0.36	0.22	0.35	0.32	0.46
	rpart	0.37	0.35	0.28	0.33	0.22	0.32	0.31	0.33	0.42
	ctree	0.39	0.39	0.37	0.34	0.32	0.29	0.26	0.34	0.46
Appliances Energy Prediction	OST	0.14	0.14	0.13	0.21	0.22	0.22	0.27	0.24	0.16
	rpart	0.13	0.14	0.16	0.16	0.21	0.22	0.22	0.21	0.17
	ctree	0.15	0.14	0.13	0.17	0.18	0.21	0.23	0.20	0.17
Automobile	OST	0.00	0.06	0.04	0.00	0.05	0.11	0.00	0.00	0.00
	rpart	0.06	0.19	0.15	0.13	0.05	0.18	0.02	-0.01	-0.10
	ctree	0.08	0.05	0.06	0.07	0.06	0.10	0.08	0.10	-0.10
AutoMPG	OST	0.59	0.60	0.59	0.54	0.49	0.58	0.49	0.49	0.57
	rpart	0.59	0.57	0.57	0.55	0.45	0.57	0.58	0.60	0.54
	ctree	0.60	0.57	0.54	0.49	0.57	0.54	0.57	0.47	0.58
Behavior Urban Traffic	OST	0.30	0.26	0.14	0.19	0.25	0.16	0.19	0.11	0.00
	rpart	0.30	0.26	0.14	0.19	0.34	0.23	0.19	0.11	0.07
	ctree	0.30	0.26	0.14	0.25	0.32	0.12	0.18	0.00	0.00
BikeSharing	OST	0.93	0.90	0.89	0.92	0.91	0.93	0.92	0.94	0.94
	rpart	0.77	0.84	0.87	0.89	0.88	0.92	0.91	0.92	0.94
	ctree	0.94	0.92	0.92	0.93	0.93	0.94	0.94	0.93	0.92
Blog Feedback	OST	0.29	0.35	0.39	0.40	0.42	0.43	0.41	0.40	0.38
	rpart	0.30	0.35	0.39	0.40	0.42	0.44	0.43	0.41	0.38
	ctree	0.31	0.36	0.37	0.38	0.40	0.41	0.41	0.39	0.37
Buzz in Social Media	OST	0.74	0.73	0.74	0.76	0.78	0.80	0.80	0.81	0.80
	rpart	0.68	0.68	0.70	0.73	0.77	0.79	0.80	0.81	0.80
	ctree	0.74	0.74	0.73	0.76	0.77	0.79	0.80	0.81	0.78
Cargo2000	OST	1.00	1.00	1.00	0.99	0.99	1.00	1.00	1.00	1.00
	rpart	1.00	1.00	1.00	1.00	0.99	1.00	0.99	1.00	1.00
	ctree	0.70	0.69	0.79	0.80	0.80	0.82	0.98	1.00	1.00
Communities Crime	OST	0.68	0.65	0.59	0.61	0.63	0.63	0.64	0.63	0.69
	rpart	0.56	0.63	0.64	0.62	0.69	0.69	0.70	0.63	0.68
	ctree	0.72	0.69	0.67	0.70	0.70	0.68	0.66	0.66	0.72
Computer Hardware	OST	0.86	0.74	0.84	0.84	0.52	0.80	0.65	0.51	0.48
	rpart	0.72	0.28	0.81	0.61	0.76	0.59	0.60	0.54	0.61
	ctree	0.82	0.85	0.83	0.77	0.52	0.72	0.28	0.74	0.35
Concrete Slump	OST	0.22	0.11	0.07	0.04	0.00	0.08	0.09	0.00	0.05
	rpart	0.15	0.11	0.07	0.04	0.06	0.08	0.20	0.31	0.18
	ctree	0.00	0.00	0.00	0.04	0.00	0.08	0.09	0.00	0.05
Concrete Strength	OST	0.52	0.49	0.45	0.31	0.35	0.35	0.45	0.38	0.45
	rpart	0.41	0.39	0.42	0.38	0.37	0.41	0.42	0.40	0.46
	ctree	0.47	0.40	0.41	0.34	0.40	0.45	0.33	0.45	0.37
CSM	OST	0.26	0.29	0.22	0.30	0.32	0.15	0.18	0.30	0.19
	rpart	0.39	0.33	0.22	0.40	0.35	0.30	0.32	0.30	0.25
	ctree	0.28	0.23	0.37	0.44	0.35	0.16	0.12	0.26	0.02
Cycle Power	OST	0.76	0.76	0.74	0.73	0.74	0.76	0.75	0.71	0.63
	rpart	0.71	0.67	0.71	0.71	0.75	0.75	0.76	0.69	0.63
	ctree	0.76	0.76	0.75	0.74	0.75	0.76	0.76	0.69	0.61
Electrical Stability	OST	0.42	0.43	0.41	0.42	0.39	0.40	0.40	0.39	0.34
	rpart	0.35	0.36	0.36	0.35	0.34	0.33	0.35	0.30	0.31
	ctree	0.41	0.41	0.40	0.41	0.41	0.38	0.40	0.37	0.32
Energy Efficiency 1	OST	0.96	0.95	0.93	0.96	0.93	0.97	0.96	0.93	0.93
	rpart	0.82	0.87	0.89	0.87	0.93	0.94	0.95	0.90	0.89
	ctree	0.94	0.92	0.89	0.89	0.89	0.91	0.92	0.90	0.89
Energy Efficiency 2	OST	0.94	0.94	0.94	0.90	0.92	0.96	0.96	0.95	0.95
	rpart	0.83	0.84	0.87	0.90	0.90	0.95	0.94	0.95	0.91
	ctree	0.93	0.93	0.89	0.87	0.89	0.89	0.91	0.89	0.88
Faceboook Comments	OST	0.47	0.53	0.59	0.60	0.60	0.60	0.59	0.57	0.53
	rpart	0.44	0.54	0.59	0.61	0.61	0.59	0.58	0.58	0.52
	ctree	0.44	0.54	0.59	0.60	0.61	0.57	0.55	0.54	0.49
	OST	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.14

