Welcome to CME 250 Introduction to Machine Learning!

Spring 2020 – Online version

April 30th 2020

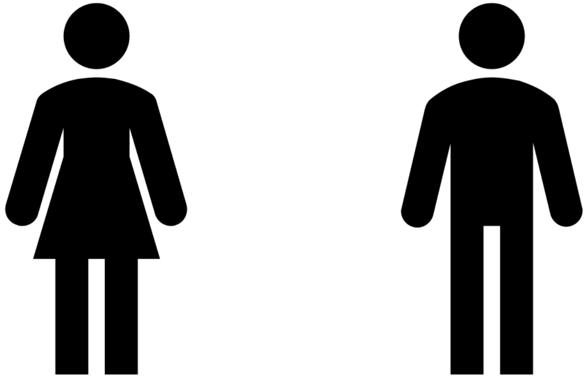
Stanford | Institute for Computational & Mathematical Engineering

Today's schedule: Tree-based methods

- The easiest way to classify: Decision Trees
- How to reduce variance:
  - Bagging: Creating multiple identically distributed trees
  - Random Forest: Creating multiple identically distributed and uncorrelated trees
  - Gradient Boosting trees: Creating incrementally better trees.

Let's get to know each other…

Breakout room



You

Another student

Name

Location

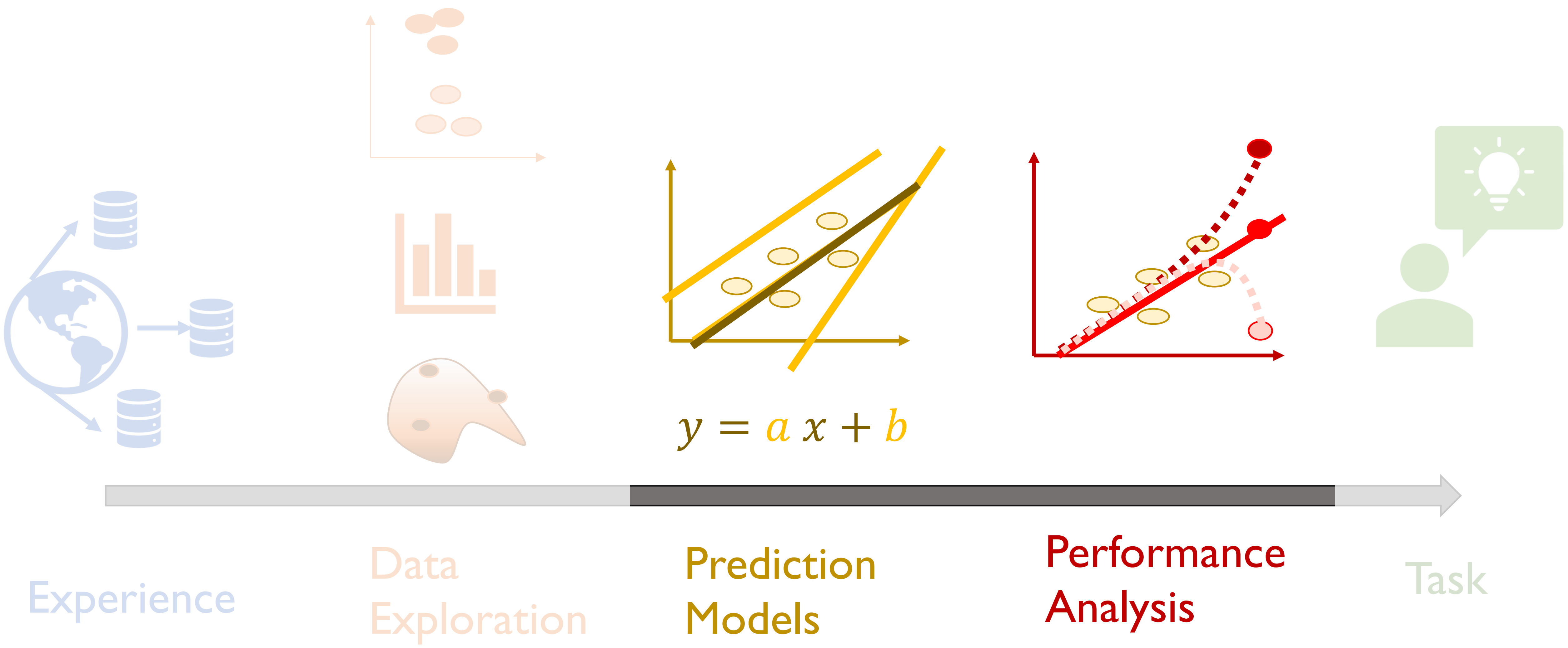
Department

Year

Who is your new office mate?
E.g.: Family, Pet, Plant, Fridge, Noisy neighbor ...

3 mins

Chat/Audio/Video



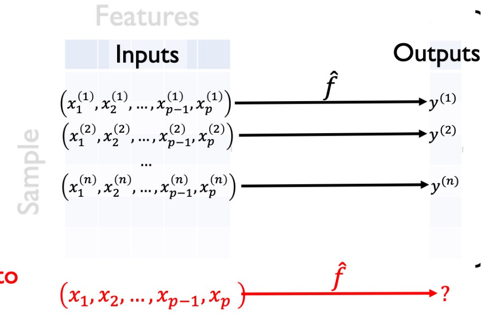
$y = a\,x + b$
Experience
Data Exploration
Prediction Models
Performance Analysis
Task

Recap

Supervised Learning

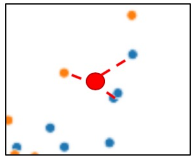
Learn from examples

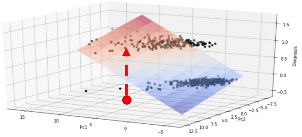
Given

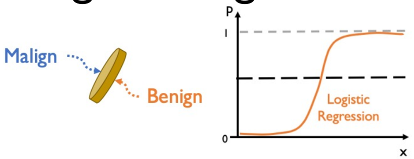


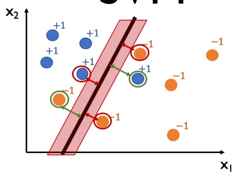
We want to predict

	Regression Y is quantitative	Classification Y is categorical
KNN	✓	✓
Linear Regression	✓	Using dummy variables?
Logistic Regression	✗	✓
SVM	✗	✓

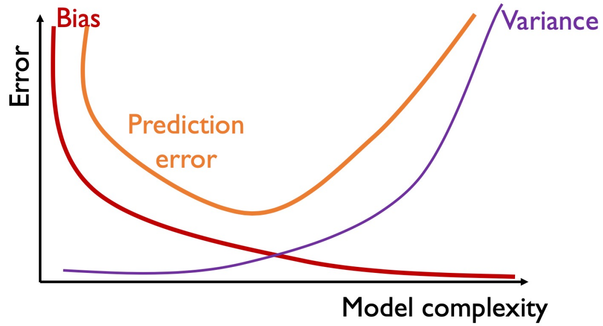




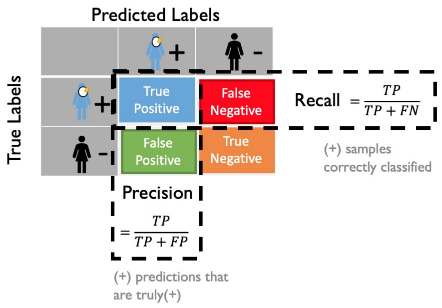




Model Selection

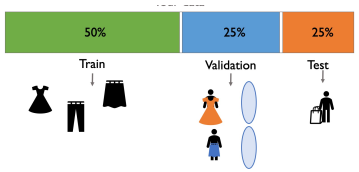


Confusion Matrix



Cross-Validation

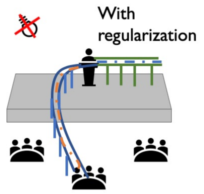
Estimate prediction error



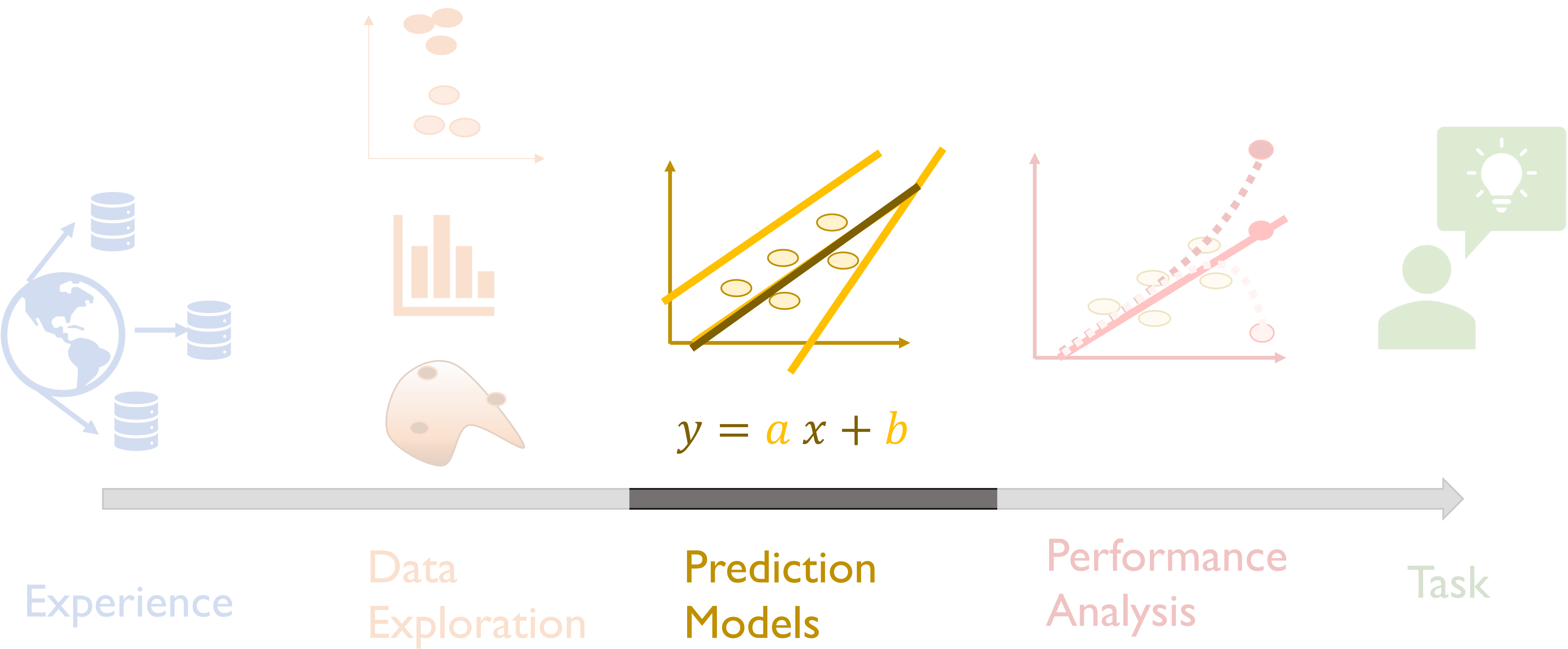
K-fold CV, LOOCV

Regularization

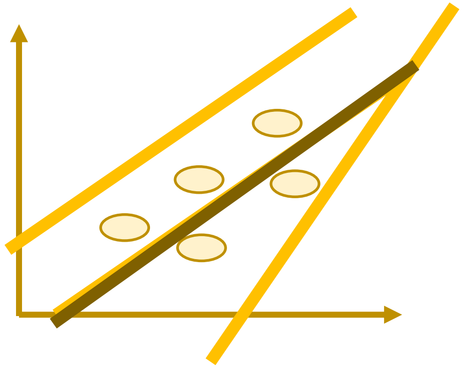
Control complexity: Hyperparameters



Ridge, Lasso



y = a x + b
Experience
Data Exploration
Prediction Models
Performance Analysis
Task



$$y = a\,x + b$$

Prediction Models

Supervised Learning Part III: Tree-based methods

Introduction to Statistical Learning

Chapter 8: Tree-based methods

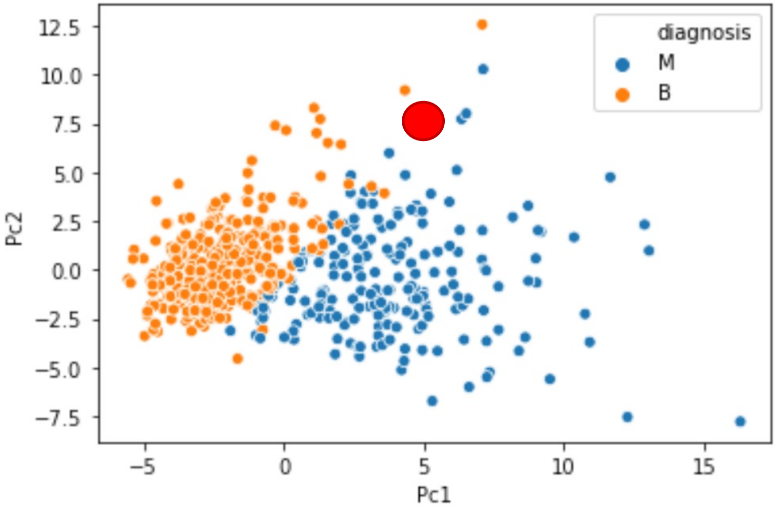
Elements Statistical Learning

Chapter 9.2: Tree-based methods

Chapter 10: Boosting and Additive Trees

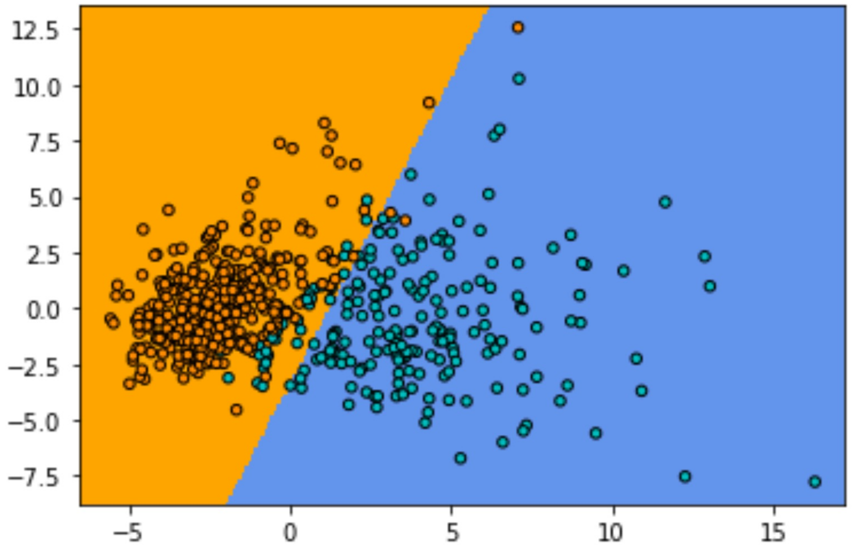
Chapter 15: Random Forest

Easy dataset to classify

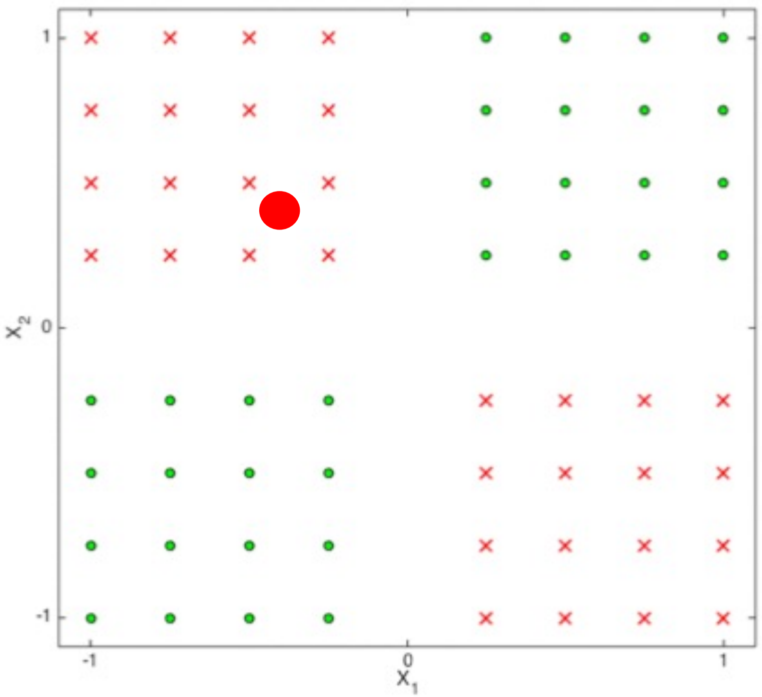


What is the diagnosis for this sample?

