Evaluating Machine Learning methods for estimation in online surveys with superpopulation modeling.

Ramón Ferri-García, Luis Castro Martín and María del Mar Rueda

Department of Statistics and Operations Research, University of Granada
{rferri,luiscastro193,mrueda}@ugr.es

Abstract

Online surveys, despite their cost and effort advantages, are particularly prone to selection bias due to the differences between target population and potentially covered population (online population). This leads to the unreliability of estimates coming from online samples unless further adjustments are applied. Some techniques have arisen in the last years regarding this issue, among which superpopulation modeling can be useful in Big Data context where censuses are accesible. This technique uses the sample to train a model capturing the behaviour of a target variable which is to be estimated, and applies it to the nonsampled individuals to obtain population-level estimates. The modeling step has been usually done with linear regression or LASSO models, but machine learning (ML) algorithms has been pointed out as promising alternatives. In this study we examine the use of these algorithms in the online survey context, in order to evaluate and compare their performance and adequacy to the problem. A simulation study shows that ML algorithms can effectively volunteering bias to a greater extent than traditional methods in several scenarios.

Keywords: Superpopulation modeling, Machine Learning, Online surveys and Simulation.

1. Introduction

Online surveys have become one of the most used modes of survey administration worldwide. They are a powerful tool for recruiting respondents fast and effortlessly with small costs in comparison to traditional survey administration modes. However, samples from online surveys are usually collected using a nonprobabilistic scheme, given that access to all members of the target population is not guaranteed in most cases and the inclusion probability cannot be obtained because of the absence of a sampling frame. As a result, selection bias derived from this procedure, defined by [8] as the presence of a substantial difference between observed and unobserved population, makes survey estimates not valid for inference [22].

Different inference procedures are proposed in the literature to correct for selection bias induced by non-random selection mechanisms. There are three important approaches: the pseudo-design based inference (or pseudo-randomization [6]), statistical matching and predictive inference.

In the pseudo-design based inference, the idea is to construct weights to correct for selection bias. The first method consists of estimating response probabilities and using them in Horvitz-Thompson type estimators to account for unequal selection probabilities. The most used method to estimate the response probabilities is propensity scoring proposed by [19] (see e.g. [13]). This method uses a probability reference sample to construct a propensity model for the non-probability sample. Sample matching is another approach also applied to reduce selection bias in non-probability samples by combining them with a probability sample.

In this paper, we consider the situation where there is only a non-probability sample available for measuring the target information, in addition to some auxiliary information of the full population of interest, and we consider several predictive inference methods. Predictive methods are based on superpopulation models. In this approach, a statistical model is fitted for the analysis variable y from the sample and used to project the sample to the full population. This approach (that can be used with probability and non-probability samples) let us use auxiliary information about covariates on different methods for predicting the unknown values. The objective of this study is to examine the use of Machine Learning algorithms in the online survey context, to evaluate and compare their performance and adequacy to the problem. A simulation study is performed for that matter.

2. Predictive inference for non-probability samples

Let s be the online sample, $\overline{s}$ the population not included in the sample, and U the complete target population so $s \cup \overline{s} = U$. The goal is to estimate the population parameter of a target variable, y, which has been measured in s but it is not available in data from $\overline{s}$.

The prediction approach is based on superpopulation models, which assume that the population under study $\mathbf{y} = (y_1, ..., y_N)'$ are observations of super-population random variables $\mathbf{Y} = (Y_1, ..., Y_N)'$ having a superpopulation model ξ. To incorporate auxiliary information $\mathbf{x}_i$ available for all $i \in U$ we assume a superpopulation for y built on some mean function of x:

$$Y_i = m(\mathbf{x}_i) + e_i, \quad i = 1, ..., N. \tag{1}$$

The random vector $e = (e_1, ..., e_N)'$ is assumed to have zero mean and a positive definite covariance matrix which is diagonal.

