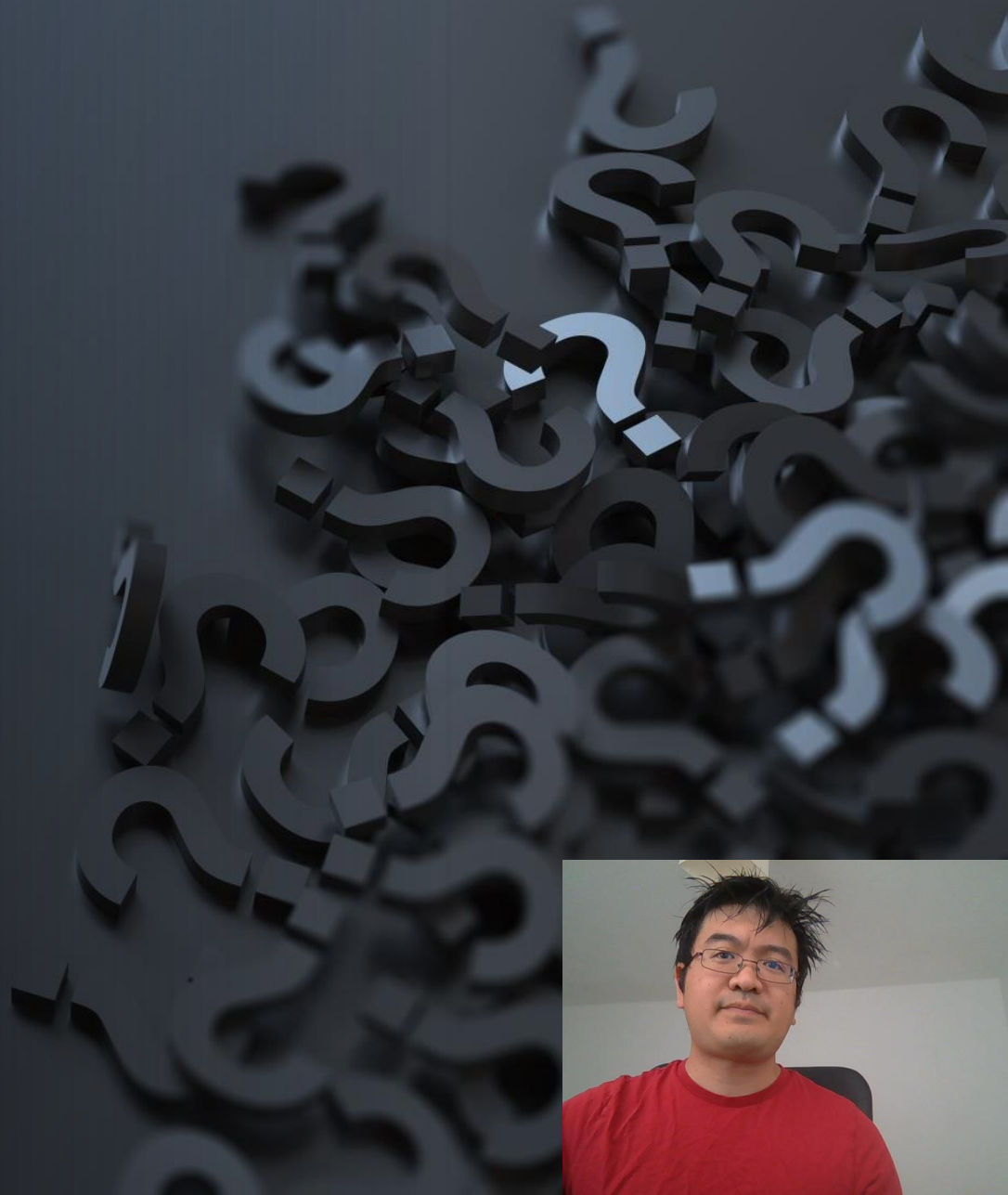

Decision Trees
Part 1: Introduction

Decision Trees

- New type of **non-linear model**
- Copes naturally with continuous and **categorical data**
- **Fast** to both train and test (highly parallelizable)
- Generates a set of **interpretable** rules

