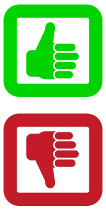


Decision Trees

CS 229: Machine Learning

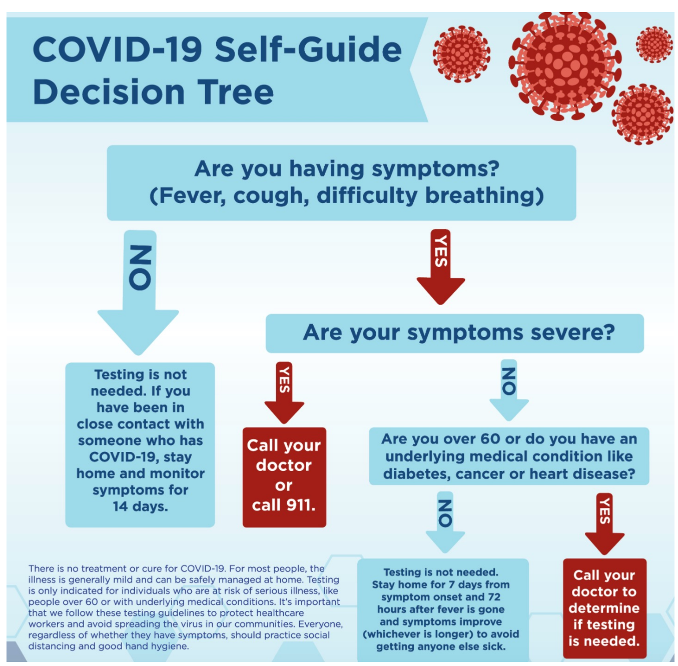
Emily Fox
Stanford University

February 12, 2024

Slides include content developed by and co-developed with Carlos Guestrin

1

How do we make decisions?



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

2

Predicting potential loan defaults

3

What makes a loan risky?

Credit History ★★★★

Income ★★★

Term ★★★★★

Personal Info ★★★

4

Credit history explained

Did I pay previous loans on time?

Example:
excellent, good, or fair

Credit History
★★★★

Income
★★★

Term
★★★★★

Personal Info
★★★

5

Income

What's my income?

Example:
$80K per year

Credit History
★★★★

Income
★★★

Term
★★★★★

Personal Info
★★★

6

Loan terms

How soon do I need to pay the loan?

Example: 3 years, 5 years,…

Credit History ★★★★

Income ★★★

Term ★★★★★

Personal Info ★★★

 CS 229: Machine Learning

7

Personal information

Age, reason for the loan, marital status,…

Example: Home loan for a married couple

Credit History ★★★★

Income ★★★

Term ★★★★★

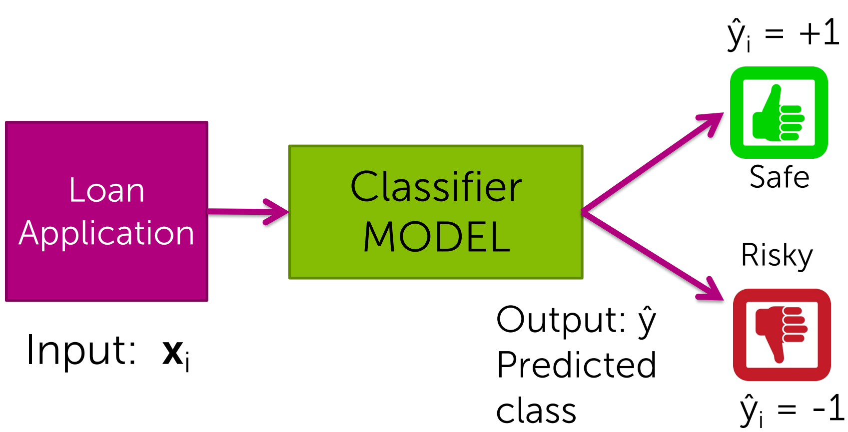
Personal Info ★★★

 CS 229: Machine Learning

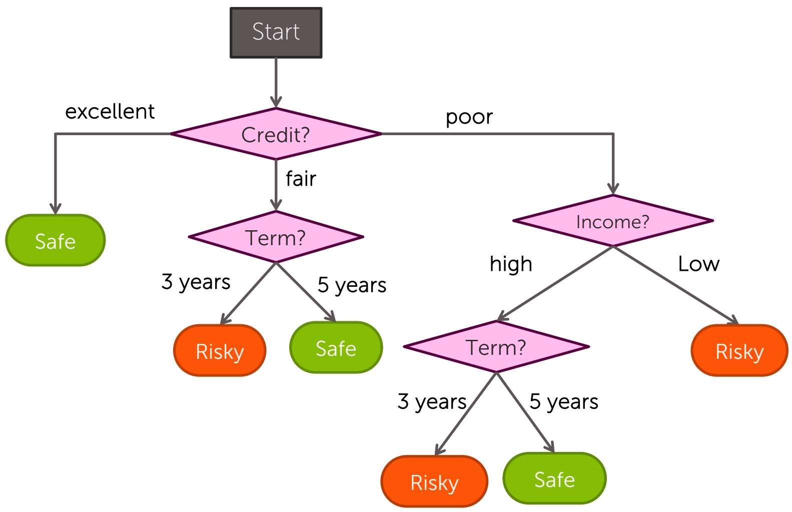
8

Intelligent application

Classifier review



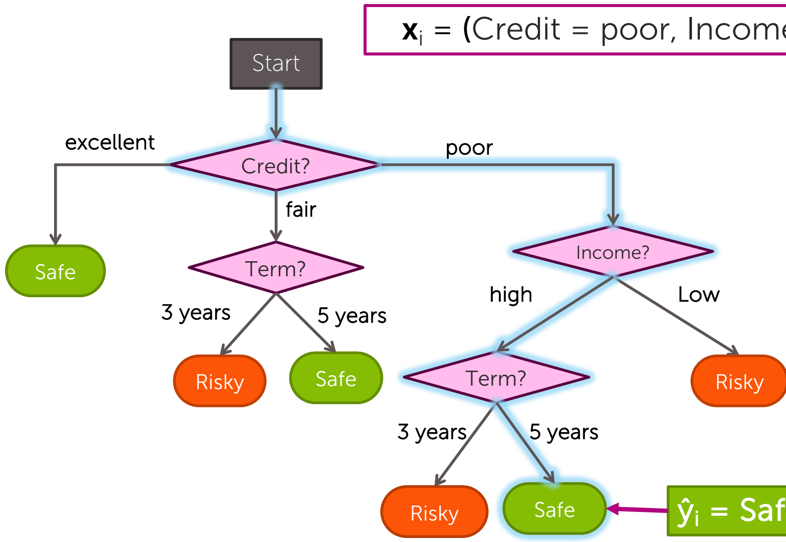
This module … decision trees



11

Scoring a loan application

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



$\hat{y}_i$ = Safe

12

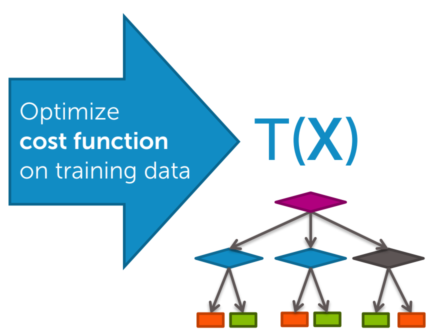
Decision tree learning task

13

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



14

Cost function: Classification error

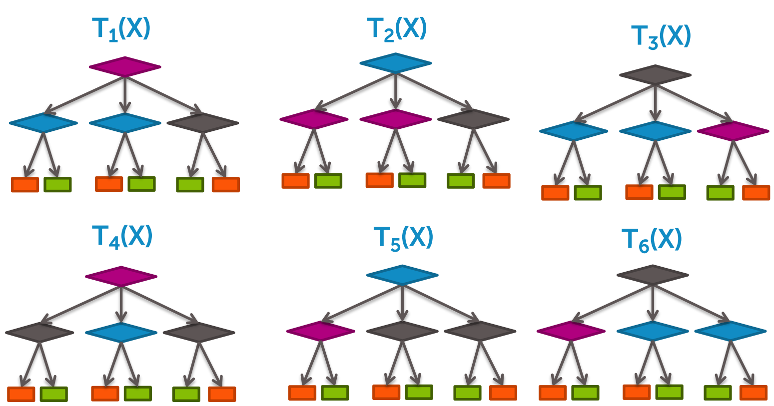
- Error measures fraction of mistakes

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

  – Best possible value : 0.0
  – Worst possible value: 1.0

How do we find the best tree?

Exponentially large number of possible trees makes decision tree learning hard!



Learning the smallest decision tree is an *NP-hard problem* [Hyafil & Rivest '76]

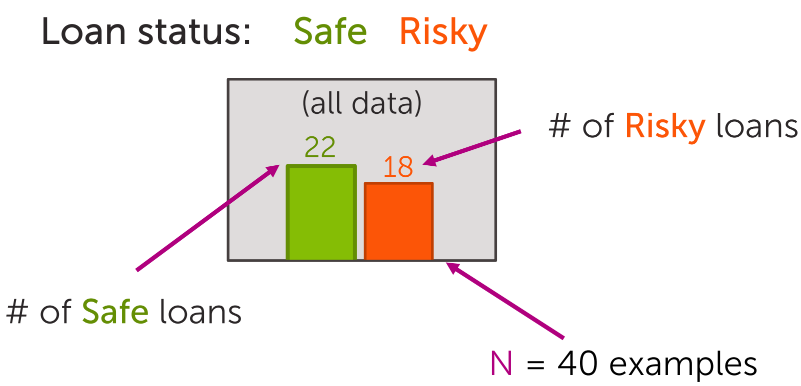
Greedy decision tree learning

Our training data table

Assume N = 40, 3 features

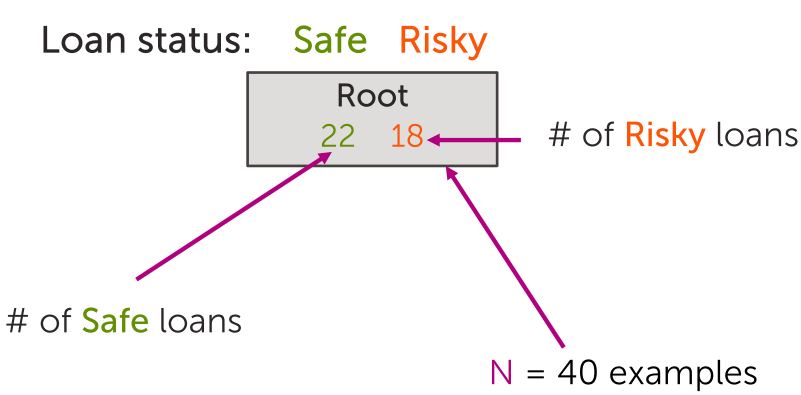
Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	risky
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe

Start with all the data



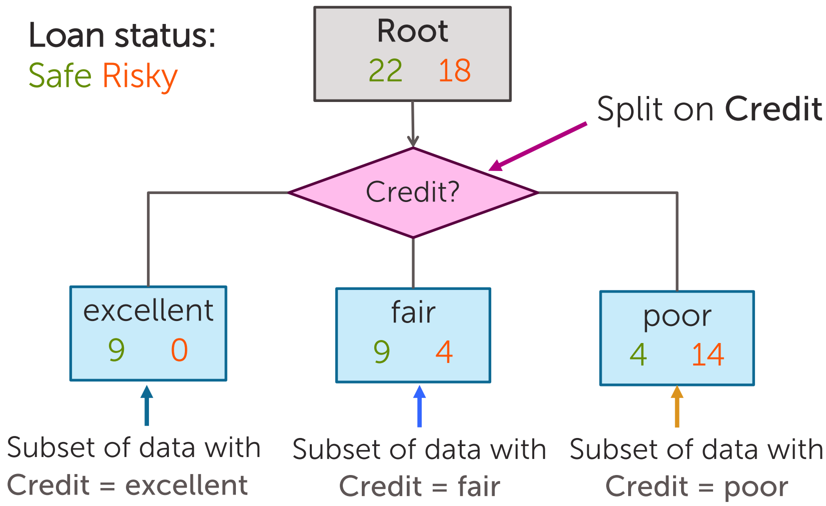
19

Compact visual notation: Root node

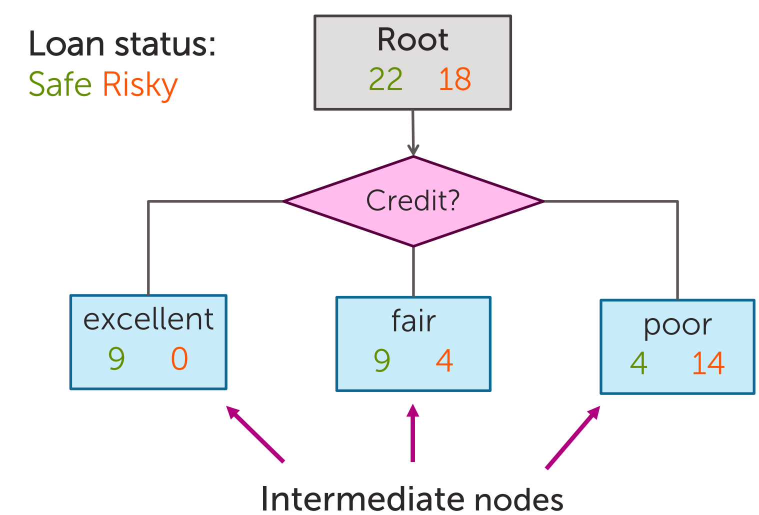


20

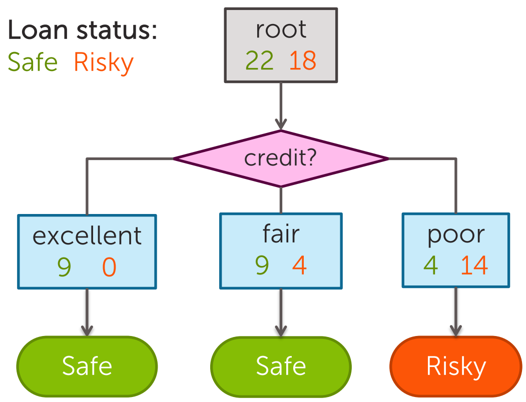
Decision stump: Single level tree



Visual notation: Intermediate nodes



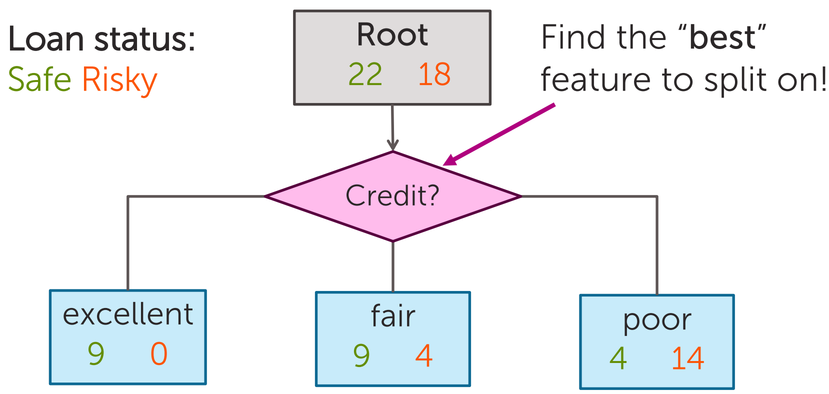
Making predictions with a decision stump



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

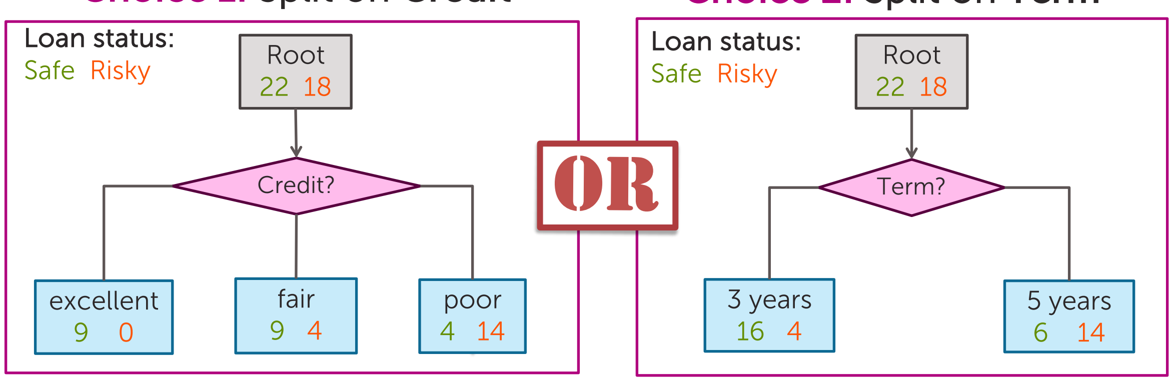
Selecting best feature to split on

How do we learn a decision stump?



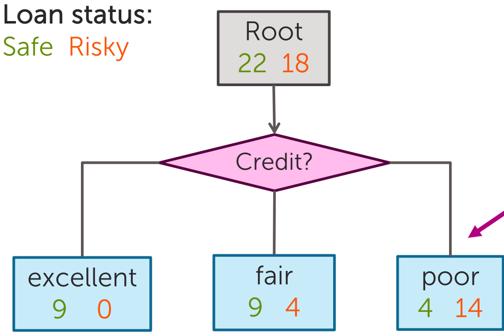
How do we select the best feature?

Choice 1: Split on **Credit** **Choice 2:** Split on **Term**



How do we measure effectiveness of a split?

Loan status:
Safe Risky



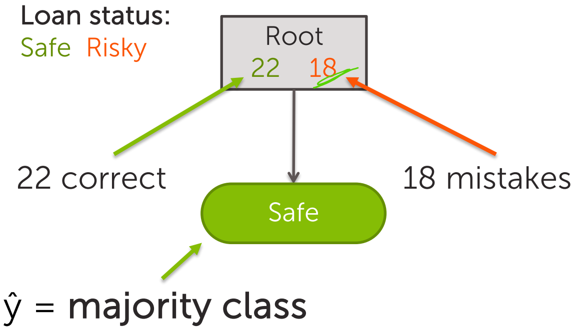
Idea: Calculate classification error of this decision stump

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

Calculating classification error

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

Loan status:
Safe Risky

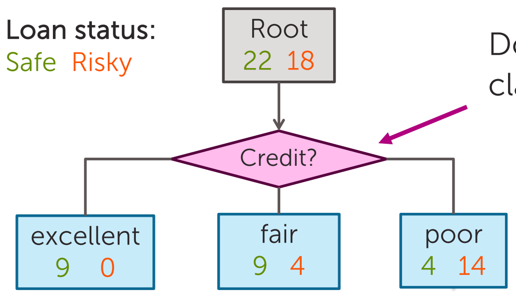


$$\text{Error} = \frac{18}{18+22=40}$$
$$= 0.45$$

Tree	Classification error
(root)	0.45

Choice 1: Split on Credit history?

Choice 1: Split on **Credit**

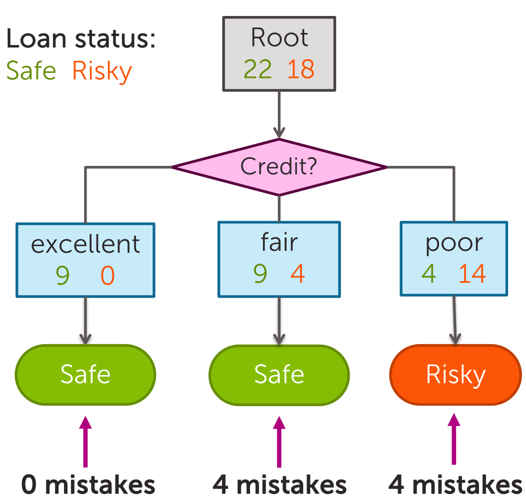


Does a **split on Credit** reduce classification error below 0.45?

29

Split on Credit: Classification error

Choice 1: Split on **Credit**



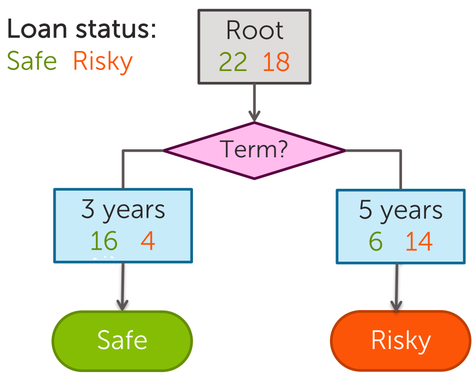
$$\text{Error} = \frac{0+4+4}{40} = 0.2$$

Tree	Classification error
(root)	0.45
Split on **credit**	0.2

30

Choice 2: Split on Term?

