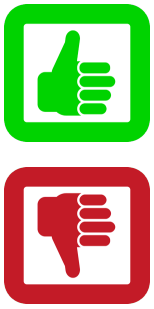


Decision Trees

Emily Fox & Carlos Guestrin

Machine Learning Specialization

University of Washington

Predicting potential loan defaults

What makes a loan risky?

I want a to buy a new house!

Loan Application

Credit ★★★★

Income ★★★

Term ★★★★★

Personal Info ★★★

Did I paid my previous bills on time?

eg: Around $100K

How soon I have to repay the loan?

eg. 2 year / 3 year

age, Marital Status, divorced etc.

Given all this info, we have to predict whether loaning me money is risky / Not

Credit history explained

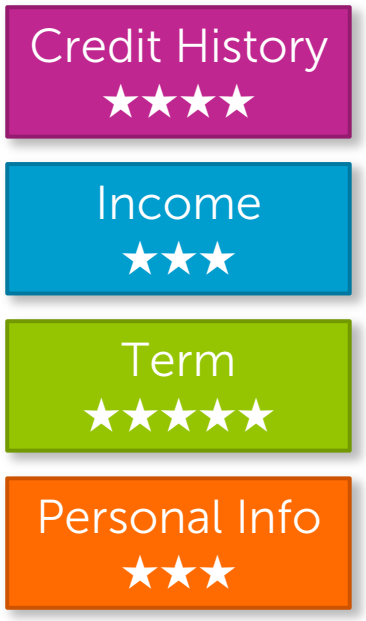
Did I pay previous loans on time?

Example: excellent, good, or fair

Income

What's my income?

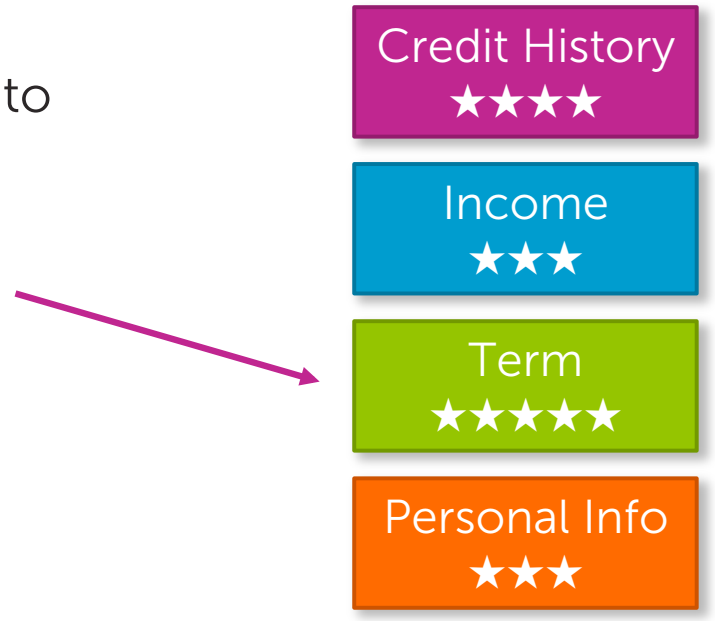
Example:
$80K per year



Loan terms

How soon do I need to pay the loan?

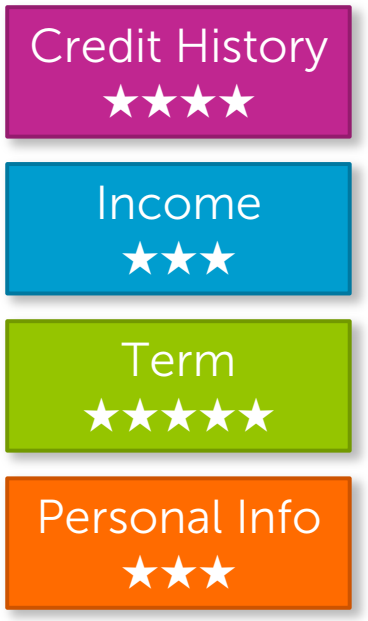
Example: 3 years, 5 years,…



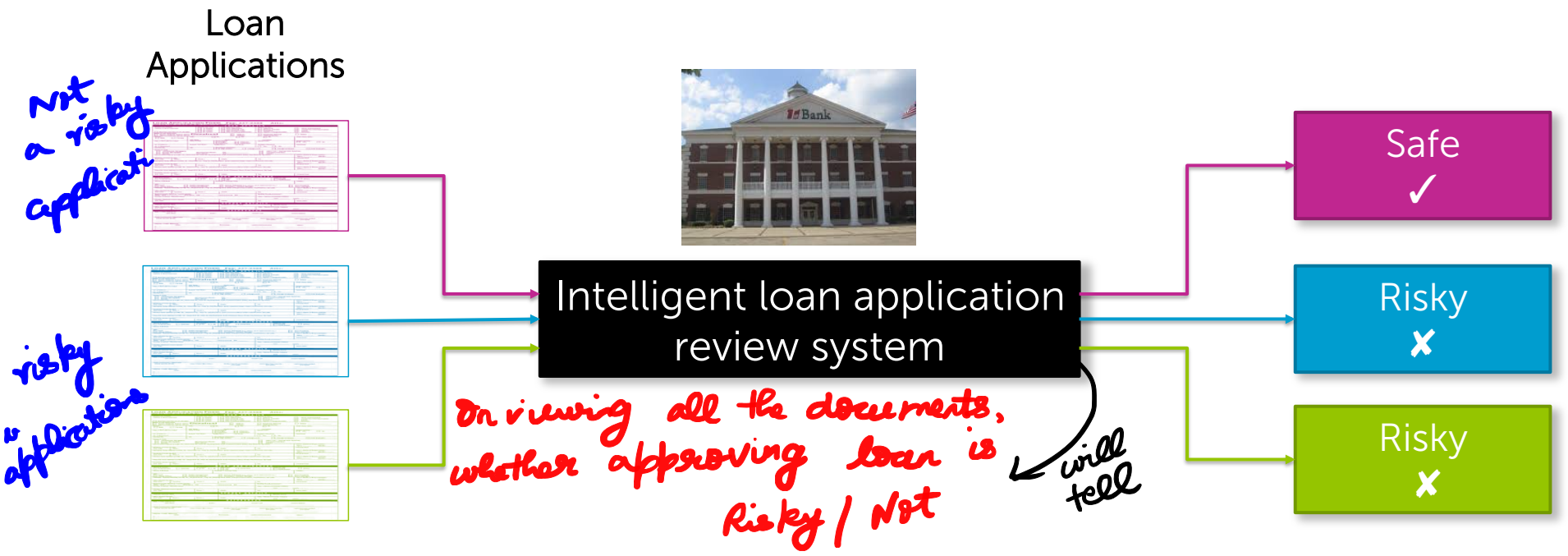
Personal information

Age, reason for the loan, marital status,…

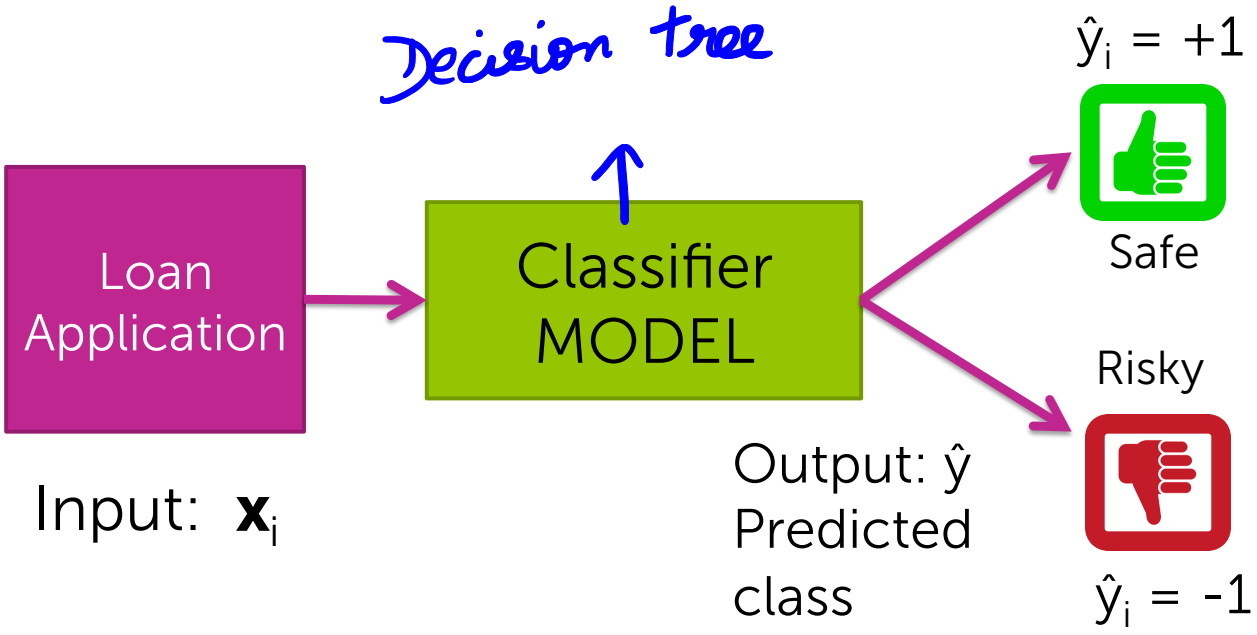
Example: Home loan for a married couple



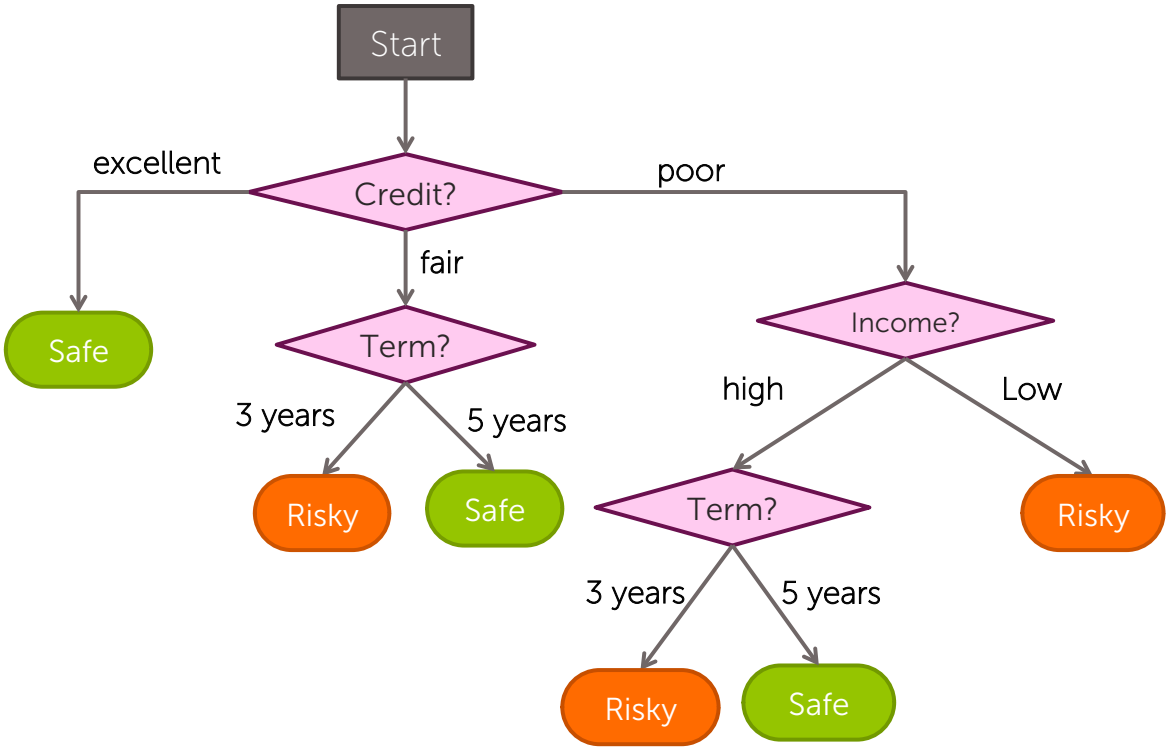
Intelligent application



Classifier review

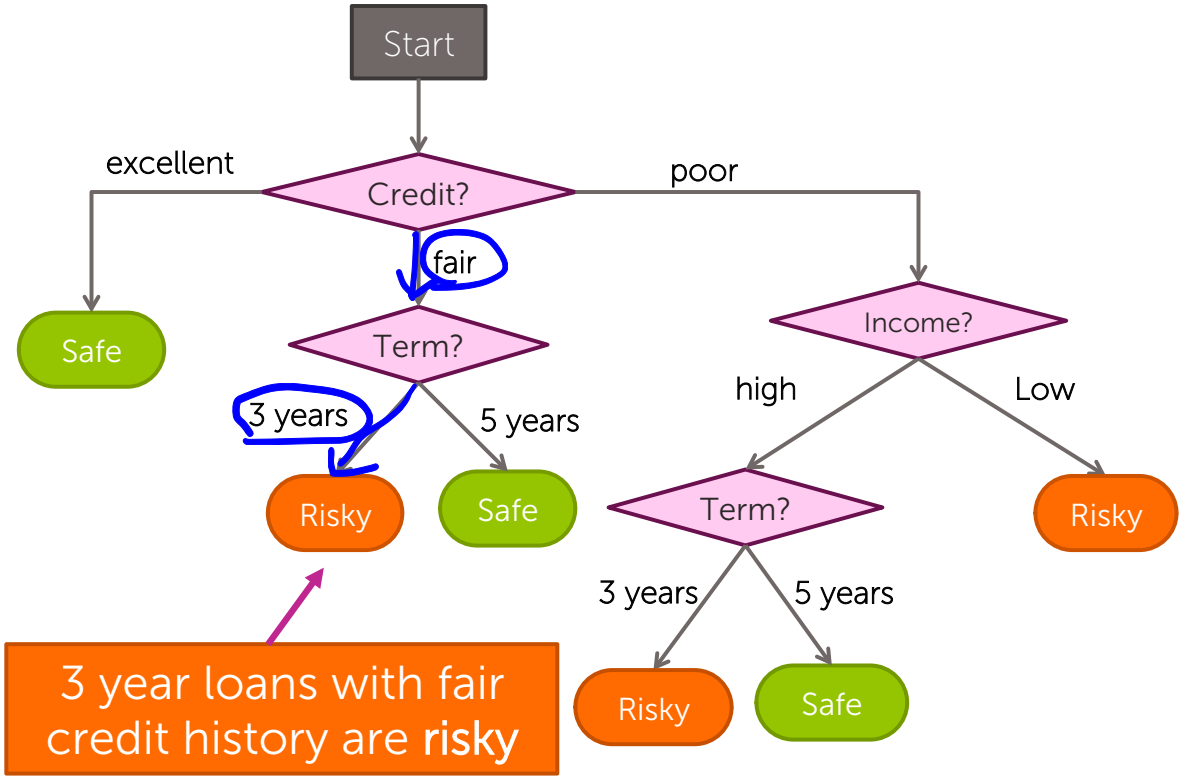


This module … decision trees

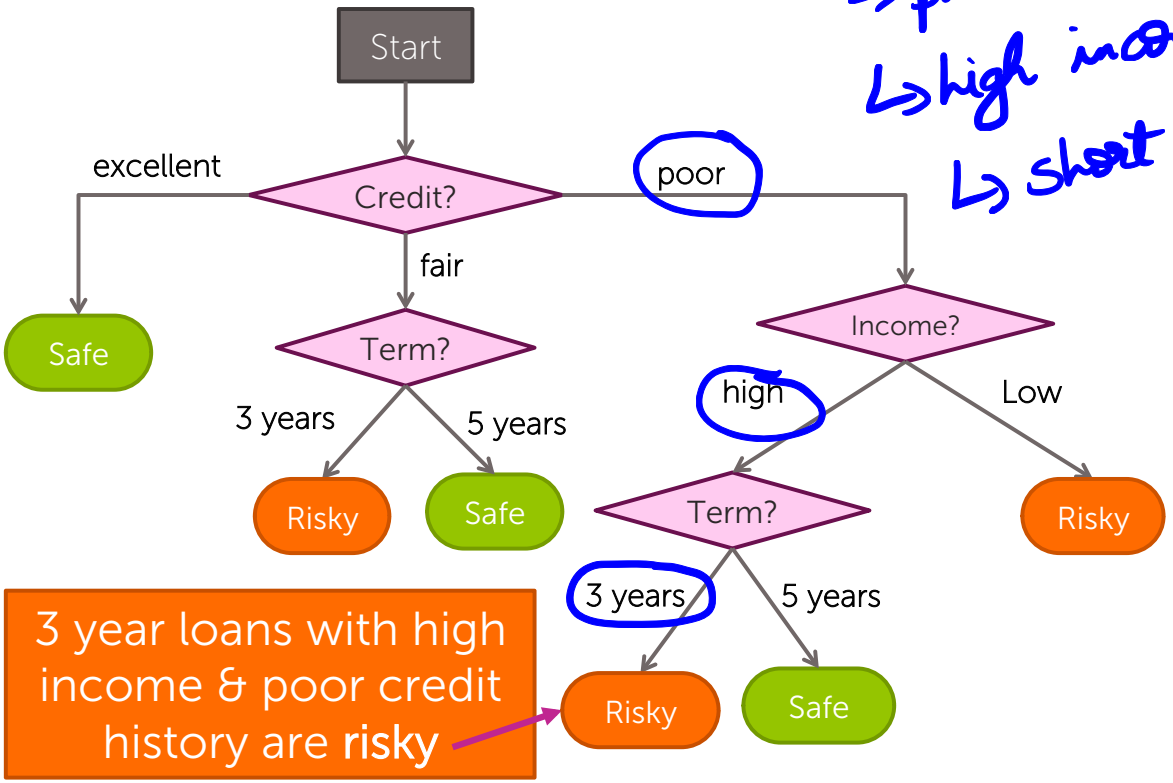


Decision trees: *Intuition*

What does a decision tree represent?



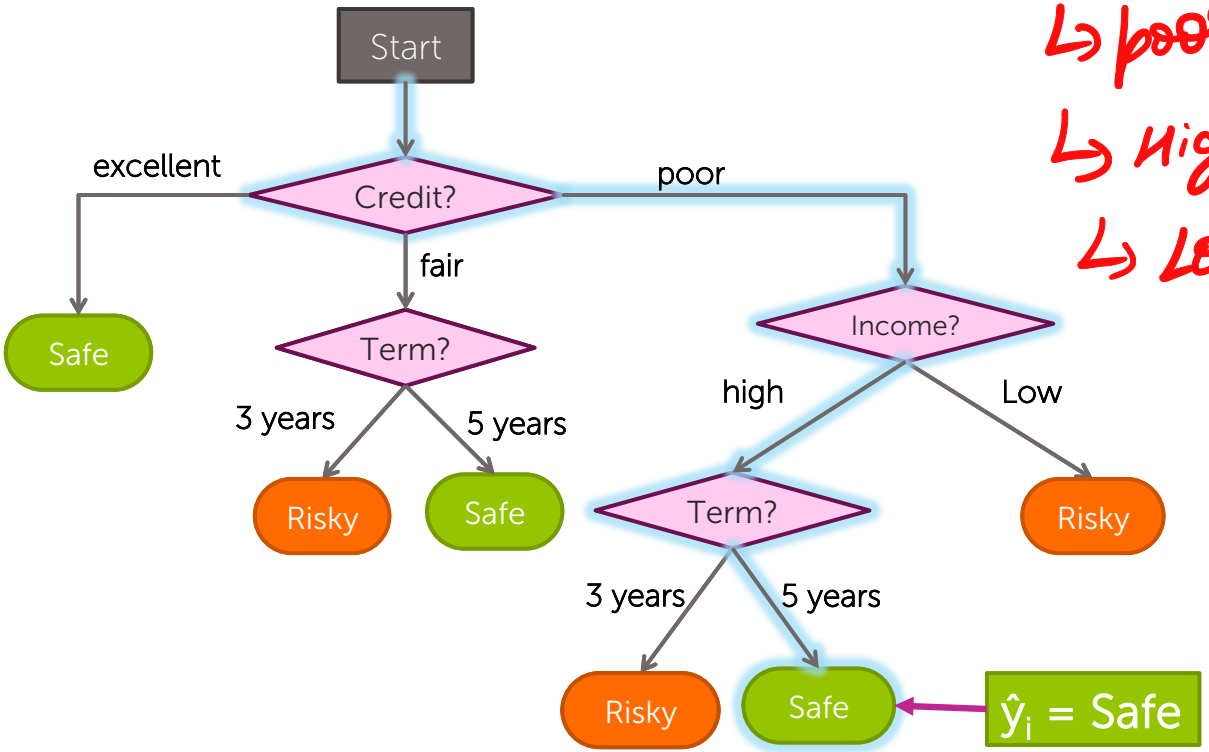
What does a decision tree represent?



