Lecture 22:
Ensembles: Bagging, Random Forests, & Intro to Boosting

Big Data and Machine Learning for Applied Economics
Econ 4676

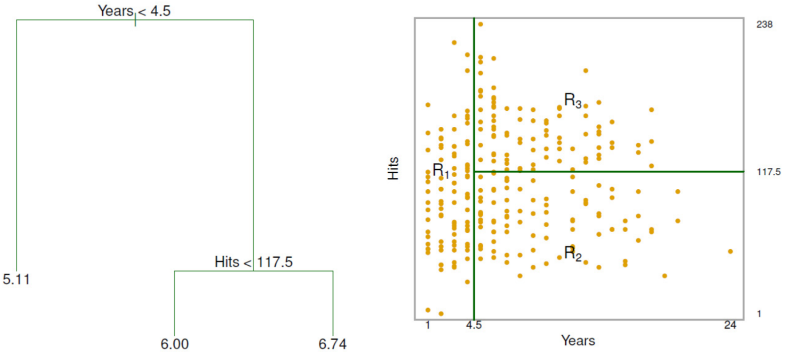
Ignacio Sarmiento-Barbieri

Universidad de los Andes

October 28, 2021

Agenda

CARTs



Years < 4.5
5.11
Hits < 117.5
6.00
6.74
238
117.5
1
Hits
R1
R2
R3
1
4.5
24
Years

CARTs

- Problem then boils down to searching the partition variable X_j and the partition point s such that

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i \in R_1(j,s)}(y-c_1)^2+\min_{c_2}\sum_{x_i \in R_2(j,s)}(y-c_2)^2\right] \tag{1}$$

- For each partition variable, and partition point, the internal minimization is the mean of each region

$$\hat{c}_m = \frac{1}{n_m}\sum(y_i|x_i \in R_m) \tag{2}$$

- Process is repeated inside each region.

CARTs

- Problem then boils down to searching the partition variable X_j and the partition point s such that

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y-c_2)^2\right] \tag{1}$$

- For each partition variable, and partition point, the internal minimization is the mean of each region

$$\hat{c}_m=\frac{1}{n_m}\sum(y_i|x_i\in R_m) \tag{2}$$

- Process is repeated inside each region.
- If the final tree has M regions then the prediction is

$$\hat{f}(x)=\sum_{m=1}^{M}\hat{c}_m I(x\in R_m) \tag{3}$$

CARTs

- Cost complexity of tree T

$$C_\alpha(T) = \sum_{m=1}^{[T]} n_m Q_m(T) + \alpha[T] \tag{4}$$

  - where $Q_m(T) = \frac{1}{n_m}\sum_{x_i \in R_m}(y_i - \hat{c}_m)^2$ for regression trees
  - $Q_m(T)$ penalizes heterogeneity (impurity) within each region, and the second term the number of regions.
  - Objective: for a given α, find the optimal pruning that minimizes $C_\alpha(T)$

CARTs

- Smart way to represent nonlinearities. Most relevant variables on top.
- Very easy to communicate.
- Reproduces human decision-making process.
- Trees are intuitive and do OK, but
    - They are not very good at prediction
    - If the structure is linear, CART does not work well.
    - Not very robust

Bagging

- We can improve performance a lot using either bootstrap aggregation (bagging), random forests, or boosting.
- Bagging & Random Forests:
  - Repeatedly draw bootstrap samples $(X_i^b, Y_i^b)_{i=1}^N$ from the observed sample.

Bagging

- We can improve performance a lot using either bootstrap aggregation (bagging), random forests, or boosting.
- Bagging & Random Forests:
  - Repeatedly draw bootstrap samples $(X_i^b, Y_i^b)_{i=1}^N$ from the observed sample.
  - For each bootstrap sample, fit a regression tree $\hat{f}^b(x)$

