On the use of machine learning methods for the treatment of unit nonresponse in surveys

Khaled LARBI[(a)], John TSANG[(b)], David HAZIZA[(b)] and Mehdi DAGDOUG[(c)]

(a) INSEE Paris, France

(b) University of Ottawa, Department of Mathematics and Statistics, Ottawa, Canada

(c) McGill University, Department of Mathematics and Statistics, Montreal, Canada

Abstract

In recent years, there has been a significant interest in machine learning in national statistical offices. Thanks to their flexibility, these methods may prove useful at the nonresponse treatment stage. In this article, we conduct an empirical investigation in order to compare several machine learning procedures in terms of bias and efficiency. In addition to the classical machine learning procedure, we assess the performance of ensemble approaches that make use of different machine learning procedures to produce a set of weights adjusted for nonresponse.

Key words: Aggregation procedure; Efficiency; Nonresponse bias; Propensity score estimation.

Introduction

In the last two decades, response rates have been steadily declining in medium to large-scale surveys conducted by National Statistical Offices, raising growing concerns about the potential nonresponse bias. Unit nonresponse, where no information is available for any of the survey variables, is typically addressed through some form of weight adjustment procedure. The underlying principle behind weight adjustment is to inflate the weight of respondents in such a way that they effectively represent the nonrespondents. The inflation factor is defined as the inverse of the estimated response probability.

The treatment of unit nonresponse starts with formulating a nonresponse model, describing the relationship between the response indicators (equal to 1 for respondents and equal to 0 for nonrespondents) and a vector of fully observed variables, which are those that are available for both the respondents and the nonrespondents. Determining a suitable model also consists of selecting of a vector of explanatory variables that are both predictive of the response indicators and related to the survey variables; see Haziza and Beaumont (2017) for a discussion.

In recent years, there has been a growing interest within National Statistical Offices in the application of machine learning techniques in the context of weighting for unit nonresponse. Some reasons for the popularity of machine learning procedures include: (i) Machine learning models can automatically learn and adapt from data, reducing the need for manual intervention. (ii) They can capture complex, non-linear relationships between variables that may be difficult to model using traditional parametric procedures such as logistic regression. (iii) A number of machine learning algorithms are known for their excellent predictive performance. However, one should exercise some caution when machine learning procedures are used for the treatment of unit nonresponse as the survey statistician faces an estimation problem rather than a prediction problem. If the aim lies in estimating a finite population total/mean, the most predictive nonresponse model may not necessarily yield the best estimator in terms of mean square error. Indeed, more accurate predictions may lead to highly dispersed weights, potentially resulting in points estimators exhibiting a large variance. This phenomenon is further discussed in Section 2. This is somewhat different from what is encountered in the context of imputation for item nonresponse, whereby highly predictive procedures are expected to produce accurate estimates of population totals/means.

In this article, we investigate the use of machine learning procedures for estimating the response probabilities. We illustrate through an empirical study that a highly predictive procedure may lead to poor estimates in terms of mean square error; see Section 2. In Section 3, we conduct an extensive simulation study to assess the performance of adjusted estimators in terms of bias and efficiency. Other empirical investigations on the use of machine learning in the context of unit nonresponse for survey data can be found in Phipps and Toth (2012), Lohr et al. (2015), Gelein (2017), and Kern et al. (2019). In Section 4, we describe a number of aggregation procedures, whereby the predictions produced by multiple machine learning procedures are combined to construct a suitable aggregate. The performance of aggregation procedures is assessed in terms of bias and efficiency. Finally, we make some remarks in Section 5.

1 Preliminaries

Consider a finite population $\mathscr{U}$ of size N; i.e., $\mathscr{U} = \{1, \ldots, k, \ldots, N\}$. The aim is to estimate the population total of a survey variable y, $t_y := \sum_{k \in \mathscr{U}} y_k$. To that end, we select a sample $\mathscr{S}$, of size n, according to a sampling design, $P(\mathscr{S} \mid \mathbf{Z})$, with first-order inclusion probabilities $\pi_k, k \in \mathscr{U}$, where $\mathbf{Z}$ denotes the matrix of design information. In the absence of nonsampling errors, a design-unbiased estimator of t_y is the well-known Horvitz-Thompson estimator

$$\widehat{t}_{y,\pi} = \sum_{k \in \mathscr{S}} d_k y_k, \tag{1}$$

where $d_k = 1/\pi_k$ denotes the design (basic) weight attached to unit k.

In the presence of unit nonresponse, the survey variable y is collected for a subset $\mathscr{S}_r \subset \mathscr{S}$. Let R_k be a response indicator attached to unit k such that $R_k = 1$ if unit k responds to the survey, and $R_k = 0$, otherwise. Let $p_k \equiv P(R_k = 1 \mid y_k, \mathbf{x}_k, k \in \mathscr{S})$ denote the response probability associated with unit k, where $\mathbf{x}_k$ denotes a vector of fully observed variable attached to unit k. We make the following assumptions: (i) The response indicators R_k are mutually independent, $k = 1, \ldots, N$; (ii) The response indicators R_k are independent of the sample selection indicators I_k, where $I_k = 1$ if $k \in \mathscr{S}$, and $I_k = 0$, otherwise. This assumption implies that the response probability of a unit is essentially determined by fixed respondent characteristics. In the context of adaptative collection designs (Groves and Heeringa, 2006), this assumption may be violated. (iii) The positivity assumption is satisfied; i.e., $\pi_k > 0$ for all k and $p_k > 0$ for all k.

An unadjusted estimator of t_y is given by

$$\widehat{t}_{y,un} = N\frac{\sum_{k\in\mathcal{S}} d_k R_k y_k}{\sum_{k\in\mathcal{S}} d_k R_k} \equiv N\widehat{\overline{Y}}_r. \tag{2}$$

The nonresponse error of $\widehat{t}_{y,un}$, defined as the difference between the unadjusted estimator and the full sample estimator, can be expressed as

$$\widehat{t}_{y,un} - \widehat{t}_{y,\pi} = N\left\{\frac{\widehat{N}_m}{\widehat{N}_\pi}\left(\widehat{\overline{Y}}_r - \widehat{\overline{Y}}_m\right)\right\}, \tag{3}$$

where $\widehat{N}_m = \sum_{k\in\mathcal{S}} d_k(1-R_k)$, $\widehat{N}_\pi = \sum_{k\in\mathcal{S}} d_k$, and

$$\widehat{\overline{Y}}_m = \frac{\sum_{k\in\mathcal{S}} d_k(1-R_k)y_k}{\sum_{k\in\mathcal{S}} d_k(1-R_k)}$$

denotes the (unestimable) mean of the nonrespondents. The term $\widehat{N}_m/\widehat{N}_\pi$ in (3) can be viewed as an estimate of the nonresponse rate. Alternatively, the population size N in (2) may be replaced by the estimated population size $\widehat{N}_\pi$. When the data are Missing Completely At Random (MCAR), we have $\mathbb{E}\left(\widehat{\overline{Y}}_r - \widehat{\overline{Y}}_m\right) \approx 0$. It follows that $\widehat{t}_{y,un}$ is virtually unbiased for t_y. In contrast, the bias may be significant if the nonresponse rate is high and/or the behaviour of the respondents differ systematically from that of the nonrespondents in terms of the y-variable.

Turning to adjusted estimators, assuming that the response probabilities p_k are known, an unbiased estimator of t_y is the so-called double expansion estimator (Särndal et al., 1992):

$$\widehat{t}_{y,DE} = \sum_{k\in\mathcal{S}} \frac{d_k}{p_k} R_k y_k. \tag{4}$$

In practice, the p_k's are unknown and are replaced with estimated response probabilities $\widehat{p}_k$. More specifically, we start by postulating the following nonresponse model:

$$\mathbb{E}(R_k \mid y_k, \mathbf{x}_k) = p(\mathbf{x}_k), \tag{5}$$

where $p(\cdot)$ is an unknown function. In the case of a parametric procedure (e.g., logistic regression), the function $p(\cdot)$ is predetermined, whereas it is left unspecified in the case of nonparametric and machine learning procedures.

An adjusted estimator of t_y is the Propensity Score-Adjusted (PSA) estimator given by

$$\widehat{t}_{y,PSA} = \sum_{k\in\mathscr{S}} \frac{d_k}{\widehat{p}(\mathbf{x}_k)} R_k y_k, \tag{6}$$

where $\widehat{p}(\mathbf{x}_k)$ denotes the fitted value attached unit to $k \in \mathscr{S}_r$. The weights adjusted for nonresponse are denoted by $w_k^* = d_k/\widehat{p}(\mathbf{x}_k)$, $k \in \mathscr{S}_r$. The nonresponse error of $\widehat{t}_{y,PSA}$ can be expressed as

$$\widehat{t}_{y,PSA} - \widehat{t}_{y,\pi} = (\widehat{t}_{y,DE} - \widehat{t}_{y,\pi}) - \sum_{k\in\mathscr{S}} \frac{d_k}{\widehat{p}(\mathbf{x}_k)} R_k y_k \left(\frac{\widehat{p}(\mathbf{x}_k) - p_k}{p_k} \right). \tag{7}$$

Since $\mathbb{E}(\widehat{t}_{y,DE} - \widehat{t}_{y,\pi}) = 0$, the estimator $\widehat{t}_{y,PSA}$ is virtually unbiased for t_y if

$$\mathbb{E}\left\{ \sum_{k\in\mathscr{S}} \frac{d_k}{\widehat{p}(\mathbf{x}_k)} R_k y_k \left(\frac{\widehat{p}(\mathbf{x}_k) - p_k}{p_k} \right) \right\} \approx 0.$$

An alternative adjusted estimator of t_y is the so-called Hájek estimator

$$\widehat{t}_{y,H} := N \frac{\sum_{k\in\mathscr{S}} \frac{d_k}{\widehat{p}(\mathbf{x}_k)} R_k y_k}{\sum_{k\in\mathscr{S}} \frac{d_k}{\widehat{p}(\mathbf{x}_k)} R_k}. \tag{8}$$

If the nonresponse model is correctly specified, we have $\mathbb{E}(\sum_{k\in\mathscr{S}} \frac{d_k}{\widehat{p}(\mathbf{x}_k)} R_k) \approx N$, which implies that both $\widehat{t}_{y,PSA}$ and $\widehat{t}_{y,H}$ would exhibit the same asymptotic bias. However, they may differ significantly in terms of variance, even in the absence of bias.

2 Estimation vs. prediction

In this section, we illustrate empirically that the most predictive model does not necessarily yield the best estimator of t_y in terms of mean square error. Indeed, including predictors that are highly predictive of R_k may produce very small estimated response probabilities $\widehat{p}_k$, which may potentially result in extreme adjusted weights w_k^*. In this case, both (6) and (8) may be inefficient. How, then, do we choose the $\mathbf{x}_k$ variables to incorporate in the nonresponse model? A common recommendation is to include the variables $\mathbf{x}_k$ that are related to both the indicator variable R_k and the survey variable y; e.g., Little and Vartivarian (2005), Beaumont (2005) and Kim et al. (2019). When an x-variable exhibits a strong correlation with R_k but is unrelated to y, excluding it from the nonresponse model is advisable. Indeed, including such a variable would not effectively mitigate nonresponse bias but could potentially lead to a significant increase in the variance of the adjusted estimator.

To illustrate this point, we conducted a limited simulation study. We generated a finite population $\mathscr{U}$ of size $N = 10,000$ with seven variables: one survey variable y and six auxiliary variables $x_1, x_2, \ldots, x_6$. We first generated the x-variables according to the following distributions: $x_1 \sim$ Gamma$(5, 1)$; $x_2 \sim$ Gamma$(1, 5)$; $x_3 \sim$ Gamma$(1, 6)$; $x_4 \sim$ Gamma$(1, 10)$; $x_5 \sim$ Gamma$(1, 20)$; $x_6 \sim$ Gamma$(0.5, 50)$. Given $x_1, \ldots, x_6$, we generated the y-variable according to the linear regression model

$$y_k = 2 - 2x_{1k} + 4x_{2k} + \epsilon_k,$$

where the errors ϵ_k were generated from a normal distribution with mean equal to zero and variance equal to 225. This led to a model R^2 approximately equal to 0.64.

