Mathematics and Computers in Simulation 225 (2024) 779–793

Contents lists available at ScienceDirect

Mathematics and Computers in Simulation

journal homepage: www.elsevier.com/locate/matcom

Original articles

Estimating response propensities in nonprobability surveys using machine learning weighted models

Ramón Ferri-García [a,*], Jorge L. Rueda-Sánchez [c], María del Mar Rueda [a], Beatriz Cobo [b]

[a] Department of Statistics and Operations Research, University of Granada, Avenida Fuentenueva, s/n, Granada, 18017, Spain
[b] Department of Quantitative Methods for Economics and Business, University of Granada, Campus Universitario de Cartuja, Granada, 18071, Spain
[c] Mathematics Institute of the University of Granada (IMAG), Calle Ventanilla, 11, 18001, Granada, Spain

ARTICLE INFO

Keywords:
Propensity score adjustment
Design weights
Nonprobability samples

ABSTRACT

Propensity Score Adjustment (PSA) is a widely accepted method to reduce selection bias in nonprobability samples. In this approach, the (unknown) response probability of each individual is estimated in a nonprobability sample, using a reference probability sample. This, the researcher obtains a representation of the target population, reflecting the differences (for a set of auxiliary variables) between the population and the nonprobability sample, from which response probabilities can be estimated.

Auxiliary probability samples are usually produced by surveys with complex sampling designs, meaning that the use of design weights is crucial to accurately calculate response probabilities. When a linear model is used for this task, maximising a pseudo log-likelihood function which involves design weights provides consistent estimates for the inverse probability weighting estimator. However, little is known about how design weights may benefit the estimates when techniques such as machine learning classifiers are used.

This study aims to investigate the behaviour of Propensity Score Adjustment with machine learning classifiers, subject to the use of weights in the modelling step. A theoretical approximation to the problem is presented, together with a simulation study highlighting the properties of estimators using different types of weights in the propensity modelling step.

1. Introduction

Novel information-gathering methods, such as online or smartphone surveys, have many advantages in terms of lower costs, higher response rates and broader questionnaire possibilities, making them attractive for practitioners considering a finite population. However, these surveys are usually self-administered, beyond the researcher's control, thus generating a nonprobability sample.

In a probability sampling design, all the individuals of the finite population of interest have a known or calculable probability of being included in the sample. If this condition does not apply, we have a nonprobability sample, which may be subject to selection bias, i.e. the sampled population may be different from the nonsampled population in a way that could affect the study variable of interest [1].

* Corresponding author.
E-mail address: rferri@ugr.es (R. Ferri-García).

https://doi.org/10.1016/j.matcom.2024.06.012
Received 6 November 2023; Received in revised form 6 June 2024; Accepted 18 June 2024
Available online 22 June 2024

Adjustment methods have been proposed to overcome or reduce the selection bias produced in nonprobability samples, but the application and effectiveness of any such method depends on the amount of auxiliary information available. Commonly, a probability sample from the same population is available, from which some auxiliary variables in common with the nonprobability sample can be measured. These are known as reference samples in this context, as they reflect the structure of the population of interest for the aforementioned auxiliary variables, enabling useful analysis even if the actual population values are unknown.

Such methods include Propensity Score Adjustment (PSA) [2], Propensity-Adjusted Probability Prediction [1], Kernel Weighting [3], Statistical Matching or Mass Imputation [4], and the combination of PSA and Mass Imputation known as Doubly Robust Estimators [5]. All of these methods are based on predictive modelling, but they differ regarding the variable that is predicted. Mass Imputation focuses on predicting the values of the variable of interest for individuals in the reference sample, while the remaining methods predict the probability of participating in the nonprobability sample, which is expressed as a binary classification problem (participation versus non-participation).

In using a reference sample, its sampling design must be taken into account, to ensure a correct, balanced representation is made of the population of interest. This sampling design is usually incorporated in linear estimators via design weights, which represent the inverse probability of an individual being selected in the reference sample. However, these weights are not always applied consistently in response propensity estimation, which can create problems, especially when nonparametric models are used for propensity prediction. In the present work, we study this issue and propose different approaches based on incorporating machine learning algorithms (XGBoost and Random Forest) into nonprobability survey sampling. Our main aims in this study are to determine the effect of weighting the algorithms (in the training step) used for propensity estimation, by training weighted algorithms leveraging the design weights available for the reference sample, and then to introduce a broader set of weighted algorithms for propensity estimation, as an alternative to logistic regression. We also provide empirical evidence of the performance of each approach in real life situations, as a valuable resource for researchers and practitioners. The rest of this paper is structured as follows: in Section 2, we introduce Propensity Score Adjustment, which is the base method for all other approaches based on propensity prediction, and discuss how design weights are used in this procedure. In Section 3, we examine the use of weighted machine learning algorithms for propensity estimation, which in our view is one of the main contributions of this paper. In Section 4, we describe the simulation study carried out to compare the modelling approaches considered, both with and without design weights, and under different scenarios. The simulation results are presented in Section 5. Finally, in Section 6 we discuss these results, summarise the main conclusions drawn and propose some recommendations for practitioners.

2. Propensity estimation

2.1. Propensity score adjustment

Let U be the population of interest, of finite size N, from which we wish to know a linear parameter of a variable of interest, y. This linear parameter can be the population mean, $\overline{Y}$, the population total, T_y, or the proportion of an attribute of interest (for example, the proportion of voters of a given political party in the population), P_y.

Let s_v be the nonprobability sample of size n_v, drawn from a potentially covered population U_{pc}, such that $U_{pc} \subseteq U$, following no sampling design, and let s_r be the reference probability sample of size n_r also drawn from U with design weights $d^r = 1/\pi^r$, where π^r is the probability of an individual being selected in s_r.

Let $R = 0, 1$ be the participation indicator where:

$$R_i = \begin{cases} 1 & i \in s_v \\ 0 & i \notin s_v \end{cases}, \quad i \in U \tag{1}$$

and let

$$\pi_i^v = Pr(i \in s_v | \mathbf{x}_i, y_i) = Pr(R_i = 1 | \mathbf{x}_i, y_i) = E_q[R_i | \mathbf{x}_i, y_i], \quad i \in U \tag{2}$$

where $\mathbf{x}$ is a set of auxiliary variables and the subscript q in the expectation refers to the model for the selection mechanism of the nonprobability sample, i.e. the propensity model, following the notation used in [5]. We call the π_i^v the propensity scores or propensities. These are unknown and require suitable model assumptions for valid estimation methods to be developed. On the other hand, $\mathbf{x}$ must have been measured for all individuals in s_r and s_v for the estimation to be performed. This set of variables can be decided according to a reference sample available before obtaining s_v, to ensure that they will be measured in s_v as well. Alternatively, they might be decided on the basis of the common variables between s_v and any probability sample that could be leveraged (for example, a government survey). Propensity estimation is more effective if these variables are associated with the selection mechanism and (especially) with the variable of interest, as shown in [6]. In this respect, we consider three basic assumptions [5]:

1. $\pi_i^v = Pr(R_i = 1 | \mathbf{x}_i), \quad i \in U$, similar to the missing at random assumption for missing data analysis.
2. $\pi_i^v > 0, i = 1, \ldots, N$, meaning that all units can be selected.
3. $R_1, \ldots, R_N$ are independent given $(\mathbf{x}_1, \ldots, \mathbf{x}_N), 1 = 1, \ldots, N$.

Assumption 1 is not very common in practice, as nonprobability surveys might have an unknown selection mechanism which is related to the variable of interest, violating the assumption. Assumption 2 is also violated when there is any kind of coverage error, which is also relatively common in nonprobability samples, especially when using new technologies for survey administration such as internet surveys or smartphone surveys, as some members of the target population will be unable to take part in the survey. The testability of these assumptions is difficult but not impossible: some measures of bias have been developed in literature that could be helpful for assessing Assumption 1 [7,8]. The coverage error in Assumption 2 could be assessed by the practitioner if it is known which segments of the population will be reached by the survey.

In a nonprobability survey, as π^v is not known (because there is no sampling design), the propensities have to be estimated in order to apply the usual Horvitz–Thompson or Hájek estimators. Propensity Score Adjustment (PSA) attempts to give an estimate, $\hat{\pi}^v$, of its value for each individual in the available samples using a model $m(\mathbf{x}, \lambda)$, where $\mathbf{x}$ is a set of covariates available in both s_v and s_r and λ is the vector of parameters of the model. The model m that is often considered in literature is logistic regression. The propensity scores are therefore estimated by pooling the non-probability sample s_v with the reference probability sample s_r and fitting a logistic regression to predict R_i^*, where $R_i^* = 1$ if $i \in s_v$ and 0 if $i \in s_r$. The estimator formula is:

$$\hat{\pi}^v = Pr(R_i^* = 1|\mathbf{x}_i) = \frac{exp(\lambda \mathbf{x}_i)}{1 + exp(\lambda \mathbf{x}_i)}, \qquad i \in s_r \cup s_v, \tag{3}$$

where λ is the vector of coefficients that minimise the logistic loss function. [9] proposed to fit a survey weighted logistic regression model. Some recent approaches involve machine learning classification algorithms for m, involving different estimators for propensities [10–12]. Estimators of linear parameters using PSA have also been developed, using the propensities to construct weights to be used in Horvitz–Thompson or Hajek estimators. The weights to be used in those estimators can be calculated using several transformations for propensities, including:

- The usual inverse probability weighting: $w_i^{IPW} = 1/\hat{\pi}_i^v, i \in s_v$.
- The modification of inverse probability weighting proposed in [13] which assumes that the individuals in the nonprobability sample do not belong to the target population of the reference sample:

$$w_i^{IPWM} = \frac{1 - \hat{\pi}_i^v}{\hat{\pi}_i^v}, \quad i \in s_v. \tag{4}$$

- Propensity stratification weighting. The propensities are classified into strata of roughly the same size, with individuals in each stratum having similar propensities. The estimation can then be performed by multiplying the original weights of the nonprobability sample (if any) by a correction factor that takes into account the design weights of the reference sample [14], such that the final weights are $w_i^{Strat1} = f_c \cdot \frac{N}{n_v}, i \in s_v$; this approach with the correction factor is further developed in Section 2.2. Another option is to calculate the mean of the propensities of each strata and to use the result as the propensity of each individual of a given stratum, thus obtaining the weights via inverse probability weighting [9].

$$w_i^{Strat2} = 1/\overline{\hat{\pi}^v}_g, \quad i \in s_v, g \ni i \tag{5}$$

where $\overline{\hat{\pi}^v}_g$ is the mean propensity of stratum g to which individual i belongs. These approaches avoid very extreme propensities that would increase the variance of the final estimates.

