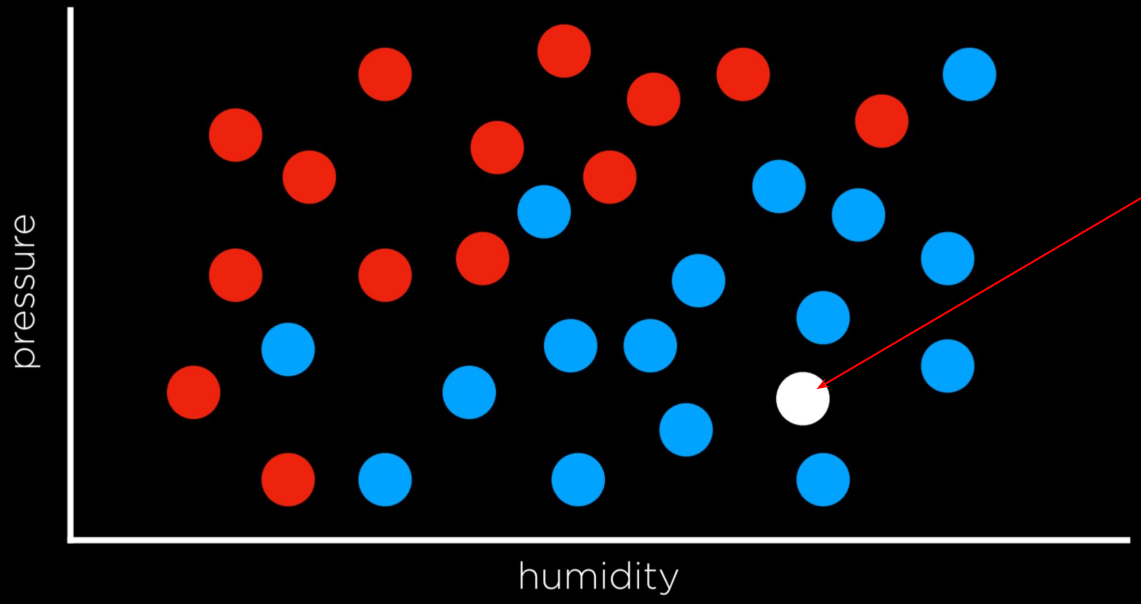
Decision Trees

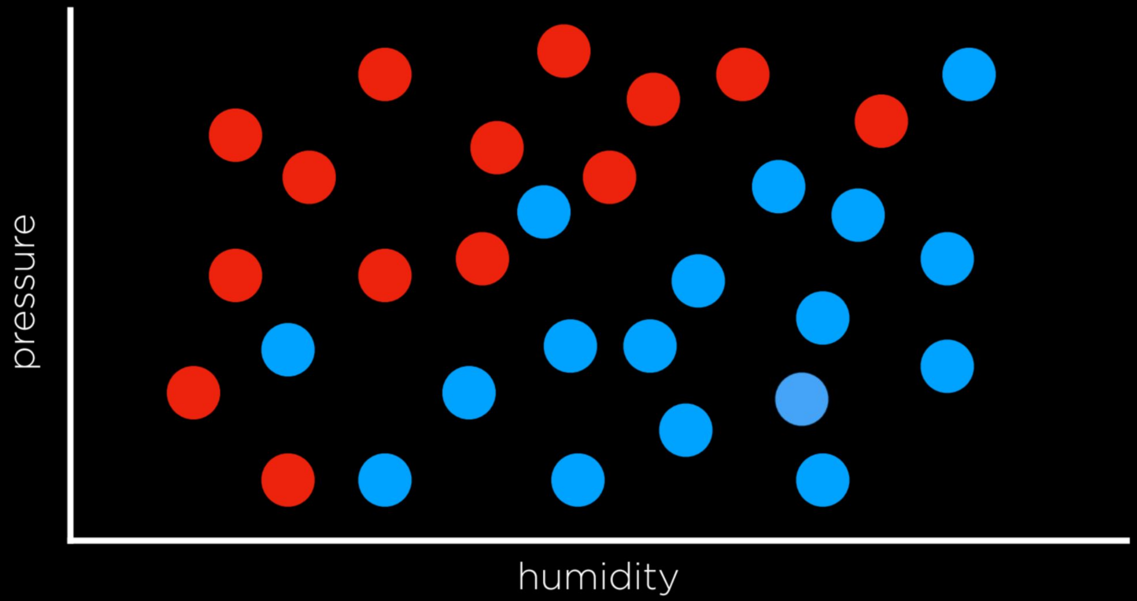
Chris made these slides :D Thanks Chris! 💀💀💀💀💀💀💀💀💀

