

# Causal Inference and Machine Learning - Heterogeneous Treatment Effects

Alexander Quispe  
The World Bank, PUCP  
[alexander.quispe@pucp.edu.pe](mailto:alexander.quispe@pucp.edu.pe)

May 14, 2024

# Motivation - CI & ML

In the last six years, an explosion has appeared in a research collaboration between economists (inference) and computer scientists/stats (Prediction).  
What are the methods that combine CI & ML?

Susan Athey School (Stanford) - CATE	Victor Chernozhukov School (MIT) - HDM
Causal Trees Causal Forest Policy Learning Simulating data using WGANs Surrogate Index Contextual Bandits	High Dimensional Metrics (Lasso, Ridge, ElasticNet) Double Debiased Machine Learning (Lasso, Ridge, Neural Nets, Trees, Forest) DML with Sensitivity Analysis (Omitted Variable Bias)

## Motivation - CI & ML

Given that most of these methodologies were proposed in academic environments, most of the packages/libraries are written in R.

However, if you transition to Industry, most Tech Firms use Python (Also Julia!!!).

Therefore, companies such as Microsoft, Uber, and Facebook have created Python packages that include these algorithms.

1. EconML (Microsoft) - <https://econml.azurewebsites.net/>
2. CausalML (UBER) - <https://github.com/uber/causalml>
3. GeoLift (Facebook) - <https://github.com/facebookincubator/GeoLift/>
4. DoubleML - <https://docs.doubleml.org/stable/index.html>

PS. Sorry about Stata Lovers

## Motivation - Causal Trees

- Most of our work in the WB is related to finding ATE/CATE estimators, but we **Pre-Specify analysis plan**
- Given that most of the time we use Surveys, and field experiments, we do not face the problem of HDM.
- **Goal: Data-driven search for heterogeneity in causal effects with valid standard errors**
- Applications : A/B Testing, Observational studies

## A/B testing for an online shop, Dataset Taken from Matteo Courthoud (Zalando)

- **Y = spend:** The amount of money spent by the customer.
- **T = discount:** Indicating whether the customer was offered a discount (1) or not (0).
- **Time:** What time the user was on the website.
- **Device:** The type of device used by the customer (mobile or desktop).
- **Browser:** The browser used by the customer.
- **Age:** Age of User

# Treatment Effect and HTE

1.

$$\text{Spend} = \beta_0 + \beta_1 \times \text{Time} + \beta_2 \times \text{Device} \\ + \beta_3 \times \text{Age} + \beta_4 \times \text{Discount} + \epsilon$$

2.

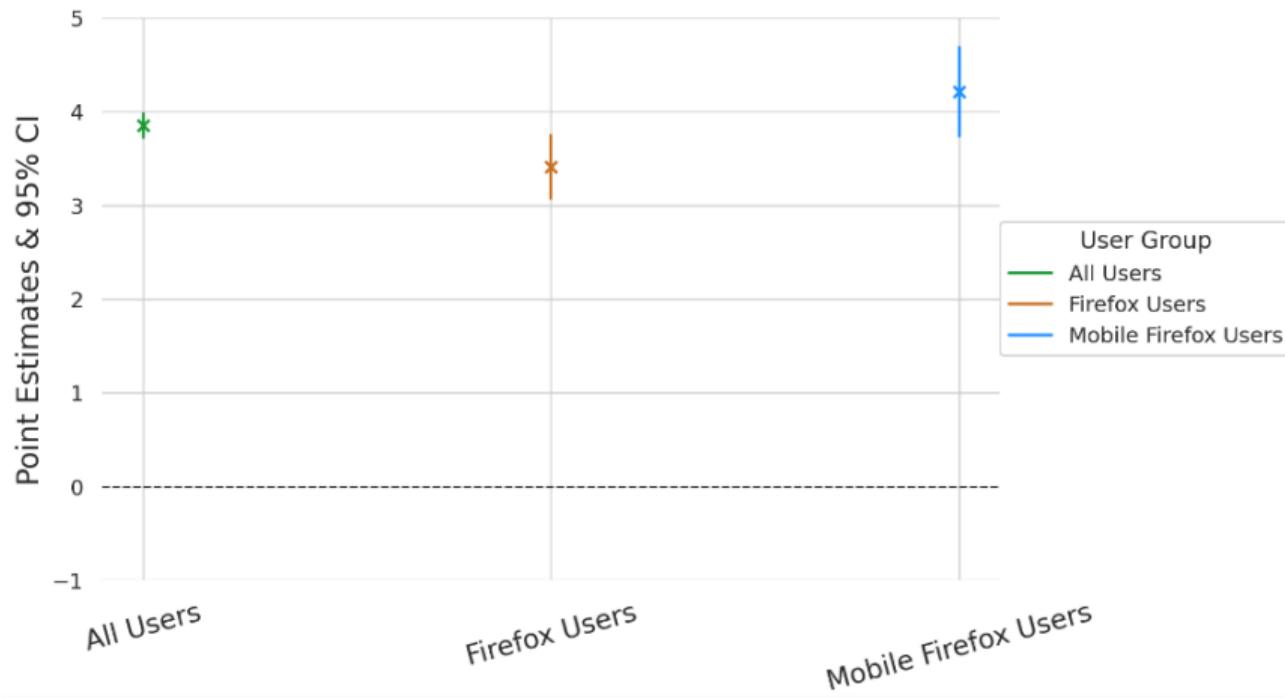
$$\text{Spend} = \beta_0 + \beta_1 \times \text{Time} + \beta_2 \times \text{Device} \\ + \beta_3 \times \text{Age} + \beta_4 \times \text{Discount} \\ + \beta_5 \times (\text{Discount} \times \text{Browser}) + \epsilon$$

3.

$$\text{Spend} = \beta_0 + \beta_1 \times \text{Time} + \beta_2 \times \text{Device} \\ + \beta_3 \times \text{Age} + \beta_4 \times \text{Discount} \\ + \beta_5 \times (\text{Discount} \times \text{Browser} \times \text{Device}) + \epsilon$$

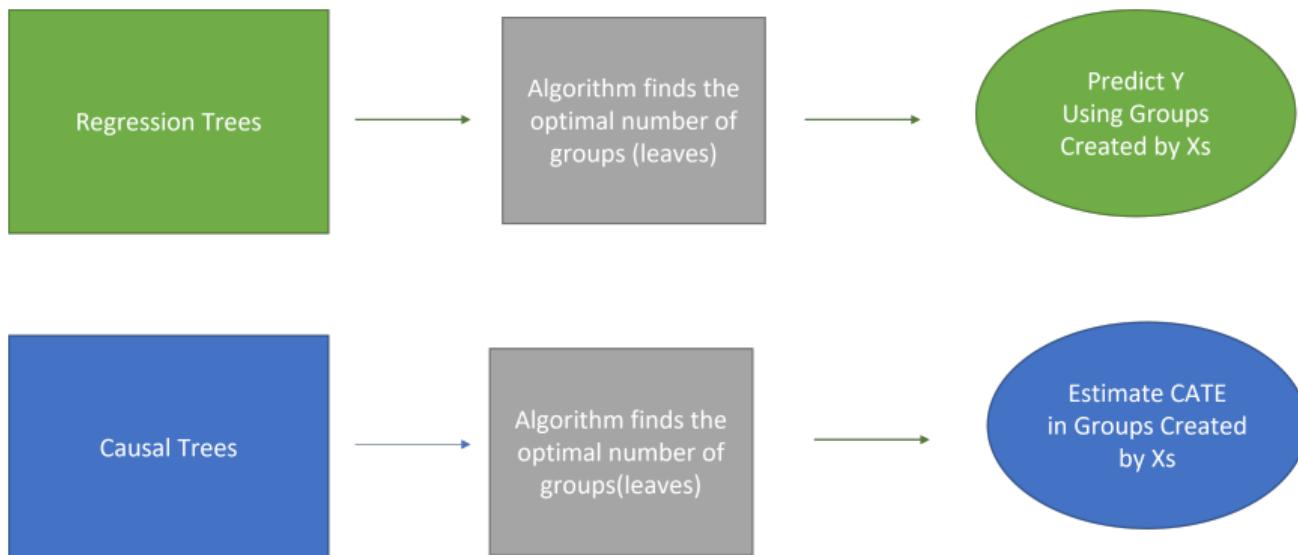
# Treatment Effect and HTE

Figure: ATE & HTE



# Main Idea Behind Causal Trees

Figure: Trees vs Causal Trees



## Conditional Average Treatment Effect

- We will learn how to estimate the conditional average treatment effect (CATE)