20 questions

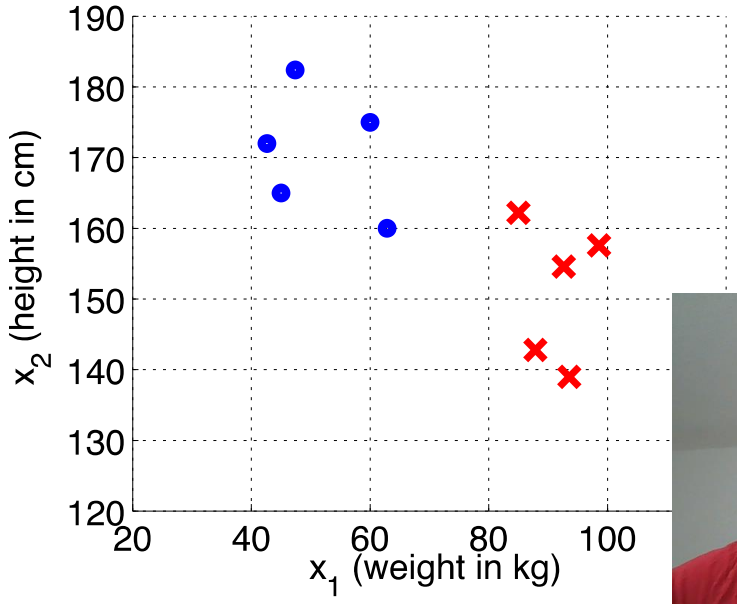
http://20q.net/



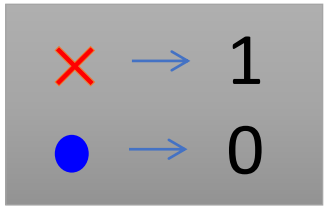
Decision Stump

Distinguish rugby players from ballet dancers.

x_1,	x_2,	y (label)
98.79,	157.59,	1
93.64,	138.79,	1
42.89,	171.89,	0
...		
...		
87.91,	142.65,	1
97.92,	162.12,	1
47.63,	182.26,	0
92.72,	154.50,	1



A simple decision tree

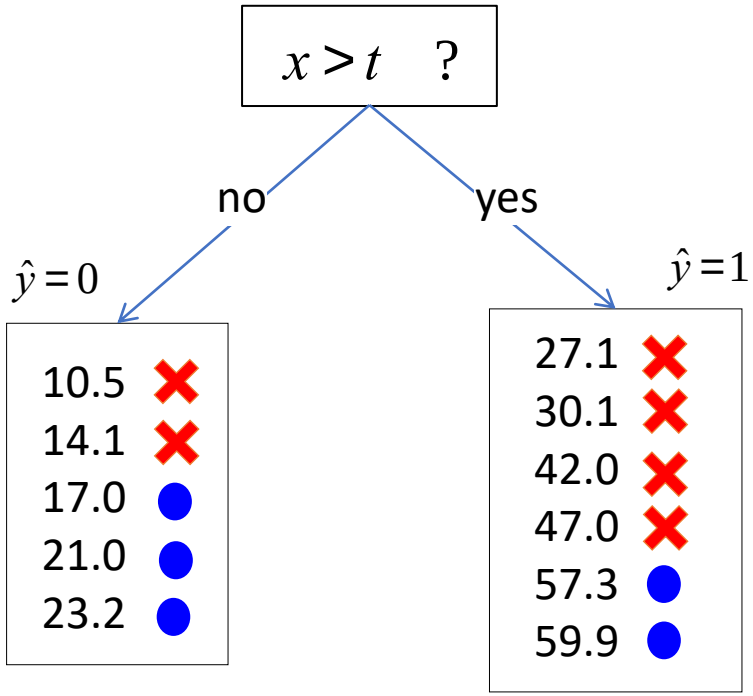


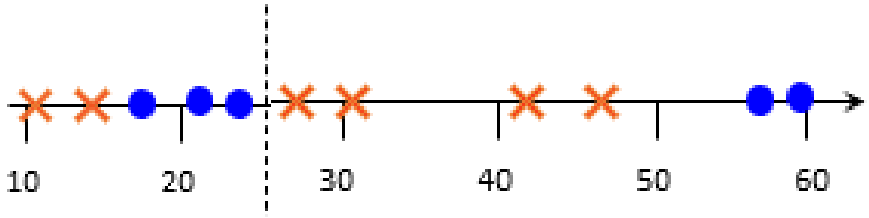
if $x > t$ then $\hat{y} = 1$ else $\hat{y} = 0$