Continued on next page

Table 6 – continued from previous page

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Faceboook Metrics	rpart	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.11	0.03
	ctree	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14
Fires	OST	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	rpart	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ctree	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GeoMusic	OST	0.02	0.00	0.03	0.00	0.02	0.07	0.07	0.00	0.10
	rpart	0.09	0.02	0.05	0.05	0.06	0.07	0.07	0.02	0.10
	ctree	0.03	0.03	0.04	0.03	0.02	0.03	0.05	0.03	0.04
Insurance Company Benchmark	OST	0.02	0.04	0.03	0.00	0.00	0.05	0.00	0.00	0.00
	rpart	0.03	0.00	0.05	0.05	0.00	0.00	0.00	0.00	0.00
	ctree	0.04	0.04	0.05	0.00	0.00	0.00	0.00	0.05	0.05
KEGG Directed	OST	0.76	0.76	0.75	0.76	0.77	0.82	0.88	0.92	0.91
	rpart	0.73	0.72	0.73	0.73	0.76	0.78	0.85	0.88	0.88
	ctree	0.73	0.72	0.72	0.73	0.75	0.80	0.87	0.88	0.88
KEGG Undirected	OST	0.86	0.85	0.85	0.86	0.87	0.87	0.88	0.90	0.91
	rpart	0.75	0.79	0.75	0.81	0.82	0.82	0.85	0.85	0.85
	ctree	0.84	0.84	0.85	0.86	0.87	0.88	0.88	0.88	0.88
Kernel Performance	OST	0.81	0.81	0.81	0.79	0.77	0.77	0.70	0.61	0.49
	rpart	0.77	0.74	0.77	0.76	0.71	0.71	0.62	0.53	0.39
	ctree	0.79	0.79	0.79	0.79	0.76	0.72	0.66	0.54	0.39
Las Vegas Strip	OST	0.00	0.00	0.00	0.02	0.08	0.07	0.00	0.00	-0.02
	rpart	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ctree	0.07	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00
Online News Popularity	OST	0.04	0.04	0.04	0.04	0.04	0.05	0.07	0.07	0.08
	rpart	0.04	0.04	0.04	0.04	0.04	0.05	0.06	0.06	0.06
	ctree	0.04	0.04	0.04	0.04	0.05	0.05	0.06	0.07	0.08
Online Video Characteristics	OST	0.70	0.71	0.75	0.76	0.77	0.76	0.75	0.77	0.79
	rpart	0.62	0.62	0.66	0.71	0.72	0.74	0.73	0.75	0.77
	ctree	0.72	0.72	0.74	0.76	0.77	0.78	0.78	0.77	0.78
Optical Interconnection Network	OST	0.30	0.55	0.11	-2.01	0.17	-1.17	0.42	0.58	0.33
	rpart	0.01	0.60	0.23	0.34	0.48	0.35	0.20	0.32	0.30
	ctree	0.00	0.10	0.22	0.27	0.45	0.30	0.34	0.37	0.36
Parkinson Telemonitoring	OST	0.61	0.69	0.55	0.55	0.55	0.56	0.50	0.58	0.69
	rpart	0.48	0.44	0.50	0.53	0.47	0.47	0.42	0.59	0.51
	ctree	0.39	0.39	0.37	0.43	0.45	0.50	0.37	0.45	0.47
PM2.5 - Beijing	OST	0.32	0.32	0.31	0.32	0.35	0.36	0.39	0.41	0.40
	rpart	0.29	0.28	0.29	0.30	0.32	0.29	0.35	0.39	0.40
	ctree	0.29	0.28	0.29	0.30	0.32	0.35	0.38	0.41	0.40
Propulsion Plant	OST	0.65	0.66	0.60	0.58	0.65	0.64	0.67	0.67	0.65
	rpart	0.40	0.43	0.44	0.41	0.43	0.35	0.48	0.46	0.45
	ctree	0.34	0.38	0.38	0.31	0.27	0.27	0.27	0.30	0.00
Protein	OST	0.28	0.28	0.26	0.28	0.31	0.35	0.33	0.33	0.33
	rpart	0.24	0.25	0.25	0.25	0.26	0.29	0.28	0.25	0.23
	ctree	0.25	0.26	0.26	0.27	0.28	0.26	0.27	0.26	0.23
Real Estate 1	OST	0.45	0.46	0.38	0.31	0.45	0.45	0.54	0.54	0.43
	rpart	0.38	0.29	0.32	0.33	0.36	0.45	0.54	0.46	0.51
	ctree	0.40	0.36	0.42	0.37	0.39	0.43	0.49	0.51	0.43
Real Estate 2	OST	0.80	0.77	0.77	0.76	0.80	0.79	0.83	0.84	0.83
	rpart	0.69	0.71	0.71	0.70	0.73	0.76	0.81	0.81	0.81
	ctree	0.74	0.74	0.74	0.75	0.75	0.78	0.78	0.79	0.79
ResidentialBuilding	OST	0.70	0.60	0.60	0.66	0.66	0.72	0.62	0.63	0.50
	rpart	0.52	0.56	0.70	0.75	0.71	0.66	0.74	0.63	0.50
	ctree	0.67	0.69	0.69	0.67	0.69	0.68	0.63	0.49	0.34
Servo	OST	0.78	0.46	0.76	0.66	0.54	0.54	0.52	0.42	-0.12
	rpart	0.47	0.42	0.48	0.68	0.59	0.41	0.30	0.16	0.09
	ctree	0.76	0.42	0.44	0.52	0.52	0.41	0.30	0.17	0.00
StockmarketIstanbul	OST	0.16	0.16	0.08	0.12	0.14	0.05	0.11	0.07	0.10
	rpart	0.10	0.16	0.16	0.22	0.17	0.08	0.11	0.05	0.10
	ctree	0.18	0.15	0.19	0.19	0.15	0.11	0.11	0.10	0.08
Stock Portfolio	OST	0.73	0.51	0.29	0.48	0.13	0.42	0.00	0.65	0.56
	rpart	0.37	0.00	0.26	0.13	-0.28	0.42	0.22	0.65	0.56
	ctree	0.11	0.21	0.04	0.19	0.13	0.19	0.43	0.60	0.68
Student Performance	OST	0.07	0.08	0.07	0.04	0.00	0.00	0.06	0.09	0.05
	rpart	0.07	0.08	0.07	0.04	0.08	0.05	0.06	0.09	0.15
	ctree	0.07	0.08	0.07	0.08	0.08	0.06	0.06	0.09	0.12
WineQuality	OST	0.19	0.17	0.17	0.17	0.17	0.16	0.15	0.16	0.14
	rpart	0.18	0.19	0.18	0.17	0.18	0.16	0.15	0.14	0.13
	ctree	0.21	0.20	0.19	0.18	0.20	0.17	0.16	0.16	0.11

Continued on next page

Table 6 – continued from previous page

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Yacht	OST	0.91	0.67	0.81	0.82	0.89	0.90	0.91	0.81	0.82
	rpart	0.63	0.67	0.76	0.85	0.89	0.92	0.86	0.80	0.82
	ctree	0.87	0.84	0.84	0.75	0.92	0.90	0.78	0.80	0.70

Table 7: Dataset specific Harrell's C Score Results for each level of censoring.

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
3D Spatial Network	OST	0.78	0.79	0.79	0.80	0.80	0.83	0.84	0.85	0.87
	rpart	0.75	0.75	0.76	0.76	0.75	0.77	0.78	0.79	0.82
	ctree	0.75	0.78	0.78	0.79	0.78	0.78	0.80	0.82	0.81
Airfoil Self Noise	OST	0.81	0.82	0.82	0.81	0.84	0.80	0.83	0.81	0.90
	rpart	0.76	0.76	0.73	0.78	0.74	0.80	0.80	0.81	0.87
	ctree	0.72	0.79	0.79	0.77	0.78	0.80	0.78	0.81	0.80
Appliances Energy Prediction	OST	0.73	0.73	0.72	0.74	0.74	0.75	0.76	0.78	0.74
	rpart	0.72	0.73	0.72	0.71	0.73	0.74	0.74	0.75	0.77
	ctree	0.72	0.72	0.71	0.73	0.73	0.74	0.76	0.75	0.76
Automobile	OST	0.50	0.57	0.56	0.50	0.57	0.58	0.50	0.50	0.50
	rpart	0.57	0.71	0.70	0.75	0.57	0.72	0.63	0.65	0.58
	ctree	0.57	0.60	0.57	0.61	0.65	0.66	0.62	0.64	0.58
AutoMPG	OST	0.87	0.87	0.77	0.76	0.84	0.87	0.86	0.86	0.92
	rpart	0.86	0.85	0.86	0.86	0.81	0.87	0.90	0.89	0.89
	ctree	0.87	0.85	0.86	0.85	0.87	0.85	0.86	0.84	0.93
Behavior Urban Traffic	OST	0.68	0.69	0.68	0.63	0.63	0.81	0.65	0.67	0.50
	rpart	0.68	0.69	0.68	0.63	0.73	0.67	0.65	0.67	0.66
	ctree	0.68	0.69	0.68	0.68	0.68	0.67	0.67	0.50	0.50
BikeSharing	OST	0.98	0.98	0.97	0.98	0.97	0.98	0.98	0.99	0.99
	rpart	0.92	0.96	0.96	0.96	0.96	0.97	0.97	0.97	0.99
	ctree	0.98	0.97	0.97	0.98	0.98	0.98	0.97	0.98	0.98
Blog Feedback	OST	0.85	0.84	0.84	0.84	0.84	0.84	0.83	0.84	0.84
	rpart	0.84	0.84	0.84	0.84	0.85	0.85	0.85	0.86	0.85
	ctree	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.86
Buzz in Social Media	OST	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.93
	rpart	0.91	0.91	0.91	0.91	0.91	0.92	0.92	0.92	0.92
	ctree	0.91	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
Cargo2000	OST	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	rpart	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ctree	0.89	0.88	0.93	0.93	0.93	0.96	1.00	1.00	1.00
Communities Crime	OST	0.88	0.89	0.87	0.86	0.87	0.88	0.90	0.91	0.92
	rpart	0.81	0.89	0.89	0.87	0.90	0.90	0.91	0.92	0.95
	ctree	0.92	0.90	0.90	0.91	0.91	0.91	0.90	0.93	0.94
Computer Hardware	OST	0.92	0.91	0.88	0.88	0.80	0.90	0.85	0.83	0.83
	rpart	0.85	0.70	0.86	0.82	0.89	0.78	0.86	0.81	0.90
	ctree	0.90	0.90	0.90	0.80	0.81	0.81	0.88	0.90	0.75
Concrete Slump	OST	0.64	0.64	0.64	0.61	0.50	0.65	0.71	0.50	0.66
	rpart	0.62	0.64	0.64	0.61	0.63	0.65	0.67	0.77	0.72
	ctree	0.50	0.50	0.50	0.61	0.50	0.65	0.66	0.50	0.66
Concrete Strength	OST	0.84	0.85	0.85	0.79	0.80	0.82	0.86	0.84	0.89
	rpart	0.78	0.80	0.82	0.81	0.80	0.83	0.85	0.87	0.88
	ctree	0.83	0.77	0.83	0.81	0.84	0.85	0.81	0.85	0.82
CSM	OST	0.74	0.74	0.70	0.71	0.72	0.67	0.69	0.72	0.73
	rpart	0.77	0.76	0.70	0.79	0.78	0.73	0.75	0.72	0.82
	ctree	0.72	0.70	0.74	0.75	0.75	0.67	0.74	0.77	0.76
Cycle Power	OST	0.91	0.91	0.91	0.91	0.92	0.92	0.90	0.93	0.92
	rpart	0.89	0.87	0.90	0.90	0.92	0.92	0.92	0.92	0.93
	ctree	0.91	0.92	0.91	0.92	0.92	0.92	0.92	0.93	0.93
Electrical Stability	OST	0.80	0.81	0.81	0.82	0.82	0.82	0.83	0.83	0.83
	rpart	0.76	0.78	0.79	0.78	0.78	0.79	0.80	0.78	0.82
	ctree	0.80	0.80	0.80	0.82	0.82	0.83	0.84	0.84	0.85
Energy Efficiency 1	OST	0.99	0.99	0.99	0.99	0.98	0.99	0.99	0.98	0.99
	rpart	0.91	0.96	0.96	0.97	0.98	0.98	0.99	0.98	0.97
	ctree	0.99	0.98	0.98	0.97	0.97	0.97	0.98	0.98	0.97
Energy Efficiency 2	OST	0.99	0.99	0.98	0.96	0.99	0.99	0.99	0.99	0.99
	rpart	0.94	0.94	0.95	0.98	0.97	0.98	0.98	0.98	0.97
	ctree	0.98	0.98	0.97	0.96	0.96	0.96	0.96	0.96	0.97