Using a set of covariates, $\mathbf{x}$, measured in s and $\overline{s}$ it is possible to estimate the values of y in $\overline{s}$ with regression modeling such that the estimated value of y for an individual i can be calculated through the following expression:

$$\hat{y}_i = E_m(y_i|\mathbf{x}_i) \tag{2}$$

m alludes to the specific model which provides the expectation of y_i, and $\mathbf{x}_i$ are the values of the i-th individual in the covariates $\mathbf{x}$.

If we want to estimate the total of y, $\overline{Y}$, we can use the auxiliary information in several ways and we can define several estimators:

- the model-based estimator:

$$\hat{\overline{Y}}_m = \frac{1}{N}\left(\sum_{i \in s} y_i + \sum_{i \in \overline{s}} \hat{y}_i\right) \tag{3}$$

- the model-assisted estimator:

$$\hat{\overline{Y}}_{ma} = \frac{1}{N}\left(\sum_{i \in U} \hat{y}_i + \sum_{i \in s} (y_i - \hat{y}_i) w_i\right) \tag{4}$$

  being w_i a weight of the unit i (set by the researcher to adjust the lack of response, lack of coverage, voluntariness, ... usually employing post-stratification).

- the model-calibrated estimator:

$$\hat{\overline{Y}}_{cal} = \frac{1}{N}\sum_{k \in s} y_k w_k^{CAL} \tag{5}$$

  where w_k^{CAL} are such that they minimize $\sum_{k \in s} G\left(w_k^{CAL}, w_k\right)$, where $G(\cdot,\cdot)$ is a particular distance function, subject to $\sum_{k \in s} w_k^{CAL}\hat{y}_i = \sum_{k \in U} \hat{y}_i$.

3. Machine learning techniques in superpopulation modelling

Usually the linear regression model is considered for estimation, $E_m(y_i|\mathbf{x}_i) = \mathbf{x}_i^T\beta$, and the above estimators can be rewritten as a type of regression estimators. Alternatively to the linear regression model, Machine Learning (ML) methods have been proposed for the estimation of the nonsampled population values. In situations where additivity and/or linearity do not hold, ML algorithms are more suitable for regression and classification. Some of these algorithms, such as decision trees and related (Random Forests, Gradient Boosting Machines) can also take interactions into account without the need of specifying the terms. The use of some ML algorithms for probabilistic samples has been studied in the last few years for deriving model-assisted estimators ([14]; [1]; [21]; [23]; [4]). In this section, we consider some of the most important ML algorithms that can be used to define model-assisted, model-based and model-calibrated estimators for a non-probability sample.

3.1. Advanced linear regression models

β coefficients of a linear regression estimated by ordinary least squares are estimated as $\beta = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$. However, as [10] stated, this estimation becomes sensitive when $\mathbf{X}'\mathbf{X}$ is far from being a unit matrix (i. e. multicollinearity is present in covariates). In such a case, ridge regression can be an alternative. It estimates regression coefficients adding an identity term to control instability, $\beta = (\mathbf{X}'\mathbf{X} + k\mathbf{I})^{-1}\mathbf{X}'\mathbf{Y}$, where $k \geq 0$ is a coefficient which depends on (unknown) real regression parameters and therefore has to be chosen arbitrarily or via hyperparameter tuning. From a Bayesian point of view, the resulting β can be considered the posterior mean of a prior Normal distribution with zero mean and a variance of $\mathbf{I}\sigma^2/k$ as described in [11]. Gibbs sampling can provide Bayesian estimates for β in such a case.

An alternative to ridge regression is the Least Absolute Shrinkage and Selection Operator (LASSO) regression, described in [20], where coefficients are estimated through minimizing the least-squares with a penalty parameter, α, subject to a restriction on a tuning parameter:

$$\begin{array}{l} argmin \sum_{i=1}^{N}(y_i - \alpha - \sum_j \beta_j x_{ij})^2 \\ subject\ to \sum_j |\beta_j| \leq t \end{array} \tag{6}$$

The restriction t is fixed to allow shrinkage of the solutions towards zero, allowing some coefficients to be equal to zero. As a consequence, this approach performs variable selection, in contrast to ridge regression where coefficients are always different from zero. LASSO estimates can be seen as posterior estimates of the distribution mode of prior Laplace independent and identical distributions. Therefore, Bayesian procedures can be used for LASSO estimation as described in [18].

