# XGBoost

## Step by step - intro2ml, trees, ensembles and finally XGBoost

**Marco Zamboni 21/04/2021**

# Who am I and contacts

- Student MSc in Data Science @ ETH Zurich
- Junior Software and Data Engineer @ 42matters
- telegram.me/ZamboniM
- https://www.linkedin.com/in/marco-zamboni

# Intro to Machine Learning

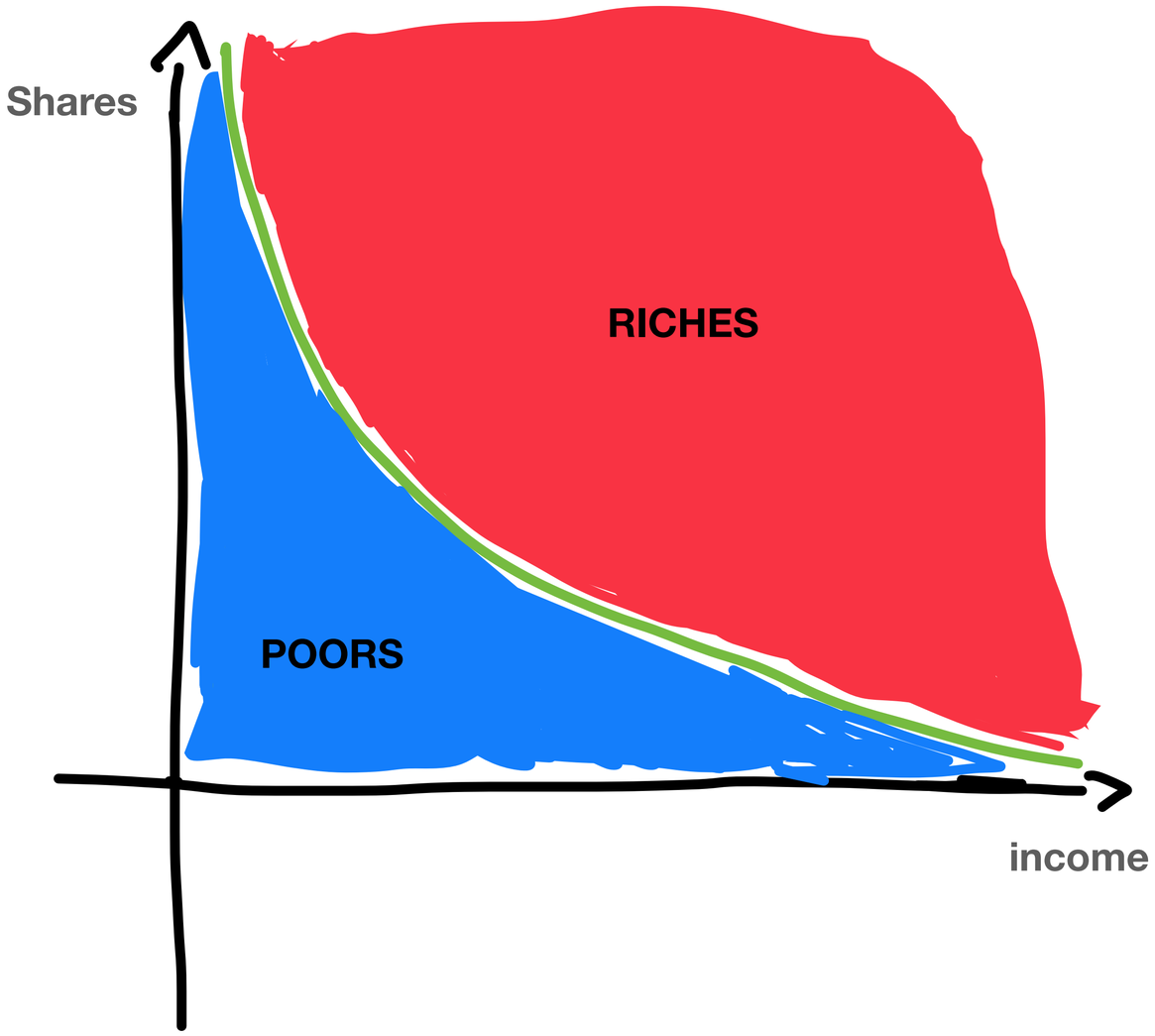
# Classification

Which are

- Bananas?
- Apples?
- Oranges?

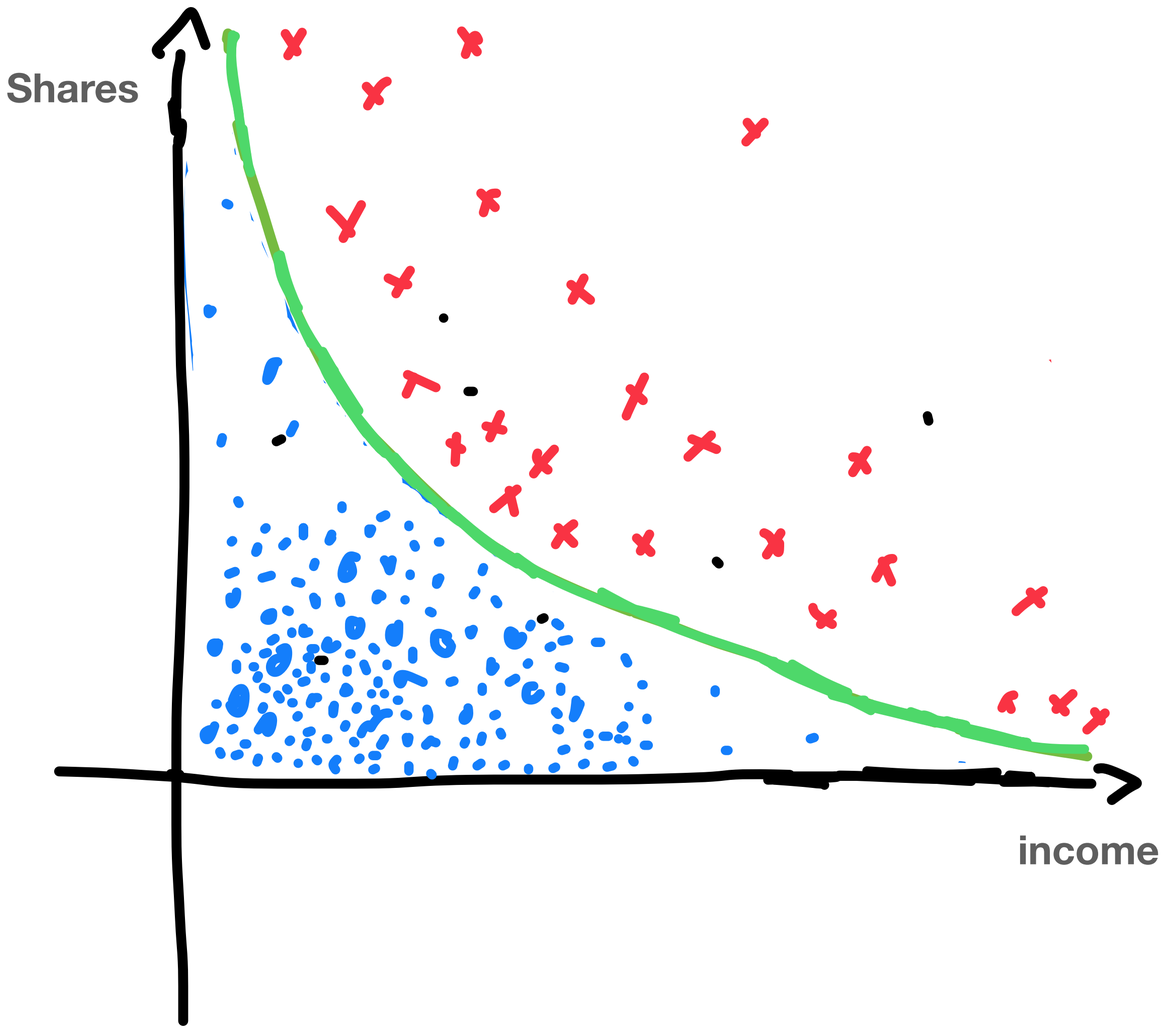
# Classification

## Rich vs Poor (priors)



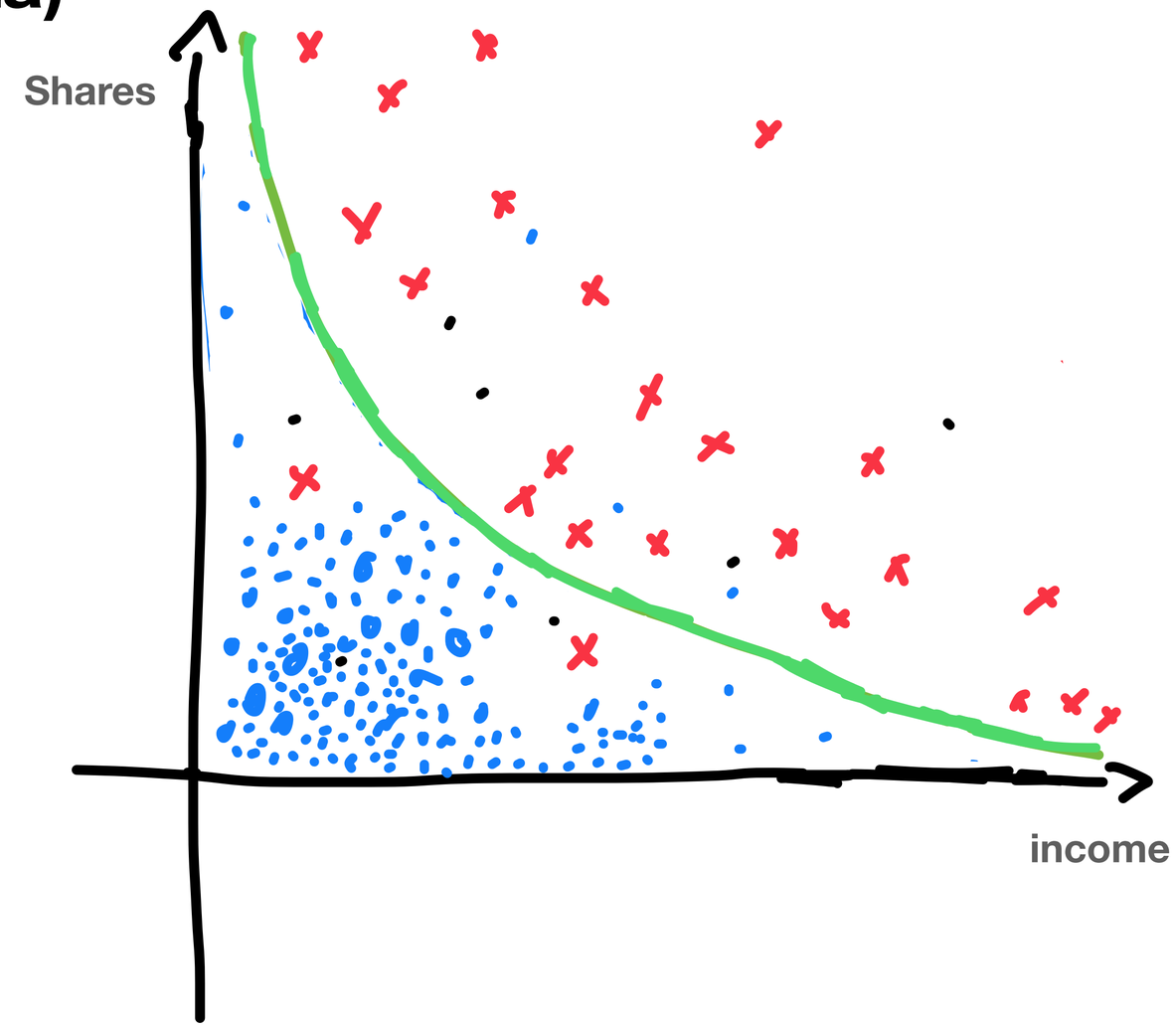
# Classification

## Rich vs Poor (Data)



# Classification

## Rich vs Poor (Real Data)



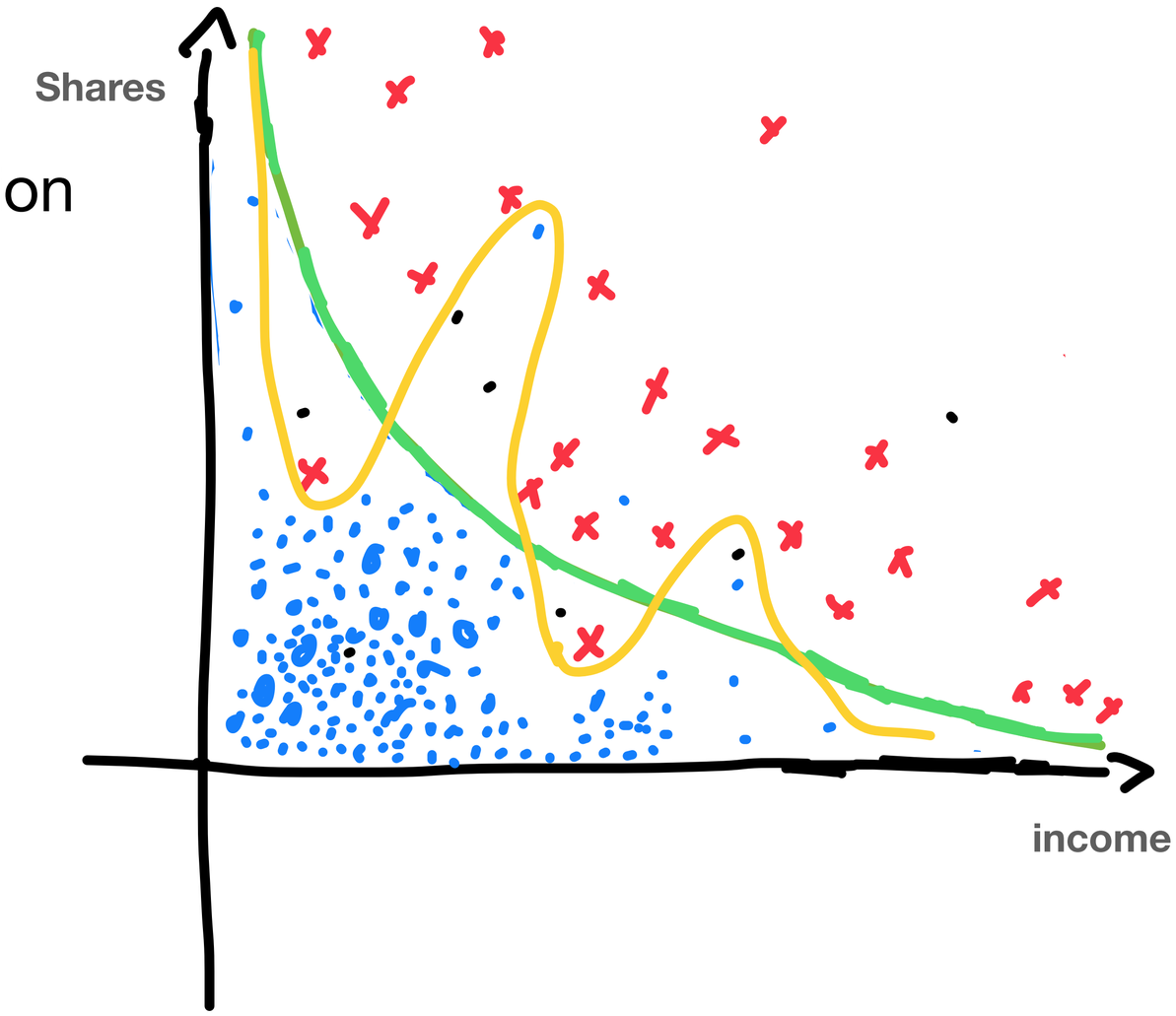
# Classification

## Rich vs Poor

Minimise an objective function

$$obj(M, \theta) = Loss(M, \theta)$$

$$Loss(M, \theta) = I(Y \neq M_\theta(X))$$



# Classification

## Rich vs Poor

Minimize an objective function

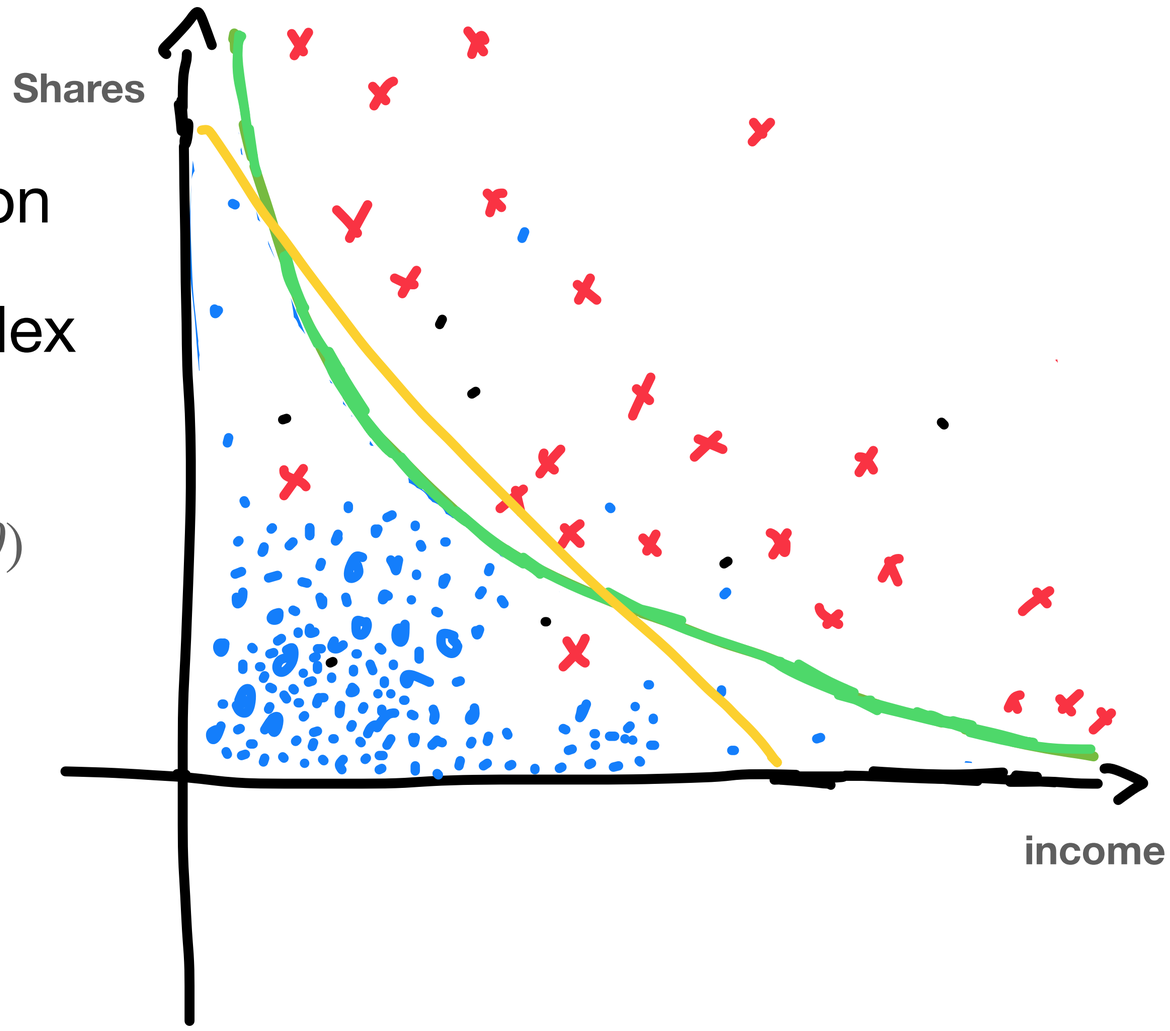
Regularize to penalize complex solutions!!

