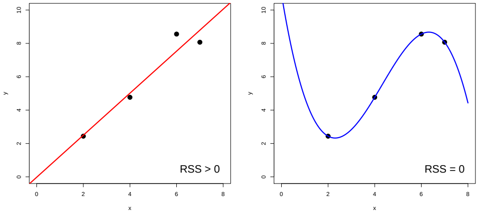
Model validation and tree pruning

Machine Learning for Ecology workshop

SEEC - Statistics in Ecology, Environment and Conservation

The problem of overfitting

- A model can be made to fit sample data arbitrarily well



- You are interested in how well your model does on *unseen* data
- Always do validation - **always always always!**

Model validation

Best practice

1. Divide your dataset in 3 parts: *training*, *validation* and *test* sets
2. Fit model on training data
3. Assess model on validation data
4. Choose model with the lowest *validation error*
5. Assess selected model on test data for final model $\Leftarrow$ this is your prediction error

Needs a lot of data

K-fold cross-validation

1. Divide data into K equal-size *folds*
2. Fit model model to all data excluding the kth fold
3. Assess performance using the kth fold
4. Repeat for all folds
5. Combine validation errors across folds

Most often $k = 10$. $K = n$, is *leave-one-out CV*

Cross-Validation

Example: 4-fold cross-validation for the linear model

	x	y
6	0.26	1.39
15	0.63	1.59
8	0.38	1.19
16	0.66	1.57
17	0.73	1.89
1	0.00	1.03
18	0.84	1.80
12	0.52	1.19
7	0.33	1.50
20	0.99	1.99
10	0.43	1.34
5	0.19	1.36
11	0.49	1.59
9	0.38	1.27
19	0.86	2.07
13	0.55	1.62
14	0.63	2.11
4	0.11	0.75
3	0.02	1.08
2	0.01	0.81

Randomise!

Cross-Validation

Example: 4-fold cross-validation for the linear model

	x	y	$\hat{y}$	$\hat{e}^2$	
6	0.26	1.39	1.24	0.023	Test set
15	0.63	1.59	1.68	0.008	
8	0.38	1.19	1.38	0.036	
16	0.66	1.57	1.71	0.021	
17	0.73	1.89	1.79	0.010	
1	0.00	1.03			Training set $\hat{y} = 0.932 + 1.184x$
18	0.84	1.80			
12	0.52	1.19			
7	0.33	1.50			
20	0.99	1.99			
10	0.43	1.34			
5	0.19	1.36			
11	0.49	1.59			
9	0.38	1.27			
19	0.86	2.07			
13	0.55	1.62			
14	0.63	2.11			
4	0.11	0.75			
3	0.02	1.08			
2	0.01	0.81			

Cross-Validation

Example: 4-fold cross-validation for the linear model

	x	y	$\hat{y}$	$\hat{e}^2$
6	0.26	1.39	1.24	0.023
15	0.63	1.59	1.68	0.008
8	0.38	1.19	1.38	0.036
16	0.66	1.57	1.71	0.021
17	0.73	1.89	1.79	0.010
1	0.00	1.03	0.87	0.026
18	0.84	1.80	2.02	0.046
12	0.52	1.19	1.58	0.149
7	0.33	1.50	1.32	0.031
20	0.99	1.99	2.22	0.053
10	0.43	1.34		
5	0.19	1.36		
11	0.49	1.59		
9	0.38	1.27		
19	0.86	2.07		
13	0.55	1.62		
14	0.63	2.11		
4	0.11	0.75		
3	0.02	1.08		
2	0.01	0.81		

Test set

Training set

$\hat{y} = 0.867 + 1.363x$

Cross-Validation

Example: 4-fold cross-validation for the linear model

	x	y	$\hat{y}$	$\hat{e}^2$
6	0.26	1.39	1.24	0.023
15	0.63	1.59	1.68	0.008
8	0.38	1.19	1.38	0.036
16	0.66	1.57	1.71	0.021
17	0.73	1.89	1.79	0.010
1	0.00	1.03	0.87	0.026
18	0.84	1.80	2.02	0.046
12	0.52	1.19	1.58	0.149
7	0.33	1.50	1.32	0.031
20	0.99	1.99	2.22	0.053
10	0.43	1.34	1.42	0.006
5	0.19	1.36	1.14	0.049
11	0.49	1.59	1.48	0.011
9	0.38	1.27	1.36	0.010
19	0.86	2.07	1.92	0.023
13	0.55	1.62		
14	0.63	2.11		
4	0.11	0.75		
3	0.02	1.08		
2	0.01	0.81		

