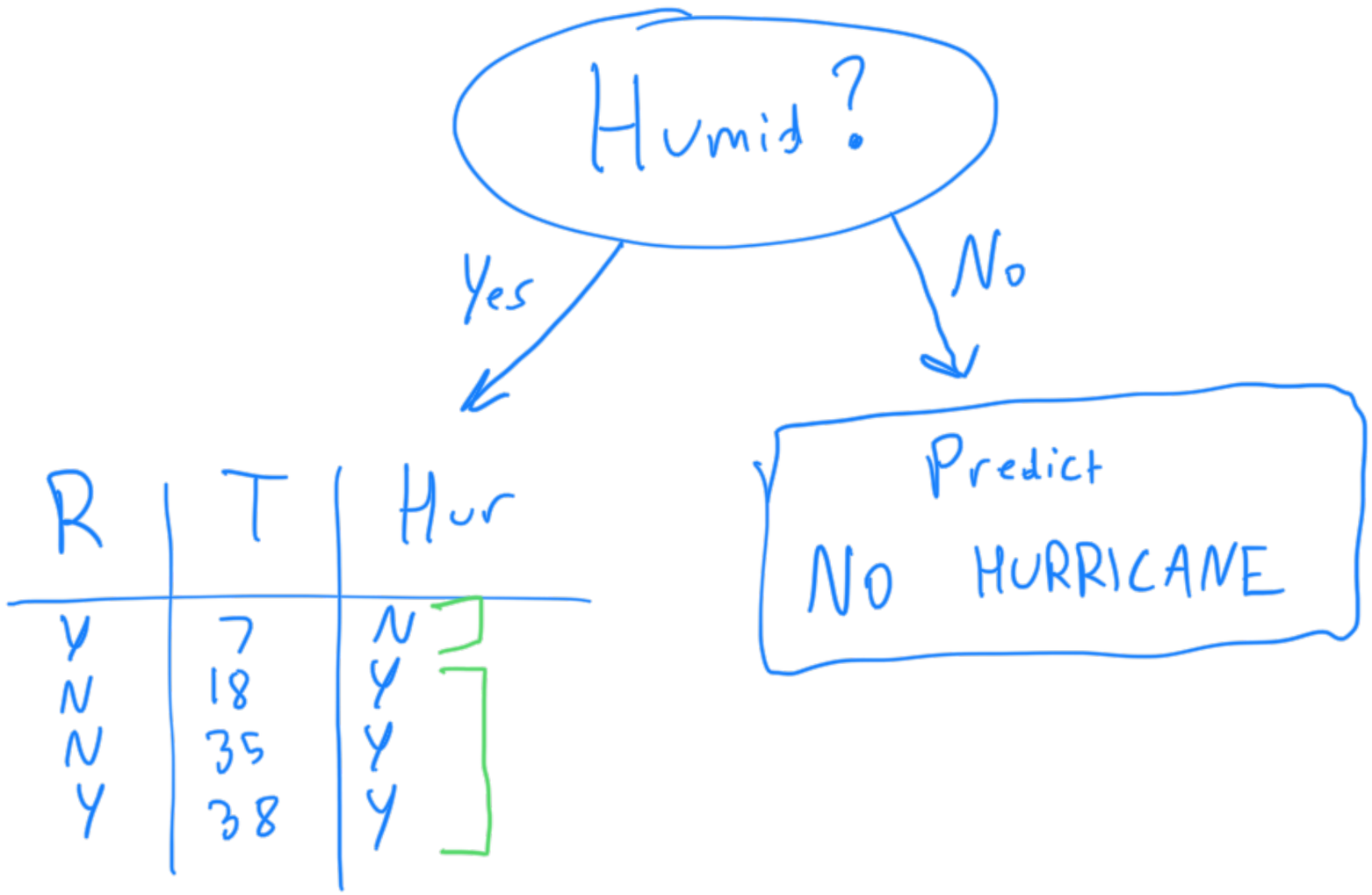
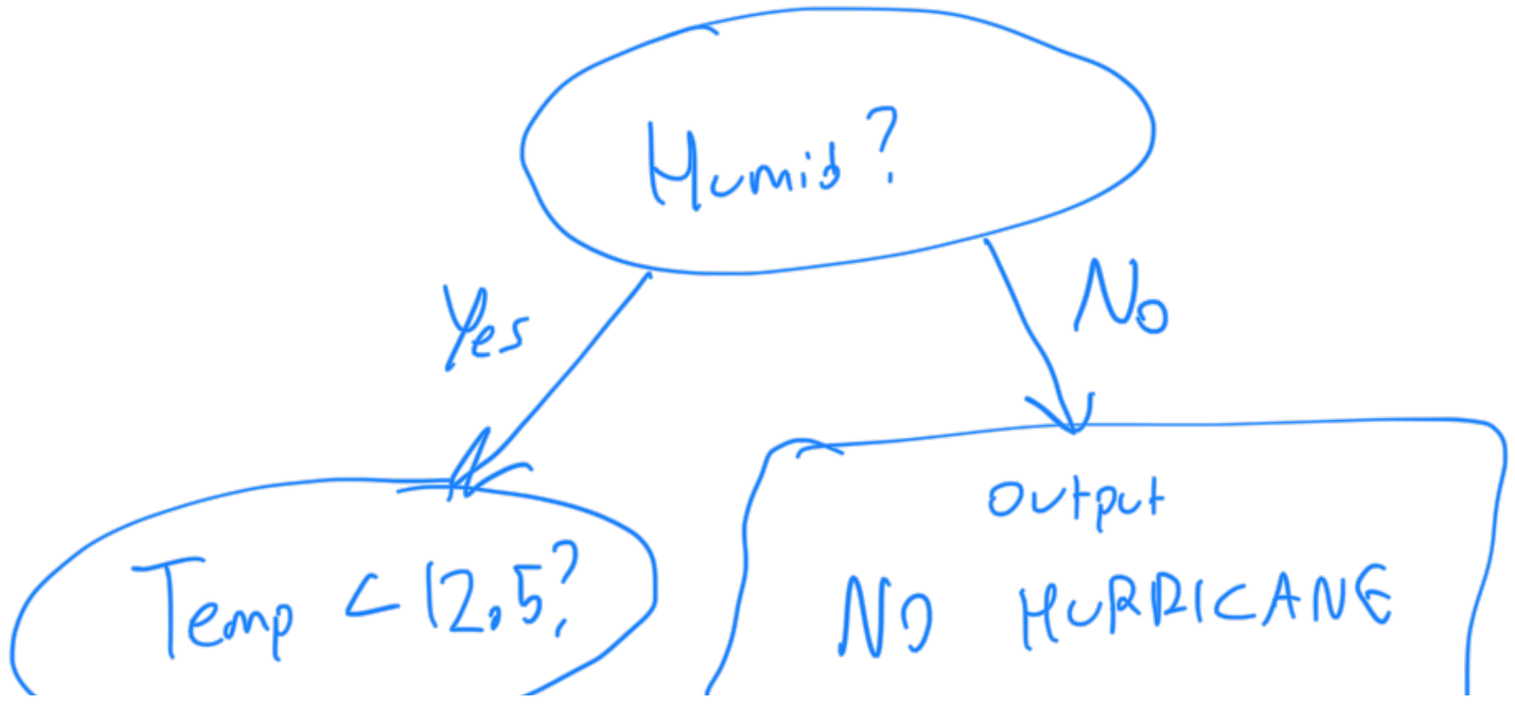