$$obj(M, \theta) = Loss(M, \theta) \; + \; \Omega(M, \theta)$$

$$Loss(M, \theta) = I(Y \neq M_\theta(X))$$

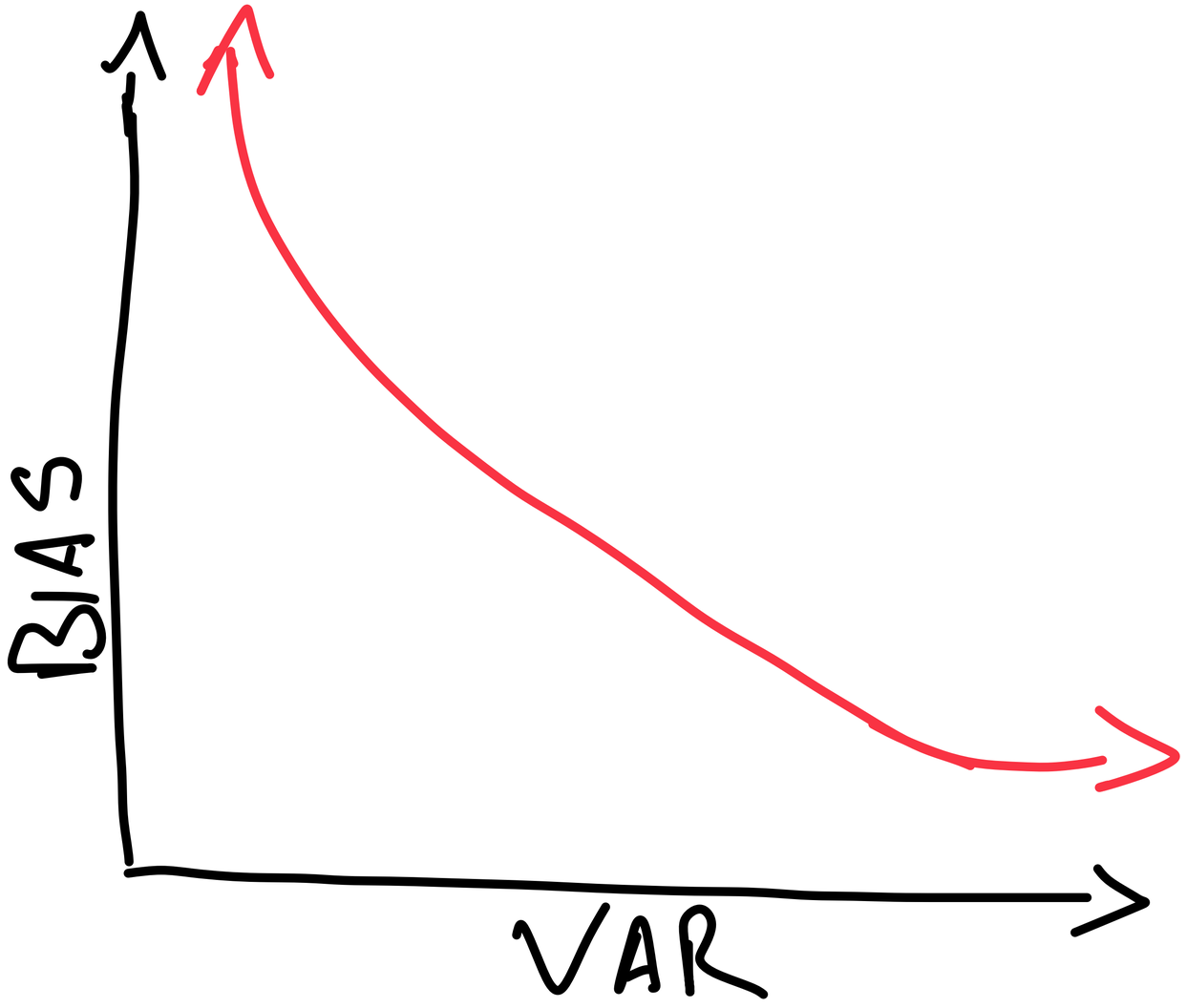
$\Omega(M, \theta) =$

- Smoothness
- Polynomial grade
- Parameters magnitude
- ...



# Bias Variance tradeoff

- High bias leads to underfitting
- Low bias leads to overfitting



# Decision Trees

# Classification

Which are

- Bananas?
- Apples?
- Oranges?

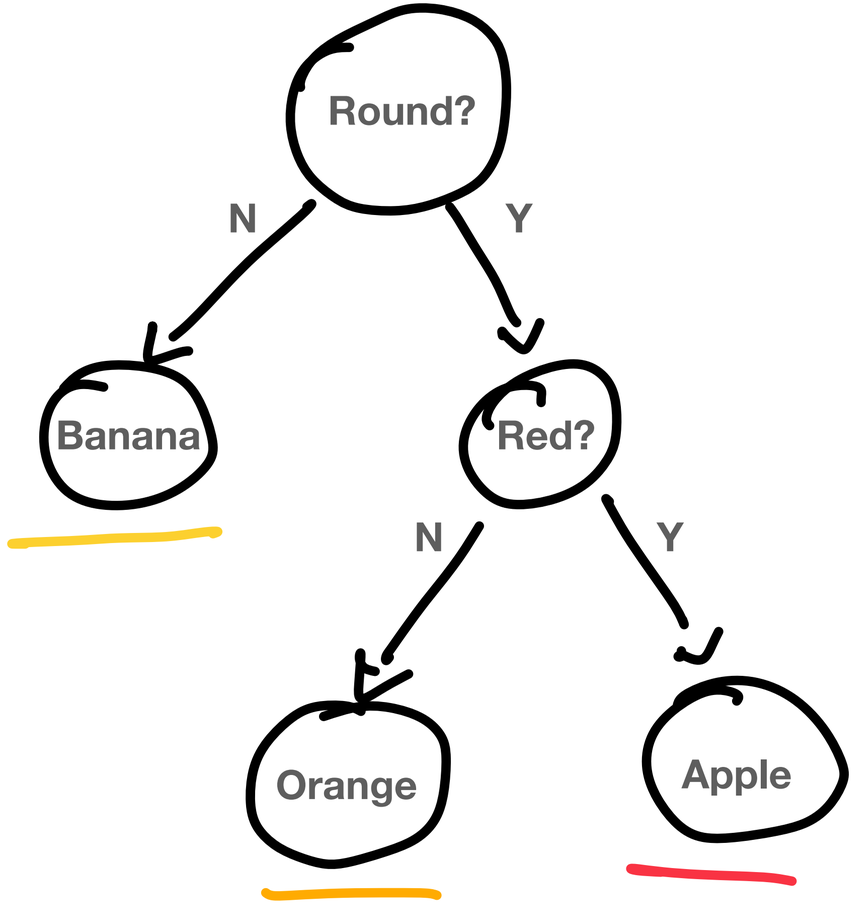
# Classification

## Code

```
def classifier(fruit):
    if is_round(fruit):
        if is_red(fruit):
            return "Apple"
        else:
            return "Orange"
    else:
        return "Banana"
```

