Social Data Science

Soc 114
Winter 2026

Supervised Machine Learning: Trees and Forests

Learning goals for today

By the end of class, you will be able to

- understand the notion of supervised machine learning
  - an input-output machine
  - learned on some learning cases
  - used to predict for new cases
- apply that notion to the specific case of regression trees
- read a prediction from a regression tree
- understand how trees can aggregate to a forest

Prediction function and supervised learning

A **prediction function** is an input-output function:

- input a vector of predictors $\vec{x}$
- output a predicted outcome $\hat{y} = \hat{f}(\vec{x})$

	Input		**Output**
	$\vec{x}$	$\longrightarrow$	$\hat{y}$
Example:	{Sex,Age}		Probability of Employment Given Sex and Age

Supervised learning includes any approach that uses observed $\{\vec{x}, y\}$ data to learn a prediction function $\hat{f}$

	Age	Sex	Employed
cases for learning	26	F	1
	40	M	1
	61	M	0
	32	F	1
case to predict	63	F	?

OLS is a prediction function

Input $\vec{x} \rightarrow$ Output $\hat{y}$

$$\hat{y} = \hat{f}(\vec{x}) = \hat{\beta}_0 + \hat{\beta}_1(\text{Sex} = \text{Male}) + \hat{\beta}_2(\text{Age})$$

- Learn $\hat{f}$ in a **learning sample** with $\{\vec{x}_i, y_i\}_{i=1}^n$
  - Computer finds $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2$ that predict well in the learning sample
- At a new $\vec{x}$ value, predict $\hat{f}(\vec{x})$

Logistic regression is a prediction function

Input $\vec{x} \rightarrow$ Output $\hat{y}$

$$\hat{y} = \hat{f}(\vec{x}) = \text{logit}^{-1}\left(\hat{\beta}_0 + \hat{\beta}_1(\text{Sex} = \text{Male}) + \hat{\beta}_2(\text{Age})\right)$$

- Learn $\hat{f}$ in a **learning sample** with $\{\vec{x}_i, y_i\}_{i=1}^n$
  - Computer finds $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2$ that predict well in the learning sample
- At a new $\vec{x}$ value, predict $\hat{f}(\vec{x})$

There are many prediction functions

- input a vector of predictors $\vec{x}$
- output a predicted outcome $\hat{y} = \hat{f}(\vec{x})$

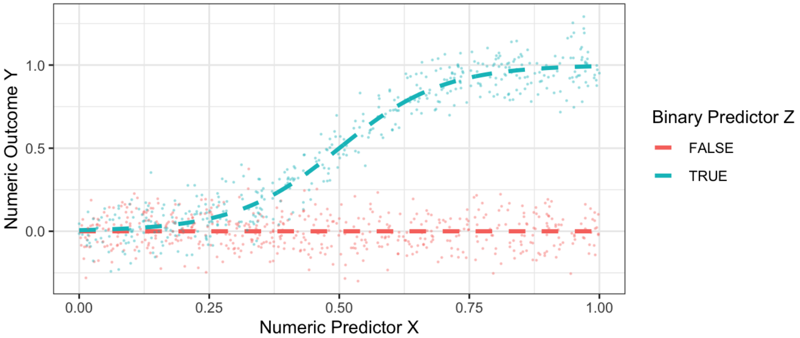
Trees as a prediction function

Tree: A series of TRUE or FALSE decisions leading to a prediction

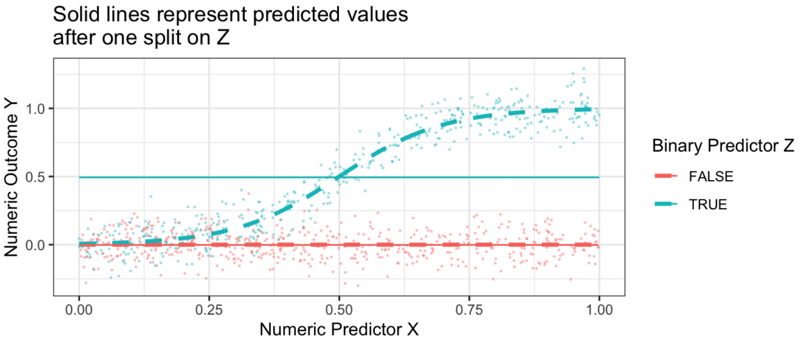
A made-up example:

- Is age greater than 40?
    - If so, is the respondent labeled female?
        - If so, predict 80% employed
        - If not, predict 85% employed
    - If age not greater than 40, predict 70% employed

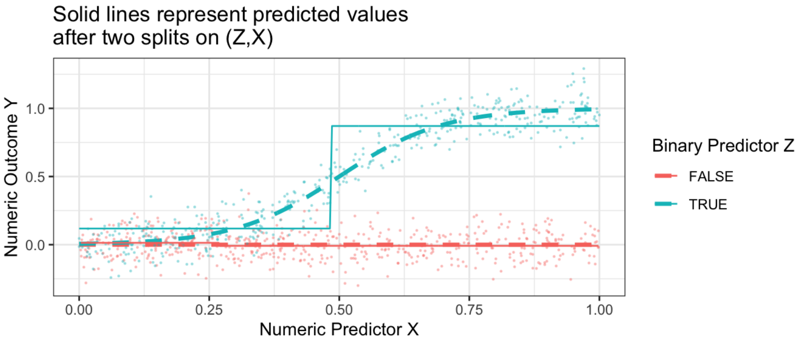
Trees as a prediction function



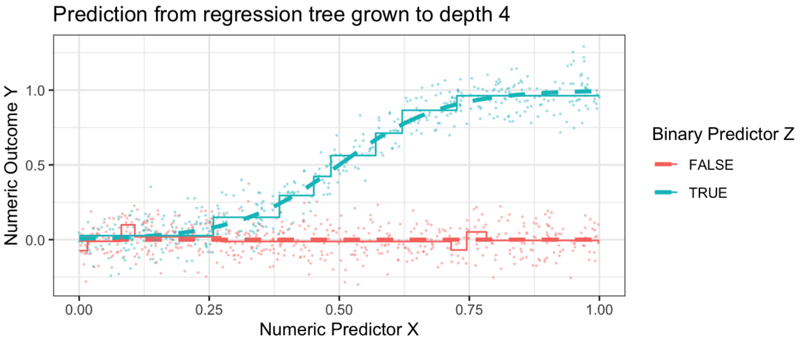
Trees as a prediction function



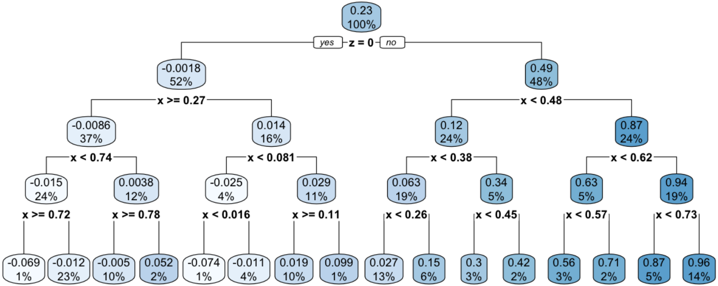
Trees as a prediction function



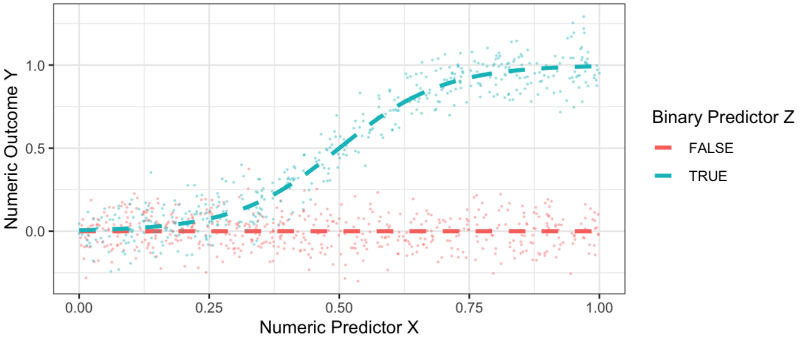