From the population, we selected $10,000$ samples, of size $n = 1,000$, according to simple random sampling without replacement. In each sample, each unit was assigned a response probability p_k:

$$p_k = 0.05 + 0.95 \left\{1 + \exp\left(-0.05x_{1k} + 0.05x_{2k} - 0.05x_{3k} + 0.05x_{4k} - 0.05x_{5k} + 0.02x_{6k}\right)\right\}^{-1}. \tag{9}$$

This led to a response rate of about 55% in each sample. The response indicators R_k were generated using a Bernoulli distribution with probability p_k.

Our goal was to estimate the population total of the y-values, $t_y = \sum_{k \in \mathscr{U}} y_k$. In our experiment, the variables $x_1, \ldots, x_6$ were fully observed, while the y-variable was prone to missing values.

In each sample, we computed two estimators of t_y:

(i) The naive estimator given by (2).

(ii) The propensity score-adjusted estimator, $\widehat{t}_{y,PSA}$, given by (6), where $\widehat{p}(\mathbf{x}_k)$ was obtained through the score method (see Section 2.1) based on different subsets of $x_1, \ldots, x_6$, and regression trees (see Section 2.2) based on the same subsets of $x_1, \ldots, x_6$.

As a measure of bias of an estimator $\widehat{t}$, we computed the Monte Carlo percent relative bias

$$\mathrm{RB}_{MC}(\widehat{t}) = 100 \times \frac{1}{10,000} \sum_{b=1}^{10{,}000} \frac{(\widehat{t}_{(b)} - t_y)}{t_y}, \tag{10}$$

where $\widehat{t}_{(b)}$ denotes the estimator $\widehat{t}$ in the bth sample, $b = 1, \ldots, 10,000$. We also computed the Monte Carlo relative efficiency of $\widehat{t}$, using the full sample estimator $\widehat{t}_{y,\pi}$ given by (1), as the reference:

$$\mathrm{RE}_{MC}(\widehat{t}) = 100 \times \frac{\mathrm{MSE}_{MC}(\widehat{t})}{\mathrm{MSE}_{MC}(\widehat{t}_{y,\pi})}, \tag{11}$$

where

$$\mathrm{MSE}_{MC}(\widehat{t}) = \frac{1}{10,000} \sum_{b=1}^{10{,}000} \left(\widehat{t}_{(b)} - t_y\right)^2$$

and $\mathrm{MSE}_{MC}(\widehat{t}_{y,\pi})$ is similarly defined.

In each sample, we also computed the Monte Carlo percent coefficient of variation of the adjusted weights $w_k^* = d_k/\widehat{p}(\mathbf{x}_k)$:

$$\mathrm{CV}_{MC}(w_k^*) = 100 \times \frac{1}{B} \sum_{b=1}^{B} \frac{s_{w^*(b)}}{\overline{w}^*_{(b)}},$$

where n_r denotes the number of respondents,

$$s_{w^*} = \sqrt{\frac{1}{n_r - 1} \sum_{k \in \mathcal{S}_r} (w_k^* - \overline{w}^*)^2}$$

and $\overline{w}^* = n_r^{-1} \sum_{k \in \mathcal{S}_r} w_k^*$. Finally, we computed the Monte Carlo mean square error of the predictions defined as

$$\mathrm{MSE}_{MC}(\widehat{p}) = 100 \times \frac{1}{B} \sum_{b=1}^{B} \frac{1}{n_r} \sum_{k \in \mathcal{S}_r} \left(\widehat{p}_{(b)}(\mathbf{x}_k) - p_k\right)^2,$$

where $\widehat{p}_{(b)}(\mathbf{x}_k)$ denotes the estimated response probability attached to unit k in the bth sample.

2.1 The score method

The score method (Little, 1986, Eltinge and Yansaneh, 1997; Haziza and Beaumont, 2007) may be described as follows:

Step 1: Obtain preliminary estimated response probabilities, $\widehat{p}^{LR}(\mathbf{x}_k)$, $k \in \mathcal{S}$, from a logistic regression.

Step 2: Form C classes based on the estimated response probabilities, $\widehat{p}^{LR}(\mathbf{x}_k)$, using an equal quantile method. We set $C = 20$, which led to classes, each of size 50.

Step 3: Adjust the weight of the respondents within a class by multiplying their design weight d_k by the inverse of the response rate observed within the same class.

Estimator	$\widehat{t}_{y,naive}$	$\widehat{t}_{y,PSA}$ x_1	$\widehat{t}_{y,PSA}$ x_1, x_2	$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_3$	$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_4$	$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_5$	$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_6$
RB$_{MC}(\widehat{t})$ in (%)	-13.4	-12.2	-0.2	-0.8	-0.3	-1.0	-0.4
RE$_{MC}(\widehat{t})$	623	561	134	141	142	161	206
CV$_{MC}(w*)$ in (%)	0	13	16	19	30	50	84
MSE$_{MC}(\widehat{p})$	4.7	5.0	4.9	4.6	4.1	1.3	0.4

Table 1: Monte Carlo measures for several estimators of t_y: The score method

The results for the score method, displayed Table 1, can be summarized as follows:

- As expected, the naive estimator was biased with a relative bias of -13.4%. This is not surprising as the naive estimator makes no use of the variables x_1 and x_2, which are related to both R_k and y.

- The propensity score estimator $\widehat{t}_{y,PSA}$ based on the variable x_1 exhibited a smaller bias than the naive estimator. Incorporating the variable x_1 in the nonresponse model helped in reducing the bias, as expected.

- The propensity score estimator $\widehat{t}_{y,PSA}$ based on the variable x_1 and x_2 was nearly unbiased with a value of relative bias of about -0.2%. In terms of relative efficiency, this estimator was the best, with a value of RE equal to 134. It is worth noting that the other propensity score estimators were nearly unbiased but were less efficient than $\widehat{t}_{y,PSA}$ based on x_1 and x_2. In other words, adding x_3 to x_6 to the model had no impact on the bias but led to an increase in variance.

- Since nonresponse was generated according to 9), the most predictive model of R_k was the one that included the variables $x_1, \ldots, x_6$. However, except for $\widehat{t}_{y,PSA}$, based on x_1 only, the estimator $\widehat{t}_{y,PSA}$ based on $x_1, \ldots, x_6$, was the worst in terms of relative efficiency, with a value of RE equal to 209. In comparison with $\widehat{t}_{y,PSA}$, based on x_1 and x_2, this corresponds to a 55% increase in terms of mean square error. This result suggests that the most predictive model may not necessarily translate into the best estimator of t_y. In fact, a quick look at the values of $\text{MSE}_{MC}(\widehat{p})$ suggests that the model that incorporates the variables $x_1, \ldots, x_6$, led to the smallest value of $\text{MSE}_{MC}(\widehat{p})$ (about 0.4), whereas the model that incorporated x_1 and x_2 led to a value of $\text{MSE}_{MC}(\widehat{p})$ of 4.9, which is about 12 times larger.

- A large dispersion of the adjusted weights w_k^* led to estimators with a large variance. This is why, in practice, limiting the dispersion of the adjusted weights w_k^* is desirable.

2.2 Regression trees

We repeated the simulation experiment with regression trees using the same setup described in Section 2.1. The simulation study was conducted using the *R* package rpart. Regression trees require the specification of some hyper-parameters such as the complexity parameter, denoted by c_p, and the minimal number of observations per terminal node, denoted by n_0. The complexity parameter is used to control the size of the tree and to prevent overfitting. We used different values of c_p: 0; 0.001; and 0.01 (the default value). We also used two values for n_0: 10 and 25. With a value of c_p set to 0.001 (say), any split that does not decrease the overall lack of fit by a factor of 0.001 is not attempted. Large values of c_p will thus lead to shallower trees.

Results for $n_0 = 10$ and $n_0 = 25$ are shown in Table 2 and Table 3, respectively. They can be summarized as follows:

- For $n_0 = 10$, we note that the estimator $\widehat{t}_{y,PSA}$, based on x_1 and x_2, was nearly unbiased for $c_p = 0$ and $c_p = 0.001$. However, the bias of $\widehat{t}_{y,PSA}$ increased as more variables were incorporated into the tree procedure. For instance, for $c_p = 0$, the estimator $\widehat{t}_{y,PSA}$, based on x_1 and x_2, showed a value of relative bias of about -0.6%, whereas the estimator $\widehat{t}_{y,PSA}$, based on x_1-x_6 showed a relative bias of about -6.5%. The same was true for all values of c_p. This is due to the fact that, as the number of predictors increased, the proportion of splits involving either x_1 or x_2 (the variables associated with both R_k and y) diminished. For instance, for $c_p = 0$ and only x_1 and x_2 were used as predictors, 100% of the splits used either x_1 or x_2. But when all the variables x_1-x_6 were included, only 16.8% of the splits used x_1, and 13.5% of the splits used x_2. In other words, above 70% of the splits did not use either x_1 or x_2.

- With an increasing value of c_p, the tree became progressively shallower, which led to larger biases. For instance for $c_p = 0$, the estimator $\widehat{t}_{y,PSA}$ based on x_1 and x_2, showed a value of RB equal to -0.6%, whereas it was equal to -8.0% for $c_p = 0.01$. Fewer terminal nodes limit the tree's ability to capture local behaviors effectively.

- Results for $n_0 = 25$ followed similar patterns as those obtained for $n_0 = 10$, except that the propensity score estimator was biased in all the scenarios.

- Like the score method, the value of $\text{MSE}_{MC}(\widehat{p})$ decreased as more predictors were incorporated in the model. Similarly, the dispersion of the adjusted weights w_k^* increased as more predictors were included.

2.3 Discussion

In Sections 2.1 and 2.2, we performed propensity score estimation based on the score method and regression trees, respectively. For regression trees, the bias of $\widehat{t}_{y,PSA}$ increased as more predictors were included in the model. This pattern was not observed for the score method. Indeed, in the case of the score method, the weighting classes were based on the preliminary score $\widehat{p}^{LR}(\mathbf{x}_k)$, which can be viewed as a scalar summary of all the information contained in

	$\text{RB}_{MC}(\widehat{t})$ in (%)	$\text{RE}_{MC}(\widehat{t})$ in (%)	$\text{MSE}_{MC}(\widehat{p})$	$\text{CV}_{MC}(w*)$ in (%)
	$c_p = 0$			
$\widehat{t}_{y,PSA}$ x_1	-11.1	572	4.0	29
$\widehat{t}_{y,PSA}$ x_1, x_2	-0.6	116	4.3	36
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_3$	-1.7	140	3.9	43
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_4$	-2.6	162	3.8	48
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_5$	-4.1	206	3.4	53
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_6$	-6.5	318	2.9	62
	$c_p = 0.001$			
$\widehat{t}_{y,PSA}$ x_1	-11.2	577	3.9	29
$\widehat{t}_{y,PSA}$ x_1, x_2	-0.7	117	4.2	36
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_3$	-1.8	142	3.8	43
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_4$	-2.8	164	3.7	48
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_5$	-4.1	209	3.3	53
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_6$	-6.6	322	2.9	62
	$c_p = 0.01$			
$\widehat{t}_{y,PSA}$ x_1	-13.7	802	3.0	5
$\widehat{t}_{y,PSA}$ x_1, x_2	-8.0	414	3.0	14
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_3$	-7.3	360	2.9	23
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_4$	-7.3	341	2.8	33
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_5$	-7.8	364	2.6	39
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_6$	-10.0	519	2.4	49

Table 2: Monte Carlo measures for several estimators of t_y: Regression trees with $n_0 = 10$

