Interpretable ML for biodiversity

An introduction using species distribution models

Timothée Poisot Université de Montréal

November 13, 2024

MAIN GOALS

1. How do we produce a model?
2. How do we convey that it works?
3. How do we talk about how it makes predictions?

BUT WHY...

... think of SDM as ML problems? Because they are! We want to learn a predictive algorithm from data

... the focus on explainability? We cannot ask people to *trust* - we must *convince* and *explain*

WHAT WE WILL *NOT* DISCUSS

1. Image recognition
2. Sound recognition
3. Generative AI

LEARNING/TEACHING GOALS

- ML basics
  - cross-validation
  - hyper-parameters tuning
  - bagging and ensembles
- Pitfalls
  - data leakage
  - overfitting
- Explainable ML
  - partial responses
  - Shapley values

BUT WAIT!

- a similar example fully worked out usually takes me 21 hours of class time
- this is an overview
- don’t care about the output, care about the process!

§ 1

Problem statement

THE PROBLEM IN ECOLOGICAL TERMS

We have information about a species, taking the form of (lon, lat) for points where the species was observed

Using this information, we can extract a suite of environmental variables for the locations where the species was observed

We can do the same thing for locations where the species was not observed

Where could we observe this species?

THE PROBLEM IN ML TERMS

We have a series of labels $\mathbf{y}_n \in \mathbb{B}$, and features $\mathbf{X}_{m,n} \in \mathbb{R}$

We want to find an algorithm $f(\mathbf{x}_m) = \hat{y}$ that results in the distance between $\hat{y}$ and y being *small*

An algorithm that does this job well is generalizable (we can apply it on data it has not been trained on) and makes credible predictions

SETTING UP THE DATA FOR OUR EXAMPLE

We will use data on observations of *Turdus torquatus* in Switzerland, downloaded from the copy of the eBird dataset on GBIF

Two series of environmental layers

1. CHELSA2 BioClim variables (19)
2. EarthEnv land cover variables (12)

Now is *not* the time to make assumptions about which are relevant!

THE OBSERVATION DATA

PROBLEM (AND SOLUTION)

We want $\mathbf{y} \in \mathbb{B}$, and so far we are missing negative values

We generate pseudo-absences with the following rules:

1. Locations further away from a presence are more likely
2. Locations less than 6km away from a presence are ruled out
3. Pseudo-absences are twice as common as presences

THE (INFLATED) OBSERVATION DATA

§ 2

Training the model

A SIMPLE DECISION TREE

Decision trees *recursively* split observations by picking the best variable and value.

Given enough depth, they can overfit the training data (we'll get back to this).

We need an initial model to get started: what if we use *all the variables*?

We shouldn't use all the variables.

But! It is a good baseline. A good baseline is important.

Can we train the model?

More specifically – if we train the model, how well can we expect it to perform?

The way we answer this question is: in many parallel universes with slightly less data, is the model good?

NULL CLASSIFIERS

What if the model guessed based on chance only?

What is chance only?

50%, based on prevalence, or always the same answer

The null classifiers tell us what we need to beat in order to perform better than chance.

Model	MCC	PPV	NPV	DOR	Accuracy
No skill	-0.00	0.34	0.66	1.00	0.55
Coin flip	-0.32	0.34	0.34	0.26	0.34
+	0.00	0.34			0.34
-	0.00		0.66		0.66

In practice, the no-skill classifier is the most informative: what if we only know the positive class prevalence?

CROSS-VALIDATION STRATEGY

- k-fold cross-validation
- no testing data here

CROSS-VALIDATION RESULTS

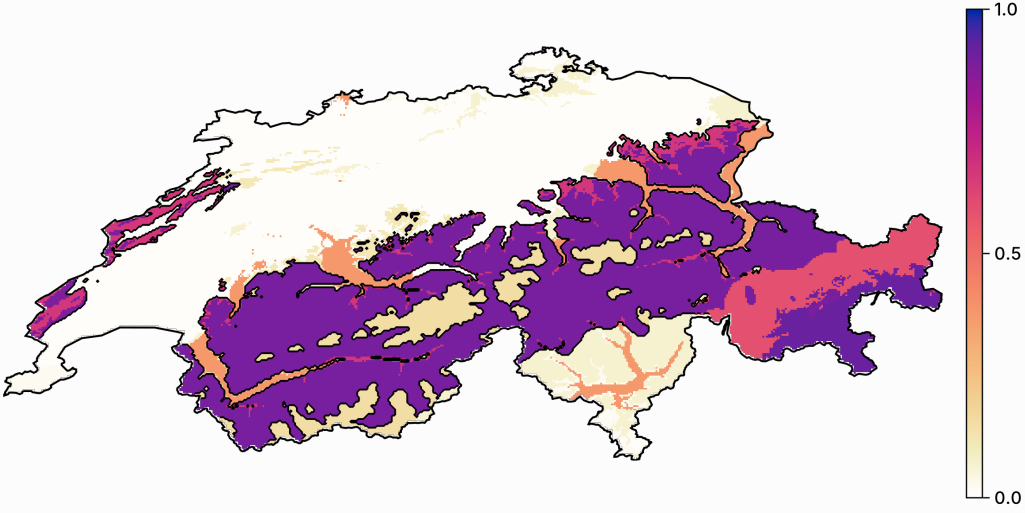
Model	MCC	PPV	NPV	DOR	Accuracy
No skill	-0.00	0.34	0.66	1.00	0.55
Dec. tree (val.)	0.80	0.83	0.96	210.06	0.91
Dec. tree (tr.)	0.84	0.86	0.97	202.00	0.93