Trees as a prediction function



Trees as a prediction function

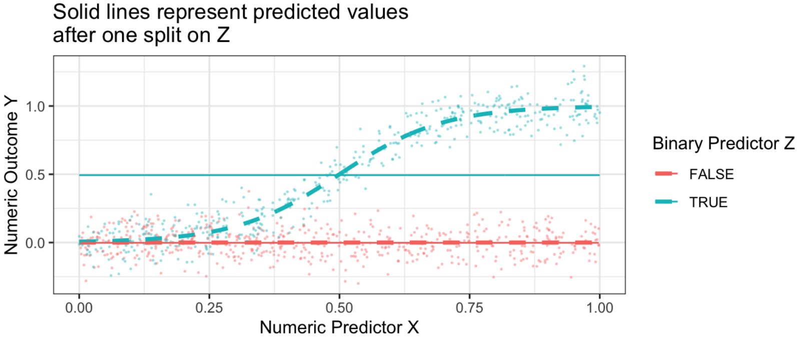


Trees as a prediction function: How that worked

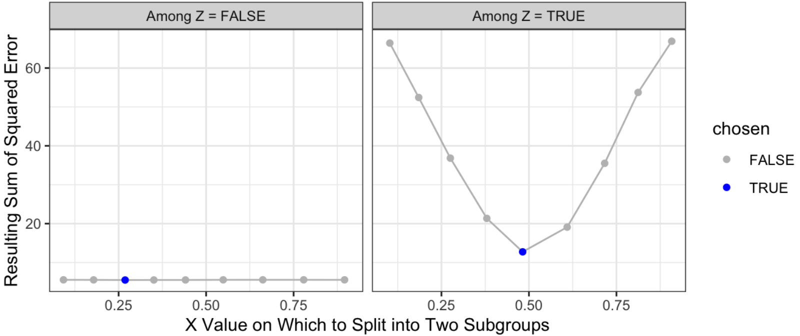


Trees as a prediction function: How that worked

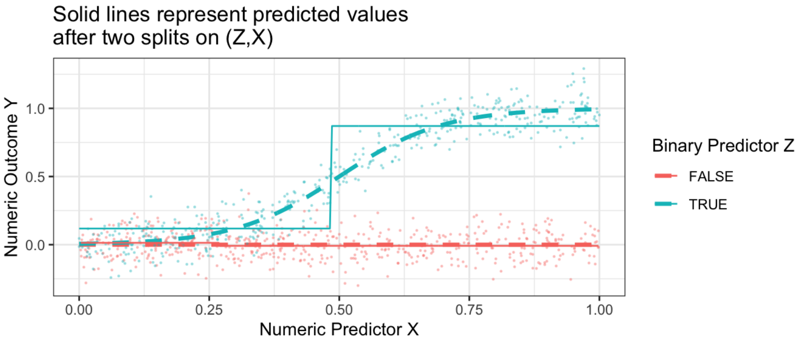
Trees as a prediction function: How that worked



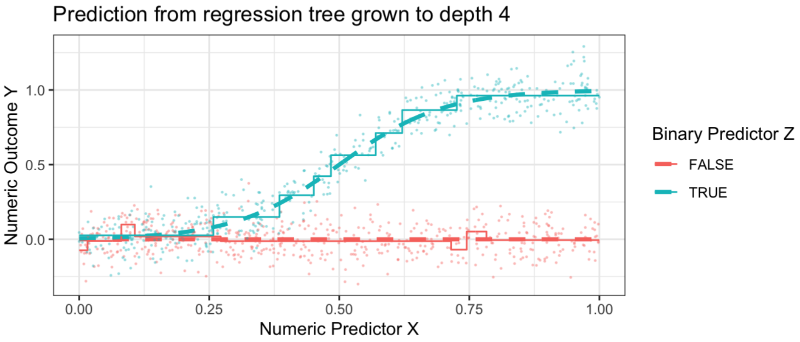
Trees as a prediction function: How that worked