Continued on next page

Table 7 – continued from previous page

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Faceboook Comments	OST	0.88	0.88	0.87	0.88	0.88	0.88	0.88	0.89	0.89
	rpart	0.88	0.87	0.88	0.88	0.88	0.88	0.88	0.89	0.89
	ctree	0.88	0.88	0.88	0.89	0.89	0.89	0.89	0.89	0.89
Faceboook Metrics	OST	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.70	0.74
	rpart	0.50	0.50	0.50	0.62	0.50	0.50	0.50	0.70	0.76
	ctree	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.74
Fires	OST	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
	rpart	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
	ctree	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
GeoMusic	OST	0.60	0.56	0.56	0.55	0.58	0.62	0.62	0.50	0.65
	rpart	0.65	0.57	0.60	0.58	0.61	0.62	0.62	0.57	0.65
	ctree	0.59	0.59	0.60	0.59	0.58	0.58	0.62	0.58	0.58
Insurance Company Benchmark	OST	0.72	0.73	0.66	0.50	0.50	0.72	0.50	0.50	0.50
	rpart	0.73	0.73	0.73	0.70	0.50	0.50	0.50	0.50	0.50
	ctree	0.70	0.70	0.71	0.50	0.50	0.50	0.50	0.73	0.71
KEGG Directed	OST	0.92	0.93	0.94	0.95	0.96	0.97	0.99	0.99	0.99
	rpart	0.91	0.92	0.93	0.94	0.95	0.96	0.98	0.99	0.99
	ctree	0.91	0.92	0.93	0.94	0.95	0.97	0.99	0.99	0.99
KEGG Undirected	OST	0.94	0.94	0.94	0.97	0.97	0.97	0.98	0.98	0.99
	rpart	0.93	0.95	0.92	0.95	0.96	0.96	0.97	0.98	0.98
	ctree	0.95	0.95	0.96	0.96	0.97	0.97	0.98	0.98	0.98
Kernel Performance	OST	0.84	0.84	0.83	0.84	0.84	0.84	0.84	0.85	0.85
	rpart	0.81	0.79	0.81	0.81	0.80	0.81	0.80	0.81	0.81
	ctree	0.82	0.82	0.82	0.83	0.83	0.82	0.81	0.82	0.81
Las Vegas Strip	OST	0.50	0.50	0.50	0.53	0.65	0.64	0.50	0.50	0.58
	rpart	0.61	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
	ctree	0.64	0.50	0.50	0.53	0.50	0.50	0.50	0.50	0.50
Online News Popularity	OST	0.61	0.61	0.61	0.61	0.61	0.61	0.62	0.62	0.63
	rpart	0.61	0.61	0.61	0.61	0.61	0.61	0.62	0.62	0.63
	ctree	0.62	0.62	0.62	0.63	0.63	0.63	0.63	0.63	0.64
Online Video Characteristics	OST	0.92	0.91	0.92	0.92	0.92	0.91	0.88	0.95	0.95
	rpart	0.90	0.88	0.88	0.90	0.91	0.92	0.93	0.94	0.96
	ctree	0.89	0.91	0.91	0.91	0.92	0.93	0.94	0.95	0.96
Optical Interconnection Network	OST	0.88	0.81	0.80	0.80	0.68	0.85	0.79	0.87	0.77
	rpart	0.75	0.80	0.86	0.79	0.79	0.81	0.81	0.81	0.72
	ctree	0.80	0.84	0.84	0.83	0.80	0.81	0.80	0.80	0.77
Parkinson Telemonitoring	OST	0.80	0.88	0.81	0.83	0.82	0.85	0.85	0.89	0.94
	rpart	0.77	0.77	0.79	0.82	0.81	0.82	0.83	0.89	0.91
	ctree	0.75	0.75	0.76	0.78	0.80	0.83	0.81	0.83	0.88
PM2.5 - Beijing	OST	0.76	0.77	0.78	0.78	0.79	0.80	0.82	0.84	0.86
	rpart	0.75	0.76	0.76	0.77	0.78	0.76	0.80	0.83	0.85
	ctree	0.75	0.76	0.77	0.78	0.79	0.80	0.81	0.83	0.86
Propulsion Plant	OST	0.87	0.88	0.87	0.88	0.86	0.87	0.92	0.85	0.95
	rpart	0.75	0.77	0.79	0.78	0.80	0.74	0.84	0.85	0.88
	ctree	0.73	0.75	0.77	0.74	0.71	0.73	0.74	0.80	0.50
Protein	OST	0.72	0.72	0.72	0.74	0.75	0.77	0.76	0.78	0.81
	rpart	0.69	0.71	0.71	0.72	0.72	0.74	0.74	0.74	0.76
	ctree	0.70	0.71	0.71	0.72	0.72	0.72	0.73	0.75	0.75
Real Estate 1	OST	0.79	0.80	0.80	0.82	0.83	0.84	0.87	0.86	0.84
	rpart	0.79	0.71	0.72	0.73	0.76	0.84	0.87	0.81	0.89
	ctree	0.78	0.77	0.79	0.78	0.81	0.83	0.82	0.82	0.86
Real Estate 2	OST	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.95	0.50
	rpart	0.88	0.89	0.90	0.90	0.90	0.92	0.94	0.95	0.97
	ctree	0.50	0.88	0.90	0.92	0.92	0.93	0.93	0.94	0.96
ResidentialBuilding	OST	0.89	0.88	0.84	0.86	0.86	0.88	0.82	0.87	0.88
	rpart	0.75	0.85	0.91	0.92	0.90	0.86	0.92	0.87	0.88
	ctree	0.88	0.88	0.88	0.87	0.85	0.90	0.85	0.85	0.85
Servo	OST	0.88	0.82	0.88	0.72	0.90	0.85	0.89	0.87	0.85
	rpart	0.69	0.70	0.70	0.80	0.82	0.70	0.70	0.79	0.66
	ctree	0.81	0.70	0.70	0.71	0.71	0.70	0.70	0.69	0.50
StockmarketIstanbul	OST	0.70	0.72	0.64	0.69	0.72	0.66	0.70	0.63	0.65
	rpart	0.62	0.71	0.71	0.73	0.72	0.65	0.66	0.71	0.65
	ctree	0.72	0.71	0.70	0.70	0.71	0.69	0.70	0.71	0.61
Stock Portfolio	OST	0.88	0.81	0.77	0.84	0.74	0.70	0.50	0.88	0.88
	rpart	0.74	0.50	0.75	0.57	0.79	0.70	0.73	0.88	0.88
	ctree	0.59	0.70	0.58	0.75	0.74	0.75	0.78	0.88	0.91
Student Performance	OST	0.59	0.61	0.60	0.55	0.50	0.50	0.58	0.64	0.68
	rpart	0.59	0.61	0.60	0.55	0.61	0.56	0.58	0.64	0.73

Continued on next page

Table 7 – continued from previous page

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
	ctree	0.59	0.61	0.60	0.61	0.61	0.61	0.58	0.64	0.68
WineQuality	OST	0.75	0.72	0.73	0.75	0.73	0.72	0.71	0.73	0.70
	rpart	0.75	0.76	0.75	0.74	0.74	0.73	0.73	0.72	0.71
	ctree	0.78	0.77	0.76	0.76	0.76	0.76	0.75	0.75	0.74
Yacht	OST	0.96	0.91	0.95	0.93	0.94	0.91	0.95	0.94	0.95
	rpart	0.87	0.86	0.88	0.93	0.94	0.95	0.91	0.91	0.95
	ctree	0.94	0.91	0.89	0.92	0.89	0.89	0.83	0.92	0.89

Table 8: Dataset specific Uno's C Score Results for each level of censoring.