KNN
Linear Regression
Logistic Regression
SVM

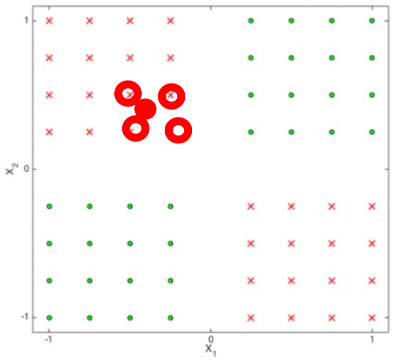


Not so easy dataset to classify



What is the diagnosis for this sample?

KNN

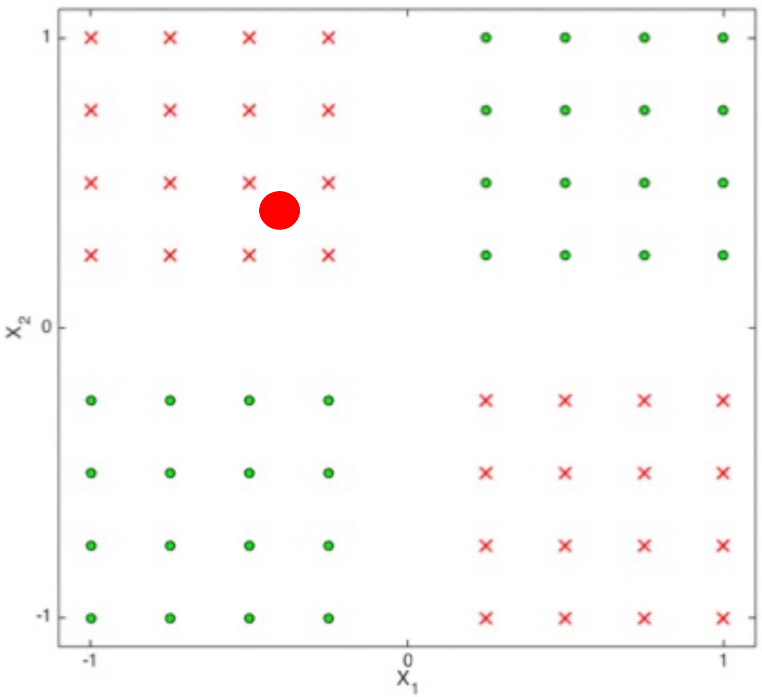


No idea how features interact

Linear Regression
Logistic Regression
SVM

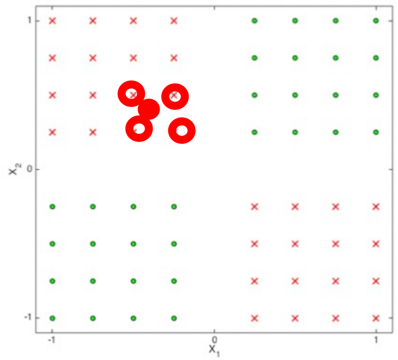
No good predicting model using x_1, x_2

Not so easy dataset to classify



What is the diagnosis for this sample?

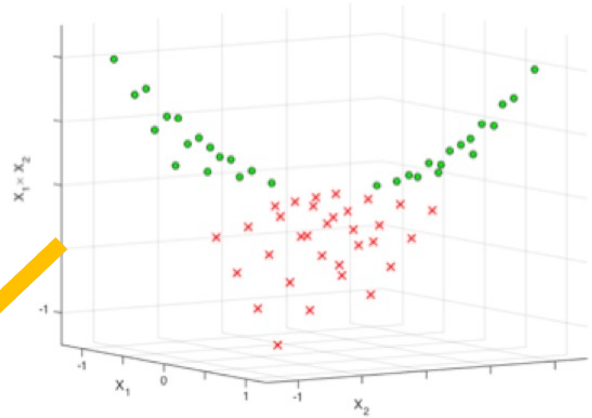
KNN



New feature space

No idea how features interact

Linear Regression
Logistic Regression
SVM

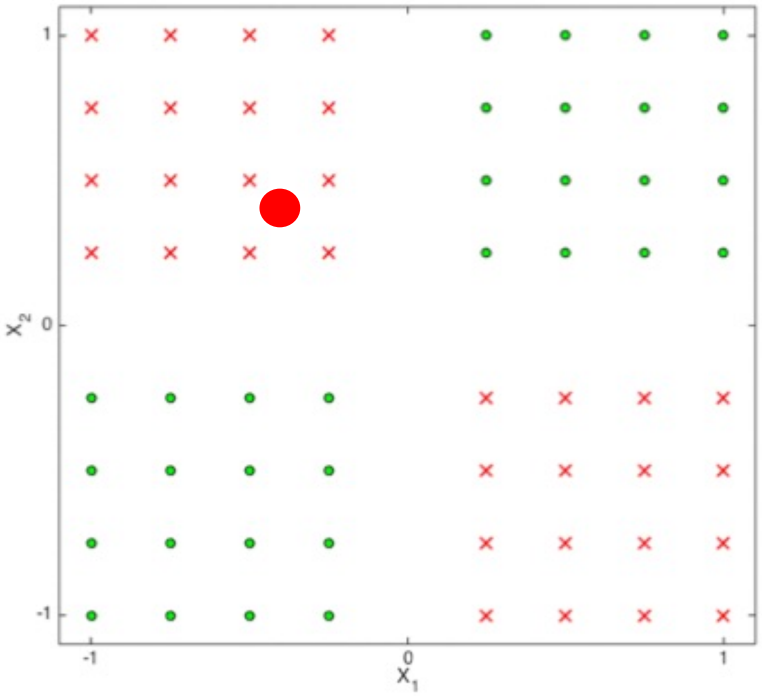


variables x_1, x_2, x_1x_2

Good predicting model using x_1, x_2, x_1x_2

How to find new features/kernels that work?

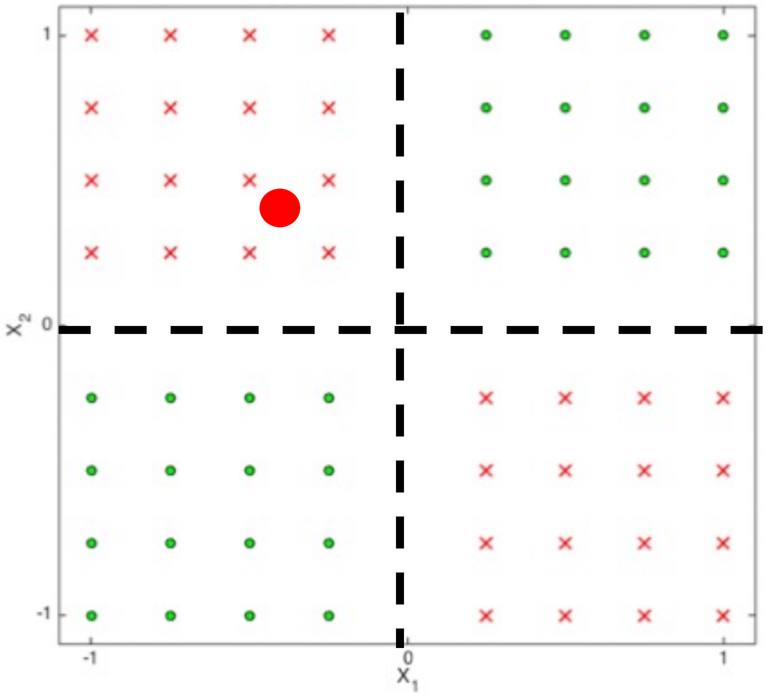
Let’s take a step back …



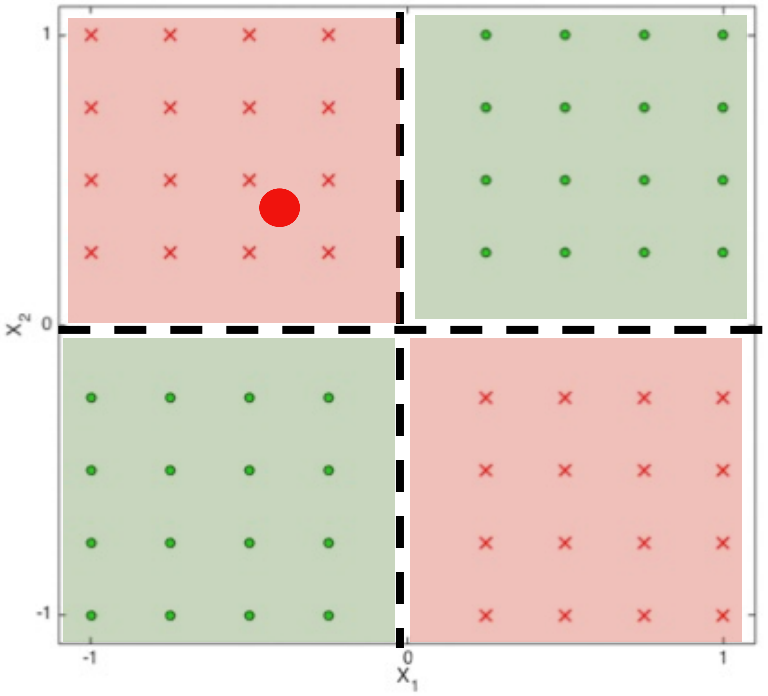
What is the diagnosis for this sample?

How would you explain this data set in simple words?

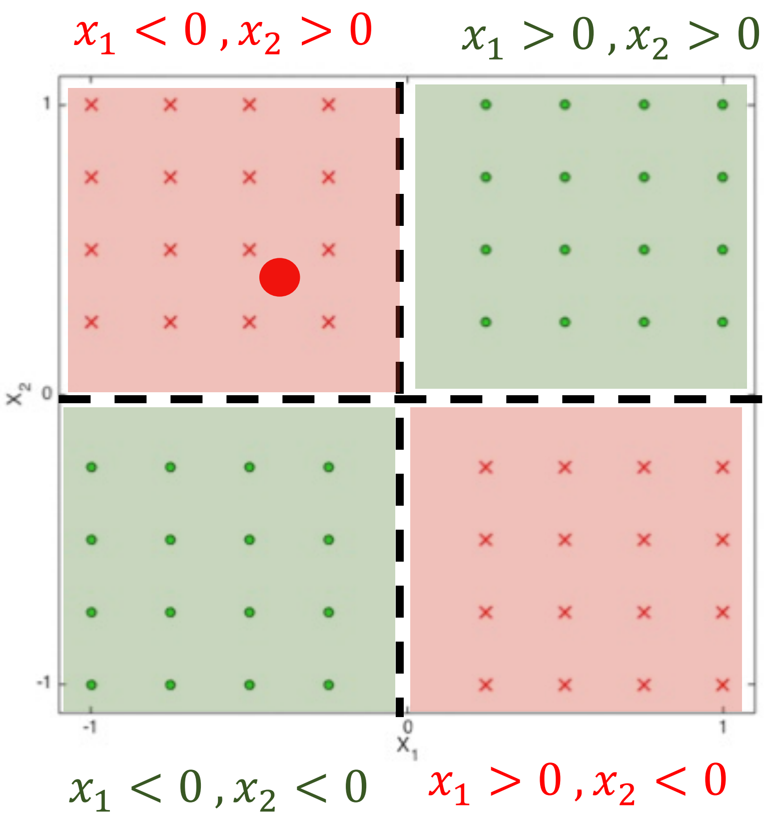
Let's take a step back …



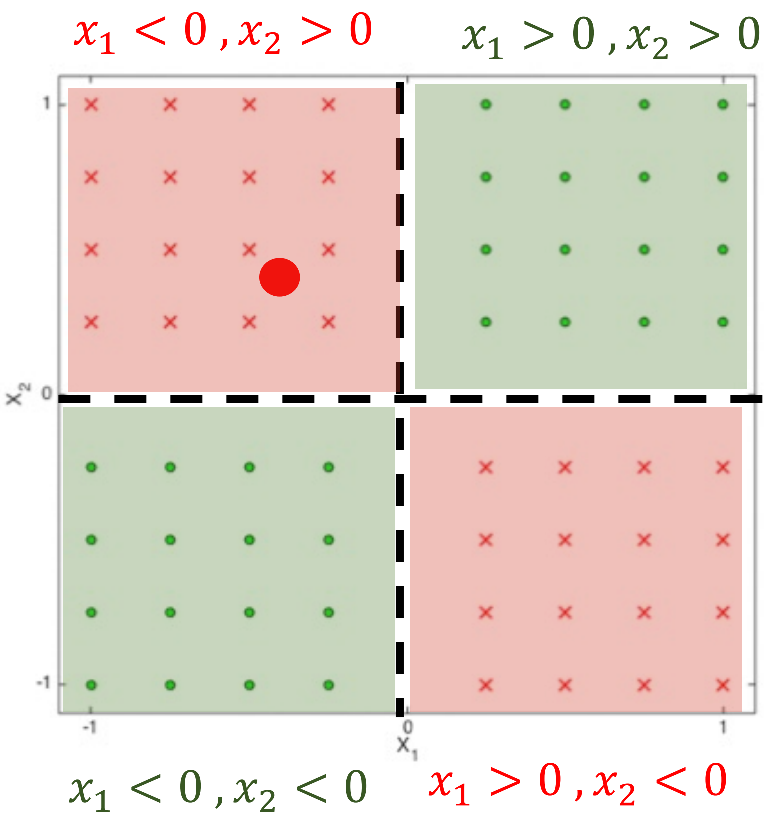
Let’s take a step back …

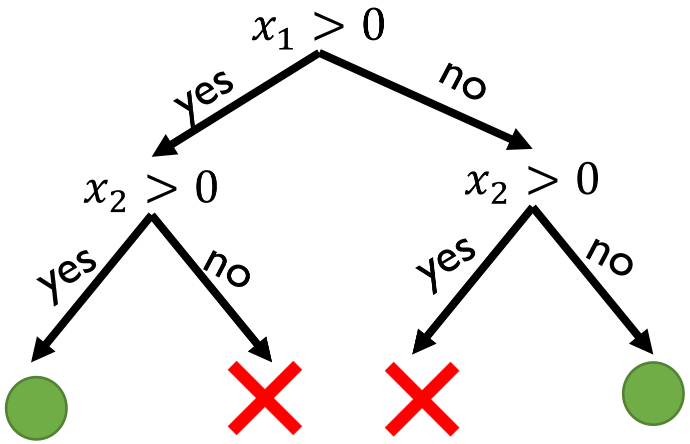


Let's take a step back …



Let's take a step back …



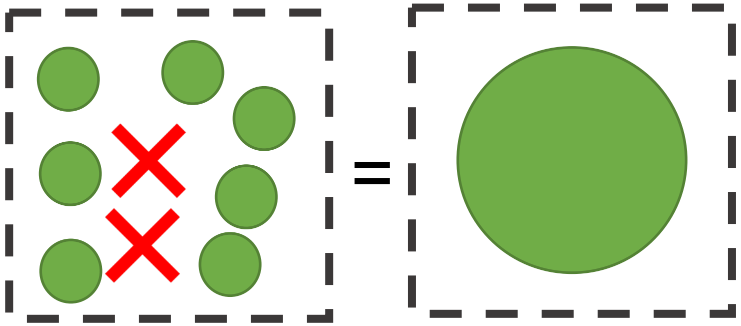


Easiest way to find decision boundaries = Decision Tree

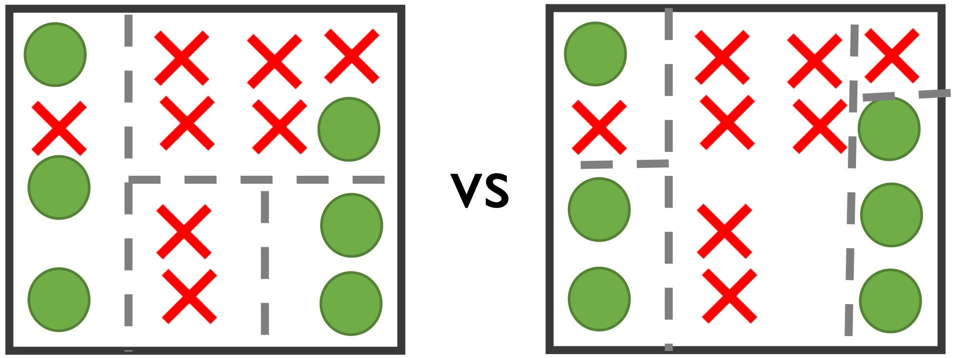
Creating CART: Classification and Regression Tree

Divide Feature space into *uniform* regions*

How to assign a value inside each region?



