CS 66: Machine Learning

Prof. Sara Mathieson

Spring 2019

Admin

- You can call me either Sara or Professor Mathieson (either one is fine!)
- Office hours: TODAY 1-3pm (Sci Center 249)
- Lab 1 due Tues night
- I will send out partner forms for Lab 2 over the weekend (complete by Tues at midnight)

Outline for January 25

- Style guidelines for Python
- Visualizations for K-nearest neighbors
- Begin: Decision Trees
- Entropy

Outline for January 25

- Style guidelines for Python
- Visualizations for K-nearest neighbors
- Begin: Decision Trees
- Entropy

Python style

- Decompose code into natural functions
- Avoid global variables, although sometimes in ML they are useful
- Include a file header with purpose and both partner names
- Include headers (in triple quotes) for each function
- No lines over 80 chars
- Include line breaks and comments!

Python style examples

```python
"""
Ask the user for their name and welcome them to CS21.
Author: Sara Mathieson
Date: 9/7/18
"""

def main():

    # ask user for their name and print greeting
    name = input("Enter your name: ")
    print("Hello", name, "!")

main()
```

```python
def factorial(n):
    """
    Given a non-negative integer n, return n! = n*(n-1)*(n-2)....3*2*1.
    """
    fact = 1 # set up an accumulator variable
    for i in range(n):
        fact = fact * (i+1) # accumulator pattern
    return fact
```

Lab 1 Notes

- May take a long time to run!

- Some data points are exactly the same between train and test (distance 0)

Outline for January 25

- Style guidelines for Python
- Visualizations for K-nearest neighbors
- Begin: Decision Trees
- Entropy

