CSE/STAT 416

Naïve Bayes and Decision Trees

Hunter Schafer
Paul G. Allen School of Computer Science & Engineering
University of Washington

April 26, 2021

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

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

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

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

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

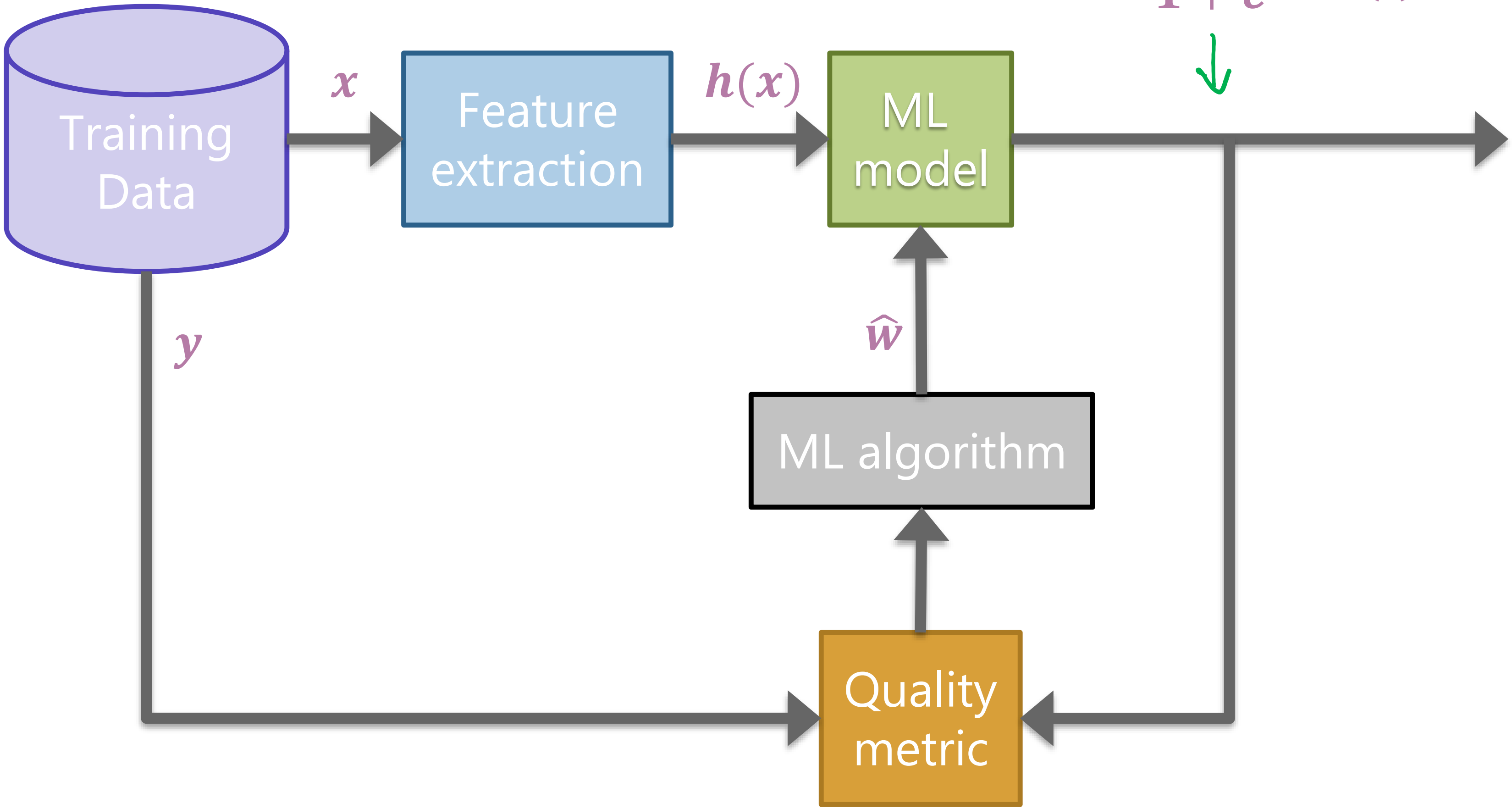
Very sure
$\hat{y}_i = +1$

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

0 | 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

$x =$ *"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!

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes Rule:

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

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

Example

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

Since we're just trying to find out which class has the greater probability, we can discard the divisor.

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.

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

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

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

Compute Probabilities

How do we compute something like

$$P(y = +1)? = \frac{\text{\# pos reviews}}{\text{\# reviews}}$$

How do we compute something like

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

$$= \frac{\text{\# of times see “awesome” in pos reviews}}{\text{\# words in all positive reviews}}$$

Zeros

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

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

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)

$\frac{0}{100} \to \frac{1}{100}$

Compare Models

Logistic Regression:

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

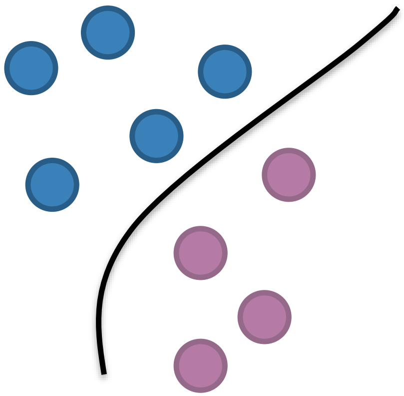
Naïve Bayes:

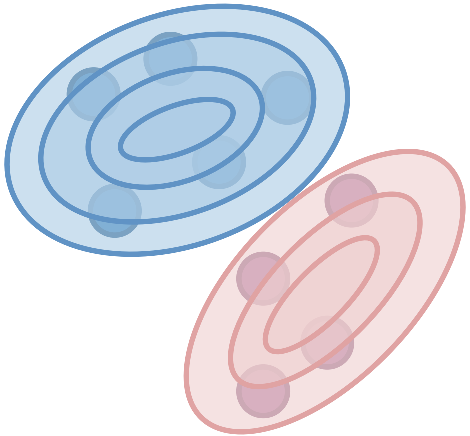
$$P(y|\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{d}) = \prod_{j=1}^{d} P(x_j|y)\ P(y)$$

Compare Models

Generative: defines a model for generating x (e.g. Naïve Bayes) $P(x|y)P(y)$

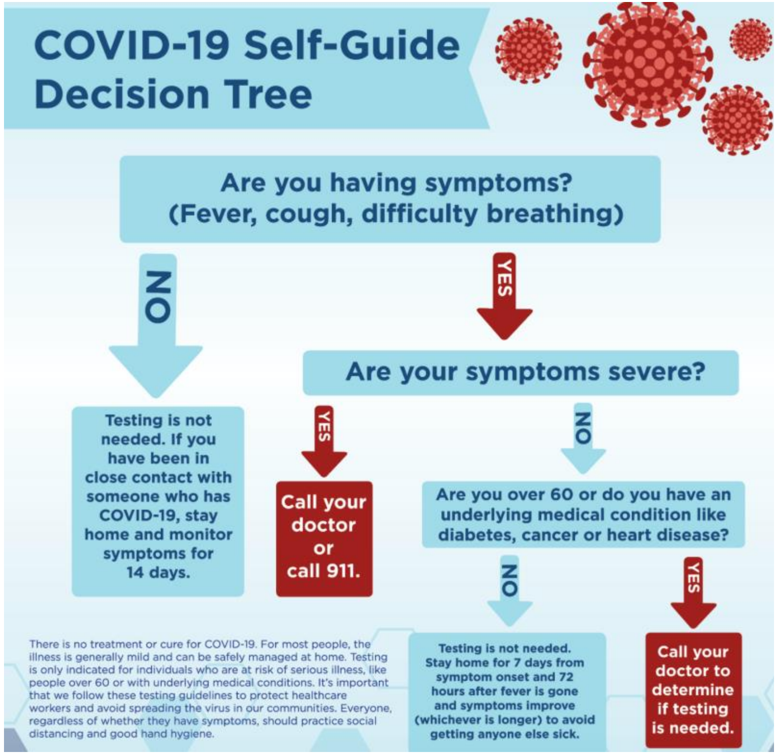
Discriminative: only cares about defining and optimizing a decision boundary (e.g. Logistic Regression) $P(y|x) \to w$





Decision Trees

How do we make decisions?

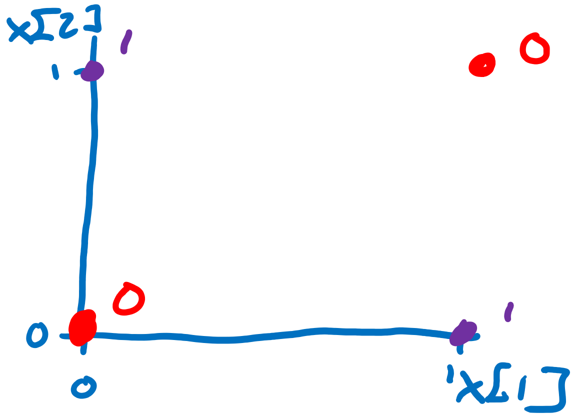


https://www.holzer.org/coronavirus-covid-19-updates/

XOR
(Exclusive Or)

- A line might not always support our decisions.