How to choose regions?



*Sounds familiar to clustering

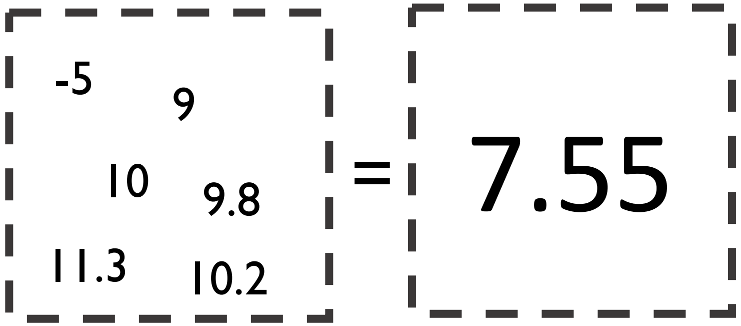
Clustering — Group samples with **similar features**

Decision Trees — Group samples with **similar outputs**

How to assign a value inside each region?

Similar to K-NN

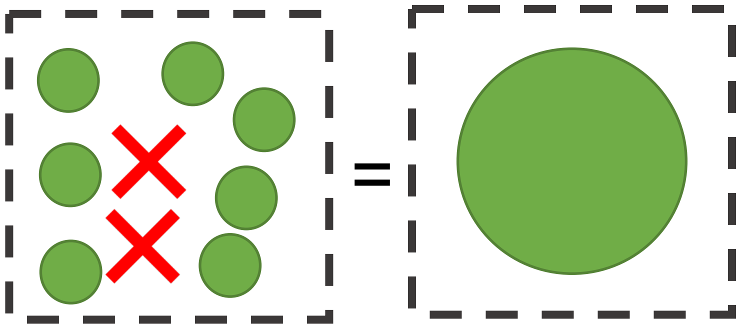
Regression



Average

$$\widehat{y_{R_k}} = \frac{1}{|R_k|} \sum_{j \in R_k} y^{(j)}$$

Classification



$\widehat{y_{R_k}} =$ Most popular category

How to choose regions?

Combinatorial problem: Get suboptimal solution

Recursive binary splitting

Find :

1. Input feature X_j
2. Cutpoint s

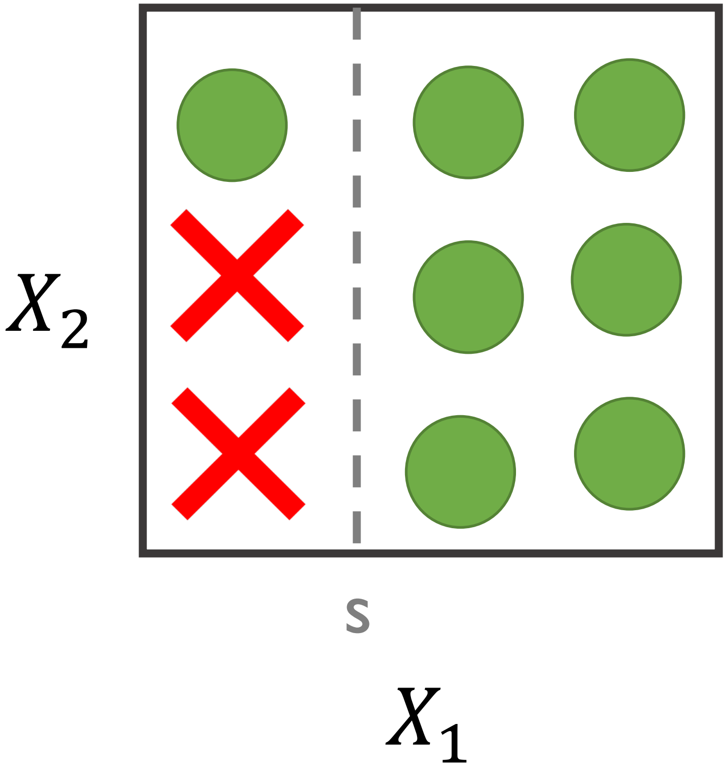
$$R_1 = \{x^{(i)} | x_j^{(i)} < s\}$$

$$R_2 = \{x^{(i)} | x_j^{(i)} \geq s\}$$

such that

$$\sum_{x^{(i)} \in R_1} L(y^{(i)}, \widehat{y_{R_1}}) + \sum_{x^{(i)} \in R_2} L(y^{(i)}, \widehat{y_{R_2}})$$

is minimized



How to choose regions?

Combinatorial problem: Get suboptimal solution

Recursive binary splitting

Find :

1. Input feature X_j
2. Cutpoint s

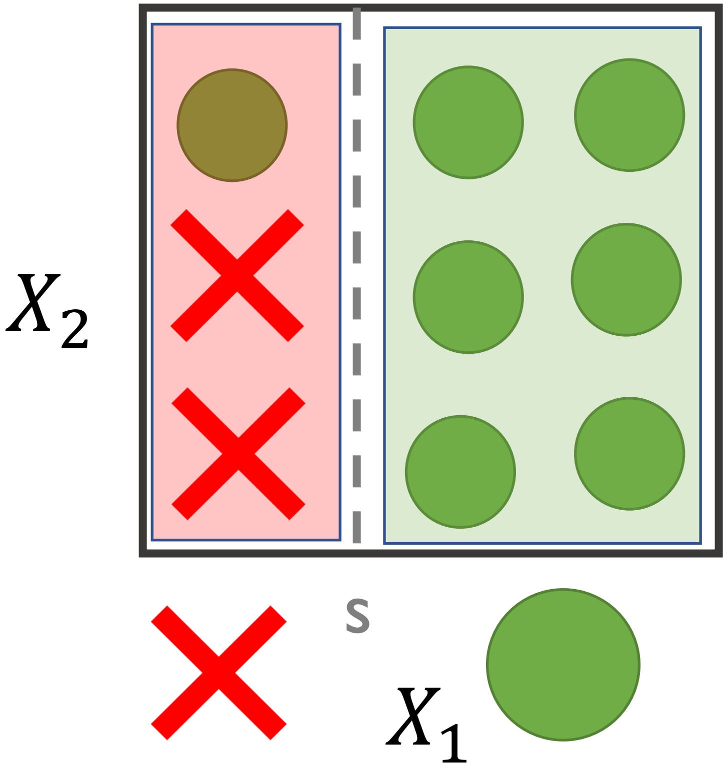
$$R_1 = \{x^{(i)} | x_j^{(i)} < s\}$$

$$R_2 = \{x^{(i)} | x_j^{(i)} \geq s\}$$

such that

$$\sum_{x^{(i)} \in R_1} L\left(y^{(i)}, \widehat{y_{R_1}}\right) + \sum_{x^{(i)} \in R_2} L\left(y^{(i)}, \widehat{y_{R_2}}\right)$$

is minimized



How to choose regions?

Combinatorial problem: Get suboptimal solution

Recursive binary splitting

Find :

1. Input feature X_j
2. Cutpoint s

$$R_1 = \left\{x^{(i)} | x_j^{(i)} < s\right\}$$

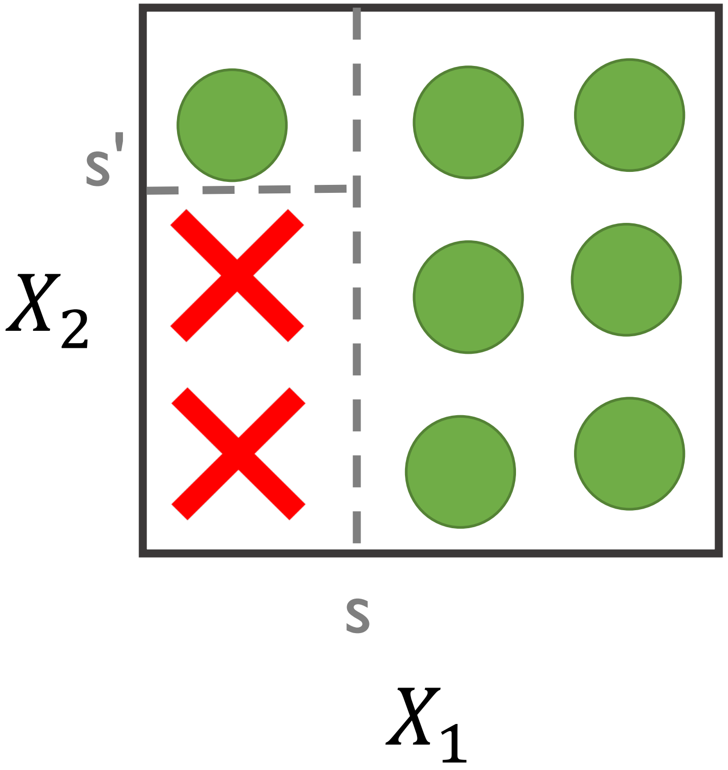
$$R_2 = \left\{x^{(i)} | x_j^{(i)} \geq s\right\}$$

such that

$$\sum_{x^{(i)} \in R_1} L\left(y^{(i)}, \widehat{y_{R_1}}\right) + \sum_{x^{(i)} \in R_2} L\left(y^{(i)}, \widehat{y_{R_2}}\right)$$

is minimized

We iterate in each of the subregions



How to choose regions?

Combinatorial problem: Get suboptimal solution

Recursive binary splitting

Find :

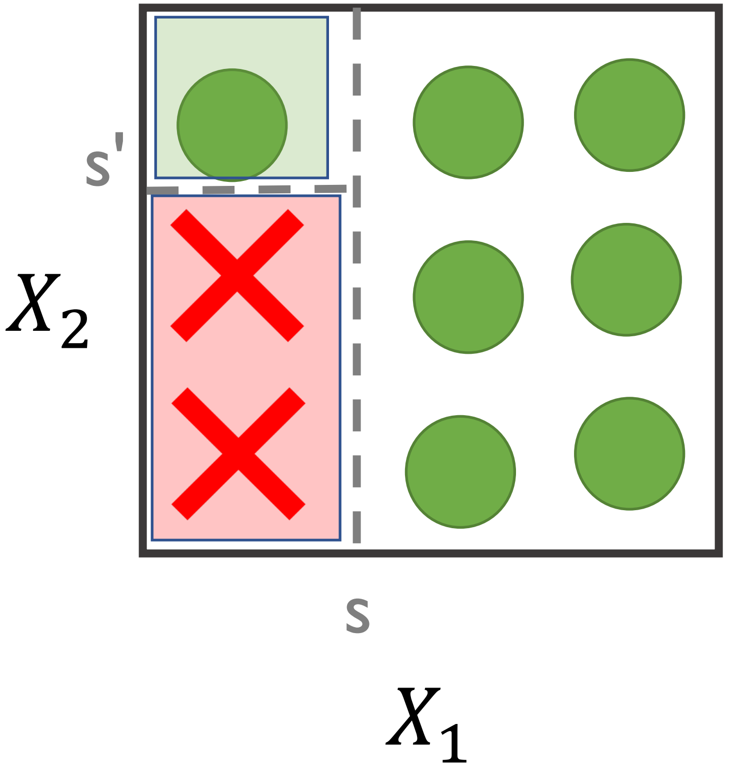
1. Input feature X_j
2. Cutpoint s

$$R_1 = \{x^{(i)} | x_j^{(i)} < s\}$$
$$R_2 = \{x^{(i)} | x_j^{(i)} \geq s\}$$

such that

$$\sum_{x^{(i)} \in R_1} L(y^{(i)}, \widehat{y_{R_1}}) + \sum_{x^{(i)} \in R_2} L(y^{(i)}, \widehat{y_{R_2}})$$

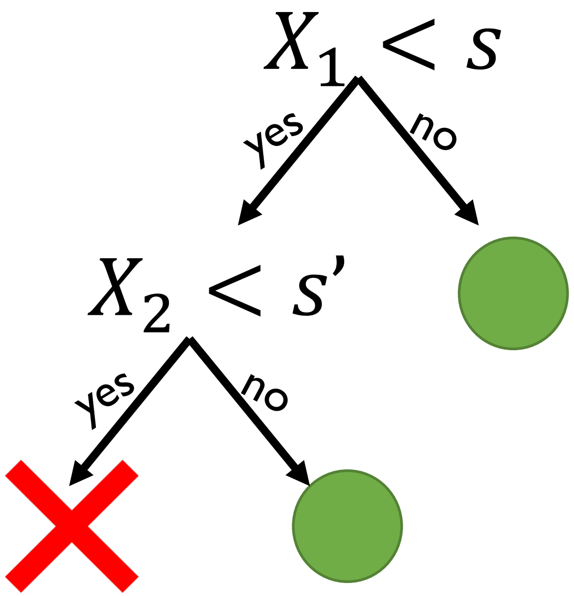
is minimized

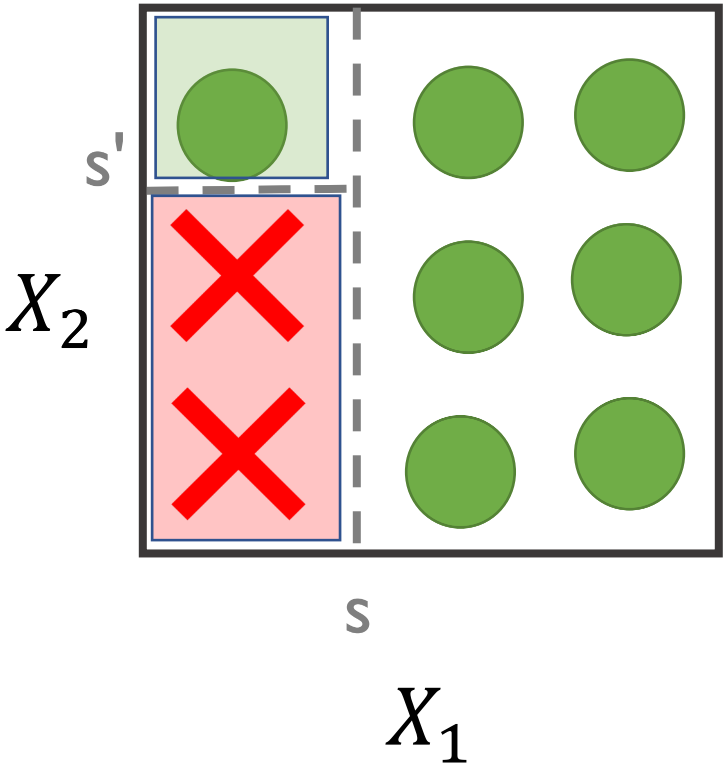


We iterate in each of the subregions

How to choose regions?