Decision Trees are surprisingly human...



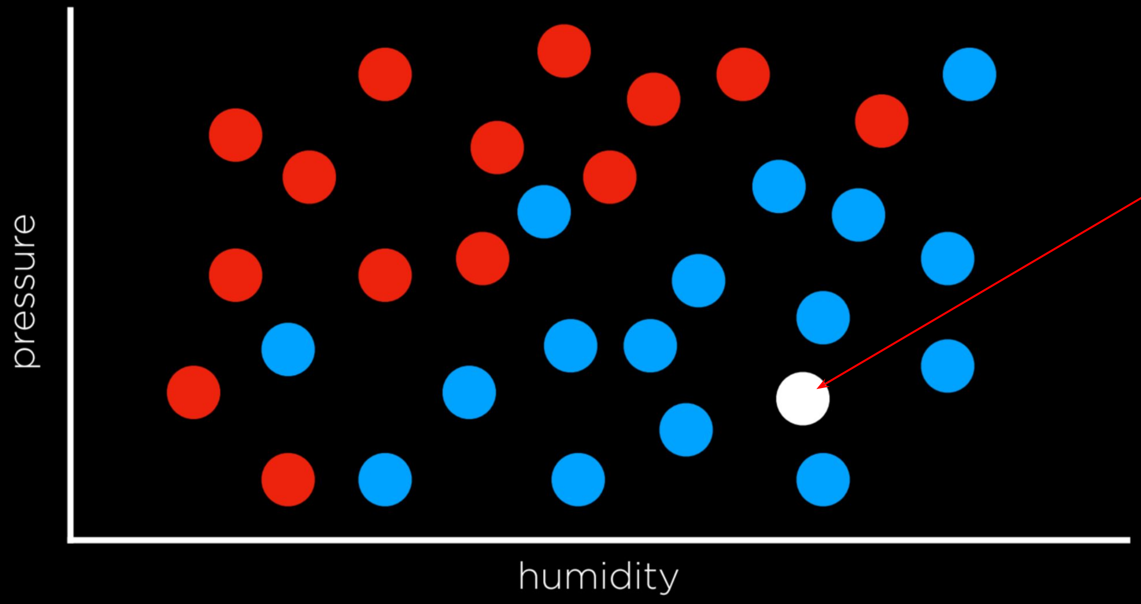
How would you classify the grey dot?