A simple decision tree

× → 1

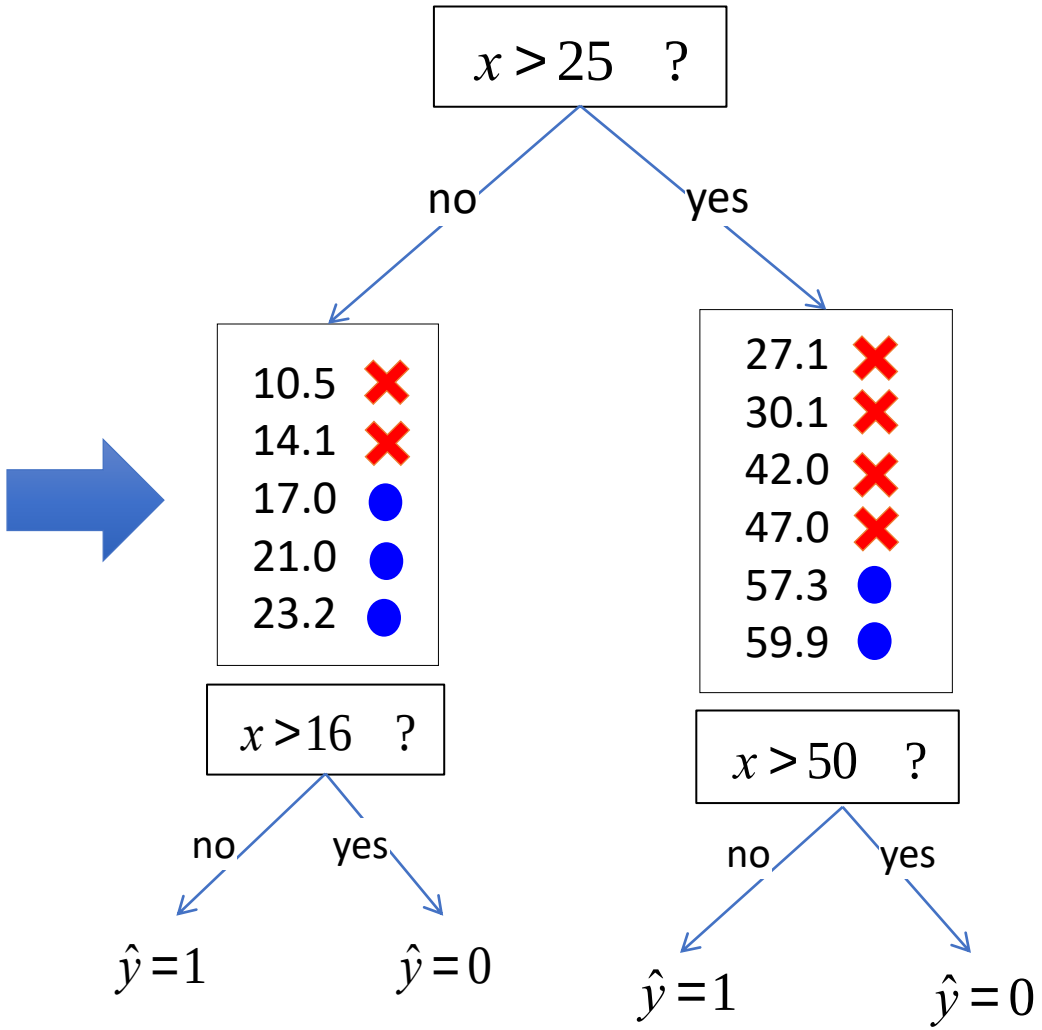
● → 0



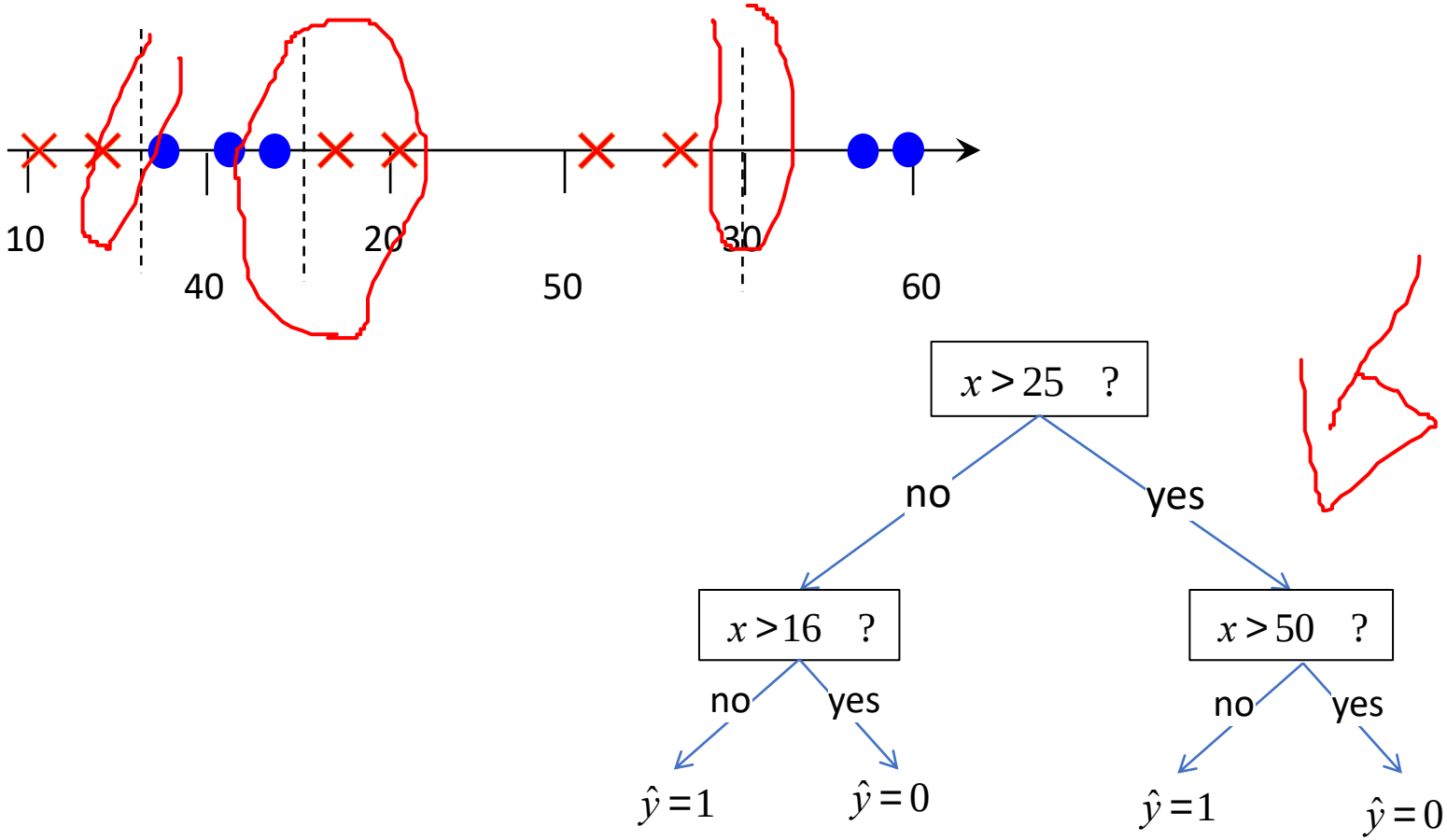


Just another dataset!

Build a stump!

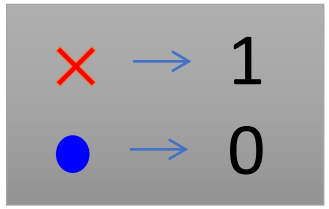


Decision Trees = nested rules




```
if x>25 then