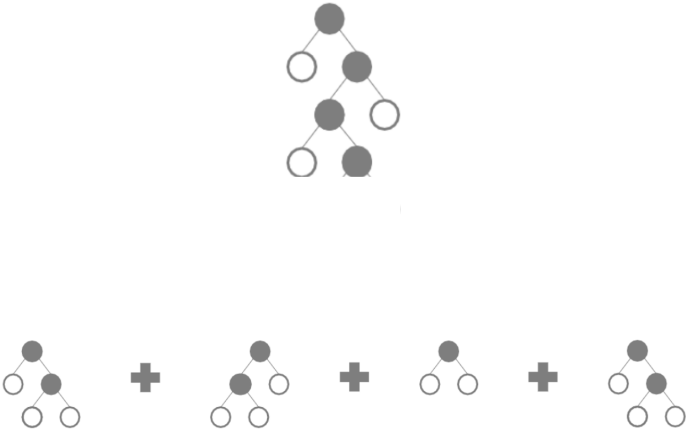
Bagging

- We can improve performance a lot using either bootstrap aggregation (bagging), random forests, or boosting.
- Bagging & Random Forests:
  - Repeatedly draw bootstrap samples $(X_i^b, Y_i^b)_{i=1}^N$ from the observed sample.
  - For each bootstrap sample, fit a regression tree $\hat{f}^b(x)$
  - Average across bootstrap samples to get the predictor

$$\hat{f}_{bag} = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x) \tag{5}$$

  - Basically we are smoothing predictions.
  - Idea: the variance of the average is less than that of a single prediction.

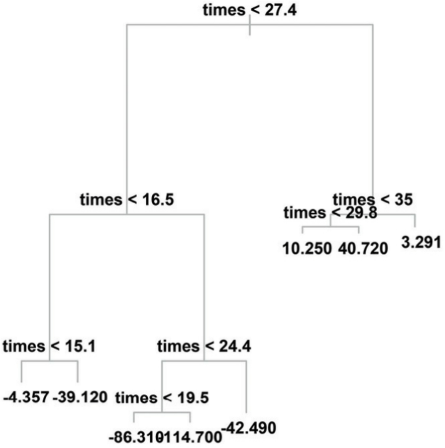
Bagging

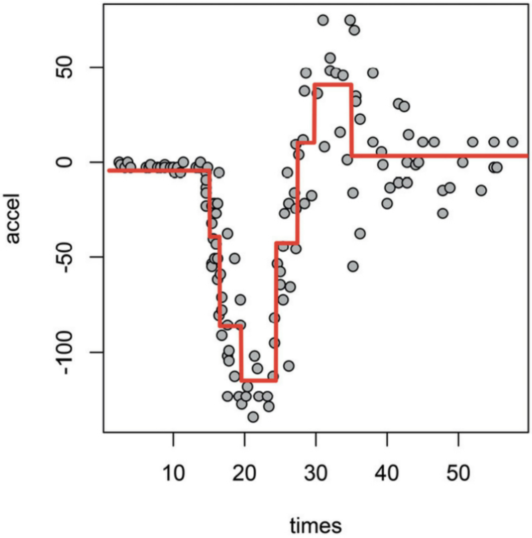


Random Forests

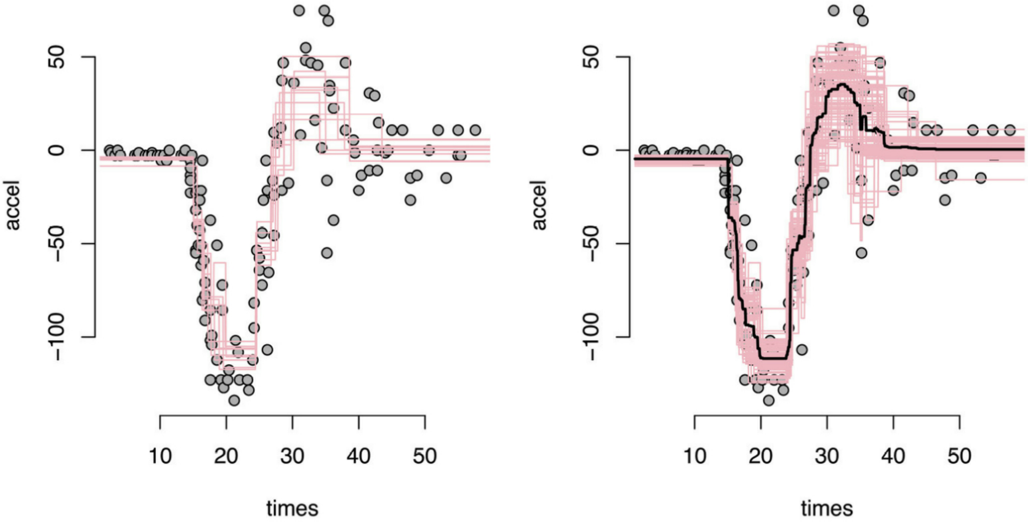
- Problem with bagging: if there is a strong predictor, different trees are very similar to each other. High correlation. Is the variance really reduced?
- Forests: lower the correlation between the trees in the boostrap.
- If there are p predictors, in each partition use only $m < p$ predictors, chosen randomly
- Bagging is forest with $m = p$ (use all predictors in each partition).
- Typically $m = \sqrt(p)$

