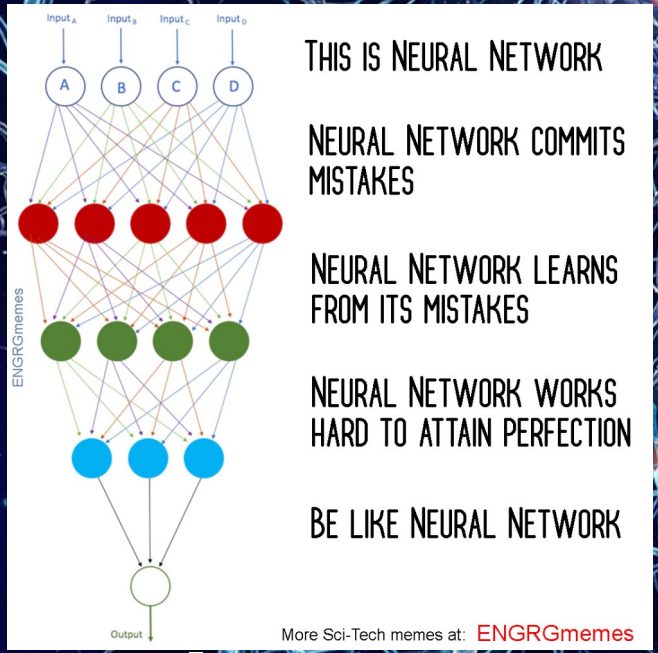
PLS 206 Applied Multivariate Modeling in Agricultural and Environmental Sciences



Random forest



Neural network

Week 8 recap

Kahoot-style warm-up

6 quick questions on clustering.

Not graded.

Use a pseudonym if you want.

Fast and correct wins.

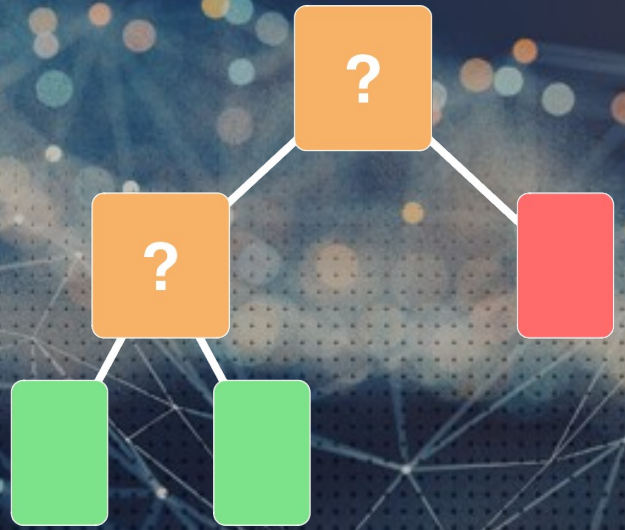


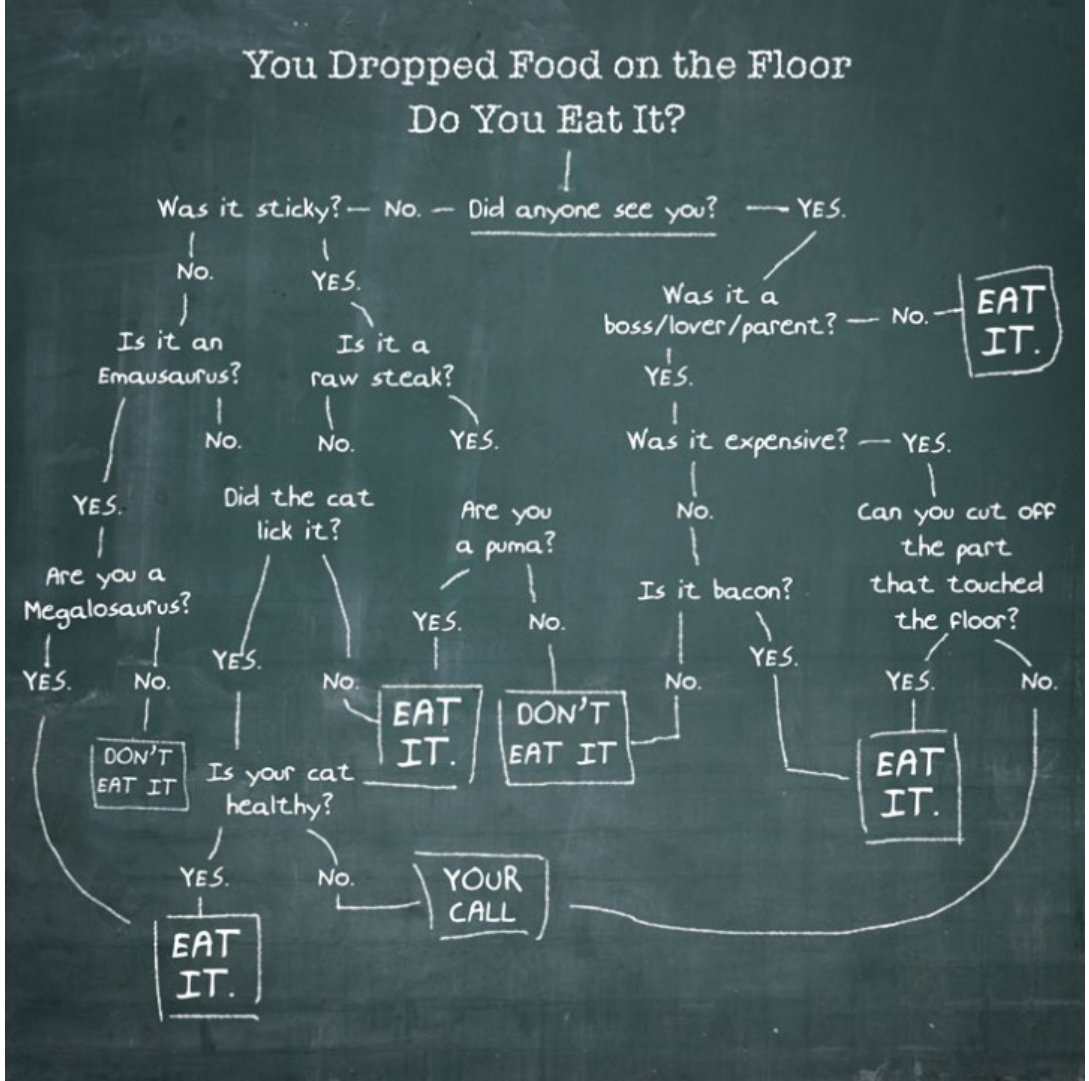
Scan with your phone

[Join on your phone or laptop: greymonroe.github.io/live-poll-test/quiz-play.html?quiz=pls206-w9-cluster-recap](greymonroe.github.io/live-poll-test/quiz-play.html?quiz=pls206-w9-cluster-recap)

[host screen]

1 Trees
2 One tree
3 Forests
4 Complex data
5 Neural nets
6 Which model?

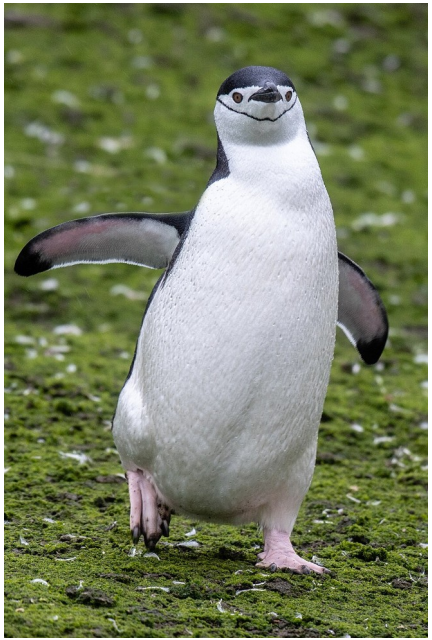



You Dropped Food on the Floor
Do You Eat It?
Was it sticky?
No.
Did anyone see you?
YES.
No.
YES.
Was it a boss/lover/parent?
No.
EAT IT.
Is it an Emausaurus?
Is it a raw steak?
YES.
Was it expensive?
YES.
No.
No.
YES.
YES.
Did the cat lick it?
Are you a puma?
No.
Can you cut off the part that touched the floor?
Are you a Megalosaurus?
Is it bacon?
YES.
No.
YES.
No.
YES.
No.
YES.
No.
YES.
No.
EAT IT.
DON'T EAT IT
DON'T EAT IT
Is your cat healthy?
EAT IT.
YES.
No.
YOUR CALL
EAT IT.

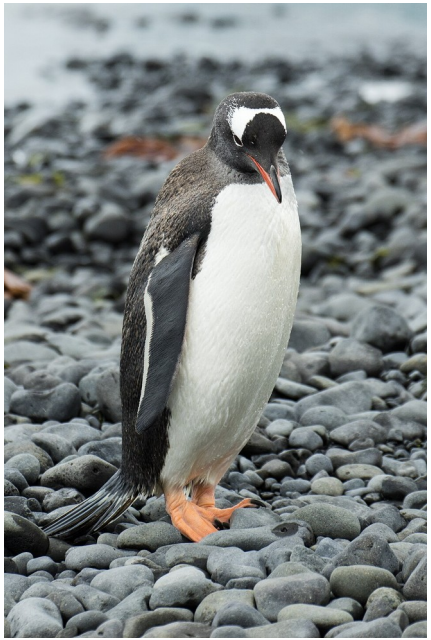
Meet the penguins



Adelie



Chinstrap

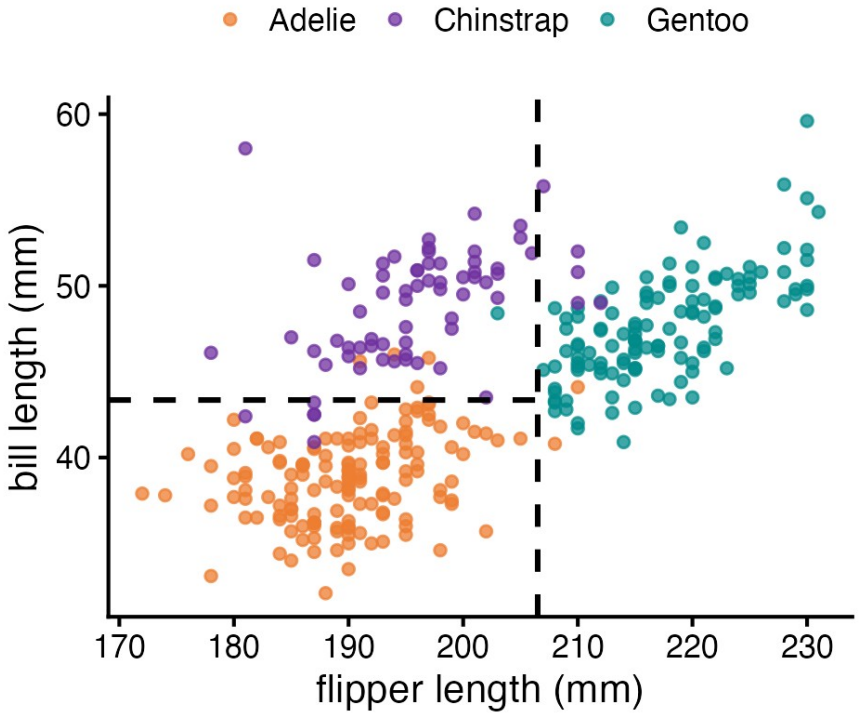


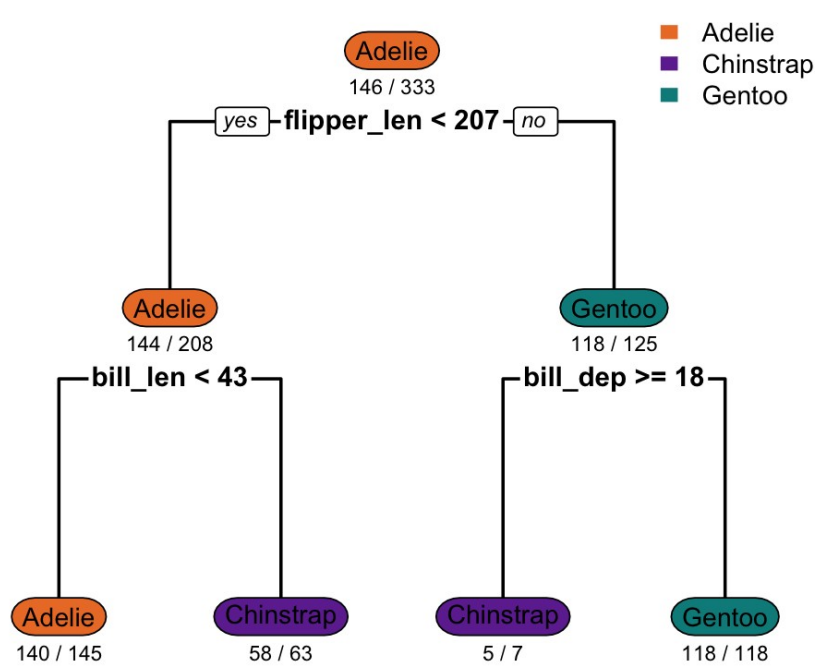
Gentoo

Can a few body measurements **tell them apart?**

Photos, Wikimedia Commons: Samuel Blanc (CC BY-SA 3.0), GRDN711 (CC BY 4.0), Godot13 (CC BY-SA 4.0)

A real tree: which penguin is it?





333 penguins, 3 species. Three questions get **96% right**. Each box: correct / total



1 Trees

2 One tree

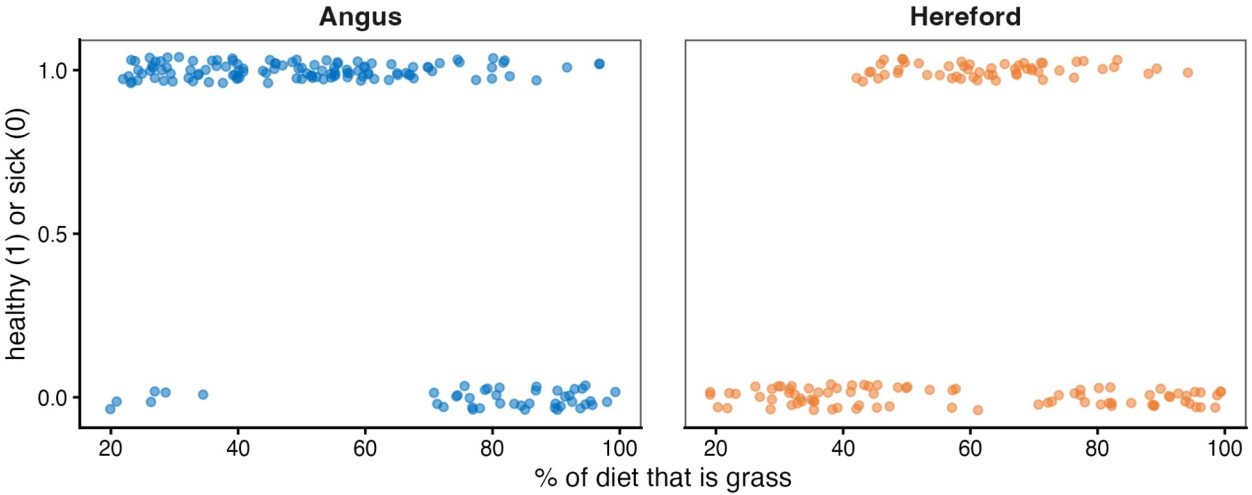
3 Forests

4 Complex data

5 Neural nets

6 Which model?

300 cattle: how much grass is too much?





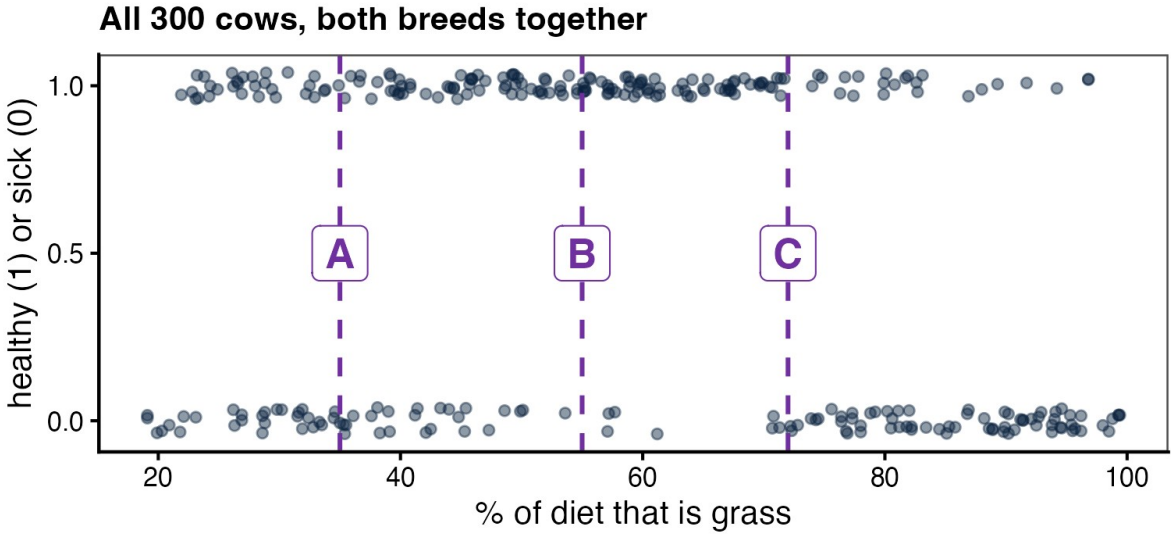


% of diet that is grass + breed → **healthy or sick?** (Hypothetical, simulated data.)

Live poll

Multiple choice — picture

First question for the tree: where should it cut grass?



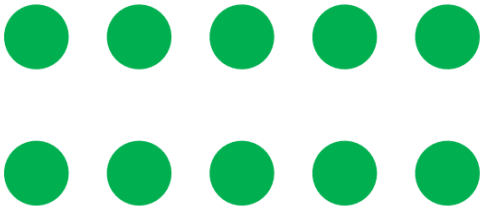
Scan with your phone

[Join on your phone or laptop: greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w9-split](greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w9-split)

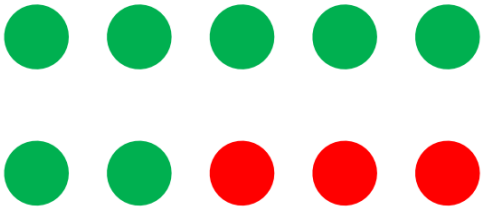
[host screen]

Impurity: how mixed is a group?

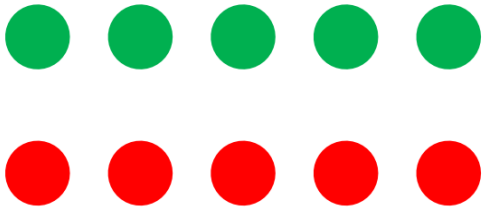
10 healthy, 0 sick
impurity = 0.00



7 healthy, 3 sick
impurity = 0.42



5 healthy, 5 sick
impurity = 0.50



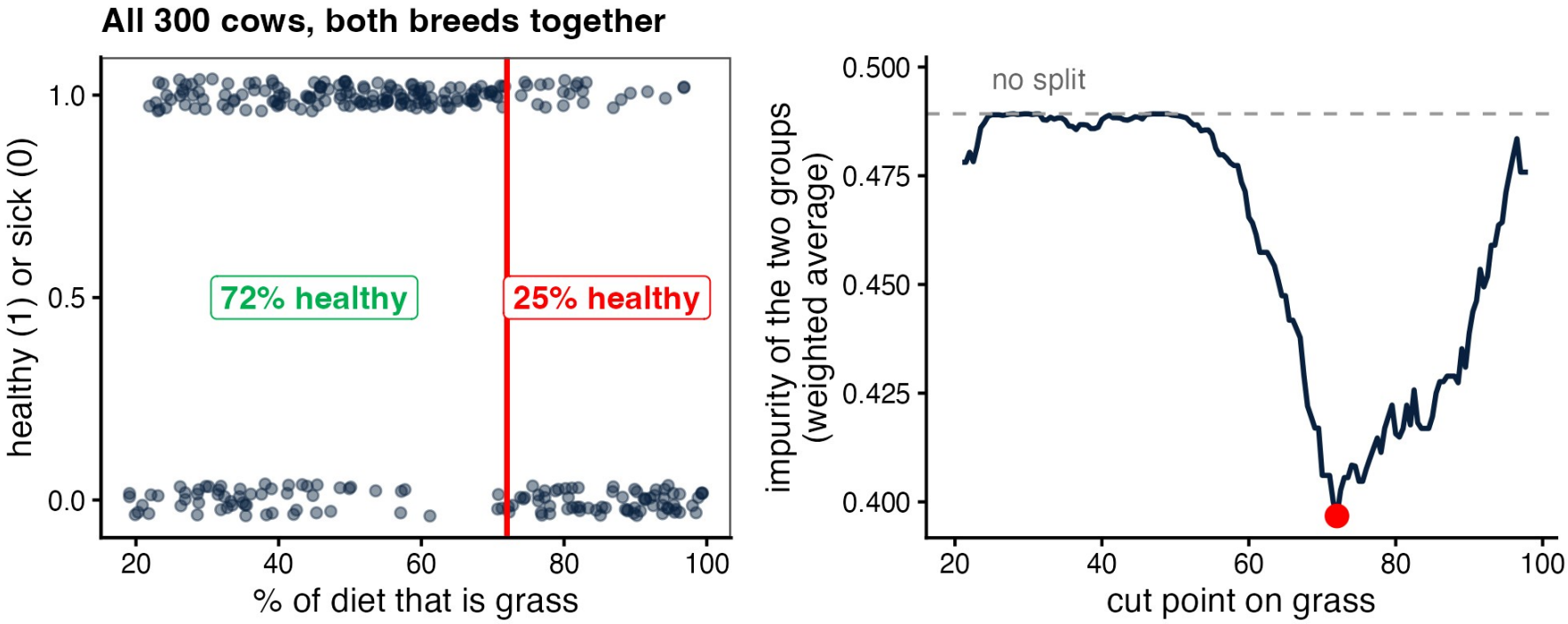


$$\text{impurity} = 2\,p\,(1-p)$$

p: share of the group that is healthy

0 = all one kind (pure). **0.5 = half and half** (as mixed as it gets).