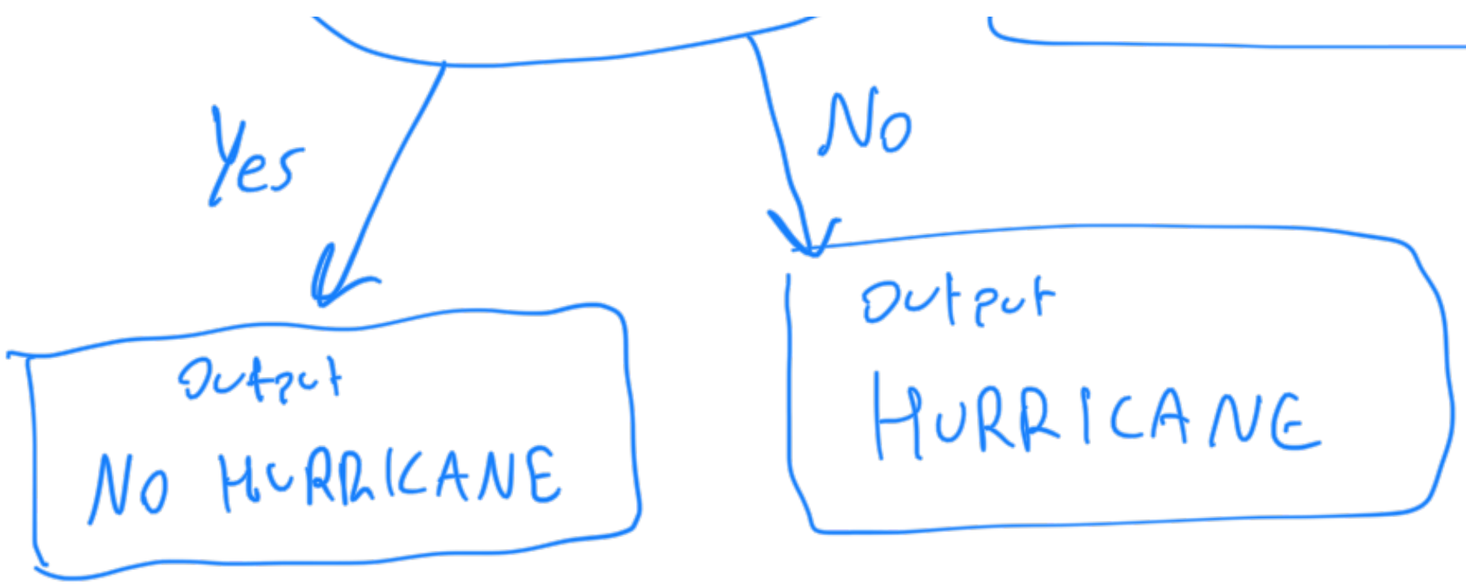
Day 12: Ensemble Models