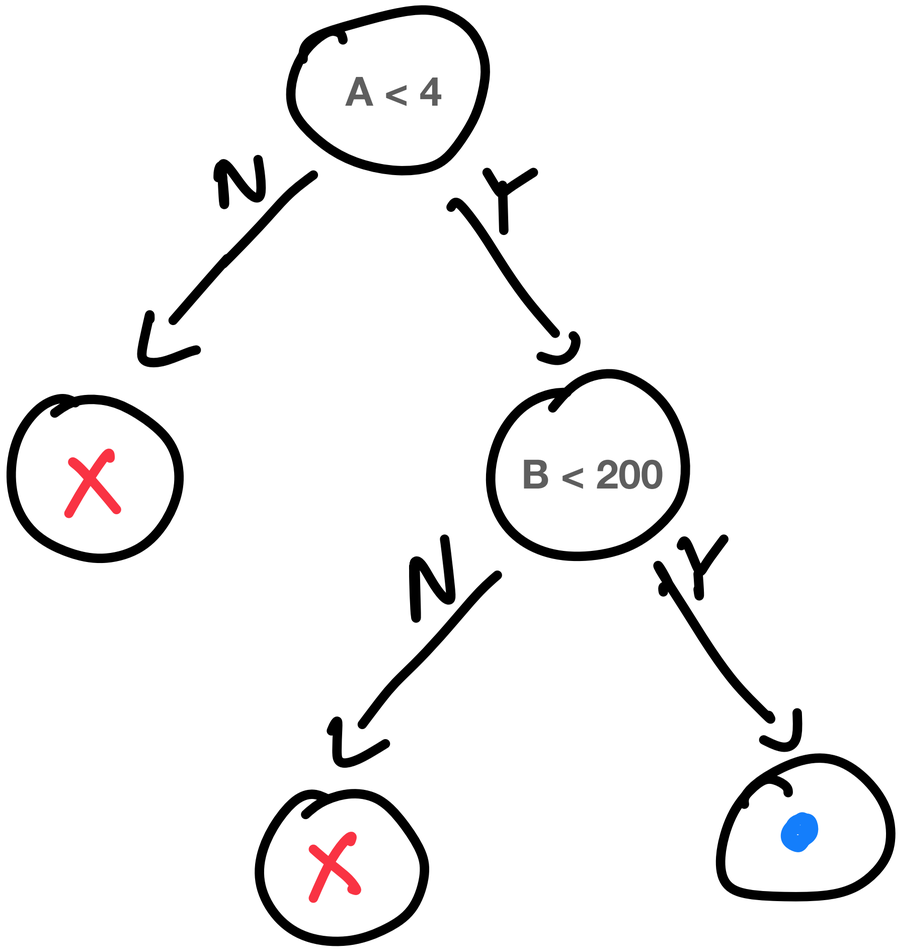
# Classification

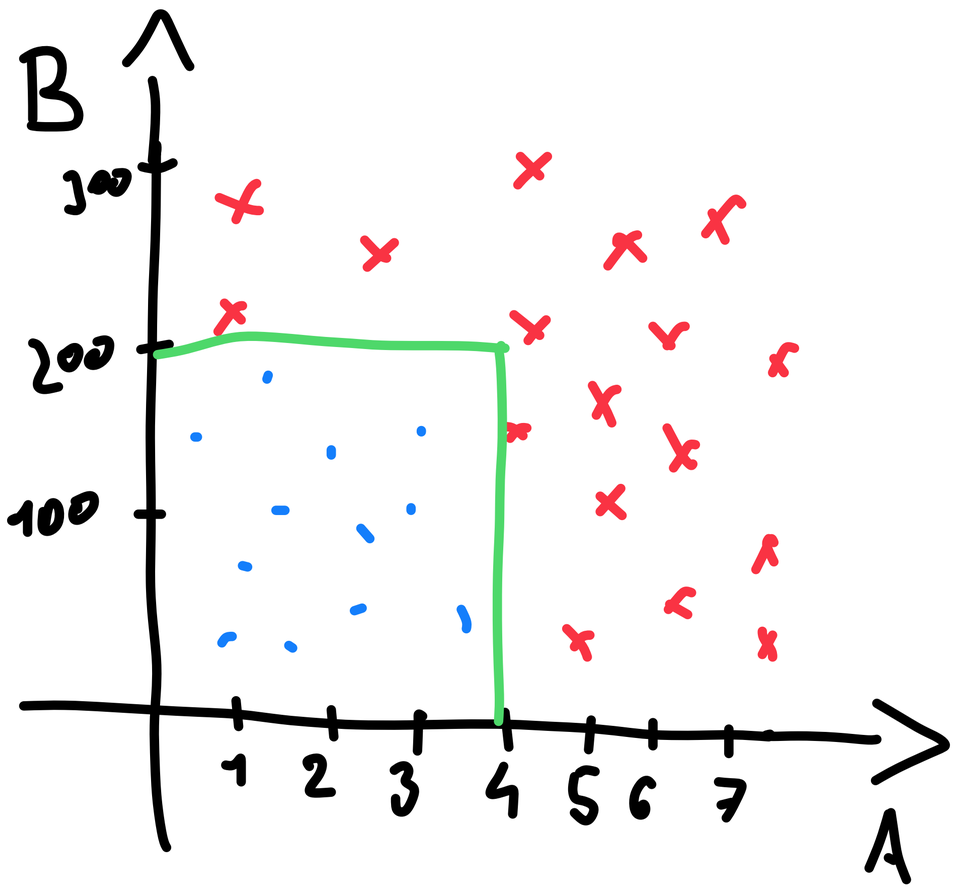
## Decision tree



# Decision trees

## Less abstract

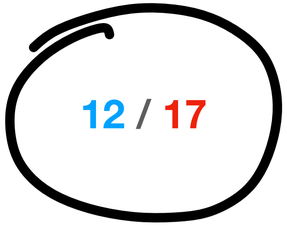


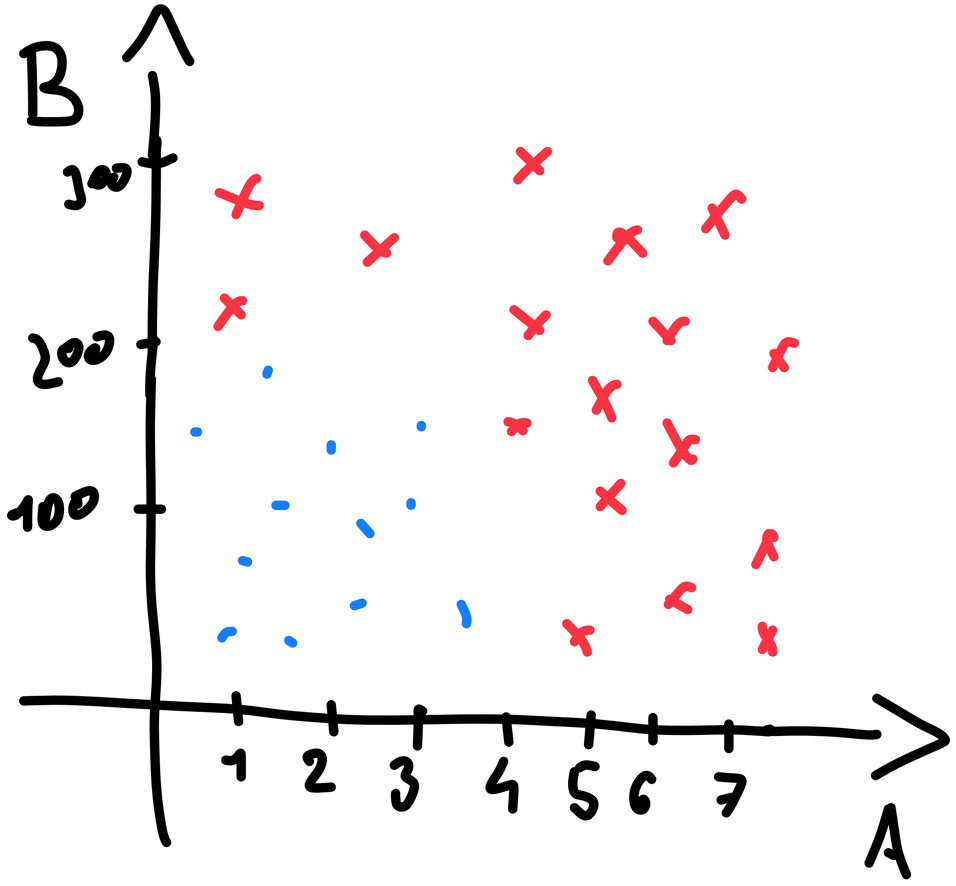



$$H(T) = -\sum_i p(T_i) log(p(T_i))$$

# Learning DTs

## Entropy



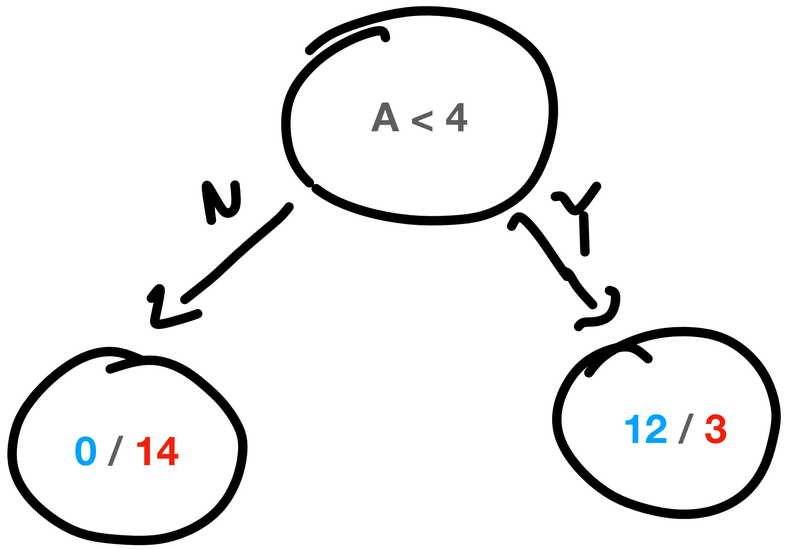




$$H(T) = -\,[(12/29)log(12/29) \;+\; (17/29)log(17/29)] \approx 0.68$$

$$H(T) = -\sum_i p(T_i) log(p(T_i))$$

# Learning DTs

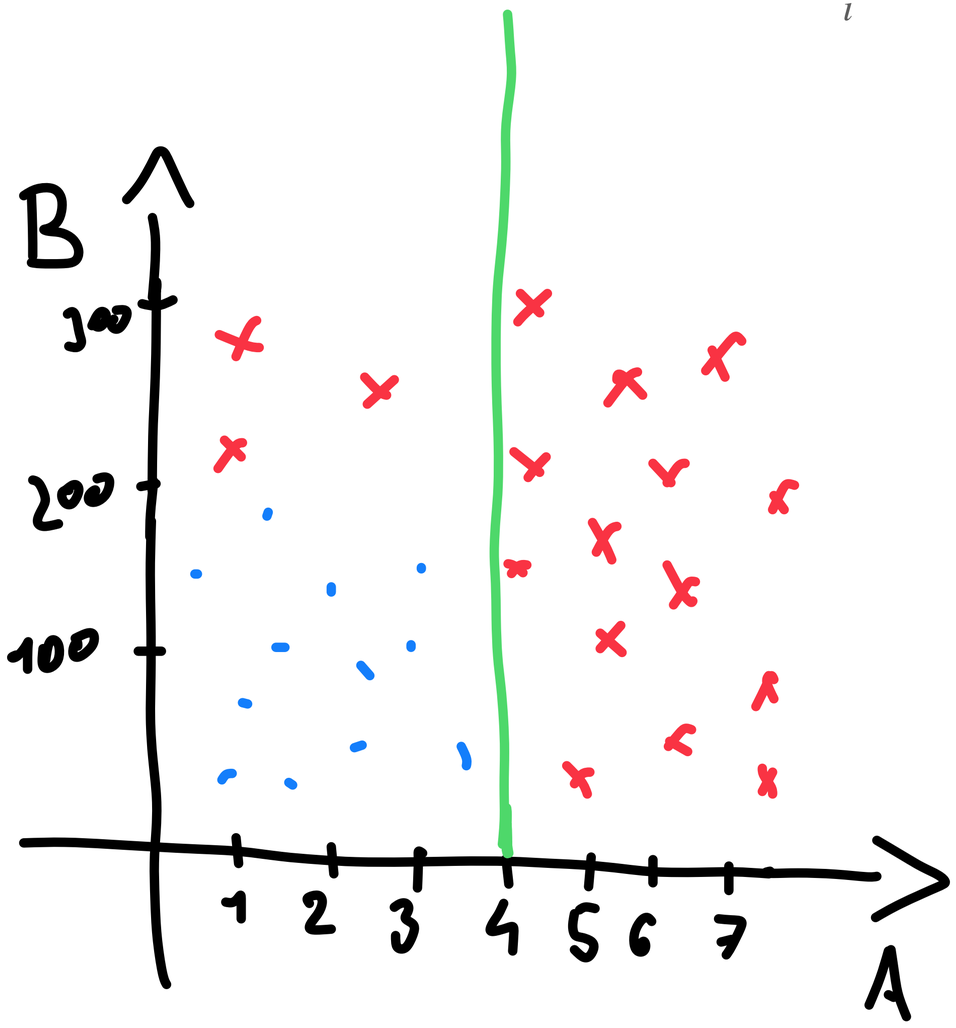
## Entropy




$H = 0$

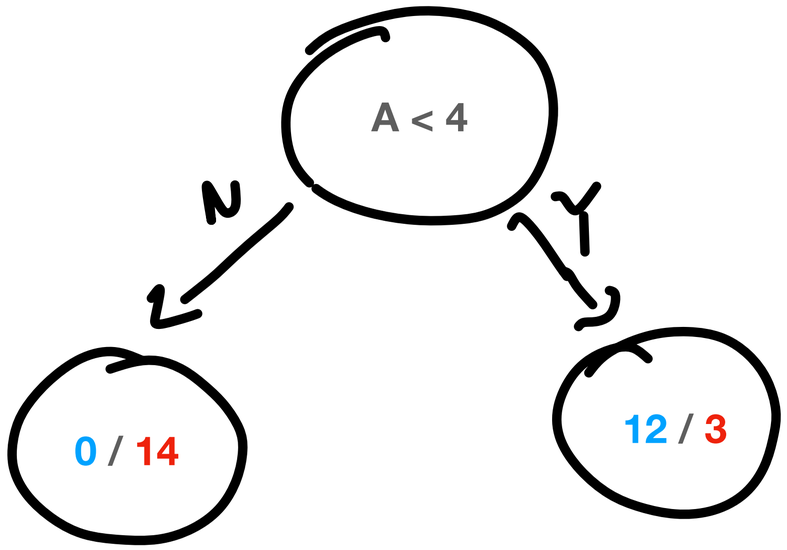
$H \approx 0.50$

$H(T, s) \approx 0.25$

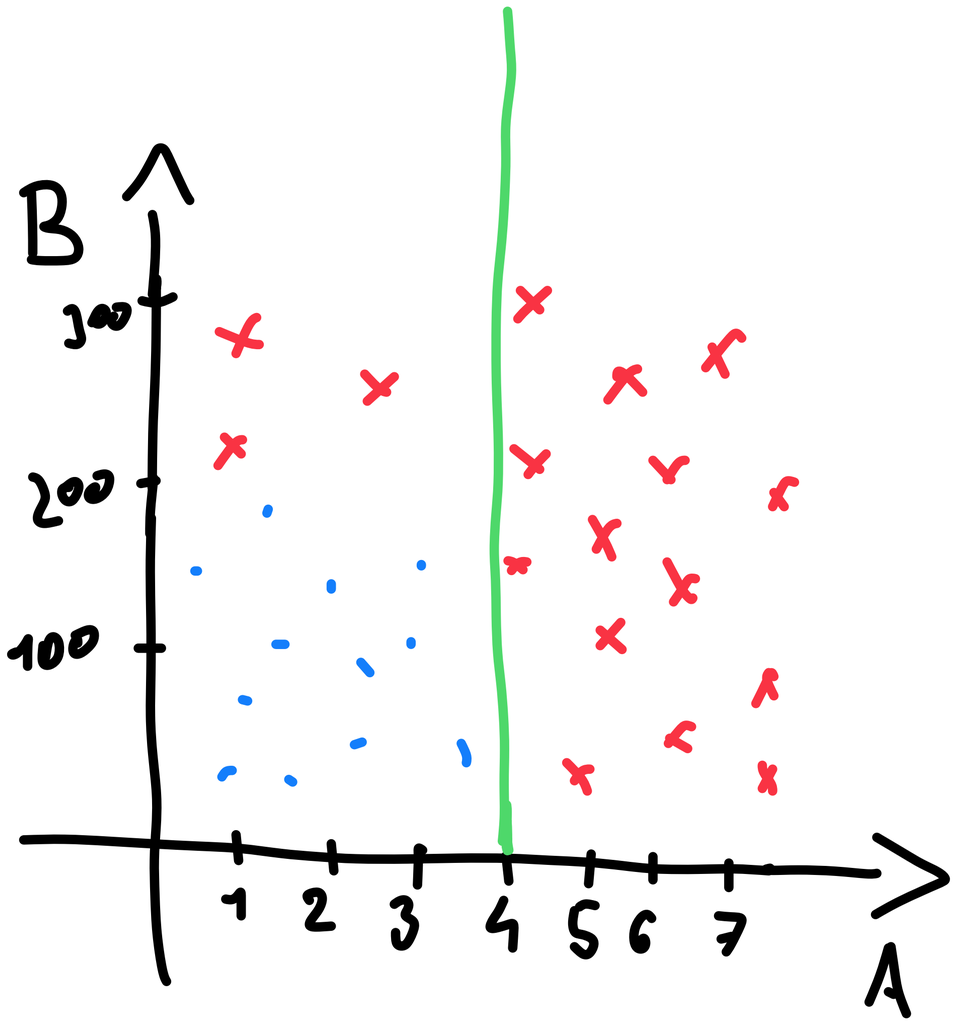


