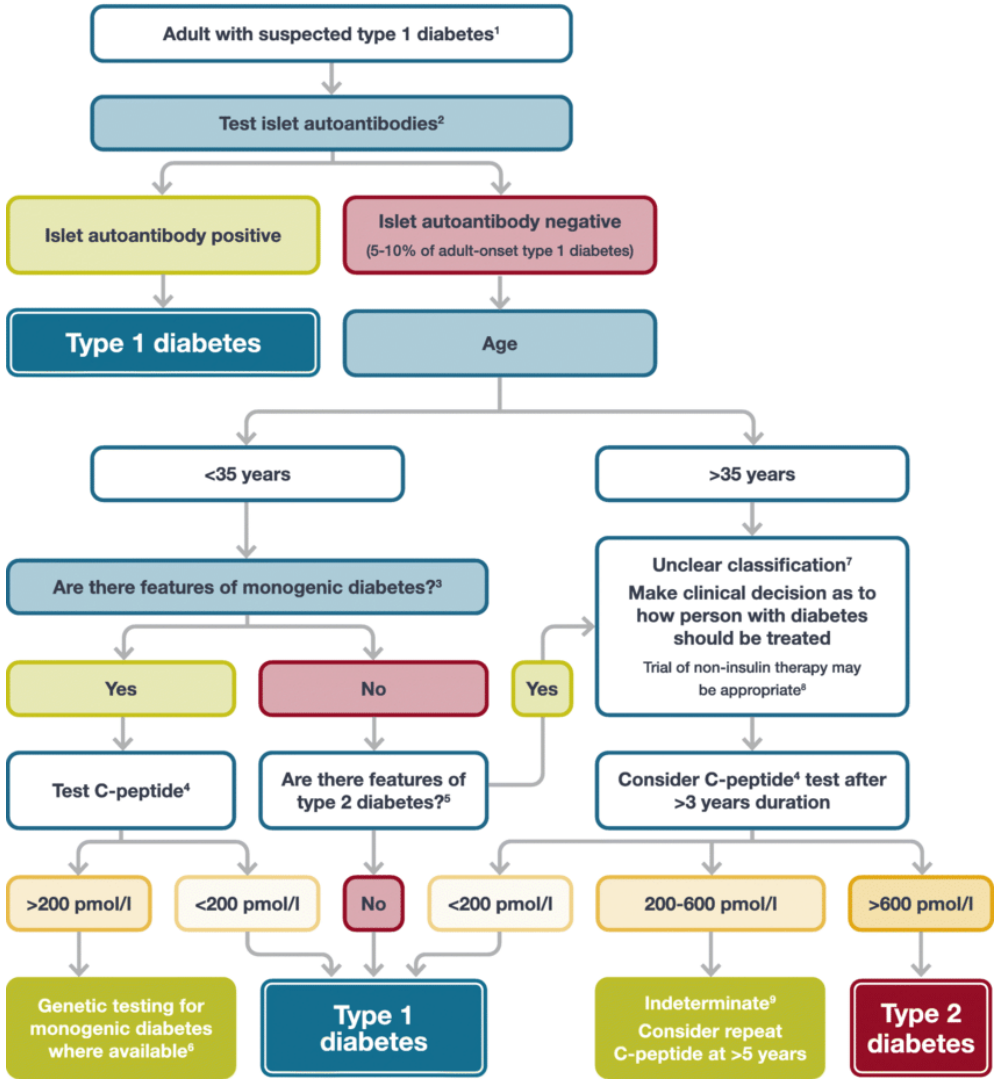
Non-parametric methods
Trees

Natasha Jaques

UNIVERSITY of WASHINGTON

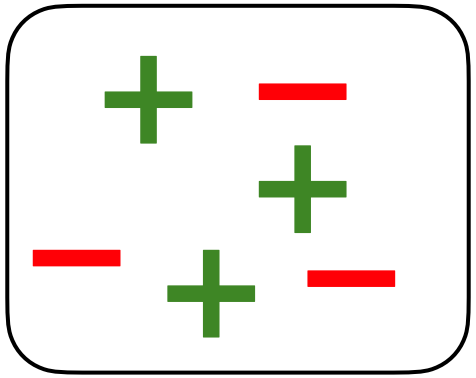
Flow chart for investigation of suspected type 1 diabetes in newly diagnosed adults, based on data from White European populations



Holt et al., 2021

Decision trees

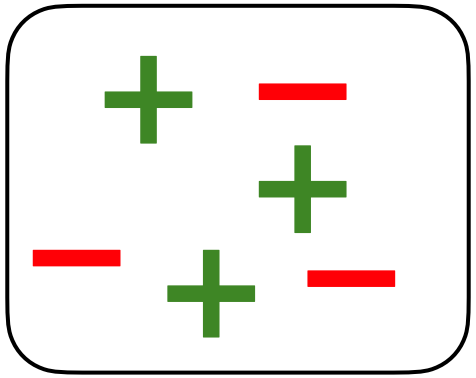
Want to classify spam from not spam using text-based features. What feature gives me the best split?