2.2. Design weights in propensity score adjustment

The use of design weights, available for the reference sample, in model fitting has been considered in various ways in the literature on PSA. Many authors fit the propensity estimation models, m, and do not make use of weights. However, according to [9], this approach gives biased estimates of linear parameters because the propensities are inflated only to the level of the combined sample $s_v \cup s_r$, rather than that of the target population.

Sometimes the design weights can be incorporated afterwards; [14] developed an approach in which the estimated propensities $\hat{\pi}^v$ are sorted and partitioned into C strata (ideally, $C = 5$ following [15,16]). A correction factor f_c is then defined for a given stratum, c, based on the design weights of both the probability and the nonprobability sample:

$$f_c = \frac{\sum_{k \in s_r^c} d_k^r / \sum_{k \in s_r} d_k^r}{n_v^c / n_v}, \tag{6}$$

where n_v^c is the number of individuals from the nonprobability sample that belong to the cth stratum, and s_r^c and s_v^c are the subset of individuals from the reference and the nonprobability sample respectively that belong to the cth stratum. The final weights of the nonprobability sample to be applied in linear estimators are defined as:

$$w_i^{Strat1} = f_c \cdot \frac{N}{n_v}, \quad i \in s_v, c \ni i \tag{7}$$

Other studies [9,17] have considered the use of weighted models, where the individuals from the nonprobability sample are assigned unitary weights, and the individuals for the probability sample are assigned their design weight with a correction factor:

$$w_i^m = \begin{cases} 1 & i \in s_v \\ d_i^r \left(1 - \frac{n_r}{\sum_{j \in s_r} d_j^r}\right) & i \in s_r \end{cases} \tag{8}$$

On the other hand, [5] has observed that this approach may not lead to a consistent estimator of the parameters in the weighted model, as the score functions are not approximately unbiased.

To overcome this problem, these authors developed a consistent and unbiased estimator of linear parameters by maximising the pseudo-log-likelihood function with respect to the vector λ of parameters of the model (for example, the regression coefficients of a logistic regression):

$$\tilde{l}(\lambda) = \sum_{i \in s_v} log\left(\frac{m(\mathbf{x}_i, \lambda)}{1 - m(\mathbf{x}_i, \lambda)}\right) + \sum_{i \in s_r} d_i^r log\left(1 - m(\mathbf{x}_i, \lambda)\right) \tag{9}$$

If a logistic regression model is used to estimate propensities, the function becomes

$$\tilde{l}(\boldsymbol{\beta}) = \sum_{i \in s_v} \mathbf{x}_i^T \boldsymbol{\beta} + \sum_{i \in s_r} d_i^r log\left(1 + exp(\mathbf{x}_i^T \boldsymbol{\beta})\right) \tag{10}$$

If we consider the approach from [9] as in Eq. (8), the pseudo-log-likelihood function would be equal to

$$\tilde{l}(\lambda) + \sum_{i \in s_v} log\left(1 - \pi_i^v\right) - \frac{n_v}{\sum_{i \in s_r} d_i^r} \sum_{i \in s_r} d_i^r log\left(1 - \pi_i^v\right)$$

If the correction factor for individuals in the reference sample is omitted, i.e.,

$$w_i^{LR} = \begin{cases} 1 & i \in s_v \\ d_i^r & i \in s_r \end{cases}, \tag{11}$$

the function would be equal to

$$\tilde{l}(\lambda) + \sum_{i \in s_v} log\left(1 - \pi_i^v\right)$$

If the sampling fraction of the nonprobability sample is sufficiently small ($n_v/N \to 0$), the above function is approximately equal to $\tilde{l}(\lambda)$ as the additional term converges to zero. Therefore, this approach could be used as an approximation for the pseudo-log-likelihood function, as noted in [6,18]. Nevertheless, the original one is preferable as it uses the equations developed in [5], thus ensuring consistent estimators for λ are obtained, while the modified version may not reach exactly the same solution (although it will be very close if n_v/N is sufficiently small).

2.3. Tree-based inverse propensity weighting

An alternative to estimate propensities using decision trees was developed in [19], in an approach that also takes into account sampling or calibration weights from s_r. This method obtains a prediction using decision trees, which can be defined as a set of rules organised in a hierarchical structure, starting from the whole dataset and ending in a smaller subset of individuals who are given a prediction according to the rule applicable.

The Tree-based Inverse Propensity Weighting (TrIPW) algorithm fits a decision tree, using a modification of the Classification And Regression Tree (CART) algorithm, on individuals from s_v using the data from s_r in an auxiliary way to predict the value of the participation indicator R, such that any individual i in the population, and more precisely in the nonprobability sample, can be assigned to a terminal node, which could be seen as a population stratum. The propensities are obtained by dividing the number of s_v members classified in a node by the projected population that would fit on that node (estimated from the sum of the design weights of the probability sample members that have been assigned to that node).

3. Weighted machine learning algorithms for propensity estimation

In several contexts, especially those that involve machine learning (ML) classification algorithms, the combination of the probability and the nonprobability sample can lead to class imbalance problems if any of the samples is significantly larger than the other one. This could easily be the case if a nonprobability large dataset is being used, and it is particularly troublesome in ML as many algorithms assume that the distribution of the target variable is uniform across all classes. Class imbalance is usually tackled in ML contexts by creating artificial individuals or by randomly removing some individuals from the dataset. The rationale behind these methods is that many ML algorithms typically used in prediction do not allow the user to assign weights to the individuals in the training samples. However, this gap is closing with the development of weighted versions of some of the most popular algorithms.

If the model allows the specification of weights for individuals, one way to tackle class imbalance is to train weighted models using unitary weights for individuals in the nonprobability sample, and the quotient between the two sample sizes for individuals of the probability sample, making the combination of samples completely balanced.

$$w_i^{mBAL} = \begin{cases} 1 & i \in s_v \\ \frac{n_v}{n_r} & i \in s_r \end{cases} \tag{12}$$

However, the aforementioned methods are only focused on balancing the sample to fulfil the requirements of uniform distribution imposed by the ML algorithms; they do not take into account the sampling design of the probability sample. In the best case scenario, they would only be able to inflate the estimations to the level of the pooled sample ($s_v \cup s_r$), which is the same issue acknowledged by [9]. For this reason, we propose some alternatives, based on the characteristics of the weighted ML algorithms available in the

current state-of-art. These alternatives use the design weights of the probability reference sample to better represent the structure of the target population in the optimisation procedures, instead of just balancing the pooled sample, thus obtaining consistent estimates of the propensities by the same principle developed in [5] and explained in Section 2.2. These design weights can be incorporated using implementations of ML algorithms that allow the use of weights; these implementations normally give more or less importance to the residuals of the model depending on the weight of the individuals, but when no optimisation is performed (for example, with bagging algorithms such as Random Forests) the weights are used in resampling to give larger or smaller selection probabilities to each individual, which can be seen as an attempt to reproduce the behaviour of the complete population. In both cases, as long as the design weights are correctly specified, the results should be similar to those that would be obtained if the entire population was used in the modelling step, which is the desirable case. These properties are further explained in the following subsections.

3.1. Random forests

The Random Forests [20] technique is based on combining of multiple decision trees for prediction in order to reduce overfitting. In the case of propensity estimation (binary classification problem), the prediction in each case is carried out by averaging the number of times that an individual is classified as belonging to the nonprobability sample (that is, $R^* = 1$) through a set of m trees. For each tree, a bootstrapped simple random sample with replacement (SRSWR) of individuals from the input dataset is selected, s_j with size n_j, $j = 1, \dots, m$, and when fitting the tree the algorithm considers only a randomly selected subset (of fixed size) of predictors drawn from the whole set of predictors available. The use of partial information for each tree is why they are known as weak classifiers.

A formula for the propensity can then be considered, following notation from [10]:

$$\hat{\pi}_i^v = \frac{\sum_{j=1}^m \phi_j(\mathbf{x}_i)}{m}, \quad i \in s_v \cup s_r, \quad \phi_j(\mathbf{x}_i) = \begin{cases} 1 & \mathbf{x}_i \in \mathcal{L}_j^1 \\ 0 & \mathbf{x}_i \in \mathcal{L}_j^0 \end{cases}, \tag{13}$$

where $\mathcal{L}_j^1$ and $\mathcal{L}_j^0$ represent the set of terminal nodes of the jth decision tree, ϕ_j, where individuals from the nonprobability sample are majority and minority, respectively.