K-nearest neighbors creates implicit decision boundaries

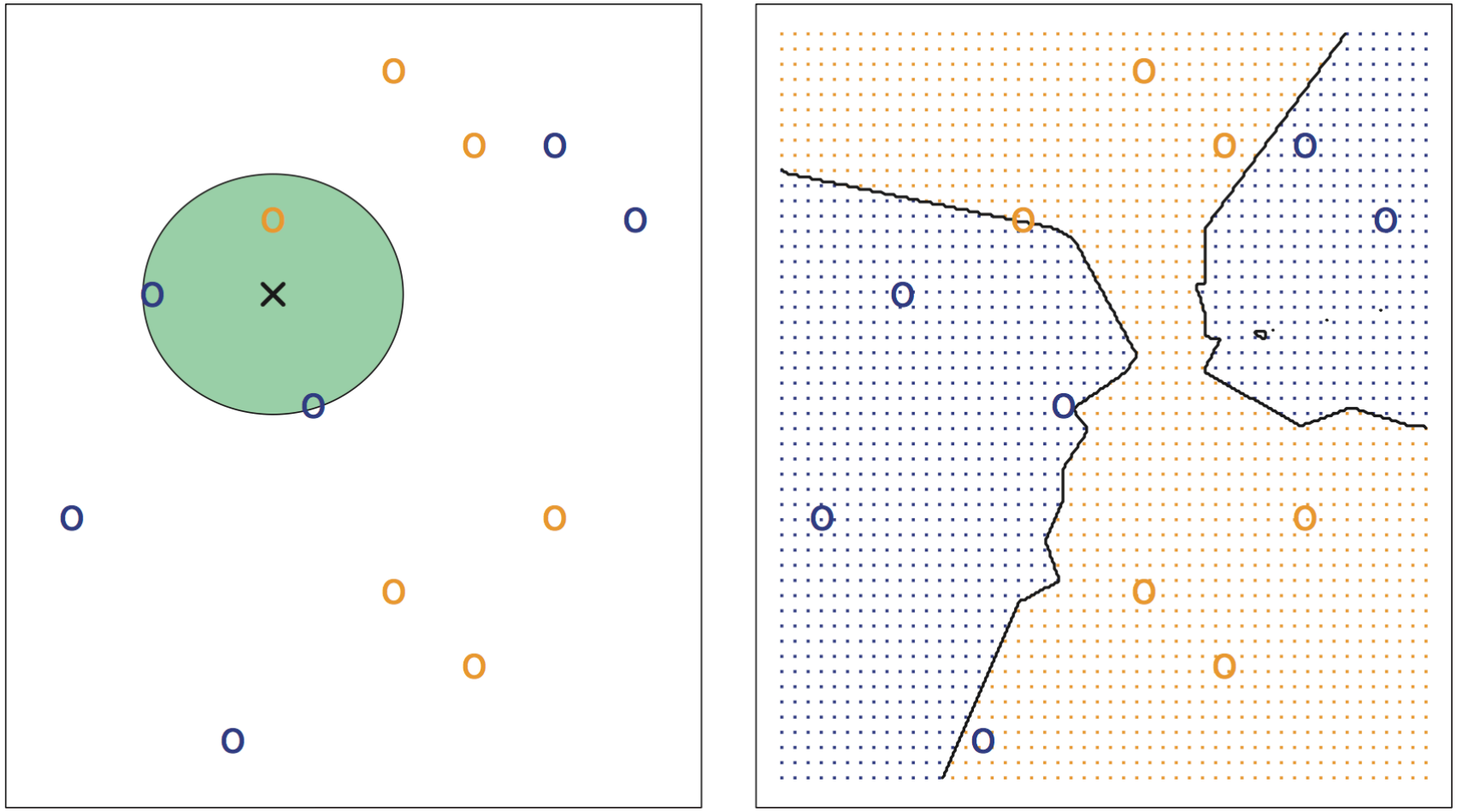
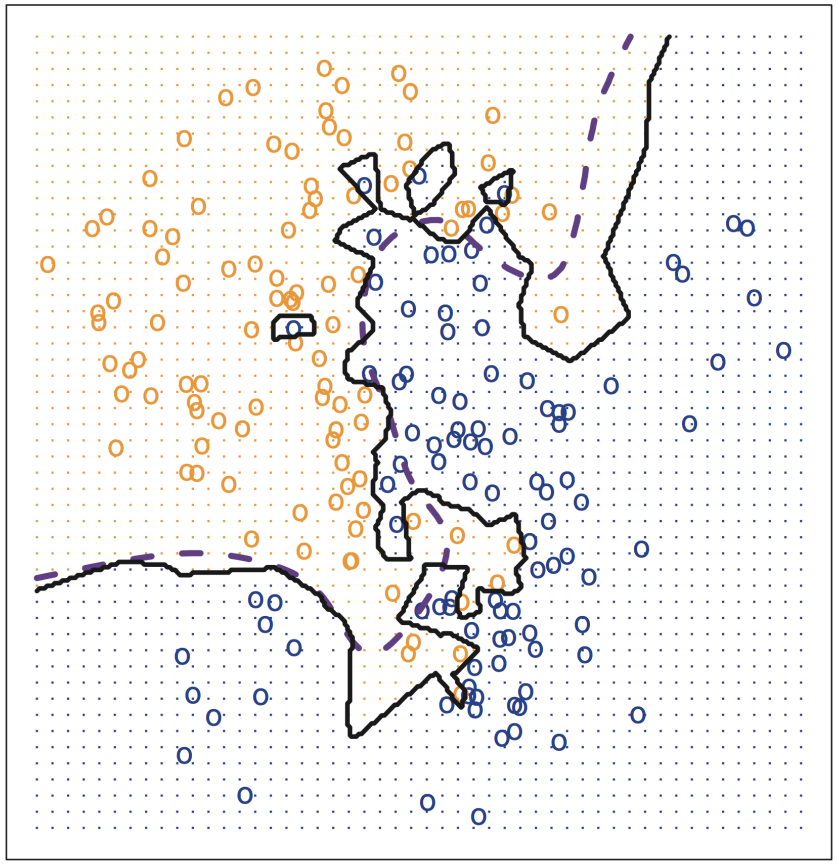