Scoring a loan application

$\mathbf{x}_i$ = (Credit = poor, Income = high, Term = 5 years)

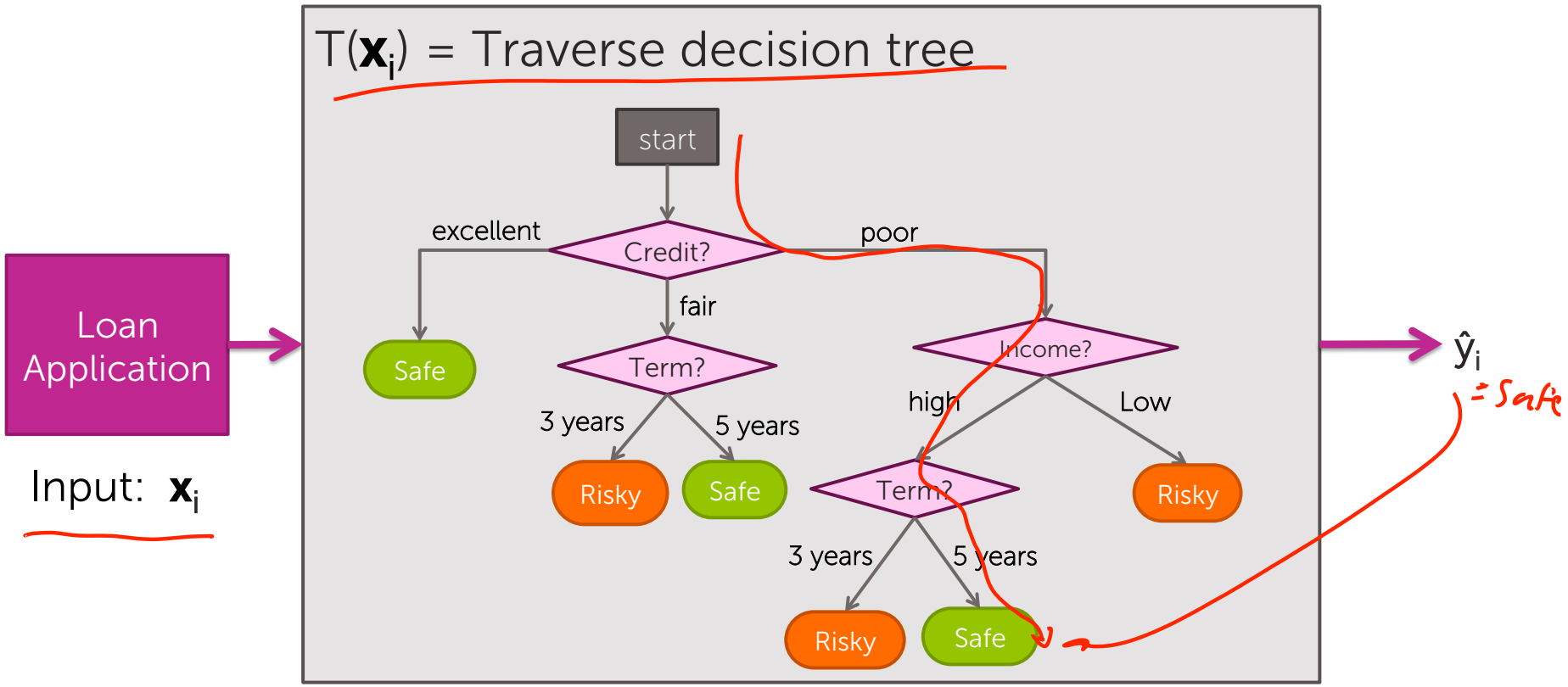
inputs



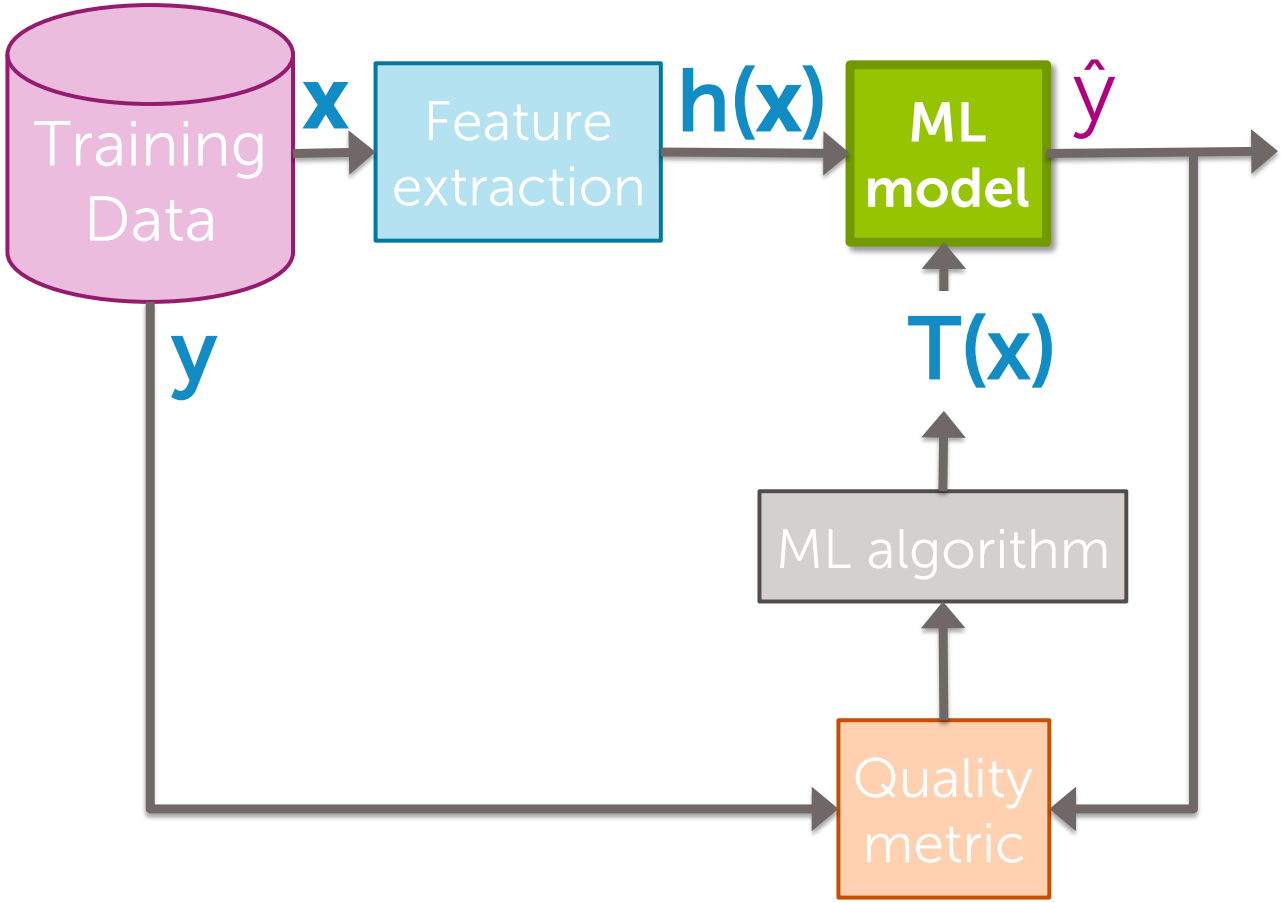
↳ poor Credit
↳ High Income
↳ Long term
} Safe

output

Decision tree model

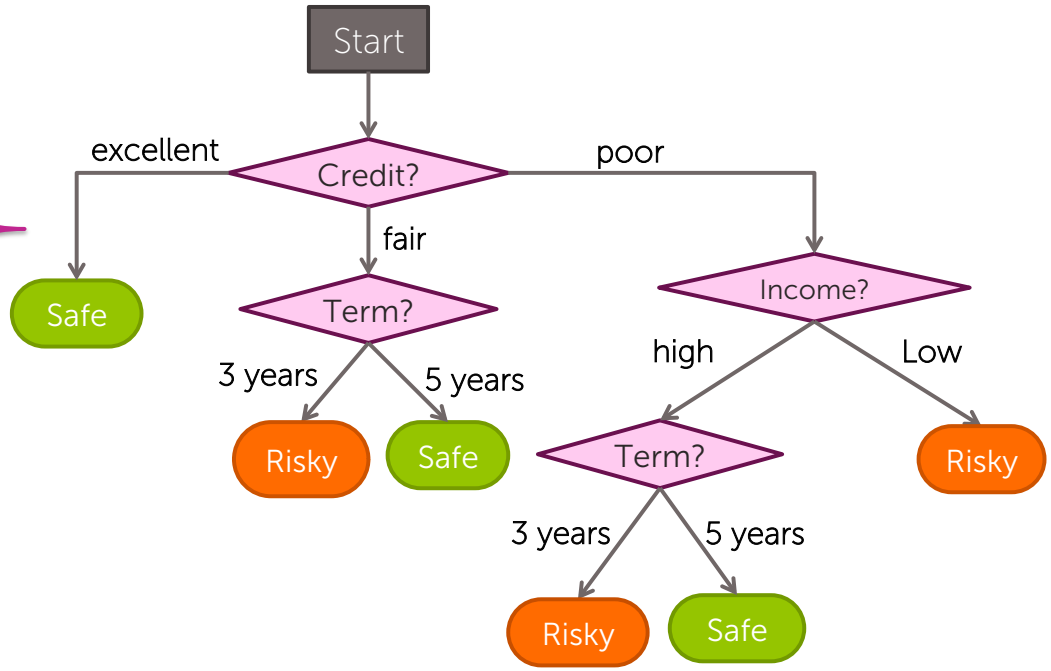


Decision tree learning task



Learn decision tree from data?

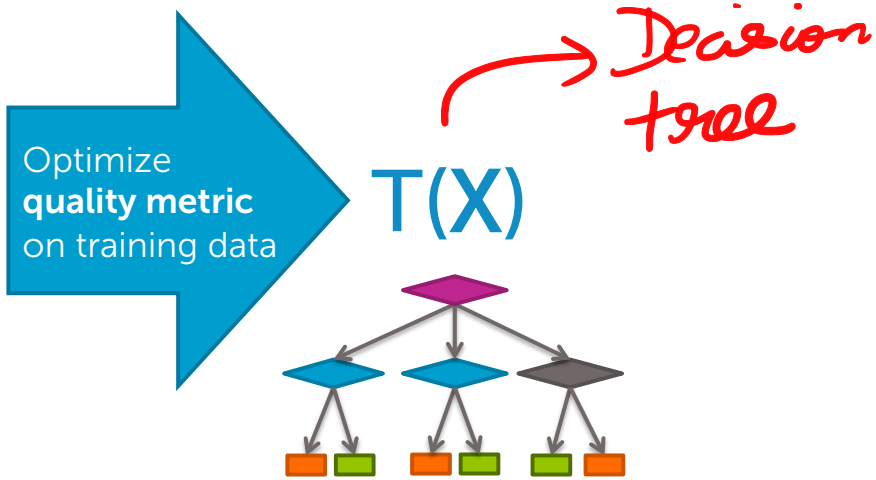
$h_1(x)$ Credit	$h_2(x)$ Term	$h_3(x)$ Income	Loan Status y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	risky
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe



Decision tree learning problem

Training data: N observations ($\mathbf{x}_i, y_i$)

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	risky
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe



Quality metric: Classification error

- Error measures fraction of mistakes

$$\text{Error} = \frac{\text{\# incorrect predictions}}{\text{\# examples}}$$

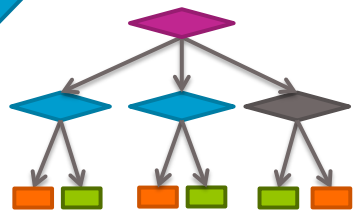
- Best possible value : 0.0
- Worst possible value: 1.0

Find the tree with lowest classification error

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	risky
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe

Minimize **classification error** on training data

T(X)

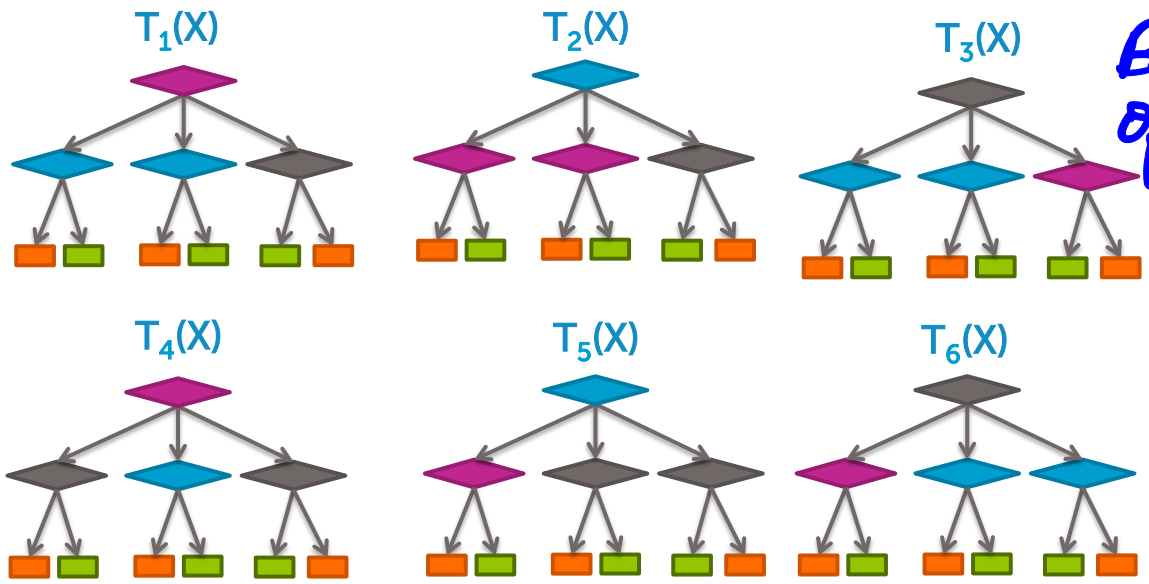


How do we find the best tree?

Exponentially large number of possible trees makes decision tree learning hard! *(NP-hard problem)*

NP-hard problem

↳ No approximation algorithm



But there are some idea of heuristics that tends to perform extremely well

idea → Greedy Algorithm

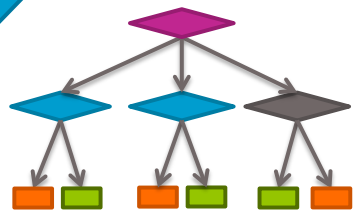
Simple (greedy) algorithm finds "good" tree

↳ Going to incrementally build a tree one layer at a time, in order to get best possible classification error at each step.

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	risky
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe

Approximately minimize **classification error** on training data

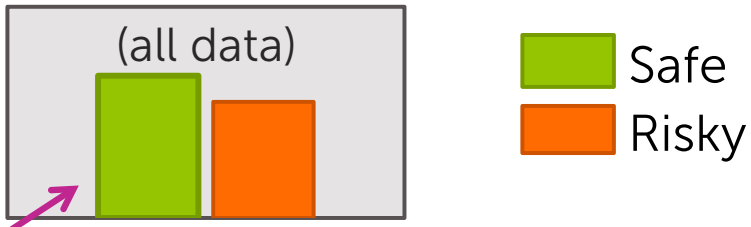
T(X)



Greedy decision tree learning:
Algorithm outline

Step 1: Start with an empty tree

root Node
↳ Contains the entire data set



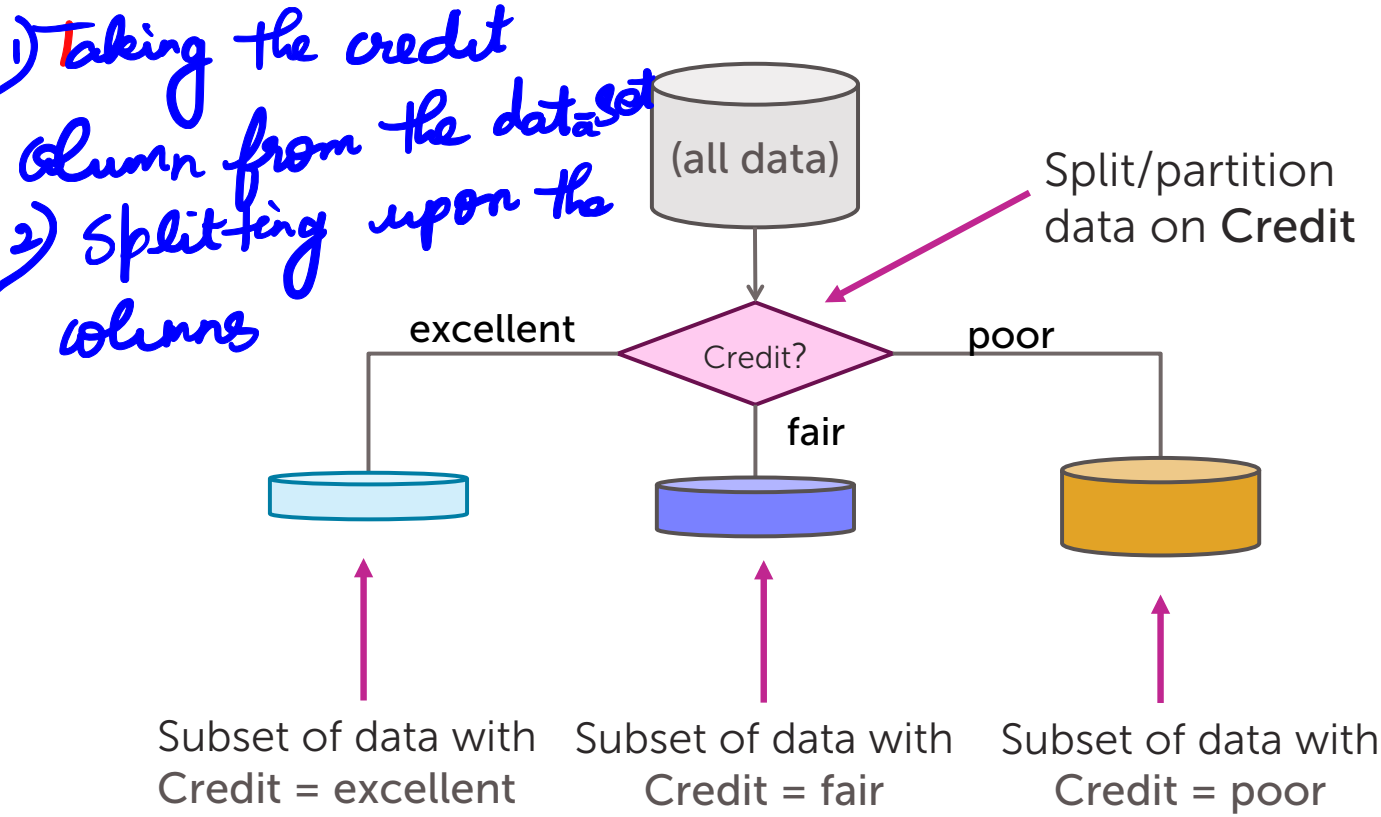
Safe
Risky

Histogram of y values

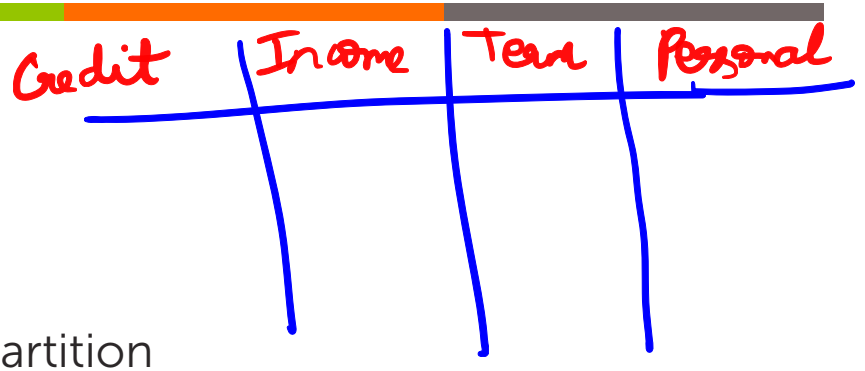
(Histogram plots)

