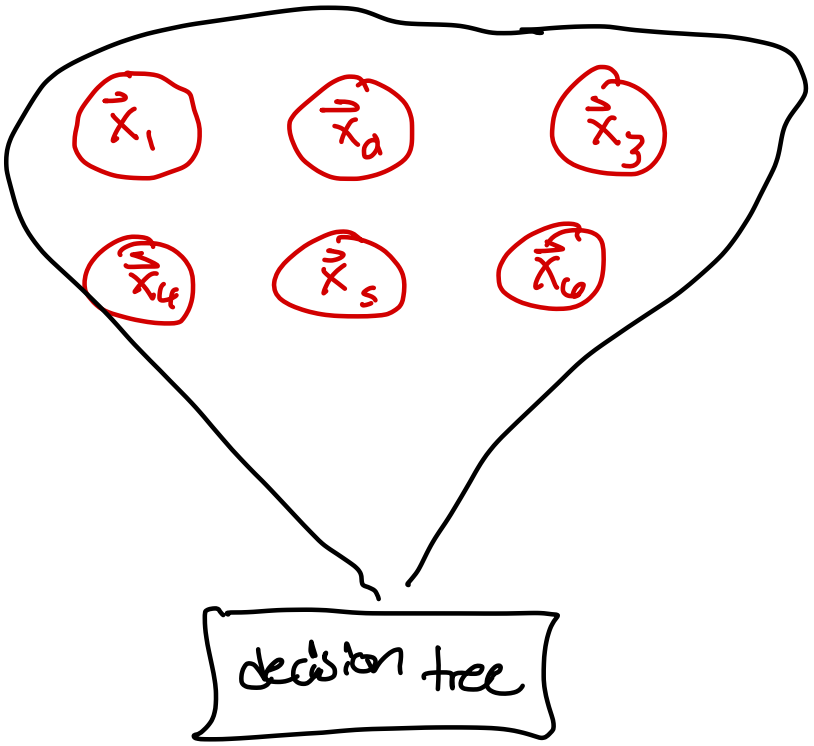
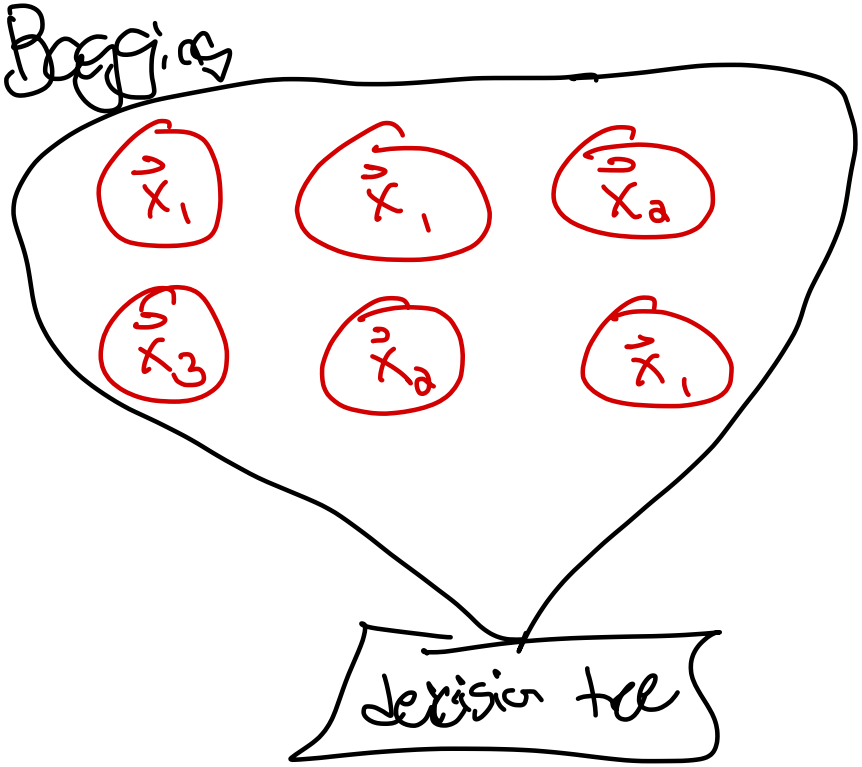
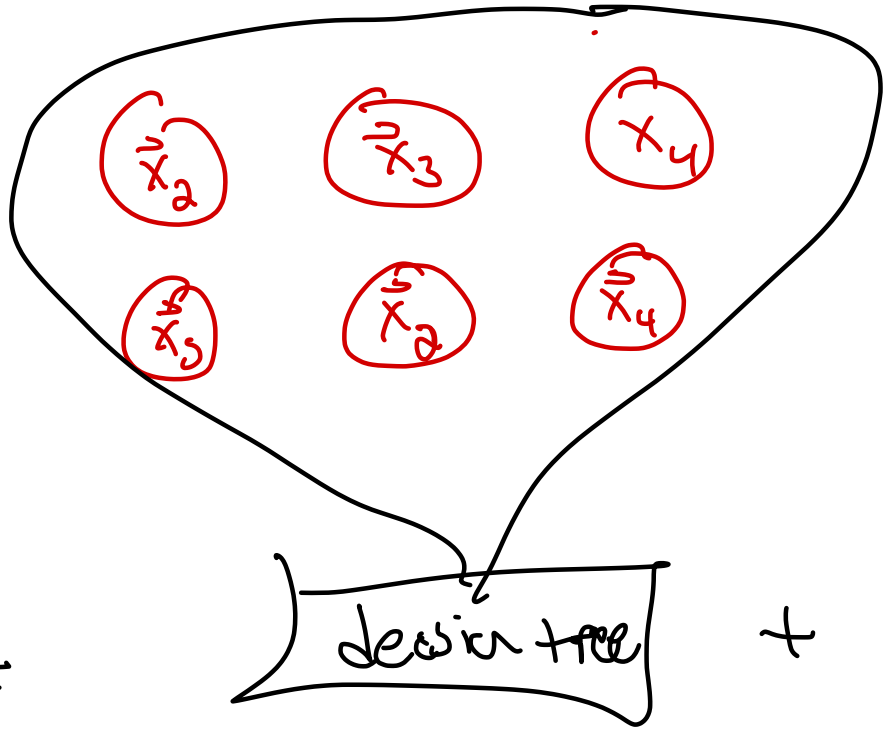


