Decision Tree ID3 Algorithm

Kavya Bohra
Frank Su

Why Trees ?

- Used in data mining to predict which class a new entry will be in
- Ease of explanation and output is human readable

ID3 Algorithm

Let's dive in a popular algorithm to see it in action

Split(node,{examples})

- A <- best attribute for node
- Split {examples} using A
- For each A, create child
- If subset is pure, stop
- If not pure, recursively split subset even more

Semester	Professor	Midterm	Report	Easy
S1	Pei	Difficult	long	No
S2	Pei	Difficult	Short	No
S3	Wang	Difficult	long	Yes
S4	Edgar	Difficult	long	Yes
S5	Edgar	Easy	long	Yes
S6	Edgar	Easy	Short	No
S7	Wang	Easy	Short	Yes
S8	Pei	Difficult	long	No
S9	Pei	Easy	long	Yes
S10	Edgar	Easy	long	Yes
S11	Pei	Easy	Short	Yes
S12	Wang	Difficult	Short	Yes
S13	Wang	Easy	long	Yes
S14	Edgar	Difficult	Short	No

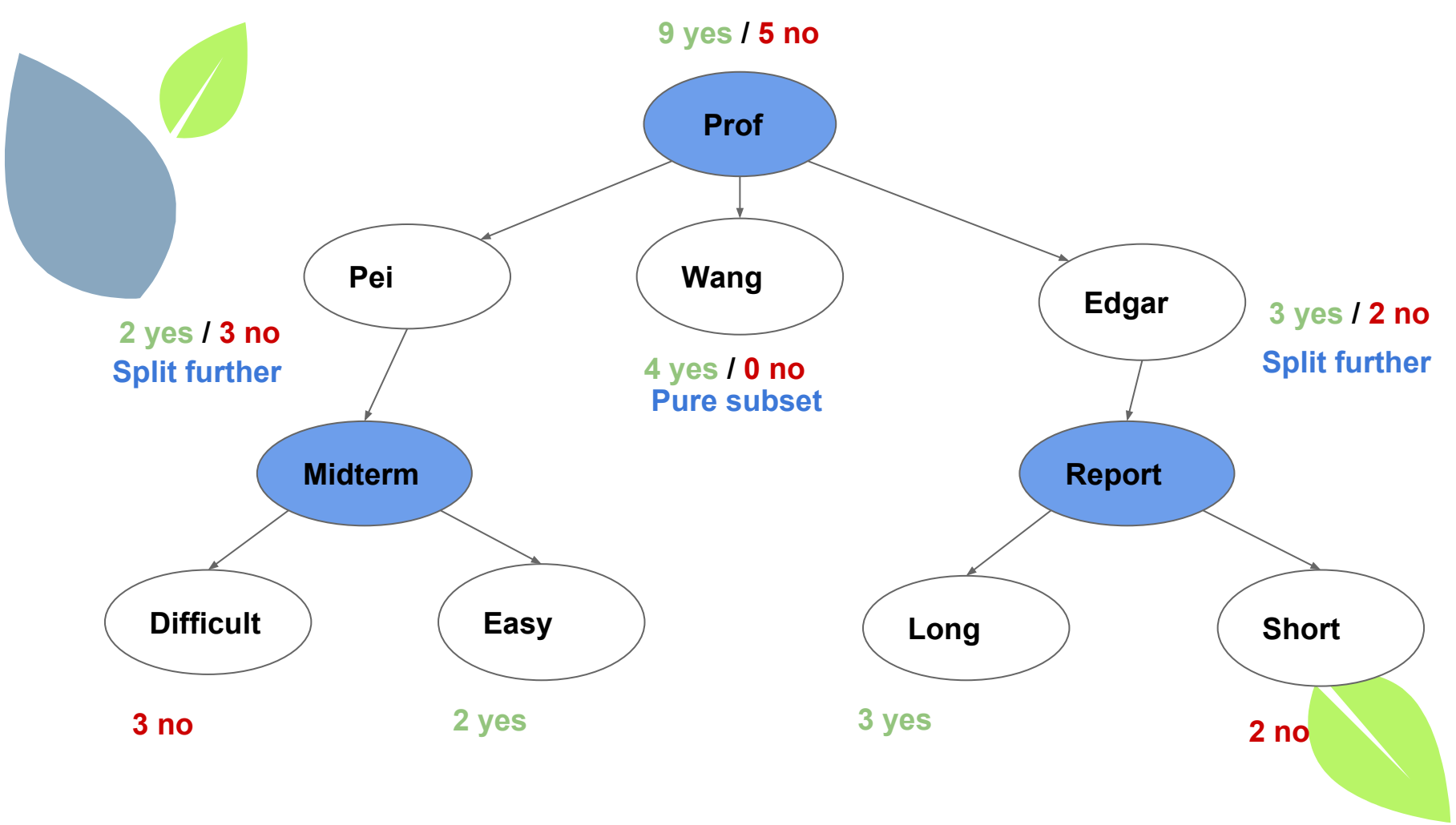
Predict if 454 final will be easy

Training examples: 9 yes / 5 no

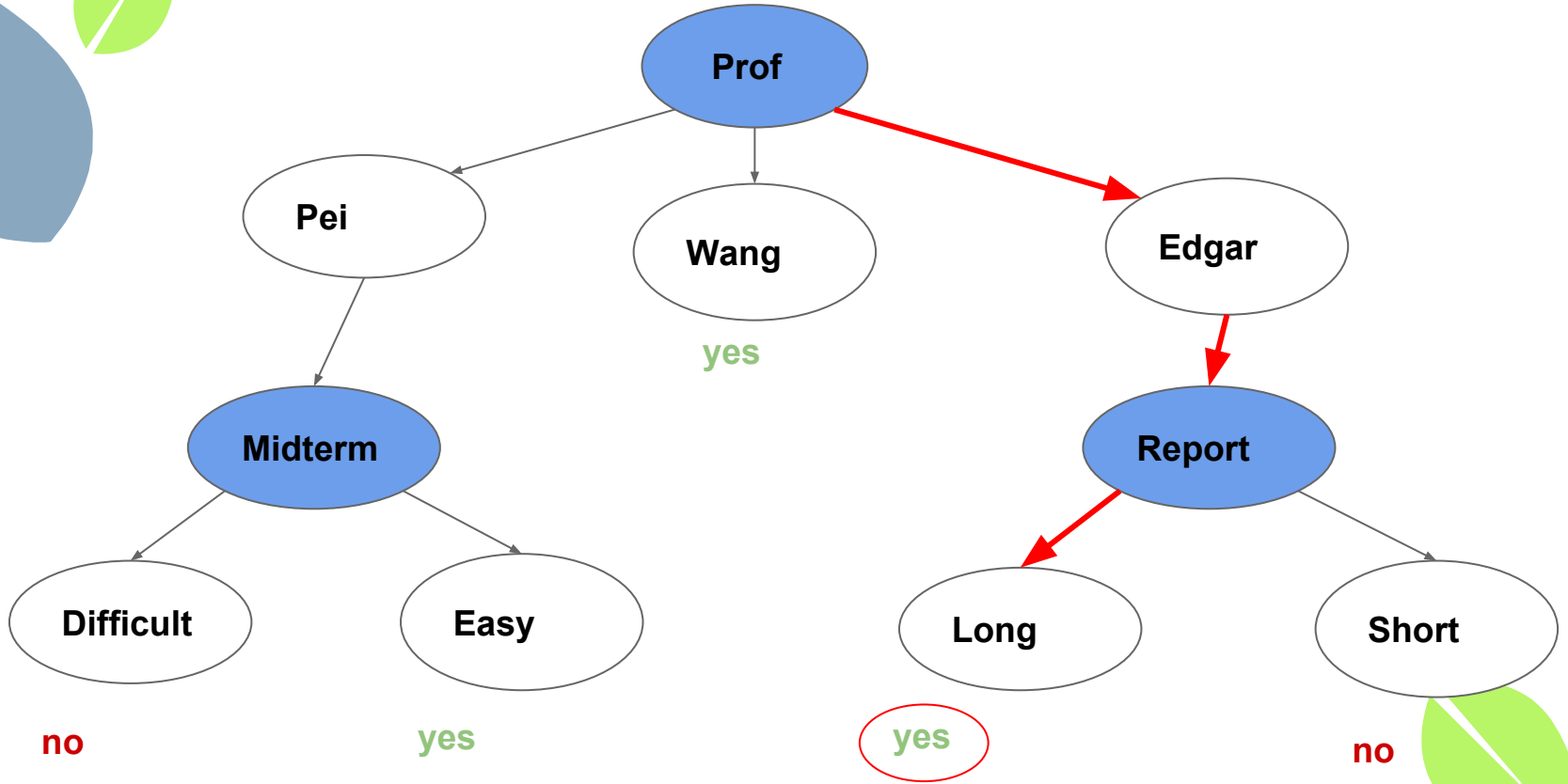
- Difficult to guess
- Try to understand when final is easy