        if x>50 then y=0 else y=1; endif
else
        if x>16 then y=0 else y=1; endif
endif
```

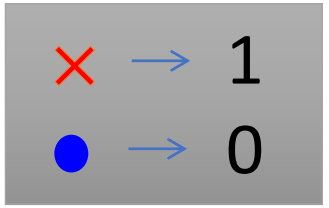
Challenge 1: selecting a good threshold



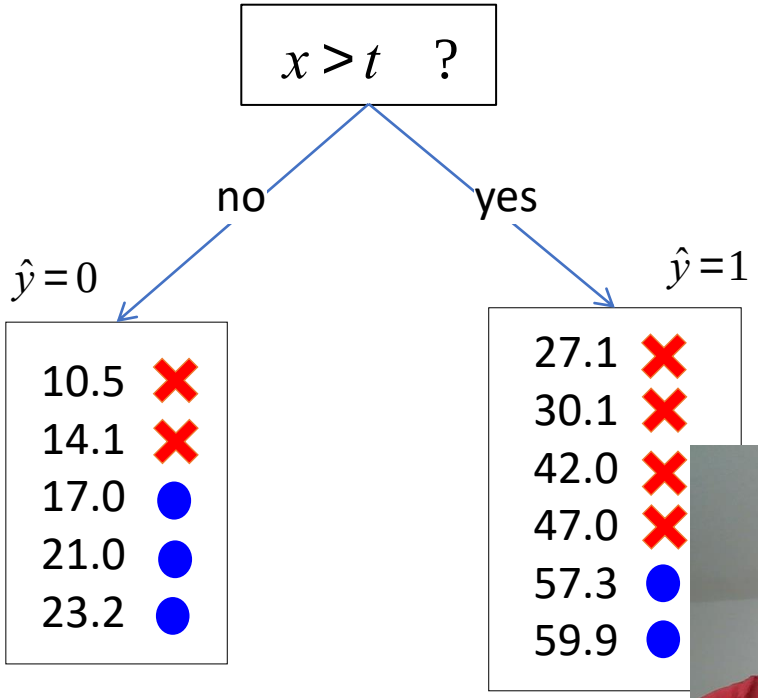
if $x > t$ then $\hat{y} = 1$ else $\hat{y} = 0$