# Learning DTs

## Information gain

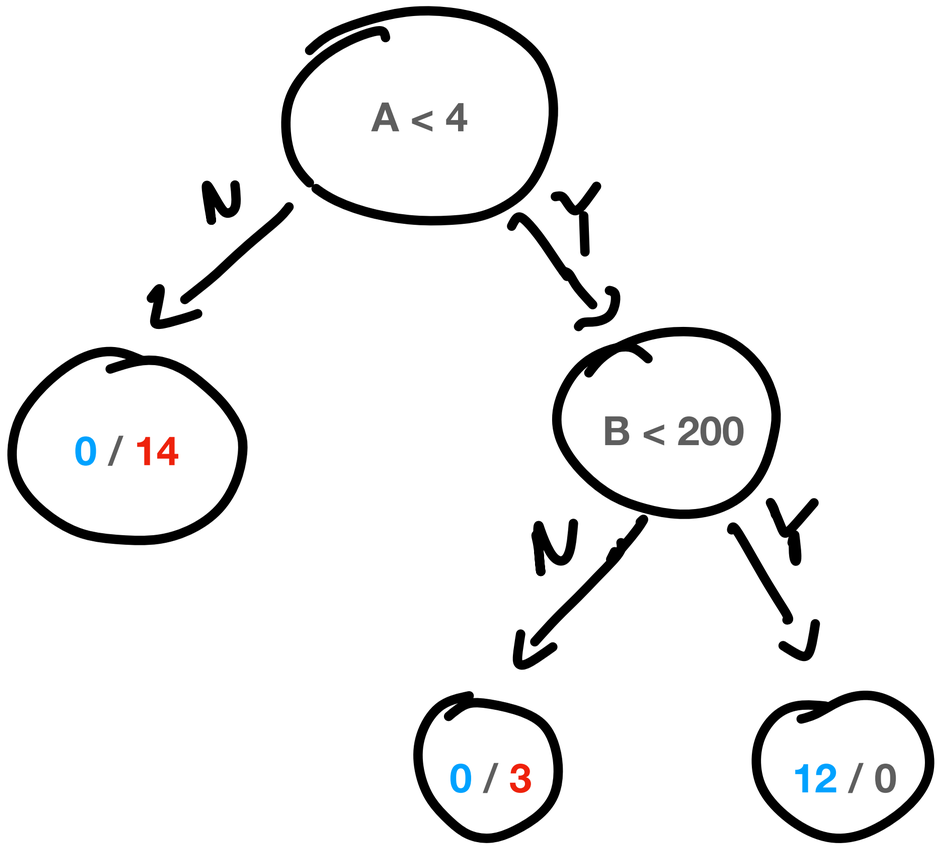



$$Ig(T, s) = H(T) - H(T, s) \approx 0.43$$

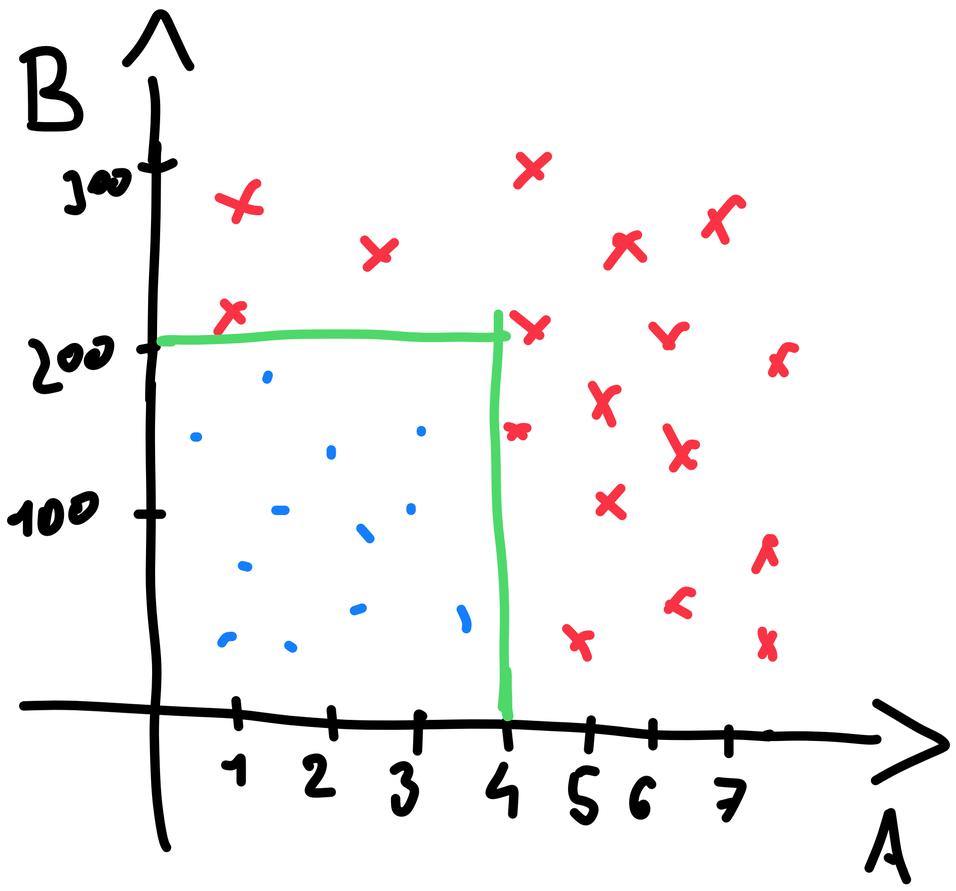


# Learning DTs

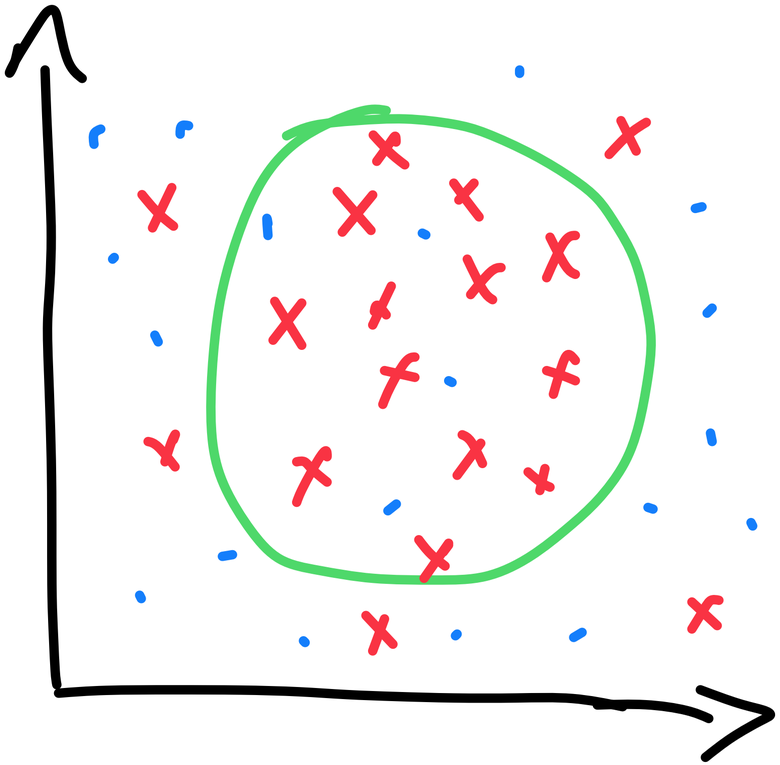
## Information gain




$Ig(T, s) \approx 0.50$



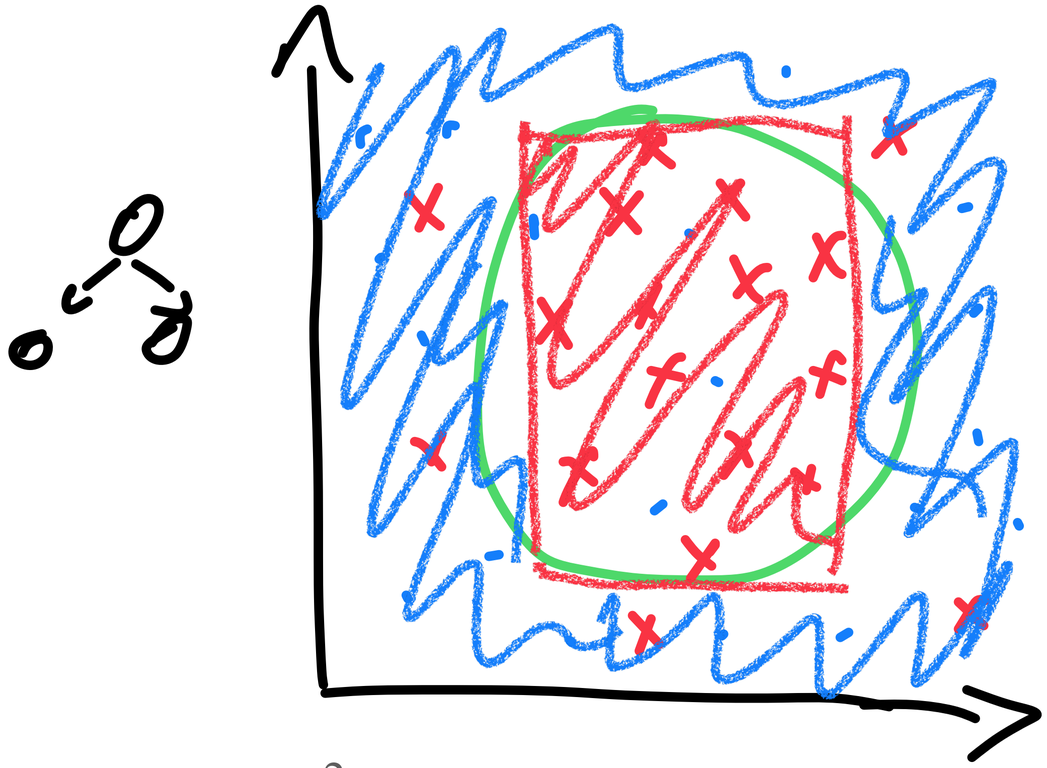
# Bias and variance in trees

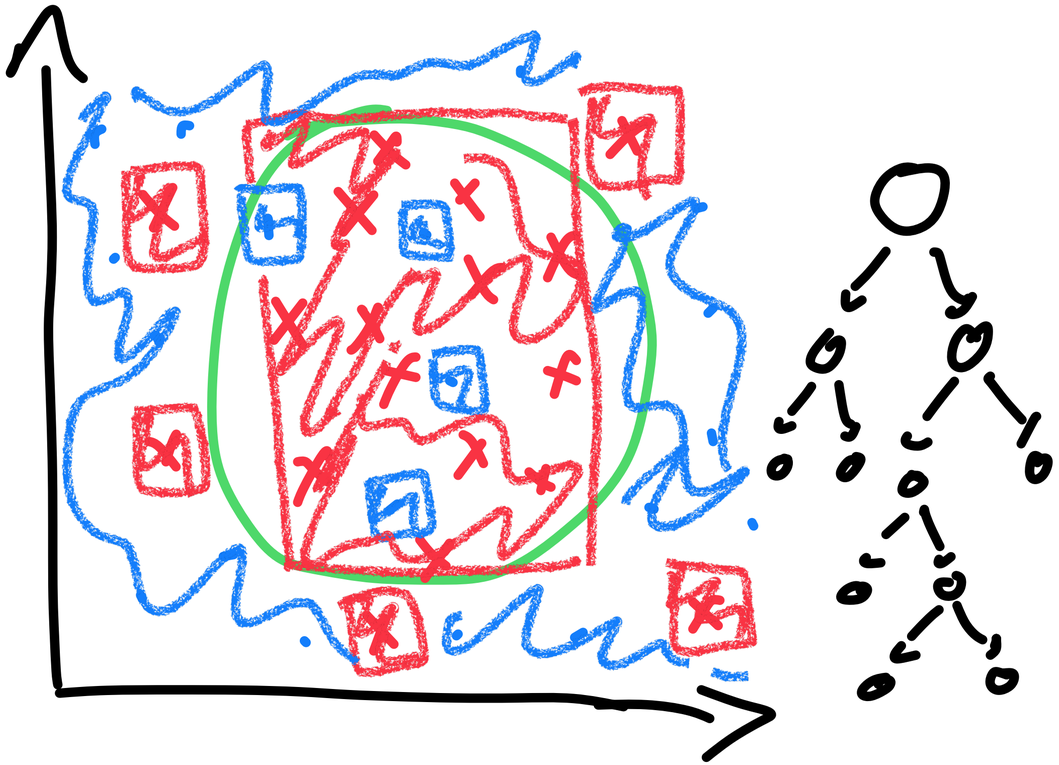


$X \sim Unif([0,1]^2)$

$Pr(Y = \times \mid X = x) = \begin{cases} 0.8 & \textbf{if inside circle} \\ 0.2 & \textbf{otherwise} \end{cases}$

# Bias and variance in trees



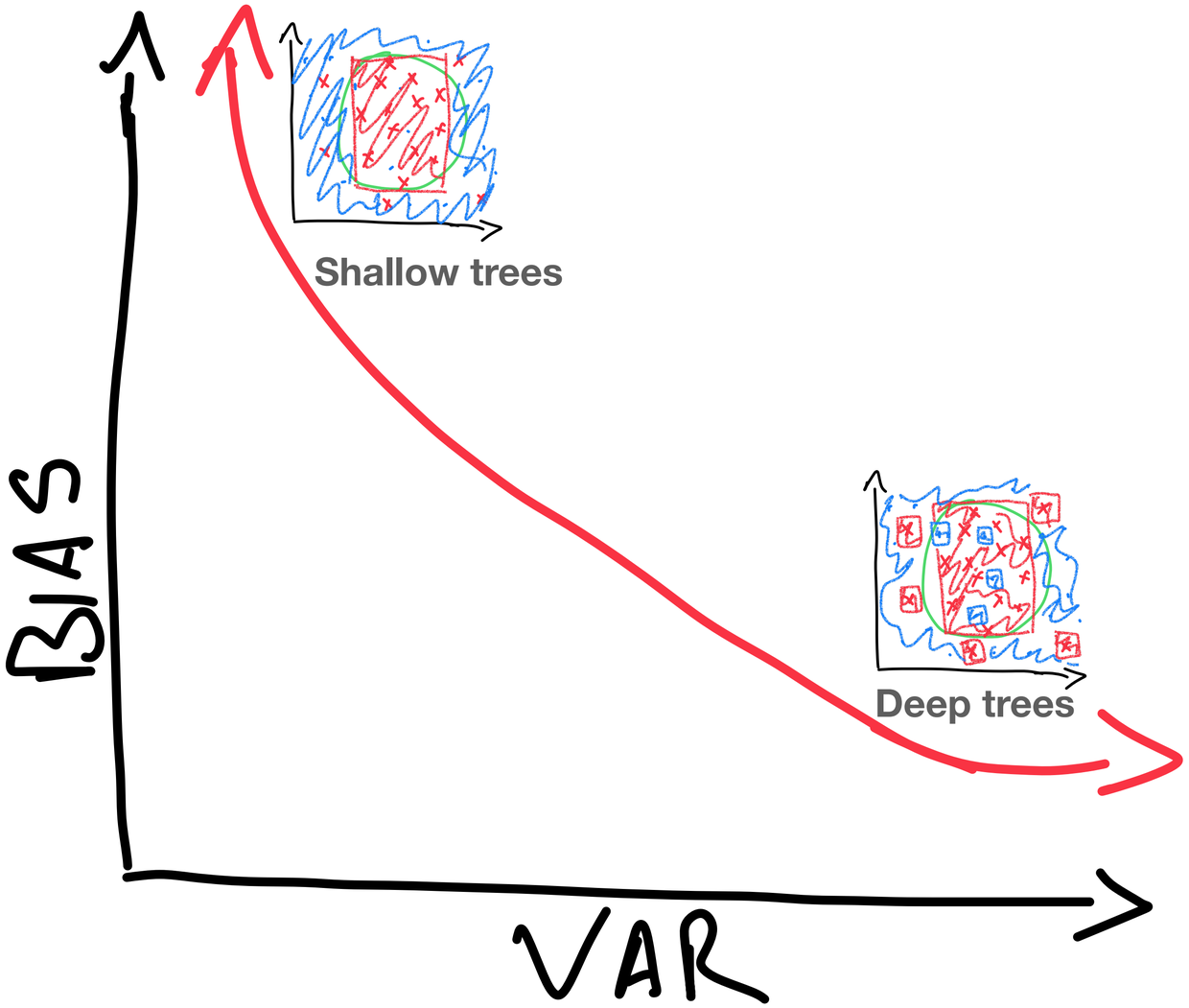


$X \sim Unif([0,1]^2)$

$Pr(Y = \times \mid X = x) = \begin{cases} 0.8 & \textbf{if inside circle} \\ 0.2 & \textbf{otherwise} \end{cases}$