3.2. Bagged Trees

Estimating the expectance $E_m(y_i|\mathbf{x}_i)$ under decision tree modeling results in the following expression:

$$E_m(y_i|\mathbf{x}_i) = \begin{cases} \overline{y(s^{J_1})} & \{i \in s/\mathbf{x}_i \in J_1\} \\ ... & ... \\ \overline{y(s^{J_k})} & \{i \in s/\mathbf{x}_i \in J_k\} \end{cases} \tag{7}$$

where $\overline{y(s^{J_i})}$ is the mean of y for the members of the sampled population, s, which meet the criteria of the ith terminal node. If considering the Bagged Trees method, predictions are made by averaging results from a range of m unpruned trees known as *weak classifiers*, each one trained in a bootstrapped subsample of the complete dataset:

$$E_m(y_i|\mathbf{x}_i) = \frac{\sum_{j=1}^{m} \phi_j(\mathbf{x}_i)}{m}, \ \phi_j(\mathbf{x}_i) = \begin{cases} \overline{y(s^{J_1^j})} & \{i \in s/\mathbf{x}_i \in J_1^j\} \\ ... & ... \\ \overline{y(s^{J_k^j})} & \{i \in s/\mathbf{x}_i \in J_k^j\} \end{cases} \tag{8}$$

where $\overline{y(s^{J_i^j})}$ is the mean of y for the members of the sampled population, s, which meet the criteria of the ith terminal node of the jth tree.

3.3. Gradient Boosting Machine

Gradient Boosting Machine (GBM) algorithm can be used for prediction in superpopulation modeling. The new formula of the estimates of y would be:

$$E_m(y_i|\mathbf{x}_i) = v^T J(\mathbf{x}_i) \tag{9}$$

where $J(\mathbf{x}_i)$ stands for a matrix of terminal nodes of m decision trees used for boosting, which is obtained through an iterative process that aims to minimize a given loss function, and v is a vector representing the weight of each tree.

3.4. k-Nearest Neighbors

k-Nearest Neighbors (k-NN) can also be used for prediction, although they constitute a much simpler algorithm. The expectance of y_i is calculated by averaging the value of y for its k nearest neighbors, this is, the k individuals closer to the ith individual according to the covariates $\mathbf{x}_i$:

$$E_m(y_i|\mathbf{x}_i) = \frac{\sum_{j \in s/d(\mathbf{x}_i,\mathbf{x}_j) \leq d(\mathbf{x}_i,\mathbf{x}_{(k)})} y_j}{k} \tag{10}$$

where $x_{(1)}$,...,$x_{(n-1)}$ denote, respectively, the closest and the furthest individual to x_i according to the distance d.

3.5. Neural networks with Bayesian Regularization

Approaches based on neural networks have been considered in the literature for superpopulation modeling [4]. In that class of models, expectance of y_i is calculated through an iterative process as defined in [17]:

$$E_m(y_i|\mathbf{x}_i) = g\left(\sum_{k=1}^{L} v_k f_k(\cdot) + b\right) \tag{11}$$

where g and f_k stand for activation functions which can have the same image, v_k are the weights of the kth neuron of the hidden layer and b is the activation threshold. The inputs are noted as $f_k(\cdot)$ given that several hidden layers can be fixed and, as a result, the inputs would go through an iterative process before reaching the last layer where the outputs are calculated. Alternatively, and as a regularization method to avoid overfitting, prior distributions can be imposed in v_k weights so they can be estimated by calculating those who maximize the posterior density or via maximum likelihood. Further details are described in [17].

4. The simulation study

4.1. Data

We have selected 3 different populations to experiment with. Also, for each one, we have tried different sampling strategies.