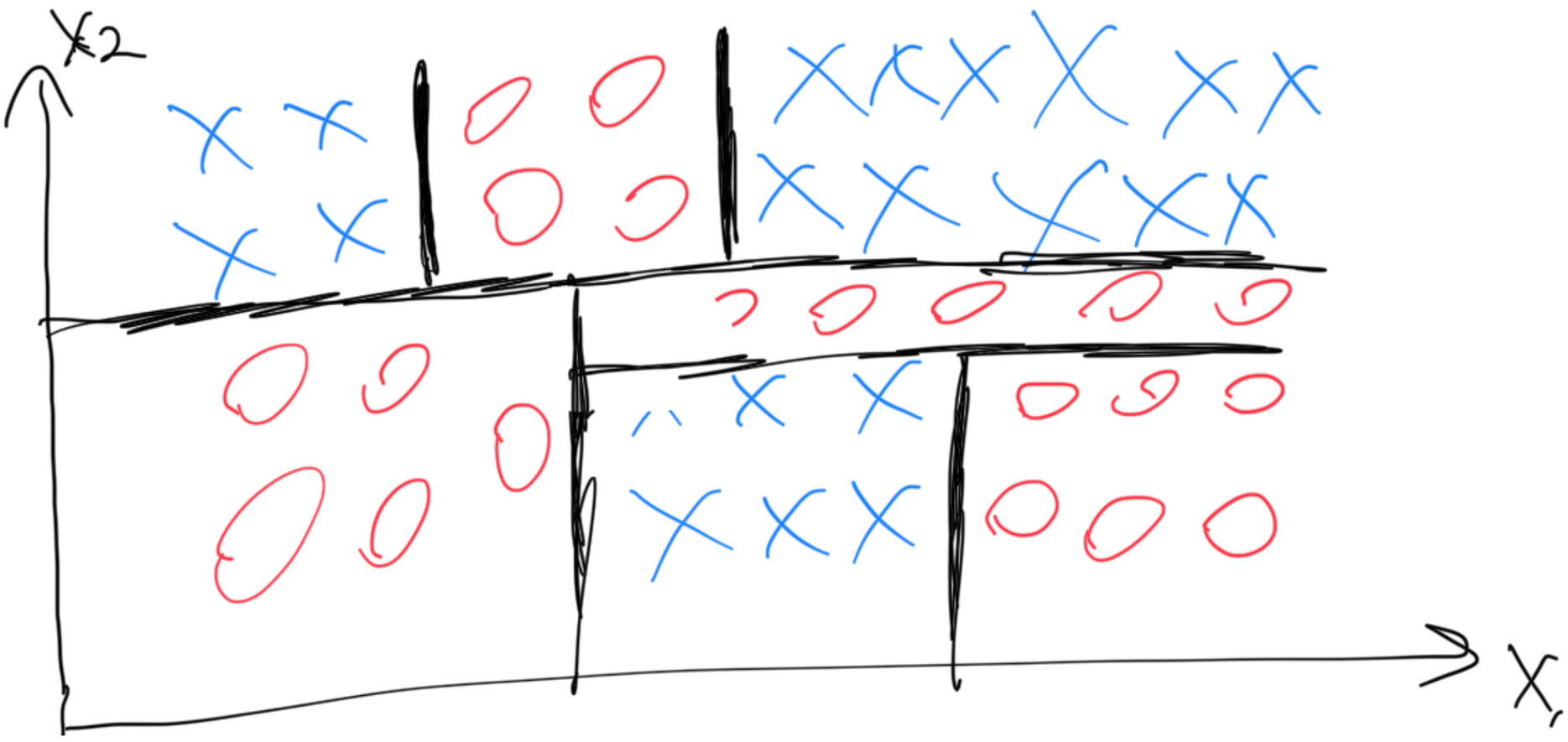


Our final decision tree is:





Decision Trees Can Learn Complex Non-Linear Separating Boundaries



Many possible Splitting metrics

Entropy

$$\text{Entropy}(x) = H(x) = -\sum_{i=1}^{n} p(x_i) \log_2 [p(x_i)]$$

n = # classes

"Disorder" "uncertainty"

"The number of bits required, on average, to encode information"

High entropy: high level of uncertainty

$$\left(\underset{a}{25\%}, \underset{b}{25\%}, \underset{c}{25\%}, \underset{d}{25\%}\right)$$

Low entropy: low level of uncertainty

(97%, 1%, 1%, 1%)
a b c d

(split on low entropy)

Information Gain

$$\text{Information Gain}(y, a) = H(y) - H(y|a)$$

$H(y)$: uncertainty/entropy before you split

$H(y|a)$: uncertainty/entropy after you split

"Expected reduction in entropy from splitting on an example"

Example

$$IG(\text{"Crash"}, \text{"Excess Gasoline"})$$

$$= H(\text{"Crash"}) - H(\text{"Crash"} \mid \text{"Excess Gasoline"})$$

