SDMs: the algorithms

Wednesday 4th, September

Grant Humphries

Overview

- Some terminology
- Broad methodologies
- The basics…
- Decision trees and related algorithms
- Hands-on fun
  - Data structuring (tidyverse)
  - Coding a model
  - Exploring parameters

Terminology

Term	Definition
Frequentist statistics	Statistics defined by frequency distributions (p-value type statistics). E.g., t-tests, logistic regression, generalized linear models
Machine learning	A school of algorithms that ‘learn’ patterns from data. Often do not require assumptions to be met about the data.
AUC	Area under the curve: a metric used to assess how accurate predictions from a classification model are
Classification	A type of modeling where the dependent variable (target variable) is a category (in our case, 0/1, but could be more)
Regression	A type of modeling where the dependent variable is continuous
Descriptor	A covariate / predictor variable in a model (values of X in Y=mX+b)
Target	The dependent variable of a model (values of Y in Y=mX+b)

Commonly used Algorithms

- Frequentist methods for presence/absence
  - Generalized Linear Models (logistic regression)
  - Generalized Additive Models
- Machine learning methods for p/a
  - Multivariate adaptive regression splines
  - Classification and Regression trees
  - Boosted regression trees (generalized boosted regression models)
  - Random forests
  - GARP (genetic algorithm for rule-set producing)