Random Forests

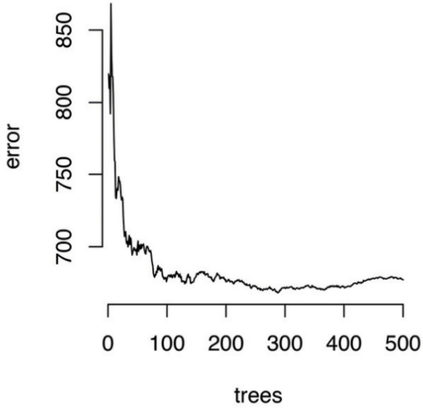




Random Forests



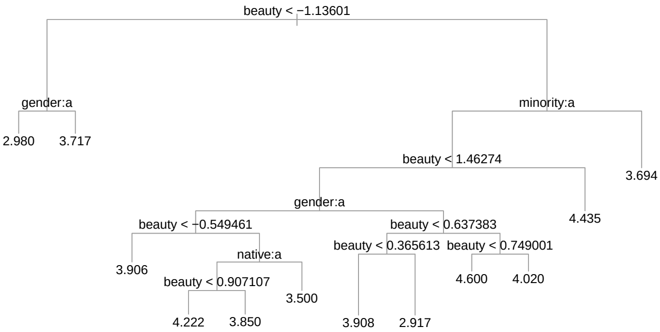
Random Forests



Random Forests and Trees

Trees

```r
pstcut <- prune.tree(pstree, best=12)
plot(pstcut, col=8)
text(pstcut)
```



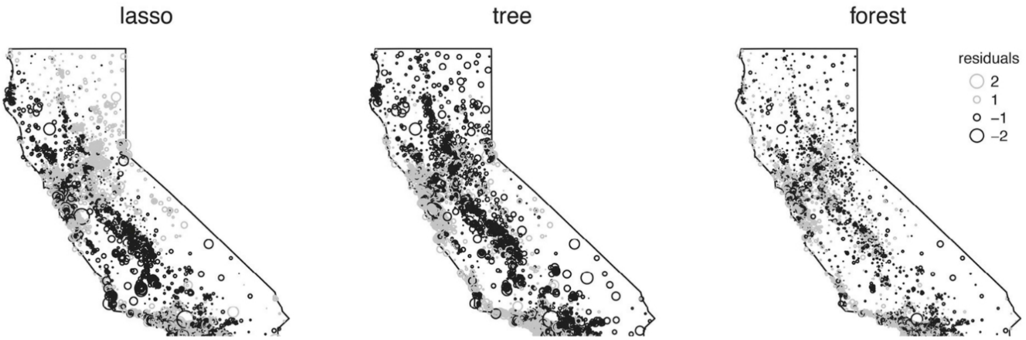
Random Forests and Trees

```
require("ranger")

rf_tree <- ranger(eval ~beauty + gender + minority + native +
         tenure + division, data=tr,
        write.forest=TRUE,num.tree=200,min.node.size=25,
        importance="impurity")
sort(rf_tree$variable.importance, decreasing = TRUE)
```