- Divide & conquer
  - split into subsets
  - Are they pure? (all yes or all no)
  - If yes: stop
  - if not: repeat
- See which subset new data falls into

New data:

S15 Edgar Difficult long ?


9 yes / 5 no
Prof
Pei
Wang
Edgar
2 yes / 3 no
Split further
4 yes / 0 no
Pure subset
3 yes / 2 no
Split further
Midterm
Report
Difficult
Easy
Long
Short
3 no
2 yes
3 yes
2 no

Predict if 454 final will be easy



New data: S15 Edgar Difficult long ?

Which Attribute To Split on?

Want to measure “purity” of split

Entropy

Entropy: H(s) = -p(+) log p(+) - p(-) log p(+) of bits

p(+) = Yes

p(-) = No

How manys bits to say whether X is positive or negative.

Range between 0 to 1

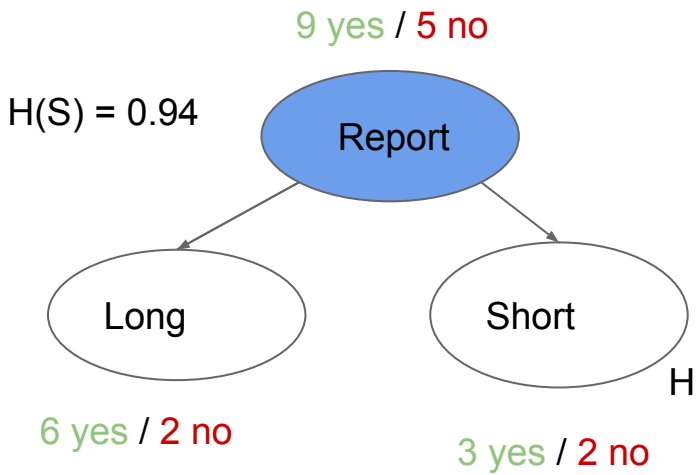
If it's pure, 0 bits.

If 10 yes, 10 no? 1 bit

Entropy tells us how pure or impure one subset is

Information Gain

- Want many items in pure subset
- Expected drop in entropy after split



H(S) = -9/14 log_2 9/14 - 5/14 log_2 5/14 = **0.940**

Gain (S, Report) = H(S) - 8/14(H(S_long)) - 6/14H(S_Short)
= **0.049**

Thanks!