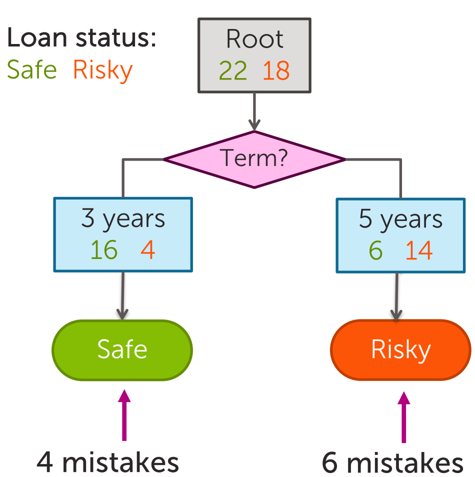
Choice 2: Split on Term



31

Evaluating the 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

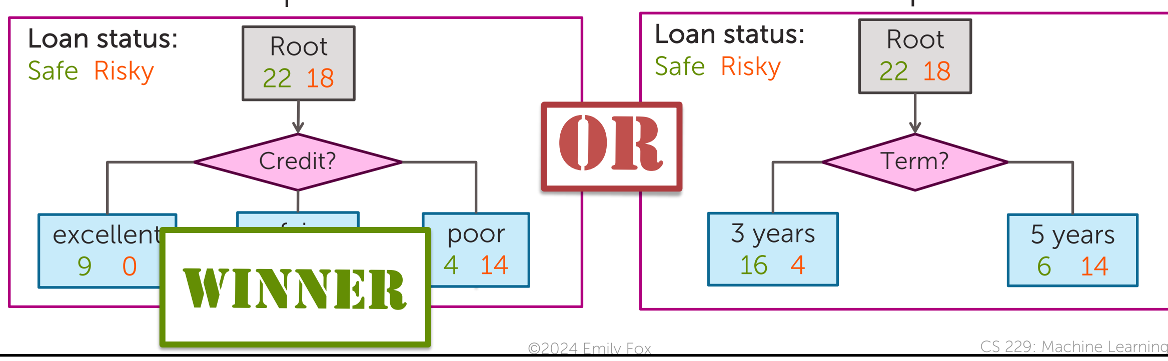
32

Choice 1 vs Choice 2: Comparing split on Credit vs Term

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

Choice 1: Split on **Credit**

Choice 2: Split on **Term**



33

Feature split selection algorithm

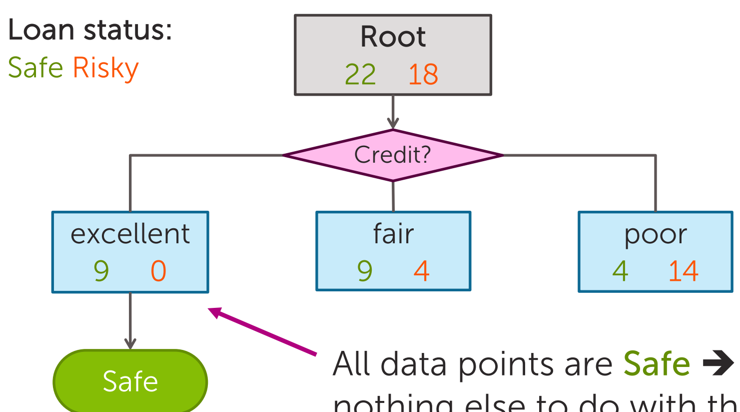
- Given a subset of data M (a node in a tree)
- For each feature $h_i(x)$:
  1. Split data of M according to feature $h_i(x)$
  2. Compute classification error of split
- Chose feature $h^*(x)$ with lowest classification error

34

Recursion & Stopping conditions

35

We've learned a decision stump, what next?



All data points are **Safe** ➔
nothing else to do with this subset of data

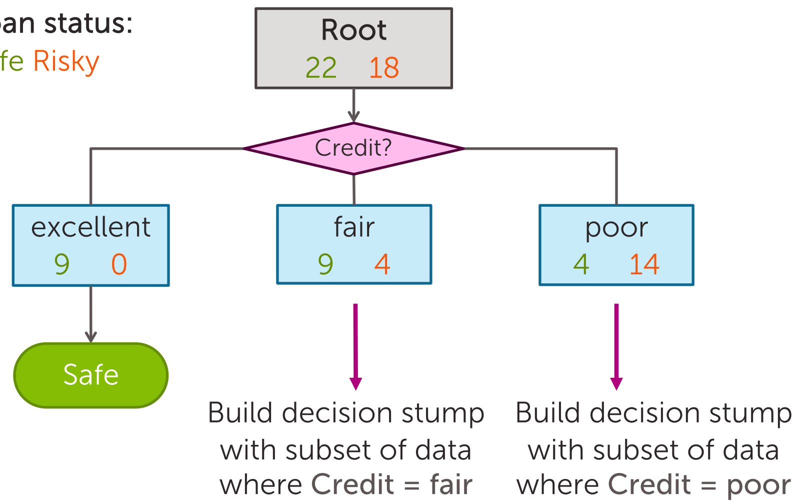
Leaf node

©2024 Emily Fox CS 229: Machine Learning

36

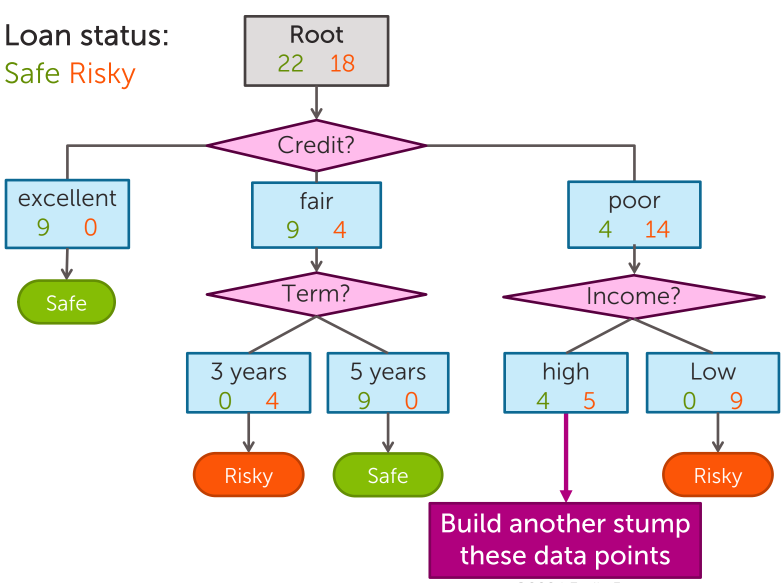
Tree learning = Recursive stump learning

Loan status:
Safe Risky

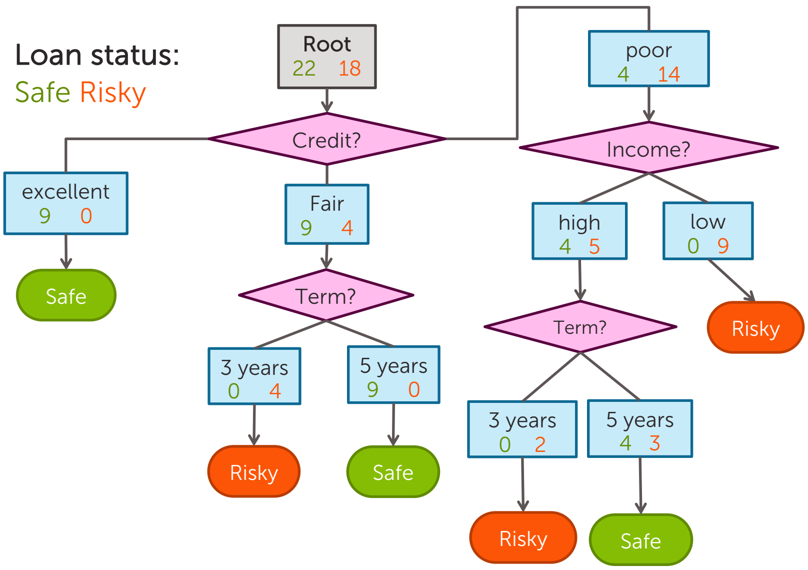


Second level

Loan status:
Safe Risky



Final decision tree



Simple greedy decision tree learning

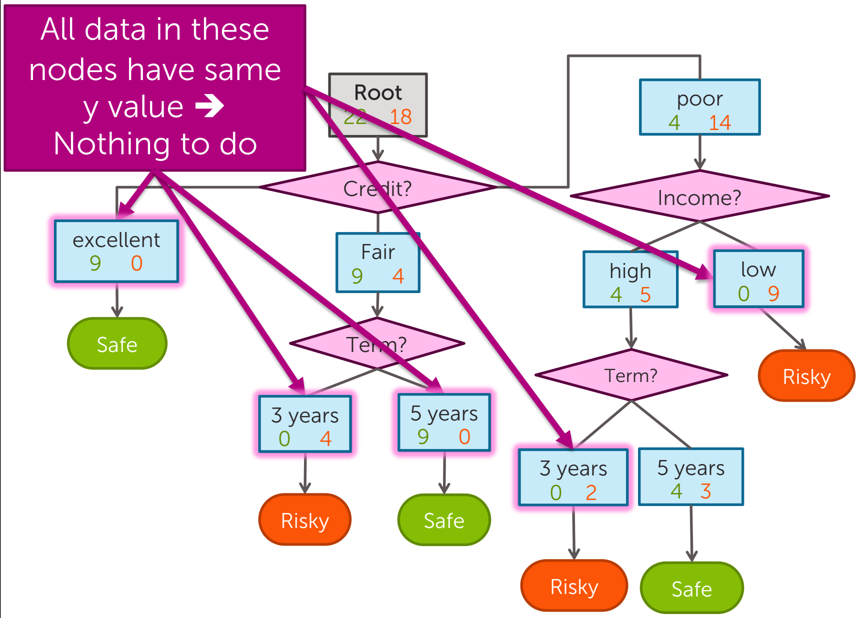
Pick best feature to split on

Learn decision stump with this split

For each leaf of decision stump, recurse

When do we stop???

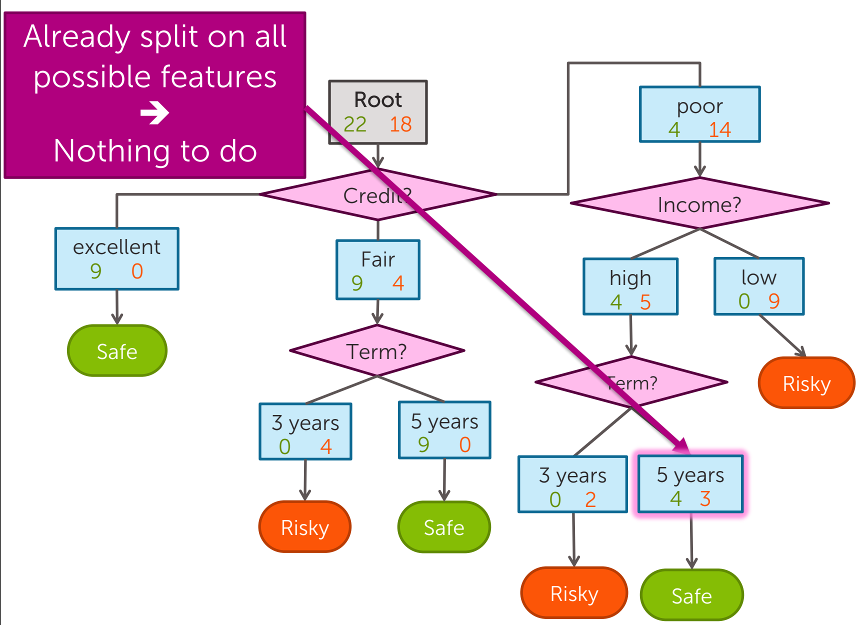
Stopping condition 1: All data agrees on y



 CS 229: Machine Learning

41

Stopping condition 2: Already split on all features



 CS 229: Machine Learning

42

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

43

Is this a good idea?

