Tree-based Machine Learning in Survey Research

Christoph Kern[1,2]

[1]University of Mannheim, School of Social Sciences

[2]University of Maryland, JPSM

UN Stats Division Brown Bag 11/20/2019

Outline

Introduction

- Machine learning (ML) methods provide a vast set of tools for exploring and analyzing diverse data
- Comprise flexible/ non-parametric methods that adapt to complex data structures
- Focus on out-of-sample prediction performance
  - Learn $f(x)$ with training set $\rightarrow$ predict on test set

- ML increasingly used by survey researchers in various contexts (Buskirk et al. 2018, Kern et al. 2019)
- A promising *supplement* in the survey methods toolkit

$\rightarrow$ This talk focuses on **tree-based methods** as an important subgroup of ML methods

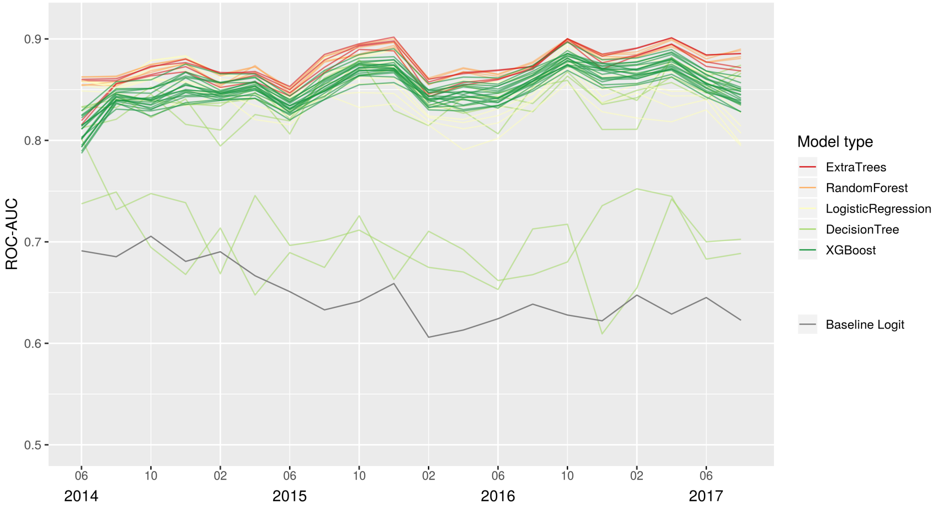
ML and Survey Research

Some examples

- Create sampling frame based on satellite images (Eckman and Qiu 2019)
- Estimate response propensities for weighting (Buskirk and Kolenikov 2015)
- Predict matches in record linkage (Schild et al. 2017)
- Create synthetic data (Drechsler and Reiter 2011)
- Predict nonreponse in panel surveys (Kern et al. 2019)
- Estimate propensity scores (McCaffrey et al. 2004)

ML and Survey Research

Figure: Predicting panel nonresponse with different model types[1]



[1] https://arxiv.org/abs/1909.13361

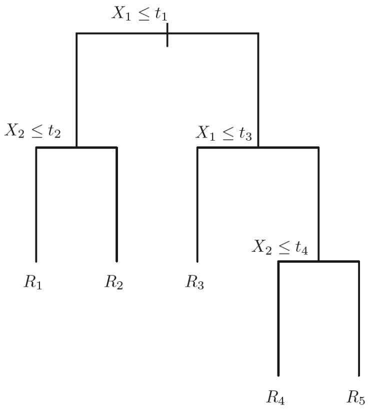
Tree-based (Ensemble) Methods

Classification and Regression Trees (CART)

Decision Trees

- Approach for partitioning the predictor space into smaller subregions
- Results in a "top-down" tree structure
- Important building block (base learner) for ensemble methods
- Many different tree building algorithms exist (Loh 2014)
  - **CART**, CHAID, CTREE...