x[1]	x[2]	y
0	0	0
1	0	1
0	1	1
1	1	0



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
★★★

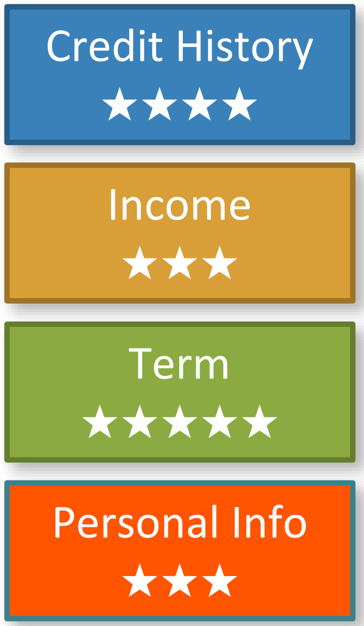
Term
★★★★★

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

Credit History
★★★★

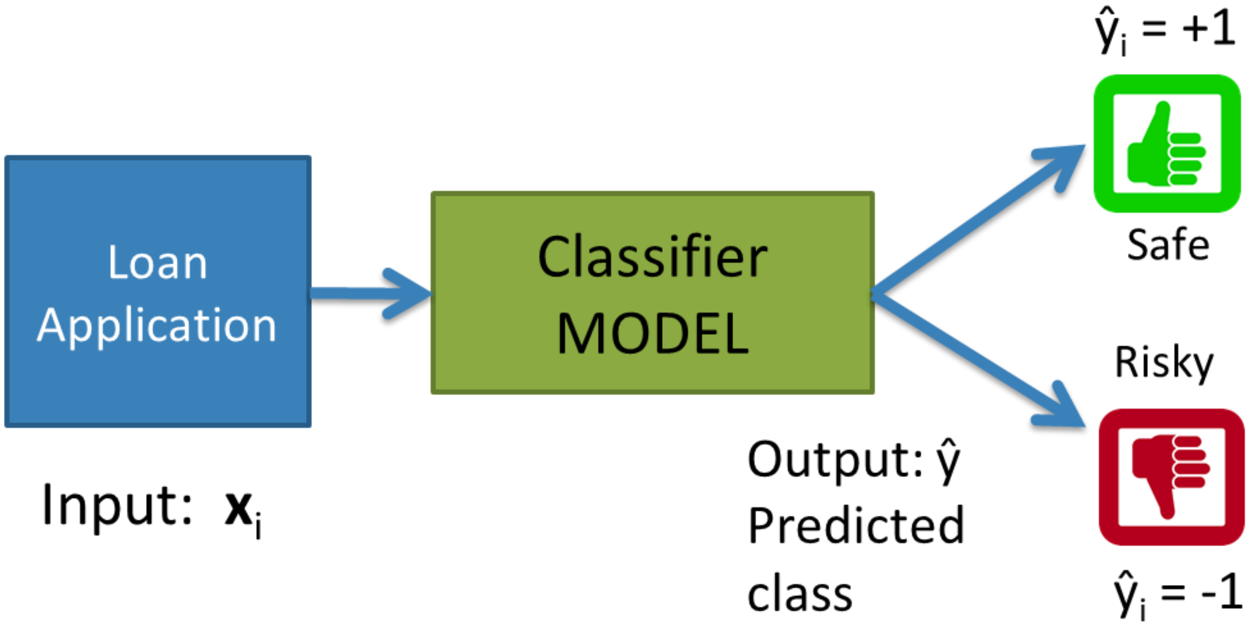
Income
★★★

Term
★★★★★

Personal Info
★★★

Intelligent application

Classifier review



Setup

N= 9 examples   D=3

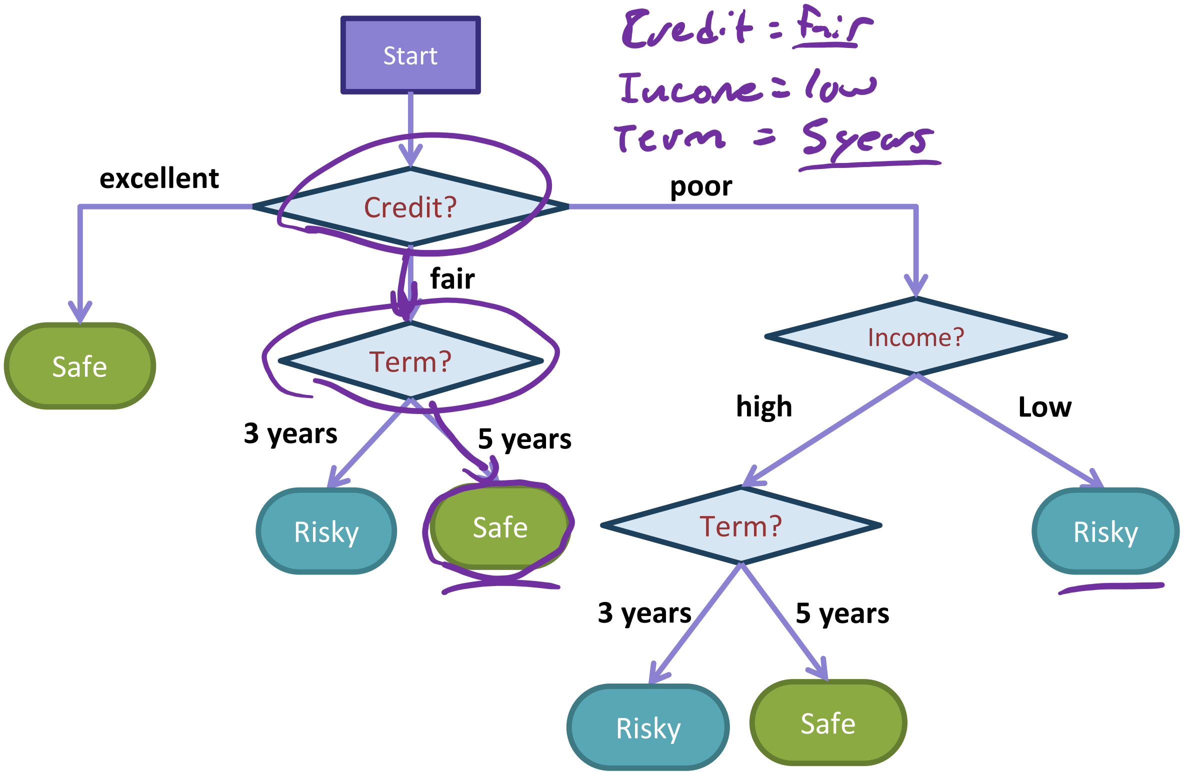
Data (N observations, 3 features)

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!

Decision Trees



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

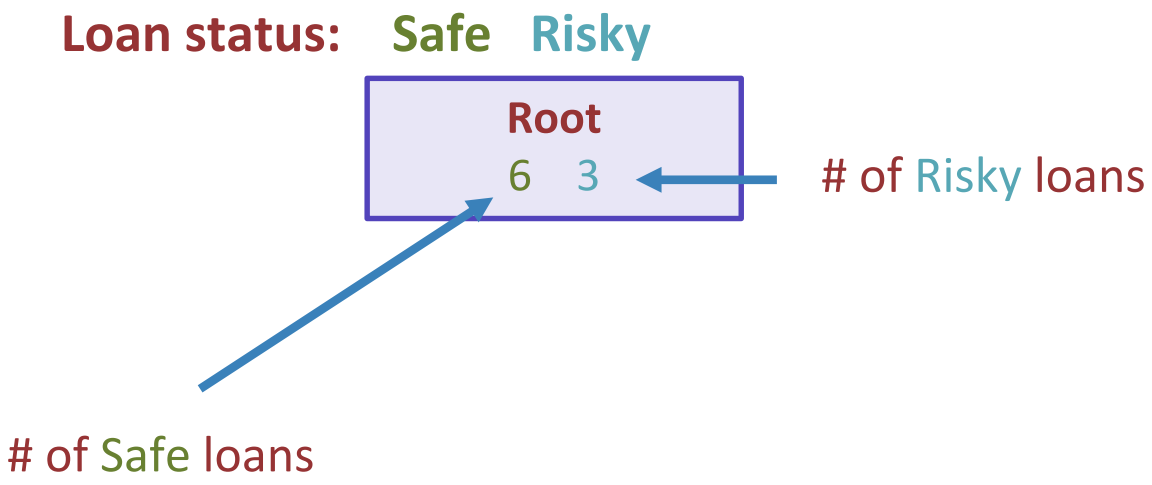
Growing Trees

- Grow the trees using a greedy approach
- What do we need?

↳ Which features are "good"

↳ When to stop growing tree

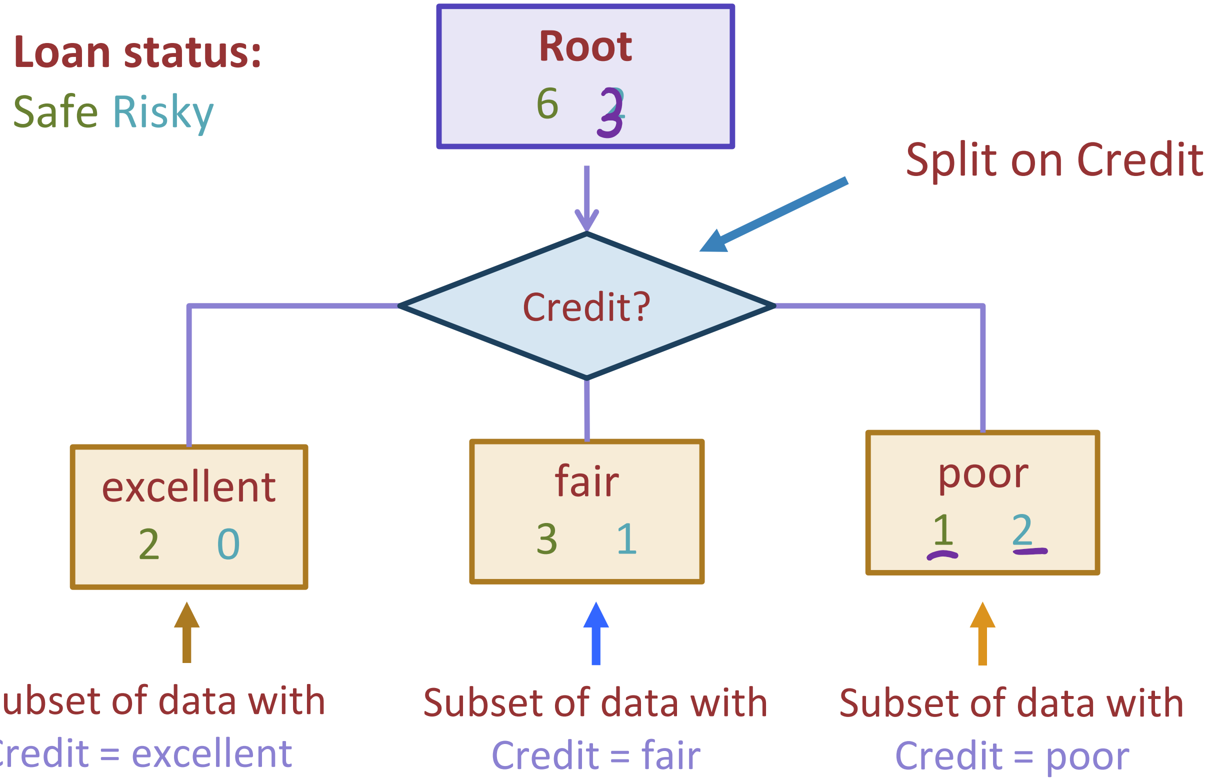
Visual Notation



N = 9 examples

Decision stump: 1 level

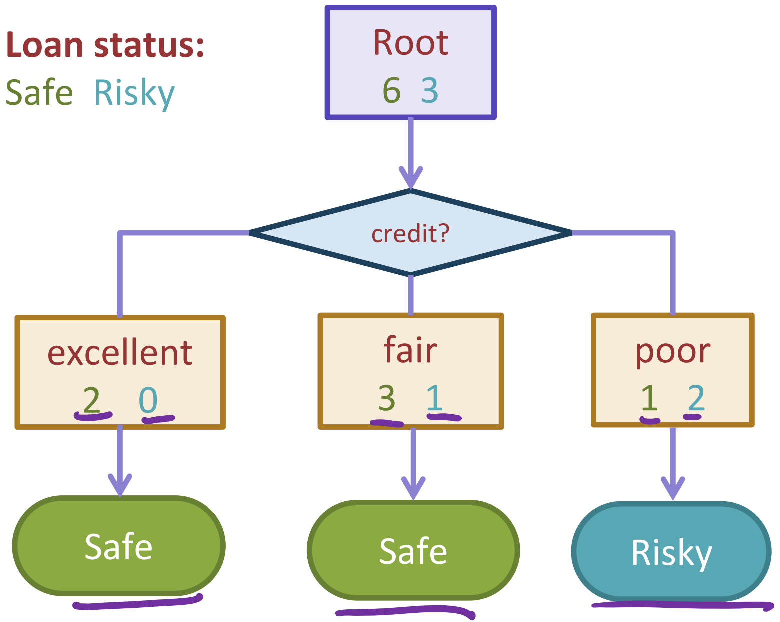
Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	safe
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe