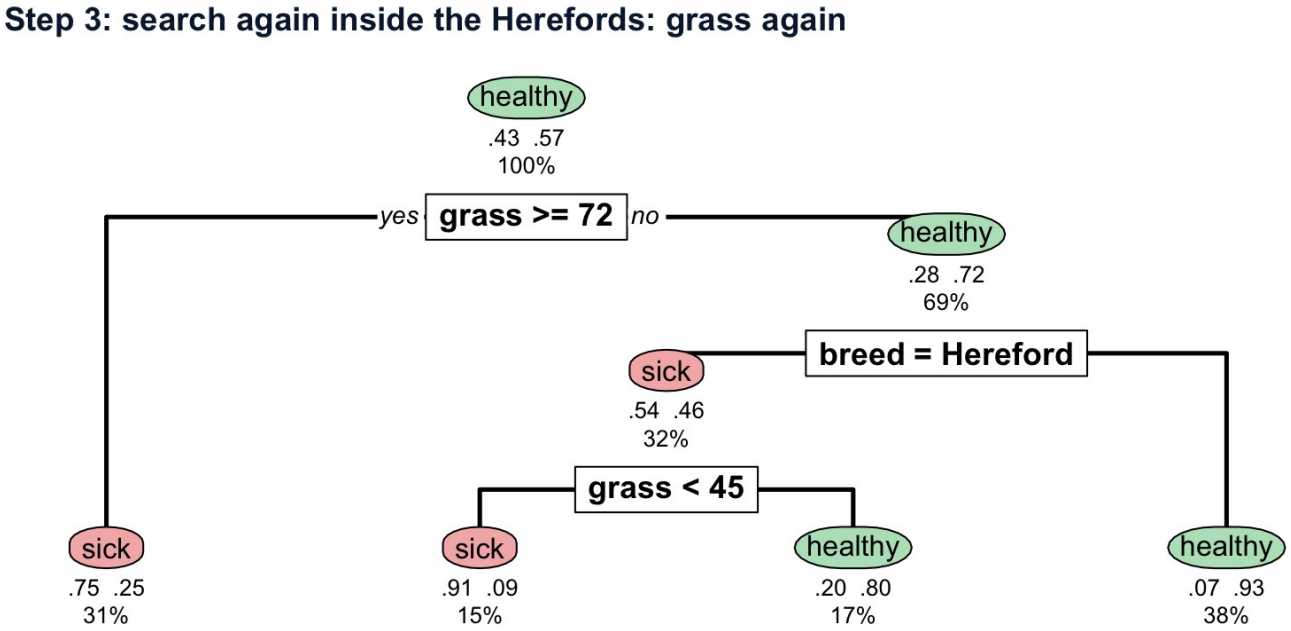
A split: one question about one predictor



Try every cut. Keep the one that leaves the **purest groups**.

Keep splitting: the tree

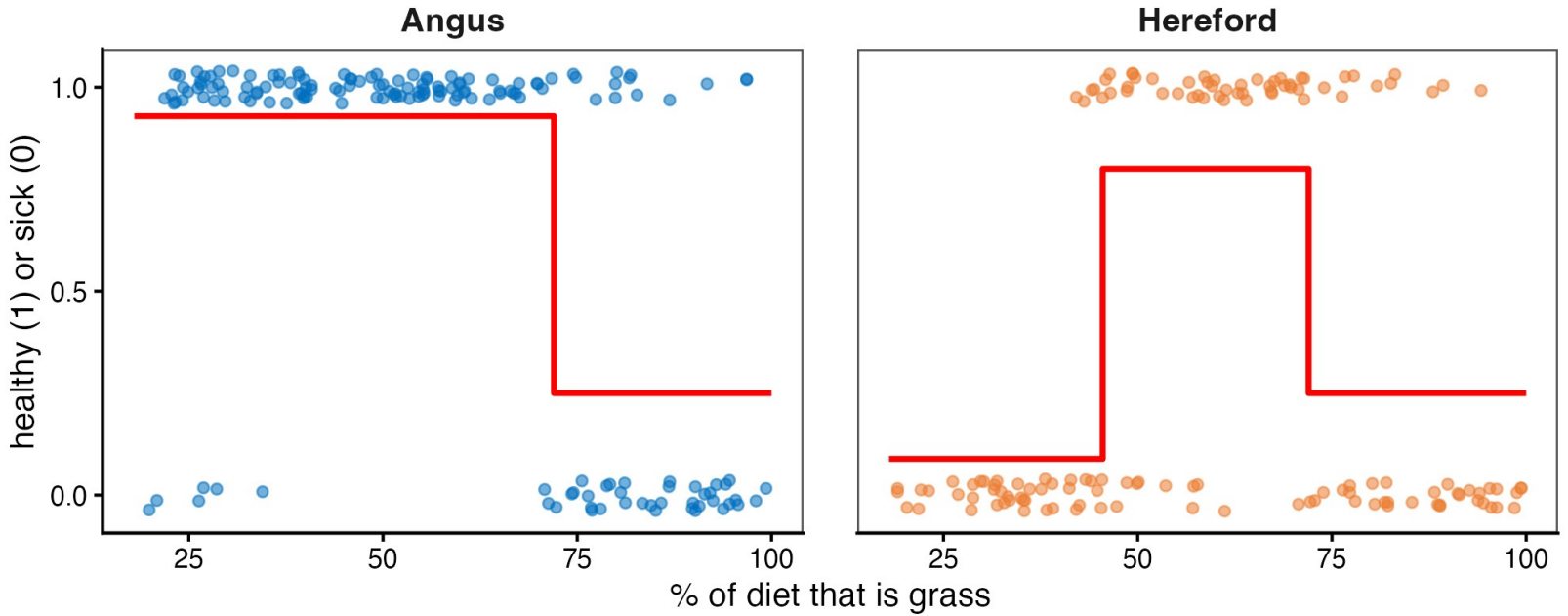
```
> tree <- rpart(status ~ grass + breed, data = cattle);  rpart.plot(tree)
```



Same search, repeated **inside each new group**. Box: share sick, share healthy; % of herd

A tree is a step function

```
> predict(tree, newdata = grid, type = "prob")
```



Rise and fall, breed differences: **nobody told it the shape**

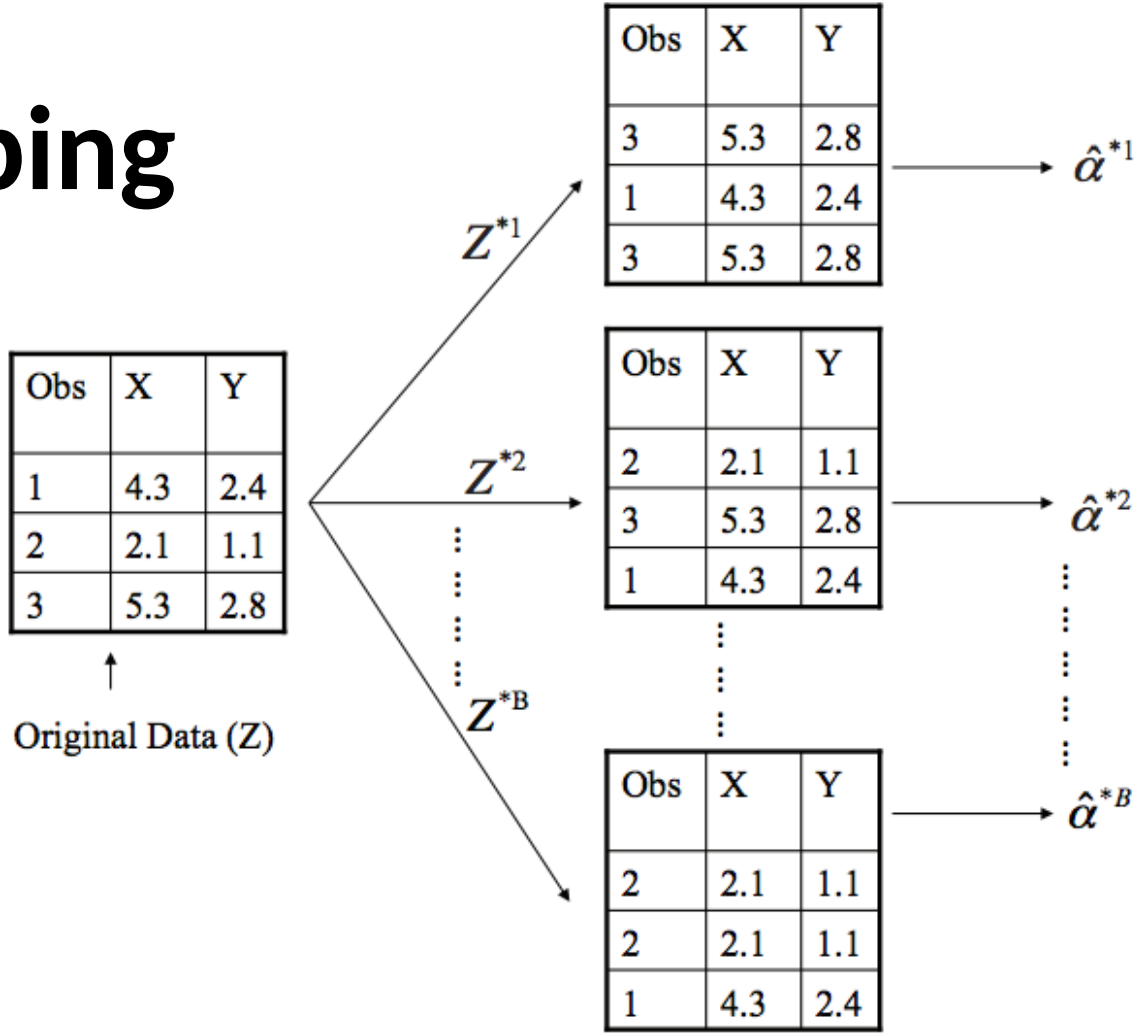
Bootstrapping: a new sample from the same data

```
> cattle[sample(nrow(cattle), replace = TRUE), ]
```

original herd	bootstrap sample 1	bootstrap sample 2	bootstrap sample 3
cow 1 40% grass healthy	cow 2 48% grass healthy	cow 1 40% grass healthy	cow 2 48% grass healthy
cow 2 48% grass healthy	cow 2 48% grass healthy	cow 2 48% grass healthy	cow 2 48% grass healthy
cow 3 65% grass healthy	cow 3 65% grass healthy	cow 2 48% grass healthy	cow 4 92% grass sick
cow 4 92% grass sick	cow 5 34% grass sick	cow 2 48% grass healthy	cow 5 34% grass sick
cow 5 34% grass sick	cow 5 34% grass sick	cow 3 65% grass healthy	cow 6 92% grass healthy
cow 6 92% grass healthy	cow 6 92% grass healthy	cow 4 92% grass sick	cow 7 96% grass sick
cow 7 96% grass sick	cow 7 96% grass sick	cow 5 34% grass sick	cow 8 72% grass sick
cow 8 72% grass sick	cow 8 72% grass sick	cow 8 72% grass sick	cow 8 72% grass sick
	missing: 1, 4	missing: 6, 7	missing: 1, 3

Draw n rows **with replacement**: some cows twice, some not at all

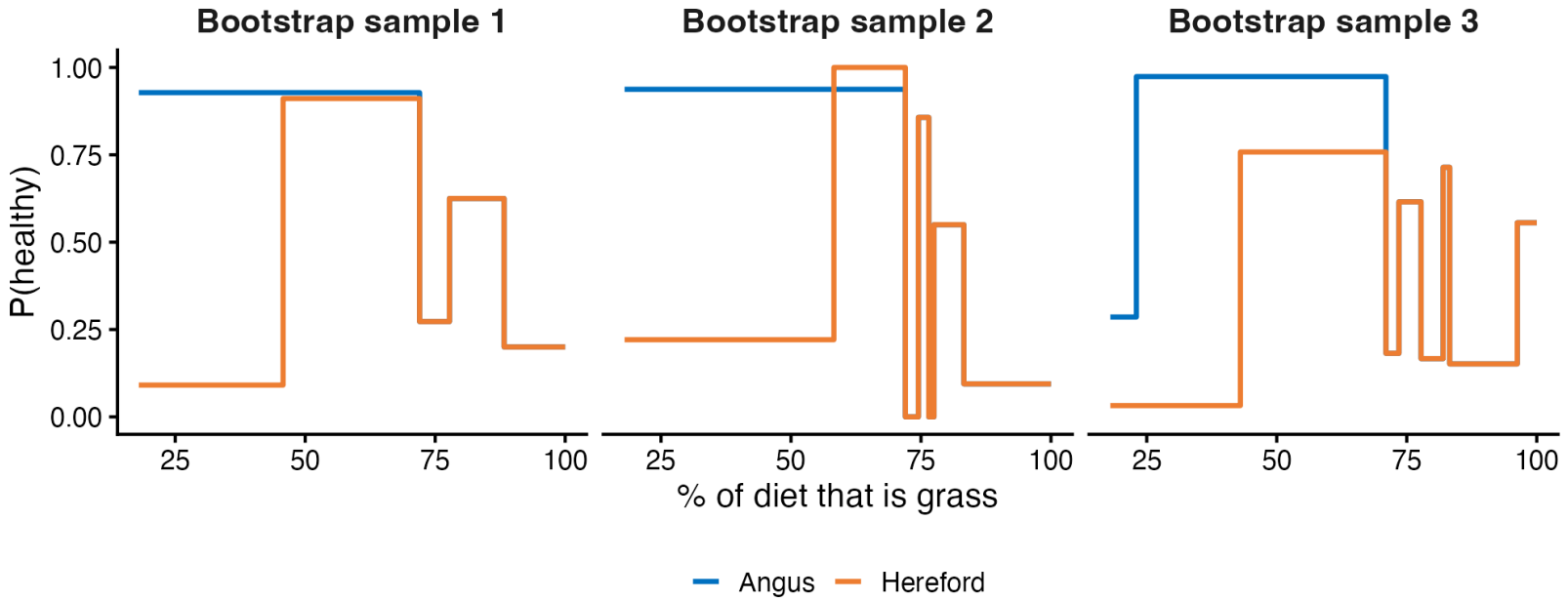
Bootstrapping



https://timvink.nl/post-balanced-trees/

The catch: one tree is unstable

```
> rpart(status ~ grass + breed, data = cattle[sample(nrow(cattle), replace = TRUE), ])
```



Three bootstrap samples of the same cows: **three different trees**. High variance.

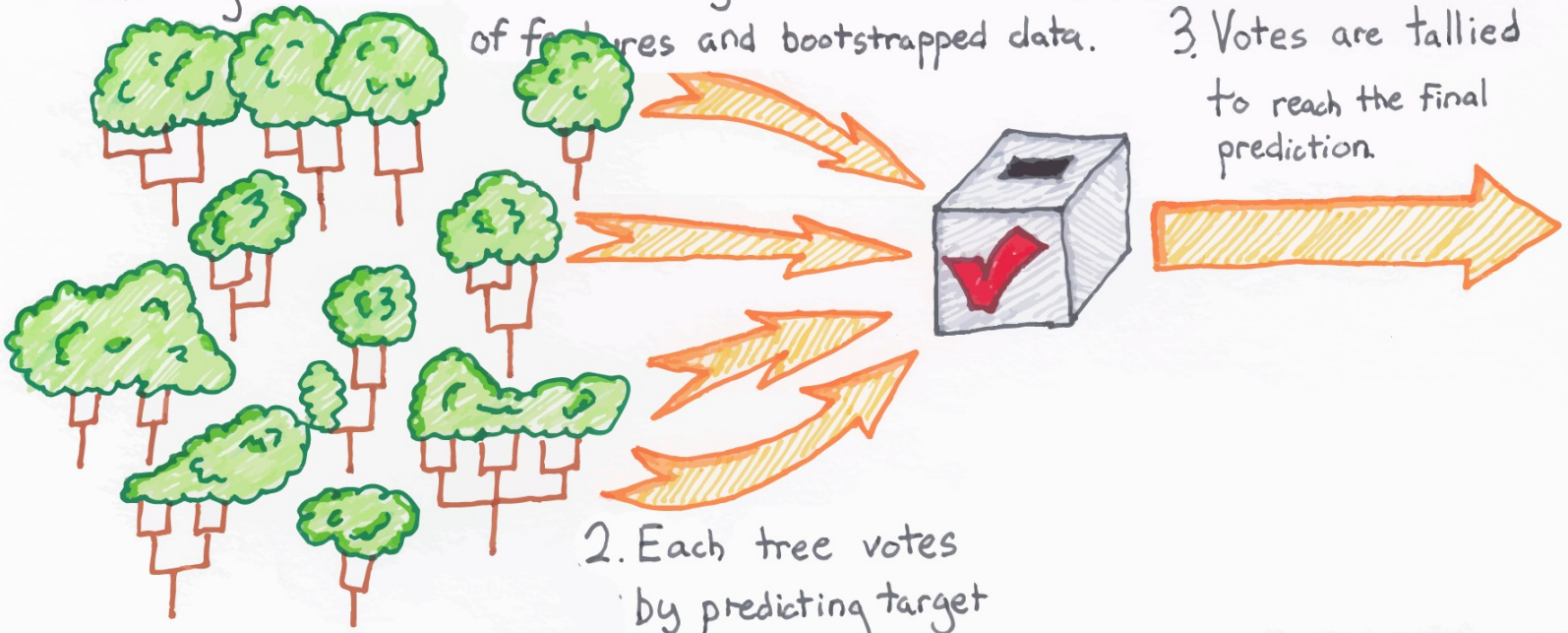
1 Trees
2 One tree
3 **Forests**
4 Complex data
5 Neural nets
6 Which model?



RANDOM FOREST

CLASSIFICATION

1) 1. Many trees are created using random subsets of features and bootstrapped data.



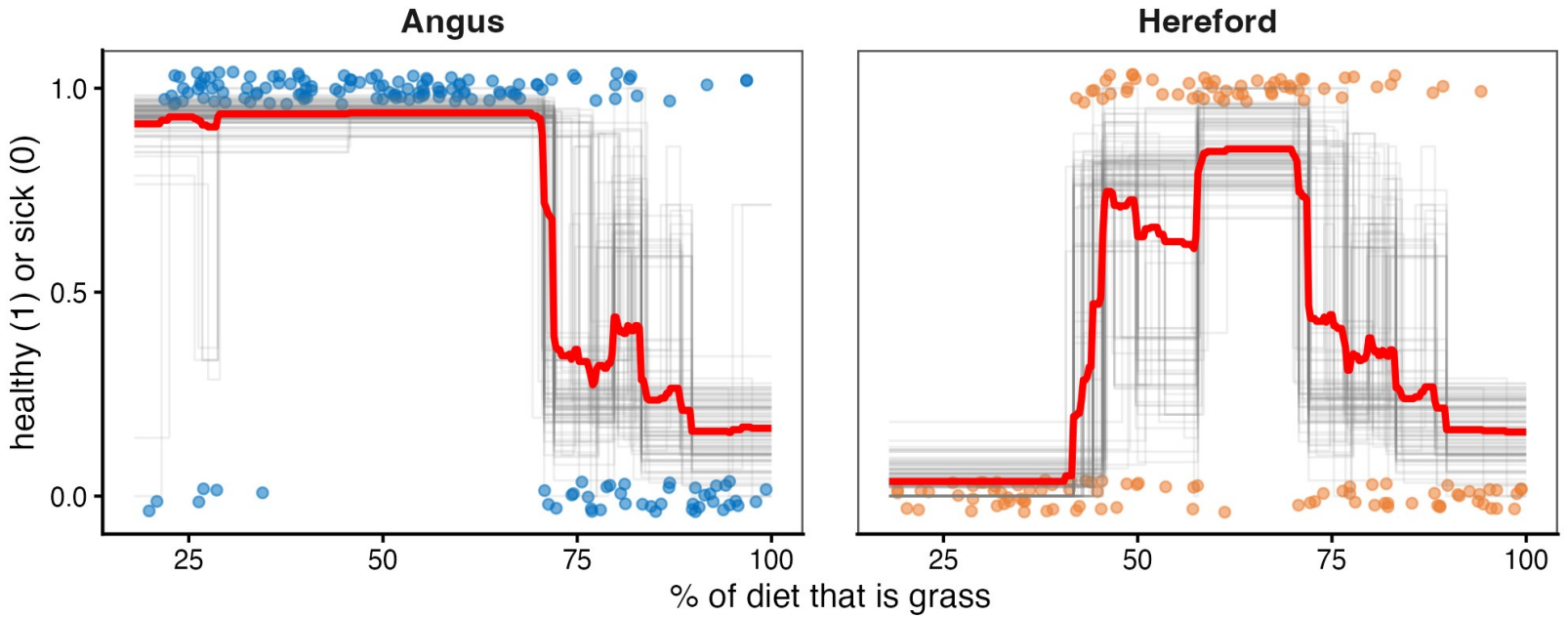
2. Each tree votes by predicting target class.