The first population, P1, consists of the 2012 edition of the Spanish Life Conditions Survey microdata [16]. The dataset contains information on economic and life conditions variables for 28,610 adult individuals. We pretend to predict the mean self-reported health on a scale from 1 to 5. For training, we used the 56 related variables. In this population, we tested two sampling strategies. The first one, P1S1, was a simple random sampling (SRS) among the population with internet access. For the second one, P1S2, a propensity to participate was considered according to the formula $Pr(yr) = \frac{yr^2 - 1900^2}{1996^2 - 1900^2}$, where yr is the year the individual was born.

The second population, P2, is BigLucy [9]. It corresponds to some financial variables of 85,396 industrial companies of a city in a particular fiscal year. We used the annual income as the target variable. For training, we took into account the level of the company (small, medium or big), the number of employees, whether it is ISO certified and the company's income tax. In this population, we tried two different sampling methods. The first one, P2S1, was SRS excluding the companies without SPAM options and the small companies. In this scenario, we tested if the algorithms could accurately predict data without any training sample (since any small company can be sampled). The second one, P2S2, only filtered by SPAM availability but it included a propensity to participate with the formula $Pr(taxes) = min(taxes^2/30, 1)$ where $taxes$ is the company's income tax.

The third population, P3, is the Bank Marketing Data Set [15], related to direct marketing campaigns (phone calls) of a Portuguese banking institution. We aimed to estimate the mean contact duration. We trained the algorithms with 18 variables. For sampling, we filtered by the number of contacts performed for each client and tested two possibilities. In the first one, P3S1, we performed SRS among those contacted more than 3 times. In the second one, P3S2, we tested another SRS among those contacted more than twice.

4.2. Procedure

For each population and sampling strategy described, we ran an experiment with 3 different sample sizes: 1000, 2000 and 5000. For each sample size, 500 simulations were executed. In each simulation, model-based, model-assisted and model-calibrated estimates were obtained using the following predictive algorithms: linear regression (*glm*), Ridge regression with and without Bayesian priors (*bridge* and *ridge* respectively), LASSO regression via penalized maximum likelihood (*glmnet*), LARS-EN algorithm (*lasso*) and using Bayesian priors on the estimates (*blasso*), k-Nearest Neighbors (*knn*), Bagged Trees (*treebag*), Gradient Boosting Machine (*gbm*) and Bayesian-regularized Neural Networks (*brnn*). Default parameters were used for every algorithm except for k-Nearest Neighbors since its results were especially sensitive to parameter optimization. The proper k is chosen via bootstrap.

The relative mean bias, relative standard deviation and the relative Root Mean Square Error in each scenario are measured as follows:

$$RBias\ (\%) = \left(\frac{\sum_{i=1}^{500} \hat{p}_{yi}}{500} - p_y \right) \cdot \frac{100}{p_y};\ RSD\ (\%) = \sqrt{\frac{\sum_{i=1}^{500} (\hat{p}_{yi} - \hat{\bar{p}}_y)^2}{499}} \cdot \frac{100}{p_y}$$

$$RMSE\ (\%) = \sqrt{RBias^2 + RSD^2}$$

with p_y the value of the target variable, $\hat{\bar{p}}_y$ the mean of the 500 estimations of p_y and $\hat{p}_{yi}$ the estimation of p_y in the i-th simulation.

To compare each estimator, we consider three metrics: its mean efficiency, its median efficiency and the number of times it has been among the best. An estimator has performed as the best when its RMSE differs from the minimum RMSE by less than 1%. The efficiency is defined as follows:

$$Efficiency\ (\%) = \frac{Baseline - RMSE}{Baseline} \cdot 100$$

where the baseline is the RMSE of using the sample average as the estimation.

Additionally, the results were analyzed using linear mixed-effects regression, to obtain estimates of the effect sizes of each algorithm on the final Root Mean Square Error (RMSE). All the analyses were performed in R.