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
3D Spatial Network	OST	0.77	0.78	0.76	0.77	0.77	0.80	0.81	0.82	0.84
	rpart	0.72	0.72	0.72	0.72	0.70	0.72	0.73	0.74	0.78
	ctree	0.73	0.77	0.76	0.76	0.75	0.74	0.77	0.78	0.76
Airfoil Self Noise	OST	0.81	0.82	0.80	0.80	0.78	0.77	0.72	0.66	0.79
	rpart	0.71	0.70	0.67	0.71	0.66	0.70	0.70	0.65	0.78
	ctree	0.71	0.75	0.74	0.71	0.67	0.69	0.66	0.68	0.81
Appliances Energy Prediction	OST	0.71	0.70	0.69	0.71	0.71	0.70	0.72	0.74	0.65
	rpart	0.70	0.70	0.69	0.64	0.69	0.69	0.69	0.70	0.70
	ctree	0.71	0.70	0.68	0.69	0.69	0.70	0.71	0.71	0.70
Automobile	OST	0.00	0.18	0.17	0.00	0.18	0.17	0.00	0.00	0.00
	rpart	0.17	0.62	0.63	0.62	0.18	0.54	0.35	0.36	0.18
	ctree	0.18	0.31	0.18	0.25	0.38	0.37	0.26	0.34	0.18
AutoMPG	OST	0.83	0.82	0.83	0.73	0.72	0.77	0.77	0.75	0.85
	rpart	0.78	0.77	0.77	0.77	0.73	0.77	0.80	0.80	0.82
	ctree	0.82	0.81	0.77	0.77	0.78	0.74	0.77	0.62	0.87
Behavior Urban Traffic	OST	0.40	0.42	0.43	0.31	0.29	0.63	0.43	0.41	0.00
	rpart	0.40	0.42	0.43	0.31	0.46	0.43	0.43	0.41	0.40
	ctree	0.40	0.42	0.43	0.40	0.41	0.43	0.47	0.00	0.00
BikeSharing	OST	0.98	0.96	0.95	0.96	0.95	0.96	0.95	0.97	0.99
	rpart	0.85	0.92	0.93	0.93	0.92	0.95	0.95	0.96	0.98
	ctree	0.97	0.97	0.96	0.96	0.96	0.96	0.95	0.96	0.94
Blog Feedback	OST	0.83	0.80	0.80	0.81	0.79	0.78	0.77	0.78	0.78
	rpart	0.82	0.80	0.80	0.80	0.81	0.81	0.79	0.81	0.80
	ctree	0.83	0.83	0.83	0.82	0.82	0.82	0.82	0.81	0.83
Buzz in Social Media	OST	0.92	0.91	0.91	0.90	0.90	0.90	0.90	0.90	0.91
	rpart	0.88	0.88	0.87	0.88	0.88	0.89	0.89	0.89	0.90
	ctree	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.89	0.88
Cargo2000	OST	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	rpart	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00
	ctree	0.77	0.77	0.88	0.88	0.88	0.94	0.95	1.00	1.00
Communities Crime	OST	0.88	0.84	0.77	0.78	0.78	0.77	0.79	0.78	0.88
	rpart	0.79	0.83	0.84	0.78	0.84	0.84	0.85	0.79	0.92
	ctree	0.89	0.87	0.84	0.86	0.86	0.84	0.80	0.83	0.87
Computer Hardware	OST	0.85	0.82	0.74	0.76	0.64	0.79	0.73	0.69	0.63
	rpart	0.73	0.41	0.73	0.62	0.77	0.54	0.78	0.65	0.77
	ctree	0.83	0.82	0.81	0.66	0.68	0.61	0.72	0.77	0.39
Concrete Slump	OST	0.35	0.34	0.35	0.26	0.00	0.31	0.46	0.00	0.33
	rpart	0.33	0.34	0.35	0.26	0.27	0.31	0.37	0.58	0.37
	ctree	0.00	0.00	0.00	0.26	0.00	0.31	0.35	0.00	0.33
Concrete Strength	OST	0.84	0.82	0.80	0.67	0.68	0.69	0.76	0.69	0.73
	rpart	0.74	0.73	0.74	0.73	0.74	0.75	0.76	0.74	0.72
	ctree	0.81	0.78	0.79	0.75	0.78	0.77	0.77	0.72	0.58
CSM	OST	0.57	0.60	0.45	0.44	0.45	0.47	0.42	0.44	0.44
	rpart	0.64	0.62	0.45	0.65	0.58	0.46	0.54	0.44	0.70
	ctree	0.54	0.53	0.63	0.55	0.59	0.43	0.58	0.66	0.62
Cycle Power	OST	0.91	0.90	0.88	0.86	0.87	0.87	0.89	0.91	0.89
	rpart	0.83	0.83	0.83	0.83	0.88	0.87	0.88	0.90	0.90
	ctree	0.90	0.89	0.89	0.88	0.88	0.89	0.88	0.90	0.90
Electrical Stability	OST	0.79	0.79	0.78	0.79	0.78	0.78	0.79	0.79	0.79
	rpart	0.74	0.75	0.75	0.75	0.74	0.74	0.75	0.75	0.78
	ctree	0.79	0.79	0.79	0.79	0.79	0.78	0.79	0.79	0.80
Energy Efficiency 1	OST	0.99	0.97	0.97	0.98	0.97	0.98	0.97	0.97	0.98
	rpart	0.89	0.93	0.94	0.95	0.97	0.97	0.98	0.97	0.97
	ctree	0.98	0.97	0.96	0.95	0.96	0.95	0.97	0.97	0.64