3. Votes are tallied to reach the final prediction.

CHRIS ALBON

Many trees, averaged

```
> randomForest(status ~ grass + breed, data = cattle, mtry = 2)    # mtry = all predictors: bagging
```



Each tree is jumpy. **Their average is smooth and stable.**

Live poll

Multiple choice

100 bagged trees. One predictor is much stronger than the rest. What do most trees split on first?

A The strong predictor
B A different predictor in each tree
C Half and half

Scan with your phone

Join on your phone or laptop: greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w9-mtry

host screen

The 'random' in random forest

rain	irrigated	heat	variety	frost
pH	soil N	elevation	slope	size

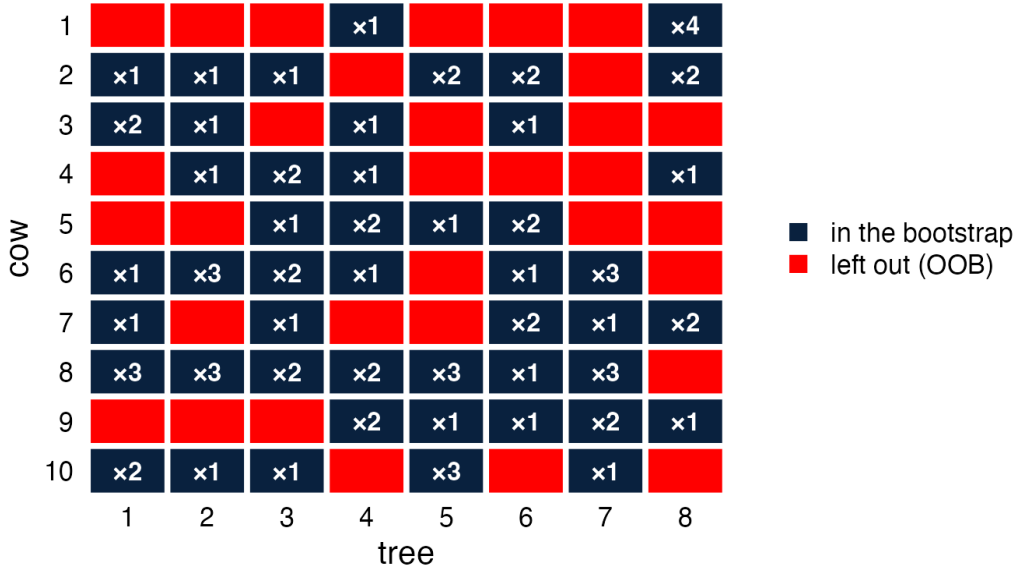
At each split the tree may only choose from `mtry` randomly drawn predictors: a **new draw at every split**

Default: **p/3** for a number, **√p** for a class

Over a whole tree every predictor gets its turn. Strong ones can't hog the top, so **trees differ more**

Free test data: out-of-bag

```
> rf   # printing the forest reports the OOB error
```



About **1/3** of rows are left out of each tree

Score each cow only with trees that **never saw it**

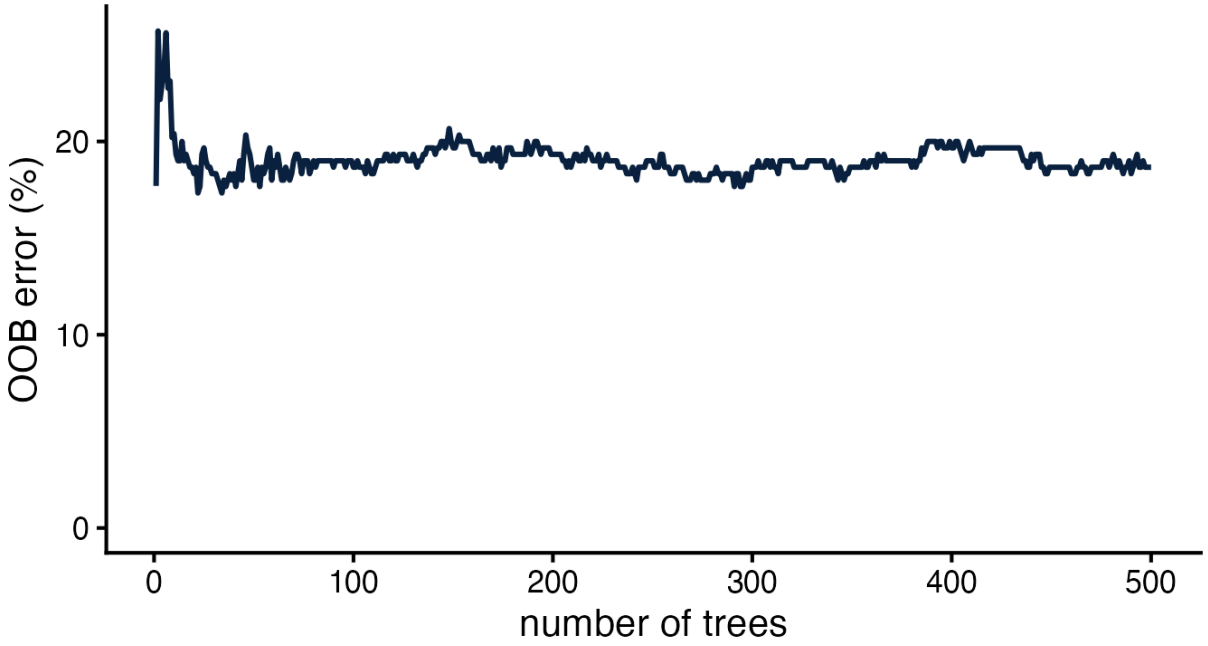
OOB error ≈ error on new data, **no test set needed**

randomForest() in R

```
> rf <- randomForest(status ~ grass + breed, data = cattle, ntree = 500)
> rf

               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 1  ← mtry

        OOB estimate of  error rate: 18.67%  ← error on cows each tree never saw
Confusion matrix:
        sick healthy class.error
sick     105      23   0.1796875
healthy   33     139   0.1918605
```

randomForest(status ~ grass + breed, data = cattle)

How many trees?



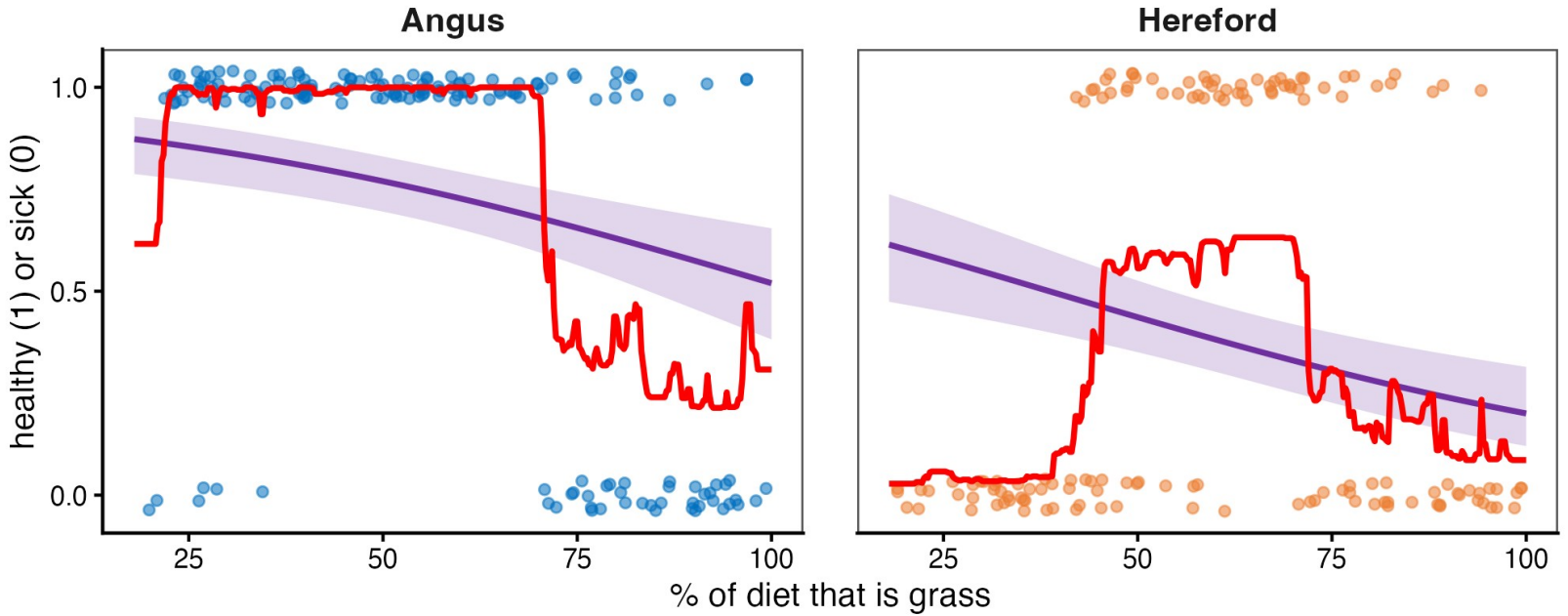
More trees never overfit

They just stop helping

plot(rf): **pick enough trees that the line is flat**

Forest vs. logistic regression

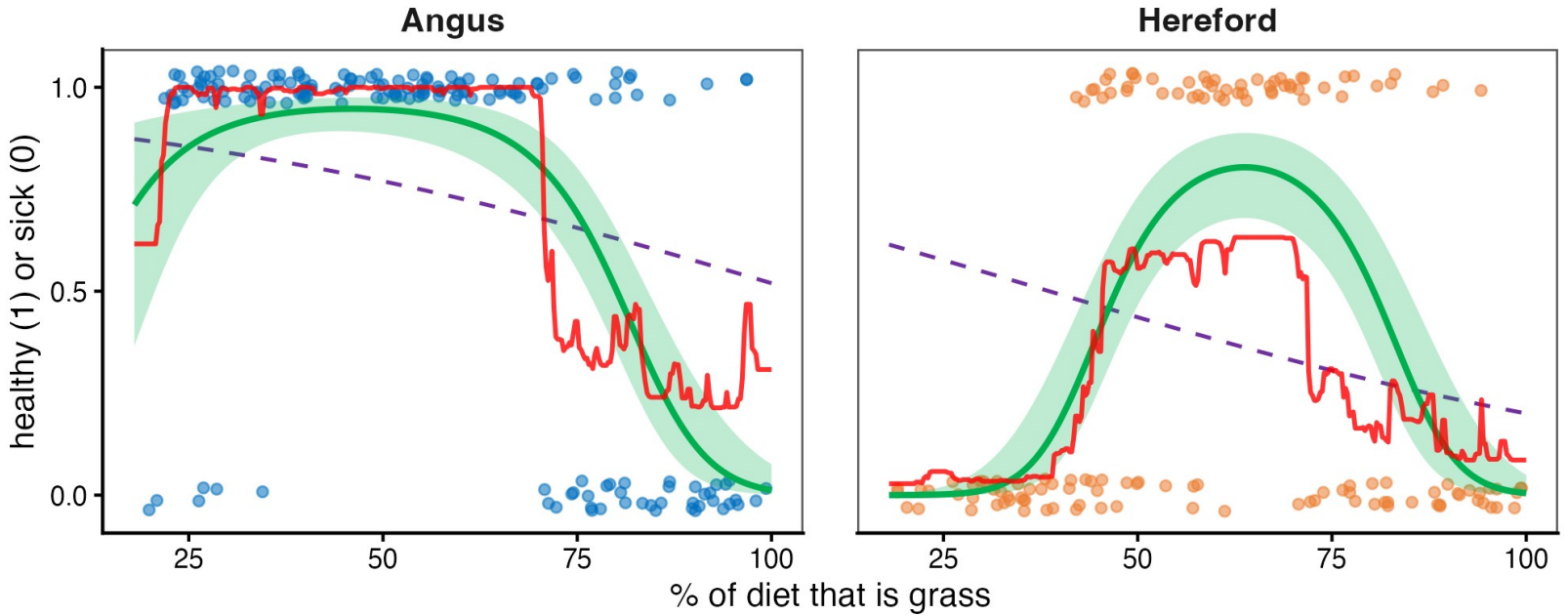
```
> predict(rf, grid, type = "prob")[, "healthy"]
```



GLM: one S-curve, misses the drop. **Forest**: follows the data.

A curved GLM works too

```
> glm(healthy ~ breed * poly(grass, 2), family = binomial, data = cattle)
```



Curved GLM: 76% of a new herd right; forest 78%. You write the shape; the forest finds it.

1 Trees
2 One tree
3 Forests
4 **Complex data**
5 Neural nets
6 Which model?



800 wheat fields, 10 predictors

Weather

- rain (mm)
- irrigated?
- days > 32 °C
- frost days

Field

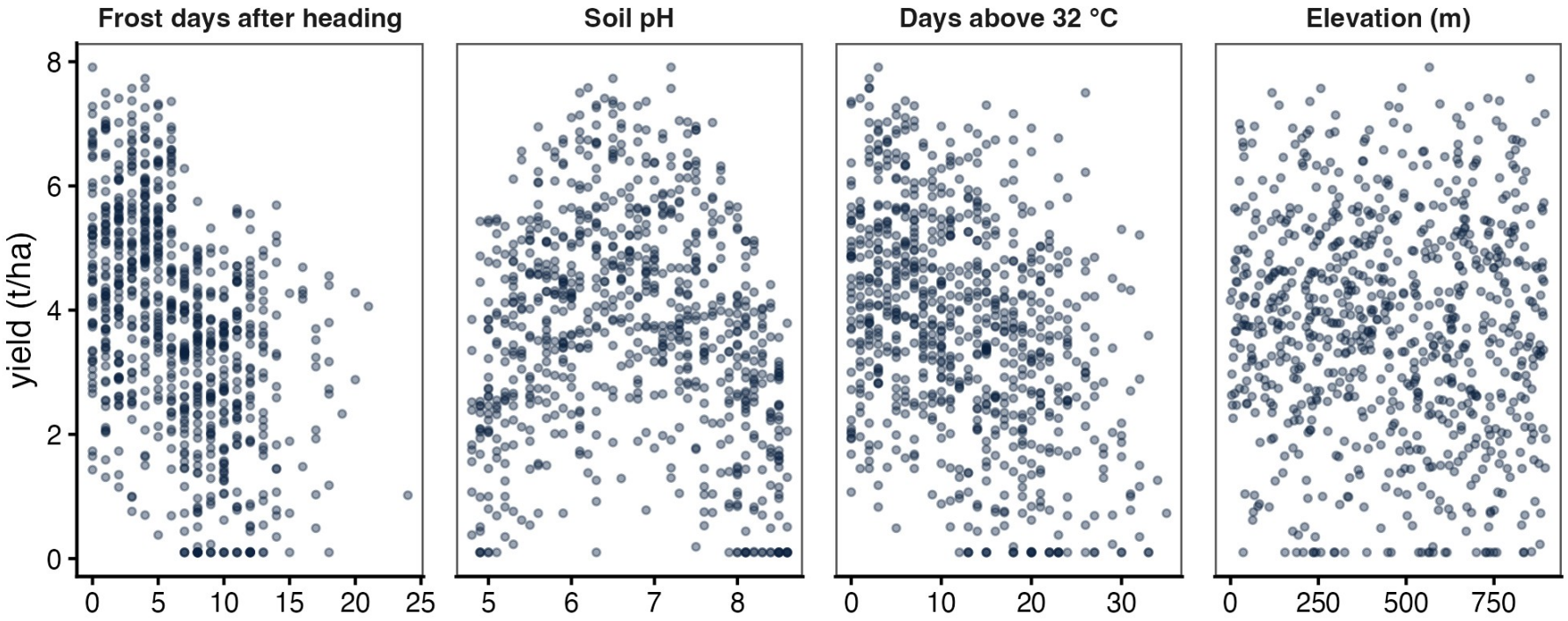
- variety (A–D)
- soil pH
- soil N

Field layout

- elevation
- slope
- field size

→ **yield (t/ha)**. Simulated, so we know the hidden rules. (Hypothetical, simulated data.)

What a scatter plot shows you

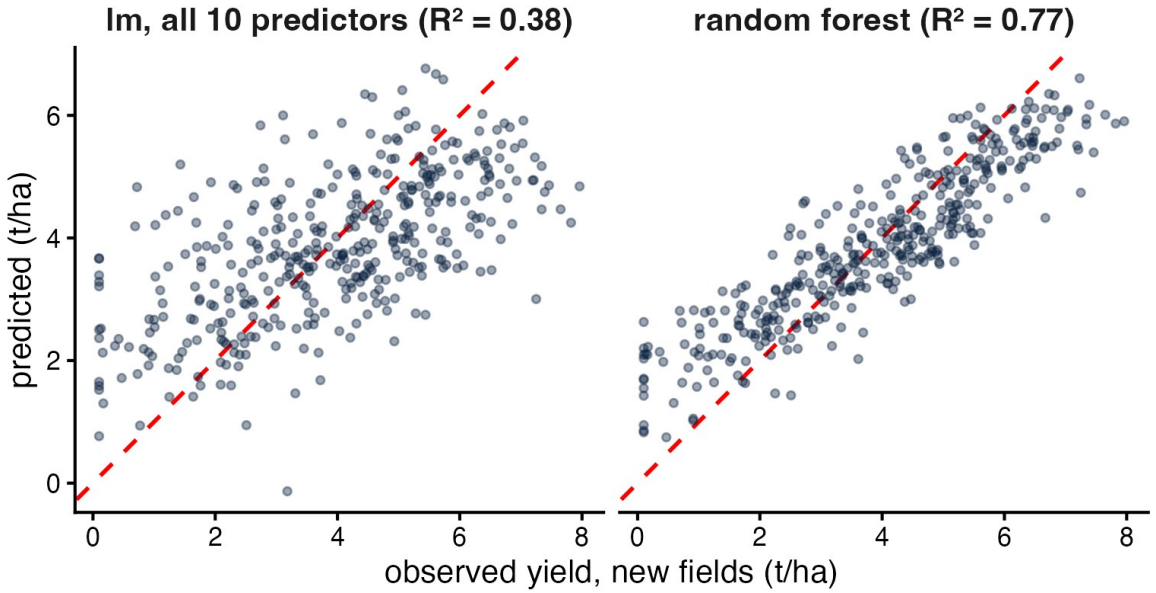


Hints of a drop and a bump, but **ten predictors** are too many to eyeball

lm with all ten vs. a random forest

```
m_lm <- lm(yield ~ ., data = fields)
predict(m_lm, new_fields)
```

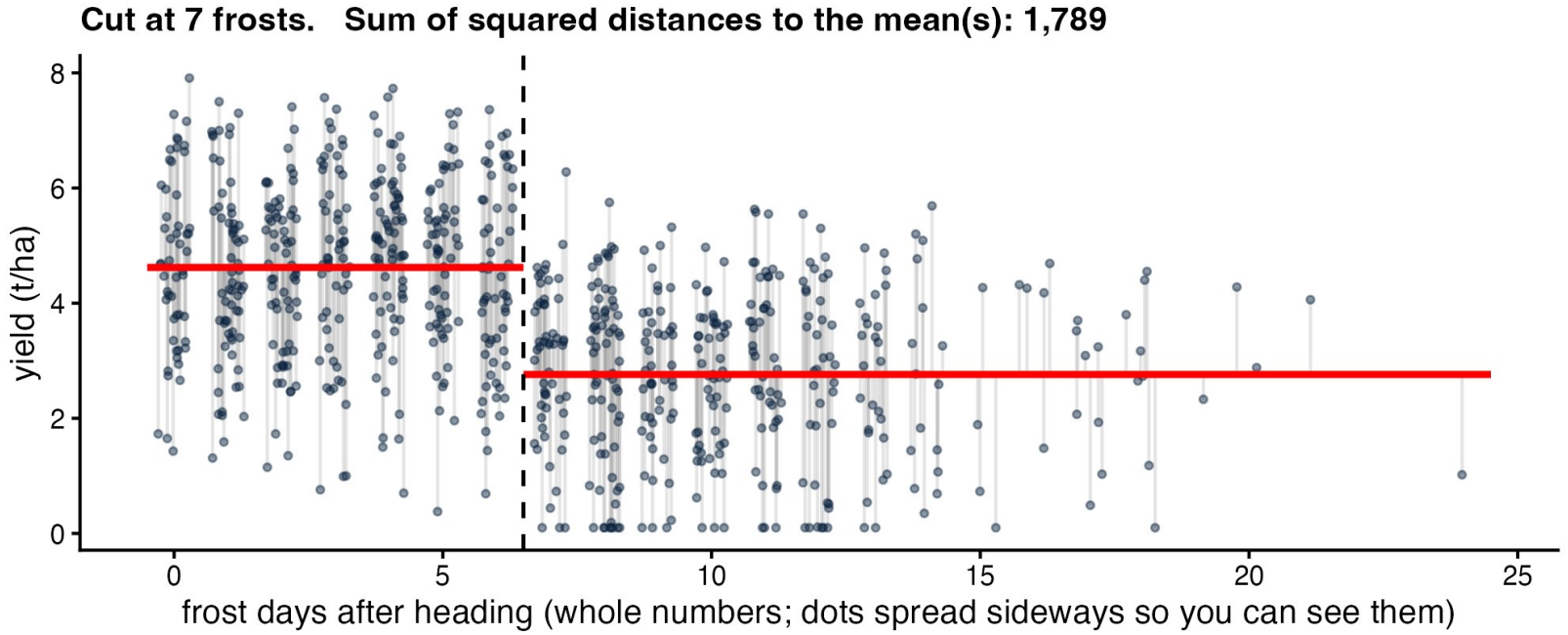
```
rf <- randomForest(yield ~ ., data = fields)
predict(rf, new_fields)
```