WHAT TO DO IF THE MODEL IS TRAINABLE?

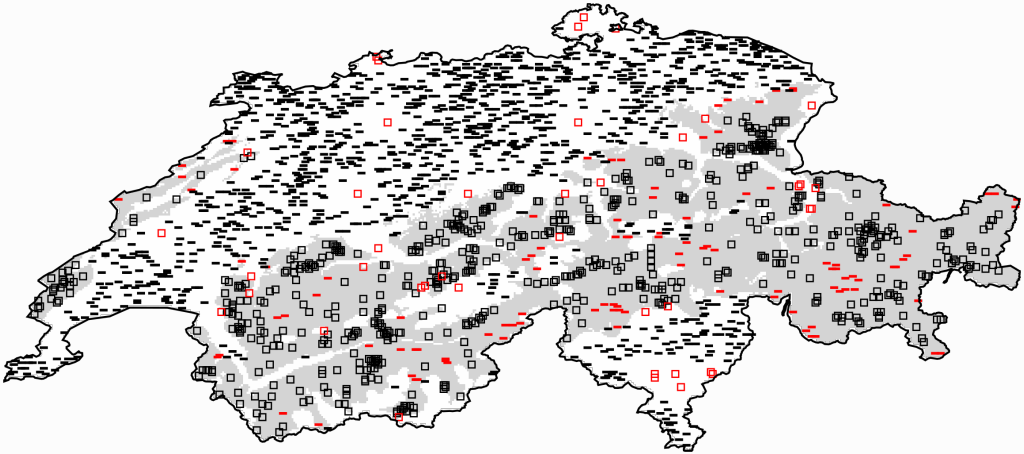
We train it!

This training is done using the *full* dataset - there is no need to cross-validate, we know what to expect based on previous steps.

INITIAL PREDICTION



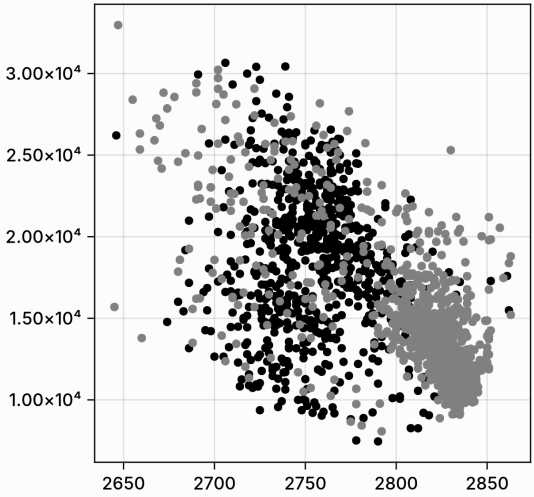
HOW IS THIS MODEL WRONG?

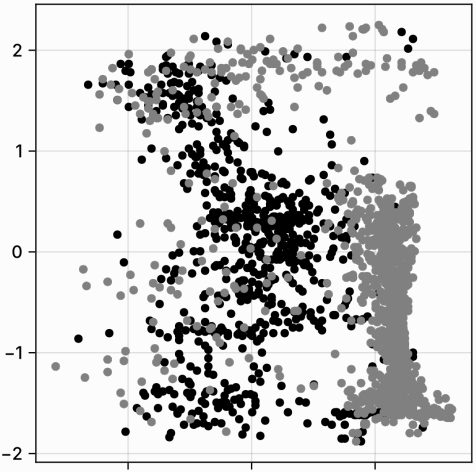


CAN WE IMPROVE ON THIS MODEL?