4.3. Results

RMSEs of each estimator for each population, sampling method and sample size can be observed in Table 1. Some algorithms achieve good results consistently, like Ridge regression. Others can greatly outperform the rest for some cases while getting poor results for the rest, like k-Nearest Neighbors. Bayesian-regularized Neural Networks are a special case since they produce promising estimations but can suffer due to a lack of data. Finally, there is a group of algorithms that never seem to be the right choice, like Bagged Trees. In any case, there is not much difference between model-based, model-assisted and model-calibrated estimates.

In order to confirm those impressions, the ranking can be seen in Table 2. Model-assisted Ridge regression has the best mean efficiency, median efficiency and the number of times it has been among the best. This is not a surprise since it is a technique for analyzing data that suffer from multicollinearity, which is expected to be the case for most biased samples.

Results of the linear mixed-effects regression (see Appendix) confirm these conclusions: there is no evidence in the simulations' results that the effect is different between Ridge regression, GLM, LASSO maximum-likelihood regression (both bayesian and non-bayesian), k-Nearest Neighbors or Bayesian-regularized Neural Networks. Nonetheless, there is evidence of a smaller RMSE reduction effect for Gradient Boosting Machines and Bagged Trees in comparison to the algorithms aforementioned, except for k-Nearest Neighbors.

5. Conclusions

This paper describes some options for estimation in non-probability samples using ML techniques in three approaches: model-based, model-assisted and model-calibrated. The paper clarifies which assumptions are required and illustrates how these proposed estimators perform empirically. The main conclusion in our simulation study is that the selection of the ML algorithm used in the process is more important than the approach used in the estimation. There is a group of ML techniques that are similar in their good performance, highlighting the Ridge regression method.

[6] also evaluates the behavior of various ML methods for model-based estimators. We have conducted a study with a broader class of estimators and more ML methods. The results obtained in our study agree on those obtained in the study by [6] in the sense that Machine Learning methods are more powerful at removing selection bias in non-probability samples than traditional estimators. However, their performance is strongly dependent on the dataset characteristics, meaning that there could not be a unique algorithm for maximizing the estimates' accuracy. Further research should consider algorithm-specific data preprocessing steps in the analysis.

References

[1] Baffetta, F., Fattorini, L., Franceschi, S. & Corona P. (2009). Design-based approach to k-nearest neighbours technique for coupling field and remotely sensed data in forest surveys. *Remote Sensing of Environment*, 113(3), 463-475.

Table 1: RMSE (%) of each estimator for each population, sampling method and sample size

