Linear classifiers:

Handling overfitting, categorical inputs, & multiple classes

STAT/CSE 416: Machine Learning

Emily Fox
University of Washington

April 24, 2018

Encoding categorical inputs

Categorical inputs

- Numeric inputs:
  - #awesome, age, salary,…
  - Intuitive when multiplied by coefficient
    - e.g., 1.5 #awesome
- Categorical inputs:

Numeric value, but should be interpreted as category
(98195 not about 9x larger than 10005)

Gender
(Male, Female,…)

Country of birth
(Argentina, Brazil, USA,…)

Zipcode
(10005, 98195,…)

How do we multiply category by coefficient???
Must convert categorical inputs into numeric features

Encoding categories as numeric features

x =

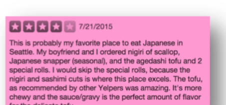
Country of birth
(Argentina, Brazil, USA,…)
196 categories

1-hot encoding

	Argentina	Brazil			Zimbabwe
x	$h_1(x)$	$h_2(x)$	…	$h_{195}(x)$	$h_{196}(x)$
Brazil	0	1	0	0	0
Zimbabwe	0	0	0	0	1

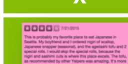
196 features

x =



Restaurant review
(Text data)
10,000 words in vocabulary

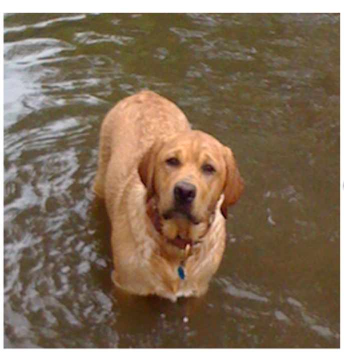
Bag of words

	#awesome	#awful			#sushi
x	$h_1(x)$	$h_2(x)$	…	$h_{9999}(x)$	$h_{10000}(x)$
	2	0	0	0	3

10,000 features

Multiclass classification using 1 versus all

Multiclass classification



Input: **x**
Image pixels

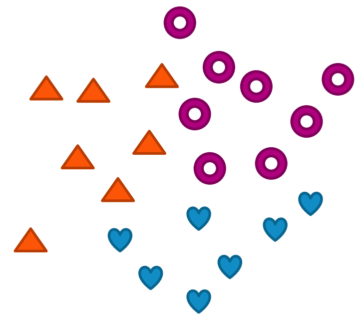
Output: y
Object in image

Multiclass classification formulation

- C possible classes:
  - y can be 1, 2,..., C

before $y \in \{-1, 1\}$

- N datapoints:



Data point	x[1]	x[2]	y
$\mathbf{x}_1, y_1$	2	1	▲
$\mathbf{x}_2, y_2$	0	2	♥
$\mathbf{x}_3, y_3$	3	3	○
$\mathbf{x}_4, y_4$	4	1	○

Learn:

$\hat{P}(y=▲\,|x)$

$\hat{P}(y=♥\,|x)$

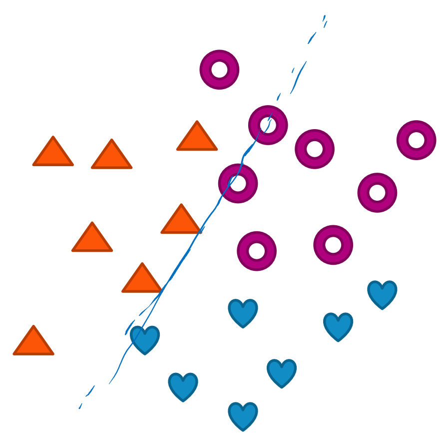
$\hat{P}(y=○\,|x)$

1 versus all:
Estimate $\hat{P}(y=▲\,|x)$ using 2-class model

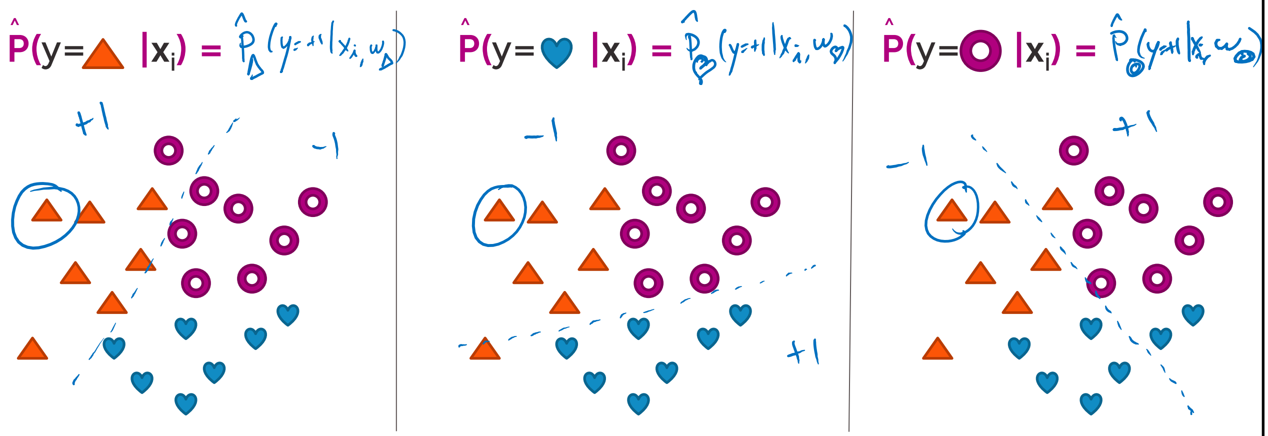
+1 class: points with y_i=▲
-1 class: points with y_i= ♥ OR ○

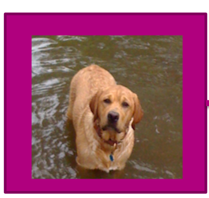
Train classifier: $\hat{P}_▲(y=+1|x)$

Predict: $\hat{P}(y=▲\,|x_i) = \hat{P}_▲(y=+1|x_i)$



1 versus all: simple multiclass classification using *C* 2-class models





Input: $\mathbf{x}_i$

Multiclass training

$\hat{P}_c(y=+1|\mathbf{x})$ = estimate of 1 vs all model for each class

Predict most likely class

max_prob = 0; $\hat{y}$ = 0

For c = 1,...,C:

    If $\hat{P}_c(y=+1|\mathbf{x}_i)$ > max_prob:

        $\hat{y}$ = c

        max_prob = $\hat{P}_c(y=+1|\mathbf{x}_i)$

Summary of overfitting in logistic regression, categorical inputs, and multiclass classification

What you can do now…

- Describe symptoms and effects of overfitting in classification
  - Identify when overfitting is happening
  - Relate large learned coefficients to overfitting
  - Describe the impact of overfitting on decision boundaries and predicted probabilities of linear classifiers
- Use regularization to mitigate overfitting
  - Motivate the form of L2 regularized logistic regression quality metric
  - Describe the use of L1 regularization to obtain sparse logistic regression solutions
  - Describe what happens to estimated coefficients as tuning parameter λ is varied
  - Interpret coefficient path plot
- Use 1-hot encoding to represent categorical inputs
- Perform multiclass classification using the 1-versus-all approach

Decision Trees

STAT/CSE 416: Machine Learning
Emily Fox
University of Washington
April 24, 2018

Predicting potential loan defaults

What makes a loan risky?

