Day 11: Naive Bayes and Decision Trees

Parametric Learning: learn model parameters (e.g., linear and logistic regression)

Non-Parametric Learning: models that don't learn parameters (e.g., KNN and Naive Bayes)

Naive Bayes

Uses Bayes' Rule:

$$\underbrace{P(y}_{\text{label}}|\underbrace{x)}_{\text{data}} = \frac{P(x|y)P(y)}{P(x)}$$

To classify:

Estimate $P(y = \text{label} \mid x = \text{data})$ for

- Evaluate ... all possible labels
- Choose label with max probability

In practice, $P(x)$ doesn't make a difference, so we don't bother calculating:

$$\frac{P(x \mid y=dog)P(dog)}{\cancel{P(x)}} \quad \text{vs.} \quad \frac{P(x \mid y=cat)P(cat)}{\cancel{P(x)}}$$

Same denomenator, so can ignore

Note on HW3: this is what HW3 means when it says "only calc the numerator"

Using conditional independence assumption:

Using

$$P(X = x_1, x_2, x_3, x_4 \mid y)$$

$$= P(x_1 \mid y) P(x_2 \mid y) P(x_3 \mid y) P(x_4 \mid y)$$

$$\underbrace{P(X \mid y = dog)} P(y = dog)$$

$$= \underbrace{P(x_1 \mid dog) P(x_2 \mid dog) P(x_3 \mid dog) P(x_4 \mid dog)} P(dog)$$

To "train": Precompute probabilities

X_1	X_2	X_3	X_4	y
0	0	1	1	car
0	1	0	1	car
1	0	1	1	car
1	1	0	0	bike
1	0	1	0	bike

$P(y=car) = 3/5$ $P(y=bike) = 2/5$

$P(x_1 | car) = 1/3$ $P(x_1 | bike) = 1$

$P(x_2 | car) = 1/3$ $P(x_2 | bike) = 1/2$

...

Problem: what if $P(x_i | y) = 0$?

$$P(x_1|y) \cdot \ldots \cdot 0 \cdot \ldots \cdot P(x_n|y) \cdot P(y) = 0$$

The Fix: Laplace Smoothing

* add 1 to the numerator
* add (# classes) to the denomenator

# of outputs

Example: Predict between "spam", "maybe spam", and "definitely not spam":

$$P(x_i = \text{"ostrich"} \mid y = \text{"spam"}) = \frac{0 + 1}{(\#\text{spam}) + 3}$$

$$P(x_i = \text{"ostrich"} \mid y = \text{"maybe spam"}) = \frac{0 + 1}{(\#\text{maybe spam}) + 3}$$

...

Common variation:

use a smoothing hyper parameter B

- adding B to the numerator
- adding $B(\#\text{classes})$ to the denomenator

hyper

In HW3, B is the smoothing parameter

A practical consideration:

underflow: $0.0000000000\ldots 003 \to 0$

standard fix: maximize $\log(P(\cdots))$

$$\log(ab) = \log(a) + \log(b)$$

So, maximizing $P(y_i = c \mid X)$ is the same as maximizing $\log P(y_i = c \mid X)$

$$\log\left(\prod_{j=1}^{d}\left[P(X_{ij} \mid y_i = c)\right] P(y_i = c)\right)$$

