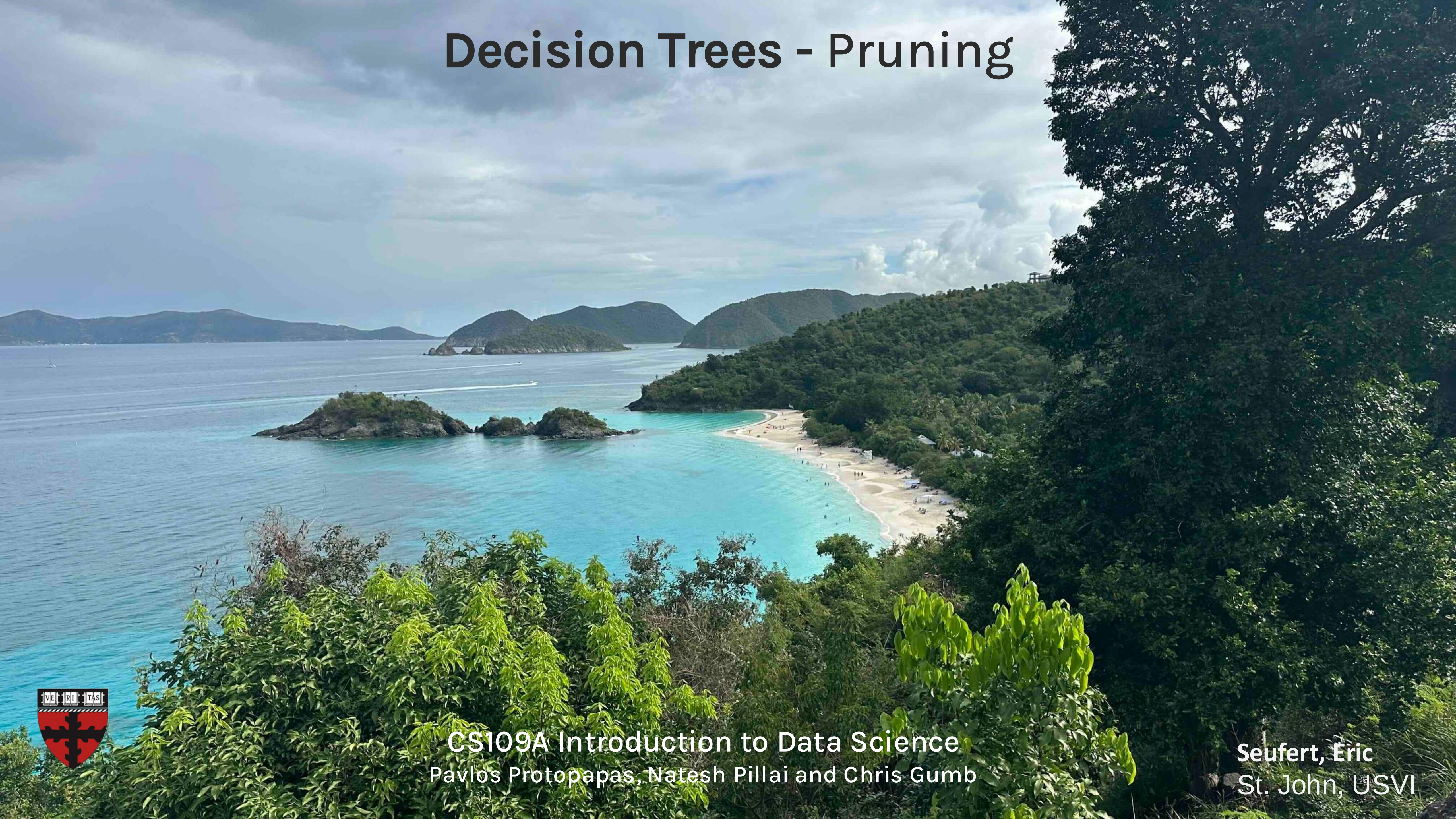
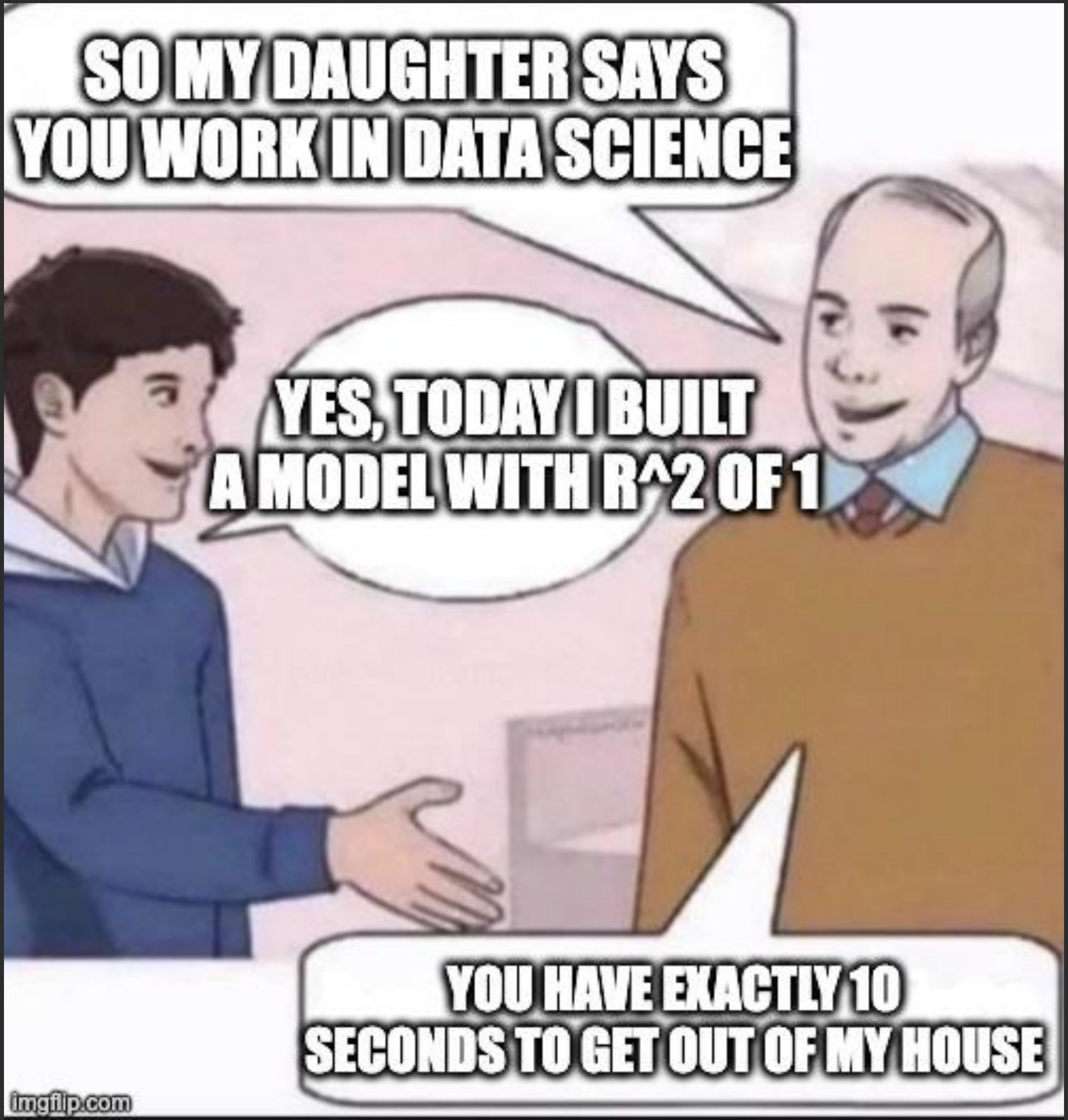
Decision Trees - Pruning



Seufert, Eric
St. John, USVI

CS109A Introduction to Data Science

Pavlos Protopapas, Natesh Pillai and Chris Gumb


SO MY DAUGHTER SAYS YOU WORK IN DATA SCIENCE
YES, TODAY I BUILT A MODEL WITH R^2 OF 1
YOU HAVE EXACTLY 10 SECONDS TO GET OUT OF MY HOUSE
imgflip.com

Outline

- Decision Trees – Regression
- Numerical vs Categorical Attributes
- Pruning

Alternatives to Using Stopping Conditions

What is the major issue with pre-specifying a stopping condition?