$$H(\text{"Crash"}) = -\left[\left(\frac{4}{6}\right)\log_2\left(\frac{4}{6}\right) + \left(\frac{2}{6}\right)\log_2\left(\frac{2}{6}\right)\right]$$

$$= 0.918$$

$$H(\text{"Crash"} \mid \text{"Excess Gasoline"})$$

"Excess Gasoline"

Yes: $\left(\frac{4}{6}\right) \cdot -\left[\left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right) + \left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right)\right]$

No: $\left(\frac{2}{6}\right) \cdot -\left[\left(\frac{2}{2}\right)\log_2\left(\frac{2}{2}\right) + \left(\frac{0}{2}\right)\log_2\left(\frac{0}{2}\right)\right]$

→ add up the two terms:

$$H(\text{"Crash"} \mid \text{"Excess Gasoline"}) = 0.667$$

Decision Trees for Regression

Similar process, except:

* use MSE or sum of square error instead of IG, G, or H
* prediction is the average of values

in a split

* Stop when # of points in a node is below a threshold, or if you exceed some # of iterations

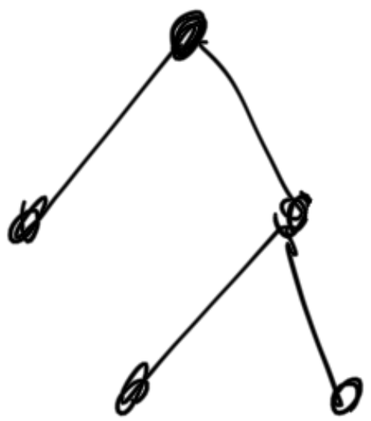
Random Forests

(and more broadly, Ensemble Learning)