Scored on **next year's 400 fields**: R^2 **0.38** vs. **0.77**

A tree for a number: each group predicts its mean

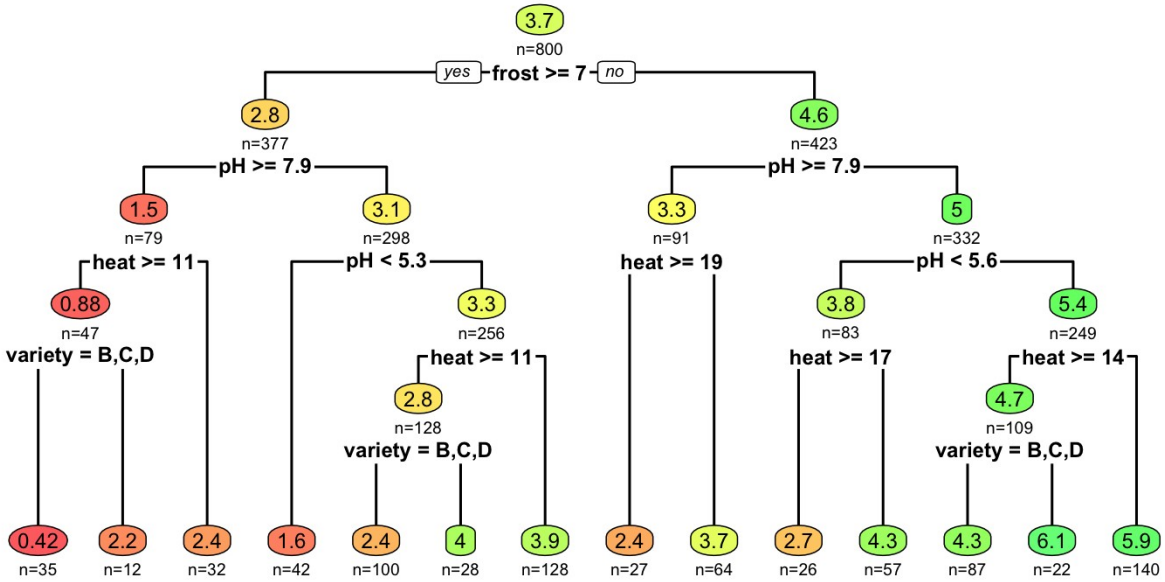
```
> rpart(yield ~ frost_days, data = fields)    # rpart grows one tree, as for the cattle
```



For a number, best cut = **smallest squared distances to the means** (not impurity)

One tree on the fields

```
> tree_fields <- rpart(yield ~ ., data = fields);  rpart.plot(tree_fields)
```



Each bubble: the **mean yield** of the fields there (t/ha). R^2 on new fields: **0.66**, better than lm

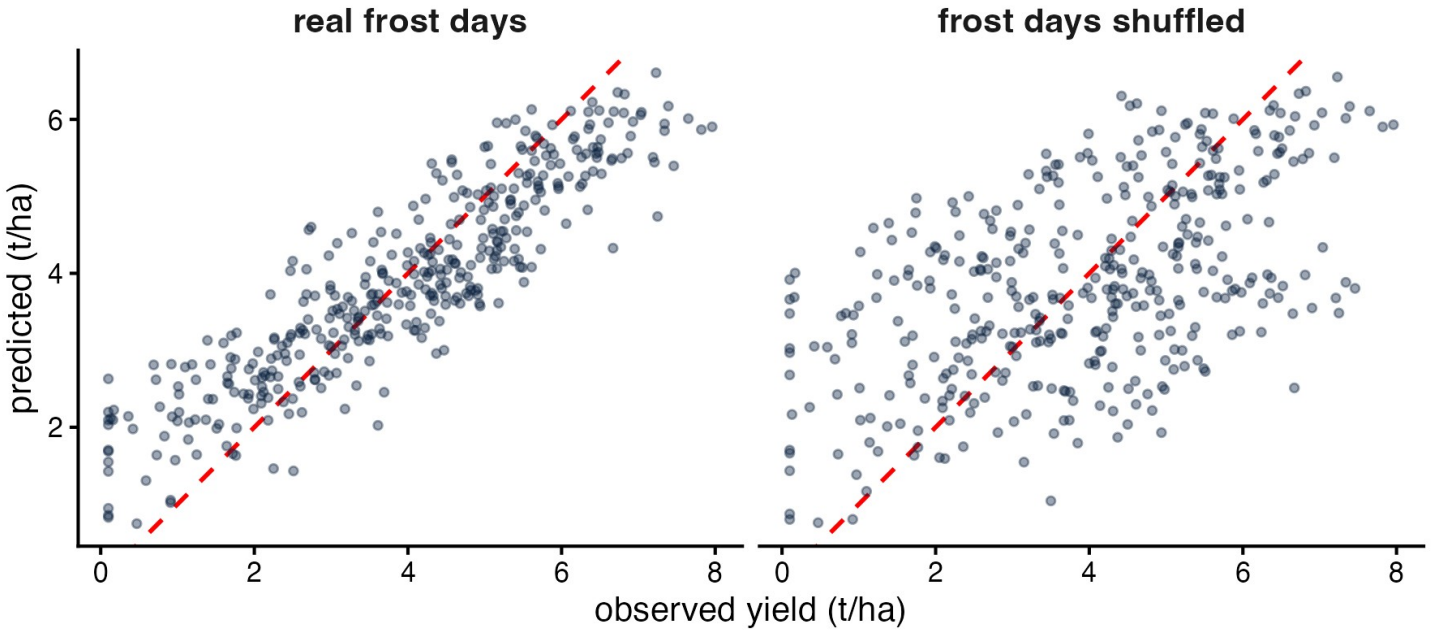
The forest in R

```
> rf_fields <- randomForest(yield ~ ., data = fields, ntree = 500,
+                           importance = TRUE)
> rf_fields

               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3  ← mtry = 10 / 3

         Mean of squared residuals: 0.7099881  ← OOB average squared miss
                   % Var explained: 77.08  ← OOB R²
```

A regression forest: each tree predicts a leaf mean; **the forest averages the 500**

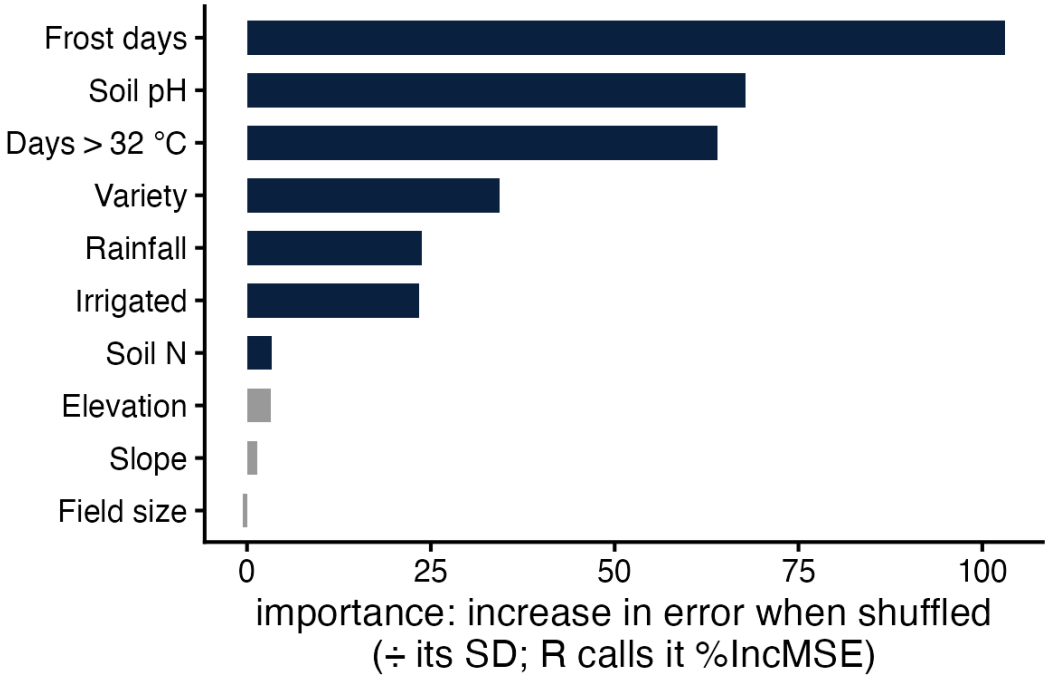
Importance: shuffle one column, watch the error

```
> shuffled$frost_days <- sample(shuffled$frost_days);  predict(rf_fields, shuffled)
```



Shuffling frost drops R^2 to **0.28**: the forest was leaning on it

Variable importance



Done while the forest is fit (importance = TRUE). For each tree:

1 error on the rows it never saw (out-of-bag)

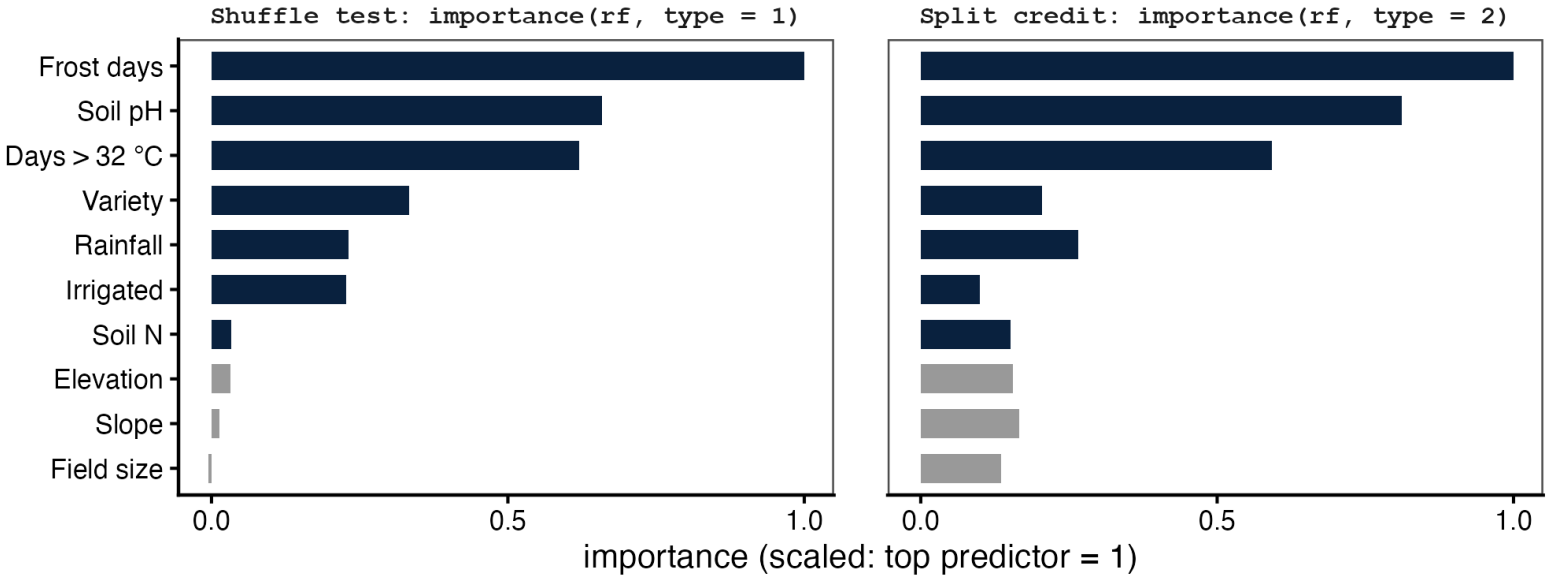
2 shuffle one column in those rows, predict again with the same tree

3 increase in error, averaged over 500 trees

importance(rf) prints these numbers; **varImpPlot(rf)** draws them. No refitting.

Two importance scores, two different questions

```
> importance(rf_fields)    # both columns
```



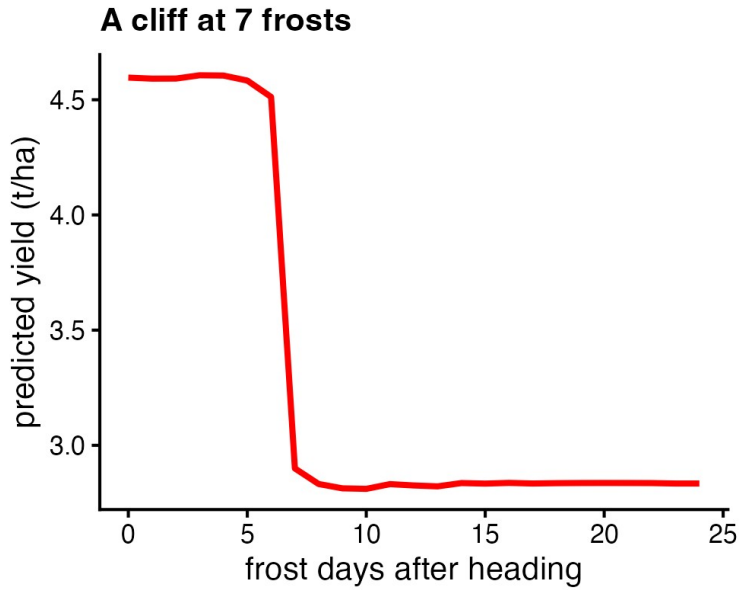
Classification names: **`MeanDecreaseAccuracy`** (type 1), **`MeanDecreaseGini`** (type 2)

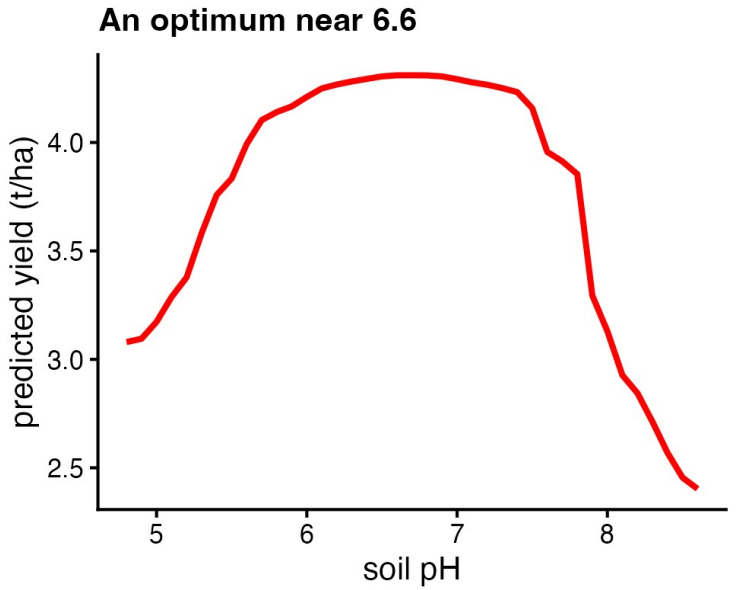
Split credit ranks **noise above irrigation**. To make inference, use the shuffle test.

Partial dependence: the shape of each effect

```
partial(rf_fields, fields, "frost_days",
        0:24)
```

```
partial(rf_fields, fields, "soil_pH",
        seq(4.8, 8.6, by = 0.1))
```



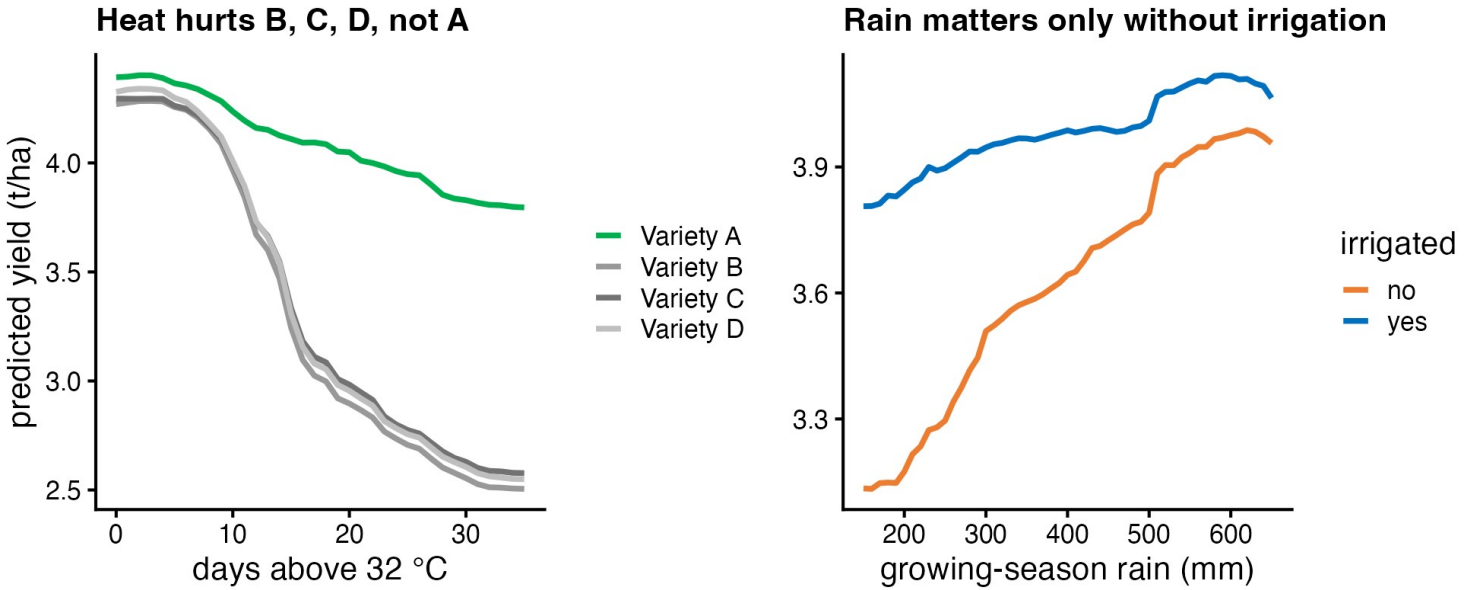


Set every field to the same value, average the predictions, **repeat along the axis**

It found the interactions too

```
partial(rf_fields, fields, "heat_days",
        0:35, by = "variety")
```

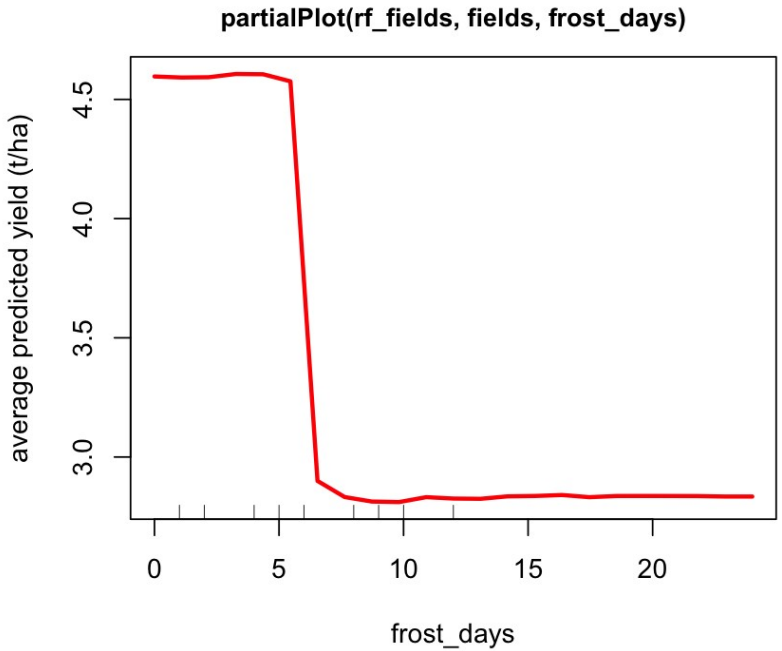
```
partial(rf_fields, fields, "rain",
        seq(150, 650, by = 10), by = "irrigated")
```



Variety × heat and **irrigation × rain**: nobody wrote an interaction term

Partial dependence in R

```
# one line: randomForest draws it
partialPlot(rf_fields, fields, frost_days)

# by hand (what it does)
pd <- data.frame(x = 0:24)
pd$yhat <- sapply(pd$x, function(v) {
  d <- fields; d$frost_days <- v  # all get v
  mean(predict(rf_fields, d))       # average
})
ggplot(pd, aes(x, yhat)) + geom_line()
```



By group (interactions): the **partial()** helper in the live script, then ggplot