- You may stop too early or stop too late.

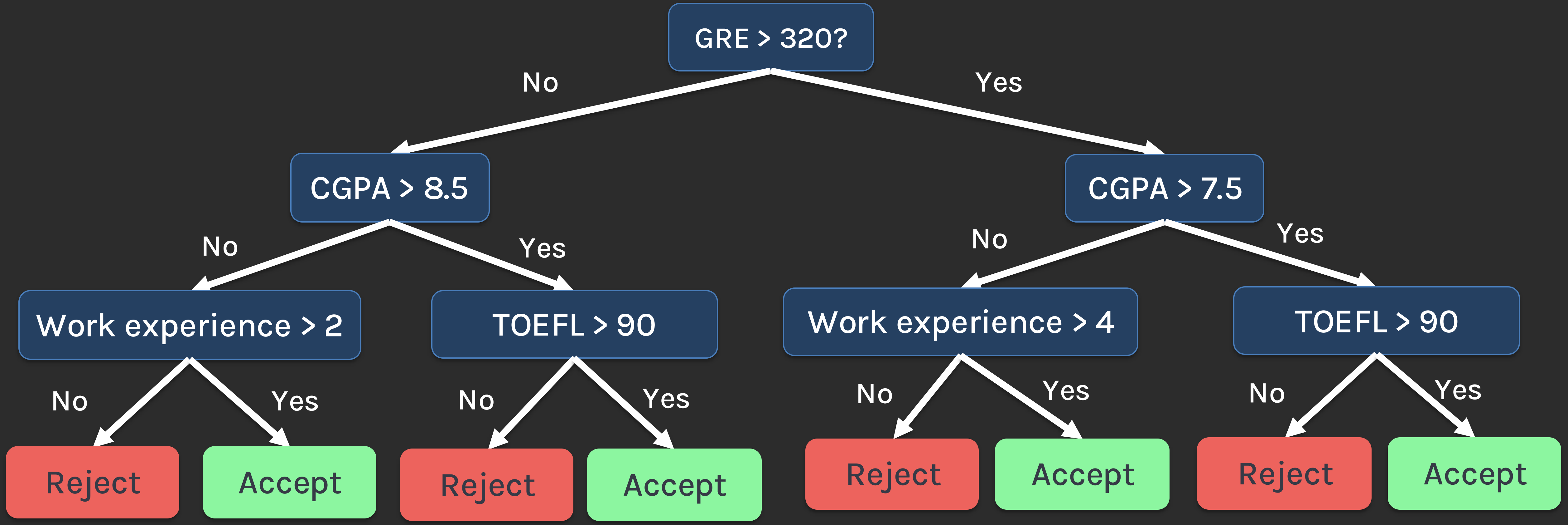
How can we fix this issue?

- Choose several stopping criteria (e.g., set minimal Gain(R) at various levels) and cross-validate to decide which one is the best.

What is an alternative approach to this method?

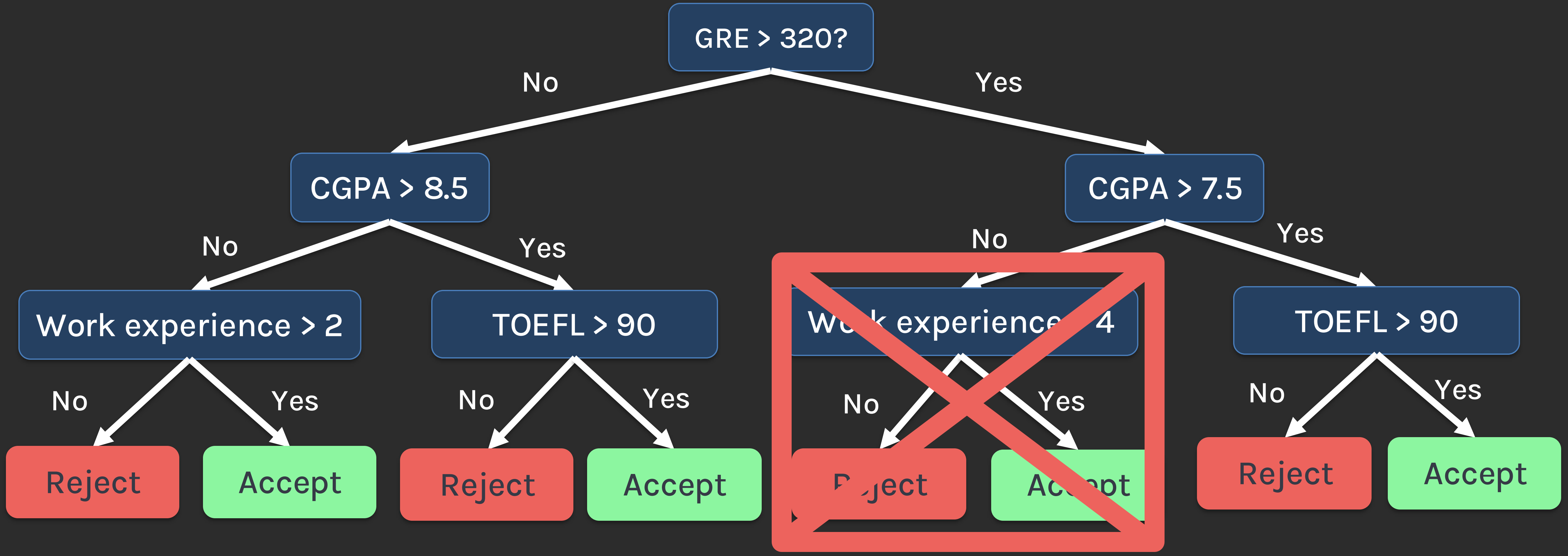
- Don't stop. Instead, prune the tree!

Example: Evaluating Applications to a GRADUATE Program



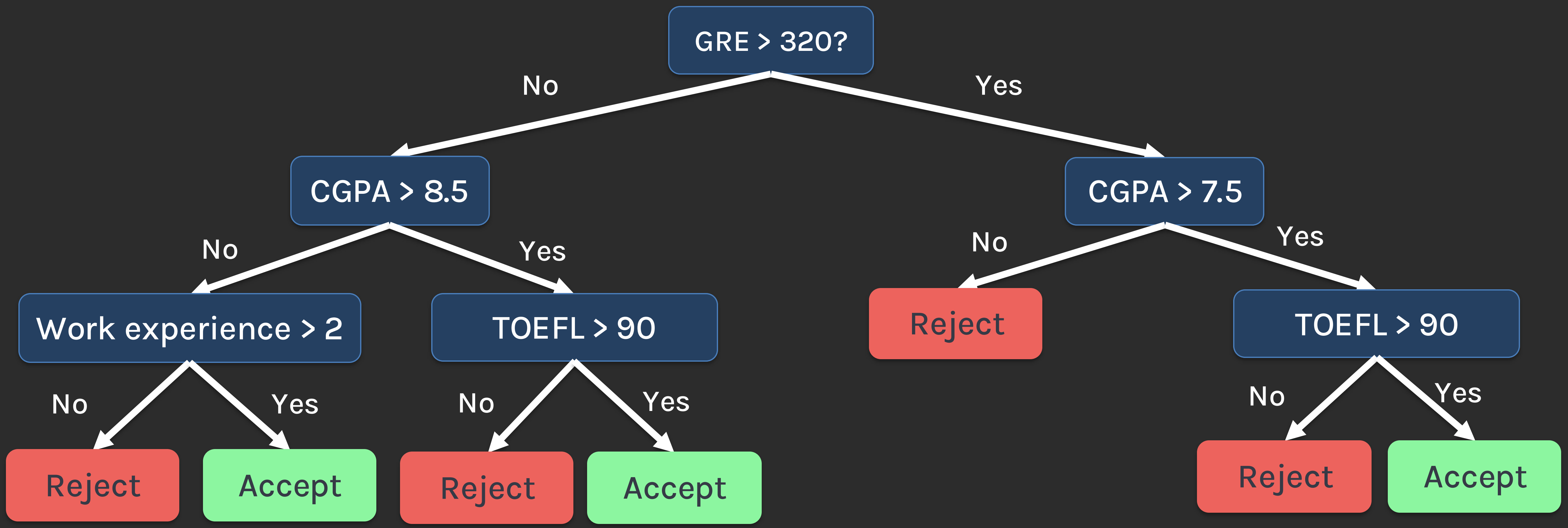
** Above representation is only for pedagogical purposes.

