

Probability Classifier

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

Probability Classifier

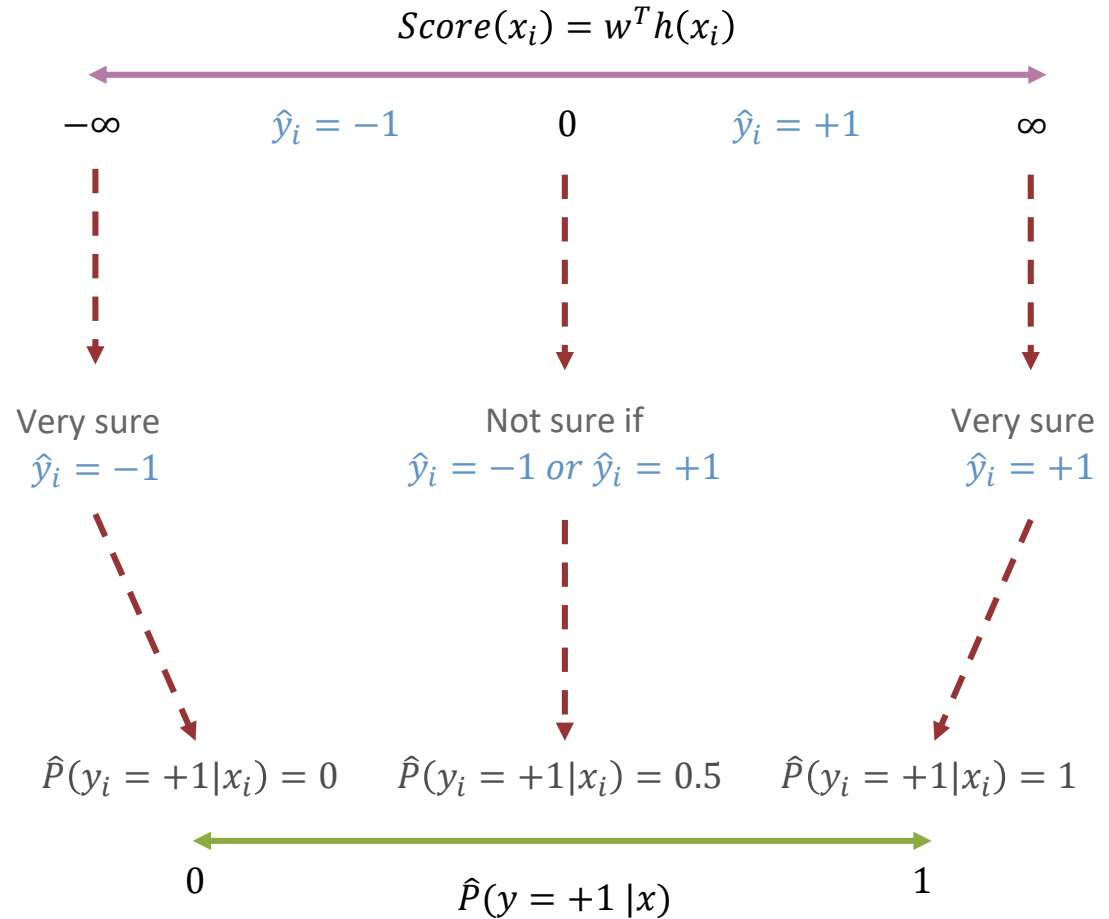
Input x : Sentence from review

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

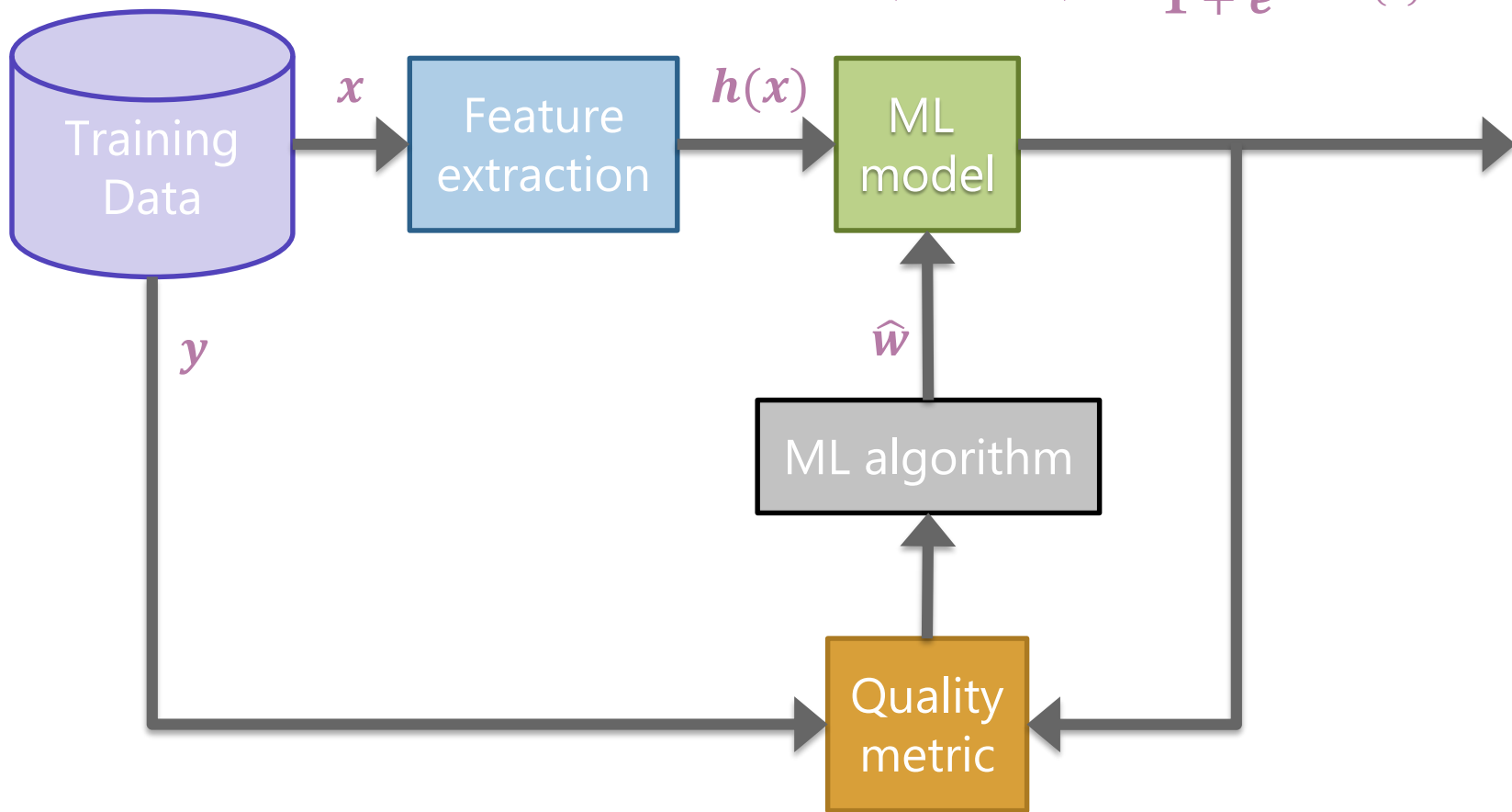
Notes:

- Estimating the probability improves **interpretability**

Interpreting Score



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



Naïve Bayes

Idea: Naïve Bayes

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

$P(y = +1 | x = \text{"The sushi \& everything else was awesome!"})?$

$P(y = -1 | x = \text{"The sushi \& everything else was awesome!"})?$

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

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

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

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

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

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

Compute Probabilities

How do we compute something like

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

How do we compute something like

$$P(\textit{“awesome”} | y = +1)?$$



Zeros

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

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

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:

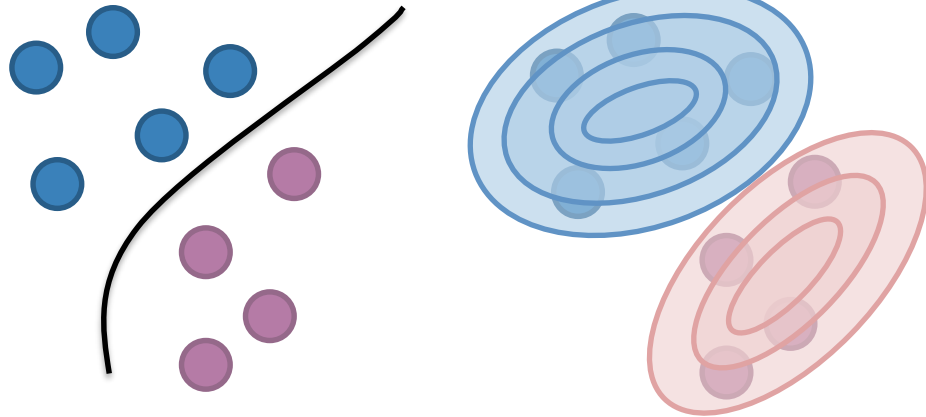
$$P(y|x_1, x_2, \dots, x_d) = \prod_{j=1}^d P(x_j|y) P(y)$$



Compare Models

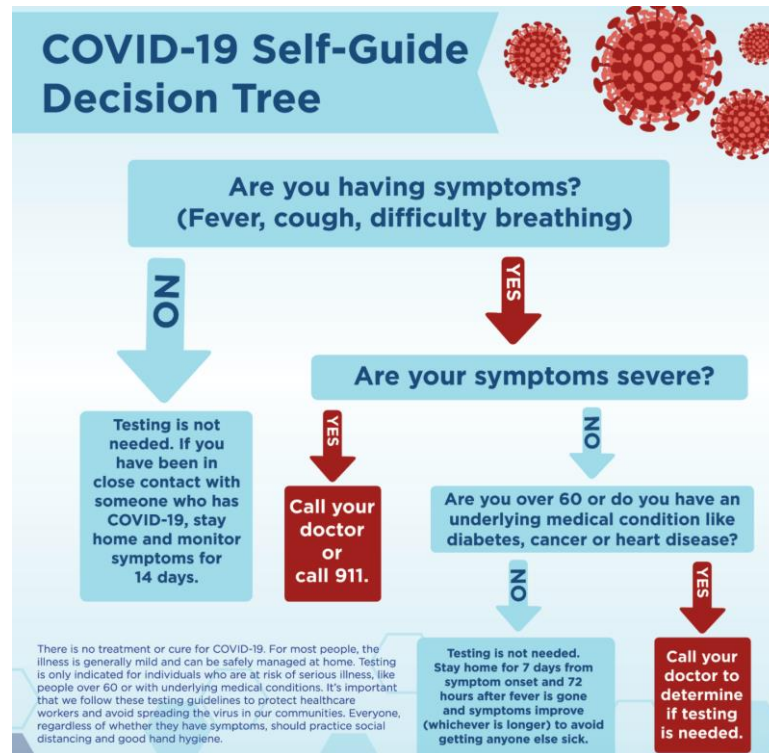
Generative: defines a model for generating x (e.g. Naïve Bayes)

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



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.