- variable selection
- data transformation (we use PCA here, but there are many other)
- hyper-parameters tuning

A NOTE ON PCA

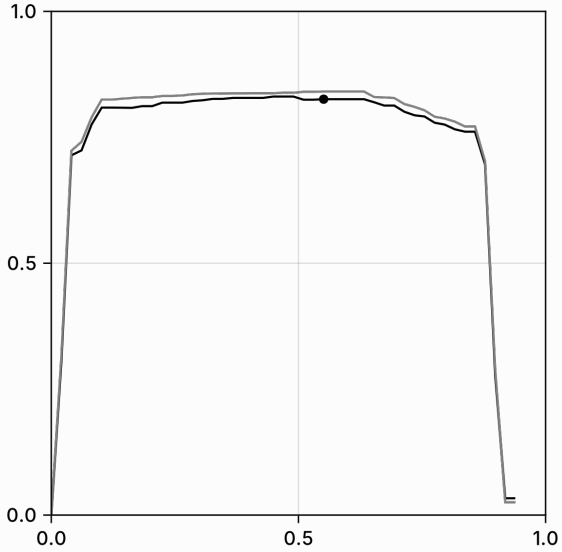

3.00×10⁴
2.50×10⁴
2.00×10⁴
1.50×10⁴
1.00×10⁴
2650
2700
2750
2800
2850


2
1
0
−1
−2
−2
0
2

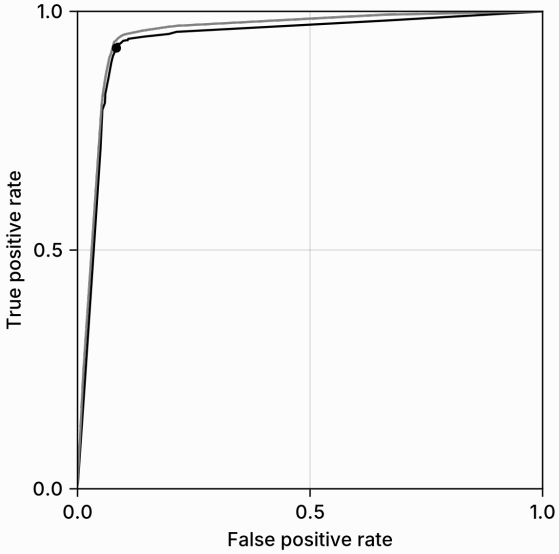
MOVING THRESHOLD CLASSIFICATION

- $P(+) > P(-)$
- This is the same thing as $P(+) > 0.5$
- Is it, though?

