Decision trees

Research Skills: Machine Learning

Grzegorz Chrupała
g.chrupala @ uvt.nl

Learning

K-Nearest-Neighbors learns by memorizing

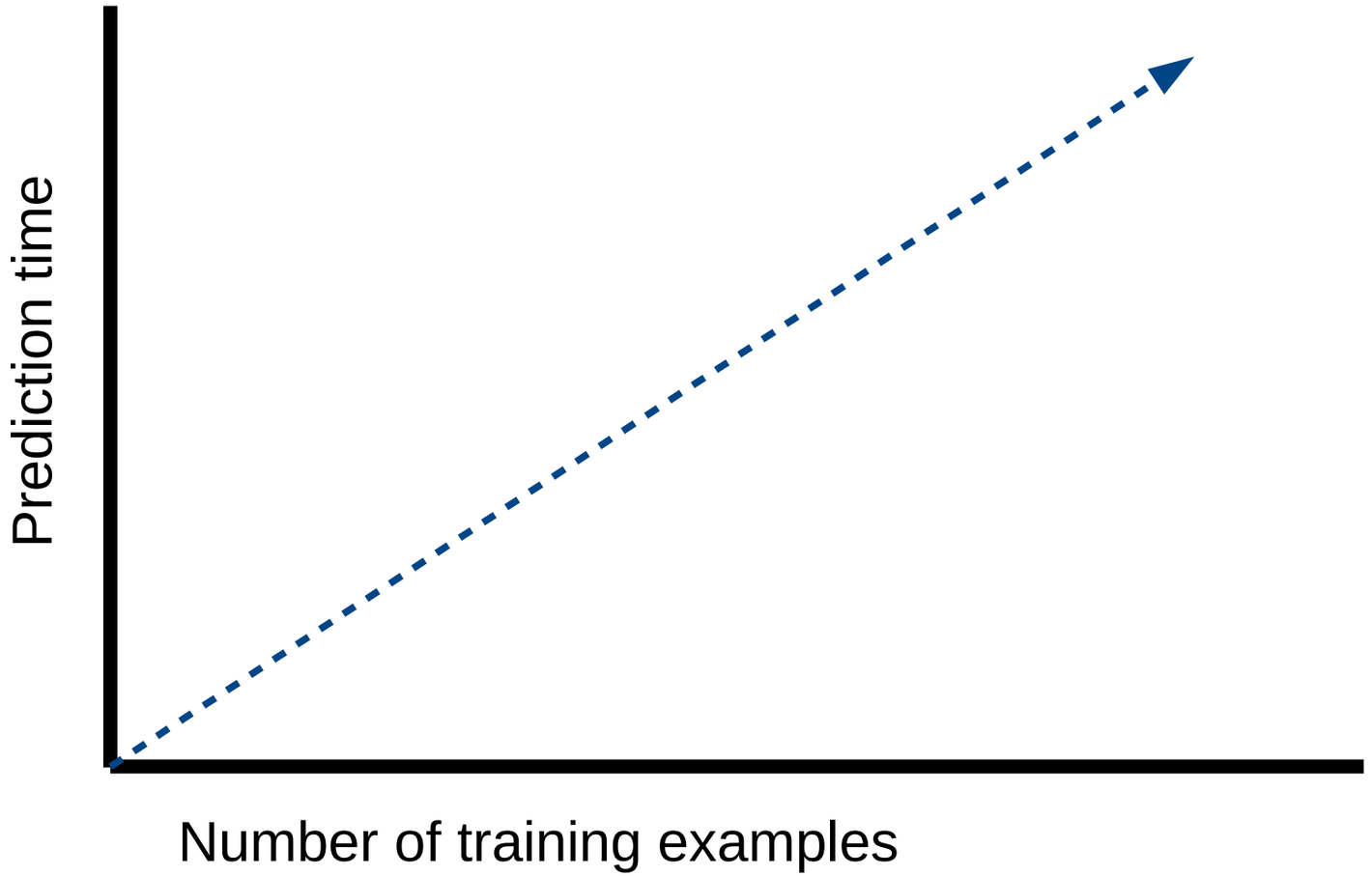
k-NN

- Advantages
  - Simple: easy to understand and implement
  - Often works well in practice
- Disadvantages
  - No abstraction
  - Slow for prediction

k-NN speed

- You have a **1,000 training** examples
- Predicting the targets of 100 new examples takes 0.1 second
- How long would it take with **10,000 training** examples?
- How about **100,000** training example?