The hidden rules, revealed

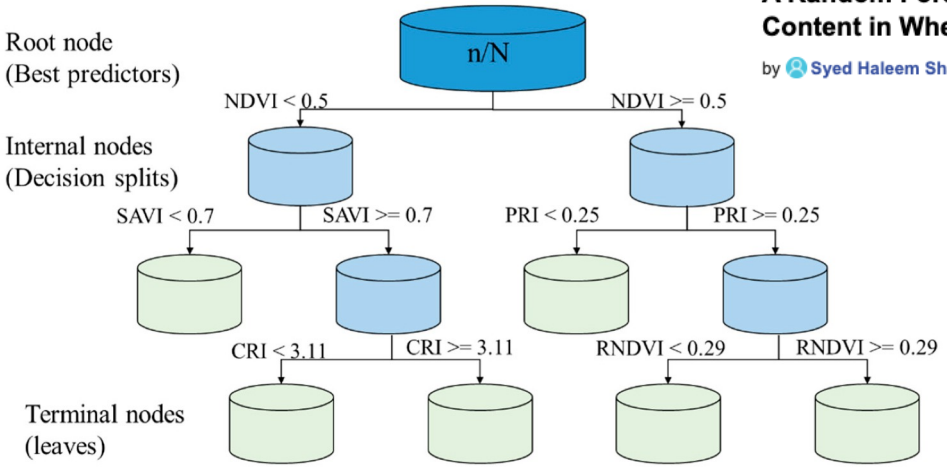
How I simulated it	What the forest found
Late frost: 7 or more → −2 t/ha	**cliff at 7 ✓**
Soil pH optimum at 6.6	**bump at 6.6 ✓**
Heat hurts every variety but A	**A flat, B–D fall ✓**
Irrigation removes the need for rain	**flat when irrigated ✓**
Elevation, slope, field size: nothing	**importance ≈ 0 ✓**

A flexible model **discovers** structure you would never have written into lm

A Random Forest Machine Learning Approach for the Retrieval of Leaf Chlorophyll Content in Wheat

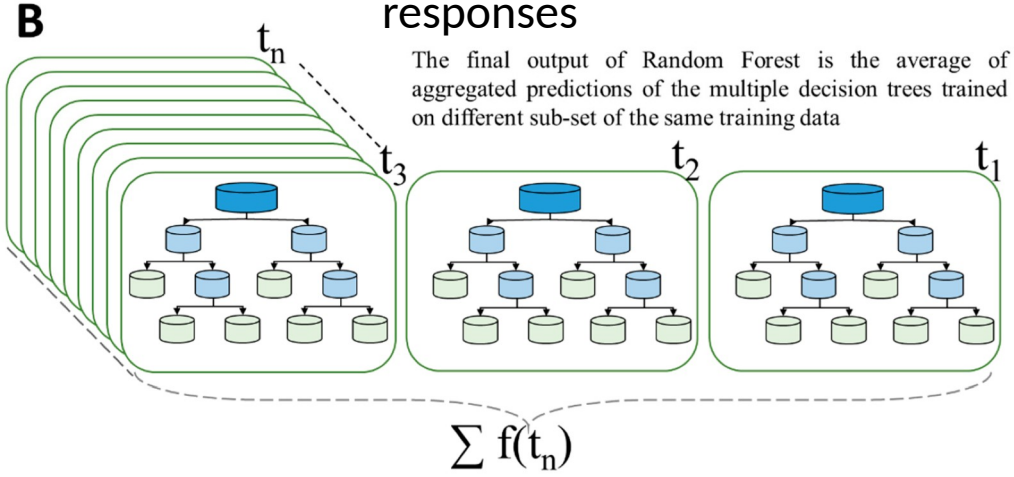
by Syed Haleem Shah [1,*], Yoseline Angel [1], Rasmus Houborg [2], Shawkat Ali [3] and Matthew F. McCabe [1]

A

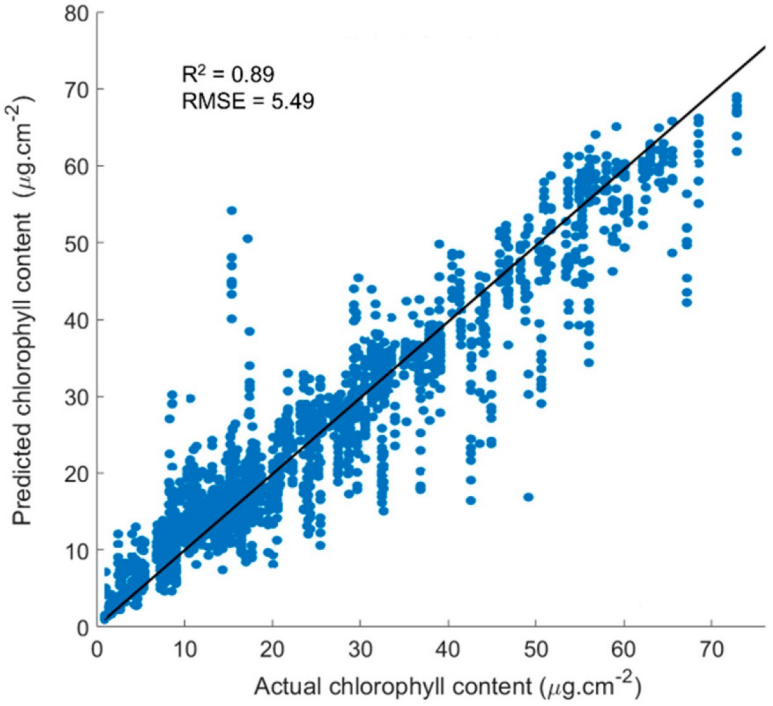


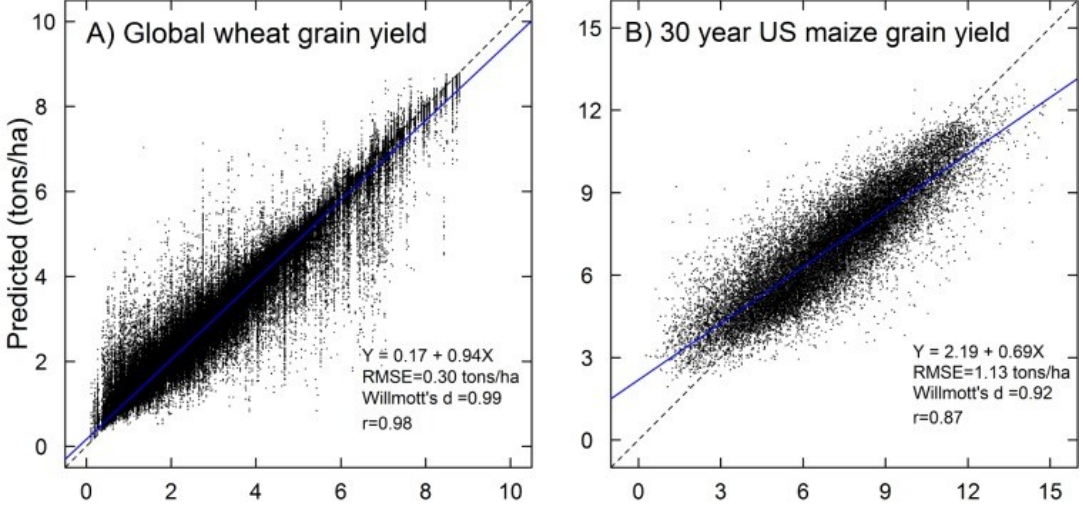
Leaves = Predicted responses

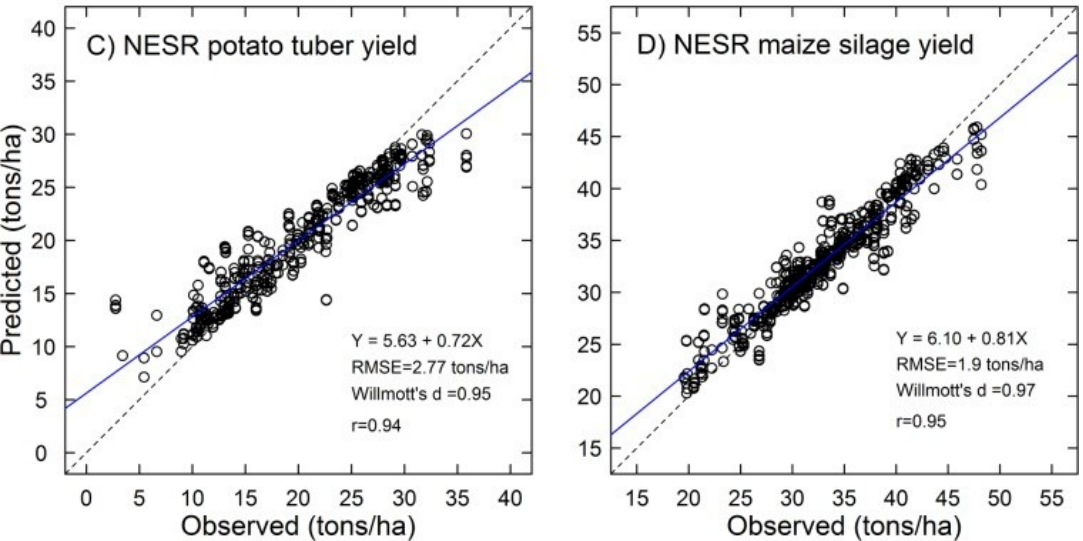
B



The final output of Random Forest is the average of aggregated predictions of the multiple decision trees trained on different sub-set of the same training data



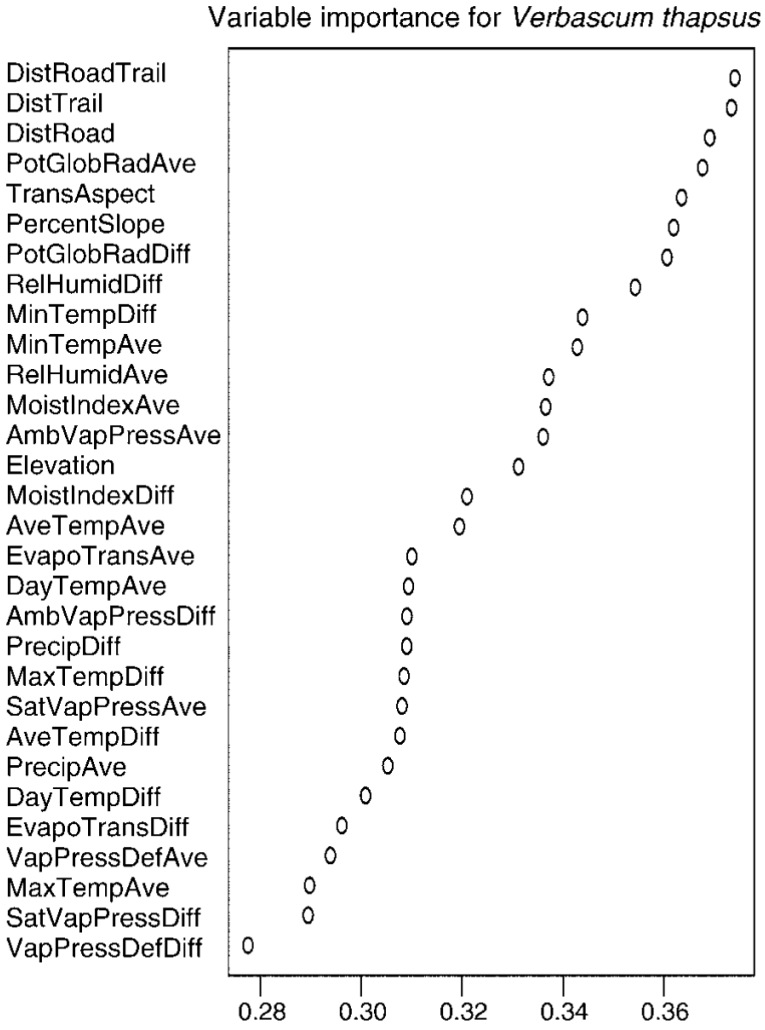




Random Forests for Global and Regional Crop Yield Predictions

Jig Han Jeong,[1] Jonathan P. Resop,[2,3] Nathaniel D. Mueller,[4,5] David H. Fleisher,[3] Kyungdahm Yun,[1] Ethan E. Butler,[6] Dennis J. Timlin,[3] Kyo-Moon Shim,[7] James S. Gerber,[8] Vangimalla R. Reddy,[3] and Soo-Hyung Kim[1,*]

Global wheat and US maize grain yields			Importance Rank	
Variable	**Abbreviation**	**Unit**	**wheat**	**maize**
Averaged monthly temperature	AVT	°C	8	9
Annual evapotranspiration	EVA	mm	2	6
Summer solstice day length	DAYL	hour	4	7
Maximum monthly temperature	MAX	°C	7	3
Mean coldest quarter Temperature	MCQ	°C	9	11
Minimum monthly temperature	MIN	°C	10	8
Mean warmest quarter temperature	MWQ	°C	6	10
Nitrogen fertilizer application rate	NFERT	kg/ha	1	2
Growing season precipitation	PRE49	mm	3	4
Annual precipitation	PRECI	mm	5	5
Year (US maize only)	YR		-	1



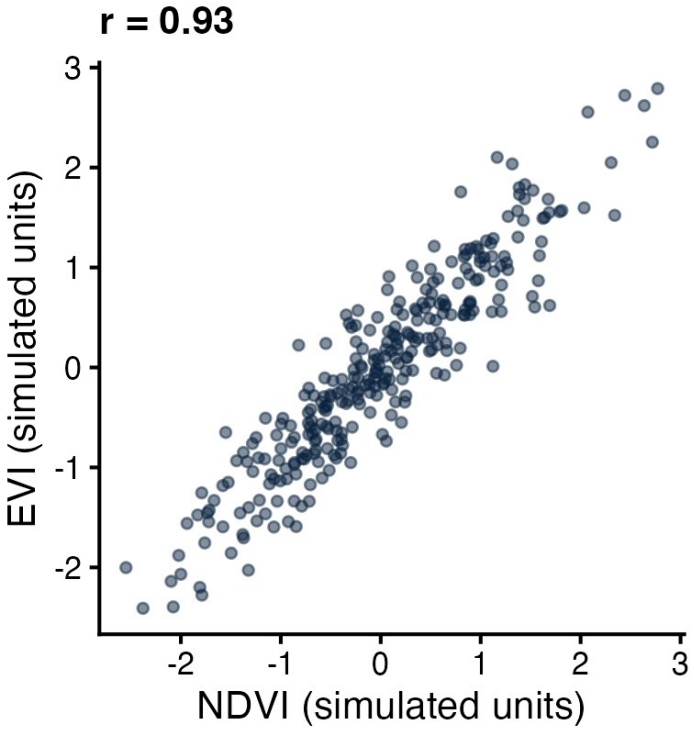
RANDOM FORESTS FOR CLASSIFICATION IN ECOLOGY

D. Richard Cutler,[1,7] Thomas C. Edwards, Jr.,[2] Karen H. Beard,[3] Adele Cutler,[4] Kyle T. Hess,[4] Jacob Gibson,[5] and Joshua J. Lawler[6]

Common Mullen



Two predictors that measure the same thing



Simulated canopy plots

biomass ← **canopy vigor** + LAI (leaf area)

NDVI and **EVI**: two greenness indices from the same images, both noisy measures of vigor

PRI: no effect

NDVI and EVI carry **almost the same information**

Live poll

Multiple choice

NDVI and EVI correlate r = 0.93. Add EVI to a forest that had NDVI. NDVI's importance...

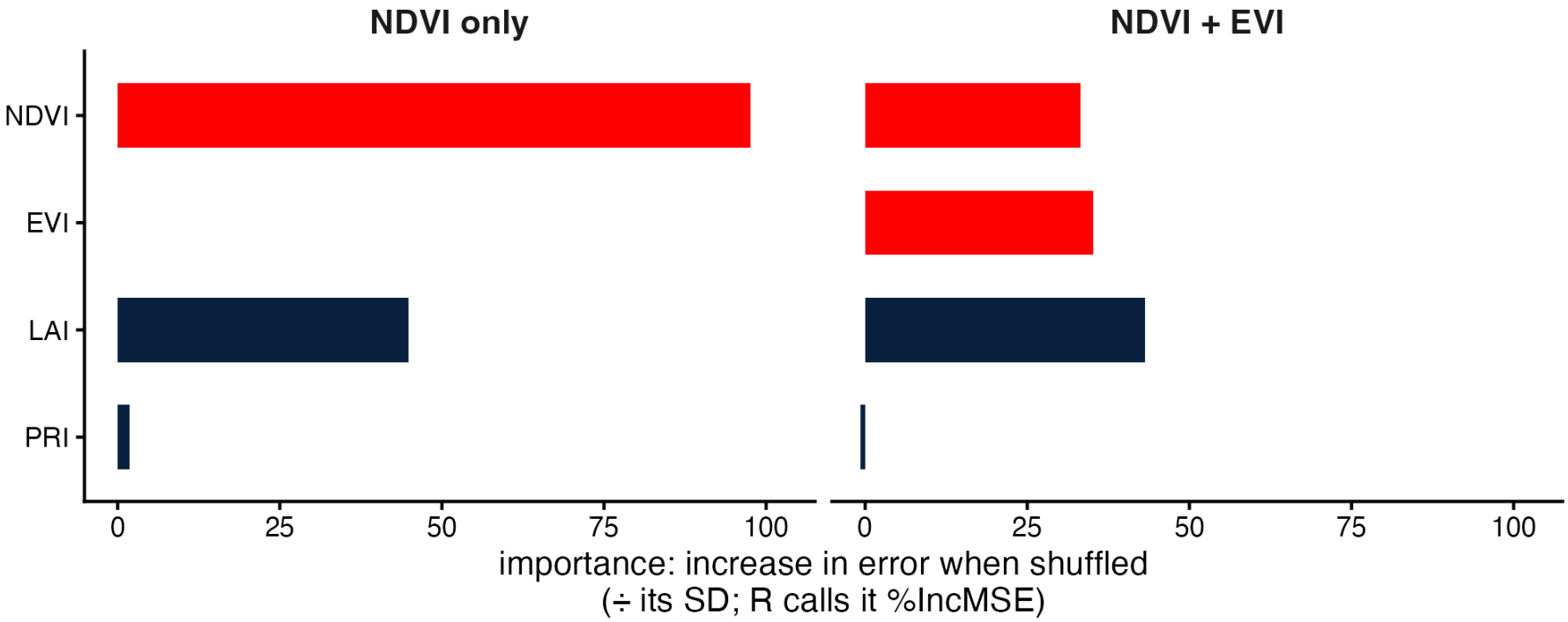
A Goes up
B Drops, about by half
C Stays the same

Scan with your phone

Join on your phone or laptop: greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w9-ndvi

host screen

Correlated predictors share the credit



NDVI alone: **98**. With EVI: **33 + 35**: shuffle one, the other covers for it

Reading importance scores

1 Correlated predictors split the credit

a low score does not mean the predictor is irrelevant

2 High importance ≠ a cause

NDVI scores high, but vigor drives biomass; NDVI only measures it

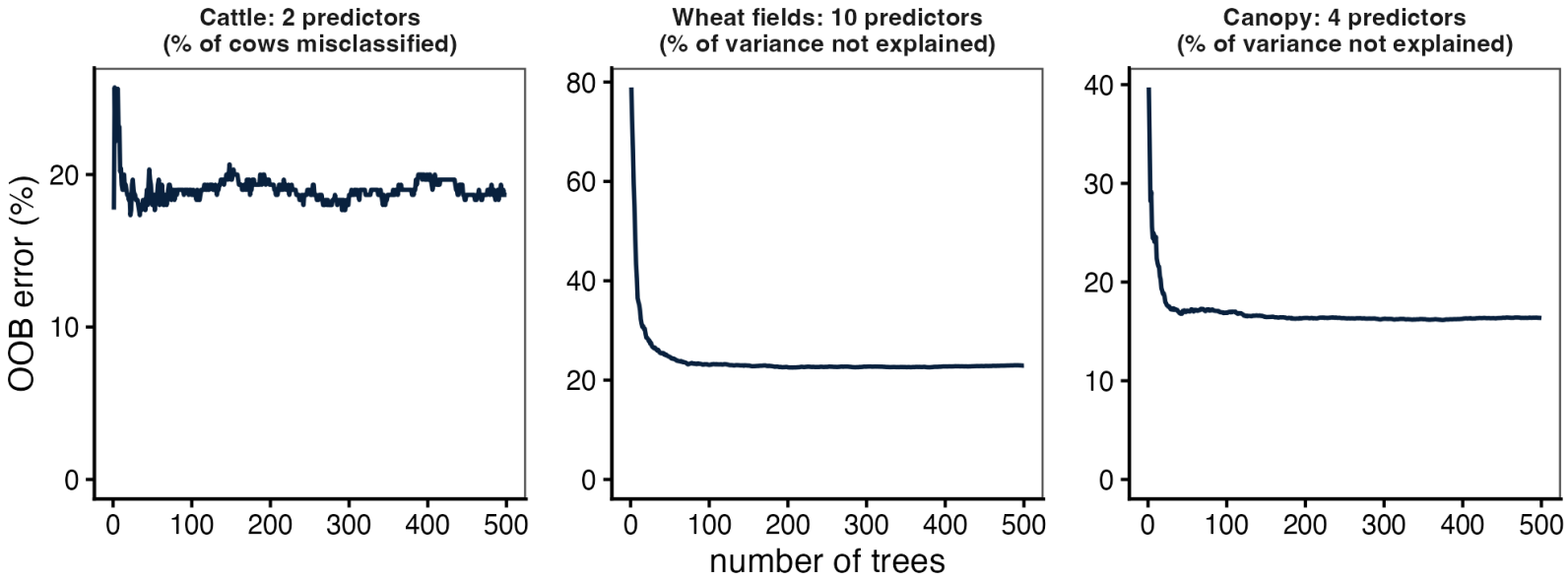
3 Check correlations first

then drop one, combine them (e.g. PCA, Week 7), or read them as a group

Importance = how much **this forest leans on a column**, not how much it matters in nature

How many trees? Every forest in this lecture

```
> plot(rf_fields)    # OOB error vs number of trees
```



One tree is bad; **by ~100 trees the curve is flat**. More trees never overfit, they just stop helping.