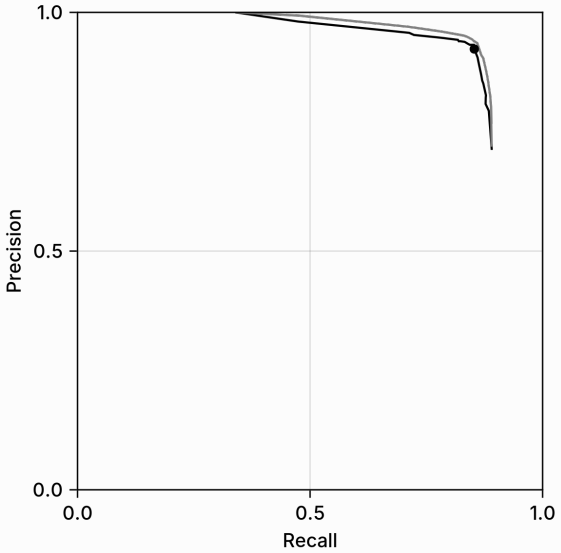
LEARNING CURVE FOR THE THRESHOLD



RECEIVER OPERATING CHARACTERISTIC



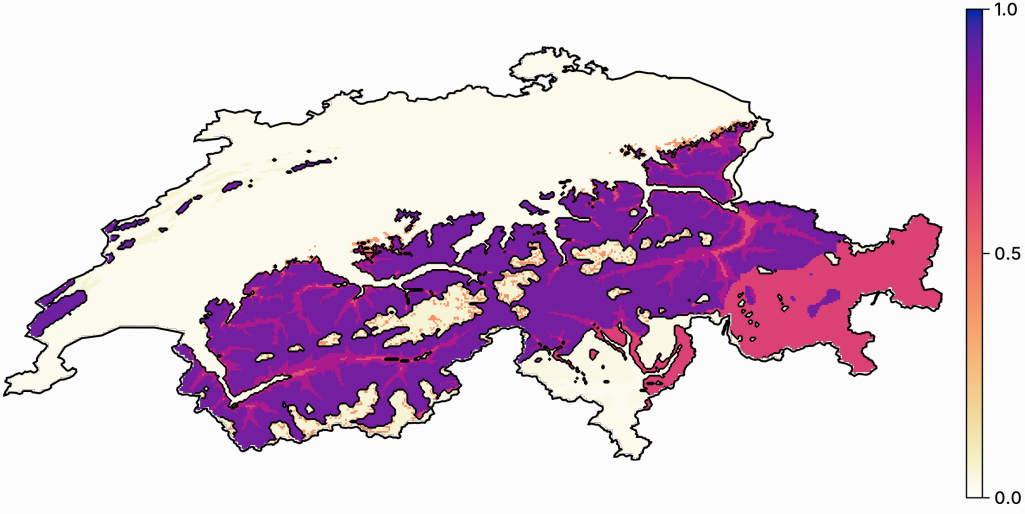
PRECISION-RECALL CURVE



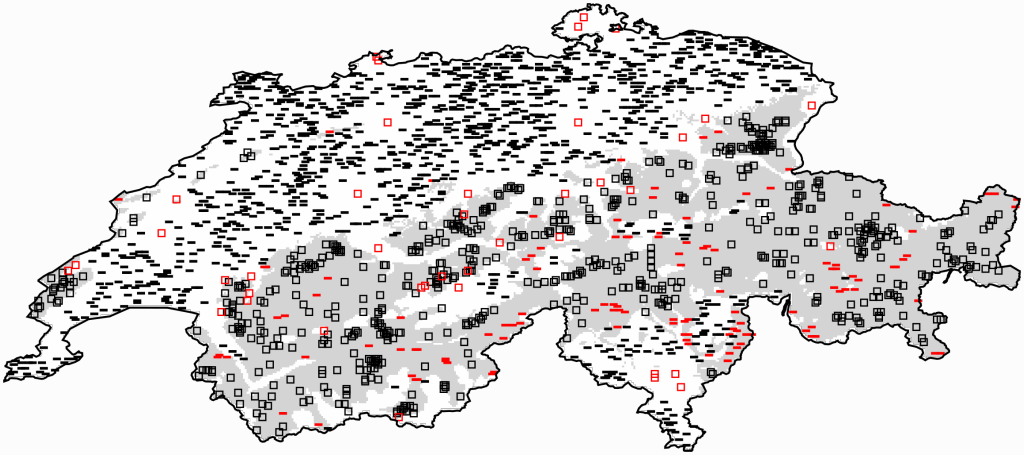
REVISITING THE MODEL PERFORMANCE

Model	MCC	PPV	NPV	DOR	Accuracy
No skill	-0.00	0.34	0.66	1.00	0.55
Dec. tree (val.)	0.80	0.83	0.96	210.06	0.91
Dec. tree (tr.)	0.84	0.86	0.97	202.00	0.93
Tuned tree (val.)	0.83	0.85	0.96	198.33	0.92
Tuned tree (tr.)	0.84	0.85	0.97	174.94	0.92

UPDATED PREDICTION

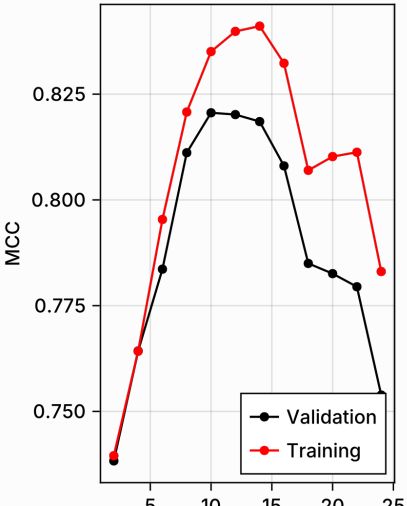


HOW IS THIS MODEL BETTER?



BUT WAIT!

Decision trees overfit: if we pick a maximum depth of 8 splits, how many nodes can we use?



§ 3

Ensemble models

LIMITS OF A SINGLE MODEL

- it’s a single model my dudes
- different subsets of the training data may have different signal
- do we need all the variables all the time?
- bias v. variance tradeoff
- fewer variables make it harder to overfit

BOOTSTRAPPING AND AGGREGATION

- bootstrap the training instances (32 samples for speed)
- randomly sample $\lceil \sqrt{n} \rceil$ variables