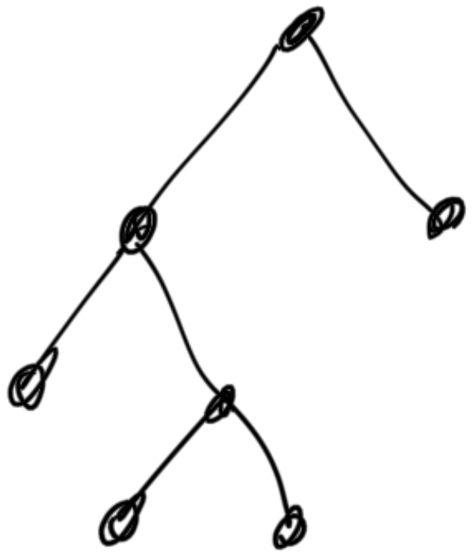
General idea

1. Build many decision trees*
2. Take the majority vote

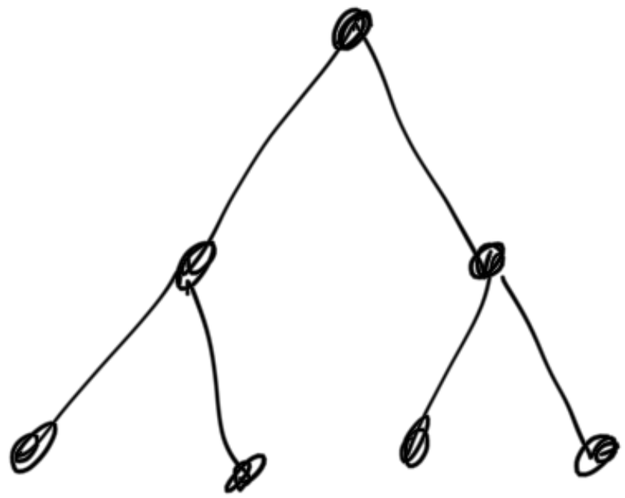
* there are many ways to build multiple decision trees



"Cancer"



"No Cancer"



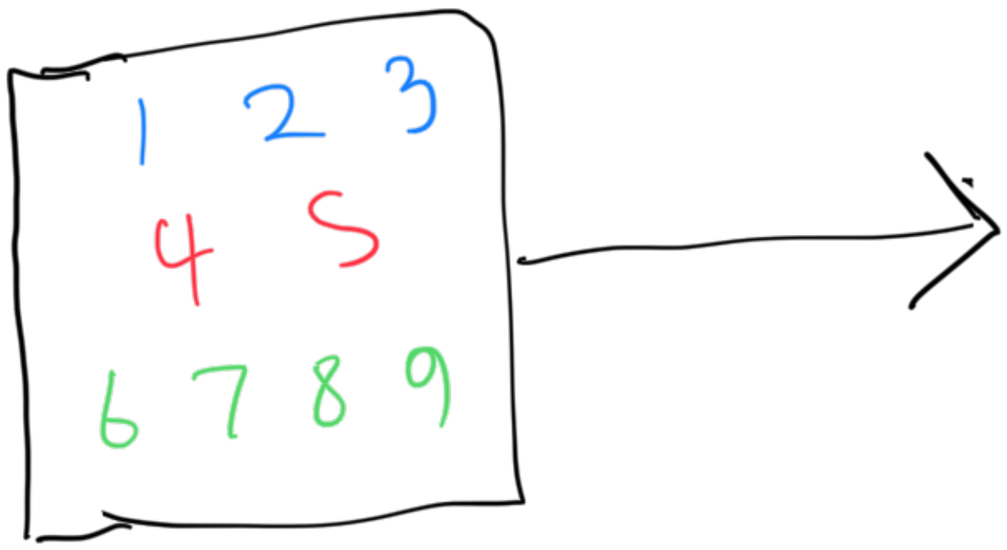
"Cancer"

predict "Cancer"

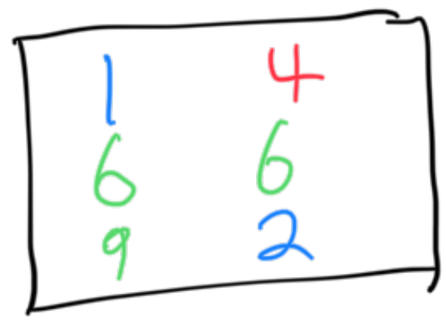
Bagging: bootstrapped aggregating

1) bootstrap (sample with replacement)

