

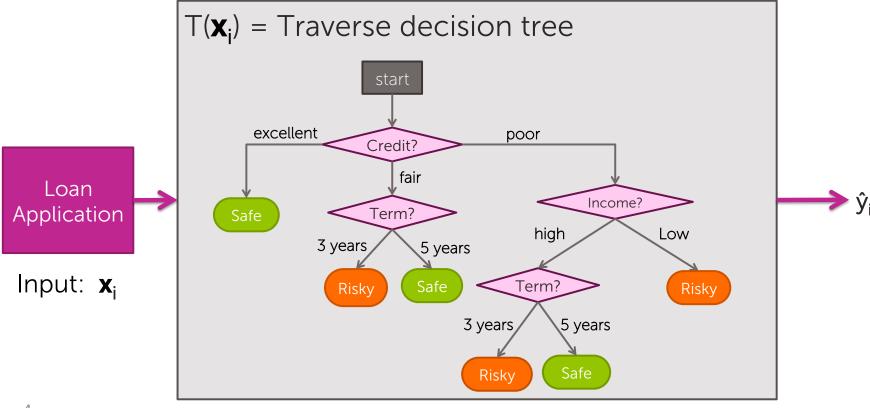
# Overfitting in decision trees

Emily Fox & Carlos Guestrin Machine Learning Specialization University of Washington