All points in the dataset

Step 2: Split on a feature

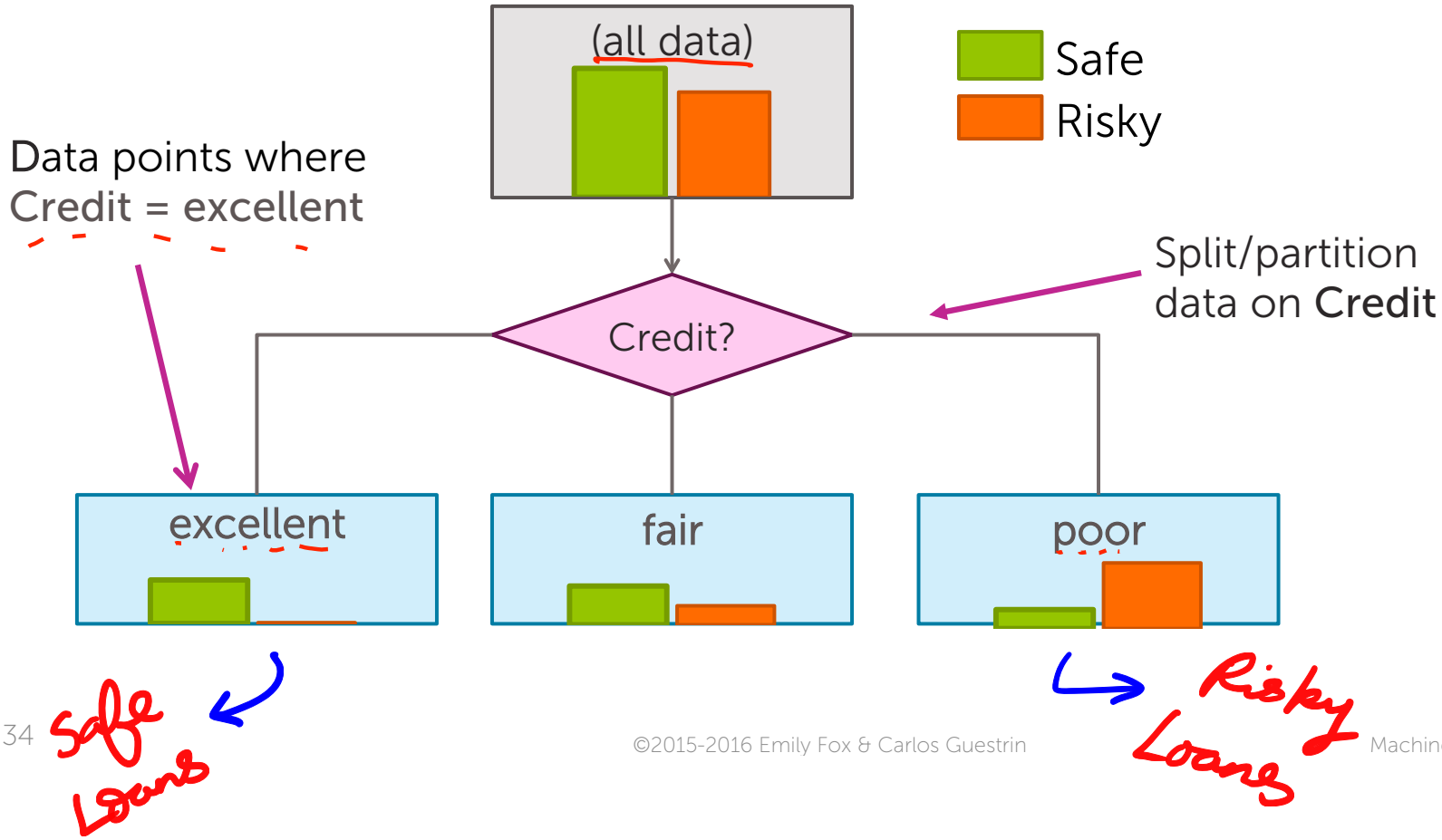


1) Taking the credit column from the dataset
2) Splitting upon the values

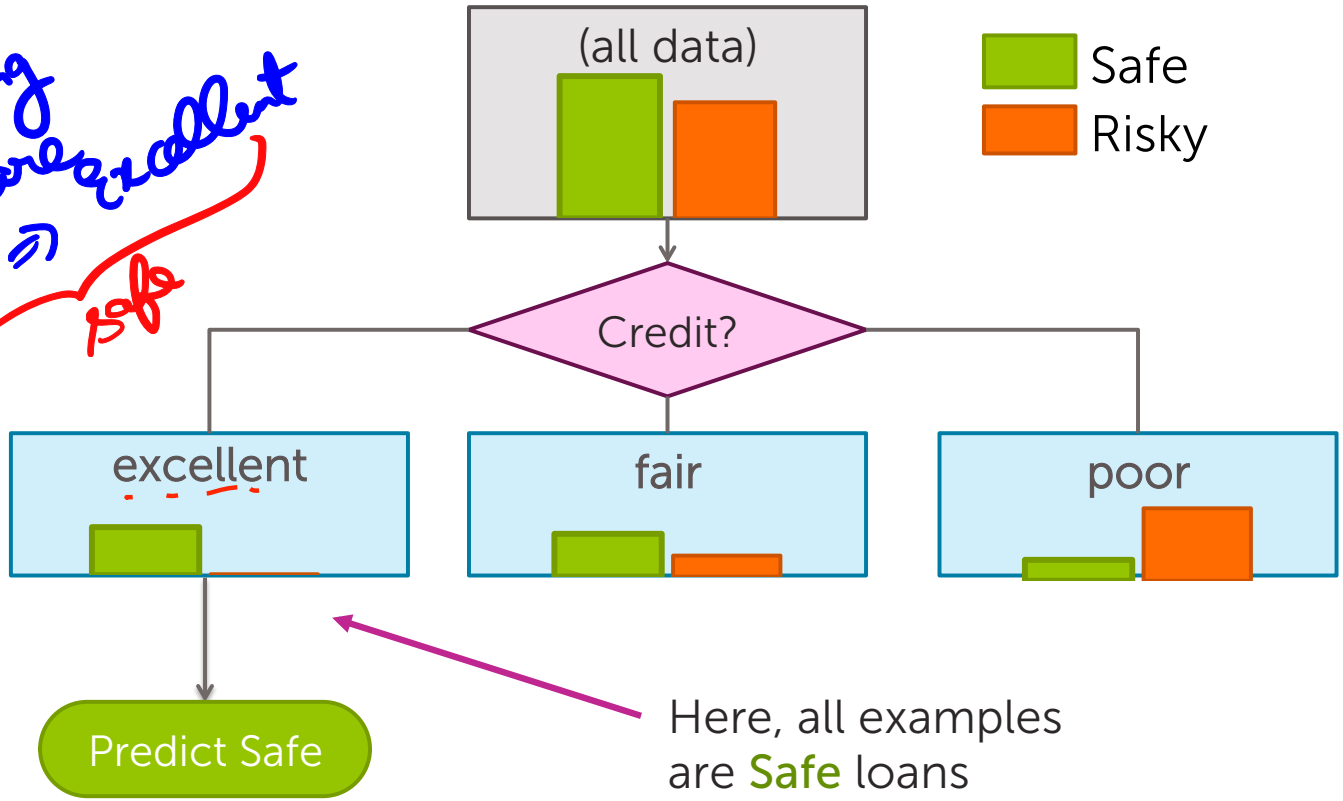


Credit
↳ excellent
↳ fair
↳ poor

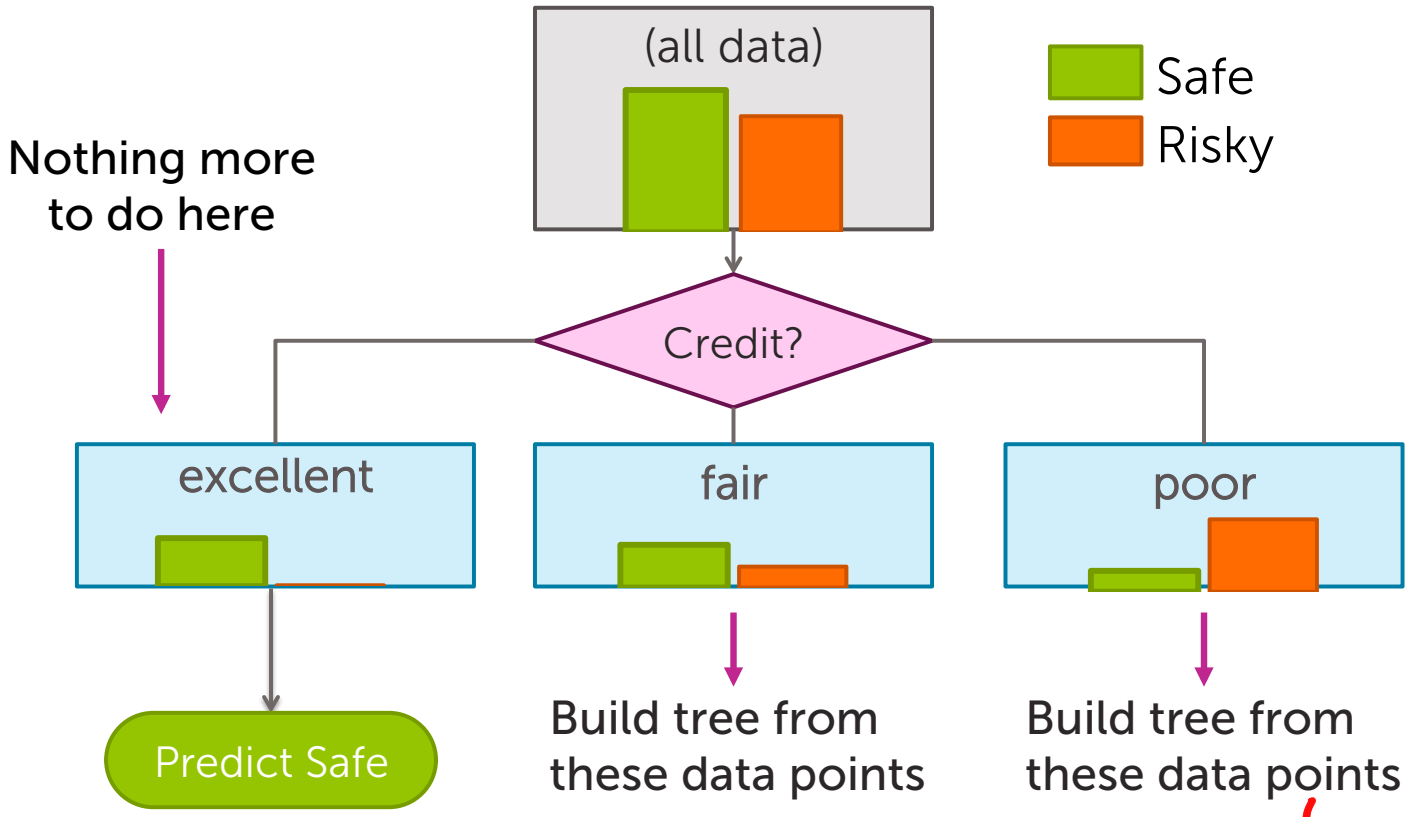
Feature split explained



Step 3: Making predictions

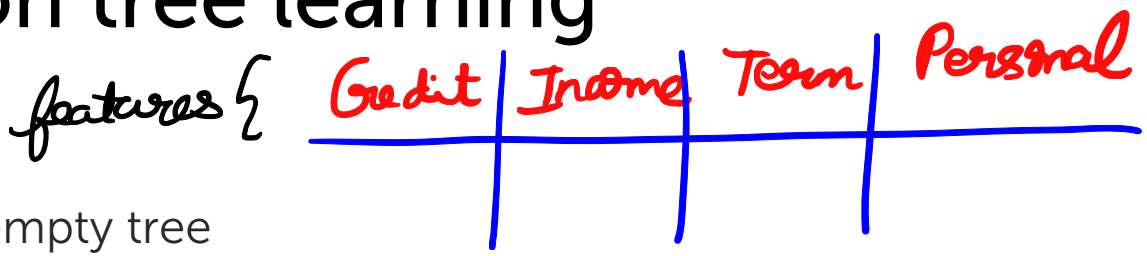


Step 4: Recursion



Should go recursively ⇒ Recursive Greedy Algorithm

Greedy decision tree learning

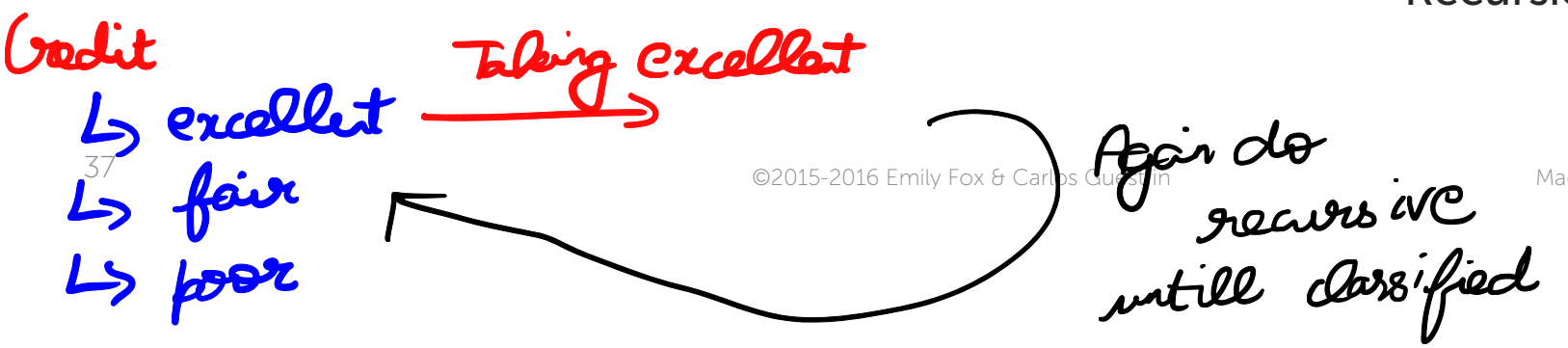


- Step 1: Start with an empty tree
- Step 2: Select a feature to split data
- For each split of the tree:
  - Step 3: If nothing more to, make predictions
  - Step 4: Otherwise, go to Step 2 & continue (recurse) on this split

Problem 1: Feature split selection

Problem 2: Stopping condition

Recursion



Feature split learning

=

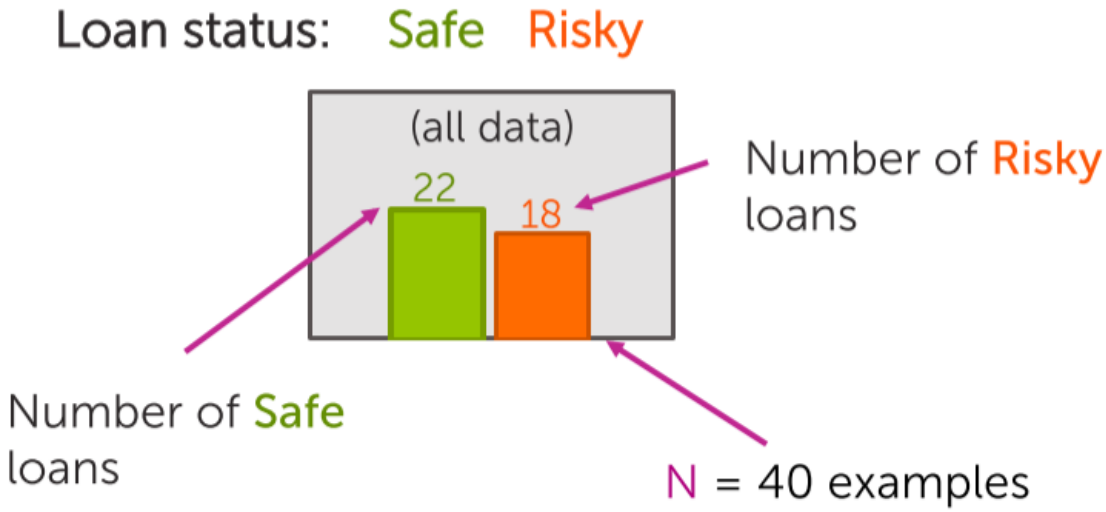
Decision stump learning

↳ One level decision tree

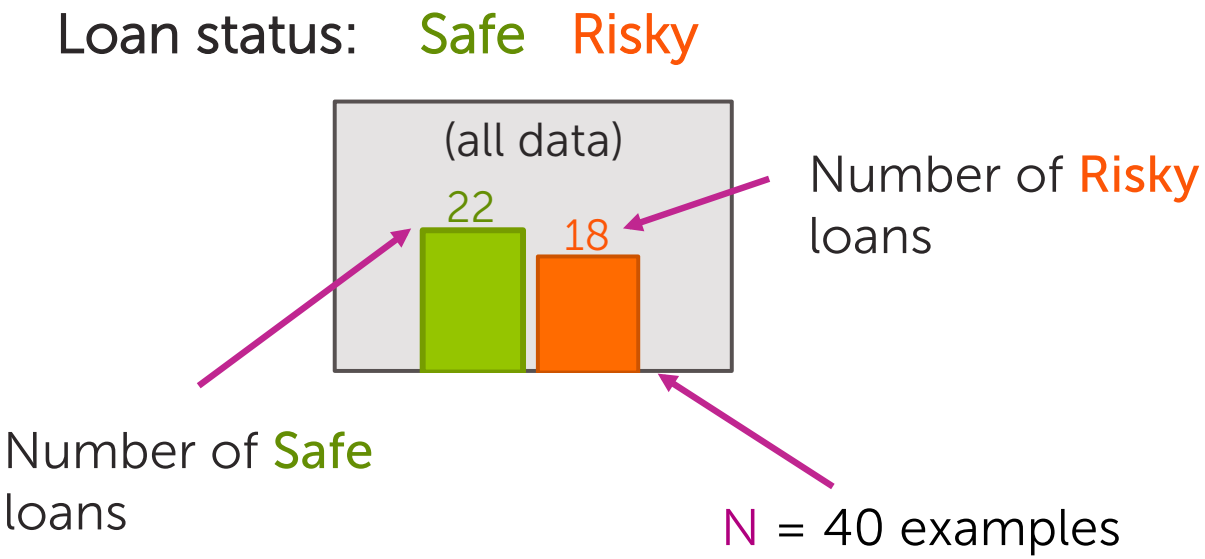
Start with the data

Assume N = 40, 3 features

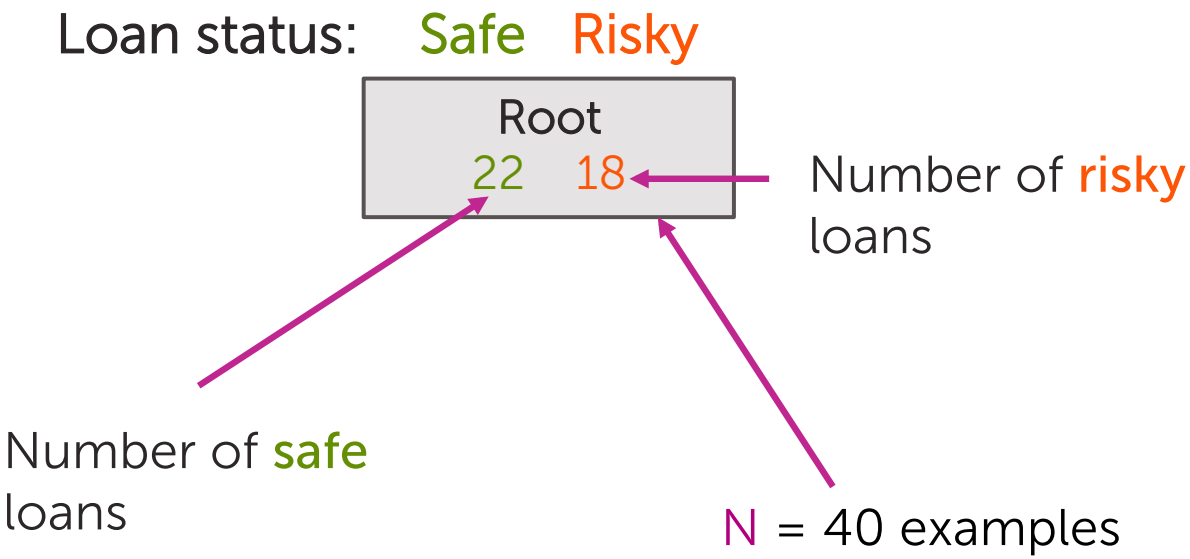
Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	risky
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe



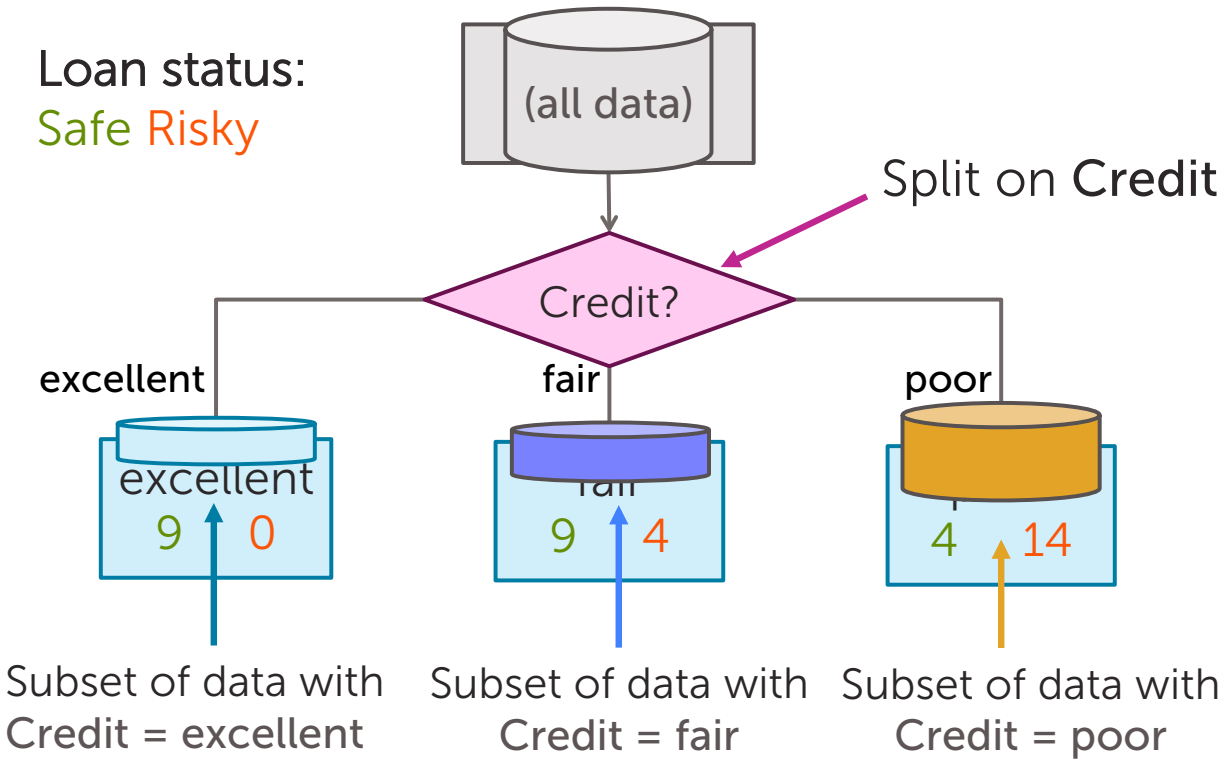
Start with all the data



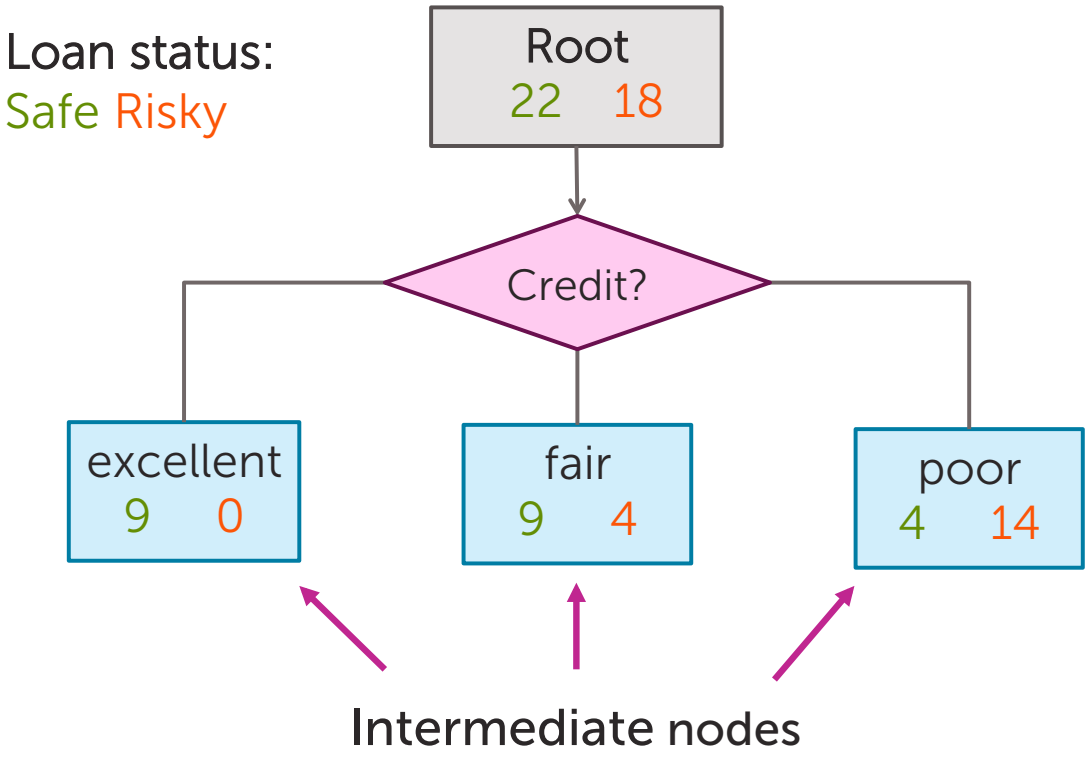
Compact visual notation: Root node



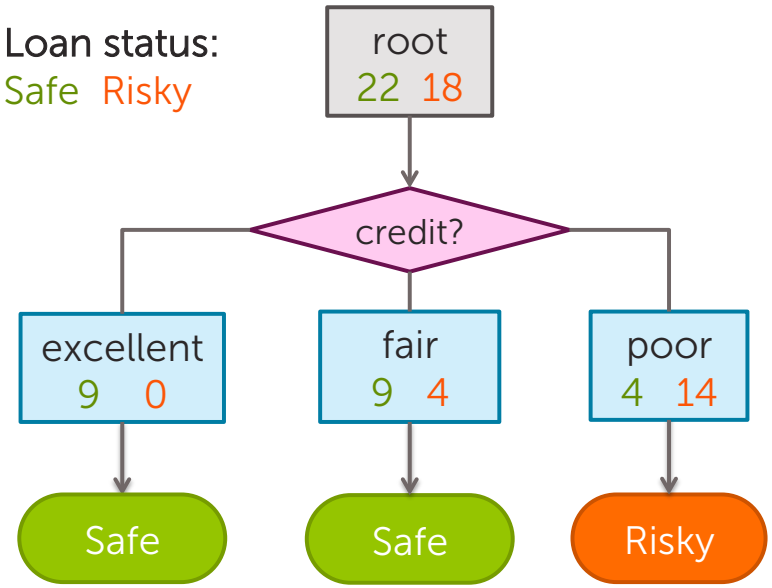
Decision stump: Single level tree



Visual Notation: Intermediate nodes



Making predictions with a decision stump



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

↳ For each Node, we are looking into Majority value for the prediction

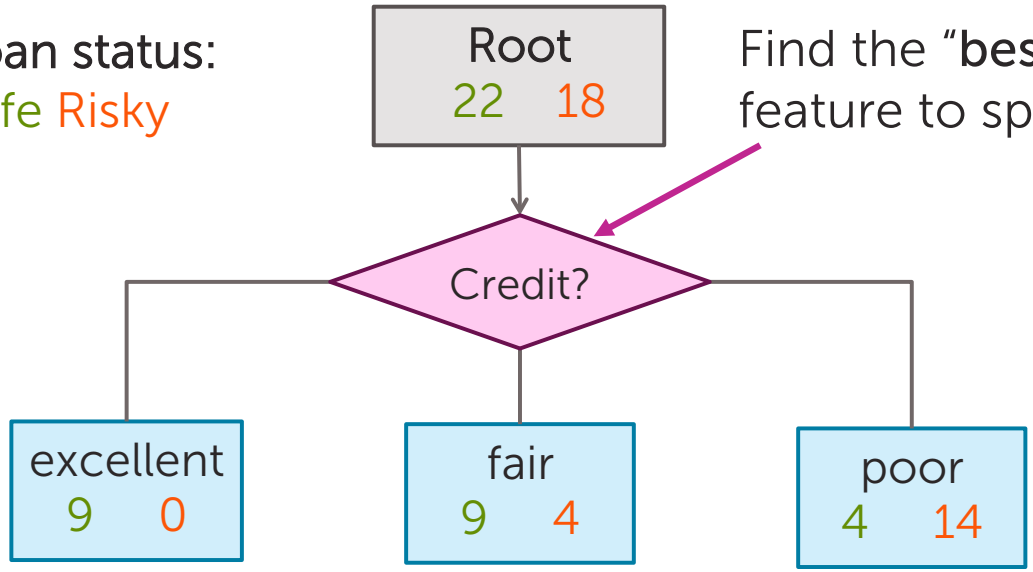
↳ 1st Decision S

Selecting best feature to split on

To get better predictions
↳ Going to split further
↳ Before split, we are going to discuss why we took CREDIT.?

