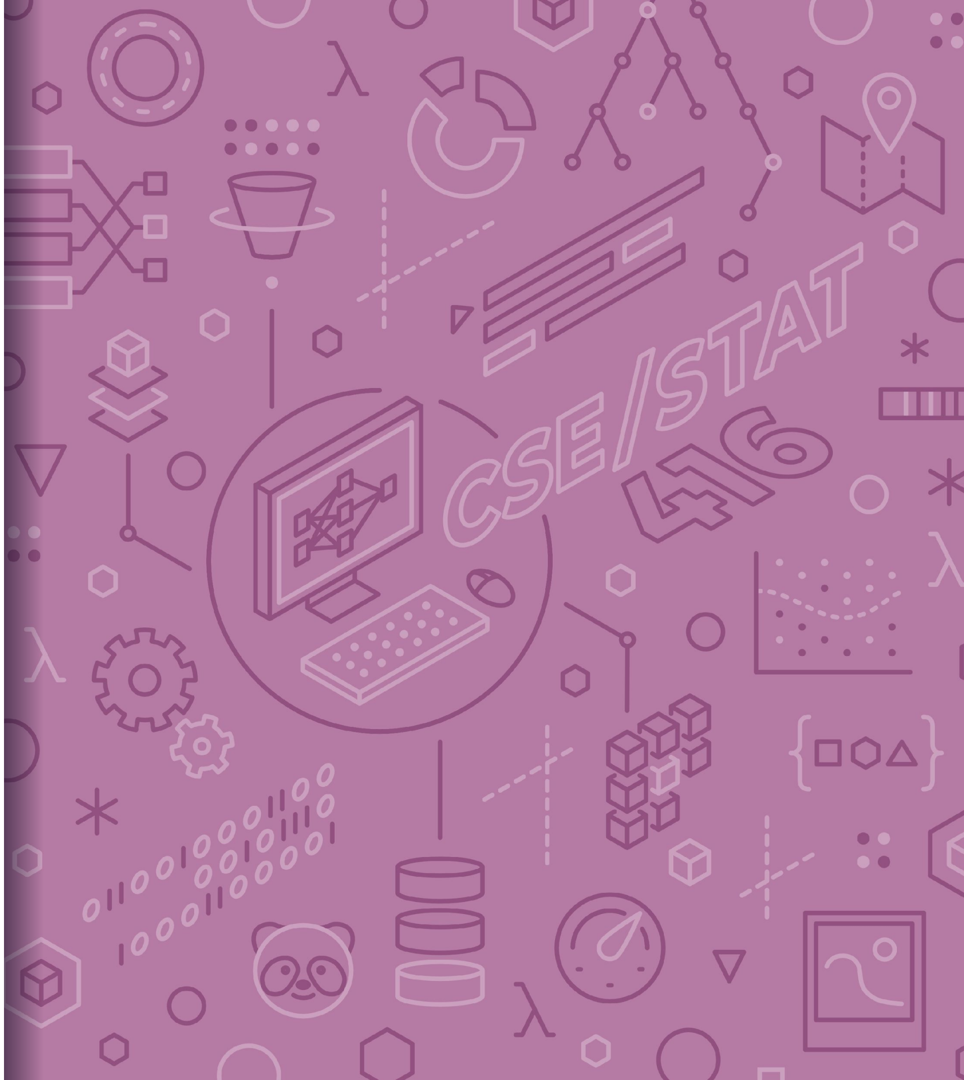
CSE/STAT 416

Naïve Bayes and Decision Trees

Tanmay Shah
Paul G. Allen School of Computer Science & Engineering
University of Washington

April 24, 2024

❓ **Questions?** Raise hand or **sli.do #cs416**
💬 **Before Class:** Pro-rain or anti-rain person?
🎵 **Listening to:** lecture



Administrivia

- Midterm due tonight
- - Post questions on Edstem (Private post as needed)

- HW3 out Friday

Probability Classifier

Idea: Estimate probabilities $\hat{P}(y|x)$ and use those for prediction

Probability Classifier

Input x: Sentence from review

- Estimate class probability $\hat{P}(y = +1|x)$
- If $\hat{P}(y = +1|x) > 0.5$:
  - $\hat{y} = +1$
- Else:
  - $\hat{y} = -1$

Notes:

- Estimating the probability improves **interpretability**

Interpreting Score

$$Score(x_i) = w^T h(x_i)$$

$-\infty$ $\quad \hat{y}_i = -1 \quad$ 0 $\quad \hat{y}_i = +1 \quad$ ∞

Very sure
$\hat{y}_i = -1$

Not sure if
$\hat{y}_i = -1 \; or \; \hat{y}_i = +1$

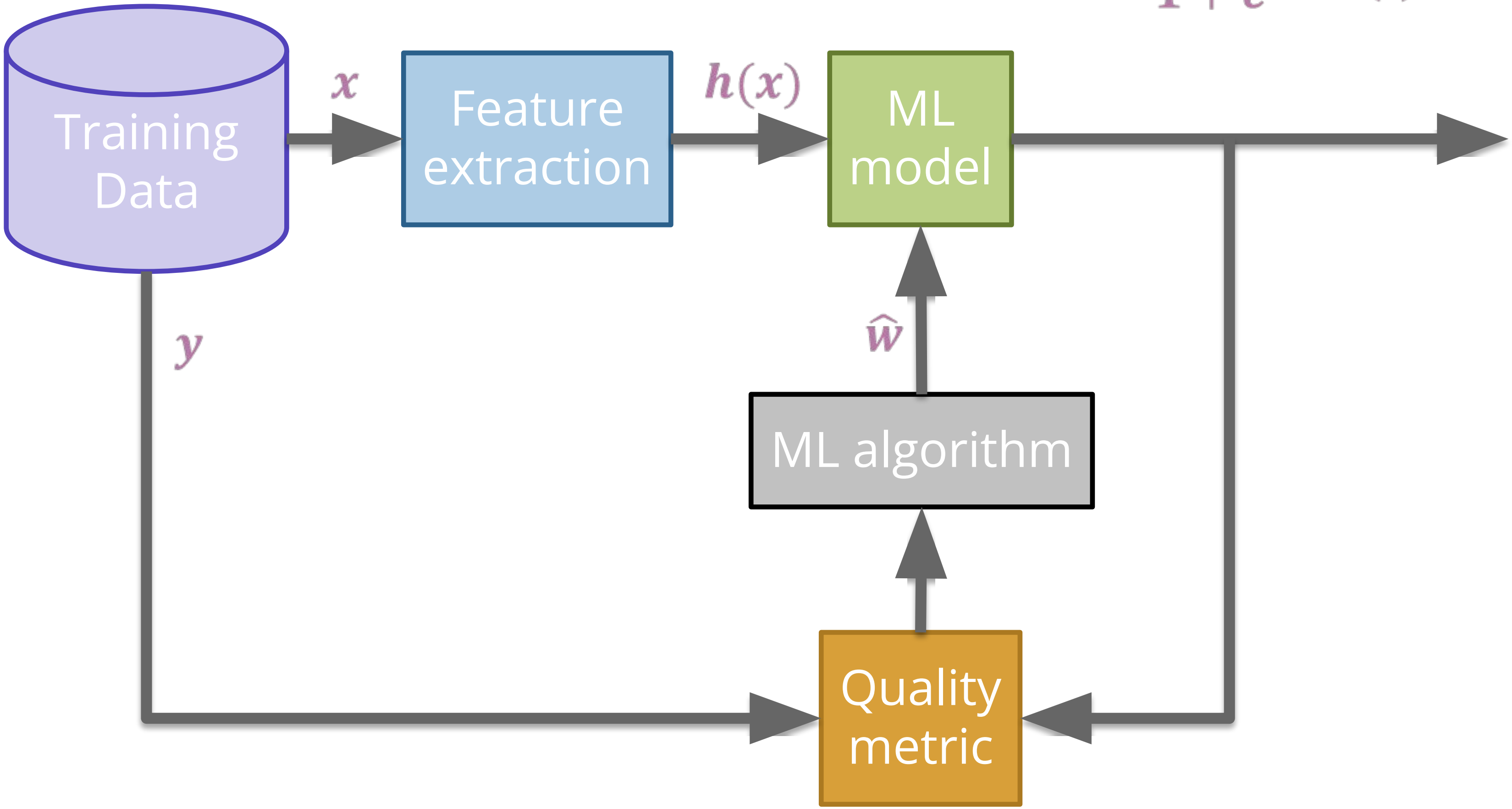
Very sure
$\hat{y}_i = +1$

$\hat{P}(y_i = +1|x_i) = 0$ $\quad$ $\hat{P}(y_i = +1|x_i) = 0.5$ $\quad$ $\hat{P}(y_i = +1|x_i) = 1$

0 $\quad$ 1

$$\hat{P}(y = +1\,|x)$$

$$\hat{P}(y=+1|x,\hat{w}) = sigmoid\left(\hat{w}^T h(x)\right) = \frac{1}{1+e^{-\hat{w}^T h(x)}}$$



Naïve Bayes

Idea: Naïve Bayes

$$x = \text{“The sushi \& everything else was awesome!”}$$

$$P\left(y = +1 \,\middle|\, x = \text{“The sushi \& everything else was awesome!”}\right)?$$

$$P\left(y = -1 \,\middle|\, x = \text{“The sushi \& everything else was awesome!”}\right)?$$

Idea: Select the class that is the most likely!

Bayes Rule:

$$P(y = +1|x) = \frac{P(x|y = +1)P(y = +1)}{P(x)}$$

Example

$$\frac{P\left(\text{“The sushi \&everything else was awesome!”} \,\middle|\, y = +1\right) P(y = +1}{P(\text{“The sushi \& everything else was awesome!”})}$$

Since we're just trying to find out which class has the greater probability

Naïve Assumption

Idea: Select the class with the highest probability!

Problem: We have not seen the sentence before.

Assumption: Words are independent from each other.

$x = \text{“}The\ sushi\ \&\ everything\ else\ was\ awesome!\text{”}$

$$\frac{P(\text{“}The\ sushi\ \&\ everything\ else\ was\ awesome!\text{”} | y = +1)\ P(y = +1)}{P(\text{“}The\ sushi\ \&\ everything\ else\ was\ awesome!\text{”})}$$

$$\begin{aligned} &P(\text{“}The\ sushi\ \&everything\ else\ was\ awesome!\text{”}\ | y = +1) \\ &= P(\text{The} \mid \text{y=+1}) * \text{P}(sushi\ | y = +1) * P(\& | y = +1) \\ &\quad * P(everything | y = +1) * P(else | y = +1) * P(was | y = +1) \\ &\quad * P(awesome | y = +1) \end{aligned}$$

Compute Probabilities

How do we compute something like

$$P(y = +1)?$$

How do we compute something like

$$P(\text{“}awesome\text{”} \mid y = +1)?$$

Zeros

If a feature is missing in a class everything becomes zero.

$$P(\text{“}The\ sushi\ \&everything\ else\ was\ awesome!\text{”}\,|y=+1)$$
$$= P(\text{The} \mid \text{y=+1}) * \text{P}(sushi\ |y=+1) * P(\&|y=+1)$$
$$* P(everything|y=+1) * P(else|y=+1) * P(was|y=+1)$$
$$* P(awesome|y=+1)$$

Solutions?

- Take the log (product becomes a sum).
  - Generally define $\log(0) = 0$ in these contexts
- Laplacian Smoothing (adding a constant to avoid multiplying by zero)

Compare Models

Logistic Regression:

$$P(y = +1|x, w) = \frac{1}{1 + e^{-w^T h(x)}}$$

Naïve Bayes:

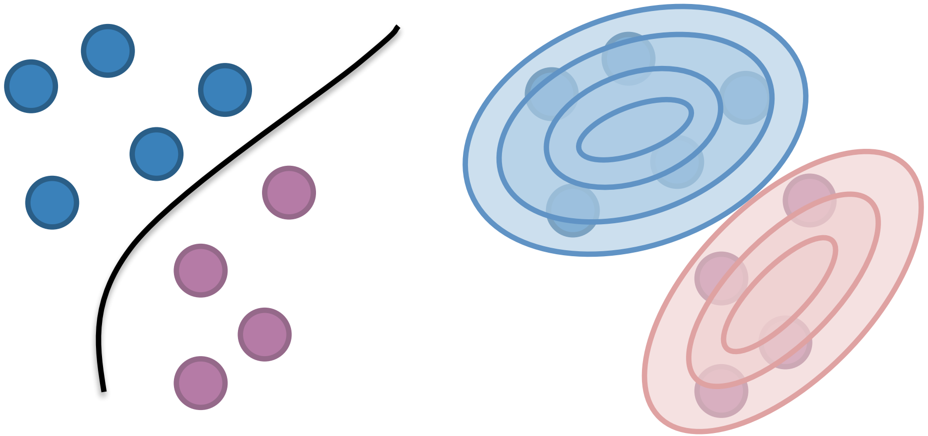
$$P(y|x_1, x_2, \dots, x_d) = \prod_{j=1}^{d} P(x_j|y)\ P(y)$$

- Based on counts of words/classes
  - Laplace Smoothing

Compare Models

Generative: defines a model for generating x (e.g. Naïve Bayes)