	RB$_{MC}(\widehat{t})$ in (%)	RE$_{MC}(\widehat{t})$ in (%)	MSE$_{MC}(\widehat{p})$	CV$_{MC}(w*)$ in (%)
	$c_p = 0$			
$\widehat{t}_{y,PSA}$ x_1	-11.6	608	3.1	15
$\widehat{t}_{y,PSA}$ x_1, x_2	-3.1	168	3.1	20
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_3$	-4.6	210	2.8	26
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_4$	-5.9	263	2.7	29
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_5$	-7.4	337	2.5	33
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_6$	-10.0	514	2.2	41
	$c_p = 0.001$			
$\widehat{t}_{y,PSA}$ x_1	-11.8	625	3.1	14
$\widehat{t}_{y,PSA}$ x_1, x_2	-3.4	174	3.1	19
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_3$	-4.7	214	2.8	26
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_4$	-6.0	268	2.7	29
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_5$	-7.4	341	2.5	33
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_6$	-10.1	517	2.2	41
	$c_p = 0.01$			
$\widehat{t}_{y,PSA}$ x_1	-14.0	824	3.1	2
$\widehat{t}_{y,PSA}$ x_1, x_2	-9.2	489	3.0	9
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_3$	-8.2	403	2.8	17
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_4$	-8.7	419	2.7	24
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_5$	-9.2	447	2.5	30
$\widehat{t}_{y,PSA}$ $x_1, \ldots, x_6$	-11.6	632	2.3	38

Table 3: Monte Carlo measures for several estimators of t_y: Regression trees with $n_0 = 25$

$x_1, \ldots, x_6$. Therefore, the sample partitions obtained through the score method implicitly made use of all the predictors, and in particular, x_1 and x_2. This is why $\widehat{t}_{y,PSA}$ was virtually unbiased as long as at least both x_1 and x_2 were included. For regression trees, the situation is more intricate. Indeed, when all the predictors $x_1, \ldots, x_6$ were included, we ended up with trees that made use of x_1 and x_2 for only a fraction of the splits. As a result, we were not able to eliminate the nonresponse bias as effectively.

These results suggest we should exercise caution if variable selection is performed prior to nonresponse adjustment. Indeed, if the variable selection method resulted in the elimination of some important predictors (which are those that are related to both R_k and y) in the presence of other predictors that are highly related to R_k but not to y, the propensity score-adjusted estimator may likely suffer from an appreciable bias.

3 Simulation study

We conducted an extensive simulation study to assess the performance of several machine learning procedures (see Section 3.2 below) in terms of bias and efficiency.

3.1 The setup

We generated several finite populations of size $N = 50,000$. Each population consisted of a survey variable Y and seven auxiliary variables, four of which were continuous and the remaining being discrete. First, the continuous auxiliary variables were generated as follows: $X^{(s)} \sim \text{Gamma}(3, 2)$, $X^{(c_1)} \sim \mathscr{N}(0, 1)$; $X^{(c_2)} \sim \text{Gamma}(3, 2)$ and $X^{(c_3)} \sim \text{Gamma}(3, 2)$. The discrete auxiliary variables were generated as follows: $X^{(d_1)} \sim \mathscr{MN}(N, 0.5, 0.05, 0.05, 0.1, 0.3)$; $X^{(d_2)} \sim \mathscr{B}(0.5)$ and $X^{(d_3)} \sim \mathscr{UD}(1; 5)$, where $\mathscr{MN}$, $\mathscr{B}$ and $\mathscr{UD}$ denote the multinomial, the Bernoulli and the uniform discrete distributions, respectively. Two configurations for these predictors were used: (i) The predictors were independently generated; (ii) The predictors were generated through Gaussian copulas to produce a level of correlation among them.

Given the values of the auxiliary variables, we generated several y-variables according to the following two models:

$$\begin{aligned} y_k = \gamma_0 + \gamma_1^{(s)} X_{1k}^{(s)} + \gamma_1^{(c)} X_{1k}^{(c)} + \gamma_2^{(c)} X_{2k}^{(c)} + \gamma_3^{(c)} X_{3k}^{(c)} + \sum_{j=2}^{5} \gamma_{1j}^{(d)} (1_{\{X_{1k}^{(d)}=j\}}) \\ + \gamma_2^{(d)} X_{2k}^{(d)} + \sum_{k=2}^{5} \gamma_{3j}^{(d)} (1_{\{X_{3k}^{(d)}=j\}}) + \varepsilon_k \qquad (12) \end{aligned}$$

and

$$y_k = \delta_1 X_{2k}^{(c)} + \delta_2 (X_{2k}^{(c)})^2 (1 - 1_{\{X_{3k}^{(d)}=2\} \cup \{X_{3k}^{(d)}=3\}}) + \log(1 + \delta_3 X_{2k}^{(c)})(1_{\{X_{3k}^{(d)}=2\} \cup \{X_{3k}^{(d)}=3\}}) + \varepsilon_k, \quad (13)$$

where $\varepsilon \sim \mathscr{N}(0, \sigma_\varepsilon^2)$. Model (12) is linear in the regression coefficients, whereas Model (13) is nonlinear.

Each population was partitioned into ten strata on the basis of the auxiliary variable $X^{(s)}$ using an equal quantile method. From each population, we selected $B = 5,000$ samples according to stratified simple random sampling without replacement of size $n = 1,000$ based on Neyman's allocation.

For the populations generated according to the linear model (12), we simulated the case of both a (virtually) non-informative sampling design and an informative sampling design. For the non-informative sampling design, the correlation between the y-variable and the design weights d_k was equal to 0.02, whereas it was equal to approximately 0.3 for the informative sampling design. For the non-informative sampling design, the vector of coefficients $\left(\gamma_0, \gamma^{(s)}, \gamma_1^{(c)}, \gamma_2^{(c)}, \gamma_3^{(c)}, \gamma_{12}^{(d)}, \gamma_{13}^{(d)}, \gamma_{14}^{(d)}, \gamma_{15}^{(d)}, \gamma_{22}^{(d)}, \gamma_{32}^{(d)}, \gamma_{33}^{(d)}, \gamma_{34}^{(d)}, \gamma_{35}^{(d)}\right)$ was set to
$(-0.2, 5.0, 5.0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)$ and to
$(-10, 5.0, 5.0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)$ for the informative sampling design. Finally, for the nonlinear model (13), the vector of coefficients $(\delta_0, \delta_1, \delta_2, \delta_3)$ was set to $(4, 4, 4, 4)$. This led to 6 different survey variables y; see Table 4. Our goal was to estimate the population total of each of the six survey variables, $t_y = \sum_{k \in U} y_k$.

In each sample, nonresponse to the survey variable Y was generated according to six nonresponse mechanisms. That is, for each $k \in \mathscr{S}$, we assigned a response probability p_k according to the following six models:

	Linear model		Nonlinear model	
	Independent predictors	Correlated predictors	Independent predictors	Correlated predictors
Informative sampling	✓	✓	X	X
Noninformative sampling	✓	✓	✓	✓

Table 4: Strategies used to generate the six survey variables

NR1: $p_k^{(1)} = \text{logit}^{-1}(-0.8 - 0.05X_{1k}^{(s)} + 0.2X_{1k}^{(c)} + 0.5X_{2k}^{(c)} - 0.05X_{3k}^{(c)} + \sum_{k=2}^{5} 0.2(1_{\{X_{1k}^{(c)}=k\}}) + 0.2X_{2k}^{(d)} + \sum_{k=2}^{5} 0.3(1_{\{X_{3k}^{(d)}=k\}}));$

NR2: $p_k^{(2)} = 0.1 + 0.9\,\text{logit}^{-1}(0.5 + 0.3X_{1k}^{(s)} - 1.1X_{1k}^{(c)} - 1.1X_{2k}^{(c)} - 1.1X_{3k}^{(c)} + \sum_{k=2}^{5} 0.8(1_{\{X_{1k}^{(c)}=k\}}) + 0.8X_{2k}^{(d)} + \sum_{k=2}^{5} 0.8(1_{\{X_{3k}^{(d)}=k\}}));$

NR3: $p_k^{(3)} = 0.1 + 0.9\,\text{logit}^{-1}\left(-1 + \text{sgn}\left(X_{1k}^{c}\right)\left(X_{1k}^{c}\right)^2 + 3 \times 1_{\left\{X_{1k}^{(d)}<4\right\} \cap \left\{X_{2k}^{(d)}=1\right\}}\right);$

NR4: $p_k^{(6)} = 0.1 + 0.6\,\text{logit}^{-1}(0.85X_{1k}^{(s)} + 0.85X_{2k}^{(c)} - 0.85X_{3k}^{(c)} - \sum_{k=2}^{5} 0.2(1_{\{X_{1k}^{(c)}=k\}}) + 0.2X_{2k}^{(d)} - \sum_{k=2}^{5} 0.3(1_{\{X_{3k}^{(d)}=k\}}));$

NR5: $p_k^{(4)} = 0.55 + 0.45\tanh\left(0.05y_k - 0.5\right);$

NR6: $p_k^{(5)} = 0.1 + 0.9\,\text{logit}^{-1}\left(0.2y_k - 1.2\right).$

The parameters in each nonresponse model were set so as to obtain a response rate approximately equal to 50% in each sample. The response indicators $R_k^{(j)}$ were generated from a Bernoulli distribution with probability $p_k^{(j)}$, $j = 1, \ldots, 6$. Note that the nonresponse mechanism NR1-NR4 involved x-variables only. Below, they will be referred to as ignorable nonresponse mechanisms. For the nonresponse mechanism NR5 and NR6, we used the generic notation y in the definition of the response probability p_k. Recall that we have generated six different survey variables (see Table 4). The y-variable in the expressions of p_k in NR5 and NR6 stands for the first survey variable when we are interested in estimating the population total of the first survey variable, stands for the second survey variable when we are interested in estimating the population total of the second survey variable, and so on. Below, NR5 and NR6 will be referred to as non-ignorable nonresponse mechanisms.

Overall, we ended up with $6 \times 6 = 36$ scenarios, each corresponding to a given survey variable and a given nonresponse mechanism. Out of the 36 scenarios, 24 were of the ignorable type, and 12 were of the non-ignorable type.