A more interesting approach developed in [21] considers the proportion of successes (in our case, individuals in the nonprobability sample) observed in the terminal nodes of each tree, instead of taking binary values depending on the majority/minority of successes:

$$\hat{\pi}_i^v = \frac{\sum_{j=1}^m \phi_j(\mathbf{x}_i)}{m}, \quad i \in s_v \cup s_r, \quad \phi_j(\mathbf{x}_i) = \frac{\#\left(l_j(i) \cap s_v\right)}{\#l_j(i)}, \tag{14}$$

where $l_j(i)$ is the terminal node where the ith individual falls according to the jth tree of the forest.

If the unweighted pooled sample is used to fit the Random Forest, the probability of an individual in s_j being in s_v will be

$$P(s_{ji} \in s_v) = \frac{n_v}{n_v + n_r}, \quad i = 1, \dots, n_j$$

However, if the complete population U is used to fit the Random Forest, the probability will be

$$P(s_{ji} \in s_v) = \frac{n_v}{n_v + N - n_v} = \frac{n_v}{N}, \quad i = 1, \dots, n_j$$

The immediate consequence of this is that the probabilities when fitting the Random Forest using the unweighted pooled sample will be inflated to $s_v \cup s_r$ instead of U, which is a similar issue to that pointed out by [9]. To tackle this issue, we propose drawing the bootstrapped samples $s_j, j = 1, \dots, m$ using the weighted pooled sample according to the design weights of each individual, meaning that the case weights of the Random Forest will be

$$w_i^{RF} = \begin{cases} 1 & i \in s_v \\ d_i^r - \frac{n_v}{n_r} & i \in s_r \end{cases}, \quad i \in s_v \cup s_r \tag{15}$$

After this adjustment, if we draw a bootstrapped unequal probability sample with replacement from $s_v \cup s_r$, with probabilities proportional to w^{RF}, the probability of an individual in s_j being in s_v will be

$$P(s_{ji} \in s_v) = \begin{array}{l} \dfrac{n_v}{n_v + \sum_{k \in s_r} \left(d_k^r - \frac{n_v}{n_r}\right)} = \dfrac{n_v}{n_v + \sum_{k \in s_r} d_k^r - n_v} = \\ \dfrac{n_v}{n_v + \hat{N} - n_v} = \dfrac{n_v}{\hat{N}} \end{array}, \quad i = 1, \dots, n_j. \tag{16}$$

where $\hat{N}$ is a consistent estimator for the population size obtained from the probability sample. The term n_v/n_r in w^{RF} becomes negligible as the sampling fraction n_v/N decreases. Considering that s_r is a probability sample, this form of weighting the bootstrapped samples should provide similar trees to those obtained using the complete target population (instead of the pooled sample), but most importantly, it should contribute to inflating the nonprobability sample propensities to the target population.

3.2. Extreme gradient boosting

The Extreme Gradient Boosting (XGBoost) method [22] is a regression and classification algorithm which has gained some acceptance in recent years, and is also used in the context of nonprobability samples [23]. It is based on the Gradient Boosting Machine (GBM) algorithm [24], and the prediction works in a similar way to that found with Random Forests, with the final values being derived from the set of predictors:

$$\hat{\pi}_i^v = \phi(\mathbf{x}_i) = \sum_{j=1}^{m} f_j(\mathbf{x}_i), \quad i \in s_v \cup s_r, \tag{17}$$

where $f_j(\mathbf{x}_i)$ represents the score obtained in the mth tree of the set. This score should not be interpreted as a prediction given from a decision tree, but as a weighted score, $\omega_j(\mathbf{x}_i)$, that also reflects the importance of each tree. In GBM, we consider the following loss function:

$$\mathcal{L}(\phi) = \sum_{i \in s_v \cup s_r} l(\hat{R}_i, R_i) + \sum_{j=1}^{m} \Omega(f_j), \quad \Omega(f) = \gamma T + \lambda \frac{1}{2} \|\omega\|^2, \tag{18}$$

where T is the number of leaves in the tree f, γ and λ are regularisation parameters that control over-fitting, and where l is a differentiable convex function which measures the difference between the predicted values, $\hat{R}$, and the real values, R. This objective function is then minimised using iterative methods, such as the Gradient Boosting Tree. In this approach, the objective function of the iteration t can be expressed as

$$\mathcal{L}^{(t)} = \sum_{i \in s_v \cup s_r} l(R_i, \hat{R}_i^{t-1} + f_t(\mathbf{x}_i)) + \Omega(f_t), \tag{19}$$

where $\hat{R}_i^{t-1}$ is the value predicted for individual i in the iteration $t-1$. Further details on the optimisation procedure for this function are given in [23]. The additional contribution of XGBoost over GBM is the inclusion of shrinkage, to limit the importance of each tree in the iterative optimisation, and the use of strategies to find split points for candidate trees, among other techniques [22].

As in Random Forests, the use of the unweighted blended sample $s_v \cup s_r$ could result in predicted probabilities that might be inflated to the size of the sample, instead of the population. If we were using the true model fitted with the whole population, the loss function could be expressed as:

$$\mathcal{L}(\phi) = \sum_{i=1}^{N} l(\hat{R}_i, R_i) + \sum_{j=1}^{m} \Omega(f_j), \tag{20}$$

Given that propensity scoring is a binary classification problem for machine learning algorithms, the function l is the binary loss, as implemented in the XGBoost algorithm by [22]. This function must then be minimised:

$$l(\hat{R}_i, R_i) = -\left(R_i log(\hat{R}_i) + (1 - R_i) log(1 - \hat{R}_i)\right) = R_i log\left(\frac{1 - \hat{R}_i}{\hat{R}_i}\right) - log(1 - \hat{R}_i) \tag{21}$$

Hence, Eq. (20) can be expressed as follows:

$$\begin{aligned} \mathcal{L}(\phi) = & \sum_{i=1}^{N} R_i log\left(\frac{1 - \hat{R}_i}{\hat{R}_i}\right) - log(1 - \hat{R}_i) + \sum_{j=1}^{m} \Omega(f_j) \\ & = \sum_{i \in s_v} log\left(\frac{1 - \hat{R}_i}{\hat{R}_i}\right) - \sum_{i=1}^{N} log(1 - \hat{R}_i) + \sum_{j=1}^{m} \Omega(f_j). \end{aligned} \tag{22}$$

This setup is very similar to the pseudo-log-likelihood optimisation presented in [5]. As $U - (s_v \cup s_r)$ remains unobserved, an estimator is needed to compute an unbiased loss function. To do so, we could use the following Horvitz–Thompson estimator:

$$\mathcal{L}^*(\phi) = \sum_{i \in s_v} log\left(\frac{1 - \hat{R}_i}{\hat{R}_i}\right) - \sum_{i \in s_r} d_i^r log(1 - \hat{R}_i) + \sum_{j=1}^{m} \Omega(f_j). \tag{23}$$

However, it might not be possible to apply this solution because it requires the design of a new loss function for XGBoost (instead of using the one already implemented), which could be a complex task. If this exact solution cannot be adopted, we could apply the following approximation based on weighting the algorithm, an approach that is already implemented in some versions of XGBoost (such as the xgboost library in R [25]), which is approximately equal to the previous equation:

$$\mathcal{L}^{**}(\phi) = \sum_{i \in s_r \cup s_v} w_i^{XGB} l(\hat{R}_i, R_i) + \sum_{j=1}^{m} \Omega(f_j), \quad w_i^{XGB} = \begin{cases} 1 & i \in s_v \\ d_i^r & i \in s_r \end{cases} \tag{24}$$

If l is the binary loss function, the following equality holds:

$$\mathcal{L}^{**}(\phi) = \mathcal{L}^*(\phi) + \sum_{i \in s_v} log(1 - \hat{R}_i) \tag{25}$$

As noted in Section 2.2, both objective functions will be approximately equal if the sampling fraction of s_v is sufficiently small.

4. Simulations

Two simulations were conducted to evaluate the behaviour of these weighting procedures. In each simulation, the population mean of the target variables was estimated using PSA with logistic regression, Random Forests, XGBoost, and TrIPW to predict propensities in three different ways: without using the design weights of the probability sample (unweighted models), using the balancing weights w^{mBAL} described in Eq. (12), and using the design weights as described for each method (w^{LR}, w^{RF} and w^{XGB}) in the previous section. For TrIPW, only one case was considered (using the design weights) given that the method is not intended to work in the other two cases. The propensities were transformed into weights using all of the four methods described in Section 2.2. For each simulation, the relative bias (RB) and efficiency ratio (weighted to unweighted estimation) of each estimator was obtained:

$$RB = \frac{|\sum_{i=1}^{B} \hat{\bar{y}}_i^k/B - \overline{Y}|}{|\sum_{i=1}^{B} \hat{\bar{y}}_i^{Unw}/B - \overline{Y}|} \qquad Eff.\ ratio = \frac{\sum_{i=1}^{B}(\hat{\bar{y}}_i^k - \overline{Y})^2}{\sum_{i=1}^{B}(\hat{\bar{y}}_i^{Unw} - \overline{Y})^2}, \tag{26}$$

where B is the number of iterations of the simulation, $\hat{\bar{y}}_i^k$ is the estimate of the mean of y provided by method k in the ith iteration, and $\hat{\bar{y}}_i^{Unw}$ is the estimate of the mean of y provided by the unweighted estimator ($\hat{\bar{y}}^{Unw} = \sum_{i \in s_v} y_i/n_v$) in the ith iteration.