Discriminative: only cares about defining and optimizing a decision boundary (e.g. Logistic Regression)



slido

Group

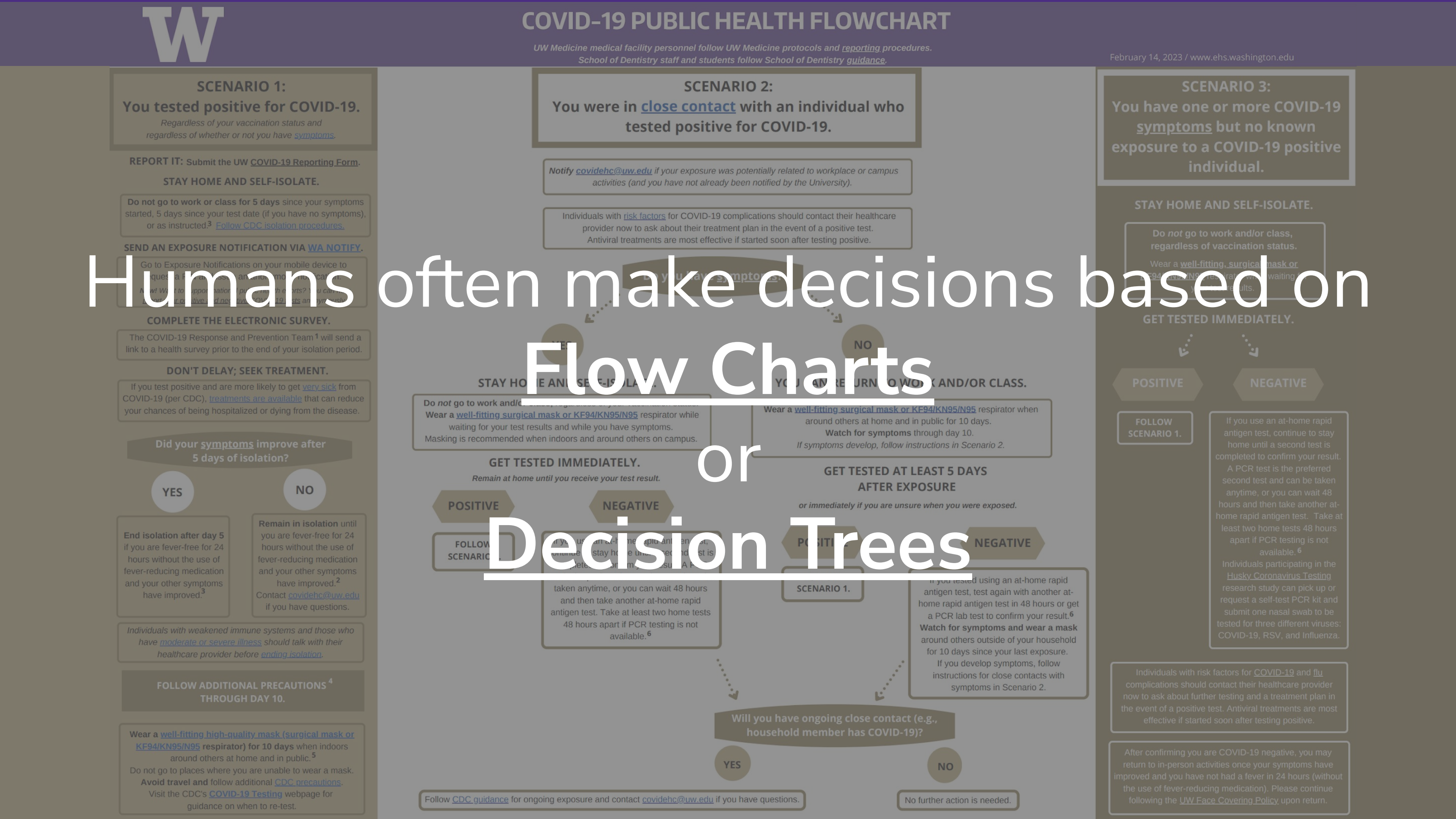
2 min

sli.do #cs416

Recap: What is the predicted class for this sentence assuming we have the following training set (no Laplace Smoothing).
“he is not cool”

Sentence	Label
this dog is cute	Positive
he does not like dogs	Negative
he is not bad he is cool	Positive

Decision Trees



Humans often make decisions based on

Flow Charts

or

Decision Trees

Parametric vs. Non-Parametric Methods

Parametric Methods: make assumptions about the data distribution

- Linear Regression ⇒ assume the data is linear
- Logistic Regression ⇒ assume probability has the shape of of a logistic curve and linear decision boundary
- Those assumptions result in a *parameterized* function family. Our learning task is to learn the parameters.

Non-Parametric Methods: (mostly) don't make assumptions about the data distribution

- Decision Trees, k-NN (soon)
- We're still learning something, but not the parameters to a function family that we're assuming describes the data.
- Useful when you don't want to (or can't) make assumptions about the data distribution.

XOR

- A line might not always support our decisions.

What makes a loan risky?

I want to buy a new house!

Loan Application

Credit History
★★★★

Income
★★★

Term
★★★★★

Personal Info
★★★

Credit history explained

Did I pay previous loans on time?

Example:
excellent, good, or fair

Credit History
★★★★

Income
★★★

Term
★★★★★

Personal Info
★★★

Income

What's my income?

Example:
$80K per year

Credit History
★★★★

Income
★★★

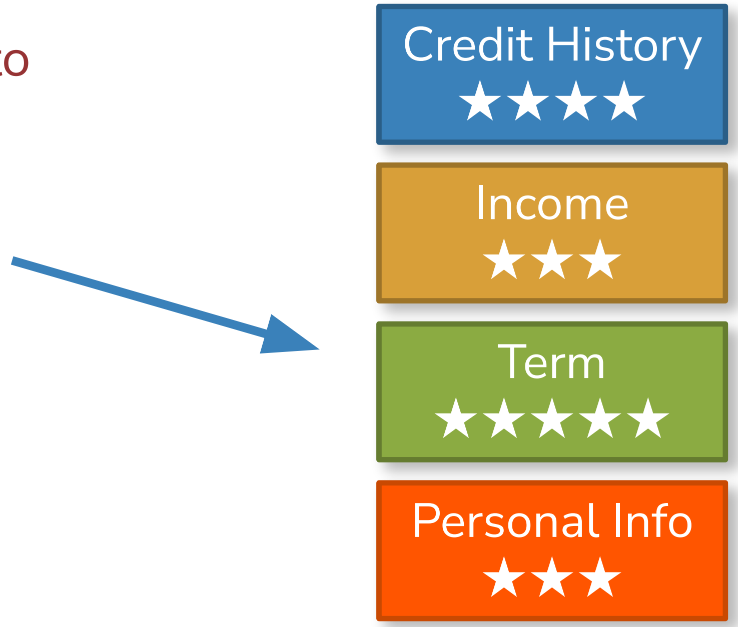
Term
★★★★★

Personal Info
★★★

Loan terms

How soon do I need to pay the loan?

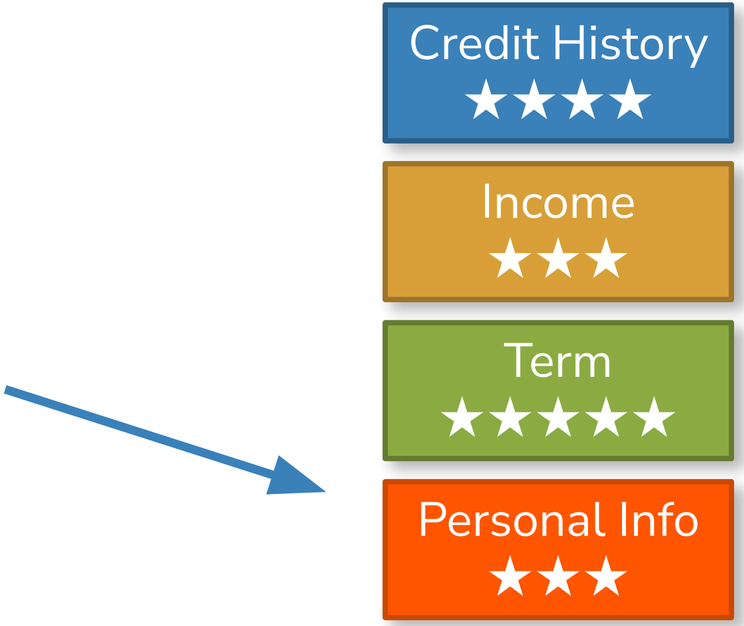
Example: 3 years, 5 years,...



Personal information

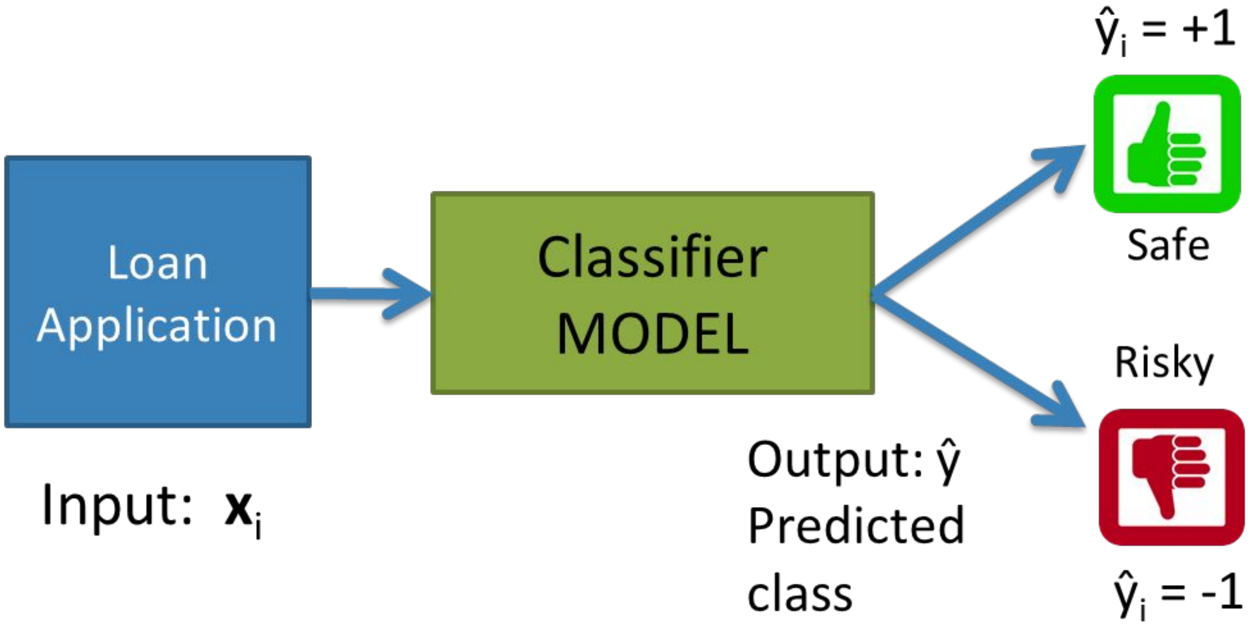
Age, reason for the loan, marital status,...

Example: Home loan for a married couple



Intelligent application

Classifier review



Setup

Data (N observations, 3 features)

Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	safe
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe

Evaluation: classification error

Many possible decisions: number of trees grows exponentially!

Poll Everywhere

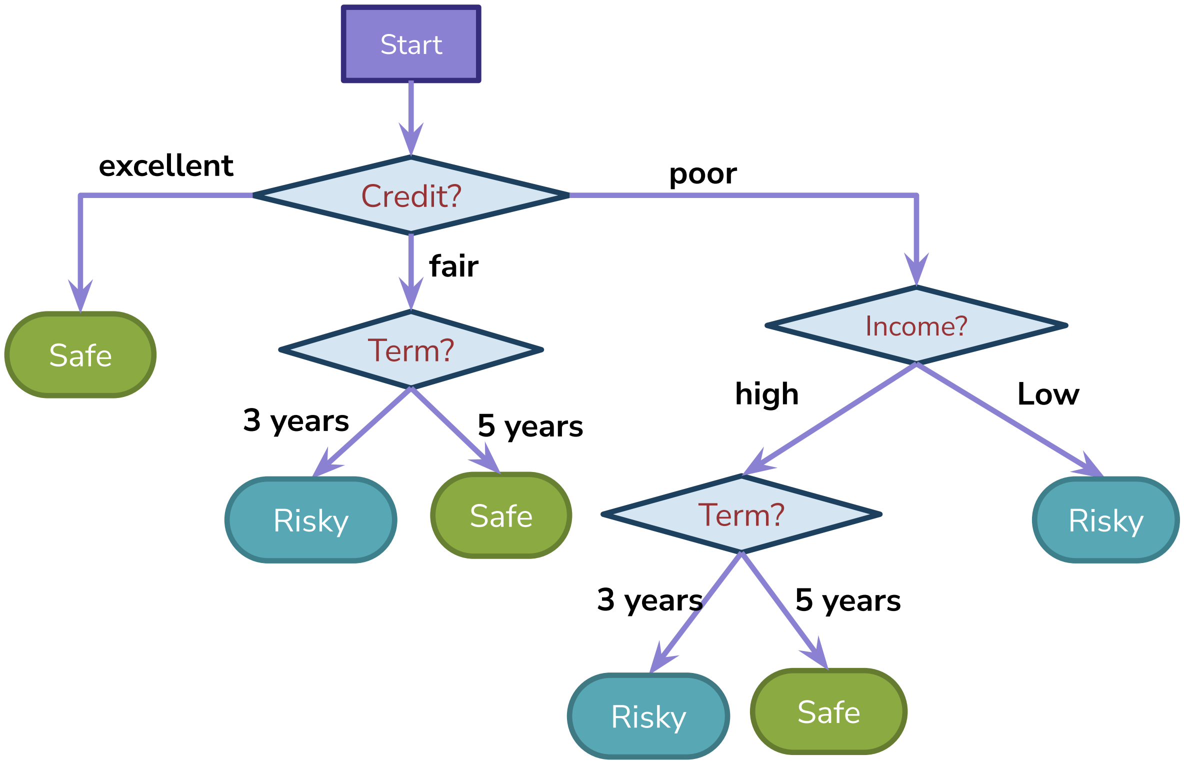
Think

2 min

No poll

With our discussion of bias and fairness from last week, discuss the potential biases and fairness concerns that might be present in our dataset about loan safety.

