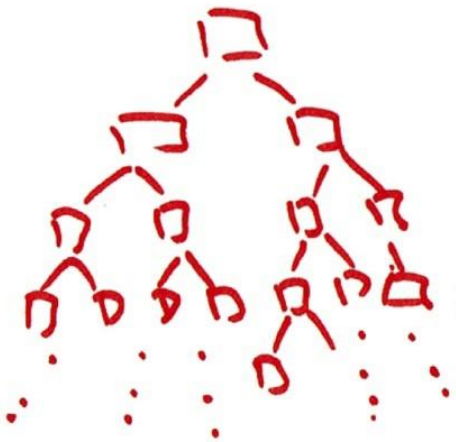
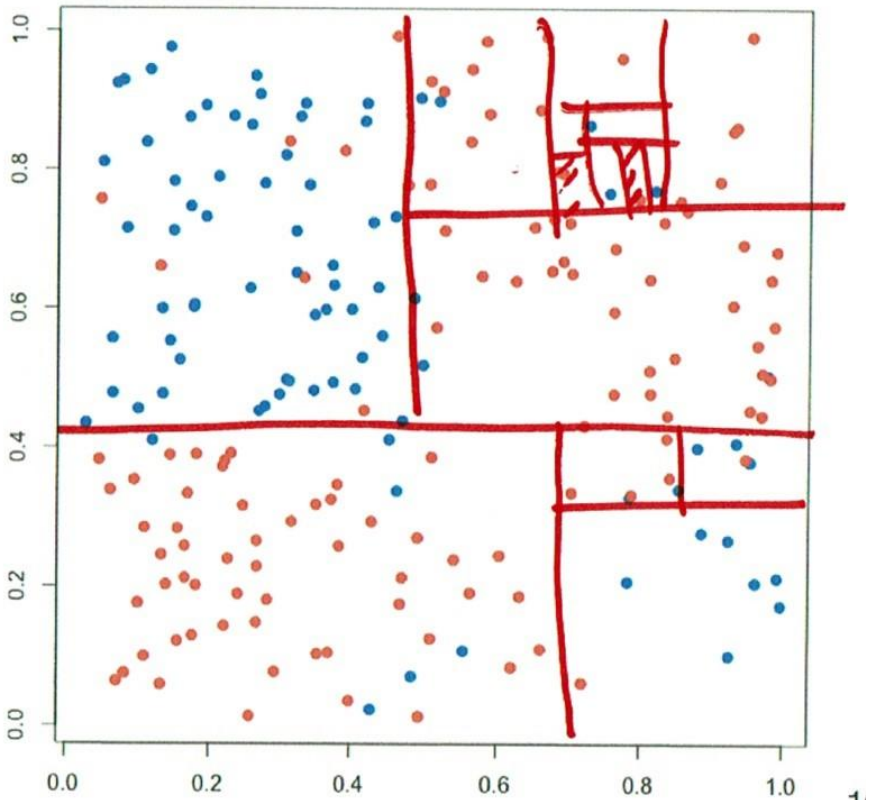
Decision trees are prone to 'overfitting'