Making predictions

Decision Stump = Tree w/ 1 internal node

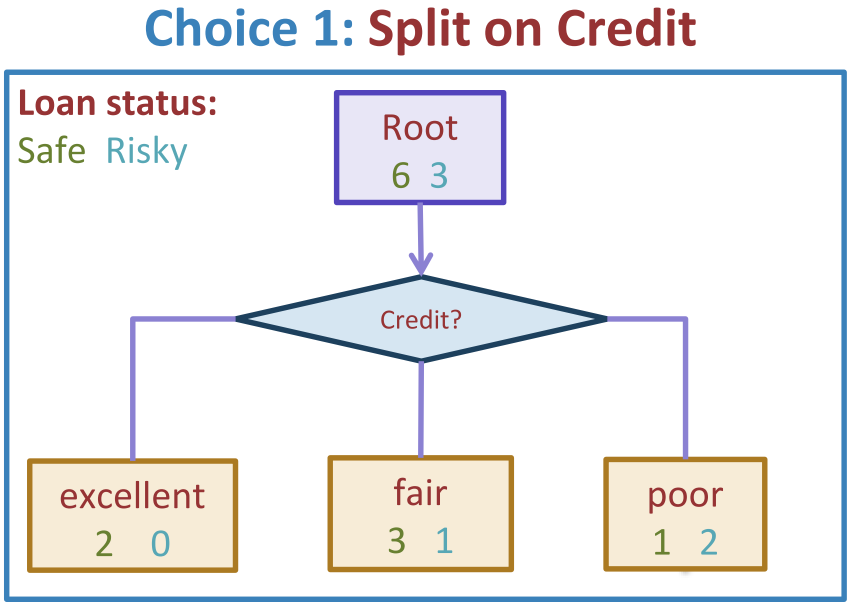
For each leaf node, set $\hat{y}$ = majority value

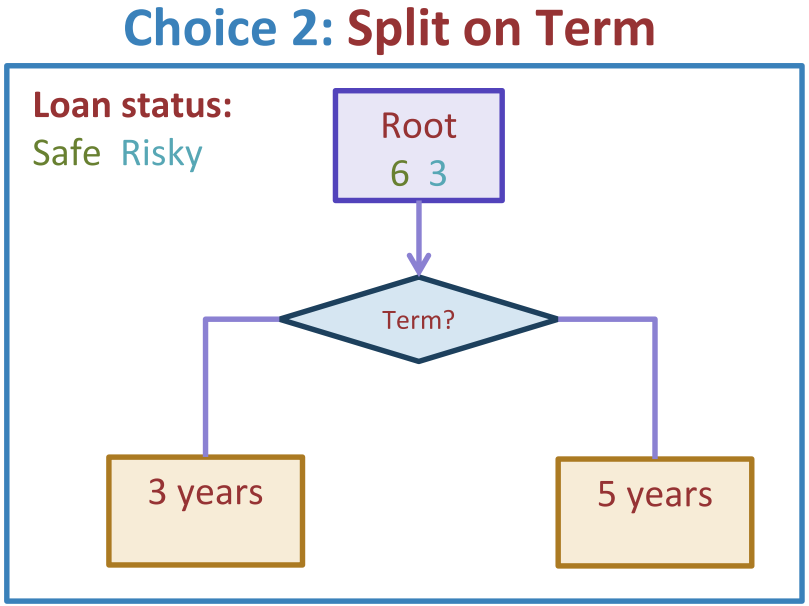


How do we select the best feature?

.

* Select the split with lowest classification error

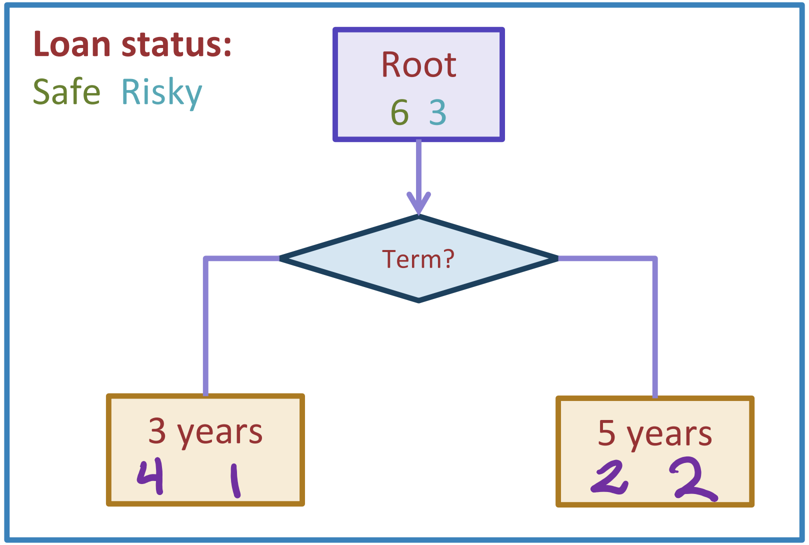




Calculate the node values.

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

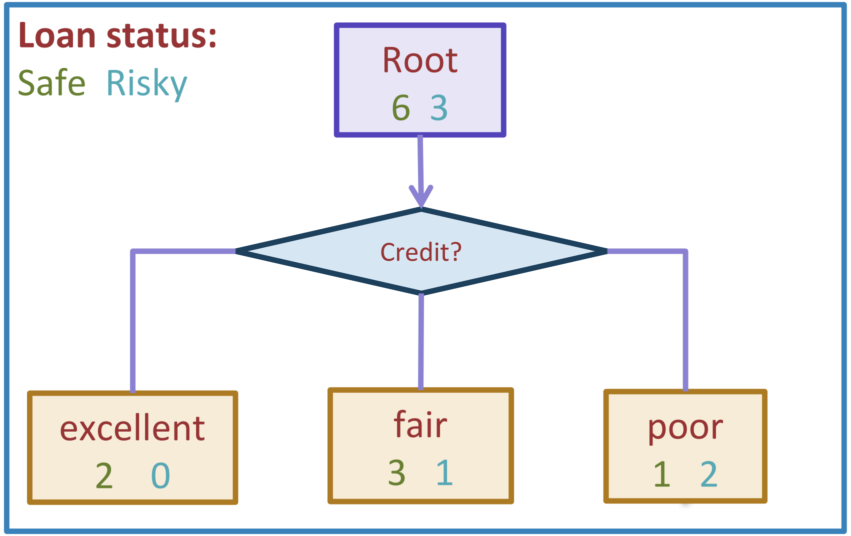
Choice 2: Split on Term



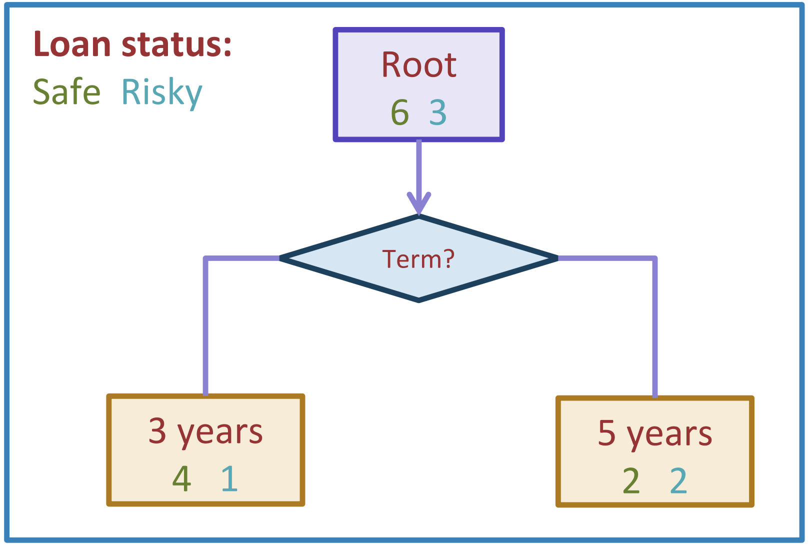
How do we select the best feature?

Select the split with lowest classification error

Choice 1: Split on Credit

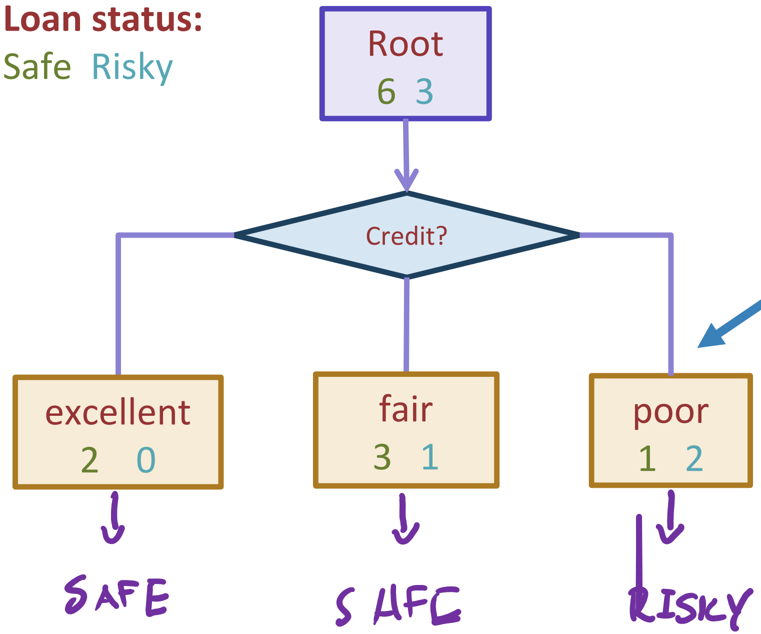


Choice 2: Split on Term



How do we measure effectiveness of a split?