	P1S1			P1S2			P2S1			P2S2			P3S1			P3S2		
Estimator	1000	2000	5000	1000	2000	5000	1000	2000	5000	1000	2000	5000	1000	2000	5000	1000	2000	5000
baseline	8.5	8.4	8.5	12.9	12.8	12.8	70.6	70.5	70.5	32.9	32.8	32.8	13.5	13.4	13.2	6.7	6.4	6
model assisted blasso	3.5	3.1	2.7	6.9	6	5.1	**24.8**	**24.8**	**24.7**	12.6	12.7	12.8	9.7	5.6	2.3	4.8	2.8	2.1
model based blasso	3.6	3.1	2.7	6.9	6	5.3	**24.5**	**24.6**	**24.5**	12.5	12.6	12.7	9.9	6	2.8	4.7	2.9	2.4
model calibrated blasso	3.6	3.1	2.7	6.8	5.9	5.1	**24.6**	**24.6**	**24.6**	12.5	12.7	12.7	9.7	5.6	2.1	4.9	2.6	2
model assisted bridge	3.7	3.2	2.7	7.1	6.1	5.2	**24.8**	**24.7**	**24.7**	12.6	12.7	12.8	10.1	5.5	3.2	5.4	3.2	2.3
model based bridge	3.8	3.3	2.9	7.3	6.3	5.4	**24.6**	**24.5**	**24.5**	12.5	12.6	12.6	10.4	5.7	3.1	5.5	3.2	2.3
model calibrated bridge	3.6	3.2	2.7	7	6.1	5.2	**24.5**	**24.5**	**24.6**	12.6	12.7	12.7	10	5.5	3	5.5	3.3	2.4
model assisted brnn	**2.9**	**2.5**	**2.2**	**5.7**	**4.9**	**4.5**	25.5	25.2	**24.8**	12.8	12.9	12.9	7.4	5.5	2.3	5.2	4.3	4.3
model based brnn	**2.7**	**2.4**	**2.3**	**5.3**	**4.7**	**4.3**	25.8	25.3	**24.9**	13.3	13.3	13.3	7.4	5.3	2.4	5.1	4.4	4.4
model calibrated brnn	**2.9**	**2.5**	**2.3**	**5.4**	**4.9**	**4.5**	25.6	25.1	**24.6**	12.9	13	12.9	7.2	10.6	2.1	5.4	4.1	4.3
model assisted gbm	5.5	5.4	5.4	9.1	8.9	8.9	46.2	46.7	47.1	17.1	17.4	17.6	7	5.3	5.2	9	7.2	4.6
model based gbm	5.5	5.4	5.5	9.3	8.9	8.9	46.2	46.6	47.1	17	17.4	17.5	7.7	4.8	5.2	8.9	7.4	4.6
model calibrated gbm	5.5	5.4	5.4	9	8.9	8.9	46	46.5	46.9	16.9	17.2	17.3	8	5.3	5.1	9	7	4.8
model assisted glm	**2.8**	**2.7**	**2.6**	**5.1**	**4.9**	**4.7**	**24.5**	**24.6**	**24.6**	12.6	12.7	12.8	**4.1**	**2.8**	**1.5**	5	4	3.5
model based glm	**2.9**	**2.6**	**2.5**	**5**	**4.8**	**4.7**	**24.4**	**24.6**	**24.6**	12.6	12.7	12.7	**4.1**	**2.9**	**1.4**	5.1	4.1	3.5
model calibrated glm	**2.9**	**2.7**	**2.5**	**5.1**	**4.8**	**4.7**	**24.4**	**24.5**	**24.5**	12.6	12.7	12.7	**4.2**	**2.7**	**1.4**	5	4.2	3.5
model assisted glmnet	**2.9**	**2.7**	**2.6**	**5.1**	**4.9**	**4.9**	25.4	25.5	25.6	12.7	12.8	12.8	**3.9**	**2.7**	**1.4**	4.8	4.1	3.4
model based glmnet	**2.9**	**2.8**	**2.6**	**5.2**	**5**	**4.9**	25.5	25.5	25.5	12.7	12.8	12.8	**4.2**	**2.8**	**1.4**	5	4	3.4
model calibrated glmnet	**2.9**	**2.8**	**2.6**	**5.1**	**5**	**4.9**	25.3	25.4	25.5	12.7	12.7	12.7	**4.2**	**2.5**	**1.3**	5	4	3.5
model assisted knn	4	3.6	3.1	6.7	6.2	5.5	34.5	34.3	34.1	**7.4**	**5.8**	**4.3**	7.5	6.8	6.3	**3.9**	**2.6**	**1.5**
model based knn	5.1	4.5	3.5	7.8	7.1	6.1	34.1	34.2	34.1	**7.1**	**5.7**	**4.3**	7.6	6.6	5.8	**4.2**	**2.7**	**1.6**
model calibrated knn	3.9	3.4	2.9	6.5	6.1	5.4	34.3	34.2	34.1	**7.2**	**5.8**	**4.2**	7.4	6.6	6.4	**3.9**	**2.7**	**1.5**
model assisted lasso	7.1	7.2	7.3	10.9	11	11.1	65.5	65.6	65.6	29.8	29.8	29.8	7.9	6.9	6.6	**4**	**2.6**	**1.8**
model based lasso	7.1	7.2	7.3	10.9	11	11.1	65.5	65.6	65.6	29.7	29.8	29.7	7.9	6.9	6.6	**3.9**	**2.8**	**1.9**
model calibrated lasso	7.2	7.2	7.2	10.8	11.1	11.1	65.4	65.4	65.5	29.8	29.7	29.7	7.9	6.7	6.5	**3.9**	**2.7**	**1.8**
model assisted ridge	**2.8**	**2.7**	**2.5**	**5**	**4.8**	**4.7**	**24.5**	**24.6**	**24.6**	12.6	12.7	12.7	**4.1**	**2.8**	**1.3**	4.8	4	3.4
model based ridge	**2.8**	**2.6**	**2.5**	**5**	**4.7**	**4.7**	**24.5**	**24.6**	**24.7**	12.6	12.7	12.8	**4.2**	**2.8**	**1.4**	5	4.2	3.5
model calibrated ridge	**2.8**	**2.7**	**2.5**	**5**	**4.9**	**4.7**	**24.4**	**24.5**	**24.5**	12.6	12.7	12.7	**4.3**	**2.9**	**1.3**	5.1	4.3	3.5
model assisted treebag	3.8	3.7	4.2	6.7	6	6.4	45.6	45.8	46	16.6	16.6	16.6	13	5.7	8.9	13.7	9.3	2.6
model based treebag	3.9	3.8	4.2	6.7	6.4	6.4	45.6	45.7	46	16.5	16.5	16.5	11.9	6.7	8.8	13.3	8.4	2.8
model calibrated treebag	3.7	3.6	4.2	6.7	6.1	6.3	45.6	45.8	46	16.4	16.6	16.6	14.5	6.3	8.7	15.9	9.1	2.7