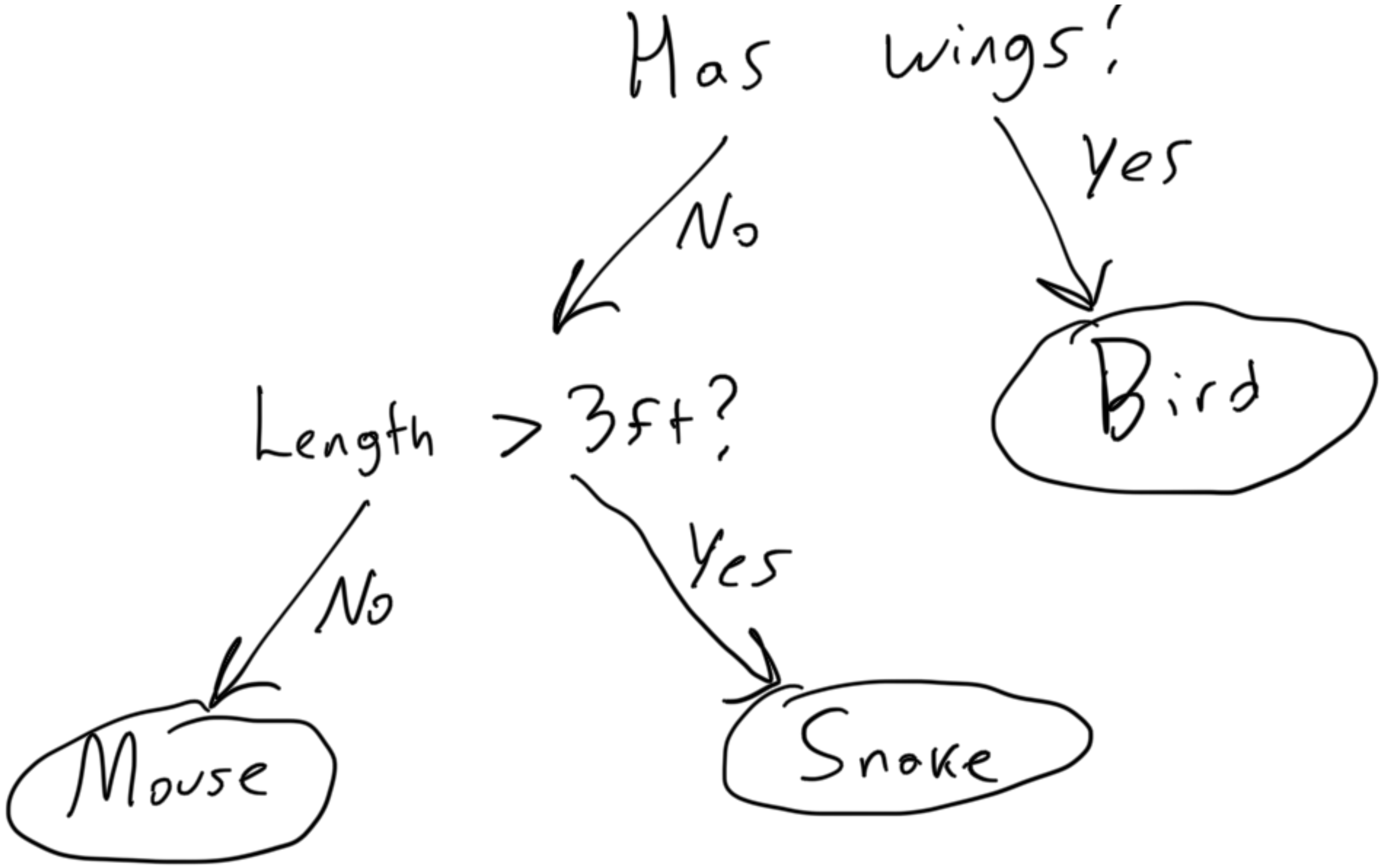
$$= \sum^{d} \log\left[P(X_{ij} \mid y_i = c)\right] + \log P(y_i = c)$$

$$= \sum_{j=1} \log[\ldots]$$

Not needed for HW3 (the log stuff), but this is always needed in the real world (when # features is large)