random forest:
a bunch of these
that all vote

Bagging



+

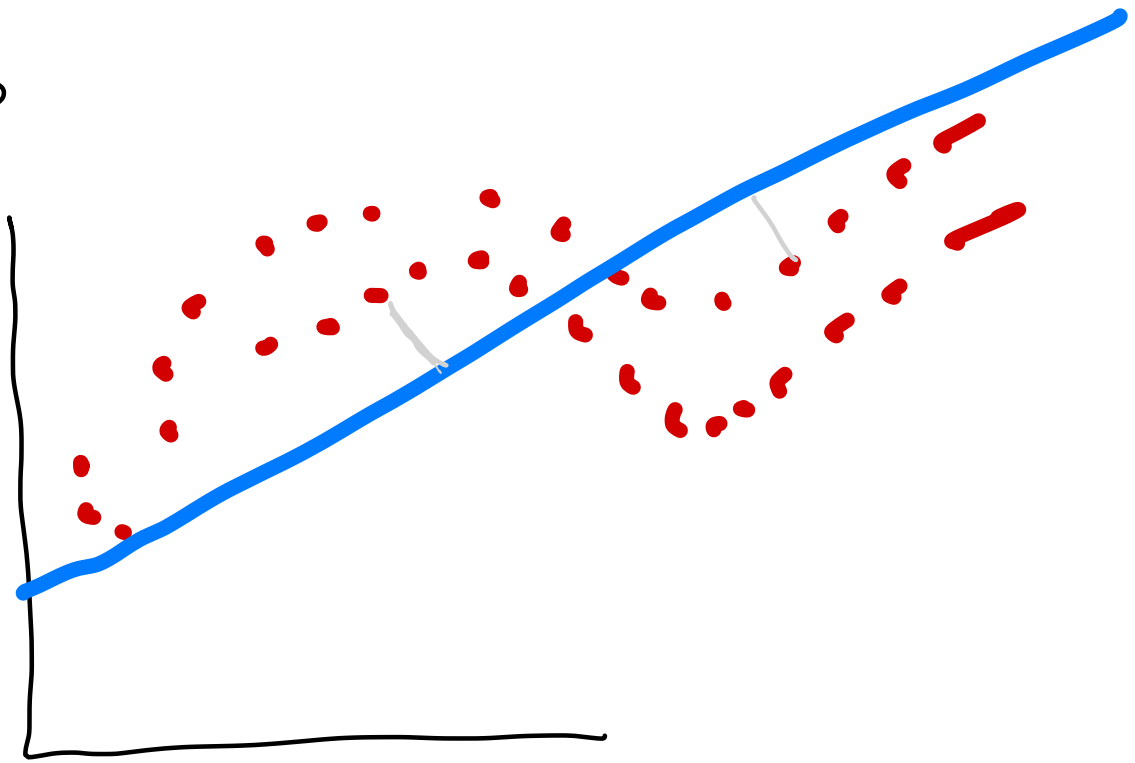


+ ...