Full dataset: 50% spam, 50% not
Entropy: 1.0

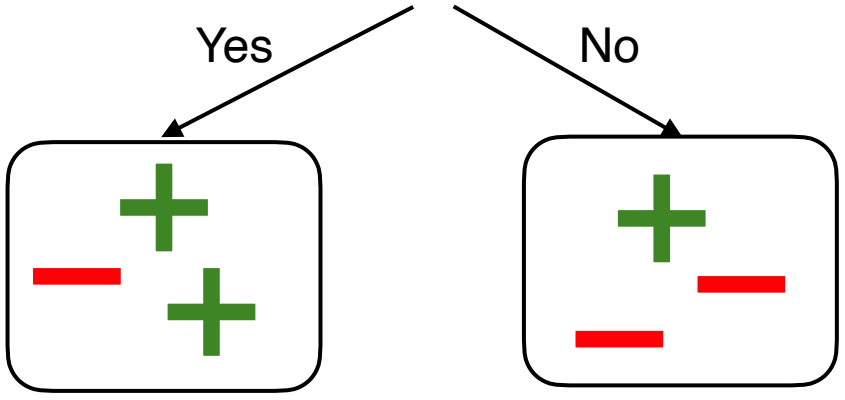
Decision trees

Want to classify spam from not spam using text-based features. What feature gives me the best split?



Full dataset: 50% spam, 50% not
Entropy: 1.0

Split on feature: contains “Limited offer”

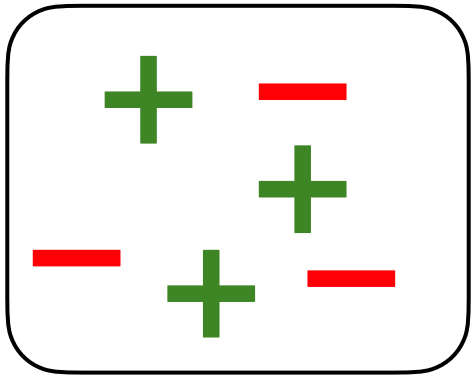


66% spam, 33% not
Entropy: 0.918

33% spam, 66% not
Entropy: 0.918

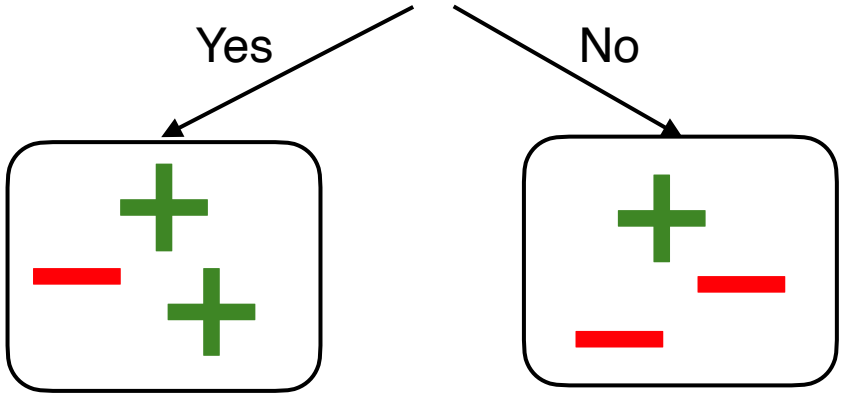
Decision trees

Want to classify spam from not spam using text-based features. What feature gives me the best split?



Full dataset: 50% spam, 50% not
Entropy: 1.0

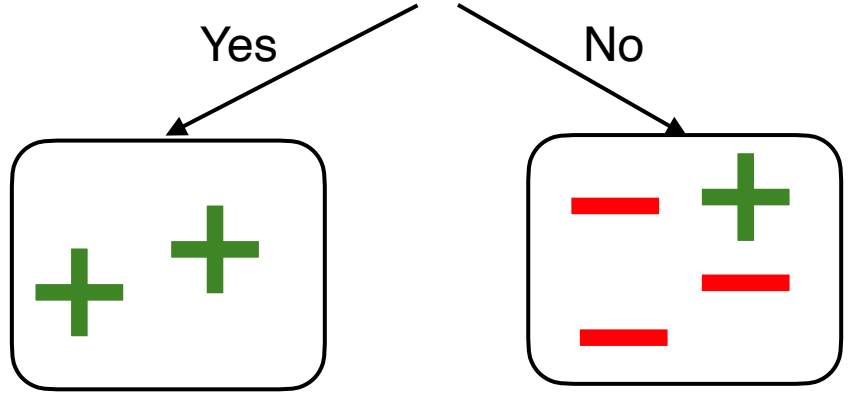
Split on feature: contains "Limited Offer"