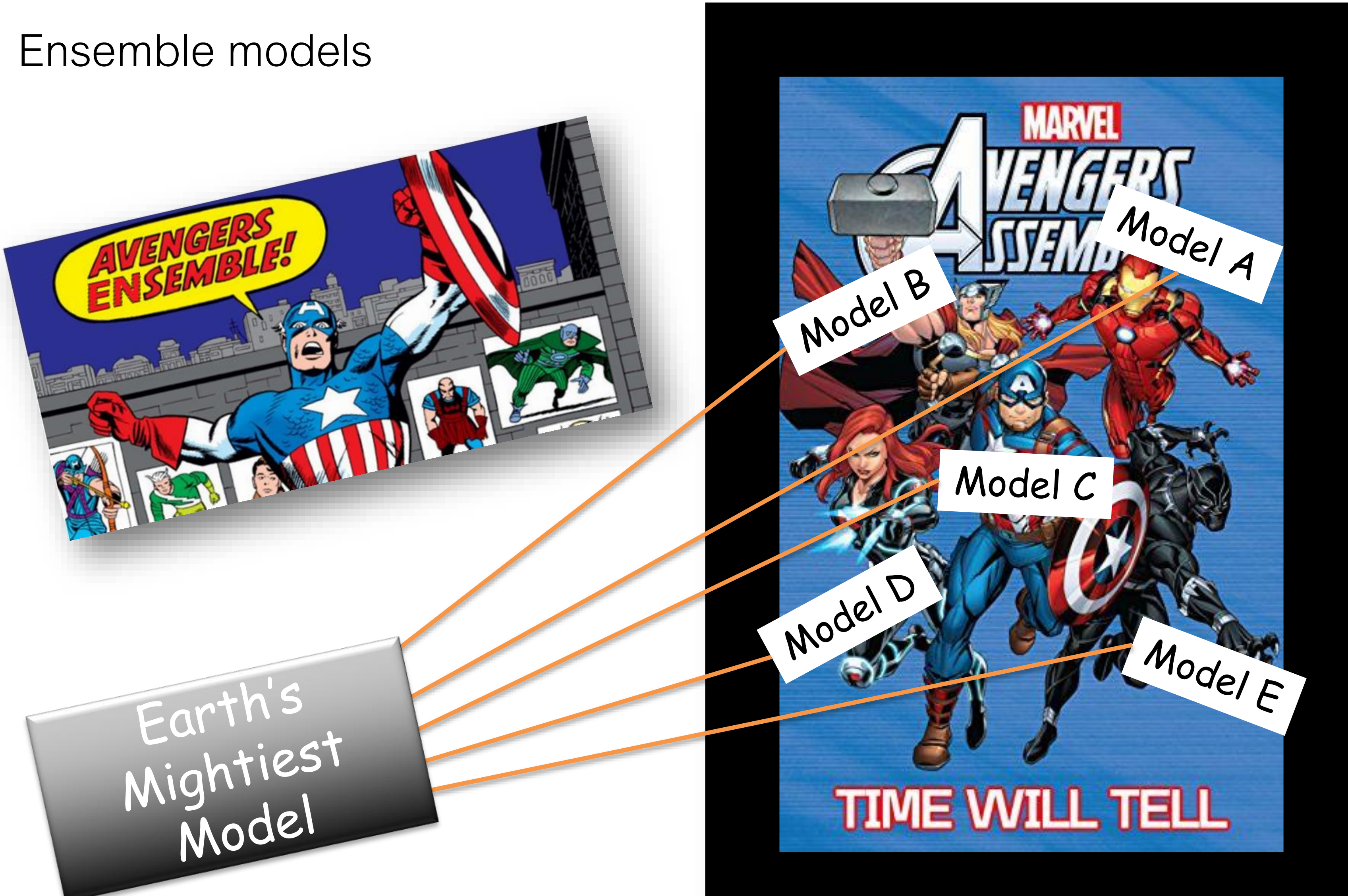
Other algorithms

- Neural networks
- Support vector machines

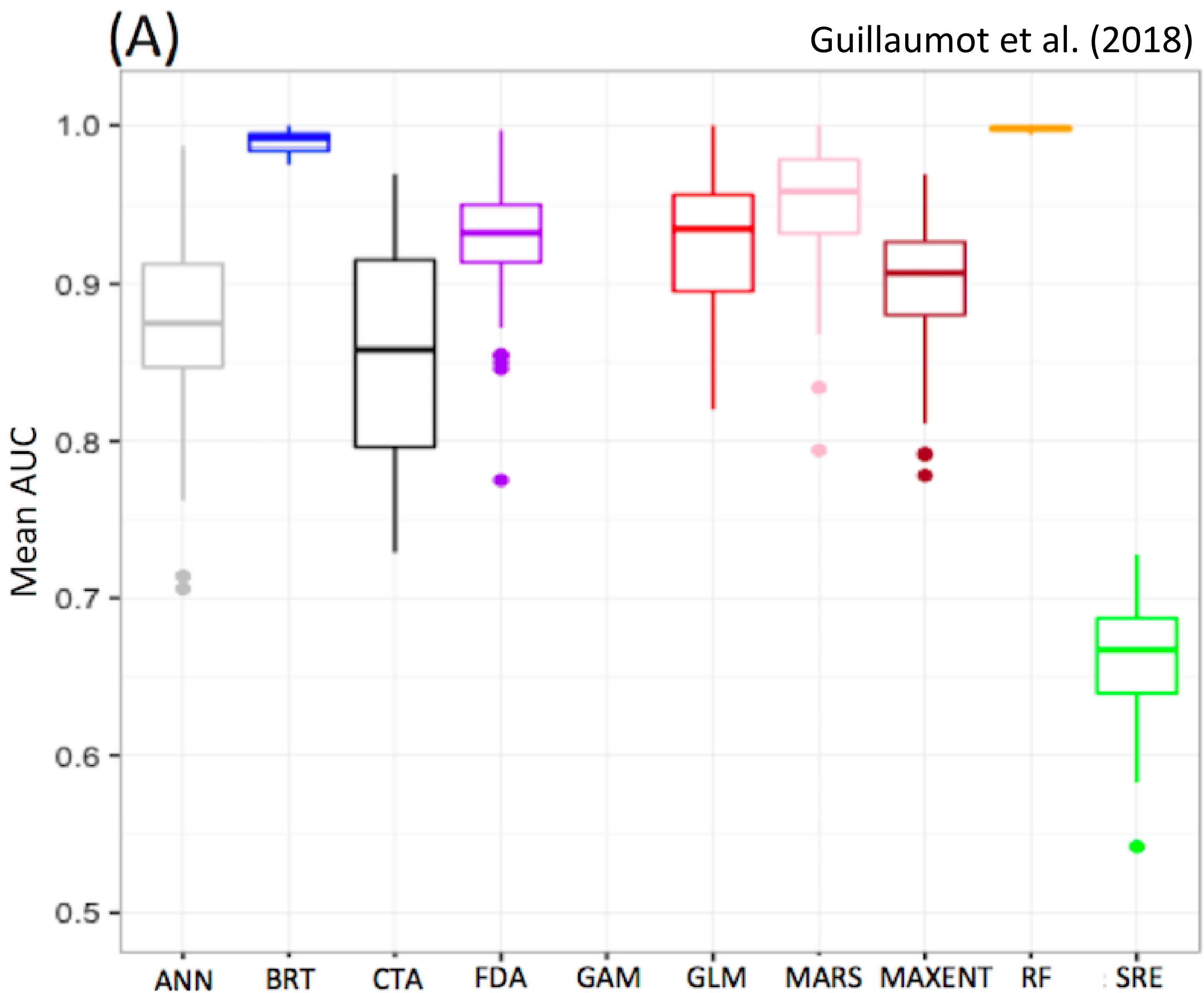
- Presence-only
  - ENFA (Ecological Niche Factor Analysis)
  - Maxent (Maximum entropy)
  - Poisson point process models

A more complete list of algorithms and some references can be found in the course material

Ensemble models

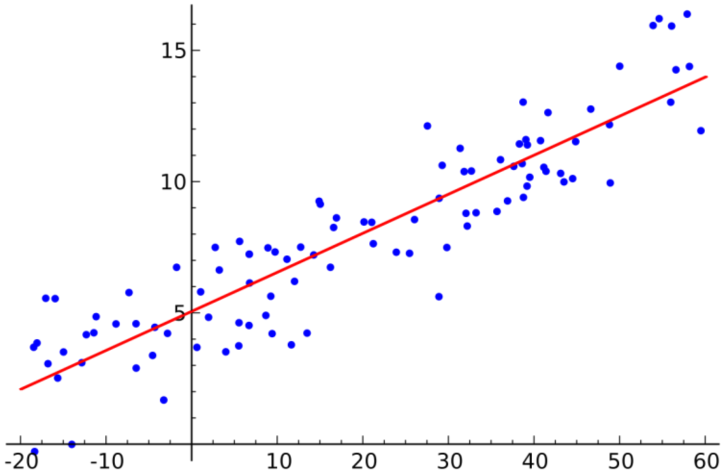


A comparison



The Generalized Linear Model (GLM)

$$Y = X\beta + e$$

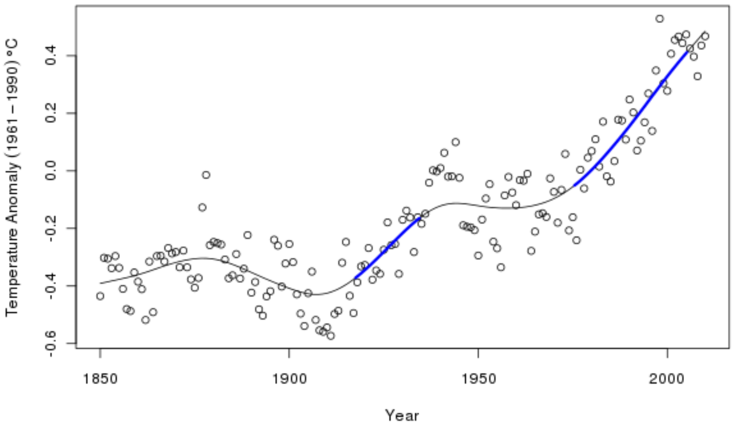


Assumptions

1. Errors are normally distributed
2. Error is the same across all observations
3. Variance is homogeneous

The Generalized Additive Model (GAM)

$$Y = s(X)\beta + e$$



Assumptions

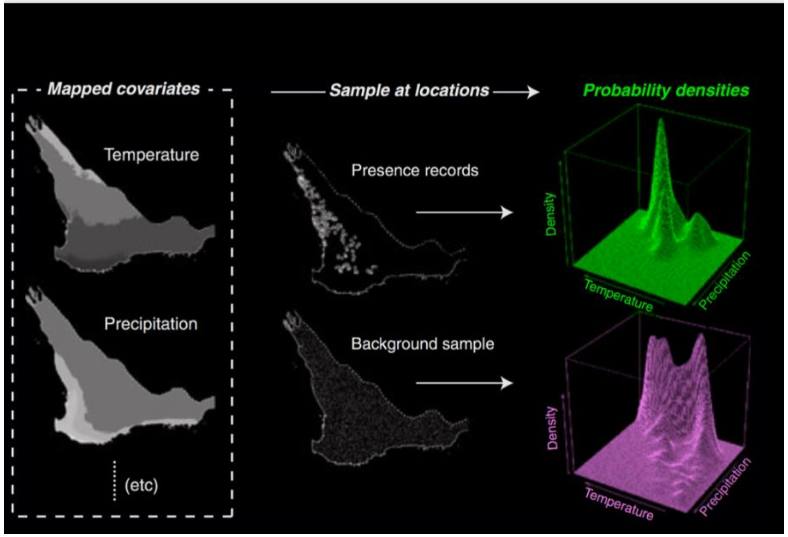
1. Errors are normally distributed
2. Error is the same across all observations
3. Variance is homogeneous

MAIN difference with GLM is that GAMs include the smoothing parameter, which makes it very flexible

MaxEnt (Maximum Entropy) – presence only modelling

Maxent uses the Gibb's probability distribution

The algorithm uses what we know of where species lie in space (i.e., the environmental envelope) and selects the parameters that create the most 'spread out' distribution (i.e. the maximum entropy).