Decision Trees are surprisingly human…



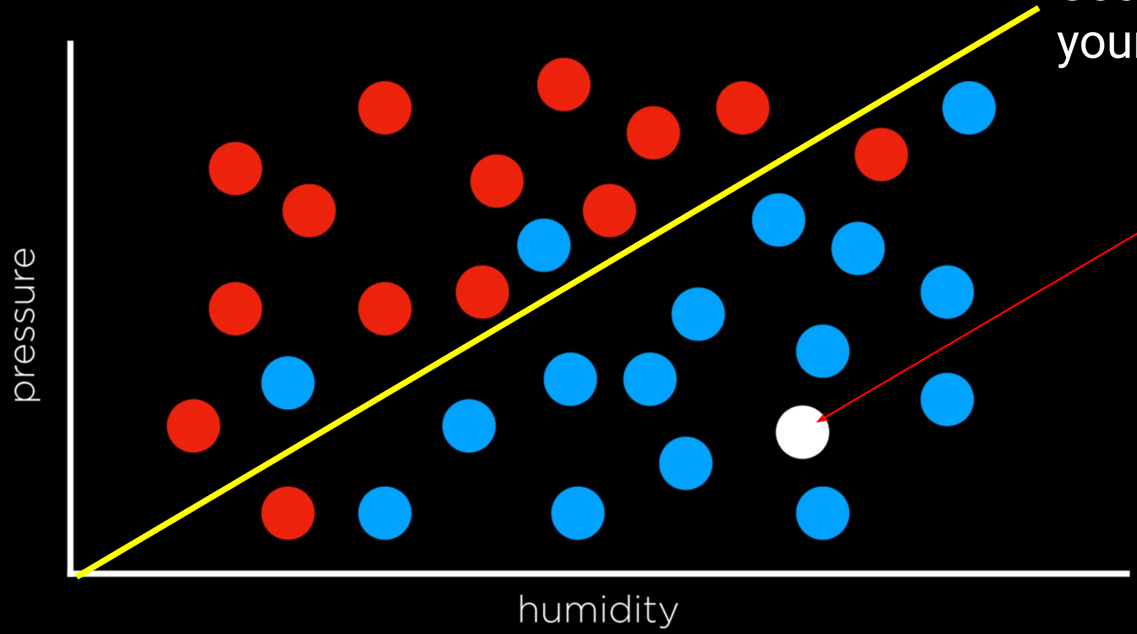
I would probably make it blue…

Decision Trees are surprisingly human...



But talk me through the process—why did we pick blue for this spot?

Decision Trees are surprisingly human…



Use this fit line in your explanation.

But talk me through the process—why did we pick blue for this spot?

If… then

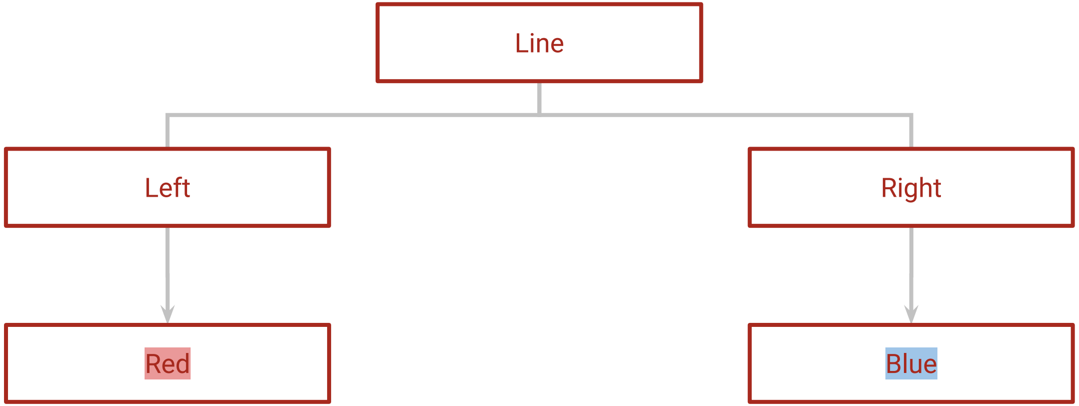
You probably thought something like:

“If the point is to the left of the yellow line, it is red.
If it is to the right, then it is blue.”

This is a selection statement. A decision, mayhaps.