# Review of loan default prediction

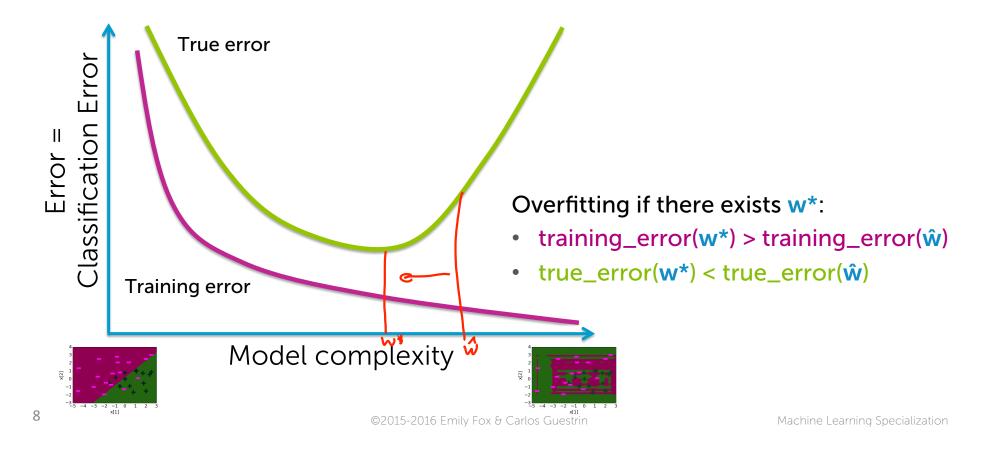


## Decision tree review

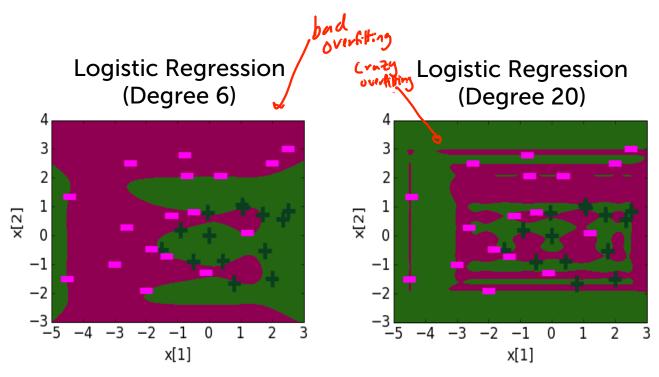


# Overfitting review

# Overfitting in logistic regression

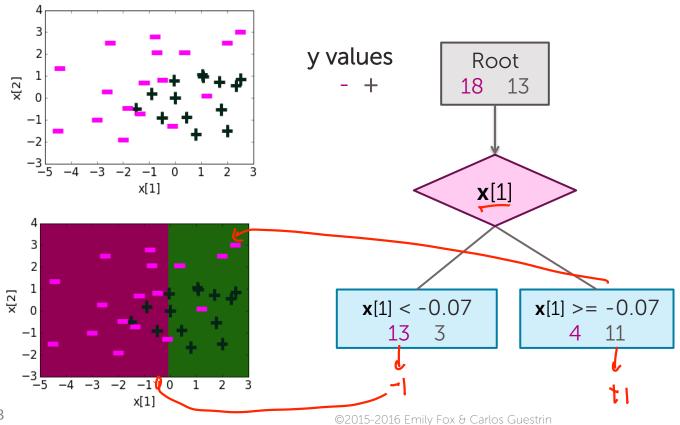


# Overfitting Overconfident predictions

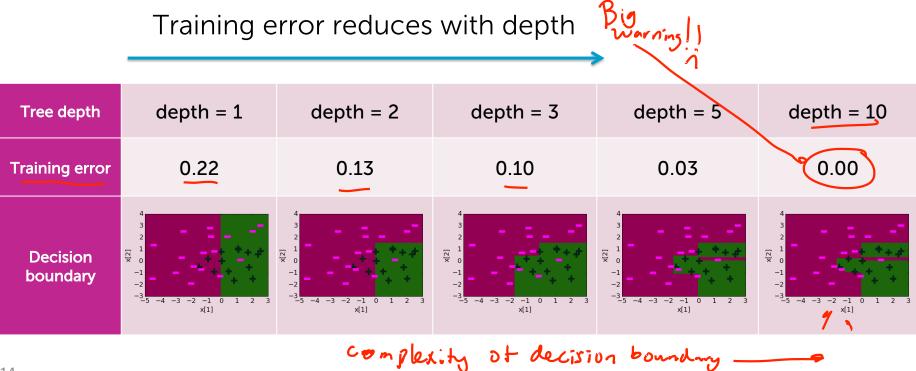


# Overfitting in decision trees

# Decision stump (Depth 1): Split on x[1]



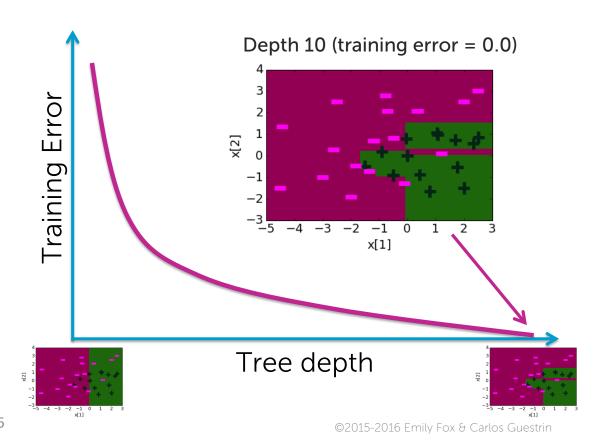
## What happens when we increase depth?



©2015-2016 Emily Fox & Carlos Guestrin

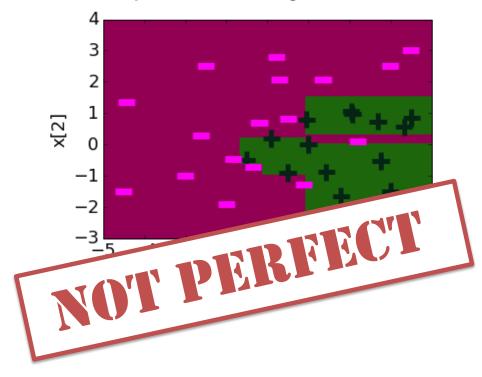
Machine Learning Specialization

# Deeper trees -> lower training error

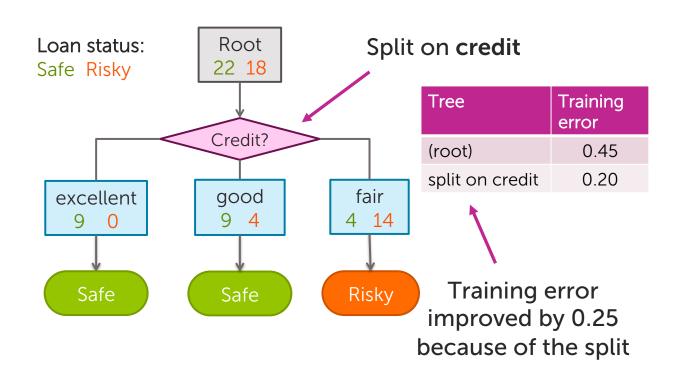


# Training error = 0: Is this model perfect?





# Why training error reduces with depth?

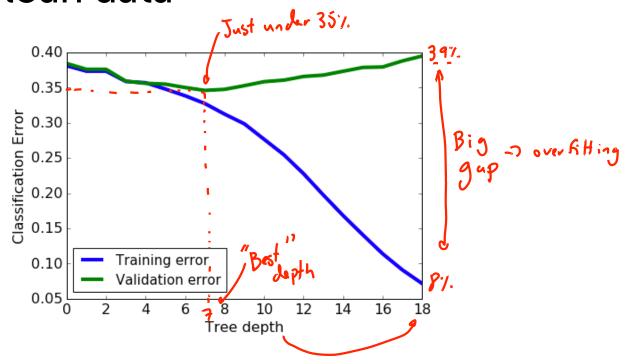


## Feature split selection algorithm

- Given a subset of data M (a node in a tree)
- For each feature h<sub>i</sub>(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

By design, each split reduces training error

# Decision trees overfitting on loan data



©2015-2016 Emily Fox & Carlos Guestrin

# Principle of Occam's razor: Simpler trees are better

# Principle of Occam's Razor



"Among competing hypotheses, the one with fewest assumptions should be selected", William of Occam, 13<sup>th</sup> Century

**Symptoms**:  $S_1$  and  $S_2$ 

**SIMPLER** 

Diagnosis 1: 2 diseases

Two diseases  $D_1$  and  $D_2$  where  $D_1$  explains  $S_1$ ,  $D_2$  explains  $S_2$ 



Diagnosis 2: 1 disease

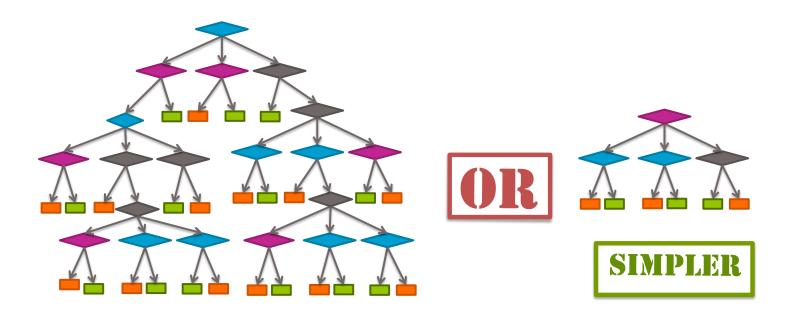
Disease  $D_3$  explains both symptoms  $S_1$  and  $S_2$ 

## Occam's Razor for decision trees

When two trees have similar classification error on the validation set, pick the simpler one

Complexity	Train error	Validation error	Same validation
Simple	0.23	0.24	error
Moderate —	0.12	0.15	Pick
Complex	0.07	0.15	bud!
Super complex 0		0.18	Overfit