What makes a loan risky?

I want to buy a new house!



Loan Application



Credit History



Income



Term



Personal Info

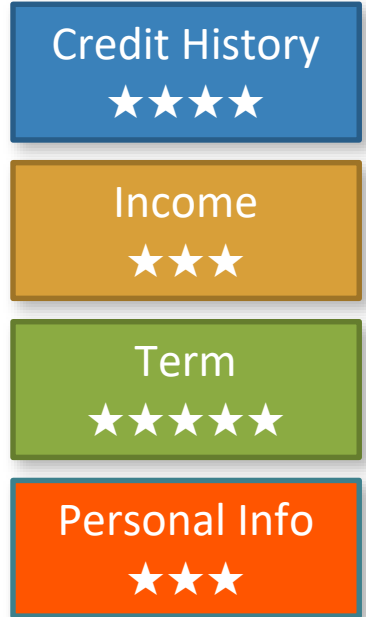


Credit history explained

Did I pay previous loans on time?



Example: excellent, good, or fair



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,...



Credit History



Income



Term



Personal Info



Personal information

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

Example: Home loan for a
married couple

Credit History



Income



Term



Personal Info



Intelligent application

Loan Applications

A pink-bordered loan application form with various fields and text.A blue-bordered loan application form with various fields and text.A green-bordered loan application form with various fields and text.

Intelligent loan application review system

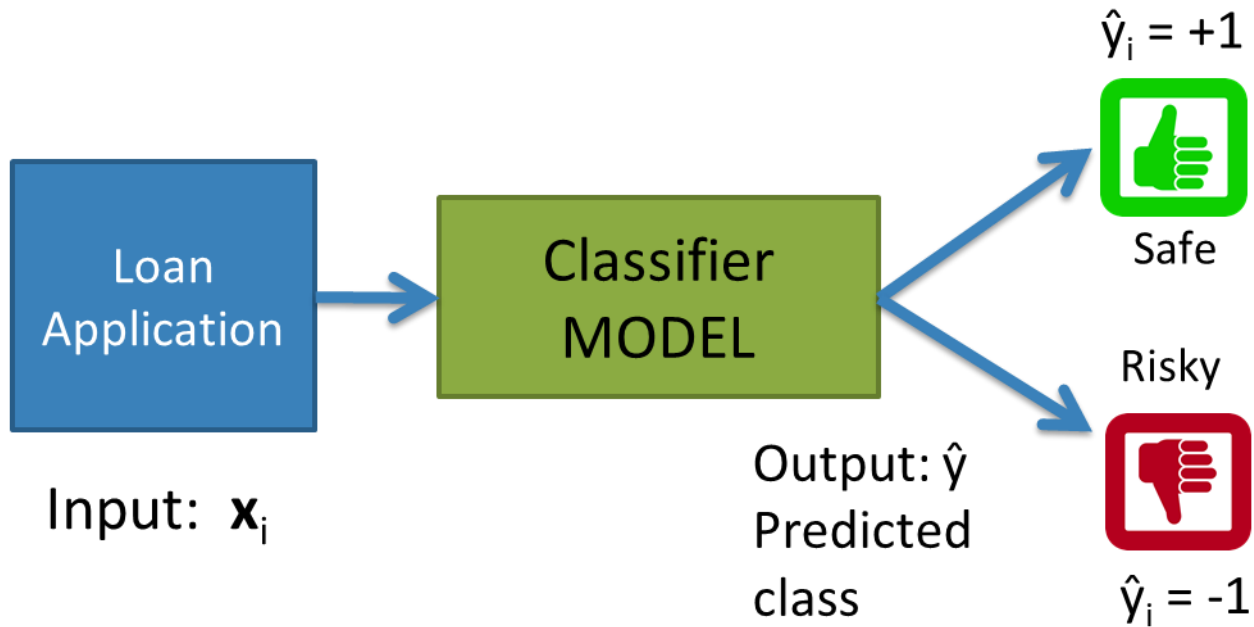
Safe
✓

Risky
X

Risky
X



Classifier review



Setup

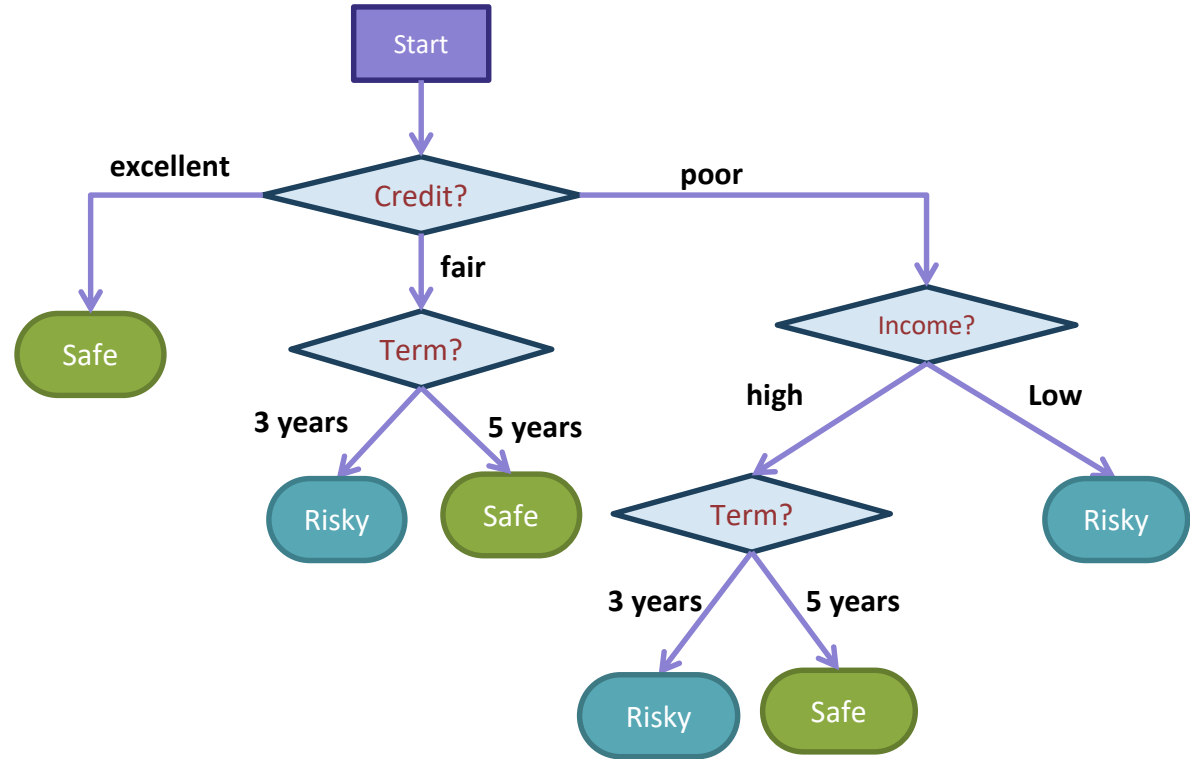
Data (N observations, 3 features)

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!

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?