# Bias Variance tradeoff

- High bias leads to underfitting
- Low bias leads to overfitting

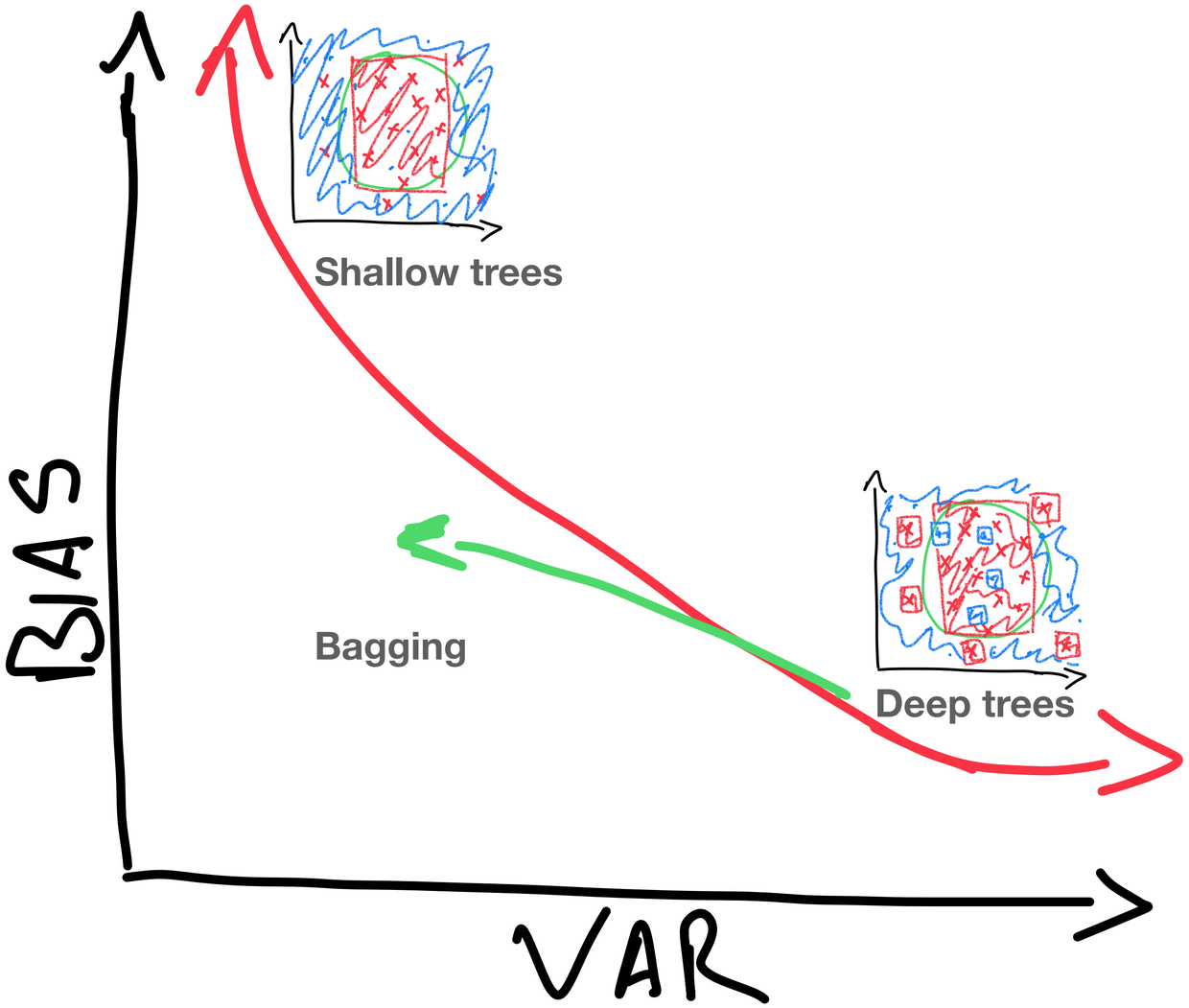


# Ensembles

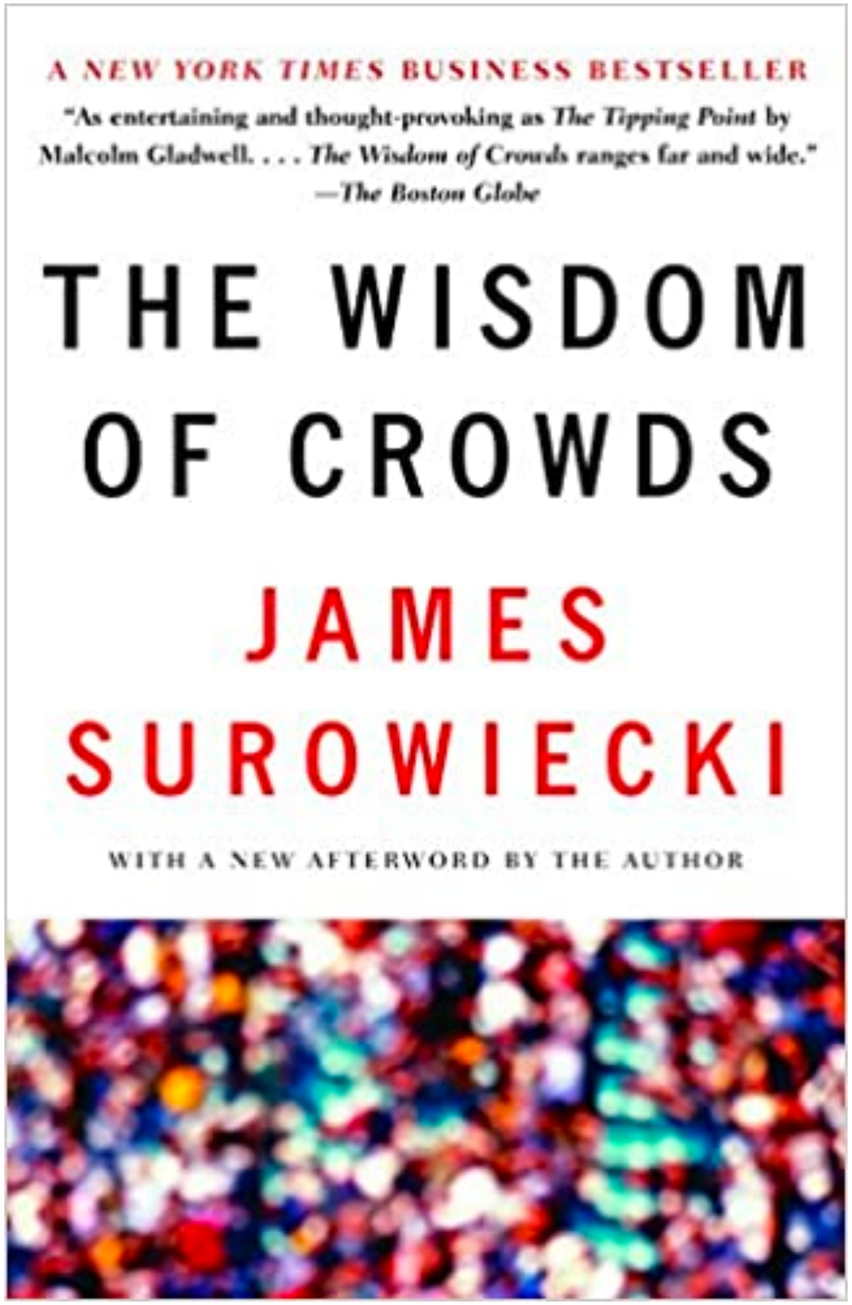
# Bias Variance tradeoff

## Bagging

- High bias leads to underfitting
- Low bias leads to overfitting



# The wisdom of crowds



# The wisdom of crowds

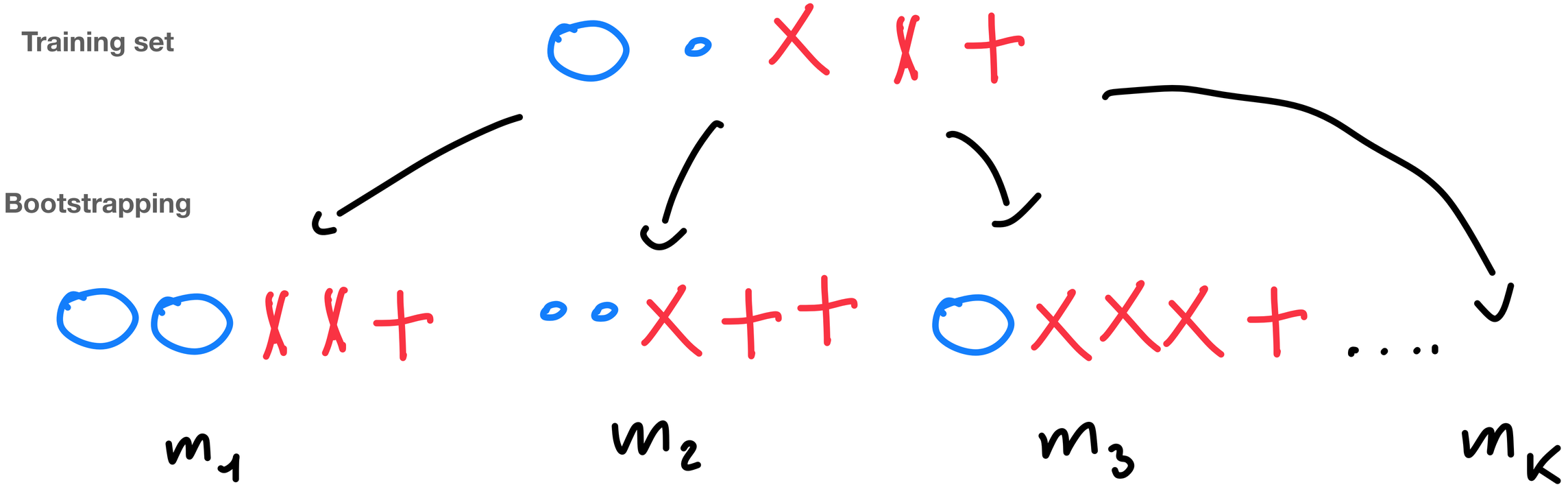
## Requirements

- Diversity
- Independence
- Be better than random
- …
- …

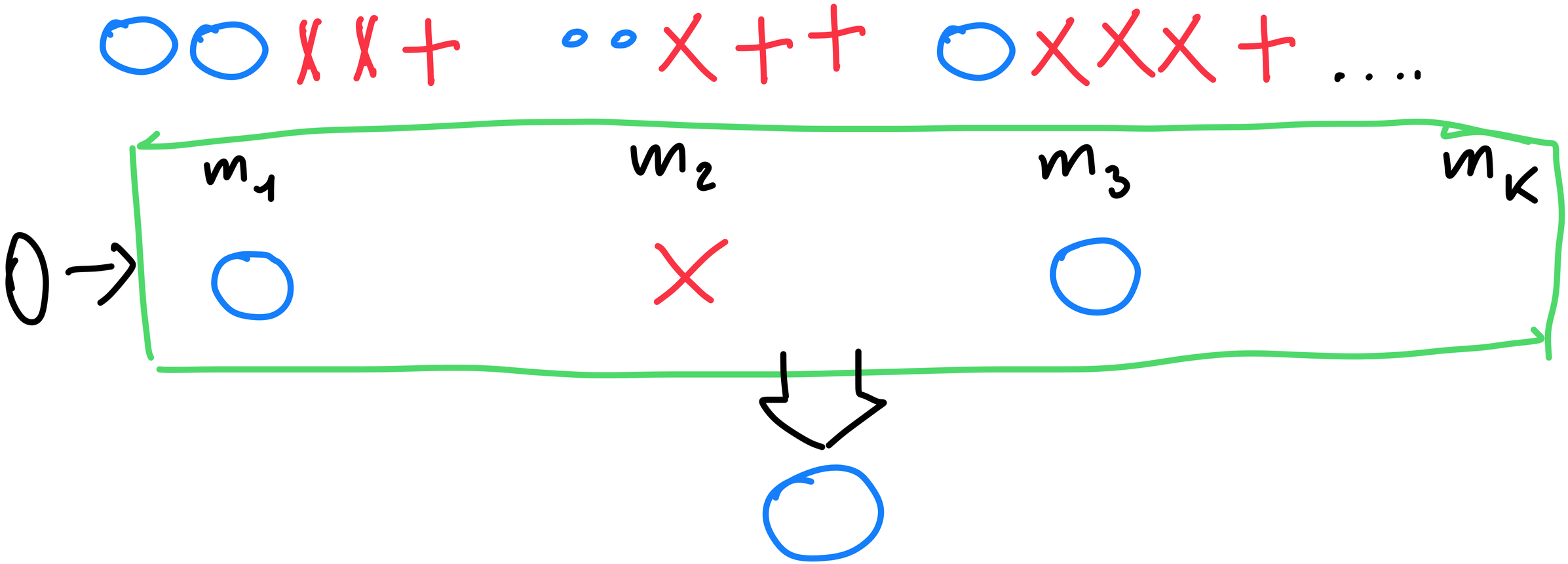
$$\mathbb{E}[Agg(M)] \leq min_{m \in M} \mathbb{E}[m]$$

$$Agg = \begin{cases} avg \\ consensus \\ \dots \end{cases}$$

# Bagging



# Bagging



# Bagging
## Example

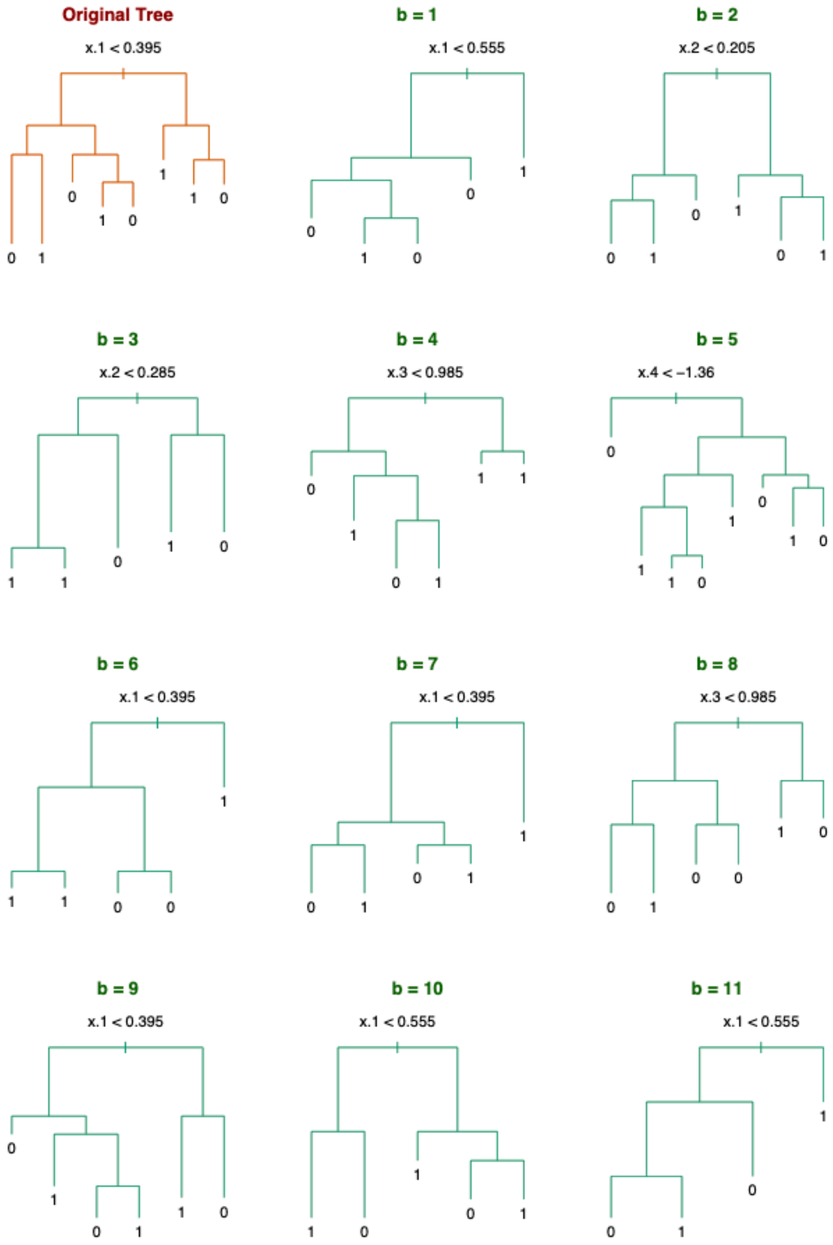
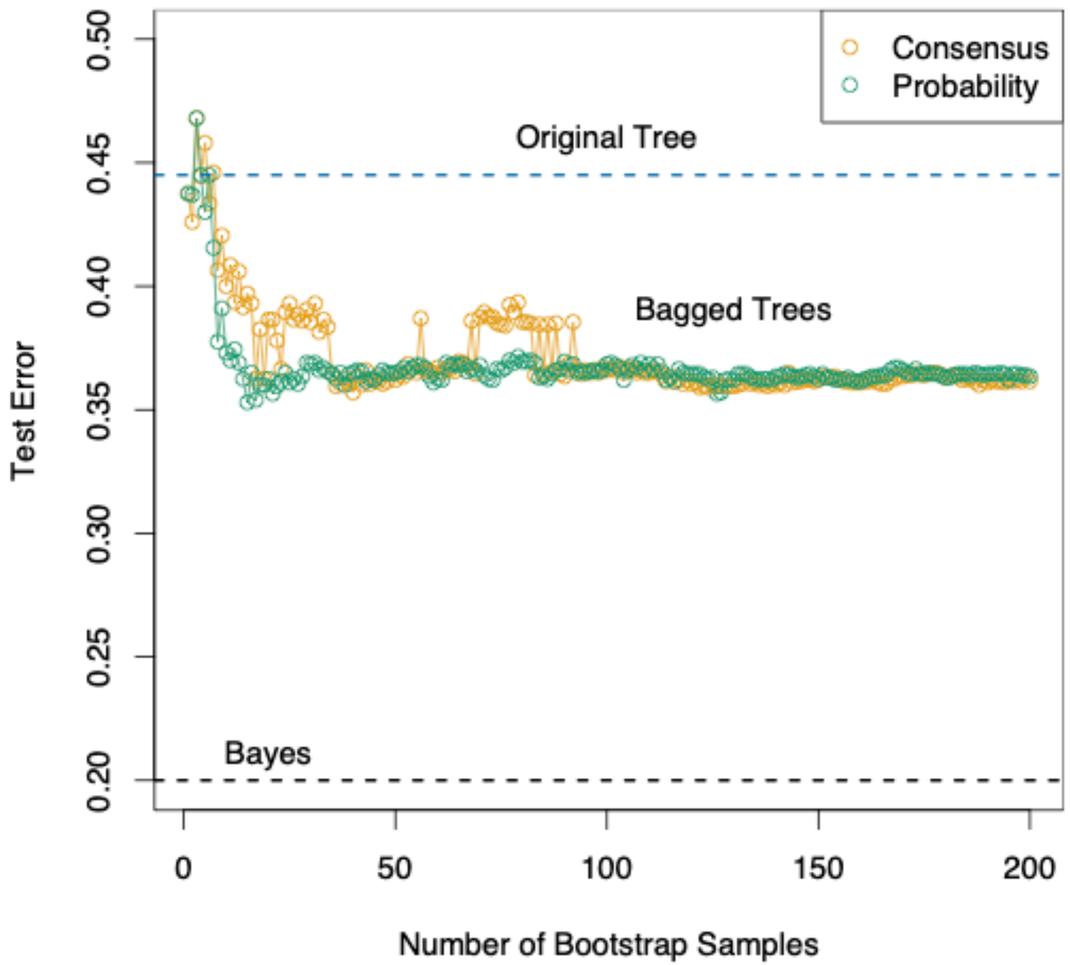



**FIGURE 8.9.** *Bagging trees on simulated dataset. The top left panel shows the original tree. Eleven trees grown on bootstrap samples are shown. For each tree, the top split is annotated.*

Elements of statistical learning, Hastie et al

# Bagging
## Example




Elements of statistical learning, Hastie et al

# State of the art

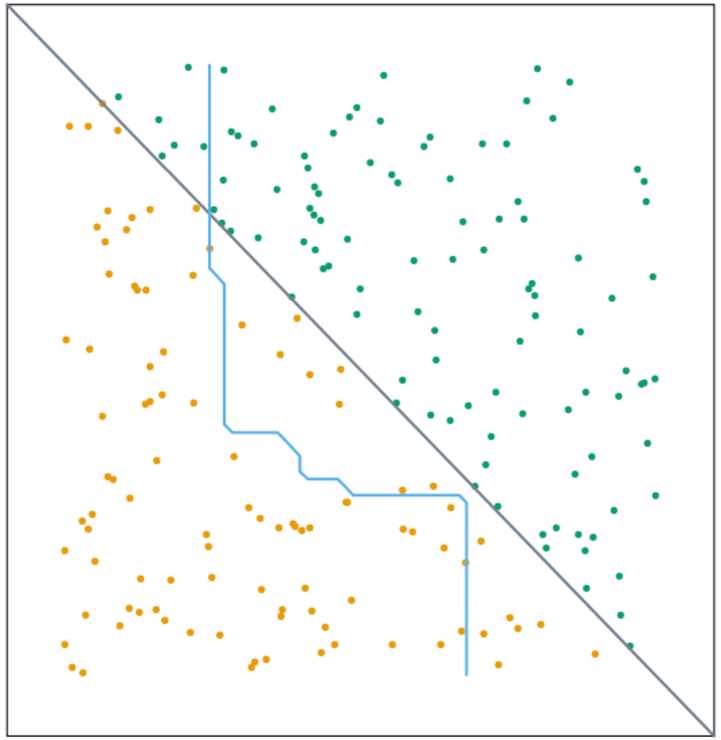
## (I mean, in 90s)

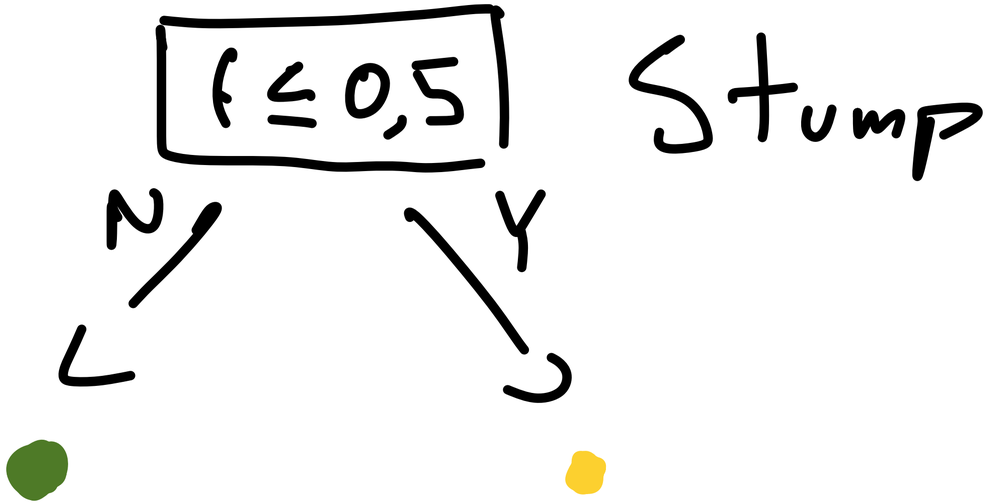
| Algorithm | Average rank |
|---|---|
| **Bagging trees** | 1.8 |
| **SVM (rbf kernel)** | 8.0 |
| **KNN** | 8.5 |
| **C4.5** | 9.3 |
| **Quadratic discriminant analysis** | 10.8 |
| **Neural Networks** | 12.8 |

**StatLog: An evaluation of machine learning and statistical algorithms, Henery and Taylor (1992)**
**Bagging predictors, Breiman (1996)**

# Diversity

Bagged Decision Rule






Elements of statistical learning, Hastie et al

Not independent!!!

# Randomized feature selection

- Start with bagging
- Each model is trained only on a **random subset of features**
- Low correlation between models
- Have the models be trees and have you ever heard of Random Forests?

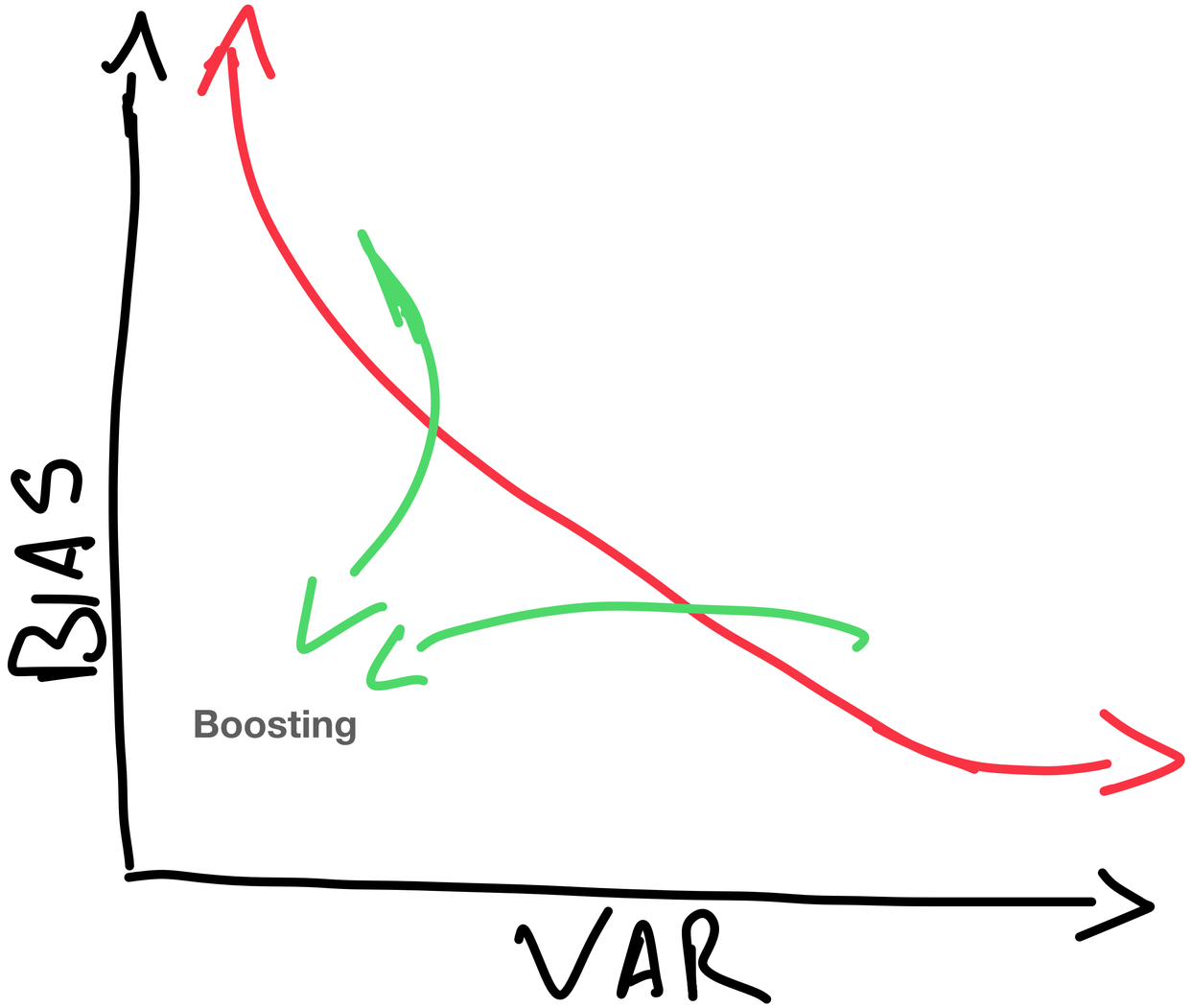
# Interesting facts

- Randomized feature selection induces implicit regularization (The Implicit Regularization of Ordinary Least Squares Ensembles, LeJune, 2020)
- Ensembles + Randomized feature selection is equivalent to stochastic gradient descent + dropout