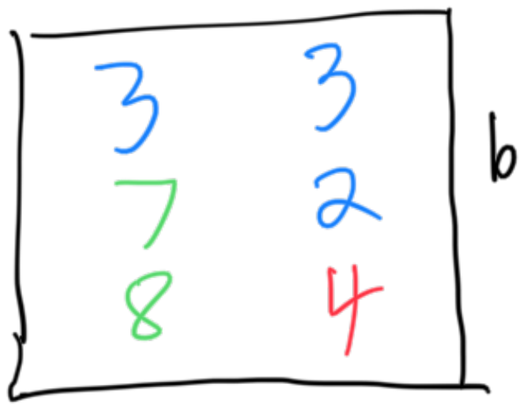
dataset into m "bootstrapped datasets"



original dataset

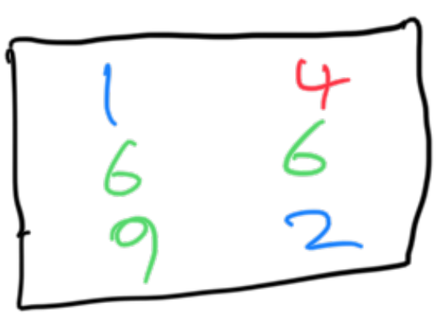


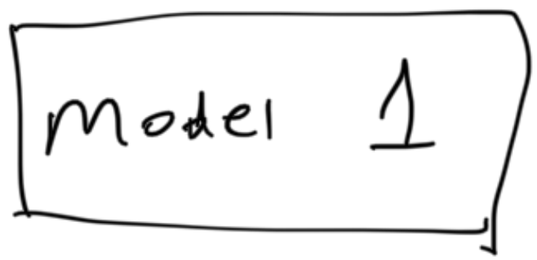
bootstrap 1



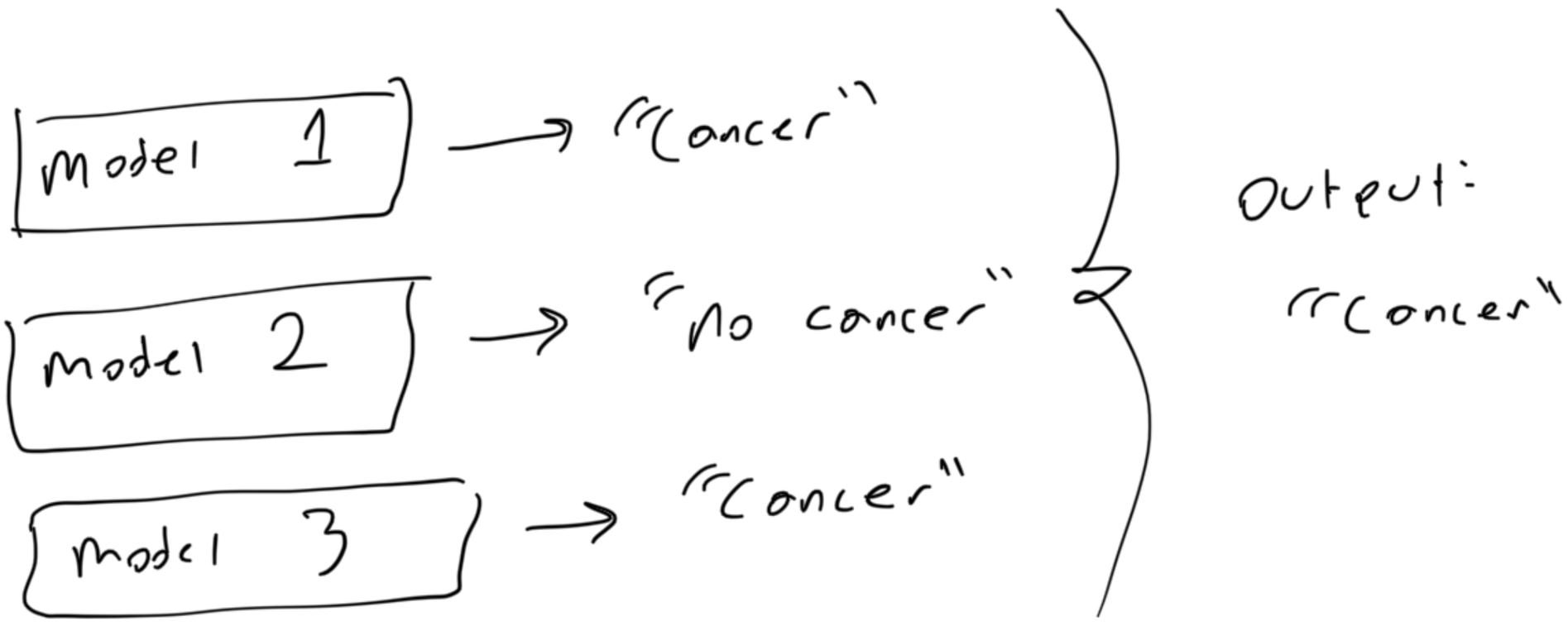
bootstrap m

(2) train a model (e.g., a decision tree) on each bootstrapped sample





(3) take majority vote of each of the models



When we apply bootstrapping to decision trees, we call that a "random forest".

But, can also bootstrap with other types of models:

logistic regression → "Cancer"

KNN → "No cancer"

decision tree → "Cancer"

Naive Bayes → "Cancer"

In addition to looking at a subset of data points, each model can also look at a subset of features.