I want a 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,…

Credit History
★★★★

Income
★★★

Term
★★★★★

Personal Info
★★★

Personal information

Age, reason for the loan, marital status,…

Example: Home loan for a married couple

Credit History ★★★★

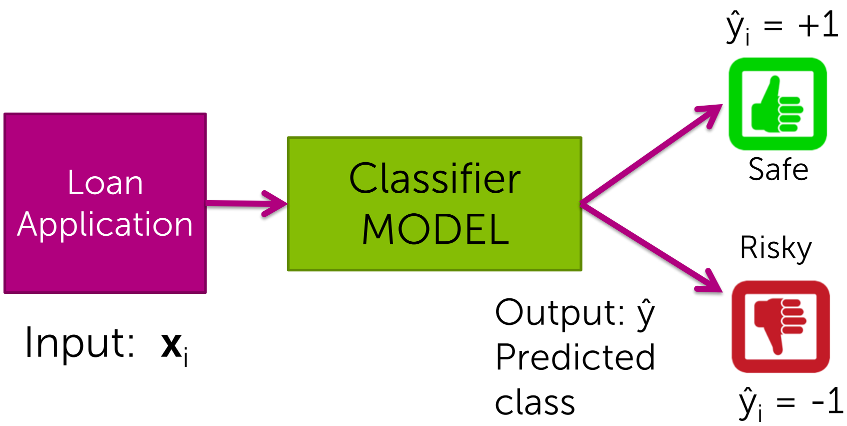
Income ★★★

Term ★★★★★

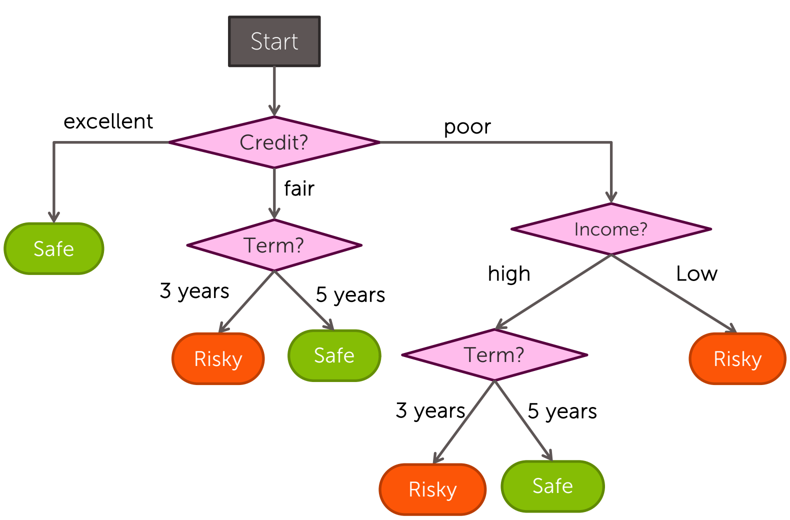
Personal Info ★★★

Intelligent application

Classifier review

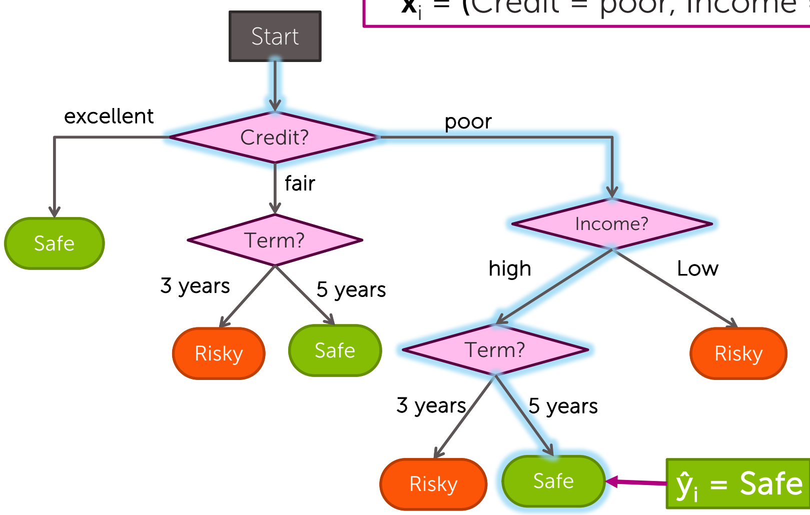


This module … decision trees



Scoring a loan application

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



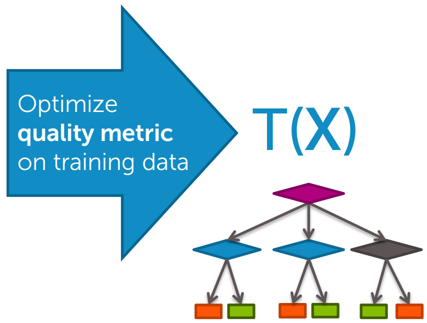
23

Decision tree learning task

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

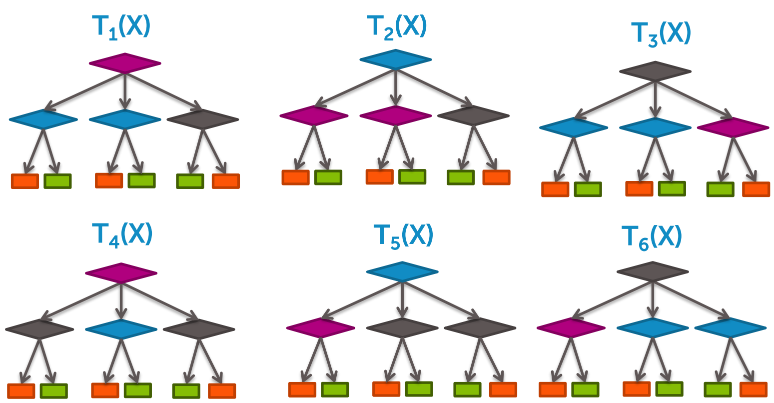
- Error measures fraction of mistakes

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

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

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

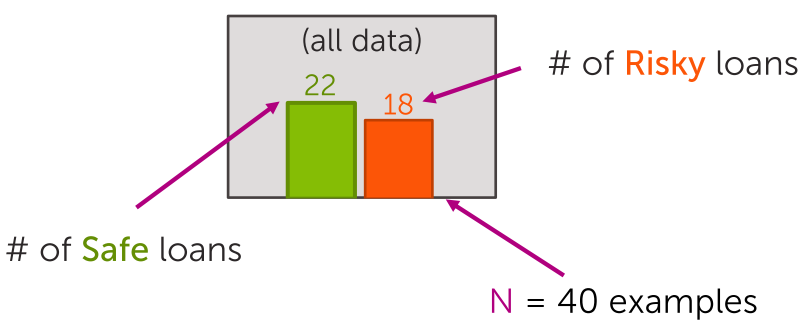
Our training data table

Assume N = 40, 3 features

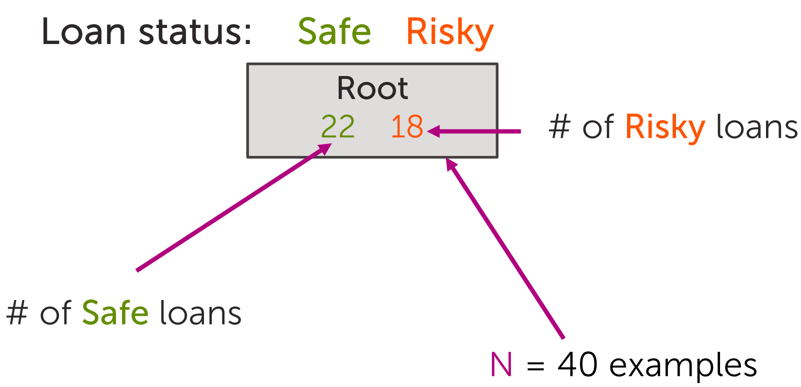
Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	risky
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe

Start with all the data

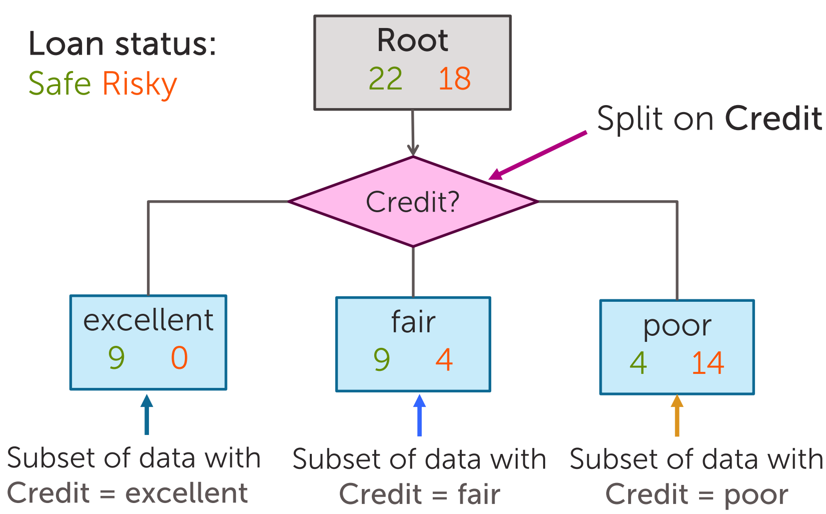
Loan status: Safe Risky



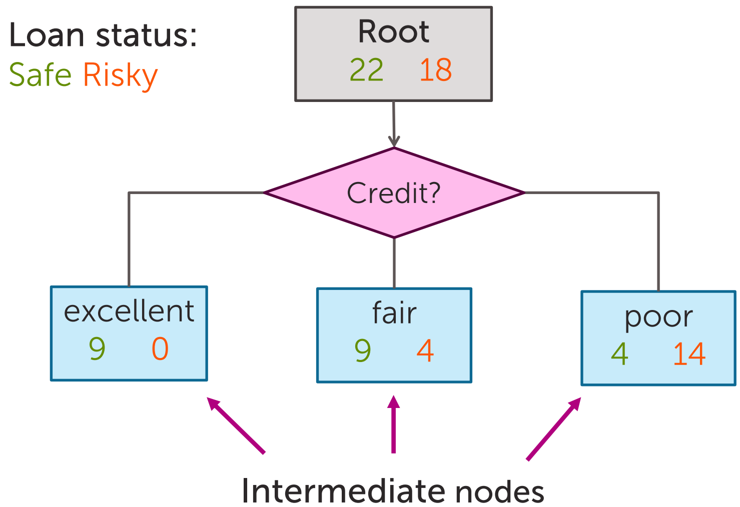
Compact visual notation: Root node



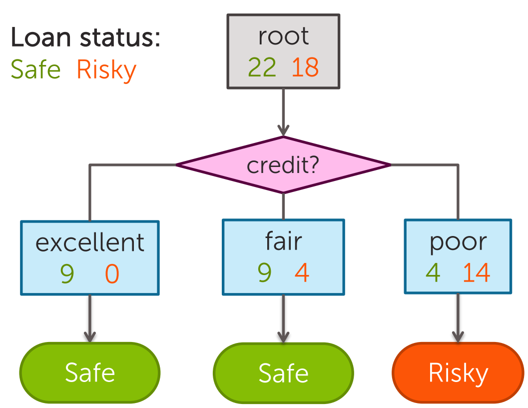
Decision stump: Single level tree



Visual notation: Intermediate nodes



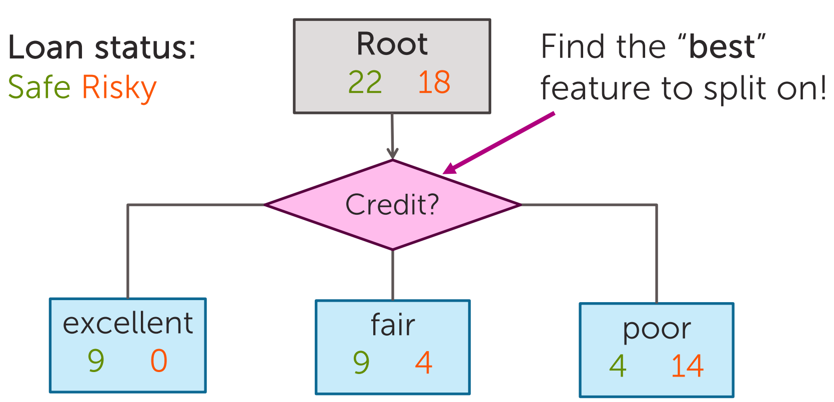
Making predictions with a decision stump



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

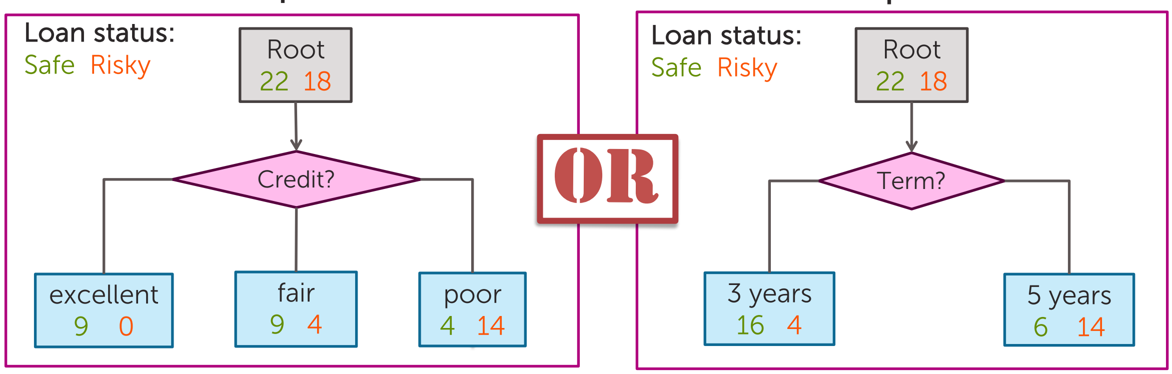
Selecting best feature to split on

How do we learn a decision stump?

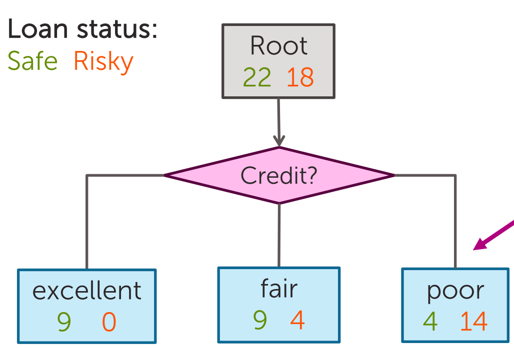


How do we select the best feature?

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



How do we measure effectiveness of a split?

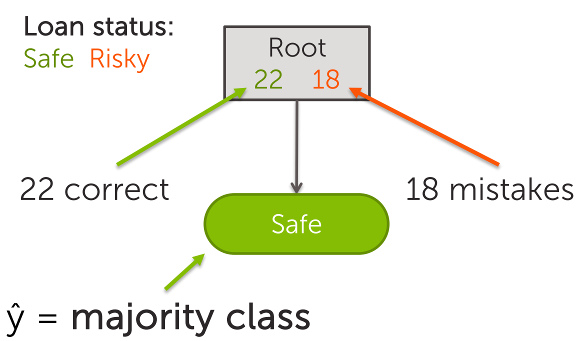


Idea: Calculate classification error of this decision stump

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

Calculating classification error

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



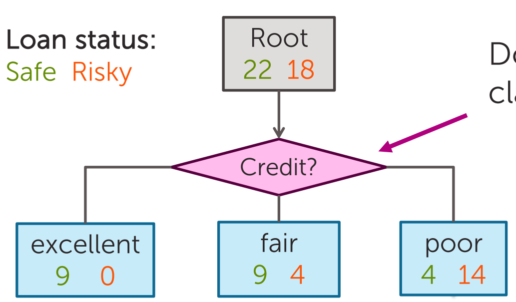
$$\text{Error} = \frac{18}{22+18}$$
$$= 0.45$$

Tree	Classification error
(root)	0.45

39 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Choice 1: Split on Credit history?

Choice 1: Split on **Credit**

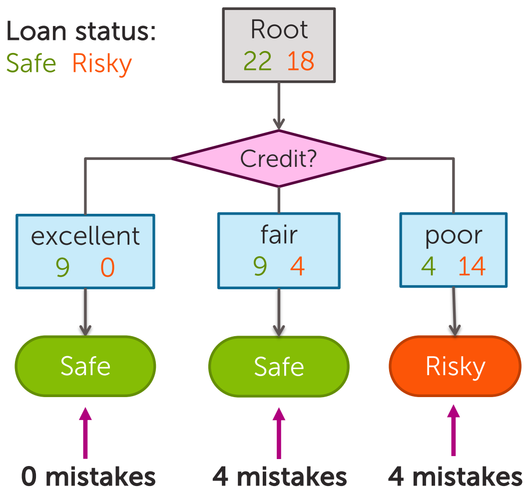


Does a **split on Credit** reduce classification error below 0.45?

40 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Split on Credit: Classification error

Choice 1: Split on Credit

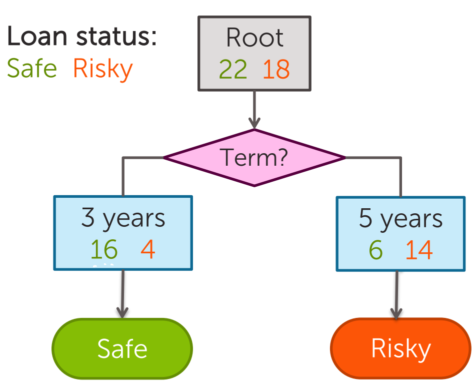


$$\text{Error} = \frac{0+4+4}{40} = 0.2$$

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

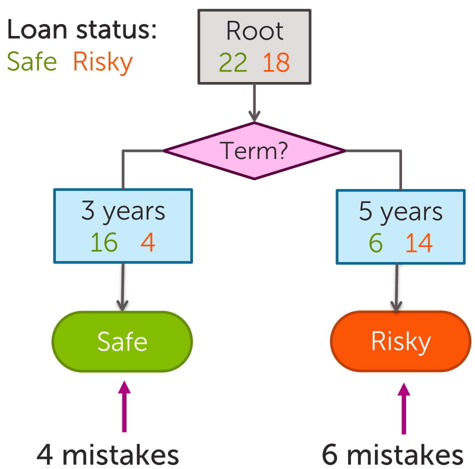
Choice 2: Split on Term?