Decision Trees

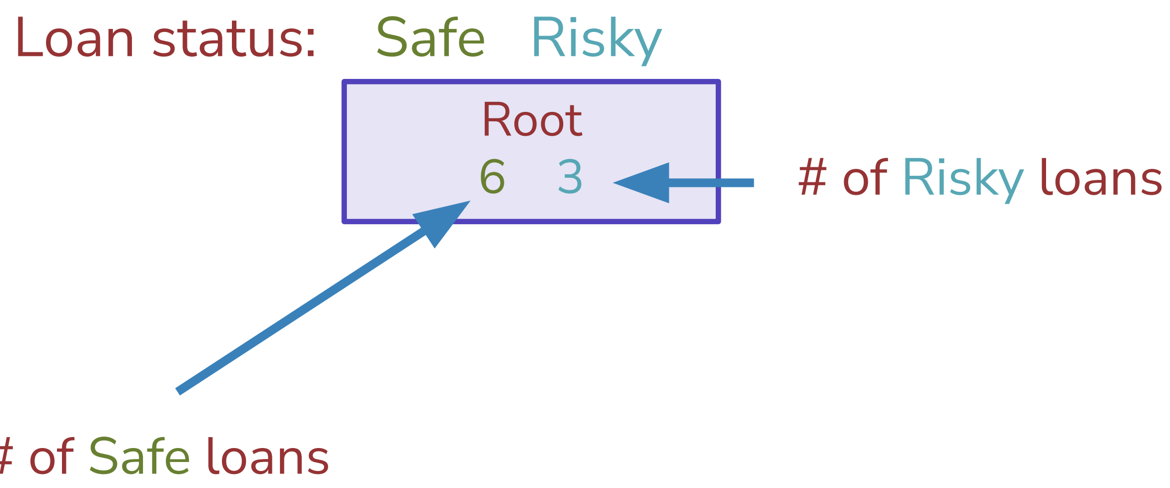


- **Branch/Internal node:** splits into possible values of a feature
- **Leaf node:** final decision (the class value)

Brain Break

Growing Trees

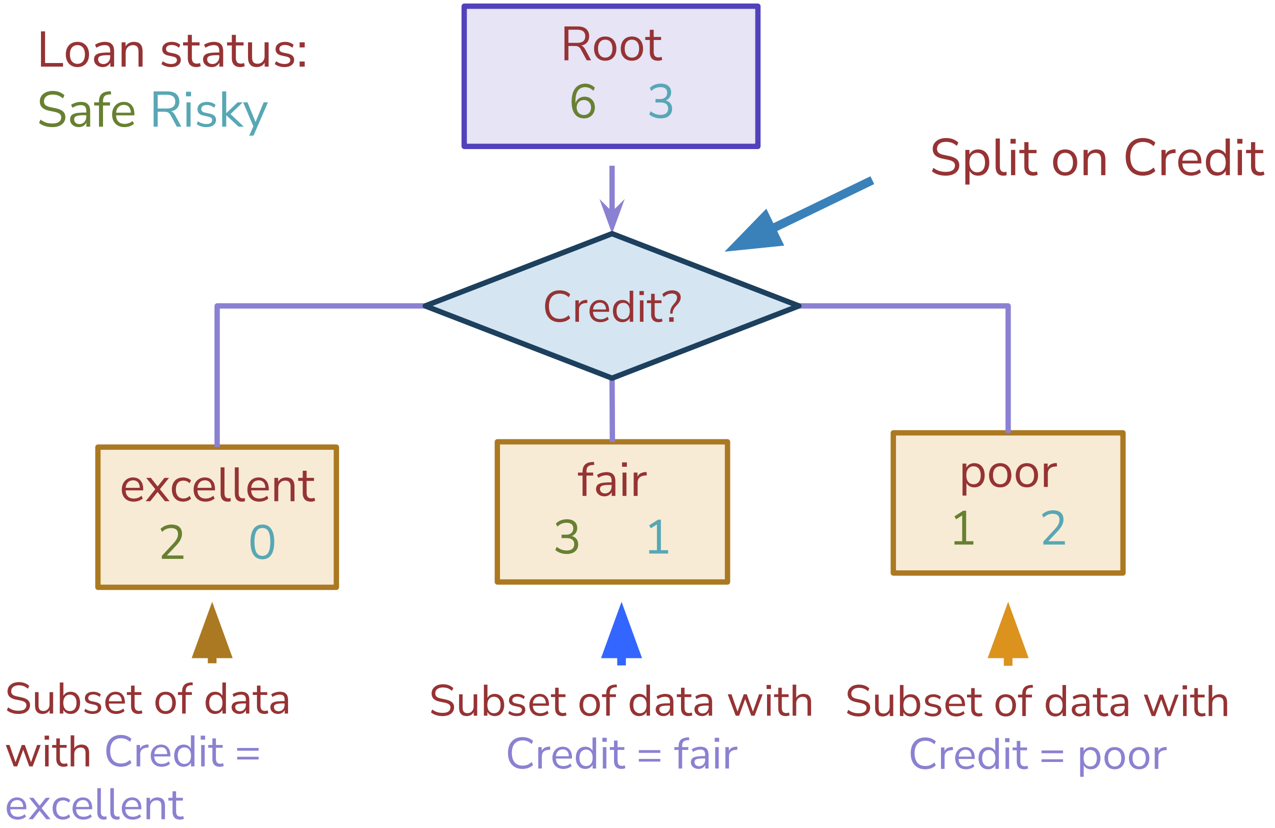
Visual Notation



N = 9 examples

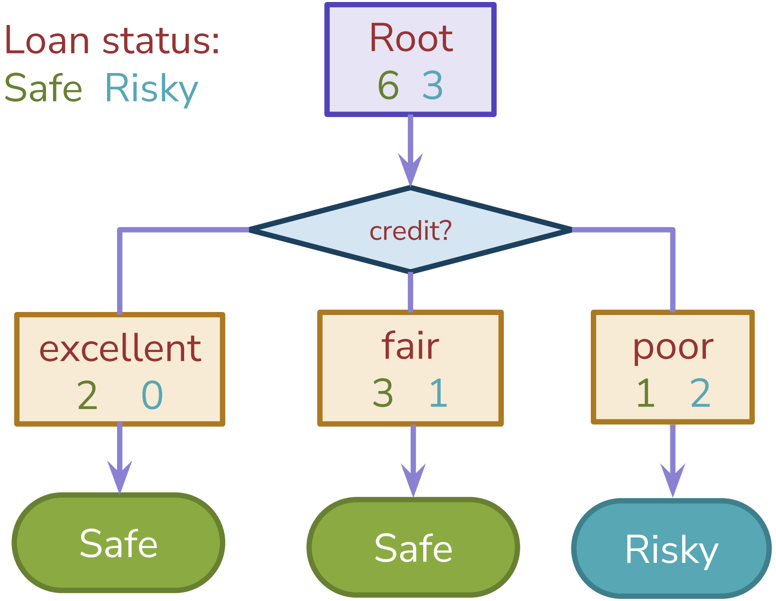
Decision stump: 1 level

Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	safe
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe



Making predictions

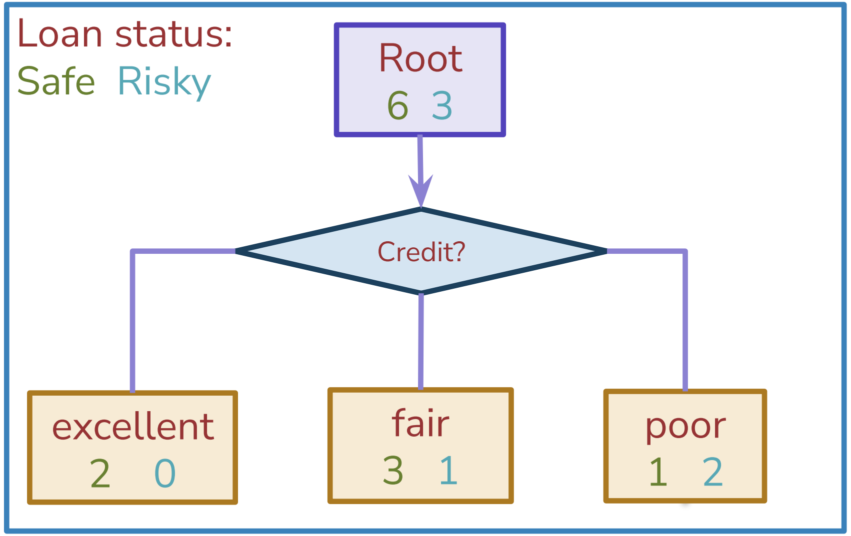
For each leaf node, set $\hat{y}$ = majority value



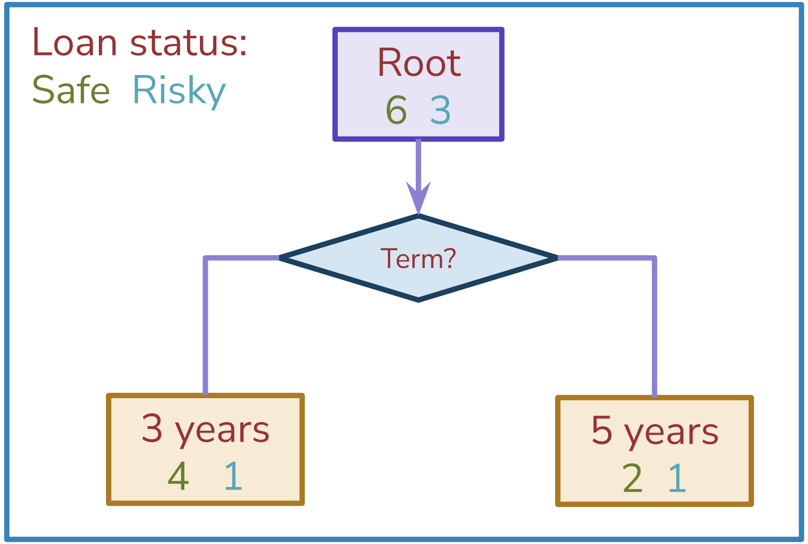
How do we select the best feature?

- Select the split with lowest classification error

Choice 1: Split on Credit



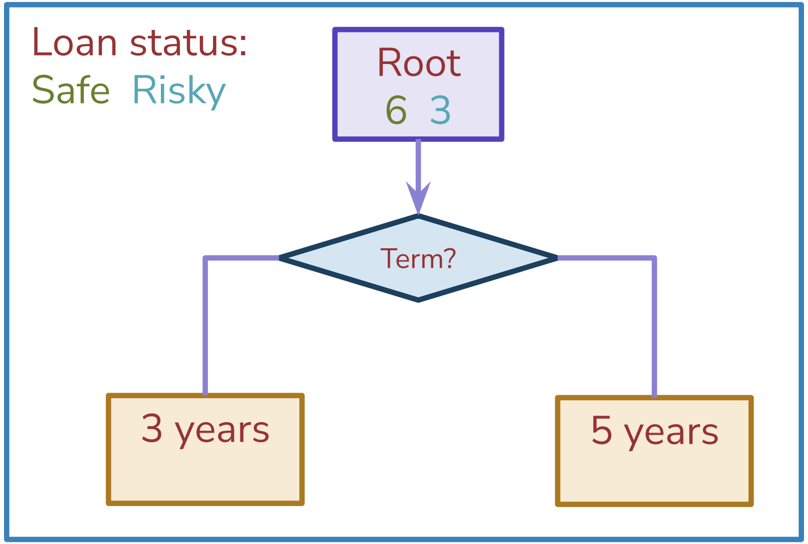
Choice 2: Split on Term



Calculate the node values.

Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	safe
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe

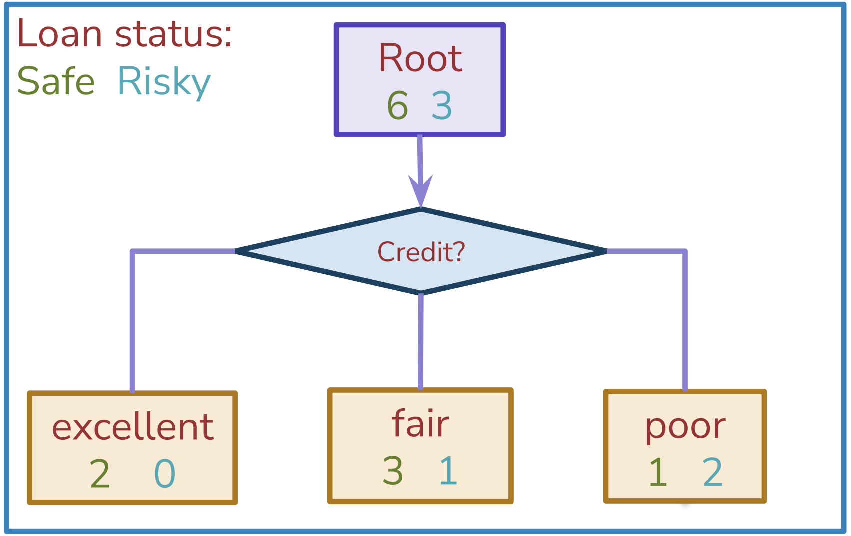
Choice 2: Split on Term



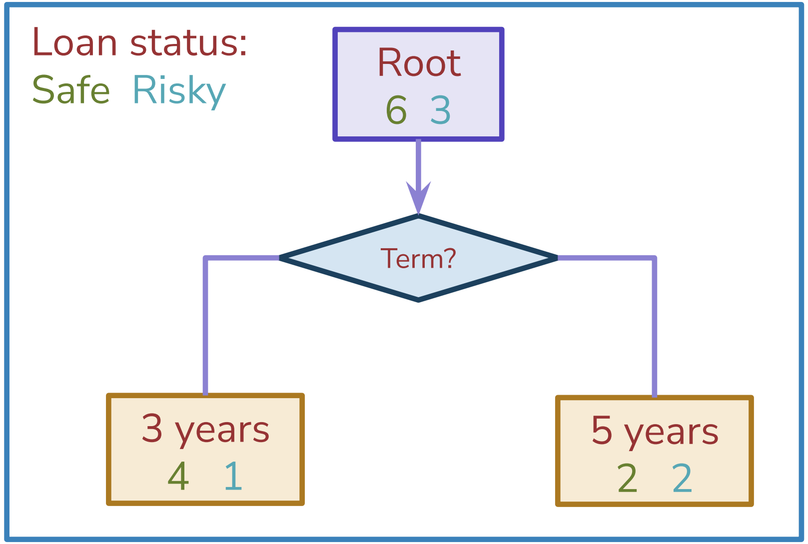
How do we select the best feature?

Select the split with lowest classification error

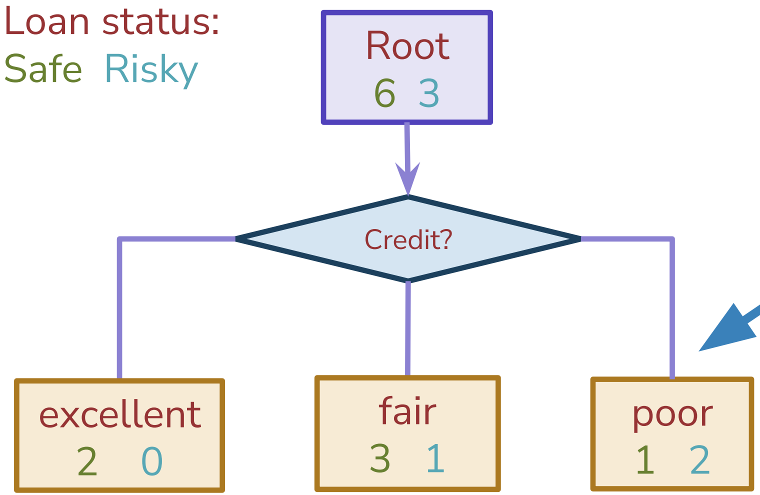
Choice 1: Split on Credit



Choice 2: Split on Term



How do we measure effectiveness of a split?



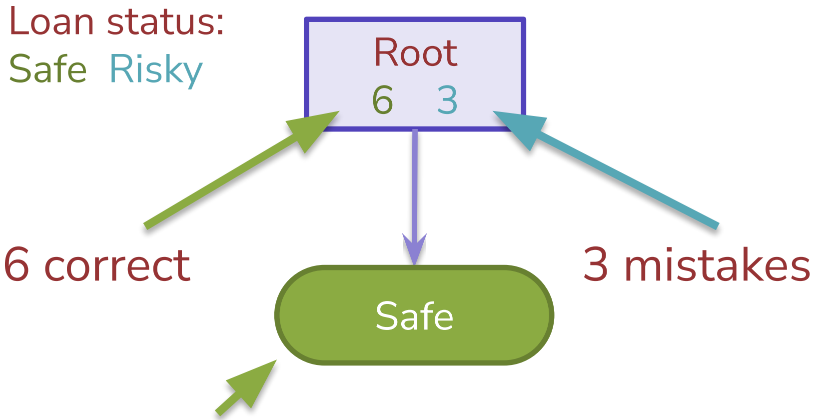
Idea: Calculate classification error of this decision stump

$$\text{Error} = \frac{\#\text{ mistakes}}{\#\text{ data points}}$$

Calculating classification error

Step 1: $\hat{y}$ = class of majority of data in node

Step 2: Calculate classification error of predicting $\hat{y}$ for this data



$\hat{y}$ = majority class

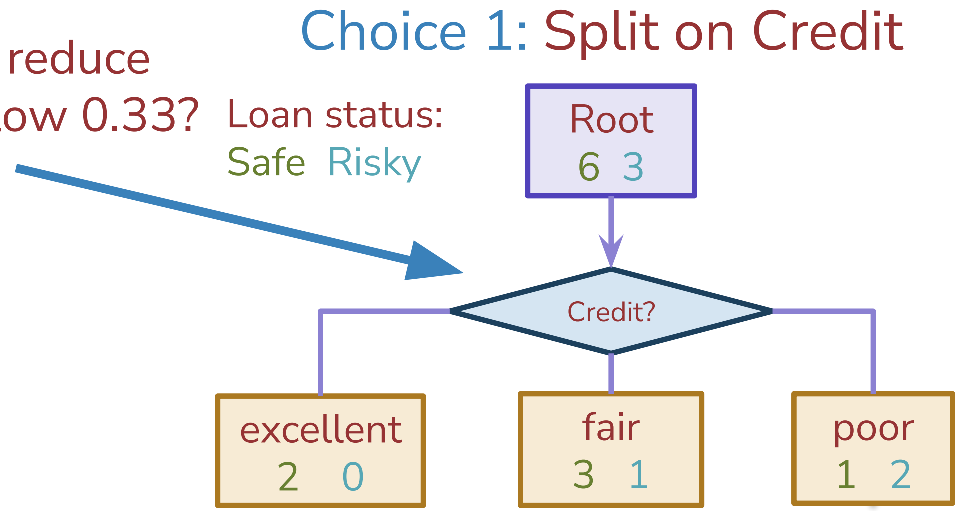
Error =

=

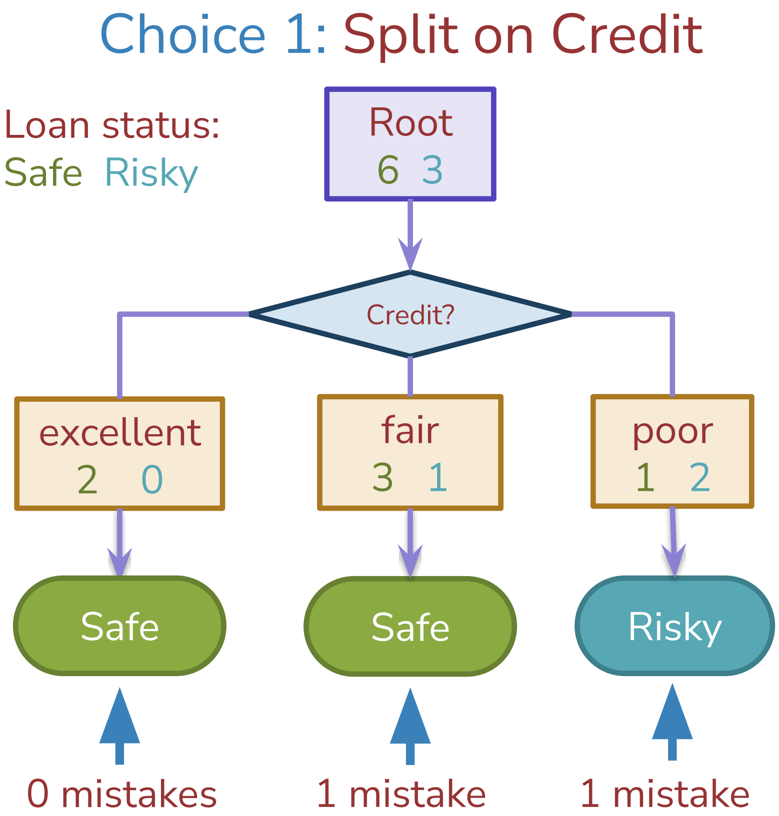
Tree	Classification error
(root)	0.33

Choice 1: Split on Credit history?

Does a split on Credit reduce classification error below 0.33?



Split on Credit: Classification error



Error = ______

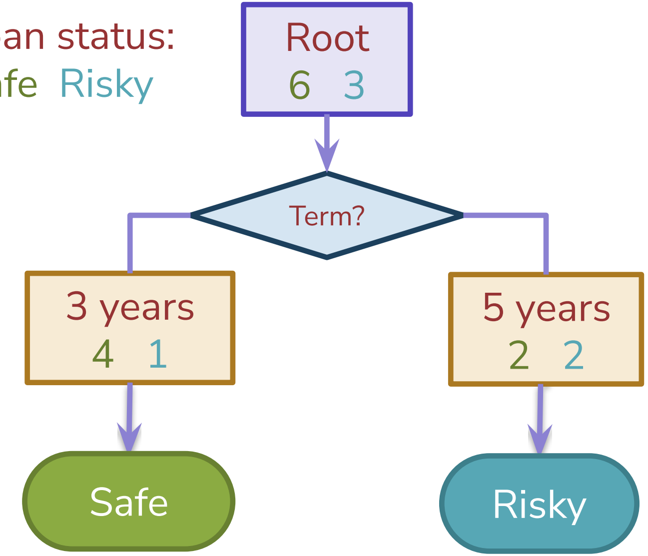
=

Tree	Classification error
(root)	0.33
Split on credit	0.22

Choice 2: Split on Term?

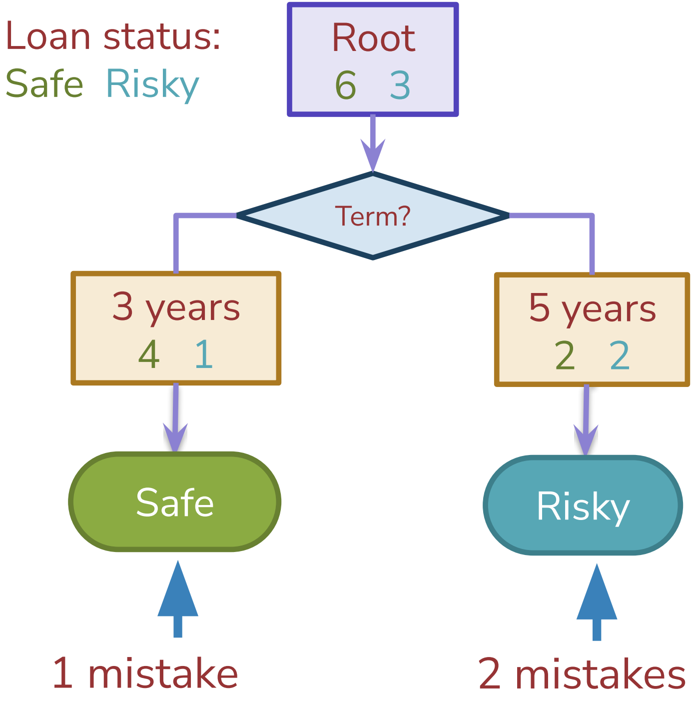
Choice 2: Split on Term

Loan status:
Safe Risky



Evaluating the split on Term

Choice 2: Split on Term



Error = ______

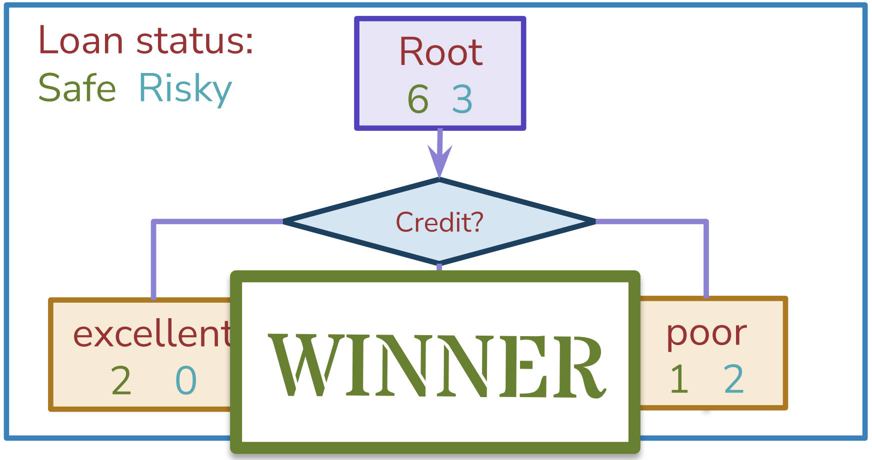
=

Tree	Classification error
(root)	0.33
Split on credit	0.22
Split on term	0.33

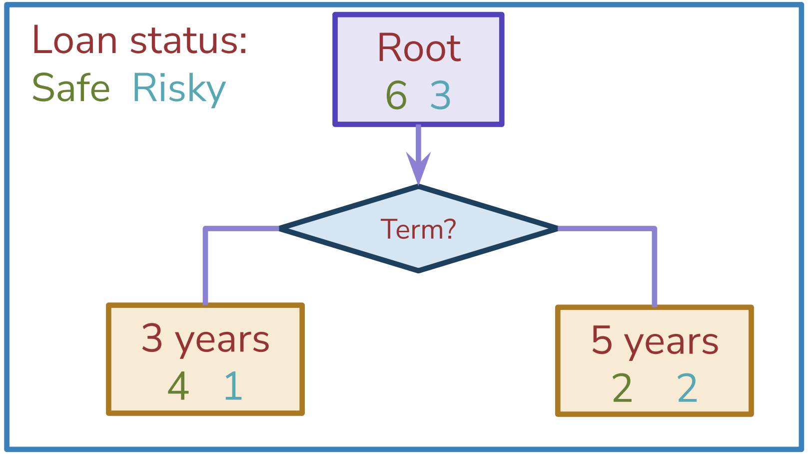
Choice 1 vs Choice 2:
Comparing split on credit vs term