How do we learn a decision stump?

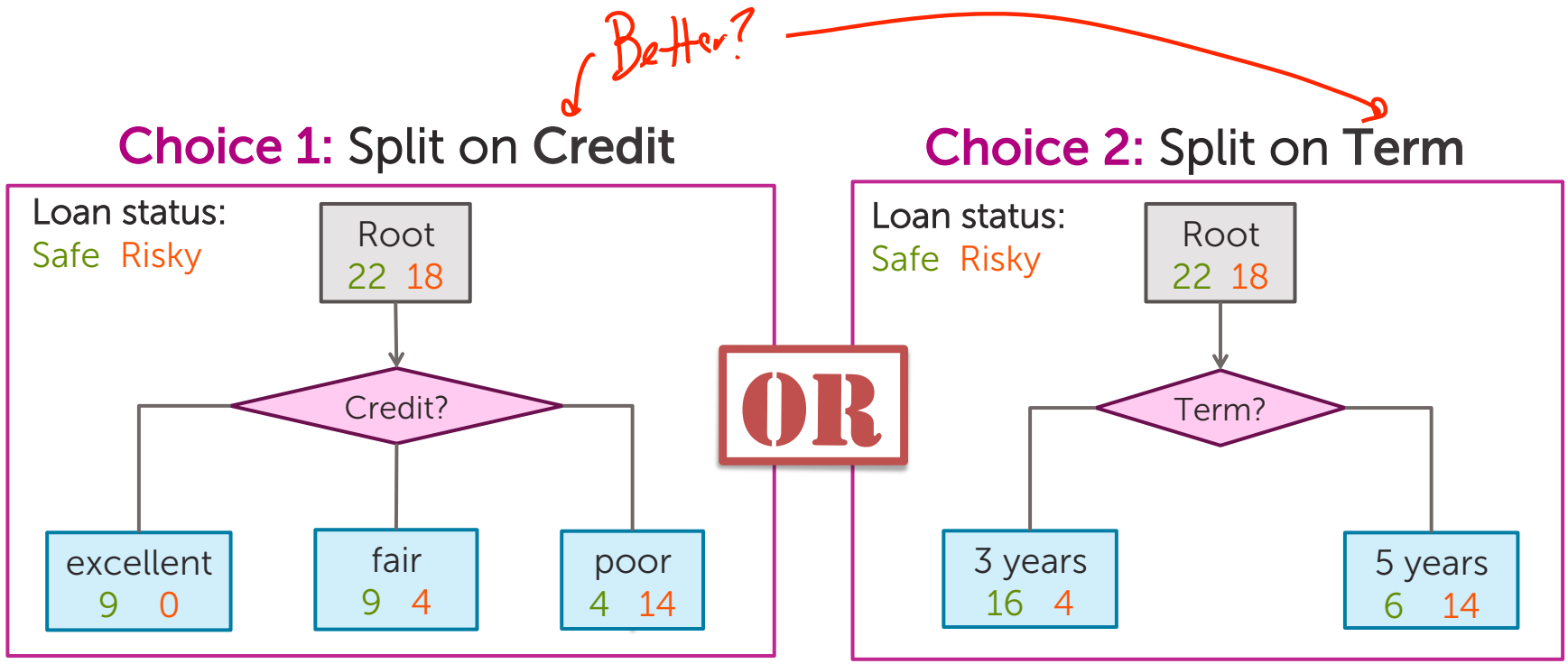
Loan status:
Safe Risky



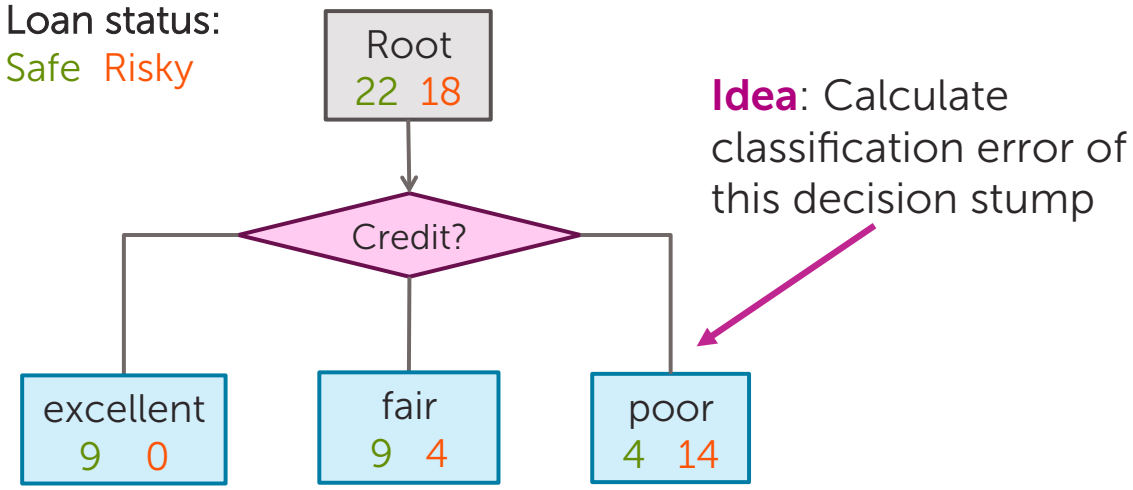
Find the "**best**" feature to split on!

How do we select the best feature?

Better split → Gives lowest classification error.



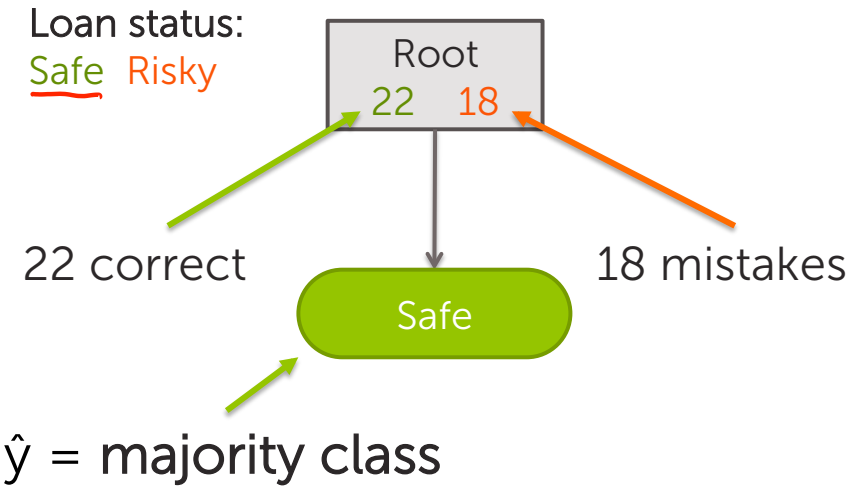
How do we measure effectiveness of a split?



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

Calculating classification error

- **Step 1:** ŷ = class of majority of data in node
- **Step 2:** Calculate classification error of predicting ŷ for this data



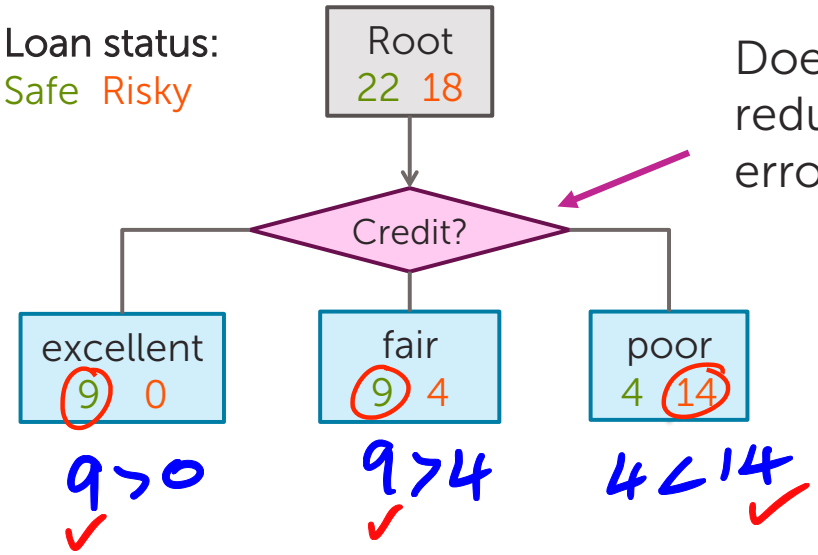
$$\text{Error} = \frac{18}{22+18} = 0.45$$

Tree	Classification error
(root)	0.45

How good is the decision stump that splits on credit?
And how does it compare to not split on anything, which has a classification error of 0.45

Choice 1: Split on **credit history**?

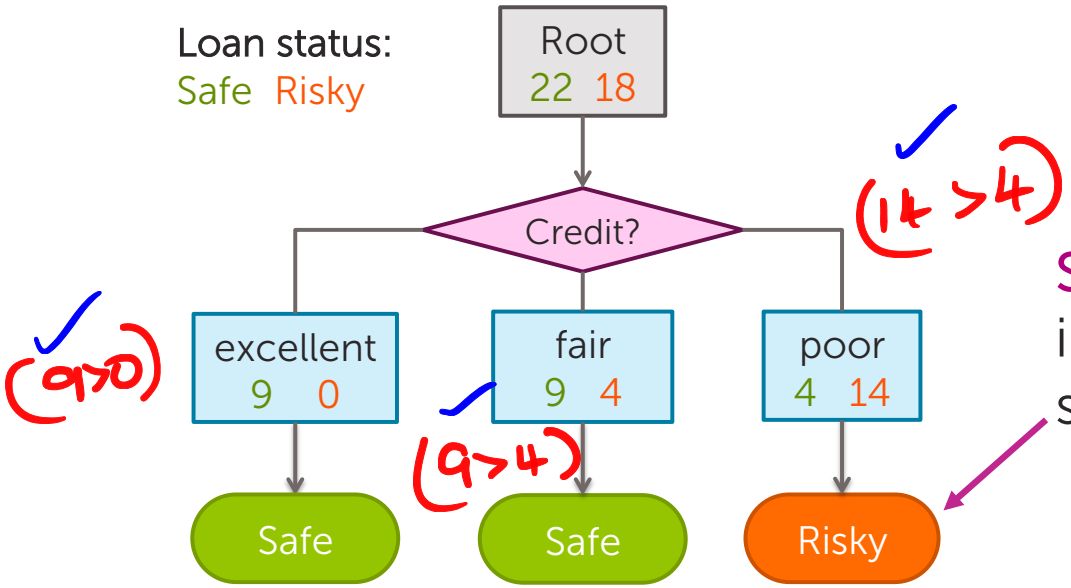
Choice 1: Split on Credit



Does a split on Credit reduce classification error below 0.45?

How good is the split on **Credit**?

Choice 1: Split on **Credit**

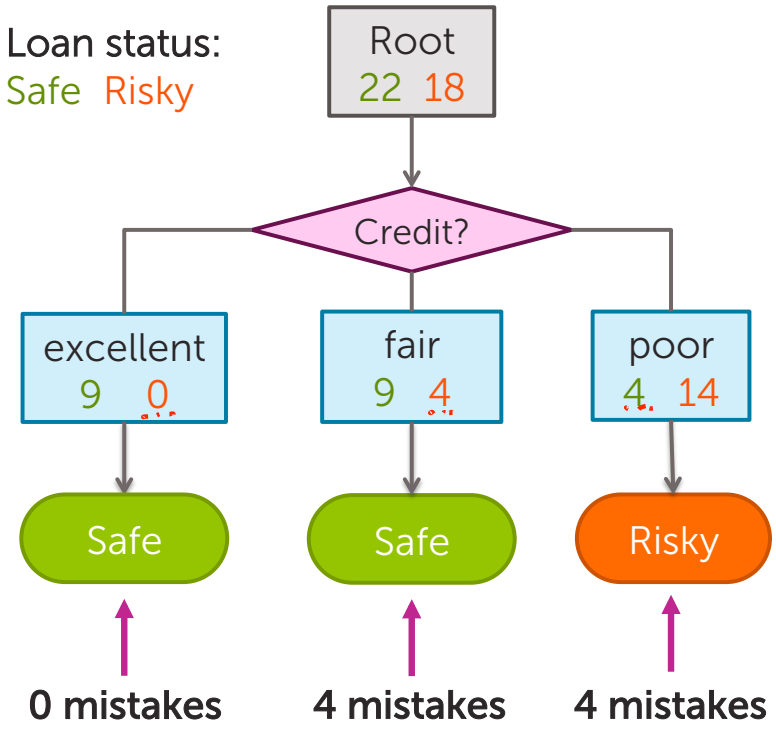


Step 1: For each intermediate node, set $\hat{y}$ = **majority value**

Majority plays a Major Role

Split on **Credit**: Classification error

Choice 1: Split on Credit



Error = $\frac{4+4}{40}$

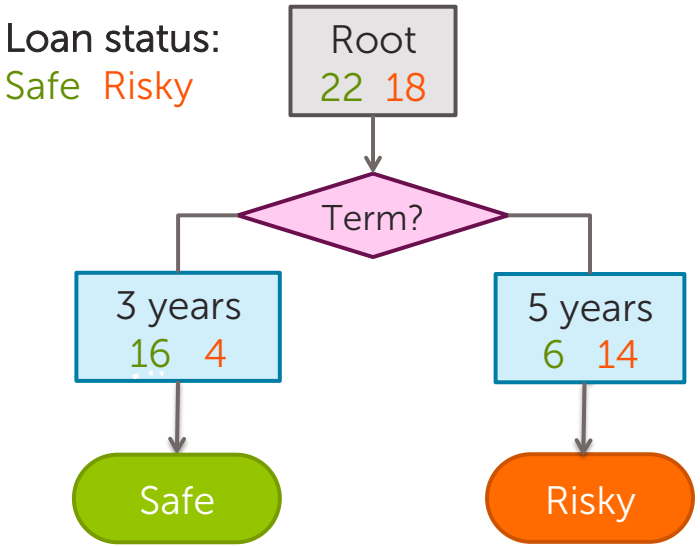
= 0.20

Tree	Classification error
(root)	0.45
Split on credit	0.2

$$\frac{0 + 4 + 4}{9 + 13 + 18} \Rightarrow \frac{8}{40} \rightarrow 0.20$$

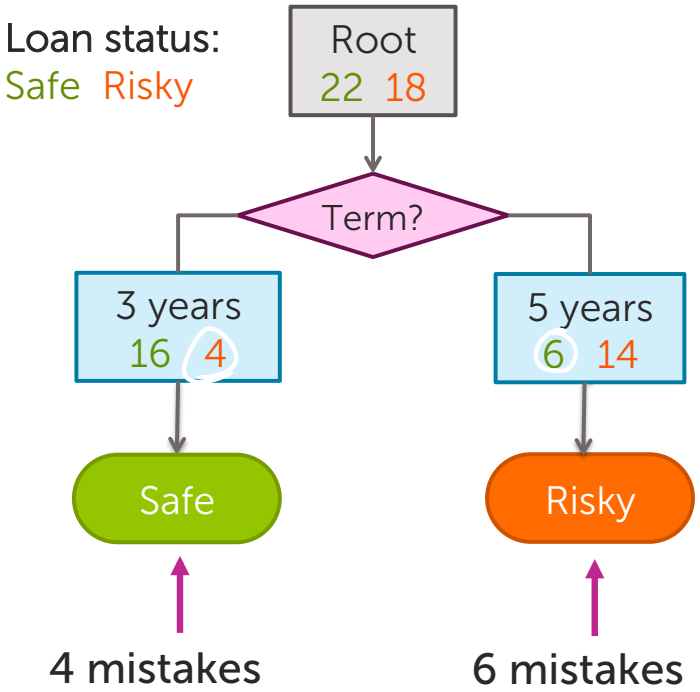
Choice 2: Split on **Term**?

Choice 2: Split on **Term**



Choice 2: Split on Term?

Choice 2: Split on Term



$$\text{Error} = \frac{4+6}{40} = 0.25$$

Tree	Classification error
(root)	0.45
Split on credit	0.2
Split on term	0.25

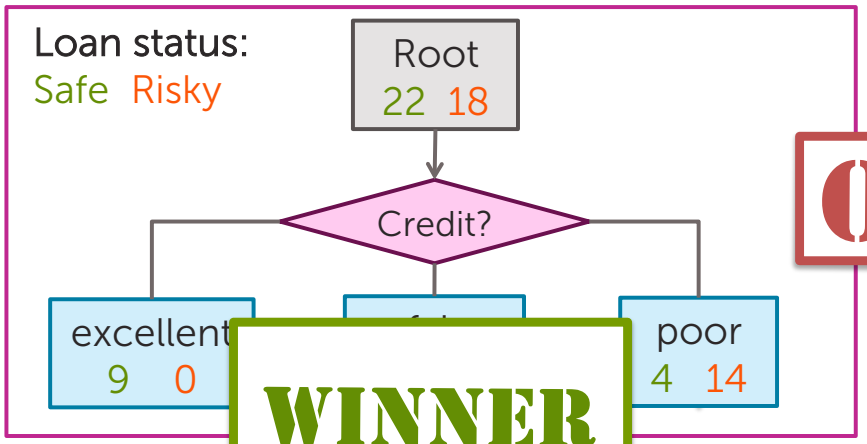
$\frac{4+6}{20+20} \Rightarrow 10/40 \Rightarrow 0.25$

Choice 1 vs Choice 2

Tree	Classification error
(root)	0.45
split on **credit**	0.2
split on **loan term**	0.25

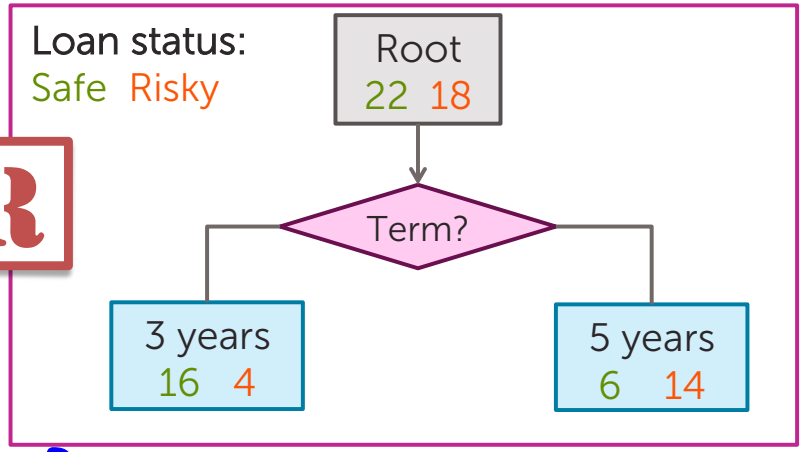
First split!