Variance

bagging → designed to combat high variance

bias



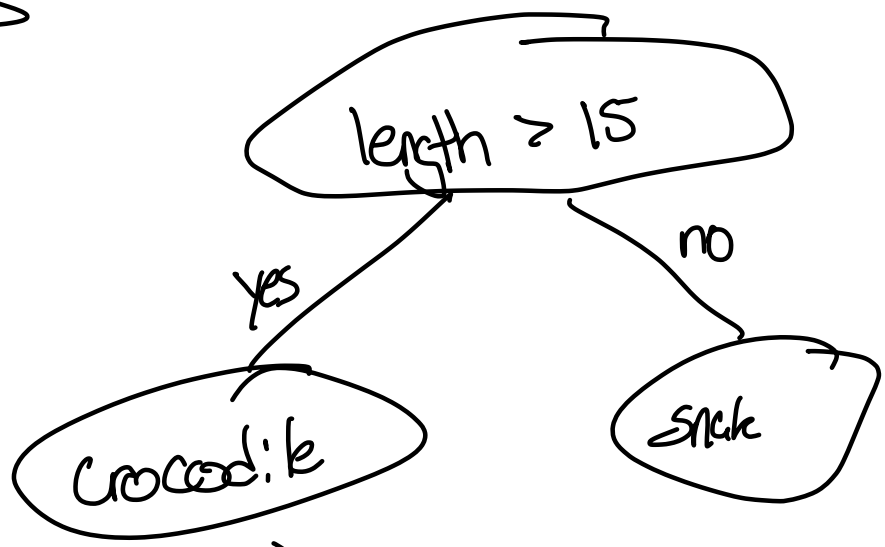
high bias

Boosting

↳ combat high bias

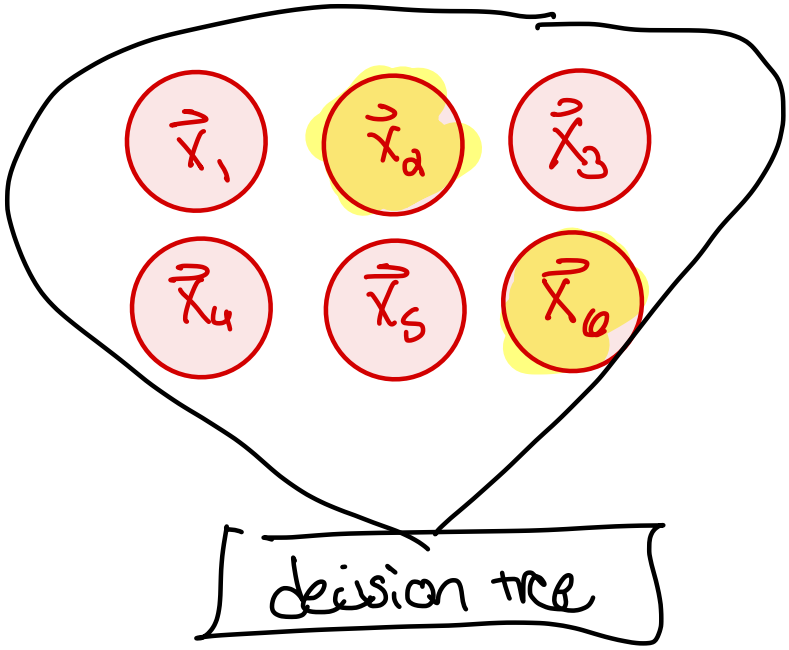
↳ "weak learners"