$$\tau(x) = E[Y_i(1) - Y_i(0) | X_i = x] \quad (1)$$

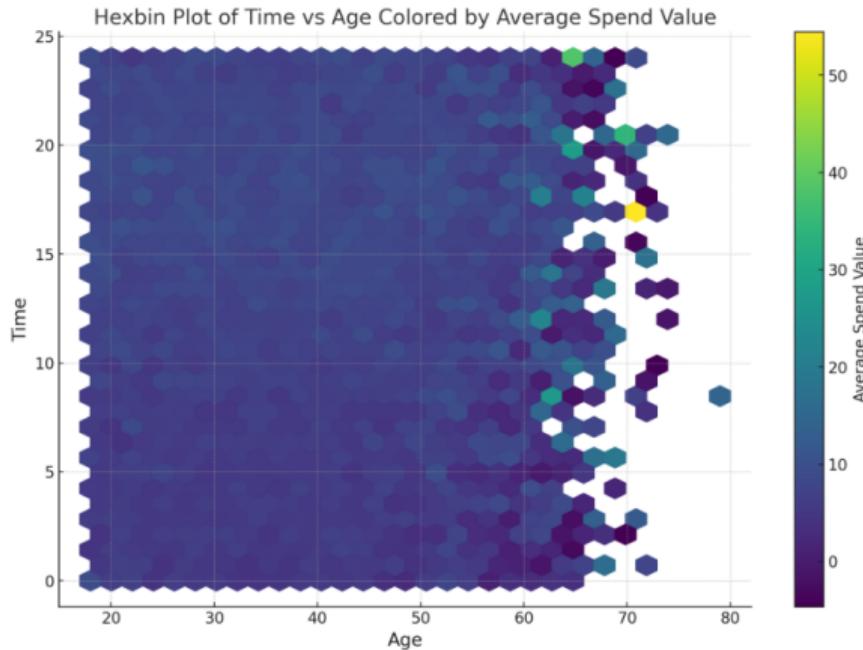
This is the average treatment effect conditional on a vector of observable characteristics.

- However, When the observable covariates are high-dimensional, It can be hard to estimate without making strong modeling. Therefore, we will try to calculate treatment effect averages for simpler groups.

$$\tau(x) = E[Y_i(1) - Y_i(0) | G_i = g] \quad (2)$$

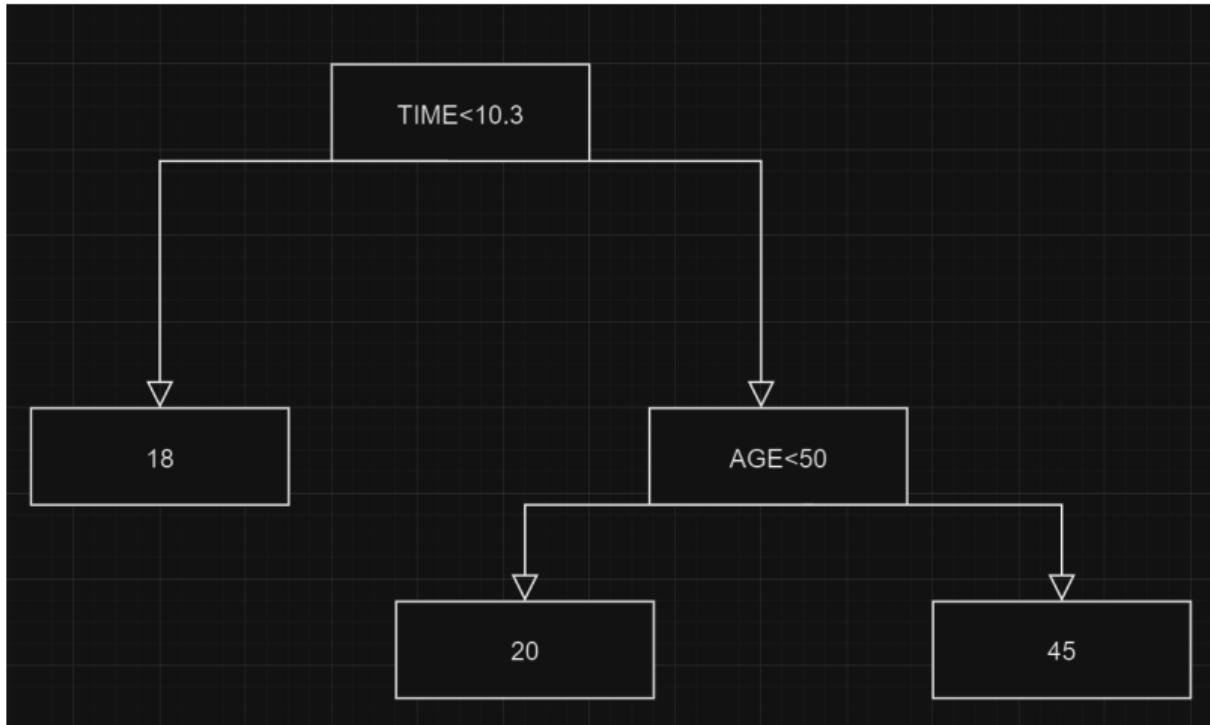
# Decision Trees: online shop data: how would you stratify it?

Figure: Spend-Time-Age



## Decision tree for these data

Figure: Decision tree for online shop



## Details of previous figure

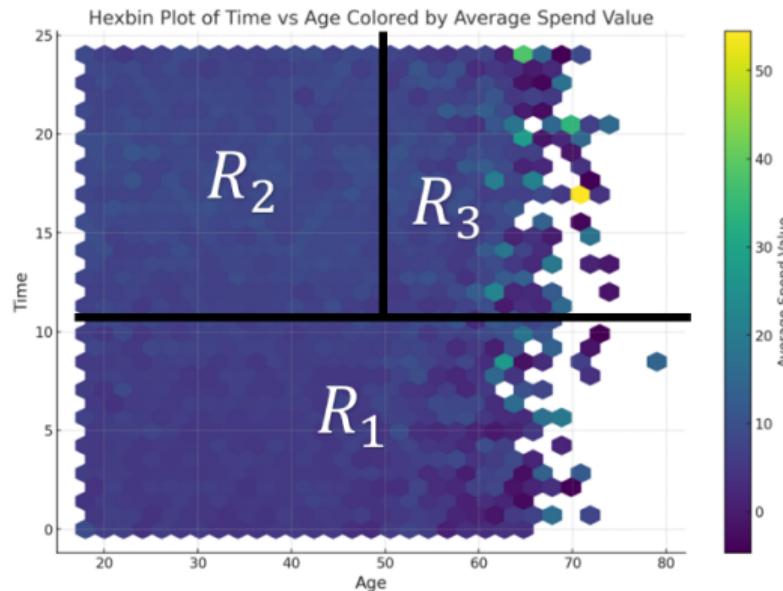
At a given node:

- $X_j < t_k$  indicates the left-hand branch emanating from that split
- $X_j \geq t_k$  is the right-hand branch
- The left-hand branch corresponds to  $Time < 10.3$
- The right-hand branch corresponds to  $Time \geq 10.3$
- The tree has **two internal nodes** and **three terminal nodes**, or **leaves**. The number in each leaf is **the mean of the response** for the observations that fall there.

## Results

- $R_1 = \{X | Time < 10.3\}$
- $R_2 = \{X | Time \geq 10.3, Age < 50\}$
- $R_3 = \{X | Time \geq 10.3, Age > 50\}$

Figure: Segments



## Terminology for Trees

- In keeping with the **tree** analogy, the areas  $R_1$ ,  $R_2$ , and  $R_3$  are known as **terminal nodes**.
- Decision trees are typically drawn **upside down**, in the sense that the leaves are at the bottom of the tree.
- The points along the tree where the predictor space is split are referred to as **internal nodes**.
- In the hitters tree, the two internal nodes are indicated by the text *Time < 10.3* and *Age < 50*.

## Tree-building process

- We divide the predictor space – that is, the set of possible values for  $X_1, X_2, \dots, X_p$  into  $J$  distinct and non-overlapping areas  $R_1, R_2, \dots, R_J$
- For every observation that falls into the area  $R_j$ , we make the same prediction, which is simply the mean of the response values for the training observations in  $R_j$ .
- The goal is to find boxes  $R_1, \dots, R_J$  that minimize the RSS, given by

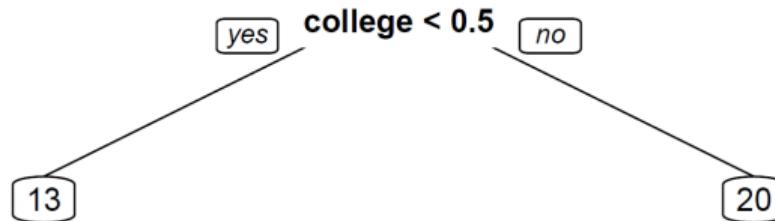
$$\sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \quad (3)$$

where  $\hat{y}_{R_j}$  is the mean response for the training observations within the  $j$ th box.

