Spatial Statistics

Session 3

Madlene Nussbaum

m.nussbaum@uu.nl

Department of Physical Geography, Utrecht University

15 Oct 2024

Machine learning for spatial prediction

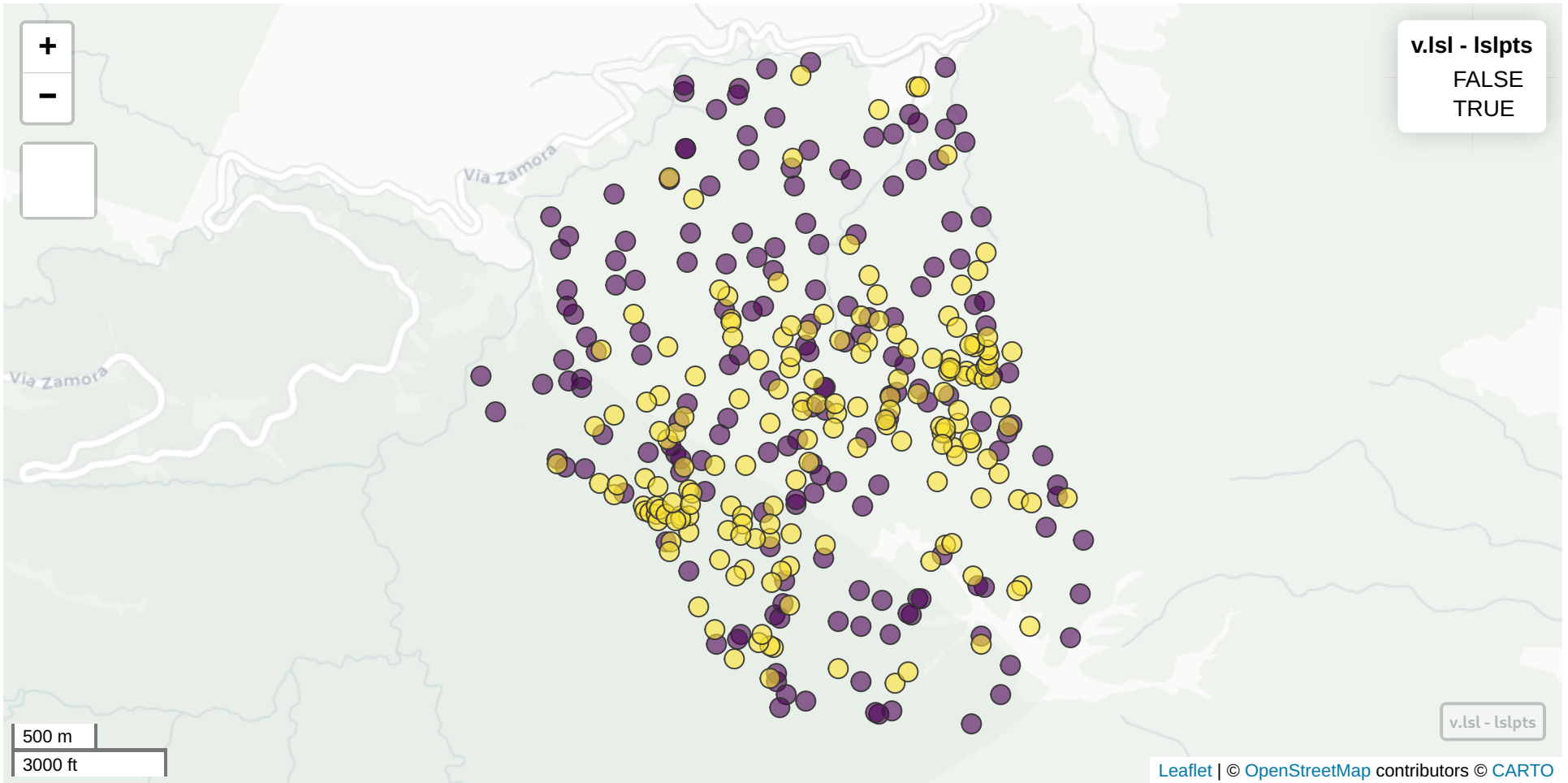
Overview this session

- Machine learning for spatial prediction
- Intro to random forest
- Model spatial location with machine learning
- Model selection
- Model interpretation
- Overview machine learning beyond random forest
- Challenges of machine learning for spatial analysis