Note: bootstrapped analyses are common in data Science and regular Science

Example:

Model A F1 score: 0.853

Model B F1 score: 0.892

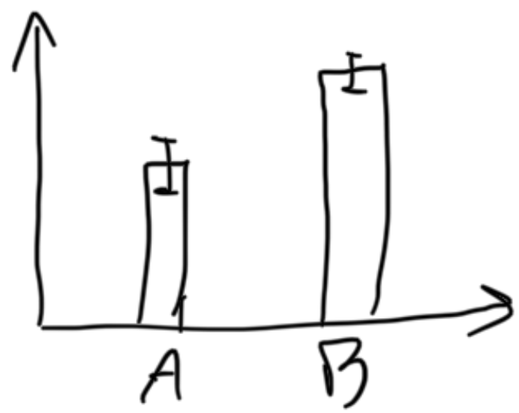
How do we know that this difference is not due to random chance?!!

Instead, we can bootstrap the test set and report mean +/- standard deviation of the bootstrapped datasets

Case 1

Model A: 0.853 +/- 0.0001

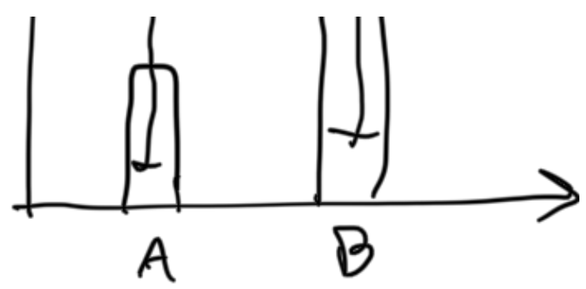
Model B: 0.892 +/- 0.0001



Case 2

Model A: 0.853 +/- 0.06

Model B: 0.892 +/- 0.01

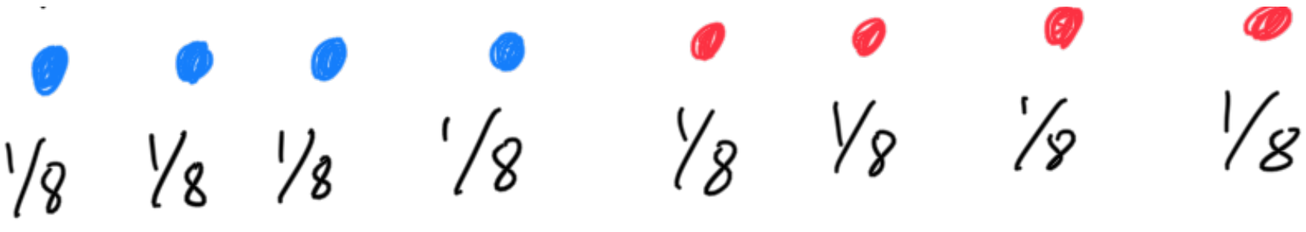


Boosting: iteratively fit models which prioritize misclassified data points in the previous iteration

AdaBoost (one boosting algorithm)