Example: Evaluating Applications to a GRADUATE Program



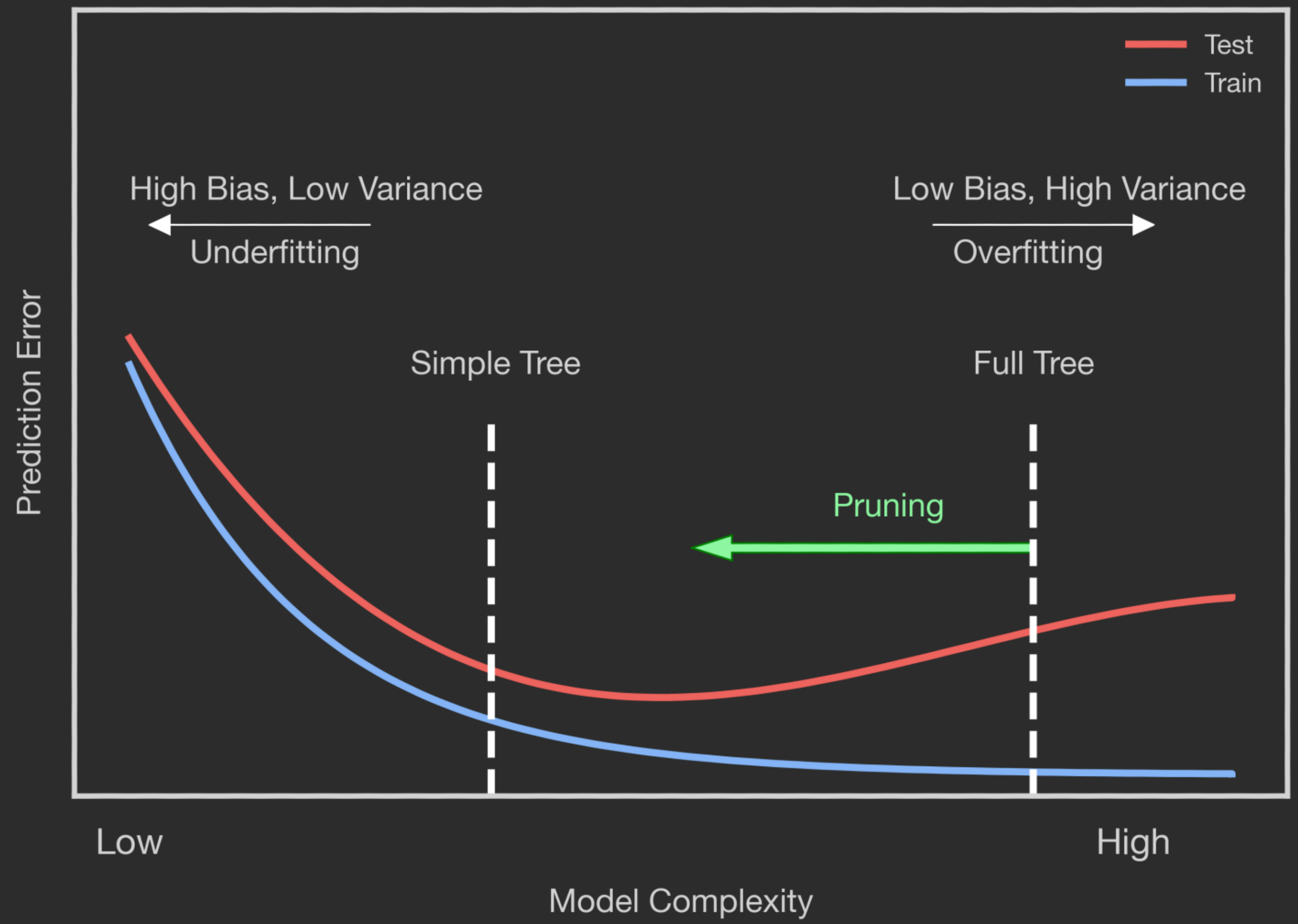
** Above representation is only for pedagogical purposes.

Example: Evaluating Applications to a GRADUATE Program



** Above representation is only for pedagogical purposes.

Motivation for Pruning



Rather than preventing a complex tree from growing, we can obtain a simpler tree by ‘pruning’ a complex one.

Pruning

There are many methods of pruning. A common one is the **cost complexity pruning:**