Visual Notation

Loan status: **Safe** **Risky**



of Risky loans

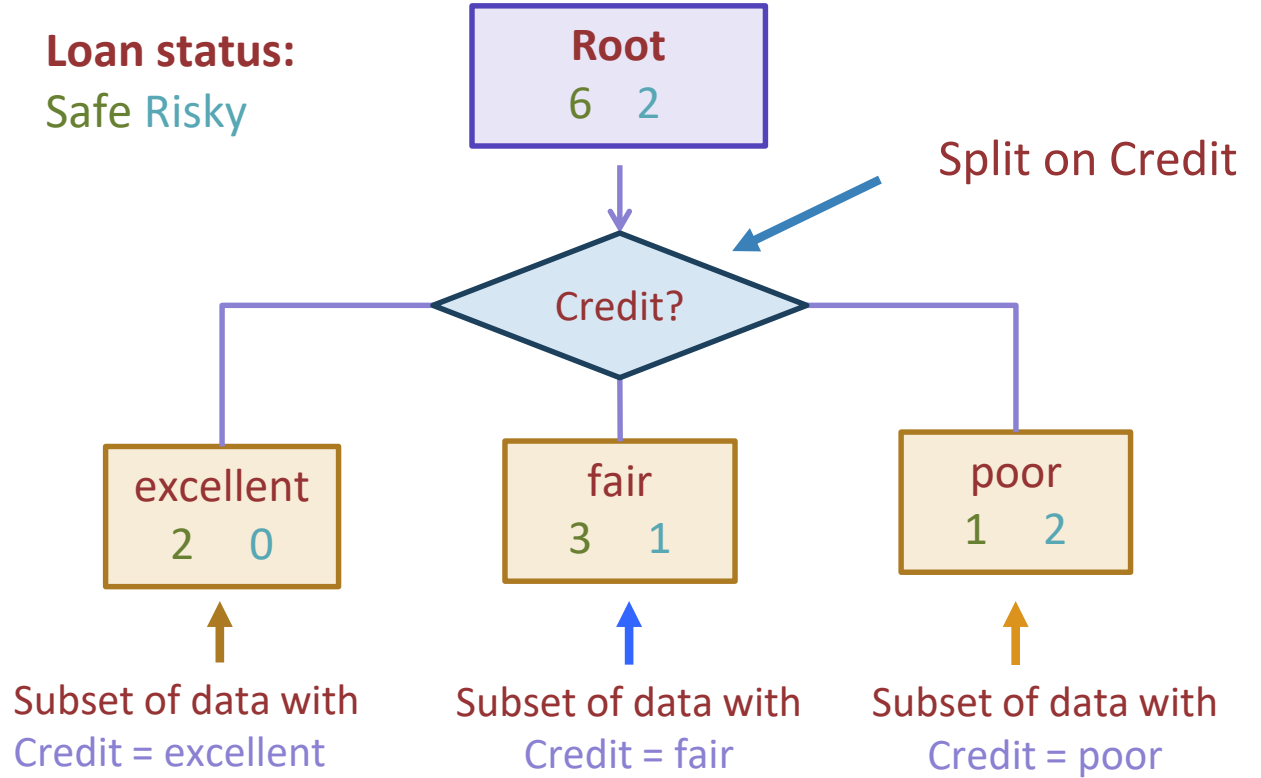
of Safe loans

$N = 9$ examples

Decision stump: 1 level

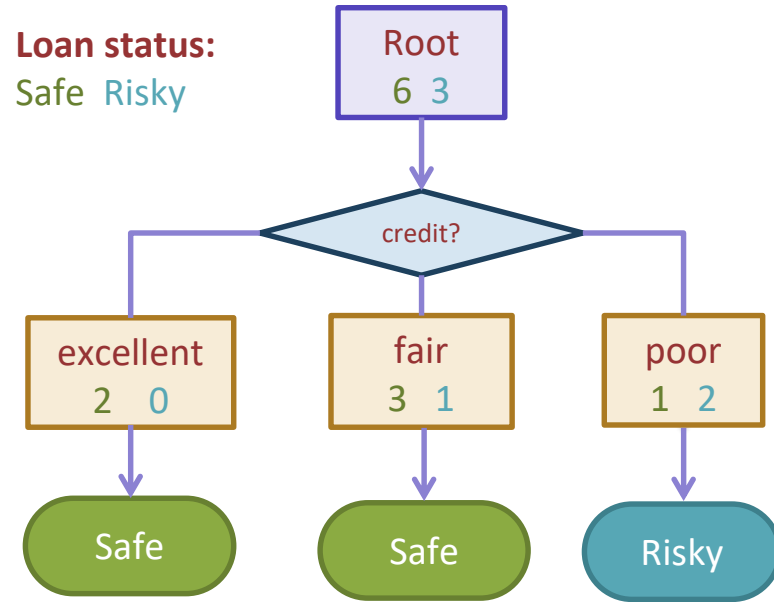
Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	safe
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe

Loan status:
Safe Risky



Making predictions

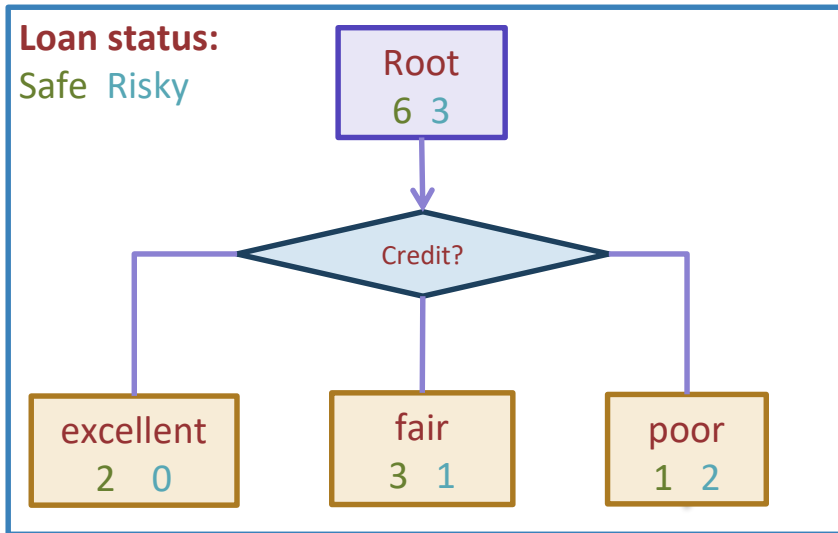
For each leaf node, set \hat{y} = majority value



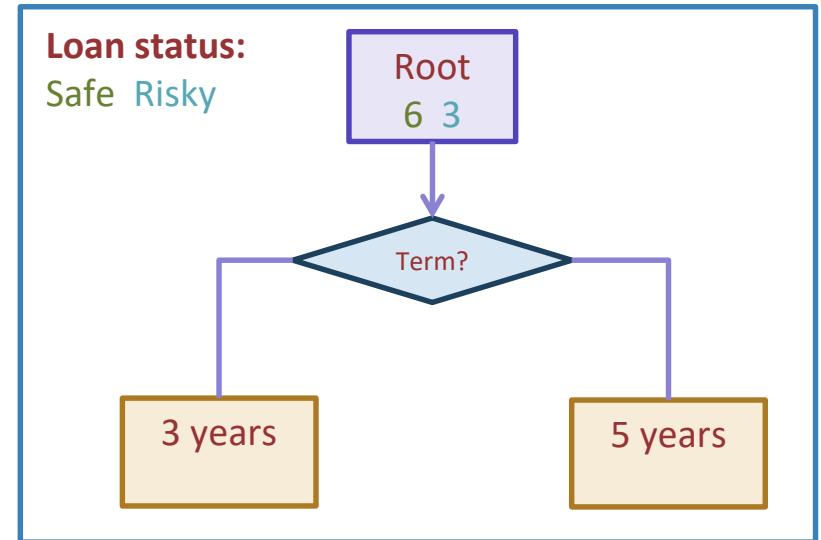
How do we select the best feature?

-
- * Select the split with lowest classification error

Choice 1: Split on Credit



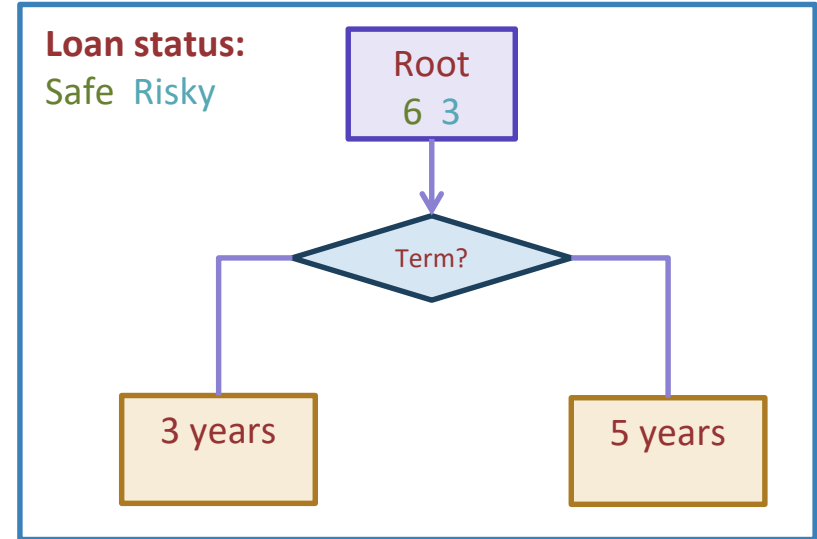
Choice 2: Split on Term



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

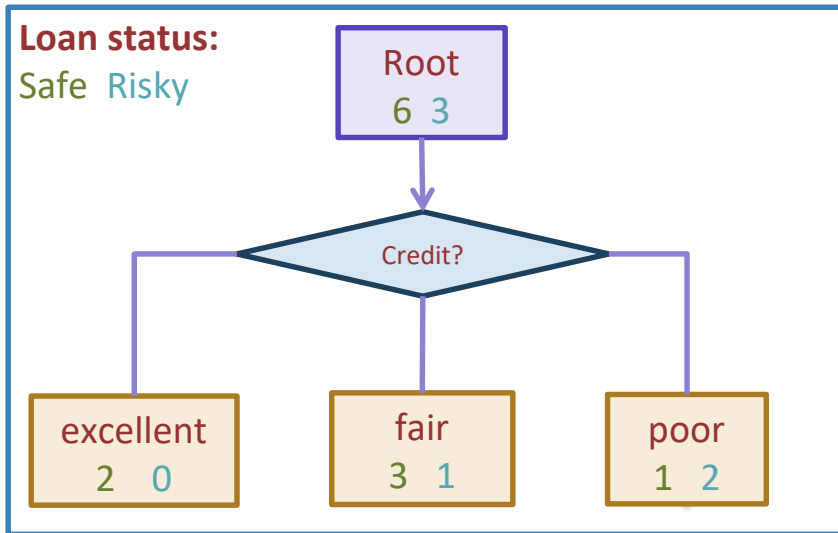
Choice 2: Split on Term



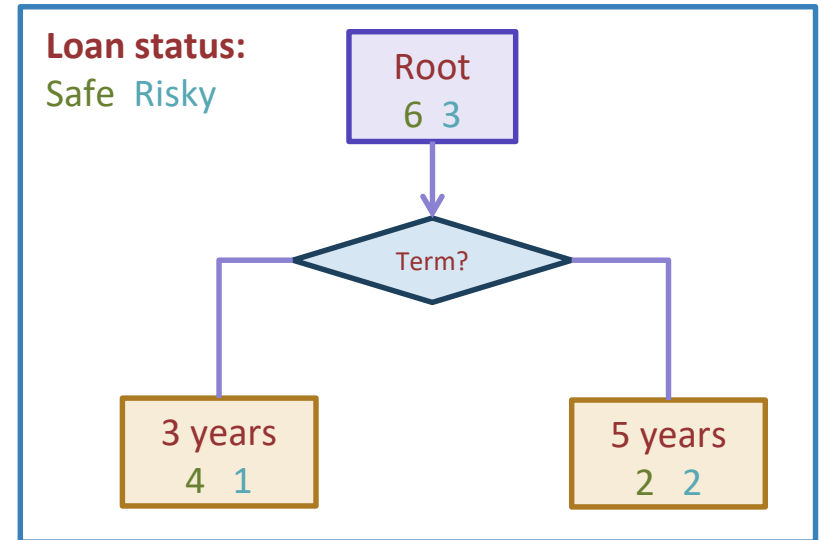
How do we select the best feature?