Continued on next page

Table 8 – continued from previous page

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Energy Efficiency 2	OST	0.98	0.98	0.96	0.95	0.97	0.97	0.99	0.99	0.98
	rpart	0.92	0.90	0.93	0.95	0.96	0.98	0.98	0.98	0.97
	ctree	0.98	0.97	0.95	0.95	0.95	0.94	0.96	0.97	0.96
Faceboook Comments	OST	0.87	0.84	0.83	0.84	0.83	0.83	0.82	0.84	0.84
	rpart	0.84	0.83	0.83	0.84	0.83	0.83	0.82	0.85	0.84
	ctree	0.86	0.86	0.86	0.86	0.87	0.87	0.86	0.86	0.86
Faceboook Metrics	OST	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.41	0.47
	rpart	0.00	0.00	0.00	0.35	0.00	0.00	0.00	0.41	0.49
	ctree	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.47
Fires	OST	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	rpart	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ctree	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GeoMusic	OST	0.44	0.31	0.33	0.32	0.38	0.35	0.35	0.00	0.37
	rpart	0.54	0.32	0.43	0.28	0.35	0.35	0.35	0.31	0.37
	ctree	0.46	0.42	0.42	0.38	0.36	0.33	0.36	0.31	0.35
Insurance Company Benchmark	OST	0.57	0.58	0.50	0.00	0.00	0.51	0.00	0.00	0.00
	rpart	0.57	0.58	0.58	0.48	0.00	0.00	0.00	0.00	0.00
	ctree	0.48	0.49	0.49	0.00	0.00	0.00	0.00	0.52	0.48
KEGG Directed	OST	0.92	0.92	0.92	0.92	0.92	0.94	0.98	0.99	0.99
	rpart	0.90	0.90	0.90	0.90	0.91	0.92	0.97	0.99	0.99
	ctree	0.90	0.90	0.90	0.90	0.91	0.94	0.97	0.98	0.98
KEGG Undirected	OST	0.96	0.96	0.96	0.96	0.96	0.96	0.97	0.98	0.99
	rpart	0.92	0.93	0.89	0.93	0.93	0.94	0.95	0.96	0.97
	ctree	0.94	0.94	0.95	0.95	0.96	0.96	0.96	0.97	0.97
Kernel Performance	OST	0.84	0.83	0.83	0.83	0.84	0.84	0.84	0.84	0.83
	rpart	0.80	0.78	0.80	0.80	0.80	0.79	0.79	0.80	0.78
	ctree	0.81	0.81	0.81	0.82	0.82	0.81	0.80	0.81	0.79
Las Vegas Strip	OST	0.00	0.00	0.00	0.09	0.44	0.43	0.00	0.00	0.13
	rpart	0.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ctree	0.41	0.00	0.00	0.09	0.00	0.00	0.00	0.00	0.00
Online News Popularity	OST	0.56	0.56	0.54	0.56	0.54	0.54	0.56	0.56	0.58
	rpart	0.57	0.58	0.57	0.57	0.57	0.57	0.59	0.56	0.53
	ctree	0.59	0.58	0.58	0.59	0.59	0.57	0.58	0.56	0.57
Online Video Characteristics	OST	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.93	0.94
	rpart	0.88	0.87	0.87	0.88	0.88	0.90	0.90	0.92	0.94
	ctree	0.87	0.89	0.90	0.90	0.90	0.91	0.92	0.92	0.94
Optical Interconnection Network	OST	0.87	0.76	0.70	0.74	0.47	0.81	0.69	0.81	0.60
	rpart	0.58	0.73	0.81	0.68	0.69	0.74	0.68	0.66	0.41
	ctree	0.71	0.80	0.78	0.77	0.74	0.73	0.70	0.69	0.67
Parkinson Telemonitoring	OST	0.81	0.87	0.79	0.79	0.80	0.77	0.76	0.75	0.89
	rpart	0.75	0.74	0.77	0.77	0.75	0.77	0.74	0.83	0.85
	ctree	0.73	0.73	0.73	0.75	0.76	0.79	0.67	0.67	0.77
PM2.5 - Beijing	OST	0.75	0.76	0.75	0.75	0.76	0.76	0.77	0.80	0.83
	rpart	0.73	0.73	0.73	0.73	0.73	0.71	0.75	0.78	0.82
	ctree	0.74	0.74	0.74	0.75	0.75	0.76	0.77	0.79	0.82
Propulsion Plant	OST	0.87	0.87	0.84	0.84	0.87	0.87	0.89	0.90	0.91
	rpart	0.70	0.72	0.73	0.71	0.73	0.68	0.76	0.80	0.80
	ctree	0.64	0.70	0.70	0.64	0.56	0.59	0.60	0.73	0.00
Protein	OST	0.69	0.71	0.68	0.70	0.72	0.74	0.74	0.75	0.78
	rpart	0.67	0.68	0.68	0.68	0.68	0.71	0.71	0.70	0.69
	ctree	0.68	0.69	0.68	0.69	0.69	0.69	0.70	0.71	0.70
Real Estate 1	OST	0.70	0.66	0.64	0.71	0.65	0.66	0.68	0.64	0.72
	rpart	0.72	0.45	0.45	0.45	0.46	0.66	0.68	0.58	0.77
	ctree	0.65	0.60	0.67	0.59	0.63	0.67	0.65	0.56	0.75
Real Estate 2	OST	0.91	0.89	0.89	0.88	0.89	0.90	0.92	0.93	0.94
	rpart	0.82	0.83	0.83	0.83	0.84	0.88	0.90	0.92	0.93
	ctree	0.86	0.86	0.88	0.88	0.87	0.89	0.89	0.90	0.91
ResidentialBuilding	OST	0.90	0.75	0.67	0.69	0.68	0.76	0.62	0.72	0.80
	rpart	0.69	0.69	0.82	0.82	0.79	0.72	0.84	0.72	0.80
	ctree	0.78	0.77	0.75	0.73	0.68	0.78	0.73	0.70	0.73
Servo	OST	0.83	0.71	0.83	0.46	0.85	0.74	0.82	0.74	0.76
	rpart	0.40	0.40	0.41	0.76	0.77	0.44	0.43	0.55	0.33
	ctree	0.73	0.40	0.40	0.43	0.43	0.44	0.43	0.42	0.00
StockmarketIstanbul	OST	0.58	0.60	0.40	0.51	0.53	0.46	0.55	0.26	0.31
	rpart	0.46	0.59	0.61	0.63	0.57	0.42	0.41	0.50	0.31
	ctree	0.61	0.57	0.56	0.55	0.57	0.52	0.48	0.49	0.18
Stock Portfolio	OST	0.75	0.62	0.61	0.64	0.40	0.31	0.00	0.74	0.68
	rpart	0.49	0.00	0.48	0.38	0.60	0.31	0.49	0.74	0.68

Continued on next page

Table 8 – continued from previous page

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
	ctree	0.18	0.42	0.16	0.51	0.40	0.44	0.41	0.53	0.85
Student Performance	OST	0.26	0.27	0.26	0.11	0.00	0.00	0.11	0.28	0.51
	rpart	0.26	0.27	0.26	0.11	0.26	0.10	0.11	0.28	0.26
	ctree	0.26	0.27	0.26	0.26	0.26	0.25	0.11	0.28	0.41
WineQuality	OST	0.69	0.60	0.61	0.70	0.61	0.60	0.60	0.64	0.61
	rpart	0.67	0.70	0.69	0.67	0.66	0.63	0.63	0.62	0.60
	ctree	0.73	0.73	0.71	0.71	0.70	0.71	0.70	0.70	0.71
Yacht	OST	0.93	0.81	0.90	0.84	0.87	0.83	0.91	0.81	0.77
	rpart	0.73	0.76	0.80	0.84	0.87	0.90	0.83	0.78	0.77
	ctree	0.89	0.82	0.77	0.81	0.80	0.79	0.66	0.75	0.69

Table 9: Dataset specific Cox Score Results for each level of censoring.

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
3D Spatial Network	OST	0.04	0.04	0.03	0.06	0.02	0.06	0.08	0.06	0.09
	rpart	0.03	0.04	0.04	0.04	0.04	0.05	0.05	0.06	0.08
	ctree	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.07	0.07
Airfoil Self Noise	OST	0.05	0.05	0.07	0.05	0.14	0.06	0.13	0.12	0.11
	rpart	0.07	0.07	0.06	0.08	0.06	0.10	0.10	0.11	0.18
	ctree	0.05	0.09	0.09	0.09	0.09	0.10	0.08	0.11	0.03
Appliances Energy Prediction	OST	0.02	0.02	0.02	0.03	0.03	0.03	0.04	0.05	0.04
	rpart	0.02	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.05
	ctree	0.02	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.04
Automobile	OST	0.00	0.02	0.02	0.00	0.02	0.02	0.00	0.00	0.00
	rpart	0.02	0.07	0.05	0.09	0.02	0.07	0.04	0.05	0.05
	ctree	0.01	0.01	0.01	0.02	0.03	0.04	0.08	0.05	0.05
AutoMPG	OST	0.19	0.19	0.12	0.04	0.16	0.22	0.22	0.21	0.34
	rpart	0.15	0.15	0.15	0.16	0.14	0.21	0.27	0.28	0.31
	ctree	0.21	0.19	0.19	0.18	0.23	0.18	0.20	0.19	0.35
Behavior Urban Traffic	OST	0.11	0.11	0.08	0.08	0.06	0.17	0.08	0.07	0.00
	rpart	0.11	0.11	0.08	0.08	0.13	0.13	0.08	0.07	0.05
	ctree	0.11	0.11	0.08	0.11	0.12	0.08	0.08	0.00	0.00
BikeSharing	OST	0.19	0.04	0.25	0.10	0.09	0.30	0.26	0.03	0.11
	rpart	0.18	0.04	0.04	0.03	0.10	0.04	0.11	0.12	0.17
	ctree	0.24	0.19	0.17	0.03	0.09	0.19	0.18	0.34	0.36
Blog Feedback	OST	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04
	rpart	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04
	ctree	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04
Buzz in Social Media	OST	0.11	0.11	0.11	0.12	0.12	0.13	0.14	0.14	0.15
	rpart	0.10	0.10	0.11	0.11	0.12	0.12	0.13	0.14	0.15
	ctree	0.11	0.11	0.11	0.12	0.12	0.13	0.13	0.14	0.14
Cargo2000	OST	0.15	0.16	0.18	0.19	0.21	0.23	0.24	0.26	0.28
	rpart	0.15	0.16	0.18	0.20	0.21	0.23	0.22	0.26	0.28
	ctree	0.08	0.09	0.10	0.12	0.13	0.16	0.24	0.26	0.28
Communities Crime	OST	0.15	0.14	0.14	0.15	0.16	0.16	0.17	0.19	0.27
	rpart	0.10	0.14	0.15	0.13	0.16	0.17	0.20	0.21	0.29
	ctree	0.19	0.18	0.17	0.16	0.18	0.19	0.19	0.22	0.29
Computer Hardware	OST	0.19	0.18	0.17	0.34	0.23	0.42	0.28	0.07	0.30
	rpart	0.29	0.06	0.30	0.22	0.34	0.24	0.30	0.27	0.38
	ctree	0.34	0.40	0.36	0.22	0.01	0.29	0.37	0.43	0.23
Concrete Slump	OST	0.05	0.04	0.04	0.03	0.00	0.06	0.07	0.00	0.05
	rpart	0.04	0.04	0.04	0.03	0.04	0.06	0.08	0.15	0.12
	ctree	0.00	0.00	0.00	0.03	0.00	0.06	0.06	0.00	0.05
Concrete Strength	OST	0.12	0.03	0.03	0.09	0.10	0.12	0.16	0.14	0.21
	rpart	0.10	0.10	0.11	0.11	0.11	0.14	0.13	0.16	0.20
	ctree	0.13	0.05	0.13	0.12	0.12	0.14	0.05	0.15	0.16
CSM	OST	0.08	0.09	0.07	0.08	0.09	0.05	0.06	0.11	0.12
	rpart	0.10	0.09	0.07	0.12	0.11	0.09	0.11	0.11	0.18
	ctree	0.06	0.06	0.09	0.11	0.10	0.05	0.07	0.15	0.13
Cycle Power	OST	0.13	0.13	0.17	0.17	0.18	0.19	0.12	0.21	0.13
	rpart	0.14	0.13	0.15	0.16	0.18	0.19	0.20	0.20	0.21
	ctree	0.16	0.17	0.17	0.17	0.18	0.19	0.20	0.20	0.21
Electrical Stability	OST	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.10	0.10
	rpart	0.05	0.06	0.06	0.06	0.06	0.06	0.08	0.07	0.09
	ctree	0.06	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.11