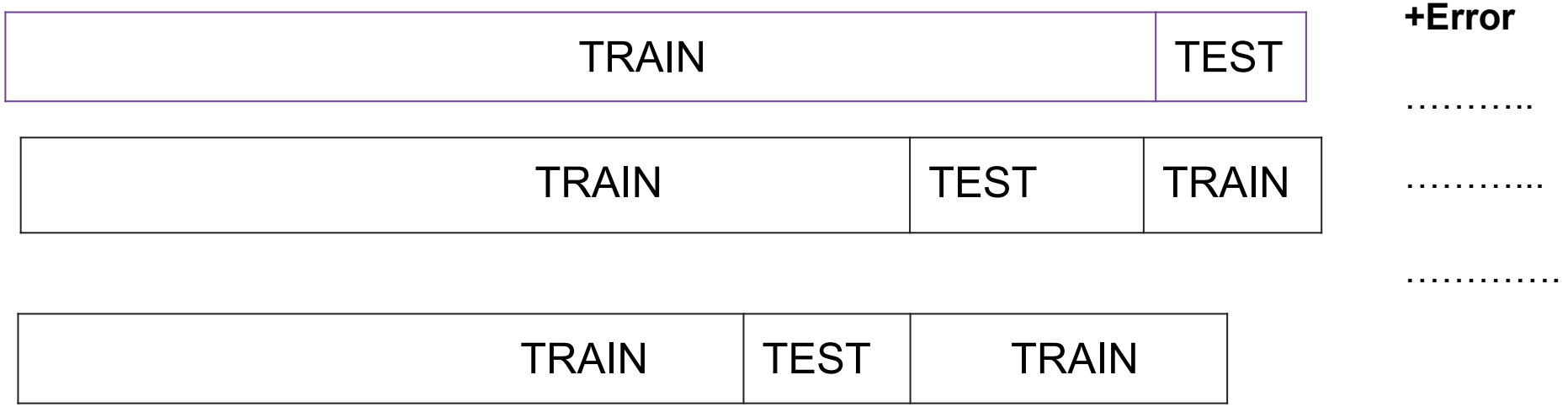
- Decision Tree is a powerful algorithm that can adapt well and capture various patterns in the data
- If allowed to grow fully, they become over-complex & tend to fit even the noise
- Thus, a fully grown tree may not 'generalize' well on test or new unseen data





Train	Test

M1	?
M2	?
M3	?



+Error

……..

……...

