Buskirk, Trent D. 2018. "Surveying the Forests and Sampling the Trees: An Overview of Classification and Regression Trees and Random Forests with Applications in Survey Research." *Survey Practice* 11 (1). https://doi.org/10.29115/SP-2018-0003.

ARTICLES

Surveying the Forests and Sampling the Trees: An overview of Classification and Regression Trees and Random Forests with applications in Survey Research

Trent D. Buskirk[1]

[1] Center for Survey Research and Department of Management Science and Information Systems, Umass Boston

Keywords: random forests, classification and regression trees

https://doi.org/10.29115/SP-2018-0003

Survey Practice

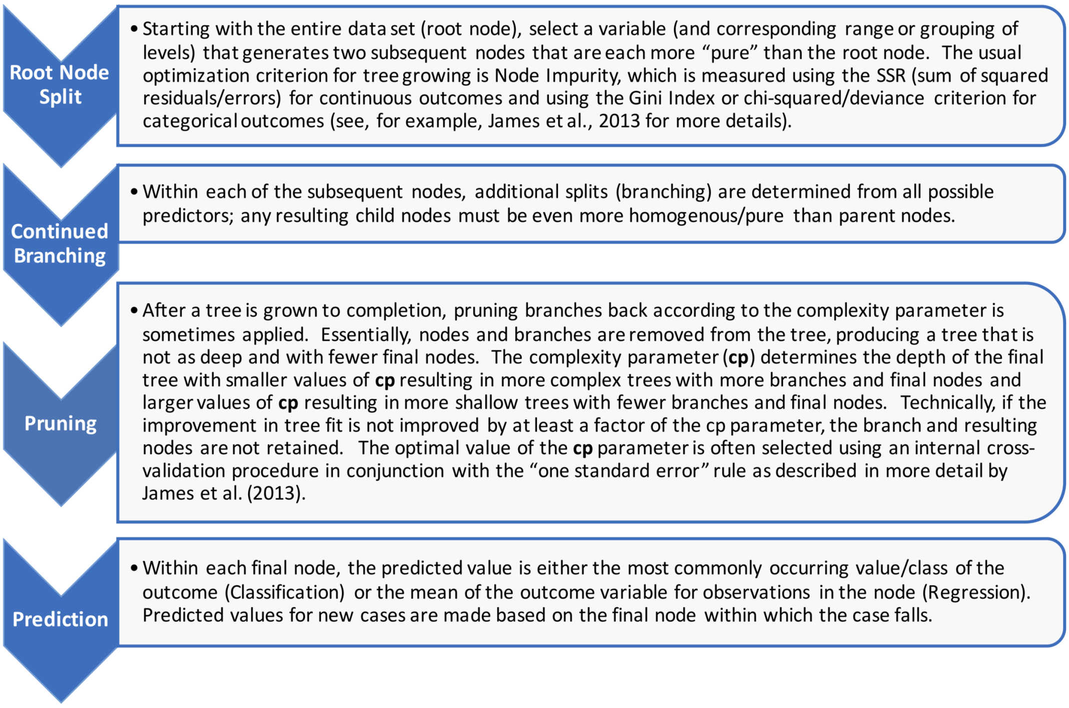
Vol. 11, Issue 1, 2018

While survey and social science researchers have become well versed in traditional modeling approaches such as multiple regression or logistic regression, there are more contemporary nonparametric techniques that are more flexible in terms of model form and distributional assumptions. Classification and regression trees (CARTs) and random forests represent two of the methods that are being applied more commonly within the survey research context for creating nonresponse adjustments and for creating propensity scores to be used within the responsive/adaptive survey context. Both of these methods can be used for regression or classification related tasks and offer researchers and practitioners excellent alternatives to the more classical approaches. CARTs and random forests can be applied when typical statistical distributional assumptions are not likely satisfied and can incorporate interactions automatically. CART models can be estimated in the presence of missing data and random forest methods can adapt to the complexity of the dataset and can be estimated when the number of predictors is large relative to the sample size. This article provides an accessible description for both of these methods and illustrates their use by developing models that predict survey response from a collection of demographic variables known for both respondents and nonrespondents.