Proposed stopping condition 3:
Stop if no split reduces the classification error

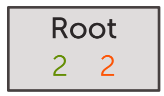
44

Stopping condition 3:
Stop if error doesn't decrease???

$y = \mathbf{x}[1]$ xor $\mathbf{x}[2]$

x[1]	**x**[2]	y
False	False	False
False	True	True
True	False	True
True	True	False

y values
True False



Error = $\frac{2}{4}$

= 0.5

Tree	Classification error
(root)	0.5

 CS 229: Machine Learning

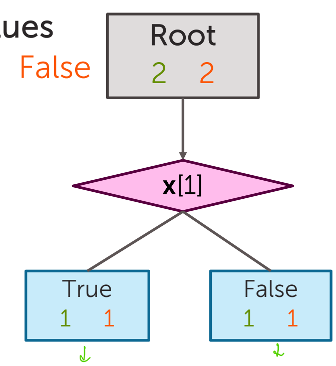
45

Consider split on **x**[1]

$y = \mathbf{x}[1]$ xor $\mathbf{x}[2]$

x[1]	**x**[2]	y
False	False	False
False	True	True
True	False	True
True	True	False

y values
True False



Error = $\frac{1+1}{4}$

= 0.5

Tree	Classification error
(root)	0.5
Split on **x**[1]	0.5

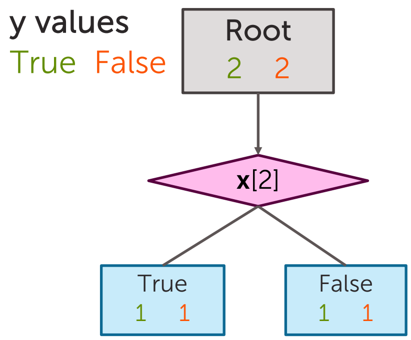
 CS 229: Machine Learning

46

Consider split on **x**[2]

y = **x**[1] xor **x**[2]

x[1]	**x**[2]	y
False	False	False
False	True	True
True	False	True
True	True	False



$$\text{Error} = \frac{1+1}{2+2} = 0.5$$

Neither features improve training error… Stop now???

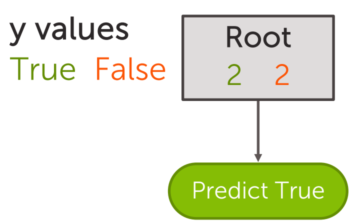
Tree	Classification error
(root)	0.5
Split on **x**[1]	0.5
Split on **x**[2]	0.5

47

Final tree with stopping condition 3

y = **x**[1] xor **x**[2]

x[1]	**x**[2]	y
False	False	False
False	True	True
True	False	True
True	True	False



Tree	Classification error
with stopping condition 3	0.5

48

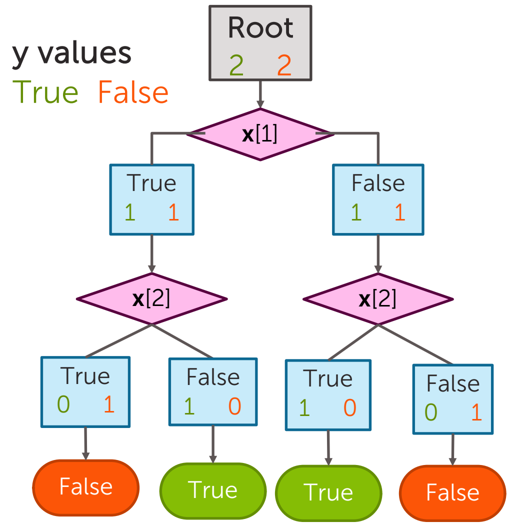
Without stopping condition 3

Condition 3 (stopping when training error doesn't't' improve) is **not** recommended!

y = **x**[1] xor **x**[2]

x[1]	**x**[2]	y
False	False	False
False	True	True
True	False	True
True	True	False

Tree	Classification error
with stopping condition 3	0.5
without stopping condition 3	0



49

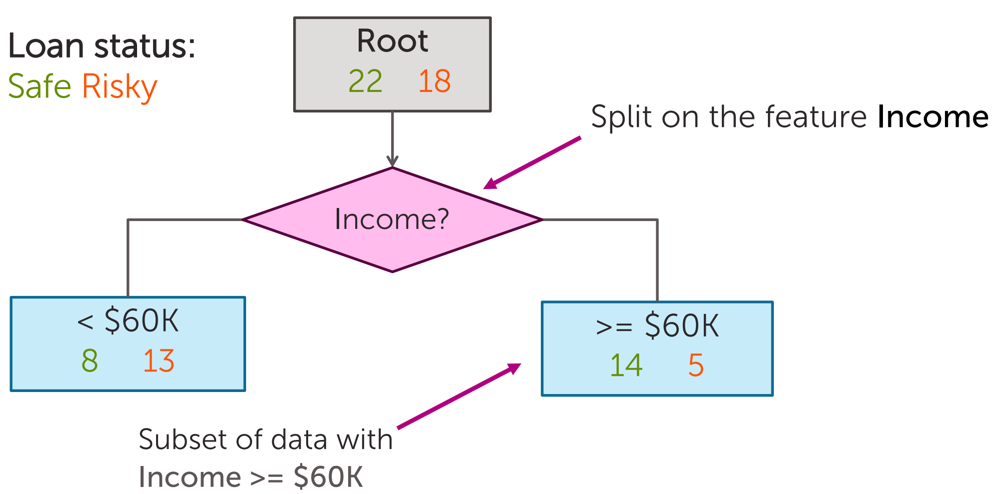
Decision tree learning:
Real valued features

50

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

Threshold split

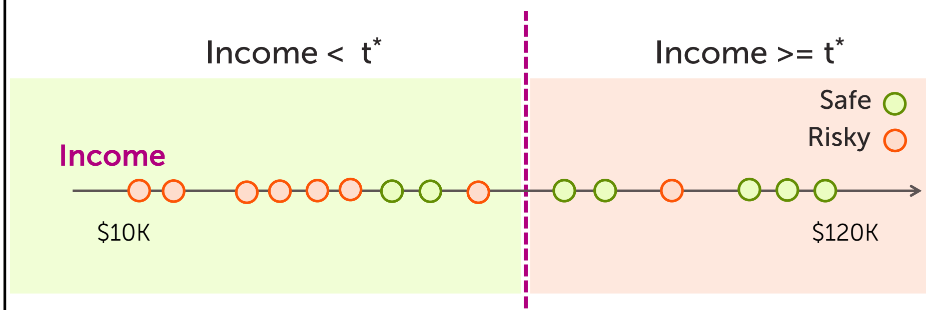


Finding the best threshold split

Infinite possible values of t

Income = t^*

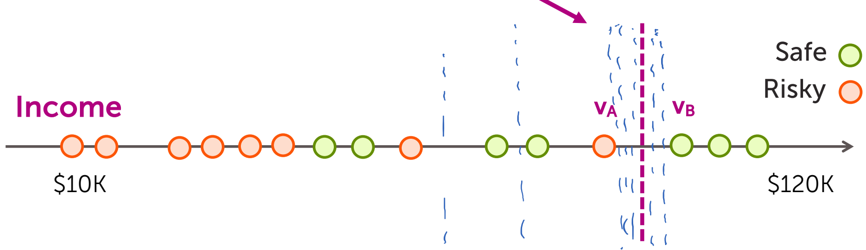
threshold to choose



53

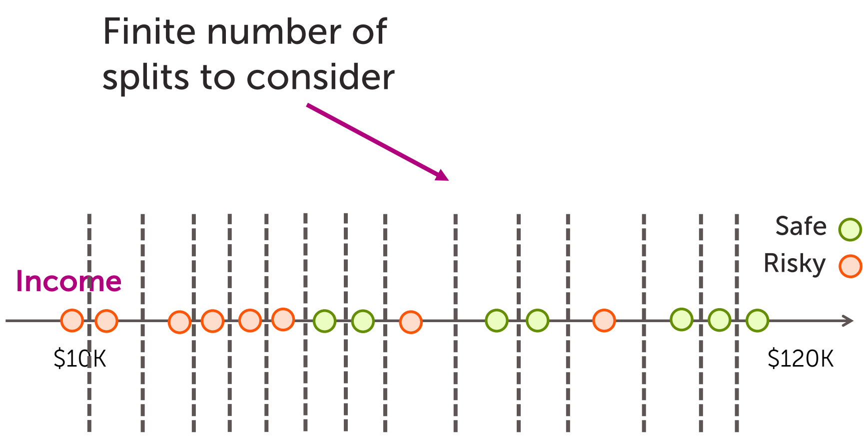
Consider a threshold between points

Same classification error for any threshold split between v_A and v_B



54

Only need to consider mid-points



 CS 229: Machine Learning

55

Threshold split selection algorithm

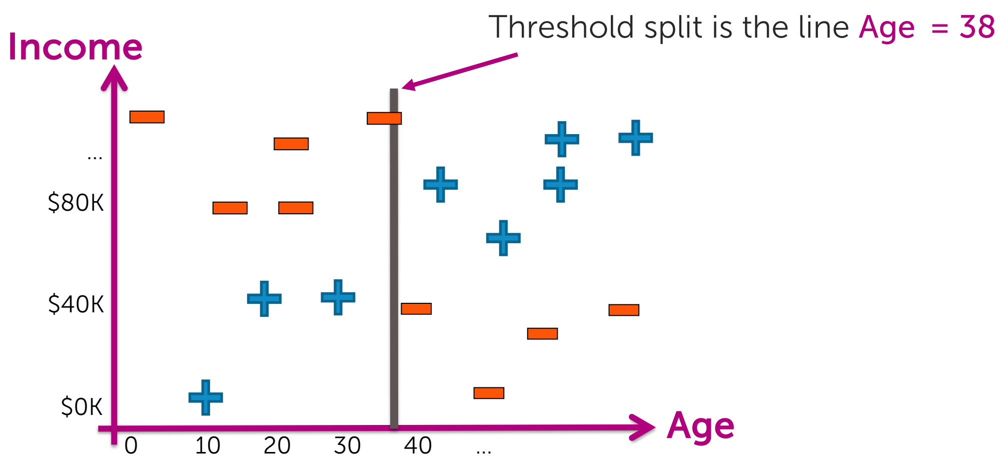
- Step 1: Sort the values of a feature $h_j(\mathbf{x})$:
  Let $\{v_1, v_2, v_3, \ldots v_N\}$ denote sorted values
- Step 2:
  - For i = 1 … N-1
    - Consider split $t_i = (v_i + v_{i+1}) / 2$ ← midpoint
    - Compute classification error for treshold split $h_j(\mathbf{x}) >= \mathbf{t}_i$
  - Chose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