Training set
$\hat{y} = 0.921 + 1.154x$

Test set

Cross-Validation

Example: 4-fold cross-validation for the linear model

	x	y	$\hat{y}$	$\hat{e}^2$
6	0.26	1.39	1.24	0.023
15	0.63	1.59	1.68	0.008
8	0.38	1.19	1.38	0.036
16	0.66	1.57	1.71	0.021
17	0.73	1.89	1.79	0.010
1	0.00	1.03	0.87	0.026
18	0.84	1.80	2.02	0.046
12	0.52	1.19	1.58	0.149
7	0.33	1.50	1.32	0.031
20	0.99	1.99	2.22	0.053
10	0.43	1.34	1.42	0.006
5	0.19	1.36	1.14	0.049
11	0.49	1.59	1.48	0.011
9	0.38	1.27	1.36	0.010
19	0.86	2.07	1.92	0.023
13	0.55	1.62	1.55	0.005
14	0.63	2.11	1.63	0.232
4	0.11	0.75	1.12	0.135
3	0.02	1.08	1.03	0.002
2	0.01	0.81	1.02	0.044

Training set
$\hat{y} = 1.012 + 0.985x$

Test set

Cross-Validation

Example: 4-fold cross-validation for the linear model

	x	y	$\hat{y}$	$\hat{e}^2$
6	0.26	1.39	1.24	0.023
15	0.63	1.59	1.68	0.008
8	0.38	1.19	1.38	0.036
16	0.66	1.57	1.71	0.021
17	0.73	1.89	1.79	0.010
1	0.00	1.03	0.87	0.026
18	0.84	1.80	2.02	0.046
12	0.52	1.19	1.58	0.149
7	0.33	1.50	1.32	0.031
20	0.99	1.99	2.22	0.053
10	0.43	1.34	1.42	0.006
5	0.19	1.36	1.14	0.049
11	0.49	1.59	1.48	0.011
9	0.38	1.27	1.36	0.010
19	0.86	2.07	1.92	0.023
13	0.55	1.62	1.55	0.005
14	0.63	2.11	1.63	0.232
4	0.11	0.75	1.12	0.135
3	0.02	1.08	1.03	0.002
2	0.01	0.81	1.02	0.044

$$\text{CV error} = \frac{1}{n}\sum_{i=1}^{n}\hat{e}_i^2$$
$$= 0.046$$

Overfitting of Regression Trees

- CART produces a complex tree with many splits
- Simpler trees may yield better out-of-sample predictions
- One alternative: grow tree only until the decrease in RSS $<$ threshold
- Better: grow a large tree and then *prune* it back to obtain a smaller *subtree*

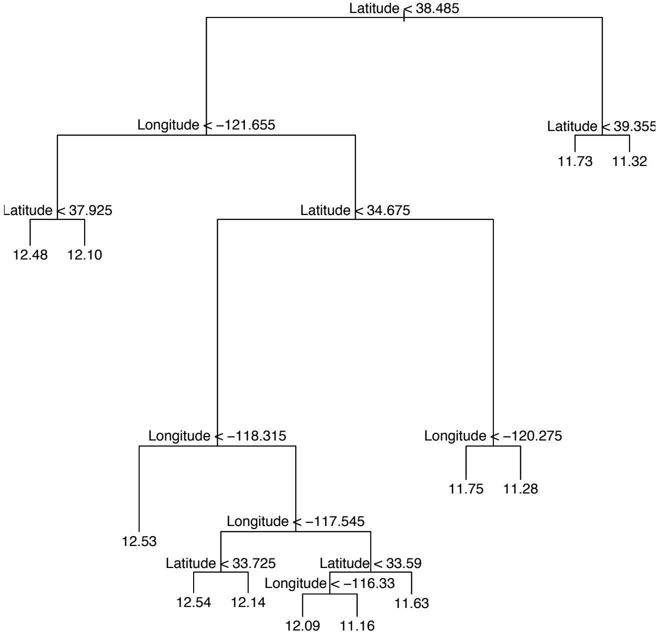
Tree Pruning

- **Goal**: find the subtree with the lowest test error
- Not computationally feasible to assess all subtrees
- Obtain a sequence of trees by iteratively pruning the full tree
- Called *cost complexity pruning*

How does Cost Complexity Pruning work?