Combinatorial problem: Get suboptimal solution





For categorical variables: use dummy variables

How to compute loss?

Regression

RSS: Residual sum of squares

$$L\left(y^{(i)}, \widehat{y_{R_k}}\right) = \left(y^{(i)} - \widehat{y_{R_k}}\right)^2$$

Classification

Misclassification error

$$L\left(y^{(i)}, \widehat{y_{R_k}}\right) = I\left(y^{(i)} \neq \widehat{y_{R_k}}\right)$$

How to compute loss?

Regression

RSS: Residual sum of squares

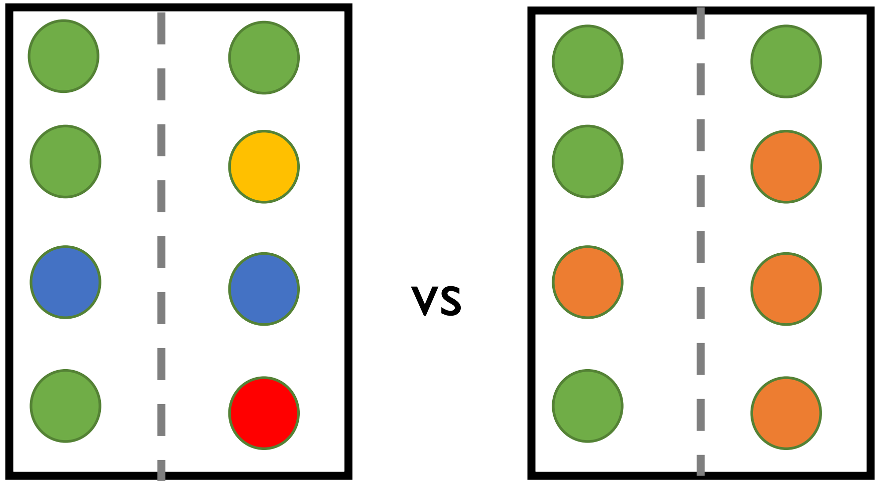
$$L\left(y^{(i)}, \widehat{y_{R_k}}\right) = \left(y^{(i)} - \widehat{y_{R_k}}\right)^2$$

Classification

Misclassification error

$$L\left(y^{(i)}, \widehat{y_{R_k}}\right) = I\left(y^{(i)} \neq \widehat{y_{R_k}}\right)$$

Not always the best



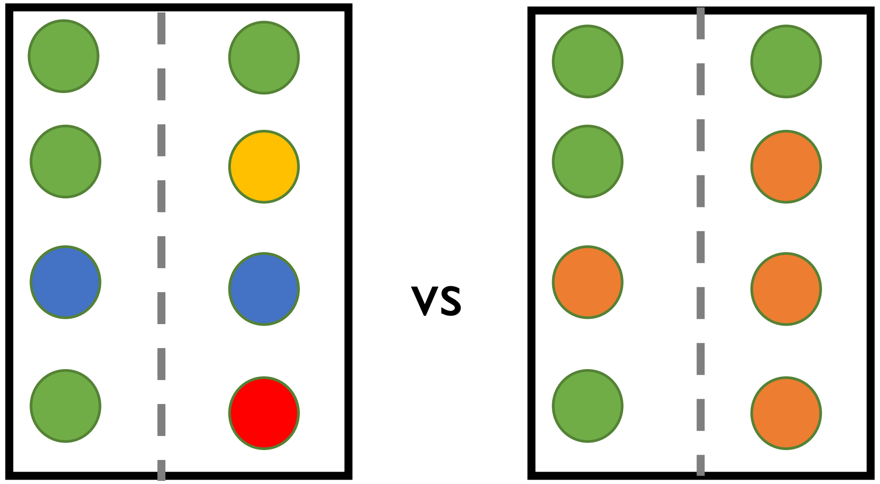
How to compute loss for classification?

Measure region **purity** with categories k

$$\hat{p}_{R_m,k} = \frac{1}{|R_m|} \sum_{x^{(i)} \in R_m} I\left(y^{(i)} = k\right)$$

Gini Index $G = \sum_{k=1}^{K} \hat{p}_{R_m,k}(1 - \hat{p}_{R_m,k})$

Cross-Entropy $D = -\sum_{k=1}^{K} \hat{p}_{R_m,k} \log(\hat{p}_{R_m,k})$

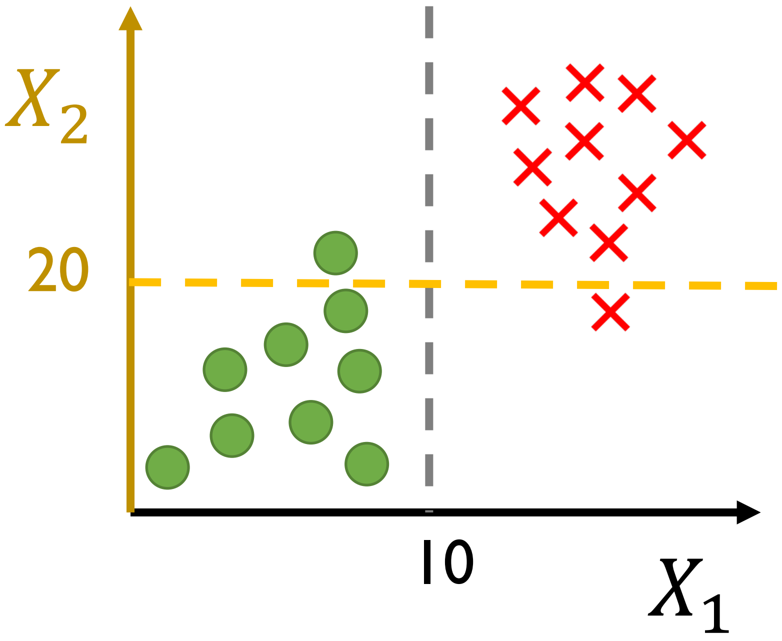


Pros of Decision Trees

- Interpretability
- Capture non-linear interactions
- Works with missing data.

  Using surrogate predictor

X_1	X_2	Y
?	10	



Cons of Decision Trees

❌ Hard to capture additivity (vs Linear Regression)

There are generalizations:

- MARS: Multivariate Adaptive Regression Splines
- MART: Multiple Additive Regression Trees

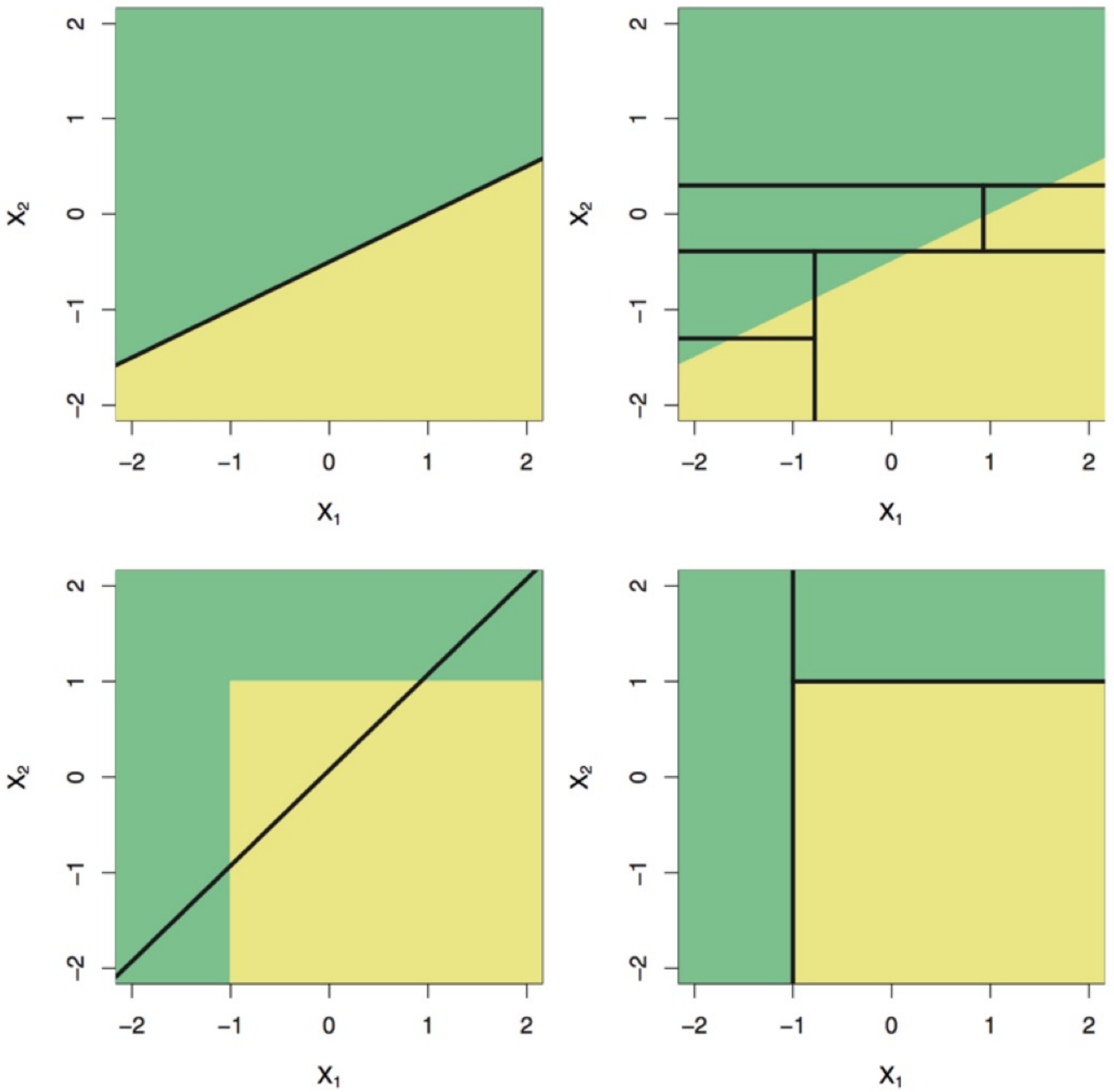
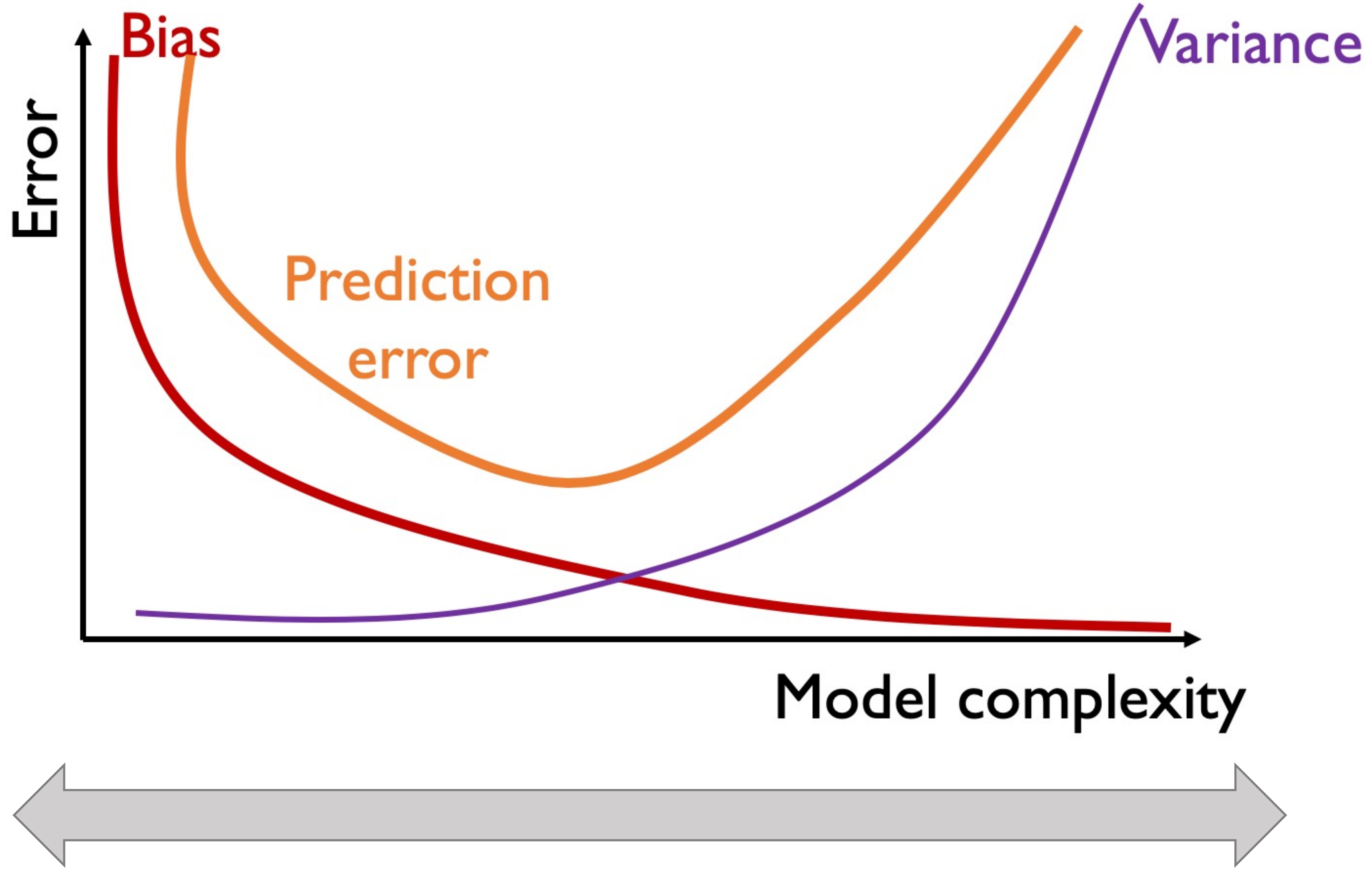


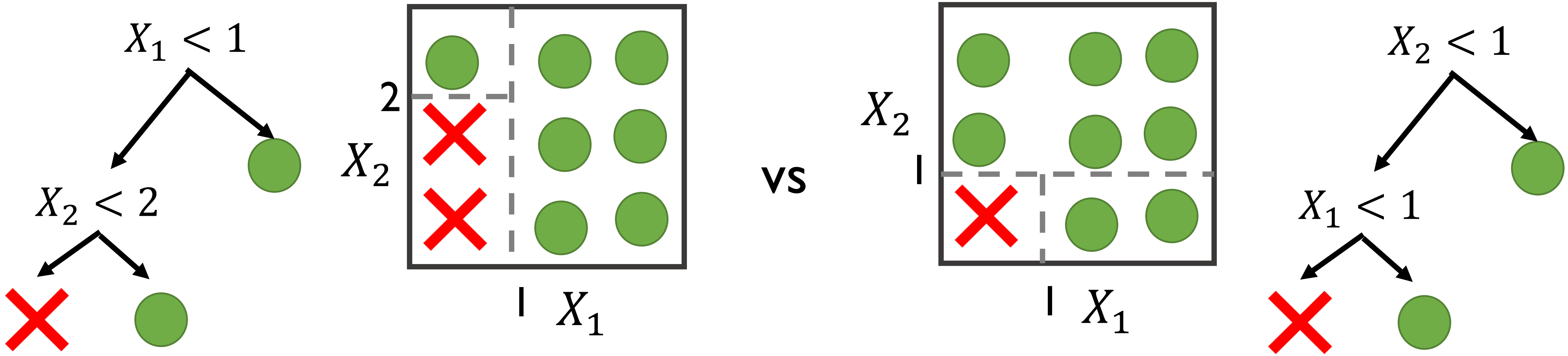
FIGURE 8.7, ISL (8th printing 2017)

How is the variance – bias tradeoff of decision trees?