Table 2: Mean and median efficiency (%) of each estimator and times it has been among the best. ma = Model-assisted, mb = Model-based, mc = Model-calibrated

	Mean	Median	Best
ma ridge	62,2	64,3	13
mb ridge	61,9	64,1	12
ma glm	61,7	64,3	12
mb glm	61,7	64,1	12
mc glm	61,7	64,3	12
mc ridge	61,6	64,3	12
ma glmnet	61,6	62,8	11
mc glmnet	61,5	63	12
mb glmnet	61,3	63	9
mc knn	59,1	53,1	7
ma knn	58,5	52,7	7
mc blasso	58,5	61,3	10
ma blasso	58,2	61,2	11
mb brnn	57,9	61,8	8
ma brnn	57,8	61,4	9
mb blasso	57,4	60,8	10
mc bridge	56,3	60,9	9
ma bridge	56,2	61,2	9
mc brnn	55,8	61,4	9
mb knn	55,7	51,6	6
mb bridge	55,7	61,3	7
ma gbm	32,6	34,7	0
mc gbm	32,4	35,1	0
mb gbm	32,4	35	0
mb treebag	32,3	49,6	0
ma treebag	31,5	49,3	0
mc treebag	29,1	49,5	1
mc lasso	25,1	14,9	3
ma lasso	24,8	14,5	3
mb lasso	24,7	14,3	3

[2] Barton, K.: MuMIn (2018). Multi-Model Inference. R package version 1.42.1. `https://CRAN.R-project.org/package=MuMIn`

[3] Bates, D., Maechler, M., Bolker, B. & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1), 1-48. doi:10.18637/jss.v067.i01.

[4] Breidt, F. J. & Opsomer, J. D. (2017). Model-assisted survey estimation with modern prediction techniques. *Statistical Science*, 32(2), 190-205.

[5] Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). Classification and regression trees. Wadsworth, Belmont, California.

[6] Buelens, B., Burger, J. & van den Brakel, JA. (2018) Comparing Inference Methods for Non-probability Samples. *International Statistical Review*, 86(2), 322343.

[7] Chen, J. K. T., Valliant, R. L., & Elliott, M. R. (2019). Calibrating nonprobability surveys to estimated control totals using LASSO, with an application to political polling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 68(3), 657-681.

[8] Elliott, M. R., & Valliant, R. (2017). Inference for nonprobability samples. *Statistical Science*, 32(2), 249-264.

[9] Gutirrez, H. A. (2009). Estrategias de muestreo. Diseo de encuestas y estimacin de parmetros. Universidad Santo Toms, Bogot (Colombia).