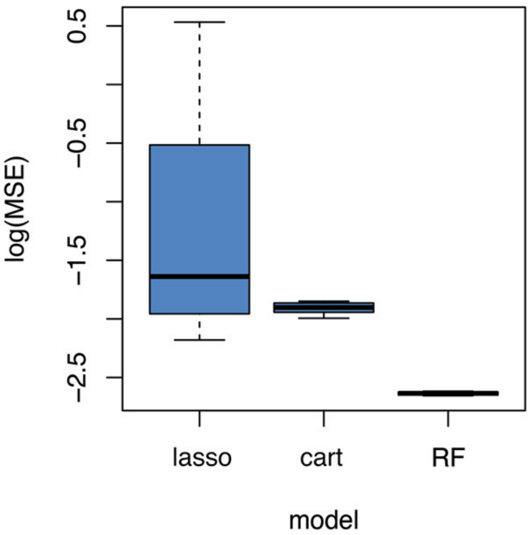
```
##    beauty  minority    gender    native    tenure  division
## 22.881176  3.089366  2.608295  2.104095  2.062075  1.627261
```

In sample residuals




```r
# Model Matrix for Lasso
XXca <- model.matrix(logMedVal~.*longitude*latitude,
      data=data.frame(scale(CAhousing)))[,-1]
```

Out of sample MSE



Boosting

- Problem with CART: high variance. Instability
- Weak classifier: marginally better classifier than flipping a coin (error rate slightly better than .5)
- E.g.: CART with few branches ('stump', two branches)
- Boosting: weighted average of a succession of weak classifiers.
- Vocab
  - $y \in -1, 1$ (for simplicity), X vector of predictors.
  - $y = G(X)$ (classifier)
  - $err = \frac{1}{N} \sum_i^N I(y_i \neq G(x_i))$

AdaBoost

1. Start with weights $w_i = 1/N$
2. For m = 1 through M:
   1. Adjust $G_m(x)$ using weights w_i .
   2. Compute prediction

$$err_m = \frac{\sum_{i=1}^{N} I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i} \tag{6}$$

   3. Compute $\alpha_m = ln\left[\frac{(1-err_m)}{err_m}\right]$
   4. Update weights: $w_i \leftarrow w_i c_i$

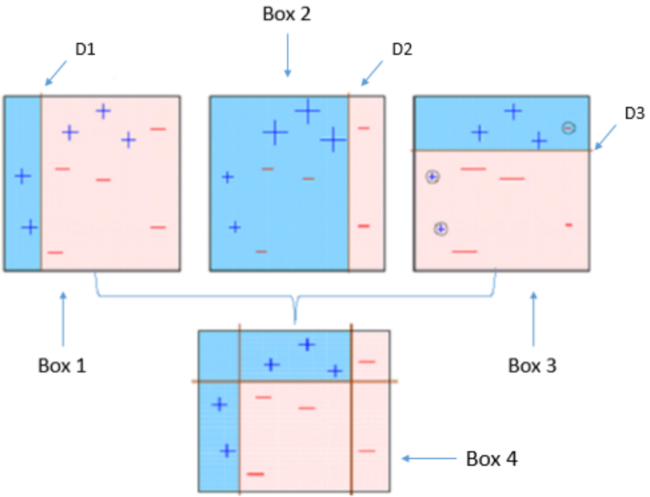
$$c_i = exp\left[\alpha_m I(yi \neq G_m(x_i))\right] \tag{7}$$

3. Output: $G(x) = sgn[\sum_{m=1}^{M} \alpha_m G_m(x)]$

AdaBoost

- $c_i = exp\left[\alpha_m I(y_i \neq G_m(x_i))\right]$
- If it was correctly predicted, $c_i = 1$. No issue.
- Otherwise, $c_i = exp(\alpha_m) = \frac{(1-err_m)}{err_m} > 1$
- At each step the method gives more relative importance to the predictions that where wrong.
- Final step: weighted average of predictions at each step.

AdaBoost



Source: https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/

Causal Forests

American Economic Review: Papers & Proceedings 2017, 107(5): 546–550
https://doi.org/10.1257/aer.p20171000

LABOR MARKETS AND CRIME[‡]

Using Causal Forests to Predict Treatment Heterogeneity: An Application to Summer Jobs[†]

By JONATHAN M.V. DAVIS AND SARA B. HELLER*

Idle hands are the devil's workshop

- The application uses two large scale RCTs of Chicago's One Summer Plus (OSP) program conducted in 2012 and 2013. OSP provides disadvantaged youth ages 14 to 22 with 25 hours a week of employment, an adult mentor, and some other programming.
- Participants are paid Chicago's minimum wage ($8.25 at the time).
- Find a 43 percent reduction in violent crime arrests in the 16 months after random assignment.