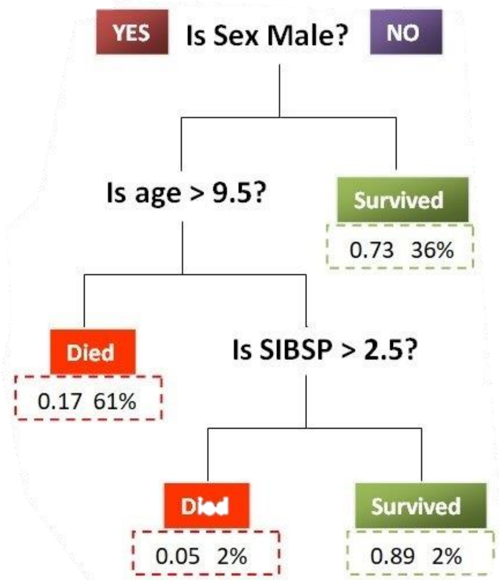


Elith et al. 2010

Machine learning!

Classification and regression trees

Developed in the 1980s by L Breiman and J Friedman



- Basically a conditional (IF/AND/OR) rule set generated by the data (a tree)
- Every split has two branches
- The number of splits and size of the 'tree' can be controlled

- The target can be categorical (classification tree), or continuous (regression), and how the splits are generated depends on the target

The basis for boosted regression trees AND Random forests

HOW IS A TREE CONSTRUCTED?

Recursive partitioning / Greedy splitting

- Tree ‘splits’ are determined by the splits that best lower the value of the **cost function**

Classification

- Gini index
= ‘purity’ of each ‘node’

- Standard deviation

Regression

- Sum squared errors

- Mean squared error

https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/

Occurrence	Sediment	Slope	Wind
0	Sand	Strong	North
1	Mud	Weak	North
1	Sand	Medium	South
0	Sand	Strong	East
0	Sand	Medium	West
1	Sand	Strong	South
1	Sand	Strong	East
1	Mud	Weak	East
0	Mud	Medium	East
1	Sand	Weak	North
1	Mud	Strong	West
0	Mud	Weak	West
0	Sand	Medium	West
0	Sand	Medium	South

What combination of predictors best describes the occurrences at value=0 or value=1??

Methodologies - CART

Occurrence	Sediment	Slope	Wind
0	Sand	Strong	North
0	Sand	Strong	East
0	Sand	Medium	West
0	Mud	Medium	East
0	Mud	Weak	West
0	Sand	Medium	West
0	Sand	Medium	sud
1	Mud	Weak	North
1	Sand	Medium	sud
1	Sand	Strong	sud
1	Sand	Strong	East
1	Mud	Weak	East
1	Sand	Weak	North
1	Mud	Strong	West

Mud

3 P / 2 A

Sand

4 P / 5 A

occurrence	sediment
0	Mud
0	Mud
1	Mud
1	Mud
1	Mud

Gini index is one method to determine splits

$$G = \sum (P_k \times (1 - P_{kw}))$$

N = 15

8 Absences

7 Presences

Pk = proportion of training rows with class k in data subset

Pkw = proportion of training rows with class k weighted by values in the parent node

Class 0: (2/5) x (1 – (2/5)) = 0.24

Class 1: (3/5) x (1 – (3/5)) = 0.24

Gini = 0.24 + 0.24 = 0.48

Methodologies - CART

Occurrence	Sediment	Slope	Wind
0	Sand	Strong	North
0	Sand	Strong	East
0	Sand	Medium	West
0	Mud	Medium	East
0	Mud	Weak	West
0	Sand	Medium	West
0	Sand	Medium	sud
1	Mud	Weak	North
1	Sand	Medium	sud
1	Sand	Strong	sud
1	Sand	Strong	East
1	Mud	Weak	
1	Sand		
1	Mud	S	