Choice 1: Split on **Credit**



OR

Choice 2: Split on **Term**



WINNER

winner

loser

In Greedy Algorithm perspective

Feature split selection algorithm

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

credit, term, income

credit

Credit is the winner

Greedy decision tree learning

- Step 1: Start with an empty tree
- Step 2: Select a feature to split data
- For each split of the tree:
  - Step 3: If nothing more to, make predictions
  - Step 4: Otherwise, go to Step 2 & continue (recurse) on this split

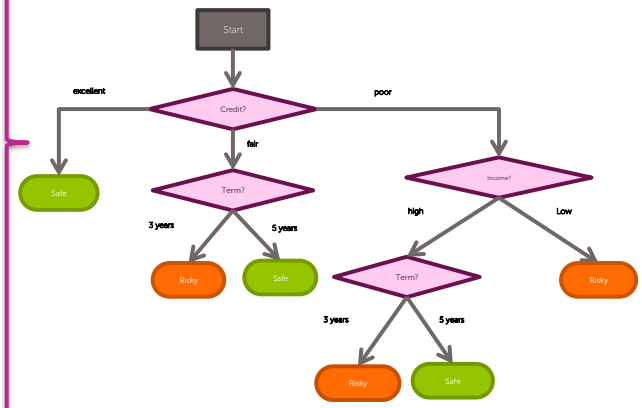
Among the three features we picked up the Credit feature

Pick feature split leading to lowest classification error

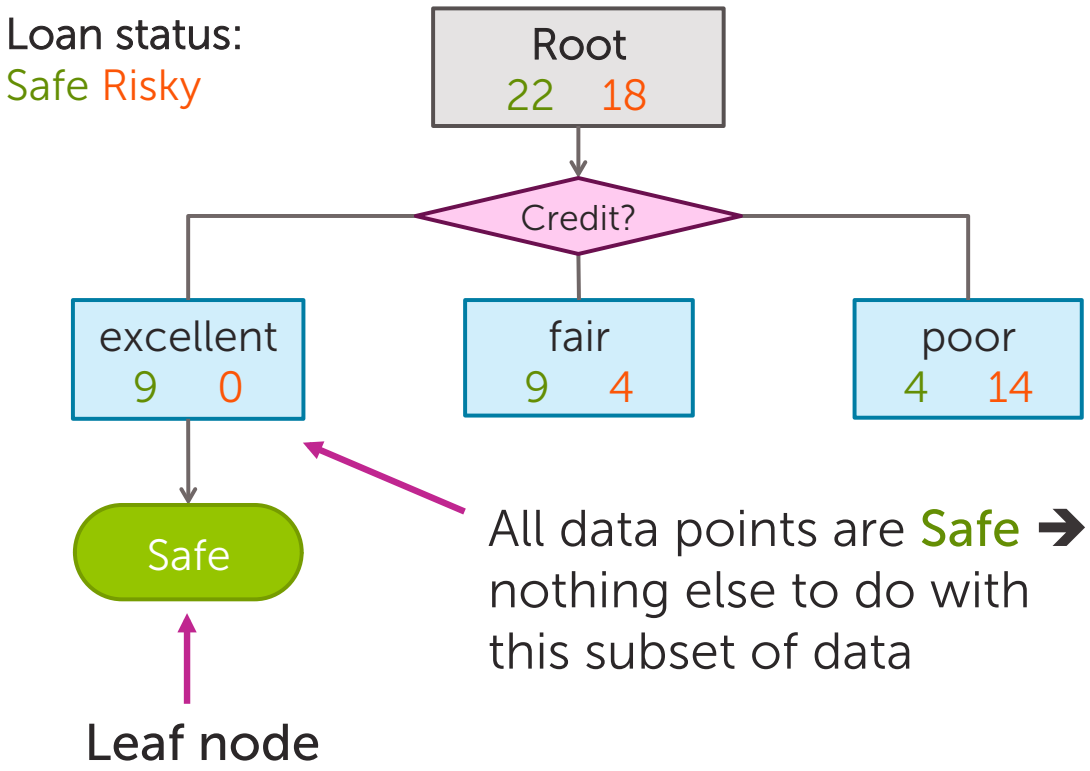
Decision Tree Learning:
Recursion & Stopping conditions

Learn decision tree from data?

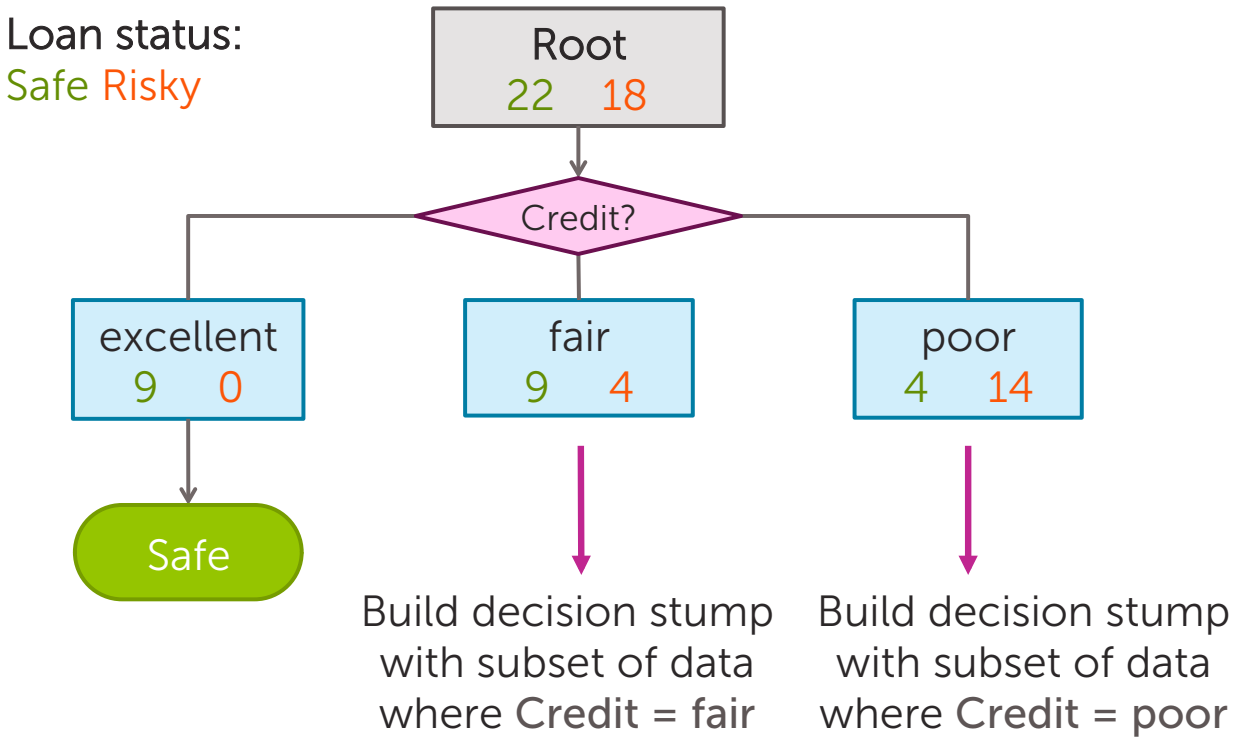
Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	risky
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe



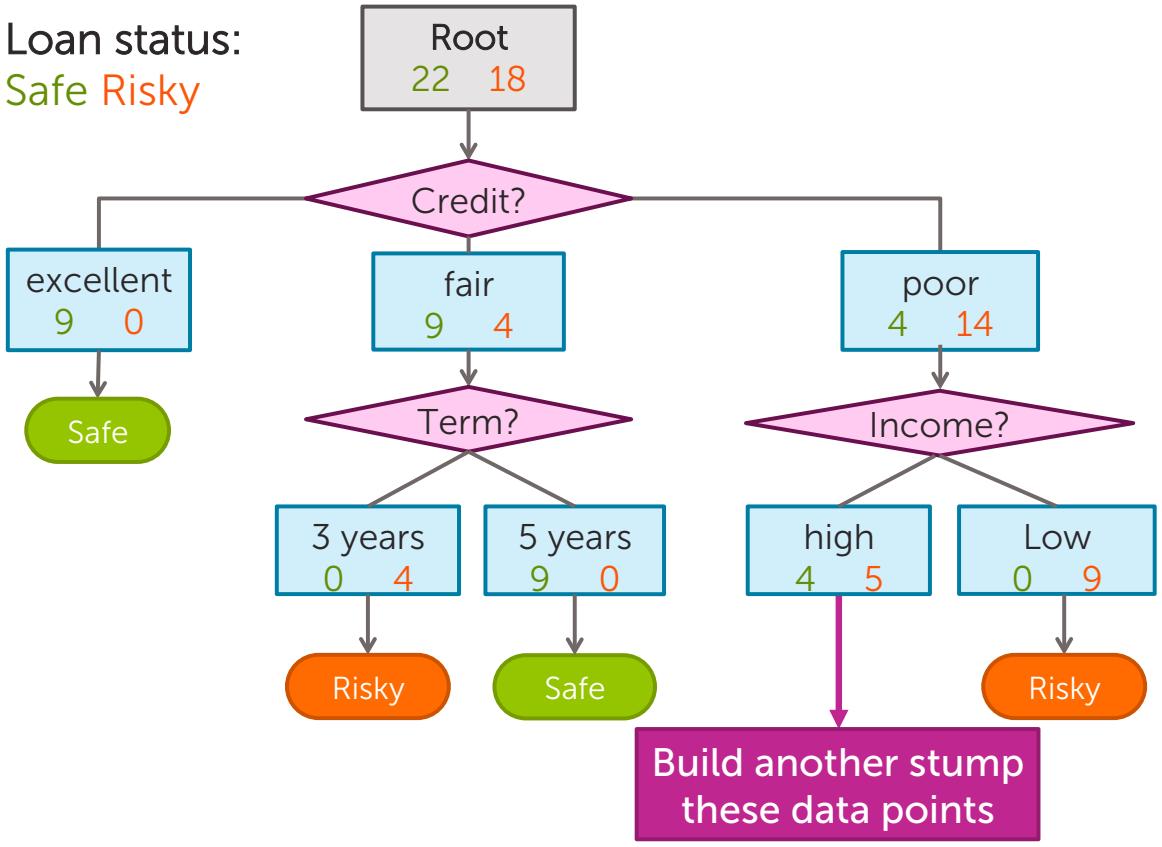
We've learned a decision stump, what next?



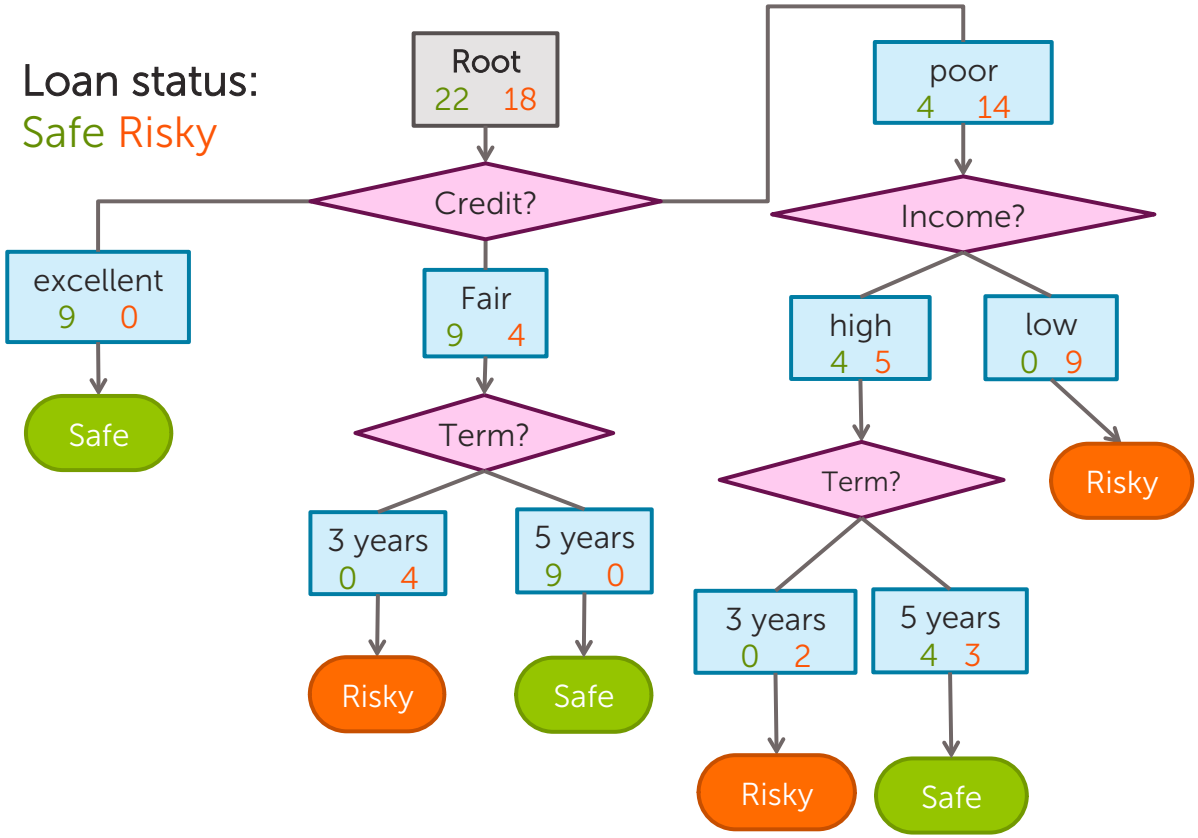
Tree learning = Recursive stump learning



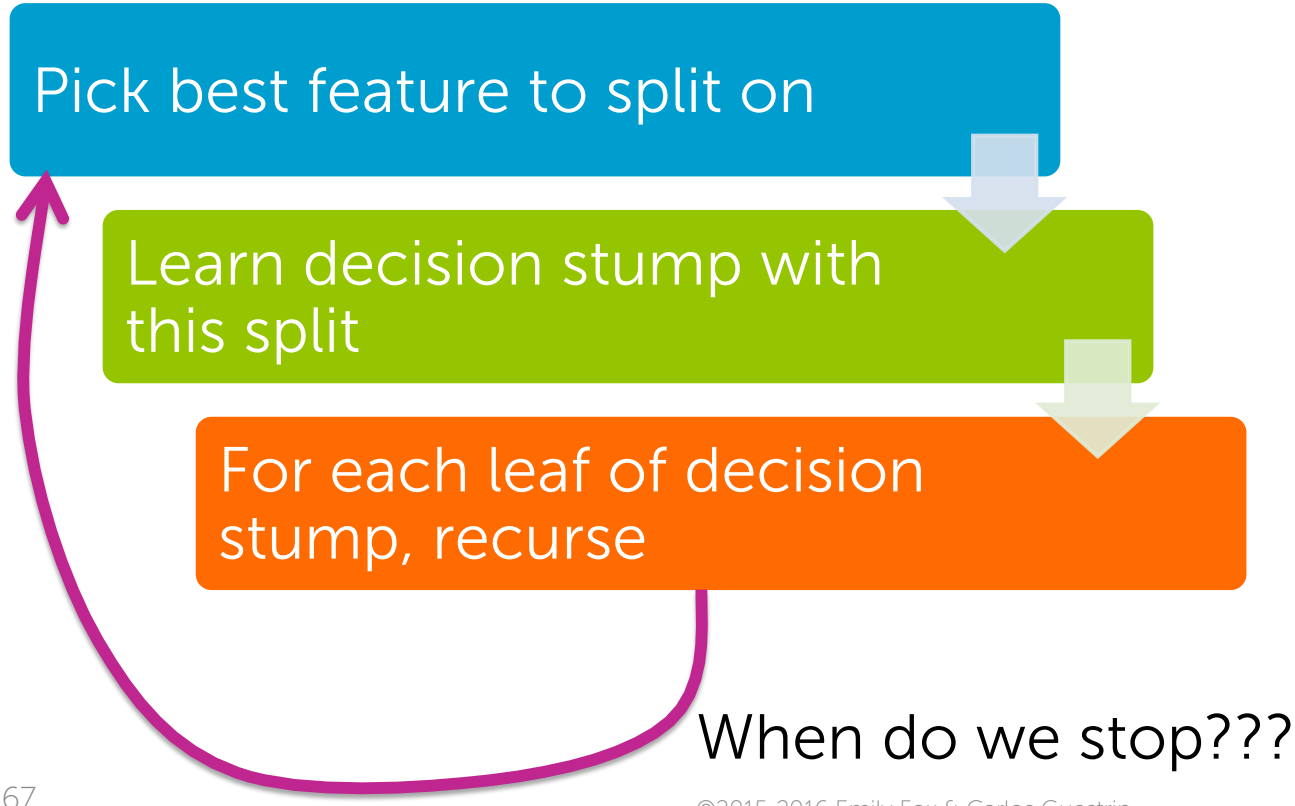
Second level



Final decision tree



Simple greedy decision tree learning

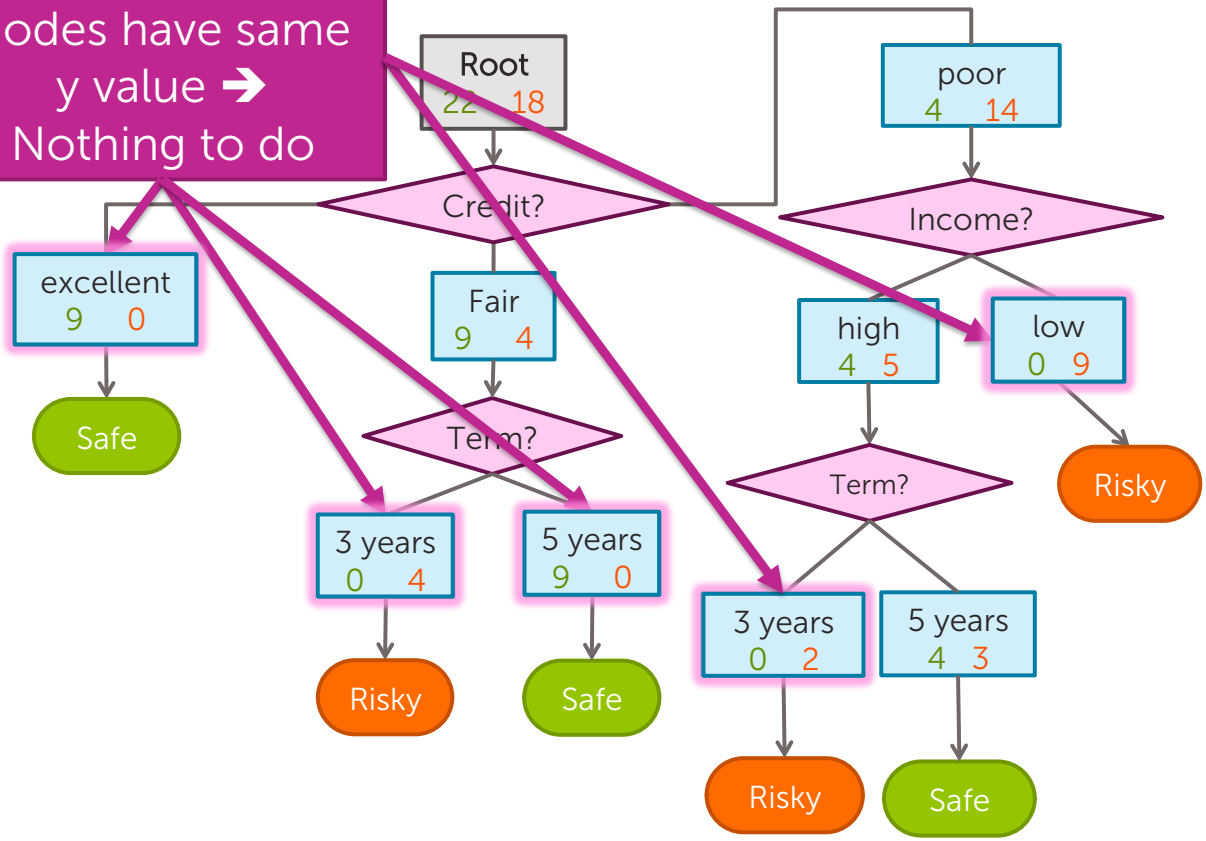


When do we stop???

What are the criteria to stop recursing?

Stopping condition 1: All data agrees on y

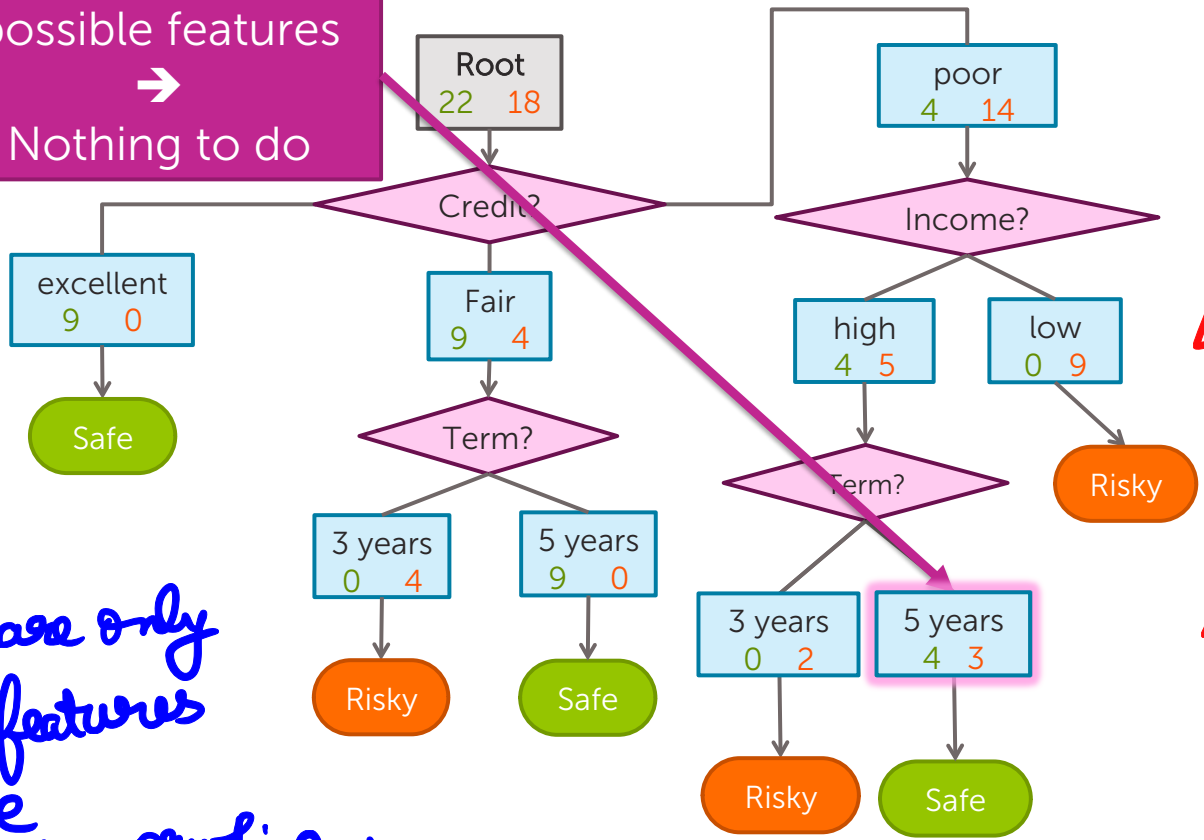
All data in these nodes have same y value ➔ Nothing to do



Stop splitting when all the data agrees on that value of Y

Stopping condition 2: Already split on all features

Already split on all possible features
➔
Nothing to do



Here are only three features and we splitted everything.