4.1. Synthetic data simulation

The first simulation used a population of size $N = 500{,}000$ created from synthetic data, with one target variable (y) and ten predictors ($x_1, x_2, \ldots, x_{10}$), with the following distributions:

$$x_1, x_2, x_4 \sim N(10, 2) \quad x_3, x_5, x_9 \sim B(0.5) \quad x_{10} \sim N(0, 1) \tag{27}$$

$$x_{6i} \sim N(10 + 2 \cdot x_{1i} + x_{3i}, 2) \quad x_7 \sim B(0.5 + 0.2 \cdot x_{1i} + 0.2 \cdot x_{2i}) \quad x_8 \sim N(10 + x_{3i}, 2) \tag{28}$$

$$y_i \sim N(10 + 2(x_{1i} + x_{2i} + x_{4i}) + 0.2(x_{3i} + x_{5i}), 1) \tag{29}$$

with $i = 1, 2, \ldots, N$. The ten predictors ($x_1, x_2, \ldots, x_{10}$) were observed in both the probability and the nonprobability samples to be drawn in each simulation run. These predictors were designed to represent a situation where some of the variables are related to the variable of interest and the selection mechanism (meaning that they should be included in the propensity estimation model) while others are unrelated (and therefore should be removed from the model). At the same time there is some level of multicollinearity among predictors. Accordingly, a model that incorporates all ten predictors, which is a common choice in propensity estimation (in order to use all the auxiliary variables available), is not the correct choice, and therefore variable selection must be performed. Thus, ML algorithms are employed to internally select appropriate variables according to their importance.

In the present case, the simulation was conducted with 1000 iterations, in each of which a nonprobability sample was drawn with an unequal probability sampling using the systematic method [26], implemented in the *UPsystematic* function from the *sampling* package in the statistical software R [27], considering inclusion probabilities π_v. The probabilities were calculated in the first step as follows:

$$ln\left(\frac{\pi_{vi}}{1 - \pi_{vi}}\right) = -10 + \frac{2}{9} y_i + x_{1i} + x_{2i} + 0.1 \cdot x_{3i} + x_{4i} + 0.1 \cdot x_{5i}, \quad i \in U \tag{30}$$

In the second step, each probability was corrected so that the sum of probabilities equalled n_v, multiplying each probability by $n_v / \sum_{i=1}^{N} \pi_{vi}^k, k = 1, 2, 3$. This correction was made in order to have fixed size samples, as in that case the sum of all inclusion probabilities must be n_v. The sizes $n_v = 500$ and $n_v = 5{,}000$ were established for the nonprobability sample.

Three probability samples of size $n_r = 500$ were also drawn in each iteration, following three different sampling schemes with sampling weights d^{r1}, d^{r2} and d^{r3}.

- The first sampling scheme was simple random sampling without replacement (SRSWOR) from the whole population, meaning that $d_i^{r1} = \frac{N}{n_r} = \frac{500000}{500} = 1000, \quad i \in s_r$ and $Deff = 1$ for this design.
- The second sampling scheme was a stratified sampling considering three strata U_1, U_2, U_3 that depended on the value of three predictors: $U_1 = \{i \in U / x_{1i} + x_{2i} + x_{4i} = 3\}$, $U_2 = \{i \in U / x_{1i} + x_{2i} + x_{4i} = 0\}$, $U_3 = \{i \in U / 0 < x_{1i} + x_{2i} + x_{4i} < 3\}$. The sample was allocated using Neyman's minimum variance method. The design effect for the estimation of the mean of y for this sampling design is $Deff = 0.479$, meaning that this design is more efficient than SRSWOR.
- The third sampling scheme was another stratified sampling considering three strata U_4, U_5, U_6 that depended on the value of two predictors: $U_4 = \{i \in U / x_{9i} = 0 \cap x_{10i} < 0\}$, $U_5 = \{i \in U / x_{9i} = 0 \cap x_{10i} \geq 0\}$, $U_6 = \{i \in U / x_{9i} = 1\}$. The allocation of the sample was arbitrary, with $n_4 = n_5 = 200$ and $n_6 = 100$. The design effect for this sampling design is $Deff = 1.568$, meaning that this design is less efficient than SRSWOR.

For each design of s_r, $\overline{Y} = \sum_{i=1}^{N} y_i/N$ was estimated using PSA as described at the beginning of Section 4, with all ten predictors as input variables of the models. This simulation has some similarities to a unit nonresponse study, except that in the unit nonresponse case a sampling design is available and the propensities are defined only for the sample, while in this case there is no sampling design and the propensities are defined for the whole population, as they are intended to replace the (nonexistent) sampling design.

Table 1
Summary table of the results of relative bias in both simulations provided by each combination of algorithms and weighting strategy for each one of them, across all scenarios considered in both simulations and in each one of them separately. Mean = mean relative bias across all scenarios. Median = median relative bias across all scenarios. Best = number of times (scenarios) each combination of algorithm and weighting strategy produced the smallest relative bias, or produced a relative bias less than 10% greater than the minimum relative bias for the same scenario.

Method	Both simulations			Synthetic data			Real-world data		
	Mean	Median	Best	Mean	Median	Best	Mean	Median	Best
Logistic regression (unweighted)	0.78	0.64	1	0.49	0.51	1	0.92	0.71	0
Logistic regression (balancing weights)	0.72	0.59	1	0.43	0.51	1	0.84	0.66	0
Logistic regression (design weights)	0.58	0.54	2	0.39	0.36	0	0.68	0.71	2
TrIPW	0.66	0.53	2	0.47	0.37	0	0.76	0.55	2
Random Forest (unweighted)	0.83	0.81	1	0.59	0.74	1	0.95	0.81	0
Random Forest (balancing weights)	0.76	0.70	3	0.54	0.62	1	0.84	0.70	2
Random Forest (design weights)	0.56	0.57	2	0.68	0.77	0	0.50	0.51	2
XGBoost (unweighted)	0.78	0.80	2	0.58	0.67	1	0.87	0.80	1
XGBoost (balancing weights)	0.71	0.68	0	0.52	0.55	0	0.78	0.68	0
XGBoost (design weights)	0.39	0.30	5	0.18	0.17	1	0.50	0.62	4

4.2. Real-world data simulation

The second simulation was performed using microdata from the 2012 edition of the Spanish Life Conditions Survey. This simulation had 1000 iterations; in each one, a reference and a convenience sample were extracted from a pseudopopulation of $N = 1,000,000$ after bootstrapping the original filtered dataset of $n = 28{,}210$ individuals ($n = 27{,}949$ after filtering individuals with any missing data in any variable). Three different sampling designs were considered for the reference sample, with size $n_r = 2,000$: a SRSWOR design, a stratified cluster sampling with Autonomous Communities as strata and households as clusters, and an unequal probability sampling with probabilities proportional to income. This approach enabled us to evaluate the behaviour of the estimator under complex survey designs. The convenience samples were drawn following an unequal probability sampling scheme, using the generalisation of the successive sampling without replacement implemented in the *sample* function in the statistical software R, with sizes $n_v = 2,000$ and $n_v = 6,000$ and probabilities

$$ln\left(\frac{\pi_{vi}}{1-\pi_{vi}}\right) \propto \begin{array}{l} 2 \cdot \text{PC} - 0.2 \cdot \text{Male} - 0.01 \cdot \text{Age (years)} - \\ 0.2 \cdot \text{Medium Density} - 0.4 \cdot \text{Low Density} \end{array}, \tag{31}$$

with $i = 1, 2, \ldots, N$. The variable PC represents having a computer at home (or not), while Medium Density and Low Density refer to the type of area the individual lives in (medium or low population density respectively). Age (in years) and male gender (or not) represent the age and gender variables. This formula is intended to capture the behaviour of respondents in real nonprobability surveys, although we do not expect these variables to be strongly related to the target variables, and so what is represented here is likely to be a Missing At Random (MAR) situation. The population parameters calculated were the mean home expenses and the proportion of households with a car, both of which are related to purchasing power and hence income. Again, this simulation has some similarities to a unit nonresponse study but with the differences highlighted in the previous section. The covariates used in PSA were five variables related to economic deprivation, four related to material deprivation and nineteen related to working conditions (including twelve reflecting the employment status of the individual in each month of the previous year).

5. Results

Table A.1 presents the results obtained for the estimation of $\overline{Y}$ in the synthetic data simulation with sample sizes $n_v = 500$ and $n_v = 5,000$. For the real-world data simulation, the results for the estimation of the population proportion of households without a car and the mean home expenses are shown in Tables A.2 and A.3 respectively.

Given the vast amount of data produced by the simulation, this section includes several tables and figures to summarise and clarify these results. Tables 1 and 2 present the relative bias and efficiency ratio results, respectively, jointly for the two simulations and also for each one separately, by means of their measures of central tendency (mean and median) and the number of times each method (algorithm and weighting strategy) produced the best result for a given scenario (considering dataset, size of n_v and sampling design of s_r) or was less than 10% greater than the best result.

The summary tables show that in general the proposed adjustments with Random Forest and XGBoost worked very well, producing mean and median relative biases and efficiency ratios below 1 (which means that they performed better than the unweighted estimator), especially when these algorithms were weighted using the approach that involves design weights (w^{LR}, w^{RF} and w^{XGB}), although the performance was fairly heterogeneous across the simulations. For instance, in the synthetic data simulation none of the methods consistently provided the best result in terms of bias (although by far the smallest mean was provided by XGBoost with design weights), but the XGBoost and Random Forests methods provided the best efficiency ratio result in most of the situations considered (4 out of 6 scenarios, 2 for each algorithm), with a mean and median efficiency ratio of almost 0 in the case of XGBoost. In the real-world data simulation too, no single approach was always best in terms of relative bias (although

Table 2
Summary table of the results of efficiency ratio in both simulations provided by each combination of algorithms and weighting strategy for each one of them, across all scenarios considered in both simulations and in each one of them separately. Mean = mean efficiency ratio across all scenarios. Median = median efficiency ratio across all scenarios. Best = number of times (scenarios) each combination of algorithm and weighting strategy produced the smallest efficiency ratio, or produced an efficiency ratio less than 10% greater than the minimum efficiency ratio for the same scenario.