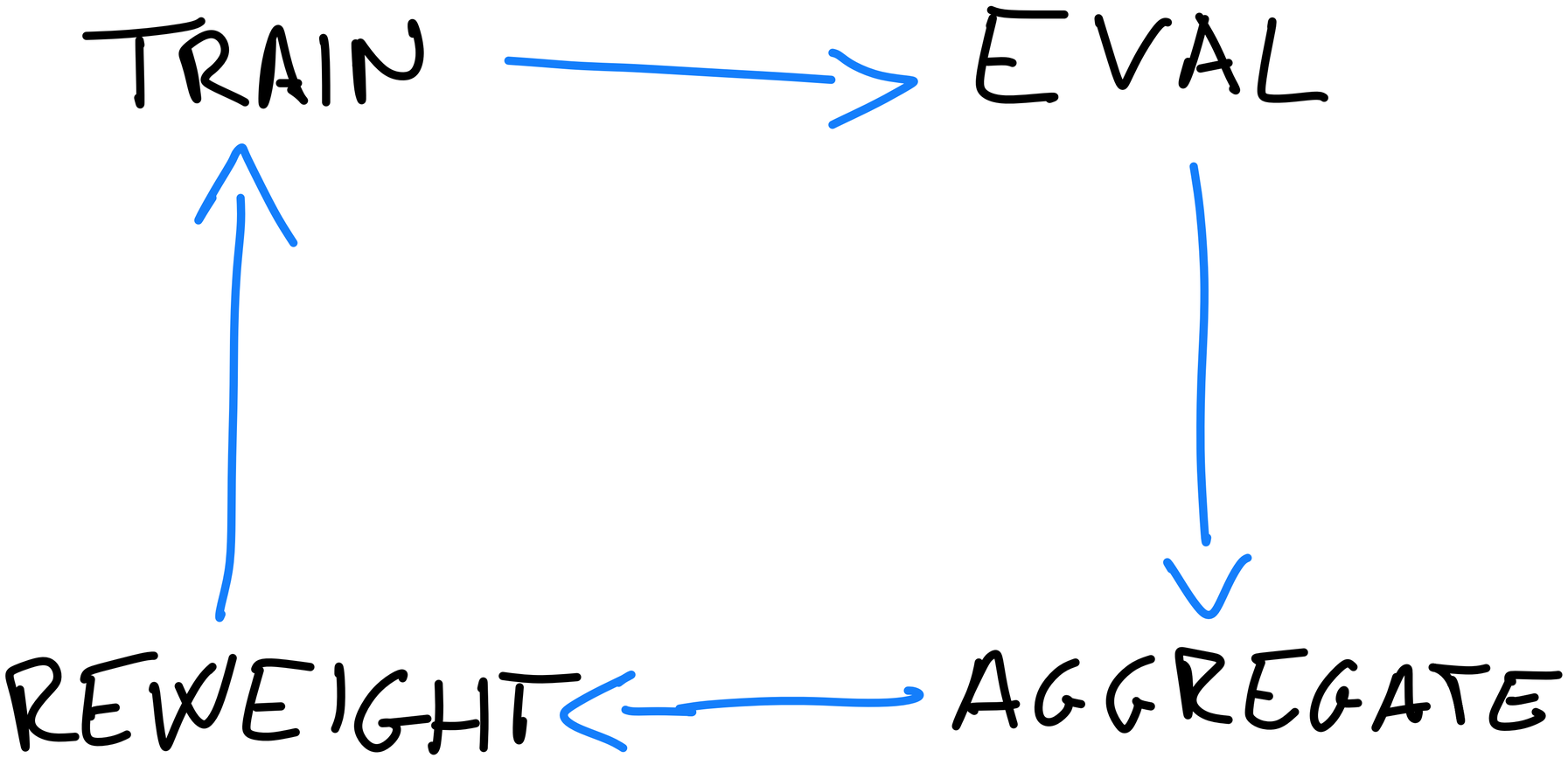
# Bias Variance tradeoff

## Boosting

- High bias leads to underfitting
- Low bias leads to overfitting

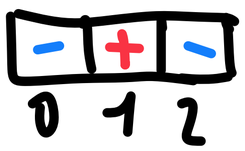


# Adaboost



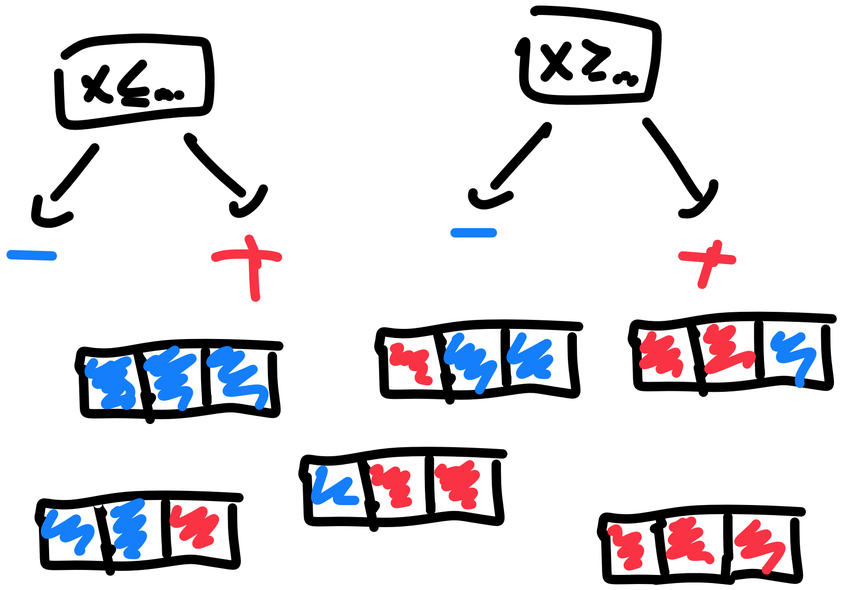
# Adaboost

## Setup



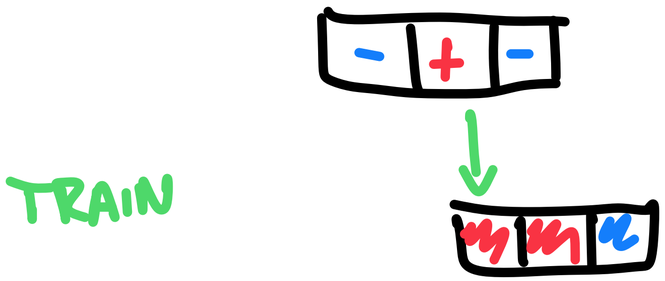

Training data

Ideal model



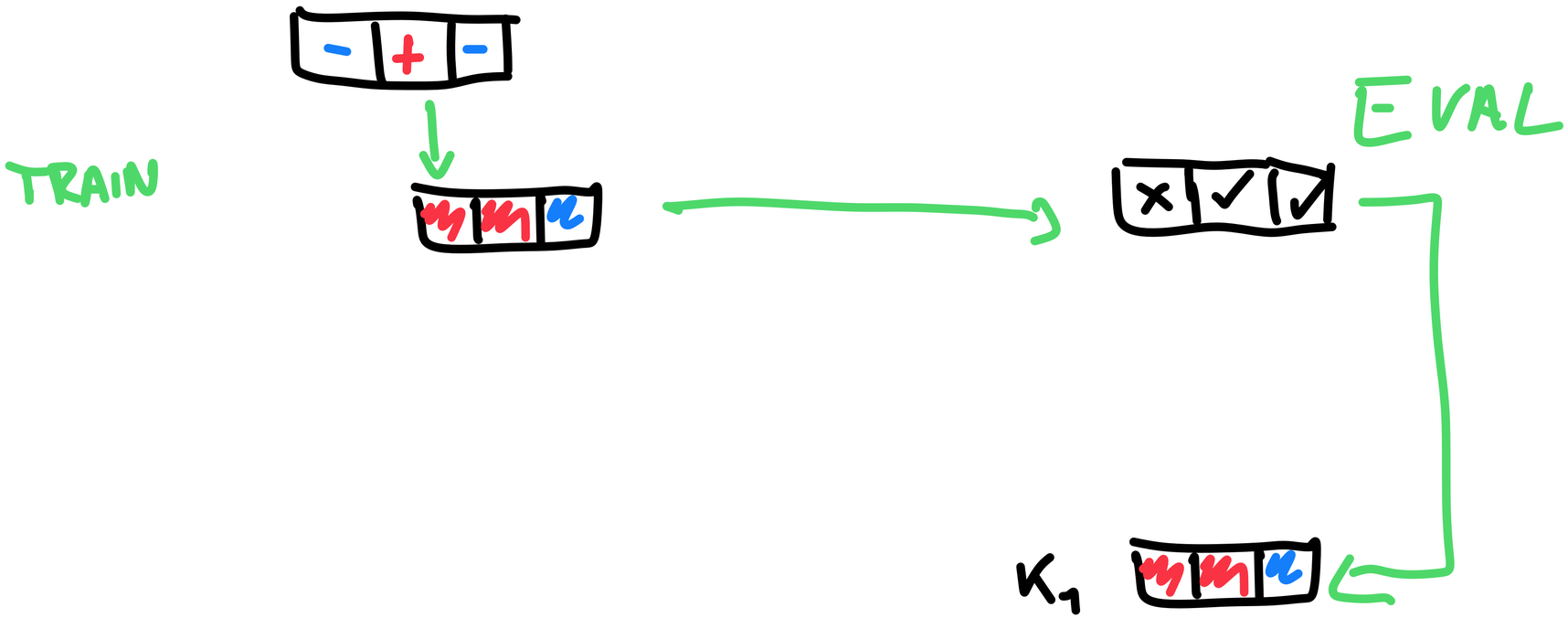
Base models

# Adaboost

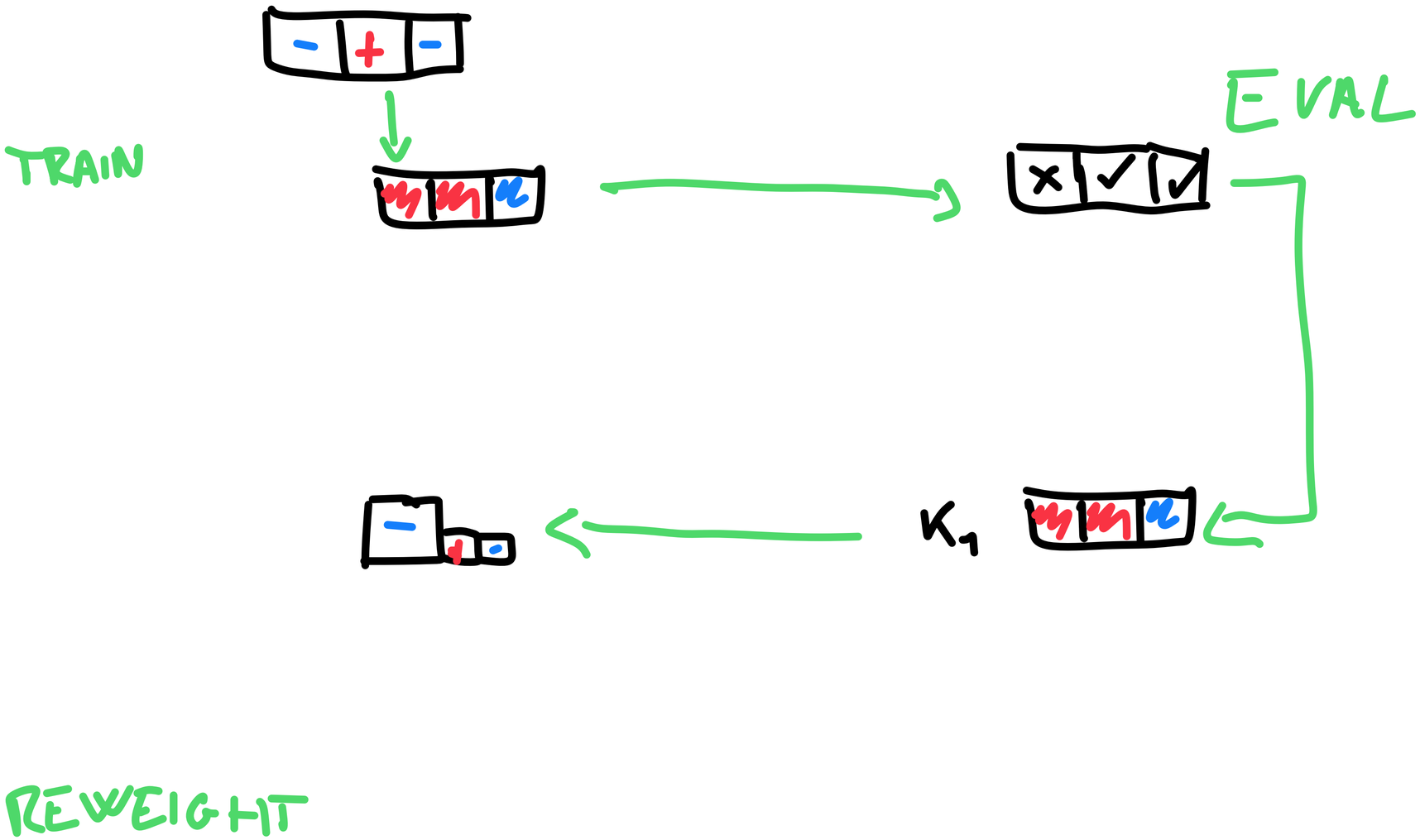


# Adaboost

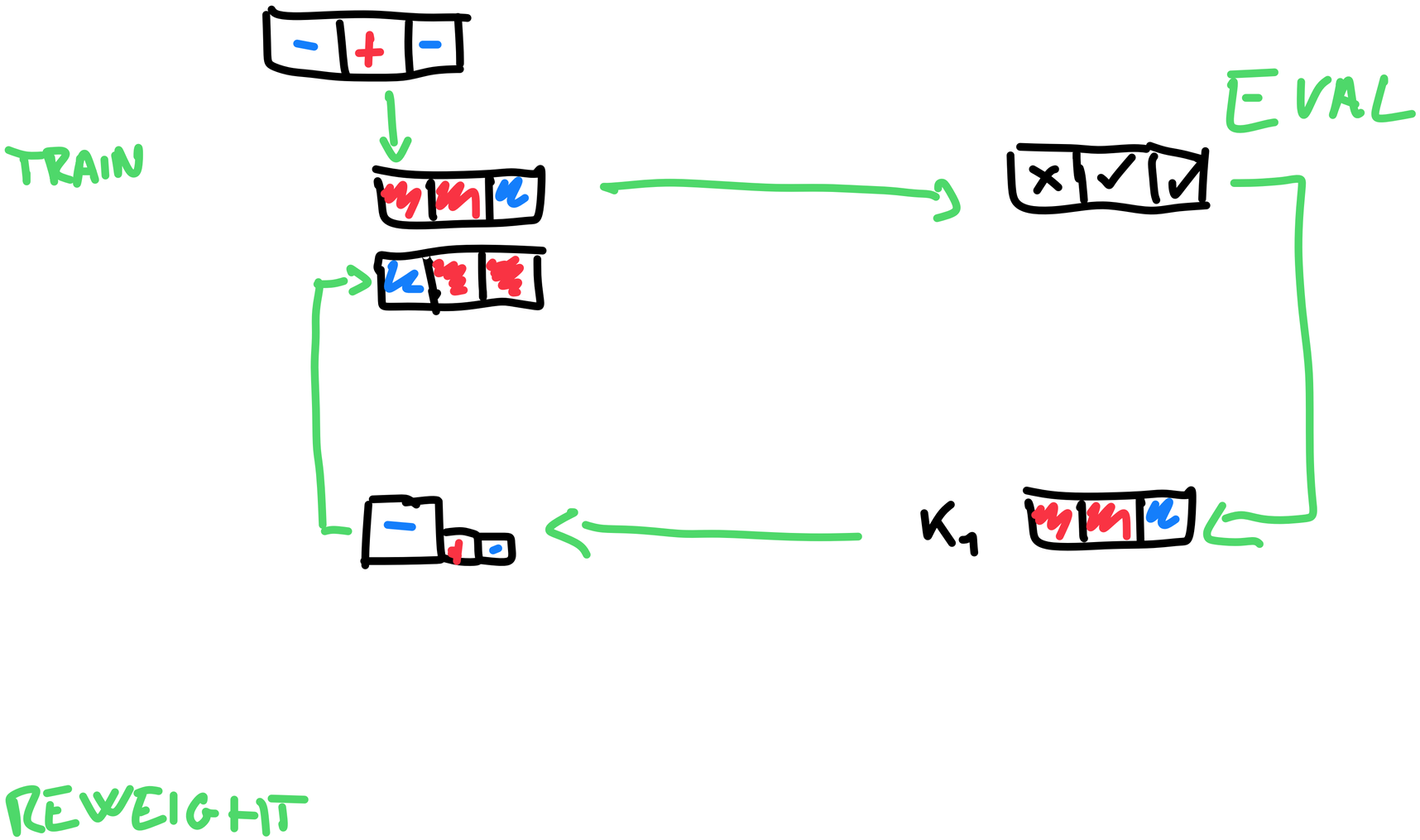
# Adaboost



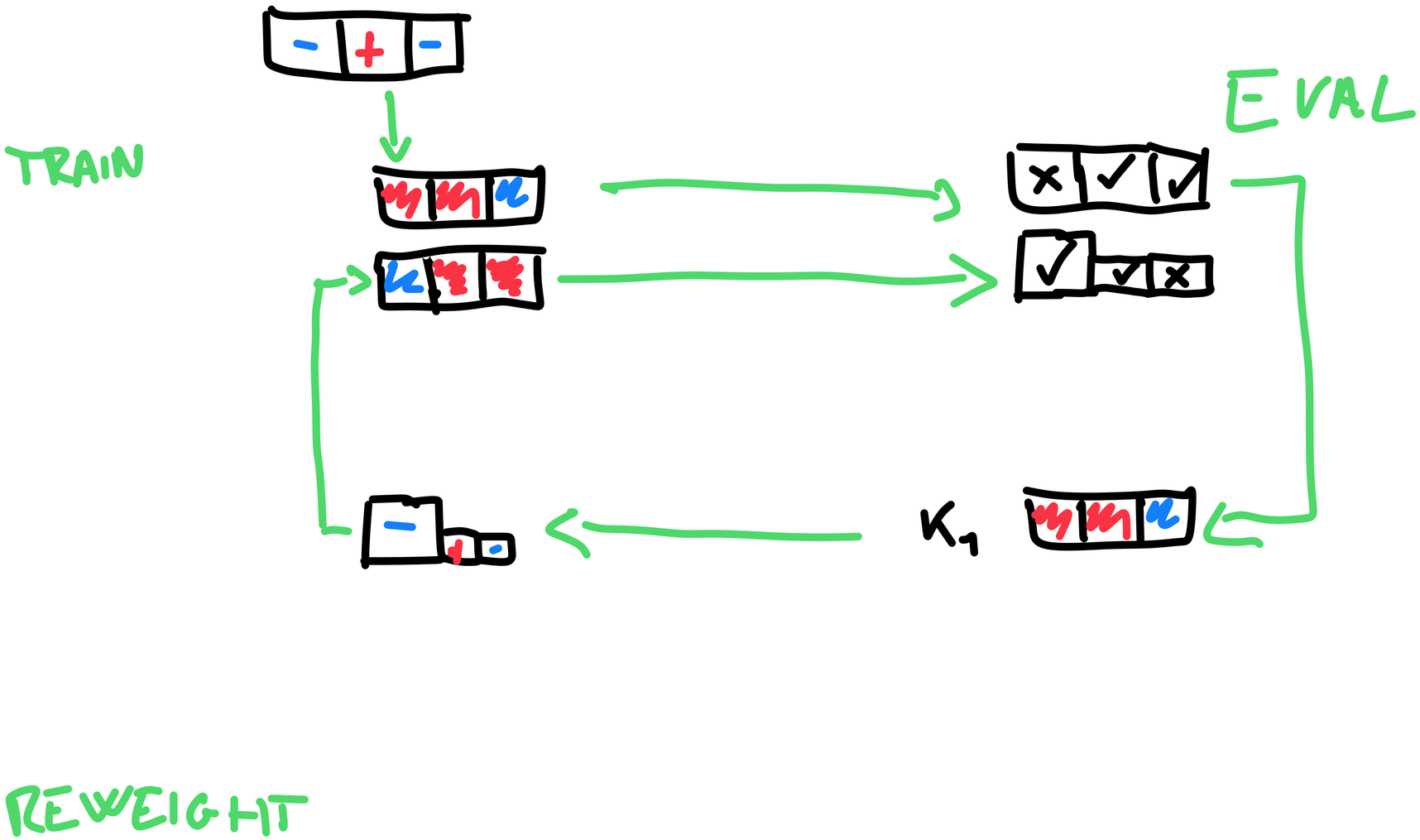