Decision Trees for Classification

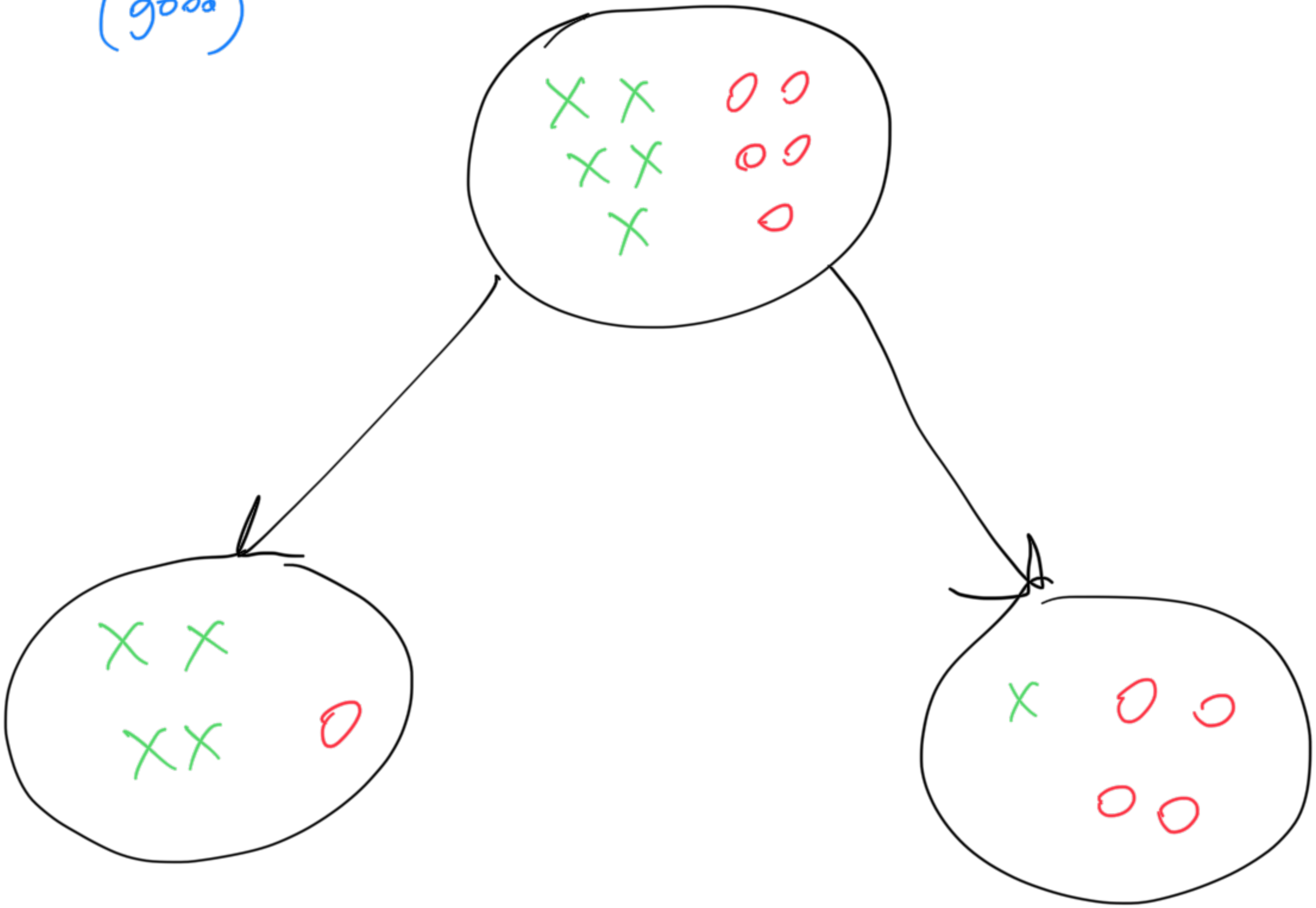
Animal Decision Tree Classifier:



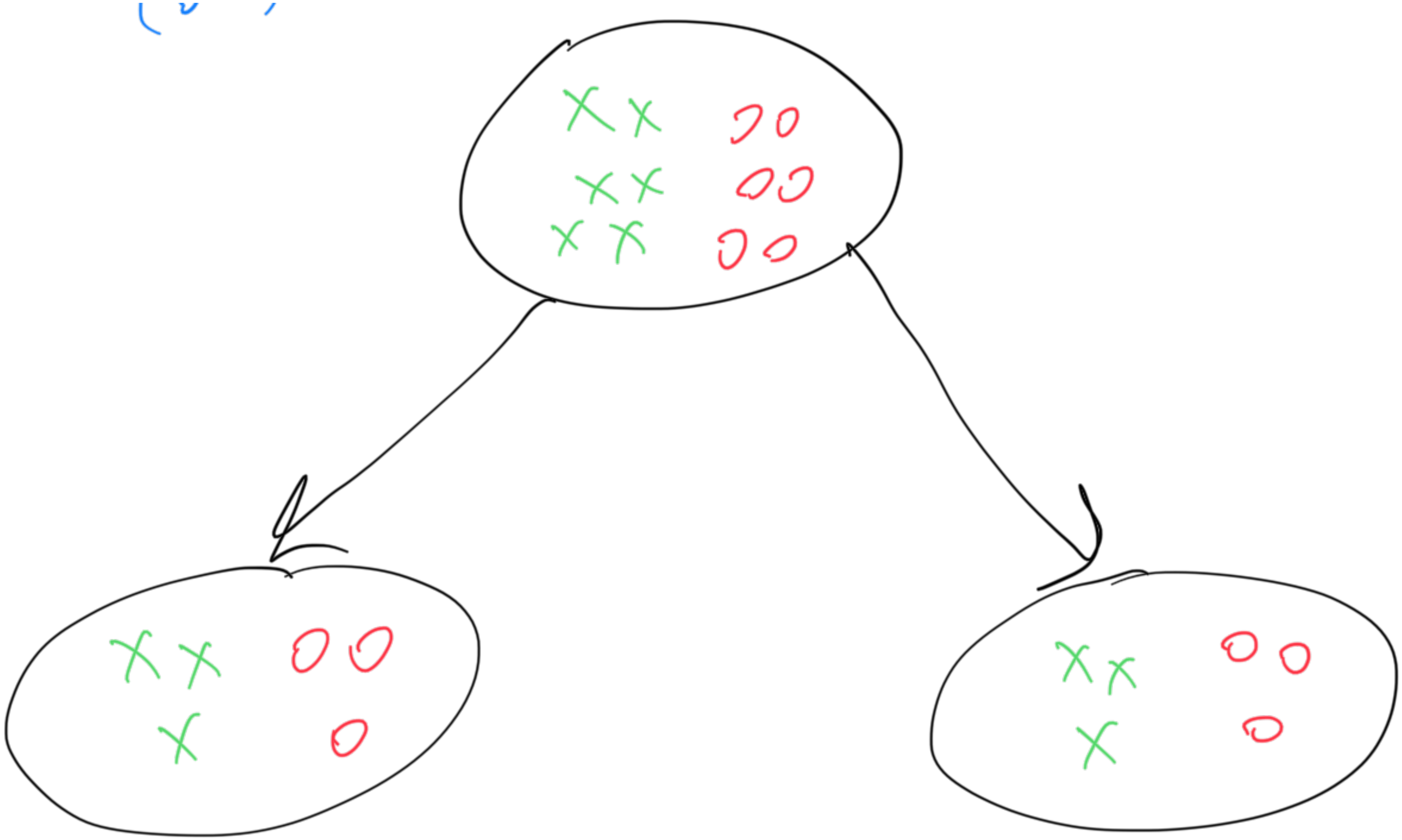
<u>Prediction</u>: go down the tree until you reach a leaf node

<u>"Training"</u>: many variants, but general idea: recursively split tree on input features with the largest "information gain"

High Information Gain:
(good)



Low Information Gain:
(Bad)



Basic Algorithm: ID3

"Iterative Dichotomiser 3"

ID3 (data D, features F):

1. calculate splitting metric for every feature $f \in F$ of dataset D
2. split D into subsets based on the splitting metric
3. make a node for the selected feature f which minimizes/maximizes the metric
4. recurse on the subsets using the non-selected features

Many possible splitting metrics;
one popular one:

Gini Impurity

$$G = 1 - \sum_{i=1}^{K} p_i^2 \, , \quad K = \text{\# of classes}, \quad p_i = \text{probability of being in class } i$$

(split on <u>lowest</u> impurity)