1 Trees
2 One tree
3 Forests
4 Complex data
5 **Neural nets**
6 Which model?



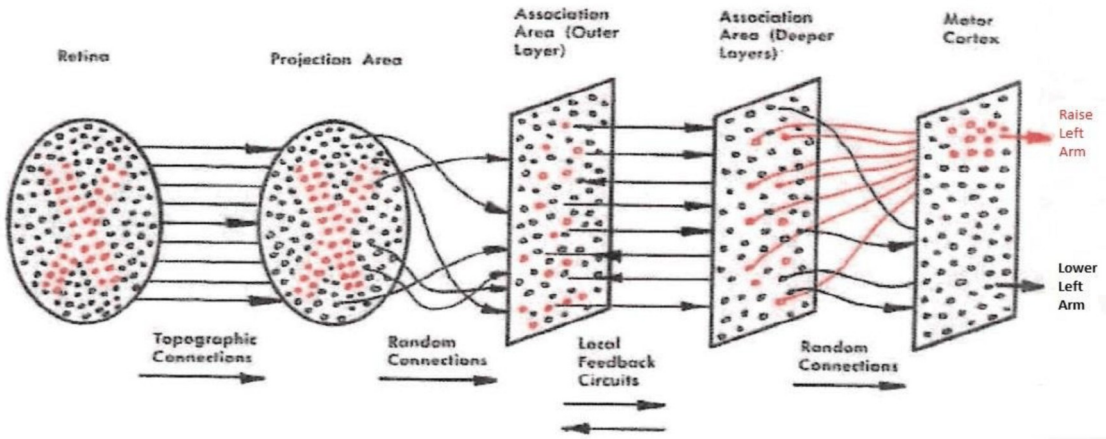


FIG. 1 — Organization of a biological brain. (Red areas indicate active cells, responding to the letter X.)

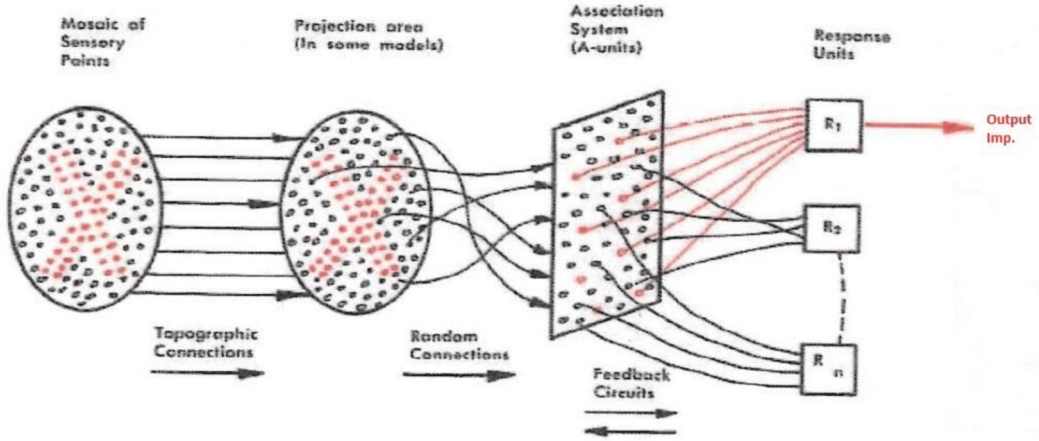


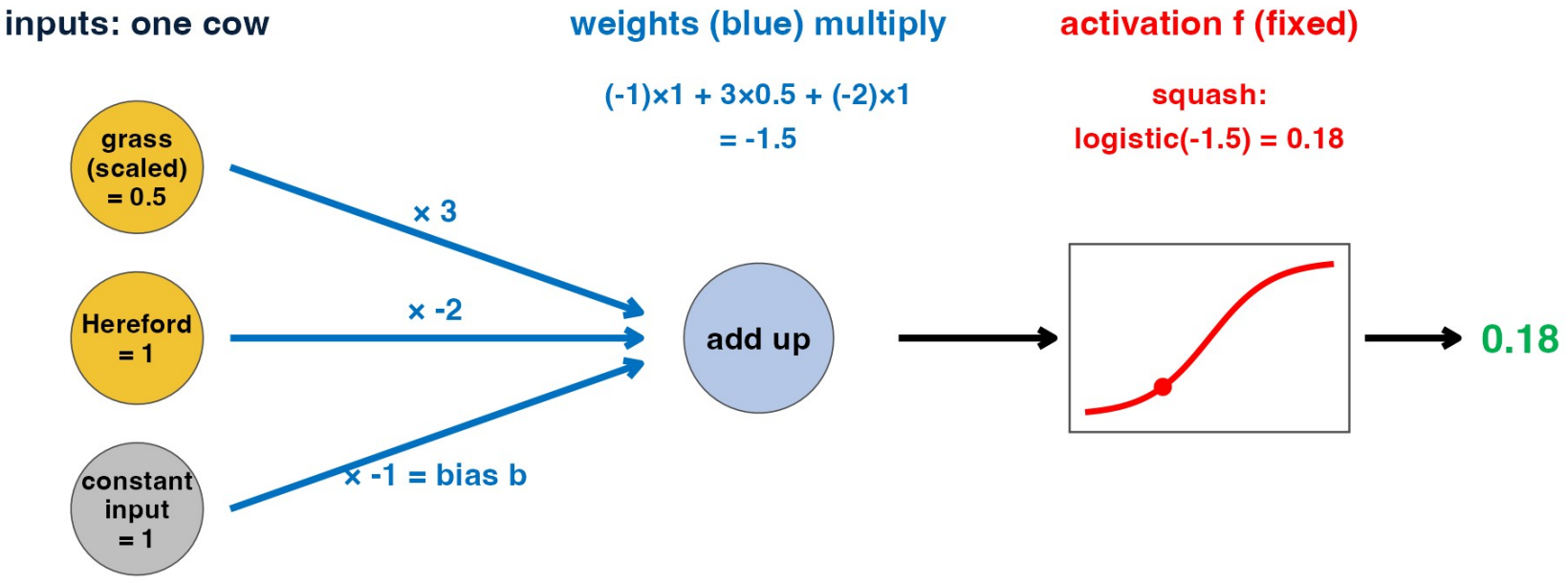
FIG. 2 — Organization of a perceptron.

From a neuron to a node

A neuron	A node in a network
gets signals from other cells	gets numbers: the inputs (grass, Hereford)
some connections are stronger	each input is multiplied by a weight
adds up what comes in	adds them up, plus a bias (an intercept)
fires if the total is high enough	squashes the total with an activation f

The analogy is loose. The node itself is **simple arithmetic**: next slide, with numbers.

One node, with numbers



Fitted from data: **the weights and the bias**. Chosen by you: **the activation f**.

One node = a regression

$$\text{output} = f\left(b + w_1 x_1 + w_2 x_2 + \dots\right)$$

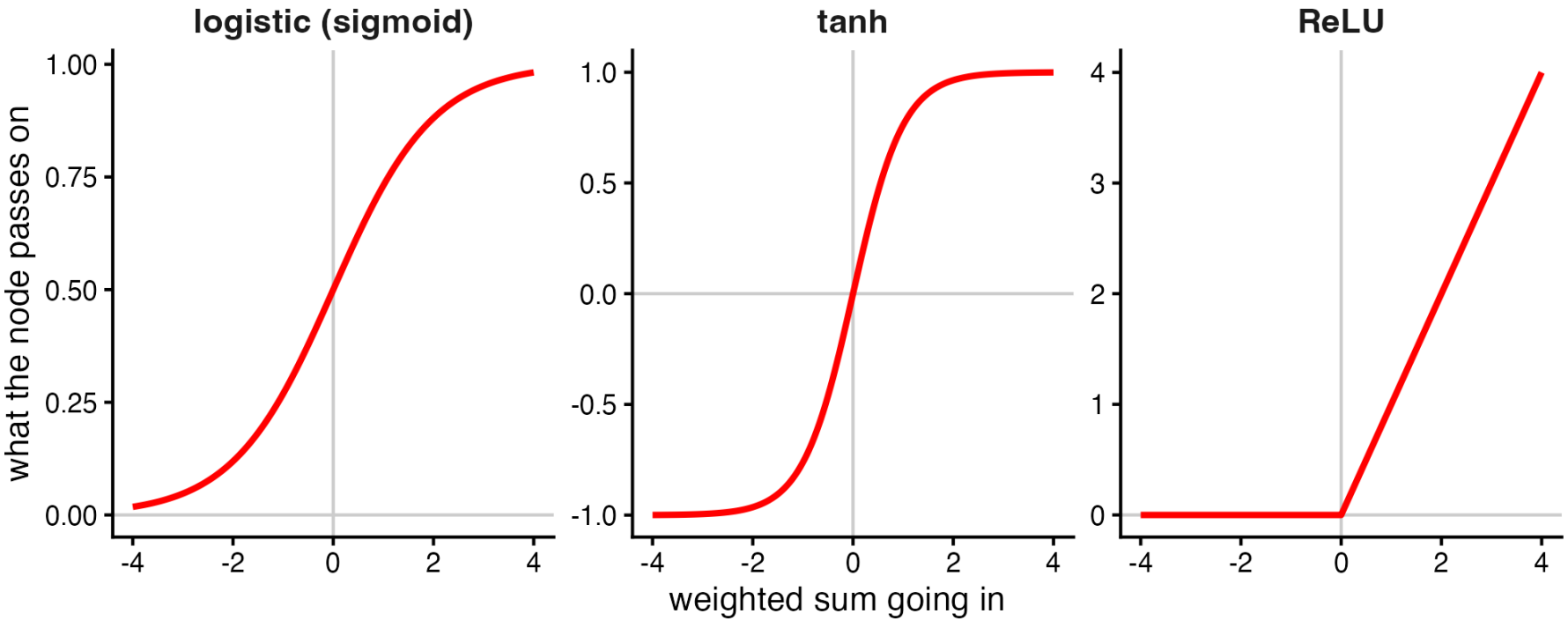
b: bias (an intercept) w: weights (slopes)

f: activation, e.g. logistic $= 1/(1 + e^{-z})$

With logistic f: this is exactly **logistic regression** (Week 3)

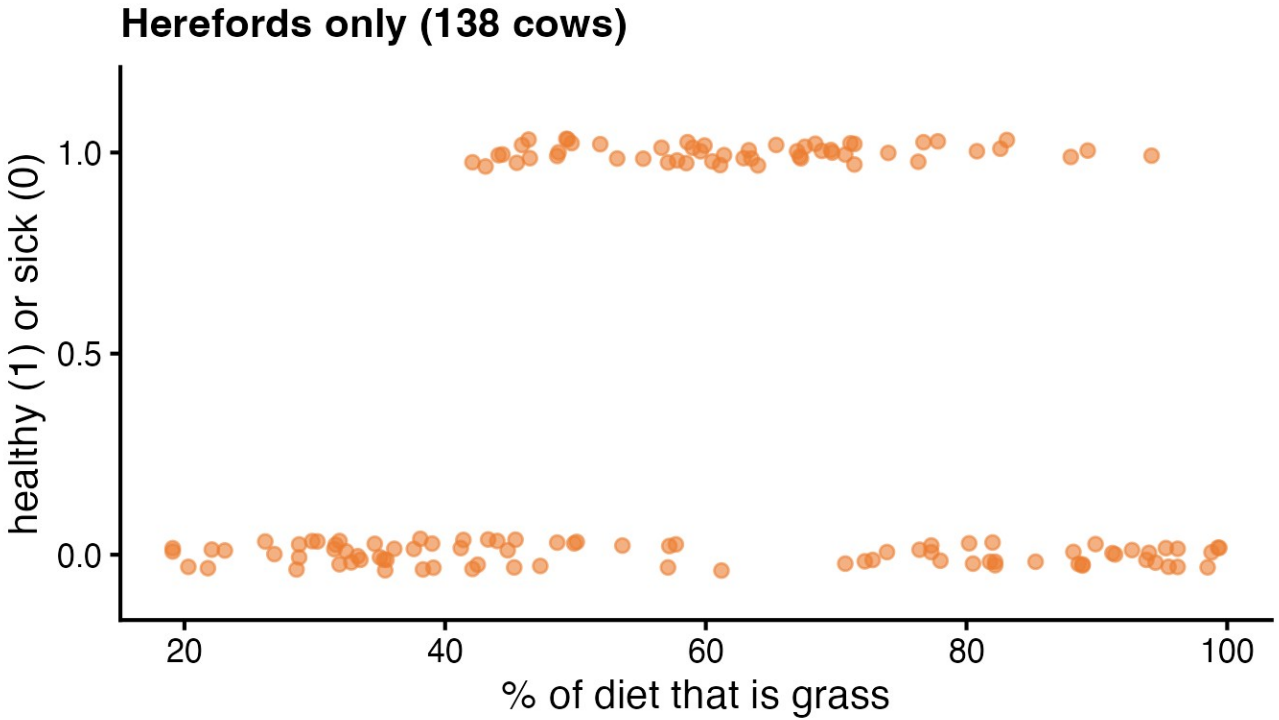
b and w are fitted, like a GLM's coefficients. **f is fixed**: no parameters.

Activation functions: the choices for f



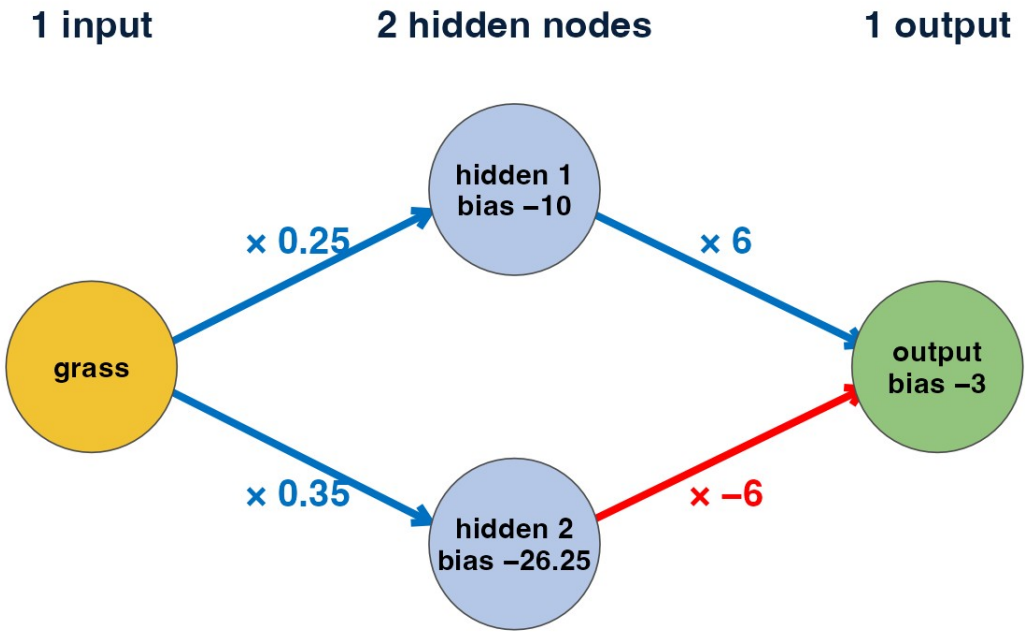
Fixed shapes, nothing fitted. The bend matters: **without it, stacked nodes are just lm**

Start smaller: one breed, one predictor



Herefords: sick at low grass, healthy in the middle, sick again at high grass. **One S-curve can't do that.**

The simplest network: grass only, two hidden nodes



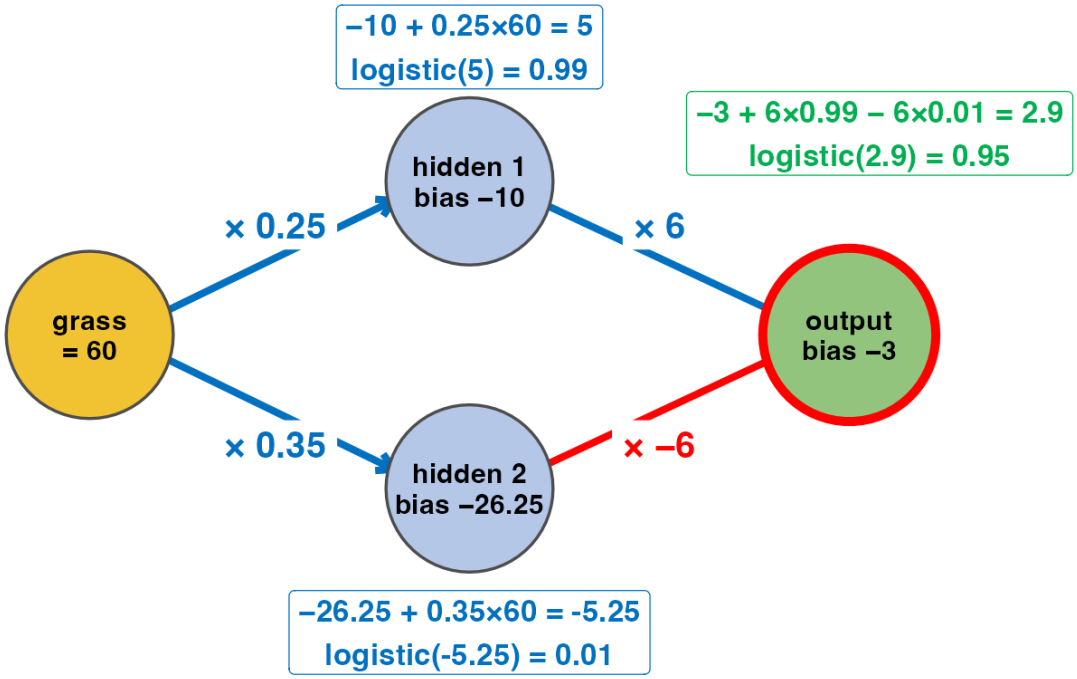
Each hidden node:
bias + weight × grass,
then logistic

Output node:
bias + weight × each hidden node, then logistic

7 numbers in all (4 weights, 3 biases)

Weights picked by hand for now. **Fitting = finding these 7 numbers from data**

One cow, step by step: 60% grass



Hidden 1
bias −10 + weight 0.25: the total passes 0 at 40% grass (0.25×40 = 10). **A switch that turns on above 40%.**

Hidden 2
bias −26.25 + weight 0.35: passes 0 at 75% grass. **A second switch, on above 75%.**

Output
bias −3: both switches off → sick. **×6**: switch 1 on pushes to healthy. **×−6**: switch 2 on cancels it.

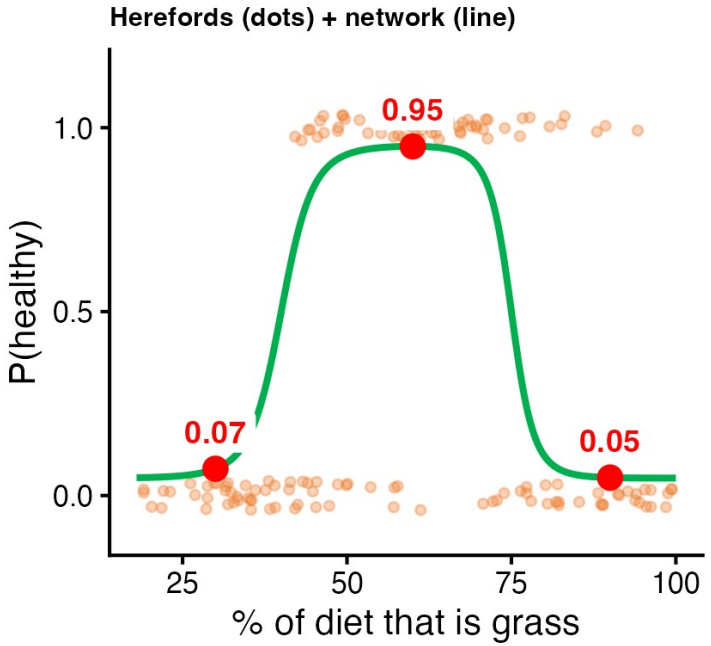
This cow: **P(healthy) = 0.95**. Every prediction is just this arithmetic.