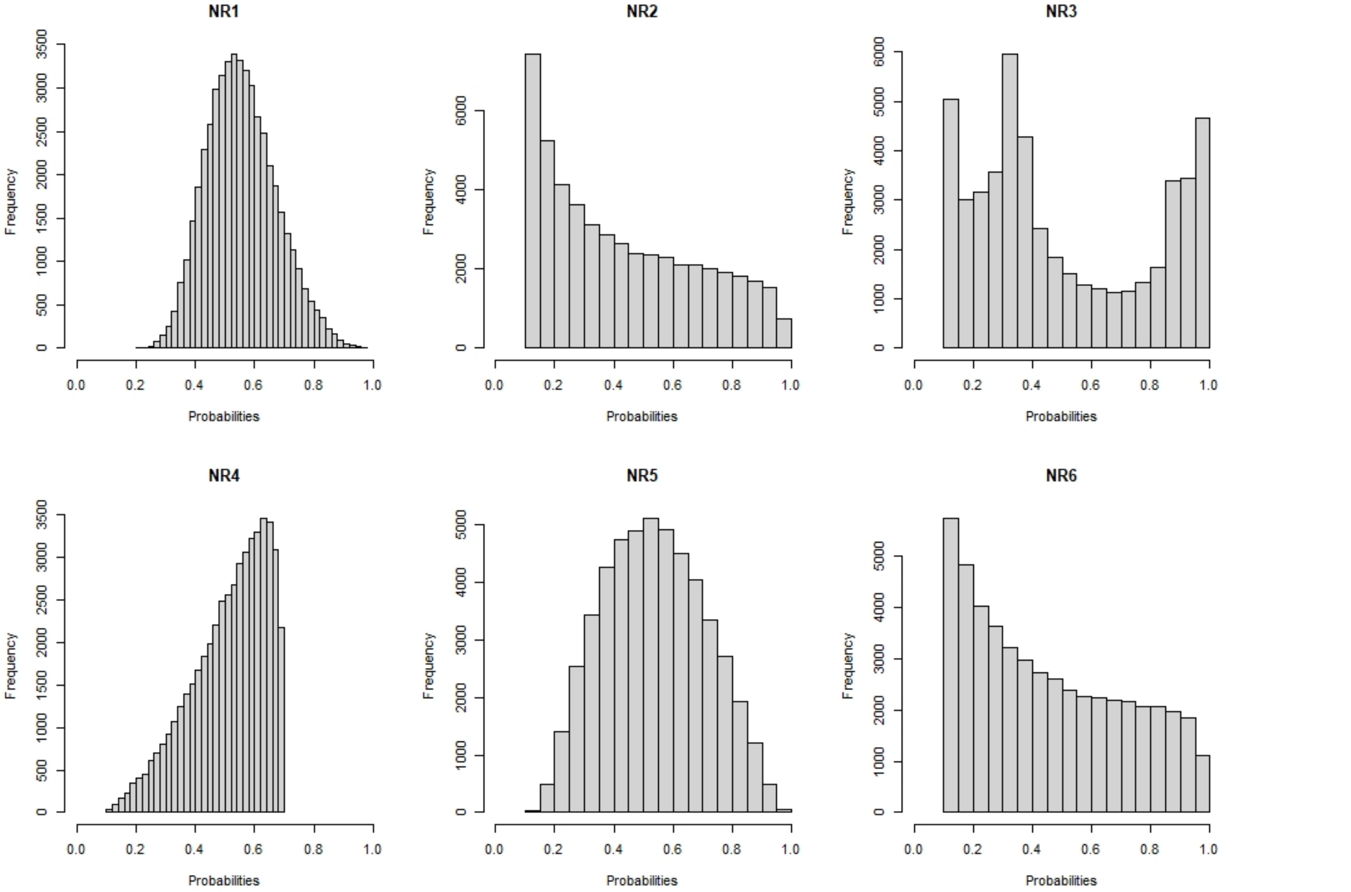

Figure 1: Distribution of response probabilities in the population $\mathscr{U}$

To estimate the response probabilities p_k, we used the following machine learning procedures based on the set of explanatory variables, $X^{(s)}$, $X_1^{(c)}$, $X_2^{(c)}$, $X_2^{(c)}$, $X_1^{(d)}$, $X_2^{(d)}$ and $X_3^{(d)}$:

(a) Logistic regression;

- `logit`.

(b) Logistic regression with variable selection based on LASSO; e.g., see Hastie et al. (2001). This procedure was implemented using the R package *glmnet*.

- `logit_lasso`: the amount of penalization λ was obtained using a 10-fold cross validation.

(c) Classification and regression trees; see Breiman et al. (1983). This procedure was implemented using the R package *rpart*.

- `cart20`: Unpruned trees, $c_p = 0$, at least 20 observations in each leaf.
- `cart30`: Unpruned trees, $c_p = 0$, at least 30 observations in each leaf.
- `cart40`: Unpruned trees, $c_p = 0$, at least 40 observations in each leaf.
- `cart50`: Unpruned trees, $c_p = 0$, at least 50 observations in each leaf.

(d) Random forests; e.g., see Breiman (2004). This procedure was implemented using the R package *ranger*.

- `rf1`: Probabilities estimation trees, at least 10 observations in each leaf, 100 trees.
- `rf2`: Probabilities estimation trees, at least 10 observations in each leaf, 500 trees.
- `rf3`: Probabilities estimation trees, at least 30 observations in each leaf, 100 trees.
- `rf4`: Probabilities estimation trees, at least 30 observations in each leaf, 500 trees.
- `rf5`: Probabilities estimation trees, at least 30 observations in each leaf, 500 trees, variable used for the allocation is selected with probability 1 at each split.

(e) k-nearest neighbors; This procedure was implemented using the R package *caret*.

- knn: k determined by 10-fold cross validation with $k \in \{3, \ldots, 12\}$.
- knn_reg: k determined by 10-fold cross validation with $k \in \{3, \ldots, 30\}$.

(f) Bayesian additive regression tree; e.g., see Chipman et al. (2010). These procedures were implemented using the R packages *dbarts* and *BART*.

- `bart` Bart as a classification method with parameters described in Chipman et al. (2010) for all priors.
- `bart_reg`: Bart as a regression method with parameters described in Chipman et al. (2010) for all priors.

(g) Extreme Gradient Boosting (XGBoost); see Chen and Guestrin (2016). This procedure was implemented using the R package *xgboost*.

- `xb1`: 500 trees, $\Gamma = 10$, proportion for subsets : 75 %, learning rate : 0.5, max depth: 2.
- `xgb2`: 2000 trees, $\Gamma = 2$, proportion for subsets : 100 %, learning rate : 0.5, max depth : 2.
- `xgb3`: 1000 trees, $\Gamma = 2$, proportion for subsets : 75 %, learning rate : 0.01, max depth : 1.
- `xgb4`: 500 trees, $\Gamma = 10$, proportion for subsets : 75 %, learning rate : 0.05, max depth : 3.

(h) Support vector machine; This procedure was implemented using the R package *e1071*.

- `svm1`: $\nu-$SVM with a Gaussian kernel, $\nu = 0.7$, $\gamma = 0.025$.
- `svm2`: $\nu-$SVM with a linear kernel, $\nu = 0.7$.

(i) Cubist algorithm; see Quinlan (1992; 1993). This procedure was implemented using the R package *Cubist*.

- `cb1`: Unbiased, 100 rules, with extrapolation, 10 committees.
- `cb2`: Unbiased, 100 rules, without extrapolation, 10 committees.
- `cb3`: Biased, 100 rules, with extrapolation, 10 committees.
- `cb4`: Unbiased, 100 rules, with extrapolation, 50 committees.
- `cb5`: Unbiased, 100 rules, with extrapolation, 100 committees.

(j) Model-based recursive partitioning; see Zeileis et al. (2008). This procedure was implemented using the R package *partykit*.

- `mob`: logit model fitted, $X^{(s)}$ for stratification.

This led to 28 machine learning procedures. The choice of packages we made to implement the method is somewhat subjective. For some machine learning procedures, several packages are available on CRAN. We acknowledge that there may be more efficient packages than the ones we used. The hyperparameters were chosen to ensure a variety of combinations commonly encountered in practice. The goal was to better understand the behavior of machine learning procedures when the hyperparameters were varied.

For some scenarios, some machine learning procedures produce extremely small estimated response probabilities or probabilities that exceed 1. To address this, we implemented a trimming procedure, which ensured that the estimated response probabilities lay in the range of $[0.025, 1]$. The estimated response probabilities that did not undergo truncation were then adjusted, so that the sum of estimated response probabilities after trimming was equal to the sum before trimming.

In each sample, we computed two estimators: (i) the propensity score-adjusted estimator, $\widehat{t}_{y,PSA}$ given by (6) and (ii) The Hájek estimator, $\widehat{t}_{y,H}$ given by (8). As a measure of bias of an estimator $\widehat{t}_y$, we computed its Monte Carlo percent relative bias given by (10). As a measure of efficiency, we computed the Monte Carlo relative efficiency, using the complete data estimator $\widehat{t}_{y,\pi}$, as the reference; see Expression (11).

3.2 Simulation results

Tables 5 and 6 show some Monte Carlo descriptive statistics regarding the relative efficiency (RE) for the PSA and Hájek estimators, respectively, across all the 36 scenarios: the minimum (Min), the first quartile (Q1), the median (Median), the third quartile (Q3) and the maximum (Max). In Tables 5 and 6, the machine learning procedures are ordered from the best to the worst with respect to the median percent RE (the median of the 36 RE-values). Figures 2 and 3 display the median percent absolute relative bias on the x-axis and the median percent RE on the y-axis for the PSA estimator; see Figures 4 and 5 for the Hájek estimator.

From Table 5, we note that three procedures stood out in terms of relative efficiency: BART, random forests, and XGboost. The commonly employed score method did not yield impressive results, with a median percent RE of about 1236. In the best-case scenario, it exhibited a minimum RE of 318, which was significantly higher than that of the best procedures that exhibited a minimum RE between 130 and 160. Similarly, in the worst-case scenario, it exhibited a value of a maximum RE of 20307, which was considerable. In contrast, the best procedures exhibited a maximum RE ranging between 1800 and 2300 approximately in the worst scenario. Finally, the procedures mob, cubist, and support vector machines performed the least favorably in our experiments. While we were unable to find a set of hyper-parameters for which they worked well, this does not mean that these methods would perform as poorly as they did with other sets of hyperparameters. For the 24 ignorable mechanisms, Figure 2 suggests that regression trees (cart) performed well in terms of median absolute RB but that they were not the most efficient in terms of RE. A similar behavior was observed for the 12 nonignorable mechanisms; see Figure 5.

Results for the Hájek estimator in Table 6 were similar to those for the PSA estimator. Again, the best machine learning procedures were: XGboost, BART, and random forests. These procedures had similar performances in terms of median percent RE. BART was especially good in the worst scenario with values of maximum percent RE equal to 1710 and 1743, which was significantly smaller than the corresponding values for XGboost and random forests. Again, the score method was outperformed by these three procedures in virtually all the scenarios.

Figures 6 and 7 display side-by-side boxplots of the distribution of the PSA estimator and the Hájek estimator for the 24 ignorable nonresponse mechanisms and the 12 nonignorable nonresponse mechanisms, respectively. For the 24 nonignorable nonresponse mechanisms, our analysis reveals that, in the worst-case scenarios, the Hájek estimator consistently outperformed the PSA estimator, as depicted in Figure 6. The Hájek estimator was thus more robust to varying conditions than the PSA estimator, at least in our experiments. In the case of the 12 nonignorable mechanisms, the results were not as clear-cut. For most machine learning procedures (except Xgboost1, Xgboost2, and Xgboost4), we observed that the Hájek estimator performed slightly better than the PSA estimator in the worst-case scenarios.

ML procedure	**Min**	**Q1**	**Median**	**Q3**	**Max**	**Mean**
bart 1	144	194	280	635	1845	489
rf 2	130	211	281	660	2799	561
rf 1	131	213	282	657	2781	560
xgb 2	132	197	295	621	2054	515
rf 5	154	207	304	717	2331	576
xgb 1	172	215	326	653	2253	552
rf 4	157	212	329	782	2359	579
rf 3	158	213	330	784	2351	579
xgb 3	171	231	336	837	2227	589
xgb 4	178	238	338	719	2574	607
knn 1	174	243	346	778	2174	576
bart 2	169	215	359	853	2087	628
knn 2	157	219	360	740	3543	693
cart 20	132	255	490	716	1904	611
cart 50	139	242	504	867	2185	602
cart 30	130	240	508	704	1924	608
cart 40	132	238	509	785	2050	605
logit	145	216	521	1233	4948	952
logit lasso	149	221	553	1242	4556	898
mob	146	254	579	1355	5287	1037
cubist 2	128	339	614	1642	37936	3128
cubist 5	151	290	648	1368	24764	1978
cubist 4	151	290	655	1396	25358	2010
cubist 1	156	323	708	1612	29335	2287
score	318	746	1236	1811	20307	2495
svm 2	251	673	2188	11525	140425	20169
svm 1	251	669	2327	9823	96179	10414
cubist 3	312	4034	10242	35640	13988674	445022

Table 5: Descriptive statistics about percent RE across the 36 scenarios: PSA estimator