Linear slowdown

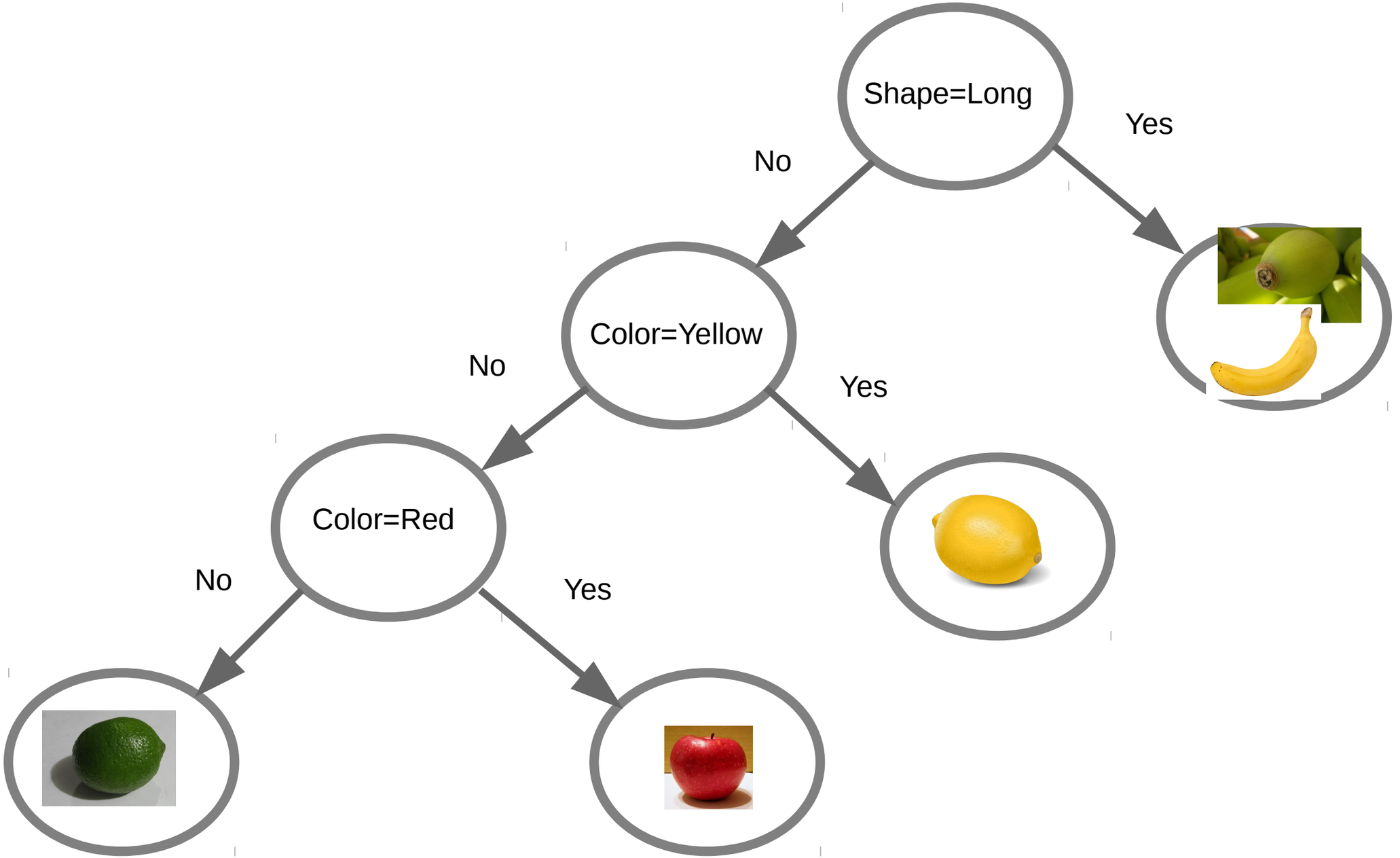


Learning rules

- If condition A:
  - If condition B:
    - Action 1
  - Else:
    - Action 2
- Else:
  - Action 3

Fruit classification

Shape	Color	*Target*
Round	Green	Lime
Round	Yellow	Lemon
Round	Green	Apple
Round	Red	Apple
Long	Yellow	Banana
Long	Green	Banana


Shape=Long
No
Yes
Color=Yellow
No
Yes
Color=Red
No
Yes

How can we build a decision tree?

- Number of possible trees grows exponentially with number of features
- Can't check them all and see which one works best
- Need to build a tree incrementally

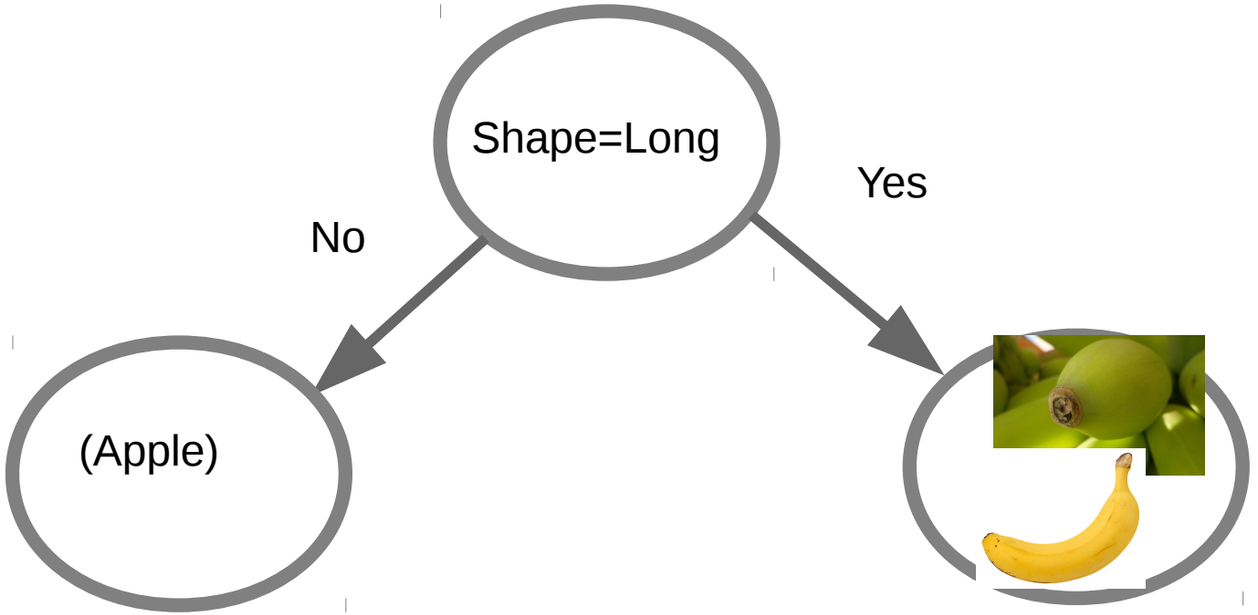
Which question to ask first?

- It's best to ask important questions first
- Which questions are important?
- The ones which help us classify:
  - if we had to classify data based only on one question, which question would do best?