Follow three cows through it

Cow on 30% grass
hidden 1: logistic(−10 + 0.25×30) = logistic(-2.5) = 0.08
hidden 2: logistic(−26.25 + 0.35×30) = logistic(-15.8) = 0.00
output: logistic(−3 + 6×0.08 − 6×0.00) = **0.07**

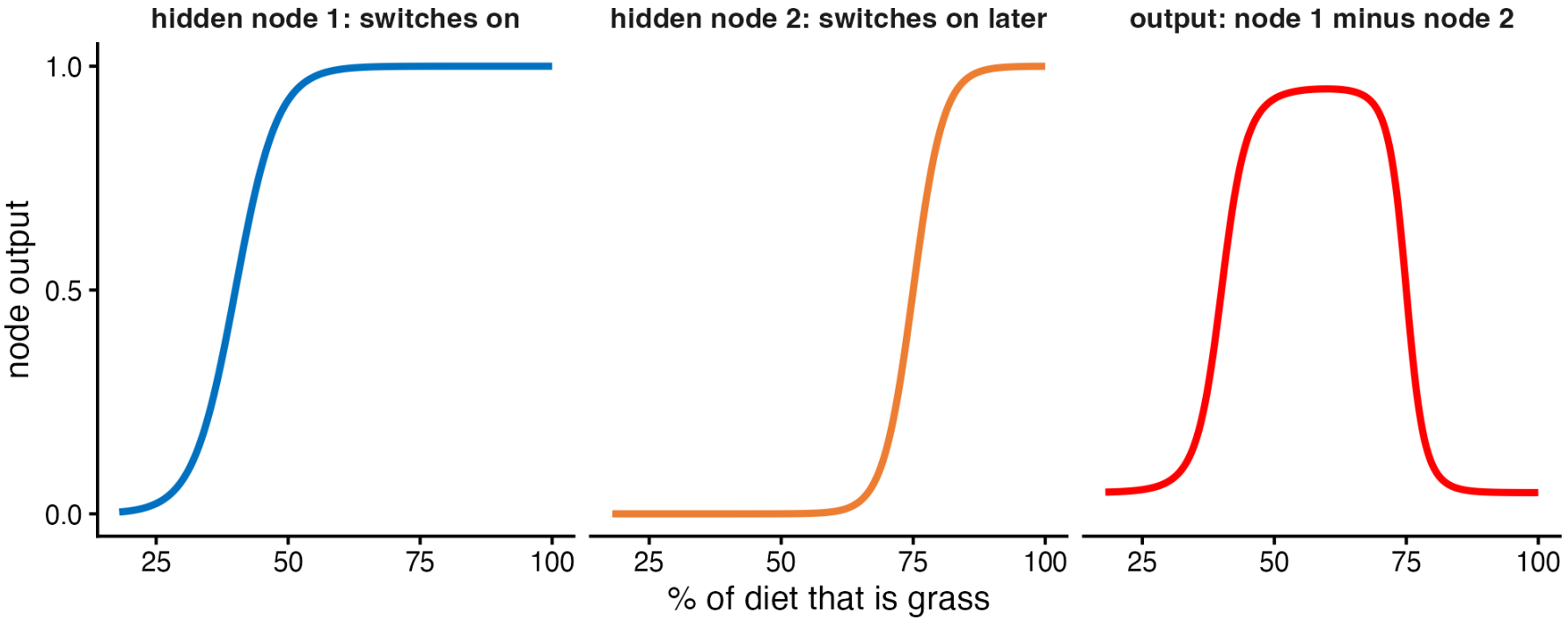
Cow on 60% grass
hidden 1: logistic(−10 + 0.25×60) = logistic(5.0) = 0.99
hidden 2: logistic(−26.25 + 0.35×60) = logistic(-5.2) = 0.01
output: logistic(−3 + 6×0.99 − 6×0.01) = **0.95**

Cow on 90% grass
hidden 1: logistic(−10 + 0.25×90) = logistic(12.5) = 1.00
hidden 2: logistic(−26.25 + 0.35×90) = logistic(5.2) = 0.99
output: logistic(−3 + 6×1.00 − 6×0.99) = **0.05**



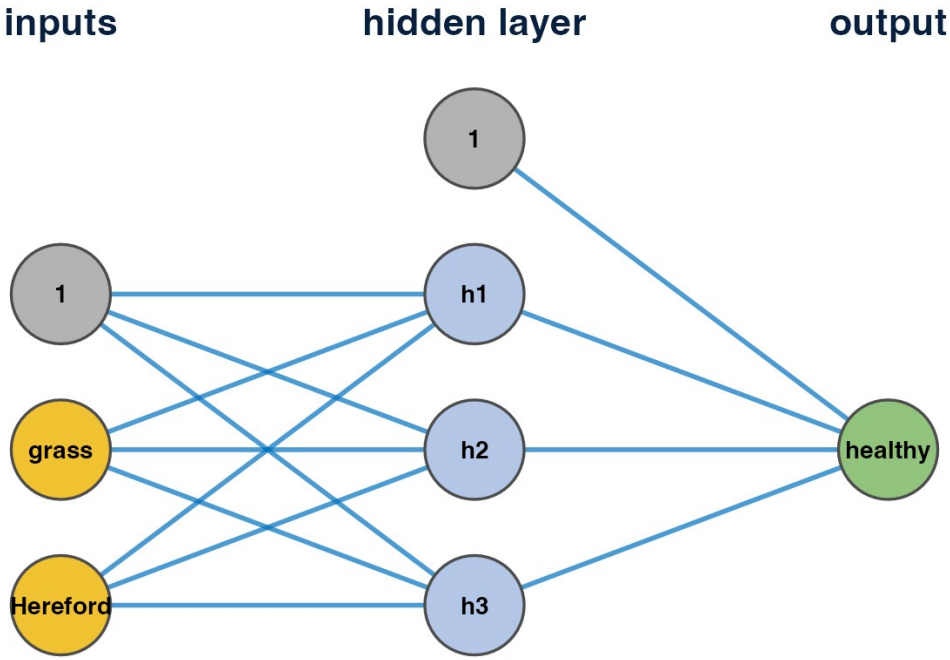
Low: neither node on. Middle: node 1 on. High: both on, **they cancel**. The line follows the Herefords.

Every grass value at once: two S-curves make a bump



Each hidden node is a switch. With enough of them a network can draw **any shape**

Two predictors, three hidden nodes: the cattle



Hidden nodes
each one is a node like before, on the inputs

Output node
a node on the hidden nodes' outputs

Every line is **a weight**. Grey 1 = a constant input; **its line's weight is the bias**

Live poll

Multiple choice

A network: 2 inputs, 3 hidden nodes, 1 output. How many numbers (weights + biases) are fitted?

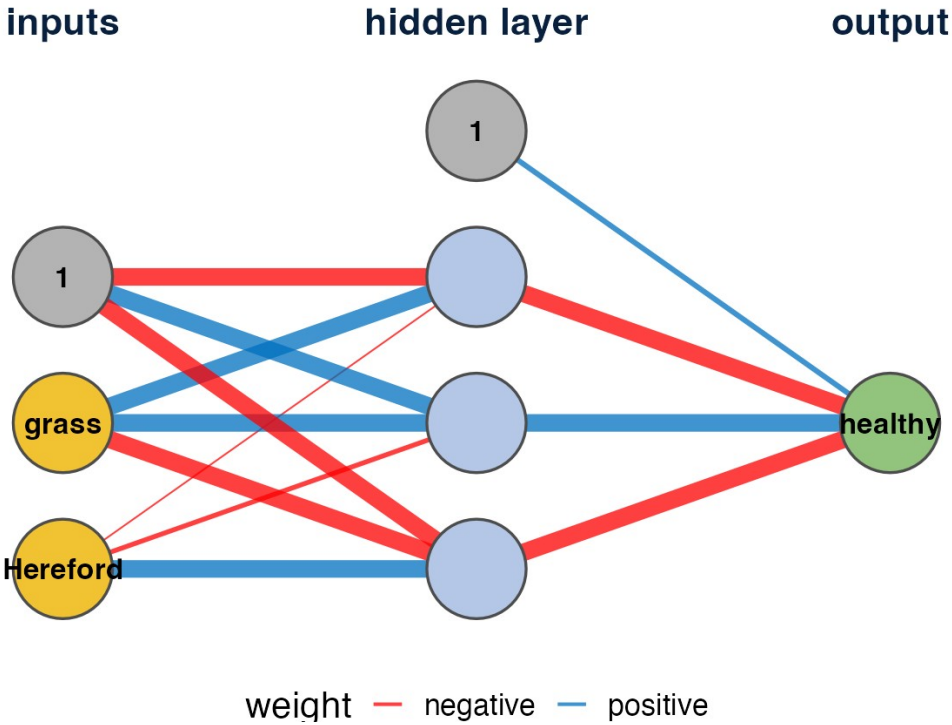
A 5
B 9
C 13
D 20

Scan with your phone

Join on your phone or laptop: greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w9-weights

host screen

Our network, fitted



13 numbers fitted

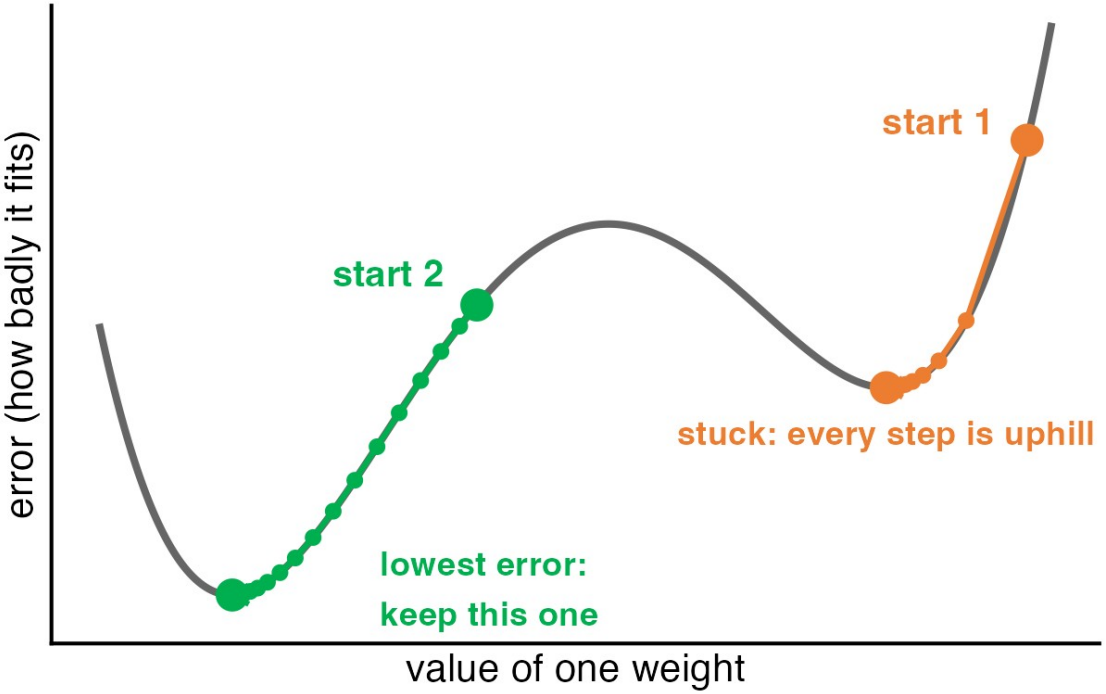
3 hidden nodes × 3
(2 weights + bias)

output: 3 weights + bias

thicker = bigger
blue +, red −

neuralnet(healthy ~ grass + breedHereford, data = cattle_nn, hidden = 3)

How a network learns: walk downhill



1 Start: random weights

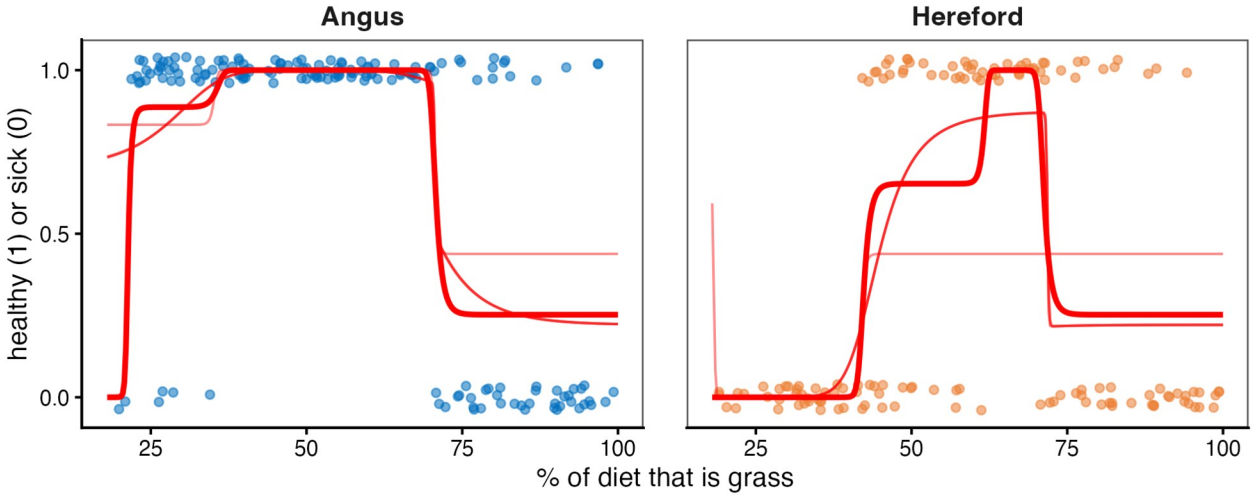
2 Measure the error

3 Nudge every weight and bias downhill

4 Repeat until it stops improving

Cartoon: 1 weight (really all 13 at once). Where you finish **depends on where you start**

Random starts: keep the best



`rep = 5`
5 random starts,
5 red curves

Thick = the start with the lowest error

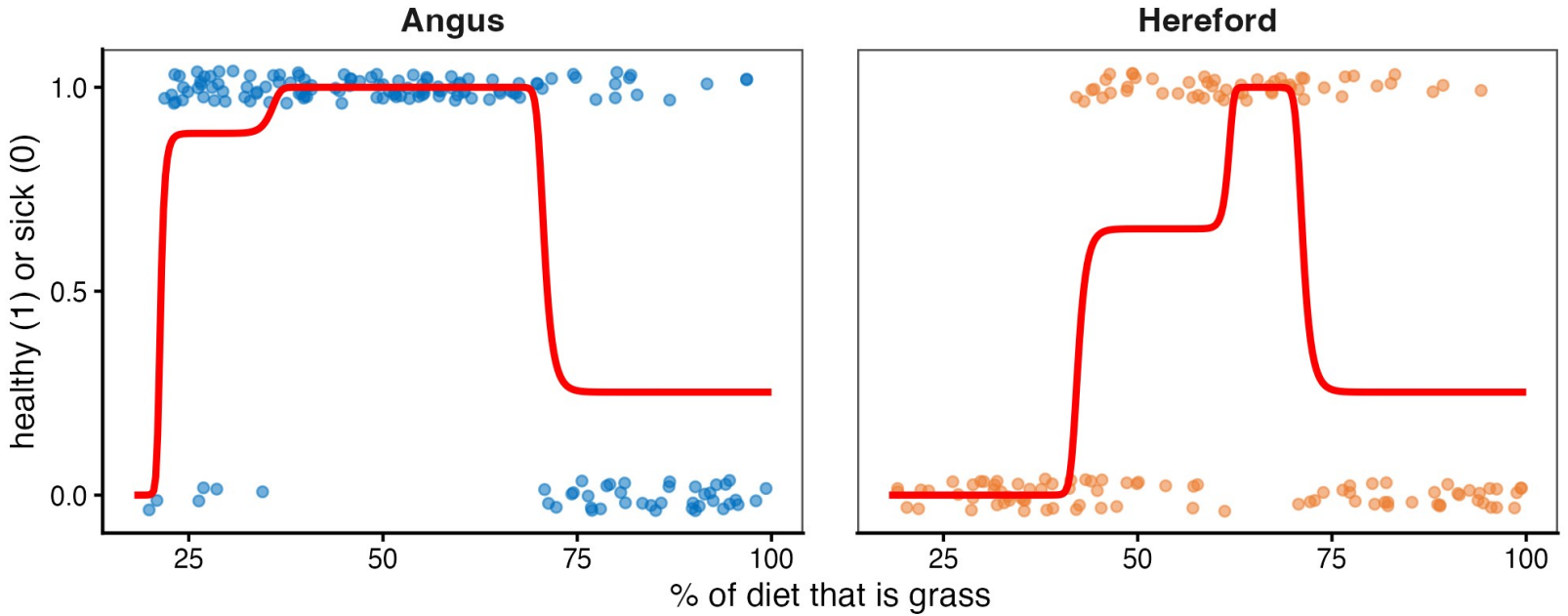
like nstart in k-means

```
> round(nn$result.matrix["error", ], 1)
[1] 124.0  99.4 104.2 104.0 103.9   ← keep the best start
```

error = **−log-likelihood** for 0/1 data (Week 5): lower = better fit

The network on the cattle

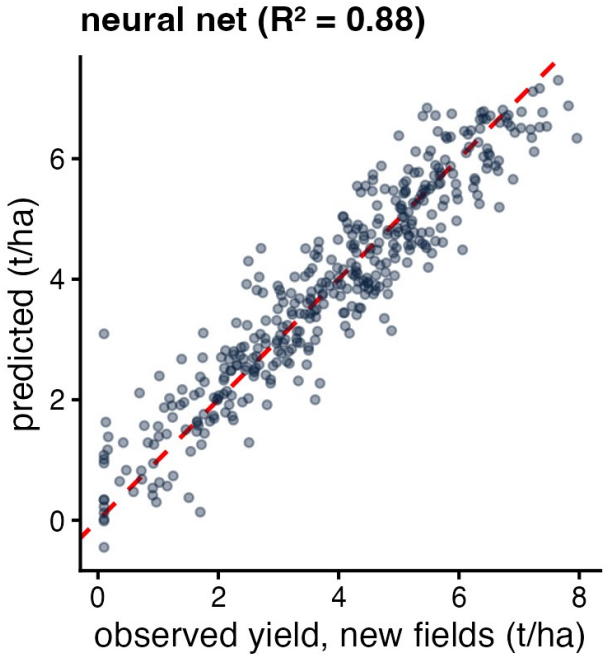
```
> predict(nn, new_data, rep = best)
```



3 hidden nodes = 3 steep S-curves, so it looks like steps: **the rise, the drop, the breed difference**

A network for a number: wheat yield

```
> neuralnet(yield_z ~ ., data = train_nn, hidden = 5, linear.output = TRUE)
```



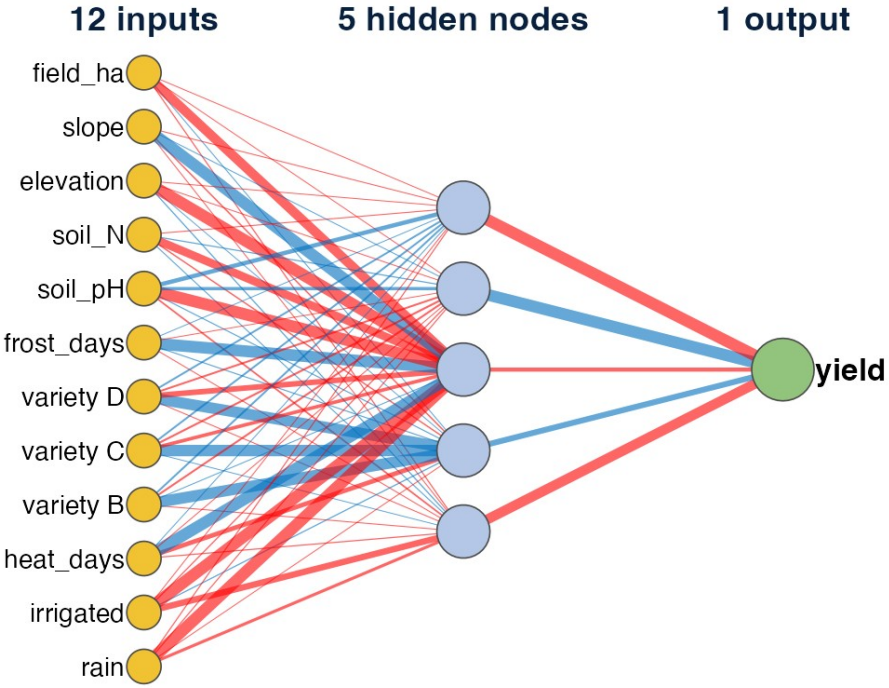
12 inputs (10 predictors; factors → 0/1)

5 hidden nodes

`linear.output = TRUE`
the output skips the S-curve, so it can be any number

R^2 **0.88** on next year's fields (forest: **0.77**)

How big was the wheat network?