Loan status:
Safe Risky



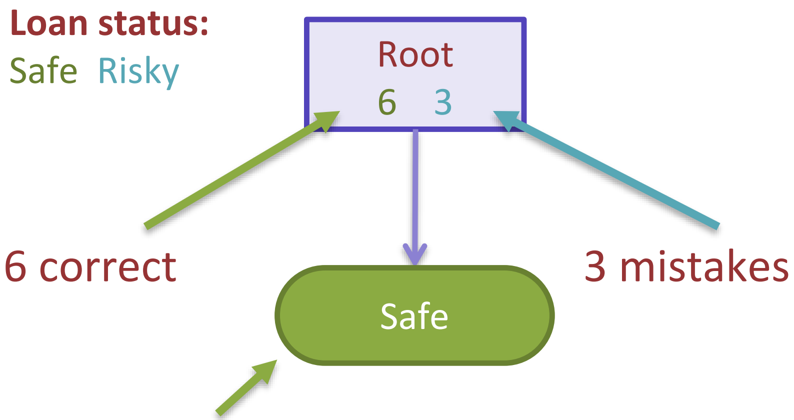
Idea: Calculate classification error of this decision stump

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

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**

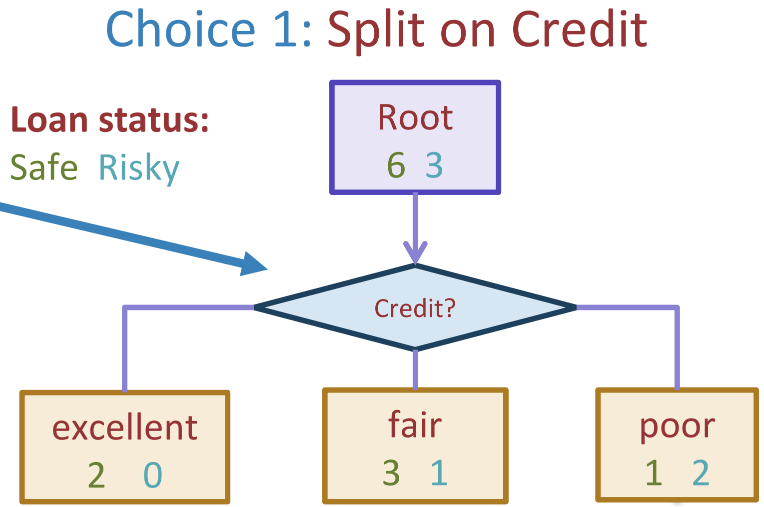
$$\text{Error} = \frac{3}{9}$$

$$= 0.33$$

Tree	Classification error
(root)	0.33

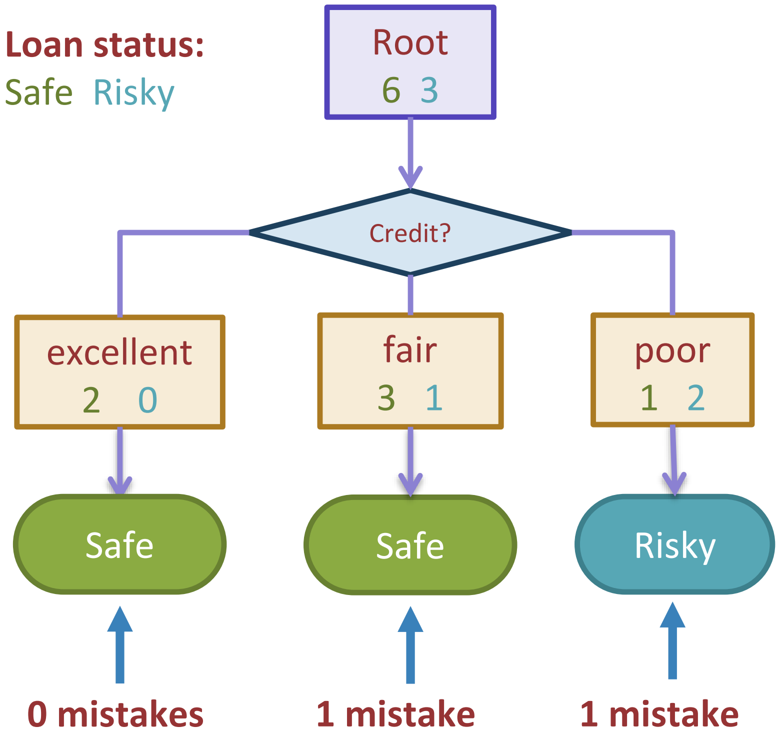
Choice 1: Split on Credit history?

Does a split on Credit reduce classification error below 0.33?



Split on Credit: Classification error

Choice 1: Split on Credit



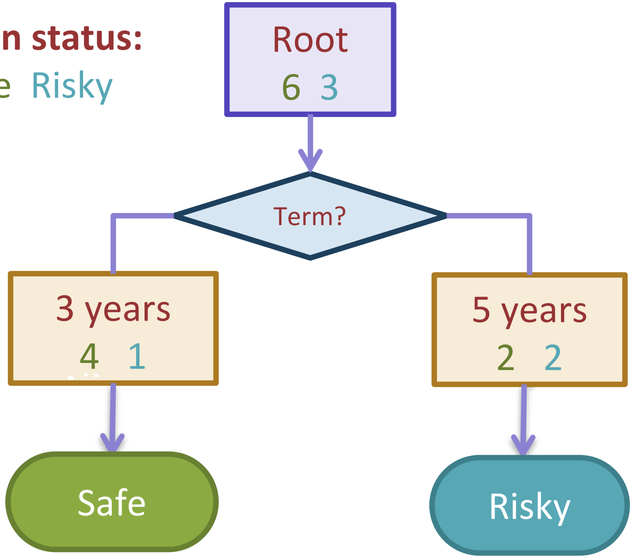
$$\text{Error} = \frac{0+1+1}{9} = \frac{2}{9}$$
$$= 0.22$$

Tree	Classification error
(root)	0.33
Split on credit	0.22

Choice 2: Split on Term?

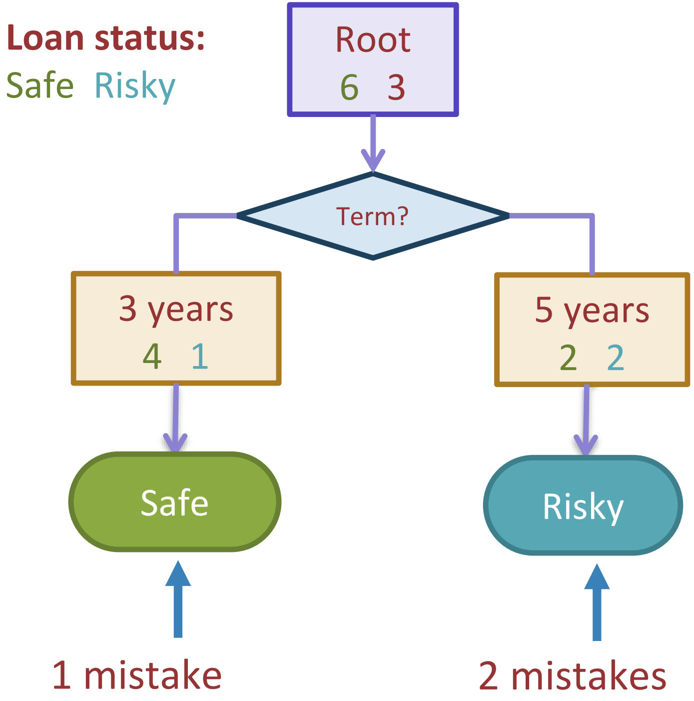
Choice 2: Split on Term

Loan status:
Safe Risky



Evaluating the split on Term

Choice 2: Split on Term

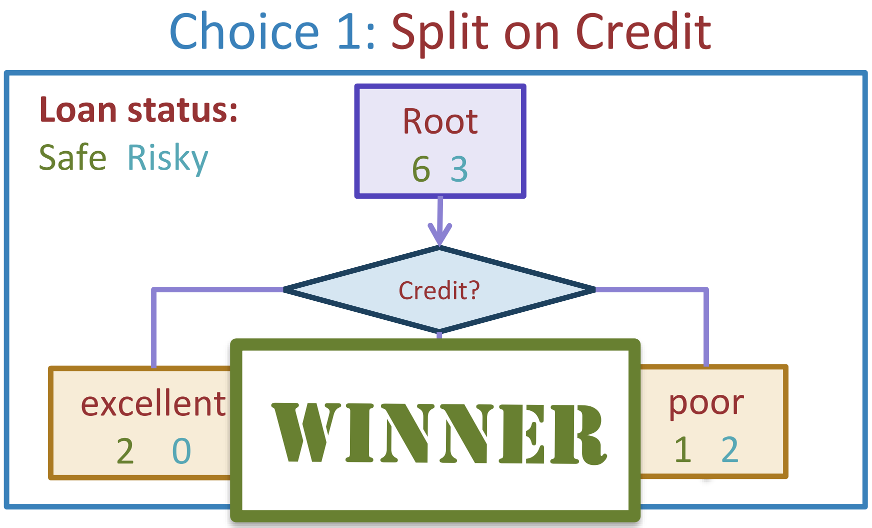


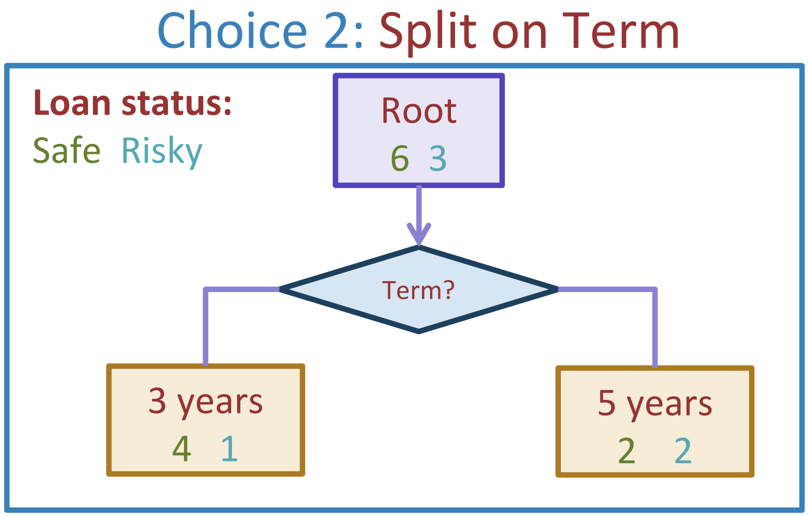
$$\text{Error} = \frac{1 + 2}{9} = \frac{3}{9}$$
$$= 0.33$$