ML procedure	**Min**	**Q1**	**Median**	**Q3**	**Max**	**Mean**
xgb 4	180	221	304	732	2912	599
bart 1	158	200	306	556	1710	478
bart 2	176	205	307	656	1743	522
xgb 1	175	209	307	643	2457	547
rf 4	174	205	314	729	2355	569
rf 3	173	205	315	729	2347	568
xgb 3	175	206	324	709	2447	577
xgb 2	159	199	325	572	2057	517
rf 5	167	215	326	770	2074	581
rf 2	170	203	328	657	2462	558
rf 1	170	204	330	656	2453	557
knn 1	179	223	337	628	1867	534
cart 50	148	211	368	602	2195	514
cart 40	141	216	380	621	2040	512
knn 2	202	238	385	818	3379	714
cart 30	140	220	400	629	1905	512
cart 20	146	237	402	621	1889	522
logit lasso	145	201	414	1031	1811	613
mob	141	213	456	1054	1793	648
logit	139	201	457	953	1903	607
cubist 2	147	293	522	882	3857	768
cubist 5	151	254	525	799	3262	713
cubist 4	152	256	527	799	3276	715
cubist 1	153	261	546	800	3348	729
score	224	505	723	1353	8356	1332
cubist 3	224	582	812	1183	4528	1106
svm 2	189	358	910	1401	5024	1161
svm 1	189	357	952	1482	4884	1122

Table 6: Descriptive statistics about percent RE across the 36 scenarios: Hájek Estimator

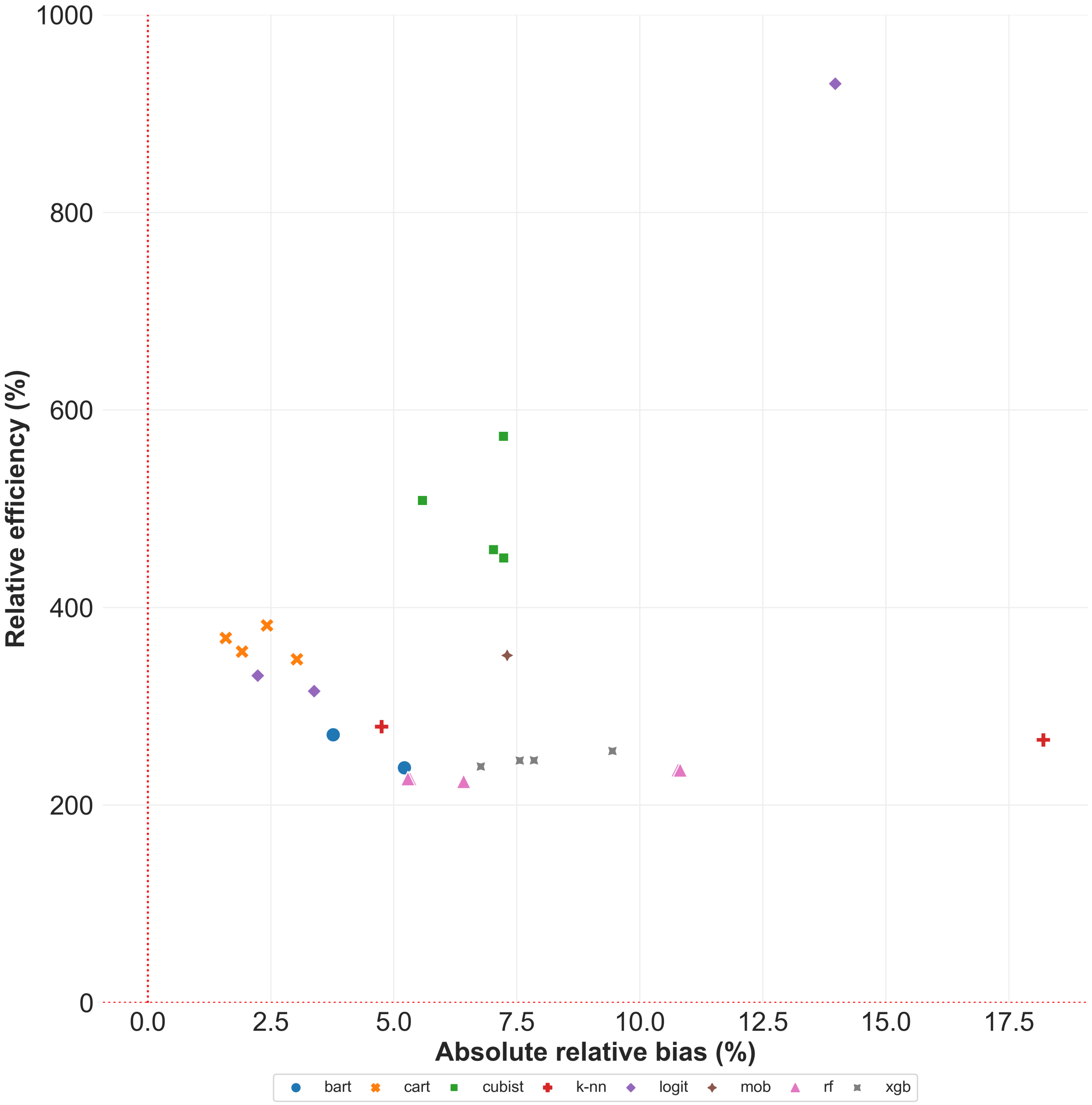

Figure 2: Median percent RE vs. median percent RB for the 24 ignorable mechanisms: PSA Estimator

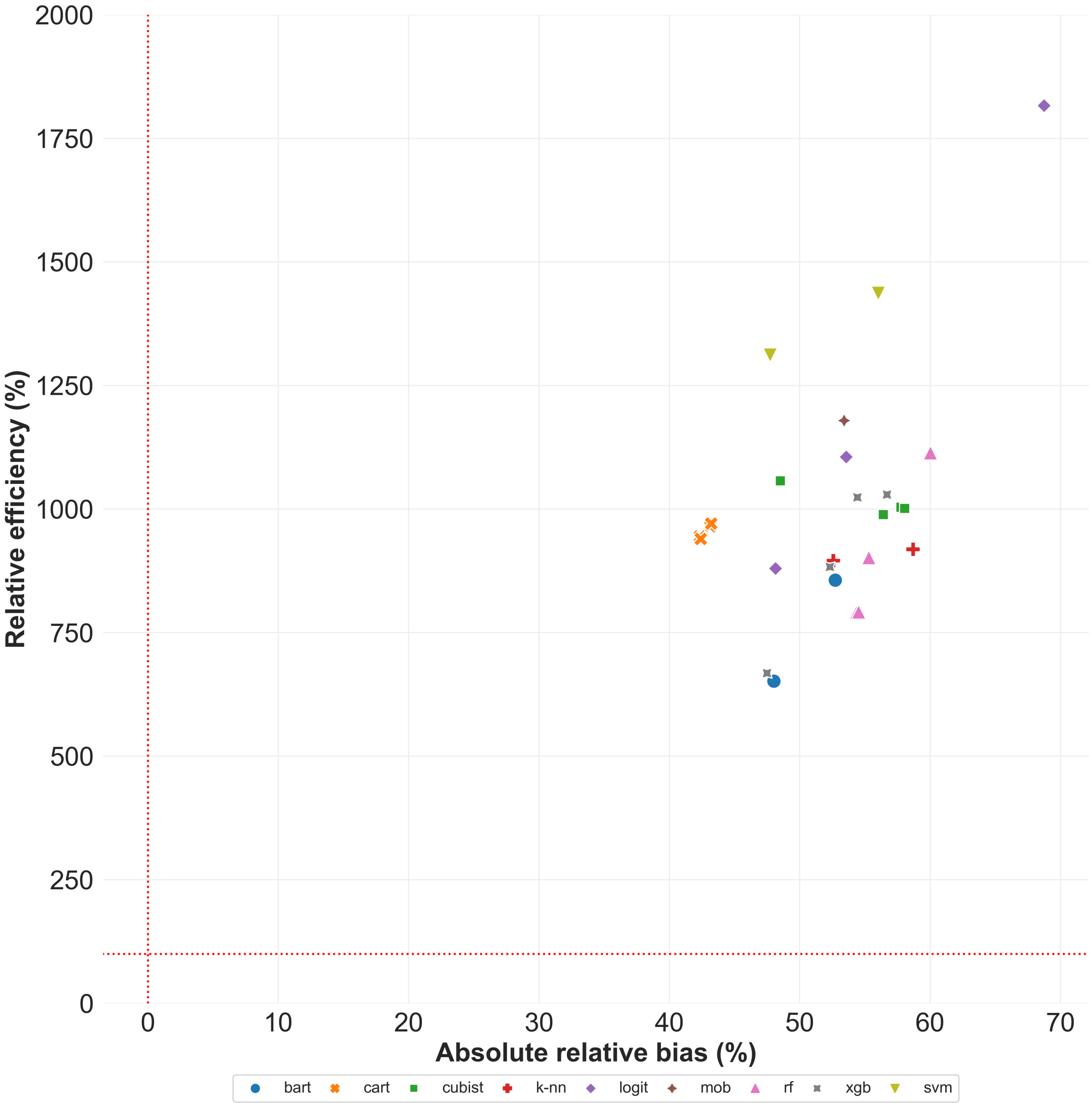

Figure 3: Median percent RE vs. median percent RB for the 12 nonignorable mechanisms: PSA Estimator

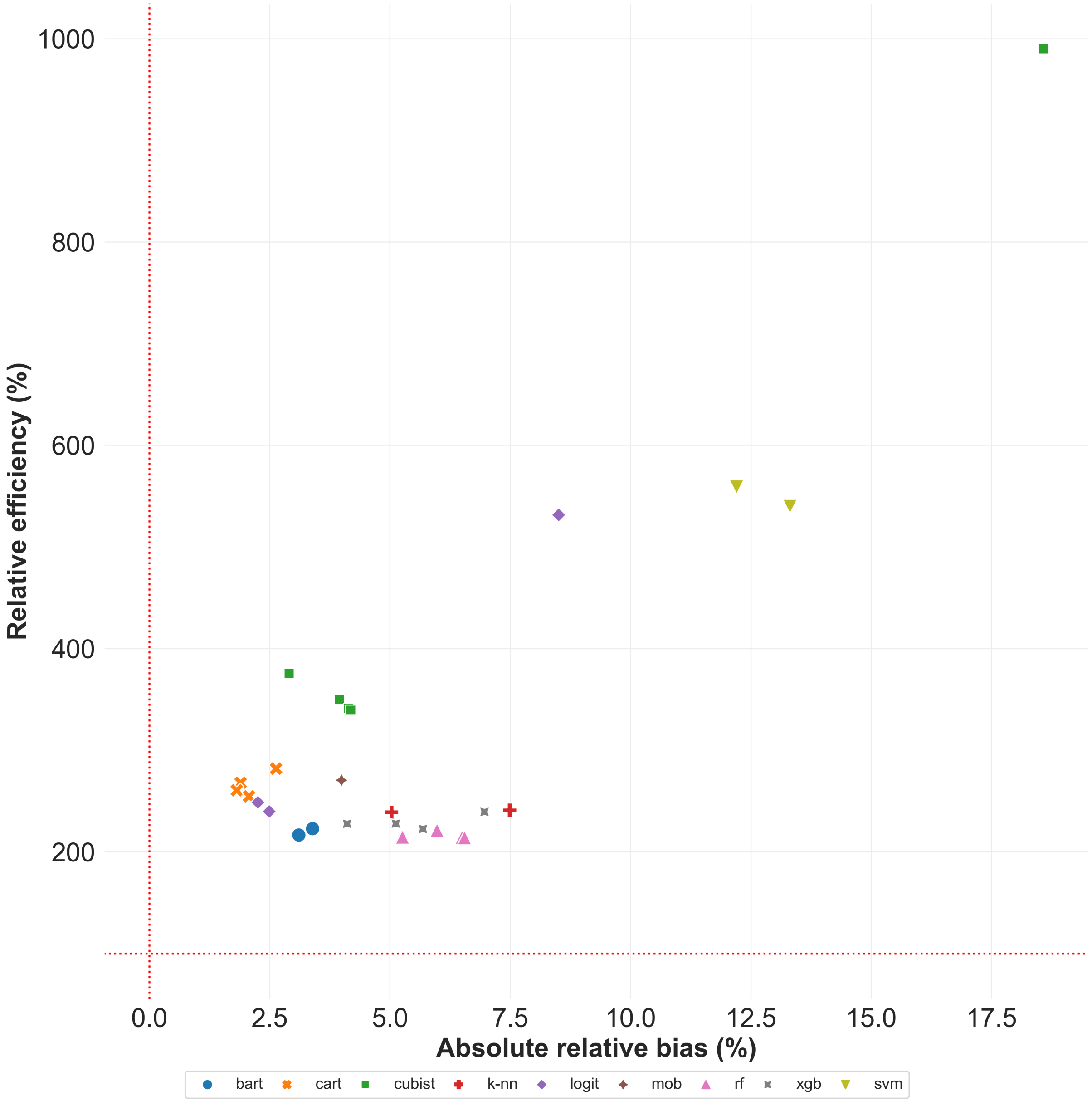

Figure 4: Median percent RE vs. median percent RB for the 24 ignorable mechanisms: Hájek Estimator