Figure: A small tree



Tree growing

Growing a **regression tree**

Define pairs of regions for all $X_1, X_2, ..., X_p$ predictors and cutpoints c

$$\tau_L(j, c) = \{X | X_j < c\} \text{ and } \tau_R(j, c) = \{X | X_j \geq c\}$$

Find split s which maximizes the reduction in RSS

$$\Delta RSS(s, \tau) = RSS(\tau) - RSS(\tau_L) - RSS(\tau_R)$$

$$RSS(\tau) = \sum_{i \in \tau} (y_i - \hat{y})^2$$

with $\hat{y}$ being the mean of y in node τ

Tree growing

Growing a **regression tree**

Define pairs of regions for all $X_1, X_2, ..., X_p$ predictors and cutpoints c

$$\tau_L(j,c) = \{X|X_j < c\} \text{ and } \tau_R(j,c) = \{X|X_j \geq c\}$$

Find split s which maximizes the reduction in RSS

$$\Delta RSS(s,\tau) = RSS(\tau) - RSS(\tau_L) - RSS(\tau_R)$$

$$RSS(\tau) = \sum_{i \in \tau} (y_i - \hat{y})^2$$

with $\hat{y}$ being the mean of y in node τ

Tree growing

Growing a **classification tree**

Find split s which maximizes the reduction in node impurity

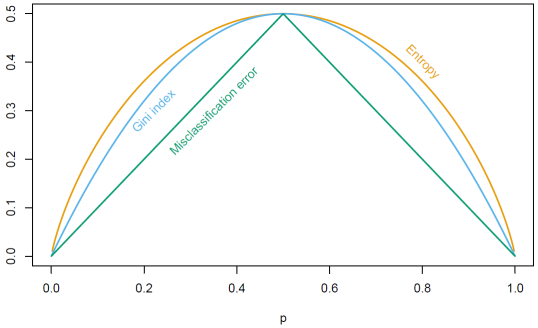
$$\Delta I(s,\tau) = I(\tau) - p(\tau_L)I(\tau_L) - p(\tau_R)I(\tau_R)$$

Gini Impurity

$$I_{Gini}(\tau) = \sum_{k=1}^{K} \hat{p}_k(1-\hat{p}_k)$$

with $\hat{p}_k$ being the proportion of observations from class k in node τ

Figure: Impurity measures (Hastie et al. 2009)



Tree growing

Algorithm 1: Tree growing process

```
Define stopping criteria;
Assign training data to root node;
if stopping criterion is reached then
    end splitting;
else
    find the optimal split point;
    split node into two subnodes at this split point;
    for each node of the current tree do
        continue tree growing process;
    end
end
```

Tree structure

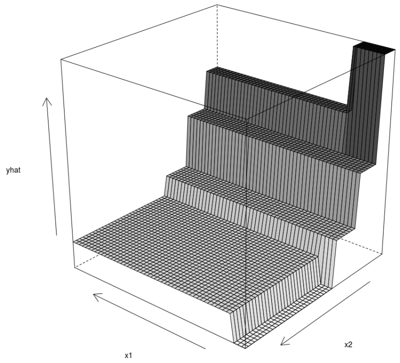
A given tree

$$\mathcal{T} = \sum_{m=1}^{M} \gamma_m \cdot 1_{(i \in \tau_m)}$$

consists of a set of $m = 1, 2, \ldots, M$ nodes which can be used for prediction by...

- Regression
  - ...using the mean of y for training observations in τ_m
- Classification
  - ...going with the majority class in τ_m

Figure: Tree prediction surface



Tree pruning

Stopping rules

- Minimum number of cases in terminal nodes
- Decrease in impurity exceeds some threshold

$\rightarrow$ However, worthless splits can be followed by good splits

Cost complexity pruning

$$R_\alpha(\mathcal{T}) = R(\mathcal{T}) + \alpha|\mathcal{T}|$$

- Find the best subtree by balancing quality $R(\mathcal{T})$ and complexity $|\mathcal{T}|$
- α controls the penalty on the number of terminal nodes
- α can be chosen through CV

Bagging and Random Forest

Bagging Trees

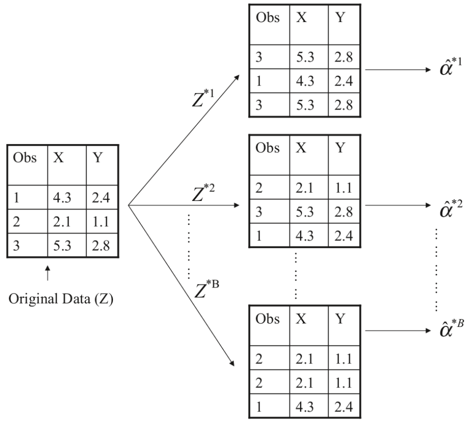
Some limitations of CART

- Lack of smoothness of prediction surface
- High variance/ instability due to hierarchical splitting process