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





3:28

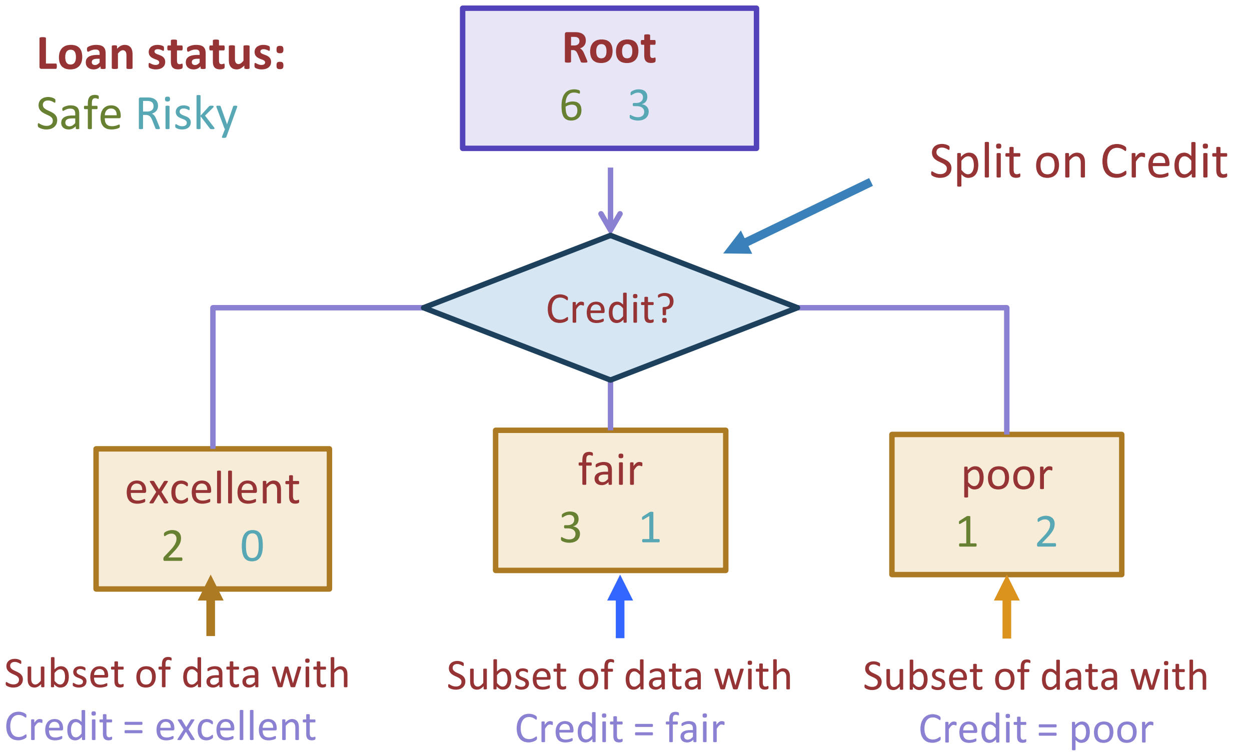
Split Selection Summary

- Given a subset of data M (a node in a tree)
- For each remaining feature $h_i(x)$:
  1. Split data of M according to feature $h_i(x)$
  2. Compute classification error of split
- Chose feature $h^*(x)$ with lowest classification error

Greedy Algorithm

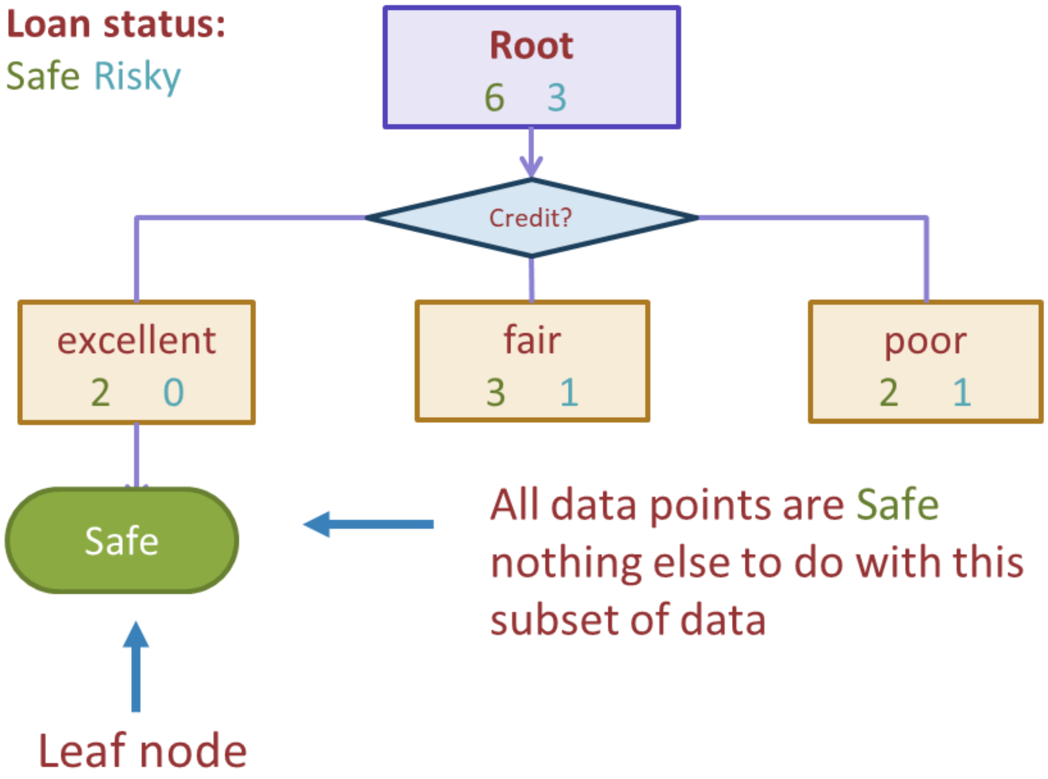
- If classification for data at current node is perfect (classification error = 0):
  - Stop
- Else:
  - repeat split selection with next stump