Step by step

Q	Correct
Long?	4
Green?	2
Red?	3
Yellow?	3

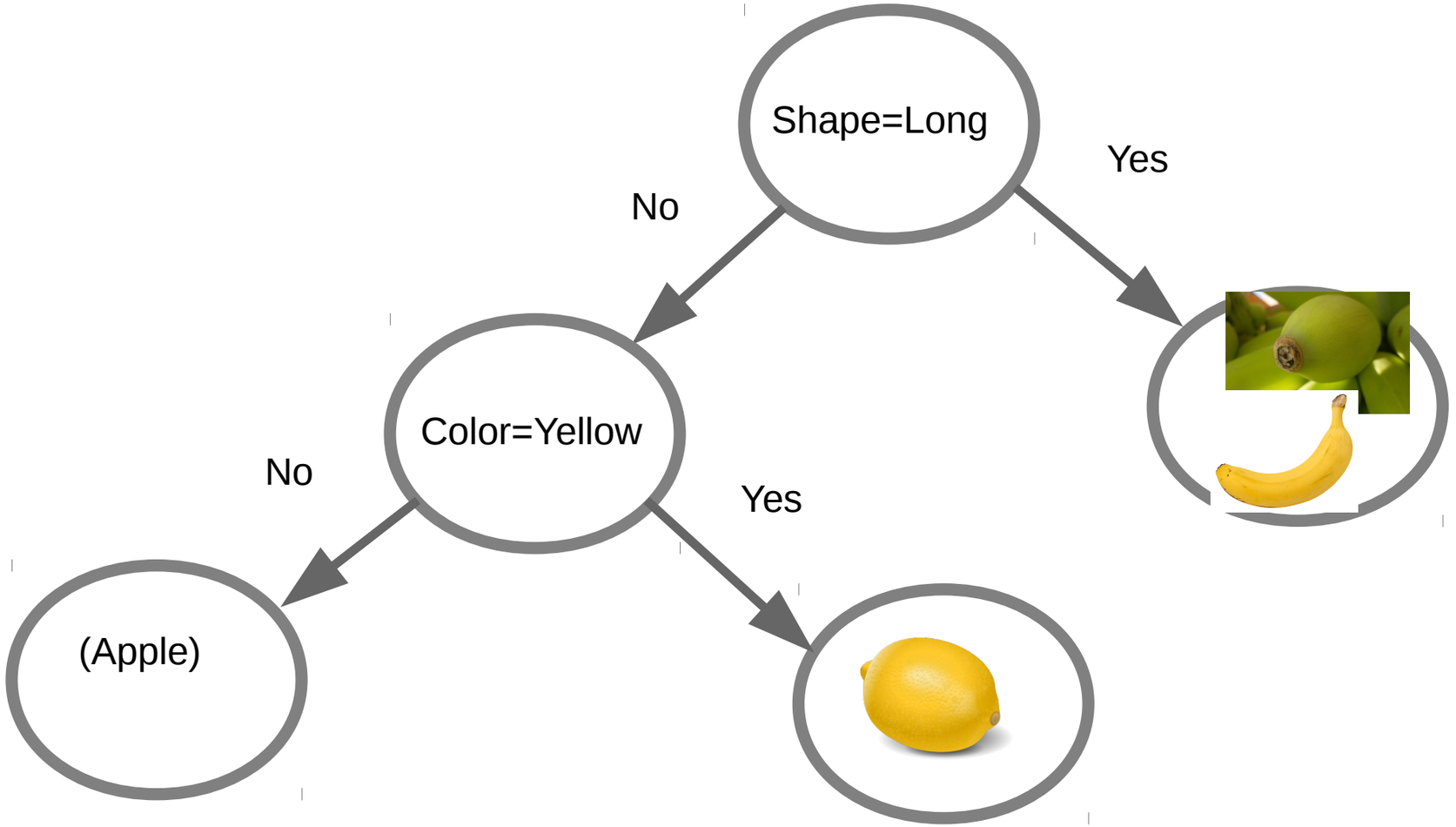
Shape	Color	*Target*
Round	Green	Lime
Round	Yellow	Lemon
Round	Green	Apple
Round	Red	Apple
Long	Yellow	Banana
Long	Green	Banana


Shape=Long
No
Yes
(Apple)