High impurity: same probability per class

$$1 - \underbrace{(0.5)^2}_{0.25} - \underbrace{(0.5)^2}_{0.25} = 0.5$$

Low impurity: high probability for one class, low probability for the other

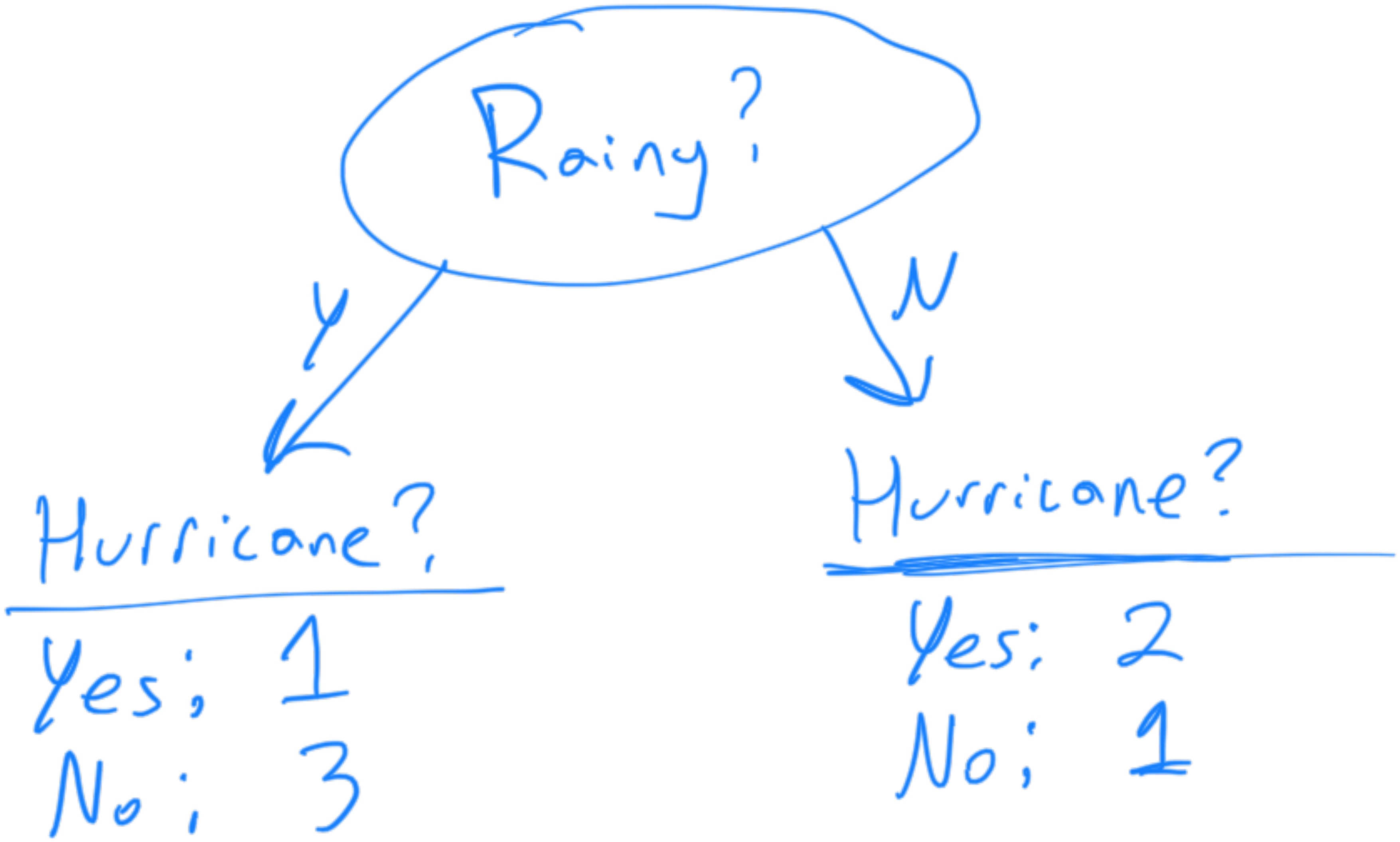
$$1 - \underbrace{(0.99)^2}_{0.9801} - \underbrace{(0.01)^2}_{0.0001} = 0.0198$$

Example (using Gini impurity as our splitting metric):

Rainy?	Humid?	Temp.	Hurricane?
Y	Y	7	No
Y	N	12	No
N	Y	18	Yes
N	Y	35	Yes
Y	Y	38	Yes
Y	N	50	No
N	N	83	No

First, see which input has the $\downarrow$ impurity:

Rainy



$$G(\text{Hurricane} = \text{yes})$$
$$= 1 - \left(\frac{1}{1+3}\right)^2 - \left(\frac{3}{1+3}\right)^2 = 0.375$$