66% spam, 33% not
Entropy: 0.918

33% spam, 66% not
Entropy: 0.918

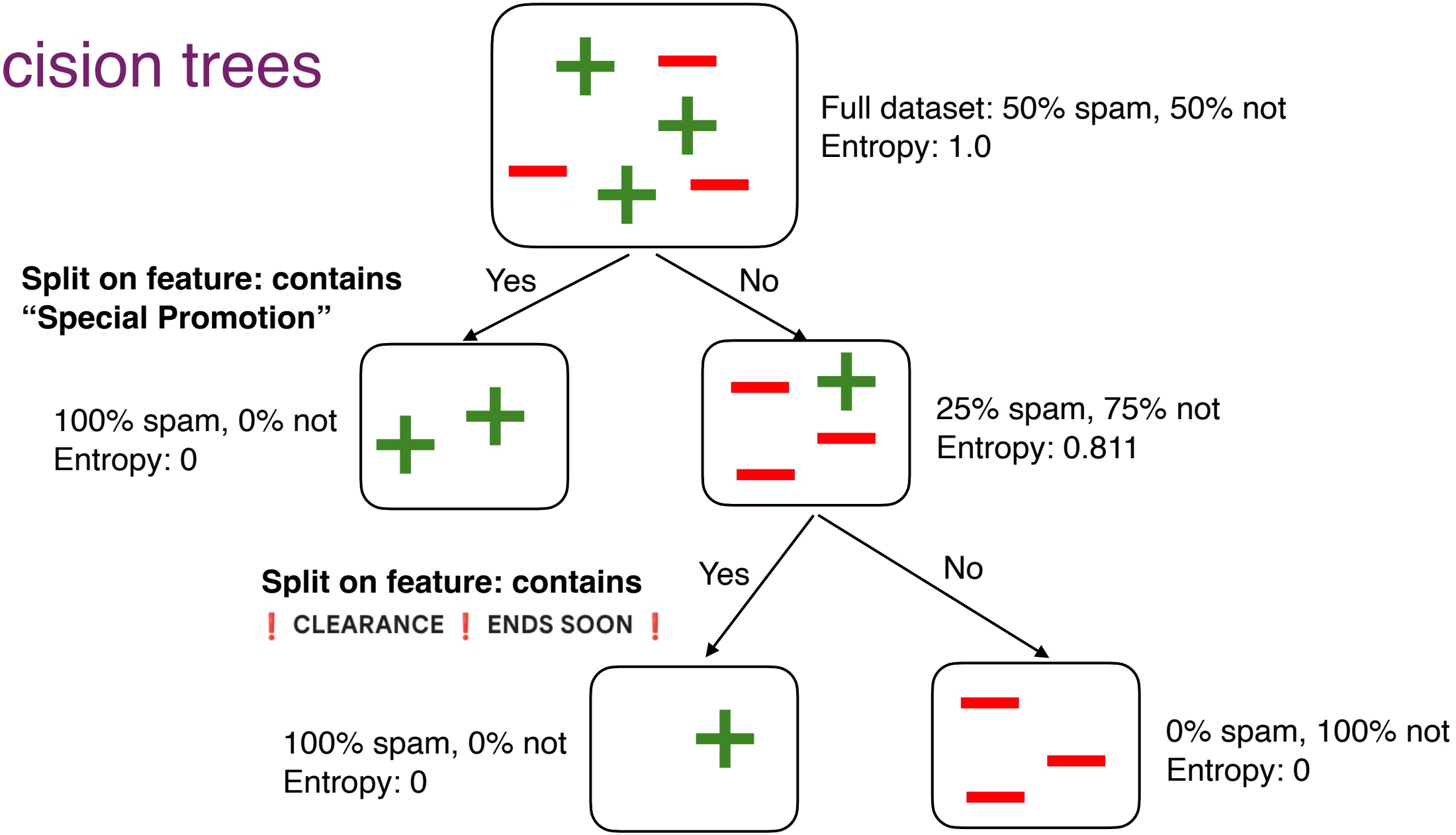
Split on feature: contains "Special Promotion"



100% spam, 0% not
Entropy: 0

25% spam, 75% not
Entropy: 0.811

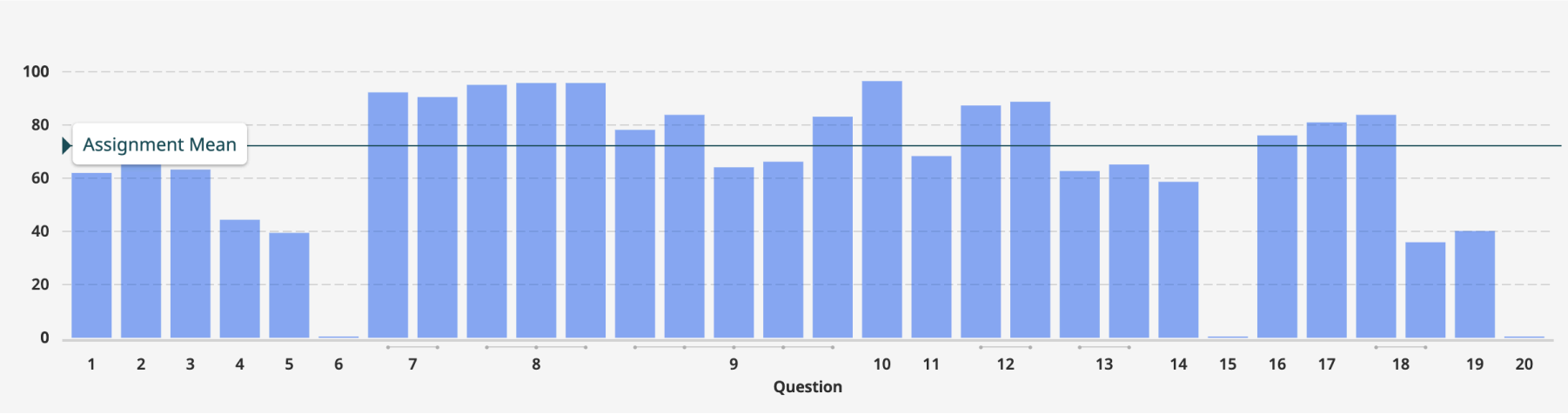
Decision trees



Today’s class is remote!