Decision stump: 1 level

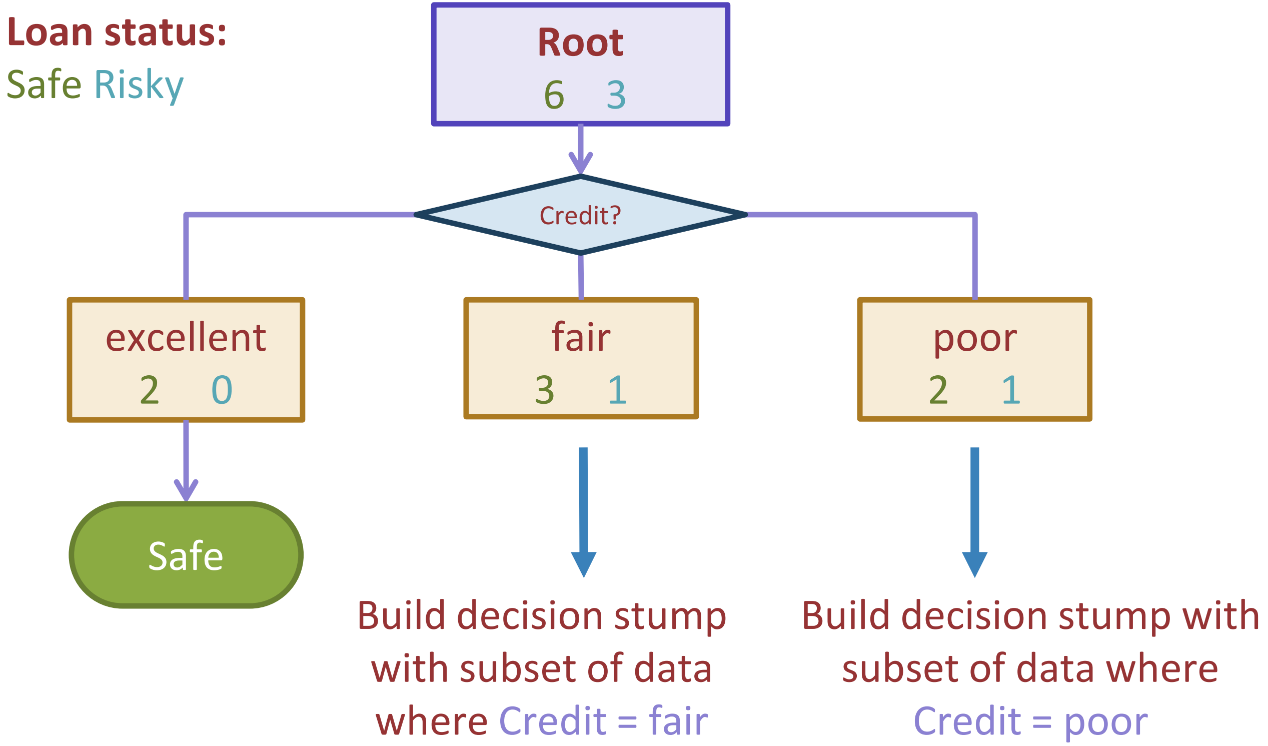


Stopping

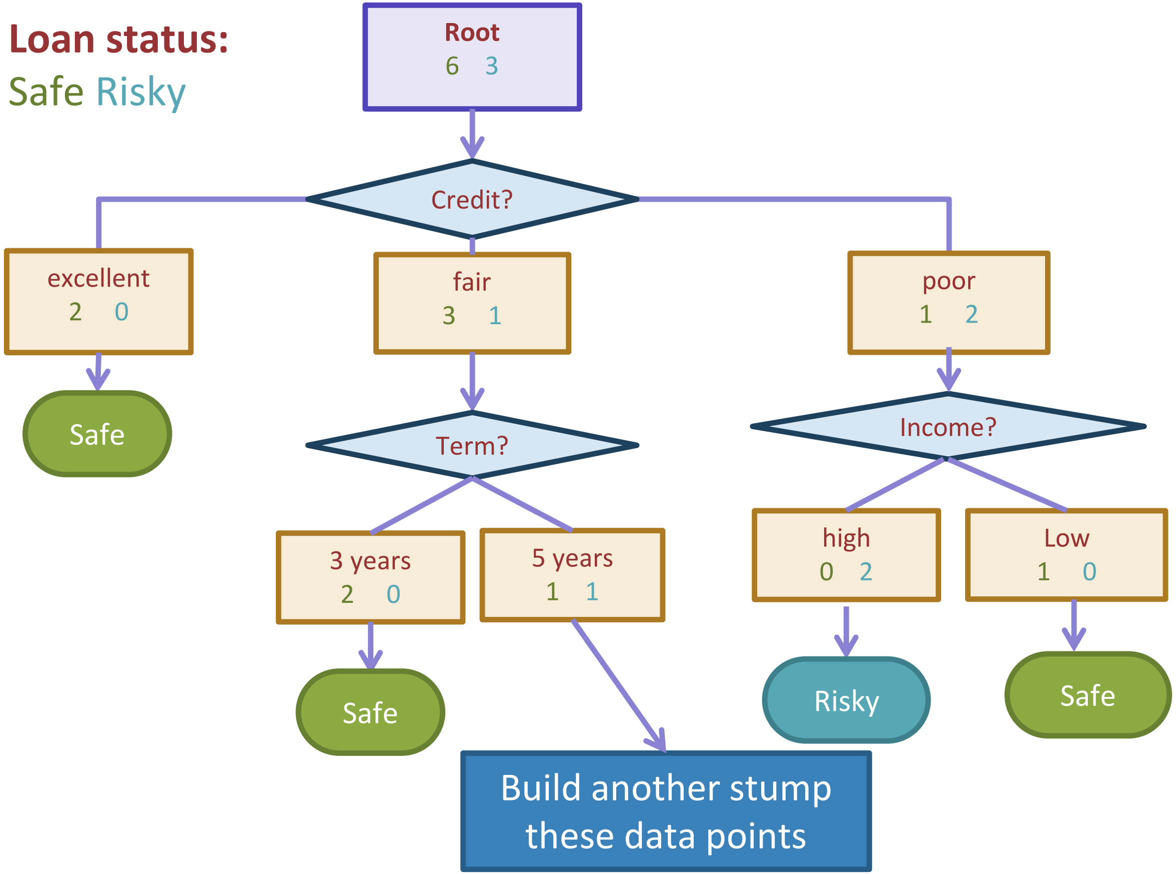
- Stop if all points are in one class



Tree learning = Recursive stump learning



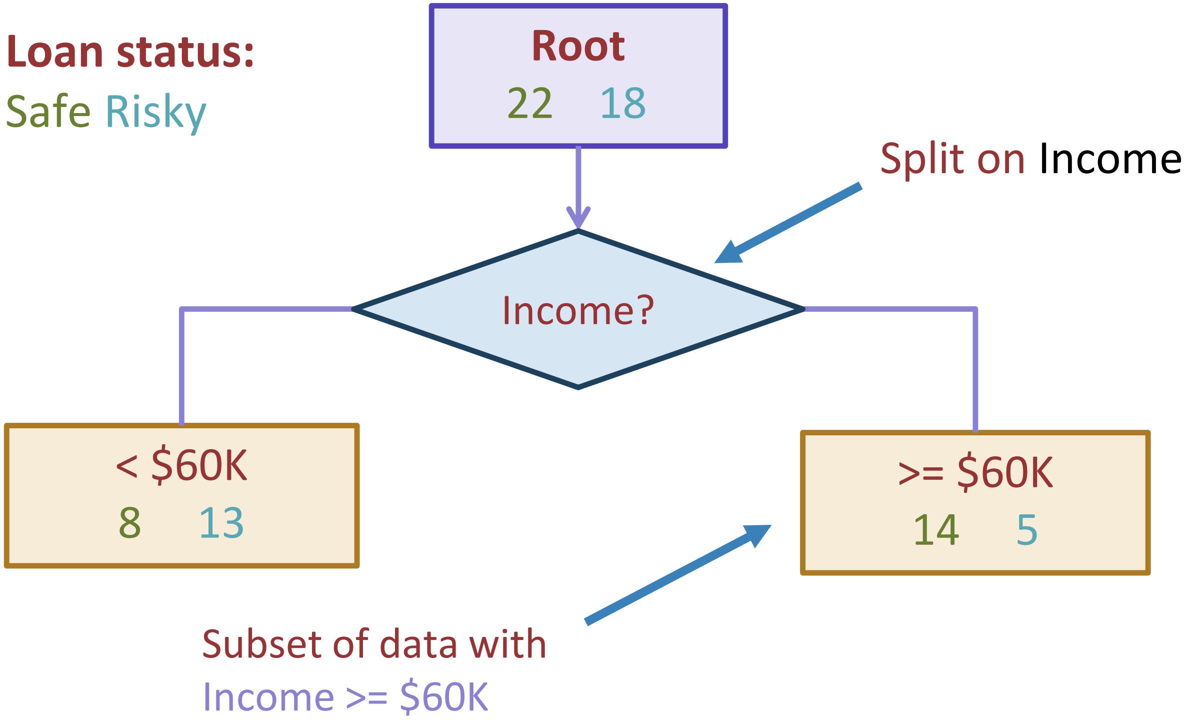
Second level



Real valued features

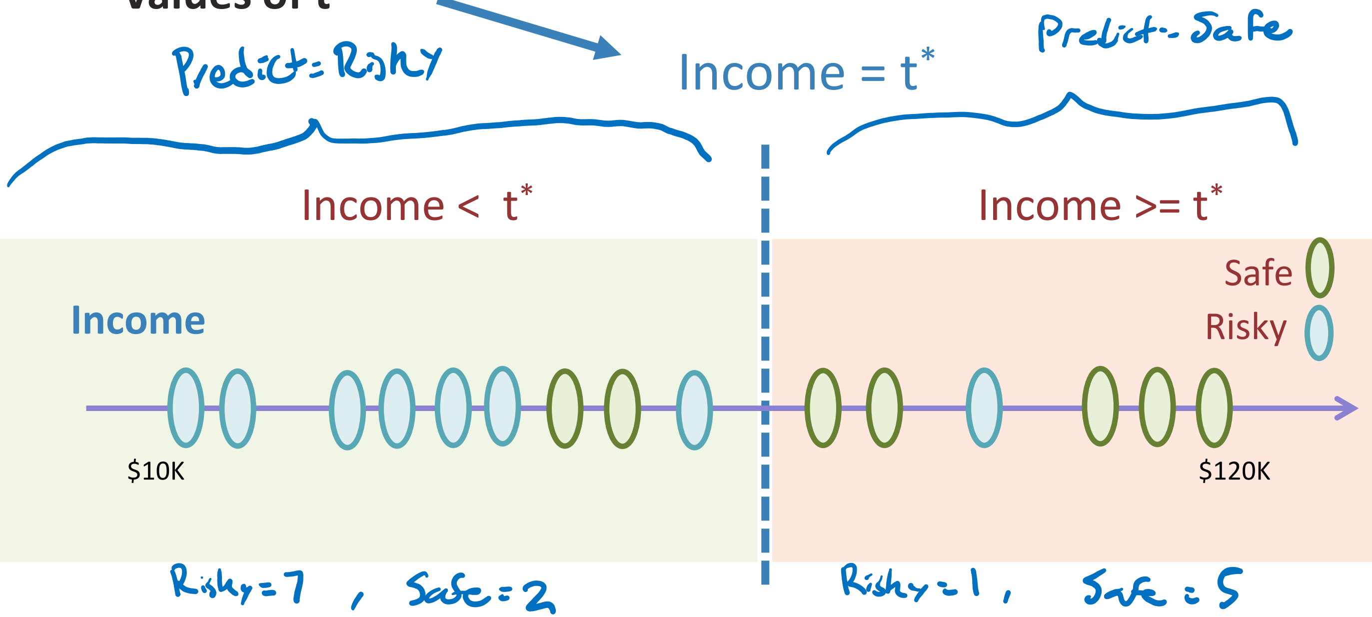
Income	Credit	Term	y
$105 K	excellent	3 yrs	Safe
$112 K	good	5 yrs	Risky
$73 K	fair	3 yrs	Safe
$69 K	excellent	5 yrs	Safe
$217 K	excellent	3 yrs	Risky
$120 K	good	5 yrs	Safe
$64 K	fair	3 yrs	Risky
$340 K	excellent	5 yrs	Safe
$60 K	good	3 yrs	Risky

Threshold split



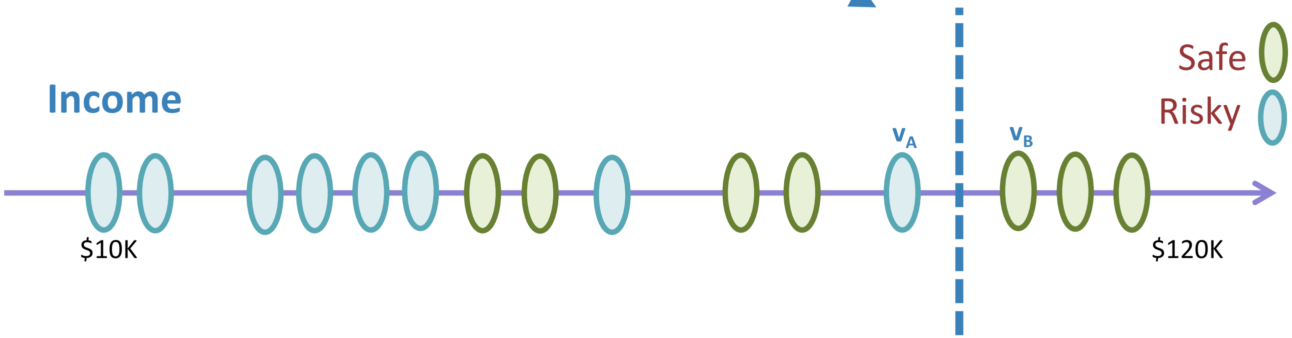
Best threshold?