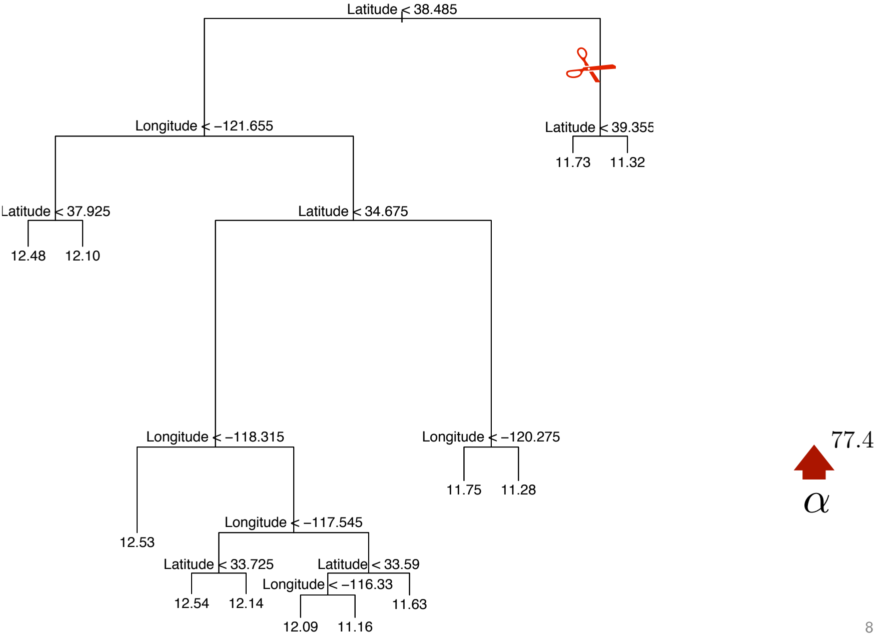
- Use the penalised RSS: $\text{RSS}_\alpha = RSS + \alpha|T|$
- $\alpha \geq 0$ is a tuning parameter and $|T|$ is number of terminal nodes
- For each value of α, find subtree T that minimises RSS_α
- As α increases, branches get pruned from the tree in a nested fashion
- Thus obtain a sequence of subtrees as a function of α