Method	Both simulations			Synthetic data			Real-world data		
	Mean	Median	Best	Mean	Median	Best	Mean	Median	Best
Logistic regression (unweighted)	2.07	1.12	1	0.34	0.26	1	2.94	1.52	0
Logistic regression (balancing weights)	1.47	0.93	1	0.24	0.26	1	1.94	1.16	0
Logistic regression (design weights)	0.74	0.56	1	0.17	0.14	0	1.03	0.78	1
TrIPW	0.93	0.33	2	0.26	0.14	0	1.27	0.34	2
Random Forest (unweighted)	1.02	0.72	0	0.48	0.57	0	1.29	0.72	0
Random Forest (balancing weights)	0.91	0.56	2	0.42	0.42	1	1.09	0.56	1
Random Forest (design weights)	0.44	0.36	7	0.56	0.63	1	0.39	0.35	6
XGBoost (unweighted)	0.92	0.66	3	0.46	0.48	0	1.15	0.66	3
XGBoost (balancing weights)	0.83	0.50	1	0.38	0.38	0	1.00	0.50	1
XGBoost (design weights)	0.28	0.20	3	0.06	0.04	2	0.39	0.40	1

XGBoost with design weights provided the best performance in 4 out of 12 scenarios). In terms of efficiency, however, Random Forest with design weights provided the best results in 6 of the 12 scenarios considered, although XGBoost also performed well, especially when it was not weighted. The differences between these modelling approaches as regards their performance might be related to the characteristics of the two datasets: the synthetic dataset contains mainly linear relationships with a smaller number of covariates, while the real-world dataset may contain more non-linear ones and a larger number of covariates (many of them categorical), a context in which Random Forests might be more suitable.

The results, however, can also vary depending on the size of the nonprobability sample, n_v, and the type of transformation applied in calculating the final weights of the estimators (w^{IPW}, w^{IPWM}, w^{Strat1}, w^{Strat2}). Fig. 1 shows the boxplots of the efficiency ratio results for each predictive algorithm and weighting strategy depending, on the one hand, on the simulation and the nonprobability sample size, and on the other hand, on the transformation used for the final weights. In both simulations, the improvement provided by adjustment methods is slightly greater when the sample sizes are not balanced, perhaps because the unbalanced cases considered are those where the nonprobability sample size was larger, leading to smaller sampling errors. In addition, the variability of the results is noticeably smaller when the algorithms are weighted in the training step with design weights instead of balancing weights or no weights at all, which generate wider boxes. This could be a consequence of elevating the estimates to the actual population size, which might swamp the effect of any other choices. Although the performance of each transformation for weighting seems to be similar across the four approaches, there is an important divide: while w^{IPW} and w^{Strat2} benefit from using design weights in the training of the algorithms, the performance of w^{IPWM} and w^{Strat1} does not differ across weighting strategies in the training step, except for providing a larger or smaller variability in some situations. The former two methods rely more strongly on the results of the propensity estimation step, while the latter two take into account other elements (such as design weights via a correction factor or the sum of the values of the variable of interest in the nonprobability sample).

For a fairer comparison of weighting and transformation approaches (as the boxplots show the aggregated results), we conducted an analysis based on the raw differences in relative bias and efficiency ratio for the results of a given method applied in a given simulation and dataset when the weighting strategy or the type of transformation was changed. We then considered how often each approach was the best, versus all other possibilities, and how often all approaches produced approximately the same results (less than 10% of difference among all of them). By doing so, we directly compared each approach with its counterfactual. In this respect, Table Table 3 shows that weighting the algorithms using design weights of the probability sample (w^{LR}, w^{RF} and w^{XGB} approaches) clearly produced the best results if the propensities were transformed into weights using the IPW approach, w^{IPW}, or with the propensity stratification approach using the mean propensity of each stratum, w^{Strat2}. In the other two choices, using design weights also produced the best results on some occasions (especially with the modified IPW approach, w^{IPWM}), but on many other occasions it was better to use balancing weights or no weighting at all; the latter approach would be the best (in terms of efficiency) when taking the propensity stratification approach with a correction factor, w^{Strat1}. In addition, regarding which type of transformation would be best for propensities, Table Table 4 shows that w^{IPWM} would be the best approach in terms of bias when using logistic regression or the TrIPW estimator, while in terms of efficiency ratio, it would be better to use w^{Strat2} for logistic regression. However, with Random Forests or XGBoost, the best choice on the majority of occasions, both in terms of relative bias and efficiency ratio, would be w^{Strat1}. This might seem contradictory given the results presented in Tables Table 1 and Table 2, according to which the best results for both algorithms are obtained when they are trained with design weights, while according to Table Table 3 w^{Strat1} works better if the algorithms are not weighted; the reason for this is that on most occasions where w^{Strat1} is a better option for these algorithms, they are used with no weighting or with balancing weights, but when design weights are used in training, the differences are significantly smaller. This outcome is to a certain extent apparent in Fig. 1 and is presented in greater detail in the tables in Appendix.

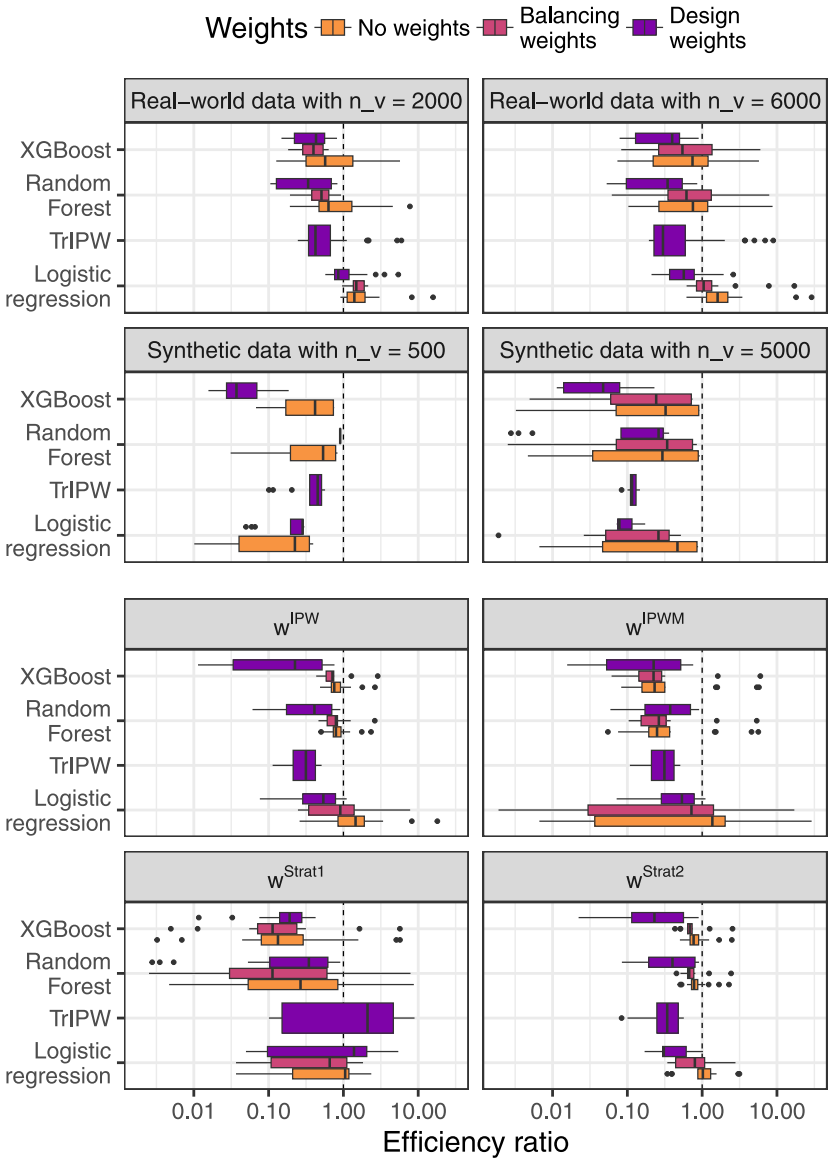

Fig. 1. Boxplots of the efficiency ratio results for each adjustment method (algorithm and weighting strategy) depending on the dataset and nonprobability sample size, and depending on the transformation applied to obtain the final weights of the estimator.

6. Discussion and conclusions

Various studies have been conducted to consider the use of design weights in propensity modelling for nonprobability survey estimation, and several estimators, based on different methodologies, have been proposed. However, in order to introduce new estimation methods for propensities it is necessary to determine the resulting properties when design weights are included or excluded. The present work is an attempt to fill this knowledge gap, both theoretically and empirically.

The results obtained from our simulations show, as in many related experiments, that there is no one-size-fits-all method in estimation. Nevertheless, some significant patterns can be observed. For example, the estimator proposed by [2], which includes a correction factor for design weights, based on the stratification of propensities without requiring the use of weighted predictive models in the propensity estimation step, does a good job in many of the simulations presented here, producing results that are close in precision to those of the estimators that instead include design weights in the modelling step, which is the ideal case [5]. The performance of this estimator is particularly good when XGBoost and Random Forest (trained with no weights or with balancing weights) are used for propensity estimation. These methods tend to give propensity estimates that are very close to 0 or 1 [28],

Table 3
Percentage of times each weighting strategy (in the algorithm training step) was the best versus all other possibilities, in terms of relative bias and efficiency ratio, depending on the transformation applied to obtain the final weights. Design w. = weighting with design weights. Bal. w. = weighting with balancing weights. No weight. = no weighting. No diff. = no difference (less than 10% of difference between all three possibilities).