Choice 2: Split on Term



Evaluating the split on Term

Choice 2: Split on **Term**



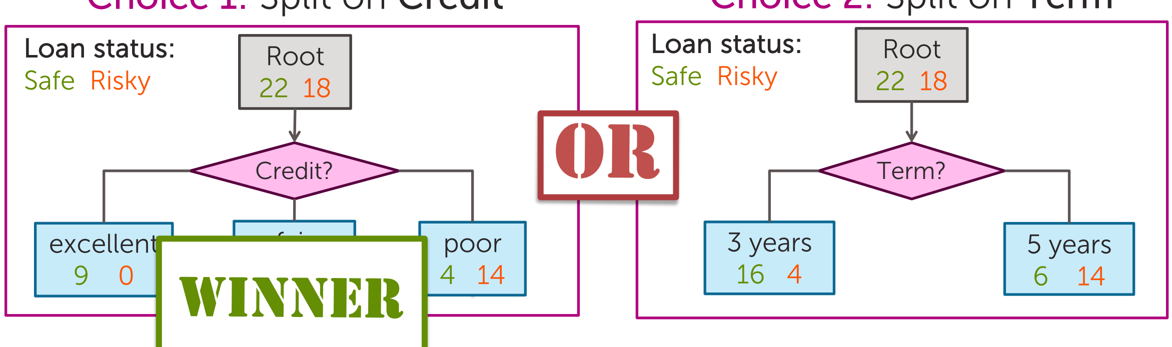
$$\text{Error} = \frac{4+6}{40} = 0.25$$

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

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**

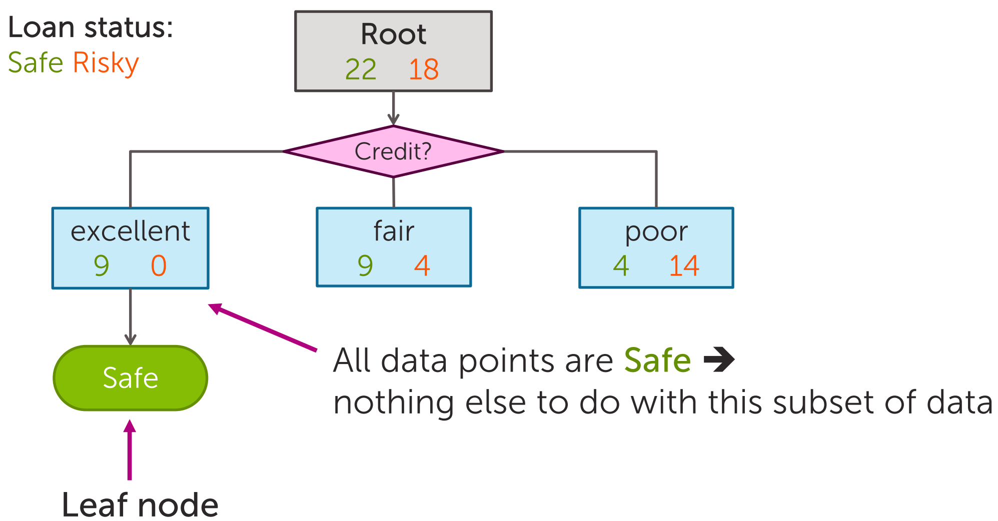


Feature split selection algorithm

- 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

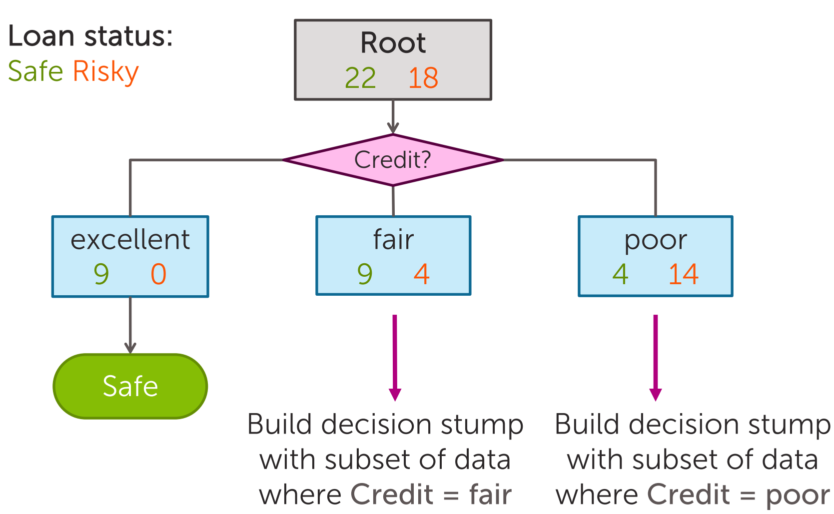
Recursion & Stopping conditions

We've learned a decision stump, what next?



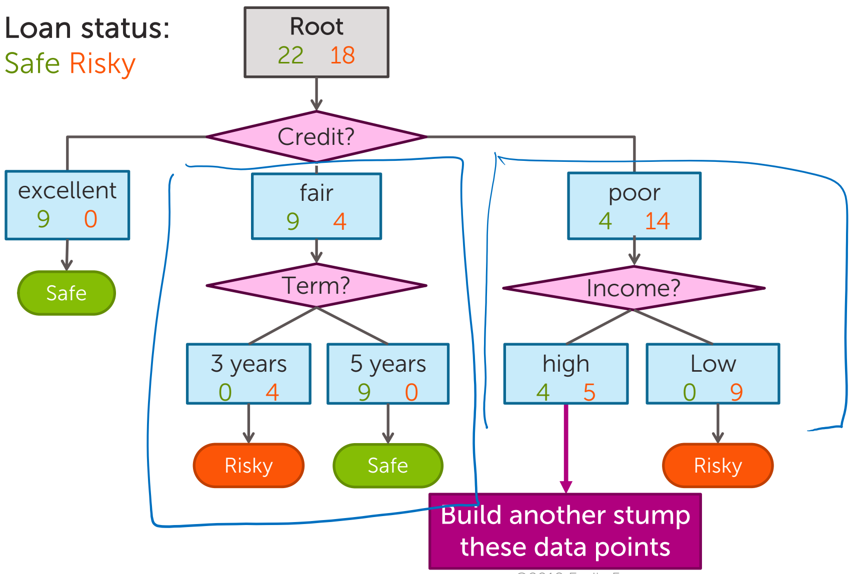
47 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Tree learning = Recursive stump learning