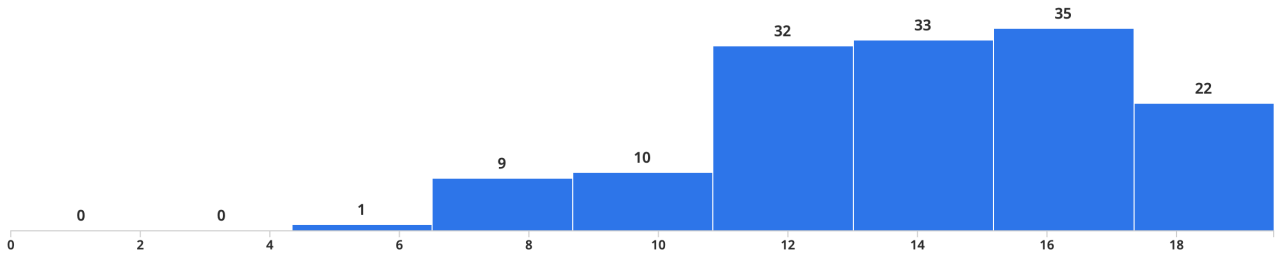
- You can use the Slido to ask questions: https://app.sli.do/event/gTDT7XWHkwhr99LLfbQeMW

Midterm statistics



Midterm Exam 19.5 points

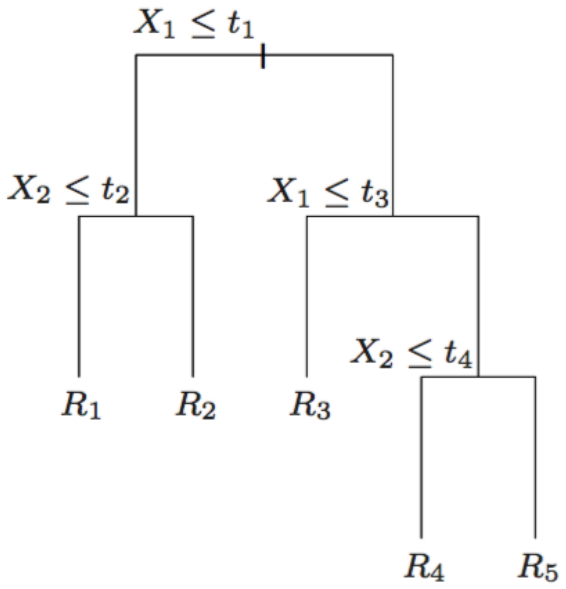
Minimum	Median	Maximum	Mean	Std Dev
30.51%	73.85%	104.1%	72.04%	15.56%

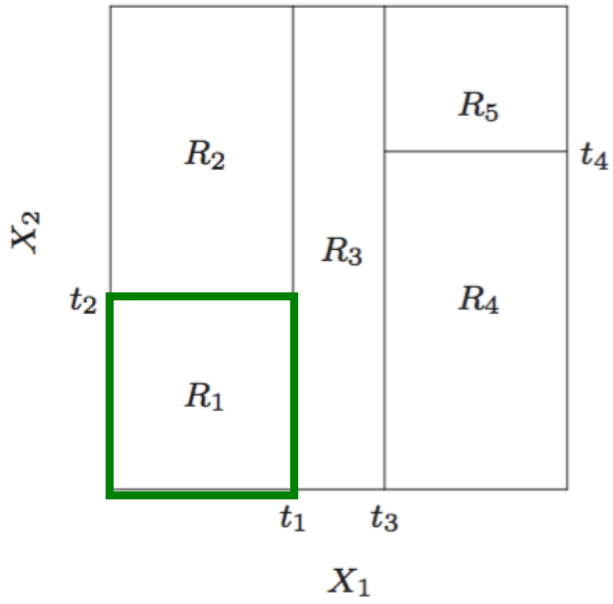


- Original average was 65%
- We dropped the two hardest questions (6 and 15) and made them bonus questions
- New class average is **72%**

Regression trees

Axis-aligned trees



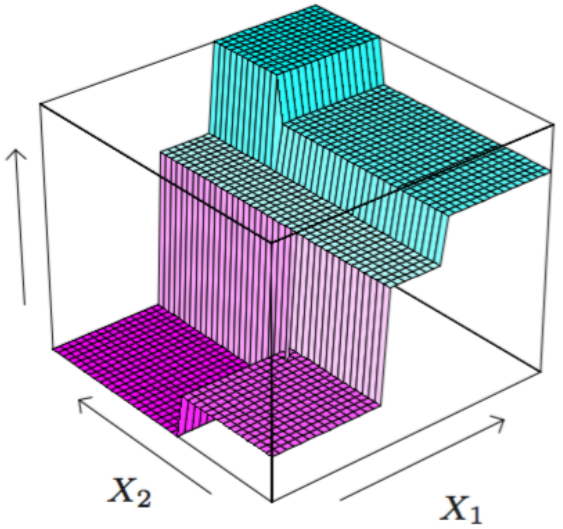


Decision boundary

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m).$$

$$I = \begin{cases} 1 & \text{if } x \in \mathbb{R}^m \\ 0 & \text{otherwise} \end{cases}$$

Actual output value for regression



When to use?

- Can be interpretable

When not to use?

- Complex, high-dimensional data (e.g. pixels)

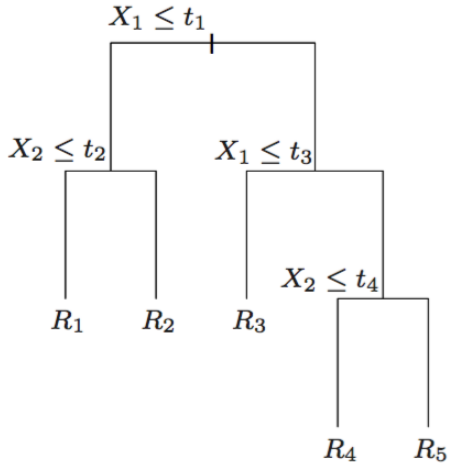
Generic algorithm for building trees

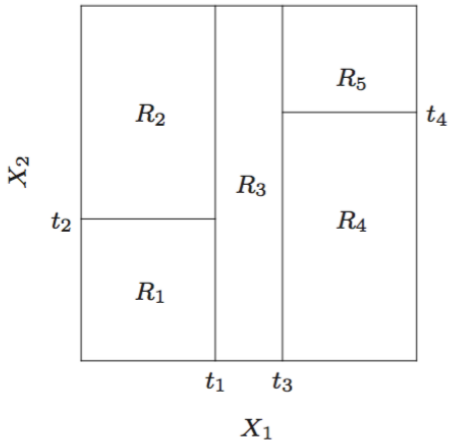
1. Start from empty decision tree
2. Recursively, for each node:
   - Iterate through all features and compute how good it'd be to split on each feature
   - Split on the "best" feature
3. Prune