Transformation	Best results in relative bias				Best results in efficiency ratio			
and sample size	Design w.	Bal. w.	No weight.	No diff.	Design w.	Bal. w.	No weight.	No diff.
Balanced ($n_r = n_v$)								
w^{IPW} transformation	57.1%		9.5%	33.3%	76.2%		14.3%	9.5%
w^{IPWM} transformation	47.6%		52.4%	0%	57.1%		42.9%	0%
w^{Strat1} transformation	38.1%		42.9%	19%	28.6%		61.9%	9.5%
w^{Strat2} transformation	71.4%		0%	28.6%	81%		19%	0%
Unbalanced ($n_r \neq n_v$)								
w^{IPW} transformation	78.8%	9.1%	6.1%	6.1%	97%	0%	0%	3%
w^{IPWM} transformation	45.5%	15.2%	39.4%	0%	63.6%	9.1%	27.3%	0%
w^{Strat1} transformation	36.4%	21.2%	33.3%	9.1%	33.3%	27.3%	36.4%	3%
w^{Strat2} transformation	90.9%	6.1%	0%	3%	97%	0%	3%	0%

Table 4
Percentage of times each transformation procedure for obtaining the final weights (w^{IPW}, w^{IPWM}, w^{Strat1}, w^{Strat2}) was the best versus all other possibilities, in terms of relative bias and efficiency ratio, depending on the algorithm used for propensity estimation. No diff. = no difference (less than 10% of difference between all four possibilities).

Algorithm	Best results in relative bias					Best results in efficiency ratio				
	w^{IPW}	w^{IPWM}	w^{Strat1}	w^{Strat2}	No diff.	w^{IPW}	w^{IPWM}	w^{Strat1}	w^{Strat2}	No diff.
Log. regr.	0%	55.3%	19.1%	25.5%	0%	0%	27.7%	27.7%	42.6%	2.1%
TrIPW	0%	50%	22.2%	22.2%	5.6%	0%	55.6%	22.2%	22.2%	0%
Random F.	0%	21.3%	55.3%	12.8%	10.6%	2.1%	23.4%	55.3%	12.8%	6.4%
XGBoost	6.4%	8.5%	68.1%	17%	0%	6.4%	17%	63.8%	12.8%	0%

meaning that this estimator could be very suitable for the case in question. The estimator incorporates a correction factor, i.e. the ratio of representation of a given stratum in the population and the nonprobability sample (essentially, this is what the estimator reflects). The propensities are used to divide the blended sample into strata, but are not directly used to calculate the final weights, meaning that disproportionately small or large propensities will not produce ill effects in this respect. On the other hand, when Random Forests and XGBoost were trained using design weights, the advantage of this estimator in the simulations disappeared (although it did not perform worse than the other approaches for transforming propensities). This fall-off might occur because using design weights both in the modelling and in the transformation could lead to overfitting (as the same information is used twice).

Furthermore, the present study provides empirical proof of the consistency of estimators that include design weights in the modelling step, which was theoretically demonstrated by [5] for the case in which logistic regression is used. In all of the scenarios considered for the simulations, weighted logistic regression produced better results than the unweighted version, and it was better to use design weights rather than our proposal of balancing weights. This improvement was especially clear with the inverse probability weighting transformation for propensities [17] was used, which was also the transformation considered in [5]. The simulations also show that this property holds in the XGBoost and Random Forest-based predictive algorithms we discuss, both of which outperform other approaches based on leveraging design weights in the modelling step, such as weighted logistic regression and the Tree-based Inverse Probability Weighted estimator.

In view of the results obtained, we recommend the use of the proposed modelling techniques in real applications, transforming propensities into weights using approaches that involve their stratification, namely those proposed by [2,9], although this choice is less relevant if design weights are used in training. In recent years, various machine learning algorithms have been proposed for propensity estimation in the nonprobability sampling estimation problem [10–12,23], but they have been applied without taking into account the sampling design of the probability sample. To the best of our knowledge, the present study is the first to specifically address this question and to compare its outcomes with those of the unweighted case, although some authors have proposed solutions which consider the impact of the sampling design [5,9,19]. These latter investigations have been taken into consideration in our study. Given the results obtained in our simulations and the theoretical development presented, we believe future investigations of the use of ML algorithms for propensity estimation should also contemplate including the probability sampling design in the modelling step, through the design weights of the probability sample.

The present study has certain limitations that should be acknowledged. Firstly, the range of methods tested was not as large as we could have wished; for example, some promising approaches such as Kernel Weighting [3] could also have been included. Moreover, the simulations only concerned two pseudopopulations with very specific characteristics and may not reflect all types of behaviour that could be present in real applications. Finally, the question of estimating the variance is not addressed here; this issue is usually tackled with jackknife or bootstrapping techniques, but a theoretical framework could be developed for the methods introduced in this work. Further studies of this matter should seek to include a wider range of techniques and populations, and also focus on the theoretical properties of the uncertainty presented by the estimators.

Table A.1
Results from the simulation with synthetic data for the estimation of the mean of y. The columns represent the sampling design of the probability sample; SRS = simple random sampling without replacement. Eff. = efficient stratified sampling. Ineff. = inefficient stratified sampling.

Algorithm	Weights in training	Weighting approach	$n_v = 500$						$n_v = 5,000$					
			Relative bias			Efficiency ratio			Relative bias			Efficiency ratio		
			SRS	Eff.	Ineff.	SRS	Eff.	Ineff.	SRS	Eff.	Ineff.	SRS	Eff.	Ineff.
LR	No	w^{IPW}	0.59	0.51	0.59	0.35	0.26	0.35	0.92	0.91	0.92	0.85	0.84	0.85
		w^{IPWM}	0.17	0.03	0.17	0.04	0.01	0.04	0.12	0.07	0.13	0.02	0.01	0.02
		w^{Strat1}	0.22	0.18	0.44	0.05	0.04	0.19	0.29	0.25	0.51	0.09	0.06	0.26
		w^{Strat2}	0.62	0.58	0.63	0.39	0.34	0.39	0.93	0.93	0.94	0.87	0.87	0.88
	Balancing	w^{IPW}							0.58	0.50	0.58	0.34	0.25	0.34
		w^{IPWM}							0.15	0.02	0.16	0.03	0.00	0.03
		w^{Strat1}							0.30	0.25	0.52	0.09	0.06	0.28
		w^{Strat2}							0.69	0.66	0.72	0.48	0.43	0.52
	Design	w^{IPW}	0.53	0.53	0.53	0.29	0.29	0.28	0.28	0.27	0.27	0.08	0.08	0.08
		w^{IPWM}	0.53	0.53	0.53	0.29	0.28	0.28	0.27	0.27	0.26	0.07	0.07	0.07
		w^{Strat1}	0.23	0.22	0.25	0.06	0.05	0.07	0.31	0.27	0.32	0.10	0.07	0.10
		w^{Strat2}	0.55	0.55	0.54	0.30	0.30	0.30	0.41	0.42	0.41	0.17	0.17	0.17
TrIPW	Design	w^{IPW}	0.67	0.71	0.67	0.45	0.51	0.46	0.33	0.38	0.35	0.11	0.15	0.12
		w^{IPWM}	0.67	0.71	0.67	0.45	0.51	0.46	0.33	0.38	0.34	0.11	0.14	0.12
		w^{Strat1}	0.30	0.32	0.44	0.10	0.11	0.20	0.35	0.37	0.32	0.13	0.14	0.11
		w^{Strat2}	0.72	0.75	0.65	0.52	0.57	0.43	0.29	0.32	0.33	0.08	0.10	0.11
RF	No	w^{IPW}	0.90	0.88	0.91	0.81	0.78	0.83	0.96	0.96	0.96	0.92	0.92	0.93
		w^{IPWM}	0.57	0.50	0.61	0.33	0.25	0.38	0.27	0.23	0.32	0.08	0.06	0.10
		w^{Strat1}	0.09	0.11	0.25	0.04	0.03	0.09	0.05	0.07	0.08	0.00	0.01	0.01
		w^{Strat2}	0.88	0.87	0.90	0.78	0.75	0.81	0.93	0.93	0.94	0.87	0.86	0.88
	Balancing	w^{IPW}							0.91	0.91	0.92	0.83	0.82	0.85
		w^{IPWM}							0.37	0.32	0.42	0.14	0.10	0.18
		w^{Strat1}							0.01	0.03	0.14	0.00	0.00	0.02
		w^{Strat2}							0.83	0.81	0.85	0.68	0.66	0.72
	Design	w^{IPW}	0.95	0.95	0.95	0.91	0.91	0.90	0.58	0.55	0.60	0.34	0.30	0.36
		w^{IPWM}	0.95	0.95	0.95	0.91	0.91	0.90	0.53	0.49	0.55	0.28	0.24	0.30
		w^{Strat1}	0.95	0.95	0.94	0.91	0.90	0.91	0.03	0.06	0.01	0.00	0.01	0.00
		w^{Strat2}	0.95	0.95	0.95	0.91	0.91	0.90	0.50	0.45	0.53	0.25	0.20	0.28
XGB	No	w^{IPW}	0.86	0.84	0.87	0.74	0.71	0.75	0.95	0.95	0.96	0.91	0.91	0.92
		w^{IPWM}	0.48	0.42	0.49	0.24	0.18	0.25	0.33	0.29	0.34	0.11	0.08	0.12
		w^{Strat1}	0.26	0.21	0.06	0.14	0.11	0.07	0.07	0.04	0.21	0.01	0.00	0.04
		w^{Strat2}	0.86	0.84	0.87	0.73	0.71	0.75	0.94	0.94	0.95	0.89	0.88	0.90
	Balancing	w^{IPW}							0.85	0.85	0.86	0.73	0.71	0.74
		w^{IPWM}							0.28	0.24	0.29	0.08	0.06	0.09
		w^{Strat1}							0.09	0.05	0.23	0.01	0.00	0.06
		w^{Strat2}							0.84	0.82	0.86	0.70	0.67	0.74
	Design	w^{IPW}	0.13	0.07	0.14	0.03	0.02	0.04	0.07	0.07	0.00	0.01	0.01	0.01
		w^{IPWM}	0.12	0.05	0.12	0.03	0.02	0.04	0.20	0.21	0.17	0.05	0.05	0.04
		w^{Strat1}	0.26	0.27	0.28	0.14	0.14	0.19	0.17	0.09	0.27	0.03	0.01	0.08
		w^{Strat2}	0.17	0.10	0.19	0.04	0.02	0.06	0.37	0.30	0.48	0.14	0.09	0.23