(1) initialize "weights" of each data point to be equal

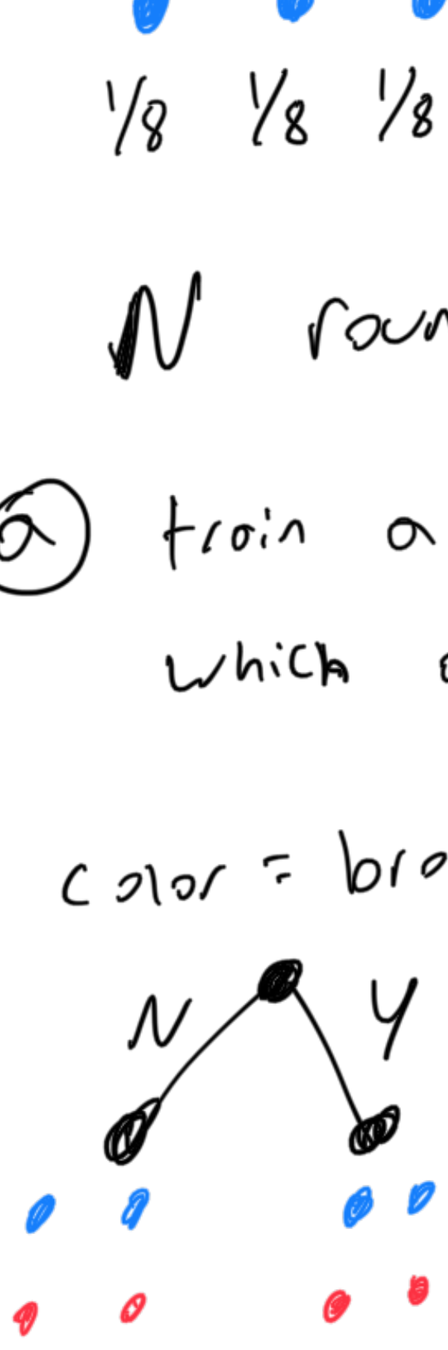
F F F F M M M M



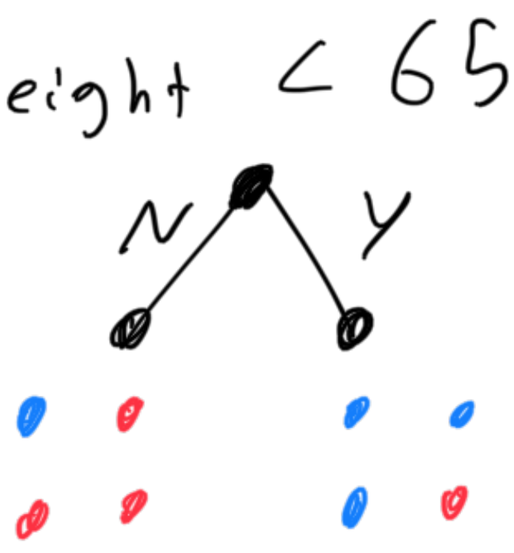
(2) for N rounds:

(a) train a model per feature which only uses that feature

color = brown?



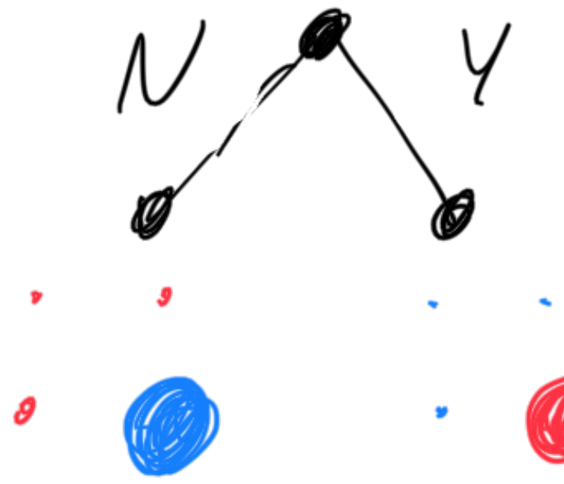
height < 65?



(b) Choose model with lowest weighted error, weighted by the weights of the data points

Select 'height < 65

(2) Update weights of each data point:
increase weight if incorrectly predicted,
decrease weight otherwise



(3) Final model is combination of
models from rounds 1 through N,
weighted based on the error
of that round

So, the final prediction is F