→ **Ensemble methods**

- Address instability via combining multiple prediction models
- Combine diverse models into a more robust ensemble
- e.g. **Bagging**: Bootstrap Aggregating

Figure: Bootstrap process (James et al. 2013)



Bagging Trees

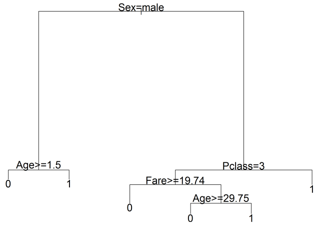
Algorithm 2: Bagging Trees

```
1  Set number of trees $B$;
2  Define stopping criteria;
3  for $b = 1$ to $B$ do
4      draw a bootstrap sample from the training data;
5      assign sampled data to root node;
6      if stopping criterion is reached then
7          end splitting;
8      else
9          find the optimal split point among the predictor space;
10         split node into two subnodes at this split point;
11         for each node of the current tree do
12             continue tree growing process;
13         end
14     end
15 end
```

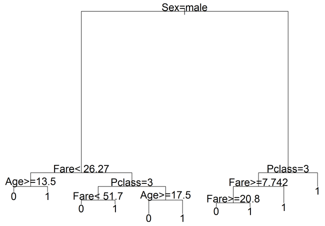
Bagging Trees

Figure: Bagging Trees

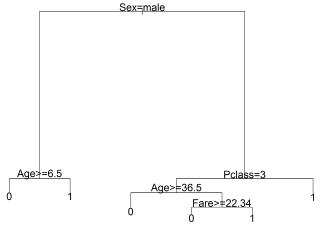
(a) b = 1



(b) b = 2



(c) b = 3



Random Forests

From Bagging to Random Forests

Variance of an average of B i.i.d. random variables

$$\frac{1}{B}\sigma^2$$

$\rightarrow$ Bagging: Averaging over B trees decreases variance

Variance of an average of B i.d. random variables with $\rho > 0$

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

$\rightarrow$ **Random Forests**: Averaging over B trees with m out of p predictors per split decreases variance and decorrelates trees

Random Forests

From Bagging to Random Forests

Variance of an average of B i.i.d. random variables

$$\frac{1}{B}\sigma^2$$

$\rightarrow$ Bagging: Averaging over B trees decreases variance

Variance of an average of B i.d. random variables with $\rho > 0$

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

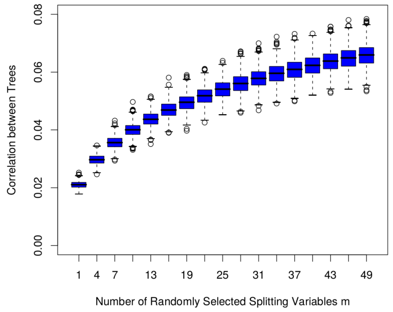
$\rightarrow$ **Random Forests**: Averaging over B trees with m out of p predictors per split decreases variance and decorrelates trees

Random Forests

The Random Forest trick (Breiman 2001)

- Randomization with respect to rows *and* columns
- Weaker predictors have more of a chance
- Results in diverse and *decorrelated* trees

Figure: Correlations between pairs of trees (Hastie et al. 2009)



Growing a Forest

Algorithm 3: Grow a Random Forest

```
Set number of trees $B$;
Set predictor subset size $m$;
Define stopping criteria;
for $b = 1$ to $B$ do
    draw a bootstrap sample from the training data;
    assign sampled data to root node;
    if stopping criterion is reached then
        end splitting;
    else
        draw a random sample $m$ from the $p$ predictors;
        find the optimal split point among $m$;
        split node into two subnodes at this split point;
        for each node of the current tree do
            continue tree growing process;
        end
    end
end
```

Growing a Forest

A Random Forest

$$\{\mathcal{T}_b\}_1^B$$

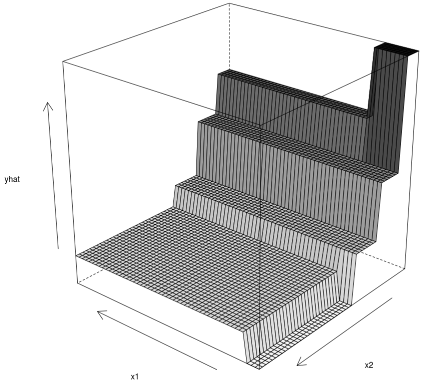
consists of a set of $b = 1, 2, \ldots, B$ trees which can be used for prediction by...