Select the split with lowest classification error

Choice 1: Split on Credit

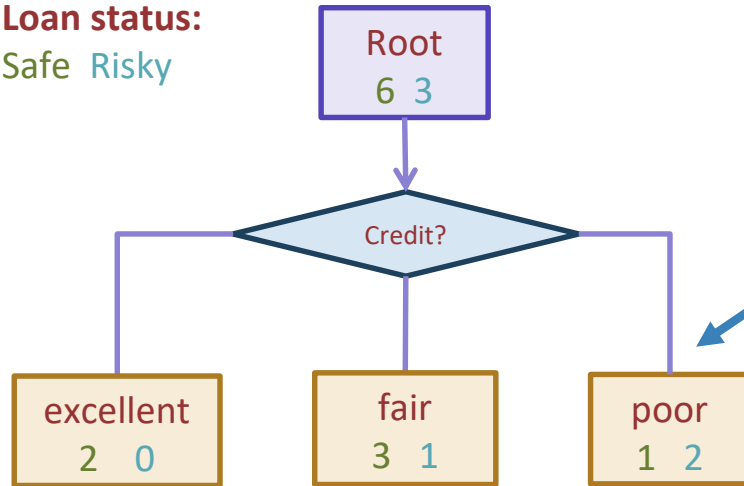


Choice 2: Split on Term



How do we measure effectiveness of a split?

Loan status:
Safe Risky



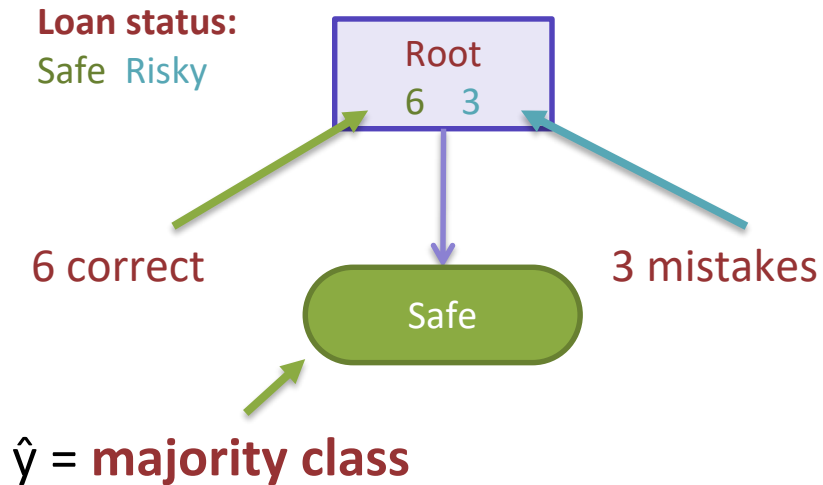
Idea: Calculate classification error of this decision stump

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

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



Error = _____

=

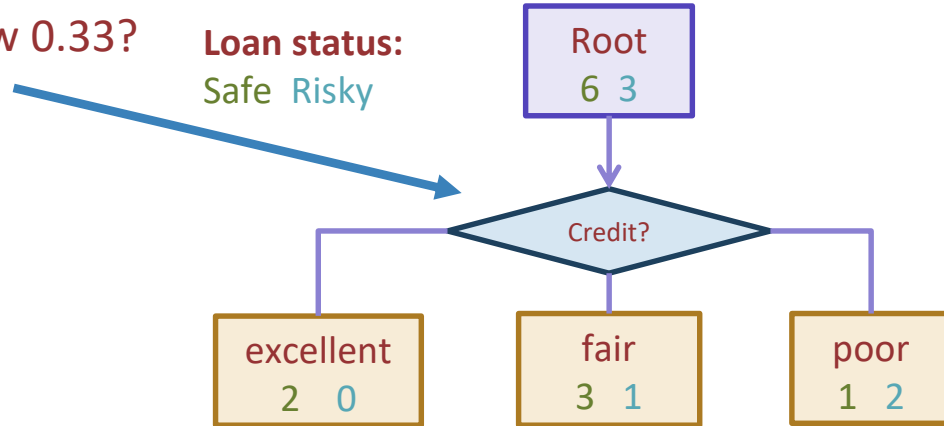
Tree	Classification error
(root)	0.33

Choice 1: Split on Credit history?

Does a split on Credit reduce classification error below 0.33?

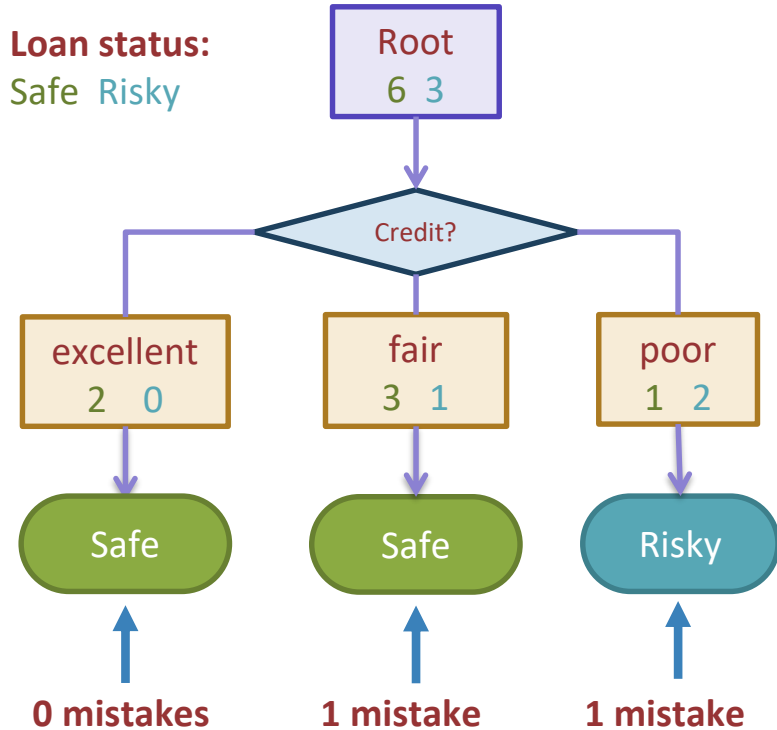
Loan status:
Safe Risky

Choice 1: Split on Credit



Split on Credit: Classification error

Choice 1: Split on Credit



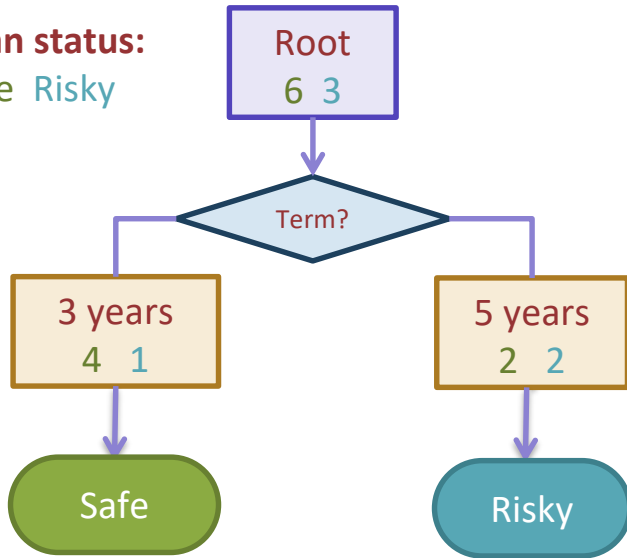
Error = _____
=

Tree	Classification error
(root)	0.33
Split on credit	0.22

Choice 2: Split on Term?

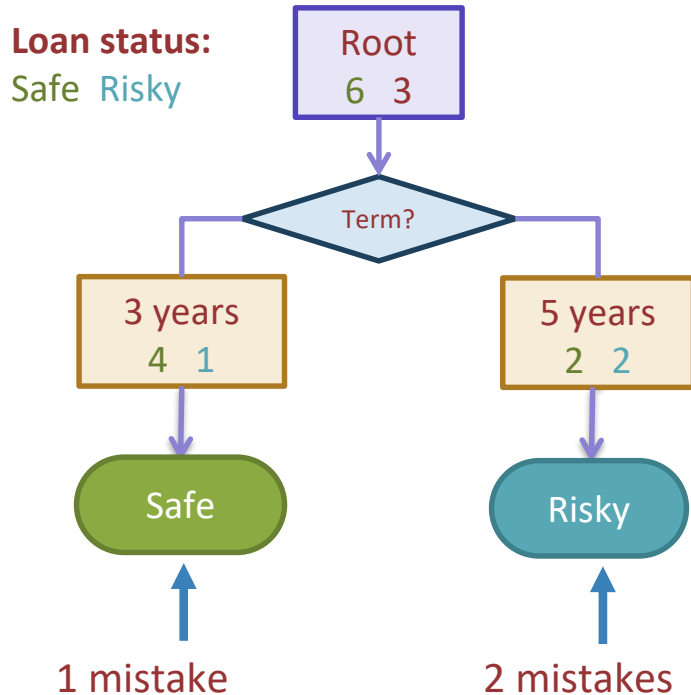
Choice 2: Split on Term

Loan status:
Safe Risky



Evaluating the split on Term

Choice 2: Split on Term



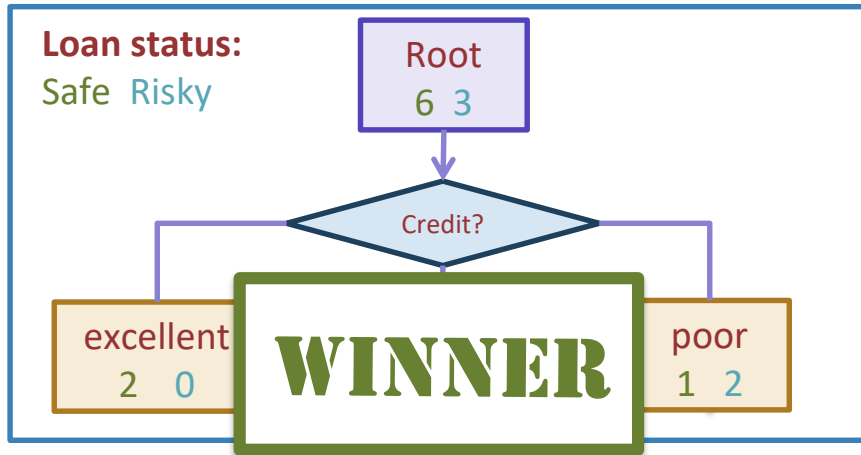
Error = _____
=

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

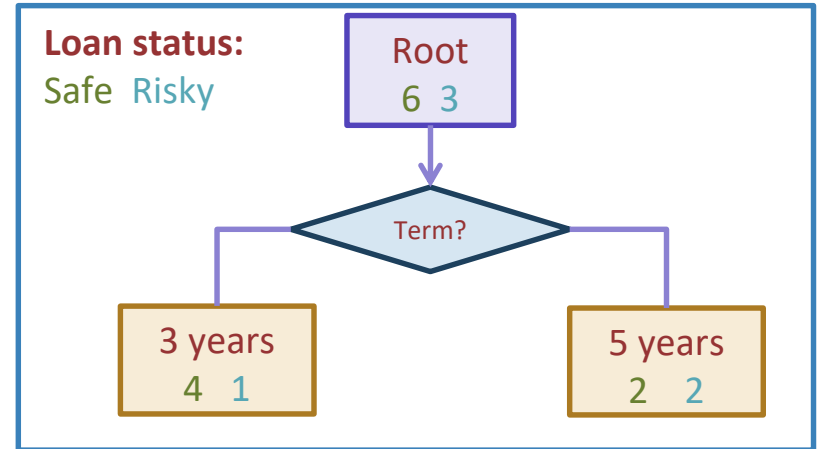
Choice 1 vs Choice 2: Comparing split on credit vs term