48 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Second level

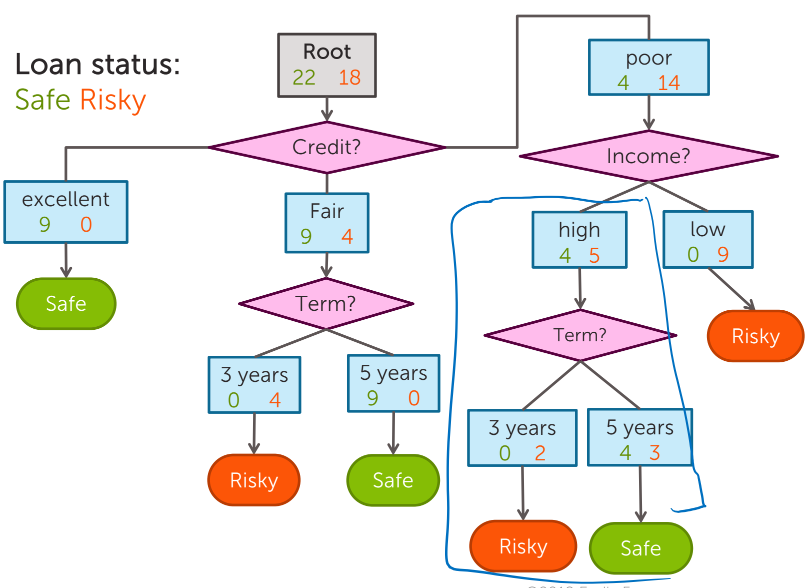


49

©2018 Emily Fox

STAT/CSE 416: Intro to Machine Learning

Final decision tree



50

©2018 Emily Fox

STAT/CSE 416: Intro to Machine Learning

Simple greedy decision tree learning

Pick best feature to split on

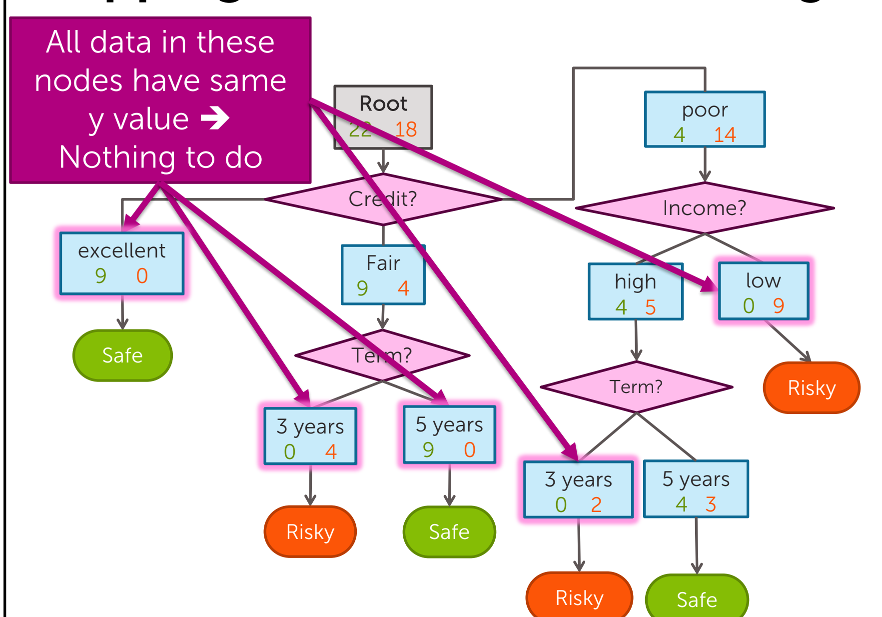
Learn decision stump with this split

For each leaf of decision stump, recurse

When do we stop???

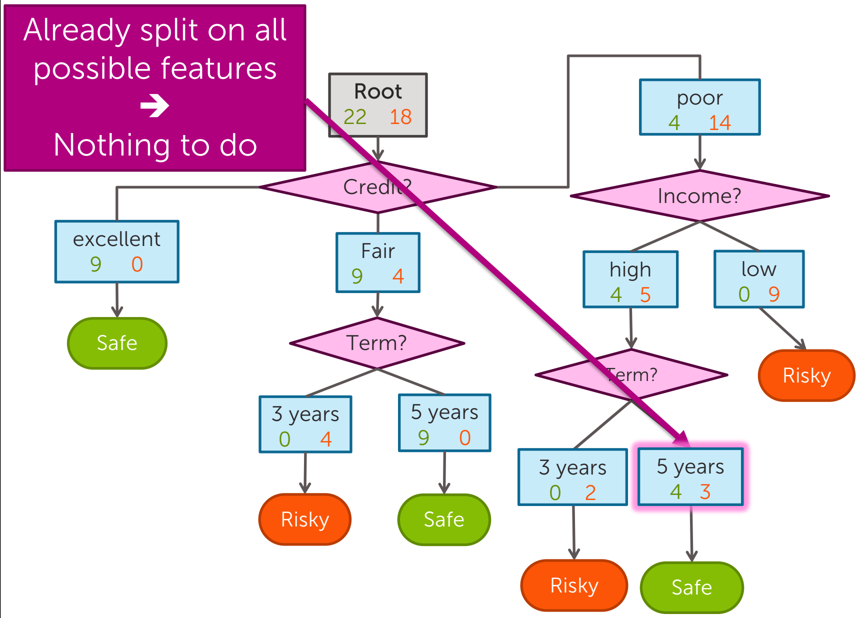
51 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Stopping condition 1: All data agrees on y



52 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Stopping condition 2: Already split on all 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

Is this a good idea?

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

Stopping condition 3:
Don't stop if error doesn't decrease???

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

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

y values
True False

Root
2 2

$\hat{y}$ = safe

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

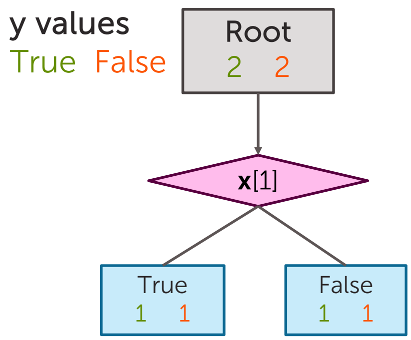
= 0.5

Tree	Classification error
(root)	0.5

Consider split on **x**[1]

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

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



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

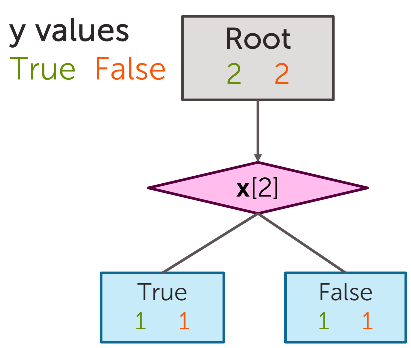
= 0.5

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

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



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

= 0.5

Neither features improve training error… Stop now???

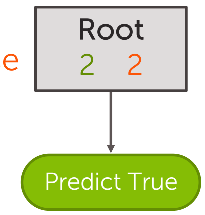
Tree	Classification error
(root)	0.5
Split on **x**[1]	0.5
Split on **x**[2]	0.5

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

y values
True False



Tree	Classification error
with stopping condition 3	0.5

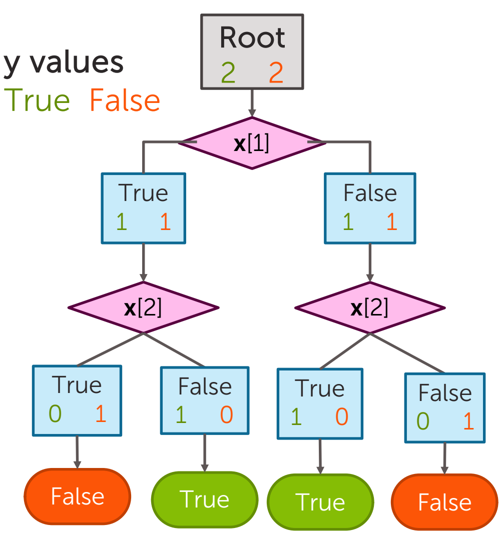
Without stopping condition 3

Condition 3 (stopping when training error doesn'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

y values
True False



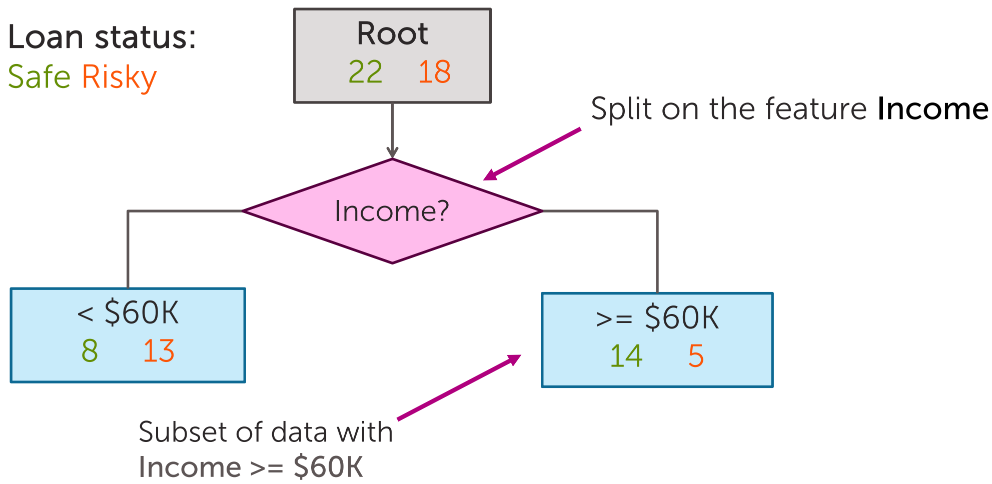
Tree	Classification error
with stopping condition 3	0.5
without stopping condition 3	0

Decision tree learning:
Real valued features

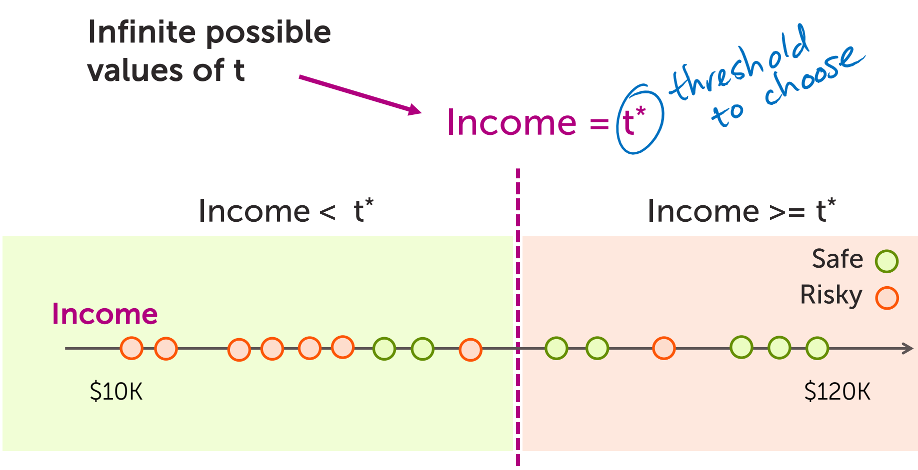
How do we use real values inputs?