What are Classification and Regression Trees and How are They Constructed?

If you have ever used the popular chi-square automatic interaction detection (CHAID) (Kass 1980) method for predicting survey response or other market segmentation, you have been building tree-based models. Classification and regression trees (CARTs) (L. et al. 1984) represent another type of tree-based method for classification or prediction. Like CHAID, CART models can be applied to both categorical outcomes as well as continuous outcomes, but CART models extend the capabilities of CHAID models by allowing both categorical ***and*** continuous predictors.

Beginning with the entire dataset (also called the root node), CART models use a series of recursive binary splits based on evaluating every possible predictor to create partitions of the sample into more homogeneous subsets or nodes. Unlike CHAID analyses that utilize statistical tests of association, CART models evaluate and base splitting and node formation on a degree of homogeneity based on the deviance score or Gini index (see James et al. 2013 for more details) for categorical outcomes and the sum of squared errors for continuous variables. For categorical outcomes, the tree models use a "classification tree," and the predicted value for any case within a particular



Root Node Split

- Starting with the entire data set (root node), select a variable (and corresponding range or grouping of levels) that generates two subsequent nodes that are each more "pure" than the root node. The usual optimization criterion for tree growing is Node Impurity, which is measured using the SSR (sum of squared residuals/errors) for continuous outcomes and using the Gini Index or chi-squared/deviance criterion for categorical outcomes (see, for example, James et al., 2013 for more details).

Continued Branching

- Within each of the subsequent nodes, additional splits (branching) are determined from all possible predictors; any resulting child nodes must be even more homogenous/pure than parent nodes.

Pruning

- After a tree is grown to completion, pruning branches back according to the complexity parameter is sometimes applied. Essentially, nodes and branches are removed from the tree, producing a tree that is not as deep and with fewer final nodes. The complexity parameter (**cp**) determines the depth of the final tree with smaller values of **cp** resulting in more complex trees with more branches and final nodes and larger values of **cp** resulting in more shallow trees with fewer branches and final nodes. Technically, if the improvement in tree fit is not improved by at least a factor of the cp parameter, the branch and resulting nodes are not retained. The optimal value of the **cp** parameter is often selected using an internal cross-validation procedure in conjunction with the "one standard error" rule as described in more detail by James et al. (2013).

Prediction

- Within each final node, the predicted value is either the most commonly occurring value/class of the outcome (Classification) or the mean of the outcome variable for observations in the node (Regression). Predicted values for new cases are made based on the final node within which the case falls.

Figure 1 Steps in constructing a classification/regression tree.

final node is simply the most commonly occurring level in that final node. On the other hand, if the outcome variable is continuous, then the tree models use a "regression tree," and the predicted value for any case falling within a particular final node is computed as the mean of the outcome variable for cases in that final node. CART models require specification of a single tuning parameter, which essentially determines the complexity of the resulting tree often referred to as *cp*. The *cp* parameter specifies the factor by which further splitting of the tree improves the overall measure of homogeneity. Larger values of *cp* result in shallow trees with fewer final nodes, while smaller values result in deeper trees with more final nodes. Setting the value of *cp* essentially controls the amount of pruning that happens for an otherwise fully grown tree. Pruning or trimming a tree is a critical step in the final tree model as it controls the amount of overfitting that is tolerated. More specific details regarding the steps taken to construct CART models are provided in Figure 1. Popular packages for implementing classification and regression trees in R are highlighted in Table 1.

Advantages and Disadvantages of Classification and Regression Trees

One of the most appealing aspects of CART models is that once constructed they can be plotted and resemble real "upside" down trees with a root node at the top and the branches and final nodes (or leaves) at the bottom of the plot. The visual nature of the tree can also be translated into a series of steps or rules that can be used to make predictions. However, tree models do have

Table 1 Popular packages for implementing classification and regression trees in R.