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 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

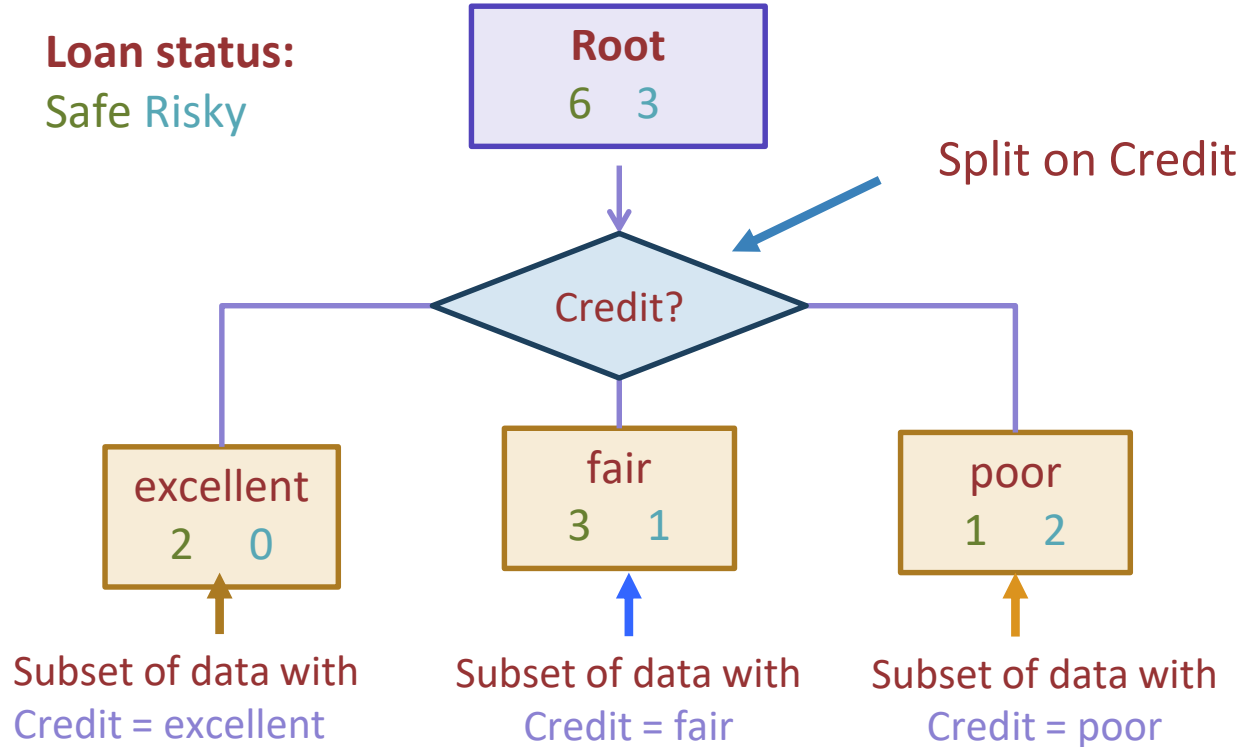


Greedy Algorithm

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

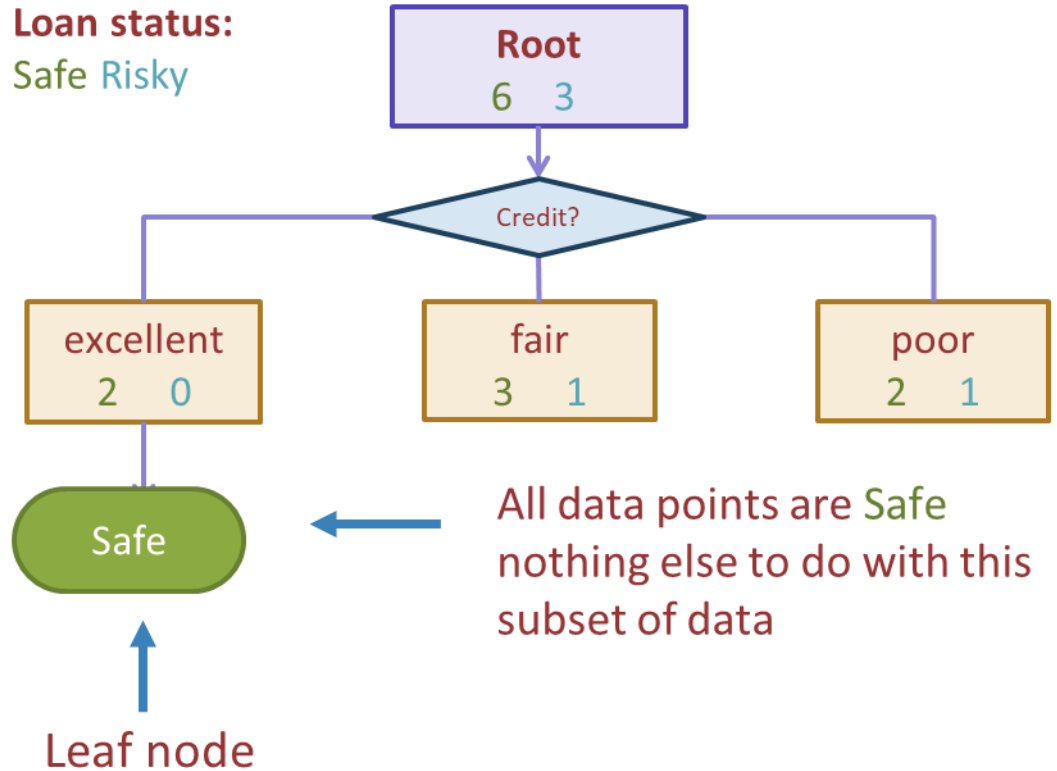


Decision
stump:
1 level

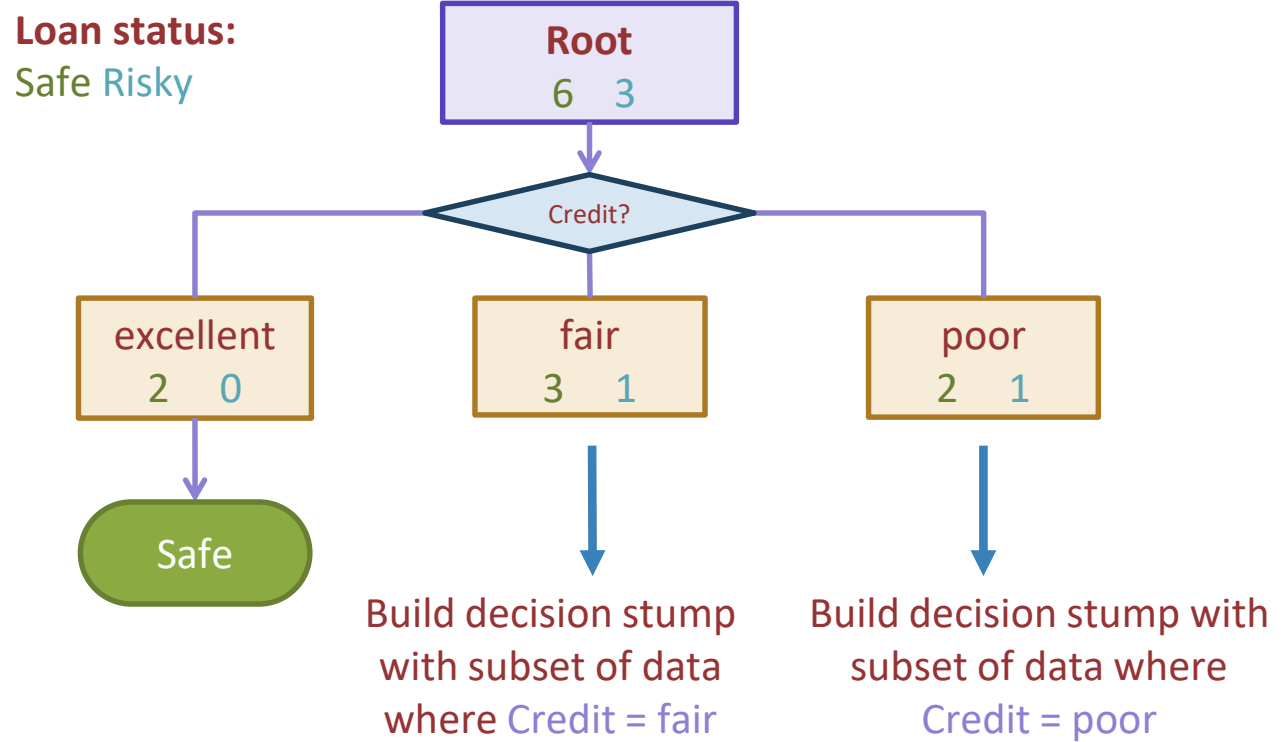


Stopping