## Top-down, greedy approach

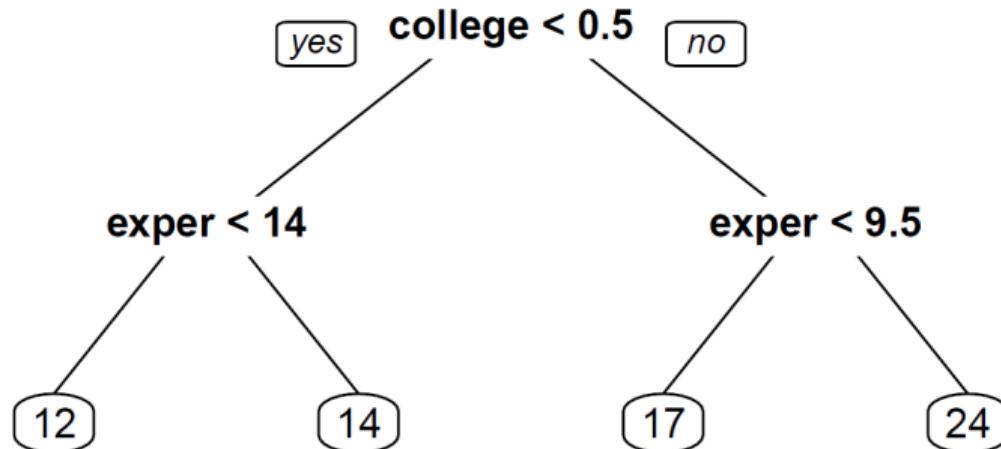
- **Top-down** because it begins at the top of the tree and then successively splits the predictor space
- **Greedy** because at each step of the tree-building process, the best split is made at that particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step.
- Example using wage data:

Figure: Wage example - Tree 1.(CMLB Ch3 Page 7)



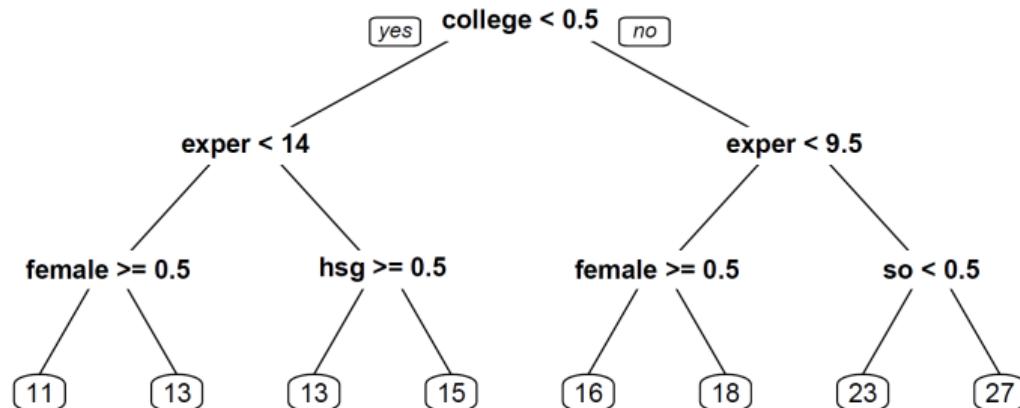
## Top-down, greedy approach

Figure: Wage example - Tree 2 (CMLB Ch3 Page 7)



## Top-down, greedy approach

Figure: Wage example - Tree 3 (CMLB Ch3 Page8)



## Pruning Regression Trees

- First, the deeper we grow the tree, the better is our approximation to the regression function  $g(Z)$
- The deeper the tree, the noisier our estimate  $\hat{g}(Z)$  becomes, since there are fewer observations per terminal node to estimate the predicted value for this node.
- From a prediction point of view, we can try to find the right depth or the structure of the tree by cross-validation. For example, in the wage example the tree of depth 2 performs better in terms of cross-validated *MSE* than the trees of depth 3 or 1. The process of cutting down the branches of the tree to improve predictive performance is called “**Pruning the Tree**”.

## Cost complexity pruning

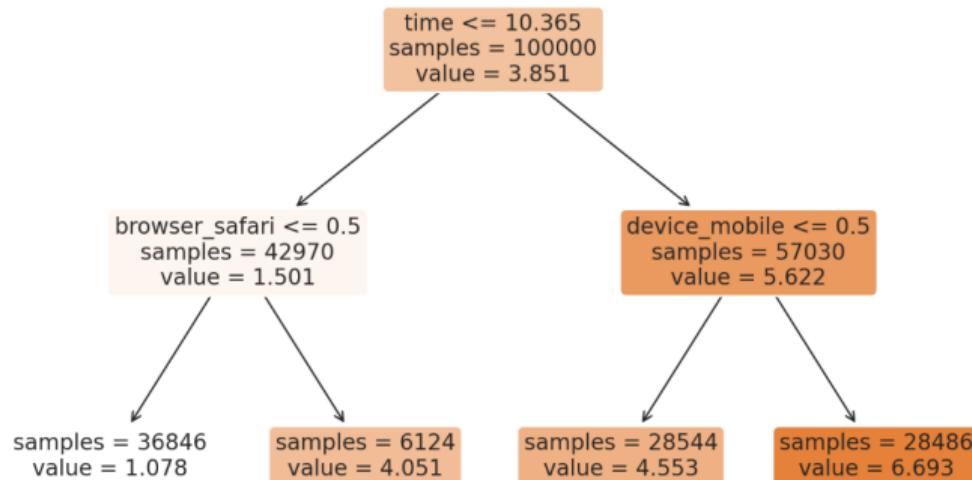
we consider a sequence of trees indexed by a nonnegative tuning parameter  $\alpha$ . For each value of  $\alpha$  there corresponds a subtree  $T \subset T_0$  such that

$$\sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T| \quad T \subset T_0 \quad (4)$$

1.  $T$  indicates the number of terminal nodes of the tree  $T$
2.  $R_m$  is the rectangle, corresponding to the  $m$ th terminal node
3.  $\hat{y}_{R_m}$  is the mean of the training observations in  $R_m$

# Online Shop Data - Tree

Figure: Online Shop Data - Tree using Cost complexity pruning



# Causal trees and Tree Regression

## Similarities

- Divide data into subgroups in order to maximize the discriminatory power of the split
- Both uses recursive binary splitting (greedy approach) into covariate space
- Solve over-fitting problem through Cross- Validation (CV) to determine the depth of the tree

# Causal trees and Tree Regression

## Differences

- **Goal: estimate treatment effect heterogeneity**
- Optimize sub-groups to estimate treatment effect heterogeneity
- Divide population into subgroups to minimize MSE in treatment effects (**objective function**) instead in outcomes
- Use honesty tree, additionally, to address over-fitting.