	$h_1(x)$	$h_2(x)$	$h_3(x)$	...	$h_{10}(x)$
t^*	39 yrs	\$60k			
error	0.7	0.4			

 CS 229: Machine Learning

56

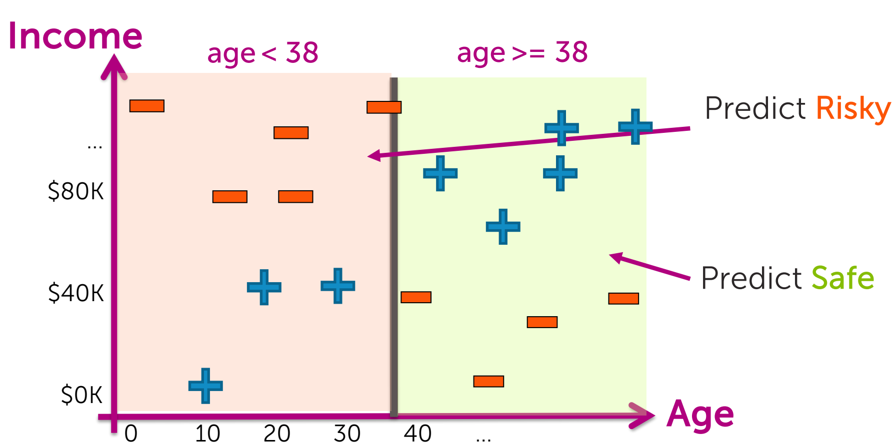
Visualizing the threshold split



CS 229: Machine Learning

57

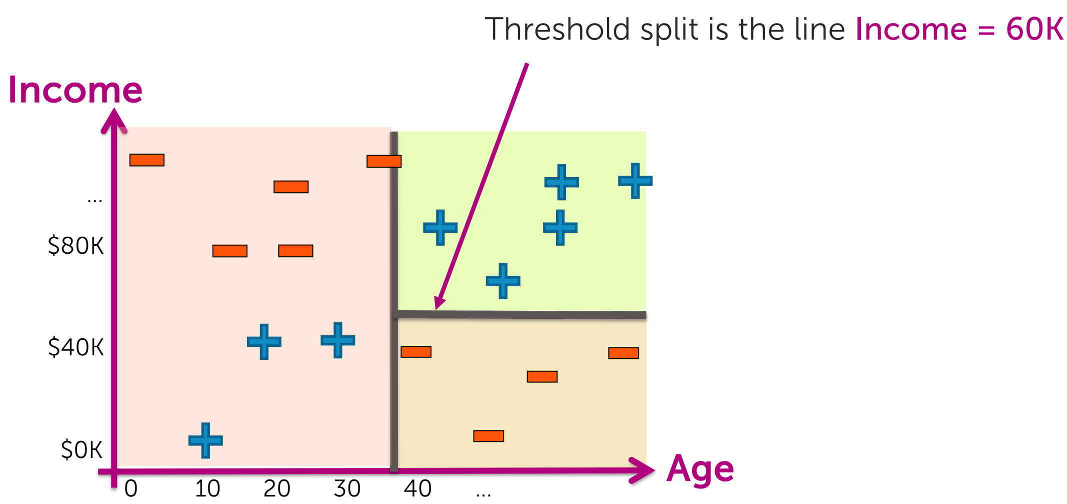
Split on Age >= 38



CS 229: Machine Learning

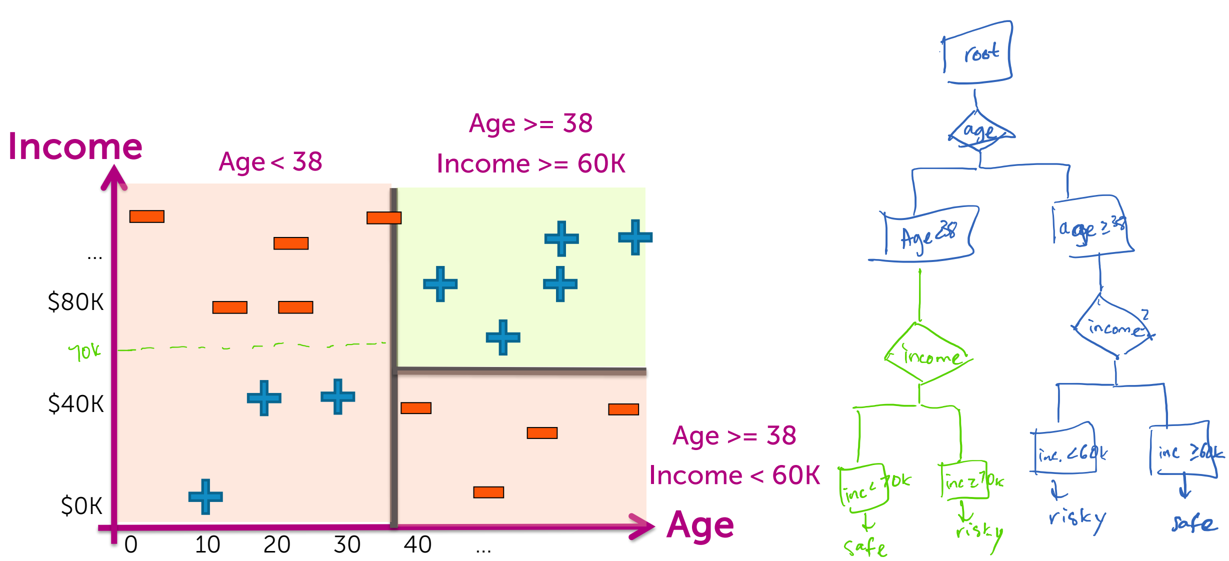
58

Depth 2: Split on Income >= $60K



59

Each split partitions the 2-D space



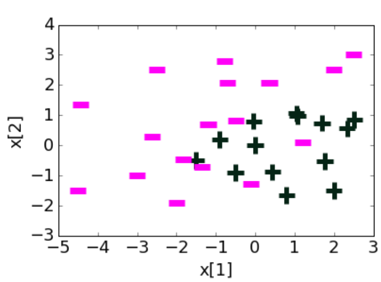
60

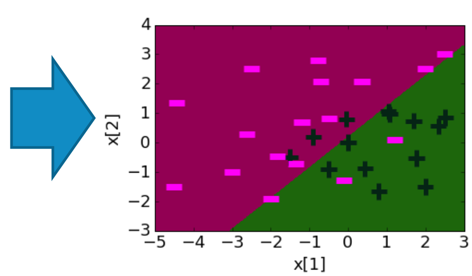
Decision trees vs logistic regression:
Example

61

Logistic regression

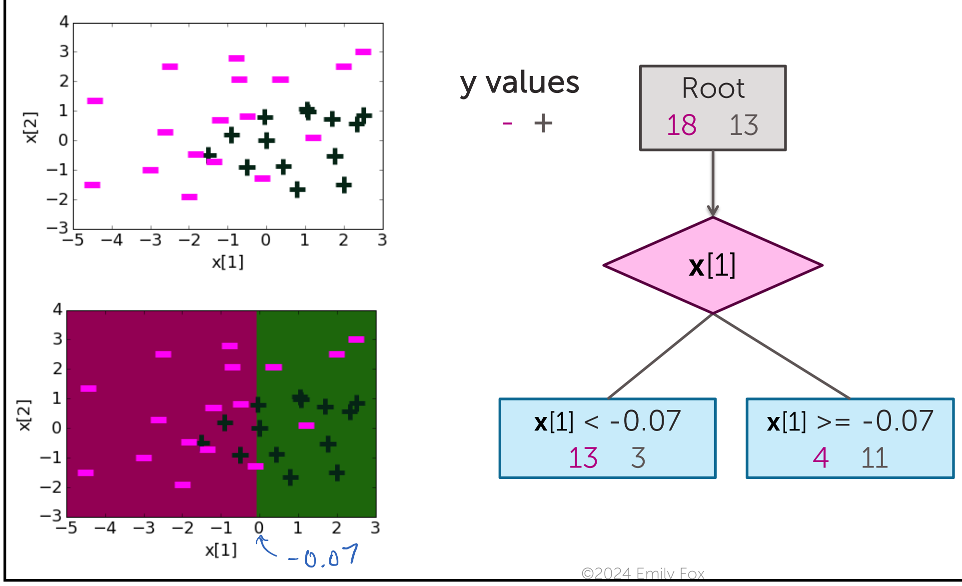
Feature	Value	Weight Learned
$h_0(\mathbf{x})$	1	0.22
$h_1(\mathbf{x})$	$\mathbf{x}[1]$	1.12
$h_2(\mathbf{x})$	$\mathbf{x}[2]$	-1.07





62

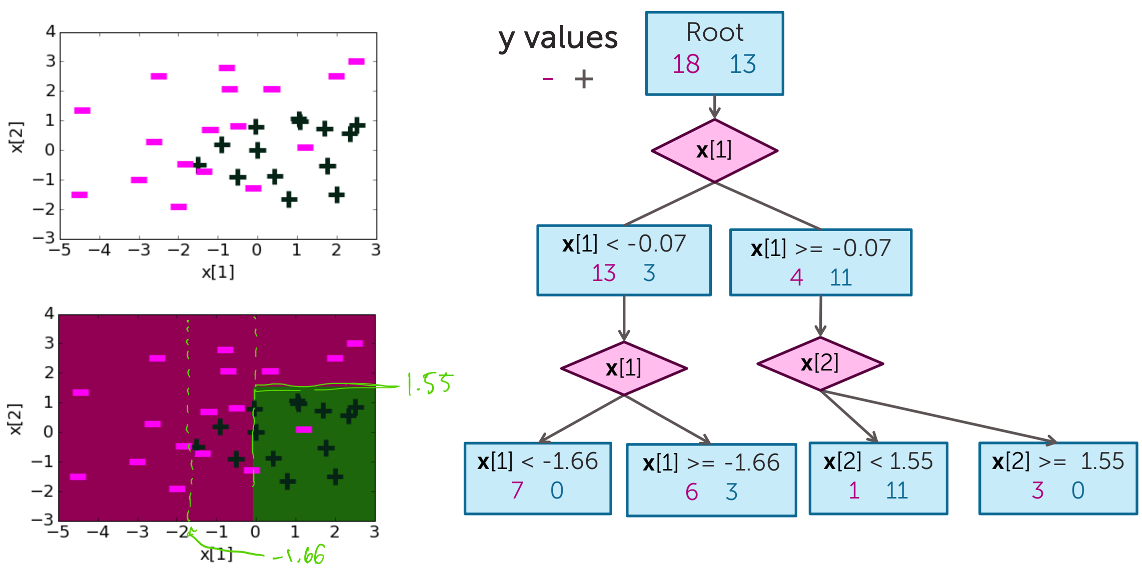
Depth 1: Split on **x**[1]