- Stop if all points are in one class

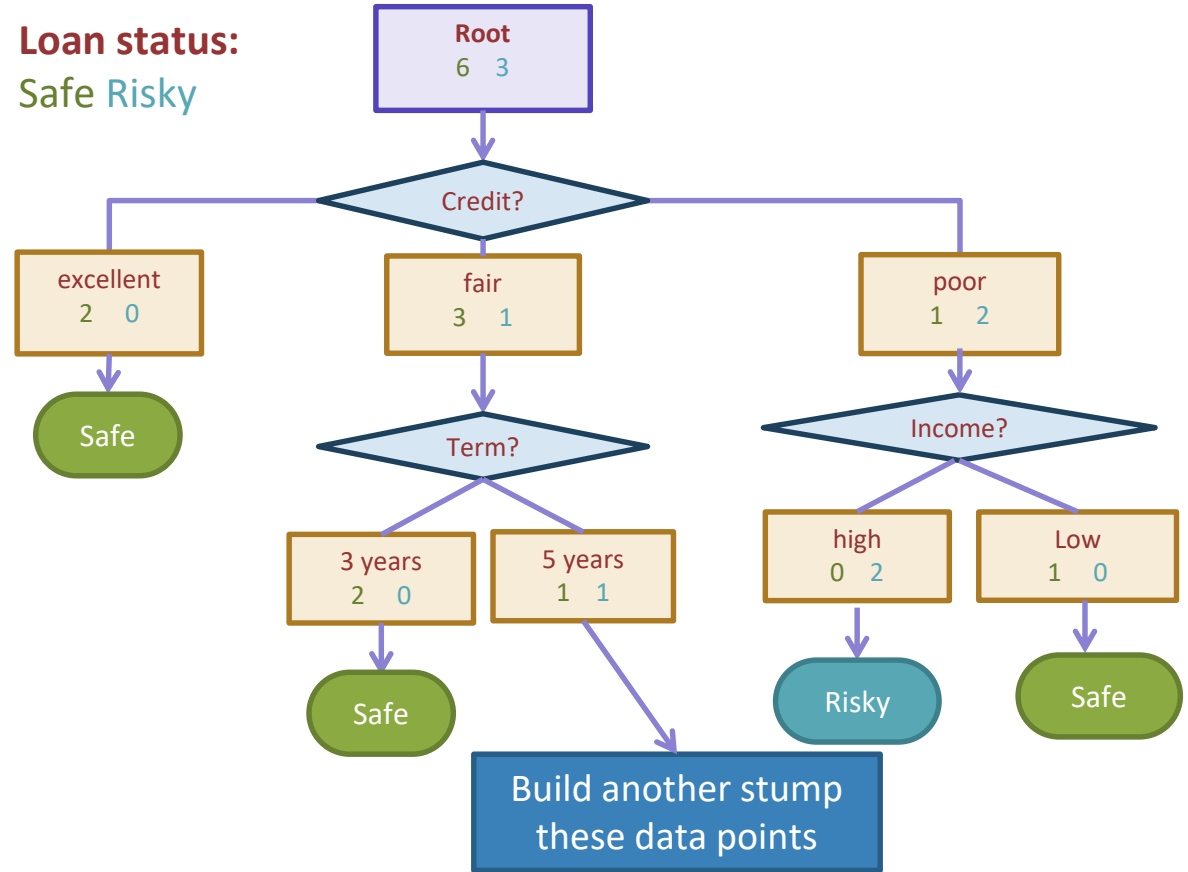


Tree learning =
Recursive
stump learning



Second level

Loan status:
Safe Risky

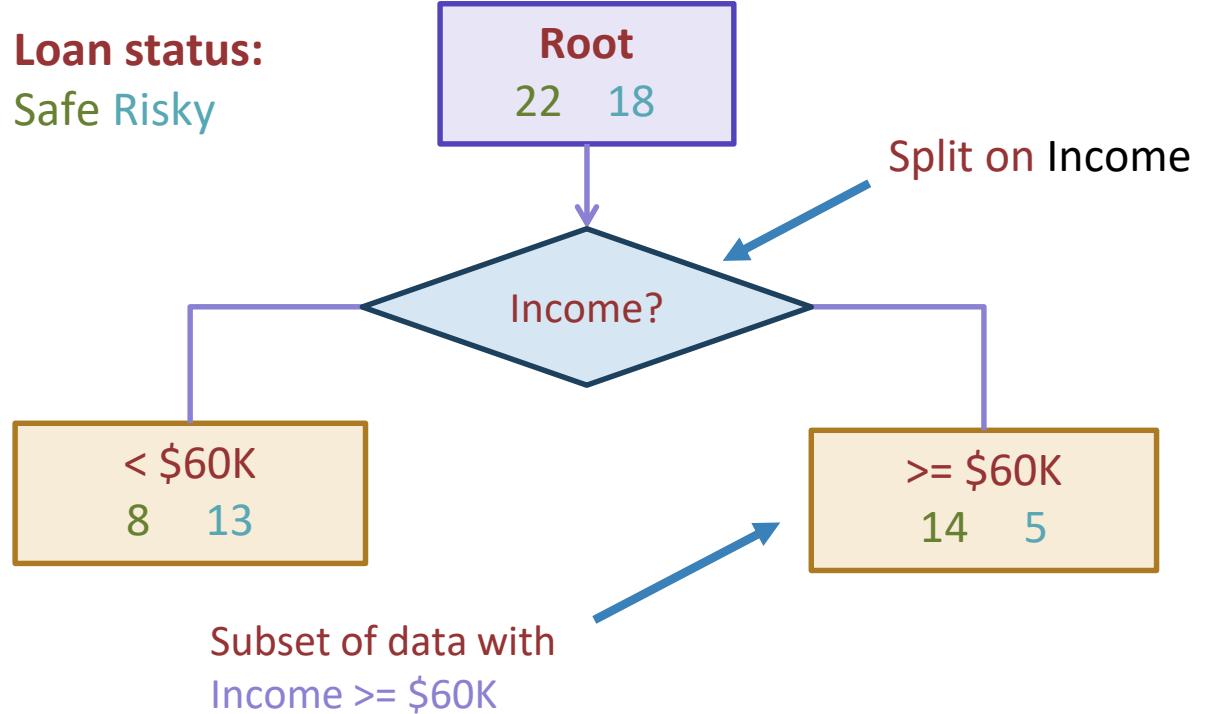


*Real valued
features*

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

Loan status:
Safe Risky

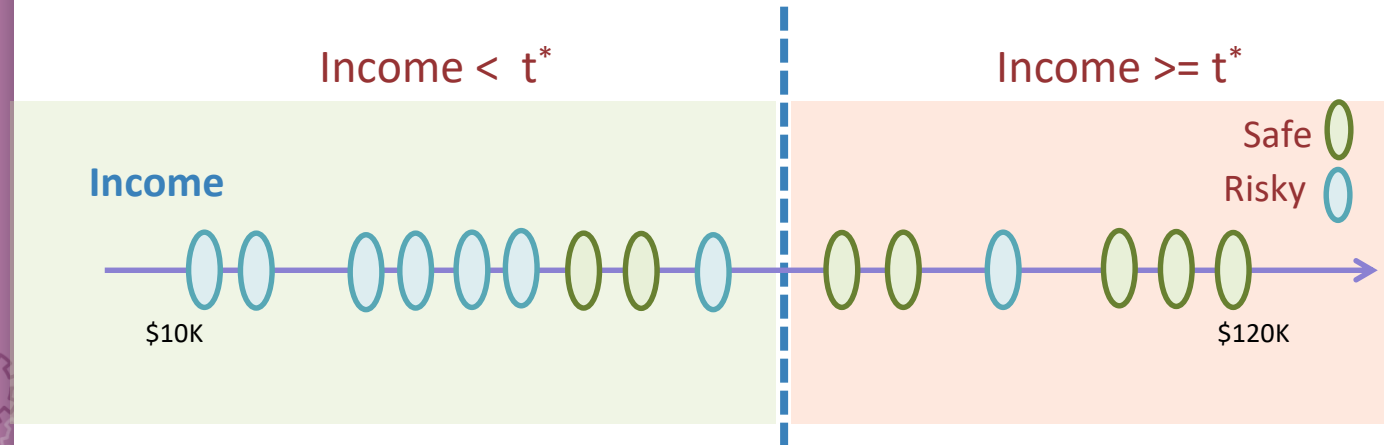


Best threshold?