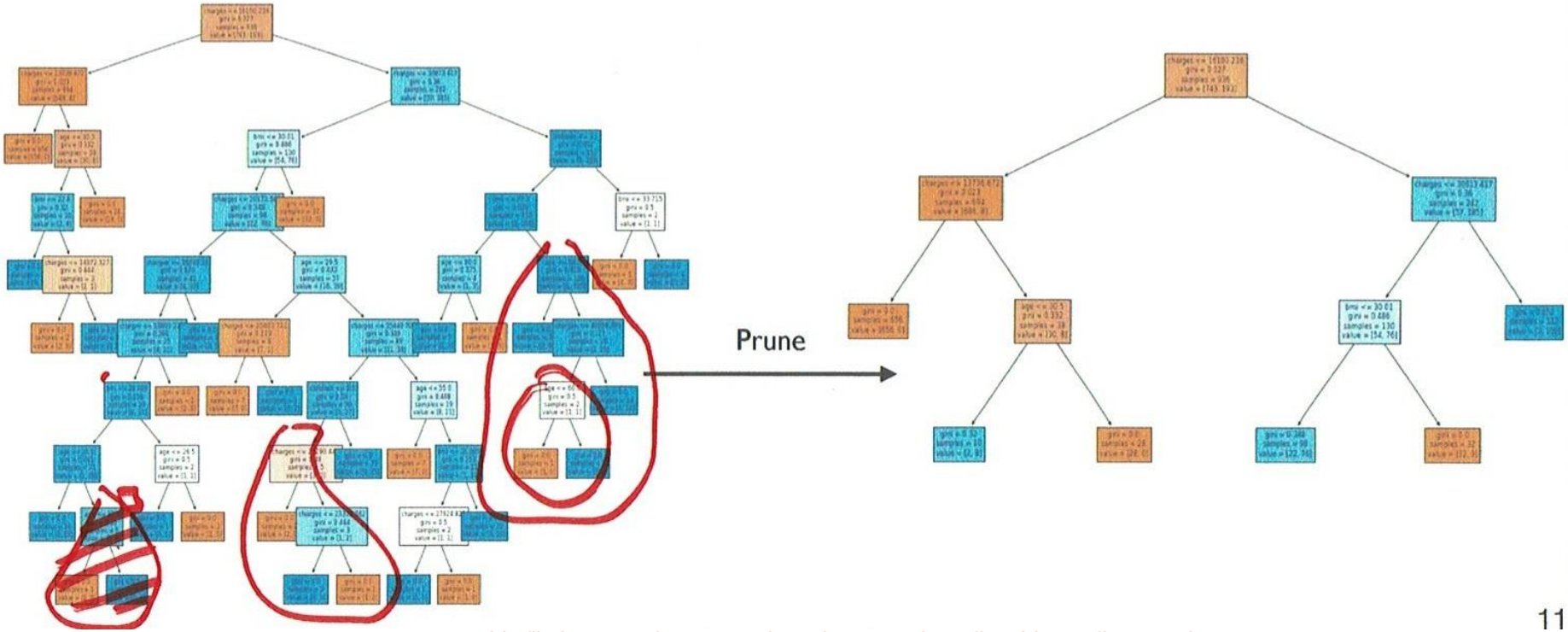
……

Post-Pruning: Cost-complexity pruning

- Starting from the Full tree, create a sequence of trees that are sequentially smaller (pruned)
- At each step the algorithm
  - try removing each possible subtree
  - find the 'relative error decrease per node' for that subtree - Complexity parameter, α
  - And remove the subtree with the minimum α
- With the list of subtrees, one usually reverts back to using cross-validation errors to find the best final pruned tree

Pruning

- Ideally we would like a tree that does not over-fit the given data
- One popular and simple way to prune a decision tree is by limiting the depth of the tree to avoid over fitting.
- For example the tree on the right below is generated with a max depth of 2 while the tree on the left has no depth restriction (and hence overfits the data)



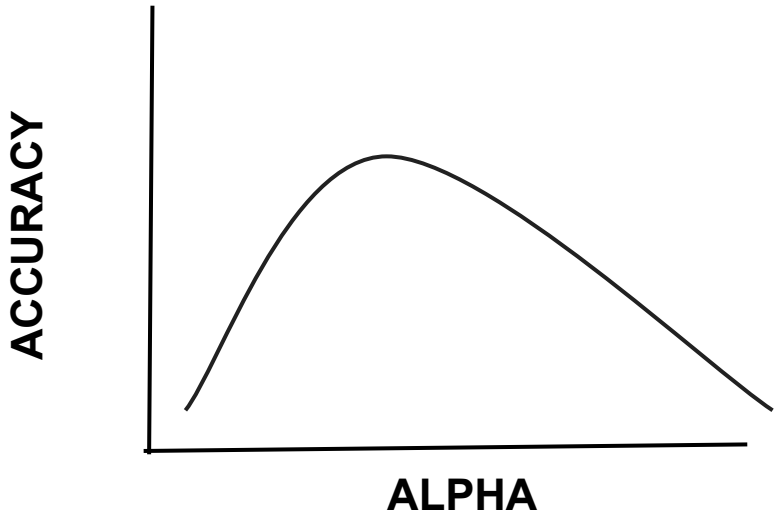
SMALL

COMPLEX

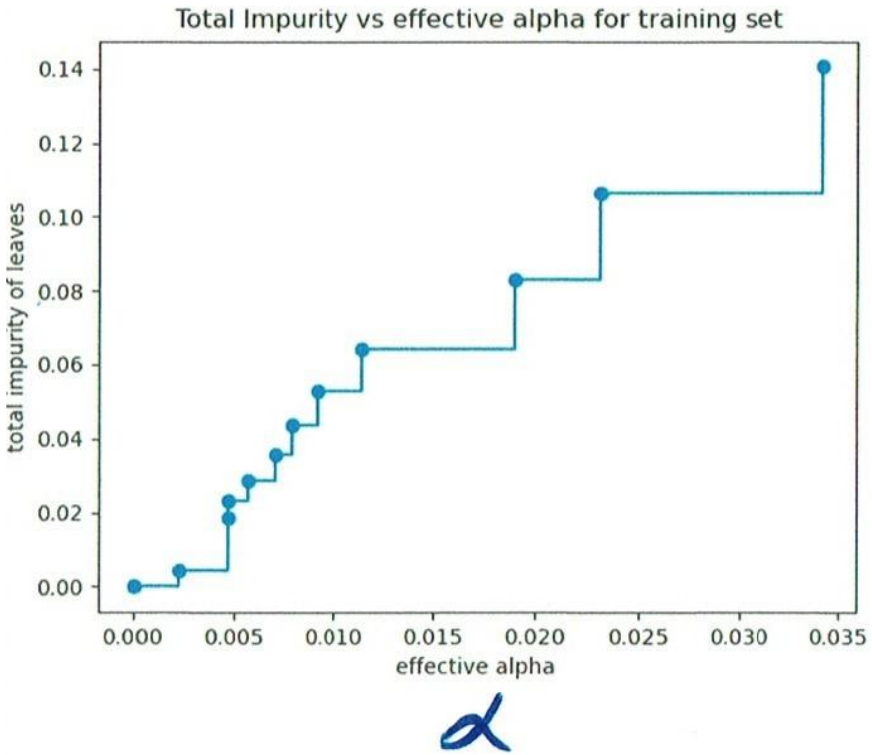
	ERROR	ALPHA
T_0		
T_1		
T_2		
T_3		
...		
....		
T_m		