Tree Pruning

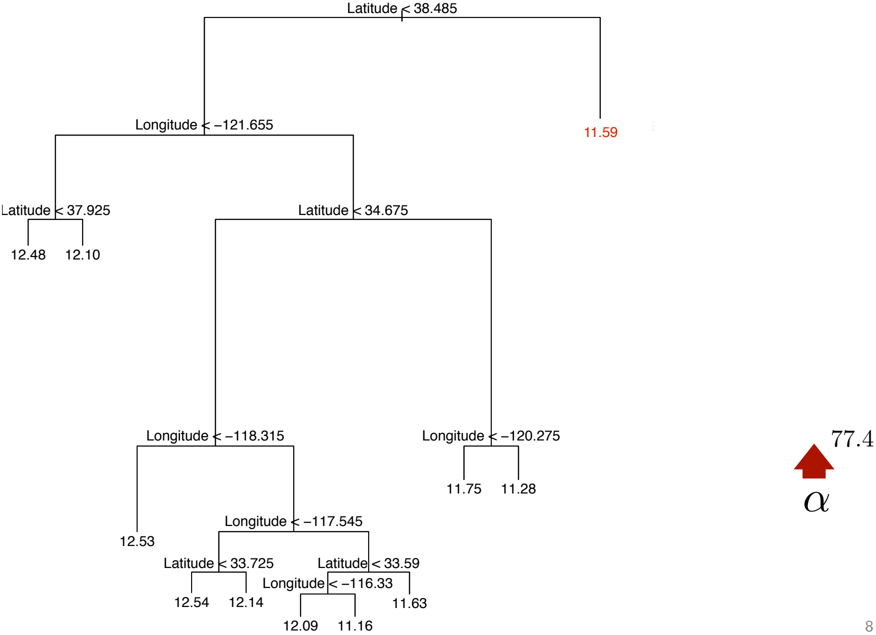


$\alpha = 0$

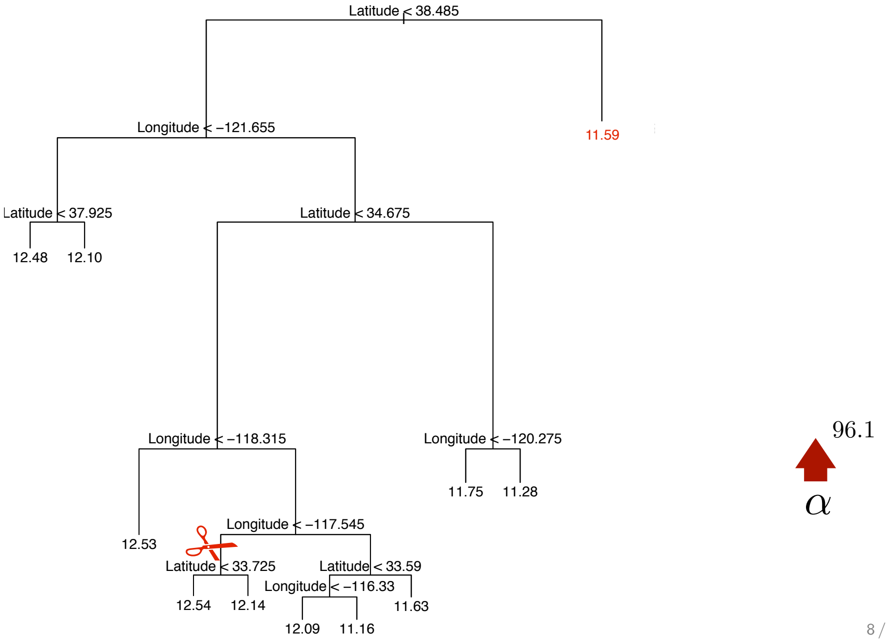
Tree Pruning



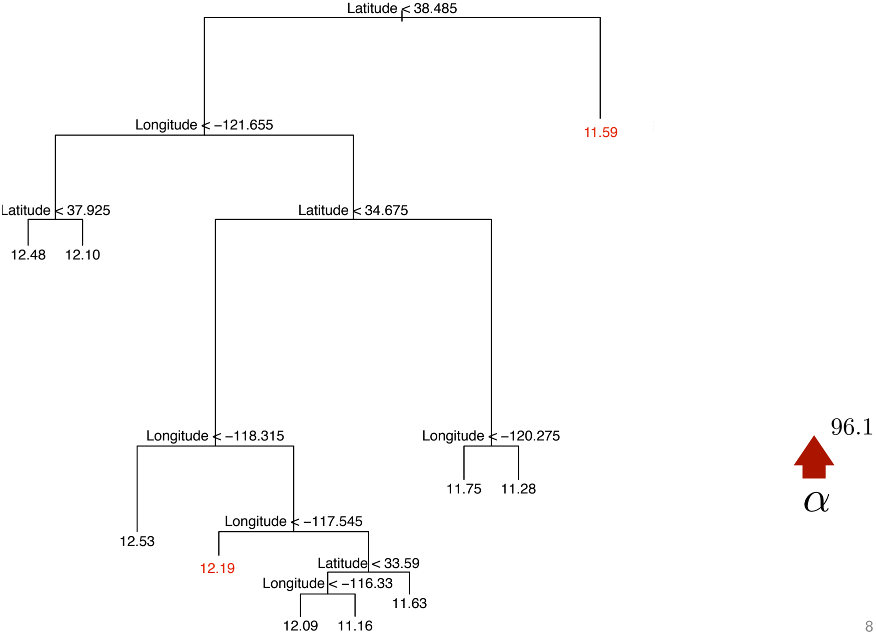
Tree Pruning



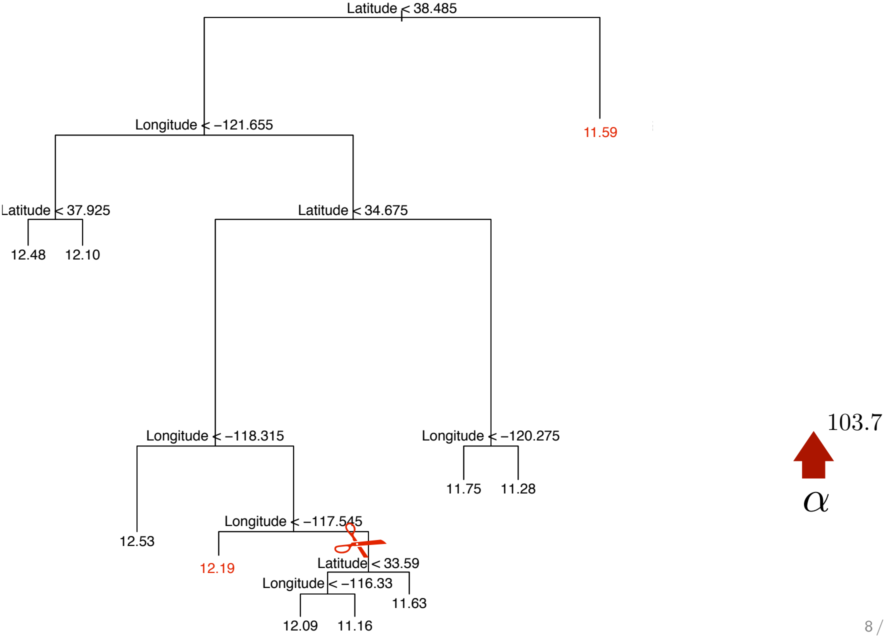
Tree Pruning



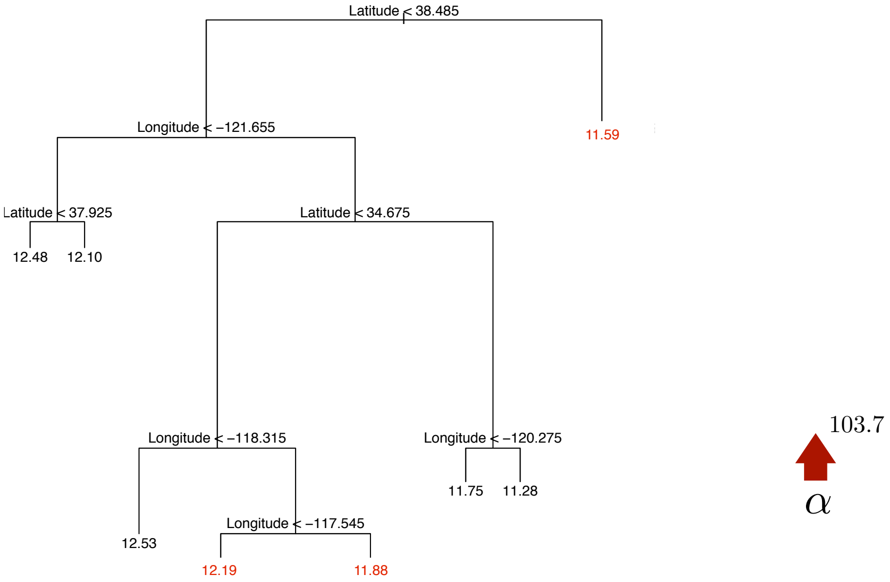