Tree	Classification error
(root)	0.33
split on credit	0.22
split on loan term	0.33

Choice 1: Split on Credit



Choice 2: Split on Term



Split Selection

Split(node)

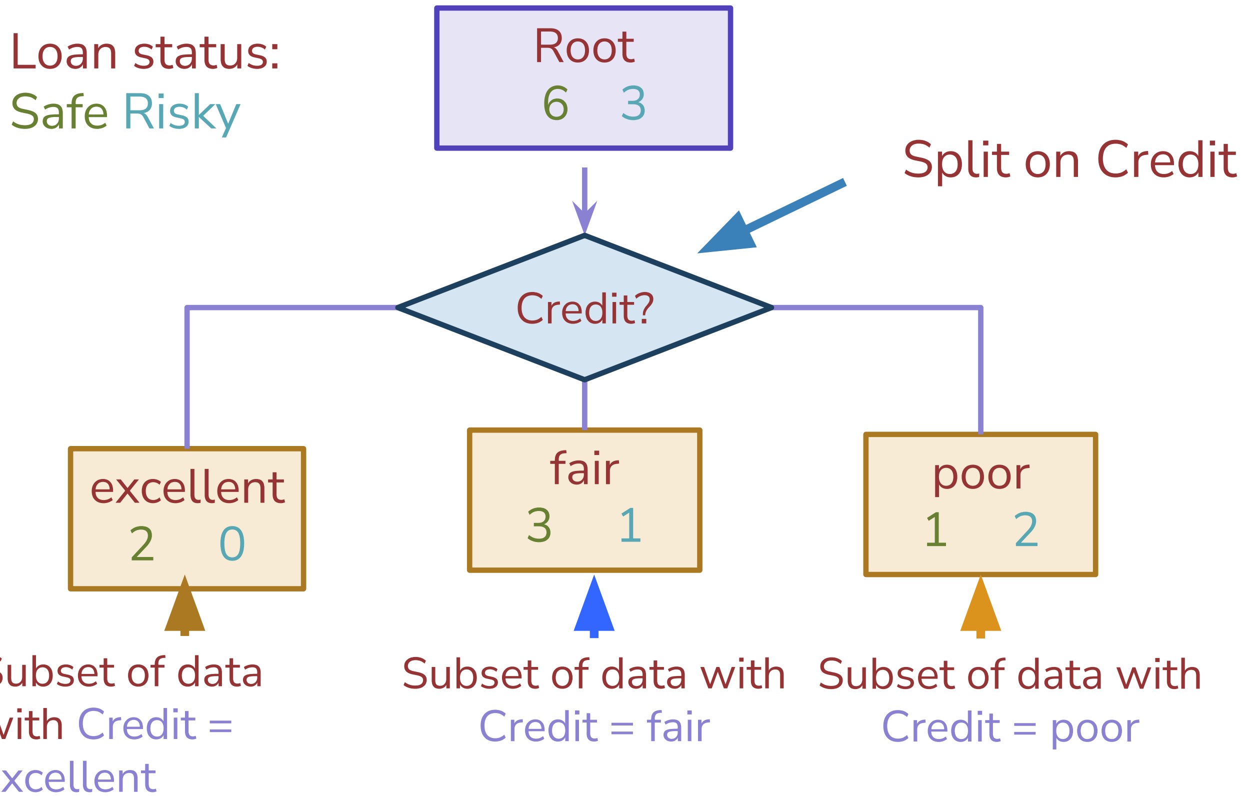
- Given M, the subset of training data at a node
- For each (remaining) feature $h_j(x)$:
    - Split data M on feature $h_j(x)$
    - Compute the classification error for the split
- Chose feature $h_j^*(x)$ with the lowest classification error

Greedy & Recursive Algorithm

BuildTree(node)

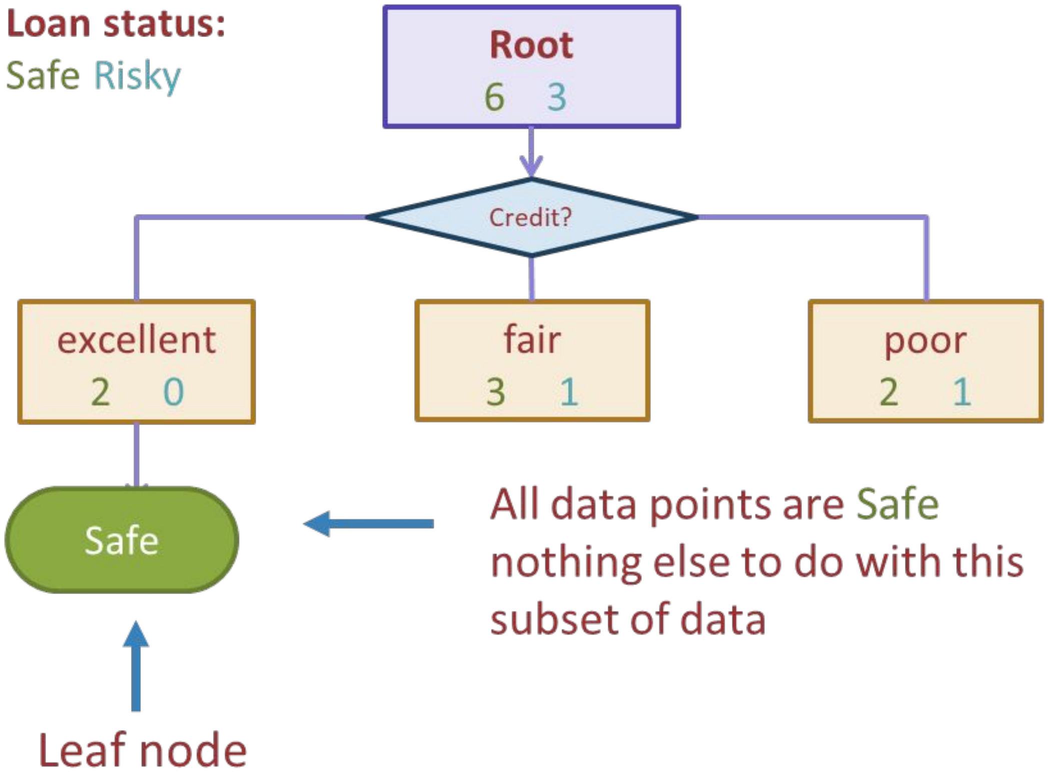
- If termination criterion is met:
    - Stop
- Else:
    - Split(node)
    - For child in node:
        - BuildTree(child)

Decision stump: 1 level

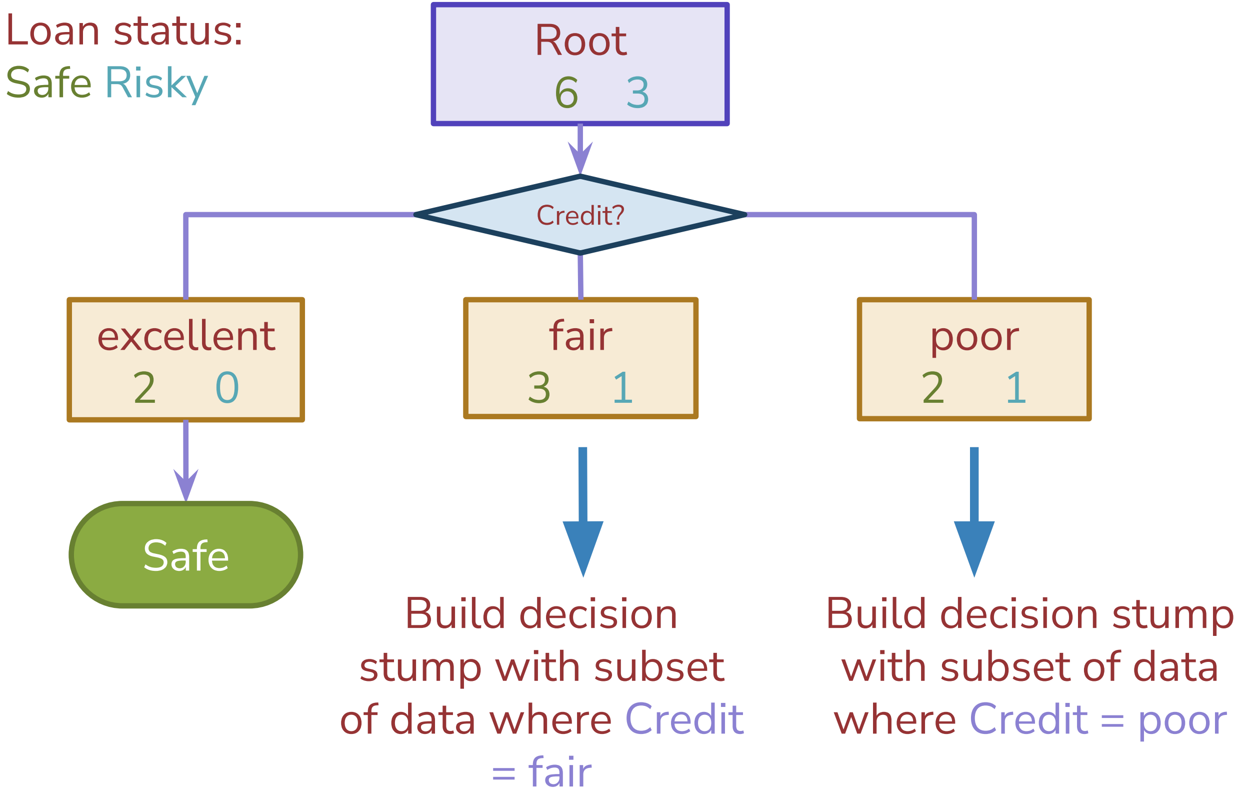


Stopping

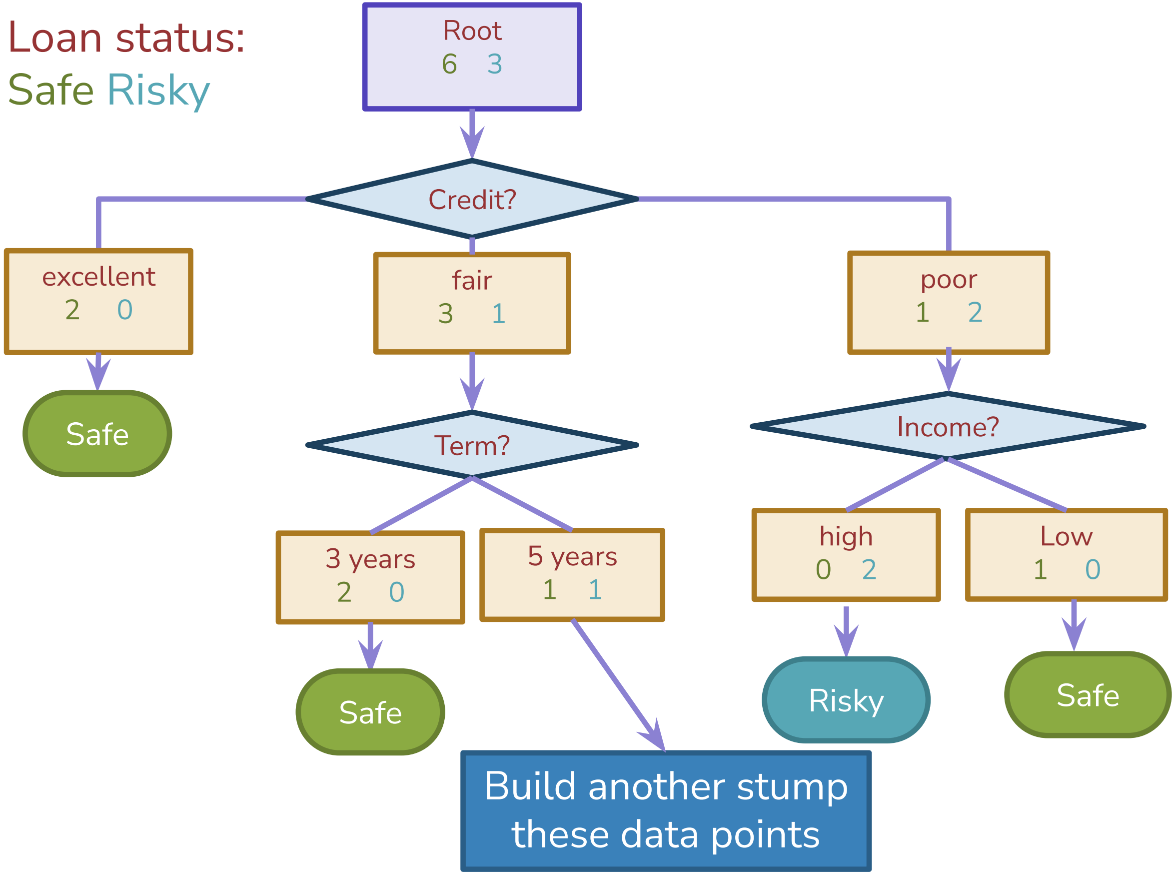
For now: Stop when all points are in one class