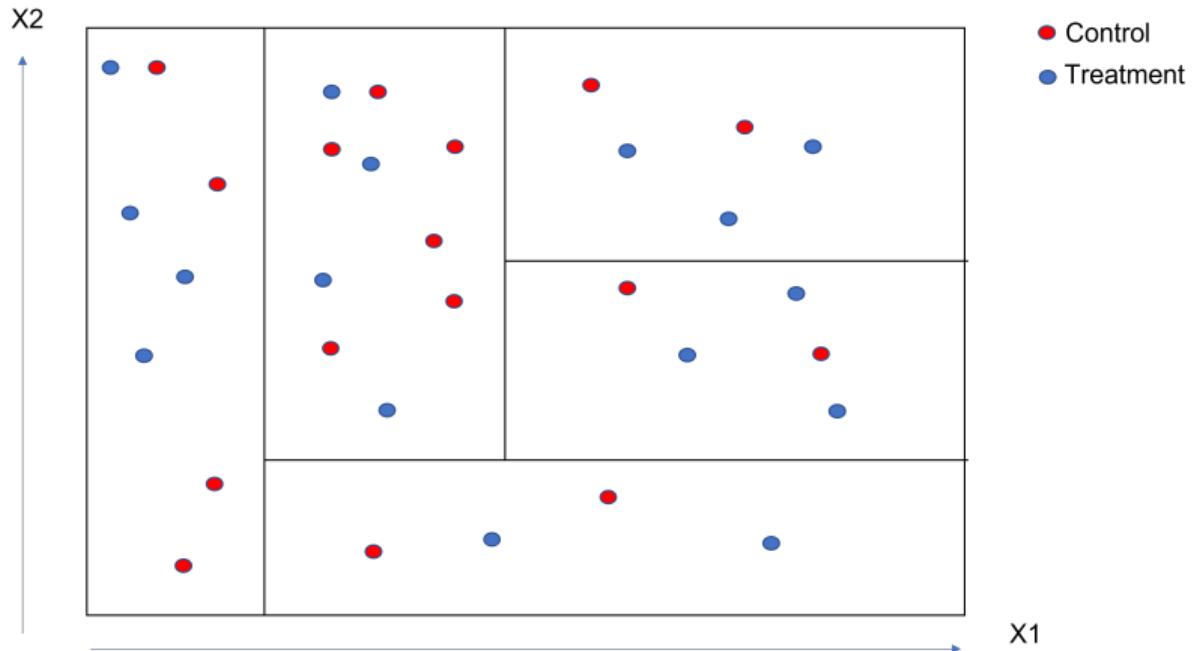
## How to address missing counterfactual

- Objective function is unfeasible  $\sum_i[(\tau_i - \hat{\tau}(X_i))^2]$  (true treatment effect unobserved)
- Transform Outcome (Athey and Imbens, 2016)
- Applying off-the-shelf prediction to estimate CATE
- $p$  is the assignment probability (if perfect RCT then  $p = 0.5$ ).

$$Y_i^* = Y_i/p, \text{ if } W_i = 1, \text{ and let } Y_i^* = -Y_i/(1-p) \text{ if } W_i = 0 \quad (5)$$

- $\mathbb{E}[Y_i^*|X_i = x] = \tau(x)$  is noisy for individual treatment effect, but unbiased estimate of CATE
- Use estimated propensity score  $\hat{p}$  or AIPW score as outcome in observational studies.

# Causal Tree Intuition



## Honest Causal Tree

- Half sample to estimates tree and another half to estimate treatment effect.
- **Tradeoff**
- COST: sample splitting means build shallower tree, less personalized predictions and lower MSE of treatment effects
- BENEFIT: valid confidence intervals with coverage rates that do not deteriorate as data generating process get more complex or more covariates are added.
- **Inference**
- Can separate tree construction from treatment effect estimation
- tree constructed on training is independent of test sample
- Holding tree form training sample fixed, can use standard methods to conduct inference within leaf of the tree on test sample

## Partition and Leaf Effects Estimates

- Three samples: model tree construction  $S^{tr}$ , estimation sample for leaf effects  $S^{est}$  and a test sample  $S^{te}$ .
- Sample average treatment effect in sample  $S^{est}$  for the leaf  $I(X_i, \Pi)$  associated with covariates  $X_i$

$$\hat{\tau}(X_i, S^{est}, \Pi) = \frac{1}{\sum_{j \in S^{est} \cap I(X_i, \Pi)} W_j} \sum_{i \in S^{est} \cap I(X_i, \Pi)} W_i Y_i - \frac{1}{\sum_{j \in S^{est} \cap I(X_i, \Pi)} (1 - W_j)} \sum_{i \in S^{est} \cap I(X_i, \Pi)} (1 - W_i) Y_i$$

## Estimating the MSE of treatment effects (EMSE)

- criterion for evaluating a partition  $\Pi$  anticipating re-estimating leaf effects using sample splitting:

$$MSE(S^{est}, S^{te}) = \frac{1}{n^{te}} \sum_{i \in S^{te}} (\tau_i - \hat{\tau}(X_i, S^{est}, \Pi))^2 \quad (6)$$

$$MSE(S^{est}, S^{te}) = \frac{1}{n^{te}} \sum_{i \in S^{te}} (\tau_i^2 - 2\tau_i * \hat{\tau}(X_i, S^{est}, \Pi) + \hat{\tau}^2(X_i, S^{est}, \Pi)) \quad (7)$$

$$EMSE = E_{S^{est}, S^{te}} [MSE(S^{est}, S^{te})] \quad (8)$$

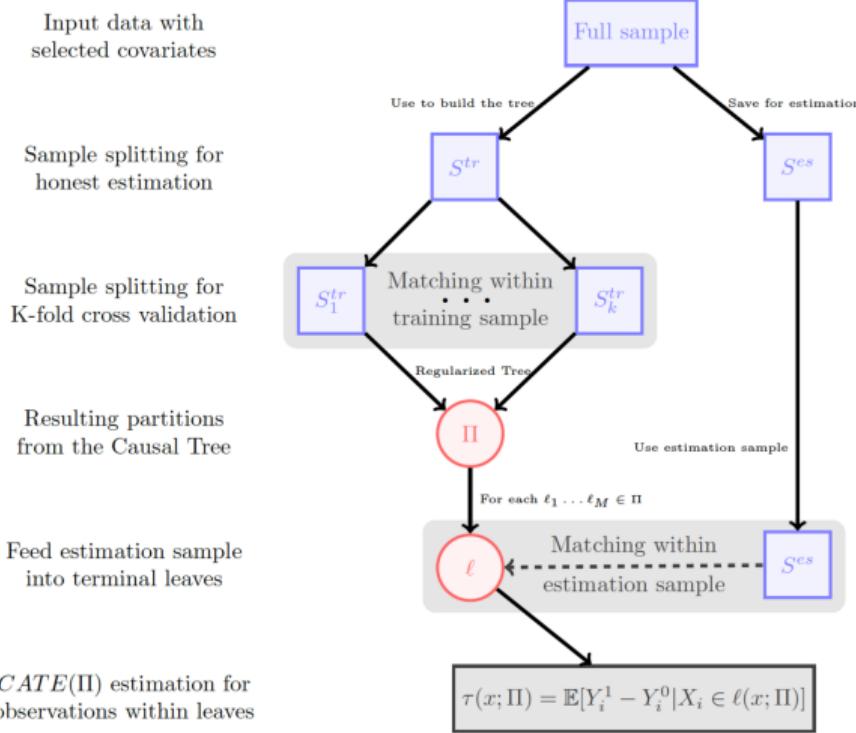
$$EMSE = \mathbf{V}_{S^{est}, S^{te}} [\hat{\tau}(X_i; \Pi, S^{est})] - E_{X_i} [\tau^2(X_i; \Pi)] + E[\tau_i^2] \quad (9)$$

## Causal Tree Algorithm

- Divide data into tree-building  $S^{tr}$  and estimation  $S^{est}$  samples
- Use recursively partition in covariate space  $X$  to deep partition  $\Pi$ 
  - Each split is selected as the one that minimizes mean - square error estimation of treatment effect over all possible binary splits
  - **Preserve minimum number of treated and control units in each child leaf**
- Use cross - validation to select the depth  $d^*$  of the partition
- Select partition  $\Pi^*$  by pruning  $\Pi$  to depth  $d^*$
- Estimate the treatment effects in each leaf of  $\Pi^*$  using estimation sample