# Adaboost



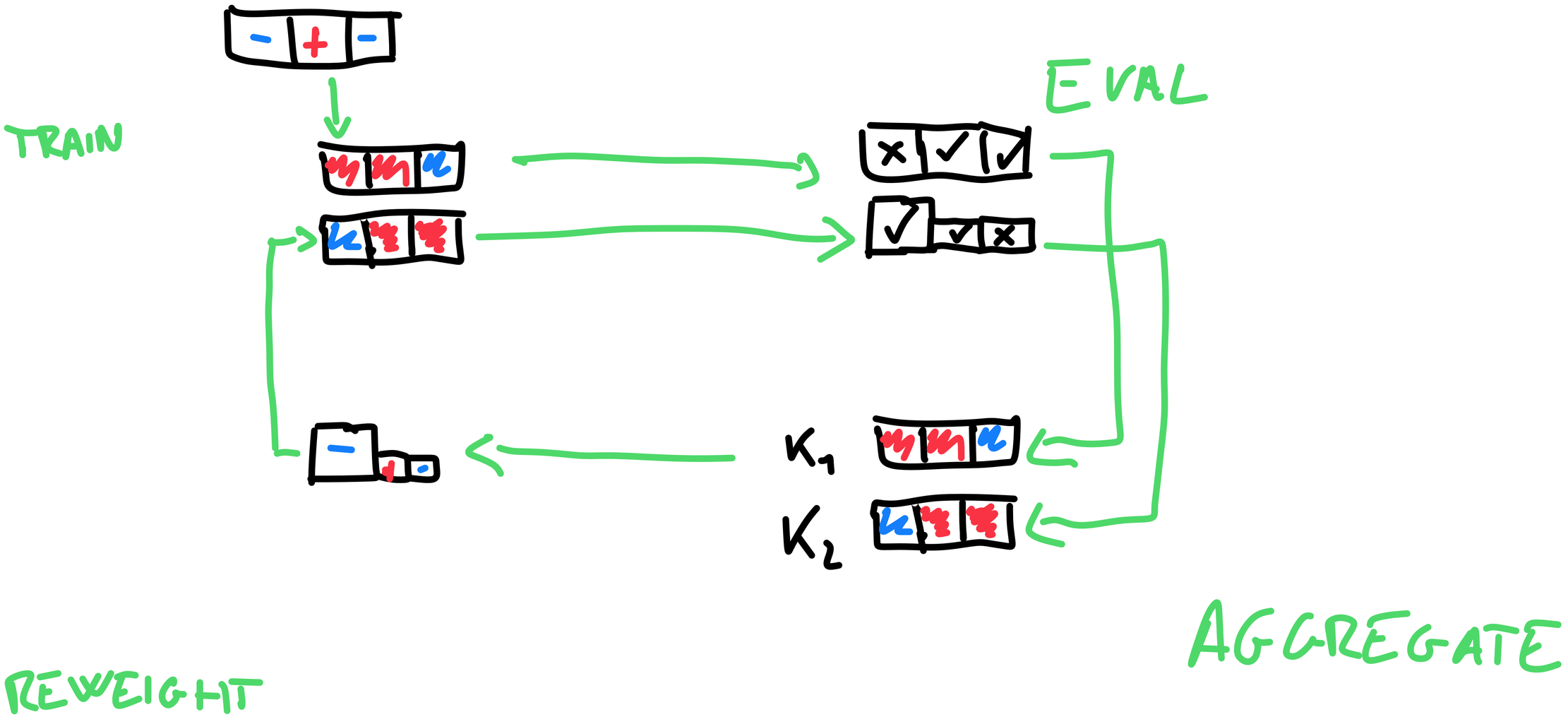
# Adaboost



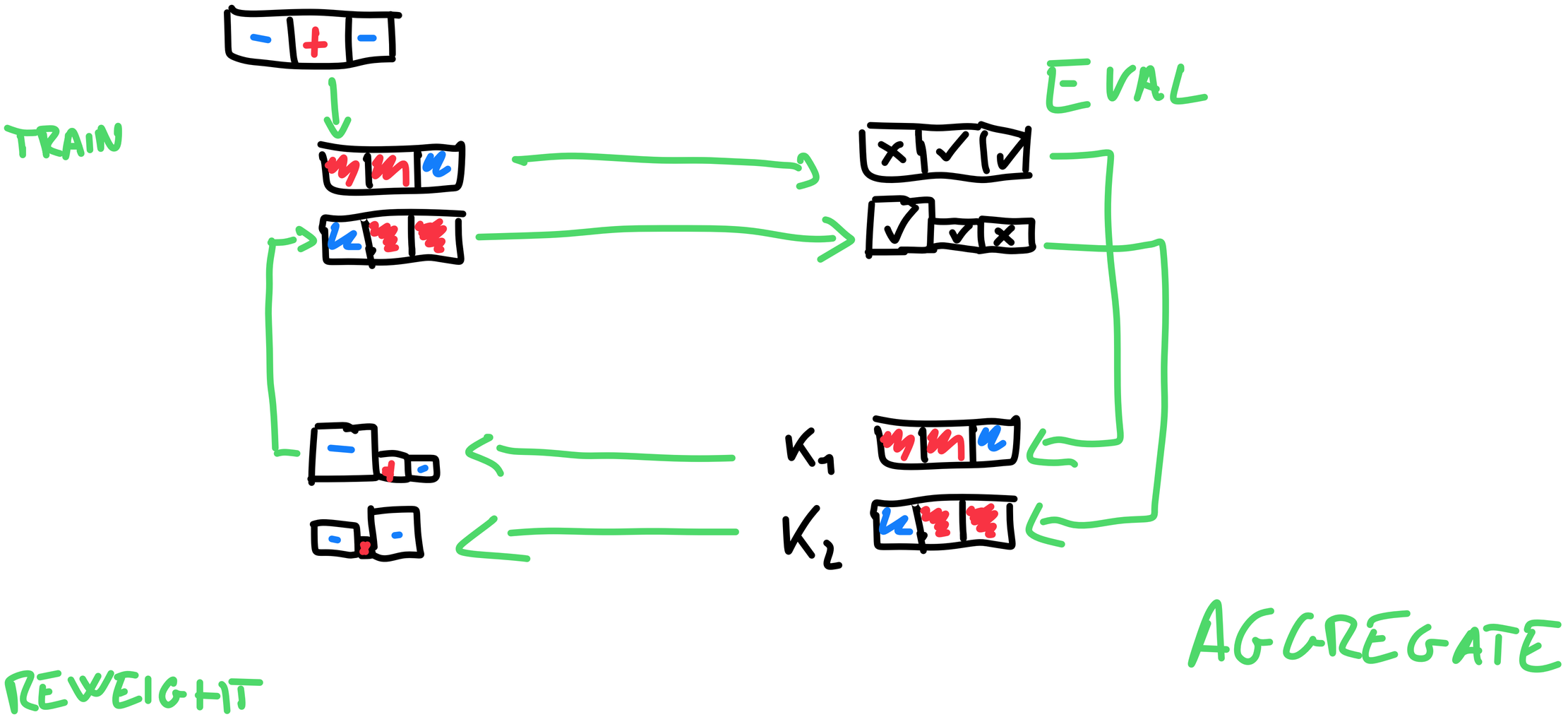
# Adaboost



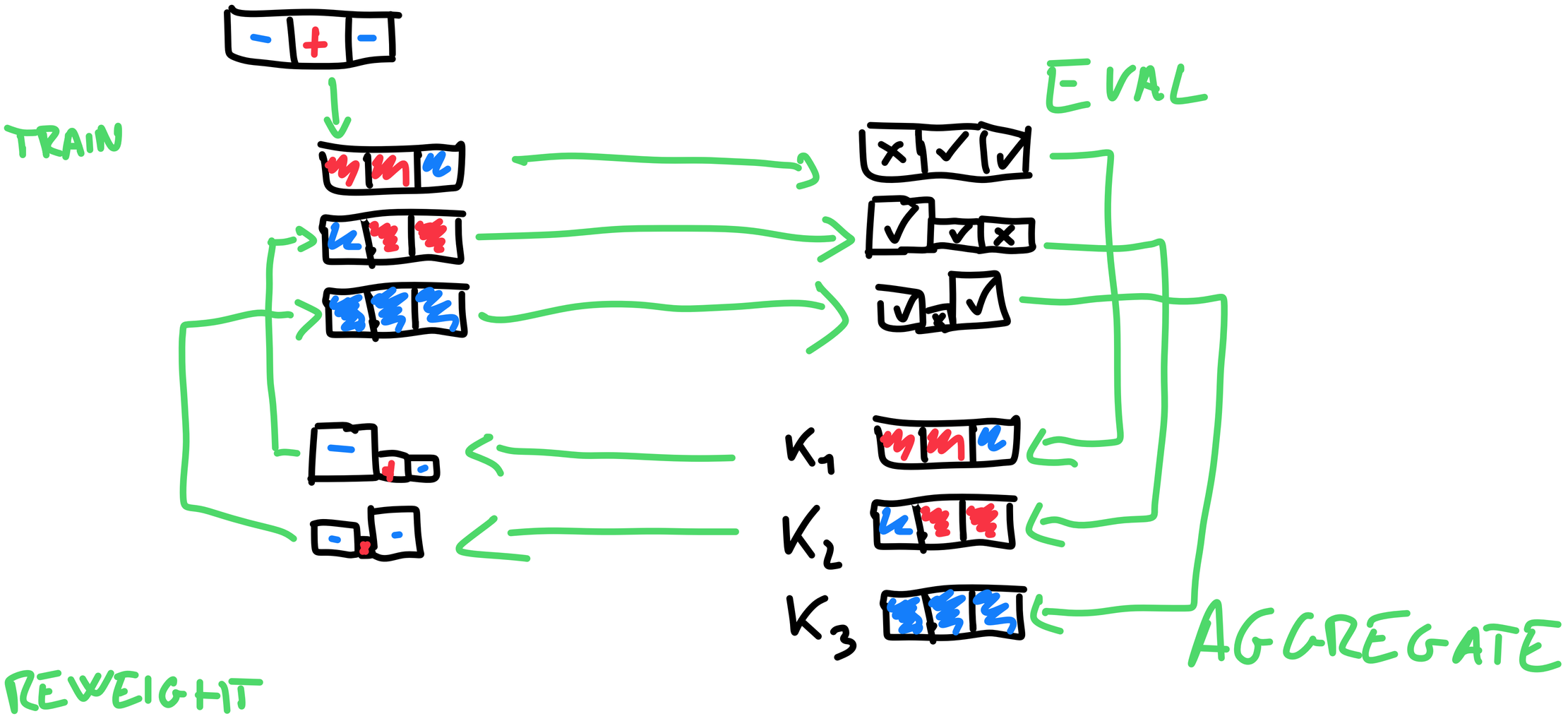
# Adaboost



# Adaboost

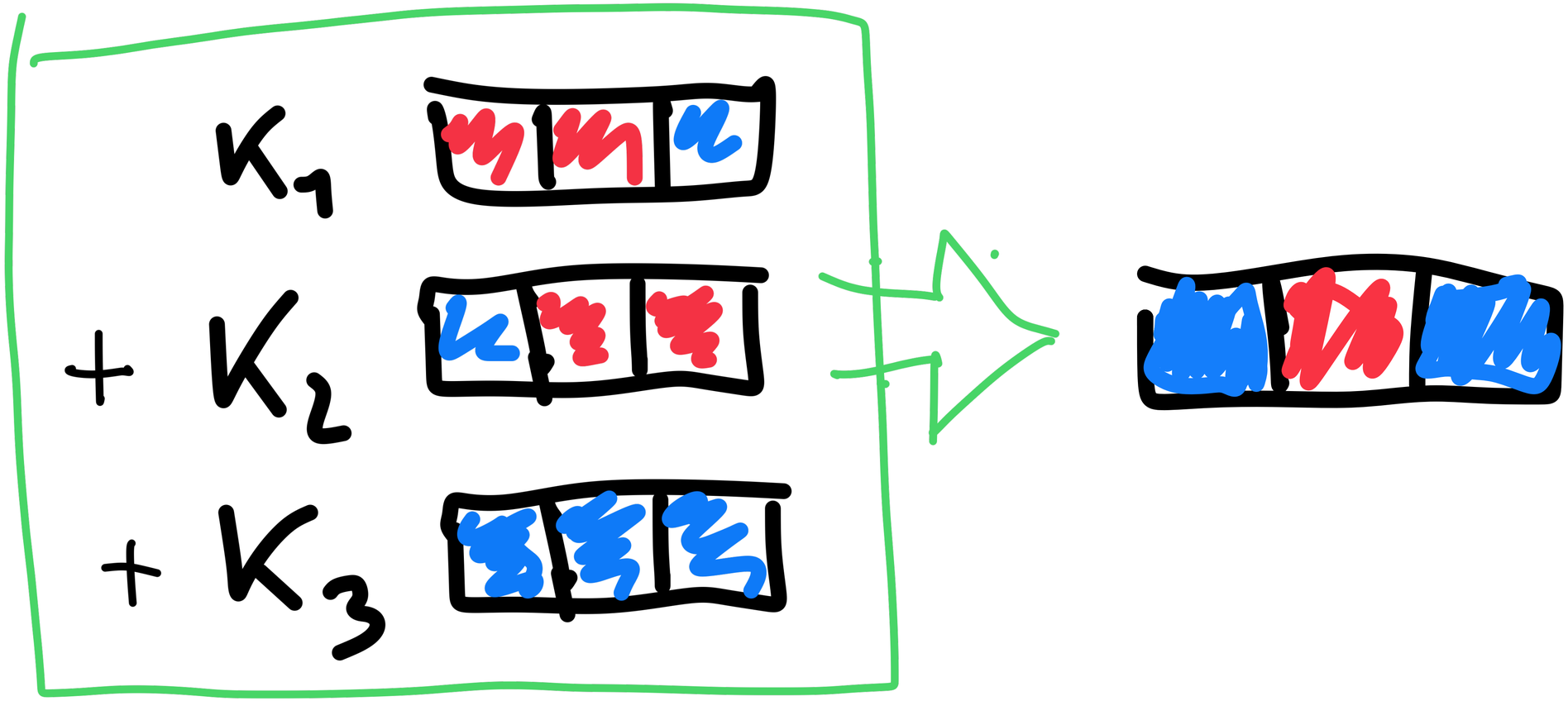


# Adaboost



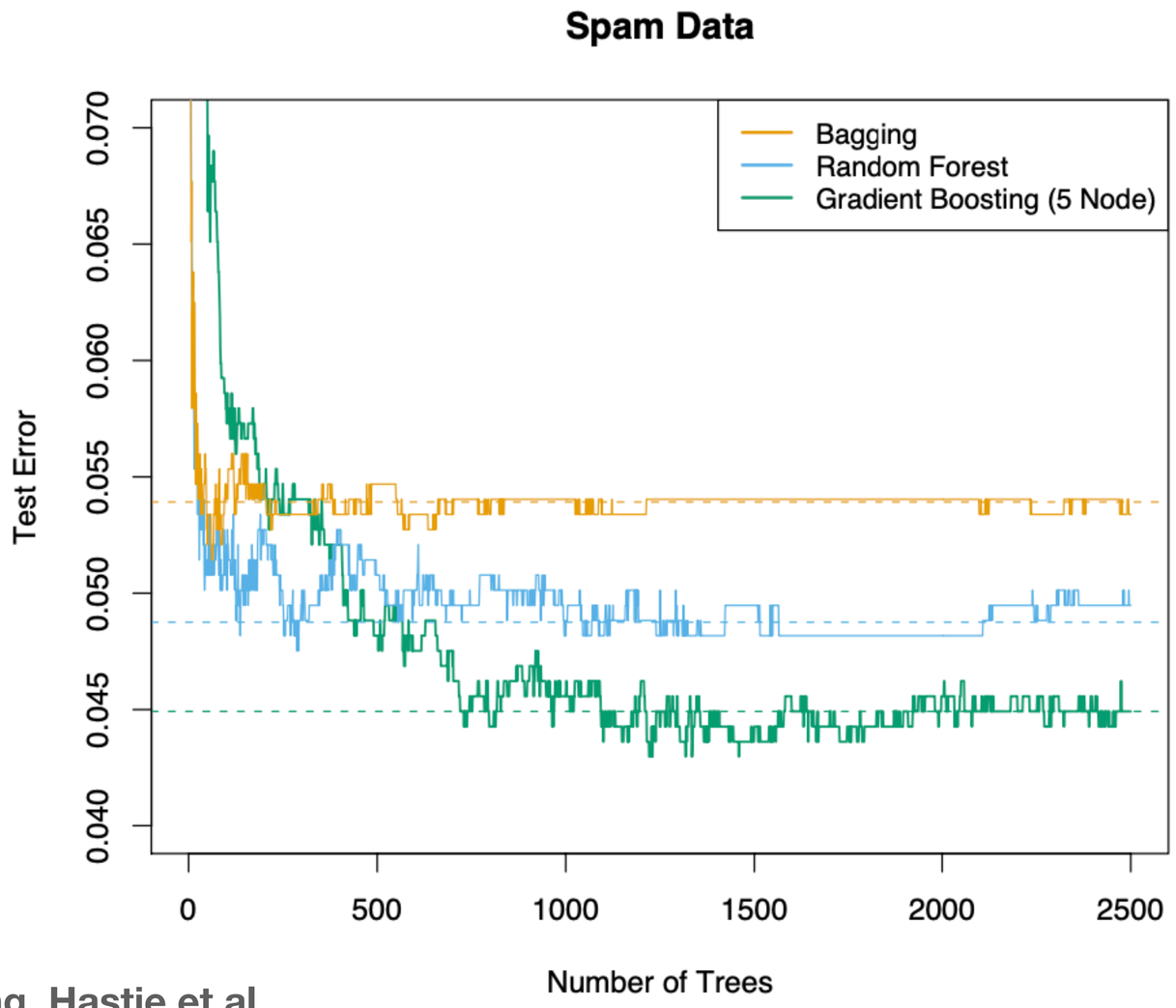
# Adaboost

## Inference



# Boosting

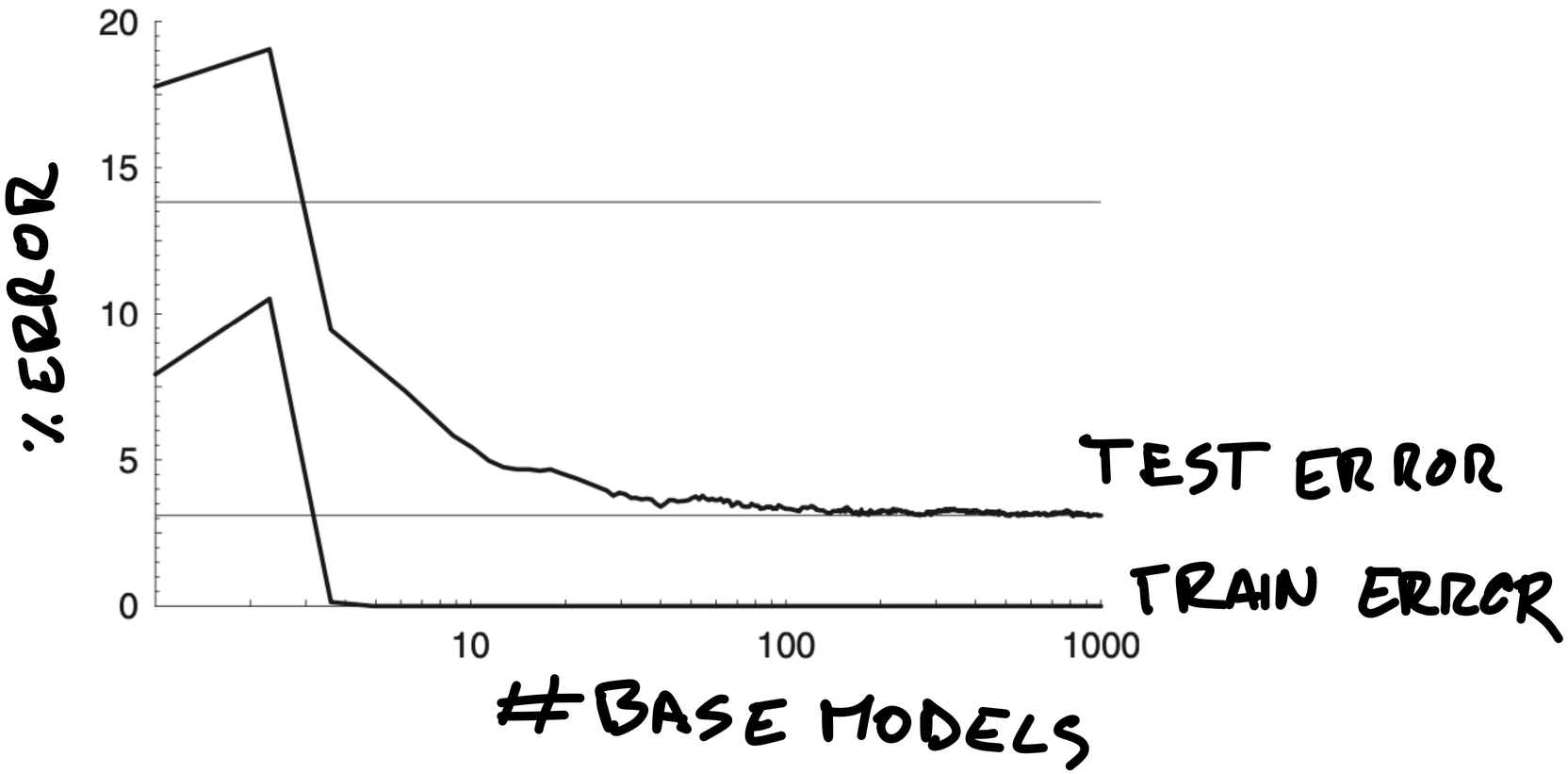
## Experiment




Elements of statistical learning, Hastie et al

# Boosting

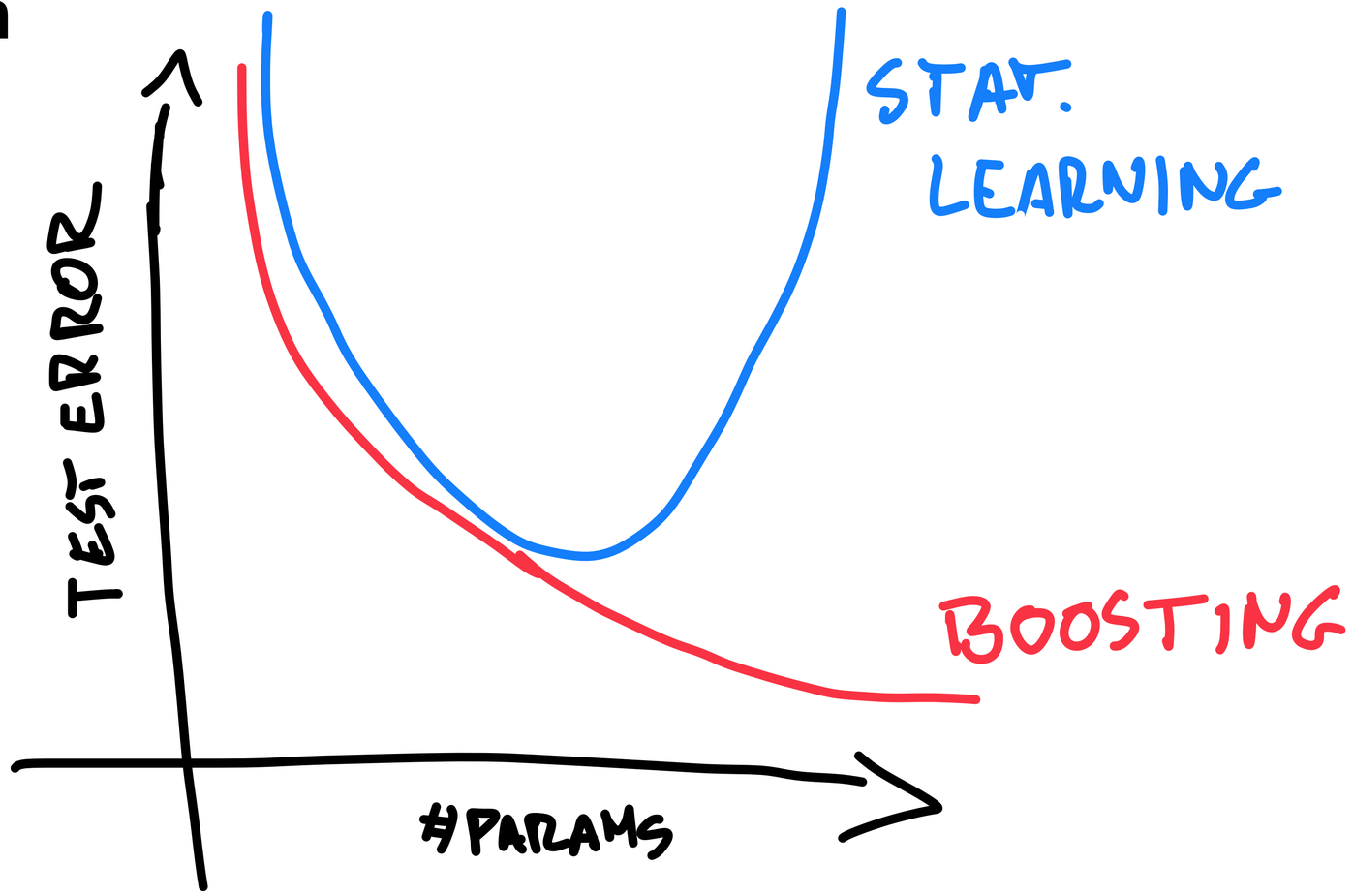
## Experiment




Boosting foundation and algorithms, Schapire and Freund