Stop

↳ If every data-point agrees on the value of y

↳ If you run out of features.

Greedy decision tree learning

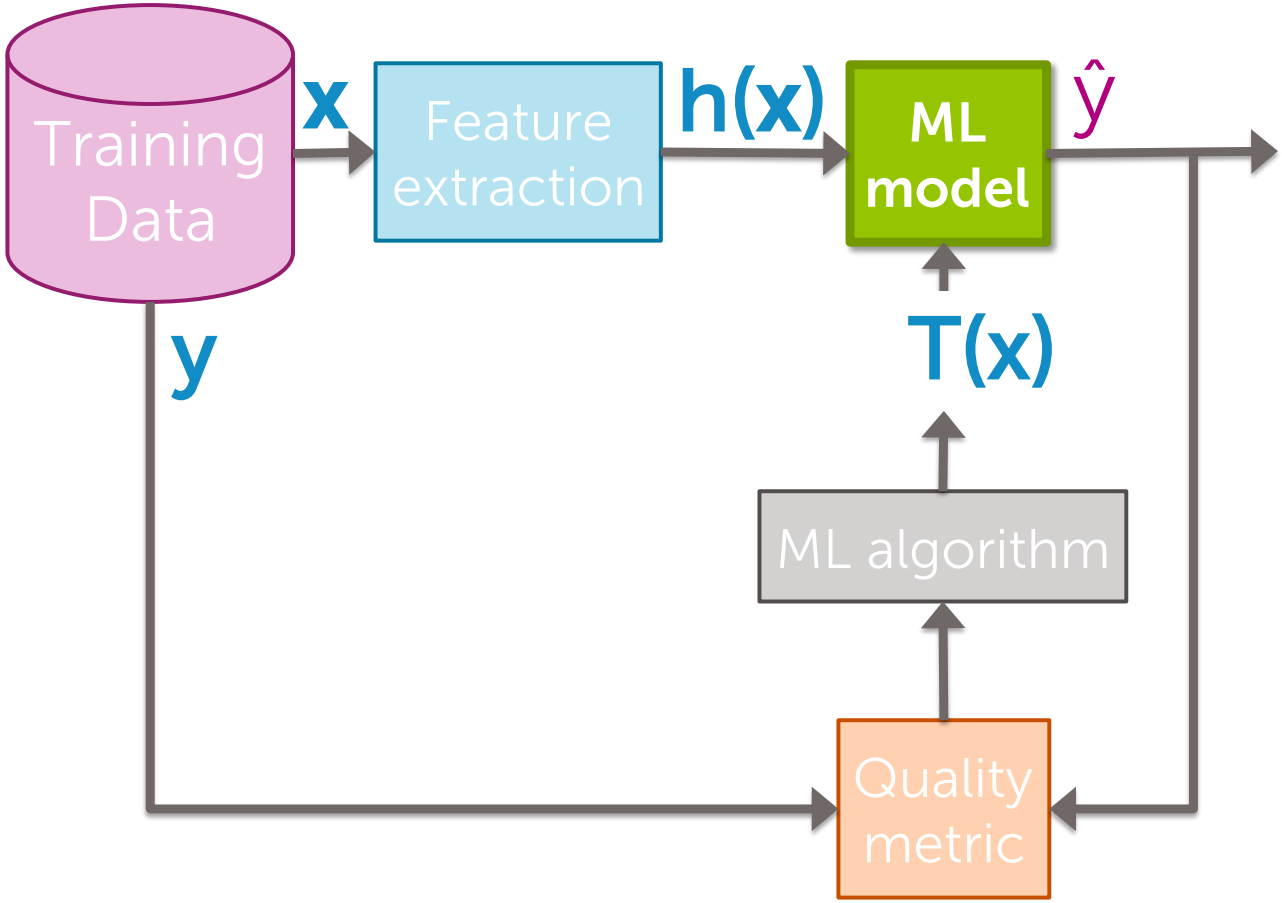
- Step 1: Start with an empty tree
- Step 2: Select a feature to split data
- For each split of the tree:
  - Step 3: If nothing more to, make predictions
  - Step 4: Otherwise, go to Step 2 & continue (recurse) on this split

Pick feature split leading to lowest classification error

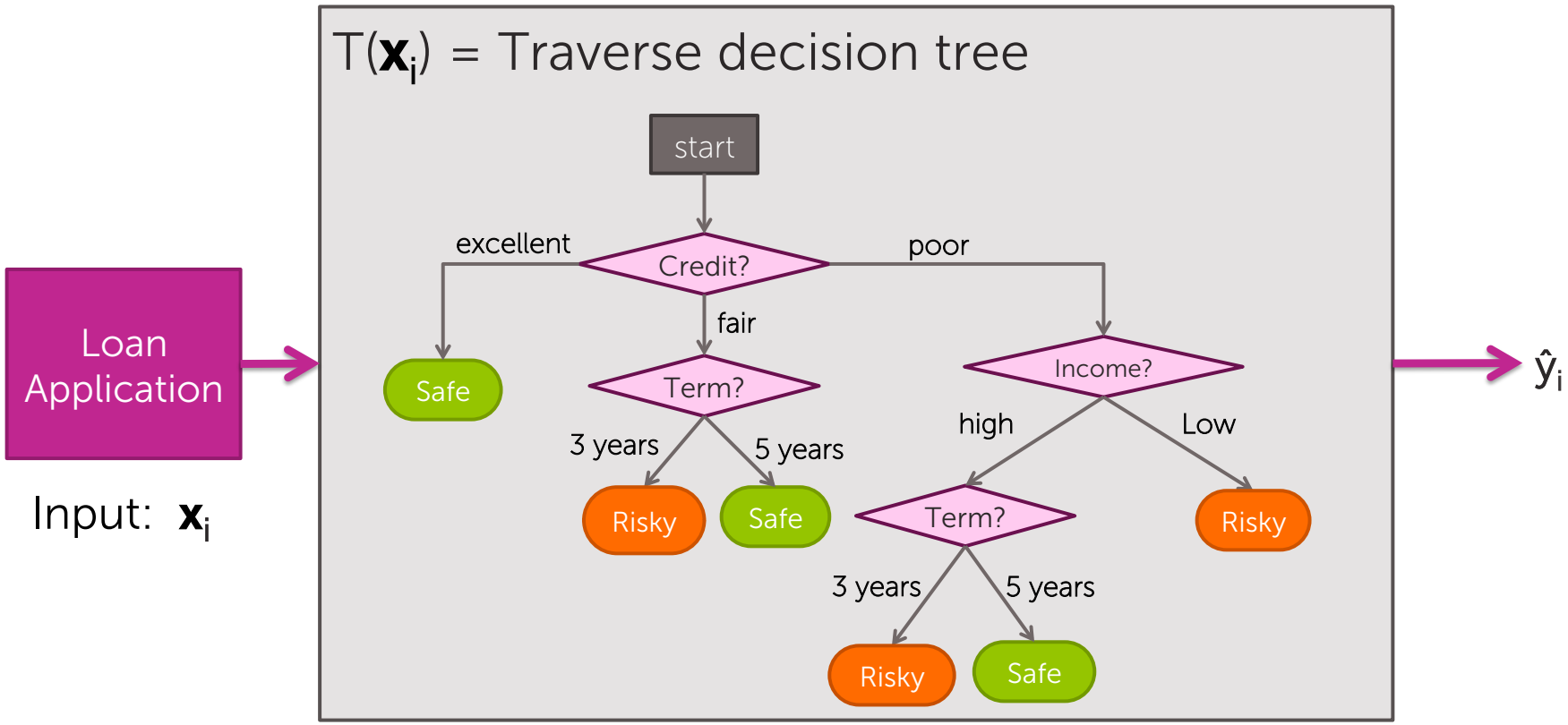
Stopping conditions 1 & 2

Recursion

Predictions with decision trees

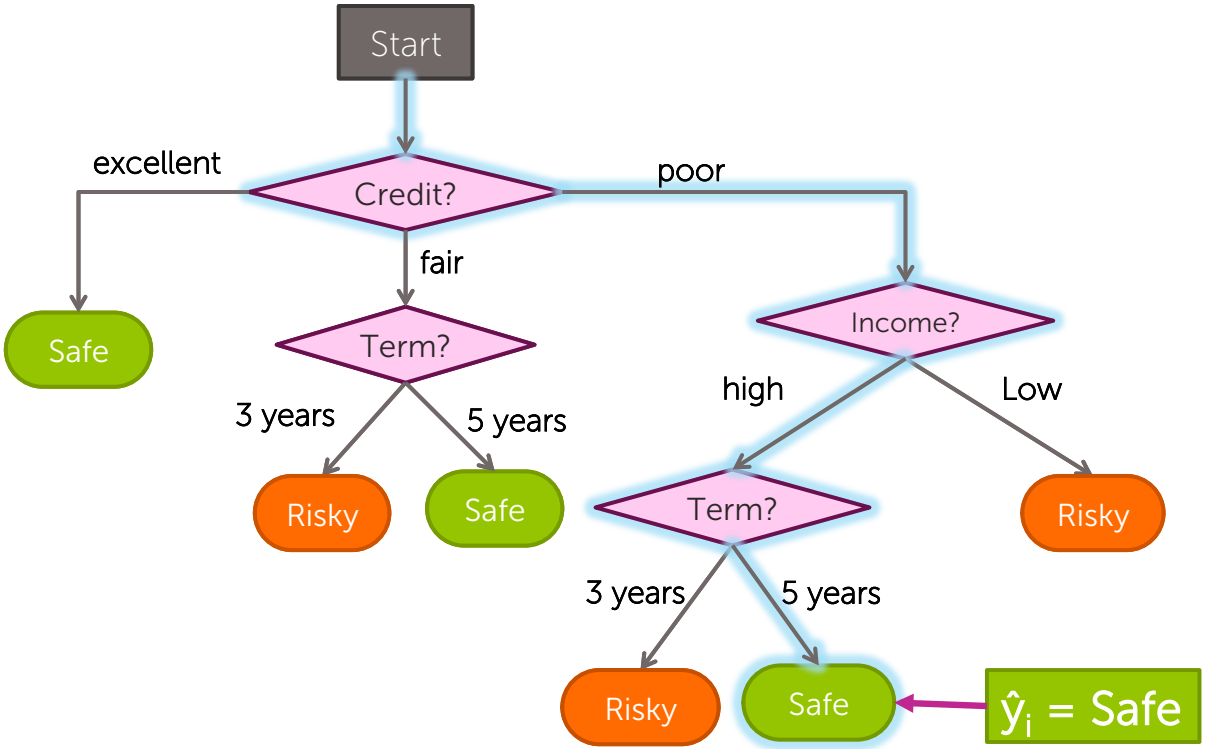


Decision tree model



Traversing a decision tree

$\mathbf{x}_i$ = (Credit = poor, Income = high, Term = 5 years)



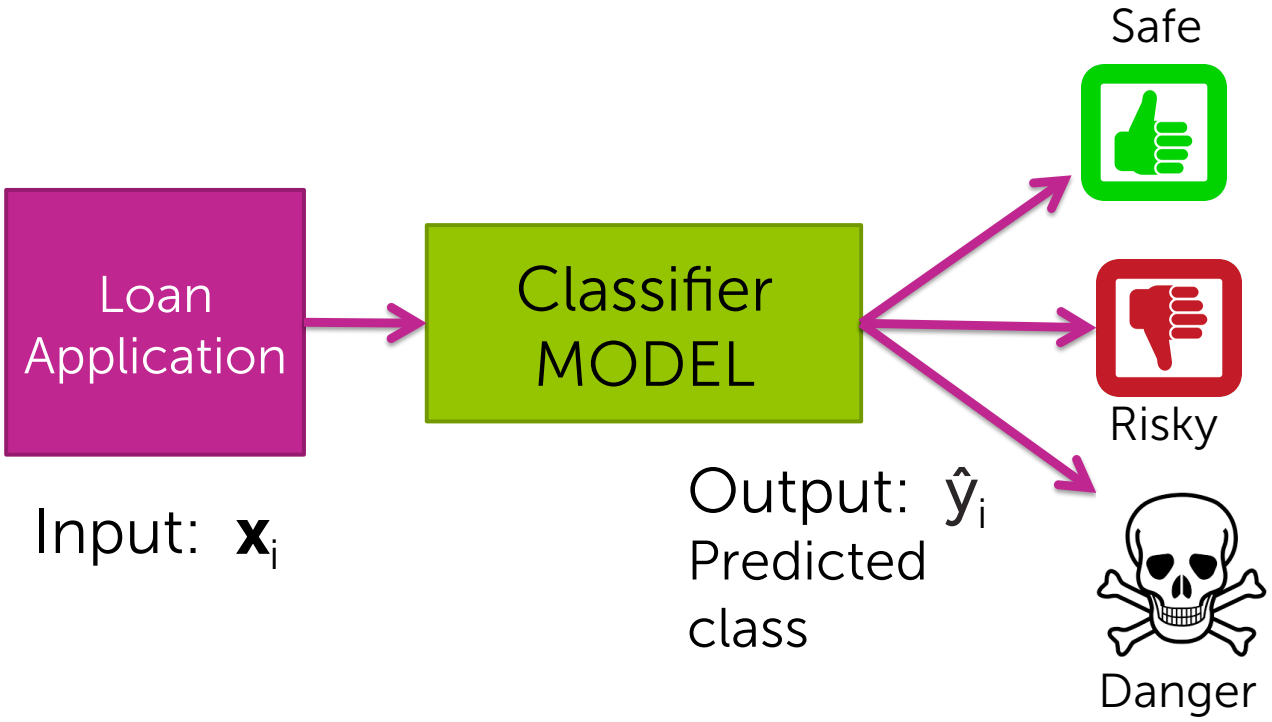
Decision tree prediction algorithm

predict(tree_node, input)

- If current tree_node is a leaf:
  - return majority class of data points in leaf
- else:
  - next_note = child node of tree_node whose feature value agrees with input
  - return predict(next_note, input)

Multiclass classification & predicting probabilities

Multiclass prediction

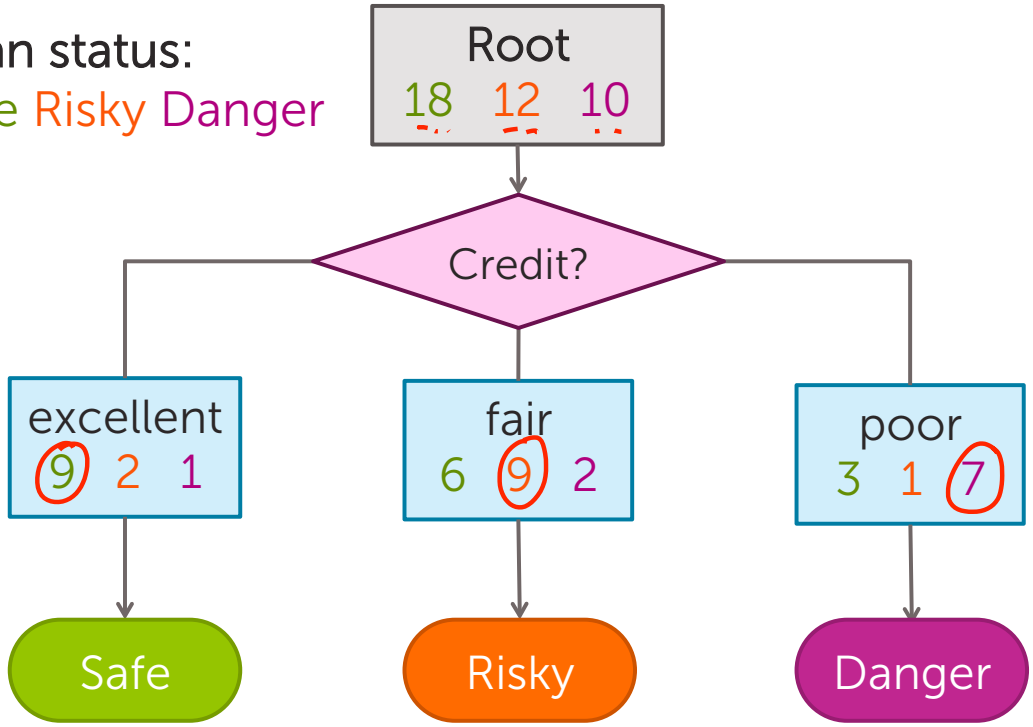


Multiclass decision stump

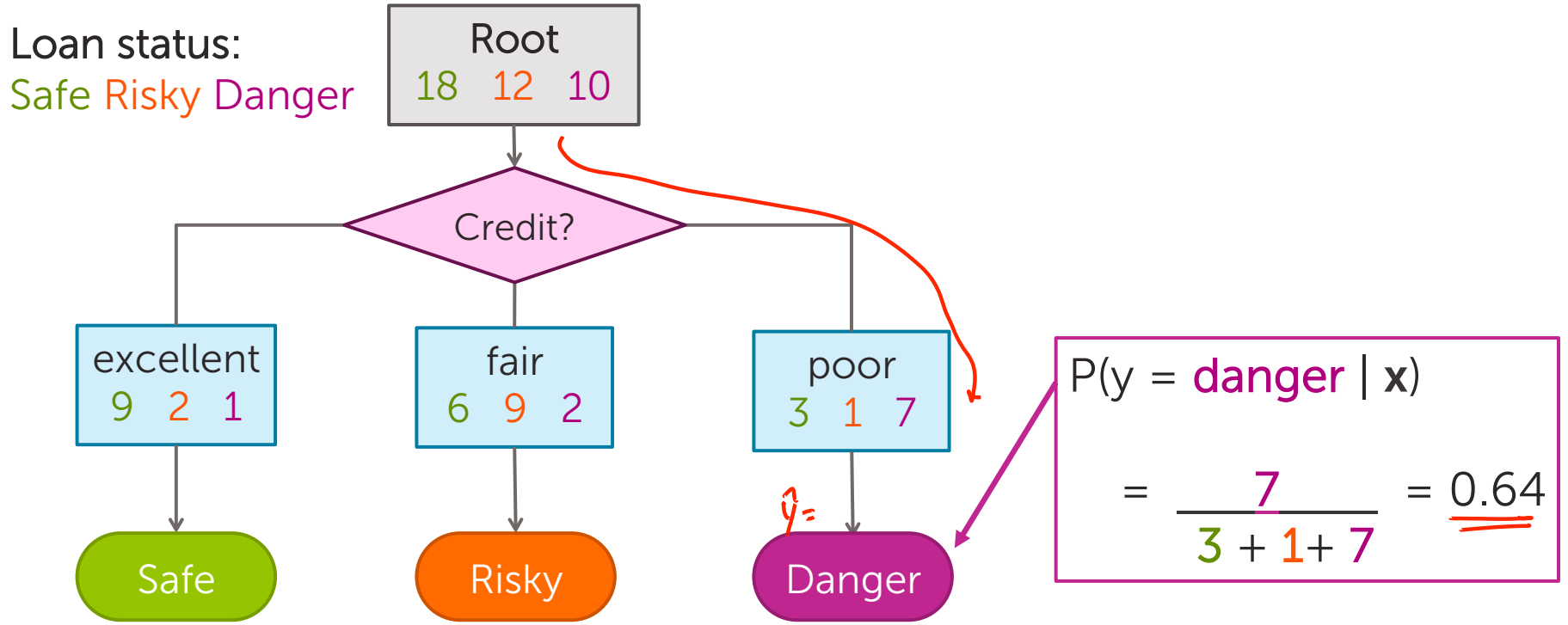
N = 40,
1 feature,
3 classes

Credit	y
excellent	safe
fair	risky
fair	safe
poor	danger
excellent	risky
fair	safe
poor	danger
poor	safe
fair	safe
...	...

Loan status:
Safe Risky Danger



Predicting probabilities with decision trees



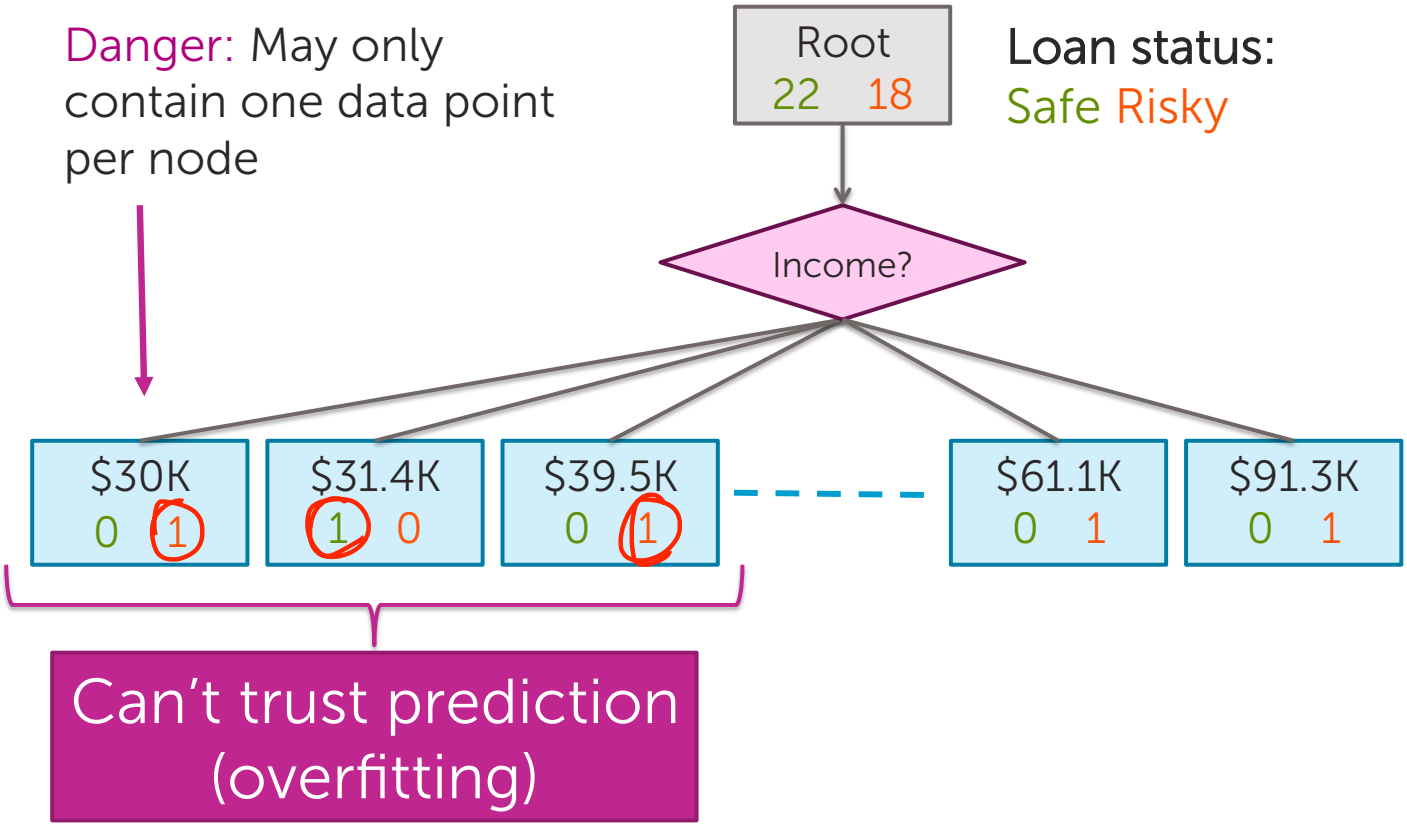
Decision tree learning:
Real valued features

How do we use real values inputs?