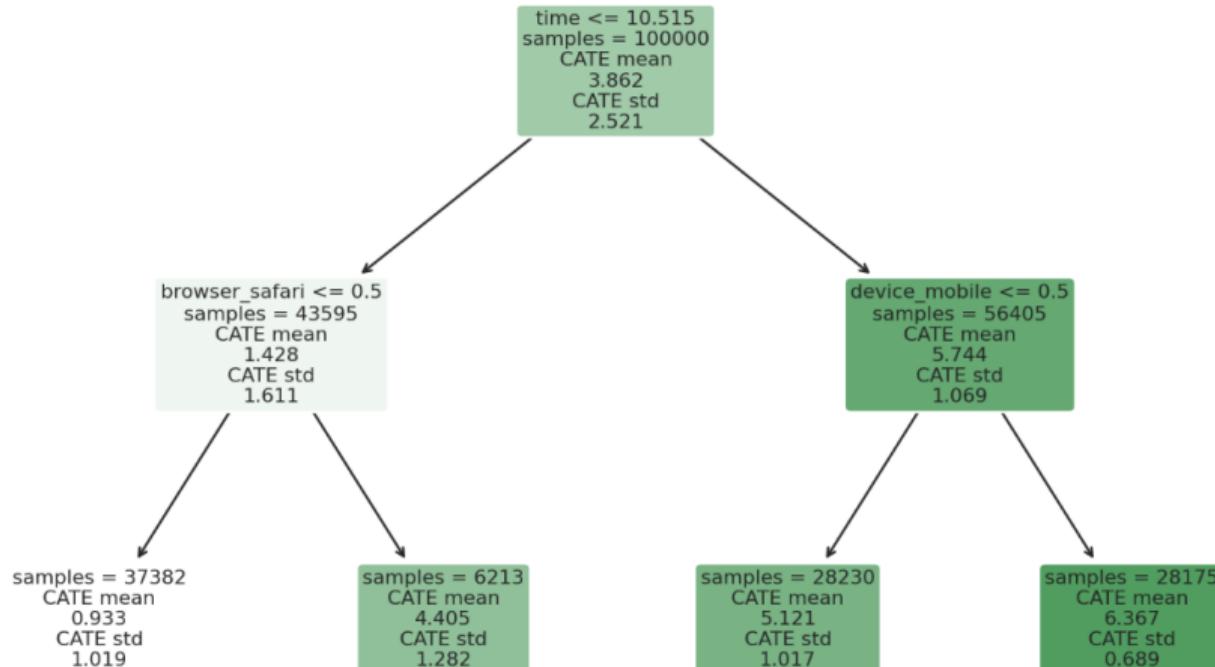
# Causal Tree Algorithm

FIGURE 1. CAUSAL TREE ALGORITHM WORKFLOW



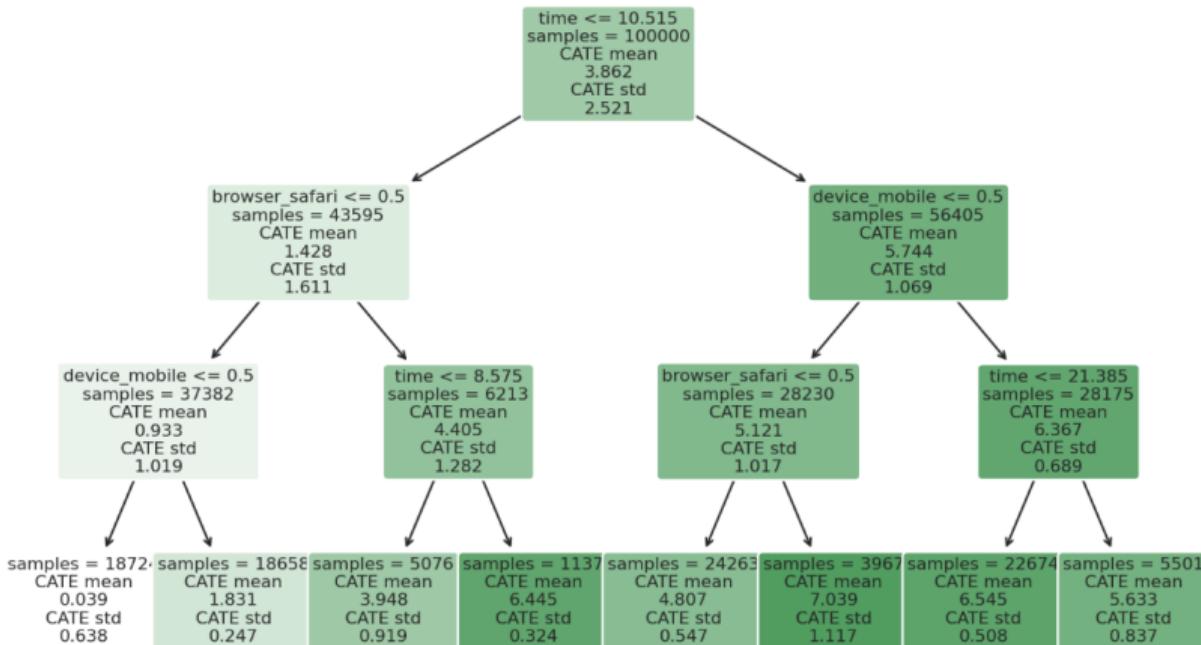
# Causal Tree - Online Shop Data

Figure: Causal Tree with max\_depth=2



# Causal Tree - Online Shop Data

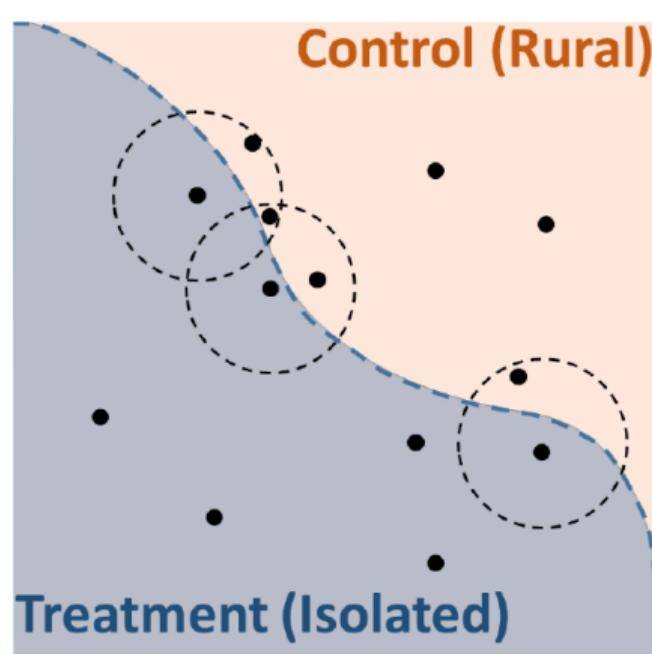
Figure: Causal Tree with max\_depth=3



## Causal Tree - Example 2

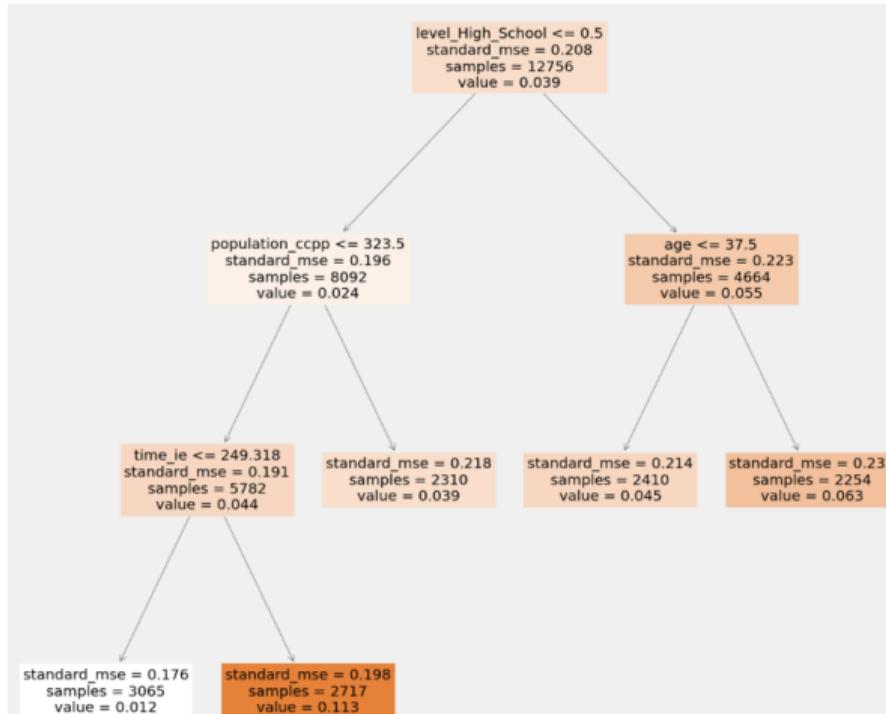
Estimating Heterogeneous Effects : How cash bonuses retain teachers in remote locations in Peru.

Figure: Unobserved Boundaries (Perez-Leon 2018)



## Causal Tree - Example 2

Estimating Heterogeneous Effects : How cash bonuses retain teachers in remote locations in Peru.