Q. Where is a good threshold?

Challenge 1: selecting a good threshold



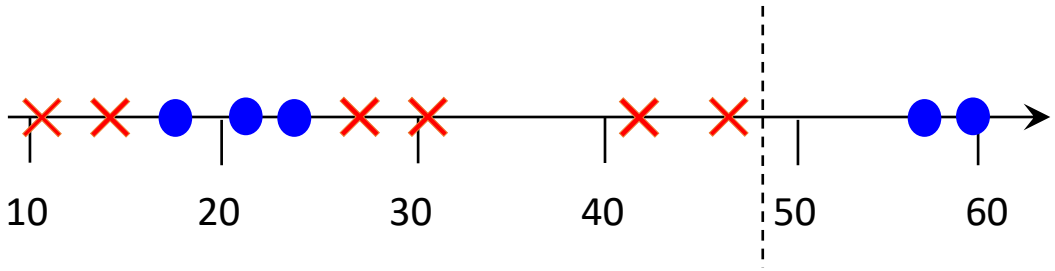
if $x > t$ then $\hat{y} = 1$ else $\hat{y} = 0$

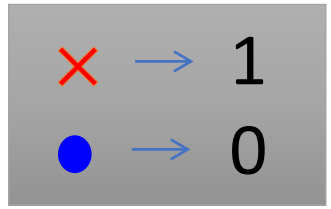


The stump "splits" the dataset.

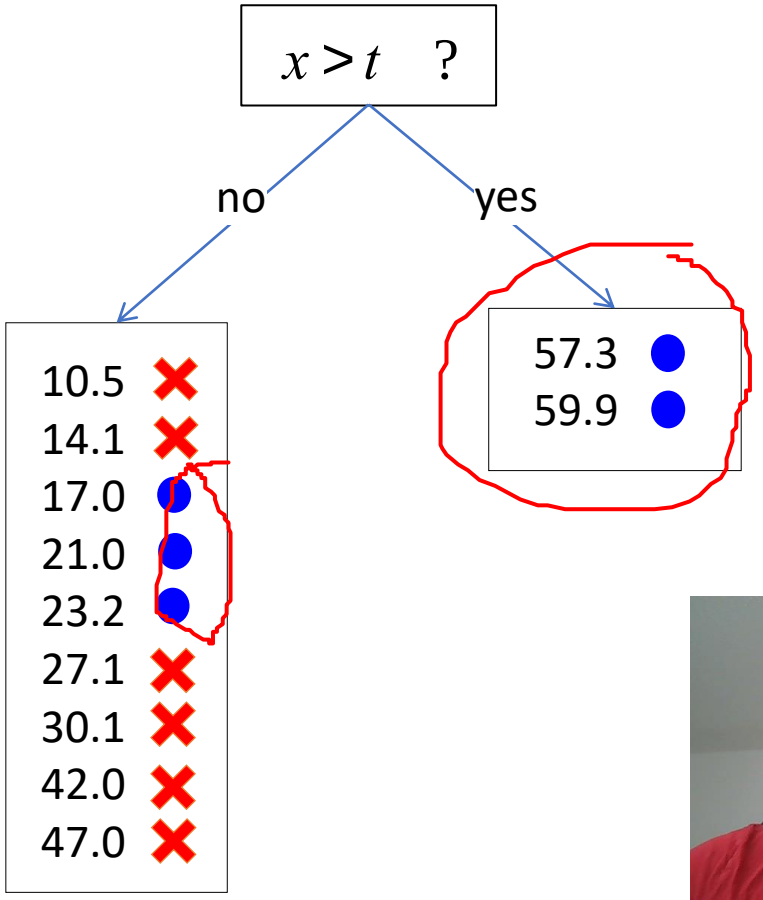
Here we have 4 classification errors.