Figure 2.14 from ISL book, KNN with two classes (C=2), and k=3

K-nearest neighbors creates implicit decision boundaries

KNN: K=1



KNN: K=100

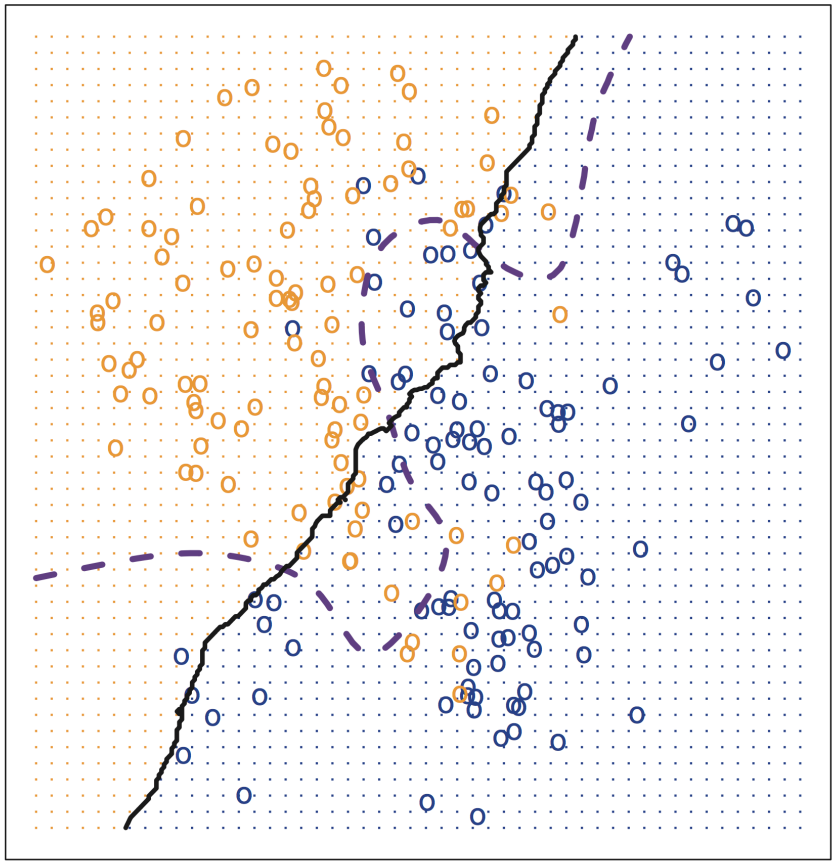


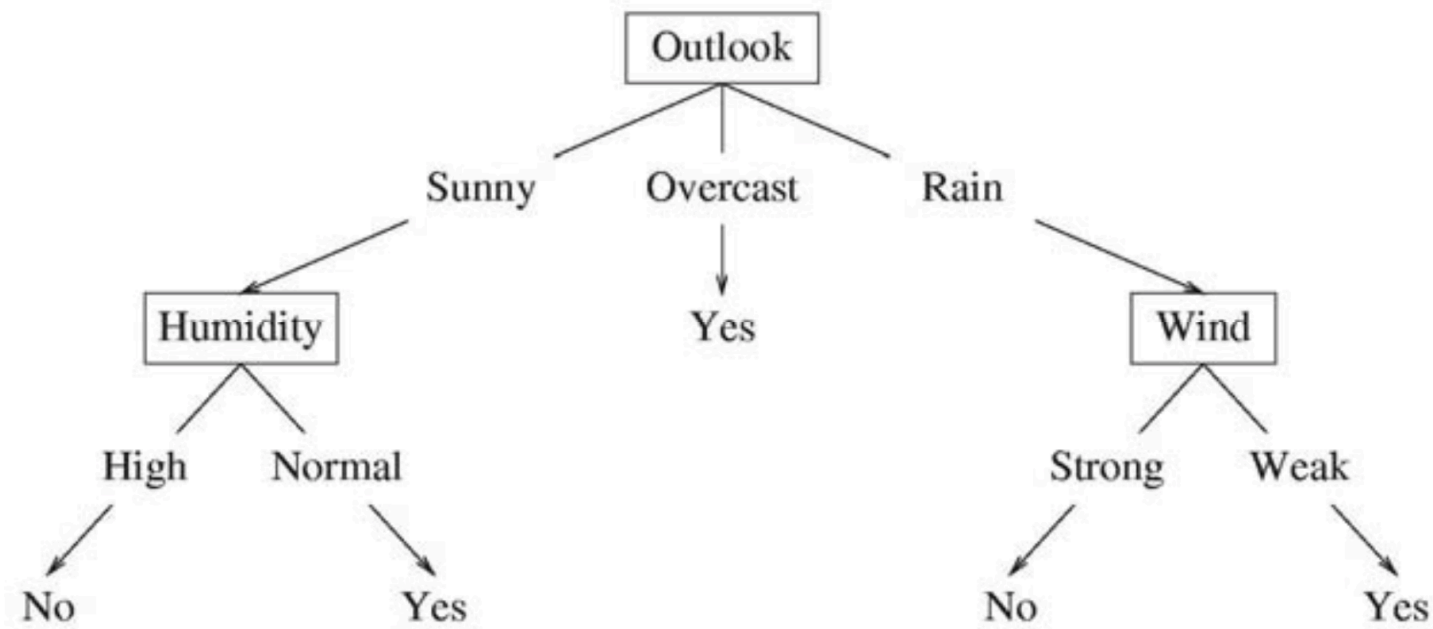
Figure 2.16 from ISL book, comparison of k=1 vs. k=100
(dashed line is “ideal” boundary)