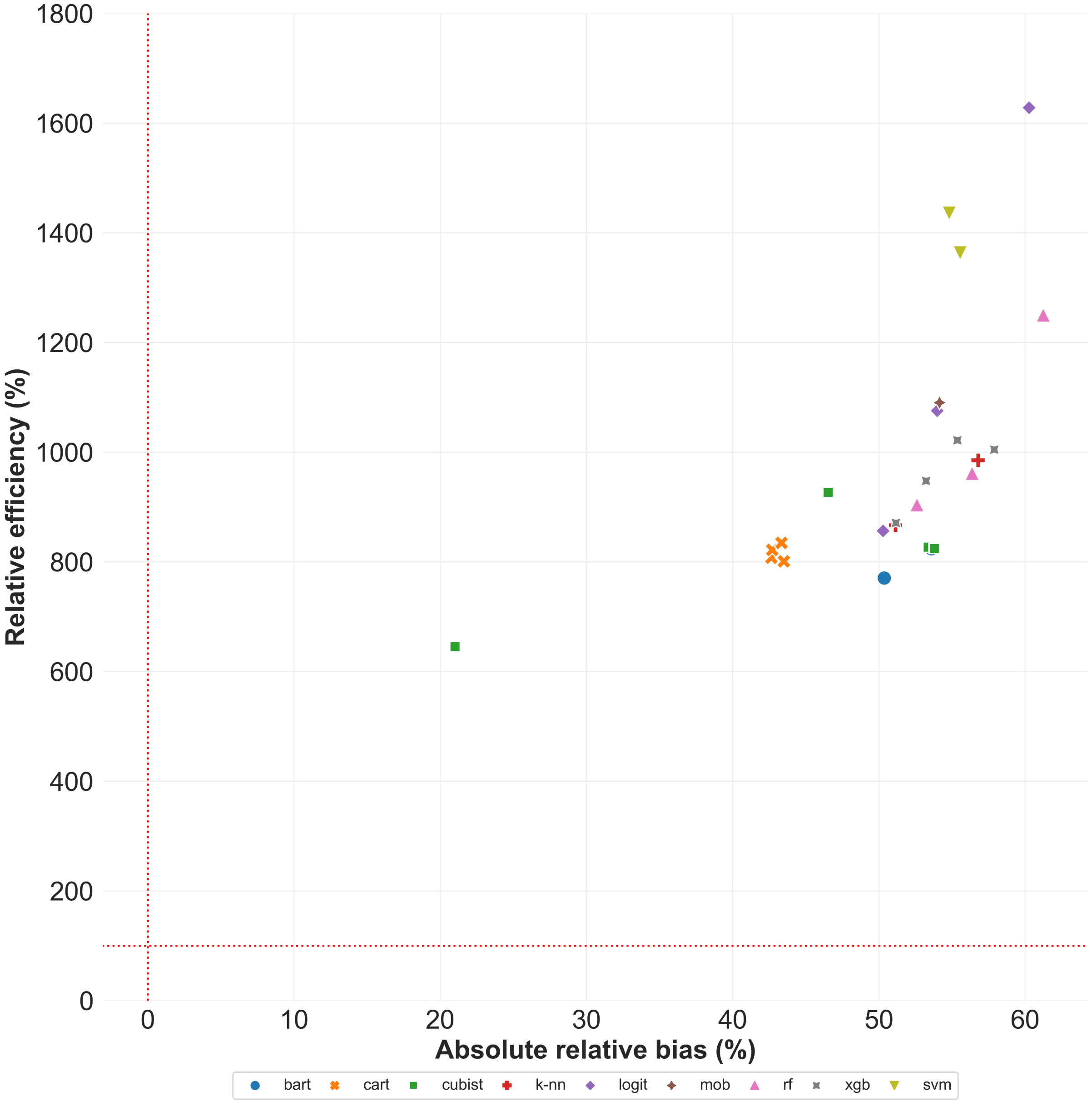

Figure 5: Median percent RE vs. median percent RB for the 12 nonignorable mechanisms: Hájek Estimator

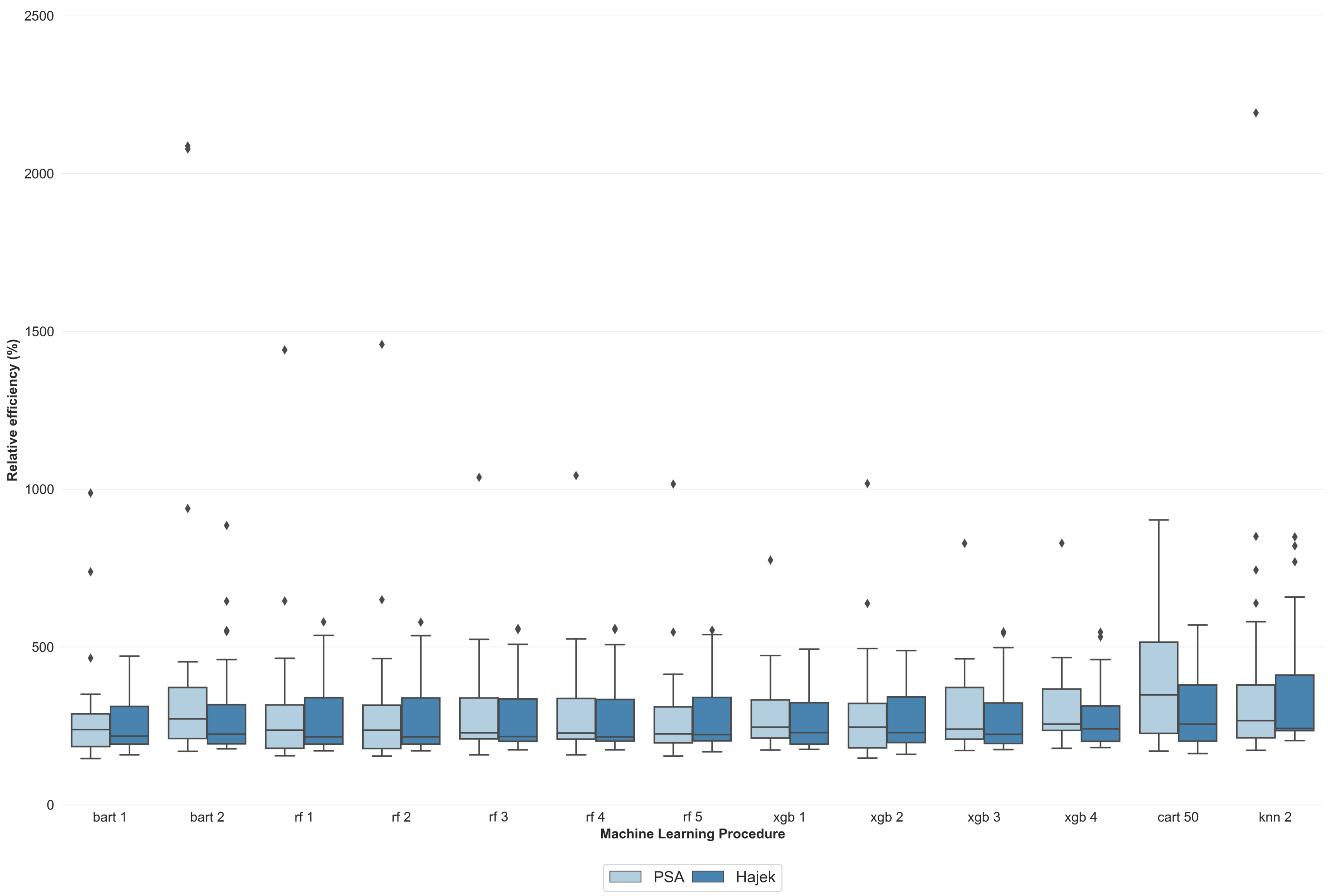

Figure 6: Distribution of the percent relative efficiency across the 24 ignorable nonresponse mechanisms for selected machine learning procedures

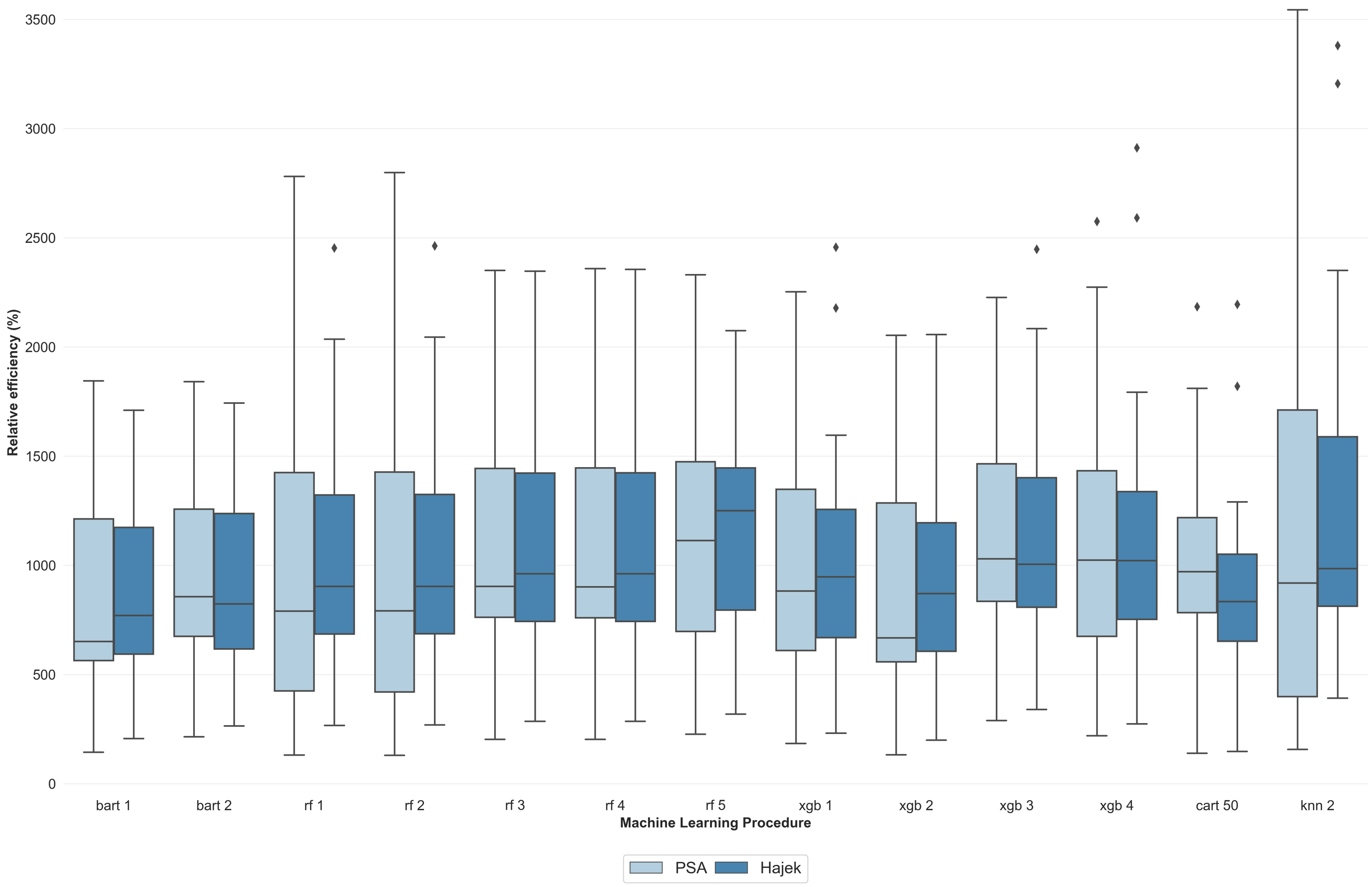

Figure 7: Distribution of the percent relative efficiency across the 12 nonignorable nonresponse mechanisms for selected machine learning procedures

4 Aggregation procedures

Aggregation procedures refer to techniques used to combine the predictions from multiple models into a single, more robust, and accurate prediction. These methods are commonly used in ensemble learning, where the goal is to improve a model's performance by leveraging the strength of multiple models (Nemiroski, 2000). In the context of unit nonresponse, multiple machine learning procedures are used to obtain a set of estimated response probabilities for each sample unit. These probabilities are then combined in some way to obtain an aggregate score. Why use an ensemble method? In general, there are no machine learning procedures that outperform all the other competitors in all the scenarios. Indeed, a given machine learning procedure may do well in a particular scenario but may perform poorly in another scenario. However, one cannot tell in advance which procedure will perform well for a specific scenario. An aggregation procedure may outperform a single procedure in terms of bias and efficiency; e.g., see Tsybakov (2003).