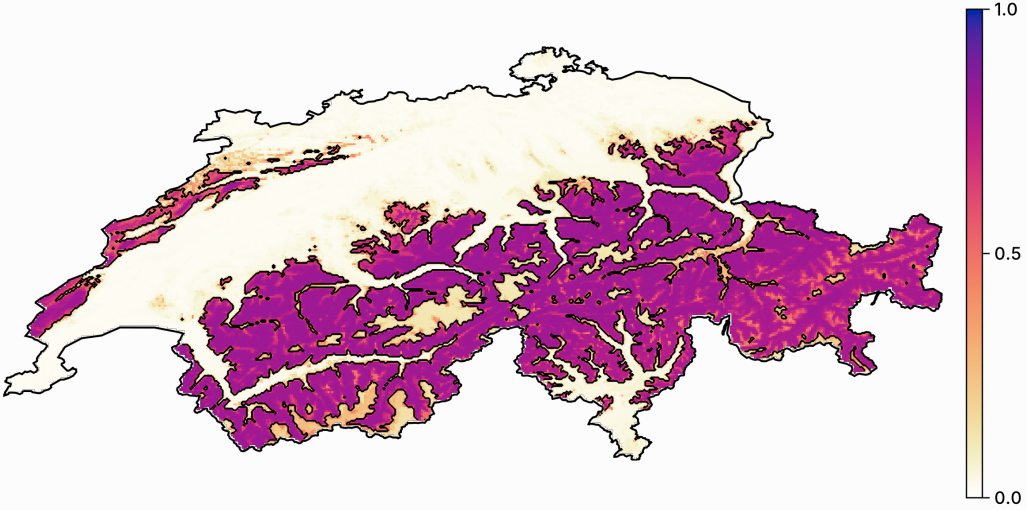
IS THIS WORTH IT?

Model	MCC	PPV	NPV	DOR	Accuracy
No skill	-0.00	0.34	0.66	1.00	0.55
Dec. tree (val.)	0.80	0.83	0.96	210.06	0.91
Dec. tree (tr.)	0.84	0.86	0.97	202.00	0.93
Tuned tree (val.)	0.83	0.85	0.96	198.33	0.92
Tuned tree (tr.)	0.84	0.85	0.97	174.94	0.92
Forest (val.)	0.77	0.79	0.96	111.07	0.89
Forest (tr.)	0.77	0.79	0.95	78.34	0.89

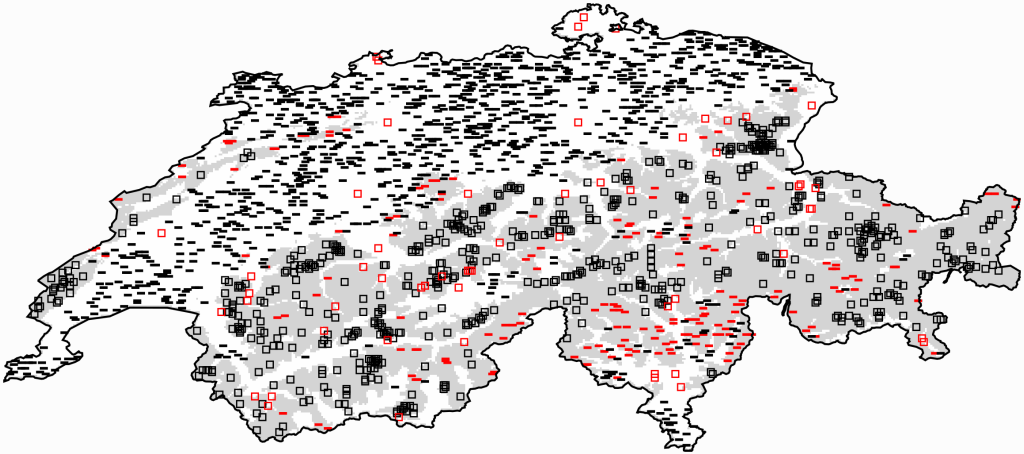
Short answer: no

Long answer: maybe? Let's talk it through!

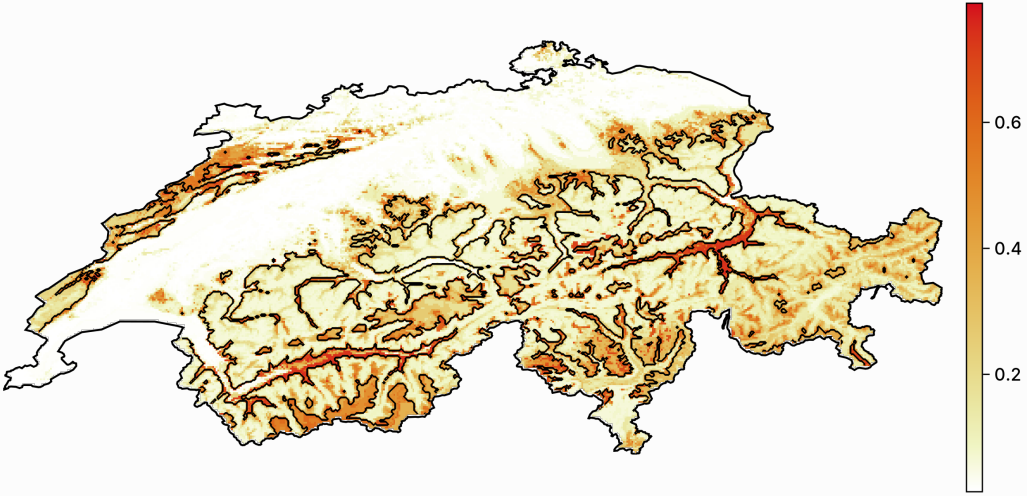
PREDICTION OF THE ROTATION FOREST



PREDICTION OF THE ROTATION FOREST



VARIATION BETWEEN PREDICTIONS



WHAT, EXACTLY, IS BOOTSTRAP TELLING US?