Tree Pruning



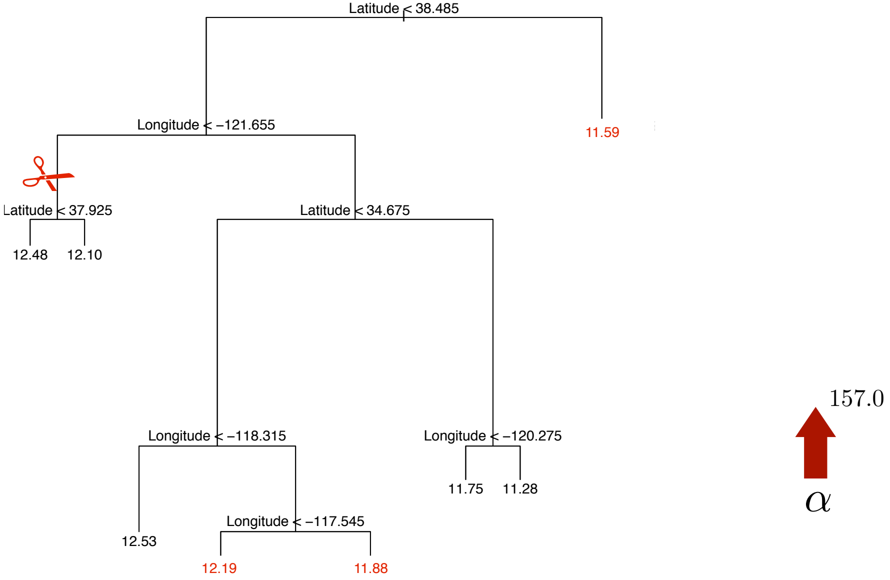
Tree Pruning



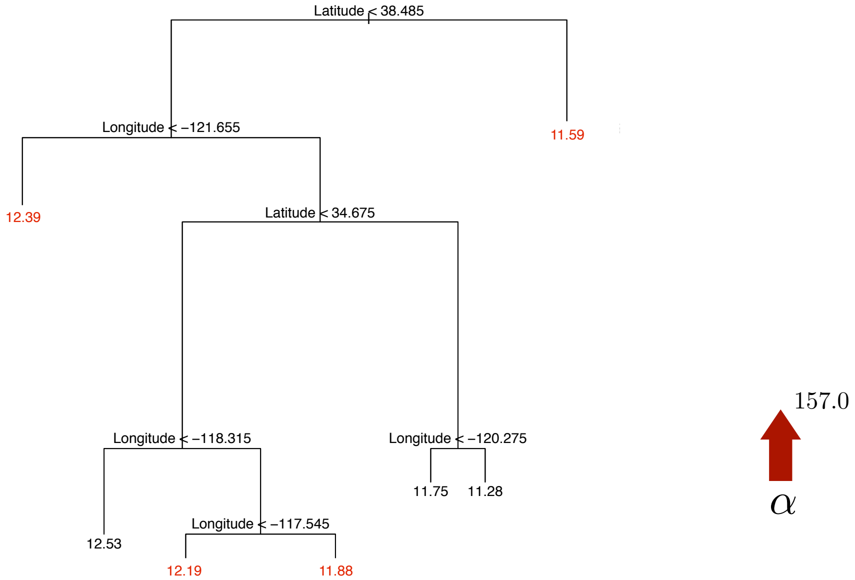
Tree Pruning



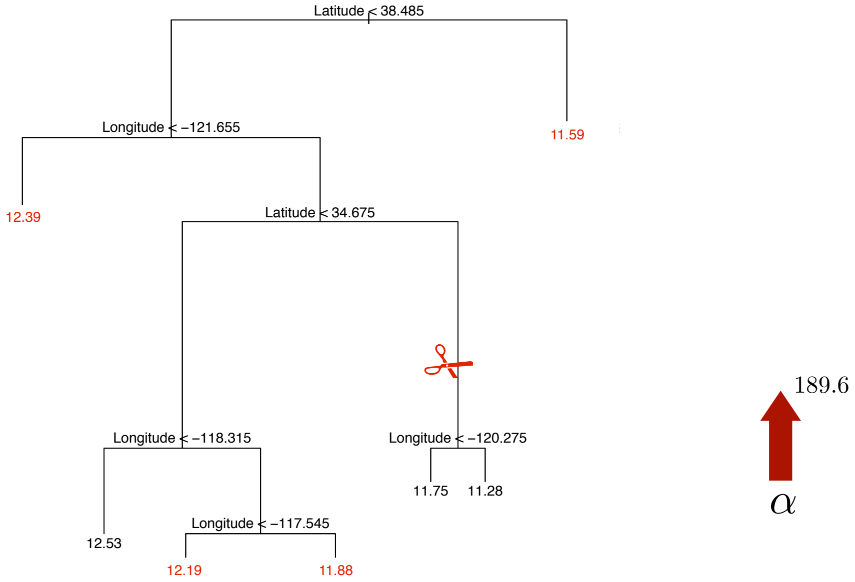
Tree Pruning



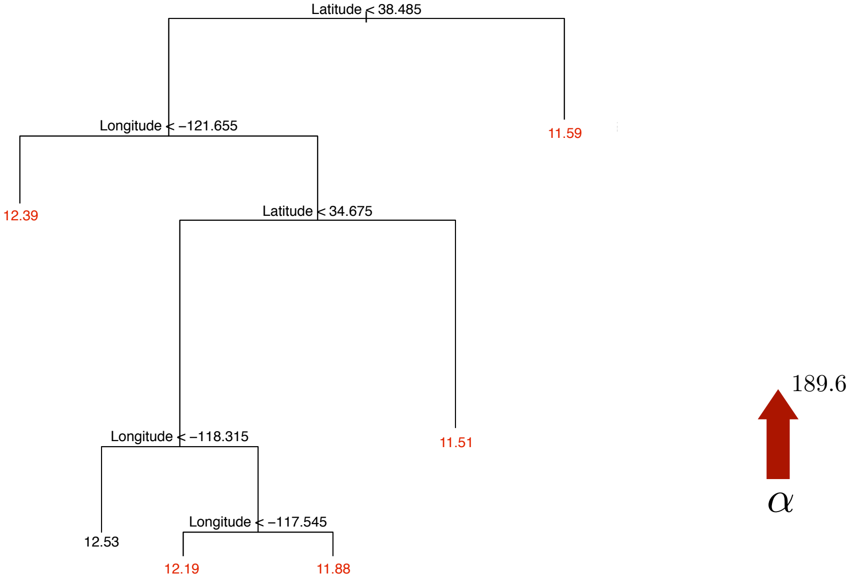
Tree Pruning



Tree Pruning