Classification Error

$$C(T) = Error(T) + \alpha \overbrace{|T|}^{\text{Regularization term}}$$

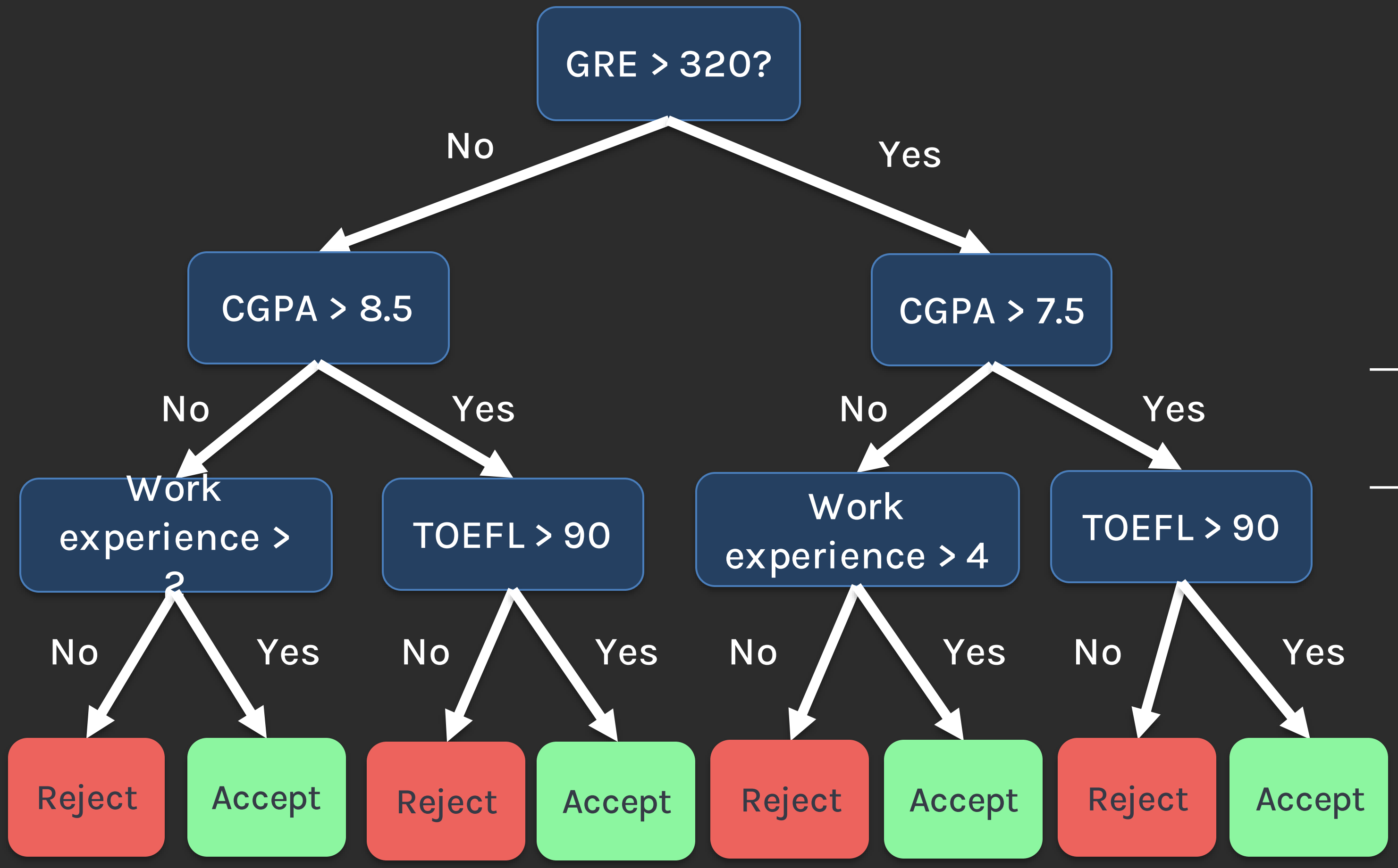
Decision tree

Complexity Parameter

Number of leaves in the tree

In other words, we add a 'regularization' term!

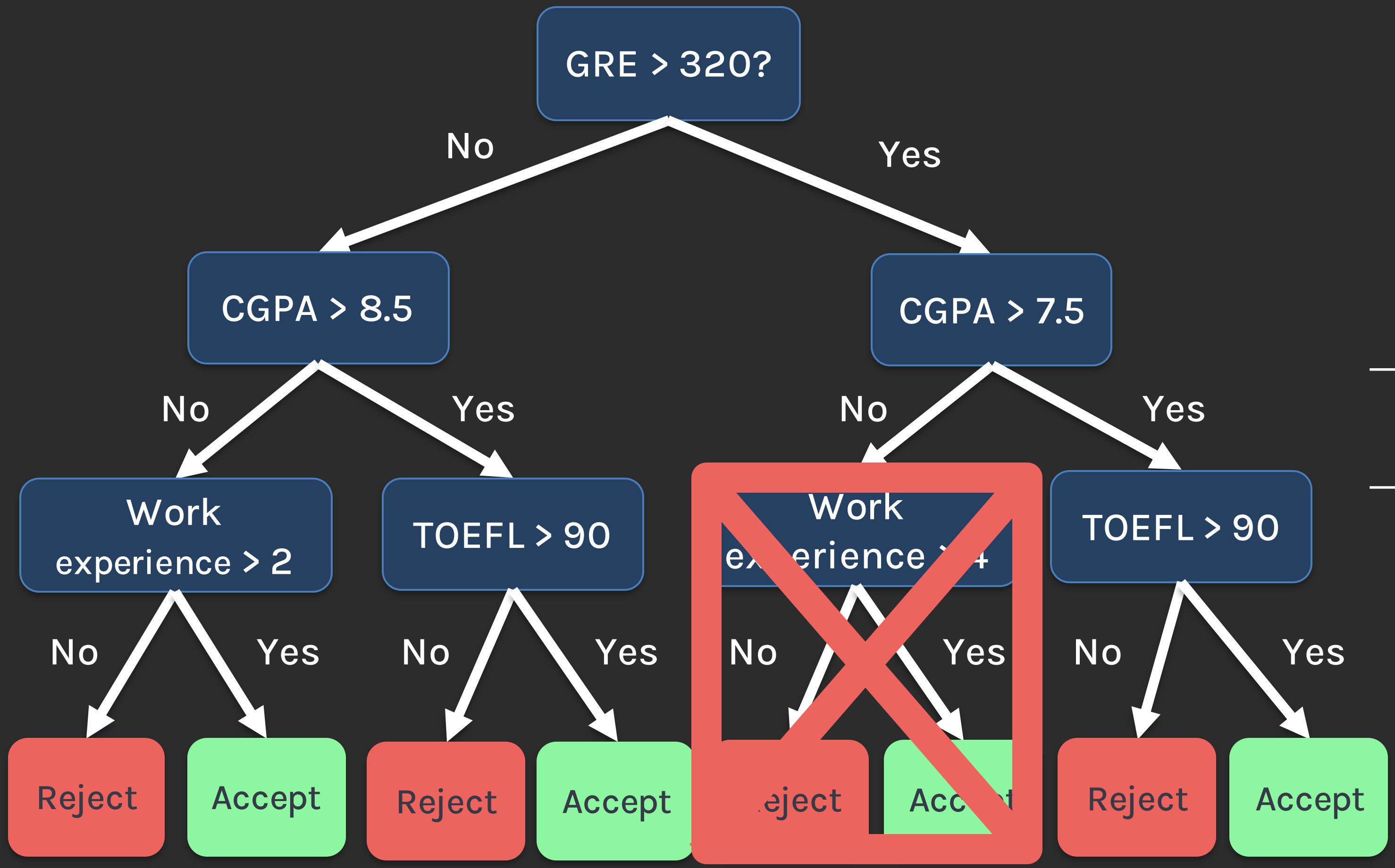
Cost-Complexity Pruning: Example



$$\alpha = 0.2$$

Tree	**Error(T)**	**|T|**	$Error(T) + \alpha\,\lvert T\rvert$
T	0.32	8	0.32 + 0.2*8 = 1.92