Mud

3 P / 2 A

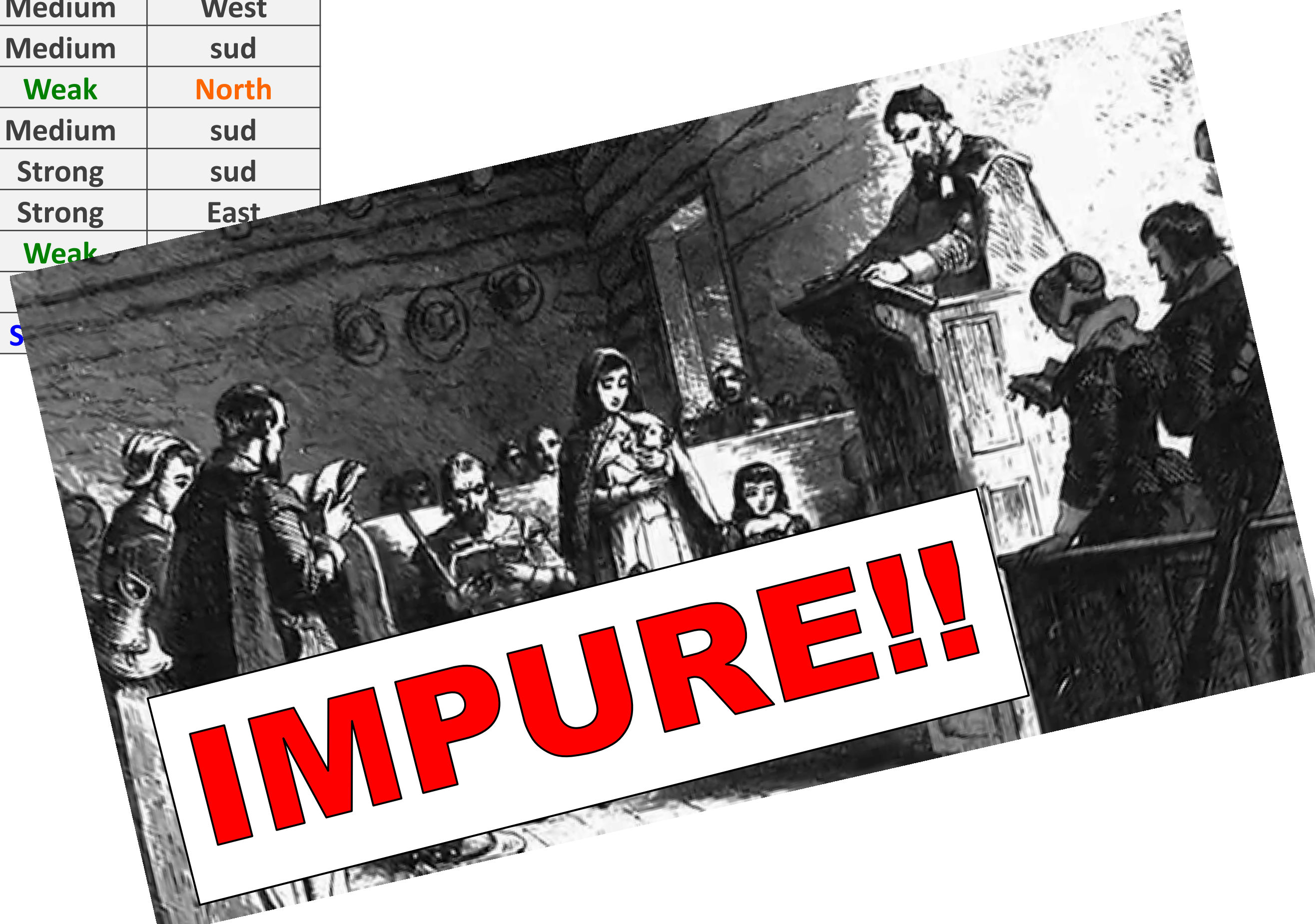
Sand

4 P / 5 A

N = 15

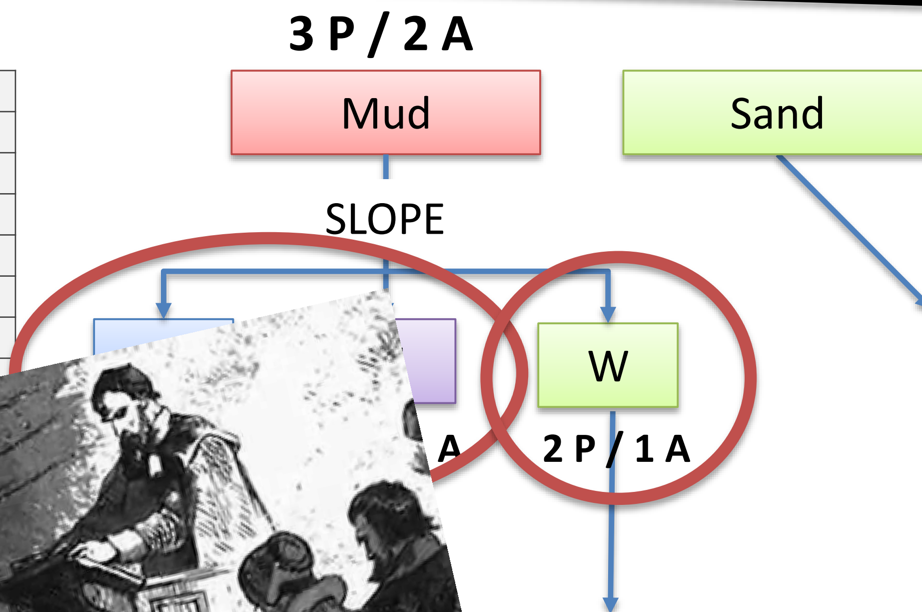
8 Absences

7 Presences



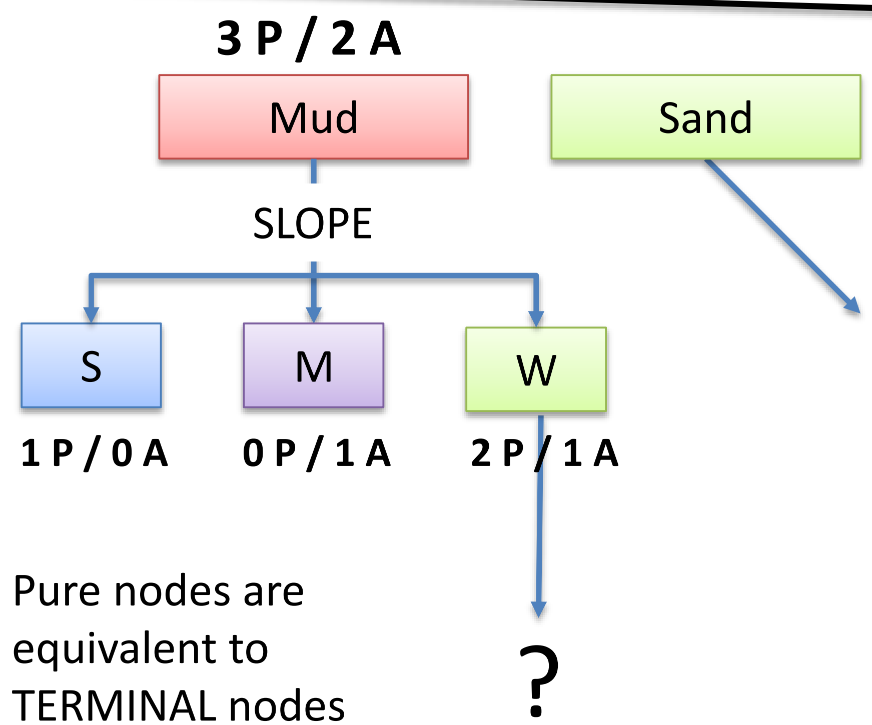
Methodologies - CART

Occurrence	Sediment	Slope	Wind
0	Sand	Strong	North
0	Sand	Strong	East
0	Sand	Medium	West
0	Mud	Medium	East
0	Mud	Weak	West
0	Sand	Medium	West
0	Sand	Medium	sud
1	Mud	Weak	



Methodologies - CART

Occurrence	Sediment	Slope	Wind
0	Sand	Strong	North
0	Sand	Strong	East
0	Sand	Medium	West
0	Mud	Medium	East
0	Mud	Weak	West
0	Sand	Medium	West
0	Sand	Medium	sud
1	Mud	Weak	North
1	Sand	Medium	sud
1	Sand	Strong	sud
1	Sand	Strong	East
1	Mud	Weak	East
1	Sand	Weak	North
1	Mud	Strong	West