R Package Name	Brief Description
tree	This package is the primary implementation of CARTs in R. The package implements both classification and regression tree models. https://cran.r-project.org/web/packages/tree/tree.pdf
rpart	Recursive partitioning tree-based models for classification, regression and survival applications that implements most of the functionality discussed in the earlier works of Breiman et al. (1984). Tends to be faster than the tree package. https://cran.r-project.org/web/packages/rpart/rpart.pdf

Table 2 Additional advantages and disadvantages of using CARTs.

Major Advantages of CARTs	Major Disadvantages of CARTs
CART models are nonparametric and flexible in that they don’t assume a functional form between outcome and predictors.	Single trees are likely to have sub-optimal predictive performance compared to other methods (Kuhn and Johnson 2013).
CART models naturally detect higher level interactions among the predictors.	CART models are based on splits that depend on previous splits; so if an error is made in a higher split it can propagate down the tree.
CART models produce a series of rules for classification that are easy to interpret and implement in field work (e.g. adaptive survey interviewing protocols).	Because of the conditional nature of the fitting, CART models can also be very sensitive to changes in the underlying data set.
CART models can handle missing data through the use of surrogate predictors.	CART models generally consider all predictor variables at each step of branching and thus cannot “force” variables to be included *a priori.* (e.g. cannot create a model to predict nonresponse that must include demographics first, for example).
CART models are computationally fast	

the potential for overfitting, resulting in estimates with little bias but higher variance. Other major advantages and disadvantages are summarized in Table 2.

How Have CARTs Been Used in Survey Research?

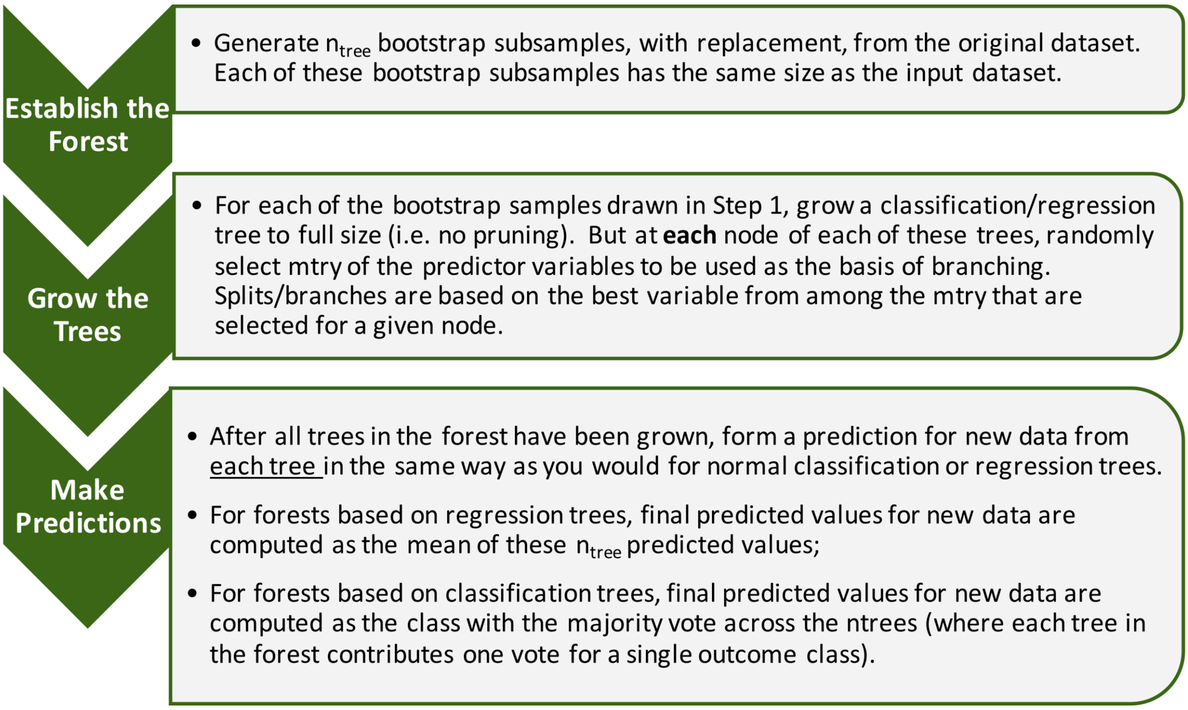
The application of CARTs to various aspects of the survey process has grown steadily in the past decade. For example, McCarthy and Earp (2009) used classification trees to investigate factors related to survey reporting errors. Garber (2009) used classification trees to predict eligibility of units included in a master mailing list for a survey targeting farms. Burgette and Reiter (2010) use regression trees as part of a multiple imputation strategy for continuous health-related survey outcomes such as birth weight. Phipps and Toth (2012) applied regression trees to data from the Occupational Employment Statistics Survey to estimate response propensities for sampled establishments. They also used a second regression tree to examine the potential of nonresponse bias in reported wages.