# Which tree is simpler?

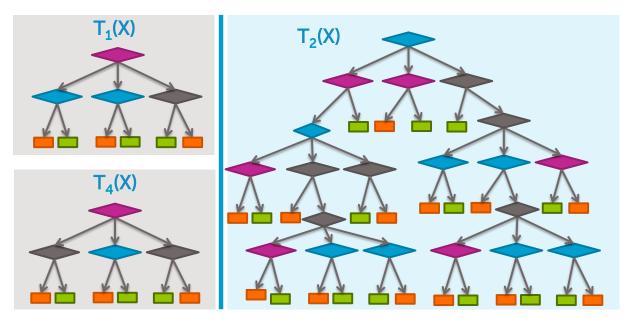


# Modified tree learning problem

Find a "simple" decision tree with low classification error

#### Simple trees

#### Complex trees



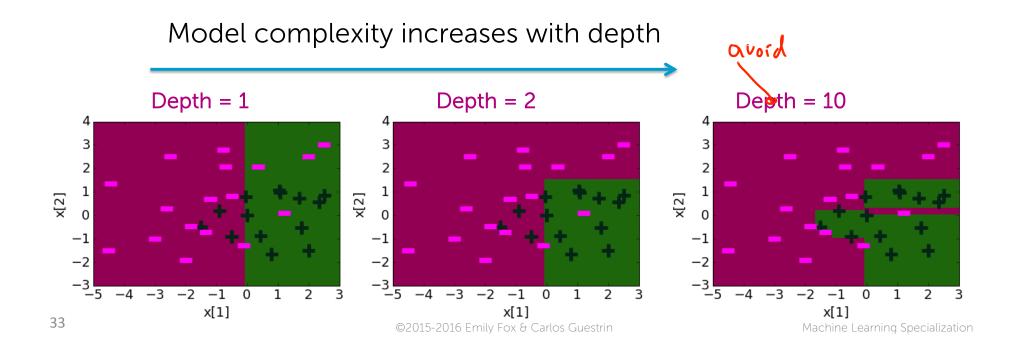
## How do we pick simpler trees?

- 1. Early Stopping: Stop learning algorithm before tree become too complex
- 2. Pruning: Simplify tree after learning algorithm terminates

# Early stopping for learning decision trees

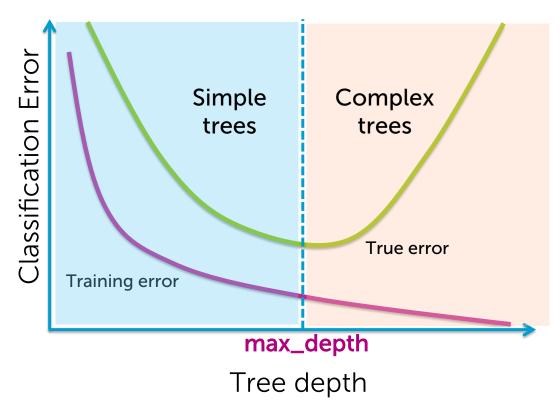
@2015\_2016 Emily Fox & Carlos Guestrin

# Deeper trees → Increasing complexity

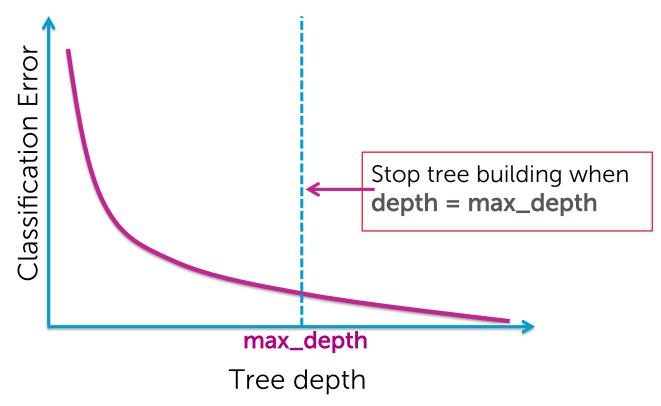


# Early stopping condition 1: Limit the depth of a tree

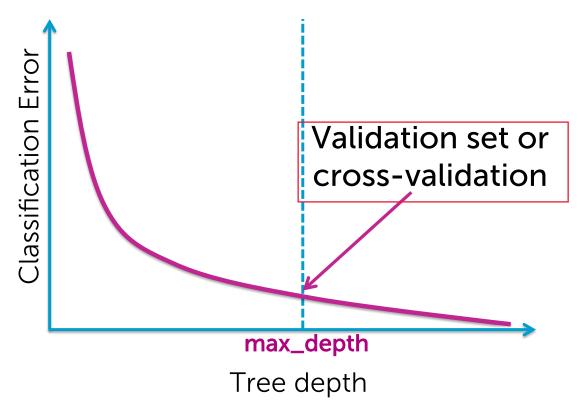
## Restrict tree learning to shallow trees?



# Early stopping condition 1: Limit depth of tree



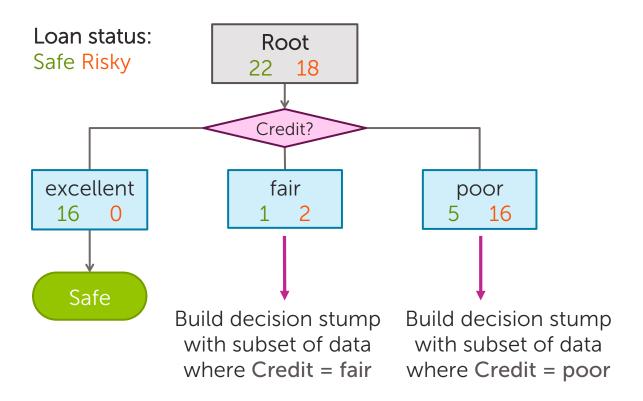
# Picking value for max\_depth???