 CS 229: Machine Learning

63

Depth 2

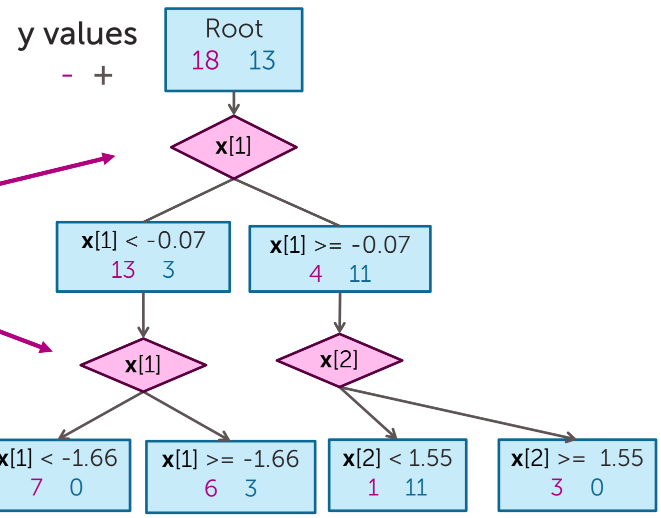


 CS 229: Machine Learning

64

Threshold split caveat

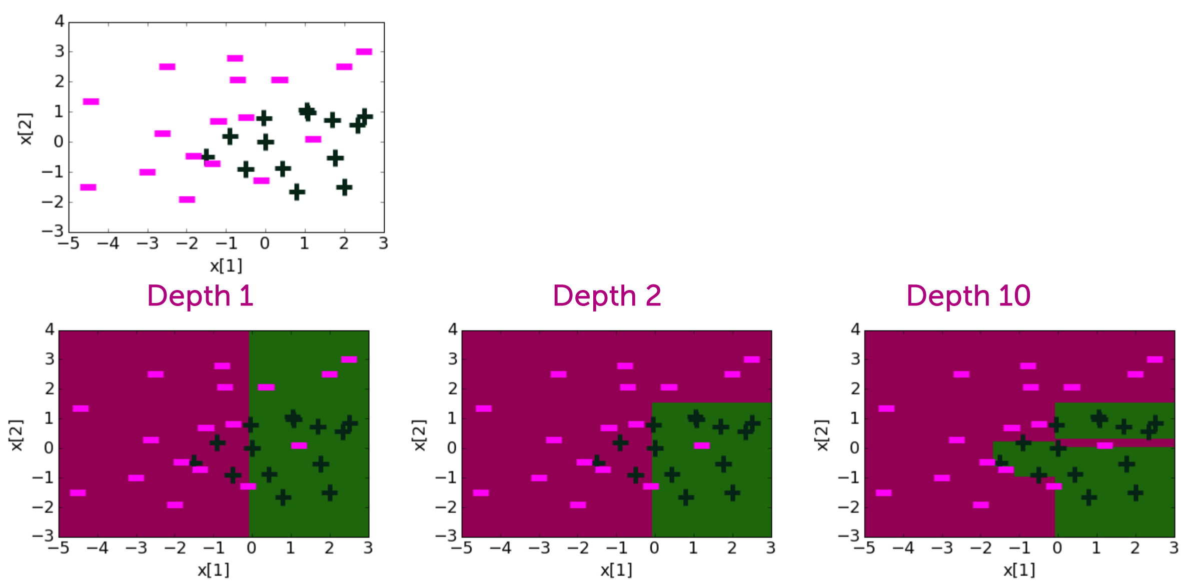
For threshold splits, same feature can be used multiple times



 CS 229: Machine Learning

65

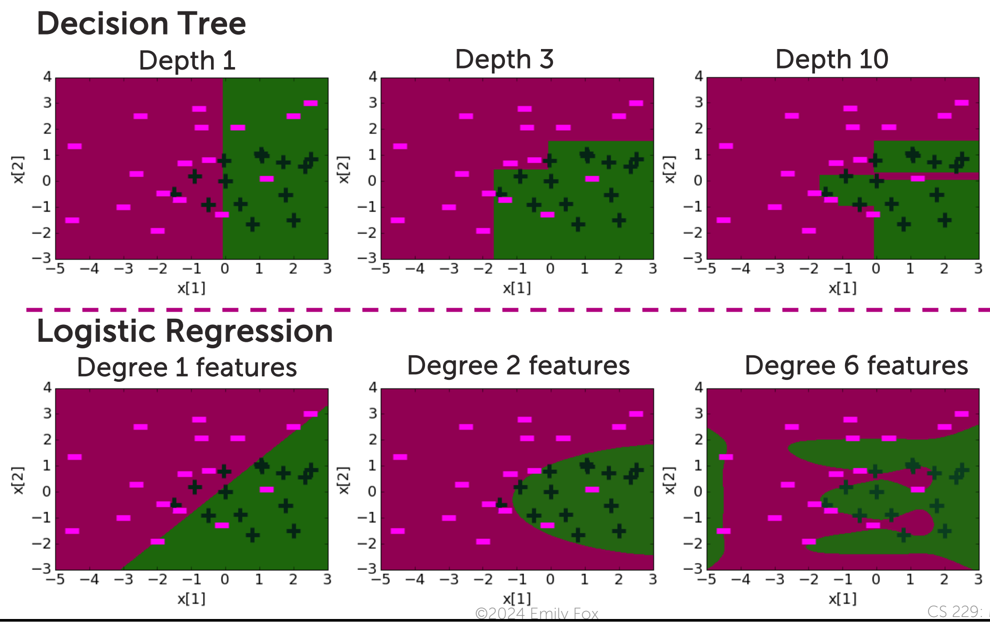
Decision boundaries



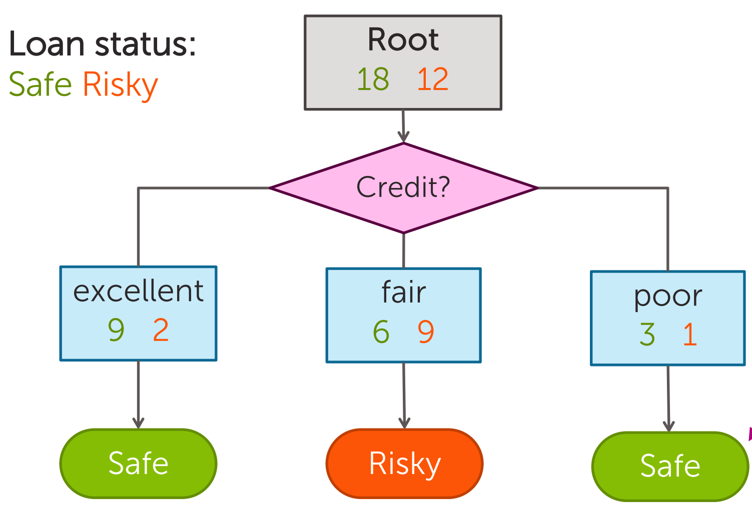
 CS 229: Machine Learning

66

Comparing decision boundaries

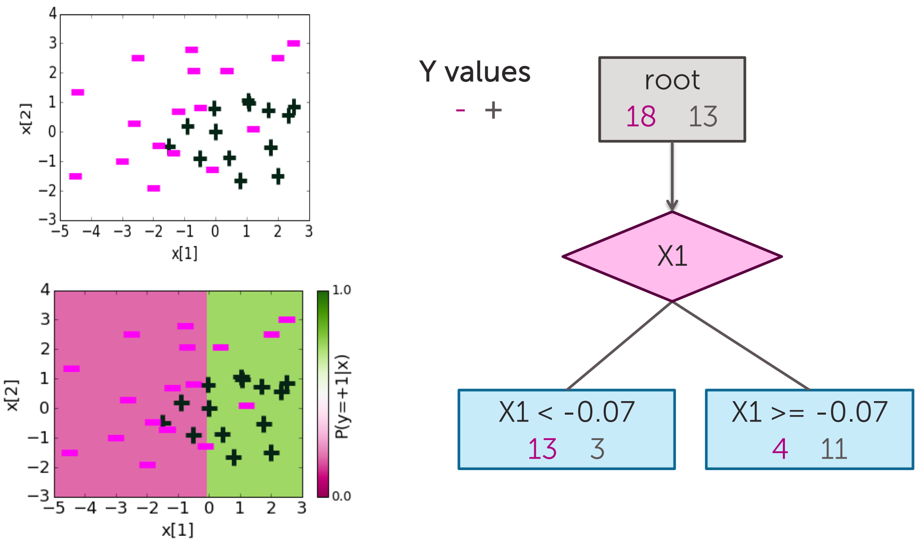


Predicting probabilities with decision trees



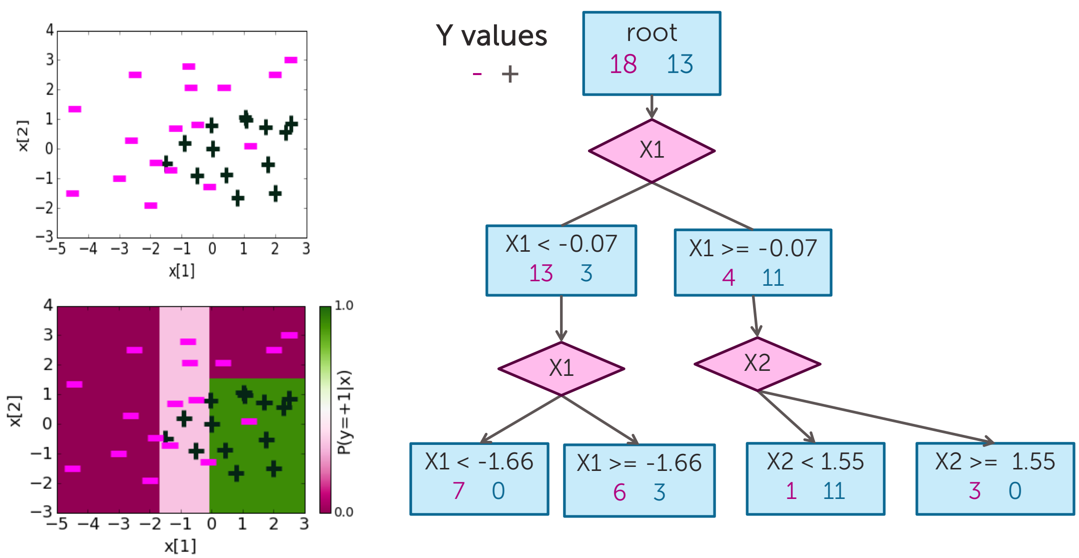
$$P(y = \text{Safe} \mid \mathbf{x}) = \frac{3}{3+1} = 0.75$$