Outline for January 25

- Style guidelines for Python
- Visualizations for K-nearest neighbors
- Begin: Decision Trees
- Entropy

Decision Tree example

- A possible decision tree for data:



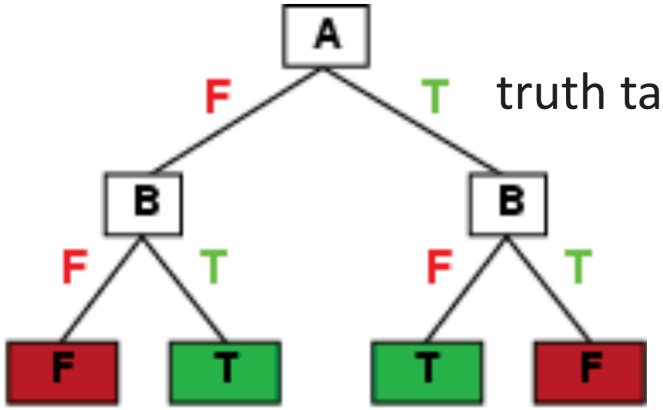
- Each internal node: test one feature
- Each branch from node: selects one value of the feature
- Each leaf node: predict y (or $p(y \mid x \text{ is a leaf})$)

Slide: Jessica Wu
Based on slide by Eric Eaton
[originally by Tom Mitchell]