What are Random Forests Models and How are They Constructed?

Developed by Breiman (2001), random forests are ensemble-based methods that generate estimates by combining the results from a collection (i.e., the ensemble) of classification or regression trees. More specifically, if the outcome of interest is continuous, then a random forests model produces an estimate of the outcome by averaging the estimates derived from a series of regression trees. On the other hand, if the outcome is binary, a random forest generates an estimate defined as the level that is predicted most often among a collection of classification trees. By combining results across an ensemble of trees, random forests avoid the overfitting tendency of any single tree and generate predictions with lower variance compared to those obtained from a single tree (Breiman 2001; James et al. 2013). Each tree in the forest is grown using an independent bootstrap subsample that is the same size as the original dataset and selected with replacement from it. While not as commonly used for this purpose, response propensities can be estimated from random forests as the fraction of trees in the forest that predict a returned survey for a given address (see, for example, Buskirk and Kolenikov 2015). We note that the more common approach with binary outcomes is for the random forests to generate an estimated class for each sampled case (e.g., respondent or not).

Splitting each tree in the forest occurs one node at a time, and each tree is grown as large as possible. The number of variables considered for splitting is restricted to a random subsample of all possible predictor variables and represents the first tuning parameter for random forests. The size of this subsample is the same for each node and each tree, and is generally referred to as “mtry.” Large values of mtry result in more correlated trees, reproducing the overfitting behavior that is typical for single trees. The most commonly used value for mtry that balances error with predictive power for classification is the square root of the total number of predictor variables rounded down to the nearest whole number and p/3 for regression with p predictors (Breiman 2001). Apart from mtry, the other tuning parameter is the number of trees to be included in the forest. In practical applications, this value typically ranges from 100 to 1,000, with more trees providing more accurate and more stable estimates at the expense of computing time. For continuous outcomes, there is one additional parameter called node size, which determines whether additional splitting on a node can occur or not. If the number of data points that fall in a node is larger than this threshold, additional splitting occurs; otherwise, the node becomes a terminal node in a given tree. The default value for node size for trees within forests applied to continuous outcomes is 5.

The overall prediction error of the random forest is generally a nonincreasing, bounded function of the number of trees, meaning that after a certain number of trees, the additional reduction in error from adding additional trees to the forest becomes negligible (Breiman 2001). However, it is also completely possible for a smaller forest to produce similar accuracy rates as a larger forest



- Generate n_{tree} bootstrap subsamples, with replacement, from the original dataset. Each of these bootstrap subsamples has the same size as the input dataset.

- For each of the bootstrap samples drawn in Step 1, grow a classification/regression tree to full size (i.e. no pruning). But at **each** node of each of these trees, randomly select mtry of the predictor variables to be used as the basis of branching. Splits/branches are based on the best variable from among the mtry that are selected for a given node.

- After all trees in the forest have been grown, form a prediction for new data from each tree in the same way as you would for normal classification or regression trees.
- For forests based on regression trees, final predicted values for new data are computed as the mean of these n_{tree} predicted values;
- For forests based on classification trees, final predicted values for new data are computed as the class with the majority vote across the ntrees (where each tree in the forest contributes one vote for a single outcome class).

Figure 2 Steps in constructing random forest models.

Table 3 Popular packages for implementing random forest models in R.