[10] Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55-67.

[11] Hsiang, T. C. (1975). A Bayesian View on Ridge Regression. *The Statistician*, 24(4), 267. doi:10.2307/2987923

[12] Kuznetsova, A., Brockhoff, P. B. & Christensen, R. H. B. (2017). lmerTest Package: Tests in Linear Mixed Effects Models. *Journal of Statistical Software*, 82(13), 1-26. doi: 10.18637/jss.v082.i13

[13] Lee, S. & Valliant, R. (2009). Estimation for Volunteer Panel Web Surveys Using Propensity Score Adjustment and Calibration Adjustment. *Sociological Methods and Research*, 37(3), 319-343.

[14] Montanari, G. E. & Ranalli, M. G. (2007). Multiple and ridge model calibration. In: *Proceedings of Workshop on Calibration and Estimation in Surveys*, Ottawa, Canada, October 31-November 1 2007.

[15] Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62, 22-31.

[16] National Institute of Statistics: Life Conditions Survey. Microdata (2012). Retrieved from `https://www.ine.es/dyngs/INEbase/en/operacion.htm?c=Estadistica_C&cid=1254736176807&menu=ultiDatos&idp=1254735976608` (accessed 7 november 2019).

[17] Okut, H. (2016). Bayesian regularized neural networks for small n big p data. In Rosa, J. L. G. (Ed.). (2016). Artificial Neural Networks: Models and Applications. BoDBooks on Demand.

[18] Park, T., & Casella, G. (2008). The bayesian lasso. *Journal of the American Statistical Association*, 103(482), 681-686.

[19] Rosenbaum, P. R. & Rubin, D. B. (1983). The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika*, 70(1), 41-55.

[20] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267-288.

[21] Tipton, J., Opsomer, J. & Moisen, G. (2013). Properties of endogenous post-stratified estimation using remote sensing data. *Remote Sensing of Environment*, 139, 130-137.

[22] Valliant, R. (2019). Comparing Alternatives for Estimation from Nonprobability Samples. *Journal of Survey Statistics and Methodology*, smz003, https://doi.org/10.1093/jssam/smz003

[23] Wang, J. C., Opsomer., J. D. & Wang, H. (2014). Bagging nondifferentiable estimators in complex surveys. *Survey Methodology*, 40, 189-209.

Appendix

Coefficient	Estimate	Std. Error	D. f.	t value	IC 95%	p-value
(Intercept)	24.073	5.118	5.641	4.704	[11.354; 36.792]	0.0039
blasso	-14.805	1.454	542.000	-10.183	[-17.661; -11.949]	2.10e-16
bridge	-14.637	1.454	542.000	-10.068	[-17.493; -11.781]	5.69e-16
brnn	-14.767	1.454	542.000	-10.157	[-17.623; -11.911]	2.63e-16
gbm	-8.880	1.454	542.000	-6.108	[-11.736; -6.024]	1.93e-03
glm	-15.444	1.454	542.000	-10.623	[-18.299; -12.588]	4.51e-18
glmnet	-15.268	1.454	542.000	-10.502	[-18.124; -12.412]	1.31e-17
knn	-14.132	1.454	542.000	-9.720	[-16.988; -11.276]	1.08e-14
lasso	-3.509	1.454	542.000	-2.413	[-6.364; -0.653]	0.0161
ridge	-15.457	1.454	542.000	-10.632	[-18.313; -12.602]	4.15e-18
treebag	-8.964	1.454	542.000	-6.166	[-11.820; -6.108]	1.37e-03

Group	Variance	Std. Dev.
Dataset	147.66	12.151
Residual	28.53	5.342

Dataset	Sampling	Intercept
P1	P1S1	16.082
P1	P1S2	18.852
P2	P2S1	47.410
P2	P2S2	27.340
P3	P3S1	17.981
P3	P3S2	16.776

Table 3: Linear mixed-effects model considering algorithms as a fixed effect and datasets as random effects.