Design choices:

- Termination condition (max depth, entropy, train/val error)
- Tree complexity
- Splitting criterion
- Pruning

Splitting regression trees





Overall goal: carve feature space X into M regions, $R_1 \ldots R_M$. For all the samples in a region R_m, predict some value c_m

Loss func? MSE

For region m, which has samples with labels $y_1, y_2 \ldots y_n$

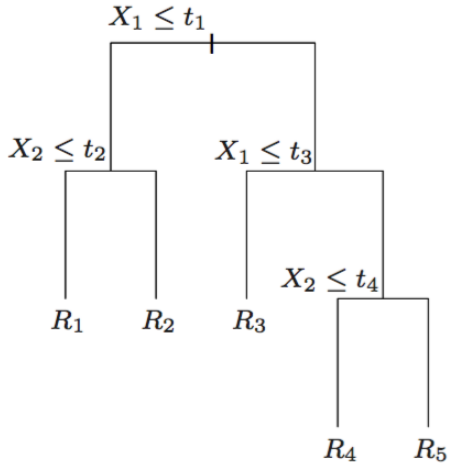
$$\min_{c_m \in \mathbb{R}} \sum_i (y_i - c_m)^2$$ # How do we find c_m?

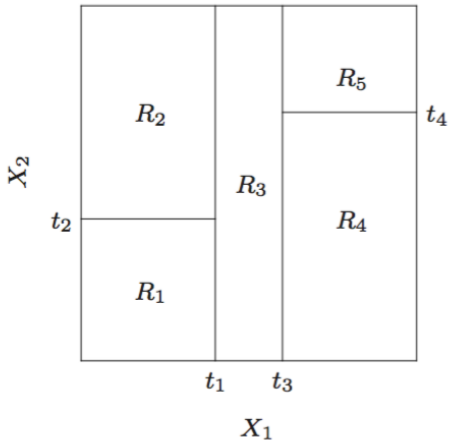
$$\frac{d}{dc} \sum_i (y_i - c)^2 = 0$$ # Take the derivative and set it to zero!

$$= -2 \sum_i (y_i - c) = 0$$

$$nc = \sum_i y_i \quad \rightarrow \boxed{c = \frac{1}{n} \sum_i y_i}$$ # Just the mean of the y values

Splitting regression trees





Overall goal: carve feature space X into M regions, $R_1 \ldots R_M$. For all the samples in a region R_m, predict some value c_m

So what is the function that predicts c_m?

$$f(x) = \sum_{m=1}^{M} c_m 1\{x \in R_m\}$$

$$R_1(j,s) = \{x, y \,|\, x_j \le s\} \rightarrow c_1$$

$$R_2(j,s) = \{x, y \,|\, x_j > s\} \rightarrow c_2$$

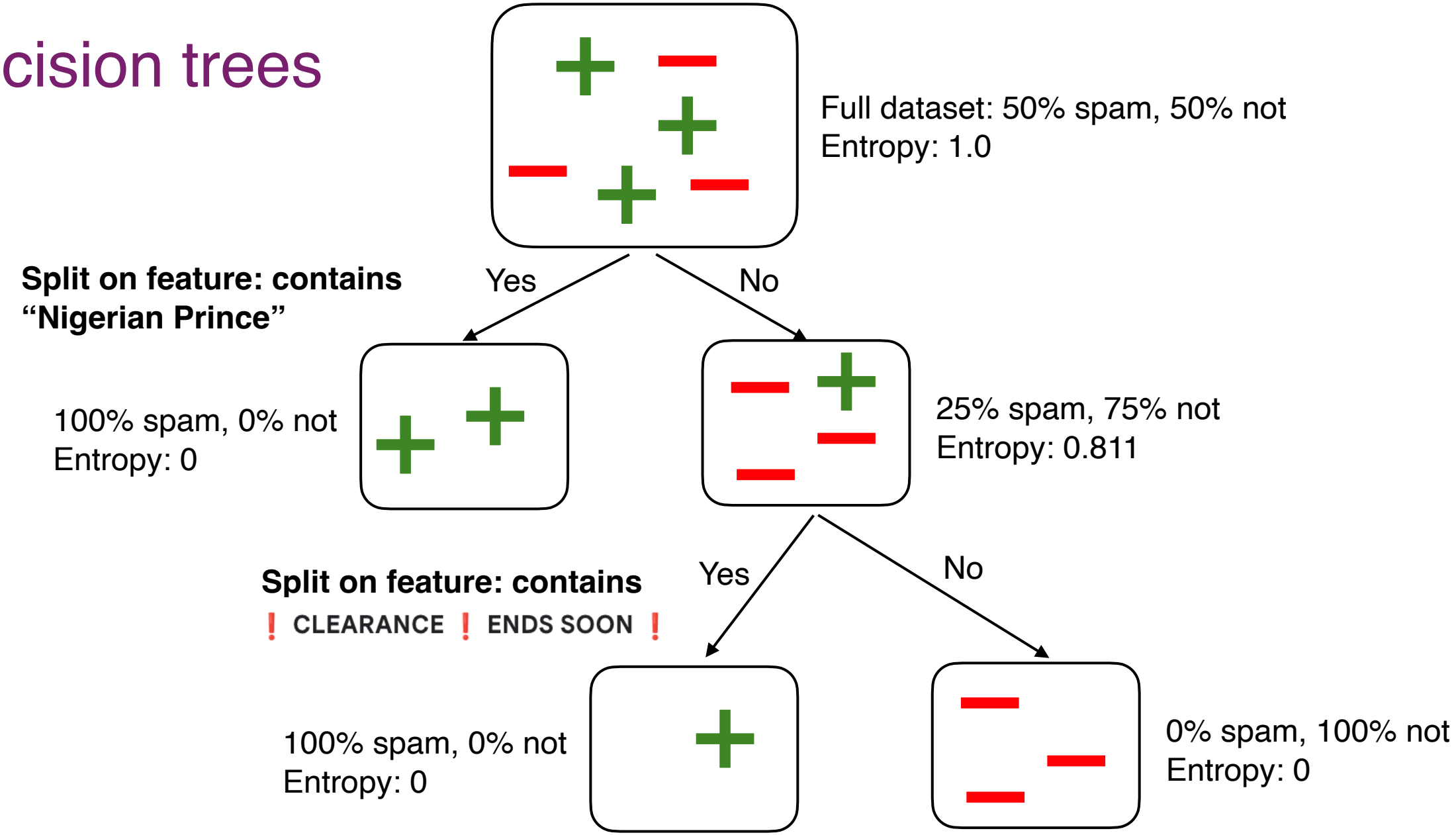
So how to learn which feature j, where to split s, and c's?