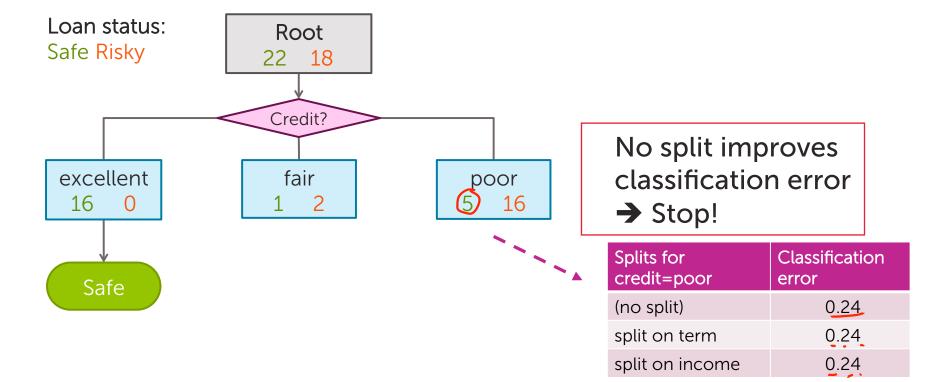
## Early stopping condition 2:

Use classification error to limit depth of tree

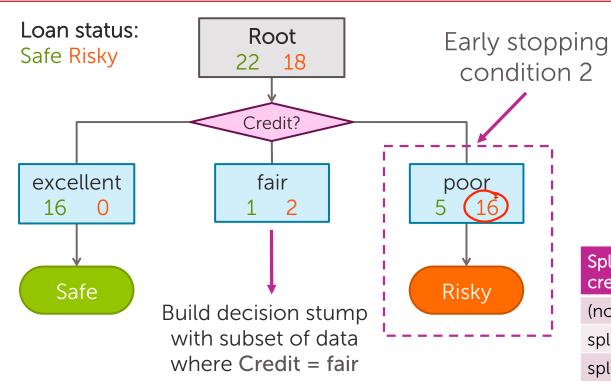
## Decision tree recursion review



## Split selection for credit=poor



# Early stopping condition 2: No split improves classification error



Splits for credit=poor	Classification error
(no split)	0.24
split on term	0.24
split on income	0.24

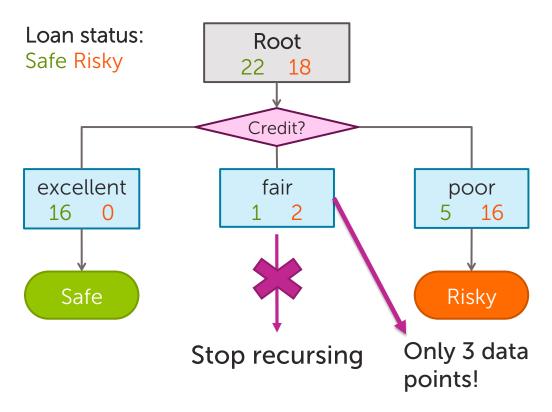
# Practical notes about stopping when classification error doesn't decrease

- 1. Typically, add magic parameter  $\epsilon$ 
  - Stop if error doesn't decrease by more than ε
- 2. Some pitfalls to this rule (see pruning section)
- 3. Very useful in practice

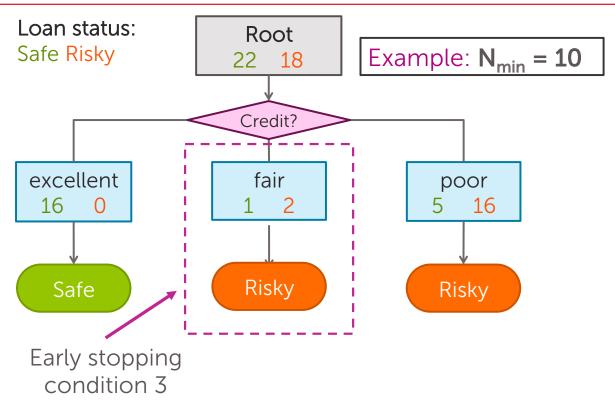
## Early stopping condition 3:

Stop if number of data points contained in a node is too small

## Can we trust nodes with very few points?



# Early stopping condition 3: Stop when data points in a node $\leq N_{min}$



# Summary of decision trees with early stopping

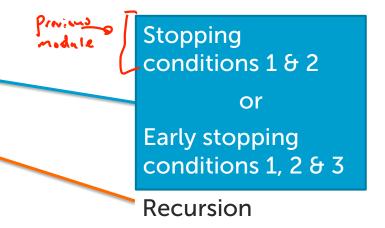
# Early stopping: Summary

- 1. Limit tree depth: Stop splitting after a certain depth
- 2. Classification error: Do not consider any split that does not cause a sufficient decrease in classification error
- 3. Minimum node "size": Do not split an intermediate node which contains too few data points