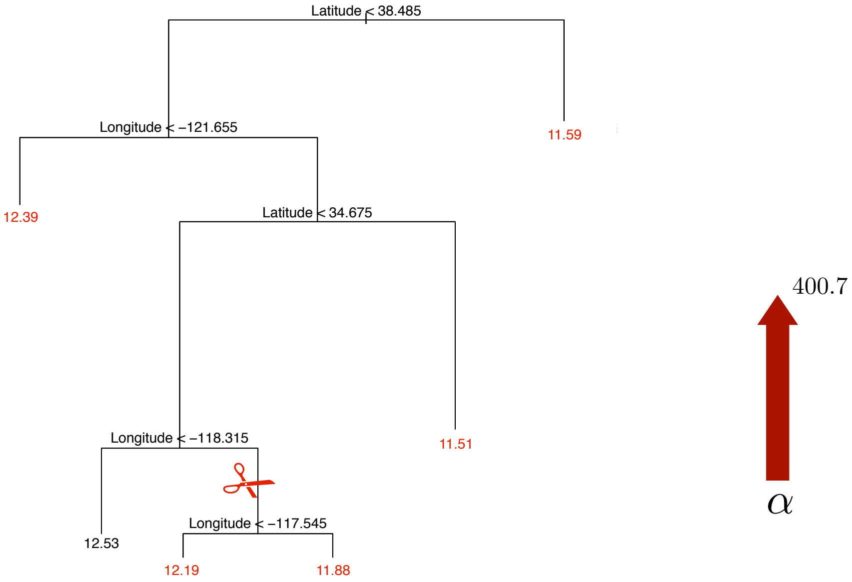


Tree Pruning

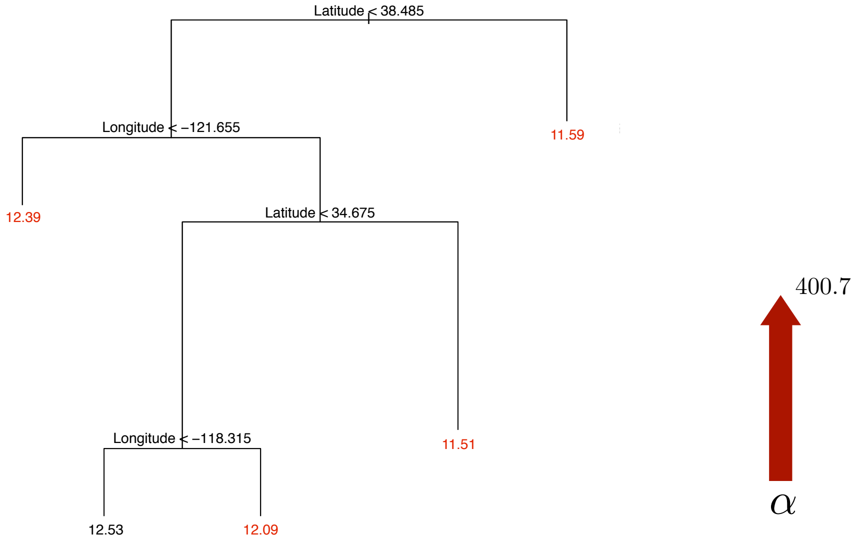


α

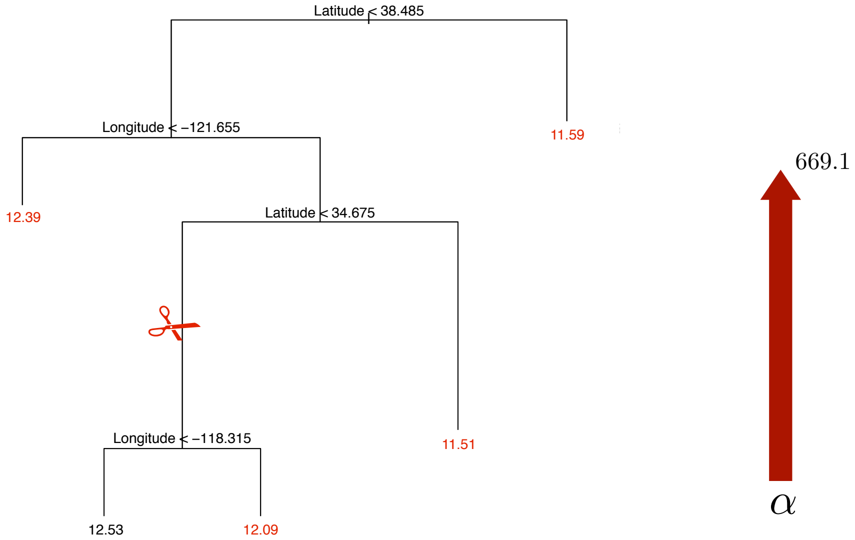
Tree Pruning



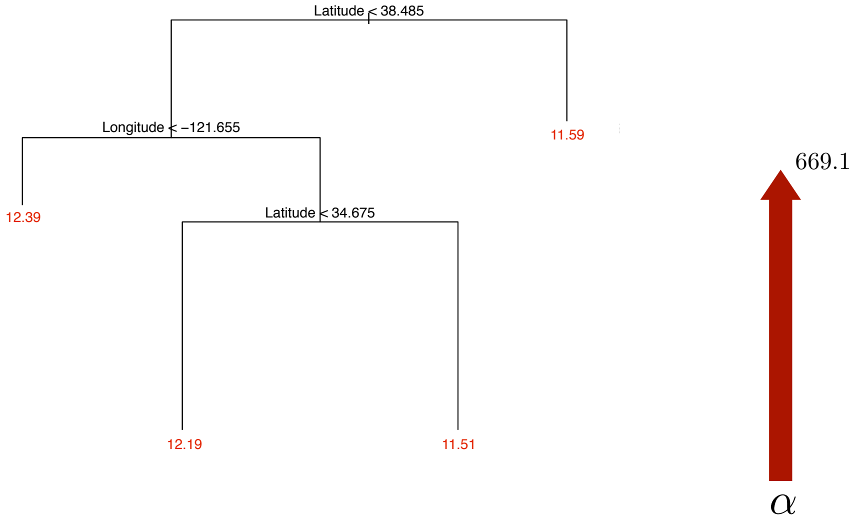
Tree Pruning



Tree Pruning



Tree Pruning



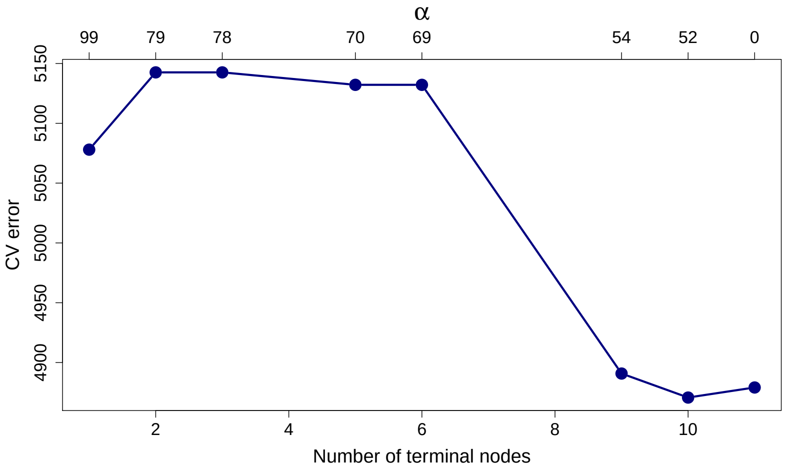
Choosing α

1. Using the full dataset, determine the critical values of $\alpha \geq 0$ that produce nested subtrees of different sizes

2. Compute the test error associated with each of these α values using K-fold cross-validation

3. Choose the α value with the lowest test error and report the corresponding subtree for the full dataset

Choosing α



Summary

- Importance of model validation
- Training, validation, test & cross-validation
- Tree pruning

- **Next:** Ensemble models: bagging, boosting, random forests