↳ trained sequentially

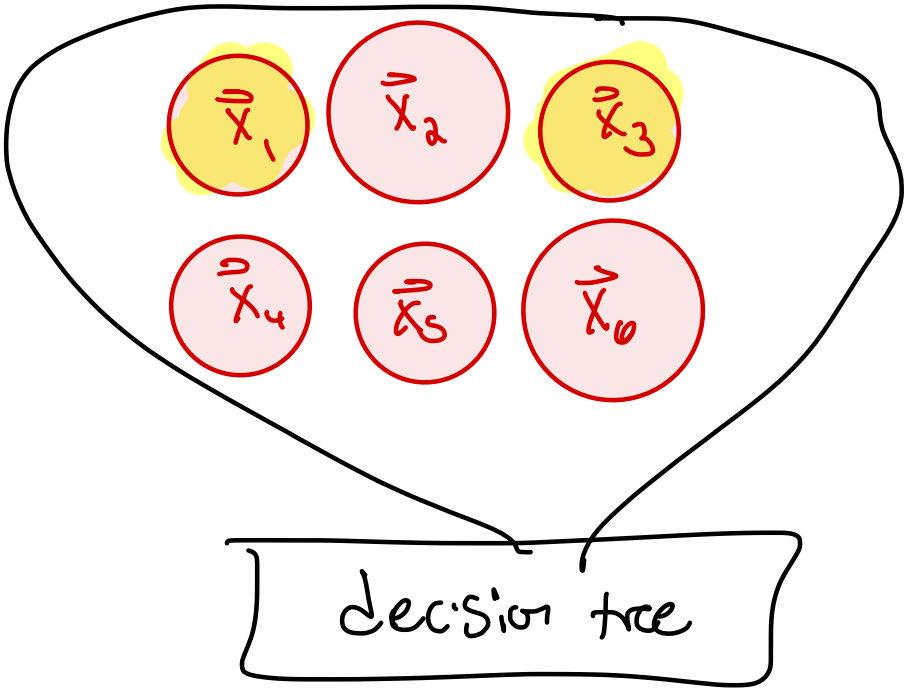


- Adaboost ("adaptive boosting")
  - adjust weights based on what it got wrong
- Gradient boosting
  ↳ next predictor is fit to errors made by the previous predictor

- XGBoost
  - ↳ uses more regularized model
- Light GBM
  - ↳ different gradient boost



$h_1(x)$



$h_2(x)$

$$\alpha_1 h_1(x) + \alpha_2 h_2(x) + \ldots + \alpha_m h_m(x)$$

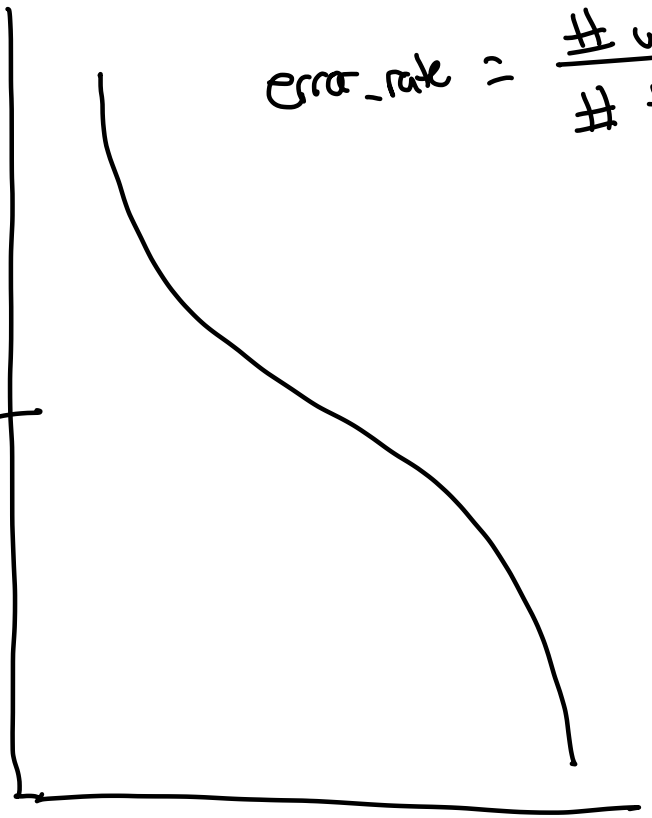
weight	length	class	sample weight
		S	1/8
		C	1/8
		C	1/8
		C	1/8
		S	1/8
		S	1/8
		C	1/8
		S	1/8

+1 or -1

$h_1(x)$

$$\alpha = \frac{1}{2} \ln\left(\frac{1-\text{error rate}}{\text{error rate}}\right)$$