Continued on next page

Table 9 – continued from previous page

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Energy Efficiency 1	OST	0.31	0.21	0.27	0.32	0.30	0.31	0.37	0.41	0.64
	rpart	0.18	0.21	0.23	0.24	0.27	0.29	0.34	0.39	0.56
	ctree	0.28	0.25	0.24	0.24	0.25	0.27	0.33	0.40	0.56
Energy Efficiency 2	OST	0.14	0.07	0.25	0.11	0.27	0.16	0.32	0.42	0.64
	rpart	0.06	0.01	0.16	0.05	0.00	0.12	0.10	0.16	0.53
	ctree	0.14	0.09	0.00	0.17	0.22	0.19	0.26	0.34	0.49
Faceboook Comments	OST	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	rpart	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	ctree	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
Faceboook Metrics	OST	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.08
	rpart	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.05	0.08
	ctree	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08
Fires	OST	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	rpart	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ctree	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GeoMusic	OST	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.00	0.03
	rpart	0.02	0.01	0.01	0.01	0.01	0.02	0.02	0.01	0.03
	ctree	0.01	0.01	0.02	0.01	0.01	0.01	0.02	0.01	0.02
Insurance Company Benchmark	OST	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	rpart	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ctree	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
KEGG Directed	OST	0.09	0.10	0.10	0.10	0.11	0.12	0.13	0.14	0.14
	rpart	0.09	0.09	0.09	0.10	0.11	0.10	0.12	0.13	0.14
	ctree	0.09	0.09	0.09	0.10	0.10	0.11	0.13	0.13	0.13
KEGG Undirected	OST	0.05	0.05	0.04	0.17	0.18	0.18	0.20	0.21	0.22
	rpart	0.13	0.14	0.13	0.15	0.16	0.17	0.18	0.19	0.21
	ctree	0.15	0.15	0.16	0.16	0.17	0.18	0.19	0.20	0.21
Kernel Performance	OST	0.09	0.09	0.08	0.09	0.09	0.09	0.09	0.09	0.09
	rpart	0.07	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.07
	ctree	0.07	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.07
Las Vegas Strip	OST	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.00	0.00
	rpart	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ctree	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Online News Popularity	OST	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	rpart	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	ctree	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Online Video Characteristics	OST	0.05	0.14	0.15	0.15	0.16	0.12	0.08	0.02	-0.35
	rpart	0.12	0.11	0.10	0.13	0.14	0.15	0.17	0.19	0.22
	ctree	0.11	0.13	0.13	0.14	0.15	0.16	0.17	0.19	0.21
Optical Interconnection Network	OST	0.09	0.15	0.11	0.15	0.07	0.10	0.10	0.25	0.06
	rpart	0.05	0.11	0.16	0.12	0.12	0.12	0.12	0.14	0.10
	ctree	0.08	0.14	0.12	0.10	0.10	0.11	0.11	0.14	0.12
Parkinson Telemonitoring	OST	0.07	0.15	0.11	0.11	0.07	0.14	0.14	0.18	0.26
	rpart	0.07	0.06	0.08	0.10	0.09	0.10	0.10	0.17	0.18
	ctree	0.06	0.05	0.05	0.07	0.08	0.10	0.09	0.12	0.15
PM2.5 - Beijing	OST	0.04	0.04	0.04	0.05	0.05	0.06	0.07	0.08	0.02
	rpart	0.04	0.04	0.04	0.04	0.05	0.04	0.06	0.07	0.09
	ctree	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.08	0.09
Propulsion Plant	OST	0.05	0.06	0.11	0.11	0.11	0.08	0.18	0.10	0.23
	rpart	0.05	0.06	0.06	0.06	0.08	0.06	0.11	0.12	0.14
	ctree	0.03	0.05	0.06	0.04	0.04	0.05	0.05	0.08	0.00
Protein	OST	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.07
	rpart	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.04	0.05
	ctree	0.02	0.02	0.02	0.03	0.03	0.03	0.04	0.04	0.05
Real Estate 1	OST	0.13	0.14	0.12	0.16	0.16	0.18	0.22	0.25	0.24
	rpart	0.13	0.05	0.08	0.09	0.11	0.18	0.22	0.19	0.32
	ctree	0.10	0.07	0.12	0.10	0.14	0.18	0.17	0.20	0.25
Real Estate 2	OST	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.00
	rpart	0.04	0.12	0.04	0.00	0.09	0.05	0.05	0.24	0.28
	ctree	0.00	0.11	0.04	0.05	0.05	0.04	0.05	0.06	0.06
ResidentialBuilding	OST	0.09	0.23	0.20	0.23	0.22	0.28	0.23	0.30	0.33
	rpart	0.14	0.22	0.29	0.29	0.27	0.24	0.32	0.30	0.33
	ctree	0.23	0.23	0.25	0.24	0.24	0.28	0.25	0.21	0.21
Servo	OST	0.31	0.21	0.31	0.21	0.21	0.27	0.34	0.30	0.21
	rpart	0.13	0.13	0.13	0.27	0.21	0.14	0.15	0.16	0.11
	ctree	0.24	0.13	0.13	0.16	0.17	0.14	0.15	0.14	0.00
StockmarketIstanbul	OST	0.05	0.05	0.03	0.04	0.05	0.04	0.06	0.03	0.05
	rpart	0.03	0.05	0.05	0.06	0.05	0.03	0.04	0.05	0.05

Continued on next page

Table 9 – continued from previous page

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
	ctree	0.06	0.05	0.06	0.06	0.05	0.05	0.06	0.05	0.03
Stock Portfolio	OST	0.48	0.35	0.34	0.40	0.23	0.24	0.00	0.55	0.47
	rpart	0.24	0.00	0.25	0.06	0.24	0.24	0.26	0.55	0.47
	ctree	0.10	0.21	0.09	0.26	0.23	0.26	0.25	0.49	0.57
Student Performance	OST	0.01	0.02	0.01	0.01	0.00	0.00	0.02	0.03	0.06
	rpart	0.01	0.02	0.01	0.01	0.01	0.01	0.02	0.03	0.03
	ctree	0.01	0.02	0.01	0.01	0.01	0.02	0.02	0.03	0.06
WineQuality	OST	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03
	rpart	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.04
	ctree	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.04	0.04
Yacht	OST	0.56	0.23	0.28	0.24	0.46	0.42	0.31	0.31	0.51
	rpart	0.31	0.31	0.33	0.45	0.46	0.50	0.42	0.40	0.51
	ctree	0.49	0.42	0.38	0.44	0.40	0.39	0.29	0.42	0.37

Table 10: Dataset specific Brier Point Ratio Results for each level of censoring.