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

Threshold split

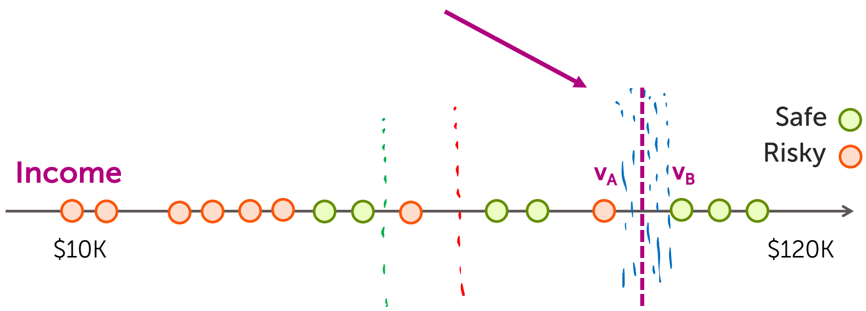


Finding the best threshold split



Consider a threshold between points

Same classification error for any threshold split between v_A and v_B



Only need to consider mid-points

Finite number of splits to consider

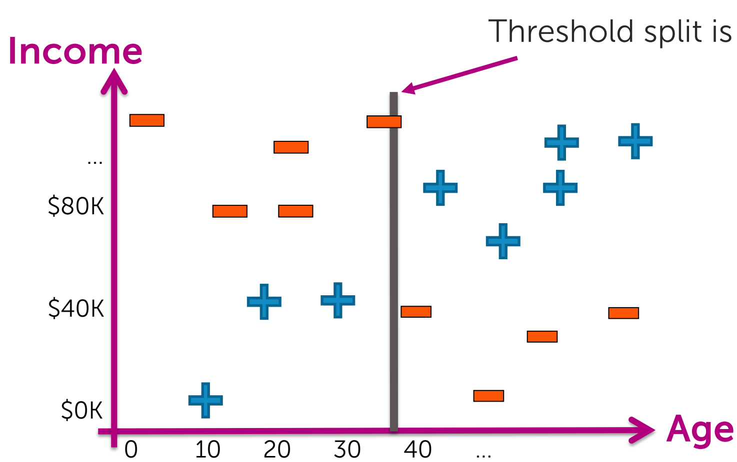
Threshold split selection algorithm

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

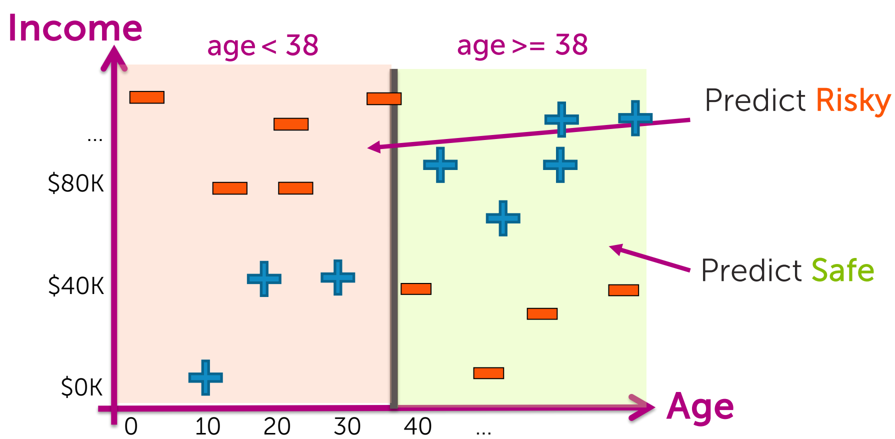
	$h_1(x)$	$h_2(x)$	. . .	$h_{10}(x)$
t^*	39 yrs	\$60 k		
error	0.7	0.4		

decision stump

Visualizing the threshold split



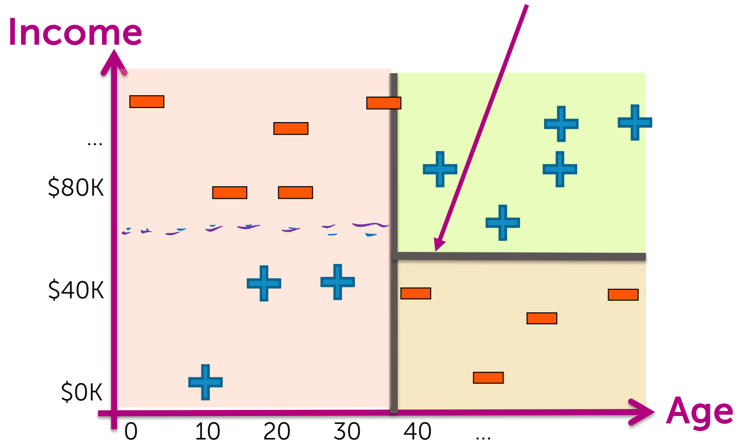
Split on Age >= 38

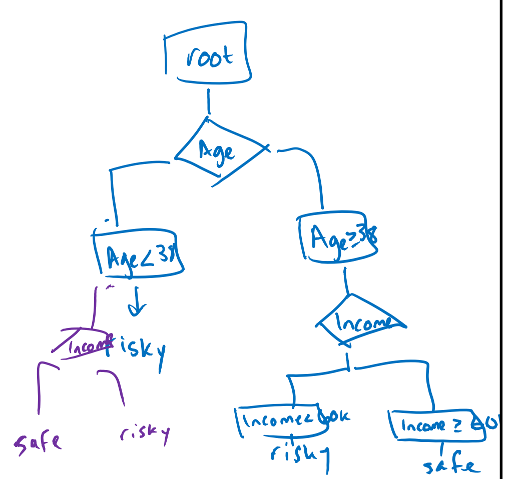


69 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Depth 2: Split on Income >= $60K

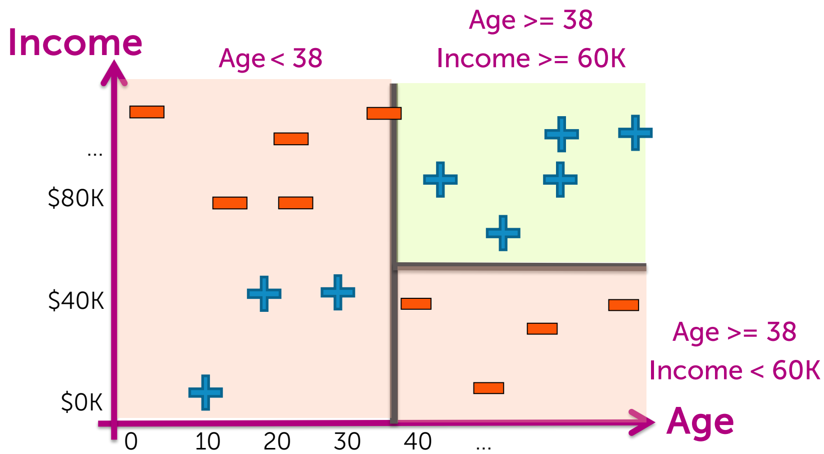
Threshold split is the line Income = 60K





70 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Each split partitions the 2-D space