Long=No

Q	Correct
Green?	2
Red?	2
Yellow?	3

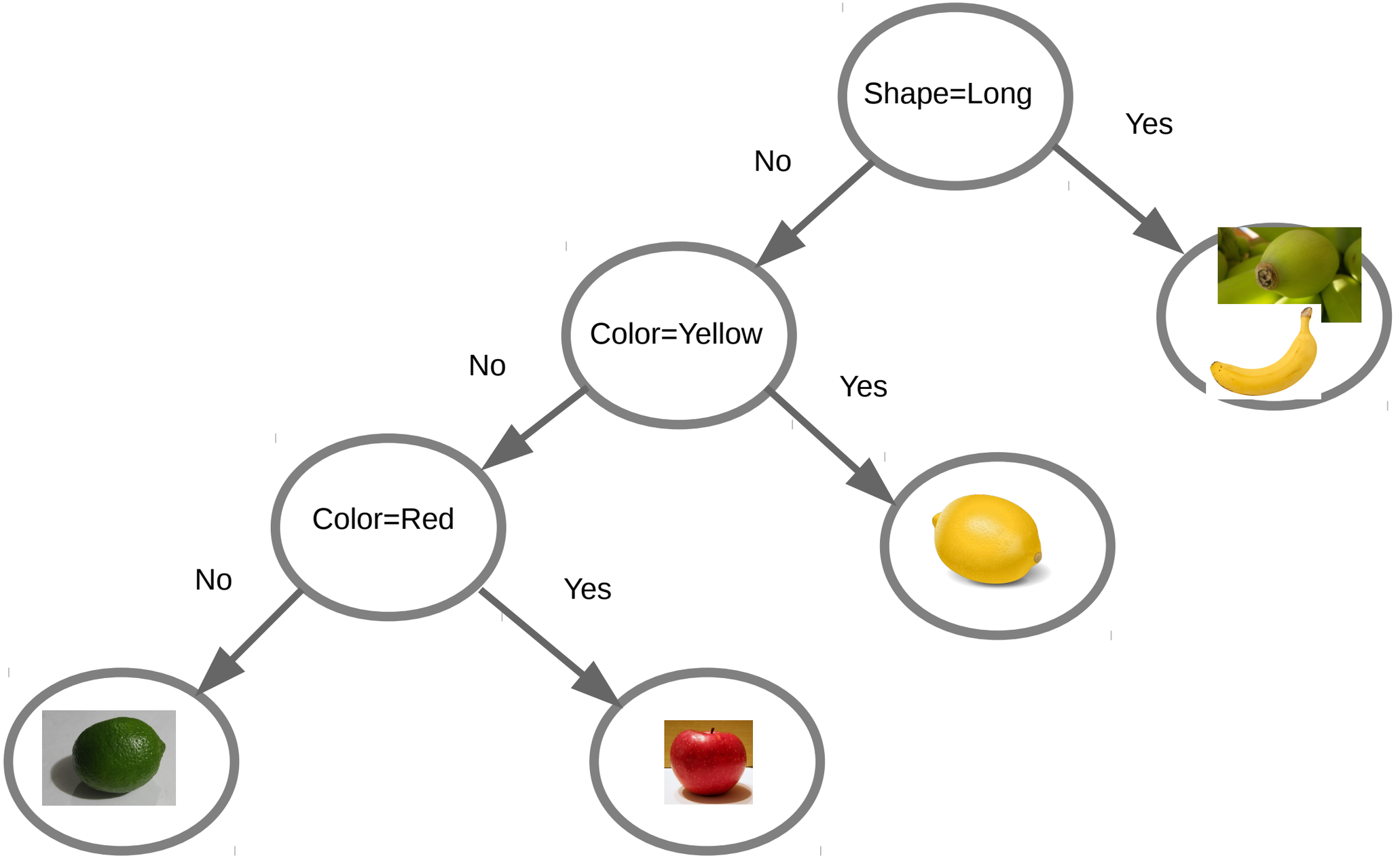
Shape	Color	*Target*
Round	Green	Lime
Round	Yellow	Lemon
Round	Green	Apple
Round	Red	Apple


Shape=Long
No
Yes
Color=Yellow
No
Yes
(Apple)

Yellow=No

Q	Correct
Red?	2
Green?	2

Shape	Color	*Target*
Round	Green	Lime
Round	Green	Apple
Round	Red	Apple


Shape=Long
No
Yes
Color=Yellow
No
Yes
Color=Red
No
Yes

Building a decision tree

- If all examples have same label
  - Create leaf node with label
- Otherwise
  - Choose most important question
  - Split data into two parts (**NO** and **YES**) according to question
  - Remove question from question set
  - Left branch ← Apply algo to **NO** examples
  - Right branch ← Apply algo to **YES** examples
  - Create node with (question, left branch, right branch)

Shape	Color
Round	Green

Using a decision tree

- Given a tree and an example
  - If tree is leaf node:
    - Prediction ← label
  - Otherwise ask the question about example
    - If **NO**
      - Prediction ← apply algo with left branch
    - If **YES**
      - Prediction ← apply algo with right branch

Decision Tree speed

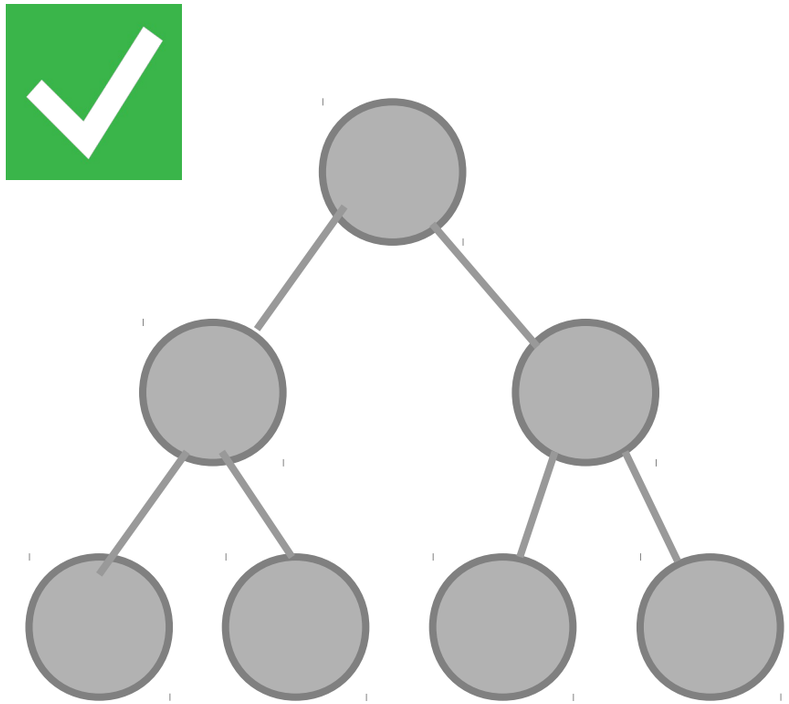
- You build a tree from **1,000 training** examples
- Predicting the targets of 100 new examples takes 0.1 second
- How long would it take with a tree made from **10,000 training** examples?
- How about **100,000** training example?