We describe two aggregation procedures for combining predictions from multiple models. Let $\widehat{p}_k^{(m)}(\mathbf{x}_k)$ be the estimated response probability attached to unit k obtained through the mth machine learning procedure $m = 1, \ldots, M$. For both aggregation procedures, the aggregate score for unit k is defined as

$$\widehat{p}_k^{agg} = \sum_{m=1}^{M} \omega_m \widehat{p}_k^{(m)}(\mathbf{x}_k), \tag{14}$$

such that $\omega_m \geq 0$ for all $m = 1, \ldots, M$, and $\sum_{m=1}^{M} \omega_m = 1$. That is, the aggregate score $\widehat{p}_k^{agg}$, is expressed as a convex combination of the individual predictions obtained from each of the M models. Assuming that the estimated response probabilities $\widehat{p}_k^{(m)}(\mathbf{x}_k), m = 1, \cdots, M$, all lie between 0 and 1, the convex combination (14) ensures that the aggregate score $\widehat{p}_k^{agg}$ also lies between 0 and 1. Machine learning procedures that perform well will be assigned a larger weight ω_m in the weighted average (14). The resulting aggregated PSA estimator is defined as

$$\widehat{t}_{PSA,agg} := \sum_{k \in \mathcal{S}} \frac{d_k}{\widehat{p}_k^{agg}} R_k y_k.$$

We now described two standard weighting procedures: linear weighting (Bunea et al., 2006, 2007) and exponential weighting (Buckland et al., 1997):

(1) *Linear weighting.*

The aggregate score $\widehat{p}_k^{agg}$ attached to unit k is obtained by fitting a linear regression model with the response indicator R_k as the dependent variable and $\widehat{p}_k^{(1)}(\mathbf{x}_k), \ldots, \widehat{p}_k^{(M)}(\mathbf{x}_k)$, as the set of explanatory variables. Let $\widehat{\beta}_1, \ldots, \widehat{\beta}_M$, denote the resulting estimated regression coefficients. Under linear weighting, to ensure a convex combination, the aggregation weights ω_m in (14) are defined as

$$\omega_m = \widehat{\beta}_m^2 / \sum_{j=1}^{M} \widehat{\beta}_j^2. \tag{15}$$

The choice (15) ensures that the weights ω_m are positive and sum to one. As a result, if the estimated response probabilities $\widehat{p}_k^{(1)}(\mathbf{x}_k), \ldots, \widehat{p}_k^{(M)}(\mathbf{x}_k)$, lie in the interval $(0,1]$, the aggregated probability will also lie in the interval $(0,1]$, which is a desirable property.

(2) *Exponential weighting.*

Let $\mathscr{L}(\cdot)$ denote a loss function. The exponential weights ω_m are given by

$$\omega_m := \frac{\exp\left\{-n \cdot T \cdot \mathscr{L}\left(\widehat{p}_m\right)\right\}}{\sum_{j=1}^{M} \exp\left\{-n \cdot T \cdot \mathscr{L}\left(\widehat{p}_j\right)\right\}}, \qquad m = 1, 2, \ldots, M, \tag{16}$$

where $T > 0$ is a hyper-parameter, often referred to as the temperature. When $T \longrightarrow 0$, the weights ω_m in (14) tend to be uniform, whereas $T \longrightarrow \infty$ will assign non-zero weights to the machine learning procedures exhibiting a small loss. For a discussion about the choice of the temperature, see Leung and Barron (2006) and Lecué (2007). We consider the following two loss functions:

(a) The misclassification error:

$$\mathscr{L}_{mis}\left(\widehat{p}_m\right) := \frac{1}{n} \sum_{k \in \mathcal{S}} \mathbb{1}_{\widehat{R}_m(\mathbf{x}_k) \neq R_k},$$

where $\widehat{R}_m(\mathbf{x}_k) := \mathbb{1}_{\widehat{p}_m(\mathbf{x}_k) \geqslant 1/2}$.

(b) The cross-entropy loss:

$$\mathscr{L}_{cross}\left(\widehat{p}_m\right) := \frac{1}{n} \sum_{k \in \mathcal{S}} \left\{-R_k \log\left(\widehat{p}_m(\mathbf{x}_k)\right) - (1 - R_k) \log\left(1 - \widehat{p}_m(\mathbf{x}_k)\right)\right\}.$$

To prevent the issue of overfitting, we consider a sample-splitting scheme that involves training/aggregation. More specifically, the aggregation procedures are implemented as follows:

Step 1: Shuffle the units in $D_S := \{(\mathbf{x}_k, R_k) \ ; \ k \in \mathscr{S}\}$ and select a fitting proportion $\rho \in (0;1)$. Let $n_{fit} := n \times \rho$. For simplicity, we assume that n_{fit} is an integer.

Step 2: Partition the data D_S into a fitting set, D_{fit}, of size n_{fit}, and an aggregation set D_{agg}, of size $n_{agg} := n - n_{fit}$.

Step 3: Fit the M models based on D_{fit} to obtain the estimated response probabilities $\widehat{p}_1 (\cdot, D_{fit}), \widehat{p}_2 (\cdot, D_{fit}), \cdots, \widehat{p}_M (\cdot, D_{fit})$.

Step 4: Determine the aggregation weights $\omega_m, m = 1, \ldots, M$, on the aggregation set D_{agg}, where ω_m is either given by (15) or (16). That is, the weights ω_m are computed with the loss $\mathscr{L}(\cdot)$ computed on D_{agg} with predictors $\widehat{p}_m (\cdot, D_{fit})$ fitted on D_{fit}, $m = 1, \ldots, M$.

Step 5: Output the aggregated response probabilities estimator $\widehat{p}_{agg} (\cdot \ , D_{fit}, D_{agg}) \equiv \widehat{p}_{agg}$ given by

$$\widehat{p}_{agg} := \sum_{m=1}^{M} \omega_m(D_{agg}) \cdot \widehat{p}_m (\mathbf{x}_k, D_{fit}), \qquad k \in \mathscr{S}_r.$$

To assess the performance of aggregation procedures, we used the same setup as the one described in Section 3.1. Again, we had $6 \times 4 = 24$ ignorable scenarios and $6 \times 2 = 12$ nonignorable scenarios. The aggregation procedures were based on the following $M = 5$ machine learning procedures: Xgboost1, cart50, rf3, knn2, and Score; see Section 3.1. The fitting proportion was set to 0 (without splitting) and to 0.7 (with splitting). The temperature T was set to $1/\mathbb{E}(n_{agg}) = 1/300$. We used both linear weighting, whereby the aggregation weights ω_m are given by (15) and exponential weighting based on both $\mathscr{L}_{mis}$ and $\mathscr{L}_{cross}$, whereby the weights ω_m are given by (16).

Tables 7 and 8 show some Monte Carlo descriptive statistics regarding the relative efficiency (RE) and the percent relative bias (RB) for the PSA estimator for the 24 ignorable scenarios and the 12 nonignorable scenarios, respectively. Tables 9 and 10 show the same Monte Carlo descriptive statistics corresponding to the Hájek estimator.

We begin by discussing the results pertaining to the PSA estimator. From Table 7, we note that the aggregation procedures based on exponential weighting performed almost as well as the best procedure, here rf3. For the 12 nonignorable nonresponse mechanisms, Table 8 shows that all the aggregation procedures outperformed each machine learning procedure individually. Similar observations can be made about the Hájek estimator; see Tables 9 and 10. In our experiments, exponential weighting was slightly more efficient than linear weighting. The effect of aggregating the predictors under splitting had limited effect in the case of exponential weighting. On the other hand, a careful examination of Tables 7-9 and 10 suggests that, in the of linear aggregation, the splitting procedure had a significant impact of the relative efficiency of the aggregated estimators in the worst-case scenarios. For instance, from Table 7, we note that linear weighting exhibited a value of RE of about 2130 in the worst case when splitting was omitted as opposed to 889 when splitting was performed. Tables 8-10 also exhibit the same phenomenon. Exponential weighting, however, does not follow this pattern: both the splitting and non-splitting versions exhibited similar performances in all our scenarios. The difference between the performance of linear with and without splitting seemed to be caused by significant differences in median absolute RB: for instance, in Table 7, the absolute RB in the worse case was equal to 22% for linear weighting with splitting, against 64% for linear weighting without splitting. Further research is needed to investigate this difference in behavior in more depth. Finally, except for Table 10, the best method with respect to the average RE, was an aggregation procedure. Overall, the performance of aggregation procedures seems promising. They allow for a data-driven "automatic" aggregation of several estimated response probabilities, and, as our results suggest, aggregation often leads to good efficiency in comparison to individual machine learning procedures.