Infinite possible values of t

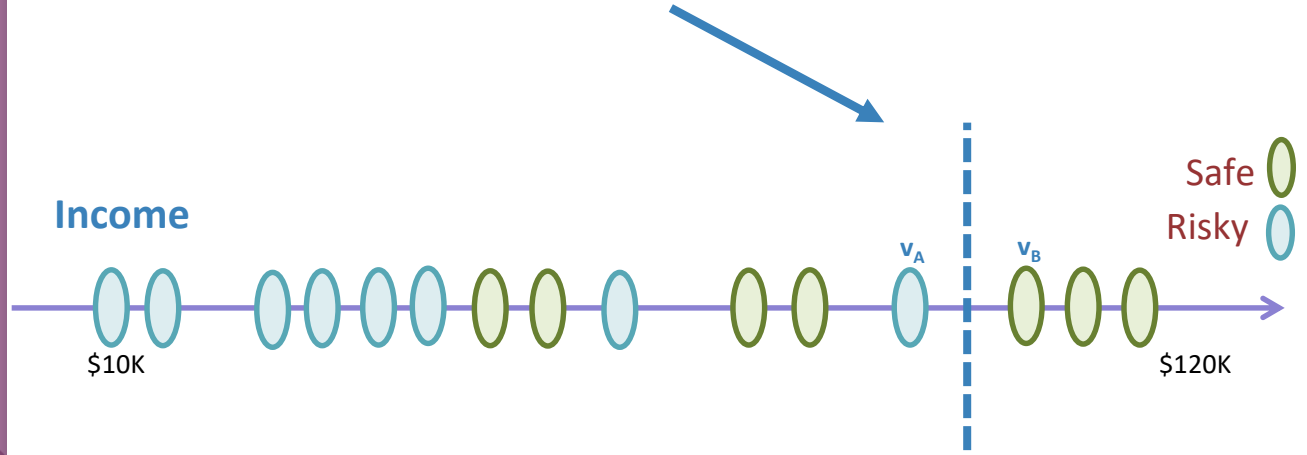


Income = t^*



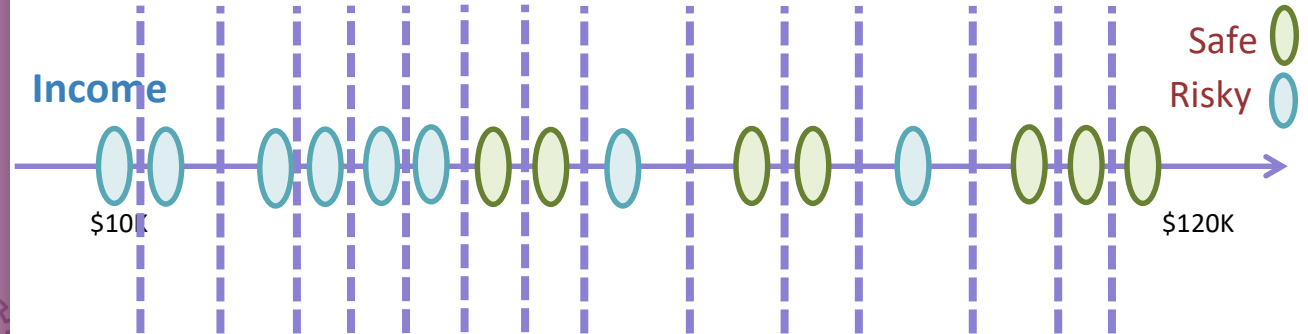
Threshold between points

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

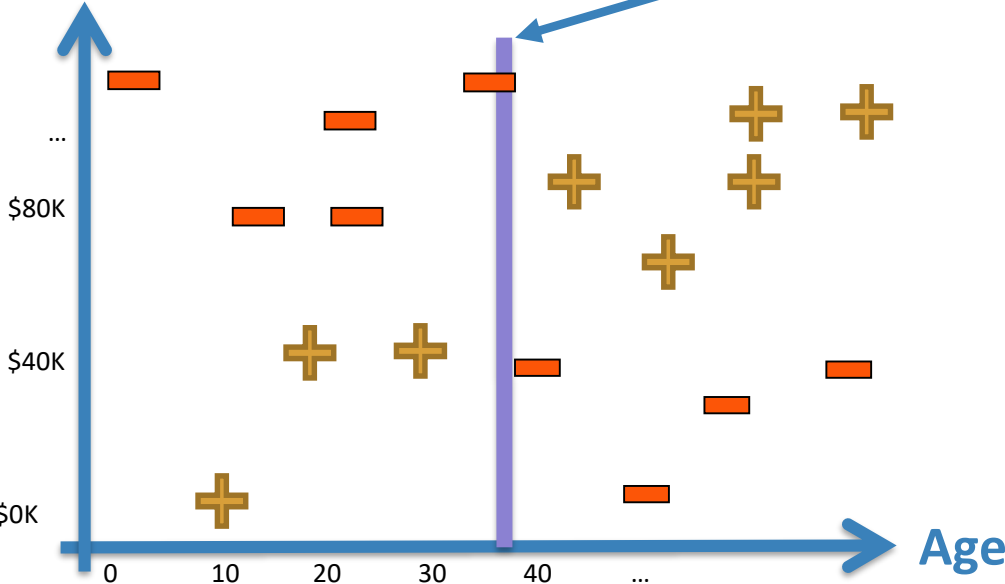
- **Step 1:** Sort the values of a feature $h_j(x)$:
Let $\{v_1, v_2, v_3, \dots, v_N\}$ denote sorted values
- **Step 2:**
 - For $i = 1 \dots N-1$
 - Consider split $t_i = (v_i + v_{i+1}) / 2$
 - Compute classification error for threshold split $h_j(x) \geq t_i$
 - Chose the t^* with the lowest classification error



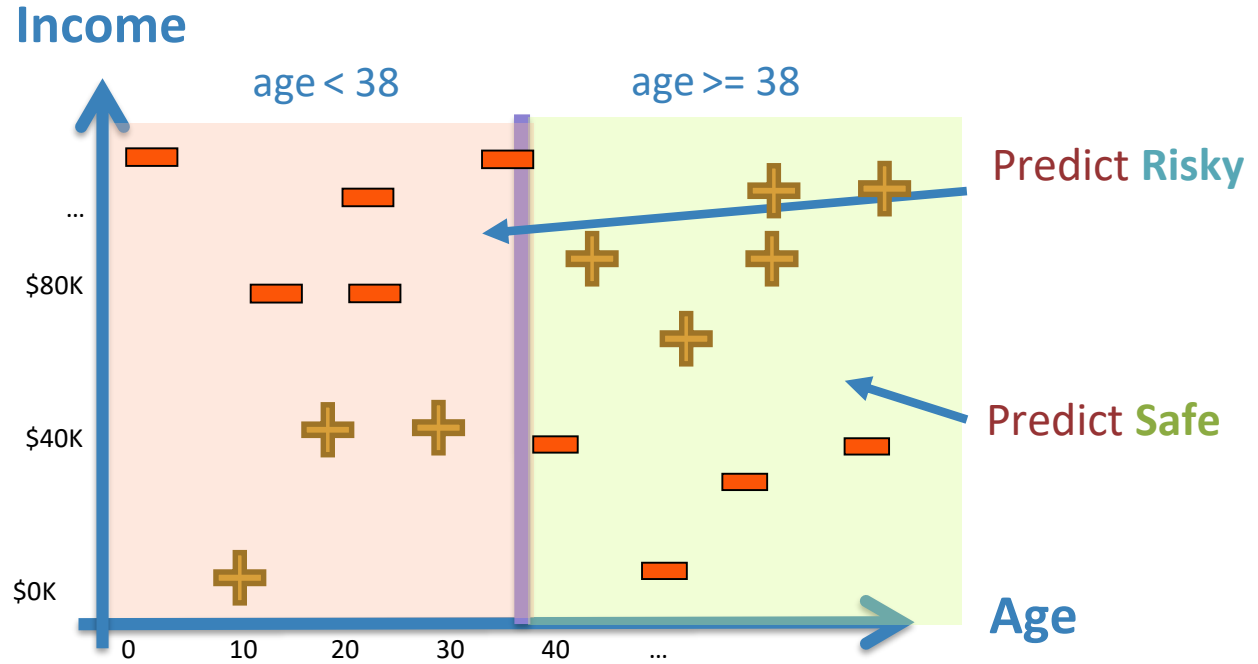
Visualizing the threshold split

Income

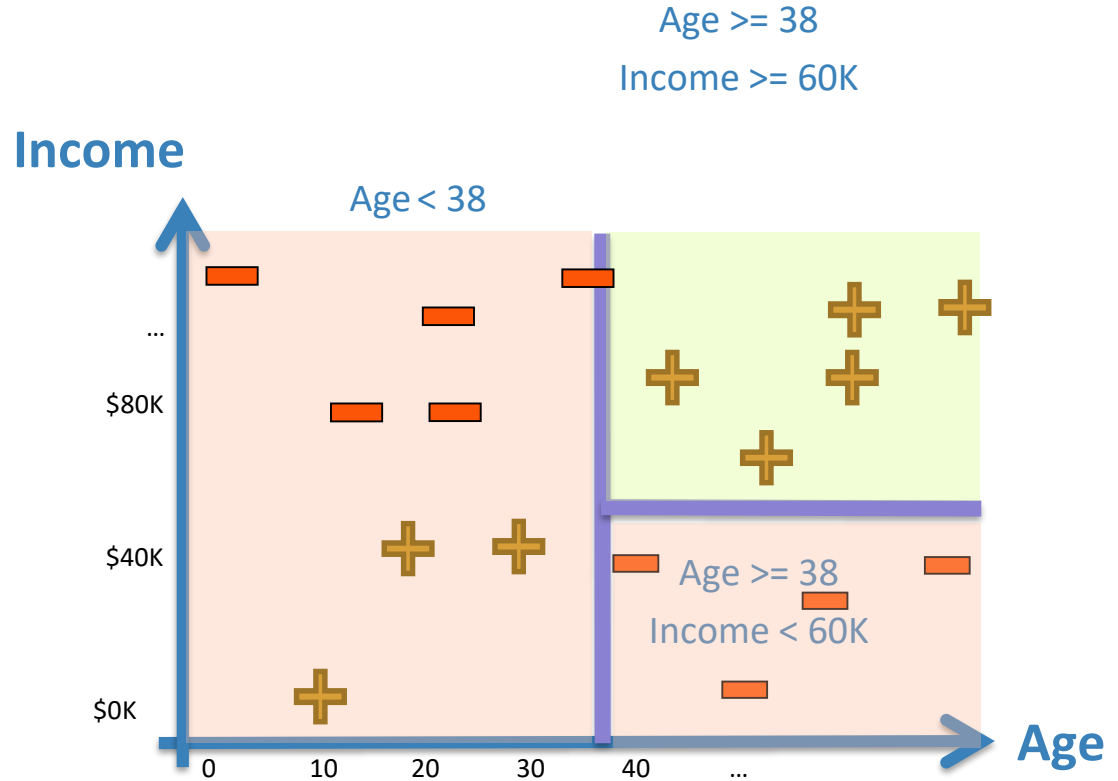
Threshold split is the line Age = 38



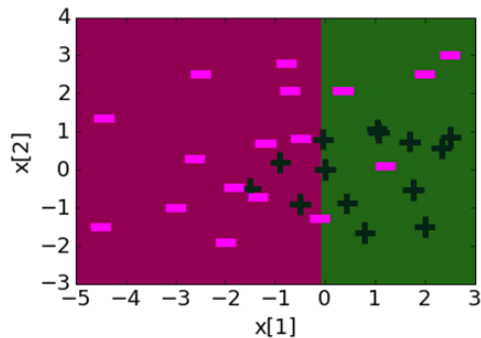
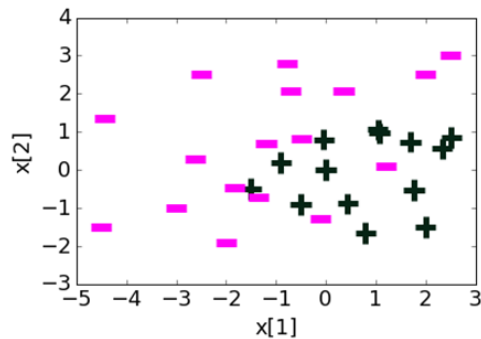
Split on Age \geq 38