Causal Tree: Theory Details

- Work well in RCTs
- Issue: we do not observe the ground truth
- Honest estimation (Innovation):
  - One sample to choose partition
  - One sample to estimate leaf effects
- Why is the split critical?
- Fitting both on the training sample risks overfitting: Estimating many "heterogeneous effects" that are really just noise idiosyncratic to the sample.
- We want to search for true heterogeneity, not noise

Heterogeneous Treatment Effects Assumptions

- There are a couple of assumptions that are key
- Assumption 1: Unconfoundedness

$$Y_i(1), Y_i(0) \perp W_i \mid X_i \tag{8}$$

  - The *unconfoundedness* assumption states that, once we condition on observable characteristics, the treatment assignment is independent to how each person would respond to the treatment.
  - i.e., the rule that determines whether or not a person is treated is determined completely by their observable characteristics.
  - This allows, for example, for experiments where people from different genders get treated with different probabilities,
  - **rules out** experiments where people self-select into treatment due to some characteristic that is not observed in our data.

Heterogeneous Treatment Effects

- Assumption 2: Overlap

$$\forall\, x \in \text{supp}\,(X), \qquad 0 < P\,(W = 1 \mid X = x) < 1 \tag{9}$$

  - The *overlap* assumption states that at every point of the covariate space we can always find treated and control individuals.
  - i.e., in order to estimate the treatment effect for a person with particular characteristics $X_i = x$, we need to ensure that we are able to observe treated and untreated people with those same characteristics so that we can compare their outcomes.

The Honest Target: Athey and Imbens Innovation

- The ultimate goal is to construct and assess an algorithm $\pi(.)$ that maximizes the honest criterion

$$max\, Q^H(\pi) = -E_{S^{te},S^{est},S^{tr}}\left[MSE_\mu(S^{te}, S^{est}, S^{tr}, \pi(S^{tr}))\right] \tag{10}$$

- In CART the target is different (adaptive target)

$$max\, Q^C(\pi) = -E_{S^{te},S^{tr}}\left[MSE_\mu(S^{te}, S^{tr}, \pi(S^{tr}))\right] \tag{11}$$

From Causal Trees to Causal Forest

The implementation steps are as follows in Davis and Heller (2017):

- (1) Draw a subsample b without replacement containing $n_b = 0.2N$ observations from the N observations in the dataset
- (2) Randomly split the n_b observations in half to form a training sample (tr) and an estimation sample (e) such that $n_{tr} = n_e = \frac{n_b}{2}$. Using just the training sample, start with a single leaf containing all n_{tr} observations.

From Causal Trees to Causal Forest

The implementation steps are as follows in Davis and Heller (2017):

- (3) For each value of each covariate, $X_j = x$, form candidate splits of the observations into two groups based on whether $X_j \leq x$. Consider only splits where there are at least ten treatment and ten control observations in both new leaves.
- Choose the single split that maximizes an objective function O capturing how much the treatment effect estimates vary across the two resulting subgroups, with a penalty for within leaf variance . If this split increases O relative to no split, implement it and repeat this step in both new leaves. If no split increases O, this is a terminal leaf.

$$O = (n_T + n_C)\hat{\tau}_l^2 - 2\left(\frac{\hat{Var}(Y_{Tl})}{n_T} + \frac{\hat{Var}(Y_{Cl})}{n_C}\right) \tag{12}$$

From Causal Trees to Causal Forest

- (4) Once no more splits can be made in step 3, the tree is defined for subsample b. Move to the estimation sample, and group the ne observations into the same tree based on their Xs.
- (5) Using just the estimation sample, calculate $\hat{\tau} = \bar{y}_{Tl} - \bar{y}_{Tc}$ within each terminal leaf. This step makes the tree honest, since treatment effect estimates are made using different observations than the ones that determined the splits.
- (6) Return to the full sample of N observations. Assign $\hat{\tau}_{l,b} = \hat{\tau}_l$ to each observation whose Xs would place it in leaf l, and save this prediction.
- (7) Repeat steps (i) to (vi) B = 25, 000 times