Expressiveness

DTs can represent any boolean function of input features

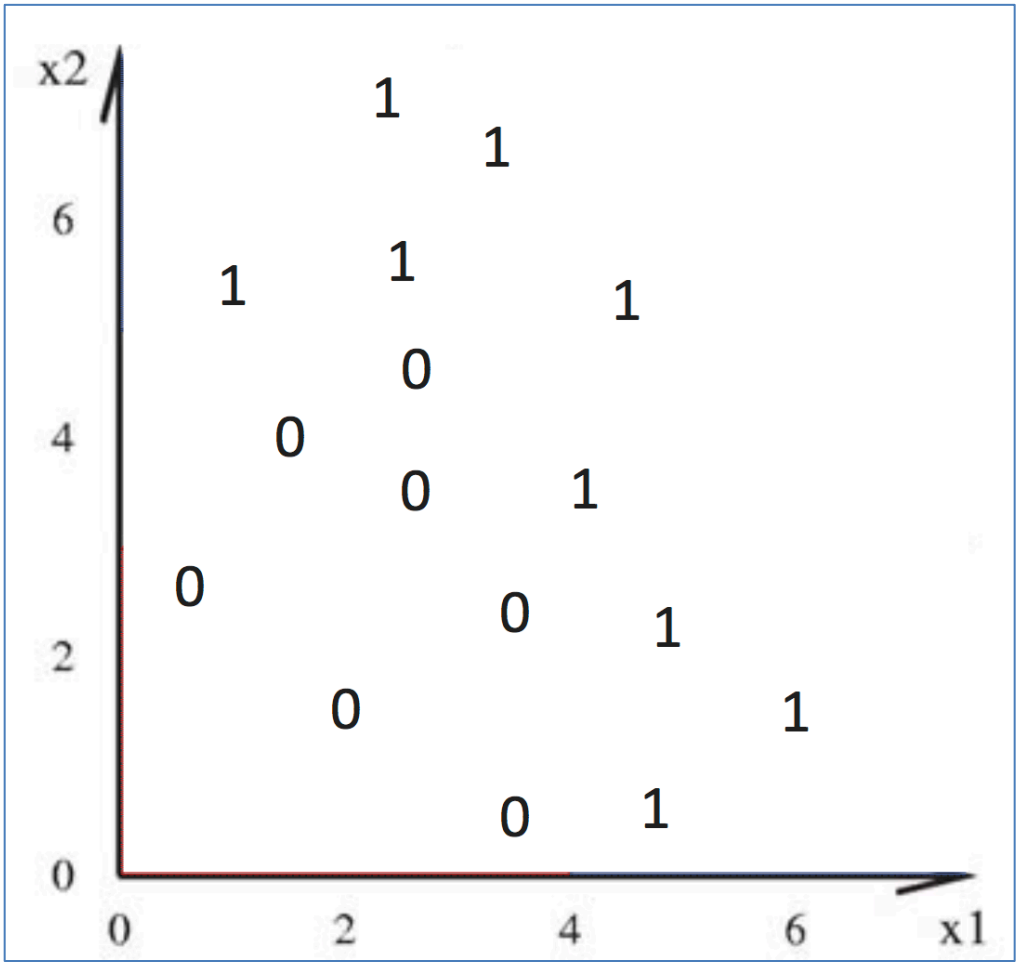
A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



truth table row → path to leaf

Slide: Jessica Wu
Original: Eric Eaton

Can also consider continuous features



Example by: Eric Eaton

How to select “best” features?

X

Color	Shape	Size
red	square	big
blue	square	big
red	circle	small
blue	square	small
red	circle	big

Y

Likes toy?
+
+
-
-
+

Example by: Ameet Soni

How to select “best” features?

X

Color	Shape	Size
red	square	big
blue	square	big
red	circle	**big**
blue	square	**big**
red	circle	big