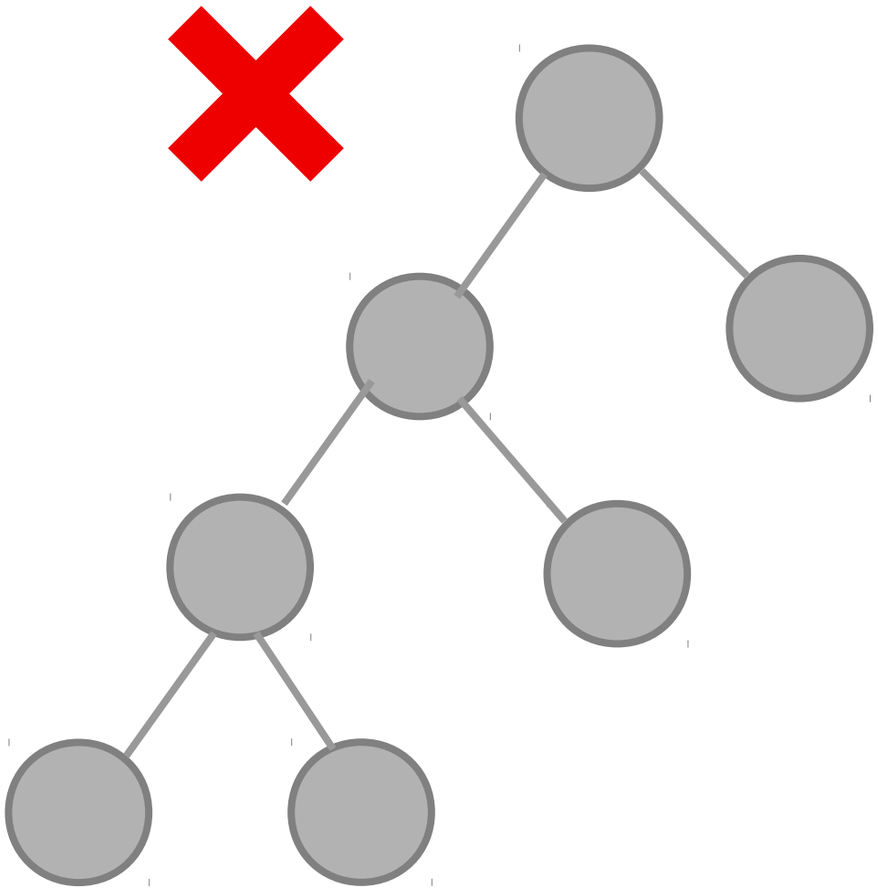
Most likely **less** than 1 second

Decision Tree speed

- Depends on number of questions needed to get to a leaf node
- Which depends on **depth** of the tree

(Un)balanced trees





Depth of balanced tree

- In a balanced binary tree, each time you ask a question
- You **halve** the number of remaining question

Repeated halving

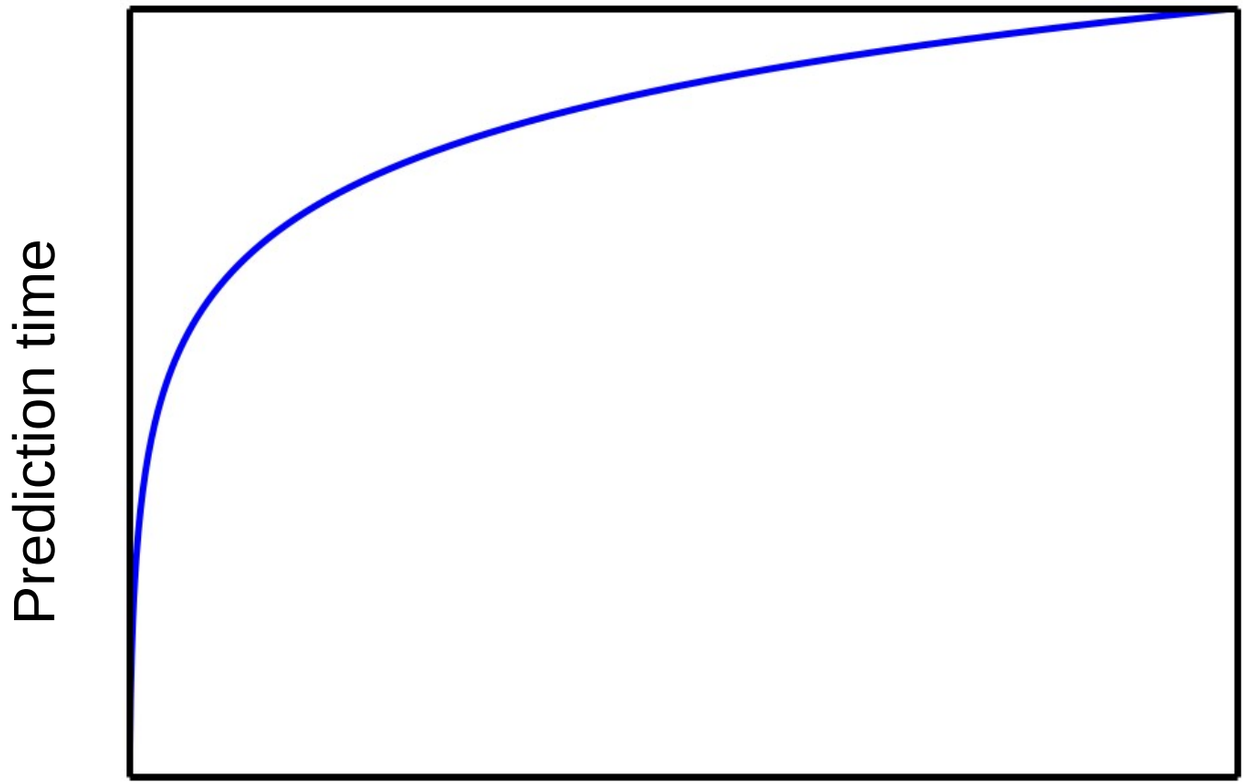
- How many halvings of **N** to get to **1**?
- How many doublings of **1** to get to **N**?