Constructing a decision tree

We can construct rather simple decision trees like this:

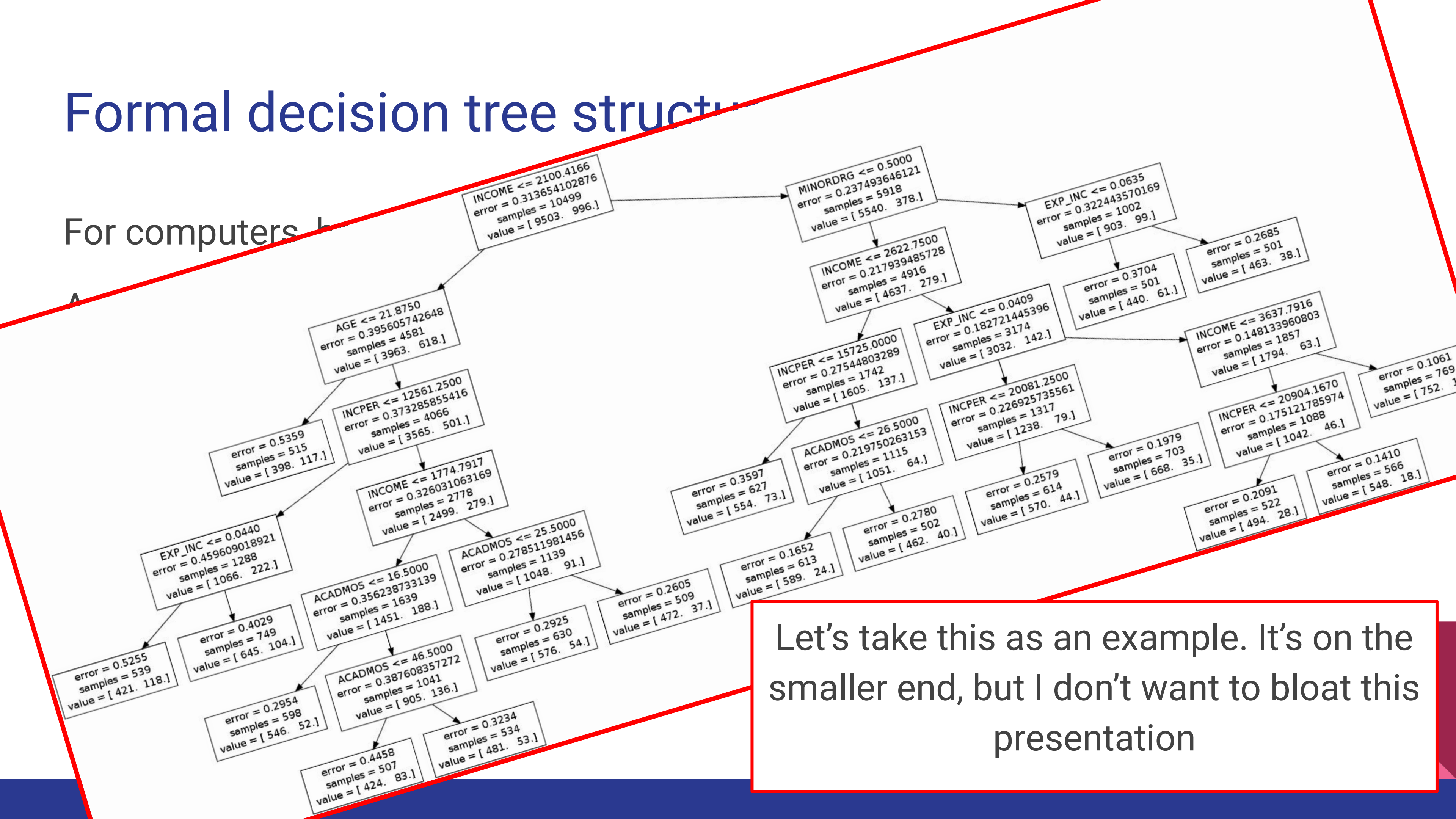


For the sake of communicating ideas, this is enough.