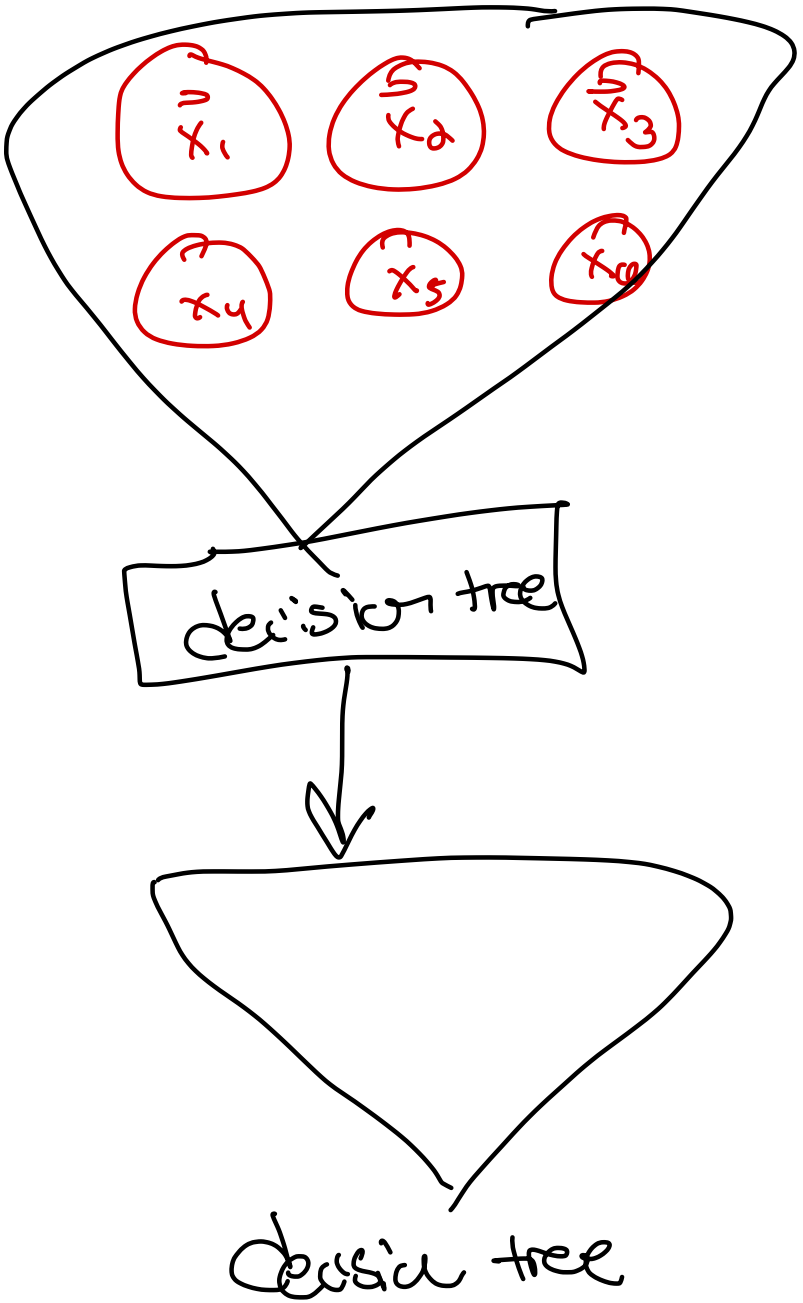
$$\text{error\_rate} = \frac{\#\text{ wrongs}}{\#\text{ total}}$$



$$\sum_i w_i \, \delta(h(x_i) \neq y(x_i))$$

1 if not equal
0 otherwise

$$w_i e^{-\alpha y h(}$$


x_1
x_2
x_3
x_4
x_5
x_6
decision tree
decision tree

Other consideration with trees/forests

1) imbalanced data
   a) be aware → compare to baseline accuracy
   b) oversample minority classes
      ↳ down sample majority class
      ] more balanced dataset to train on
   c) penalize minority mistakes more

↳ Balanced Random forest
   ↳ draw bootstrap samples from minority class
   ↳ draw same # of samples, with replacement, from majority class
   ↳ repeat: train decision tree incorporating randomness

↳ Weighted Random forest
   a) assigns weight to each
   b) weight GINI when choosing split
   c) uses weighted majority vote when determining predictions

- different costs for features
  - replace information gain with $\frac{\text{gain}}{\text{cost}}$