$$(((1 \times 2) \times 2) \times 2) = 8$$

$$2^3 = 8$$

$$\log_2(8) = 3$$

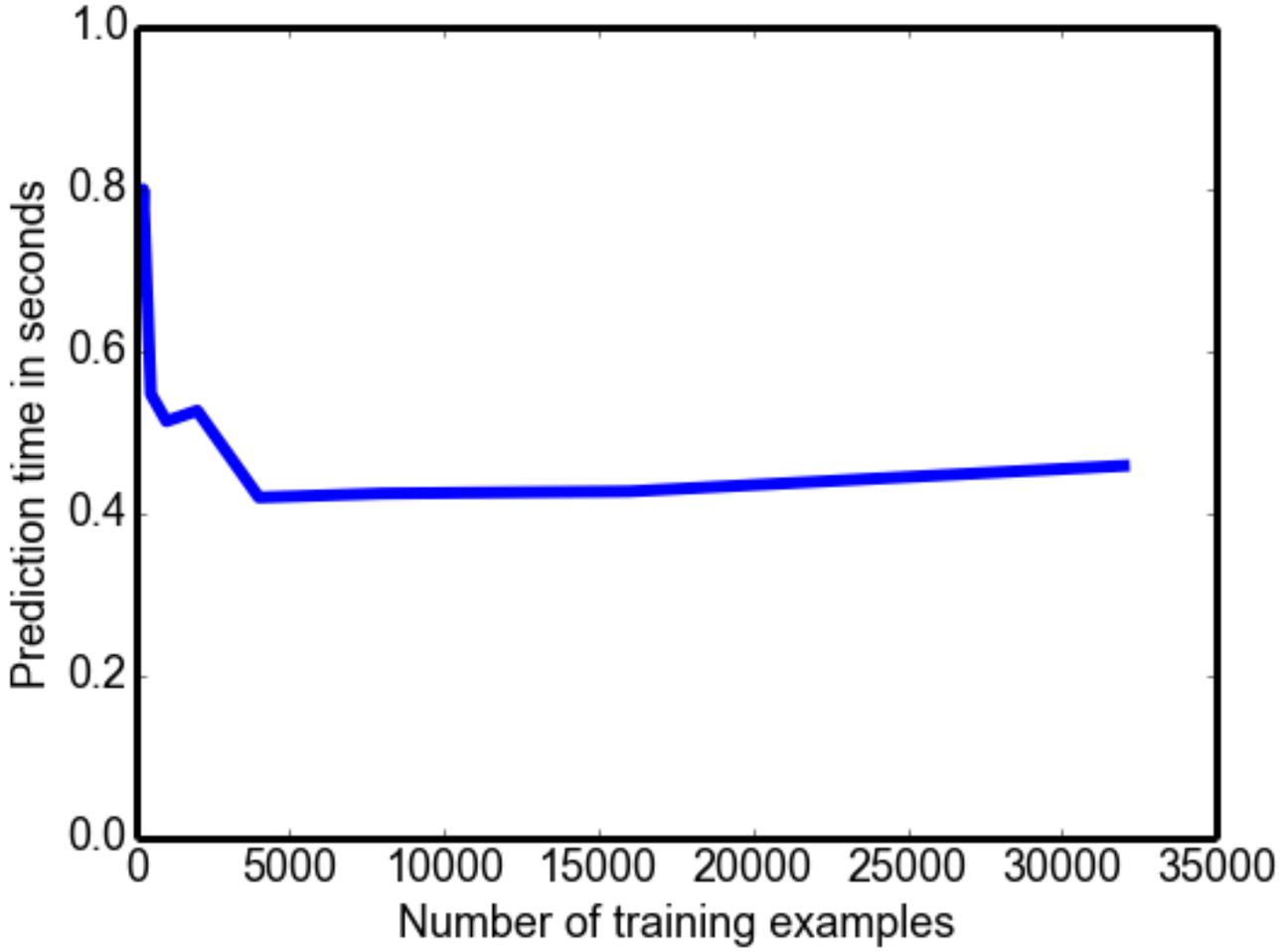
Prediction speed (balanced trees)