# Gracias

## Motivation - CI & ML

What if we want to start learning CI & ML from Zero to Hero? You can visit my open source project [Dive into Causal Machine Learning](#)

### **MGTECON 634: Machine Learning and Causal Inference - Standford**

1. [R Version](#)
2. [Python Version](#)

### **14.388: Inference on Causal and Structural Parameters Using ML and AI - MIT**

1. [R Version](#)
2. [Python Version](#)
3. [Julia Version](#)

PS. Again, Sorry about Stata Lovers

## Citation

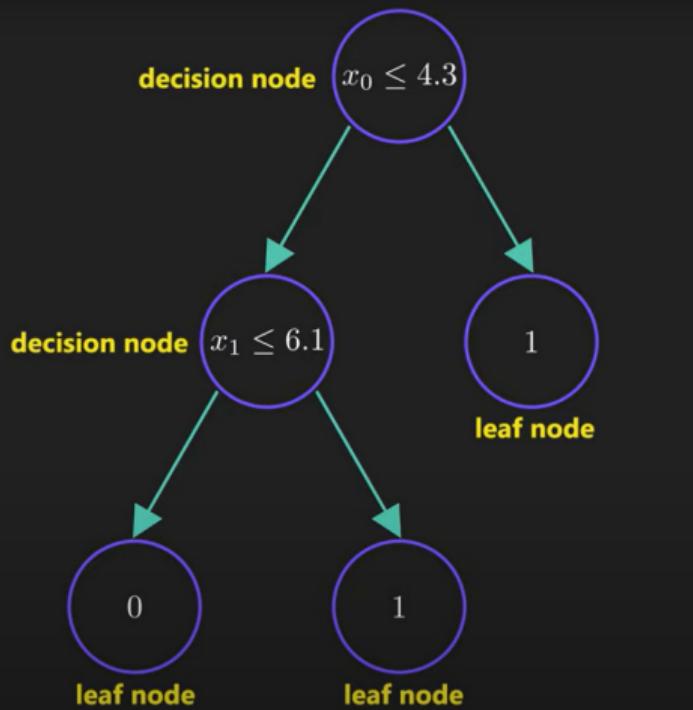
Images are taken from [Sujan Dutta](#) notes on Random Forest. Notes on Causal Trees and Causal Forest are taken from Susan Athey Lecture Notes on Machine Learning and Causal Inference, 2021.

# Random Forest

$id$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

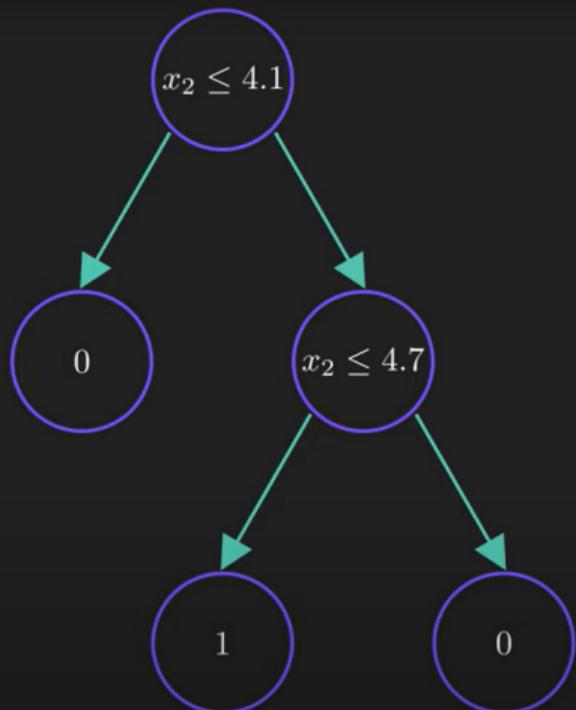
# Random Forest

<i>id</i>	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1



# Random Forest

$id$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	6.5	4.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1