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
3D Spatial Network	OST	0.44	0.46	0.45	0.47	0.48	0.53	0.51	0.49	0.45
	rpart	0.36	0.36	0.37	0.39	0.33	0.35	0.34	0.35	0.33
	ctree	0.38	0.46	0.43	0.46	0.42	0.39	0.41	0.41	0.33
Airfoil Self Noise	OST	0.52	0.52	0.54	0.61	0.58	0.53	0.48	0.46	0.52
	rpart	0.37	0.41	0.38	0.52	0.33	0.47	0.47	0.44	0.41
	ctree	0.41	0.46	0.52	0.49	0.45	0.47	0.44	0.42	0.51
Appliances Energy Prediction	OST	0.20	0.21	0.16	0.18	0.23	0.05	0.06	0.22	-0.06
	rpart	0.18	0.20	0.18	0.18	0.24	0.02	0.03	0.19	-0.04
	ctree	0.18	0.16	0.16	0.16	0.22	0.04	0.04	0.17	-0.07
Automobile	OST	0.00	0.02	-0.06	0.00	-0.05	0.07	0.00	0.00	0.00
	rpart	0.08	0.29	0.33	0.19	-0.05	0.14	0.10	-0.04	-0.01
	ctree	0.15	0.13	0.10	0.13	0.13	0.16	0.10	0.08	-0.01
AutoMPG	OST	0.66	0.62	0.66	0.65	0.65	0.60	0.51	0.38	0.54
	rpart	0.73	0.62	0.69	0.69	0.57	0.63	0.58	0.51	0.40
	ctree	0.71	0.67	0.63	0.63	0.65	0.49	0.50	0.39	0.59
Behavior Urban Traffic	OST	0.25	0.29	0.22	0.06	0.08	0.10	0.06	0.10	0.00
	rpart	0.25	0.29	0.22	0.06	0.23	0.14	0.06	0.10	0.09
	ctree	0.25	0.29	0.22	0.09	0.11	0.17	0.10	0.00	0.00
BikeSharing	OST	0.97	0.93	0.94	0.94	0.93	0.94	0.93	0.93	0.94
	rpart	0.83	0.89	0.91	0.91	0.92	0.92	0.93	0.92	0.94
	ctree	0.95	0.94	0.94	0.93	0.95	0.95	0.94	0.95	0.95
Blog Feedback	OST	-0.25	-0.24	-0.23	-0.24	-0.24	-0.24	1.00	1.00	1.00
	rpart	-0.25	-0.24	-0.23	-0.23	-0.23	-0.24	1.00	1.00	1.00
	ctree	-0.25	-0.25	-0.24	-0.23	-0.24	-0.24	1.00	1.00	1.00
Buzz in Social Media	OST	0.81	0.80	0.78	0.77	0.73	0.63	0.28	1.00	1.00
	rpart	0.80	0.78	0.76	0.75	0.71	0.62	0.26	1.00	1.00
	ctree	0.81	0.80	0.78	0.76	0.72	0.60	0.24	1.00	1.00
Cargo2000	OST	0.02	0.02	0.03	0.05	1.00	1.00	1.00	0.99	-2.10
	rpart	0.02	0.03	0.03	0.04	1.00	1.00	0.99	0.99	-2.07
	ctree	0.03	0.03	0.04	0.04	0.65	0.81	1.00	1.00	-2.10
Communities Crime	OST	0.76	0.70	0.59	0.62	0.65	0.64	0.70	0.72	0.75
	rpart	0.58	0.70	0.69	0.69	0.71	0.76	0.77	0.71	0.69
	ctree	0.76	0.75	0.72	0.76	0.77	0.69	0.74	0.76	0.77
Computer Hardware	OST	0.88	0.83	0.77	0.58	0.51	0.63	0.55	0.91	0.90
	rpart	0.87	0.42	0.76	0.69	0.72	0.50	0.51	0.91	0.73
	ctree	0.84	0.87	0.75	0.34	0.28	0.49	0.51	0.77	0.69
Concrete Slump	OST	0.14	0.15	0.16	0.11	0.00	0.09	0.18	0.00	0.17
	rpart	0.14	0.15	0.16	0.11	0.11	0.09	0.02	0.27	0.11
	ctree	0.00	0.00	0.00	0.11	0.00	0.09	0.07	0.00	0.17
Concrete Strength	OST	0.54	0.51	0.56	0.37	0.41	0.42	0.52	0.49	0.64
	rpart	0.39	0.43	0.54	0.39	0.47	0.44	0.48	0.54	0.58
	ctree	0.54	0.48	0.54	0.43	0.54	0.49	0.48	0.54	0.58
CSM	OST	0.44	0.34	0.46	0.44	0.42	0.12	0.22	0.35	0.27
	rpart	0.45	0.49	0.46	0.48	0.45	0.43	0.35	0.35	0.32
	ctree	0.28	0.20	0.22	0.48	0.31	0.23	0.15	0.23	0.22
Cycle Power	OST	0.84	0.81	0.78	0.75	0.73	0.74	0.73	0.71	0.63
	rpart	0.78	0.72	0.73	0.72	0.75	0.74	0.74	0.70	0.63
	ctree	0.84	0.82	0.76	0.74	0.74	0.75	0.73	0.70	0.63

Continued on next page

Table 10 – continued from previous page

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Electrical Stability	OST	0.50	0.50	0.46	0.49	0.44	0.42	0.42	0.38	0.32
	rpart	0.41	0.43	0.44	0.37	0.39	0.34	0.37	0.29	0.28
	ctree	0.46	0.48	0.44	0.47	0.45	0.45	0.44	0.40	0.34
Energy Efficiency 1	OST	-0.91	-0.93	-0.91	-0.89	-0.65	0.96	0.81	0.75	0.79
	rpart	-0.51	-0.50	-0.92	-0.90	-0.88	0.95	0.88	0.75	0.74
	ctree	-0.91	-0.92	-0.91	-0.89	-0.88	0.94	0.85	0.75	0.75
Energy Efficiency 2	OST	-1.06	-1.01	-1.06	-1.00	-0.99	0.99	1.00	0.97	0.92
	rpart	-0.44	-0.41	-1.06	-1.00	-1.00	1.00	0.95	0.97	0.88
	ctree	-1.06	-1.01	-1.05	-1.00	-1.00	0.97	0.94	0.91	0.86
Faceboook Comments	OST	-0.44	-0.41	-0.41	-0.41	-0.40	-0.40	-0.40	1.00	1.00
	rpart	-0.42	-0.41	-0.42	-0.40	-0.40	-0.40	-0.39	1.00	1.00
	ctree	-0.43	-0.42	-0.43	-0.43	-0.42	-0.43	-0.43	1.00	1.00
Faceboook Metrics	OST	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.24	0.19
	rpart	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.24	0.17
	ctree	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.19
Fires	OST	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
	rpart	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
	ctree	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
GeoMusic	OST	0.02	0.00	0.02	0.00	-0.00	0.07	0.05	0.00	0.06
	rpart	0.11	0.02	0.06	0.09	0.07	0.07	0.05	0.00	0.06
	ctree	0.04	0.02	0.03	0.05	0.01	0.03	0.05	0.01	-0.01
Insurance Company Benchmark	OST	-0.01	-0.00	-0.01	0.00	0.00	-0.00	1.00	1.00	1.00
	rpart	-0.01	-0.01	-0.01	-0.00	0.00	0.00	1.00	1.00	1.00
	ctree	-0.00	-0.00	-0.00	0.00	0.00	0.00	1.00	1.00	1.00
KEGG Directed	OST	0.83	0.85	0.84	0.87	0.86	0.88	0.91	-2.77	-2.71
	rpart	0.80	0.83	0.85	0.85	0.86	0.85	0.87	-2.66	-2.65
	ctree	0.79	0.81	0.85	0.86	0.87	0.88	0.90	-2.68	-2.63
KEGG Undirected	OST	0.89	0.86	0.87	0.87	0.82	0.90	0.87	0.89	0.87
	rpart	0.80	0.83	0.77	0.80	0.80	0.83	0.81	0.84	0.77
	ctree	0.84	0.83	0.85	0.85	0.82	0.91	0.88	0.87	0.81
Kernel Performance	OST	0.62	0.60	0.58	0.56	0.55	0.52	0.48	0.45	0.37
	rpart	0.53	0.48	0.49	0.47	0.45	0.42	0.39	0.38	0.29
	ctree	0.56	0.55	0.53	0.53	0.52	0.46	0.40	0.40	0.30
Las Vegas Strip	OST	0.00	0.00	0.00	-0.00	0.07	0.06	0.00	0.00	0.02
	rpart	-0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ctree	0.04	0.00	0.00	-0.00	0.00	0.00	0.00	0.00	0.00
Online News Popularity	OST	0.08	0.08	0.08	0.08	0.06	0.03	0.03	-0.00	1.00
	rpart	0.08	0.09	0.08	0.07	0.06	0.05	0.02	-0.00	1.00
	ctree	0.09	0.09	0.10	0.09	0.09	0.06	0.03	-0.00	1.00
Online Video Characteristics	OST	0.77	0.77	0.78	0.76	0.75	0.73	0.75	0.77	0.80
	rpart	0.73	0.71	0.71	0.71	0.70	0.71	0.75	0.76	0.79
	ctree	0.69	0.75	0.76	0.75	0.74	0.76	0.78	0.79	0.78
Optical Interconnection Network	OST	0.73	0.56	0.60	0.61	0.44	0.34	1.00	1.00	1.00
	rpart	0.61	0.56	0.77	0.57	0.52	0.07	1.00	1.00	1.00
	ctree	0.74	0.72	0.67	0.50	0.37	0.24	1.00	1.00	1.00
Parkinson Telemonitoring	OST	0.60	0.70	0.53	0.55	0.62	0.69	0.72	0.79	0.79
	rpart	0.52	0.43	0.49	0.52	0.54	0.62	0.52	0.76	0.55
	ctree	0.39	0.35	0.36	0.41	0.48	0.57	0.53	0.62	0.61
PM2.5 - Beijing	OST	0.36	0.40	0.41	0.42	0.43	0.45	0.47	0.47	0.42
	rpart	0.35	0.37	0.38	0.40	0.41	0.36	0.42	0.45	0.42
	ctree	0.36	0.38	0.39	0.41	0.43	0.45	0.45	0.47	0.42
Propulsion Plant	OST	0.71	0.75	0.68	0.69	0.68	0.68	0.69	0.71	0.71
	rpart	0.47	0.48	0.52	0.45	0.44	0.34	0.50	0.47	0.49
	ctree	0.42	0.50	0.50	0.36	0.27	0.27	0.25	0.32	0.00
Protein	OST	0.32	0.33	0.29	0.32	0.33	0.34	0.30	0.32	0.36
	rpart	0.28	0.30	0.30	0.29	0.27	0.29	0.26	0.24	0.24
	ctree	0.30	0.29	0.28	0.30	0.28	0.26	0.25	0.26	0.24
Real Estate 1	OST	0.54	0.63	0.59	0.62	0.69	0.60	0.62	0.55	0.48
	rpart	0.39	0.46	0.61	0.65	0.67	0.60	0.62	0.48	0.60
	ctree	0.53	0.49	0.67	0.65	0.62	0.61	0.57	0.50	0.42
Real Estate 2	OST	0.84	0.84	0.81	0.79	0.82	0.79	0.83	0.82	0.92
	rpart	0.72	0.75	0.71	0.71	0.72	0.74	0.81	0.79	0.91
	ctree	0.75	0.76	0.76	0.76	0.76	0.77	0.78	0.75	0.89
ResidentialBuilding	OST	0.80	0.69	0.63	0.74	0.63	0.71	0.52	0.52	0.44
	rpart	0.56	0.86	0.83	0.73	0.78	0.64	0.73	0.52	0.44
	ctree	0.80	0.87	0.79	0.80	0.70	0.74	0.66	0.46	0.34
Servo	OST	0.62	0.61	0.59	0.23	0.57	0.74	0.47	0.49	0.05
	rpart	0.31	0.31	0.30	0.42	0.49	0.18	0.17	0.32	0.08