Speed in reality – Census income

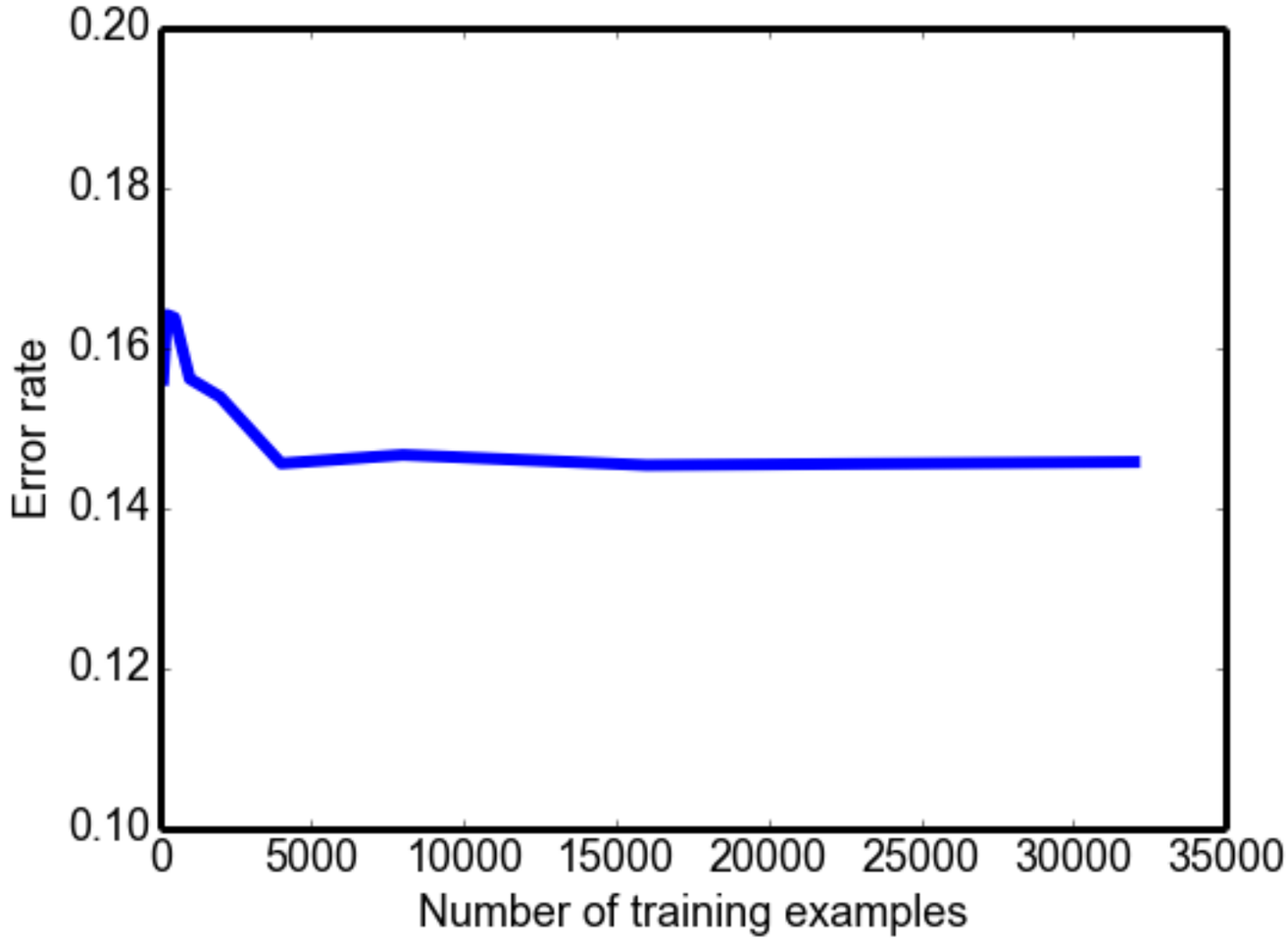
INPUT					TARGET
age	edu	occupation	race	sex	income
39	13	Adm-clerical	White	Male	<=50K
50	13	Exec-managerial	White	Male	<=50K
38	9	Handlers-cleaners	White	Male	<=50K
53	7	Handlers-cleaners	Black	Male	<=50K
28	13	Prof-specialty	Black	Female	<=50K
37	14	Exec-managerial	White	Female	<=50K
49	5	Other-service	Black	Female	<=50K
52	9	Exec-managerial	White	Male	>50K
31	14	Prof-specialty	White	Female	>50K
42	13	Exec-managerial	White	Male	>50K
37	10	Exec-managerial	Black	Male	>50K

Speed in reality



Larger datasets tend to produce better-balanced trees.

Error rates


0.20
0.18
0.16
0.14
0.12
0.10
Error rate
0
5000
10000
15000
20000
25000
30000
35000
Number of training examples

How do we generate questions?

- Categorical values
  - Binarize (convert to 1/0 or YES/NO)
- Numerical values
  - Discretize
  - Questions of the form $\text{is } x_i <= \text{threshold}_j?$
  - Thresholds: values found in data (or between pairs of adjacent values)

Discretization

YearsEducation
13
13
9
7
13
14

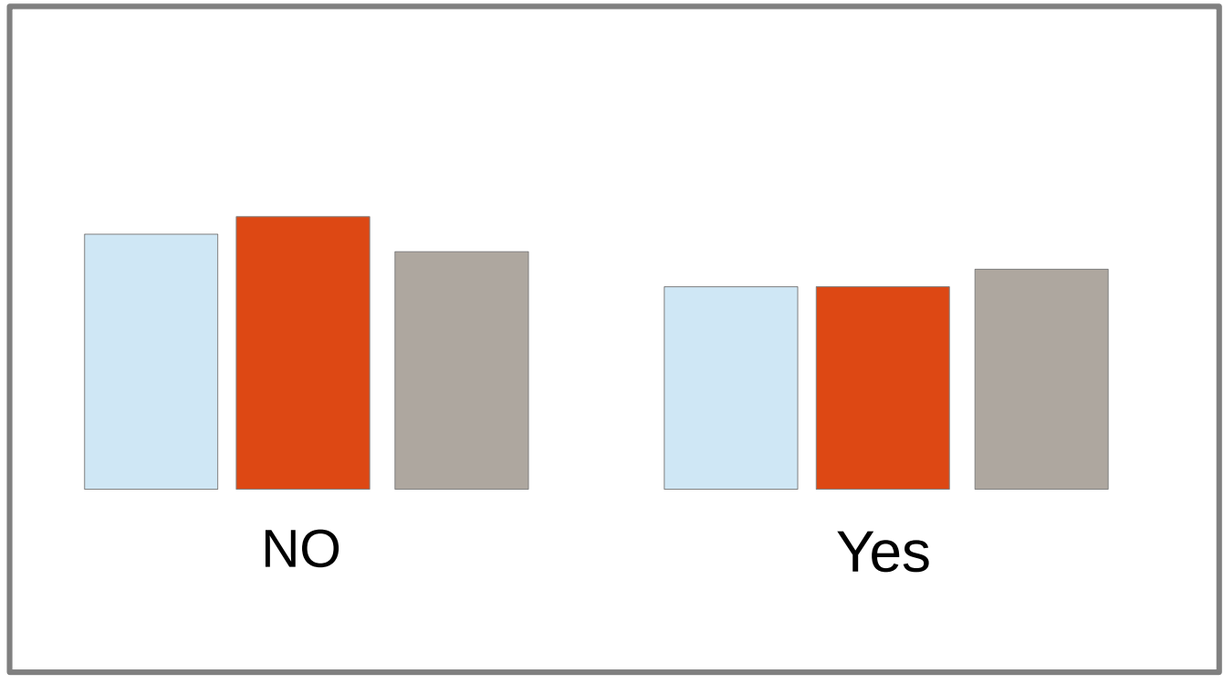
<=7	<=9	<=13	<=14
No	No	Yes	Yes
No	No	Yes	Yes
No	Yes	Yes	Yes
Yes	Yes	Yes	Yes
No	No	Yes	Yes
No	No	No	Yes