$$\arg\min_{j,s} \left[\min_{c_1} \sum_{(x_i,y_i \in R_1(j,s))} (y - c_1)^2 + \min_{c_2} \sum_{(x_i,y_i \in R_2(j,s))} (y - c_2)^2 \right]$$

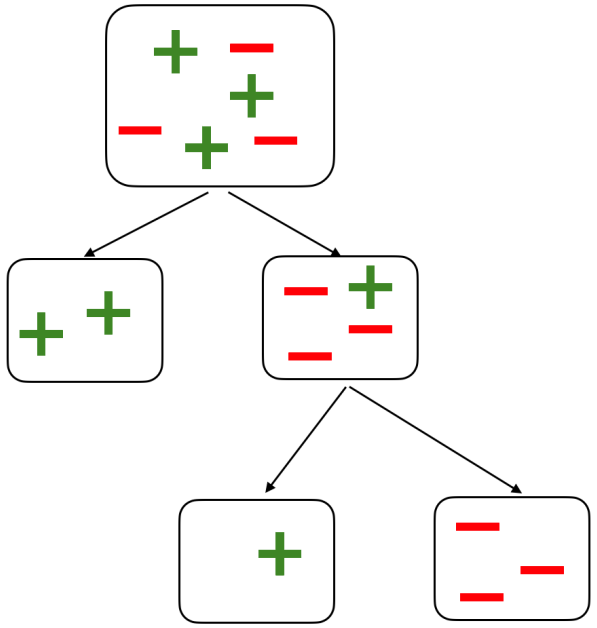
Loss on LHS of tree # Loss on RHS of tree

$$c_1 = \frac{1}{|R_1|} \sum_{(x_i,y_i \in R_1)} y$$ # From last slide

Decision trees



Splitting decision trees



Overall goal: choose features which can best sort positive and negative examples. For all the samples in a region R_m, predict the majority class c_m

Loss func? 0/1 error / binary classification error

$$f(x) = \sum_{m=1}^{M} c_m 1\{x \in R_m\}$$ # Same

$$\text{split on } x_j \le s, \quad x_j > s$$

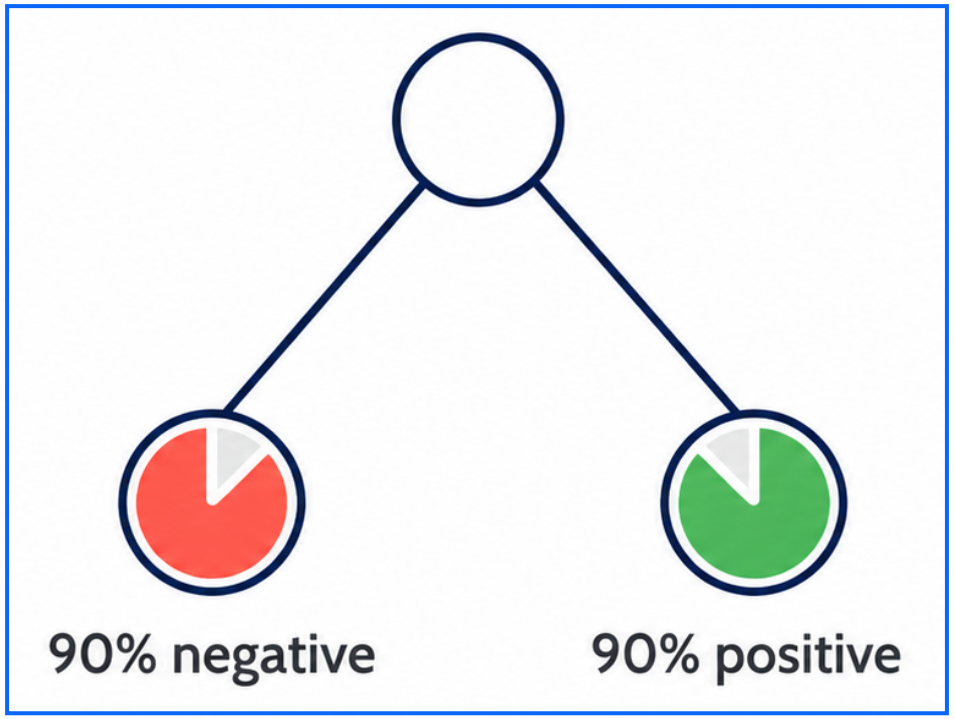
Same for illustrative purposes. In reality you can also have categorical features. E.g. “Islet Antibody Positive Test”, or “Blood Type”