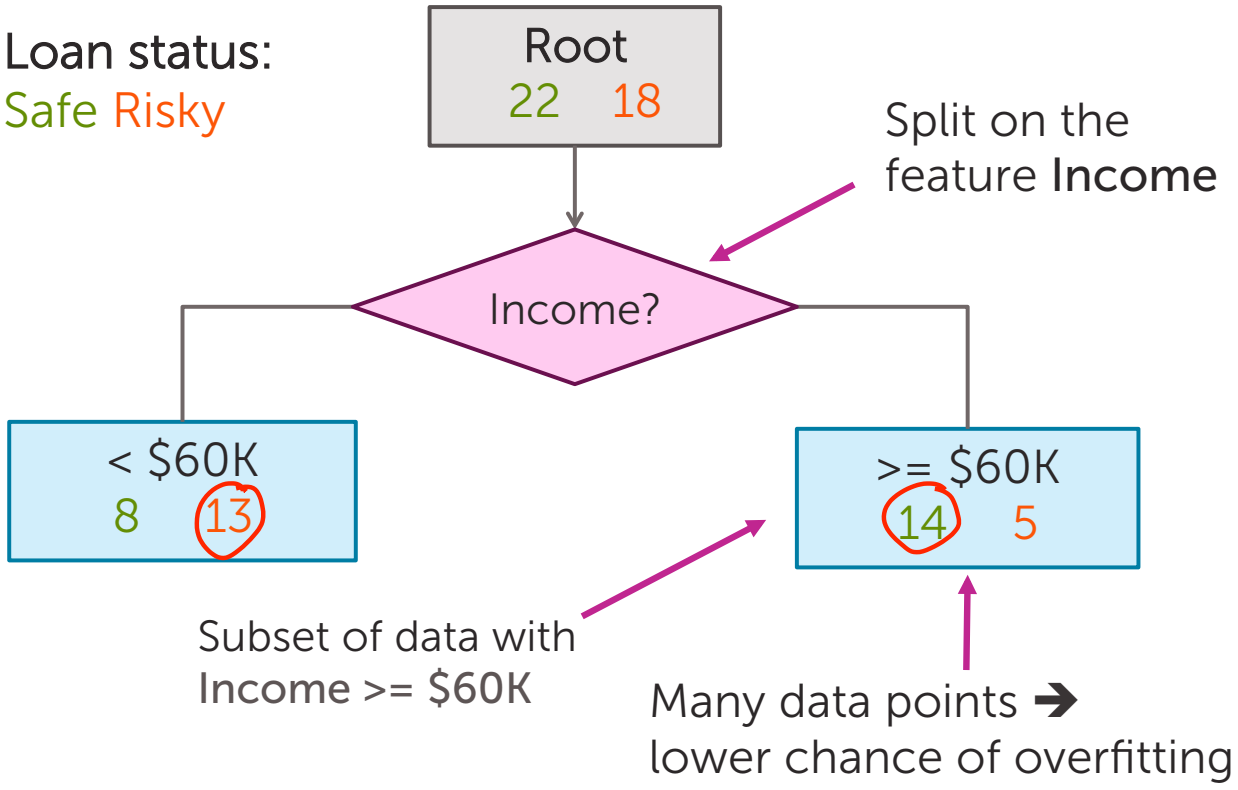
Income	Credit	Term	y
$105 K	excellent	3 yrs	Safe
$112 K	good	5 yrs	Risky
$73 K	fair	3 yrs	Safe
$69 K	excellent	5 yrs	Safe
$217 K	excellent	3 yrs	Risky
$120 K	good	5 yrs	Safe
$64 K	fair	3 yrs	Risky
$340 K	excellent	5 yrs	Safe
$60 K	good	3 yrs	Risky

Split on each numeric value?

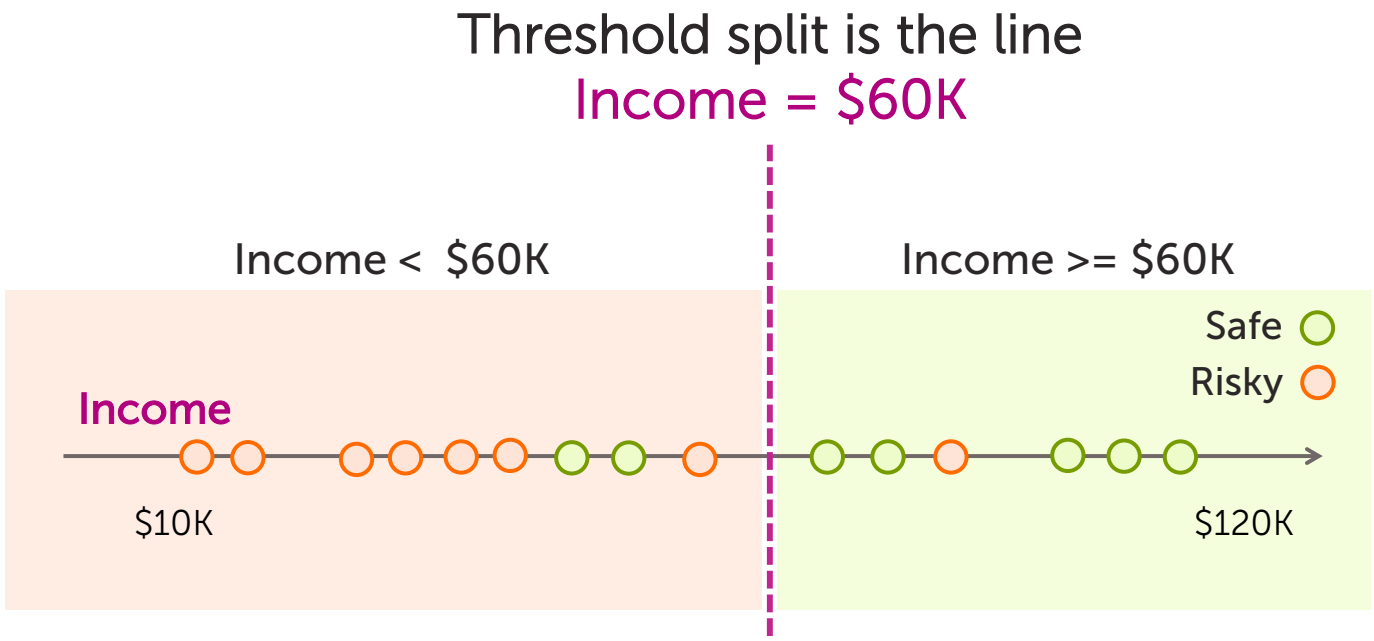
Danger: May only contain one data point per node