Formal decision tree structu

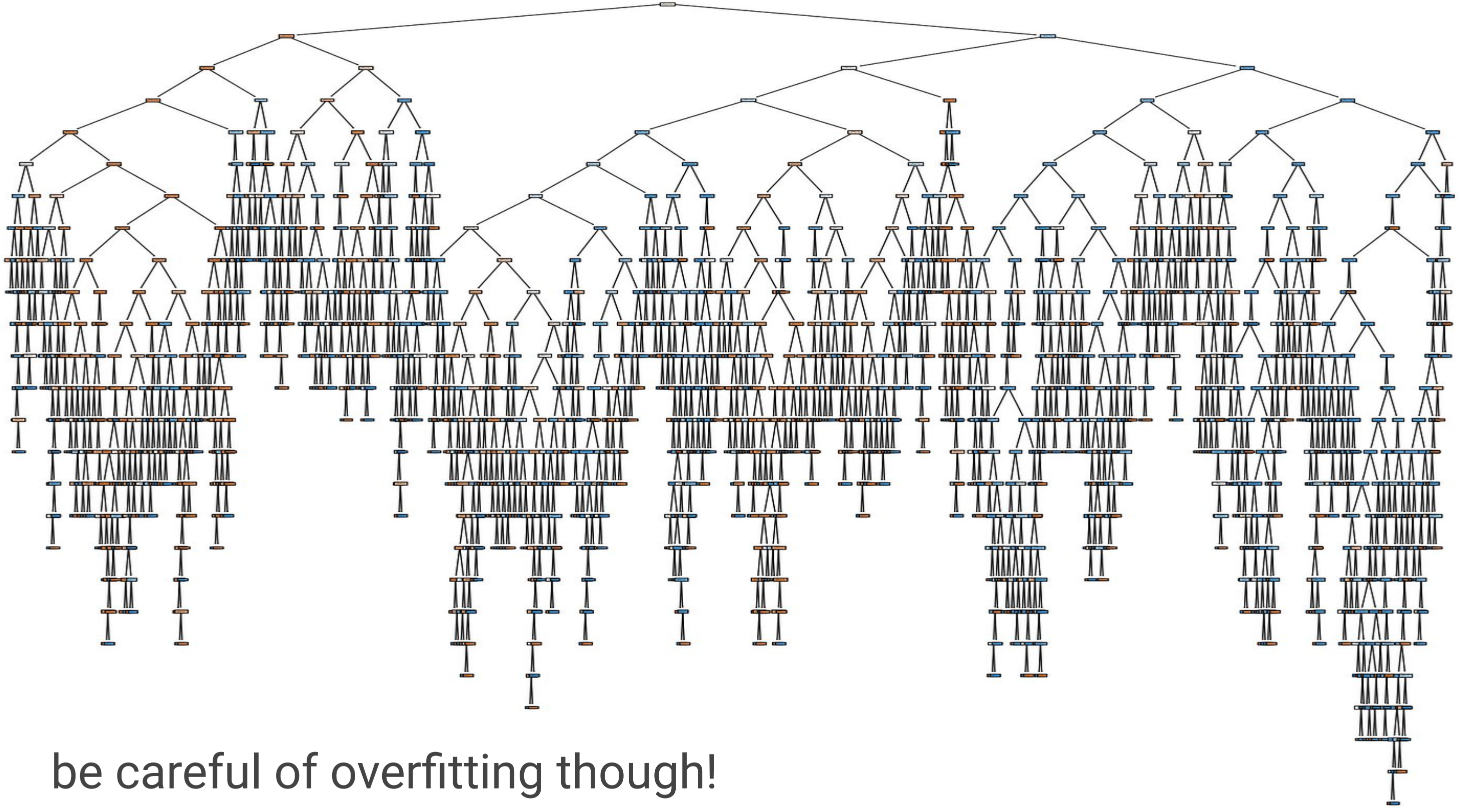
For computers, b

A



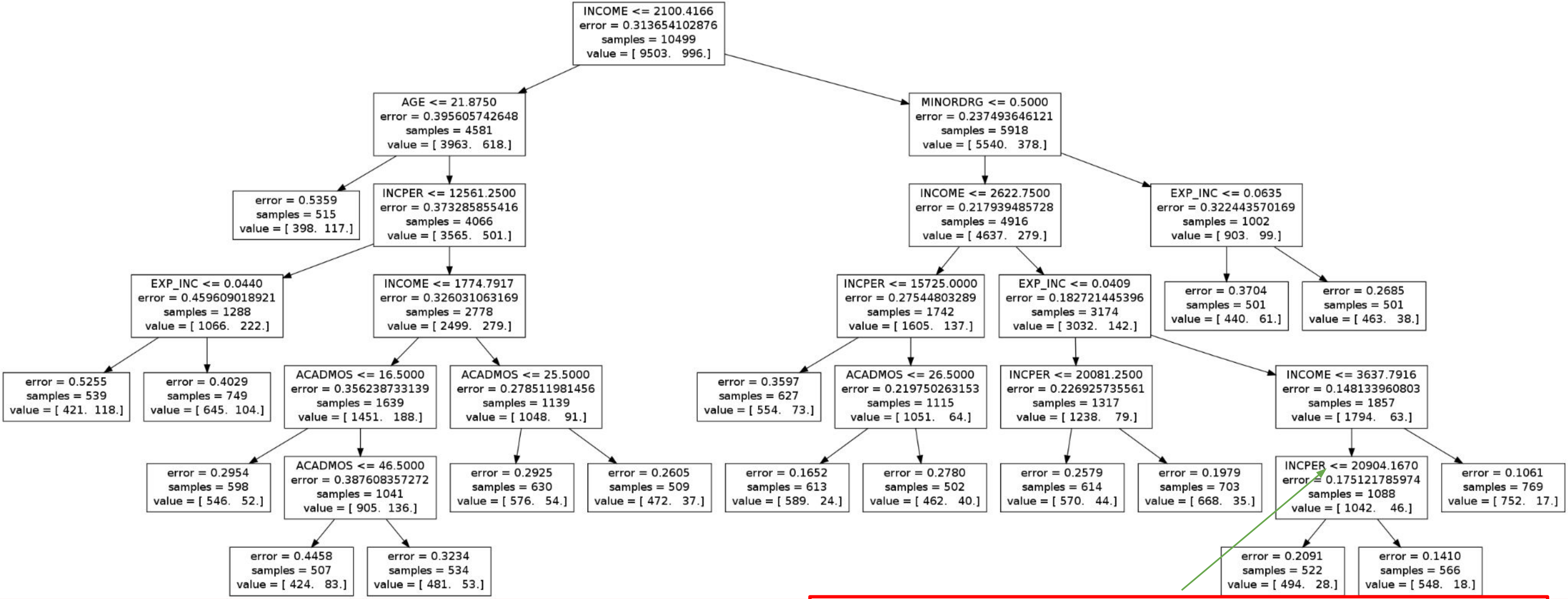
Let’s take this as an example. It’s on the smaller end, but I don’t want to bloat this presentation

A “normal” decision tree



be careful of overfitting though!