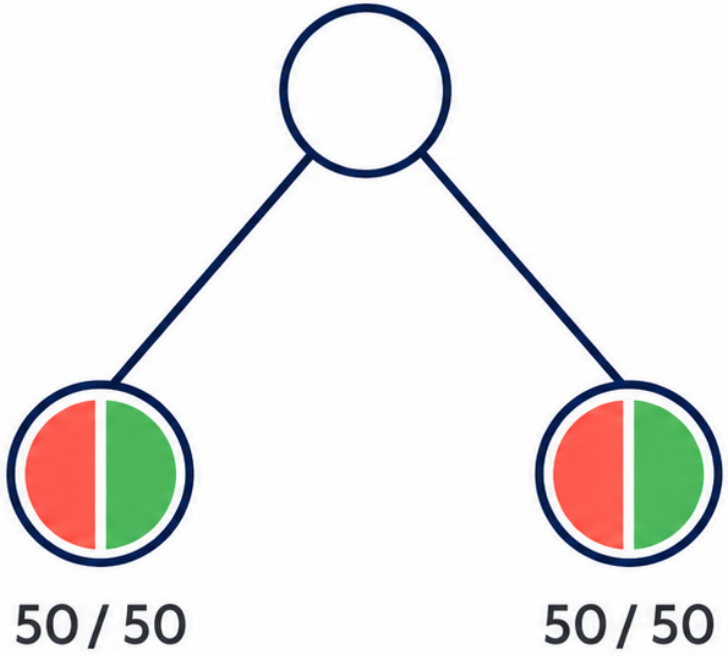
Define “impurity” measure I (measures how well positive class examples are sorted from negative)

Find split that minimizes impurity

Splitting decision trees

Find split that minimizes impurity… so which tree should I prefer?





What is the classification accuracy of each tree on the training data?

90%

50%

Entropy

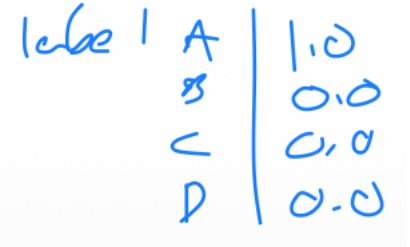
Entropy: $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$

Example: $Y = \{0, 0, 1\}, \quad p_0 = \frac{2}{3}, \quad p_1 = \frac{1}{3}$

$$H = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) \approx 0.918$$

Entropy range?

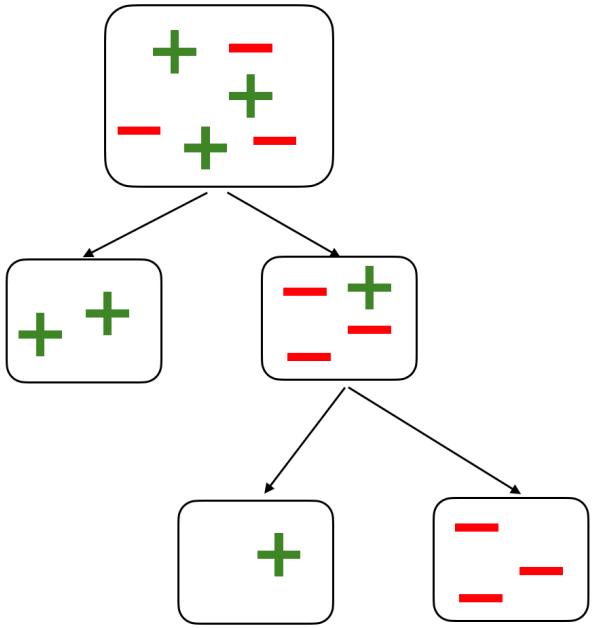
0 when data is perfectly sorted



1 for a uniform distribution

Impurity Measure: Information Gain

Information Gain: $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$



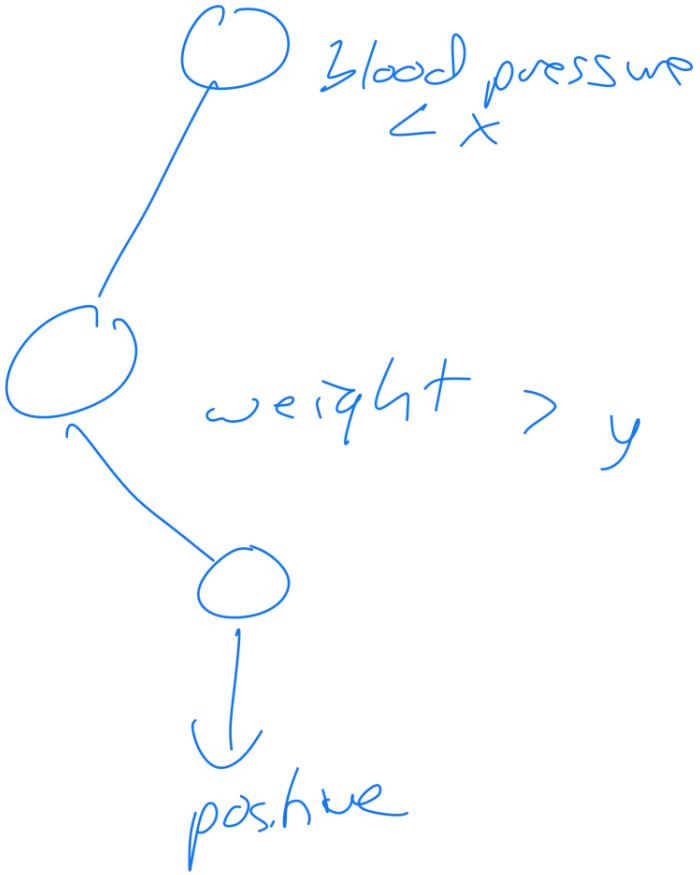
For decision trees, the decision criteria for splitting n points X based on whether feature x_j <= s, resulting in k=2 regions:

$$= H(X) - \sum_{i=1}^{k} \frac{|R_i(j,s)|}{n} H(R_i(j,s))$$

Interpreting trees

Trees are “easy” to interpret:

- You can explain <u>how</u> the classifier came to the conclusion it did



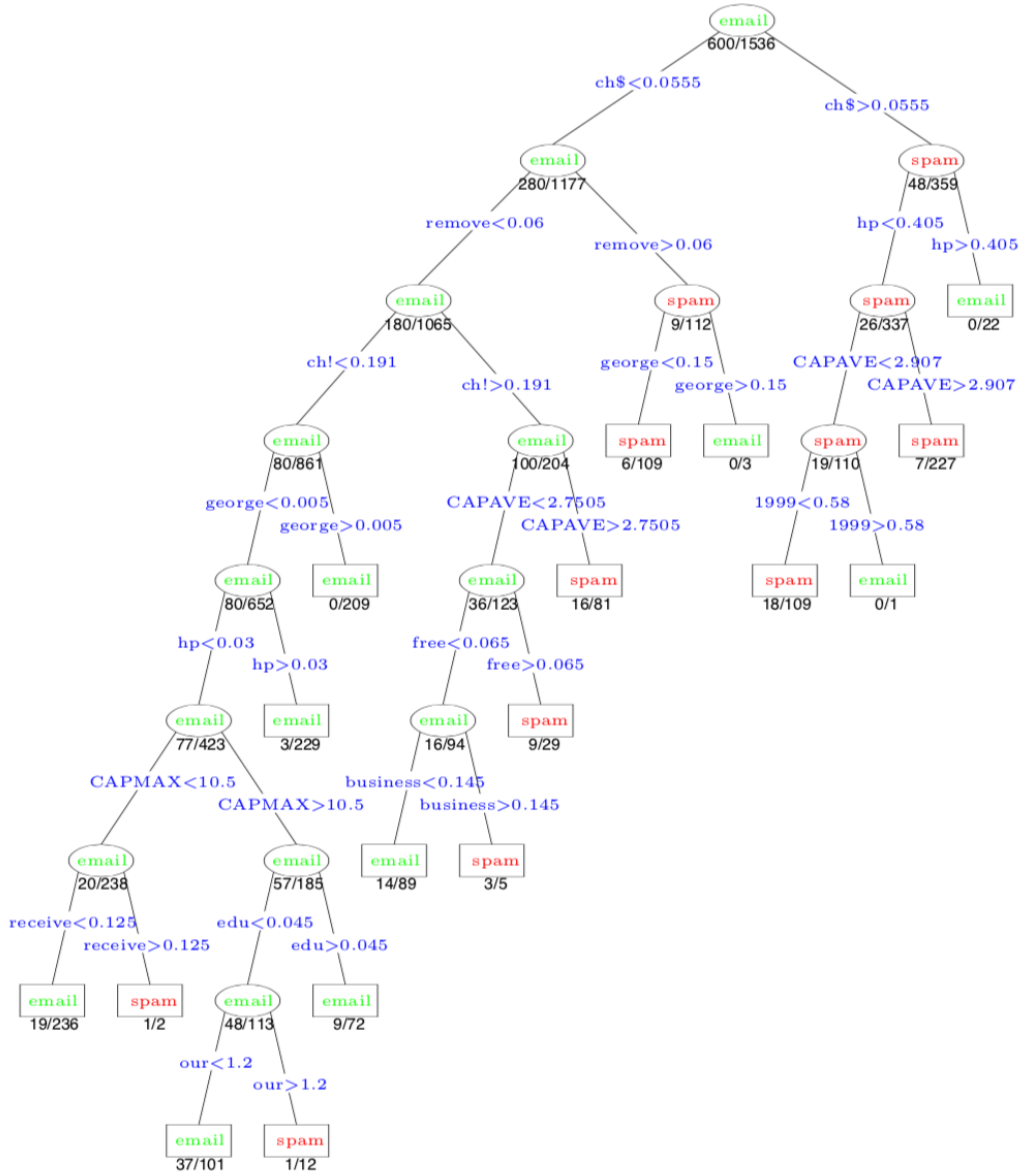
Interpreting trees

Trees are “easy” to interpret:

- You can explain how the classifier came to the conclusion it did

But they can be complex

- Small changes in data can result in large difference in trees



Summary so far

- Trees have low bias, high variance
- Deal with categorial variables well
- Intuitive, “interpretable”
- Good software exists
- Some theoretical guarantees

Where don’t they work?

Many related continuous features (pixels)

Why low bias? If you allow enough depth / splits, you can fit anything

How to regularize? We will say more, but you can limit the number of nodes, the depth, or the requirements on entropy for leaves