- what if we had a little less data (it's conceptually close to cross-validation!)
- uncertainty about locations, not predictions

Do we expect the model predictions to change at this location when we add more training data?

VARIABLE IMPORTANCE

Layer	Variable	Import.
10	BIO10	0.28209
5	BIO5	0.253606
6	BIO6	0.1741
13	BIO13	0.0832986
15	BIO15	0.0797567
26	Cultivated and Managed Vegetation	0.0793417
12	BIO12	0.044542
29	Snow/Ice	0.0032655

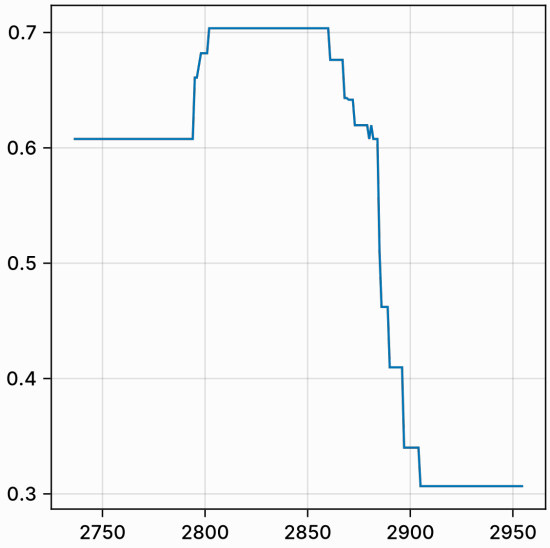
§ 4

But why?

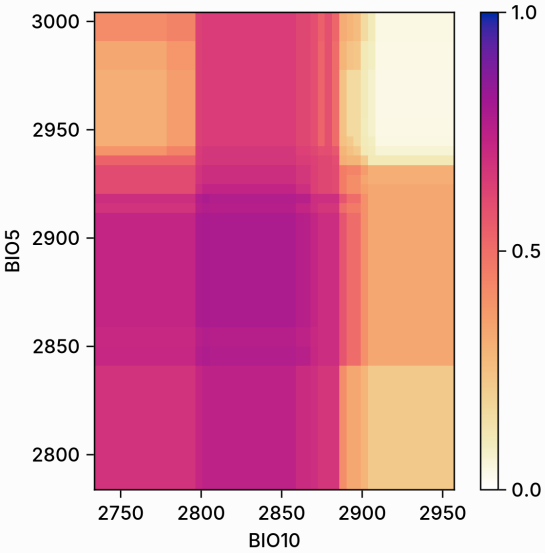
If we assume that all the variables except one take their average value, what is the prediction associated to the value that is unchanged?