Challenge 1: selecting a good threshold





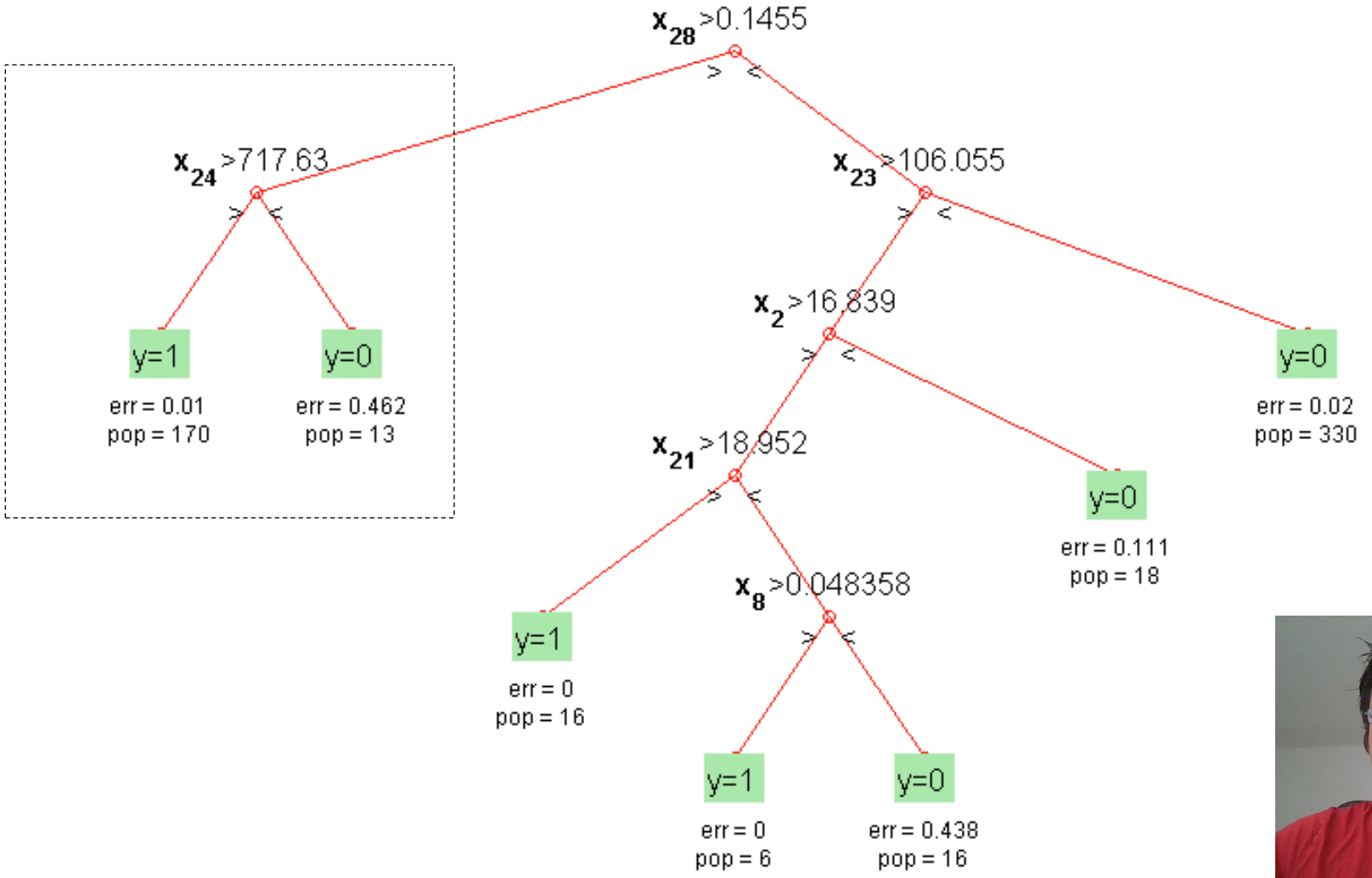
if $x > t$ then $\hat{y} = 1$ else $\hat{y} = 0$



Here we have 3 classification errors.

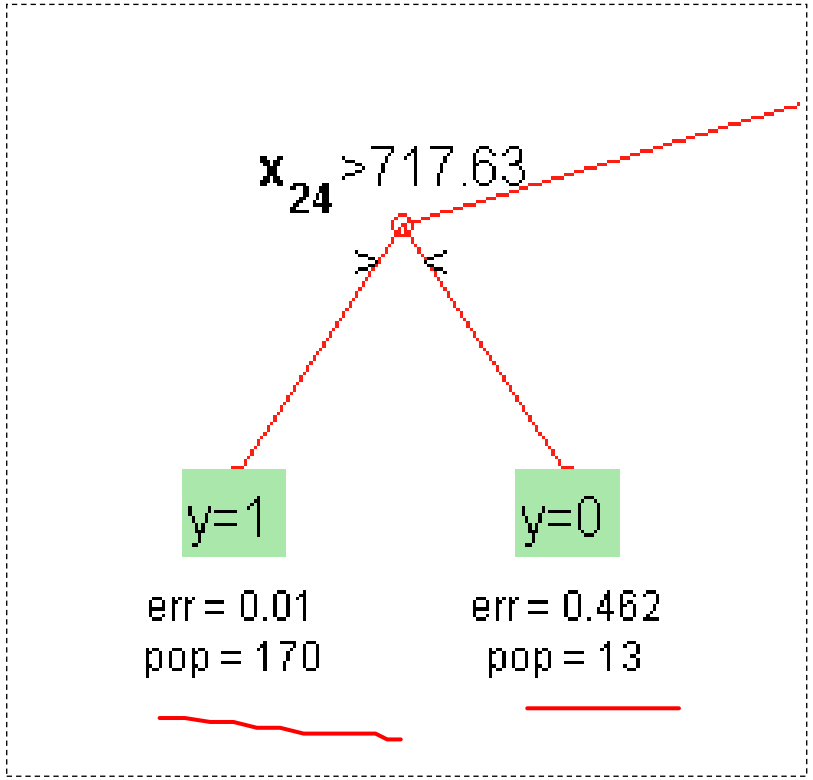
Challenge 1: selecting a good threshold

Here's one I made earlier…..