#### 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 

     Majoring
  - Step 4: Otherwise, go to Step 2 & continue (recurse) on this split



#### Overfitting in Decision Trees: Pruning



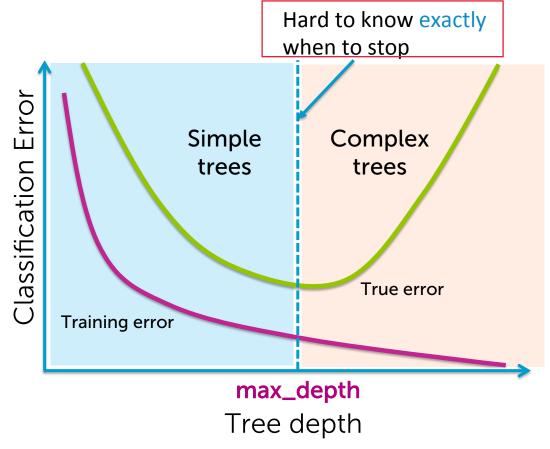
©2015-2016 Emily Fox & Carlos Guestrir

#### Stopping condition summary

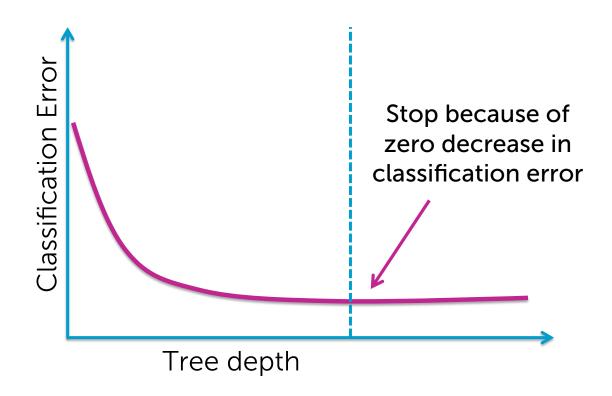
- Stopping condition:
  - 1. All examples have the same target value
  - 2. No more features to split on
- Early stopping conditions:
  - 1. Limit tree depth
  - 2. Do not consider splits that do not cause a sufficient decrease in classification error
  - 3. Do not split an intermediate node which contains too few data points

# Exploring some challenges with early stopping conditions

#### Challenge with early stopping condition 1

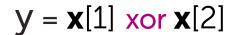


#### Is early stopping condition 2 a good idea?

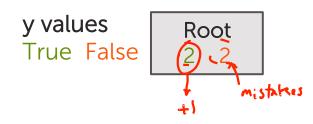


#### Early stopping condition 2:

#### Don't stop if error doesn't decrease???



<b>x</b> [1]	<b>x</b> [2]	У
False	False	False
False	True	True
True	False	True
True	True	False



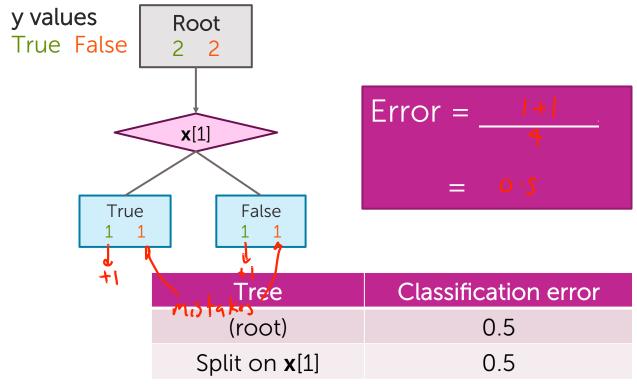
$$Error = \frac{2}{4}$$

$$= \frac{0.5}{5}$$

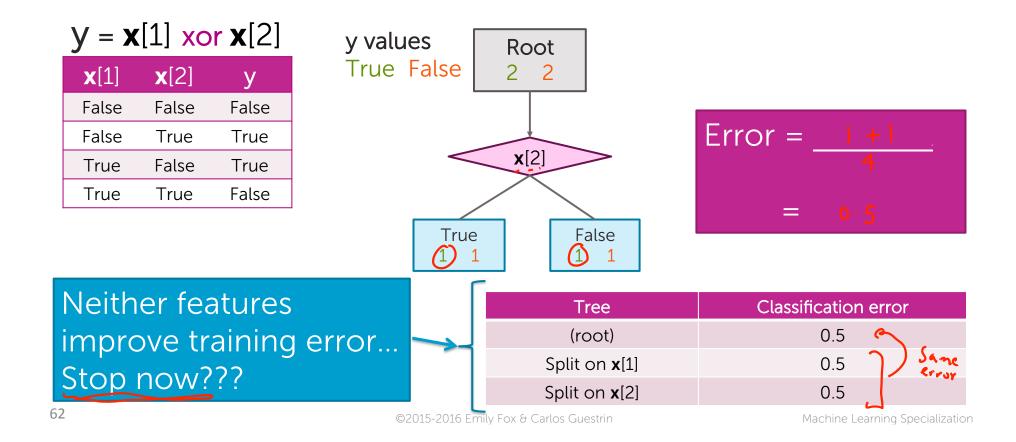
Tree	Classification error
(root)	0.5

#### Consider split on x[1]

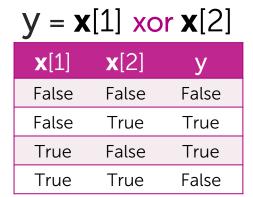


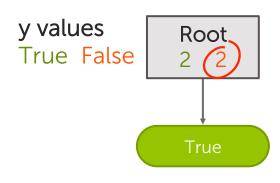


#### Consider split on x[2]



#### Final tree with early stopping condition 2





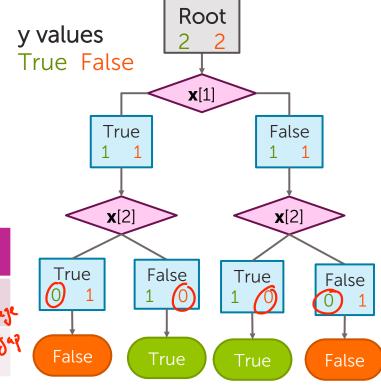
Tree	Classification error
with early stopping condition 2	0.5

#### Without early stopping condition 2



<b>x</b> [1]	<b>x</b> [2]	у
False	False	False
False	True	True
True	False	True
True	True	False

Tree	Classification error
with early stopping condition 2	0.5 7 huze
without early stopping condition 2	0.0



64

©2015-2016 Emily Fox & Carlos Guestrin

#### 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

#### Tree pruning

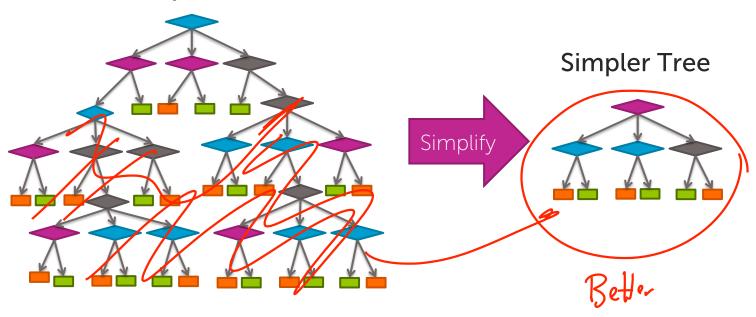
#### Two approaches to picking simpler trees