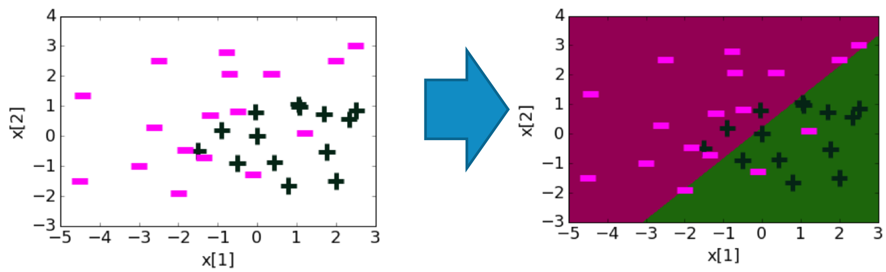


Same tree as previous slide

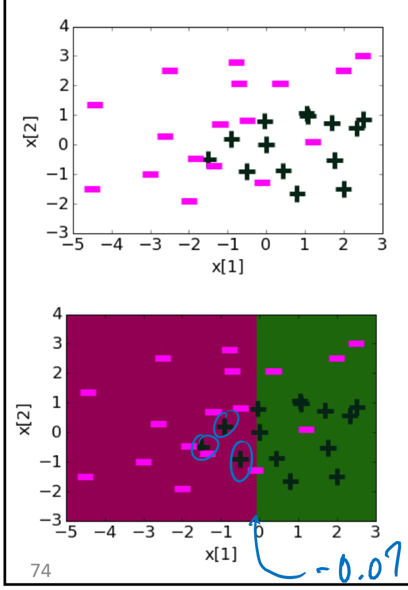
Decision trees vs logistic regression: *Example*

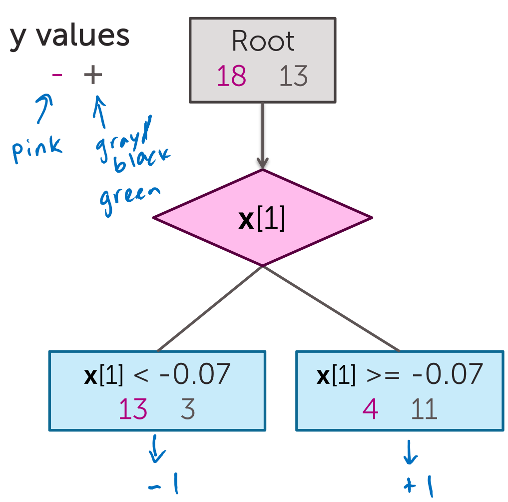
Logistic regression

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

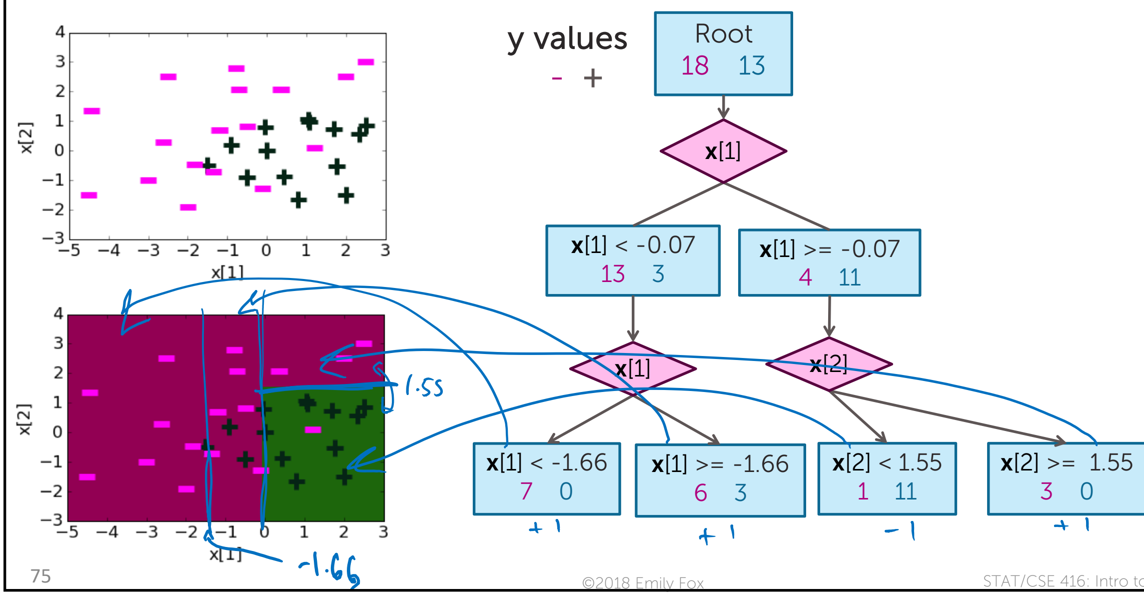


Depth 1: Split on **x**[1]