CRediT authorship contribution statement

Ramón Ferri-García: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jorge L. Rueda-Sánchez:** Writing – review & editing, Validation, Supervision, Software, Methodology, Formal analysis, Data curation. **María del Mar Rueda:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition. **Beatriz Cobo:** Writing – review & editing, Validation, Supervision, Software, Methodology, Formal analysis, Data curation.

Acknowledgements

This work is part of grant PDC2022-133293-I00 funded by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR, and partially funded by Consejería de Universidad, Investigación e Innovación (C-EXP-153-UGR23, Andalusia, Spain), Plan Propio de Investigación y Transferencia (PPJIA2023-030, University of Granada) and IMAG-Maria de Maeztu CEX2020-001105-M/AEI/10.13039/501100011033. The second author has a FPI grant from Ministerio de Educación y Ciencia (PRE2022-103200) associated with the aforementioned IMAG-Maria de Maeztu funding. The authors thank Kenneth C. Chu (Statistics Canada) and Jean-François Beaumont (Statistics Canada) for their assessment of the application of TrIPW algorithm, including the R package to perform the simulations. Funding for open access charge: Universidad de Granada / CBUA.

Table A.2
Results from the simulation with real-world data for the estimation of the proportion of individuals whose household has a car. The columns represent the sampling design of the probability sample; SRS = simple random sampling without replacement. Strat = stratified sampling. Uneq = unequal probabilities sampling.

Algorithm			$n_v = 2,000$						$n_v = 6,000$					
	Weights in training	Weighting approach	Relative bias			Efficiency ratio			Relative bias			Efficiency ratio		
			SRS	Strat	Uneq	SRS	Strat	Uneq	SRS	Strat	Uneq	SRS	Strat	Uneq
LR	No	w^{IPW}	0.67	0.65	1.02	1.12	1.12	1.40	0.63	0.56	1.07	1.58	1.72	1.73
		w^{IPWM}	0.58	0.59	1.08	1.01	1.07	1.54	0.42	0.42	1.20	1.34	1.64	2.06
		w^{Strat1}	0.75	0.83	0.92	1.16	1.12	1.19	0.68	0.71	0.97	0.62	1.06	1.17
		w^{Strat2}	0.72	0.69	1.04	0.91	1.01	1.31	0.85	0.83	1.07	0.95	1.03	1.22
	Balancing	w^{IPW}		0.56			1.49		0.70	0.69	1.07	0.81	1.02	1.42
		w^{IPWM}		0.51			1.49		0.56	0.60	1.16	0.62	0.91	1.65
		w^{Strat1}		0.83			1.12		0.68	0.76	0.96	0.62	0.86	1.08
		w^{Strat2}		0.65			0.96		0.72	0.71	1.08	0.76	0.85	1.29
	Design	w^{IPW}	0.82	0.90	0.81	0.79	0.92	0.73	0.71	0.83	0.71	0.54	0.78	0.54
		w^{IPWM}	0.82	0.90	0.80	0.79	0.92	0.73	0.71	0.83	0.71	0.54	0.78	0.53
		w^{Strat1}	0.91	1.21	0.91	1.74	2.09	1.46	0.74	1.02	0.72	0.76	1.39	0.73
		w^{Strat2}	0.67	0.87	0.66	0.62	0.86	0.57	0.48	0.69	0.49	0.31	0.60	0.30
TrIPW	Design	w^{IPW}	0.63	0.63	0.63	0.42	0.42	0.42	0.54	0.52	0.55	0.30	0.30	0.32
		w^{IPWM}	0.63	0.63	0.63	0.42	0.42	0.42	0.53	0.52	0.55	0.30	0.30	0.31
		w^{Strat1}	0.91	0.11	0.87	2.14	2.18	1.11	1.81	1.14	1.96	3.71	2.01	4.99
		w^{Strat2}	0.73	0.74	0.56	0.56	0.57	0.35	0.55	0.56	0.62	0.32	0.34	0.40
RF	No	w^{IPW}	0.77	0.79	1.11	0.61	0.64	1.23	0.87	0.88	1.05	0.76	0.77	1.10
		w^{IPWM}	0.52	0.53	1.24	0.31	0.33	1.52	0.46	0.47	1.21	0.23	0.25	1.47
		w^{Strat1}	0.34	0.34	1.37	0.26	0.39	1.84	0.23	0.31	1.31	0.10	0.22	1.72
		w^{Strat2}	0.78	0.79	1.11	0.63	0.64	1.22	0.88	0.87	1.05	0.78	0.77	1.10
	Balancing	w^{IPW}		0.79			0.64		0.77	0.79	1.12	0.60	0.64	1.25
		w^{IPWM}		0.53			0.33		0.51	0.54	1.25	0.28	0.32	1.57
		w^{Strat1}		0.35			0.39		0.10	0.12	1.36	0.06	0.14	1.84
		w^{Strat2}		0.79			0.64		0.80	0.80	1.11	0.64	0.65	1.24
	Design	w^{IPW}	0.81	0.81	0.88	0.68	0.68	0.78	0.67	0.71	0.83	0.46	0.53	0.71
		w^{IPWM}	0.81	0.81	0.88	0.67	0.68	0.78	0.67	0.70	0.83	0.46	0.53	0.70
		w^{Strat1}	0.77	0.68	0.84	0.63	0.53	0.73	0.56	0.53	0.76	0.34	0.36	0.60
		w^{Strat2}	0.84	0.83	0.91	0.72	0.71	0.83	0.75	0.78	0.92	0.58	0.63	0.86
XGB	No	w^{IPW}	0.76	0.77	1.13	0.60	0.62	1.26	0.87	0.87	1.06	0.76	0.77	1.12
		w^{IPWM}	0.51	0.51	1.27	0.30	0.32	1.59	0.46	0.47	1.24	0.23	0.26	1.52
		w^{Strat1}	0.20	0.13	1.27	0.13	0.18	1.58	0.27	0.18	1.22	0.10	0.10	1.49
		w^{Strat2}	0.79	0.78	1.12	0.63	0.63	1.24	0.89	0.89	1.05	0.80	0.79	1.11
	Balancing	w^{IPW}		0.77			0.62		0.76	0.78	1.13	0.58	0.61	1.28
		w^{IPWM}		0.50			0.32		0.49	0.51	1.27	0.26	0.30	1.62
		w^{Strat1}		0.13			0.18		0.27	0.20	1.28	0.11	0.12	1.64
		w^{Strat2}		0.79			0.64		0.81	0.81	1.12	0.65	0.66	1.26
	Design	w^{IPW}	0.71	0.71	0.86	0.53	0.55	0.76	0.62	0.64	0.80	0.40	0.45	0.66
		w^{IPWM}	0.70	0.70	0.86	0.53	0.55	0.76	0.61	0.64	0.80	0.40	0.45	0.66
		w^{Strat1}	0.34	0.41	0.54	0.20	0.29	0.35	0.43	0.51	0.63	0.22	0.32	0.42
		w^{Strat2}	0.70	0.70	0.90	0.52	0.56	0.82	0.70	0.75	0.94	0.51	0.60	0.90

Table A.3
Results from the simulation with real-world data for the estimation of the mean of monthly home expenses. The columns represent the sampling design of the probability sample; SRS = simple random sampling without replacement. Strat = stratified sampling. Uneq = unequal probabilities sampling.