- Regression
  - Averaging predictions over all trees
  - $\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^B \mathcal{T}_b(x)$
- Classification
  - Using most commonly occurring class among all trees
  - $\hat{C}_{rf}^B(x) = \text{majority vote}\{\hat{C}_b(x)\}_1^B$
- Probability estimation
  - Using the proportion of class votes of all trees
  - Averaging predicted probabilities over all trees

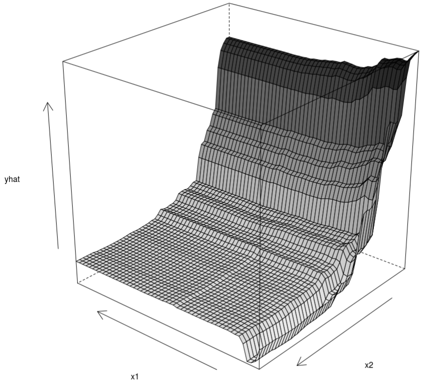
RF vs. CART

Figure: Prediction surface (example)

(a) CART



(b) Random Forest



Tuning RF

Tuning Random Forests

- Predictor subset size m out of p (`mtry`)
  - Most important tuning parameter in RF
  - Starting value; $m = \sqrt{p}$ (classification), $m = p/3$ (regression)
  - Can be chosen using OOB errors based on different m
- Number of trees
  - sufficiently high (e.g. 500)
- Node size (number of observations in terminal nodes)
  - sufficiently low (e.g. 5)

Boosting

Boosting

Boosting methods

- Class of ensemble methods which combine **sequential** prediction models
- Adaptive approach with focus on “difficult observations”
- Different flavors exist
  - AdaBoost
  - Gradient Boosting Machines (GBM)
  - XGBoost
  - ...
- Can be applied to different (weak) base learners
  - Boosting trees
  - Model-based boosting
  - ...

AdaBoost

- Algorithm for classification problems ($Y \in \{-1, 1\}$)
- Estimate a sequence of classifiers using reweighted data
- **AdaBoost process**
  1. Fit classifier $G_m(x)$ to weighted data (intitial weights $w_i = \frac{1}{n}$)
  2. Compute the misclassification rate

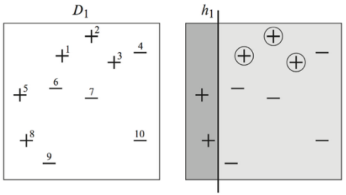
$$\text{err}_m = \frac{\sum_{i=1}^{n} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{n} w_i}$$

  3. Compute the classifier weight $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$
  4. Recalculate weights $w_i = w_i \exp(\alpha_m I(y_i \neq G_m(x_i)))$
- Majority vote classification: $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$

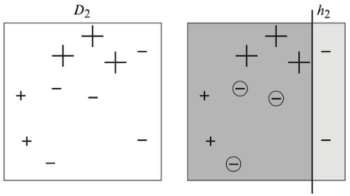
Boosting Stumps

Figure: (Ada)Boosting stumps (example)[2]

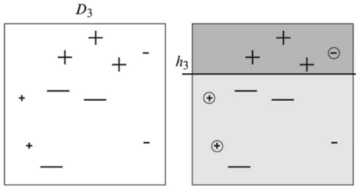
(a) Step 1: $\alpha_1 = 0.42$



(b) Step 2: $\alpha_2 = 0.65$



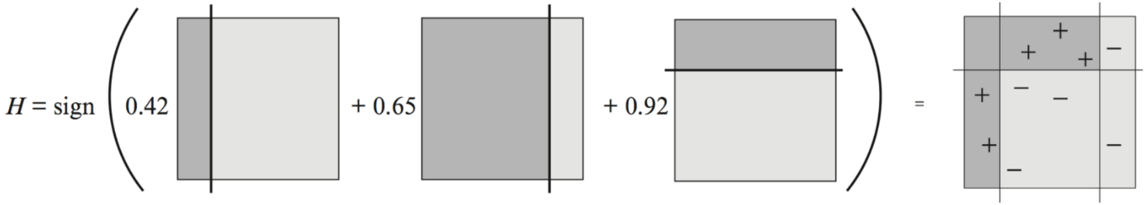
(c) Step 3: $\alpha_3 = 0.92$



[2] Source: Shapire & Freund 2012

Boosting Stumps

Figure: Step 4: Combine models



Gradient Boosting Machines

- General approach to sequential learning
- Applicable with various loss functions
- **Boosting trees**
  1. Initialize model (with a constant $f_0(x)$)
  2. Compute pseudo-residuals based on current model