‘**pop**’ is the number of training points that arrived at that node.

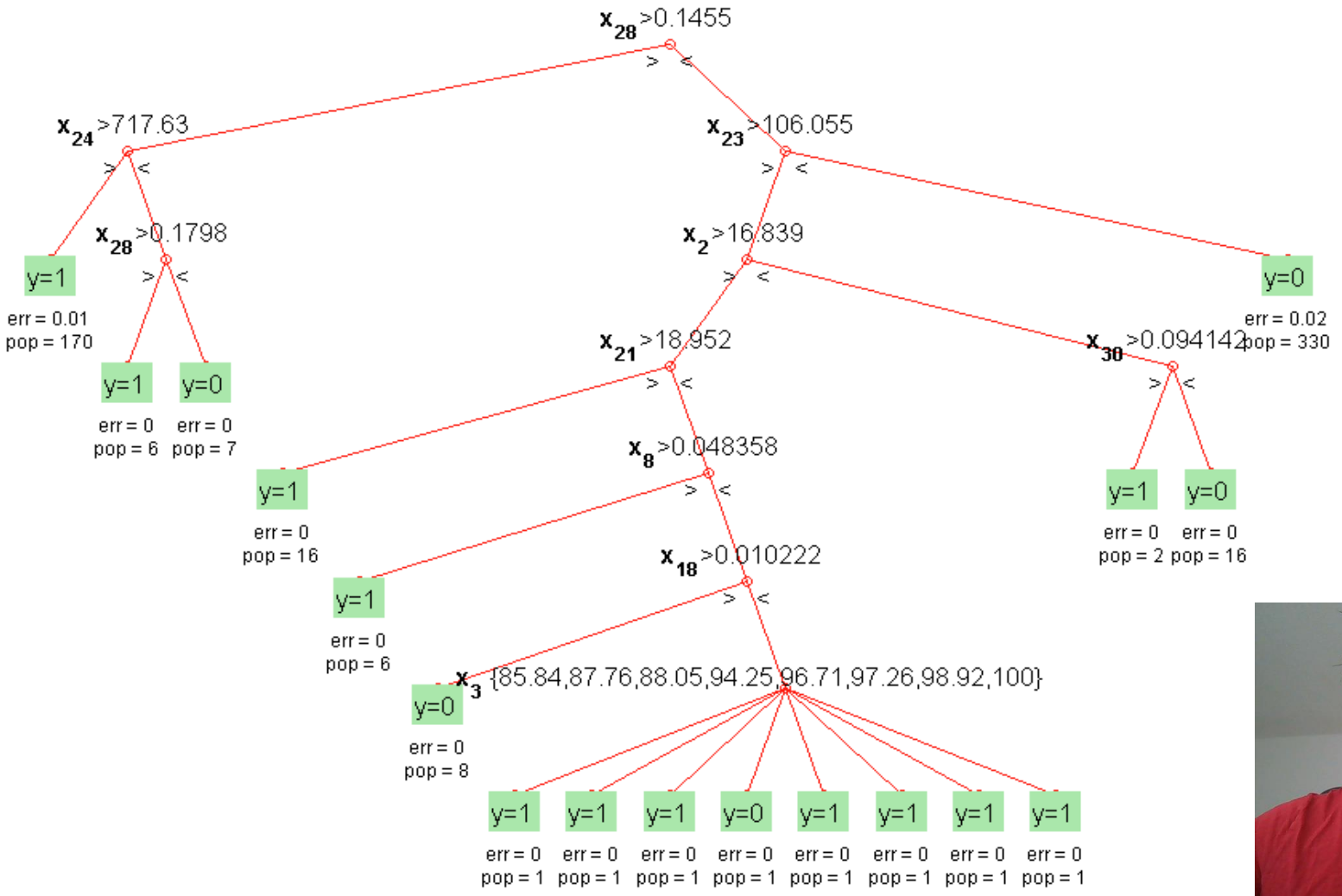
‘**err**’ is the fraction of those examples incorrectly classified.



Challenge 2: how complex should I make the tree?