Cons of Decision Trees

Overfitting and Instability



Cons of Decision Trees

Overfitting and Instability

Bias

Solution 1: **Tree Pruning** Reduce model complexity

Variance

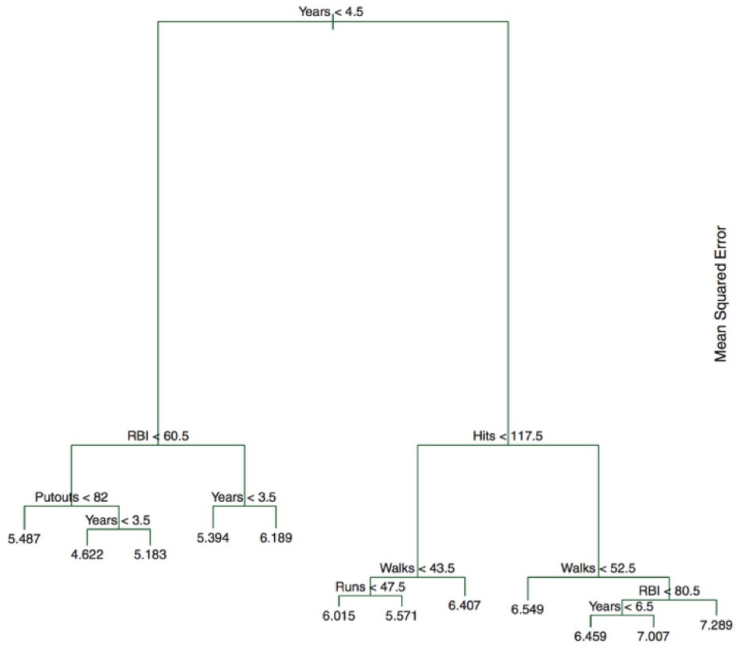


FIGURE 8.4, ISL (8th printing 2017)

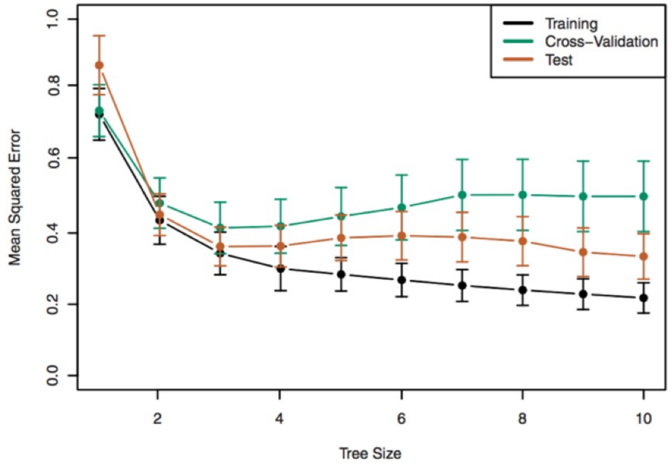


FIGURE 8.5, ISL (8th printing 2017)

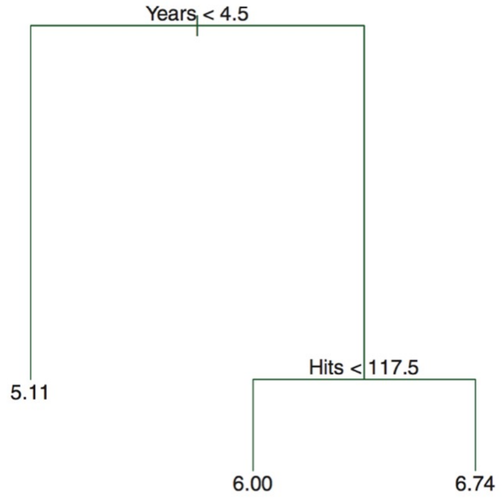


FIGURE 8.1, ISL (8th printing 2017)

Cons of Decision Trees

❌ Overfitting and Instability

Solution 2: **Tree "Averaging"**

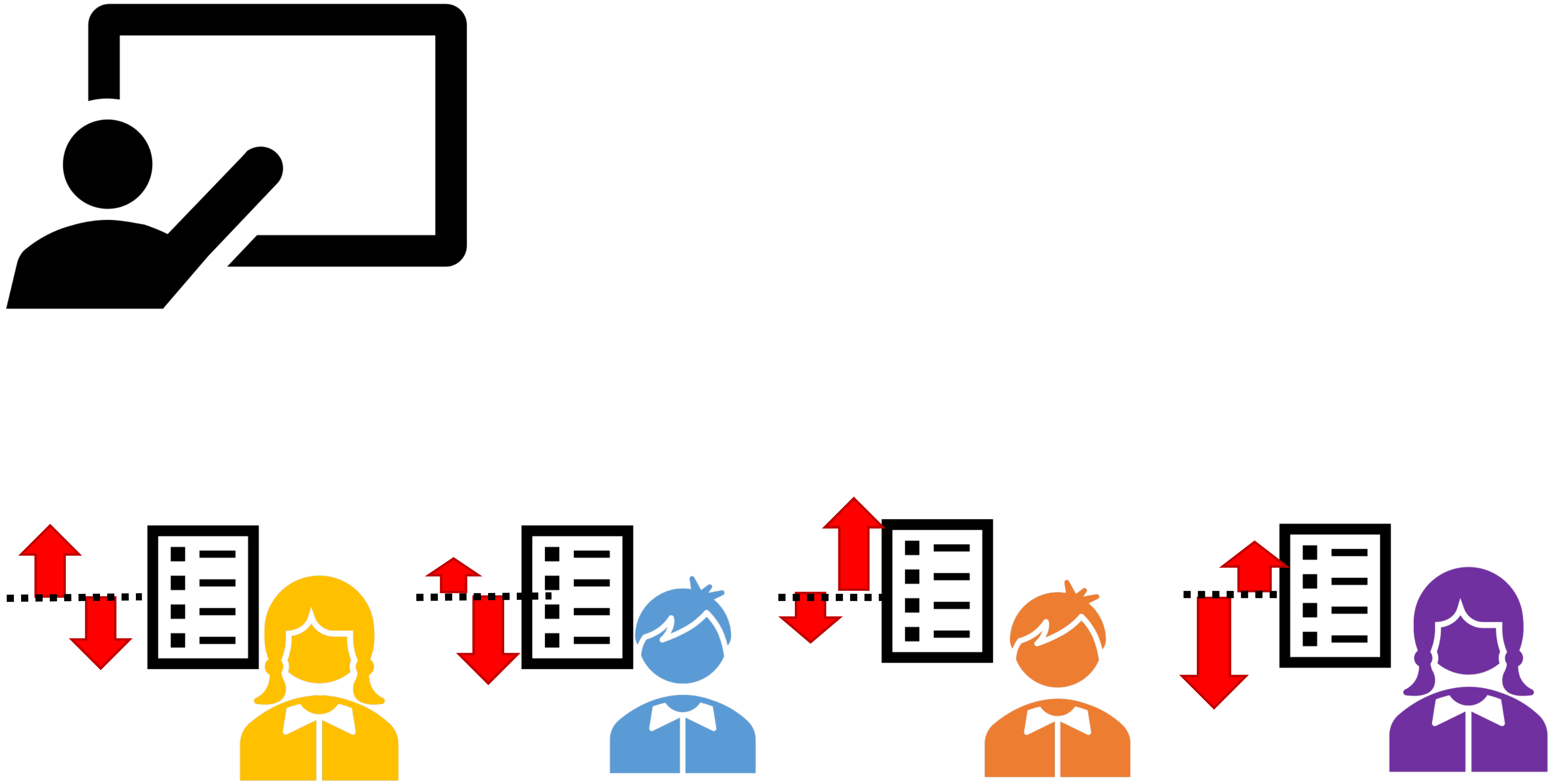
Reduce variance

Bias =

Variance ↓

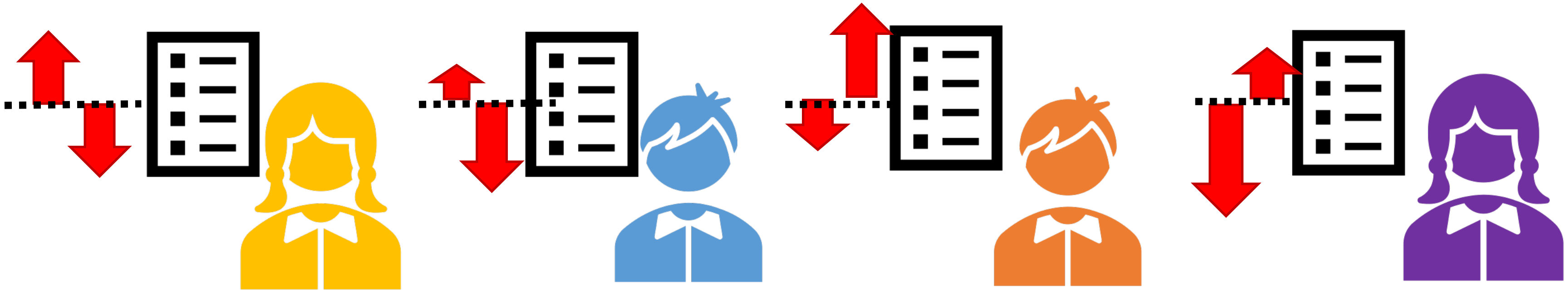
Motivation: If we have N **independent** random variables with variance σ^2, then the variance of the **average** is $\frac{\sigma^2}{N}$

Some intuition: How can we average?

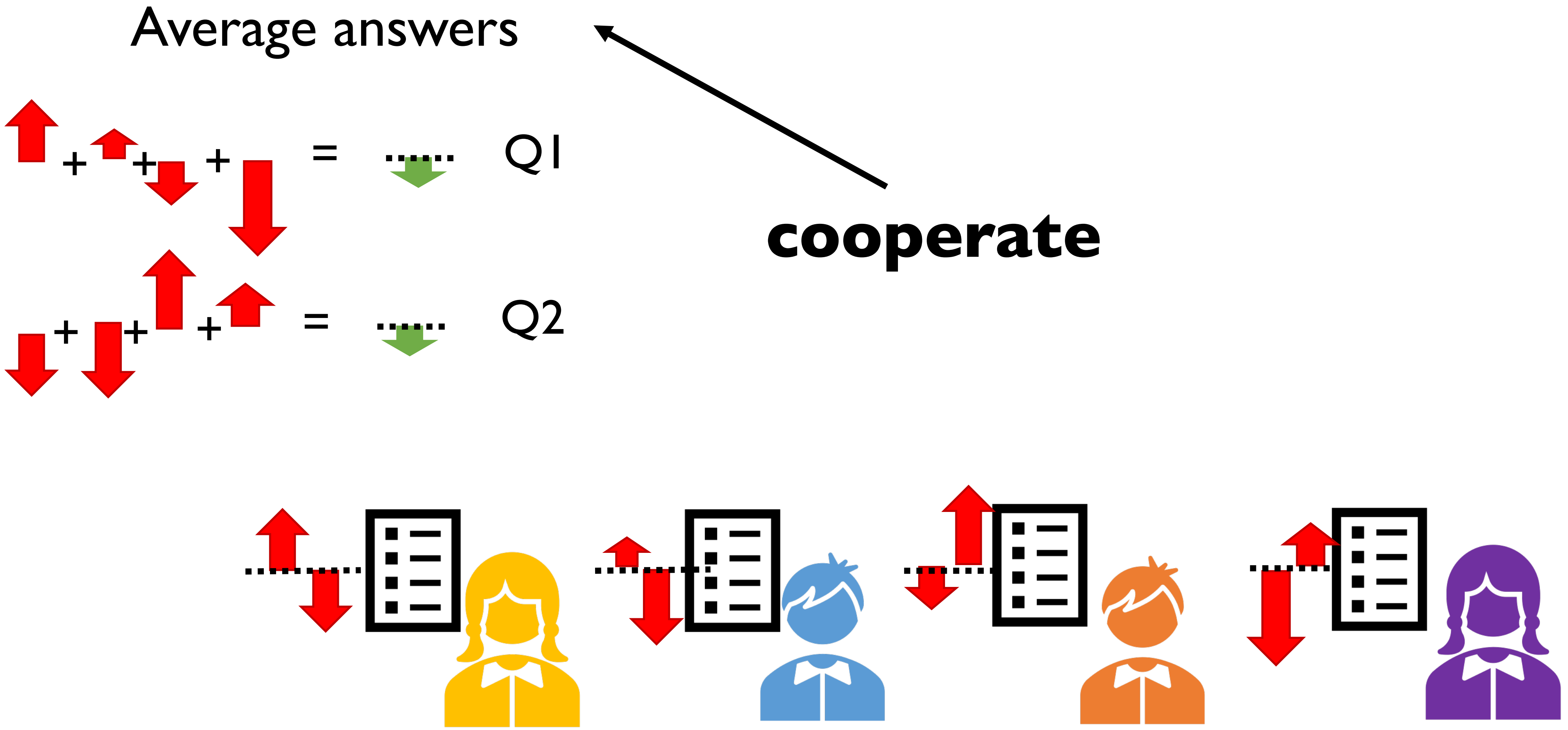


Some intuition: How can we average?