Infinite possible values of t



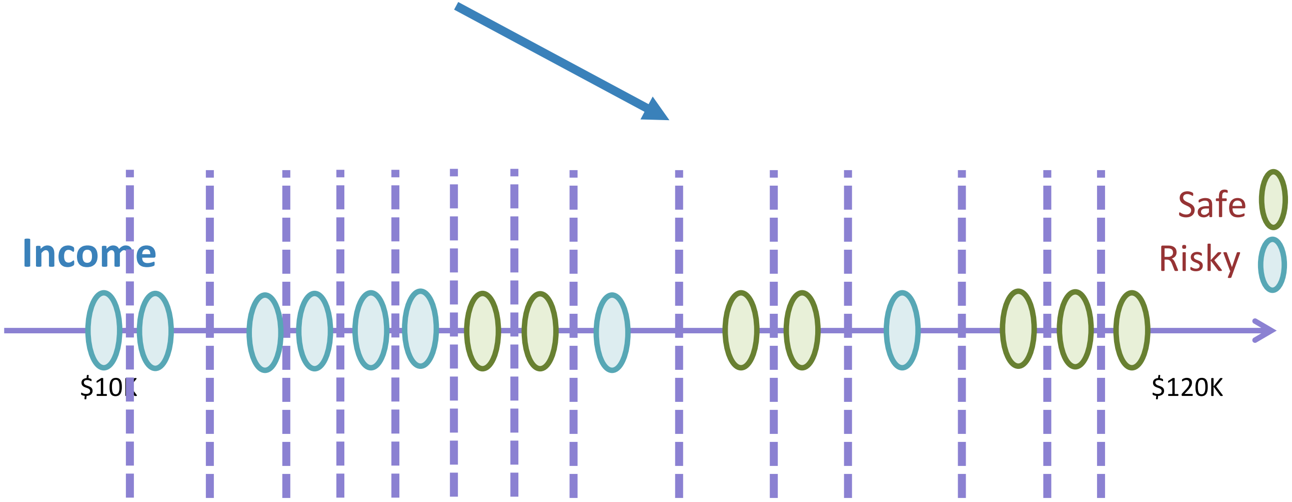
Threshold between points

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



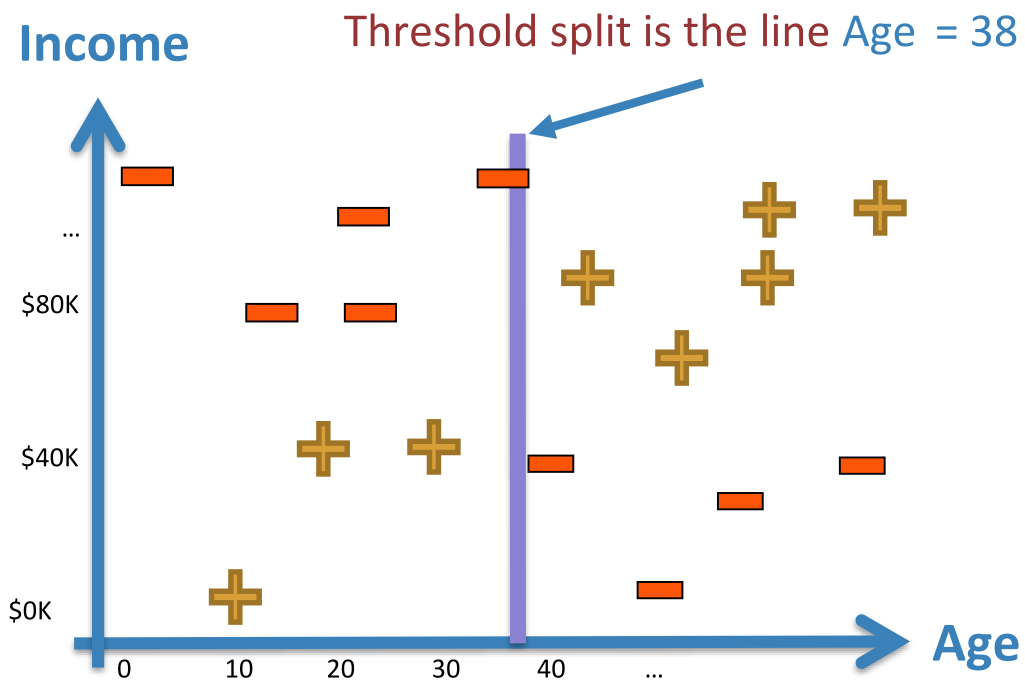
Threshold split selection algorithm

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

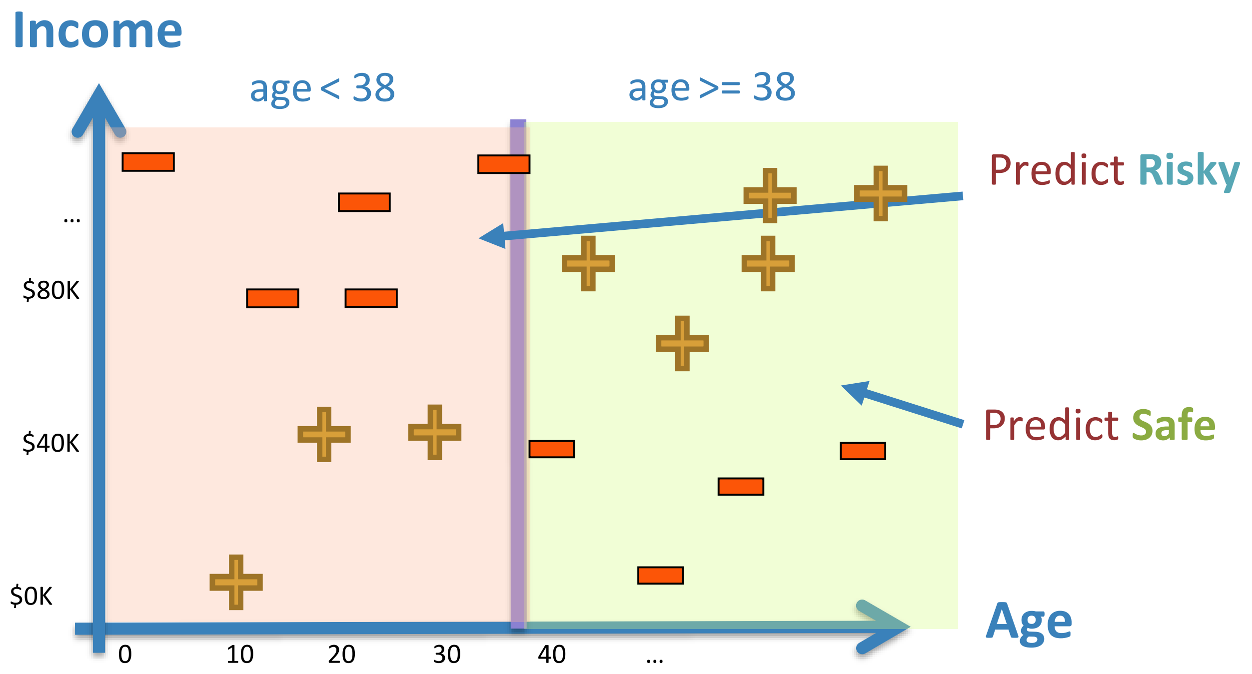
  Let $\{\mathbf{v_1, v_2, v_3, \ldots v_N}\}$ denote sorted values

- Step 2:
  - For i = 1 … N-1
    - Consider split $t_i = (\mathbf{v_i} + \mathbf{v_{i+1}}) / 2$
    - Compute classification error for threshold split $h_j(x) \geq \mathbf{t_i}$
  - Chose the $\mathbf{t^*}$ with the lowest classification error