Algorithm	Weights in training	Weighting approach	$n_v = 2,000$						$n_v = 6,000$					
			Relative bias			Efficiency ratio			Relative bias			Efficiency ratio		
			SRS	Strat	Uneq	SRS	Strat	Uneq	SRS	Strat	Uneq	SRS	Strat	Uneq
LR	No	w^{IPW}	0.49	0.50	2.17	1.51	1.97	8.22	0.64	0.57	2.54	2.40	3.39	18.04
		w^{IPWM}	0.35	0.41	2.92	1.40	1.94	15.81	0.29	0.28	4.07	2.14	3.45	29.06
		w^{Strat1}	0.67	0.78	0.96	1.71	2.36	2.20	0.49	0.56	1.08	0.76	1.05	1.72
		w^{Strat2}	0.56	0.55	1.57	1.26	1.55	3.04	0.75	0.74	1.42	1.25	1.56	3.15
	Balancing	w^{IPW}		0.40			2.13		0.55	0.52	2.44	1.02	1.32	7.80
		w^{IPWM}		0.32			2.15		0.30	0.33	3.59	0.83	1.23	17.09
		w^{Strat1}		0.76			1.83		0.50	0.64	0.98	0.69	1.20	1.56
		w^{Strat2}		0.49			1.45		0.54	0.55	1.59	0.84	1.12	2.79
	Design	w^{IPW}	0.71	0.86	0.72	0.84	1.11	0.77	0.52	0.66	0.54	0.37	0.81	0.37
		w^{IPWM}	0.71	0.86	0.72	0.84	1.11	0.77	0.52	0.66	0.54	0.37	0.81	0.36
		w^{Strat1}	0.70	0.96	0.44	5.42	3.54	2.70	0.45	0.79	0.49	1.38	2.60	1.94
		w^{Strat2}	0.38	0.78	0.39	0.71	1.03	0.58	0.00	0.38	0.06	0.31	0.62	0.21
TrIPW	Design	w^{IPW}	0.52	0.50	0.50	0.33	0.34	0.31	0.41	0.41	0.41	0.21	0.23	0.20
		w^{IPWM}	0.51	0.50	0.50	0.33	0.34	0.31	0.41	0.40	0.41	0.21	0.22	0.19
		w^{Strat1}	1.59	0.52	1.36	5.97	5.22	2.07	2.48	1.55	2.83	6.95	3.76	8.93
		w^{Strat2}	0.65	0.64	0.41	0.47	0.48	0.25	0.42	0.42	0.48	0.22	0.25	0.26
RF	No	w^{IPW}	0.69	0.68	1.57	0.52	0.50	2.34	0.85	0.85	1.34	0.74	0.73	1.76
		w^{IPWM}	0.35	0.29	2.21	0.20	0.19	4.58	0.39	0.35	2.40	0.19	0.20	5.64
		w^{Strat1}	0.72	0.77	2.88	0.65	0.92	7.75	0.41	0.50	2.99	0.27	0.53	8.74
		w^{Strat2}	0.70	0.68	1.55	0.53	0.51	2.28	0.86	0.84	1.30	0.74	0.71	1.68
	Balancing	w^{IPW}		0.68			0.51		0.68	0.67	1.64	0.48	0.46	2.63
		w^{IPWM}		0.29			0.19		0.33	0.25	2.34	0.15	0.14	5.37
		w^{Strat1}		0.77			0.93		0.51	0.65	2.84	0.36	0.70	7.89
		w^{Strat2}		0.68			0.51		0.71	0.66	1.58	0.51	0.45	2.44
	Design	w^{IPW}	0.19	0.09	0.31	0.11	0.11	0.17	0.15	0.02	0.42	0.06	0.08	0.20
		w^{IPWM}	0.19	0.09	0.31	0.11	0.11	0.16	0.15	0.01	0.41	0.06	0.08	0.19
		w^{Strat1}	0.13	0.26	0.20	0.11	0.21	0.13	0.21	0.48	0.03	0.10	0.40	0.05
		w^{Strat2}	0.26	0.15	0.34	0.15	0.13	0.19	0.25	0.09	0.48	0.10	0.08	0.26
XGB	No	w^{IPW}	0.69	0.66	1.67	0.52	0.49	2.64	0.83	0.82	1.35	0.70	0.69	1.79
		w^{IPWM}	0.35	0.26	2.39	0.21	0.21	5.38	0.32	0.24	2.43	0.14	0.15	5.75
		w^{Strat1}	0.03	0.17	2.46	0.15	0.33	5.71	0.03	0.17	2.28	0.07	0.20	5.10
		w^{Strat2}	0.71	0.67	1.62	0.54	0.50	2.49	0.85	0.82	1.31	0.73	0.68	1.69
	Balancing	w^{IPW}		0.66			0.49		0.67	0.64	1.71	0.47	0.43	2.88
		w^{IPWM}		0.27			0.21		0.32	0.20	2.48	0.15	0.14	6.00
		w^{Strat1}		0.15			0.32		0.03	0.24	2.41	0.08	0.26	5.69
		w^{Strat2}		0.68			0.51		0.70	0.64	1.61	0.51	0.43	2.55
	Design	w^{IPW}	0.44	0.17	0.76	0.27	0.19	0.62	0.28	0.00	0.69	0.13	0.11	0.50
		w^{IPWM}	0.44	0.16	0.76	0.27	0.19	0.62	0.28	0.01	0.69	0.13	0.12	0.49
		w^{Strat1}	0.02	0.18	0.25	0.15	0.28	0.20	0.05	0.29	0.30	0.08	0.26	0.15
		w^{Strat2}	0.36	0.06	0.70	0.23	0.19	0.56	0.23	0.00	0.73	0.11	0.12	0.57

Appendix. Simulation results

See Tables A.1–A.3.

References

[1] M.R. Elliott, R. Valliant, Inference for nonprobability samples, Statist. Sci. 32 (2) (2017) 249–264.
[2] S. Lee, Propensity score adjustment as a weighting scheme for volunteer panel web surveys, J. Off. Stat. 22 (2) (2006) 329–349.
[3] L. Wang, B.I. Graubard, H.A. Katki, A.Y. Li, Improving external validity of epidemiologic cohort analyses: A kernel weighting approach, J. R. Stat. Soc. Ser. A: Stat. Soc. 183 (3) (2020) 1293–1311.
[4] D. Rivers, Sampling for web surveys, 2007, Presented in Joint Statistical Meetings. Salt Lake City, UT.
[5] Y. Chen, P. Li, C. Wu, Doubly robust inference with nonprobability survey samples, J. Amer. Statist. Assoc. 115 (532) (2020) 2011–2021.
[6] R. Ferri-García, J.F. Beaumont, K. Bosa, J. Charlebois, K. Chu, Weight smoothing for nonprobability surveys, TEST 31 (3) (2022) 619–643.
[7] R.R. Andridge, B.T. West, R.J. Little, P.S. Boonstra, F. Alvarado-Leiton, Indices of non-ignorable selection bias for proportions estimated from non-probability samples, J. R. Stat. Soc. Ser. C. Appl. Stat. 68 (5) (2019) 1465–1483.
[8] R.J. Little, B.T. West, P.S. Boonstra, J. Hu, Measures of the degree of departure from ignorable sample selection, J. Surv. Stat. Methodol. 8 (5) (2020) 932–964.
[9] R. Valliant, J.A. Dever, Estimating propensity adjustments for volunteer web surveys, Sociol. Methods Res. 40 (1) (2011) 105–137.

[10] R. Ferri-García, M.D.M. Rueda, Propensity score adjustment using machine learning classification algorithms to control selection bias in online surveys, PLoS One 15 (4) (2020) e0231500.
[11] L. Castro-Martín, M.D.M. Rueda, R. Ferri-García, Inference from non-probability surveys with statistical matching and propensity score adjustment using modern prediction techniques, Mathematics 8 (6) (2020) 879.
[12] C. Kern, Y. Li, L. Wang, Boosted kernel weighting—Using statistical learning to improve inference from nonprobability samples, J. Surv. Stat. Methodol. 9 (5) (2020) 1088–1113, http://dx.doi.org/10.1093/jssam/smaa028.
[13] M. Schonlau, M.P. Couper, Options for conducting web surveys, Statist. Sci. 32 (2) (2017) 279–292.
[14] S. Lee, R. Valliant, Estimation for volunteer panel web surveys using propensity score adjustment and calibration adjustment, Sociol. Methods Res. 37 (3) (2009) 319–343.
[15] W.G. Cochran, The effectiveness of adjustment by subclassification in removing bias in observational studies, Biometrics (1968) 295–313.
[16] P.R. Rosenbaum, D.B. Rubin, Reducing bias in observational studies using subclassification on the propensity score, J. Amer. Statist. Assoc. 79 (387) (1984) 516–524.
[17] R. Valliant, Comparing alternatives for estimation from nonprobability samples, J. Surv. Stat. Methodol. 8 (2) (2020) 231–263.
[18] J.F. Beaumont, Are probability surveys bound to disappear for the production of official statistics, Survey Methodol. 46 (1) (2020) 1–28.
[19] K.C.K. Chu, J.F. Beaumont, The use of classification trees to reduce selection bias for a non-probability sample with help from a probability sample, in: Proceedings of the Survey Methods Section: SSC Annual Meeting, vol. 26, Calgary, AB, Canada, 2019.
[20] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.
[21] J.D. Malley, J. Kruppa, A. Dasgupta, K.G. Malley, A. Ziegler, Probability machines, Methods Inf. Med. 51 (01) (2012) 74–81.
[22] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[23] L. Castro-Martín, M.d.M. Rueda, R. Ferri-García, C. Hernando-Tamayo, On the use of gradient boosting methods to improve the estimation with data obtained with self-selection procedures, Mathematics 9 (2991) (2021) http://dx.doi.org/10.3390/math9232991.
[24] J.H. Friedman, Greedy function approximation: A gradient boosting machine, Ann. Stat. (2001) 1189–1232.
[25] T. Chen, He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, M. Li, J. Xie, M. Lin, Y. Geng, Y. Li, J. Yuan, Xgboost: Extreme gradient boosting. r package version 1.7.7.1, 2024, https://CRAN.R-project.org/package=xgboost.
[26] W.G. Madow, On the theory of systematic sampling, II, Ann. Math. Stat. 20 (3) (1949) 333–354.
[27] Y. Tillé, A. Matei, Sampling: Survey sampling, 2023, R package version 2.10. https://CRAN.R-project.org/package=sampling.
[28] T.D. Buskirk, S. Kolenikov, Finding respondents in the forest: A comparison of logistic regression and random forest models for response propensity weighting and stratification, Surv. Methods: Insights Field (2015) 1–17.