# Random Forest

$id$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

$id$
2
0
2
4
5
5

# Random Forest

$id$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

$id$
2
0
2
4
5

$id$
2
1
3
1
4

$id$
4
1
3
0
0
2

$id$
3
3
2
5
1
2

# Random Forest

<i>id</i>	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

<i>id</i>
2
0
2
4
5

<i>id</i>
2
1
3
1
4

<i>id</i>
4
1
3
0
0
2

<i>id</i>
3
3
2
5
1
2

Bootstrapped Datasets

# Random Forest

$id$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

$id$
2
0
2
4
5

$id$
2
1
3
1
4

$id$
4
1
3
0
0
2

$id$
3
3
2
5
1
2

$x_0, x_1$

# Random Forest

$id$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

$id$
2
0
2
4
5
5

$id$
2
1
3
1
4
4

$id$
4
1
3
0
0
2

$id$
3
3
2
5
1
2

$x_0, x_1$

$x_2, x_3$

$x_2, x_4$

$x_1, x_3$

# Random Forest

$id$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

$id$
2
0
2
4
5
5

$id$
2
1
3
1
4
4

$id$
4
1
3
2
0
0
2

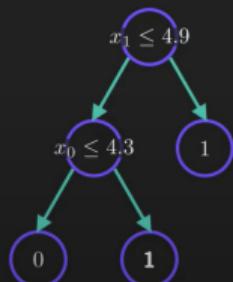
$id$
3
3
2
5
1
2

$x_0, x_1$

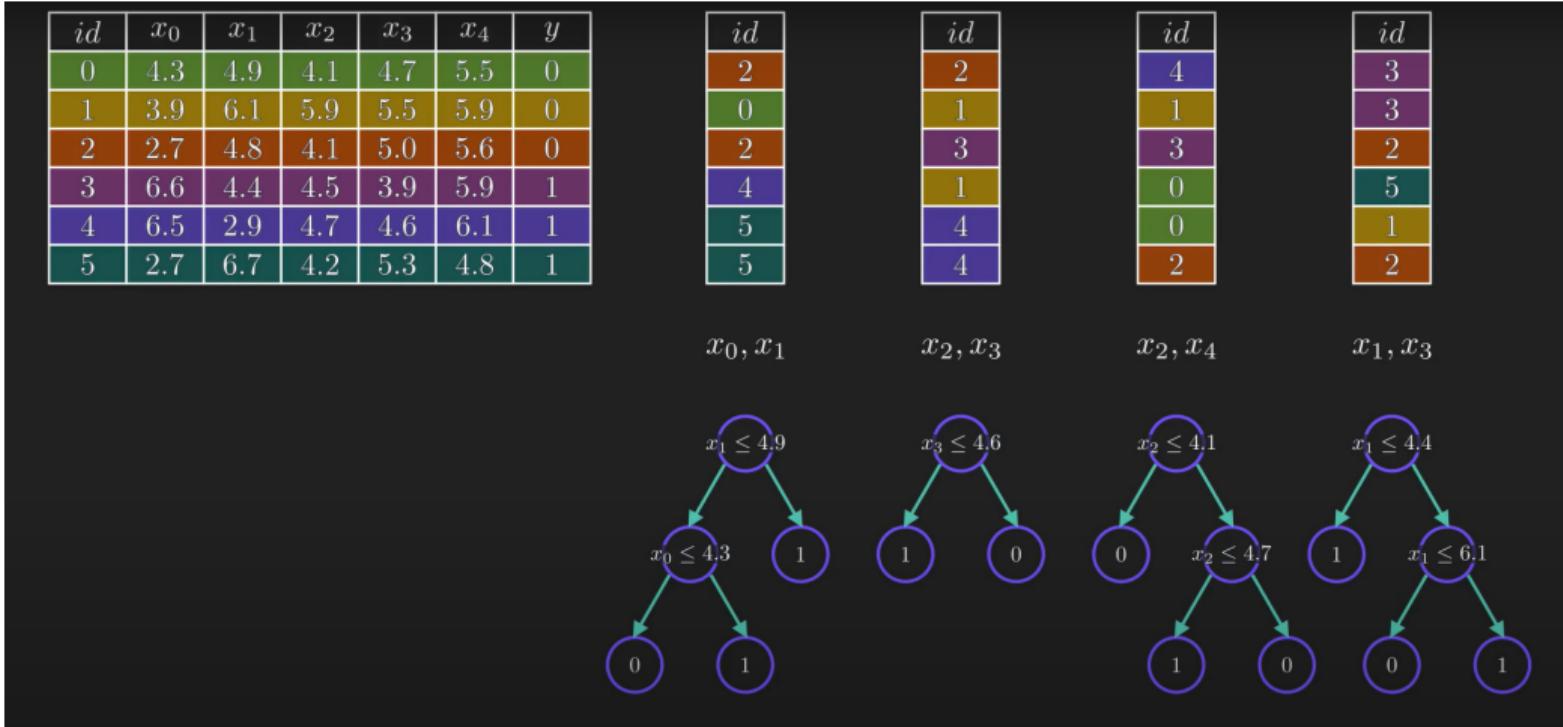
$x_2, x_3$

$x_2, x_4$

$x_1, x_3$



# Random Forest



# Random Forest

$id$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

$id$
2
0
2
4
5
5

$id$
2
1
3
1
0
0
4

$id$
4
1
3
3
0
0
2

$id$
3
3
2
5
1
2

$x_0, x_1$

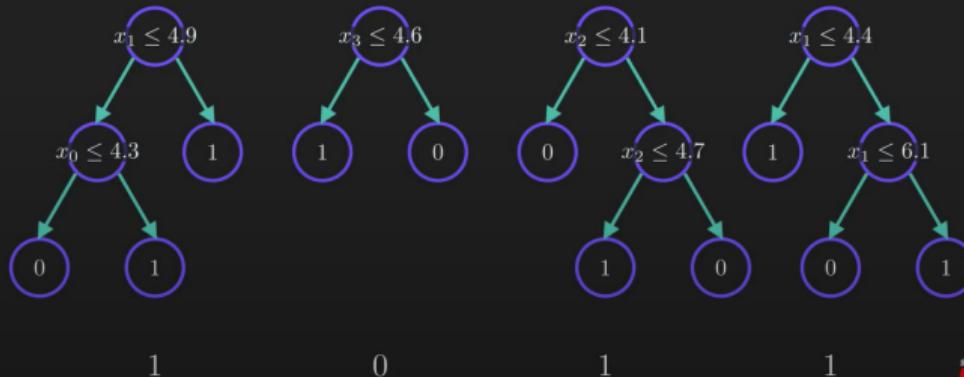
$x_2, x_3$

$x_2, x_4$

$x_1, x_3$

2.8	6.2	4.3	5.3	5.5
-----	-----	-----	-----	-----

Bootstrap + Aggregating  
(Bagging)



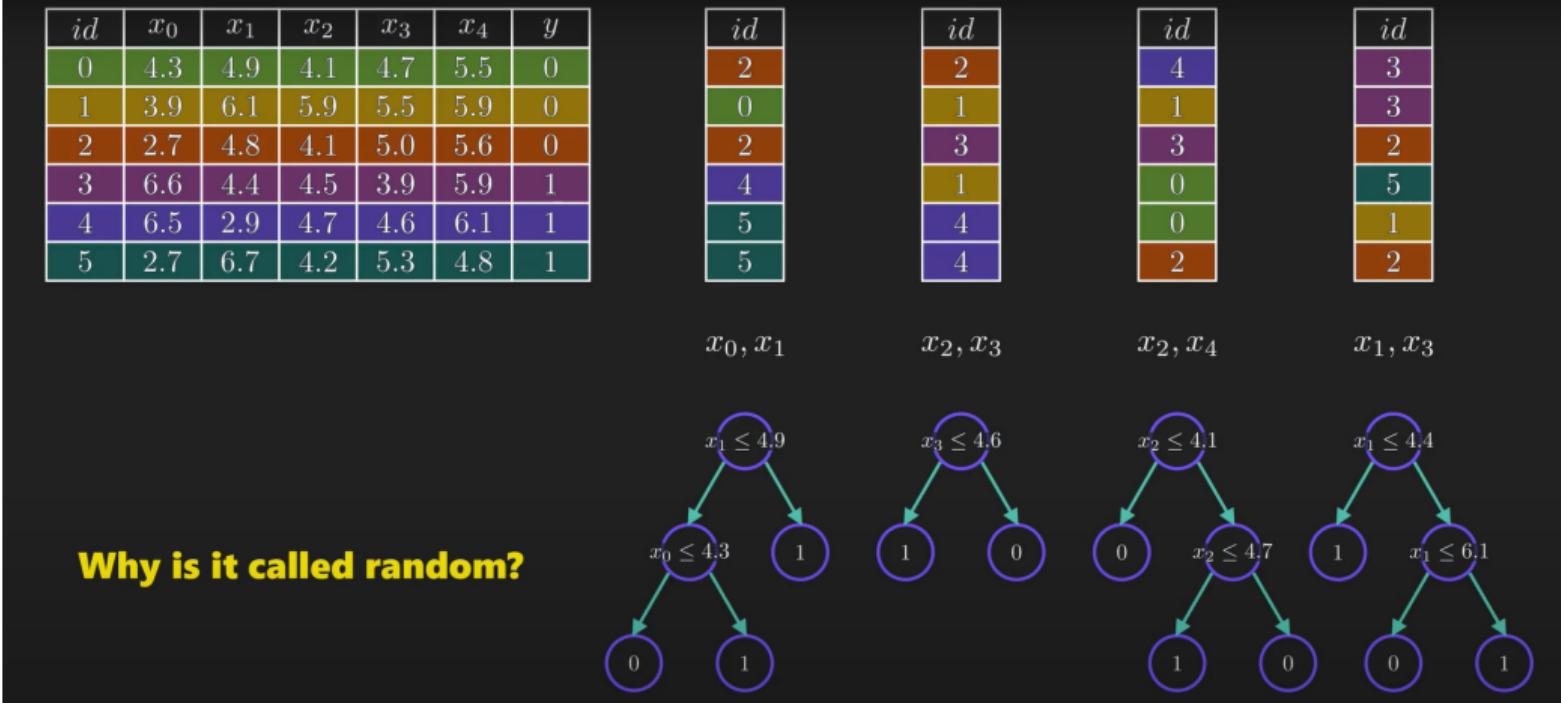
1

0

1

1

# Random Forest



# Random Forest

$id$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

$id$
2
0
2
4
5
5

$id$
2
1
3
1
0
4
4

$id$
4
1
3
3
0
0
2

$id$
3
3
2
5
1
2

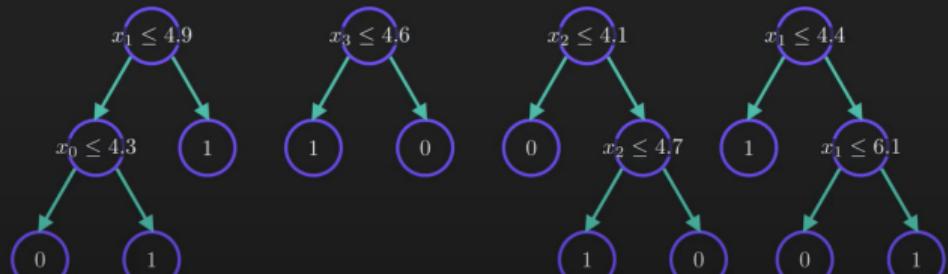
$x_0, x_1$

$x_2, x_3$

$x_2, x_4$

$x_1, x_3$

**Why Bootstrapping and Feature Selection?**



# Random Forest

$id$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

$id$
2
0
2
4
5
5

$id$
2
1
3
1
0
4
0
4

$id$
4
1
3
3
0
5
1
2

$id$
3
3
2
5
1
2

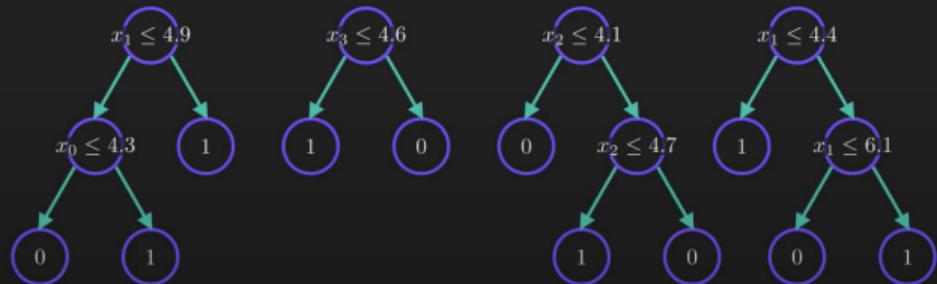
$x_0, x_1$

$x_2, x_3$

$x_2, x_4$

$x_1, x_3$

**How many features  
to consider?**



## Random Forest - Concluding Remarks

From ISL2 we have :

- Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees. This reduces the variance when we average the trees.
- But when building these decision trees, each time a split in a tree is considered, **a random selection of  $m$  predictors is chosen** as split candidates from the full set of  $p$  predictors. The split is allowed to use only one of those  $m$  predictors.
- A fresh selection of  $m$  predictors is taken at each split, and typically we choose  $m \approx \sqrt{p}$ , that is, the number of predictors considered at each split is approximately equal to the square root of the total number of predictors

## Baseline method: k-NN matching

Consider the  **$k$ -NN matching** estimator for  $\tau(x)$ :

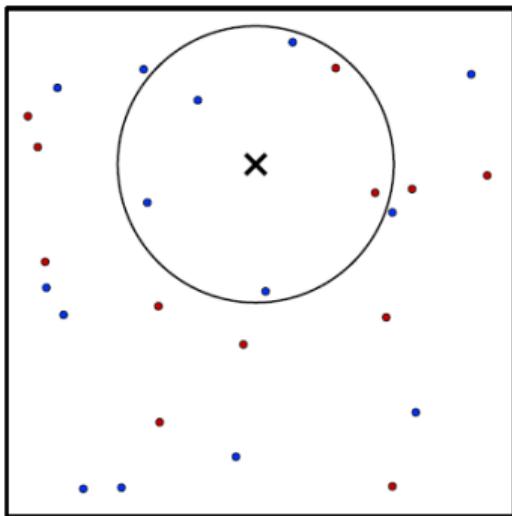
$$\hat{\tau}(x) = \frac{1}{k} \sum_{S_1(x)} Y_i - \frac{1}{k} \sum_{S_0(x)} Y_i,$$

where  $S_{0/1}(x)$  is the set of  $k$ -nearest cases/controls to  $x$ . This is consistent given **unconfoundedness** and regularity conditions.