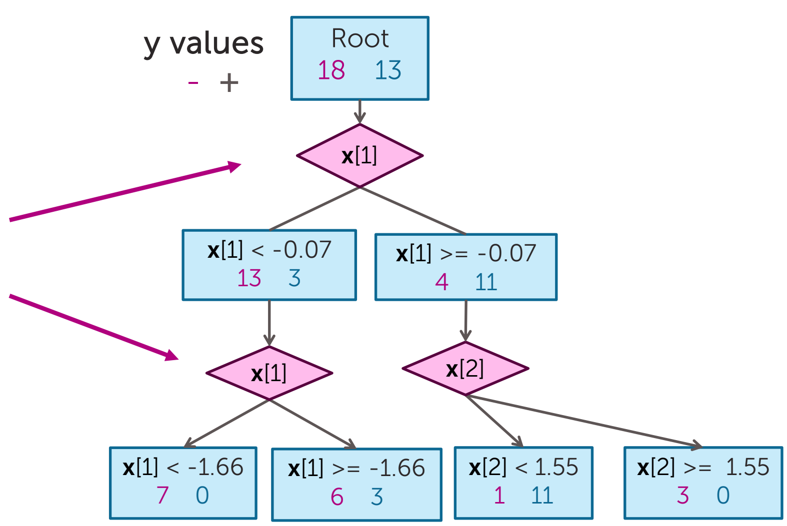


Depth 2

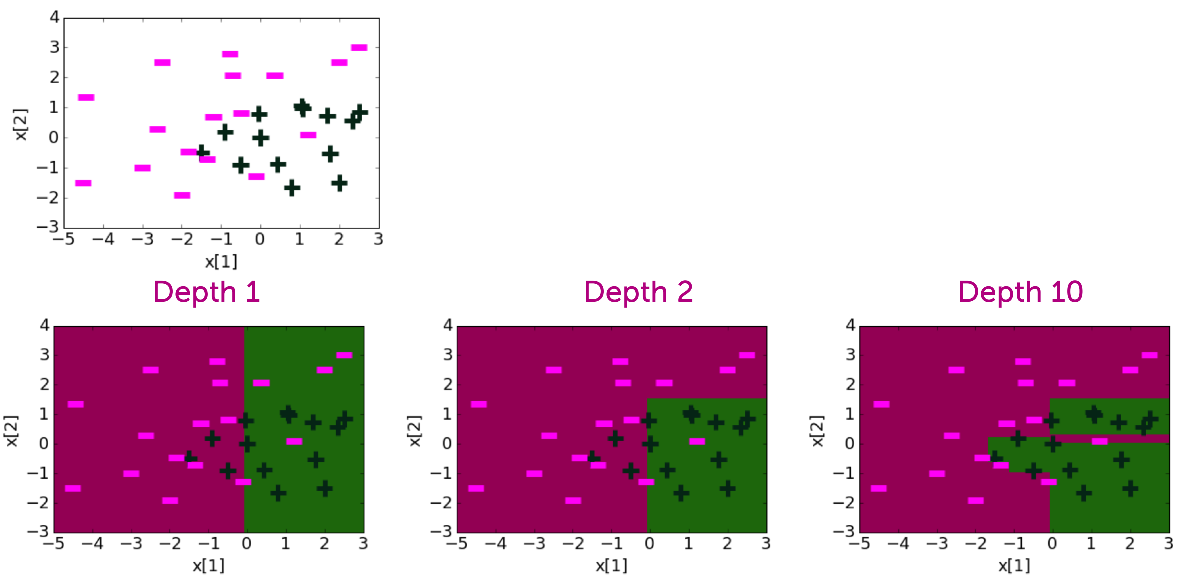


Threshold split caveat

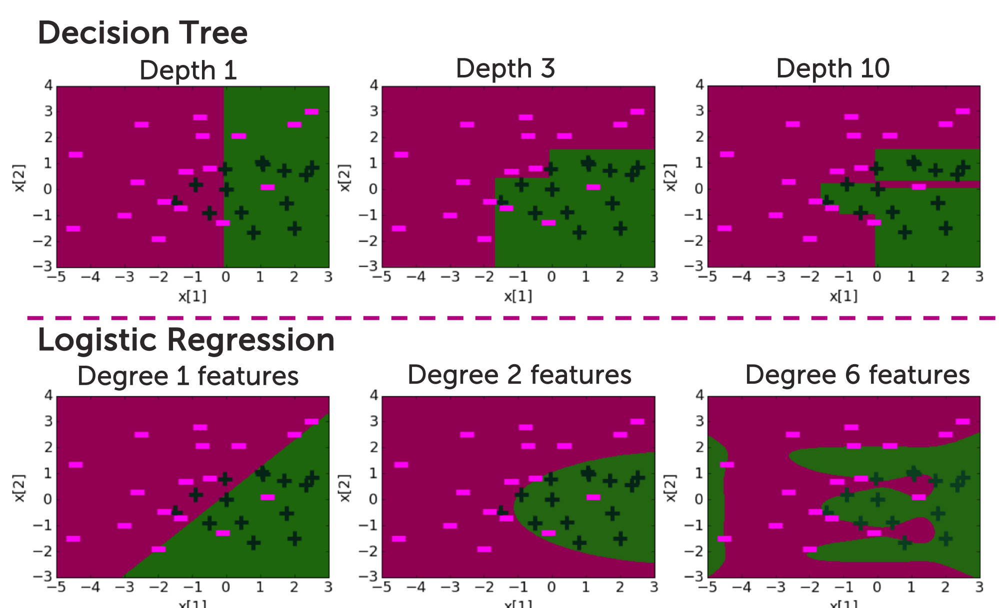
For threshold splits, same feature can be used multiple times



Decision boundaries

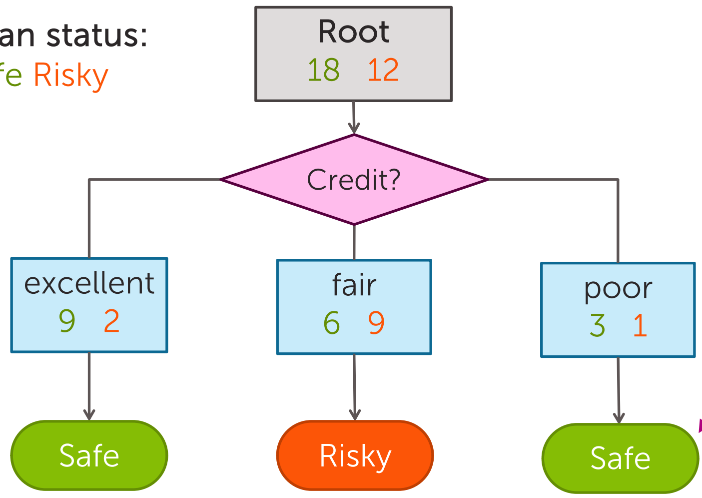


Comparing decision boundaries



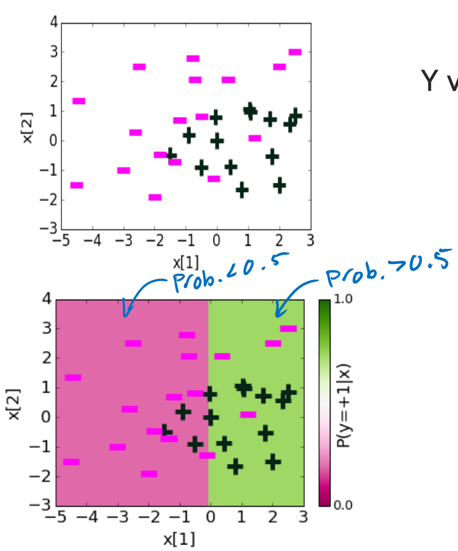
Predicting probabilities with decision trees

Loan status:
Safe Risky

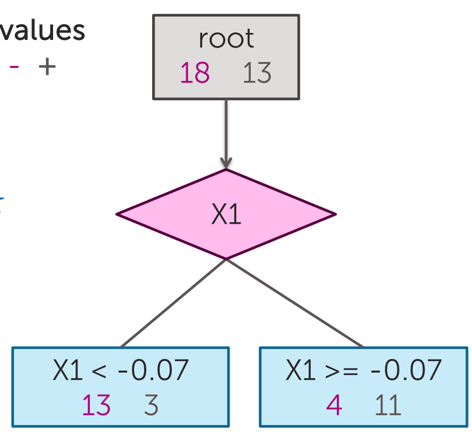


$$P(y = \text{Safe} \mid \mathbf{x}) = \frac{3}{3+1} = 0.75$$

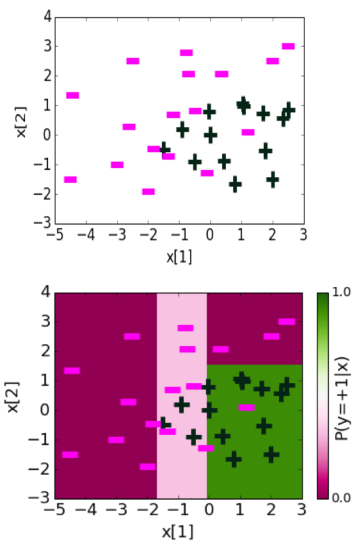
Depth 1 probabilities

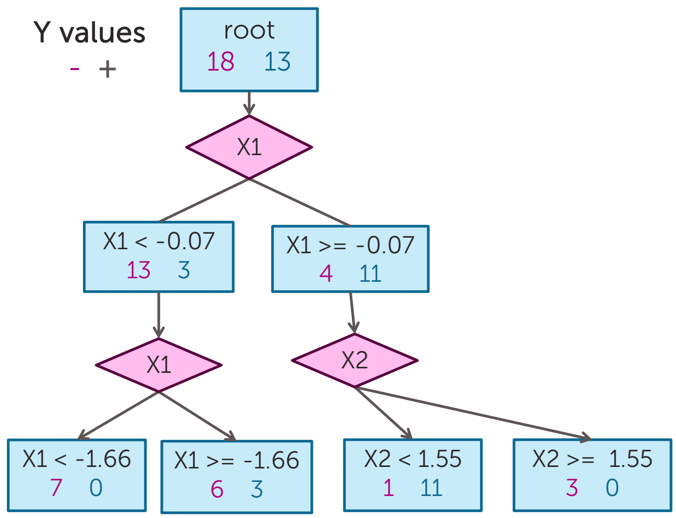


Y values
\- +



Depth 2 probabilities

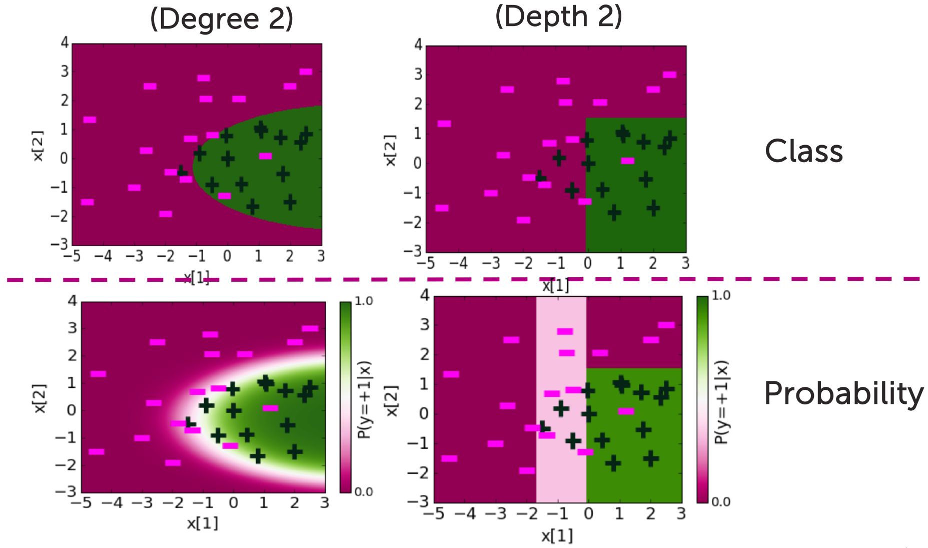




Comparison with logistic regression

Logistic Regression (Degree 2)

Decision Trees (Depth 2)



Class

Probability

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

©2018 Emily Fox
STAT/CSE 416: Intro to Machine Learning