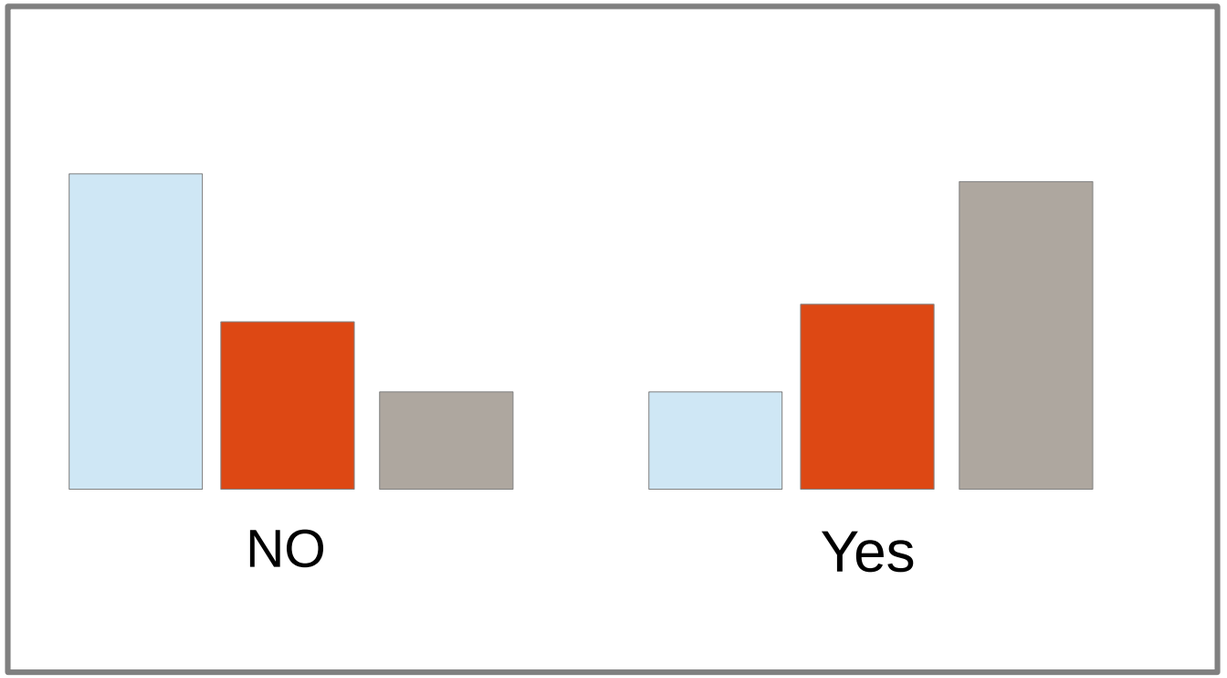
Question goodness

- Classification accuracy – very simple
- Purity measures – typically perform better
- **Entropy**

Entropy

- Measure of uniformity of distribution





Digression: Recursion

- We build and use DT with recursive functions
- Recursive function
  - Base case
  - Recursive call – applies itself

```
def factorial(n):
    if n == 1:
        return 1
    else:
        return n * factorial(n - 1)
```

Recursion exercise: Power

- Define a recursive function **`power`** which raises number x to the n^{th} power
- You will need to repeated multiply the number by itself

```
>>> print power(2, 3)
8
>>> print power(2, 8)
256
```

Traverse a decision tree

- Write function **`predict`** which takes a decision tree and a new example, and returns the prediction

```
model = \
{'q': 1,
 'left': {'q': 4,
  'left': {'q': 3,
           'left': {'guess': 'Lime'},
           'right': {'guess': 'Apple'}},
  'right': {'guess': 'Lemon'}},
 'right': {'guess': 'Banana'}}
```

```
                # Round  Long   Green  Red    Yellow
>>> new = [ True, False, True,  False, False ]
>>> print predict(model, new)
Lime
```

Image credits

- Red apple http://upload.wikimedia.org/wikipedia/commons/2/24/Redapple.jpg
- Banana http://upload.wikimedia.org/wikipedia/commons/8/8a/Banana-Single.jpg
- Lime http://upload.wikimedia.org/wikipedia/commons/5/55/Lime_closeup.jpg
- Lemon https://openclipart.org/image/300px/svg_to_png/189589/lemon-citrina.png
- Green apple http://upload.wikimedia.org/wikipedia/commons/5/55/GreenApple.png
- Green Banana http://pixabay.com/en/green-bananas-tip-garden-banana-108109/
- Green lemon http://pixabay.com/static/uploads/photo/2013/12/15/11/33/lemon-228857_640.jpg
- Cross mark http://pixabay.com/static/uploads/photo/2012/04/12/20/12/x-30465_640.png
- Check mark https://openclipart.org/image/300px/svg_to_png/196364/check.png