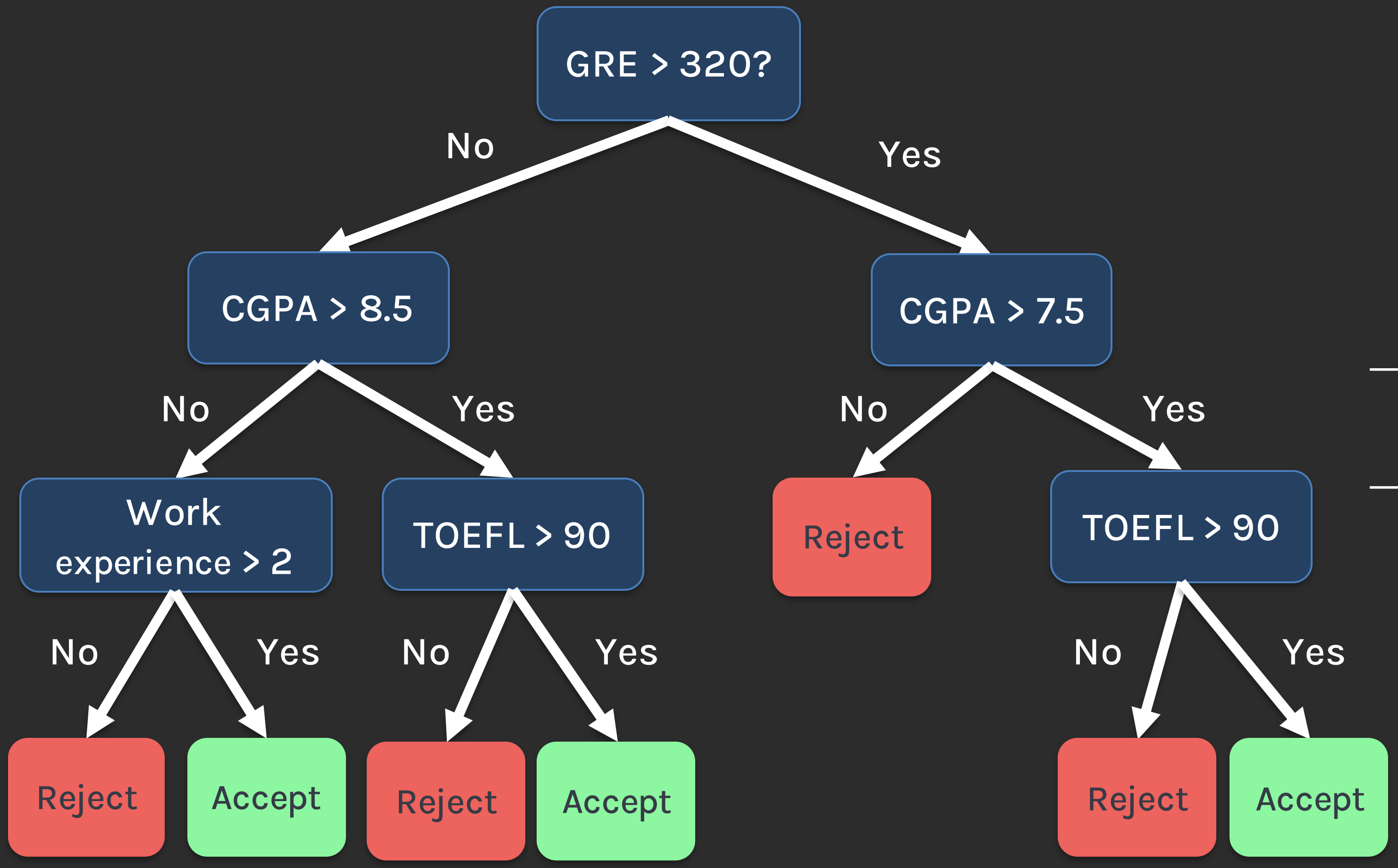
Cost-Complexity Pruning: Example



$\alpha = 0.2$

Tree	Error(T)	|T|	$Error(T) + \alpha\,\|T\|$
T	0.32	8	1.92

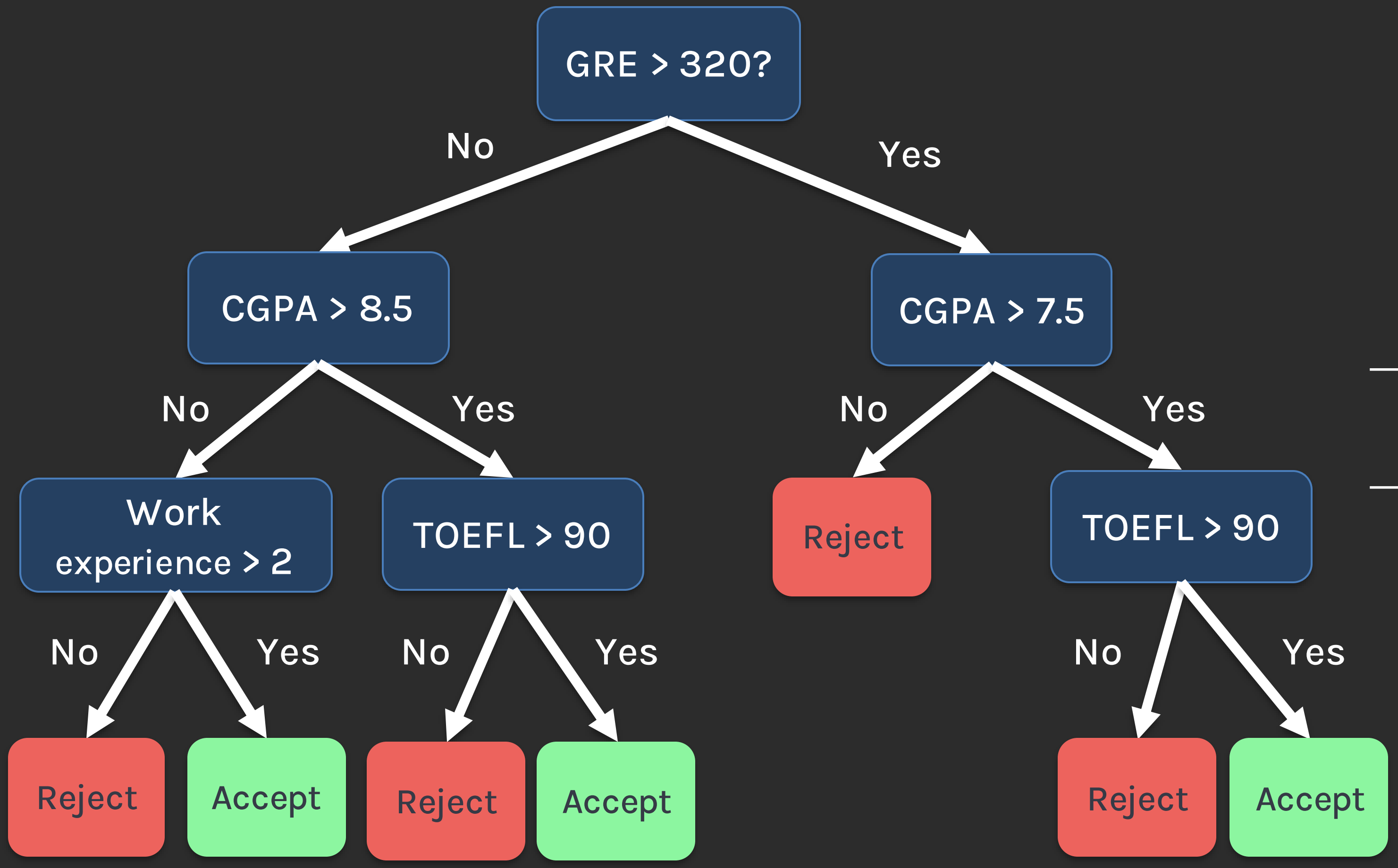
Cost-Complexity Pruning: Example



$$\alpha = 0.2$$

Tree	**Error(T)**	**|T|**	$Error(T) + \alpha\,\|T\|$
T	0.32	8	1.92
T_{small}	0.33	7	0.33 + 0.2*7 = 1.73