Pure nodes are equivalent to TERMINAL nodes

N = 15

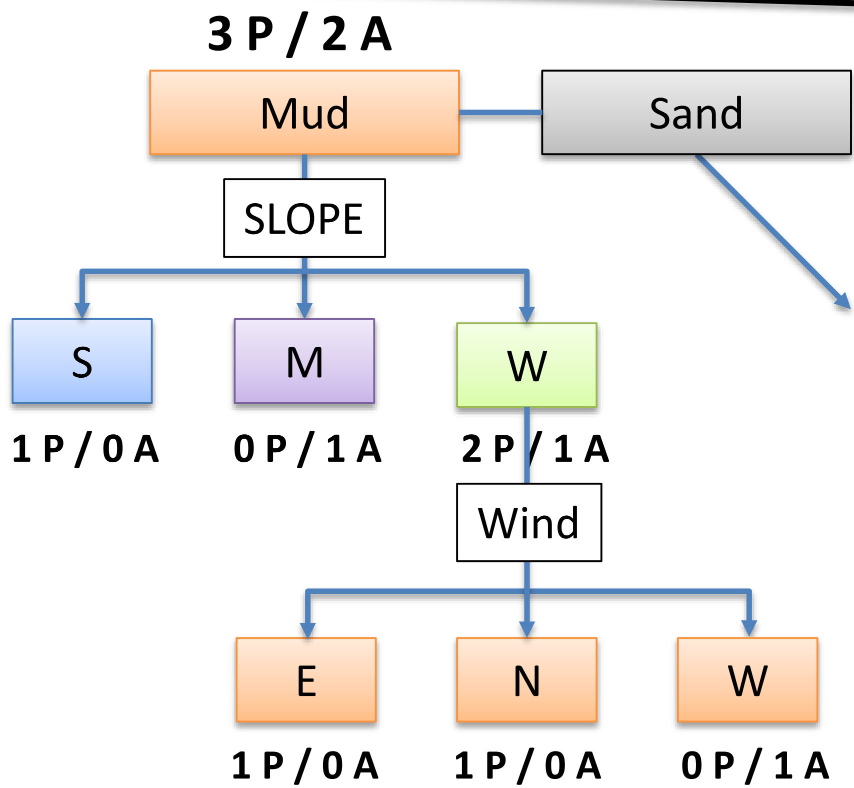
8 Absences

7 Presences

Splits will continue to be calculated until :

a) All terminal nodes are 'pure'
b) The 'purity' in the terminal nodes reaches a pre-determined value
c) Standard deviation of observations in the node reaches a certain % of the initial standard deviation

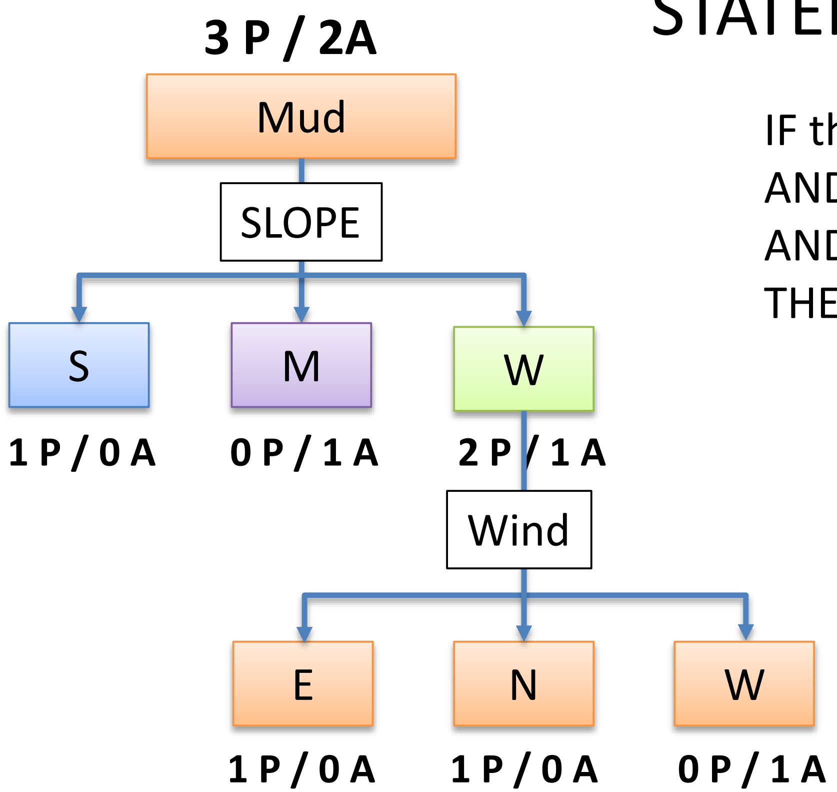
Occurrence	Sediment	Slope	Wind
0	Sand	Strong	North
0	Sand	Strong	East
0	Sand	Medium	West
0	Mud	Medium	East
0	Mud	Weak	West
0	Sand	Medium	West
0	Sand	Medium	sud
1	Mud	Weak	North
1	Sand	Medium	sud
1	Sand	Strong	sud
1	Sand	Strong	East
1	Mud	Weak	East
1	Sand	Weak	North
1	Mud	Strong	West



N = 15

7 Absences

7 Presences

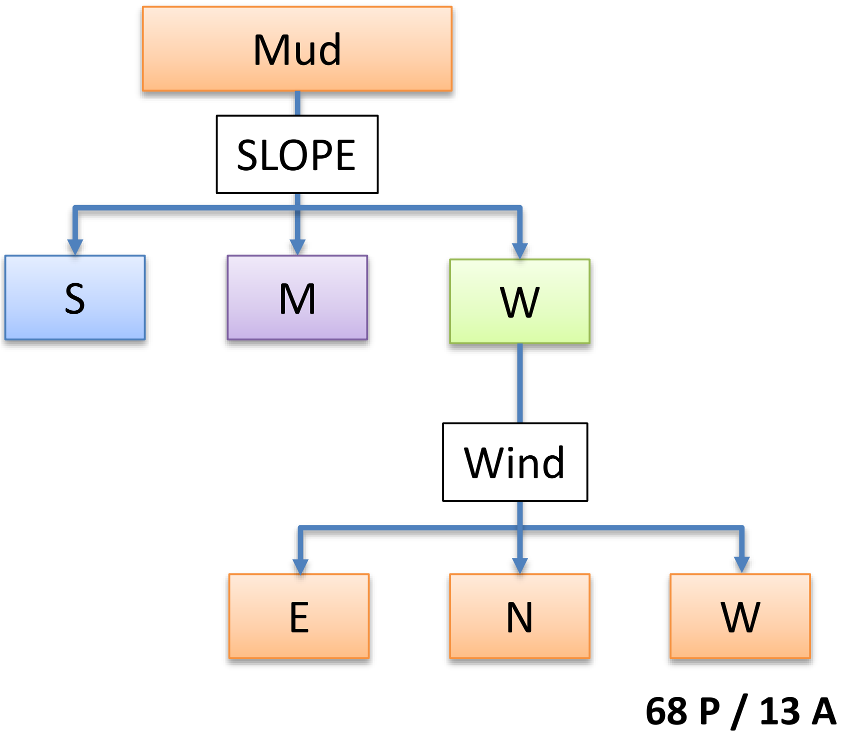


CONVERT TO CONDITIONAL STATEMENTS:

IF the sediment is mud
AND the slope is W (weak)
AND the wind is from the N (North)
THEN the value of the occurrence is a PRESENCE

Given new data: MUD, WEAK, WEST …… ?