Tree learning = Recursive stump learning



Second level

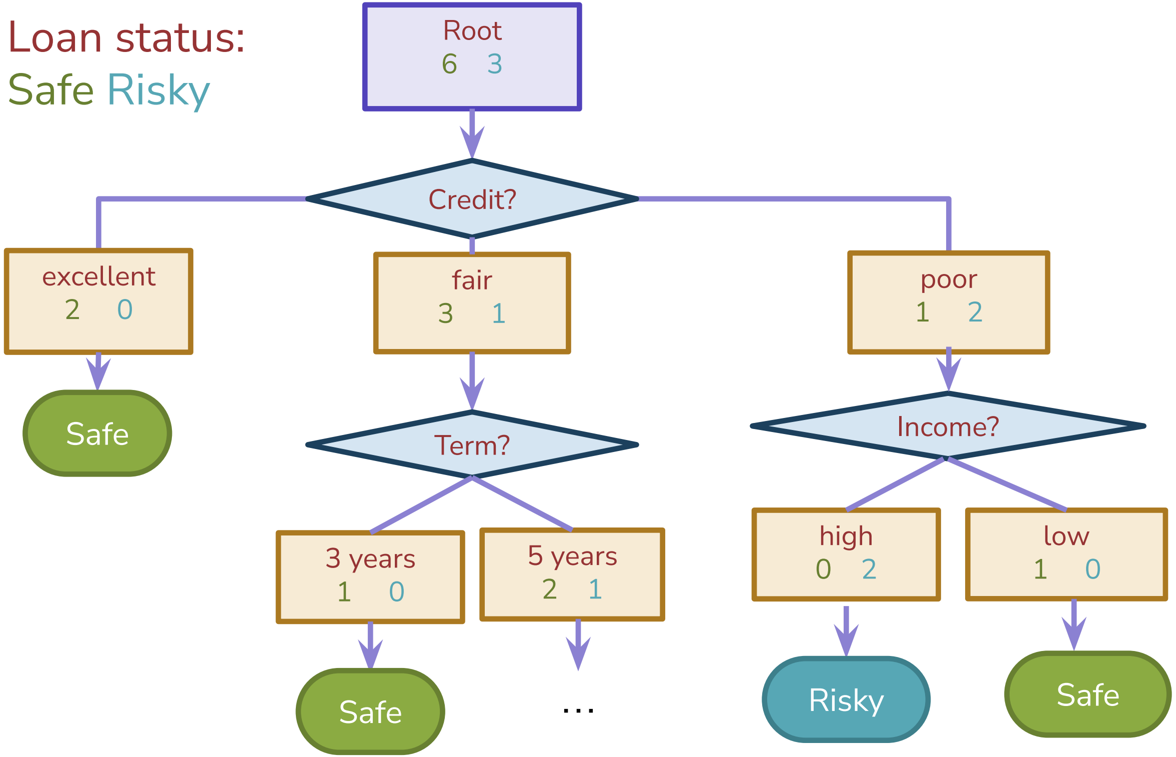


slido

Think

1 min

What predictions **should** the below decision tree output for the following datapoints?



Credit	Term	Income
excellent	5 yrs	high
fair	3 yrs	low
poor	5 yrs	(missing)

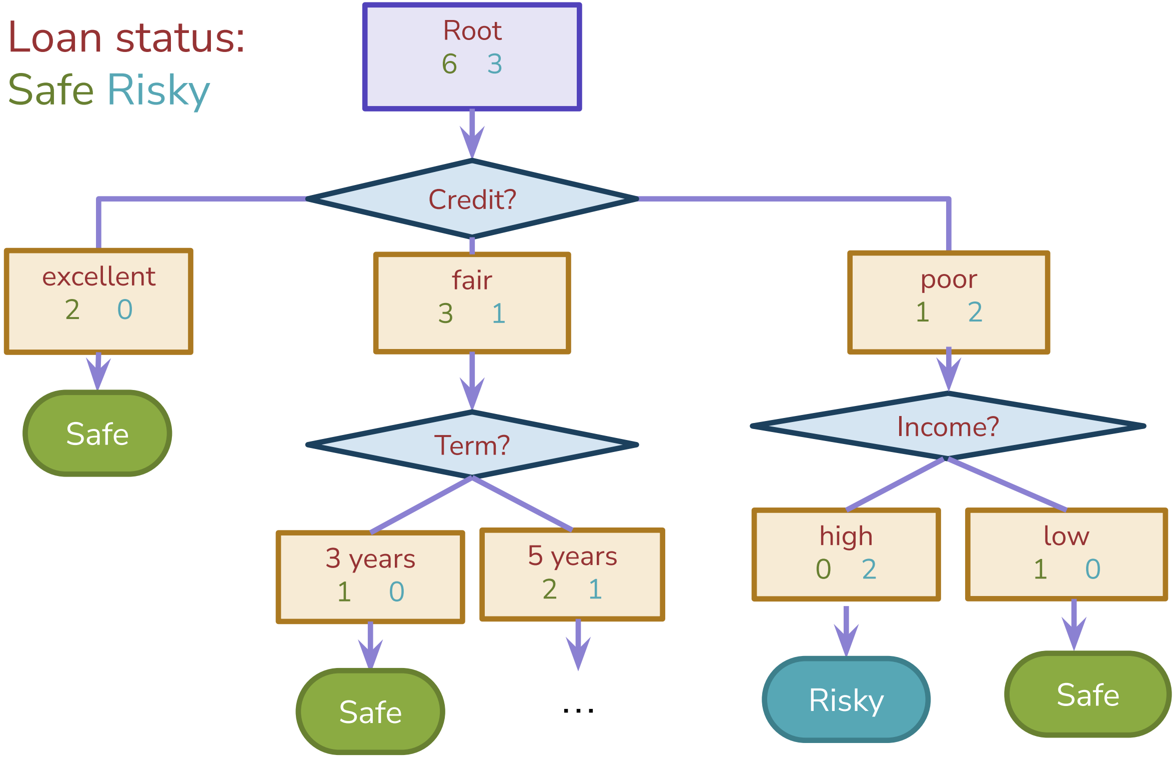
slido

Group

2 min

Credit	Term	Income
excellent	5 yrs	high
fair	3 yrs	low
poor	5 yrs	(missing)

- What predictions **should** the below decision tree output for the following datapoints?

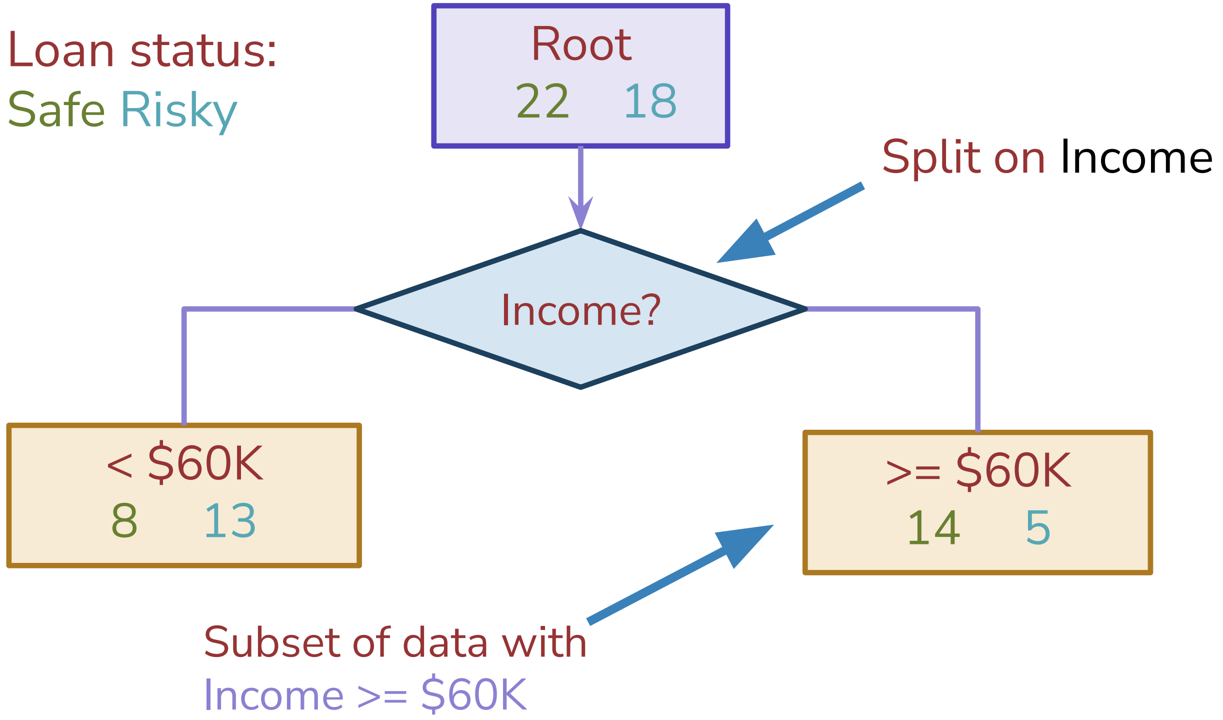


Brain Break

Real valued features

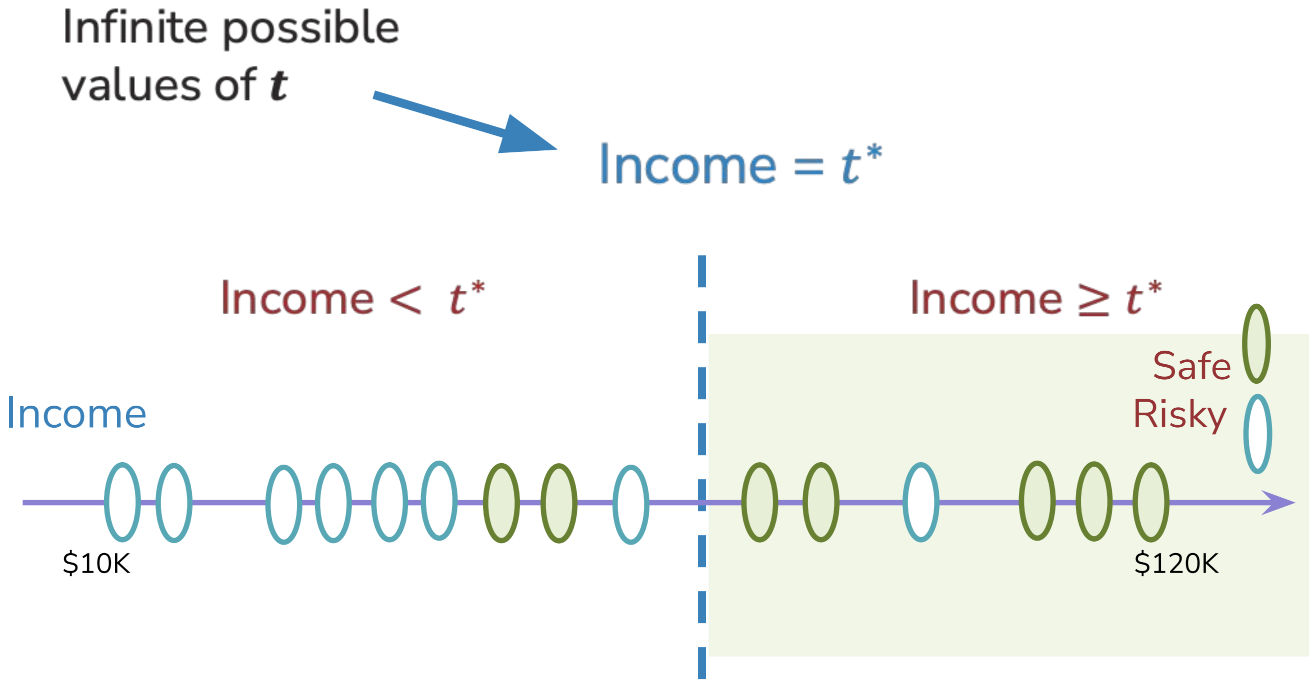
Income	Credit	Term	y
$105 K	excellent	3 yrs	Safe
$112 K	good	5 yrs	Risky
$73 K	fair	3 yrs	Safe
$69 K	excellent	5 yrs	Safe
$217 K	excellent	3 yrs	Risky
$120 K	good	5 yrs	Safe
$64 K	fair	3 yrs	Risky
$340 K	excellent	5 yrs	Safe
$60 K	good	3 yrs	Risky