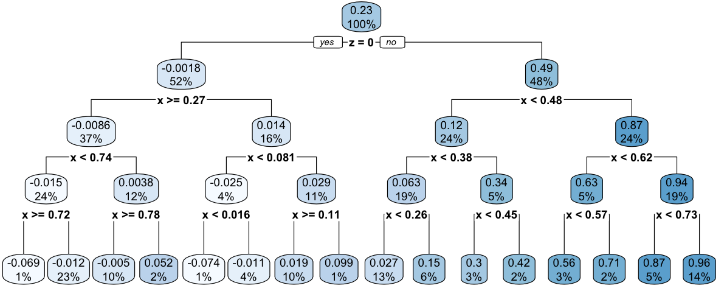
Trees as a prediction function: How that worked



Trees as a prediction function: How that worked



Trees as a prediction function: How that worked



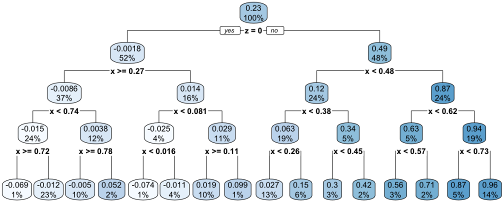
Trees as a prediction function: How that worked.
Summary.

1. Begin with all data
2. Consider many ways to partition into two parts
3. Estimate the mean squared prediction error for each:
   $\mathrm{E}((\hat{Y} - Y)^2)$
4. Choose the split that minimizes mean squared prediction error

Repeatedly, apply steps (1–4) to each subgroup.
Stop by a data-driven rule.

Trees: Some terminology

- Branch = one direction of a split
- Leaf = terminal node at the bottom



When presented with a new case, find its leaf.
Predict the mean of Y among learning cases in that leaf.

A tree can be interpretable: Realistic example

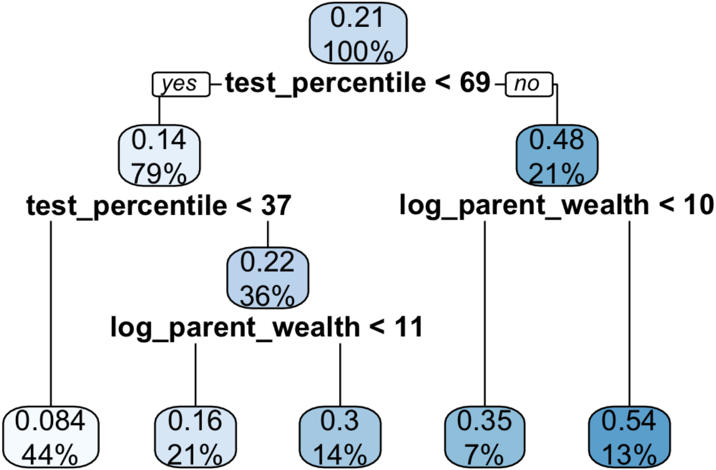
- Outcome: Has spouse or partner with BA degree at age 35
- Predictors: Demographics and measures of family background

A tree can be interpretable: Realistic example

```
library(tidyverse)
library(rpart)
library(rpart.plot)

all_cases <- read_csv("https://soc114.github.io/data/nlsy97_simulated.csv")

rpart.out <- rpart(
  y ~ sex + race + mom_educ + dad_educ + log_parent_income +
    log_parent_wealth + test_percentile,
  data = all_cases
)

rpart.plot(rpart.out)
```

A tree can be interpretable: Realistic example

Y = has spouse or partner with BA degree at age 35



Pruning a tree

Sometimes you want a simpler decision rule

- you worry you are fitting to noise
- you want to explain predictions more easily