Equivalent to a mean-field approximation

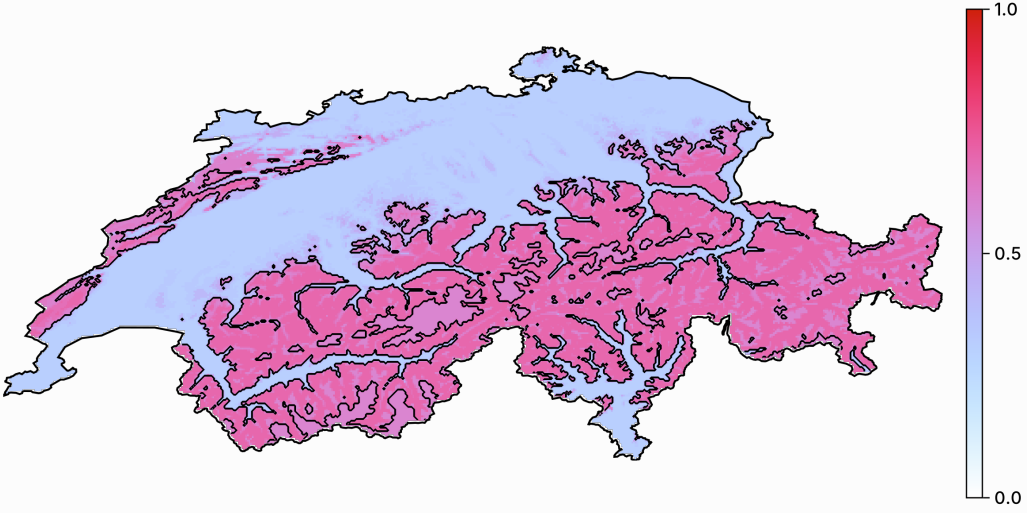
EXAMPLE WITH TEMPERATURE



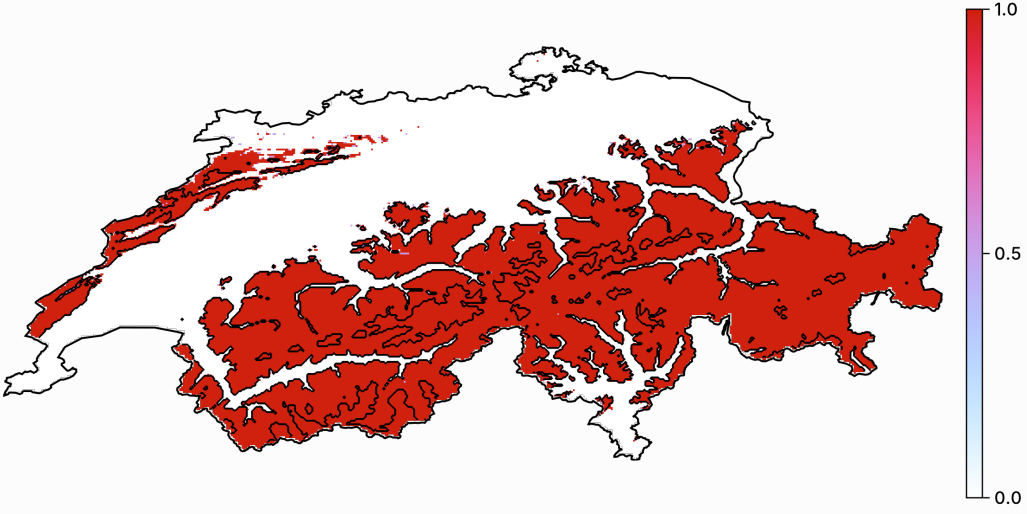
EXAMPLE WITH TWO VARIABLES



SPATIALIZED PARTIAL RESPONSE PLOT



SPATIALIZED PARTIAL RESPONSE (BINARY OUTCOME)



INFLATED RESPONSE CURVES

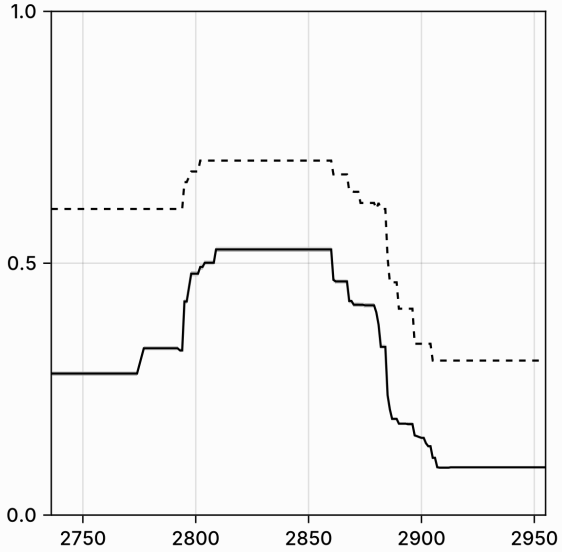
Averaging the variables is masking a lot of variability!

Alternative solution:

1. Generate a grid for all the variables
2. For all combinations in this grid, use it as the stand-in for the variables to replace

In practice: Monte-Carlo on a reasonable number of samples.

EXAMPLE

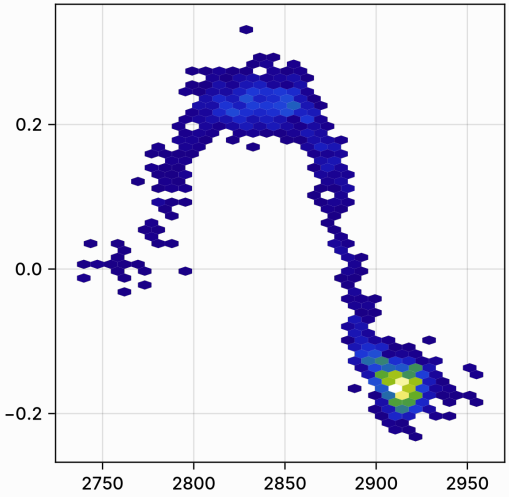


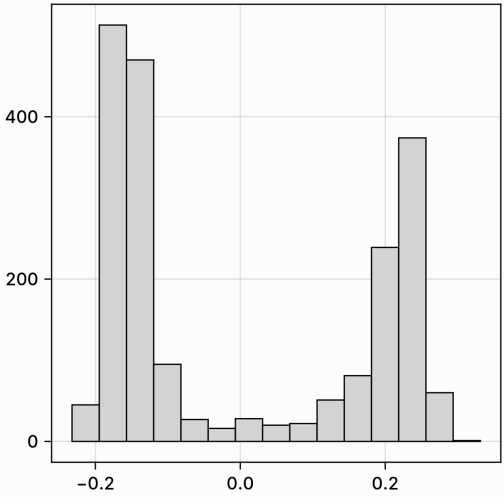
LIMITATIONS

- partial responses can only generate model-level information
- they break the structure of values for all predictors at the scale of a single observation
- their interpretation is unclear