R package name	Brief description
randomForest	RandomForest is one of the more popular and implementations of Brieman's (2001) random forest method. https://cran.r-project.org/web/packages/randomForest/index.html
randomForestSRC	randomForestSRC provides a unified treatment of Brieman's (2001) random forest models applied to regression, classification, and survival analysis problems. https://cran.r-project.org/web/packages/randomForestSRC/index.html
ggRandomForests	This package provides graphical methods for exploring random forest models created using either the randomForest or randomForestSRC packages. https://cran.r-project.org/web/packages/ggRandomForests/index.html
ranger	This package offers a fast implementation of random forest models for classification, regression, survival, and probability prediction trees. https://cran.r-project.org/web/packages/ranger/index.html

(Goldstein, Polley, and Briggs 2011). While the value of mtry can impact the overall prediction accuracy of the forest, studies have indicated that the overall results tend to be fairly robust with similar performance being achieved across a fairly wide range of values (Pal 2005). For continuous outcomes, it has been shown in practice that prediction error rates can be reduced by using larger values of the node size parameter beyond the default (Segal 2004). More details about random forests construction are provided in Figure 2. Popular packages for implementing random forest models in R are highlighted in Table 3.

Table 4 Additional advantages and disadvantages of random forest models.

Major advantages of random forests	Major disadvantages of random forests
Like tree methods, random forests can handle predictors that are continuous, categorical, skewed, and sparse data.	Missing data must be handled before applying random forests models.
Random forests are aptly suited for the “large p, small n” scenario (Strobl et al. 2007).	Measures of variable importance can be biased if the predictors are correlated.
Random forests can also be very effective for estimating outcomes that are a complex functions of predictors with many interactions or possibly a non-linear function of the parameters (Mendez et al., 2008).	Random forests can be computationally intensive.

Advantages and Disadvantages of Random Forest Models

As mentioned previously, the fact that random forests create estimates by aggregating over a series of trees generally implies less overfitting than a single tree model. Moreover, since random forests are grown based on bootstrap subsamples taken with replacement, they produce an internally valid and nearly unbiased estimate of performance. However, unlike tree models that are easy to visualize, random forests are not easily visualized. However, they can produce a ranking of variable importance for each possible predictor that can easily be displayed graphically. Other major advantages and disadvantages of random forests are provided in Table 4.

How Have Random Forests Been Used in Survey Research?

The use of random forest models in survey research has not been as common compared to tree-based models, but their use has steadily been increasing within the past 5 years. For example, Caiola and Reiter (2010) illustrated how random forests could be used to generate partially synthetic categorical data using data from the 2000 U.S. Current Population Survey. Buskirk, West, and Burks (2013) investigated the use of random forests for estimating response propensities, which were then applied to sampled units on subsequent cross-sectional surveys at later time points to estimate the propensity to respond. Earp et al. (2014) investigated the use of a random forest-like ensemble of trees for evaluating nonresponse bias for establishment surveys. Buskirk and Kolenikov (2015) compared logistic regression and random forest models for nonresponse adjustments to sampling weights based on propensity scores.

Classification Example

Using the National Health Interview Survey (NHIS) example training dataset, we estimated a main effects logistic regression, classification tree and random forest model to predict the simulated survey respondent outcome based on a set of core demographics as described previously in this paper. The three models were developed using the training dataset and applied to the testing dataset to evaluate various performance metrics including percentage correctly classified, sensitivity, specificity and the area under the ROC curve, and a measure of balanced accuracy — defined as the mean of the sensitivity and