12 inputs
10 predictors; factors → 0/1 columns

1 hidden layer, 5 nodes

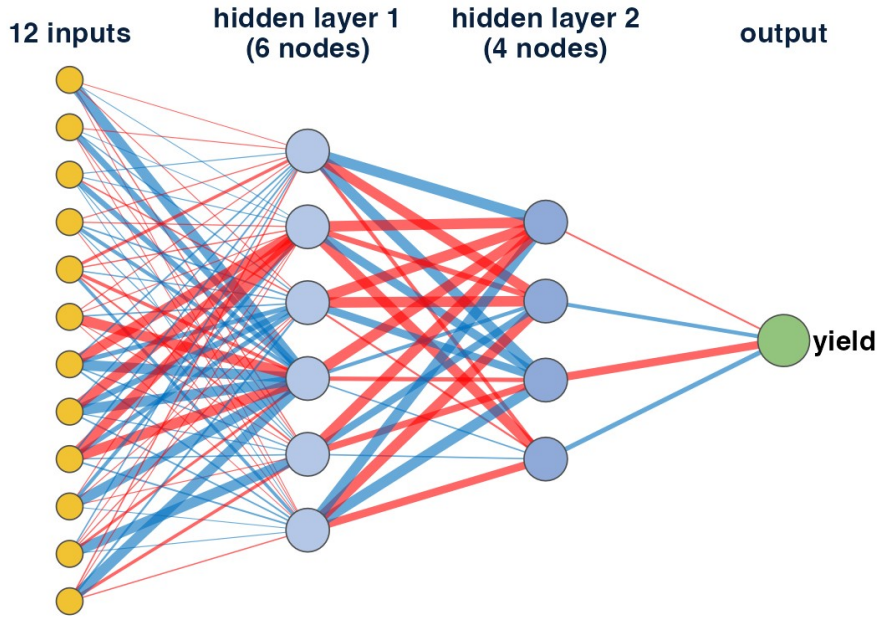
71 numbers fitted
12×5 weights in, 5 out, + 5 + 1 biases

Each line = one weight (biases not drawn)

Image and language models: many hidden layers and **millions to billions of numbers**

More layers: hidden = c(6, 4)

```
> neuralnet(yield_z ~ ., data = train_nn, hidden = c(6, 4), linear.output = TRUE)
```



Layer 2's nodes take layer 1's outputs as their inputs

111 numbers fitted (one layer: 71)

R^2 on new fields: **0.85**
one layer of 5: 0.88

Deeper is more flexible, **not automatically better**: test on new data

Three rules for neuralnet()

1 Numbers only

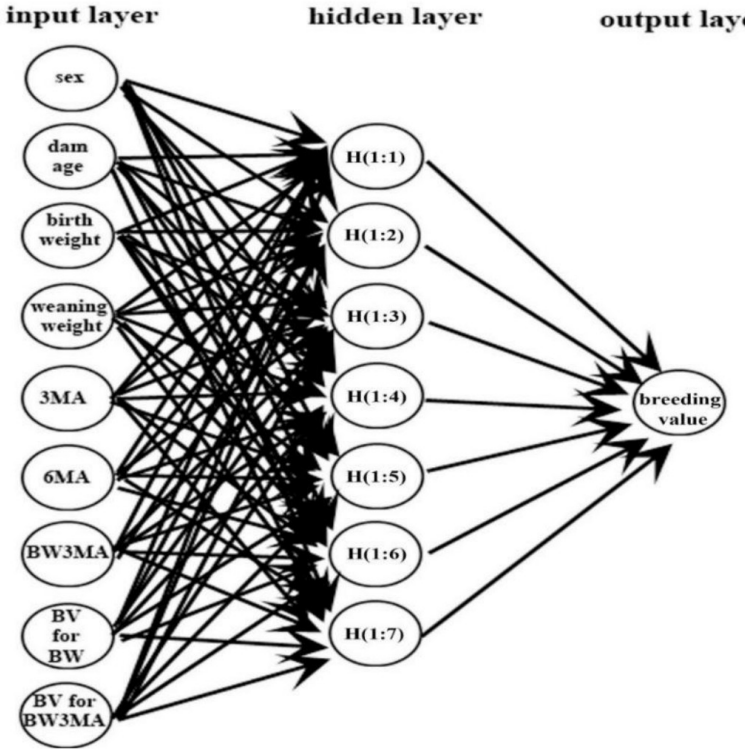
factors → 0/1 columns with model.matrix()

2 Scale the inputs

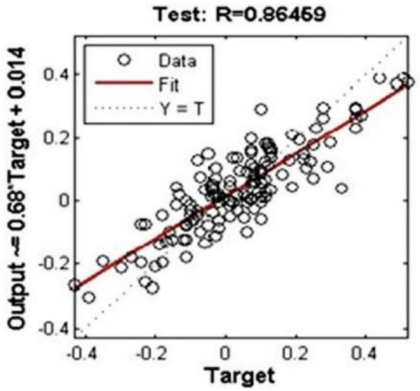
scale(), using the training means and SDs

3 Many starts

rep = 5 or more; keep the lowest error



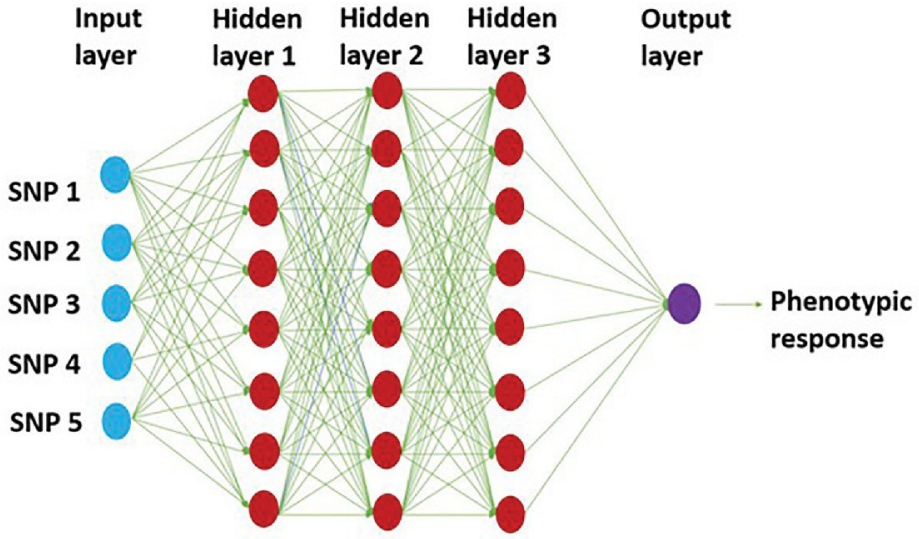




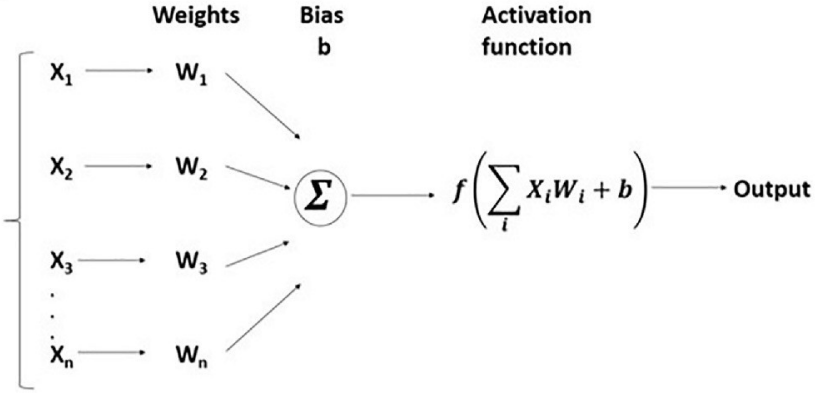
Predicting breeding value of body weight at 6-month age using Artificial Neural Networks in Kermani sheep breed

Hamidreza Ghotbaldini[1], Mohammadreza Mohammadabadi[1*], Hossein Nezamabadi-pour[1], Olena Ivanivna Babenko[2], Maryna Vitaliivna Bushtruk[1] and Serhii Vasyliovych Tkachenko[1]

A



B



Deep Learning for Predicting Complex Traits in Spring Wheat Breeding Program

Karansher S. Sandhu[1], Dennis N. Lozada[2], Zhiwu Zhang[1], Michael O. Pumphrey[1] and Arron H. Carter[1*]

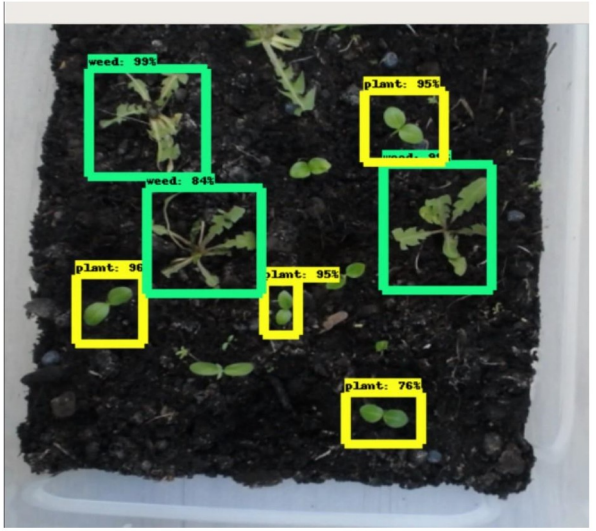


Model	Grain yield	Grain protein content	Test weight	Plant height	Heading date
rrBLUP	0.39	0.48	0.45	0.52	0.46
MLP	**0.44**	**0.53**	**0.48**	**0.57**	**0.51**
CNN	0.39	0.48	0.47	0.55	0.49

The highest prediction accuracy is bolded for each trait under each model scenario.





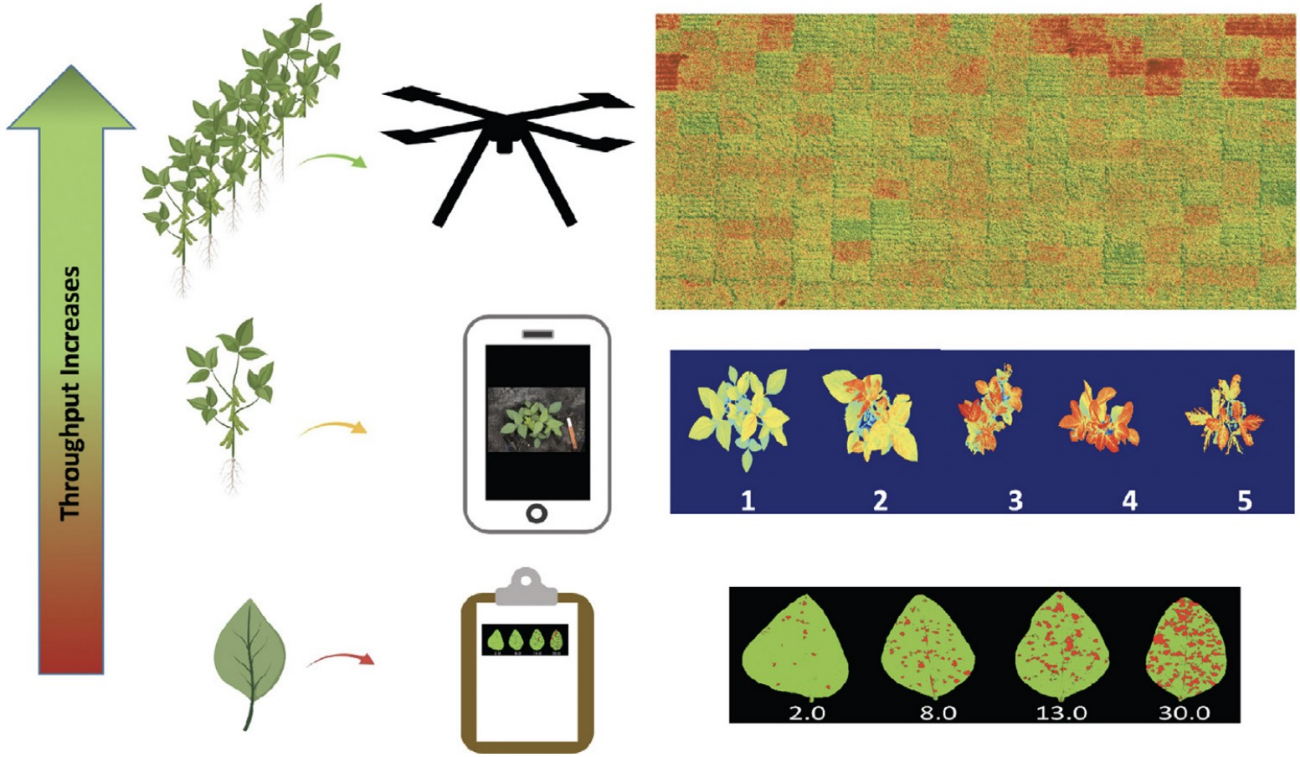


Plant and Weed Identifier Robot as an Agroecological Tool Using Artificial Neural Networks for Image Identification

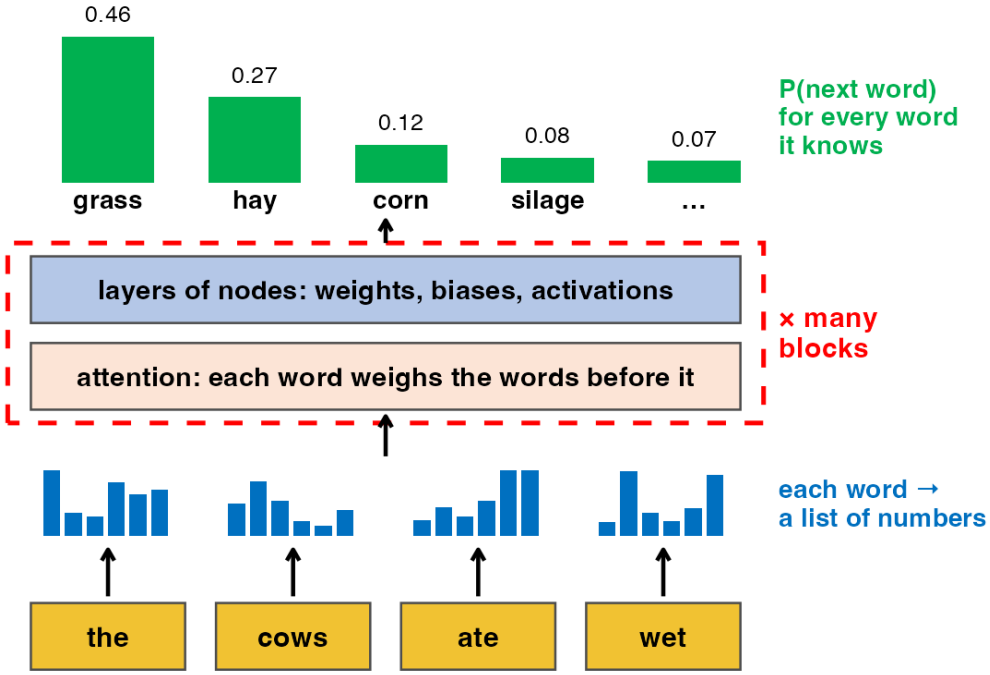
by Tavseef Mairaj Shah *,†, Durga Prasad Babu Nasika † and Ralf Otterpohl

Challenges and Opportunities in Machine-Augmented Plant Stress Phenotyping

Arti Singh,[1,*] Sarah Jones,[1] Baskar Ganapathysubramanian,[2] Soumik Sarkar,[2] Daren Mueller,[3] Kulbir Sandhu,[1] and Koushik Nagasubramanian[4]



Rounding off: a language model



Shared with our networks

- nodes: weights, biases, activations
- fitted downhill on −log-likelihood
- judged on text it never saw

New

- attention: which earlier words matter
- billions of numbers, huge amounts of text
- output: a probability for every word

Pick a word, add it to the text, run again: **one word at a time**. (Probabilities made up.)

1 Trees

2 One tree

3 Forests

4 Complex data

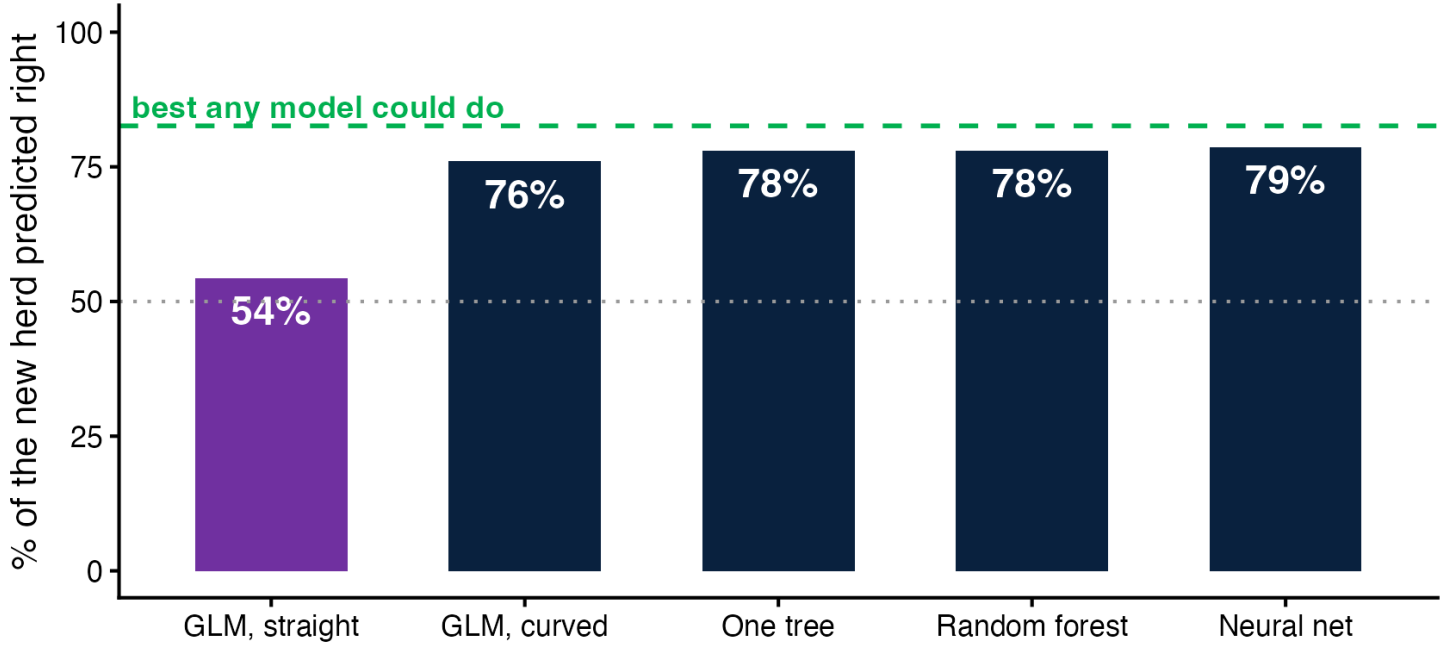
5 Neural nets

► **6 Which model?**



The cattle: a new herd of 300

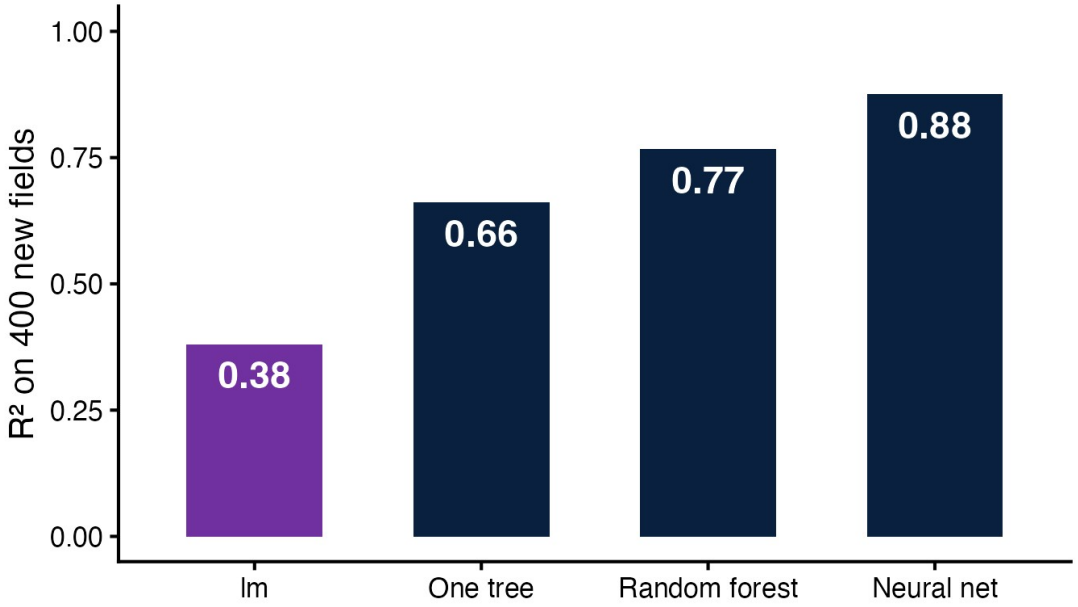
```
> mean((predict(m, new_herd, type = "response") > 0.5) == new_herd$healthy)
```



Anything flexible ≈ **78%**. The ceiling is the noise, not the model.

The wheat fields: 400 new fields

```
> r2(new_fields$yield, predict(m, new_fields))   # r2() helper in the live script
```



Many predictors, thresholds, interactions:

flexible models win big

Which model?

	lm / GLM	Random forest	Neural net
Shapes	ones you write	finds them	finds them
How to read it	coefficients, p-values	importance, partial dependence	hard: weights mean little
Data needed	little	moderate	lots
Choices to make	the shape (terms)	ntree, mtry (defaults work)	scaling, rep, hidden nodes

Want to **explain**? lm/GLM. Want to **predict**? Try them all, test on new data.

Live coding: forests and nets



`Week9_live_1_forests.R`

Cattle, then wheat fields

rpart() and randomForest()

OOB error, importance

partial dependence

neuralnet() and the comparison

After Thanksgiving

Mon Nov 30: structural equation models

MANOVA & transformations: self-study page on the website

PS 9 due Mon Nov 30, 1:10 PM