Cost-Complexity Pruning: Example



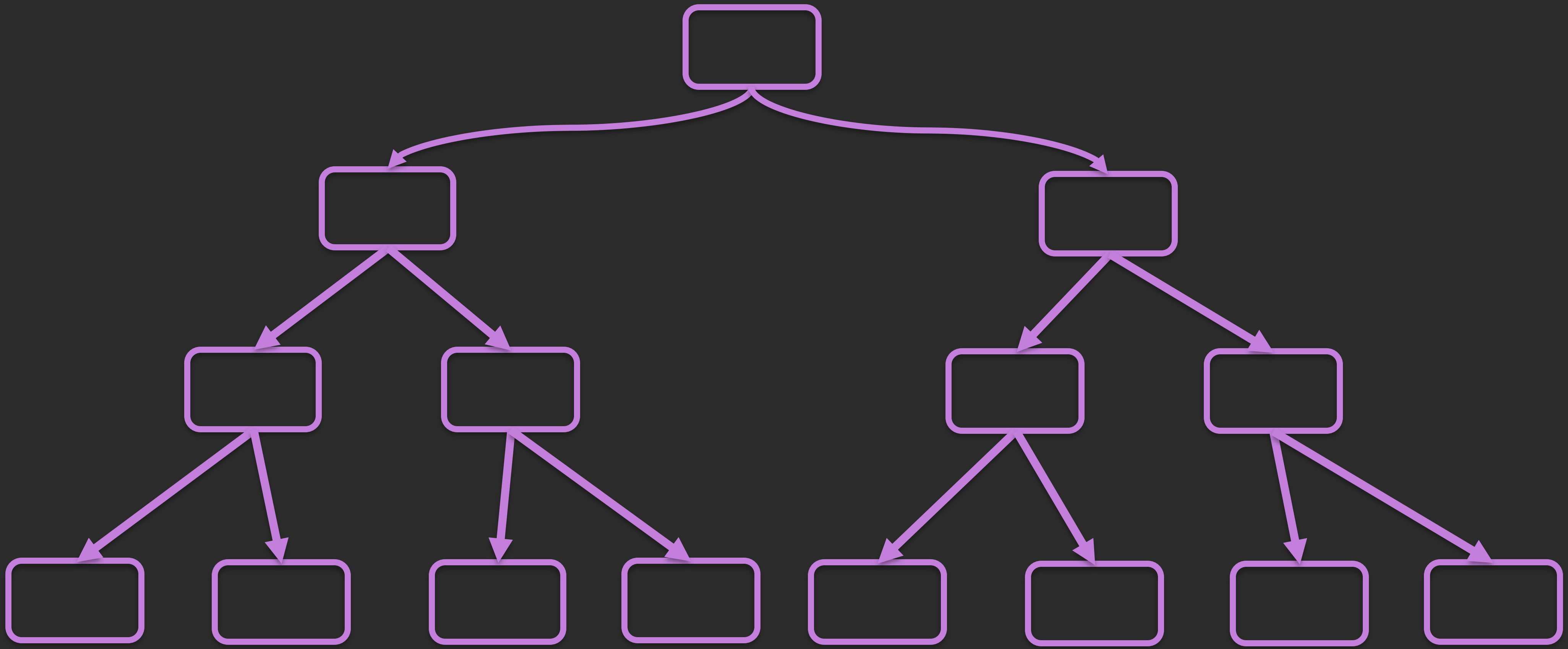
$$\alpha = 0.2$$

Tree	**Error(T)**	**|T|**	$Error(T) + \alpha \|T\|$
T	0.32	8	1.92
T_{small}	0.33	7	1.73

The smaller tree has a larger error `Error(T)` but smaller complexity score `C(T)`.

Pruning

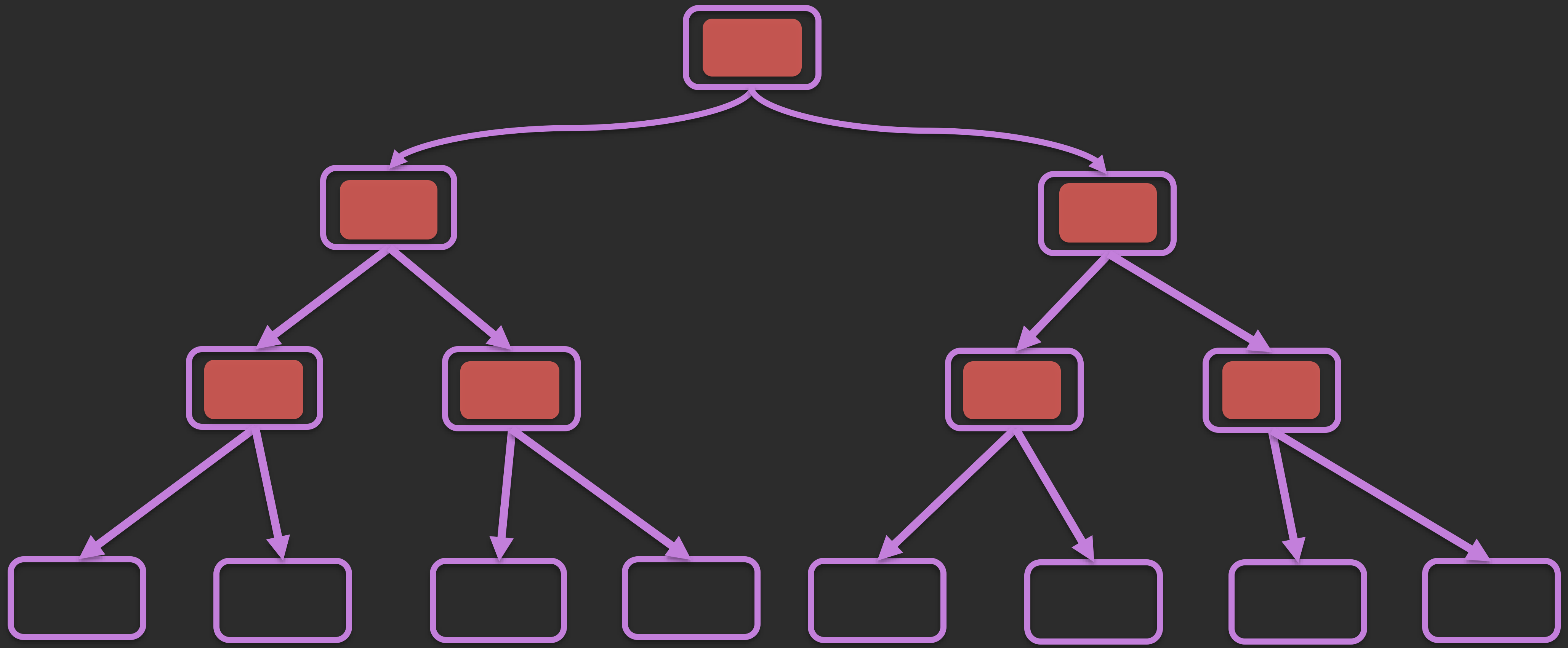
Suppose we have **a full tree** T_0 as shown below.



How many possible ways are there to prune this tree?

Pruning

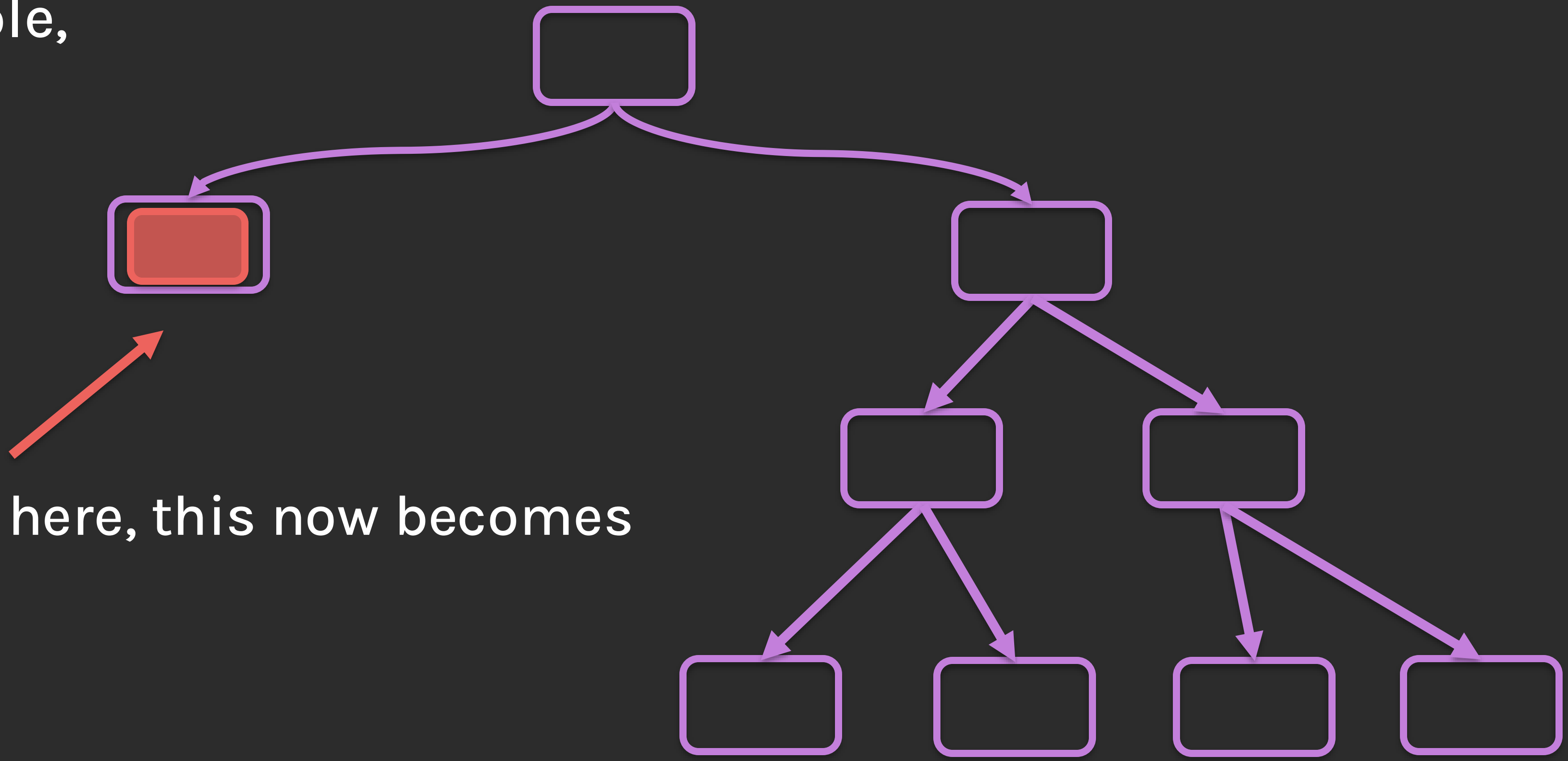
There are **7** possible pruning locations, which are shown with red rectangles.