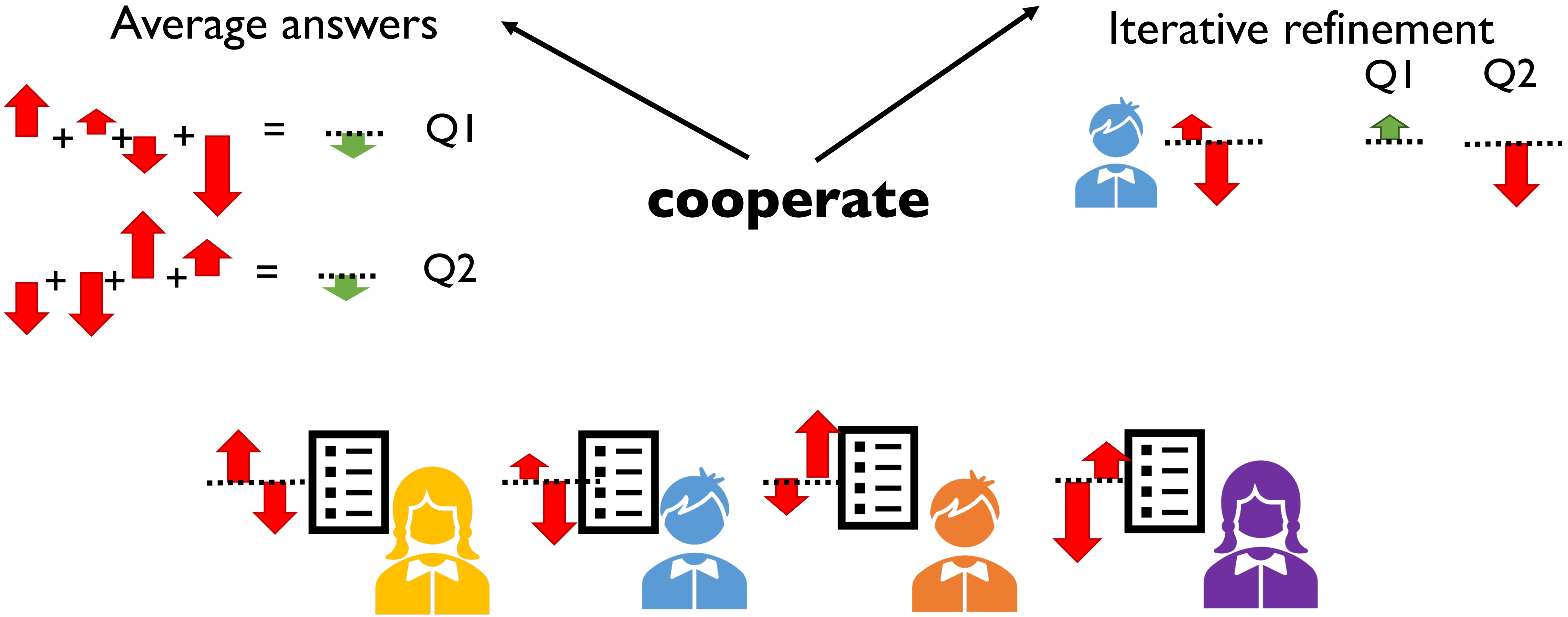
“You can cooperate”



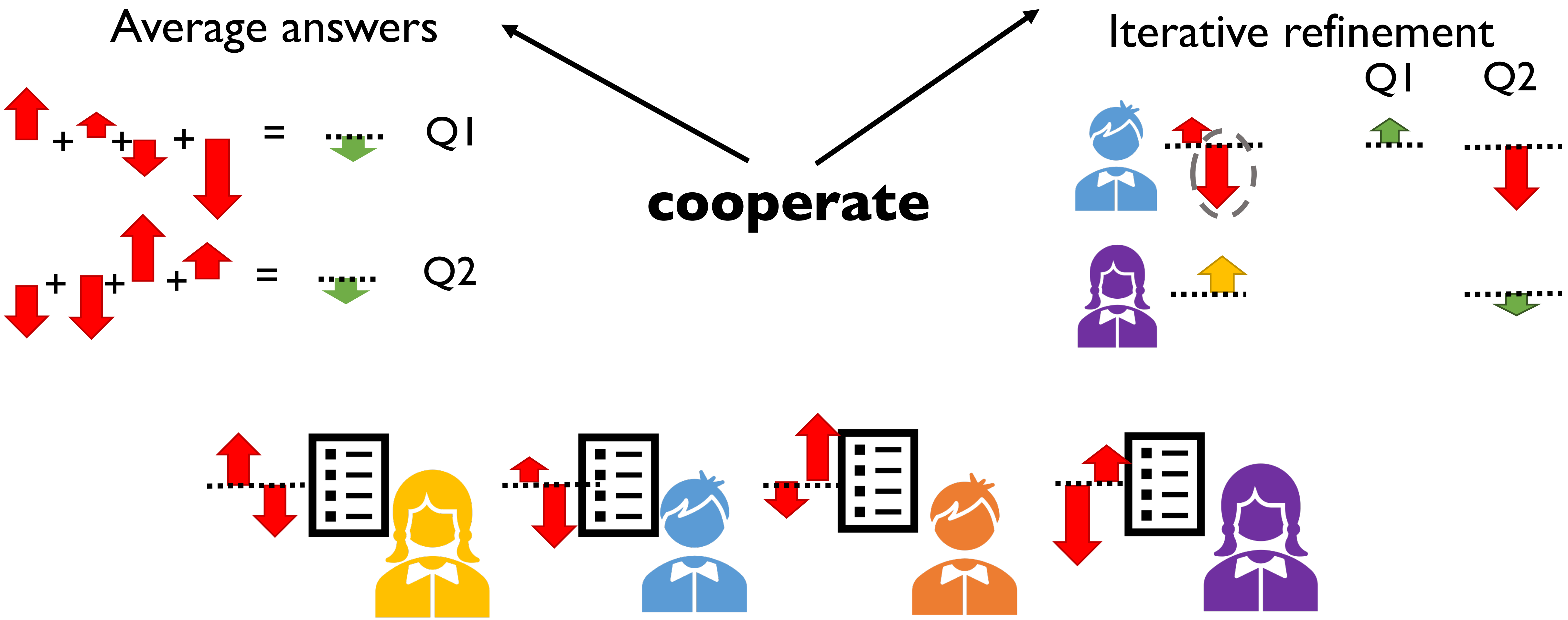
Some intuition: How can we average?



Some intuition: How can we average?

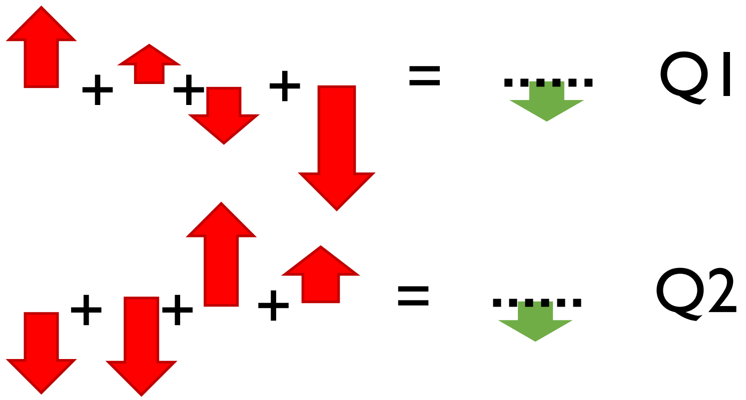


Some intuition: How can we average?



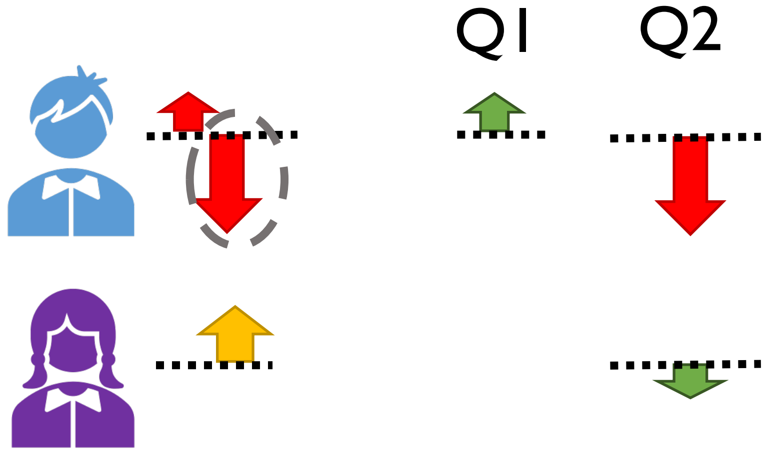
How can we combine trees?

Average answers



Bagging

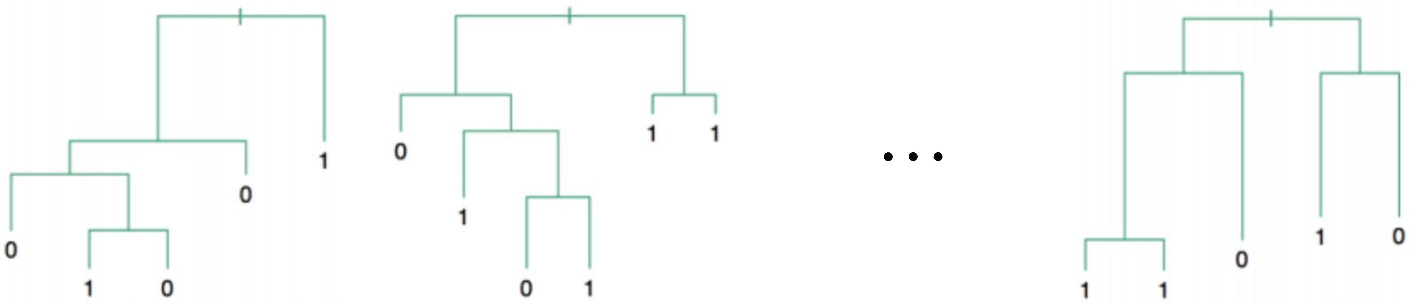
Iterative refinement



Boosting

Bagging: Averaging Trees

If we have B trees



$\hat{f}^1(x)$ $\hat{f}^2(x)$ … $\hat{f}^B(x)$

Then

$$\hat{f}_{\text{avg}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x)$$ (regression) or Most popular category (classification)

How do we construct B trees?

How do we construct B trees?

Option 1)

Create B subsets of the data

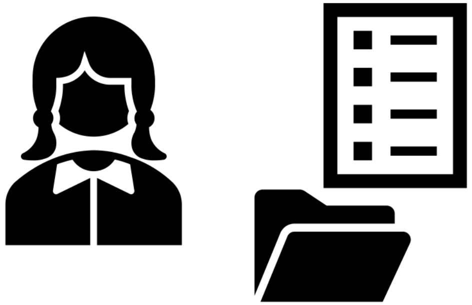
Each subset will be smaller, bias will increase

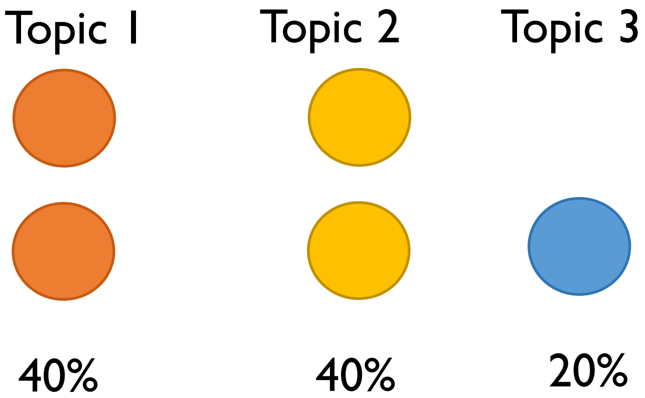
Option 2)

Create B identically distributed samples of size N

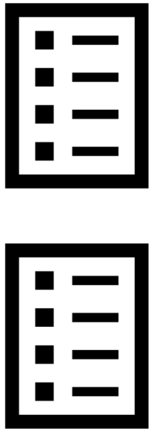
Bootstrap

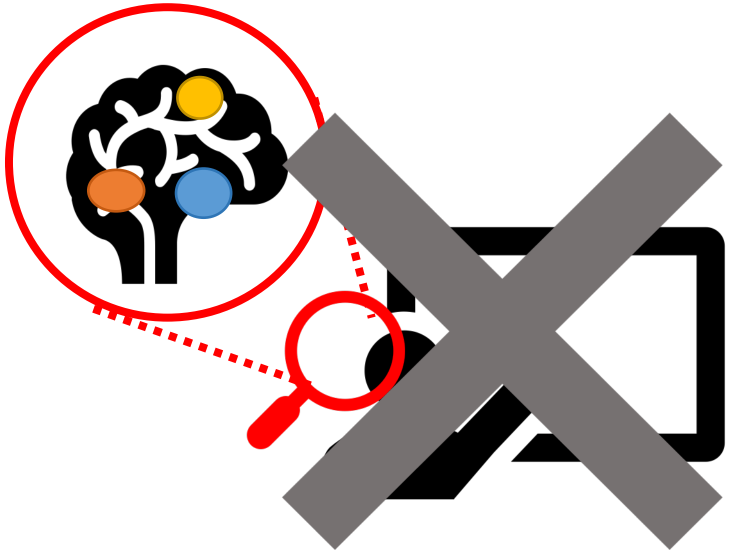
What is Bootstrap?



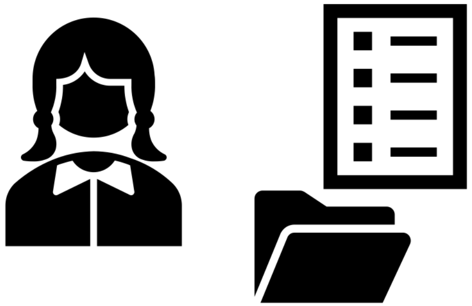


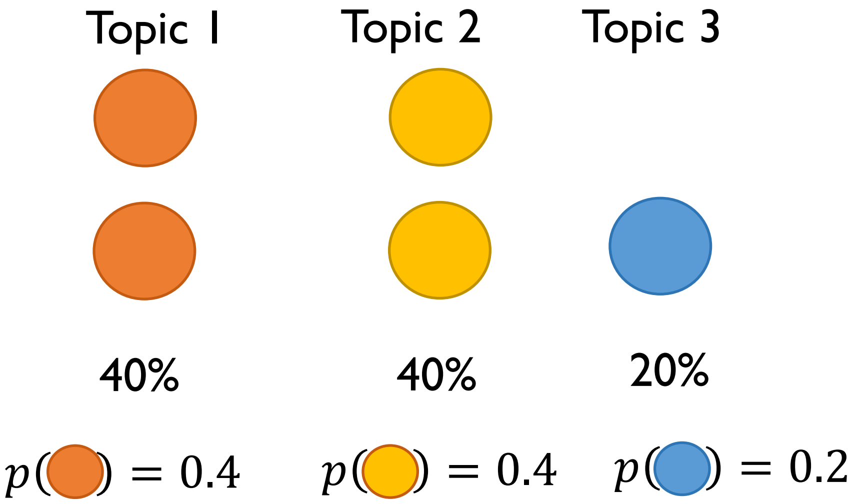
How to create new samples?



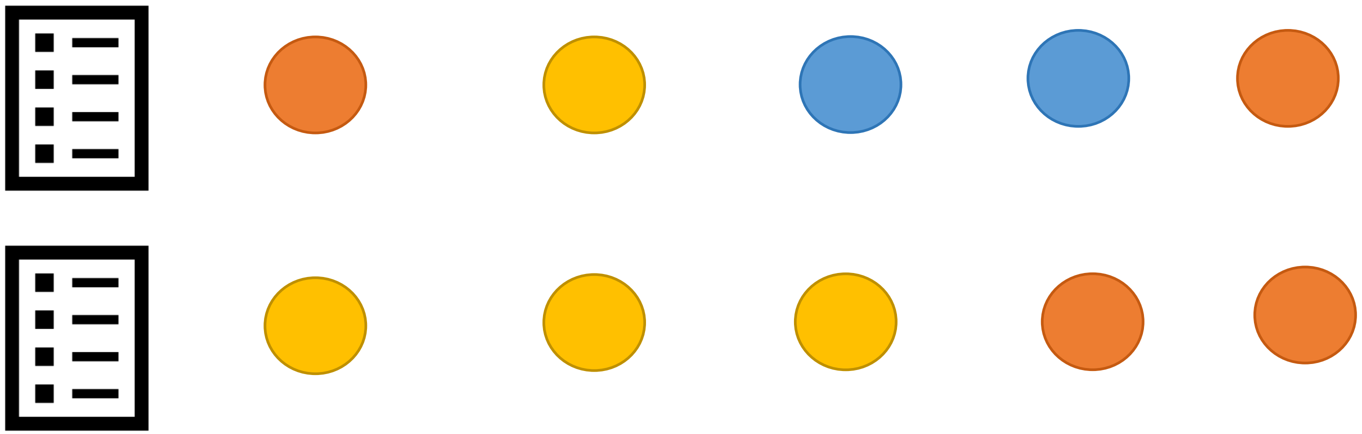


What is Bootstrap?