From Causal Trees to Causal Forest

- Define observation i's predicted CATE as $\hat{\tau}_i^{CF}(x) = \frac{1}{B}\sum \hat{\tau}_{l,b}$

From Causal Trees to Causal Forest

- Define observation i's predicted CATE as $\hat{\tau}_i^{CF}(x) = \frac{1}{B}\sum \hat{\tau}_{l,b}$
- The procedure requires the researcher to select three parameters: the number of trees, the minimum number of treatment and control observations in each leaf, and the subsample size.
- In the absence of formal criteria to guide our choices, we used a large number of trees (more trees reduce the Monte Carlo error introduced by subsampling; we found moving from 10,000 to 25,000 improved the stability of estimates across samples).
  - Increasing the minimum number of observations in each leaf trades off bias and variance; bigger leaves make results more consistent across different samples but predict less heterogeneity.
  - Smaller subsamples reduce dependence across trees but increase the variance of each estimate (larger subsamples made little difference in our application).

From Causal Trees to Causal Forest

- We run the entire CF procedure using only S_{in}, then use the trees grown in S_{in} to generate predictions for all observations in S_{in} and S_{out}.
- This allows to assess the performance of the predictions in a hold-out sample (albeit with reduced statistical power) and to check whether heterogeneity is more distinct in S_{in} than S_{out}, which could be a sign of overfitting.
- Within each sample, we group youth by whether they are predicted to have a positive or negative treatment effect ($\hat{\tau}_i^{CF} > 0$ is desirable for employment and adverse for arrests).
- We estimate separate treatment effects for these two subgroups by regressing each outcome on the indicator:

$$y_{ib} = \beta_1 I[\hat{\tau}_i^{CF} > 0] + \beta_2 T_i I[\hat{\tau}_i^{CF} > 0] + \beta_3 T_i \left(1 - I[\hat{\tau}_i^{CF} > 0]\right) + X\theta + \alpha_b + u_{ib} \tag{13}$$

Causal Forests

TABLE 1—TREATMENT EFFECTS BY PREDICTED RESPONSE

Subgroup	No. of violent crime arrests	Any formal employment
Panel A. In sample		
$\hat{\tau}_i^{CF}(x) > 0$	0.22	0.19
	(0.05)	(0.03)
$\hat{\tau}_i^{CF}(x) < 0$	−0.05	−0.14
	(0.02)	(0.03)
H_0: subgroups equal, $p =$	0.00	0.00
Panel B. Out of sample		
$\hat{\tau}_i^{CF}(x) > 0$	−0.01	0.08
	(0.05)	(0.03)
$\hat{\tau}_i^{CF}(x) < 0$	−0.02	−0.01
	(0.02)	(0.03)
H_0: subgroups equal, $p =$	0.77	0.02
Panel C. Adjusted in sample		
$\hat{\tau}_i^{CF}(x) > 0$	−0.06	0.05
	(0.04)	(0.03)
$\hat{\tau}_i^{CF}(x) < 0$	−0.02	−0.04
	(0.02)	(0.03)
H_0: subgroups equal, $p =$	0.41	0.02

Review & Next Steps

- Bagging and Random Forests
- Comparisons: Lasso, CART, Random Forests
- AdaBoost
- Causal Forests
- Next class: More on boosting
- Questions? Questions about software?

Further Readings

- Athey, S., & Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. Proceedings of the National Academy of Sciences, 113(27), 7353-7360.
- Davis, Jonathan M.V., and Sara B. Heller. 2017. ”Using Causal Forests to Predict Treatment Heterogeneity: An Application to Summer Jobs.” American Economic Review, 107 (5): 546-50.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.
- Green, D. P., & Kern, H. L. (2012). Modeling heterogeneous treatment effects in survey experiments with Bayesian additive regression trees. Public opinion quarterly, 76(3), 491-511.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- Taddy, M. (2019). Business data science: Combining machine learning and economics to optimize, automate, and accelerate business decisions. McGraw Hill Professional.