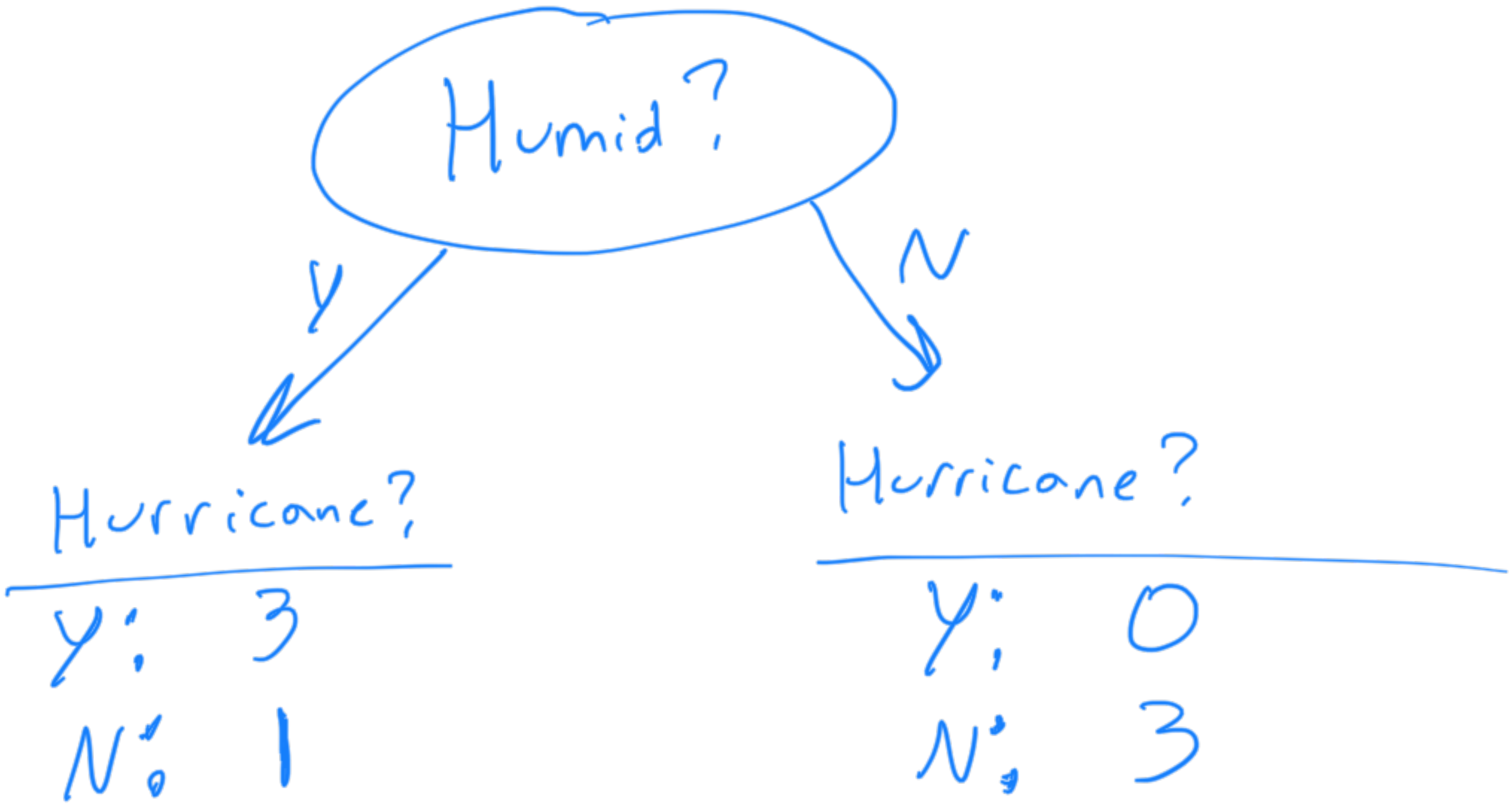
$$G(\text{Hurricane} = \text{no})$$
$$= 1 - \left(\frac{2}{2+1}\right)^2 - \left(\frac{1}{2+1}\right)^2 = 0.444$$

$$\left(\frac{4}{\;}\right)0.375 + \left(\frac{3}{4+3}\right)0.444$$

$G_{total} = \left(\frac{\ }{4+3}\right) \ldots \left(\ \right)$

$= 0.405$

Humid



...

$G_{total} = 0.214$

Temp.

Temp	Hurricane?
7	N
12	N
18	Y
35	Y
38	Y
50	N
83	N

Split	Gini
9.5	G = 0.429
15	G = 0.343
26.5	G = 0.476
36.5	G = 0.476
44	G = 0.343
66.5	G = 0.429

Temp. < 9.5 ?

Y →

Hurricane?
Y: 0

N →

Hurricane?
Y: 3
N: 3

N: 1 N:

$$G_{total} = \left(\frac{1}{1+6}\right) G_{temp<9.5} + \left(\frac{6}{1+6}\right) G_{temp \geq 9.5}$$

$$= 0.429$$

Deciding on the root node:

Since "Humid?" has lowest G,

"Humid?" is our root node:

R	T	Hur

R	T	Hur

Y	7	N
N	18	Y
N	35	Y
Y	38	Y

(Recursively)
Run through the same process but on this smaller sub-dataset

Y	12	N
Y	50	N
N	83	N

100% of the data points have "Hurricane = no", so stop splitting
(make this a leaf node)