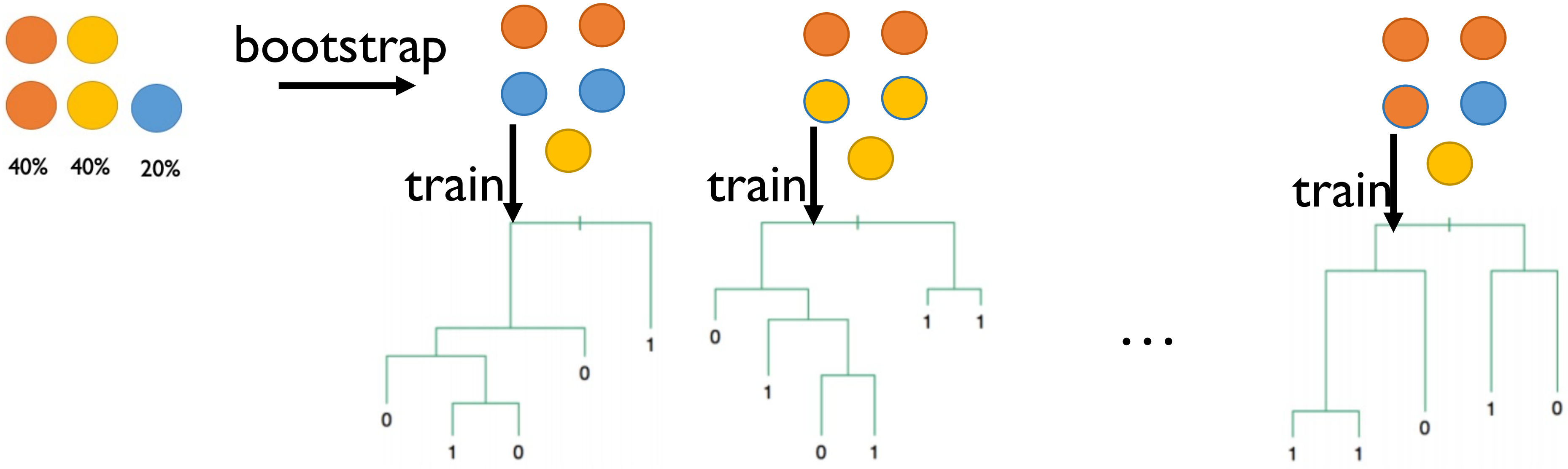


How to create new samples?



Draw each sample independently

Bagging: Bootstrap Aggregation



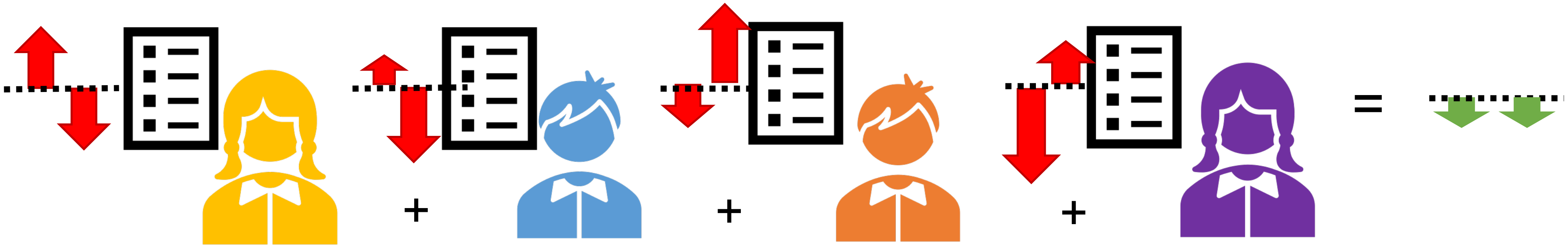
$\hat{f}^1(x)$ $\hat{f}^2(x)$ $\hat{f}^B(x)$

Then

$$\hat{f}_{\text{avg}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x)$$ (regression) or Most popular category (classification)

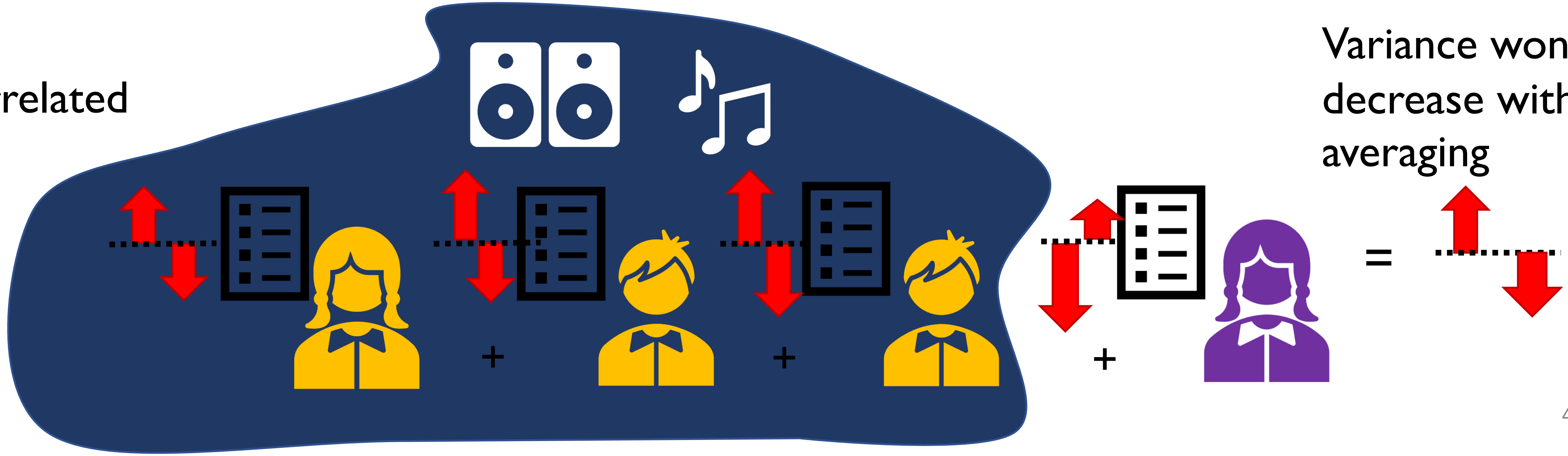
Bootstrap samples are not uncorrelated

Uncorrelated



vs

Correlated

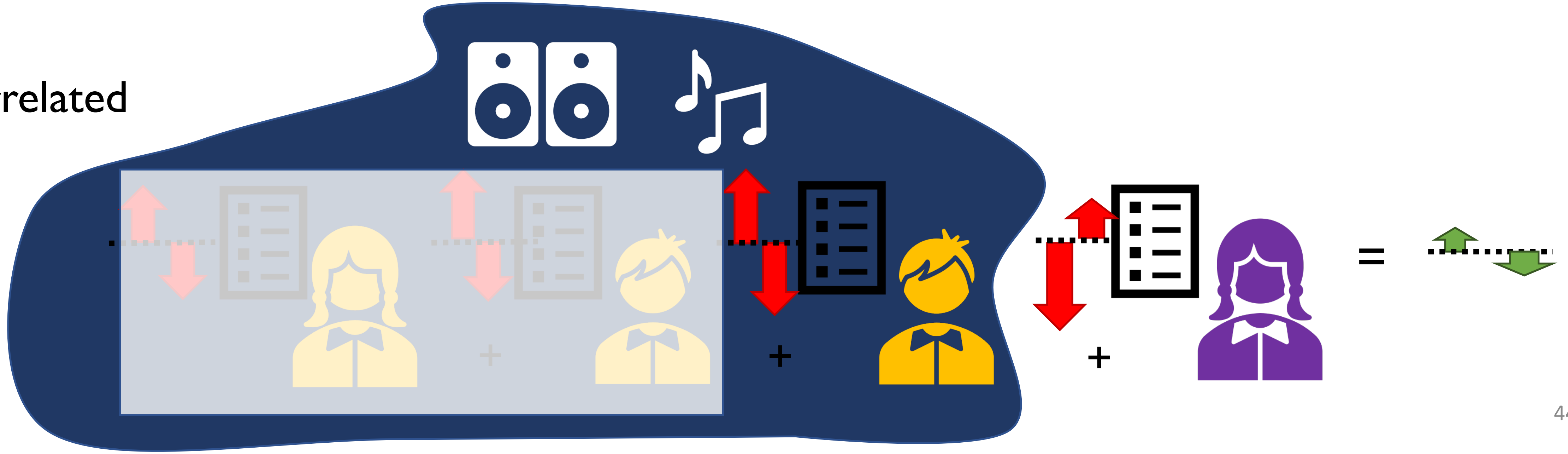


Variance won't decrease with averaging

How to create uncorrelated samples?

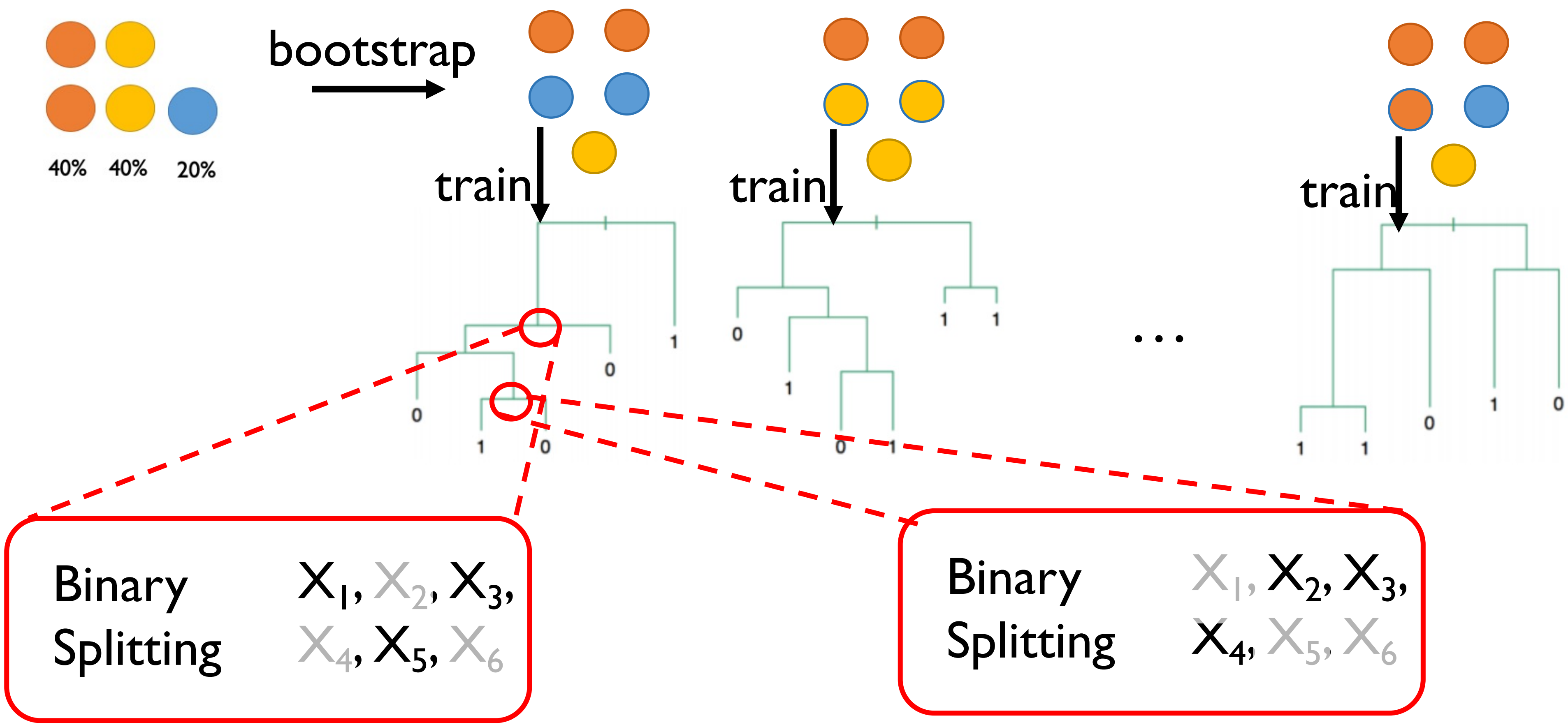
Subsample features at random!

Correlated

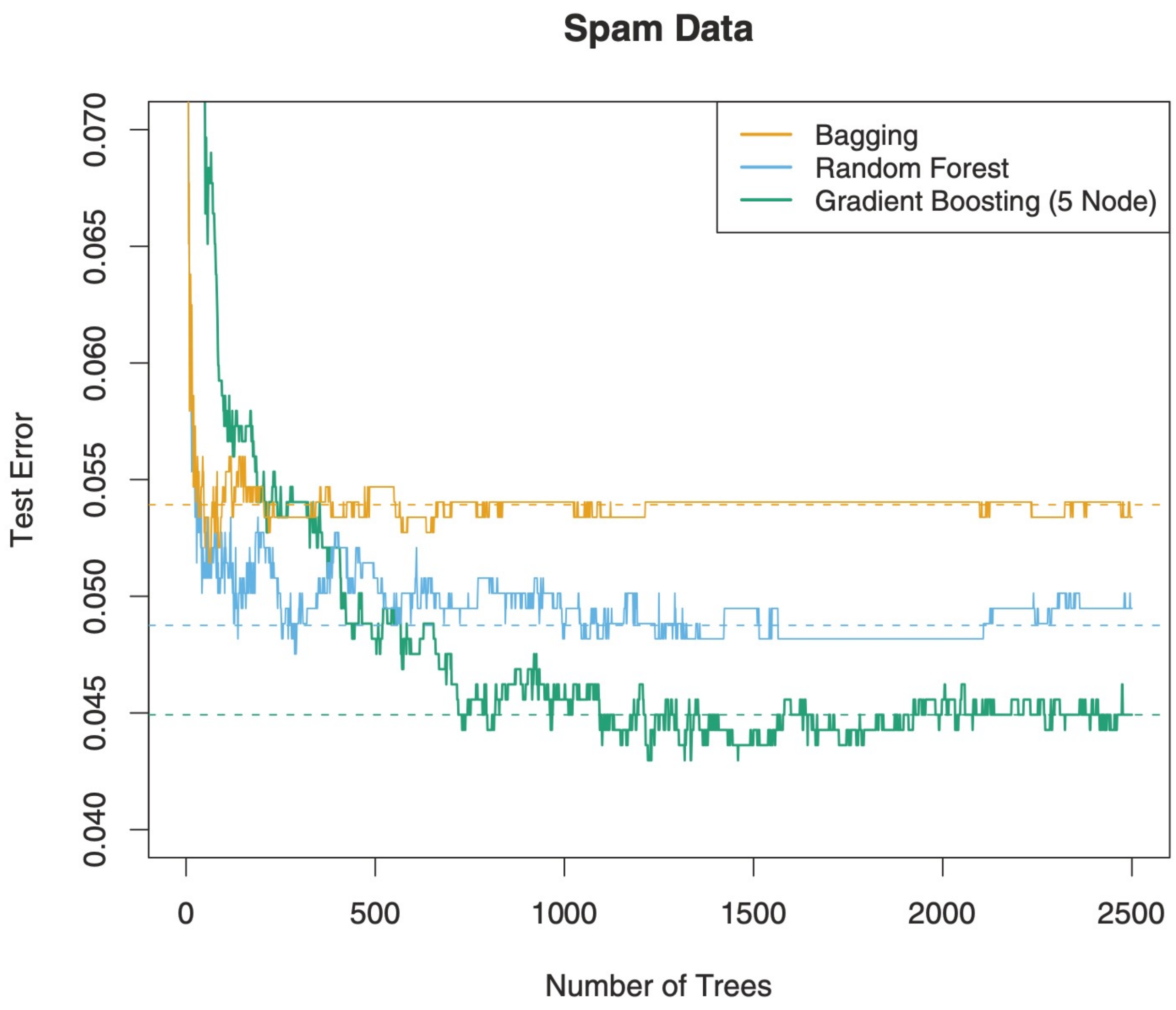


Random Forests

Subsample features at random! $selected \approx \sqrt{\frac{total}{features}}$



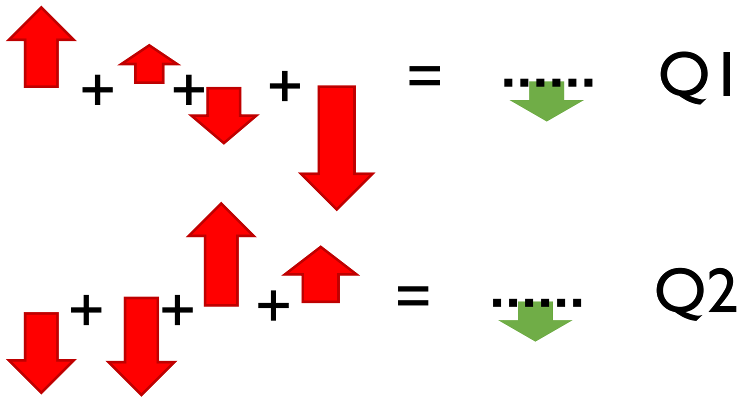
Random Forests vs. Bagging



ESL Fig. 15.1

How can we combine trees?