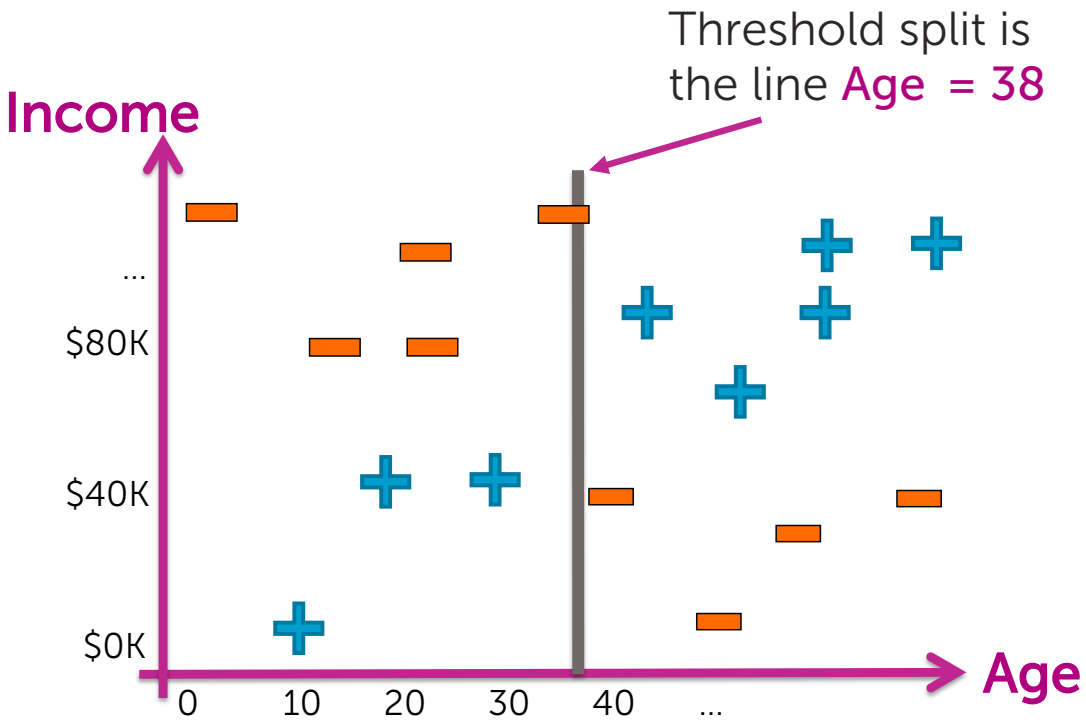
Alternative: Threshold split



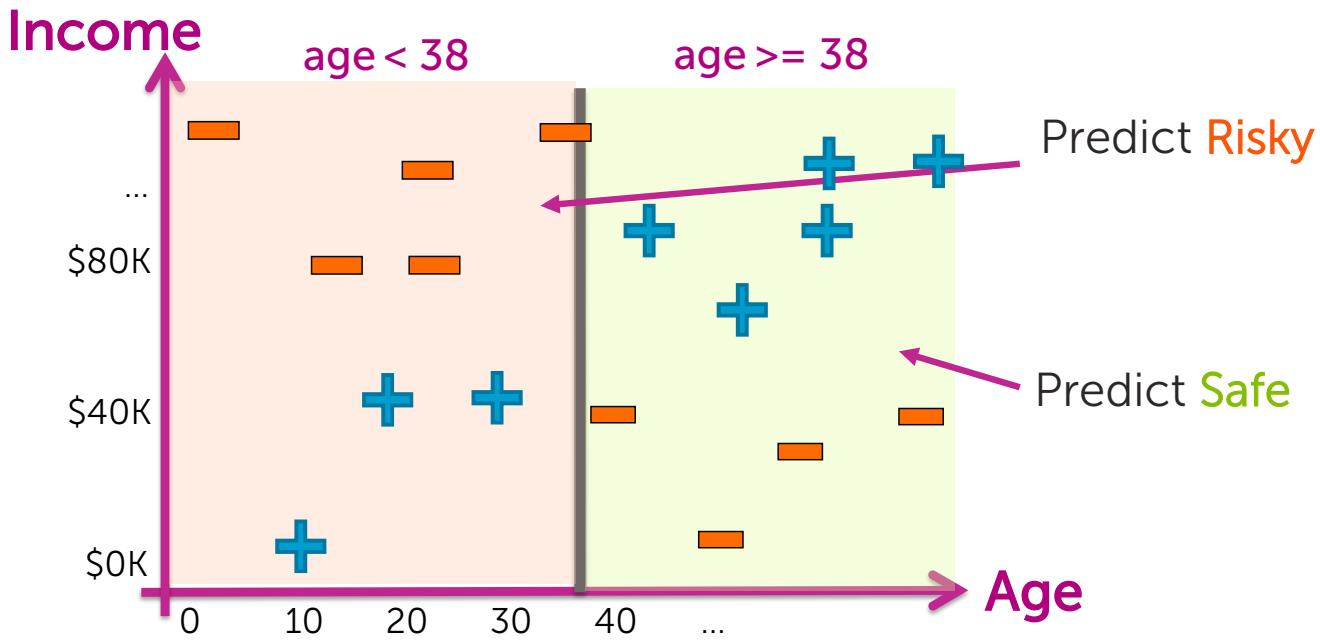
Threshold splits in 1-D



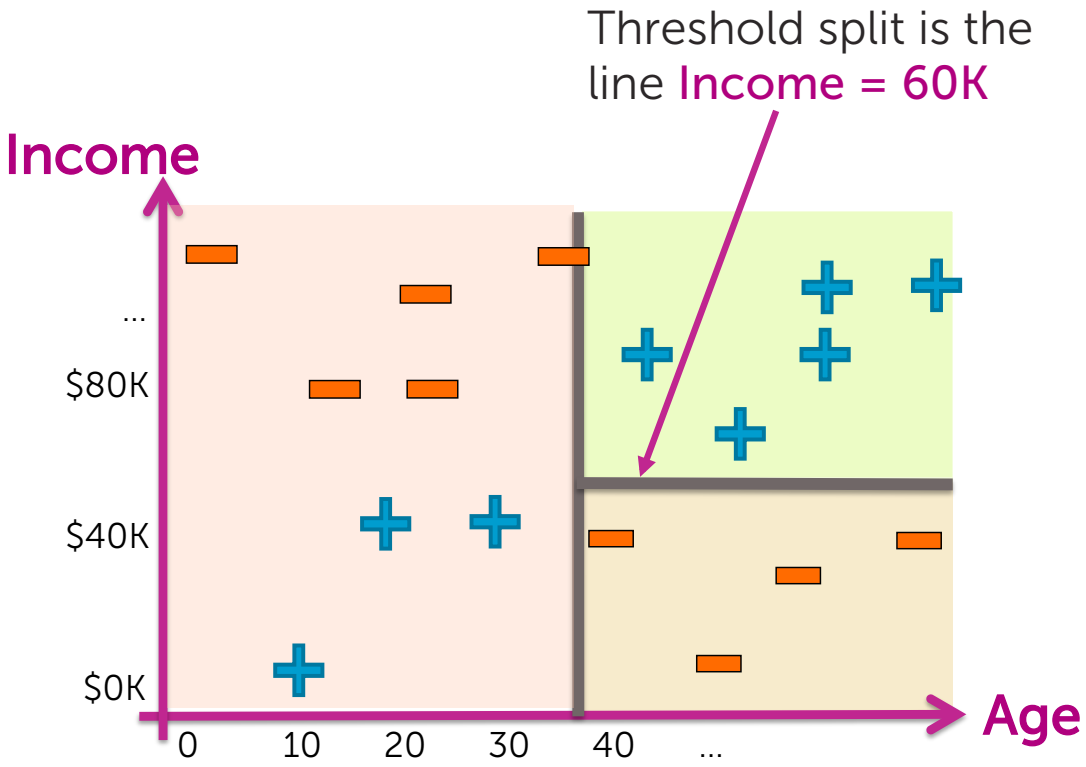
Visualizing the threshold split



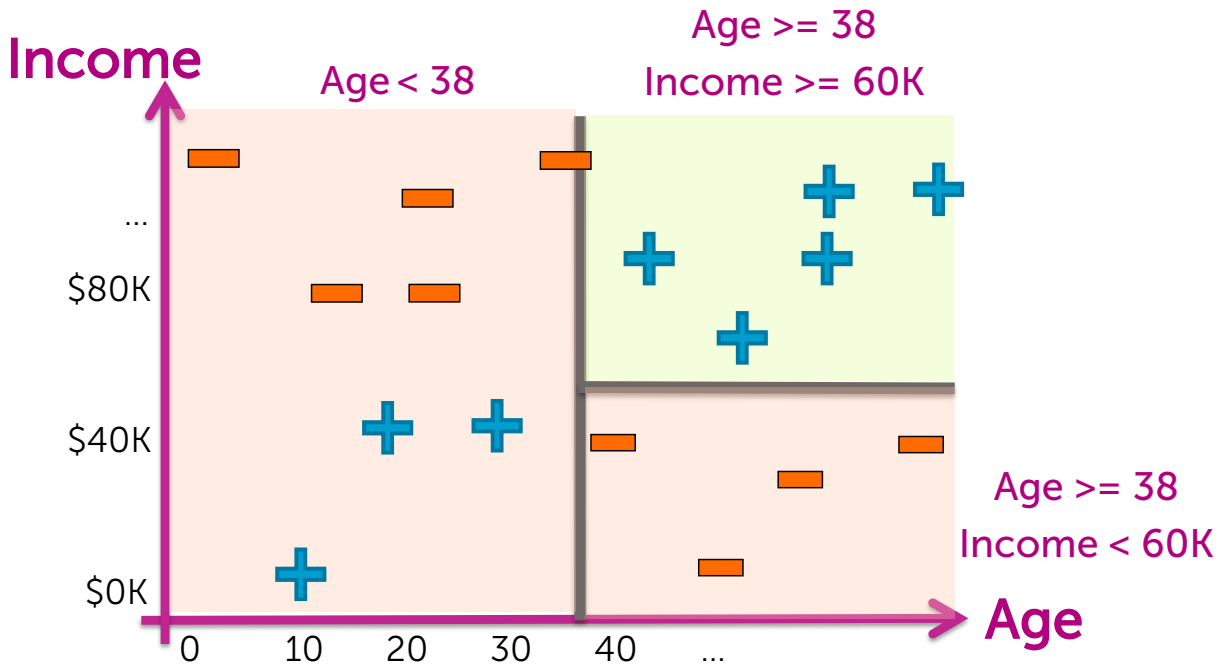
Split on Age >= 38

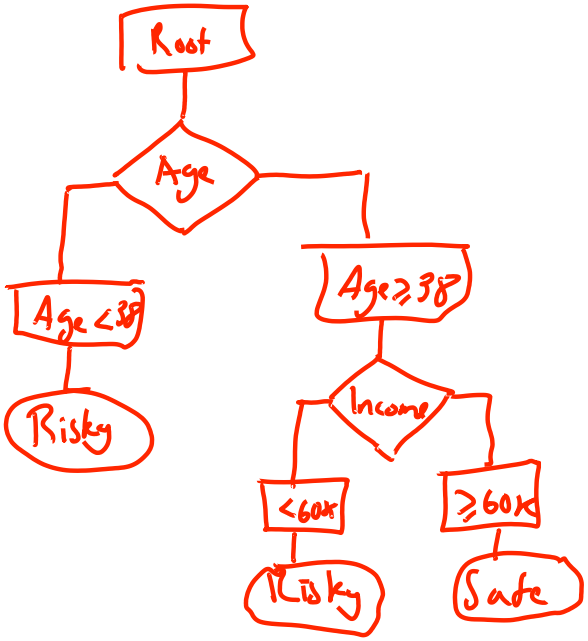


Depth 2: Split on Income >= $60K



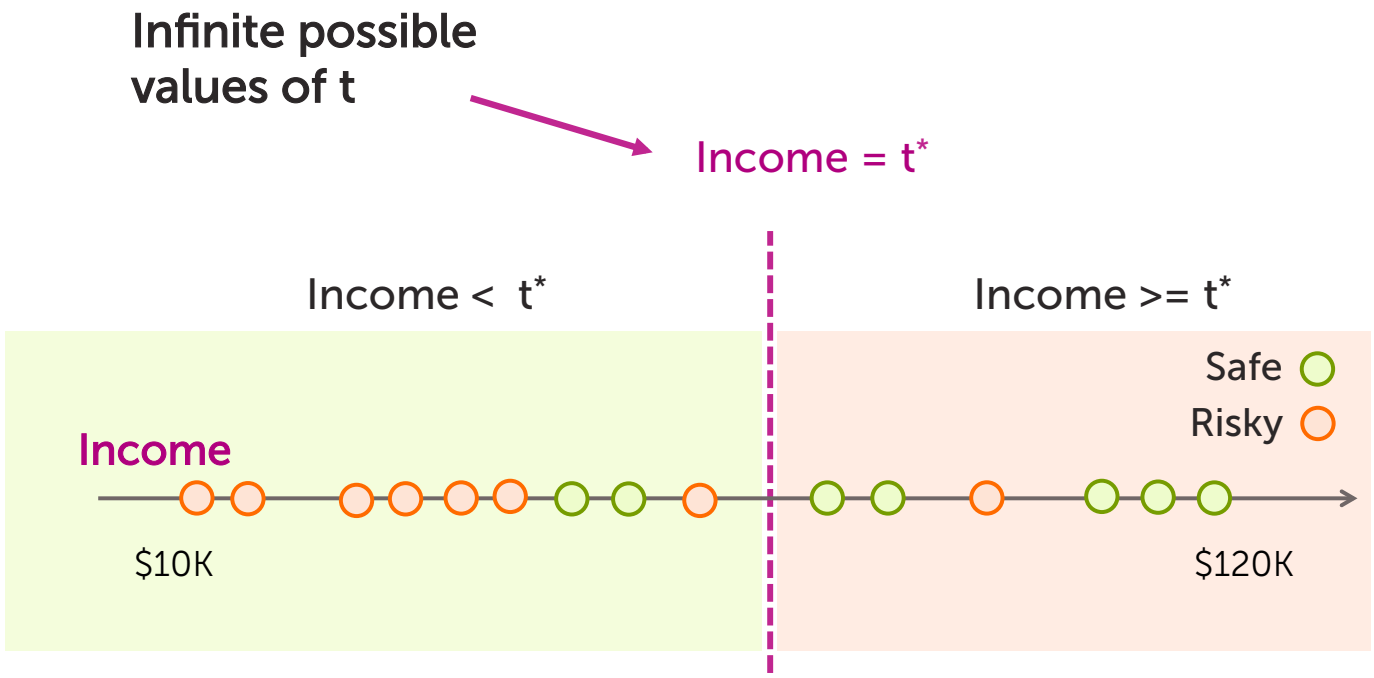
Each split partitions the 2-D space





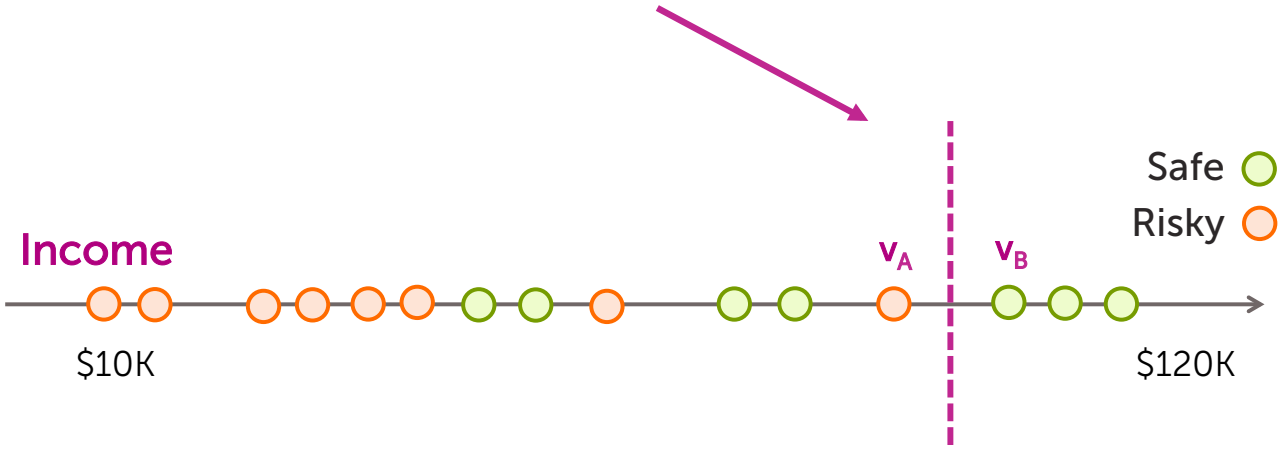
Finding the best threshold split

Finding the best threshold split

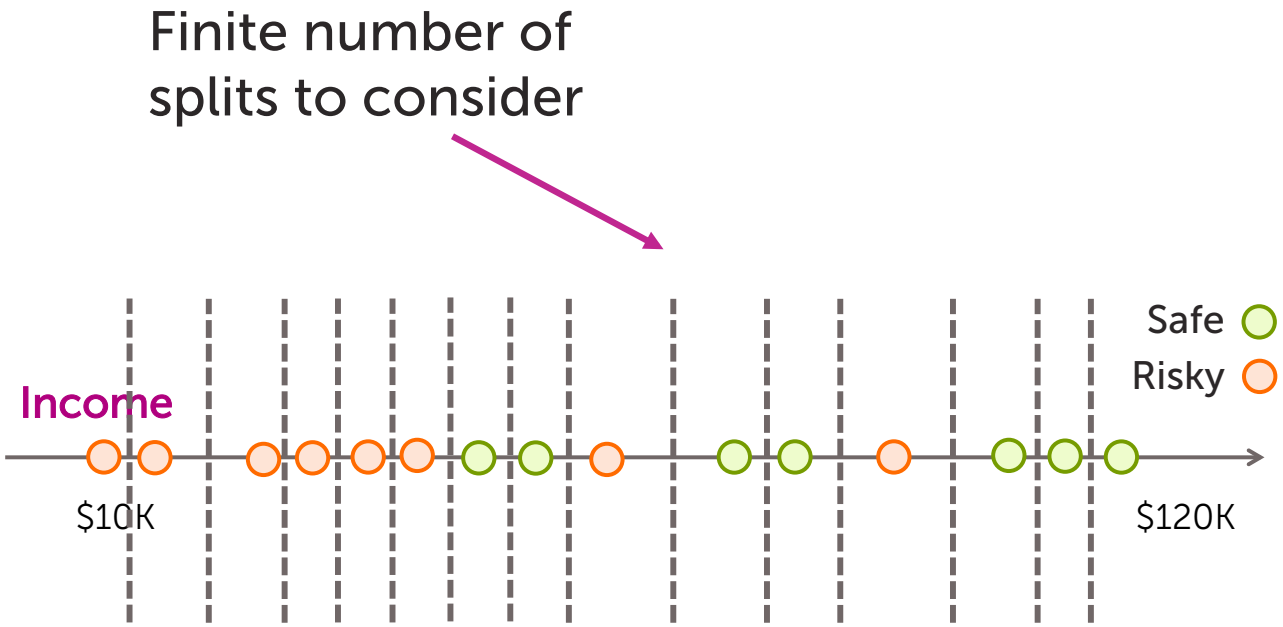


Consider a threshold between points

Same classification error for any threshold split between v_A and v_B



Only need to consider mid-points



Threshold split selection algorithm

- Step 1: Sort the values of a feature $h_j(\mathbf{x})$:

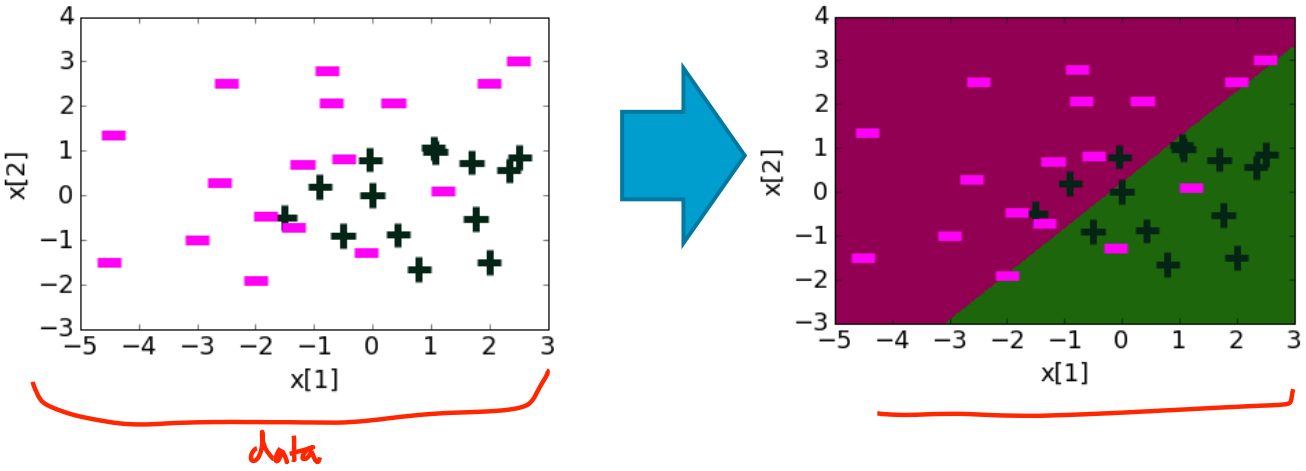
  Let $\{v_1, v_2, v_3, \ldots v_N\}$ denote sorted values

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

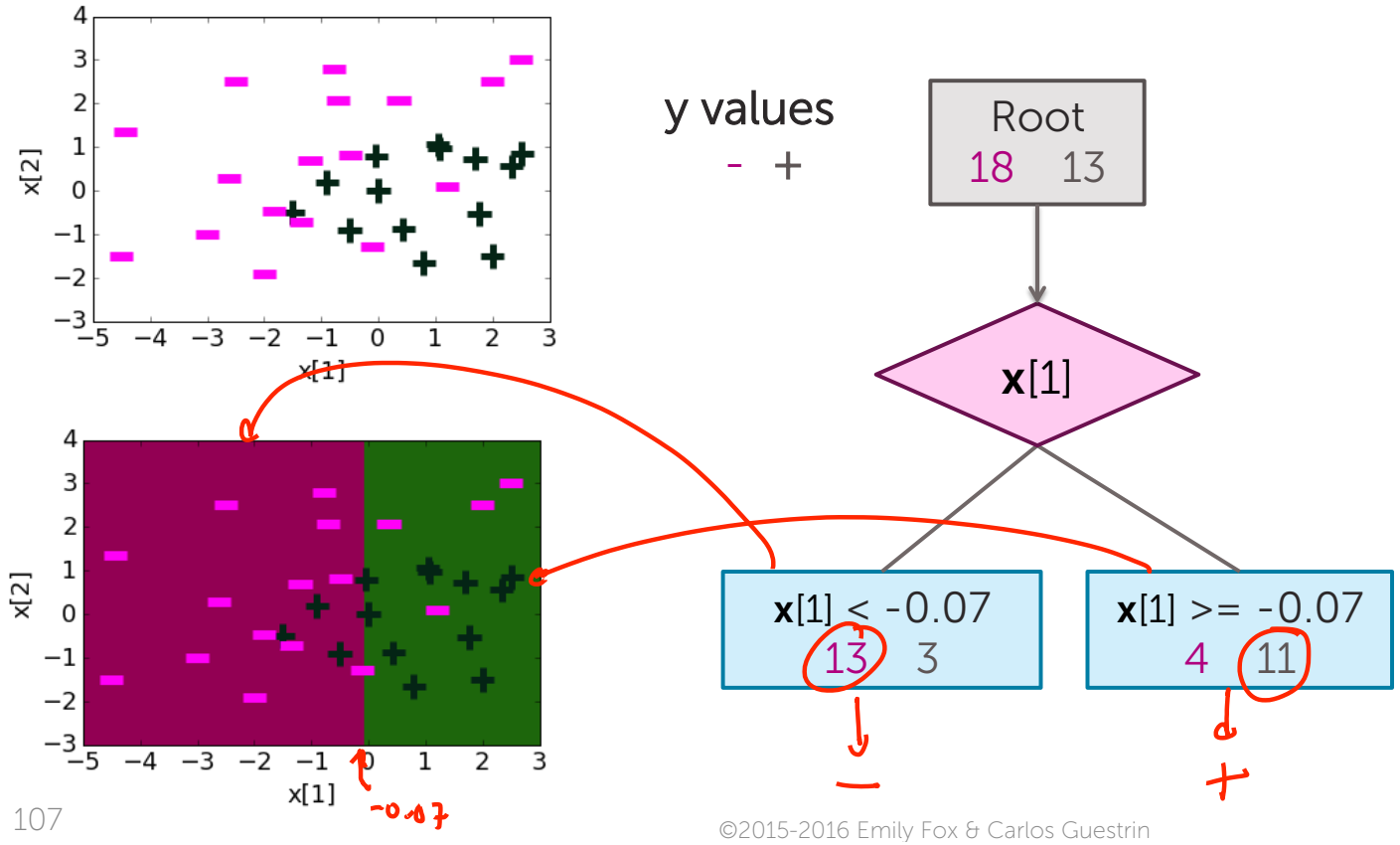
Decision trees vs logistic regression:
Example

Logistic regression

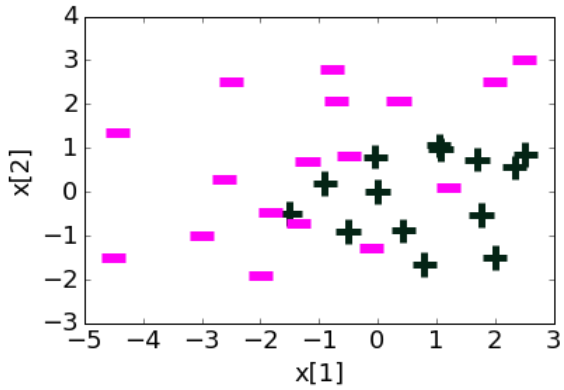
Feature	Value	Weight Learned
$h_0(\mathbf{x})$	1	0.22
$h_1(\mathbf{x})$	$\mathbf{x}[1]$	1.12
$h_2(\mathbf{x})$	$\mathbf{x}[2]$	-1.07

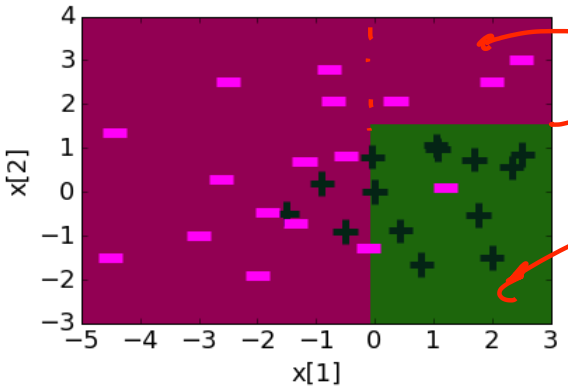


Depth 1: Split on **x**[1]



Depth 2





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

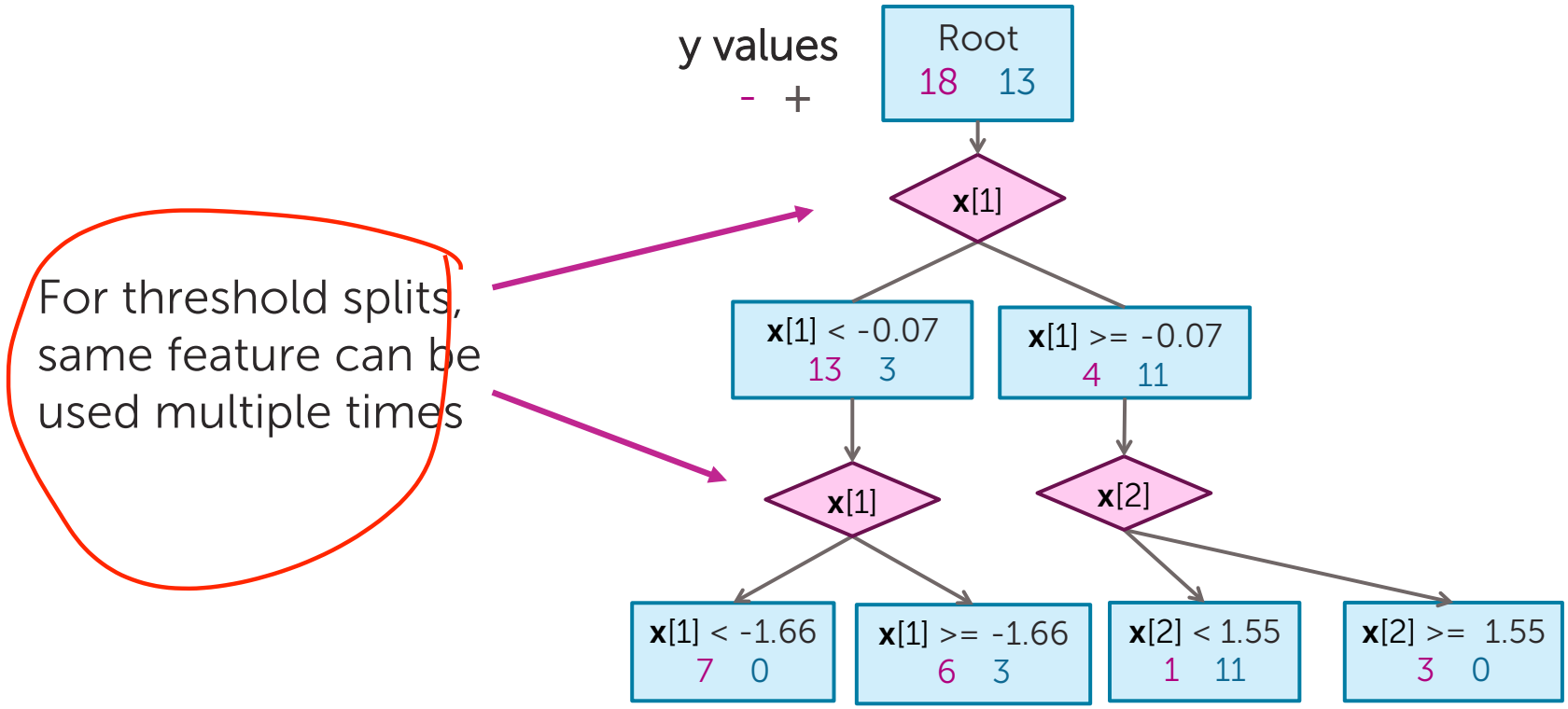
-1

-1

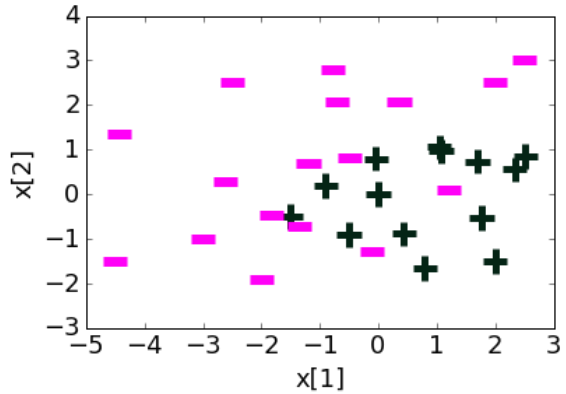
11

-1

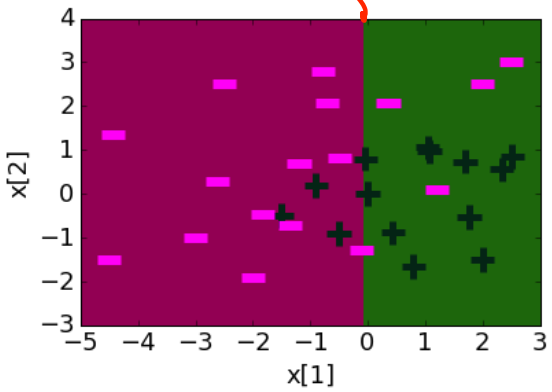
Threshold split caveat



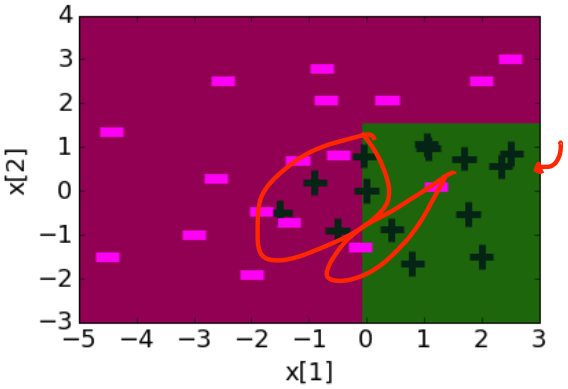
Decision boundaries



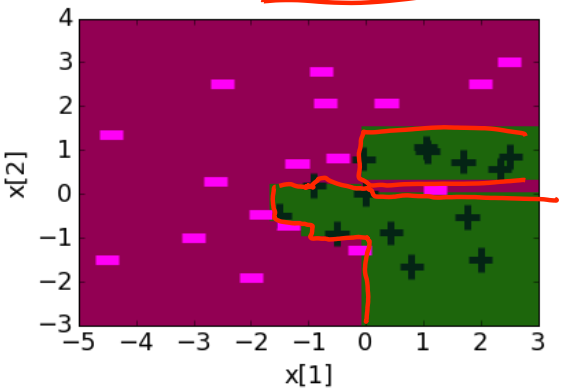
Depth 1



Depth 2

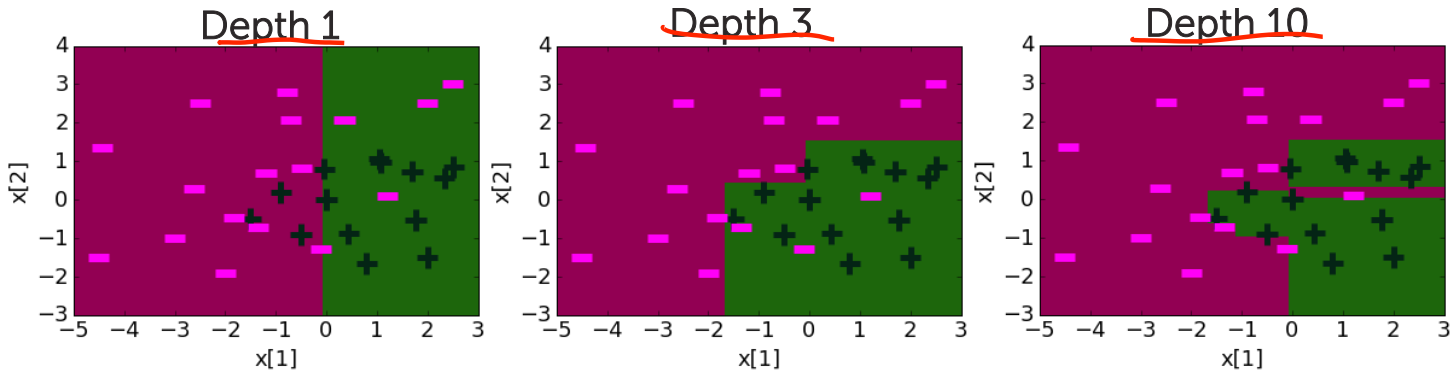


Depth 10

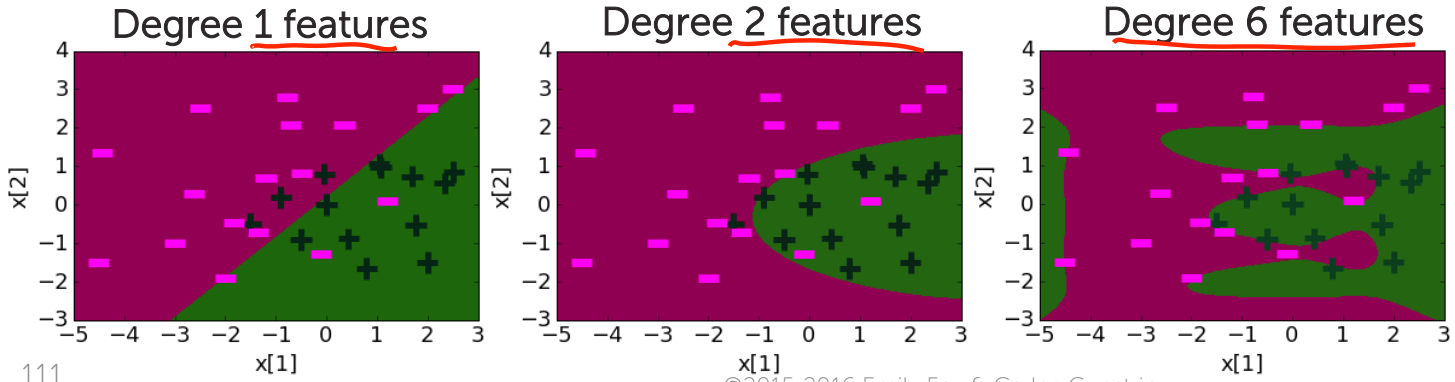


Comparing decision boundaries

Decision Tree



Logistic Regression



Summary of decision trees

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
  - Probability predictions
  - Multiclass classification

Thank you to Dr. Krishna Sridhar

Dr. Krishna Sridhar

Staff Data Scientist, Dato, Inc.