- 1. Early Stopping: Stop the learning algorithm before the tree becomes too complex
- 2. Pruning: Simplify the tree after the learning algorithm terminates

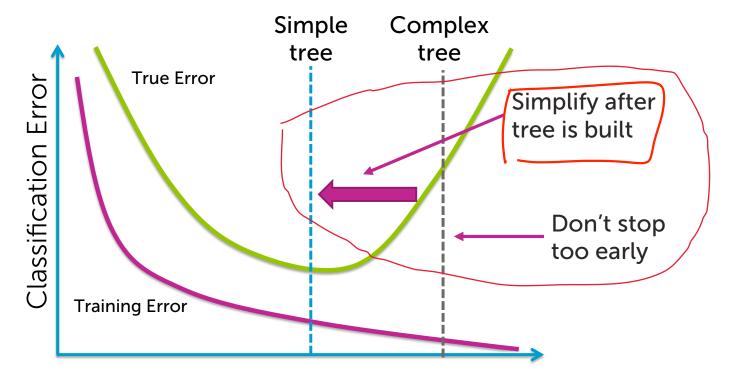
Complements early stopping

# Pruning: *Intuition*Train a complex tree, simplify later

#### Complex Tree

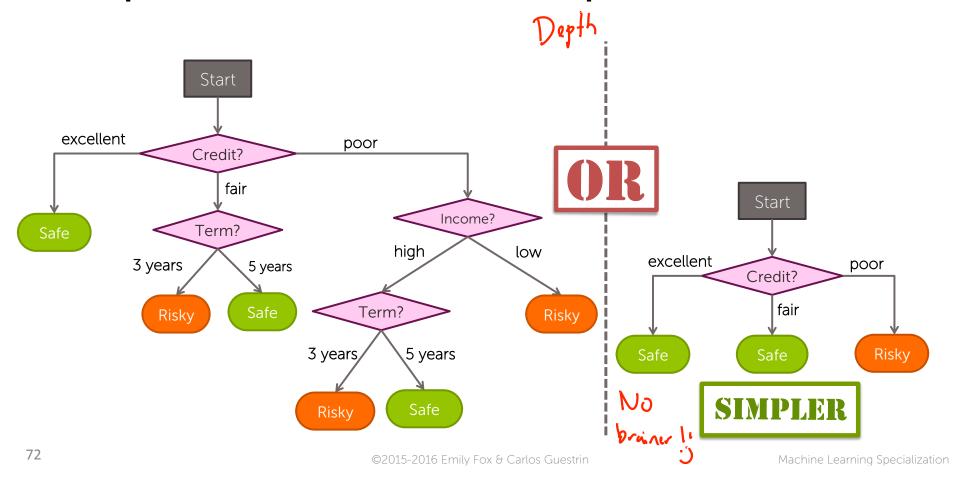


## Pruning motivation

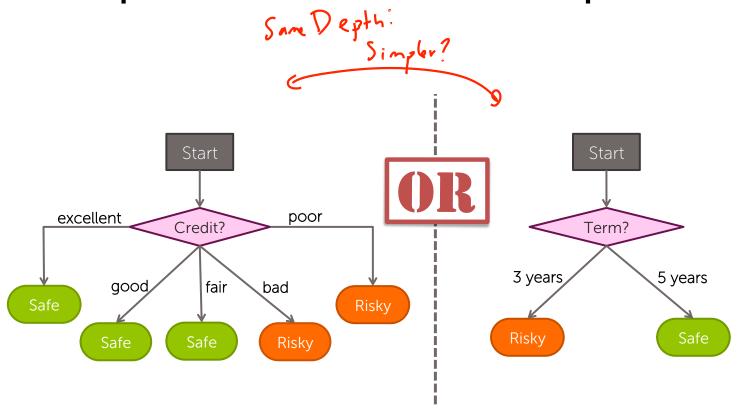


Tree depth

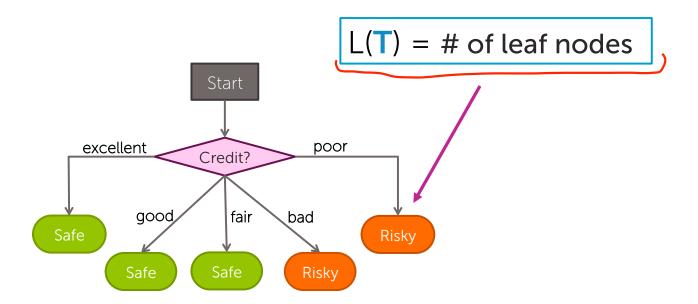
#### Example 1: Which tree is simpler?



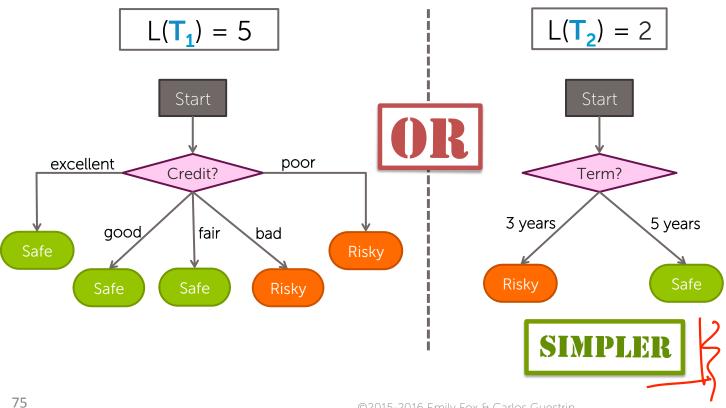
## Example 2: Which tree is simpler???