Each split
partitions the
2-D space

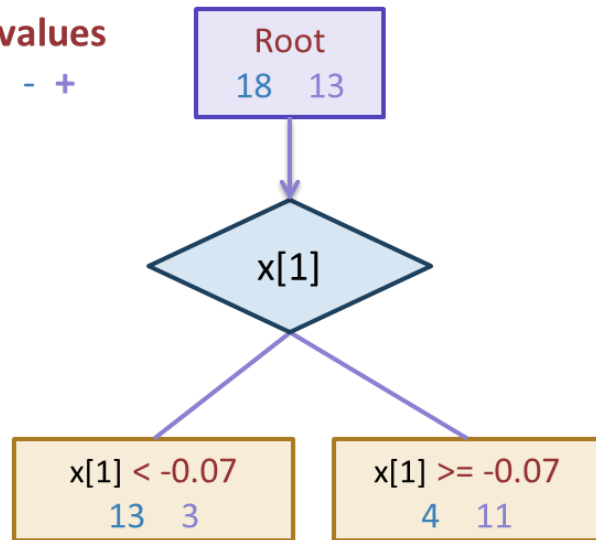


Depth 1: Split on $x[1]$

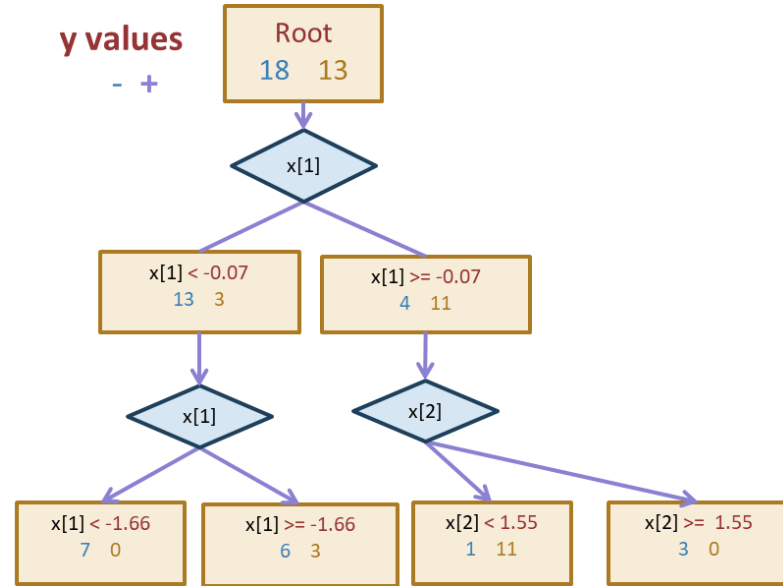
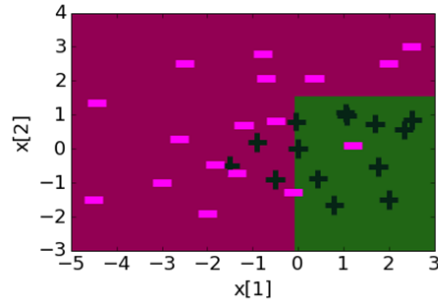
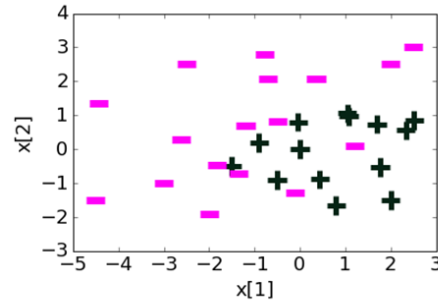


y values

- +

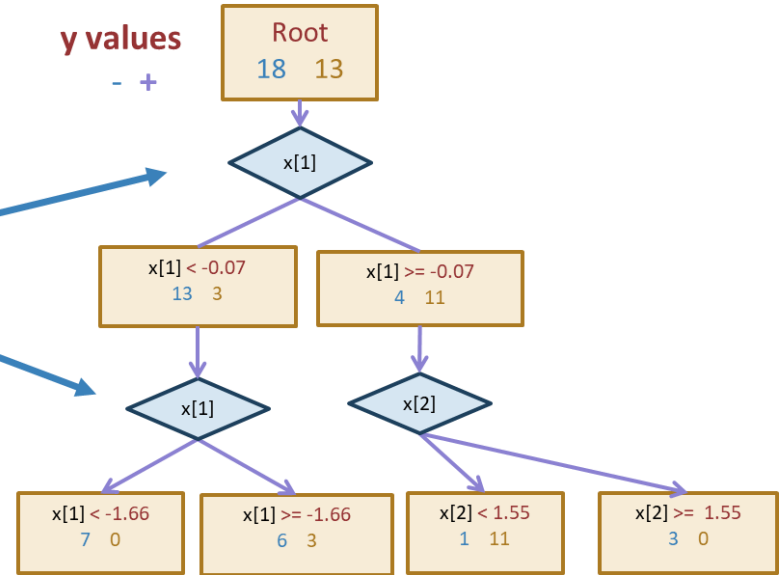


Depth 2



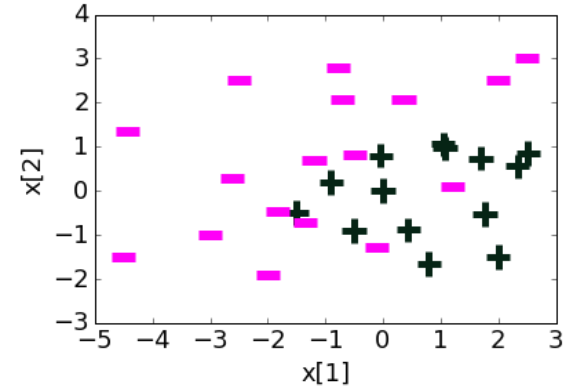
Threshold split caveat

For threshold splits, same feature can be used multiple times

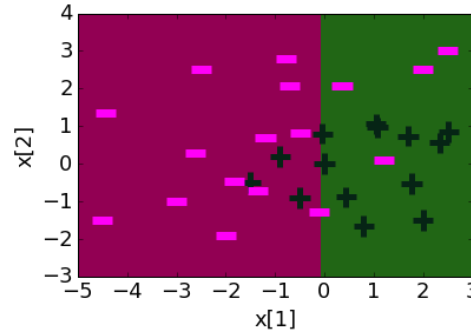


Decision boundaries

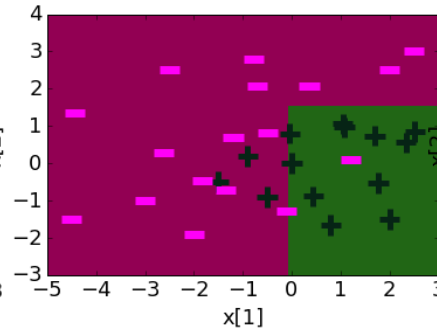
- Decision boundaries can be complex!



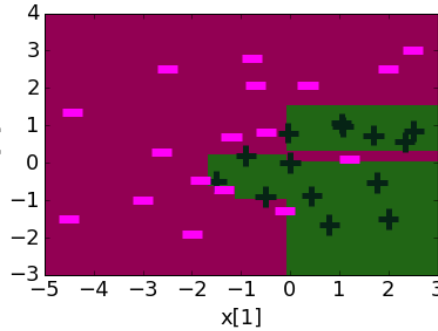
Depth 1



Depth 2



Depth 10



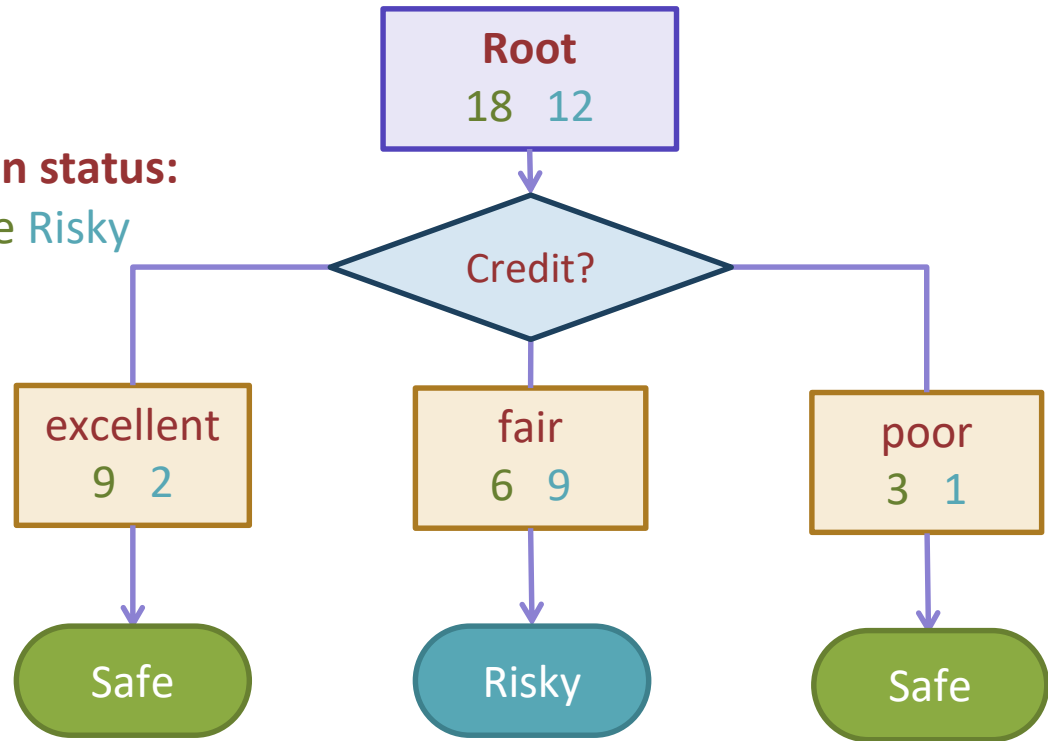
Overfitting

- 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