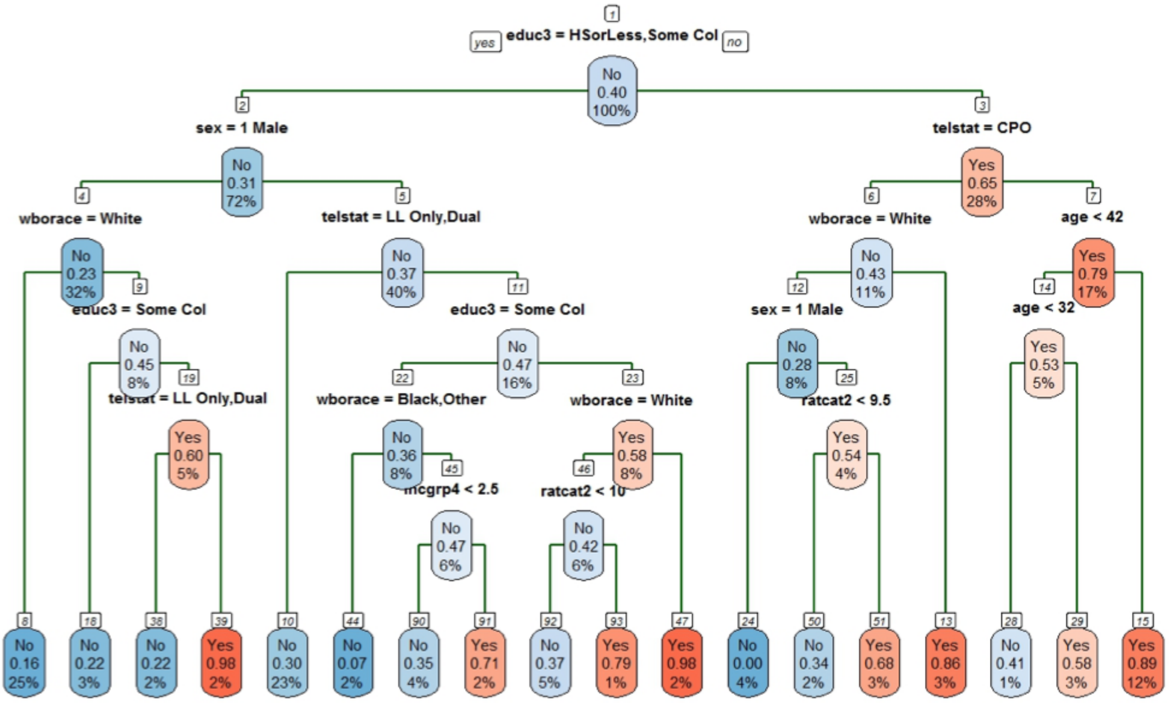


Figure 3 Classification tree model predicting survey response from demographic variables using the NHIS sample dataset.

specificity measures. The classification tree was computed using a *cp* value of 0.0022, which was determined using 10-fold cross-validation of the training data along with a one-standard error rule. The random forest model used a total of 1,001 trees based on preliminary testing along with the default value of the mtry tuning parameter.

The final classification tree model is provided in Figure 3, and the summary statistics for the accuracy of this model in comparison to the random forest and main effects logistic regression models are provided in Table 5. The final tree model contains 18 final nodes that are shown on the very bottom of the tree in Figure 3. Generally, tree models are read from the top down — if the condition specified at any given node is true, then proceed downward and to the left; otherwise, proceed downward and to the right. Continue making your way down the tree until you reach one of the final nodes. The left-most final node (number 8) represents white males who have less than a BS/BA degree. These individuals comprise 25% of the entire sample, and a minority of these individuals were respondents (16%). A future individual who is male, white, and has less than a BS/BA degree would be predicted to be a nonrespondent using this tree model.

Generally speaking, the random forest model outperformed both the tree and logistic regression models on a majority of the metrics, but both the forest and tree models outperformed the logistic regression model on all metrics. In particular, both the random forest and tree models were more specific than the logistic regression model (i.e., higher correct detection of non-respondents)

Table 5 Various statistics of model accuracy for predicting survey response status by applying the respective models, constructed using the training sample, to the test sample.

Statistic (estimated using a 16% hold-out test sample)	Main effects logistic regression model	Final classification tree model	Final random forest model
Accuracy (i.e., percentage correctly classified)	69.8%	78.3%	78.5%
Sensitivity (i.e., true positive rate)	49.6%	57.5%	60.3%
Specificity (i.e., true negative rate)	83.2%	92.1%	90.6%
Balanced accuracy (mean of sensitivity and specificity)	66.4%	74.8%	75.5%
Area under the ROC curve	74.2%	81.9%	83.8%

and had between 7 and 10 percentage points higher sensitivity values (i.e., higher correct detection of respondents). The same spread for the area under the curve was also realized for the forest and tree models compared to the logistic regression model. Since the binary outcome was simulated through a series of probit models involving nonlinear and interaction terms, we would expect lower performance from the main effects logistic regression. In addition, it is important to note that the nonparametric nature of the forest and tree models were able to approximate these more nonlinear and complex probit models and create predictions that had a relatively high level of accuracy and performance without having to specify the shape/structure of the underlying survey outcome model.