BUT, Don't we get probabilities??



Probability class P

68/(68+13) = 0.8395

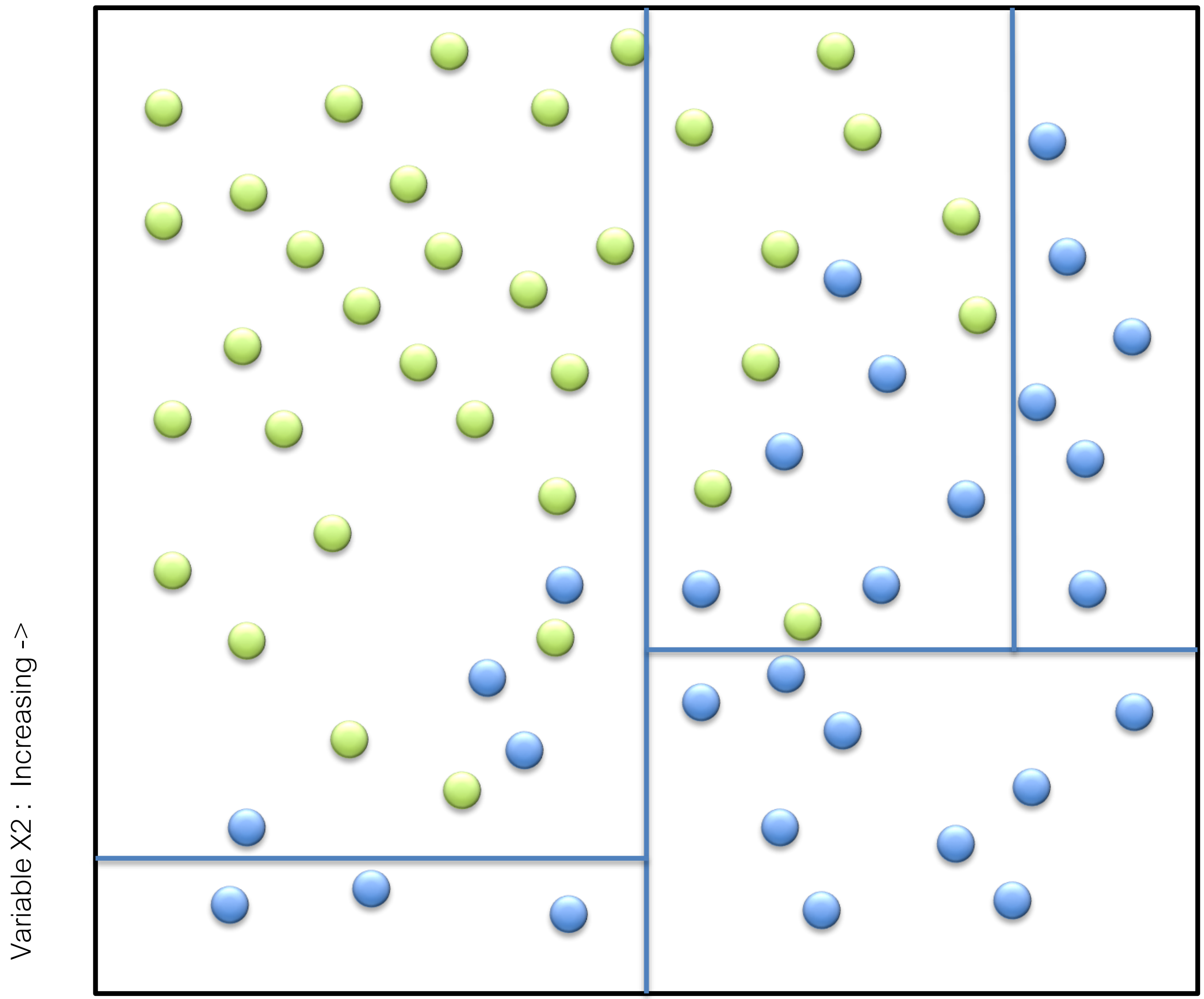
Probability class A

1 – P(class A) = 0.1605

But what if we have 40 P and 41 A?? → THRESHOLDS

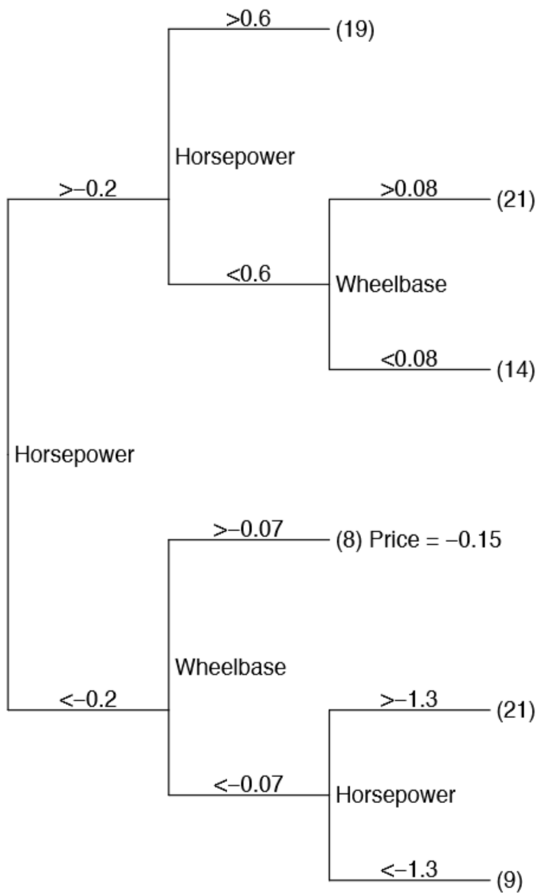
WHAT ABOUT CONTINOUS PREDICTORS!?