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$$

  3. Fit a regression tree to the pseudo-residuals
  4. Compute $\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$
  5. Update the current model: $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$
- Output $\hat{f}(x) = f_M(x)$

$\rightarrow$ Analogue to steepest descent

Gradient Boosting Machines

Table: GBM components for different loss functions

Setting	Loss function	r_i	$f_0(x)$
Regression	$\frac{1}{2}(y_i - f(x_i))^2$	$y_i - f(x_i)$	mean(y_i)
Regression	$\lvert y_i - f(x_i) \rvert$	sign$(y_i - f(x_i))$	median(y_i)
Classification	Deviance	$I(y_i = G_k) - p_k(x_i)$	prior p's

Shrinkage, Subsampling, Tuning

Shrinkage

- Additional tweak in Gradient boosting
- Slow down learning rate to avoid overfitting
- Learning rate is controlled by λ
    - $f_m(x) = f_{m-1}(x) + \lambda \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

Subsampling

- Optional add-on in Gradient boosting
- Use a random sample (w/o replacement) of pseudo-residuals in each step
- Can be introduced to improve performance and speed
    - "Stochastic gradient boosting"

Shrinkage, Subsampling, Tuning

Tuning Gradient Boosting Machines

- Number of trees M
    - Number of "iterations"
    - Overfitting can occur for large M
- Interaction depth D
    - Number of splits for each tree
    - Boosting stumps: $D = 1$
- Shrinkage parameter λ
    - e.g. $\lambda = 0.01$, $\lambda = 0.001$
    - Smaller λ needs larger M
- ...

Boosting for Regression

Algorithm 4: Gradient Boosting for regression

```
1  Set number of trees $M$;
2  Set interaction depth $D$;
3  Set shrinkage parameter $\lambda$;
4  Use $\bar{y}$ as initial prediction;
5  for $m = 1$ to $M$ do
6      compute residuals based on current predictions;
7      assign data to root node, using the residuals as the outcome;
8      while current tree depth $< D$ do
9          tree growing process;
10     end
11     compute the predicted values of the current tree;
12     add the shrinked new predictions to the previous predicted values;
13 end
```

Interpretable ML

Interpretable ML

Interpreting Random Forests, Boosting models

- Inspect each tree of the ensemble
  - Inefficient for 500+ trees
- Variable importance
  - Summary of “effect size”
- Partial dependence plots
  - Graphical representation of “effect structure”
- ...

Variable Importance

Variable importance with CART

$$\mathcal{I}_\ell^2(T) = \sum_{t=1}^{J-1} \hat{\imath}_t^2 I(v(t) = \ell)$$

- Sum of squared improvements $\hat{\imath}^2$ over all internal nodes with predictor X_ℓ
  - Regression: Overall reduction in RSS caused by X_ℓ
  - Classification: Overall reduction of impurity caused by X_ℓ

Importance with Random Forests

$$\mathcal{I}_\ell^2 = \frac{1}{M} \sum_{m=1}^{M} \mathcal{I}_\ell^2(T_m)$$

- Average improvement caused by predictor X_ℓ over all trees

Variable Importance

Permutation feature importance (Fisher et al. 2018)

1. Estimate the original model error $e_{orig}(\hat{f}) = L(Y, \hat{f}(X))$
2. For each feature $j \in 1, \ldots, p$
    1. Generate feature matrix $X_{perm\,j}$ by permuting the values of feature X_j in X
    2. Estimate error $e_{perm} = L(Y, \hat{f}(X_{perm\,j}))$ based on the predictions of the permuted data
    3. Calculate permutation feature importance $FI_j = \frac{e_{perm}(\hat{f})}{e_{orig}(\hat{f})}$ or via $FI_j = e_{perm}(\hat{f}) - e_{orig}(\hat{f})$
3. Output FI for all variables

Partial Dependence Plots

Plotting feature effects in “black box” learning methods

1. Choose a range of values $\{x_{11}, x_{12}, \ldots, x_{1k}\}$ of x_1
2. For each $i \in \{1, 2, \ldots, k\}$
    1. Generate an artificial dataset by fixing x_1 to x_{1i} for all cases
    2. Compute predictions for all cases using the prediction model (e.g. RF)
    3. Average the predictions over all cases
3. Plot the obtained average predictions against x_{1i} for $i = 1, 2, \ldots, k$