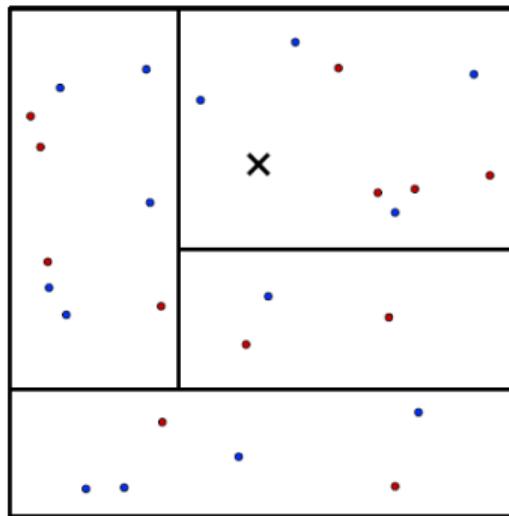
- ▶ **Pro:** Transparent asymptotics and good, robust performance when  $p$  is small.
- ▶ **Con:** Acute curse of dimensionality, even when  $p = 20$  and  $n = 20k$ .

**NB:** Kernels have similar qualitative issues as  $k$ -NN.

## Making k-NN matching adaptive



Euclidean neighborhood,  
for  $k$ -NN matching.



Tree-based neighborhood.

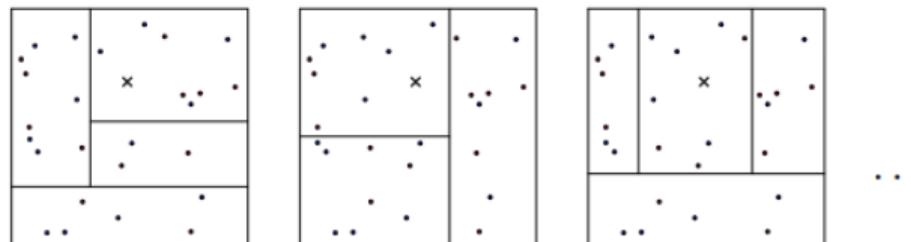
## From Trees to Random Forest

- Training set  $(X_i, Y_i, W_i)_{i=1}^n$ , while tree predictor for a test point  $(x)$

$$\hat{\tau} = T(x; X_i, Y_i, W_{i=1}^n) \quad (10)$$

- Random Forest: build and average many different trees  $T^*$
- Create alternative trees ( $T_b^*$ ) by bagging (sampling with replacement) or sub-sampling the training set

$$\hat{\tau} = \frac{1}{B} \sum_{b=1}^B T_b^*(x; X_i, Y_i, W_{i=1}^n) \quad (11)$$



## Statistical inference with regression forest

Regression forest are asymptotically Gaussian and centered:

$$\frac{\hat{\mu}_n(x) - \mu(x)}{\sigma_n(x)} \rightarrow \mathbb{N}(0, 1), \sigma_n^2(x) \rightarrow_p 0 \quad (12)$$

technical conditions

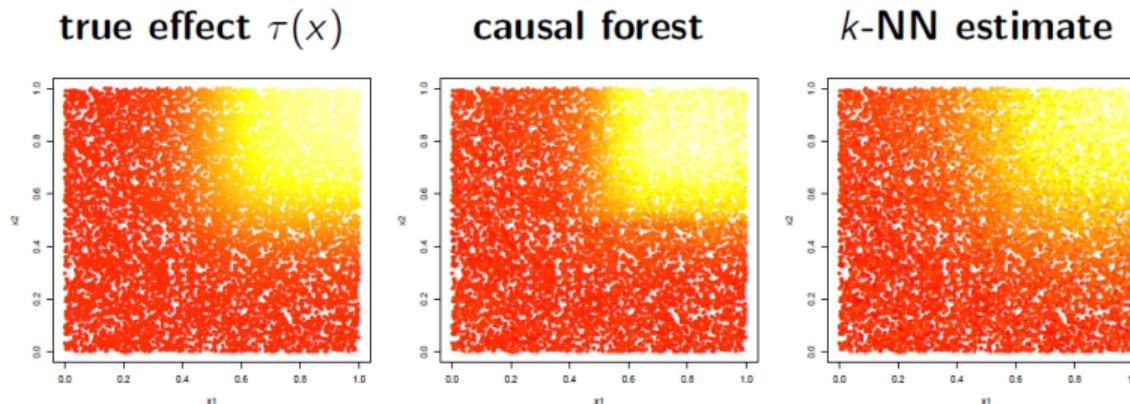
- Individual trees are honest (**Honesty**)
- Individual trees built random sub-samples of size  $s n^\beta$ , where  $\beta_{min} < \beta < 1$  (**Subsampling**)
- $X_i$  density from 0 and  $\infty$  (**Continuos features**)
- Conditional mean function  $\mu(x) = \mathbb{E}[Y|X = x]$  is Lipschitz continuous (**Lipschitz response**)

# Causal Forest Example

Figure: True effect, Causal Forest, KNN estimate

We have  $n = 20k$  observations whose features are distributed as  $X \sim U([-1, 1]^p)$  with  $p = 6$ ; treatment assignment is random. All the signal is concentrated along two features.

The plots below depict  $\hat{\tau}(x)$  for 10k random test examples, projected into the 2 signal dimensions.

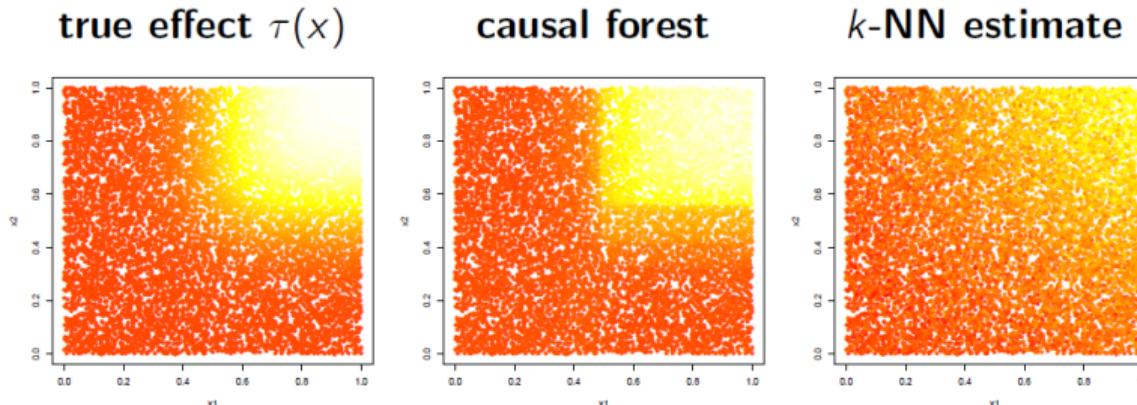


# Causal Forest Example

Figure: True effect, Causal Forest, KNN estimate

We have  $n = 20k$  observations whose features are distributed as  $X \sim U([-1, 1]^p)$  with  $p = 20$ ; treatment assignment is random.  
**All the signal is concentrated along two features.**

The plots below depict  $\hat{\tau}(x)$  for 10k random test examples, projected into the 2 signal dimensions.



## Sources

- Lecture Notes of Susan Athey's Machine Learning and Causal Inference Course at Stanford.
- Lecture Notes of Victor Chernozhukov's nference on Causal and Structural Parameters Using ML and AI at MIT .
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- Perez-Leon (2018). Inducing Teacher Retention in Remote Locations: Evidence from Peru.