Methodologies - CART



Variable X1: Increasing ->

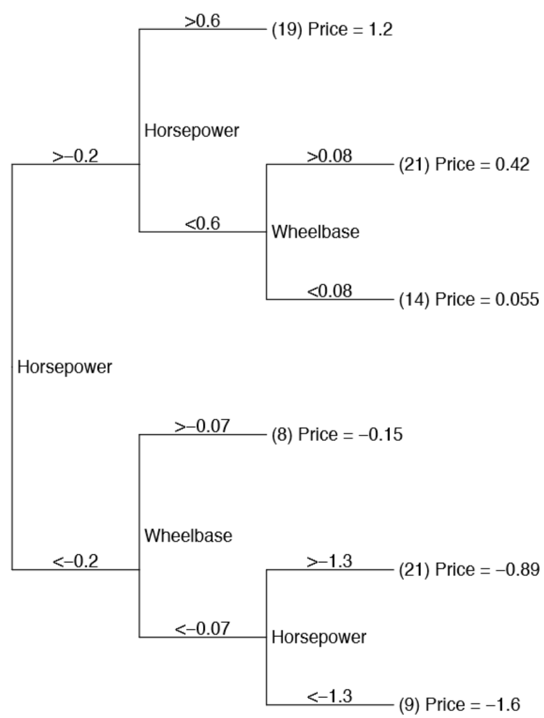
Methodologies - CART



Target / Dependent variable = Continuous?

Splits are determined by reduction of variance

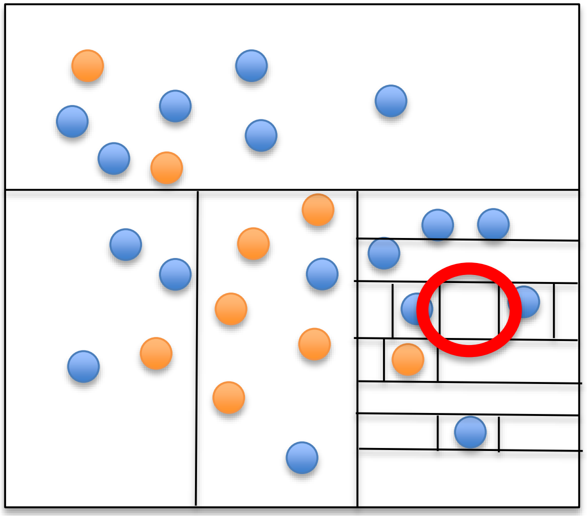
Predictions are made based on the mean values of the observations in the terminal nodes



A note on over-'fitting'? / over-'splitting'

- CART does not explicitly 'fit' anything (e.g., a line).
- Over-'learning' / 'splitting' can occur if you tell it to keep splitting!!! BE CAREFUL

- A method to limit this is by "PRUNING" the tree:
    - Can be done by CROSS-VALIDATION
    - Or a PRUNING ALGORITHM



Methodologies - CART

- Advantages
  - The model is visualised intuitively (conditionals)
  - Can include many variable types
  - Can include missing predictor data**
  - Generally insensitive to extreme values

- Disadvantages
  - True linear relationships are difficult to capture
  - Uncertainty is difficult to measure
  - A single tree does not have much predictive power

CAN WE MAKE CART BETTER??

Brought to you by:

ENTER: Boosting and Bagging!

BOOSTING

Increasing the power of your model by iteratively building on the previous version until you get the 'best' model.
Think: **Bruce Banner transforming to Hulk by increasing in size, making him all powerful**

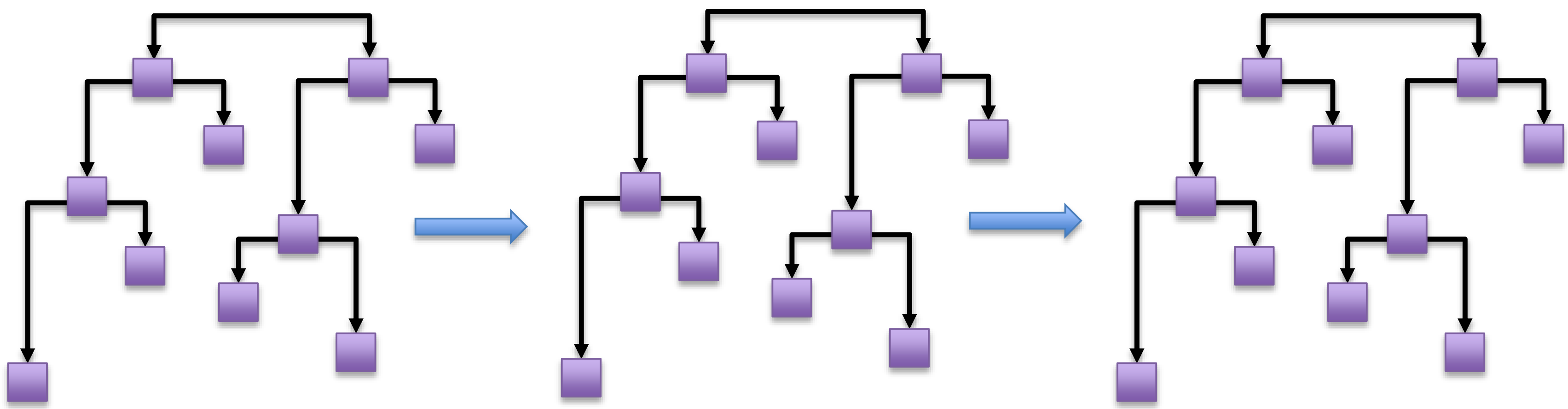
BAGGING

Increasing the power of your model by building many models and averaging to get the 'best' model.
Think: **Doctor Strange multiplying himself to launch a powerful attack against the forces of evil**