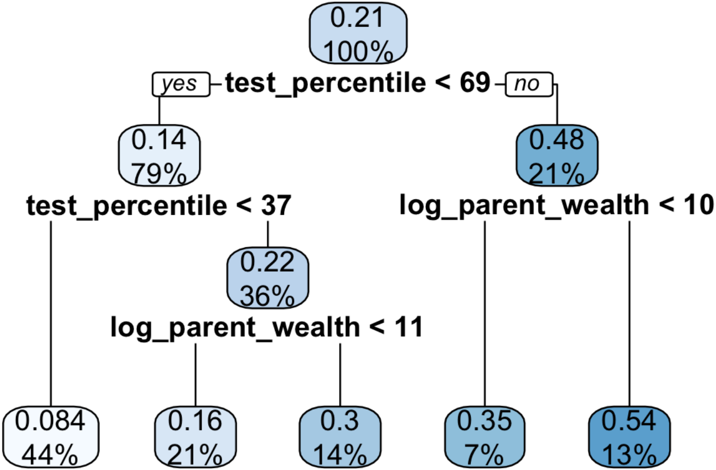
Pruning a tree

Sometimes you want a simpler decision rule

- you worry you are fitting to noise
- you want to explain predictions more easily

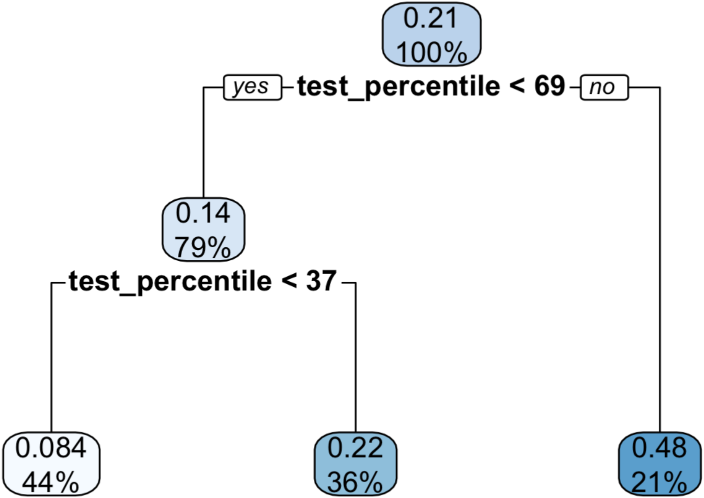
Then you prune the tree: Trim back some branches

Pruning a tree: Original tree



Pruning a tree: Pruned tree

pruned <- prune(rpart.out, cp = .02)



Discussion: Why prefer a tree vs OLS?

- Reasons to prefer a tree

- Reasons to prefer OLS

Discussion: Why prefer a tree vs OLS?

- Reasons to prefer a tree
  - No need to assume a functional form
  - Easy to explain how a prediction is made: follow the decision branches
- Reasons to prefer OLS

Discussion: Why prefer a tree vs OLS?

- Reasons to prefer a tree
  - No need to assume a functional form
  - Easy to explain how a prediction is made: follow the decision branches
- Reasons to prefer OLS
  - More widely known in social science
  - Better if the functional form is correct

From trees to forests

Trees are **high-variance** estimators

- Suppose we all have different samples
- We each estimate a tree
- Trees will look very different

From trees to forests

Forests aggregate trees to **reduce variance**

- For tree $1, \ldots, n_{\text{Trees}}$
  - Bootstrap the data
  - Randomly sample $p_{\text{Selected}} < p$ columns of the data
  - Learn a tree
- Then predict the average of the trees

From trees to forests

Together, we will try a forest on the course website page

From trees to forests

Why might you prefer a tree?
Why might you prefer a forest?

Learning goals for today

By the end of class, you will be able to

- understand the notion of supervised machine learning
  - an input-output machine
  - learned on some learning cases
  - used to predict for new cases
- apply that notion to the specific case of regression trees
- read a prediction from a regression tree
- understand how trees can aggregate to a forest