Pruning

For each of those pruning locations, we will get 7 possible pruned trees, T^*.

For example,



If we pruned here, this now becomes a leaf node

Question: How do we choose the best pruned tree?

Pruning

We will choose the one that maximizes the difference of **cost complexity score** between a full tree and a pruned tree.

Quick recap: Cost complexity score is

Classification Error

$$C(T) = Error(T) + \alpha\,|T|$$

Decision tree

Complexity Parameter

Number of leaves in the tree

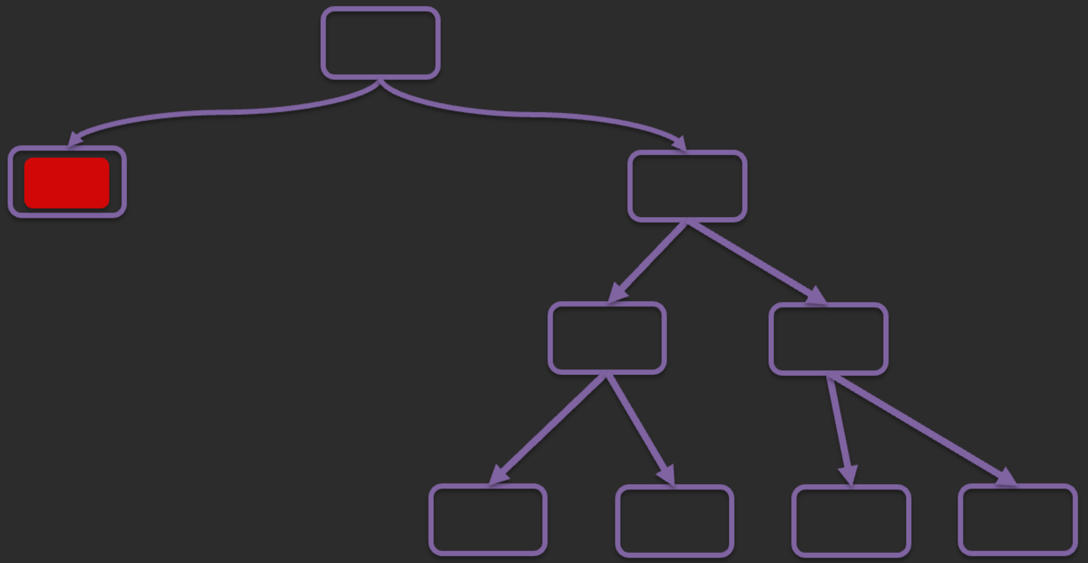
Pruning

Our goal is to maximize $C(T) - C(T^*)$

T^* is the pruned tree where T is the unpruned tree

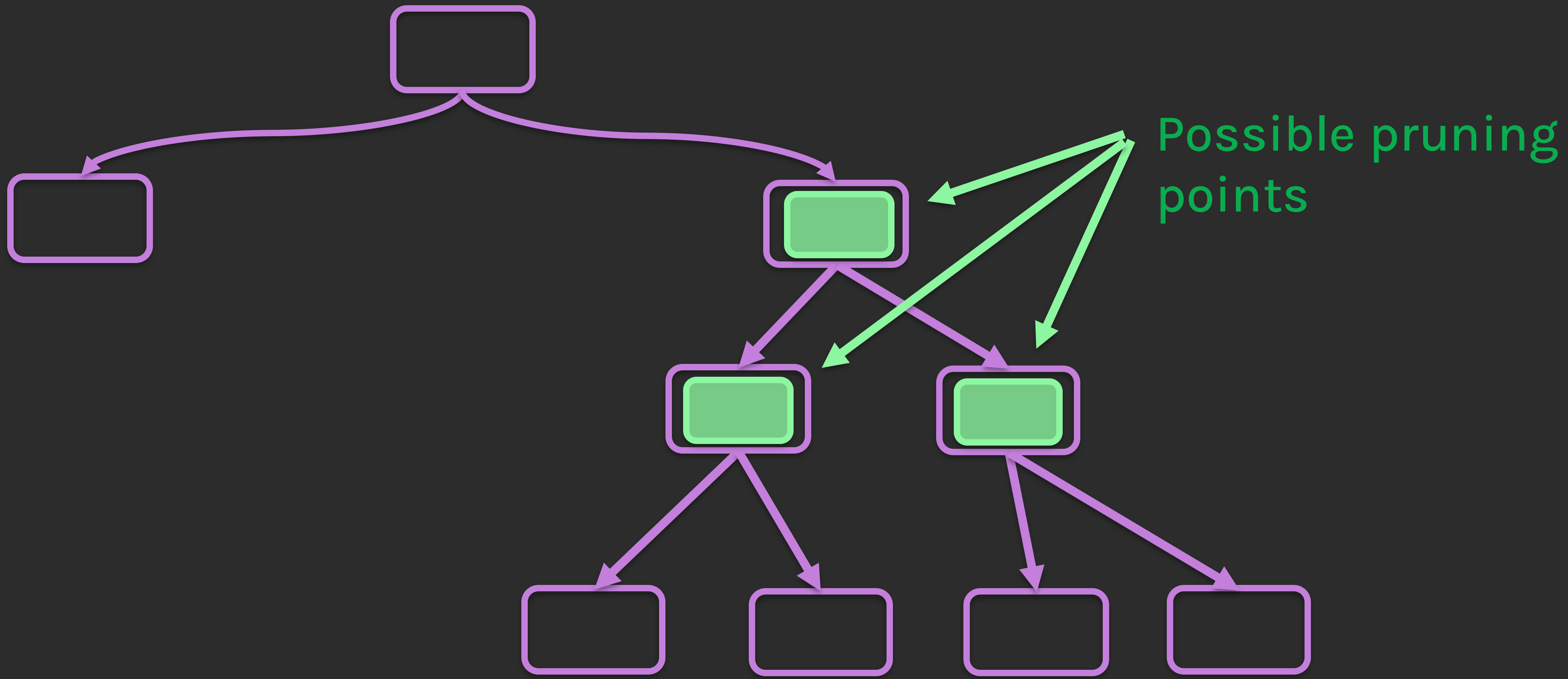
1. $\underset{T*}{argmax}\,[C(T) - C(T^*)]$
2. $\underset{T*}{argmax}\,[E(T) - E(T^*) + \alpha|T| - \alpha|T^*|]$
3. $\underset{T*}{argmax}\left[\frac{E(T)-E(T^*)|}{\alpha|T^*|-\alpha|T|} - 1\right]$ (divide by $\alpha|T^*| - \alpha|T|$)
4. $\underset{T*}{argmax}\left[\frac{E(T)-E(T^*)|}{\alpha|T^*|-\alpha|T|}\right]$
5. $\underset{T*}{argmin}\left[\frac{E(T)-E(T^*)|}{\alpha|T|-\alpha|T^*|}\right]$