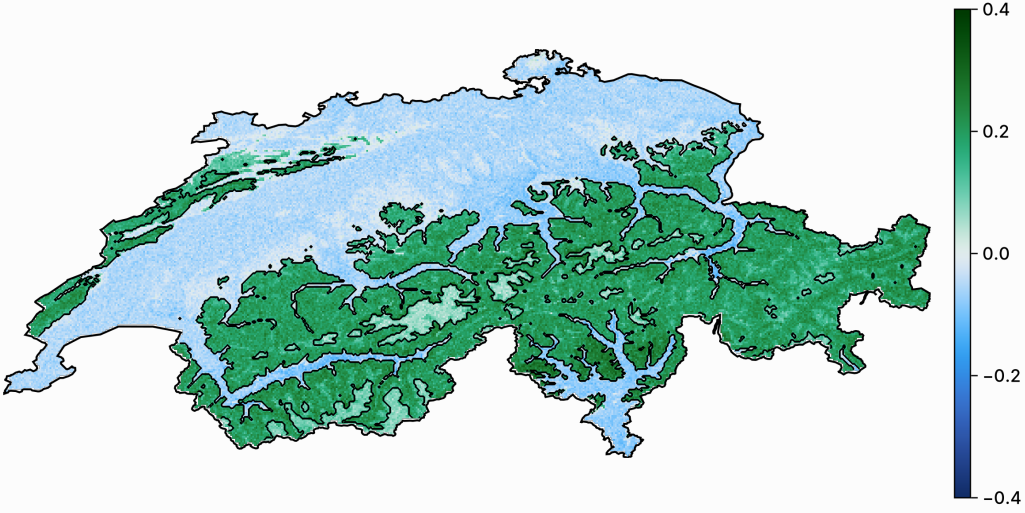
- how much is the average prediction modified by a specific variable having a specific value?
- it's based on game theory (but it's not *actually* game theory)
- many highly desirable properties!

RESPONSE CURVES REVISITED





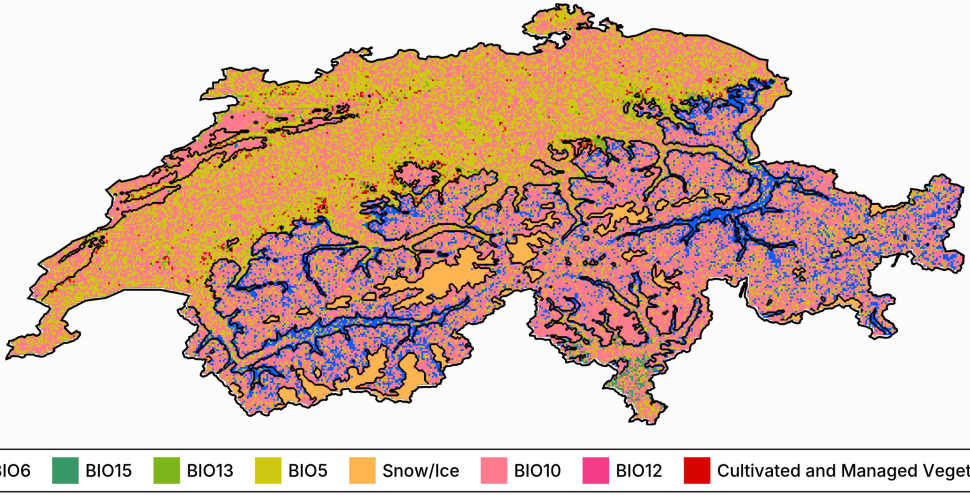
ON A MAP



VARIABLE IMPORTANCE REVISITED

Layer	Variable	Import.	Shap. imp.
10	BIO10	0.28209	0.310325
5	BIO5	0.253606	0.252689
6	BIO6	0.1741	0.215058
13	BIO13	0.0832986	0.0639503
26	Cultivated and Managed Vegetation	0.0793417	0.0634541
12	BIO12	0.044542	0.0370903
15	BIO15	0.0797567	0.0315063
29	Snow/Ice	0.0032655	0.025927

MOST IMPORTANT PREDICTOR



§ 5

Summary

SDMS ARE (APPLIED) MACHINE LEARNING

- models we can train
- parameters can (should!) be tuned automatically
- we can use tools from explainable ML to give more clarity