ML procedure	**Min**	**Q1**	**Median**	**Q3**	**Max**	**Mean**
rf 3	158	208	227	338	1037	298
	(0.1)	(2.7)	(5.3)	(17.9)	(31.8)	(10.3)
Exponential weighting: $\mathscr{L}_{mis}$ (with splitting)	160	182	234	292	1143	294
	(0.5)	(4.0)	(11.7)	(20.5)	(38.4)	(13.2)
Exponential weighting: $\mathscr{L}_{mis}$ (without splitting)	159	182	235	292	1114	293
	(0.6)	(4.0)	(11.6)	(19.8)	(37.8)	(13.0)
Exponential weighting: $\mathscr{L}_{cross}$ (with splitting)	160	183	235	292	1169	296
	(0.5)	(4.0)	(11.3)	(19.4)	(37.3)	(12.8)
Exponential weighting: $\mathscr{L}_{cross}$ (without splitting)	159	182	236	292	1080	291
	(0.3)	(4.0)	(11.9)	(21.1)	(38.8)	(13.4)
xgb 1	172	210	245	332	775	288
	(0.8)	(2.9)	(7.6)	(16.9)	(23.8)	(9.7)
Linear weighting (with splitting)	170	207	246	329	889	308
	(0.0)	(2.2)	(6.9)	(14.6)	(22.0)	(8.6)
Linear weighting (without splitting)	159	181	250	349	2130	383
	(0.6)	(3.4)	(17.2)	(24.5)	(64.3)	(18.8)
knn 2	172	211	266	379	2192	410
	(3.1)	(6.3)	(18.2)	(31.6)	(66.9)	(21.1)
cart 50	170	226	348	515	901	381
	(0.0)	(0.5)	(3.0)	(5.1)	(25.9)	(4.4)
score	318	489	930	1329	11111	1712
	(0.6)	(3.9)	(14.0)	(21.8)	(44.3)	(15.7)

Table 7: Descriptive statistics of percent RE and percent RB (in parentheses) across the 24 ignorable scenarios: the propensity score estimator

ML procedure	**Min**	**Q1**	**Median**	**Q3**	**Max**	**Mean**
Exponential weighting: $\mathscr{L}_{cross}$ (without splitting)	150	573	765	1410	2335	1054
	(3.1)	(33.5)	(51.1)	(66.8)	(111.8)	(52.9)
Exponential weighting: $\mathscr{L}_{mis}$ (without splitting)	152	571	768	1423	2371	1060
	(3.3)	(34.2)	(51.6)	(66.4)	(111.9)	(53.1)
Exponential weighting: $\mathscr{L}_{mis}$ (with splitting)	157	576	773	1449	2425	1070
	(3.8)	(35.2)	(52.5)	(65.9)	(111.9)	(53.4)
Exponential weighting: $\mathscr{L}_{cross}$ (with splitting)	161	578	776	1465	2474	1078
	(4.2)	(35.2)	(53.1)	(65.5)	(112.1)	(53.7)
Linear weighting (without splitting)	158	555	792	1549	2913	1151
	(4.6)	(34.0)	(55.6)	(63.5)	(120.4)	(55.2)
Linear weighting (with splitting)	180	641	858	1333	2082	1046
	(7.4)	(33.9)	(51.9)	(68.5)	(108.3)	(53.4)
xgb 1	184	610	883	1348	2253	1080
	(7.8)	(34.0)	(52.3)	(70.5)	(113.4)	(54.9)
rf 3	204	762	904	1444	2351	1141
	(10.2)	(40.3)	(55.3)	(71.8)	(111.1)	(56.7)
knn 2	157	399	919	1711	3543	1260
	(2.4)	(24.9)	(58.7)	(64.5)	(128.6)	(56.3)
cart 50	139	783	971	1219	2185	1043
	(2.8)	(25.4)	(43.2)	(73.5)	(104.7)	(47.8)
score	767	1630	1816	3148	20307	4062
	(19.6)	(49.9)	(68.7)	(87.0)	(137.6)	(71.9)

Table 8: Descriptive statistics of percent RE and percent RB (in parentheses) across the 12 nonignorable scenarios: the propensity score estimator

ML procedure	**Min**	**Q1**	**Median**	**Q3**	**Max**	**Mean**
rf 3	173	200	215	334	558	277
	(0.2)	(3.1)	(5.2)	(14.1)	(35.8)	(9.7)
Exponential weighting: $\mathscr{L}_{mis}$ (with splitting)	177	198	220	330	534	273
	(0.6)	(3.2)	(5.8)	(13.9)	(38.9)	(10.8)
Exponential weighting: $\mathscr{L}_{cross}$ (with splitting)	178	199	220	331	535	273
	(0.7)	(3.3)	(5.9)	(14.3)	(39.3)	(10.9)
Exponential weighting: $\mathscr{L}_{mis}$ (without splitting)	175	197	220	326	535	272
	(0.6)	(3.1)	(5.6)	(13.6)	(38.5)	(10.6)
Exponential weighting: $\mathscr{L}_{cross}$ (without splitting)	174	196	221	323	535	272
	(0.6)	(3.1)	(5.5)	(13.3)	(38.1)	(10.5)
Linear weighting (with splitting)	175	200	223	324	493	271
	(0.2)	(2.5)	(5.7)	(11.0)	(26.5)	(7.9)
xgb 1	175	191	228	323	493	266
	(0.0)	(2.3)	(5.1)	(13.2)	(31.9)	(8.7)
Linear weighting (without splitting)	180	200	231	392	765	325
	(1.4)	(4.0)	(7.3)	(19.8)	(57.1)	(15.8)
knn 2	202	234	241	411	848	359
	(1.5)	(5.6)	(7.5)	(21.5)	(66.2)	(17.7)
cart 50	161	201	255	379	569	298
	(0.2)	(1.1)	(2.1)	(7.2)	(24.5)	(5.2)
score	224	351	532	736	4629	842
	(0.2)	(2.6)	(8.5)	(21.1)	(33.6)	(12.0)

Table 9: Descriptive statistics of percent RE and percent RB (in parentheses) across the 24 ignorable scenarios: the Hájek estimator

ML procedure	Min	Q1	Median	Q3	Max	Mean
cart 50	148	653	835	1051	2195	947
	(3.4)	(26.3)	(43.3)	(66.2)	(105.5)	(47.1)
Exponential weighting: $\mathscr{L}_{cross}$ (without splitting)	249	689	914	1281	2410	1108
	(13.4)	(34.0)	(53.4)	(70.9)	(115.3)	(56.3)
Exponential weighting: $\mathscr{L}_{mis}$ (without splitting)	255	702	916	1297	2419	1117
	(13.8)	(34.3)	(53.7)	(70.9)	(115.6)	(56.6)
Linear weighting (without splitting)	287	764	924	1404	2769	1240
	(16.0)	(37.8)	(55.2)	(70.1)	(129.9)	(60.1)
Exponential weighting: $\mathscr{L}_{mis}$ (with splitting)	273	731	924	1326	2420	1132
	(14.9)	(34.9)	(54.6)	(70.7)	(115.7)	(57.2)
Linear weighting (with splitting)	235	687	930	1258	2252	1065
	(12.3)	(32.0)	(53.3)	(70.5)	(110.6)	(54.8)
Exponential weighting: $\mathscr{L}_{cross}$ (with splitting)	288	761	932	1346	2433	1146
	(15.8)	(35.3)	(55.2)	(70.6)	(116.1)	(57.6)
xgb 1	231	669	948	1256	2457	1108
	(12.0)	(32.4)	(53.2)	(73.5)	(116.5)	(56.5)
rf 3	286	743	961	1423	2347	1150
	(16.3)	(36.5)	(56.3)	(68.6)	(113.7)	(57.4)
knn 2	391	813	985	1589	3379	1423
	(21.6)	(42.7)	(56.8)	(67.6)	(144.3)	(64.4)
score	656	1264	1628	2300	8356	2313
	(22.5)	(49.4)	(60.3)	(86.4)	(121.9)	(66.2)

Table 10: Descriptive statistics of percent RE and percent RB (in parentheses) across the 12 nonignorable scenarios: the Hájek estimator

5 Final remarks

In this paper, our primary focus was to evaluate the performance of various machine learning procedures within the context of unit nonresponse. Our findings revealed that among the tested methods, XGBoost, random forests, and Bayesian Additive Regression Trees (BART) emerged as the best procedures, showcasing their potential to reduce the potential nonresponse bias effectively. These procedures performed well in a wide variety of settings and, in the case of complex or nonlinear nonresponse mechanisms, exhibited significantly better performance than the commonly employed score method. Moreover, our study highlighted the effectiveness of aggregation methods in improving the overall performance of machine learning procedures.

In this work, we used aggregation procedures to combine the predictions from different machine learning algorithms. Aggregation procedures may also prove useful for combining the predictions produced by several architectures (that correspond to a set of hyperparameters) of a given machine learning procedure. For instance, we may combine several predictions obtained through the use of XGBoost, whereby each prediction is obtained with specific values of the learning rate, the depth of the tree, etc. The hope is that the resulting aggregated estimator will perform as well as the estimator obtained with the best architecture.

In practice, we often ask the question of which methods should be used or what the best method is for our specific scenario. As argued in Section 2, the best machine learning procedure is not necessarily the one that yields the best predictions. Selecting the best machine learning procedure or the best architecture requires an optimal criterion; e.g., the estimated mean square error of the adjusted estimator. This topic is currently under investigation.

Variance estimation for propensity score adjusted estimators constitutes another significant gap in the existing literature. This is currently under investigation and will be presented in a separate publication.

References

Beaumont, J-F. (2005). On the use of data collection process information for the treatment of unit nonreponse through weight adjustment. *Survey Methodology* **31**, 227–231.

Breiman, L. (2001). Random forests. *Machine Learning* **45**, 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1983). *Classification and regression trees.* Routledge.

Buckland, S. T., Burnham, K. P., and Augustin, N. H. (1997). Model selection: an integral part of inference. *Biometrics* **53**, 603–618.

Bunea, F., Tsybakov, A. B., and Wegkamp, M. H. (2006). Aggregation and sparsity via l_1 penalized least squares. *International Conference on Computational Learning Theory.* Berlin, Heidelberg : Springer Berlin Heidelberg, pp. 379–391.

Bunea, F., Tsybakov, A. B., and Wegkamp, M. H. (2007). Aggregation for Gaussian regression. *The Annals of Statistics* **35**, 1674–1697.

Chen, T. and Guestrin, C. (2016). Xgboost : A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

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

Eltinge, J. L., and Yansaneh, I. S. (1997). Diagnostics for formation of nonresponse adjustment cells, with an application to income nonresponse in the US Consumer Expenditure Survey. *Survey Methodology* **23**, 33–40.

Gelein, B. (2017). Handling missing data with superpopulation model, design-based approach and machine learning. Unpublished PhD thesis.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning.* Springer Series in Statistics. New York, USA.

Haziza, D., and Beaumont, J. F. (2007). On the construction of imputation classes in surveys. *International Statistical Review* **75**, 25–43.

Haziza, D., and Beaumont, J. F. (2017). Construction of Weights in Surveys: A Review. *Statistical science* **32**, 206–226.

Kern, C, Klausch, T., and Kreuter, F. (2019). *Survey Research Methods* **13**, 73–93.

Kim, J.K., Park, S., and Kim, K. (2019). A note on propensity score weighting method using paradata in survey sampling. *Survey Methodology* **45**, 451–463.

Leung, G., and Barron, A. R. (2006). Information theory and mixing least-squares regressions. *IEEE Transactions on Information Theory* **52**, 3396–3410.

Lecué, G. (2007). *Méthodes d'agrégation: optimalité et vitesses rapides.* PhD thesis, Paris 6.

Little, R. J. (1986). Survey nonresponse adjustments for estimates of means. *International Statistical Review* **54**, 139–157.

Little, R.J.A. and Vartivarian, S. (2005). Does weighting for nonresponse increase the variance of survey means? *Survey Methodology* **31**, 161–168.

Lohr, S., Hsu, V., and Montaquila, J. (2015). *Using classification and regression trees to model survey nonresponse.* Proceedings of the Survey Research Methods Section, American Statistical Association, Alexandria, pp. 2071–2085.

Nemirovski, A. (2000). *Topics in Non-parametric Statistics*, vol. 1738 of Ecole d'été de Probabilités de Saint-Flour 1998, Lecture Notes in Mathematics. Springer, N.Y.

Phipps, P. and Toth, D. (2012). Analyzing establishment nonresponse using an interpretable regression tree model with linked administrative data. *Annals of Applied Statistics* **6**, 772–794.

Quinlan, J. R. (1992) Learning with Continuous Classes. *Proceedings of Australian Joint Conference on Artificial Intelligence*, Hobart, pp. 343–348.

Quinlan, J. R. (1993). Combining instance-based and model-based learning. *In Proceedings of the tenth international conference on machine learning*, pp. 236–243.

Särndal, C.-E., Swensson, B., and Wretman, J. (1992). *Model-Assisted Survey Sampling*, Springer-Verlag, New York.

Tsybakov, A. B. (2003). Optimal rates of aggregation. *In Learning Theory and Kernel Machines: In the proceedings of the 16th Annual Conference on Learning Theory and 7th Kernel Workshop*, COLT/Kernel 2003, Washington, DC, USA, pp. 303-313. Springer Berlin Heidelberg.

Zeileis, A., Hothorn, T., and Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics* **17**, 492–514.