Threshold split



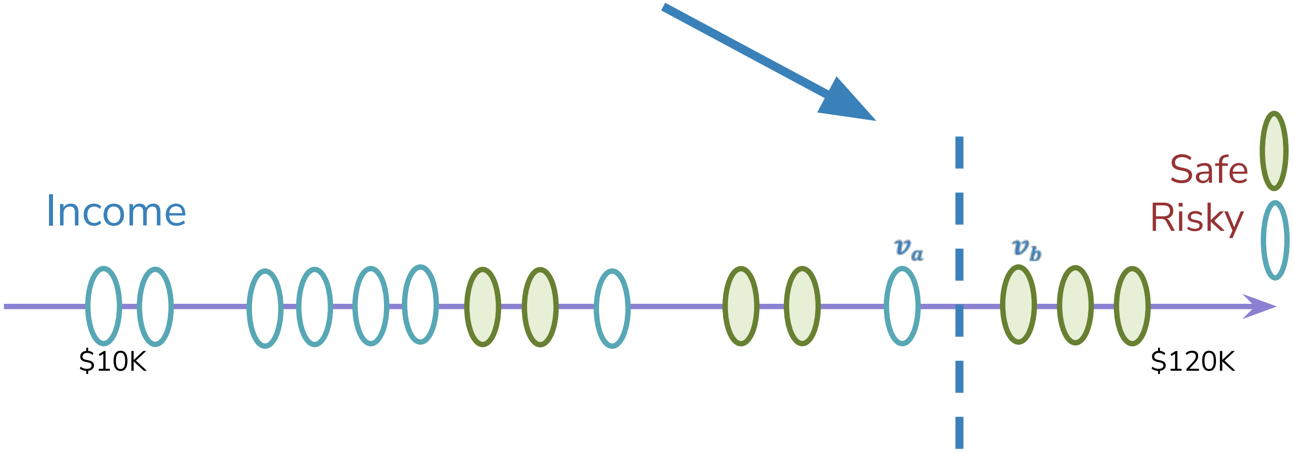
Best threshold?

Similar to our simple, threshold model when discussing Fairness!



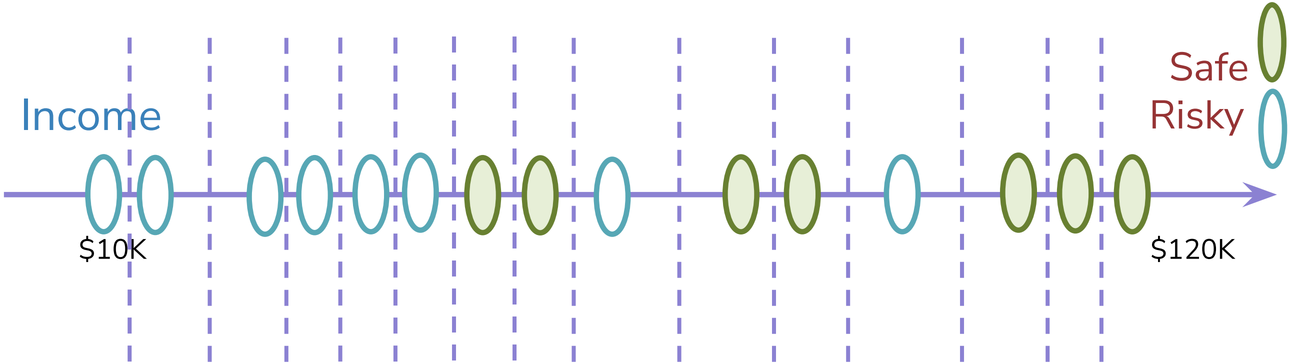
Threshold between points

Same classification error for any threshold split between v_a and v_b



Only need to consider mid-points

Finite number of splits to consider



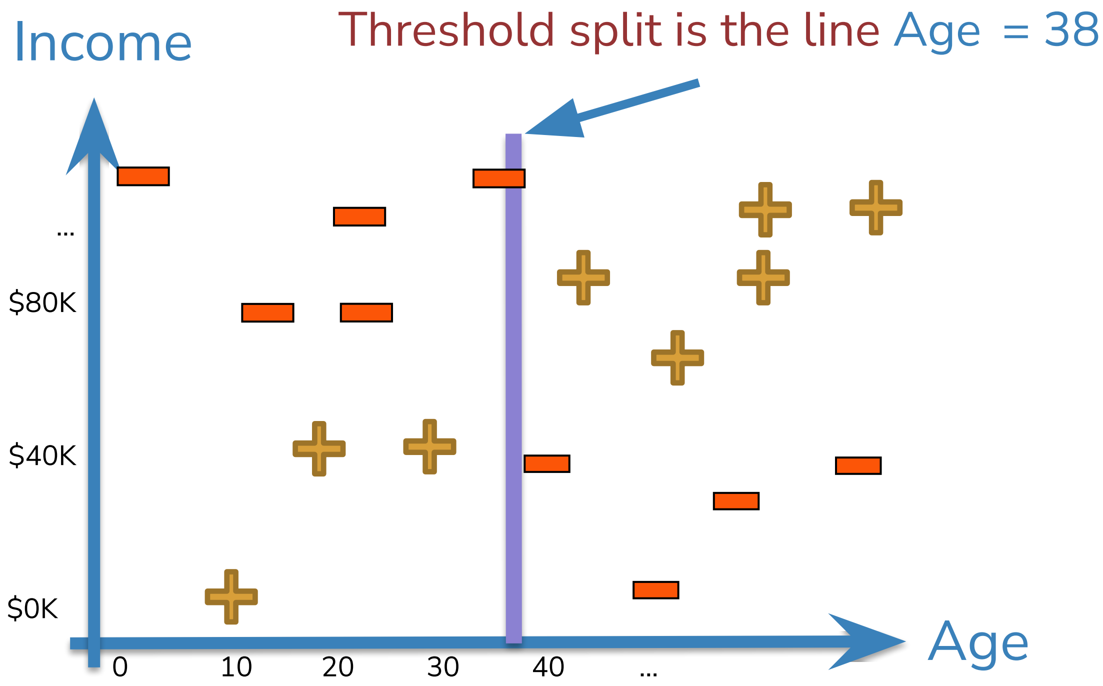
Threshold split selection algorithm

- Step 1: Sort the values of a feature $h_j(x)$:

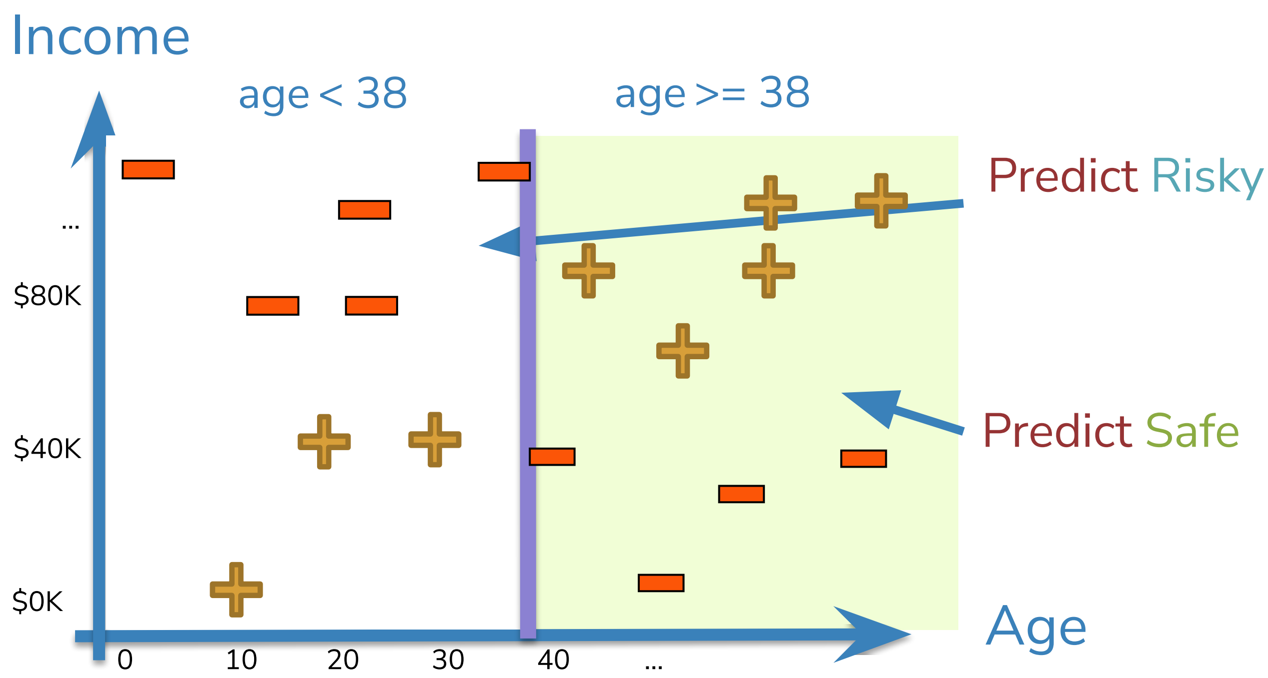
  Let $[v_1, v_2, \ldots, v_N]$ denote sorted values

- Step 2:
  - For $i = [1, \ldots, N \; - 1]$
    - Consider split $t_i = \frac{v_i + v_{i+1}}{2}$
    - Compute classification error for threshold split $h_j(x) \geq t_i$
  - Chose the $\boldsymbol{t}^*$ with the lowest class. error