Y

Likes toy?
+
+
-
-
+

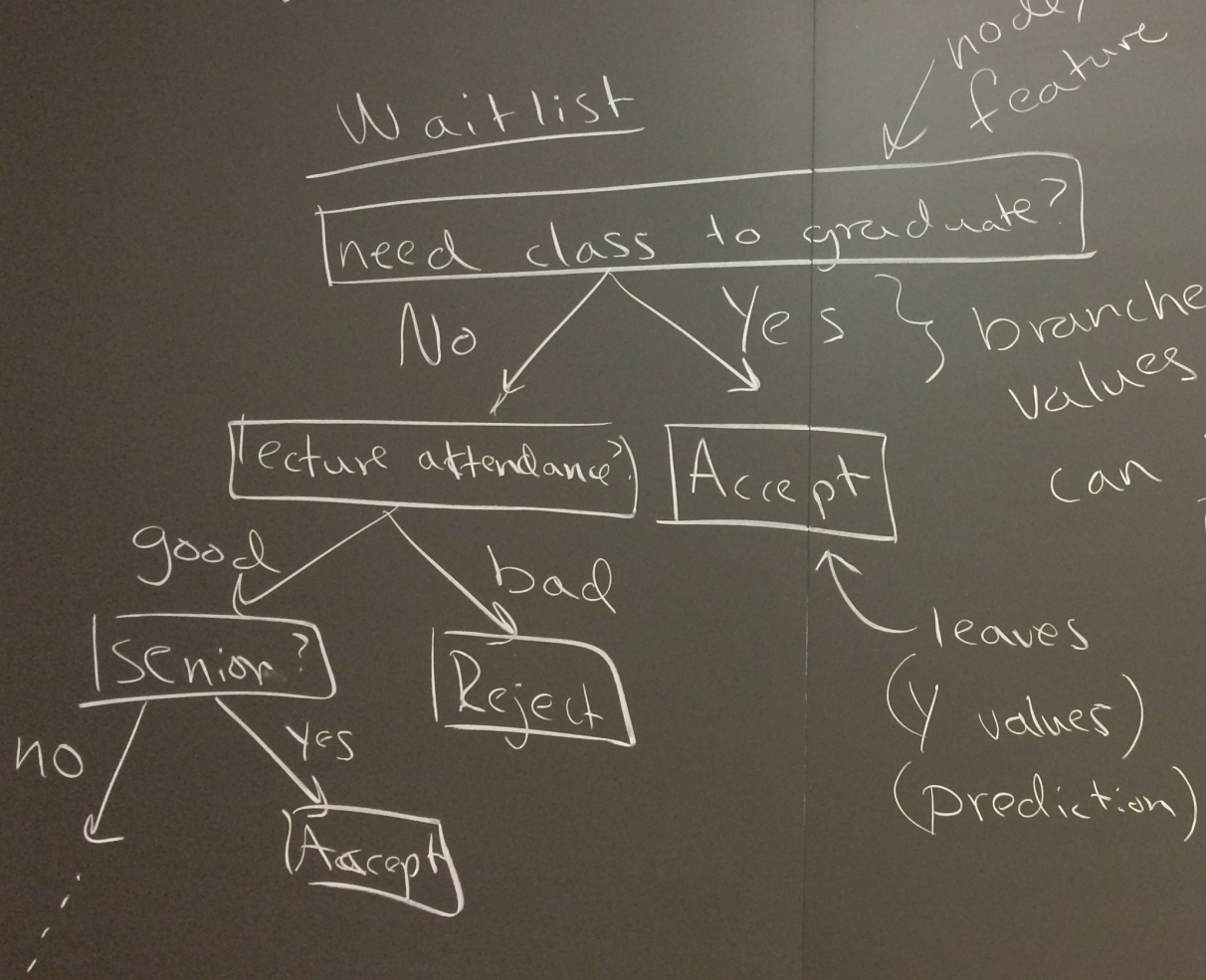
Example by: Ameet Soni

Outline for January 25

- Style guidelines for Python
- Visualizations for K-nearest neighbors
- Begin: Decision Trees
- Entropy

Example Decision Tree

Waitlist



$Y \in \{\text{accept}, \text{reject}\}$

} branches => values feature can take on

STOP criteria:

*all data points in partition have same label

ID3 Algorithm (Quinlan)

1. select the "best" feature that informs label/output

2. divide: partition data into branches based on their value for that feature

3. conquer: recurse on each partit

Entropy

class	prob	fixed len	prefix encoding
senior	0.5	00	0
junior	0.25	01	10
soph	0.125	10	110
first-year	0.125	11	111

$\underbrace{\text{class}}_{Y}$

① send raw text "senior"