REFERENCES

Breiman, L. 2001. “Random Forests.” *Machine Learning* 45 (1): 5–32.

Burgette, L.F., and J.P. Reiter. 2010. “Multiple Imputation for Missing Data via Sequential Regression Trees.” *American Journal of Epidemiology* 172 (9): 1070–76.

Buskirk, T.D., and S. Kolenikov. 2015. “Finding Respondents in the Forest: A Comparison of Logistic Regression and Random Forest Models for Response Propensity Weighting and Stratification.” *Survey Insights: Methods from the Field, Weighting: Practical Issues and “How to” Approach*. http://surveyinsights.org/?p=5108.

Buskirk, T.D., B.T. West, and A.T. Burks. 2013. “Respondents: Who Art Thou? Comparing Internal, Temporal, and External Validity of Survey Response Propensity Models Based on Random Forests and Logistic Regression Models.” In *The 2013 Joint Statistical Meetings*. Montreal, Canada.

Caiola, G., and J.P. Reiter. 2010. “Random Forests for Generating Partially Synthetic, Categorical Data.” *Transactions on Data Privacy* 3 (1): 27–42.

Earp, M., M. Mitchell, J. McCarthy, and F. Kreuter. 2014. “Modeling Nonresponse in Establishment Surveys: Using an Ensemble Tree Model to Create Nonresponse Propensity Scores and Detect Potential Bias in an Agricultural Survey.” *Journal of Official Statistics* 30 (4): 701–19.

Garber, S.C. 2009. “Census Mail List Trimming Using SAS Data Mining.” In *Department of Agriculture, National Agricultural Statistics Service, RDD Report 09-02*. Fairfax, VA.

Goldstein, B.A., E.C. Polley, and F.B.S. Briggs. 2011. “Random Forests for Genetic Association Studies.” *Statistical Applications in Genetics and Molecular Biology* 10 (1): 32.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. *An Introduction to Statistical Learning with Applications in R*. New York, NY: Springer.

Kass, G.V. 1980. “An Exploratory Technique for Investigating Large Quantities of Categorical Data.” *Applied Statistics* 29 (2): 119–27.

Kuhn, M., and K. Johnson. 2013. *Applied Prediction Modeling*. New York, NY: Springer.

L., Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. 1984. *Classification and Regression Trees*. Boca Raton, FL: Wadsworth; Taylor & Francis Group.

McCarthy, J., and M. Earp. 2009. “Who Makes Mistakes? Using Data Mining Techniques to Analyze Reporting Errors in Total Acres Operated.” In *US Department of Agriculture, National Agricultural Statistics Service, RDD Report 09-05*. Fairfax, VA.

Pal, M. 2005. “Random Forest Classifier for Remote Sensing Classification.” *International Journal of Remote Sensing* 26 (1): 217–22.

Phipps, P., and D. Toth. 2012. “Analyzing Establishment Nonresponse Using an Interpretable Regression Tree Model with Linked Administrative Data.” *The Annals of Applied Statistics* 6 (2): 772–94.

Segal, Mark R. 2004. “Machine Learning Benchmarks and Random Forest Regression. Center for Bioinformatics & Molecular Biostatistics.” 2004. https://escholarship.org/uc/item/35x3v9t4.

Strobl, C., A.-L. Boulesteix, A. Zeileis, and T. Hothorn. 2007. “Bias in Random Forest Variable Importance Measures: Illustrations, Sources and a Solution.” *BMC Bioinformatics* 8 (25).

SUPPLEMENTARY MATERIALS

Example R-Code

Download: https://www.surveypractice.org/article/2709-surveying-the-forests-and-sampling-the-trees-an-overview-of-classification-and-regression-trees-and-random-forests-with-applications-in-survey-resear/attachment/9426.zip