This will result in a subtree, $\boldsymbol{T^{(1)}}$



Pruning

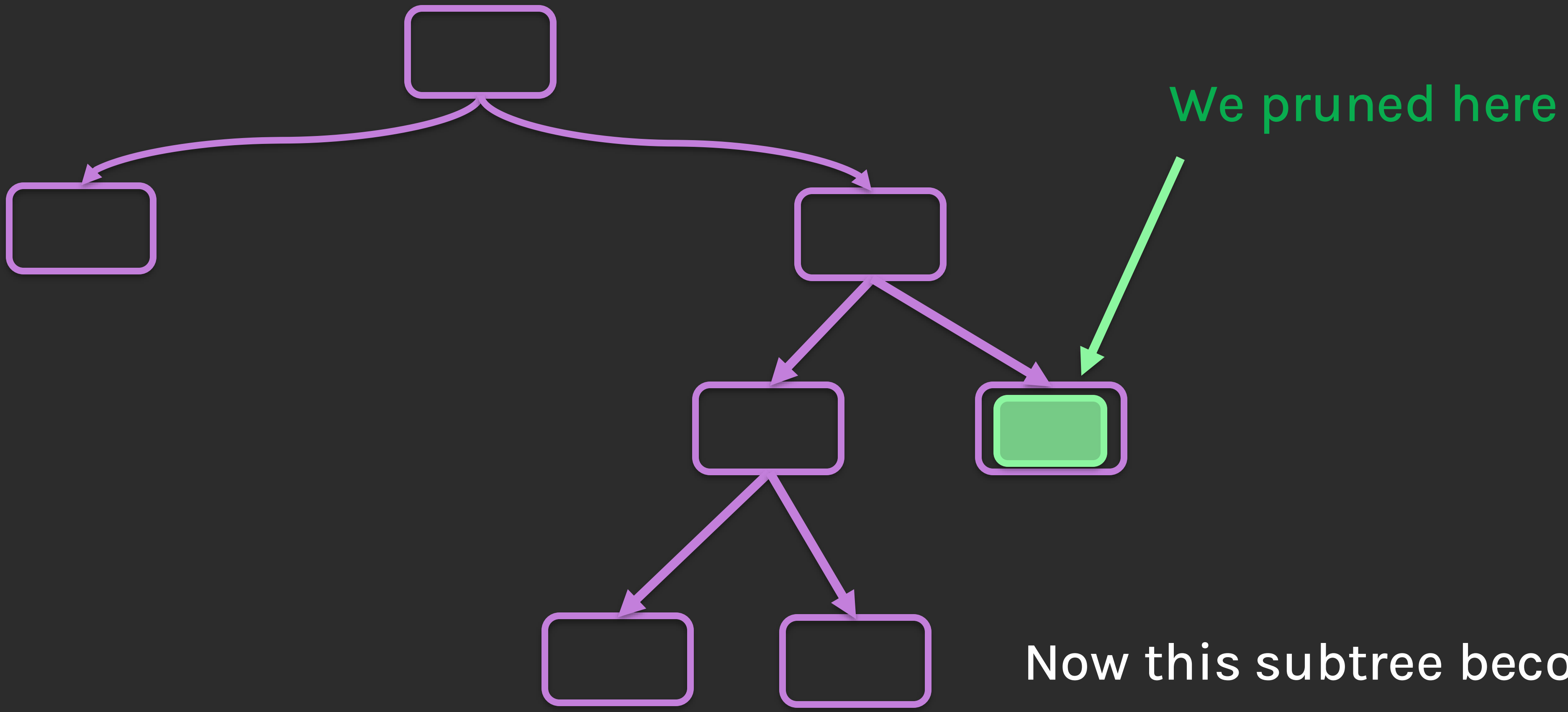
Now, we again consider all possible pruning from $\boldsymbol{T}^{(\mathbf{1})}$.



Pruning

We again minimize the ratio of the difference of the errors **over the difference in the complexity**

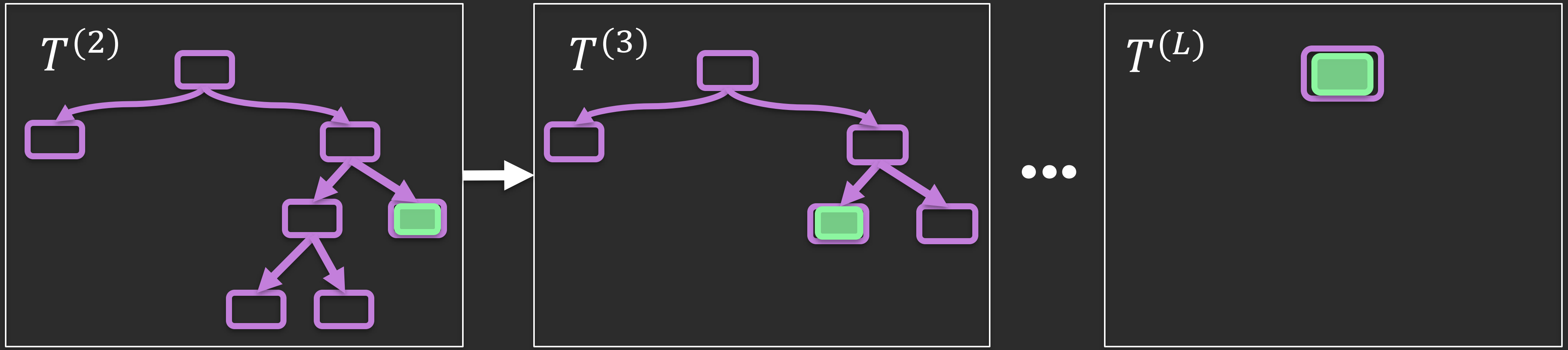
$$\underset{T*}{argmin}\left[\frac{E\left(T^{(1)}\right)-E(T^*)|}{\alpha|T^{(1)}|-\alpha|T^*|}\right]$$



Now this subtree becomes $T^{(2)}$

Pruning

We iterate this pruning process to obtain $T^{(2)}, T^{(3)}, \ldots, T^{(L)}$ where $T^{(L)}$ is the tree containing just the root of $T^{(0)}$.



We select the optimal tree $T^{(i)}$ by cross validation.

This is the T^* given α. Finally, we choose **the optimal** α by cross validation!

Summary

What is the main drawback of pre-defining stopping criteria for decision tree growth?

Pre-defined stopping criteria may lead to trees that are either too simple (underfitting) or too complex (overfitting) since it's challenging to determine the optimal stopping point beforehand.

Explain the alternative approach to using stopping conditions for decision tree construction.

Instead of stopping conditions, an alternative approach is to grow a large tree first, then prune it back to find a balanced structure, offering flexibility for optimal complexity.

What is the core concept behind pruning in decision trees?

Pruning simplifies a decision tree by removing branches with minimal impact on overall accuracy, enhancing the tree's ability to generalize.

Summary

Describe the formula used for calculating the cost complexity of a decision tree.

Cost complexity (C(T)) is calculated as: $C(T) = Error(T) + \alpha|T|$, where Error(T) is the classification error, α is the complexity parameter, and $|T|$ is the number of leaves.

In cost-complexity pruning, how does the complexity parameter (α) influence the trade-off between tree size and error?

The complexity parameter (α) adjusts the penalty for tree size, with higher values favoring smaller trees with potentially higher error and lower values favoring more complex trees with potentially lower error.

How do you determine the best pruned tree from a set of candidate trees generated by pruning?

The best pruned tree is selected by maximizing the difference in cost complexity scores, choosing the tree with the most significant reduction in complexity without a major increase in error.

Thank you