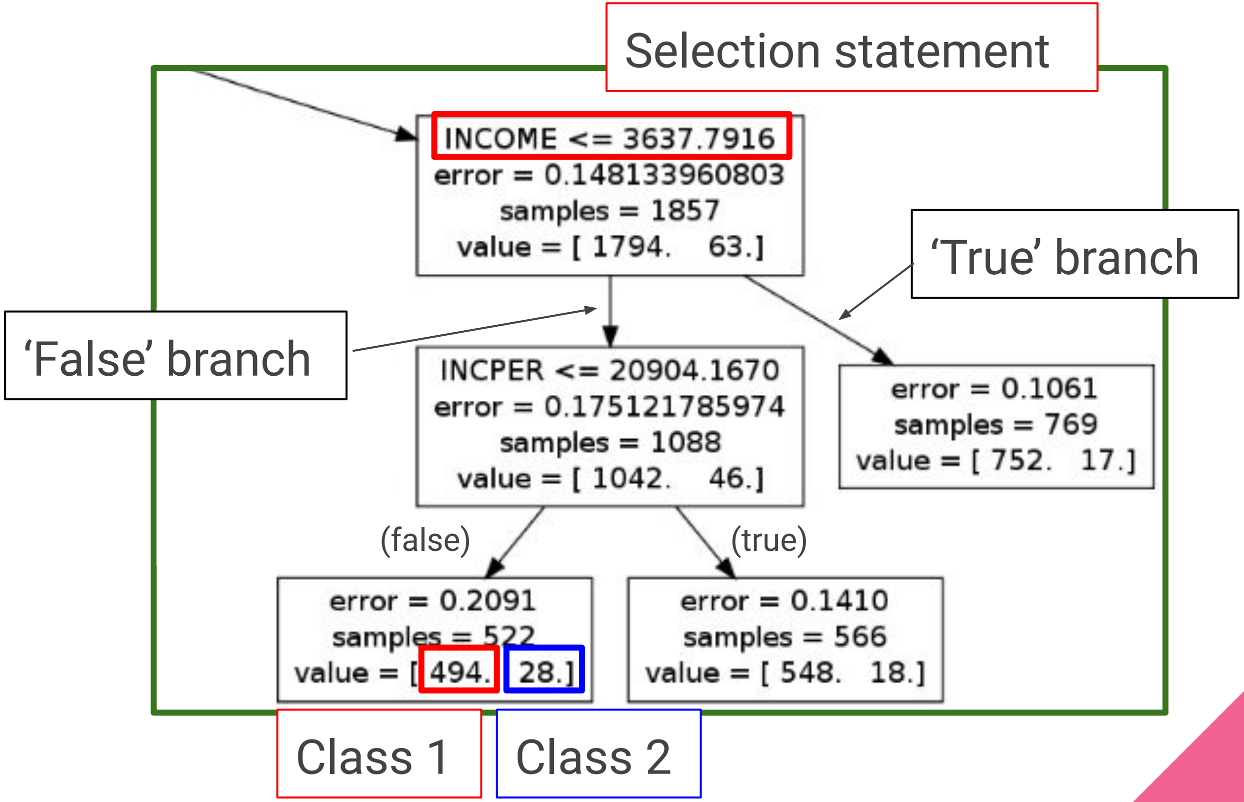
Oh shit what is this?!?!?!?!?!?!?



```
INCOME <= 2100.4166
error = 0.313654102876
samples = 10499
value = [ 9503.   996.]

AGE <= 21.8750
error = 0.395605742648
samples = 4581
value = [ 3963.   618.]

MINORDRG <= 0.5000
error = 0.237493646121
samples = 5918
value = [ 5540.   378.]

error = 0.5359
samples = 515
value = [ 398.  117.]

INCPER <= 12561.2500
error = 0.373285855416
samples = 4066
value = [ 3565.   501.]

INCOME <= 2622.7500
error = 0.217939485728
samples = 4916
value = [ 4637.   279.]

EXP_INC <= 0.0635
error = 0.322443570169
samples = 1002
value = [ 903.   99.]

EXP_INC <= 0.0440
error = 0.459609018921
samples = 1288
value = [ 1066.   222.]

INCOME <= 1774.7917
error = 0.326031063169
samples = 2778
value = [ 2499.   279.]

INCPER <= 15725.0000
error = 0.27544803289
samples = 1742
value = [ 1605.   137.]

EXP_INC <= 0.0409
error = 0.182721445396
samples = 3174
value = [ 3032.   142.]

error = 0.3704
samples = 501
value = [ 440.   61.]

error = 0.2685
samples = 501
value = [ 463.   38.]

error = 0.5255
samples = 539
value = [ 421.  118.]

error = 0.4029
samples = 749
value = [ 645.  104.]

ACADMOS <= 16.5000
error = 0.356238733139
samples = 1639
value = [ 1451.   188.]

ACADMOS <= 25.5000
error = 0.278511981456
samples = 1139
value = [ 1048.    91.]

error = 0.3597
samples = 627
value = [ 554.   73.]

ACADMOS <= 26.5000
error = 0.219750263153
samples = 1115
value = [ 1051.    64.]

INCPER <= 20081.2500
error = 0.226925735561
samples = 1317
value = [ 1238.    79.]

INCOME <= 3637.7916
error = 0.148133960803
samples = 1857
value = [ 1794.    63.]

error = 0.2954
samples = 598
value = [ 546.   52.]

ACADMOS <= 46.5000
error = 0.387608357272
samples = 1041
value = [ 905.  136.]

error = 0.2925
samples = 630
value = [ 576.   54.]

error = 0.2605
samples = 509
value = [ 472.   37.]

error = 0.1652
samples = 613
value = [ 589.   24.]

error = 0.2780
samples = 502
value = [ 462.   40.]

error = 0.2579
samples = 614
value = [ 570.   44.]

error = 0.1979
samples = 703
value = [ 668.   35.]

INCPER <= 20904.1670
error = 0.175121785974
samples = 1088
value = [ 1042.    46.]

error = 0.1061
samples = 769
value = [ 752.   17.]

error = 0.4458
samples = 507
value = [ 424.   83.]

error = 0.3234
samples = 534
value = [ 481.   53.]

error = 0.2091
samples = 522
value = [ 494.   28.]

error = 0.1410
samples = 566
value = [ 548.   18.]
```