Depth 1 probabilities



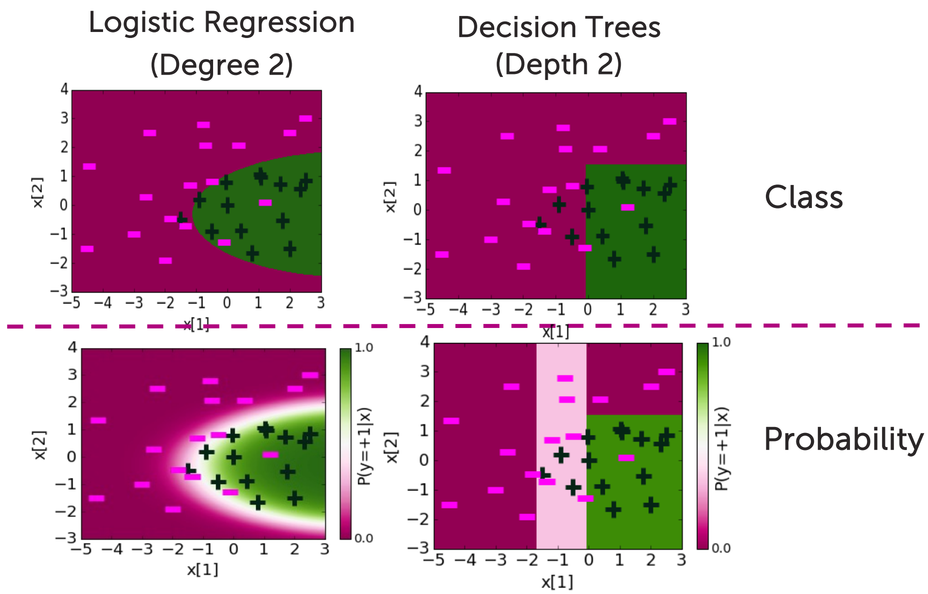
69

Depth 2 probabilities



70

Comparison with logistic regression



71

Summary of decision trees

72

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

73

Decision Trees: Overfitting

CS 229: Machine Learning
Emily Fox
Stanford University
February 12, 2024

Slides include content developed by and co-developed with Carlos Guestrin

74

Overfitting in decision trees

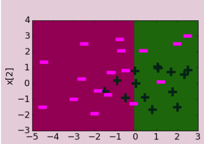
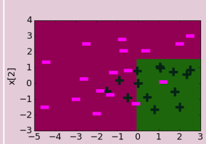
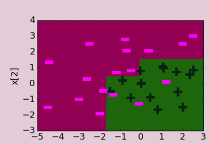
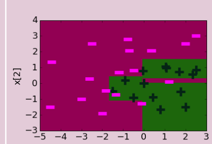
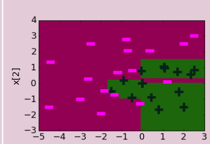
 CS 229: Machine Learning

75

What happens when we increase depth?

Training error reduces with depth

big warning!

Tree depth	depth = 1	depth = 2	depth = 3	depth = 5	depth = 10
Training error	0.22	0.13	0.10	0.03	0.00
Decision boundary					

 CS 229: Machine Learning

76

Two approaches to picking simpler trees

1. **Early Stopping:**
   Stop the learning algorithm **before** tree becomes too complex

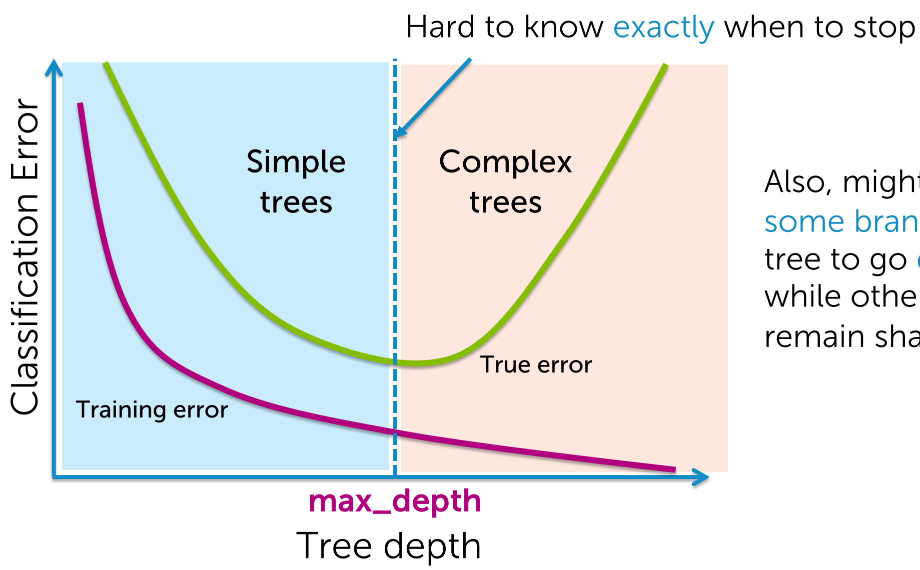
2. **Pruning:**
   Simplify the tree **after** the learning algorithm terminates

Technique 1: Early stopping

- Stopping conditions (recap):
  1. All examples have the same target value
  2. No more features to split on

- Early stopping conditions:
  1. Limit tree depth (choose *max_depth* using validation set)
  2. Do not consider splits that do not cause a sufficient decrease in classification error
  3. Do not split an intermediate node which contains too few data points

Challenge with early stopping condition 1



Also, might want some branches of tree to go deeper while others remain shallow

Early stopping condition 2: Pros and Cons

- Pros:
  - A reasonable heuristic for early stopping to avoid useless splits
- Cons:
  - Too short sighted: We may miss out on "good" splits may occur right after "useless" splits
  - Saw this with "xor" example

Two approaches to picking simpler trees

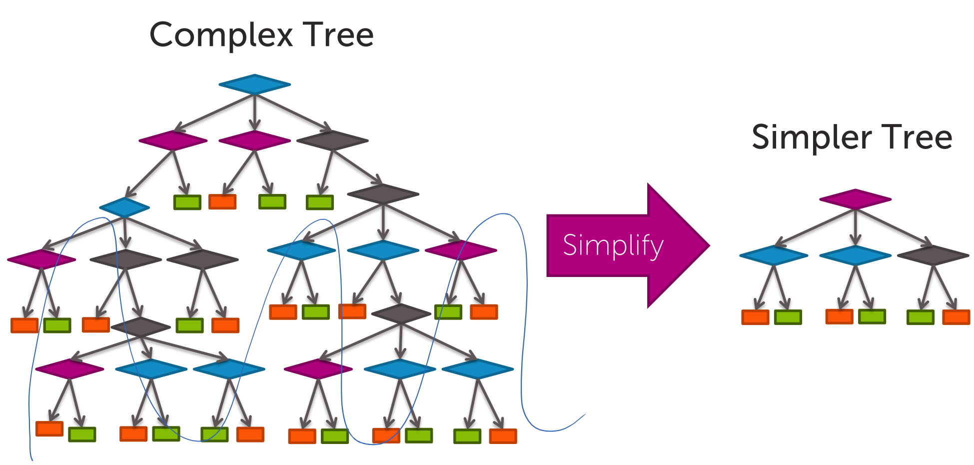
1. **Early Stopping:**
   Stop the learning algorithm **before** tree becomes too complex

2. **Pruning:**
   Simplify the tree **after** the learning algorithm terminates

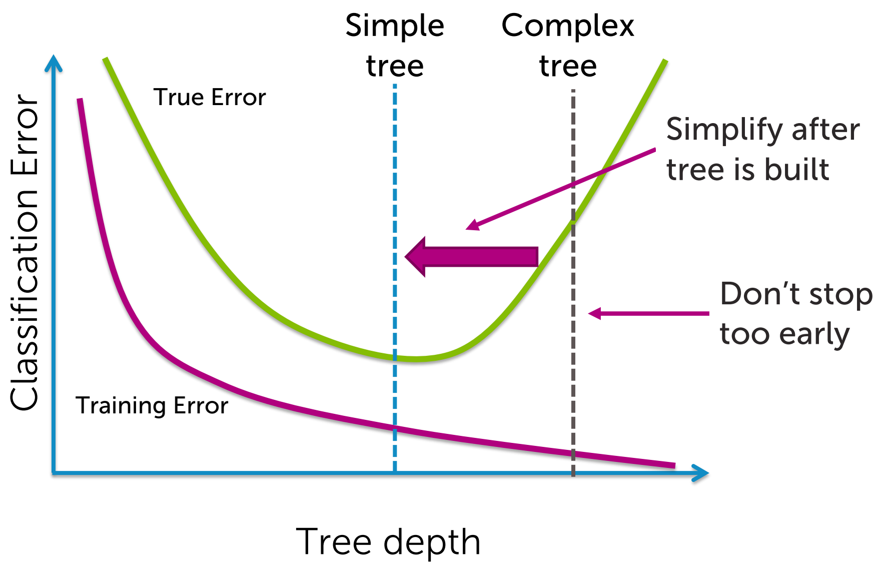
Complements early stopping

Pruning: *Intuition*

Train a complex tree, simplify later



Pruning motivation



 CS 229: Machine Learning

83

Scoring trees: Desired total quality format

Want to balance:

i. How well tree fits data

ii. Complexity of tree

want to balance

Total cost =

measure of fit + measure of complexity

(classification error)
Large # = bad fit to training data

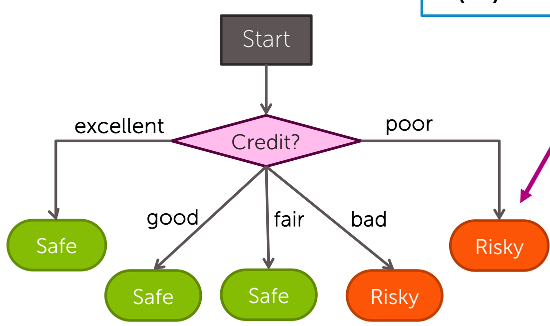
Large # = likely to overfit

 CS 229: Machine Learning

84

Simple measure of complexity of tree

L(T) = # of leaf nodes



85

Balance simplicity & predictive power

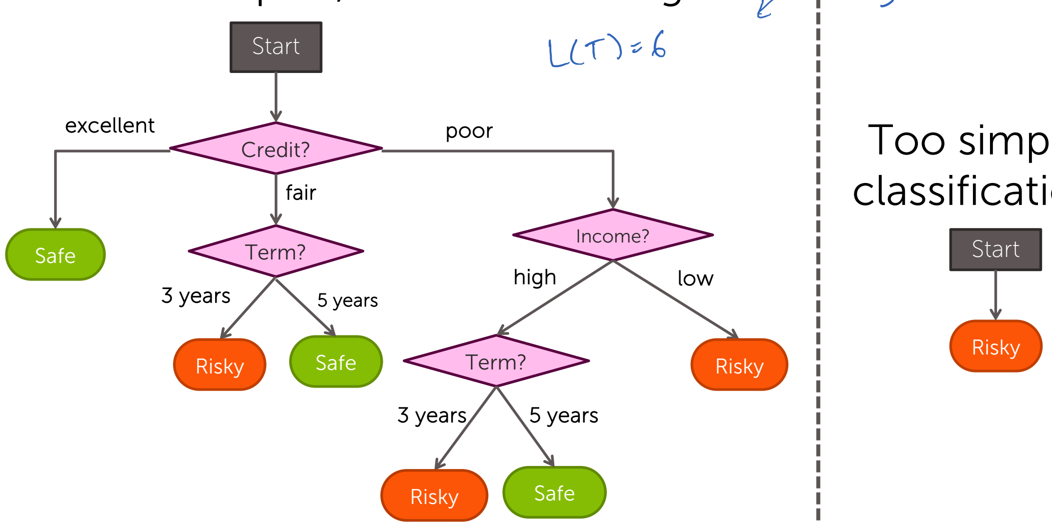
Too complex, risk of overfitting

L(T) = 6

solution in between

L(T) = 1

Too simple, high classification error



86

Balancing fit and complexity

Total cost $C(\mathbf{T}) = \text{Error}(\mathbf{T}) + \lambda L(\mathbf{T})$

tuning parameter

If $\lambda=0$: standard decision tree learning

If $\lambda=\infty$: ∞ penalty $\rightarrow$ root $\quad \hat{y}$ = majority vote (of all training data)

If λ in between: balance of fit + complexity

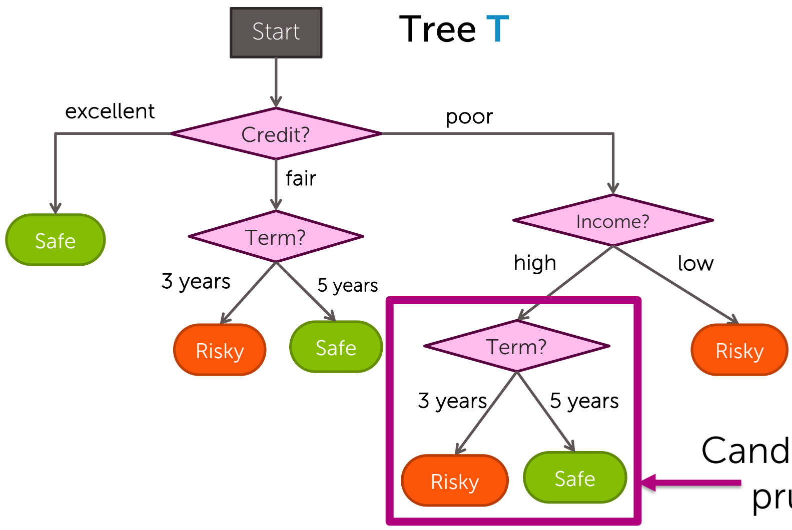
87

Tree pruning algorithm

88

Step 1: Consider a split

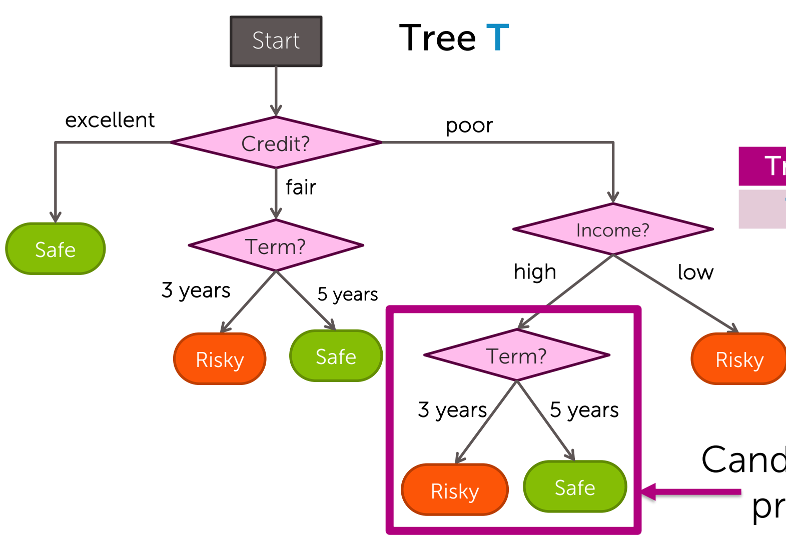
Tree **T**



Candidate for pruning

Step 2: Compute total cost C(T) of split

Tree **T**



Candidate for pruning

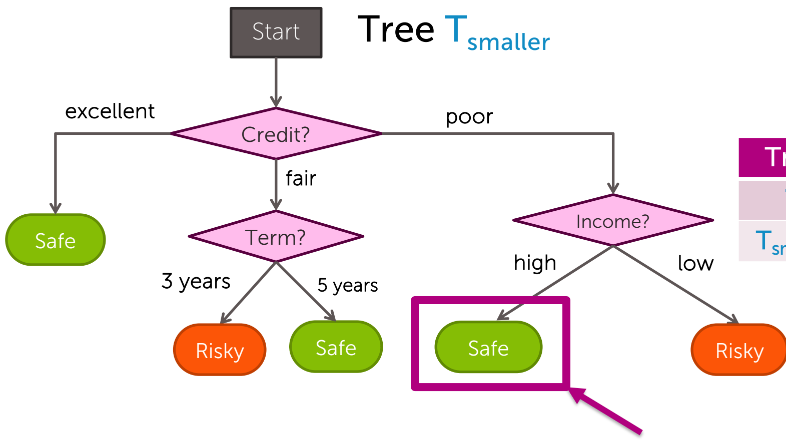
0.25
+0.03(6)

$\lambda = 0.03$

Tree	Error	#Leaves	Total
T	0.25	6	0.43

$$C(T) = \text{Error}(T) + \lambda\, L(T)$$

Step 2: "Undo" the splits on Tsmaller



$\lambda = 0.03$

Tree	Error	#Leaves	Total
T	0.25	6	0.43
$T_{smaller}$	0.26	5	0.41

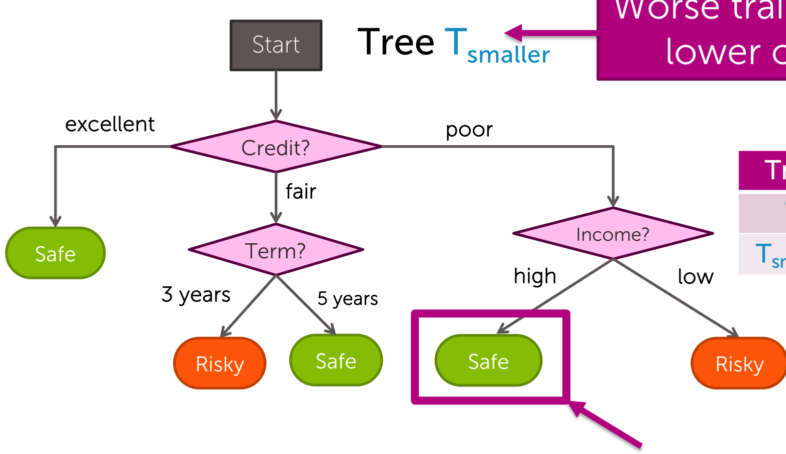
$C(T) = Error(T) + \lambda\, L(T)$

Replace split
by leaf node?

91

Prune if total cost is lower: $C(T_{smaller}) \leq C(T)$

Worse training error but lower overall cost



$\lambda = 0.03$

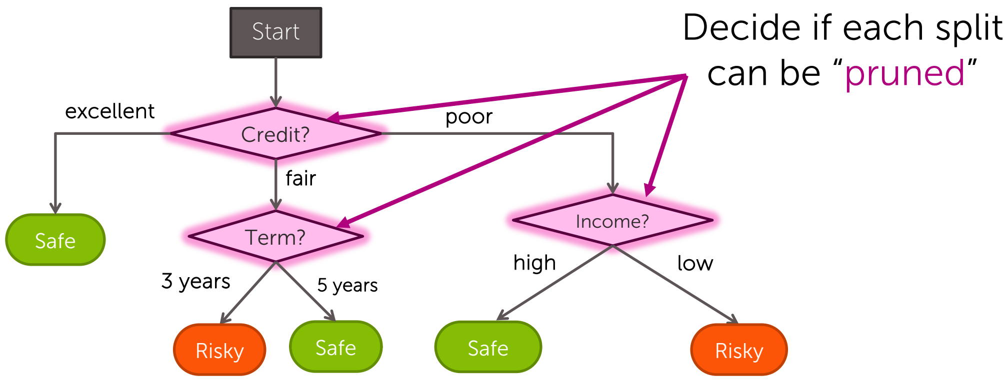
Tree	Error	#Leaves	Total
T	0.25	6	0.43
$T_{smaller}$	0.26	5	0.41

$C(T) = Error(T) + \lambda\, L(T)$

Replace split
by leaf node? ← YES!

92

Step 5: Repeat Steps 1-4 for every split



93

How to choose hyperparameters?
(e.g., λ or max_depth) ← tuning params

validation error on valid. set
or
Cross validations

94

Summary of overfitting in decision trees

What you can do now…

- Identify when overfitting in decision trees
- Prevent overfitting with early stopping
  - Limit tree depth
  - Do not consider splits that do not reduce classification error
  - Do not split intermediate nodes with only few points
- Prevent overfitting by pruning complex trees
  - Use a total cost formula that balances classification error and tree complexity
  - Use total cost to merge potentially complex trees into simpler ones