#### Simple measure of complexity of tree

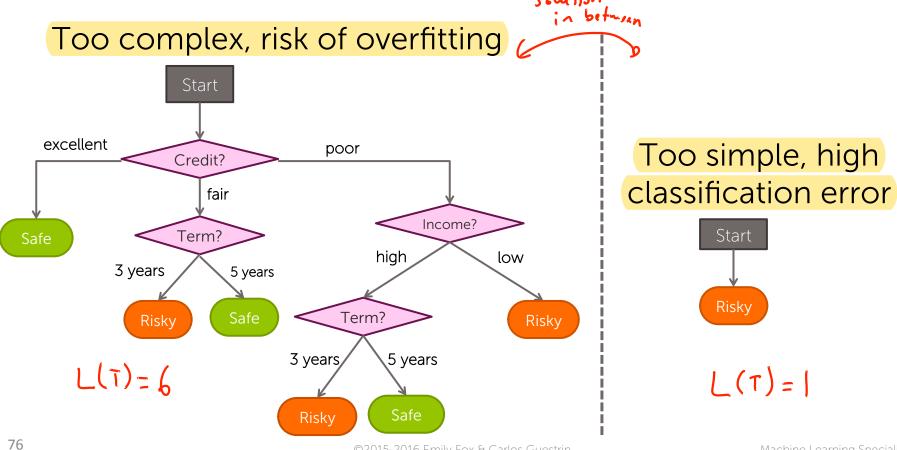


#### Which tree has lower L(T)?



©2015-2016 Emily Fox & Carlos Guestrin

#### Balance simplicity & predictive power

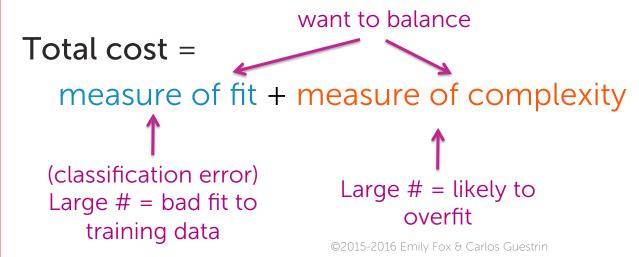


©2015-2016 Emily Fox & Carlos Guestrin

#### Desired total quality format

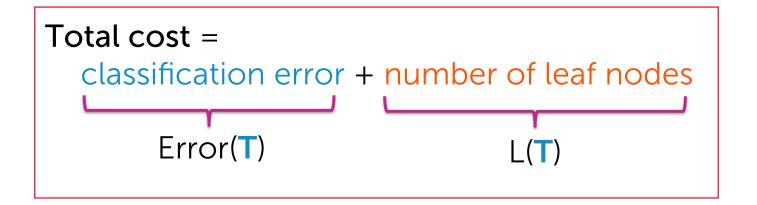
#### Want to balance:

- i. How well tree fits data
- ii. Complexity of tree



7Ω

#### Consider specific total cost



#### Balancing fit and complexity

```
Total cost C(T) = Error(T) + \lambda L(T)

tuning parameter

If \lambda = 0:

Standard decision true learning

If \lambda = \infty:

\infty panally \Rightarrow

\mathcal{G} = \text{Majority class}

or panally \Rightarrow

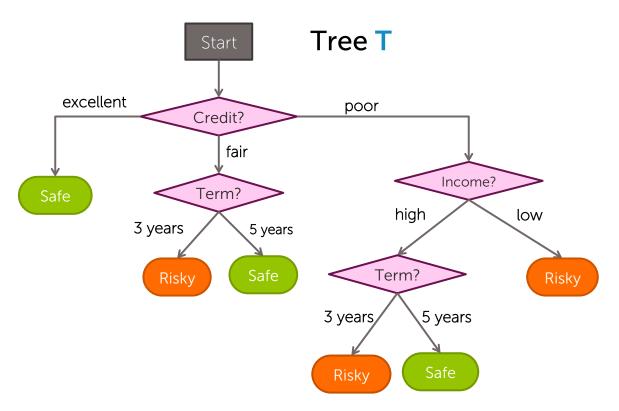
If \lambda in between: Balance fit & complexity
```

#### Use total cost to simplify trees

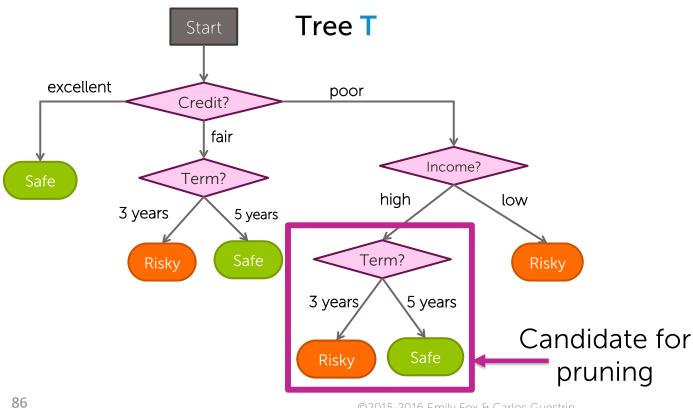
# Complex tree Simpler tree Total quality based pruning Error (T) + \( \lambda \) (T)