Let’s focus on this part

Tree's Anatomy



Selection statement

'True' branch

'False' branch

Class 1

Class 2

Some notes

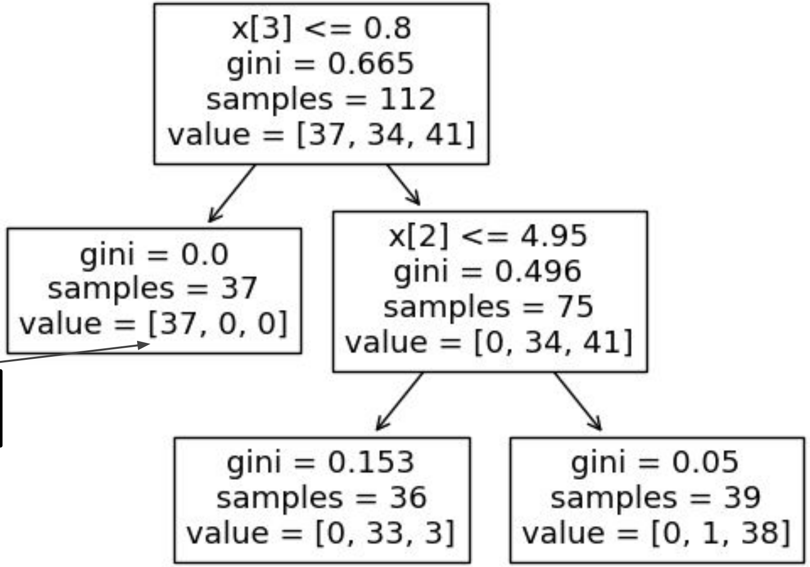
You may notice that it doesn’t actually say what the decision made is (which class it is put into). That’s because… classification is ambiguous. It could be:

- Always guessing the one with the higher incidence (bad)
- Guessing with a rate dependent on the incidence (better)

In this case, it’s the latter.

Also, another name for the error is “gini”

An ideal tree



A tree with 0 error/gini will have a 100% homogeneity

Coding resources

I use `sklearn`. It includes compatibility with matplotlib to easily show structures like above.

`sklearn.tree.DecisionTreeClassifier` handles classification very easily. There are many tutorials online on how to use it.

https://colab.research.google.com/drive/1eKXB2weI0hmKFZGRmLUEydJpXNM8rP8T?usp=sharing