THE INCREDIBLE HULK

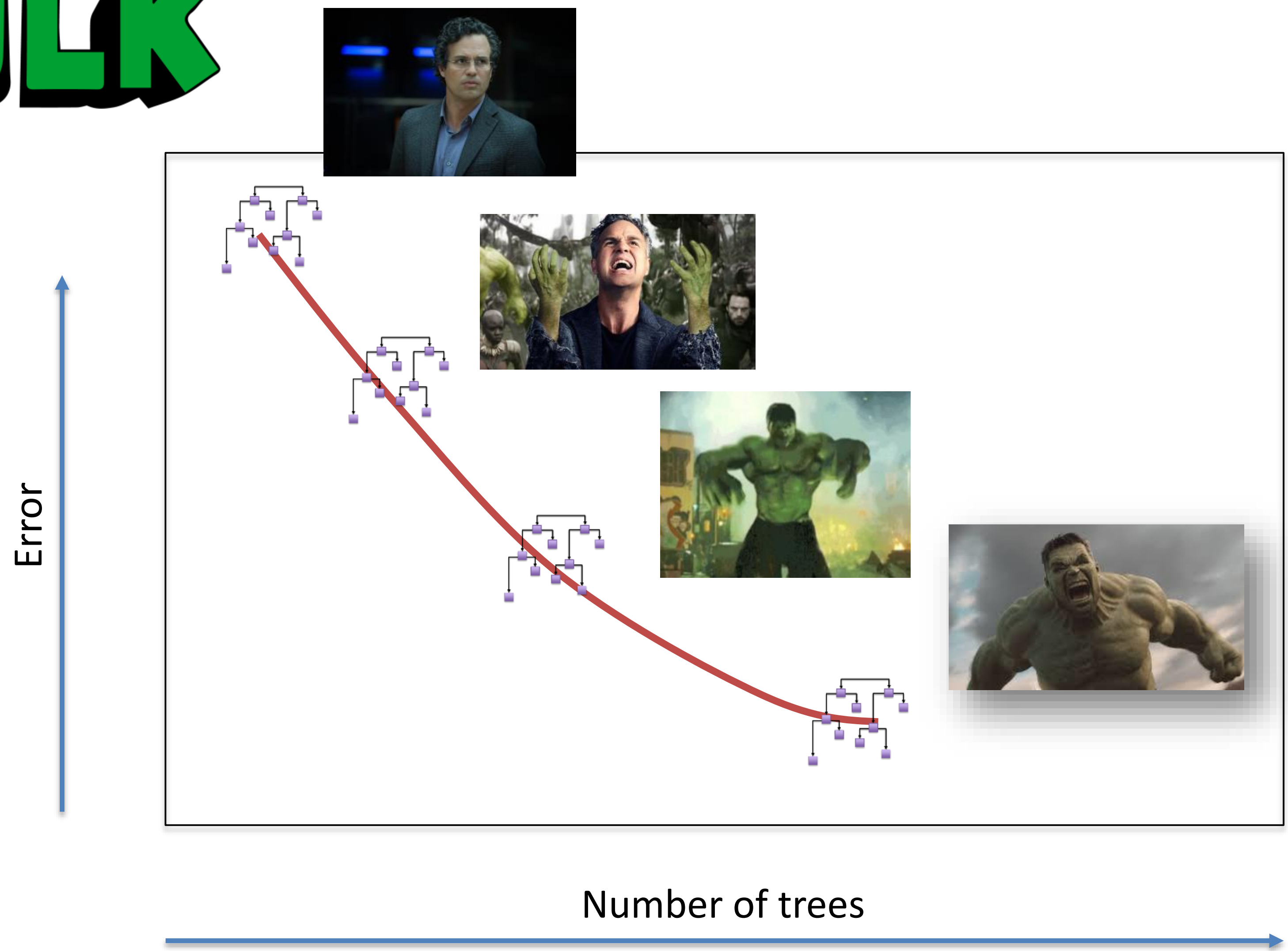
Boosted regression trees

aka. Stochastic gradient boosting, generalized boosted regression modelling, gradient descent modelling



THE LOSS FUNCTION: Most commonly RMSE

THE INCREDIBLE HULK



THE INCREDIBLE HULK

IMPORTANT PARAMETERS:

Number of trees: Total number of trees to build (typically in the thousands)

Learning rate / shrinkage rate: A weighting factor to slow ‘learning’ (lower values will increase the number of corrections to each tree, thus decreasing the number of trees you might have to grow to minimize error). Typical values are 0.3, 0.1, 0.05, 0.01 (a value of 1 is no weighting)

Step size: The number of trees at each ‘step’ that is used for determining the best model

Tree complexity: The number of terminal nodes / the number of splits (depends on implementation)

DOCTOR STRANGE

Random forests

DOCTOR STRANGE

Random Forests bags predictions by AVERAGING the results of the models across the many trees (i.e., forests)

Interpretability can be lost as relationships (partial dependence) are difficult to elucidate.

Each tree is built with a DIFFERENT SUBSET of the data (determined by the ‘**out of bag**’ fraction parameter)

The **number and complexity** of trees can also be controlled

Over-learning can occur in both RF and BRT

- In RF, this is limited by cross-validation within and between forests
- In BRT this is limited by cross-validation between steps (number of trees): Once the error rate starts to increase on predictions, we have found the model that best generalises

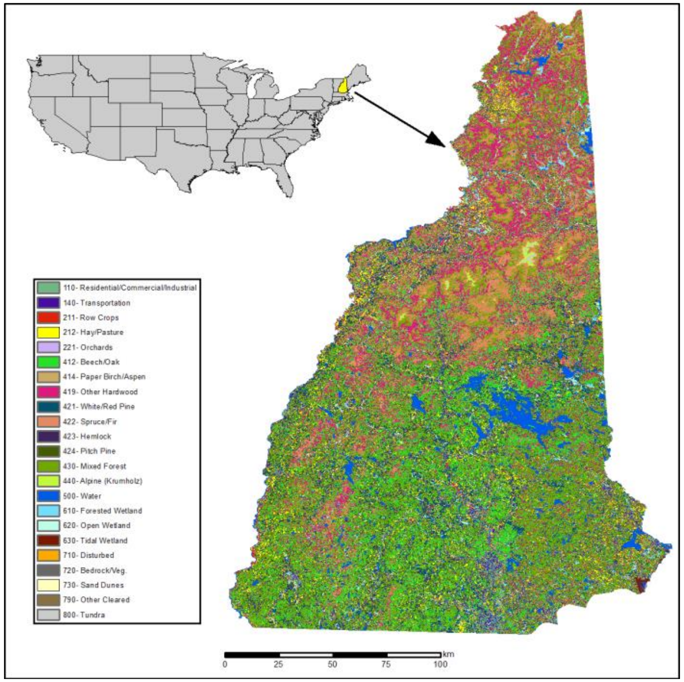
A quick note on One Hot Encoding!

- A common problem in landscape ecology is the existence of too many categories! Can cause one variable to 'over-power' the model - One Hot Encoding can help fix this.

P/A	Class
0	A
0	A
1	B
1	C
0	D
1	E
0	F

P/A	A	B	C	D	E	F
0	1	0	0	0	0	0
0	1	0	0	0	0	0
1	0	1	0	0	0	0
1	0	0	1	0	0	0
0	0	0	0	1	0	0
1	0	0	0	0	1	0
0	0	0	0	0	0	1

DON'T WORRY: BRT and RF are VERY good at handling MANY variables!



Some useful libraries

dismo**
biomod2**
gbm
randomForest
party
SSDM**
xgboost
h2o
rPart
MRSea**
caret
mgcv

Let’s play!

Time for some hands-on practice. We’re going to structure data, build models and briefly look at some output.

TASKS

- Use tidyverse to one-hot-encode
- Create a decision tree in the party package
- Run the same model with gbm.step and view test/train plot
  - Change learning rate and time the model
- Run the same model with randomForest
- Take a look at variable importance plots/rankings