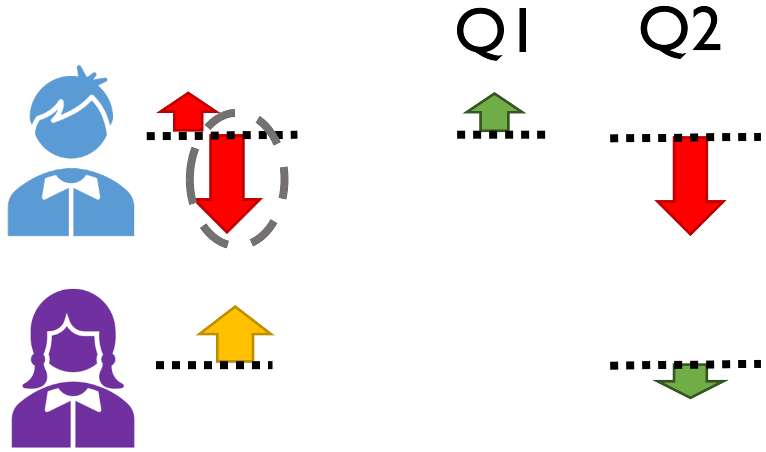
Average answers



Bagging

Hyperparameter = # trees

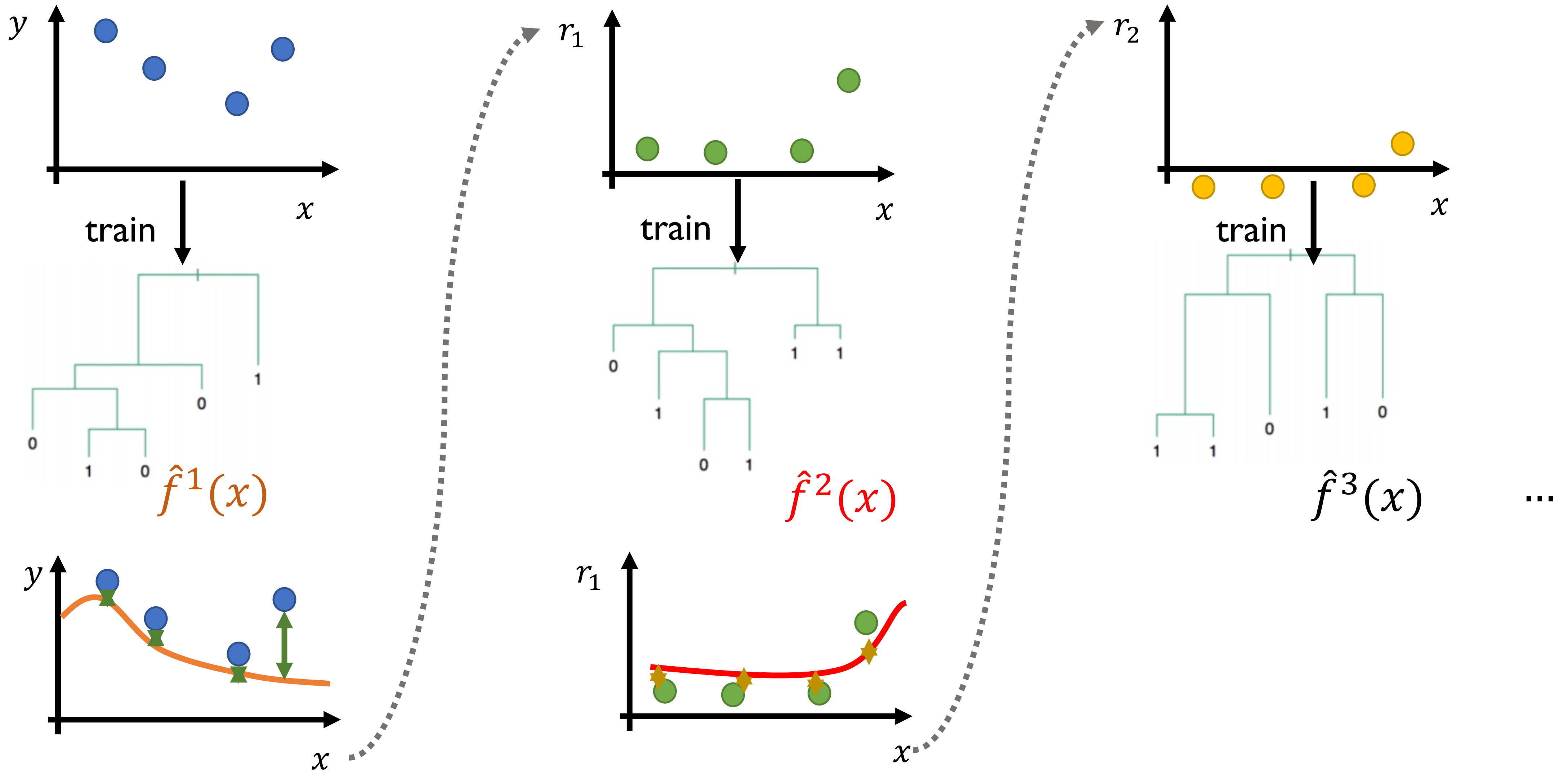
Iterative refinement



Boosting

Gradient Boosting Trees

Best off-the-shelf classifier

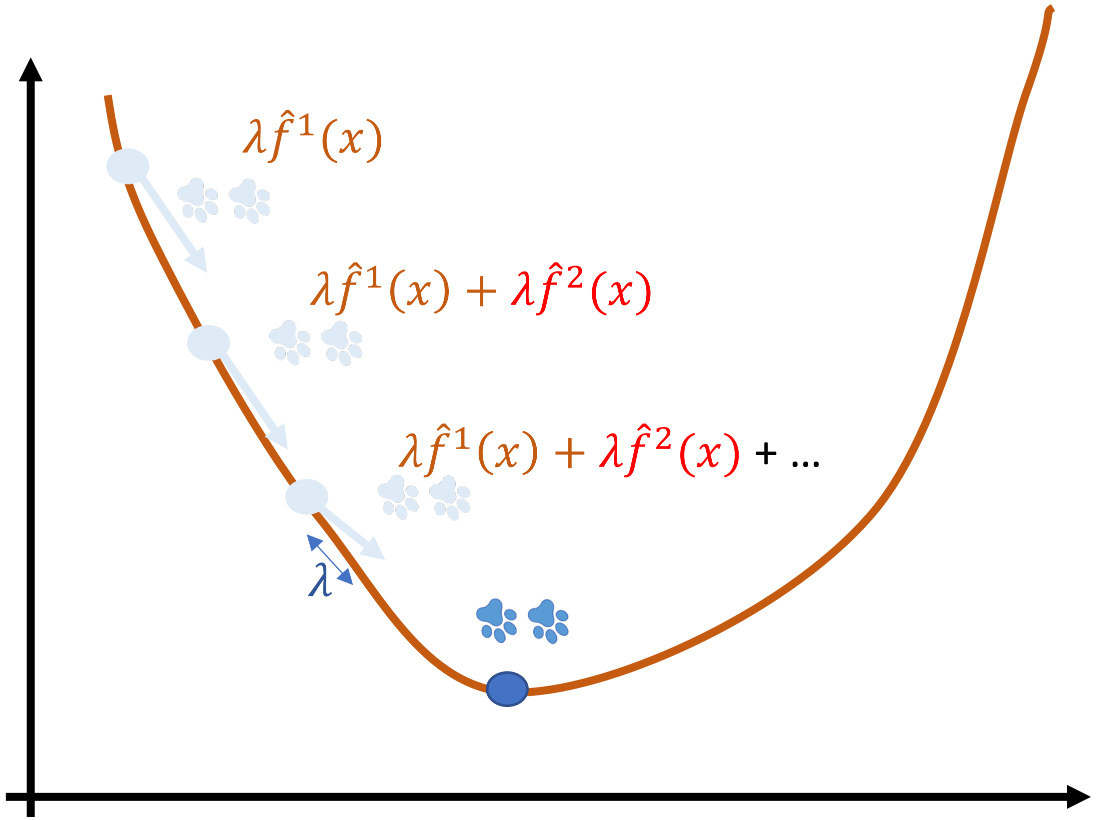


$r_1^{(i)} = y^{(i)} - \lambda \hat{f}^1(x)$

$r_2^{(i)} = r_1^{(i)} - \lambda \hat{f}^2(x)$

Gradient Boosting Trees

Best off-the-shelf classifier



$$\hat{f}_{\text{avg}}(x) = \lambda \sum_{b=1}^{B} \hat{f}^b(x)$$

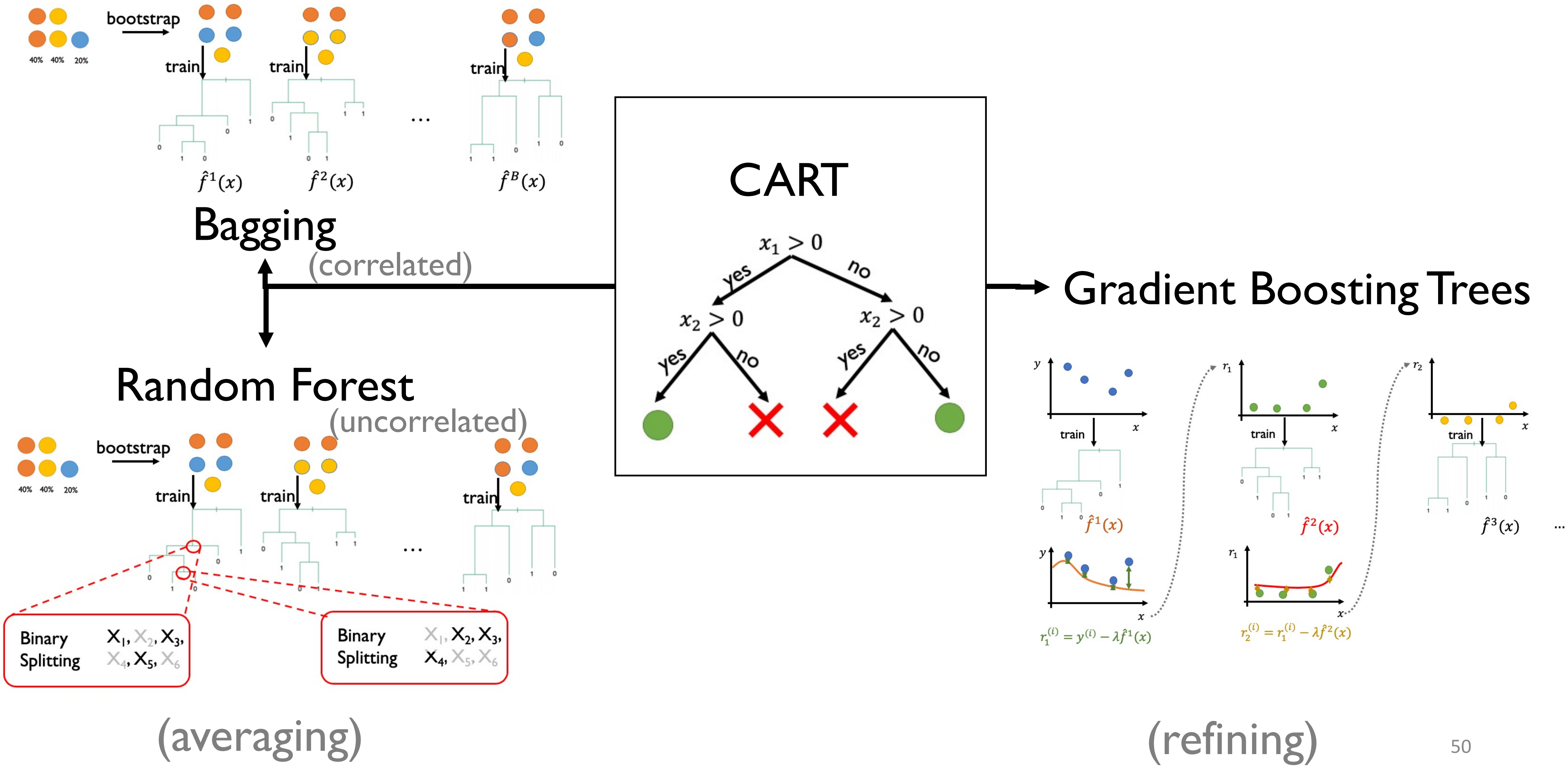
Hyperparameters

B = # trees ⟶ Overfitting

λ= Learning rate (small)

d = depth tree

Recap


bootstrap
40% 40% 20%
train
$\hat{f}^1(x)$ $\hat{f}^2(x)$ … $\hat{f}^B(x)$
Bagging
(correlated)
Random Forest
(uncorrelated)
Binary Splitting X1, X2, X3, X4, X5, X6
Binary Splitting X1, X2, X3, X4, X5, X6
(averaging)
CART
$x_1 > 0$
yes
no
$x_2 > 0$
Gradient Boosting Trees
$\hat{f}^1(x)$ $\hat{f}^2(x)$ $\hat{f}^3(x)$ …
$r_1^{(i)} = y^{(i)} - \lambda \hat{f}^1(x)$
$r_2^{(i)} = r_1^{(i)} - \lambda \hat{f}^2(x)$
(refining)

Coming up … (Last class)

Practical Example:

How to choose the **best** method?

Intro to Neural Networks
& Deep Learning