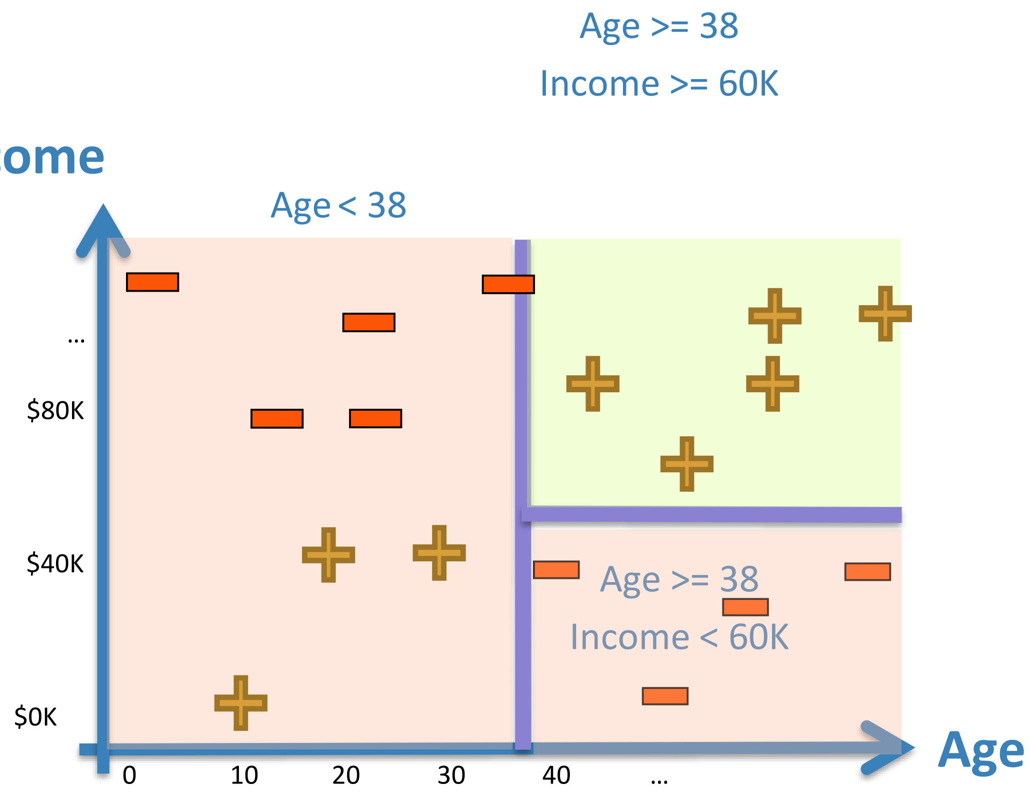
Visualizing the threshold split



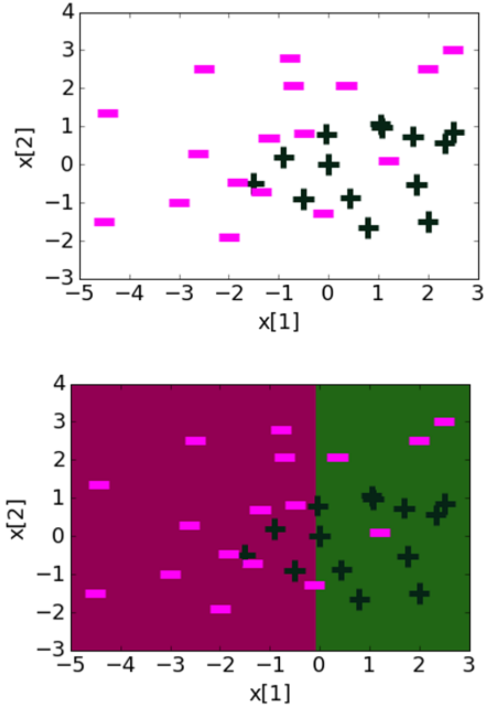
Split on Age >= 38

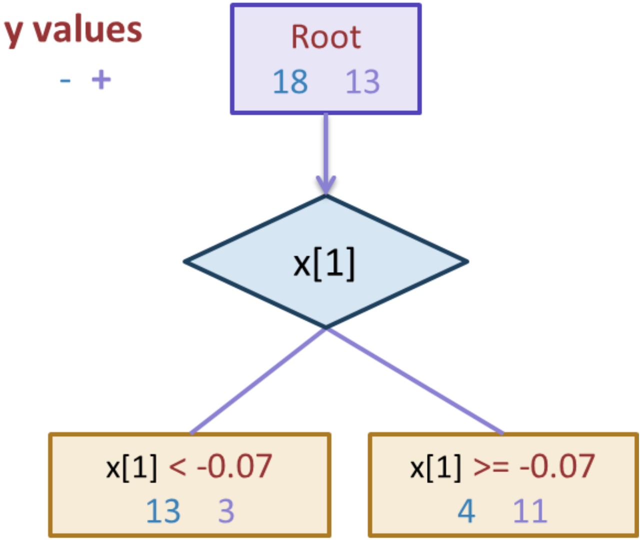


Each split partitions the 2-D space

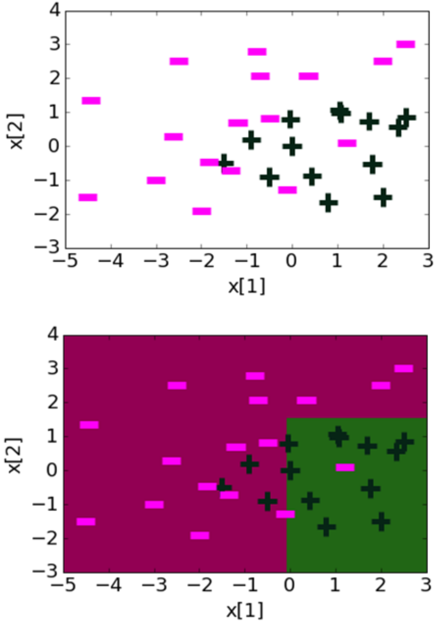


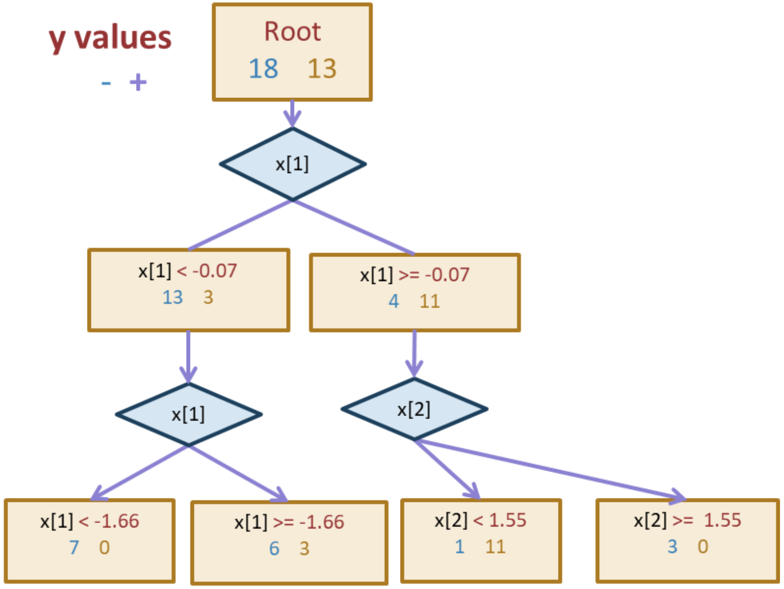
Depth 1: Split on x[1]





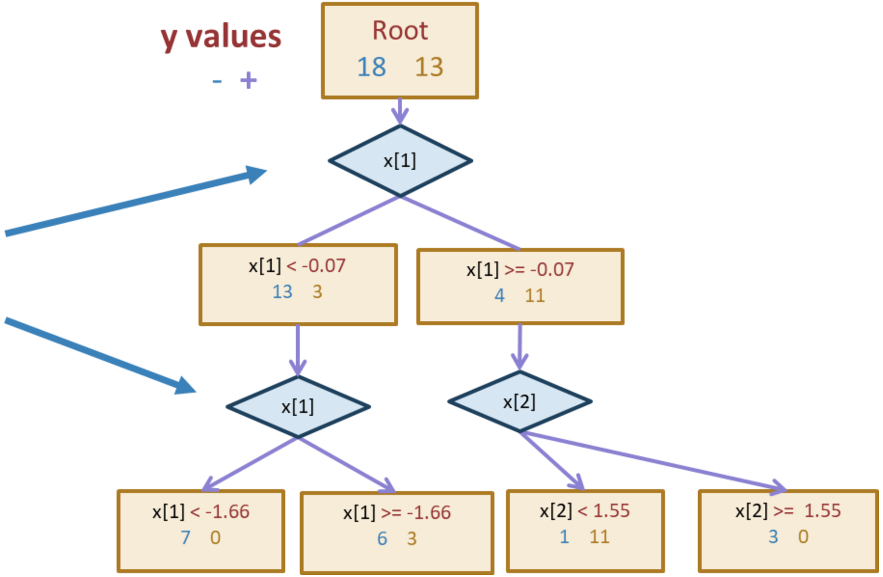
Depth 2


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

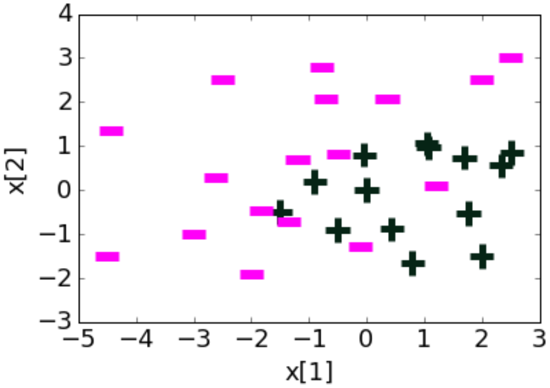
Threshold split caveat

For threshold splits, same feature can be used multiple times

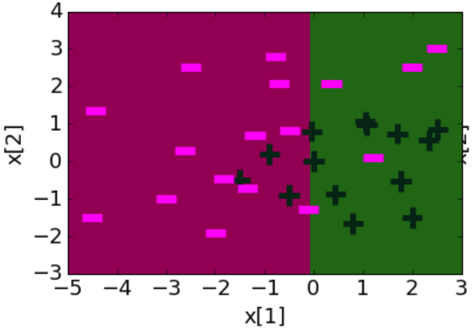


Decision boundaries

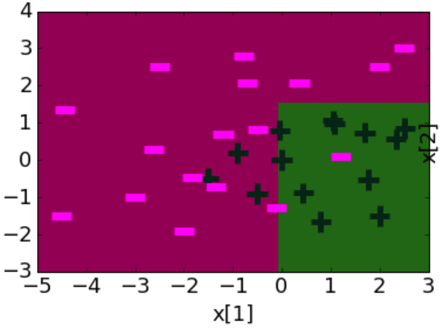
- Decision boundaries can be complex!



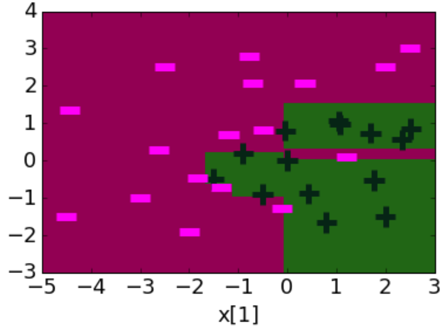
Depth 1



Depth 2



Depth 10

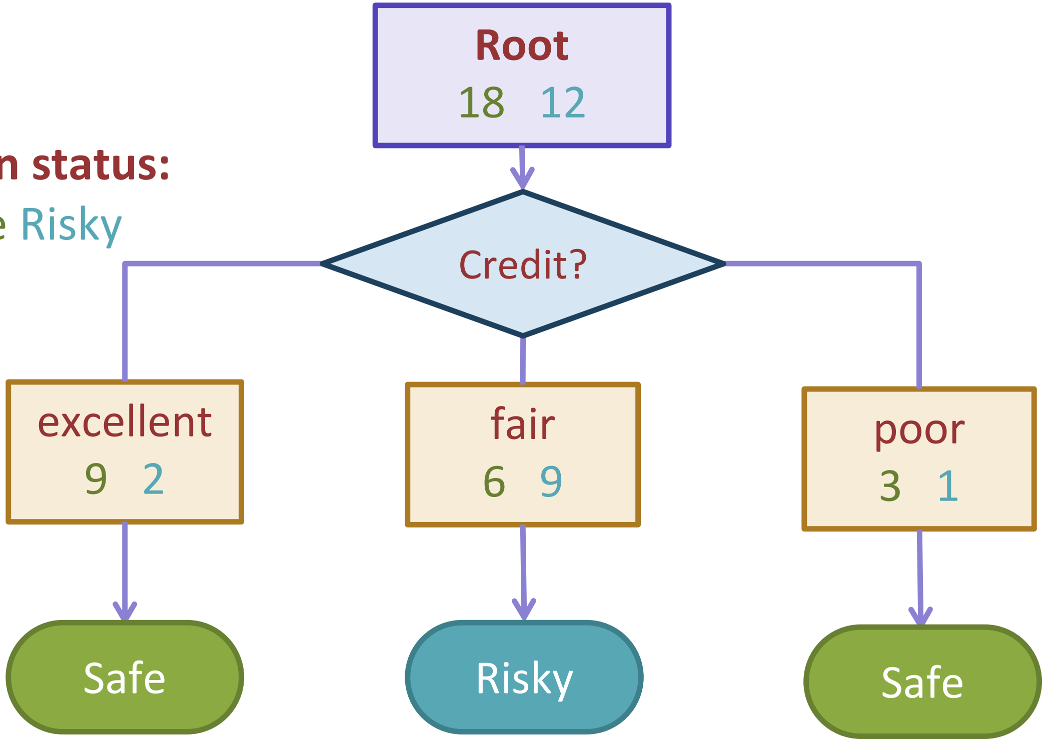


Overfitting

- 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:
  - Early stopping
    - Fixed length depth
    - Stop if error does not considerably decrease
  - Pruning
    - Grow full length trees
    - Prune nodes to balance a complexity penalty

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 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