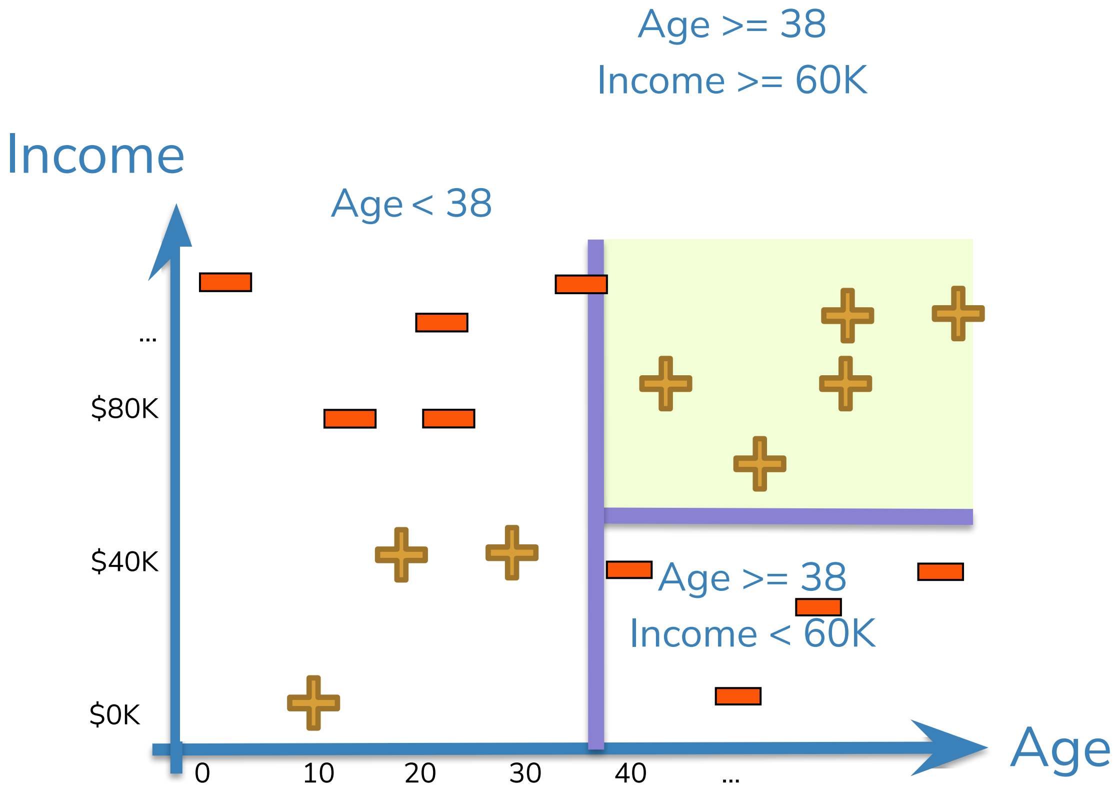
Visualizing the threshold split



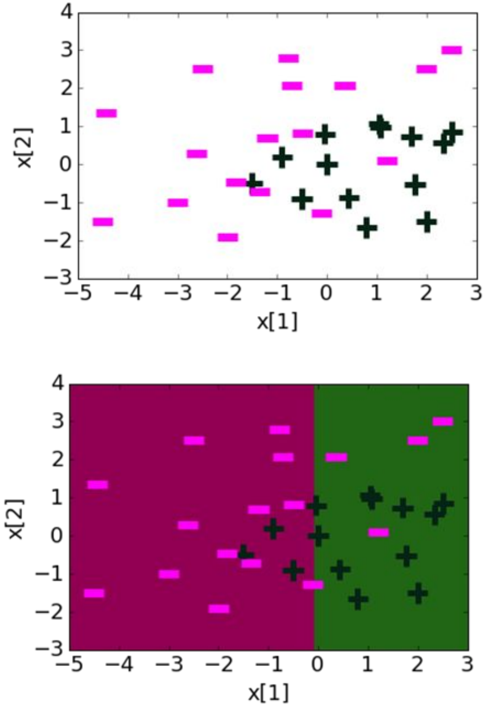
Split on Age >= 38

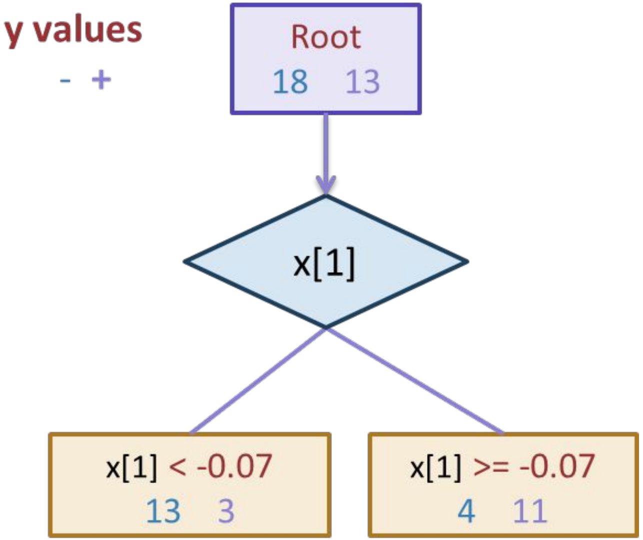


Each split partitions the 2-D space

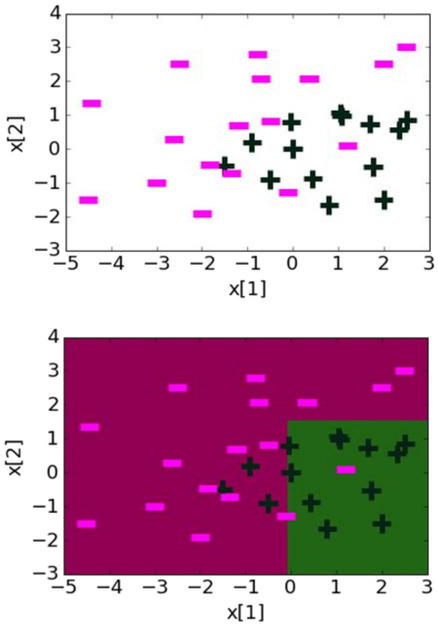


Depth 1: Split on x[1]

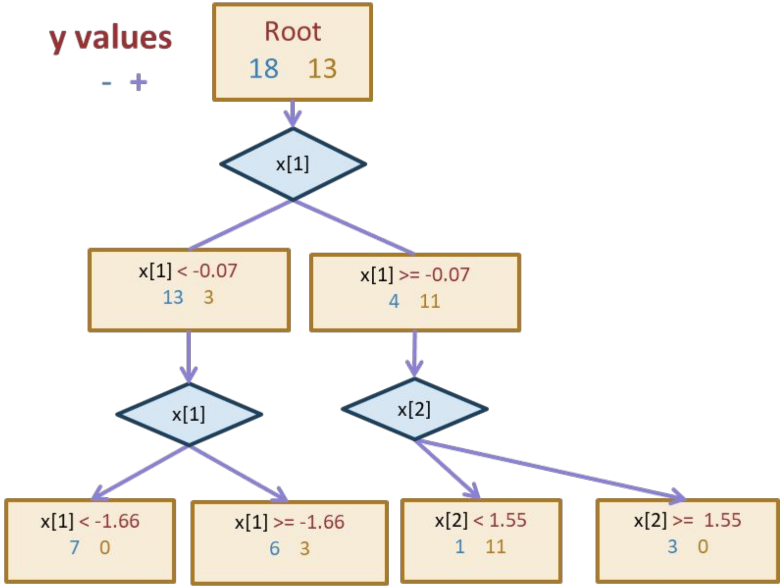




Depth 2



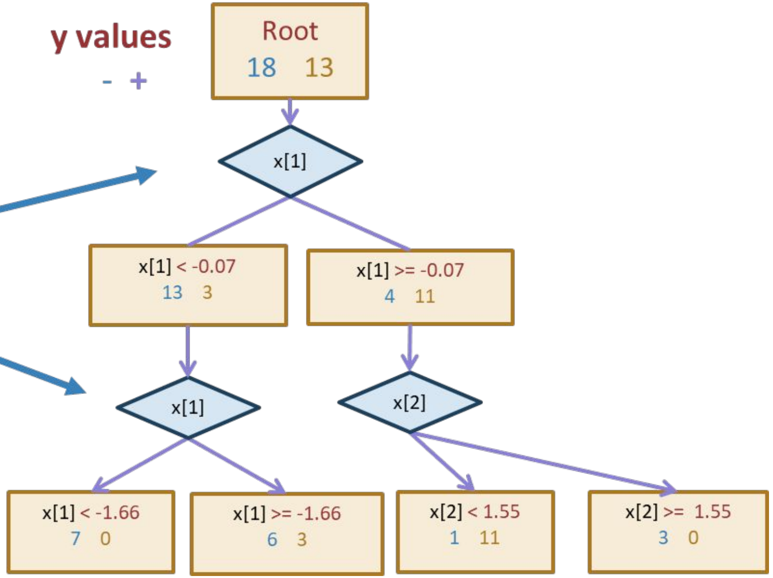
x[2]
x[1]
x[2]
x[1]



y values
- +
Root
18 13
x[1]
x[1] < -0.07
13 3
x[1] >= -0.07
4 11
x[1]
x[2]
x[1] < -1.66
7 0
x[1] >= -1.66
6 3
x[2] < 1.55
1 11
x[2] >= 1.55
3 0

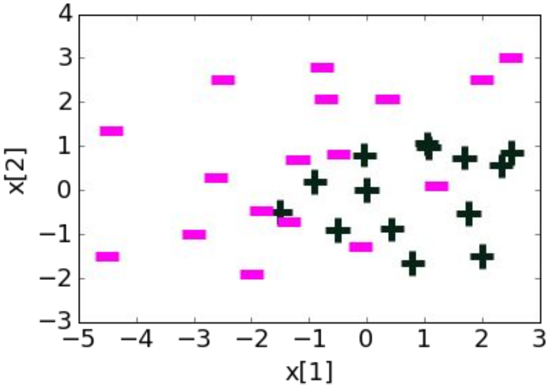
Threshold split caveat

For threshold splits, same feature can be used multiple times

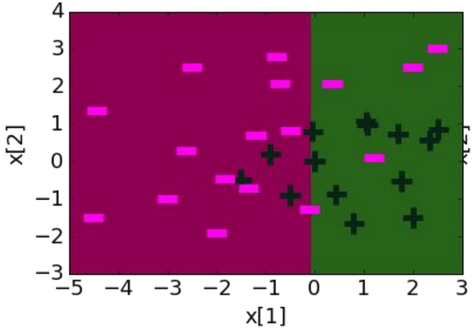


Decision boundaries

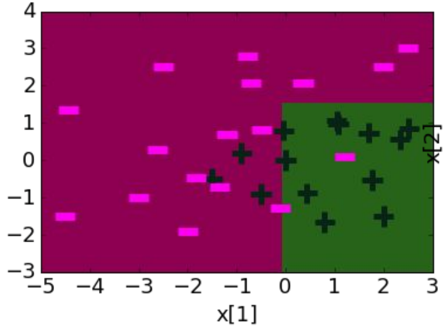
- Decision boundaries can be complex!



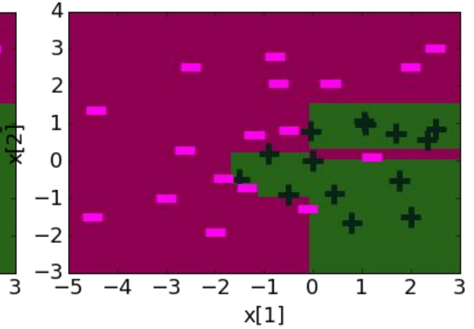
Depth 1



Depth 2



Depth 1

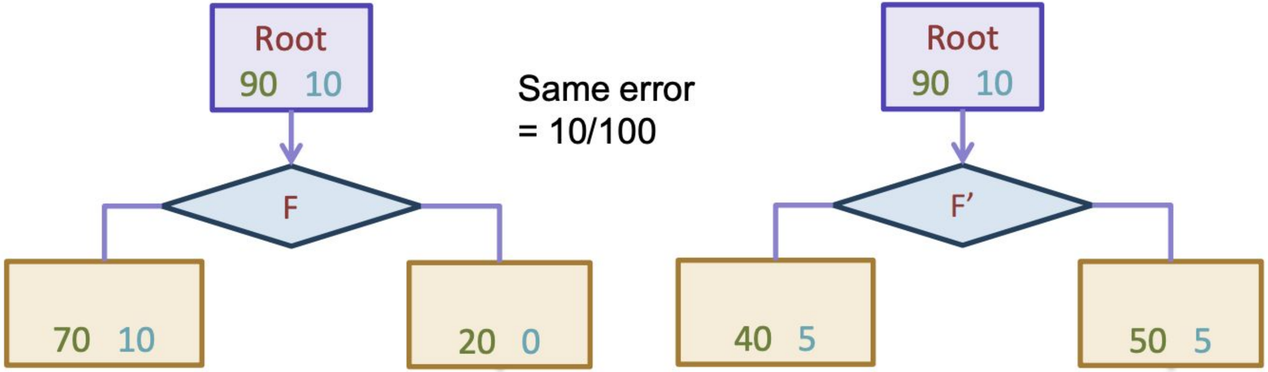


Overfitting

- Deep decision trees are prone to overfitting
  - Decision boundaries are interpretable but not stable
  - Small change in the dataset leads to big difference in the outcome
- Overcoming Overfitting:
  - Stop when tree reaches certain height (e.g., 4 levels)
  - Stop when leaf has ≤ some num of points (e.g., 20 pts)
    - Will be the stopping condition for HW
  - Stop if split won't significantly decrease error by more than some amount (e.g., 10%)
- Other methods include growing full tree and pruning back
- Fine-tune hyperparameters with validation set or CV

In Practice

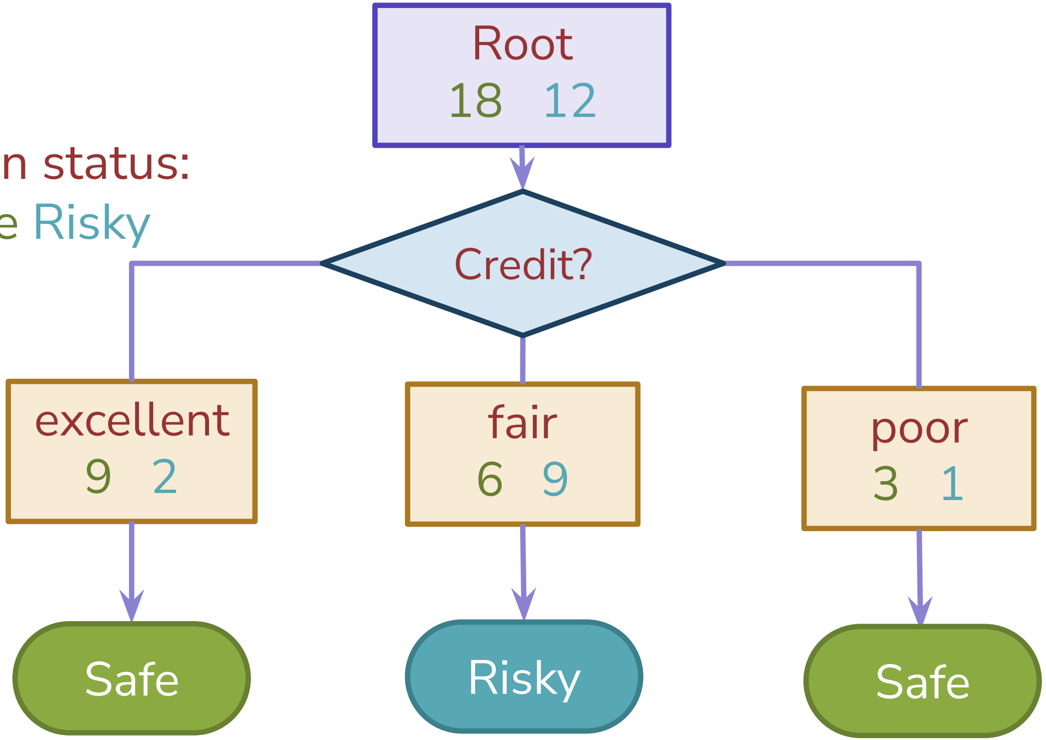
- Trees can be used for classification or regression (CART)
  - Classification: Predict majority class for root node
  - Regression: Predict average label for root node
- In practice, we don't minimize classification error but instead some more complex metric to measure quality of split such as **Gini Impurity** or **Information Gain** (not covered in 416)



- Can also be used to predict probabilities

Predicting probabilities

Loan status:
Safe Risky



$$P(y = \text{Safe} \mid x) = \frac{3}{3 + 1} = 0.75$$

Recap

What you can do now:

- Define the assumptions and modeling for Naïve Bayes
- Define a decision tree classifier
- Interpret the output of a decision trees
- Learn a decision tree classifier using greedy algorithm
- Traverse a decision tree to make predictions
  - Majority class predictions