ICE and ALE

ICE plots (Goldstein et al. 2014)

- Individual PDPs for all cases w/o final averaging
- One line represents the predictions for one case over the range of x
- Can uncover heterogeneous effects that are driven by interactions
- Centered ICE plots
  - Adjust for different individual baselines

ALE plots (Apley 2016)

- With correlated features, PDPs can (artificially) construct very unlikely combinations
- ALE solution:
  1. Use only cases with (similar) x-values within a given interval
  2. Calculate differences in predictions between upper and lower limit of this interval

More interpretable ML

- Global and local surrogate models
  - `https://arxiv.org/abs/1602.04938`
- Shapley values and SHAP
  - `https://arxiv.org/abs/1705.07874`
- Feature interaction (H-statistic)
  - `https://arxiv.org/abs/0811.1679`
- Partial dependence-based variable importance
  - `https://arxiv.org/pdf/1805.04755.pdf`
- Representative trees from ensembles
  - `https://www.ncbi.nlm.nih.gov/pubmed/22302520`

Summary and Resources

Summary

Tree methods

- Divide-and-conquer strategy that splits the data into subgroups
- No need to specify the functional form in advance (unlike regression)
- Non-linearities and interactions are handled automatically
- Limitations of CART: Instability, competition among correlated predictors, biased variable selection
- Bagging, RF stabilize predictions from high-variance methods
- Boosting sequentially combines multiple models into a powerful ensemble

Software Resources

Resources for R

- Standard package to build CARTs: `rpart`
  - Unified infrastructure for tree representation: `partykit`
- Standard package to grow RFs: `randomForest`
  - Fast implementation of RFs: `ranger`
- Standard package for Gradient Boosting: `gbm`
  - Extreme Gradient Boosting: `xgboost`
- Interpretable Machine Learning in R: `iml`

Books

Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer.

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning*. New York, NY: Springer.

Molnar, C. (2019). Interpretable Machine Learning. A Guide for Making Black Box Models Explainable. `https://christophm.github.io/interpretable-ml-book/`.

Contact

c.kern@uni-mannheim.de

References

Buskirk, T. D., Kolenikov, S. (2015). Finding respondents in the forest: A comparison of logistic regression and random forest models for response propensity weighting and stratification. Survey Methods: Insights from the Field. https://surveyinsights.org/?p=5108

Buskirk, T. D., Kirchner, A., Eck, A., and Signorino, C. S. (2018). An introduction to machine learning methods for survey researchers. Survey Practice, 11(1).

Eckman, S., Qiang Qiu, Q. (2019). Detecting Housing Units from Satellite Imagery with Computer Vision. https://www.stepheckman.com/talk/aapor_201905/

Drechsler, J., Reiter, J. P. (2011). An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets. *Computational Statistics & Data Analysis 55*(12), 3232–3243.

Kern, C., Klausch, T., and Kreuter, F. (2019). Tree-based machine learning methods for survey research. Survey Research Methods, 13(1):73–93.

McCaffrey, D. F., Ridgeway, G., and Morral, A. R. (2004). Propensity score estimation with boosted regression for evaluating causal effects in observational studies. *Psychological Methods 9*(4), 403–425.

Schild, C.-J., Schultz, S., F. Wieser (2017). Linking Deutsche Bundesbank Company Data using Machine-Learning-Based Classification. Technical Report 2017-01, Deutsche Bundesbank Research Data and Service Centre.

References

Apley, D. W. (2016). Visualizing the effects of predictor variables in black box supervised learning models. https://arxiv.org/abs/1612.08468.

Breiman, L. (2001). Random forests. *Machine Learning 45*(1), 5–32.

Fisher, A., Rudin, C., Dominici, F. (2018). Model Class Reliance: Variable importance measures for any machine learning model class, from the 'Rashomon' perspective. http://arxiv.org/abs/1801.01489.

Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E. (2014). Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. https://arxiv.org/abs/1309.6392.

Loh, W.-Y. (2014). Fifty Years of Classification and Regression Trees. *International Statistical Review 82*(3), 329–348.

Schapire, R. E. and Freund, Y. (2012). *Boosting: Foundations and Algorithms*. Cambridge, MA: MIT Press.