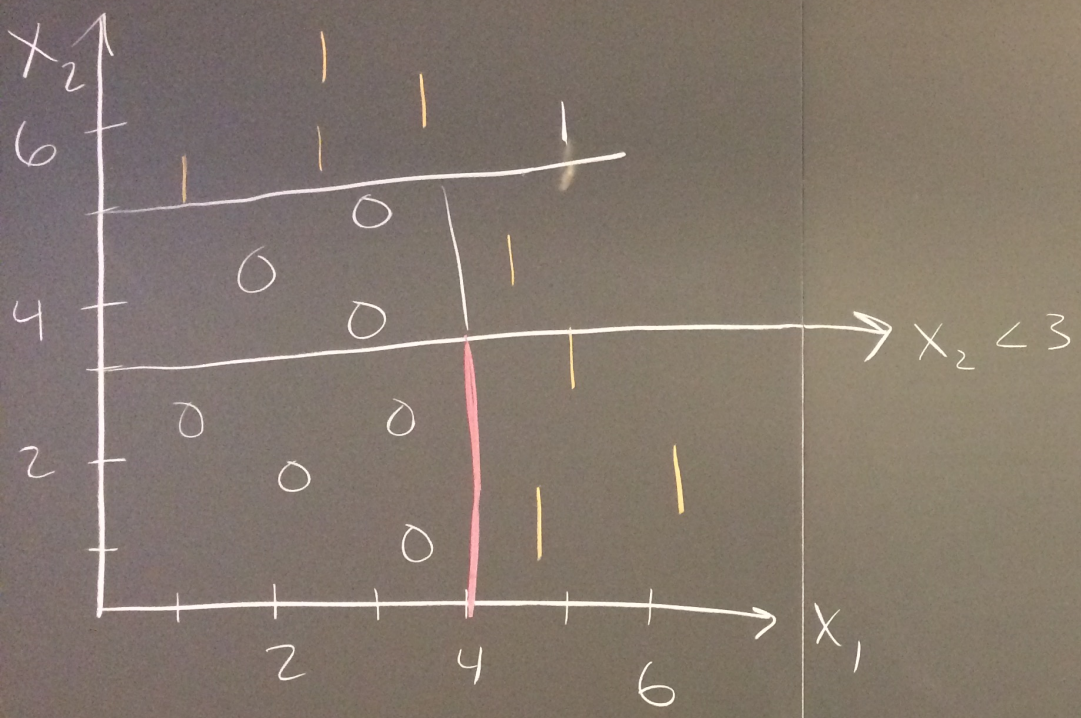
② fixed length binary encoding

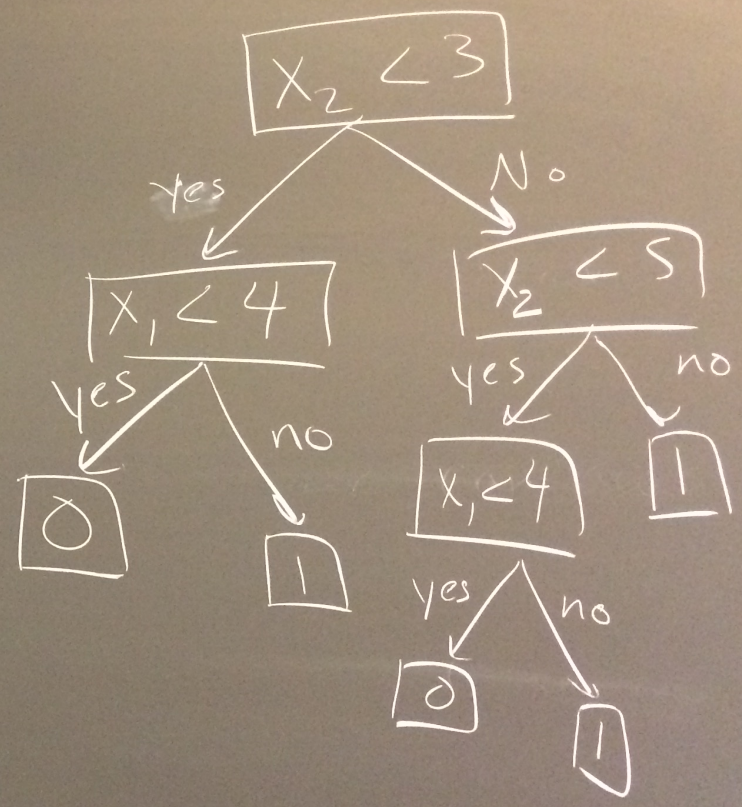
avg # of bits to send one value

$$\frac{1}{2}\cdot 1 + \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 3 + \frac{1}{8}\cdot 3$$

$$1 + \frac{3}{4} \rightarrow \boxed{1.75}$$

Same as entropy!





Note: there could be more optimal (i.e. smaller) trees!