# Boosting

## Observation

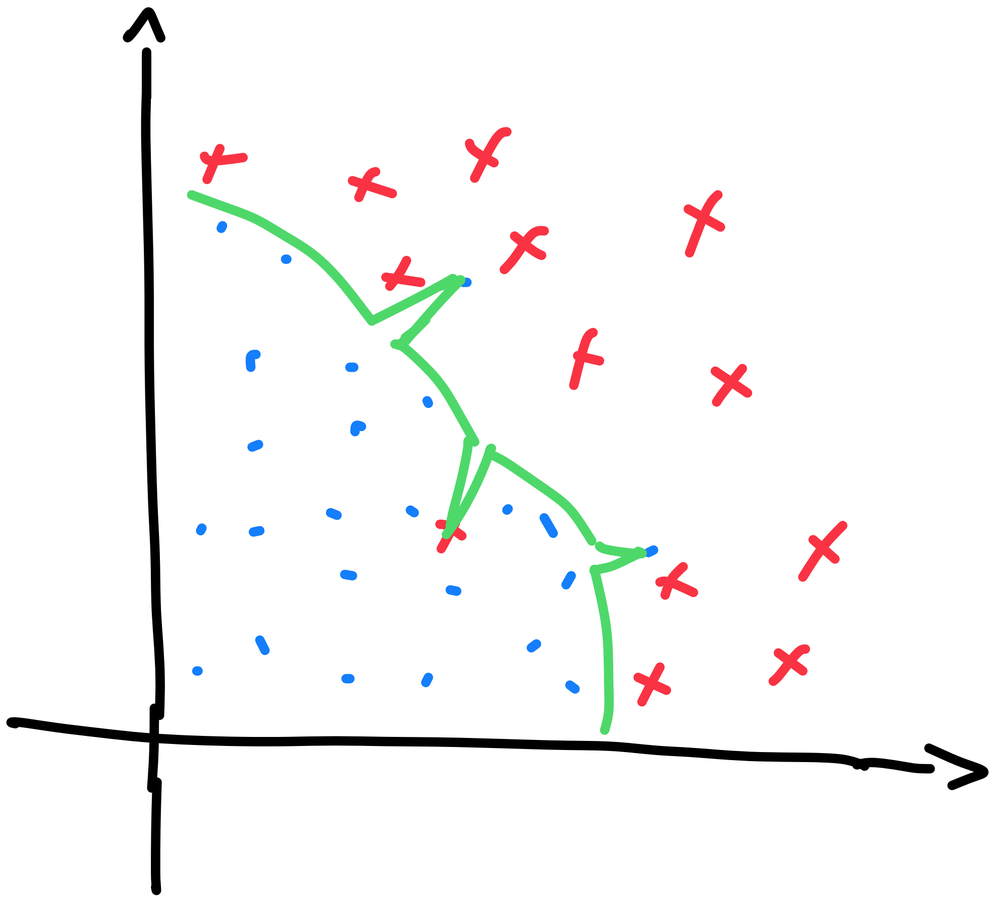


# Boosting

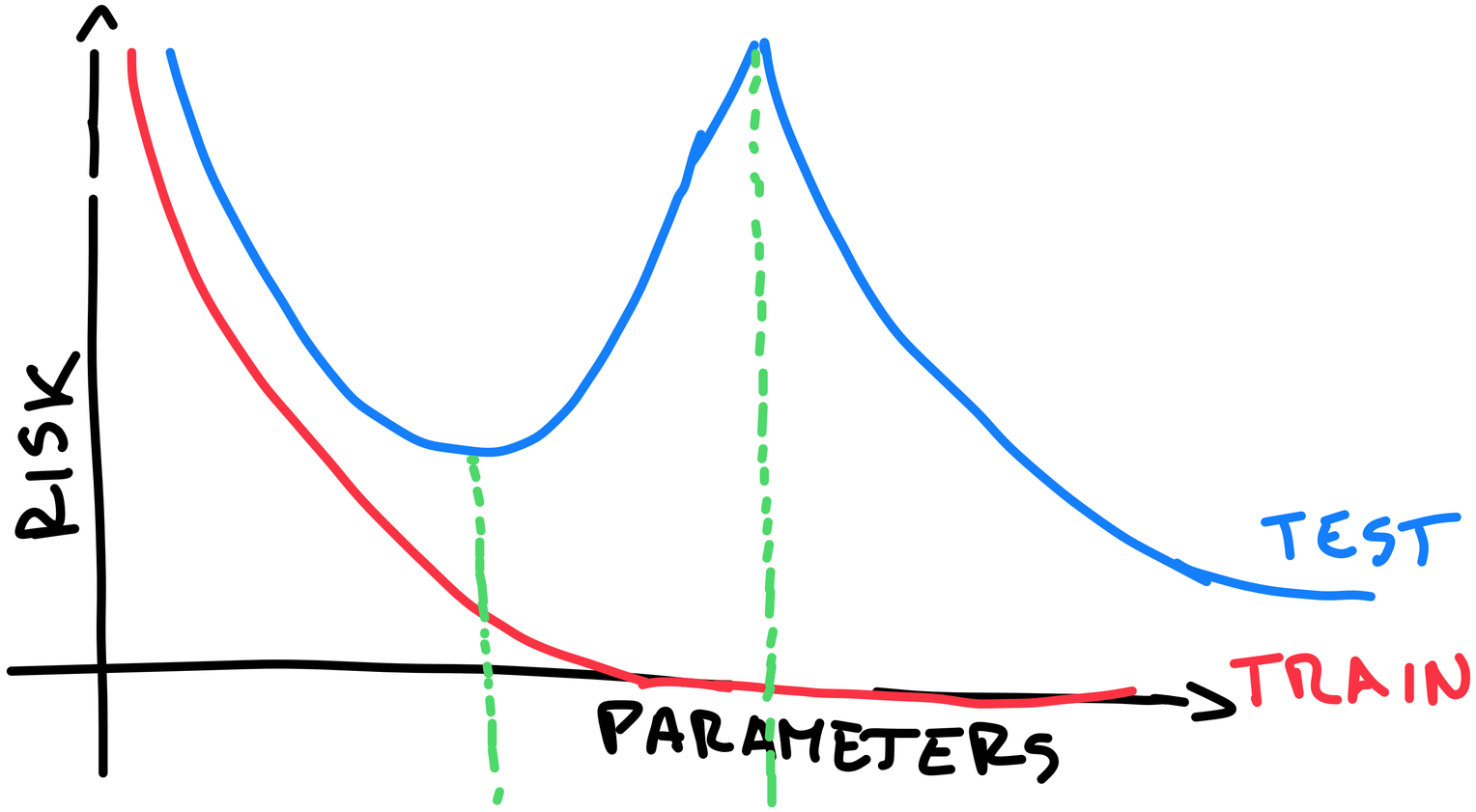
## Explanations

- (Friedmann) Adaboost is just forward additive stepwise modeling with exponential loss
- (Shapire and Freund) Boosting trains maximum-margin classifiers
- (Wyner et al) Adaboost and random forests train spiky, interpolating and self-averaging models
- (Belkin et al) Boosting reaches the overparameterized regime in double descent

# Spiky interpolators



# Double descent



# XGBoost

https://xgboost.readthedocs.io/en/stable/tutorials/model.html