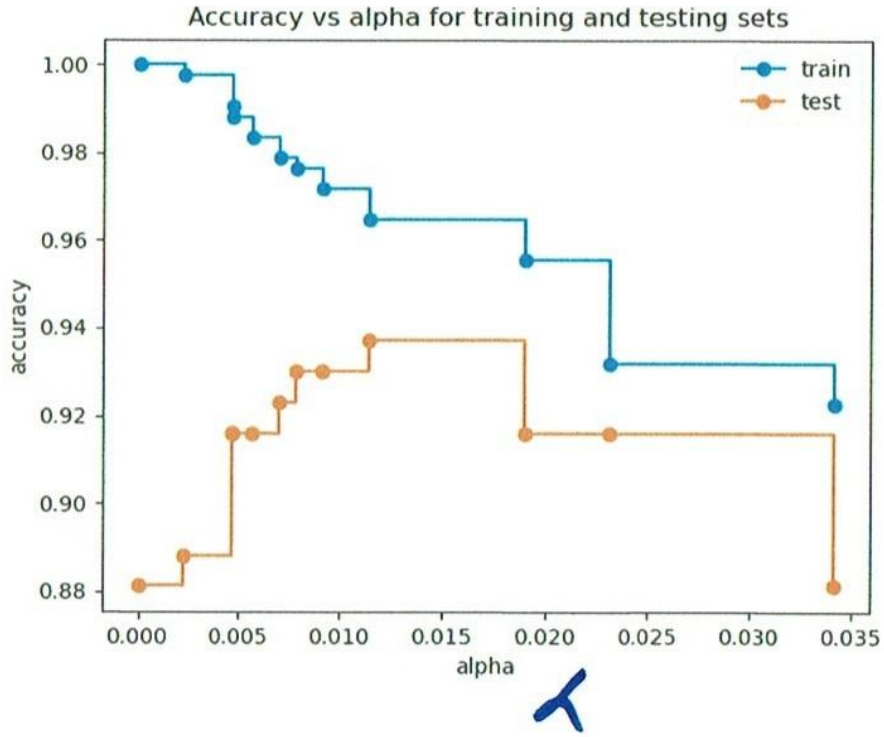
SIMPLE LARGE

Alpha decreases, impurity increases, complexity decreases



ALPHA = (Error (Pruned) - Error (original)) / (Number of nodes reduced)





Impurity Measures in Decision Trees

	GINI INDEX	ENTROPY	INFORMATION GAIN	VARIANCE
When to use	Classification	Classification	Classification	Regression
Formula	$1 - \Sigma p_i^2$	$-\Sigma p_i \log(p_i)$	$E(Y) - E(Y\|X)$	$\Sigma(x - \bar{x})^2/N$
Range	0 to 0.5 0 = most pure 0.5 = most impure	0 to 1 0 = most pure 1 = most impure	0 to 1 0 = less gain 1 = more gain	>=0
Characteristics	Easy to compute Non-additive	Computationally intensive Additive	Computationally intensive	The most common measure of spread