#### Tree pruning algorithm

### **Pruning Intuition**

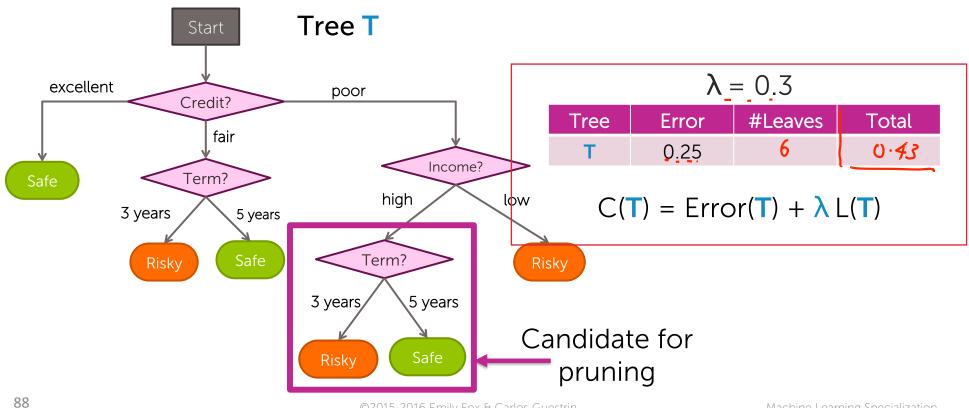


### Step 1: Consider a split



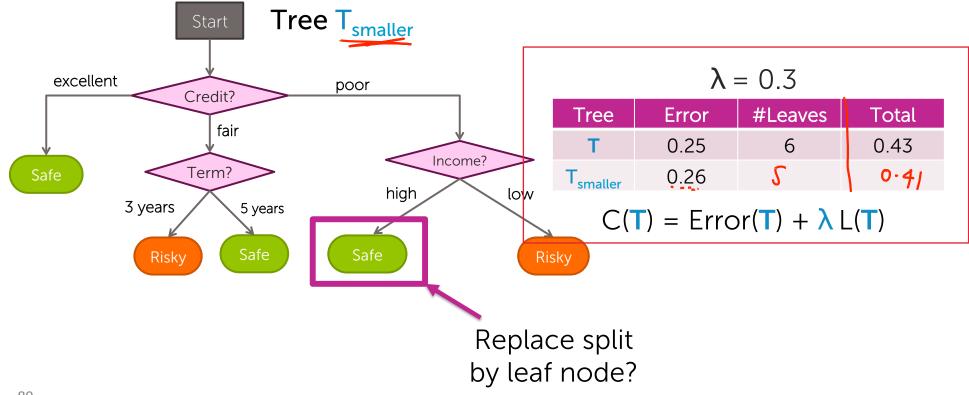
©2015-2016 Emily Fox & Carlos Guestrin

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



©2015-2016 Emily Fox & Carlos Guestrin

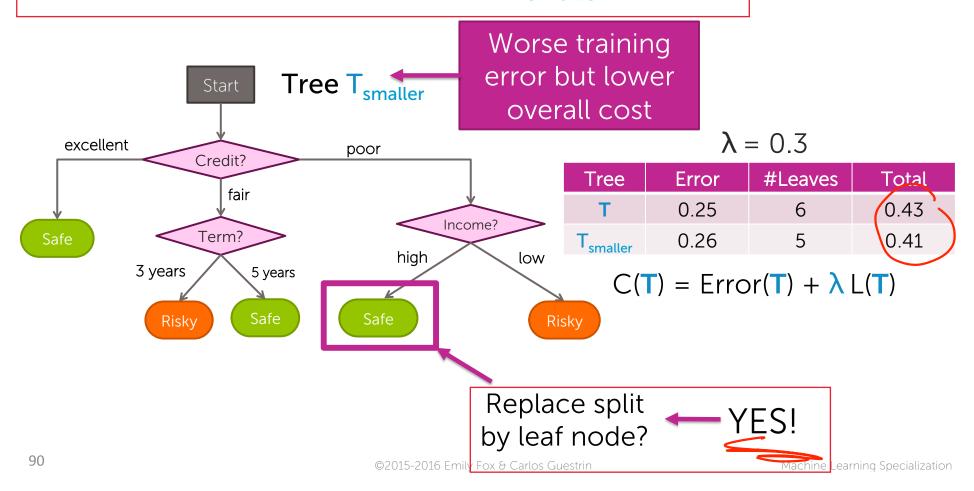
## Step 2: "Undo" the splits on T<sub>smaller</sub>



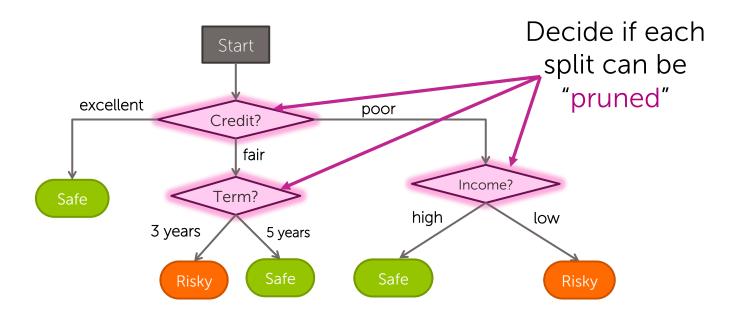
89

©2015-2016 Emily Fox & Carlos Guestrin

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



#### Step 5: Repeat Steps 1-4 for every split



#### Decision tree pruning algorithm

- Start at bottom of tree T and traverse up, apply prune\_split to each decision node M
- prune\_split(T,M):
  - 1. Compute total cost of tree T using  $C(T) = Error(T) + \lambda L(T)$
  - 2. Let T<sub>smaller</sub> be tree after pruning subtree below M
  - 3. Compute total cost complexity of  $T_{\text{smaller}}$   $C(T_{\text{smaller}}) = Error(T_{\text{smaller}}) + \lambda L(T_{\text{smaller}})$
  - 4. If  $C(T_{smaller}) < C(T)$ , prune to  $T_{smaller}$

# 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

#### Thank you to Dr. Krishna Sridhar



Dr. Krishna Sridhar Staff Data Scientist, Dato, Inc.