Increasing the maximum depth (10)

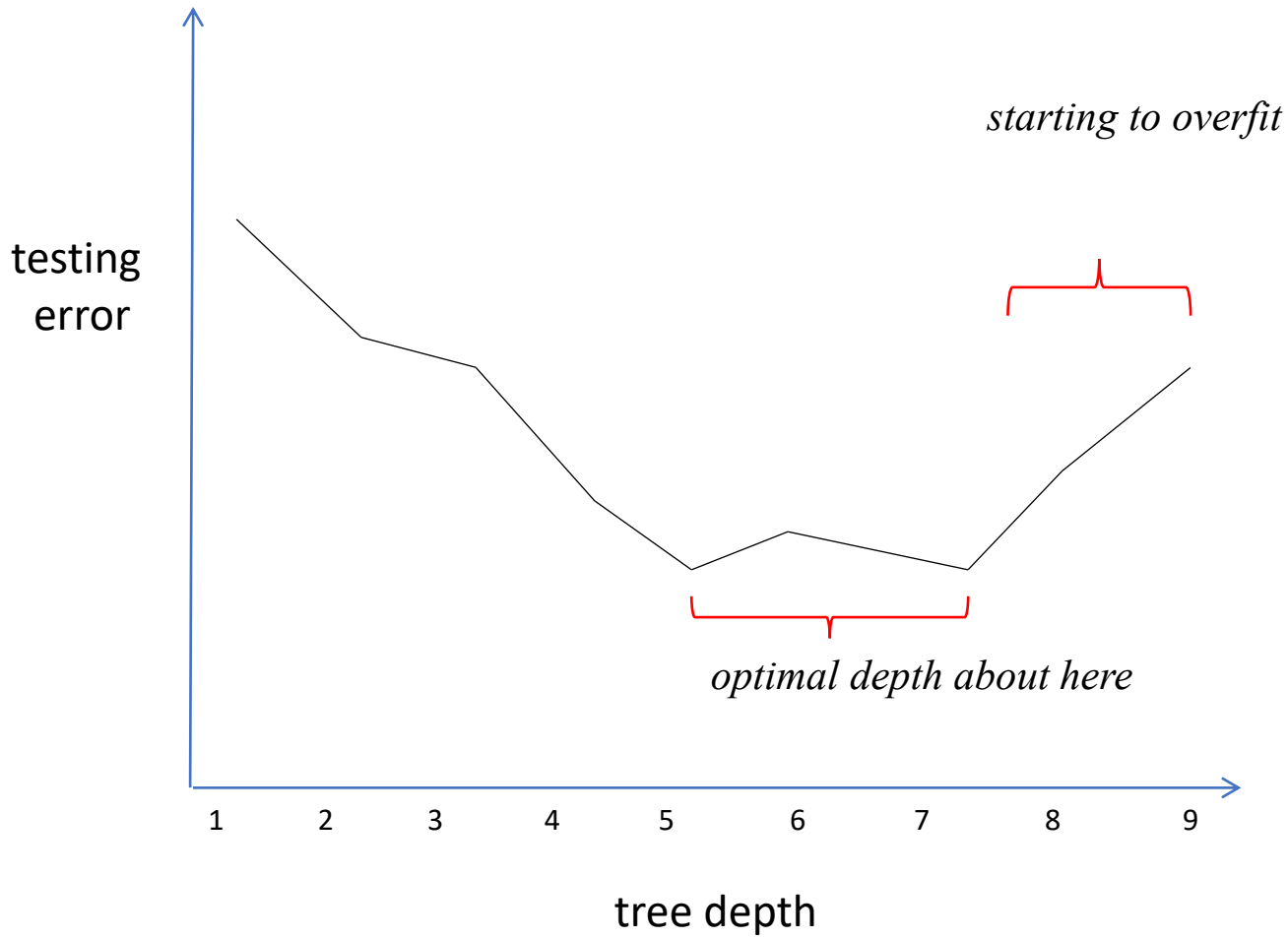
Decreasing the minimum number of examples required to make a split (5)



- The number of possible paths tells you the number of rules.
- More rules = more complicated.
- We could have N rules where N is the size of the dataset. This would mean no generalisation outside of the training data, or the tree is *overfitted*

Overfitting = fine tuning

Overfitting….



Decision Tree Learning Algorithm (sometimes called "ID3")

```
1: function BUILDTREE( subsample, depth )
2:
3:     //BASE CASE:
4:     if (depth == 0) OR (all examples have same label) then
5:         return most common label in the subsample
6:     end if
7:
8:     //RECURSIVE CASE:
9:     for each feature do
10:        Try splitting the data (i.e. build a decision stump)
11:        Calculate the cost for this stump
12:    end for
13:    Pick feature with minimum cost
14:
15:    Find left/right subsamples
16:    Add left branch ← BUILDTREE( leftSubSample, depth − 1 )
17:    Add right branch ← BUILDTREE( rightSubSample, depth − 1 )
18:
19:    return tree
20:
21: end function
```

Decision Trees

- New type of **non-linear model**
- Copes naturally with continuous and **categorical data**
- **Fast** to both train and test (highly parallelizable)
- Generates a set of **interpretable** rules