Continued on next page

Table 10 – continued from previous page

Dataset	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
	ctree	0.60	0.31	0.30	0.23	0.21	0.18	0.17	0.13	0.00
StockmarketIstanbul	OST	0.26	0.29	0.20	0.26	0.29	0.13	0.08	0.08	0.01
	rpart	0.13	0.26	0.21	0.28	0.27	0.16	0.08	0.08	0.01
	ctree	0.25	0.25	0.30	0.27	0.23	0.18	0.15	0.18	0.02
Stock Portfolio	OST	0.71	0.42	0.86	0.49	0.59	0.48	0.00	0.48	0.39
	rpart	0.60	0.00	0.86	0.01	0.59	0.48	0.16	0.48	0.39
	ctree	0.02	0.12	-0.16	0.37	0.59	0.44	0.65	0.66	0.41
Student Performance	OST	0.08	0.12	0.11	0.09	0.00	0.00	0.08	0.09	0.04
	rpart	0.08	0.12	0.11	0.09	0.12	0.06	0.08	0.09	0.24
	ctree	0.08	0.12	0.11	0.10	0.12	0.11	0.08	0.09	0.07
WineQuality	OST	0.10	0.10	0.22	0.24	-0.25	-0.26	-0.25	-0.27	0.01
	rpart	0.10	0.09	0.22	0.22	-0.28	-0.25	-0.25	-0.27	0.02
	ctree	0.11	0.11	0.24	0.25	-0.34	-0.35	-0.34	-0.36	0.01
Yacht	OST	0.91	0.78	0.80	0.83	0.76	0.69	0.87	0.80	0.80
	rpart	0.90	0.62	0.62	0.82	0.76	0.80	0.73	0.80	0.80
	ctree	0.94	0.98	0.80	0.85	0.86	0.94	0.56	0.85	0.59

B.2 Aggregate Results for different Censoring Levels

		Integrated Brier Score		Harrel's C Score		Uno's C Score		Cox Partial Likelihood		Brier Point Ratio	
% Censoring	Method	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
	ctree	0.43	0.047	0.769	0.022	0.665	0.041	0.097	0.016	0.325	0.068
0	OST	**0.48**	0.049	**0.79**	0.023	**0.704**	0.043	**0.1**	0.018	**0.366**	0.07
	rpart	0.404	0.042	0.767	0.018	0.654	0.035	0.084	0.011	0.332	0.055
	ctree	0.422	0.046	0.776	0.021	0.656	0.042	**0.096**	0.015	0.323	0.067
0.1	OST	**0.464**	0.046	**0.788**	0.022	**0.687**	0.041	0.087	0.012	**0.352**	0.067
	rpart	0.403	0.043	0.767	0.02	0.636	0.04	0.075	0.01	0.323	0.053
	ctree	0.423	0.046	0.776	0.021	0.647	0.043	0.091	0.014	0.313	0.067
0.2	OST	**0.443**	0.049	0.778	0.022	**0.669**	0.041	**0.097**	0.014	**0.349**	0.068
	rpart	0.421	0.045	**0.781**	0.019	0.66	0.038	0.093	0.013	0.339	0.066
	ctree	0.43	0.045	0.779	0.021	0.644	0.042	0.097	0.013	0.316	0.064
0.3	OST	0.399	0.074	0.772	0.023	0.642	0.045	**0.1**	0.015	**0.331**	0.065
	rpart	**0.434**	0.045	**0.784**	0.02	**0.659**	0.038	0.094	0.014	0.315	0.063
	ctree	0.434	0.046	0.778	0.022	0.635	0.044	0.097	0.013	0.307	0.065
0.4	OST	**0.442**	0.048	0.774	0.023	0.639	0.045	**0.105**	0.015	**0.346**	0.066
	rpart	0.429	0.049	**0.784**	0.021	**0.646**	0.042	0.104	0.015	0.332	0.066
	ctree	0.429	0.047	0.784	0.021	0.641	0.042	0.109	0.014	0.387	0.055
0.5	OST	0.428	0.061	**0.793**	0.022	**0.674**	0.041	**0.12**	0.017	**0.418**	0.056
	rpart	**0.443**	0.046	0.782	0.022	0.635	0.043	0.108	0.015	0.384	0.055
	ctree	0.422	0.047	0.787	0.021	0.637	0.043	0.114	0.014	0.434	0.057
0.6	OST	**0.455**	0.049	0.781	0.024	0.645	0.048	**0.118**	0.016	**0.456**	0.058
	rpart	0.439	0.047	**0.791**	0.022	**0.648**	0.043	**0.118**	0.015	0.436	0.057
	ctree	0.429	0.047	0.797	0.022	0.642	0.044	0.136	0.02	0.404	0.091
0.7	OST	**0.461**	0.048	0.803	0.024	0.65	0.048	0.132	0.02	**0.438**	0.092
	rpart	0.439	0.046	**0.807**	0.021	**0.668**	0.04	**0.138**	0.018	0.425	0.09
	ctree	0.39	0.049	0.792	0.023	0.629	0.046	0.145	0.023	0.357	0.106
0.8	OST	**0.435**	0.05	0.806	0.024	**0.675**	0.045	0.146	0.027	**0.402**	0.108
	rpart	0.424	0.047	**0.816**	0.021	0.667	0.043	**0.168**	0.023	0.378	0.105

Table 11: Average scores for OCT, **rpart**, **ctree** models for real world datasets for each level of censoring.