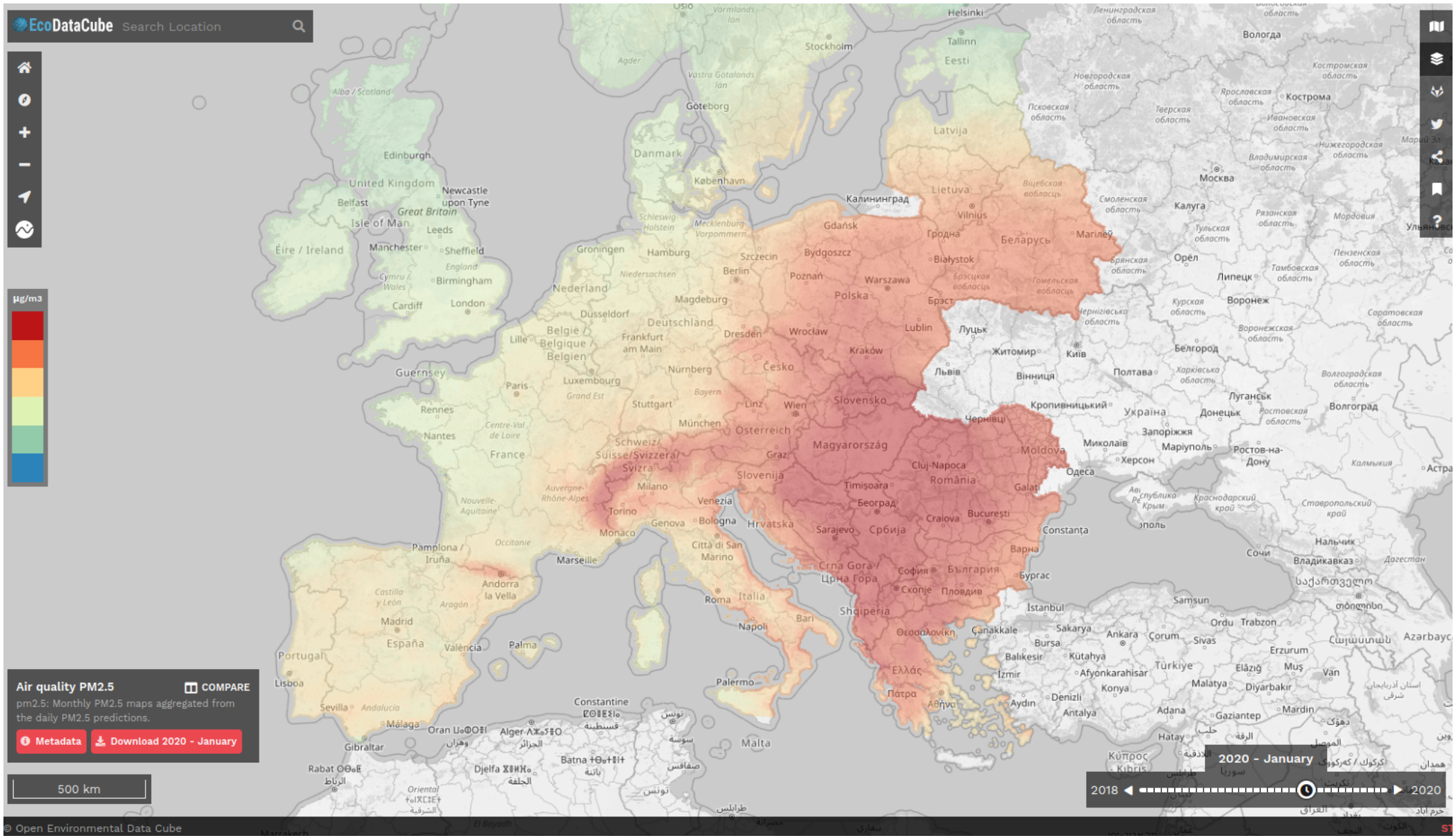
1 Geostatistical data analysis today

1.1 Spatial data analysis - examples

Binary responses – Landslide occurrence (lab 1)

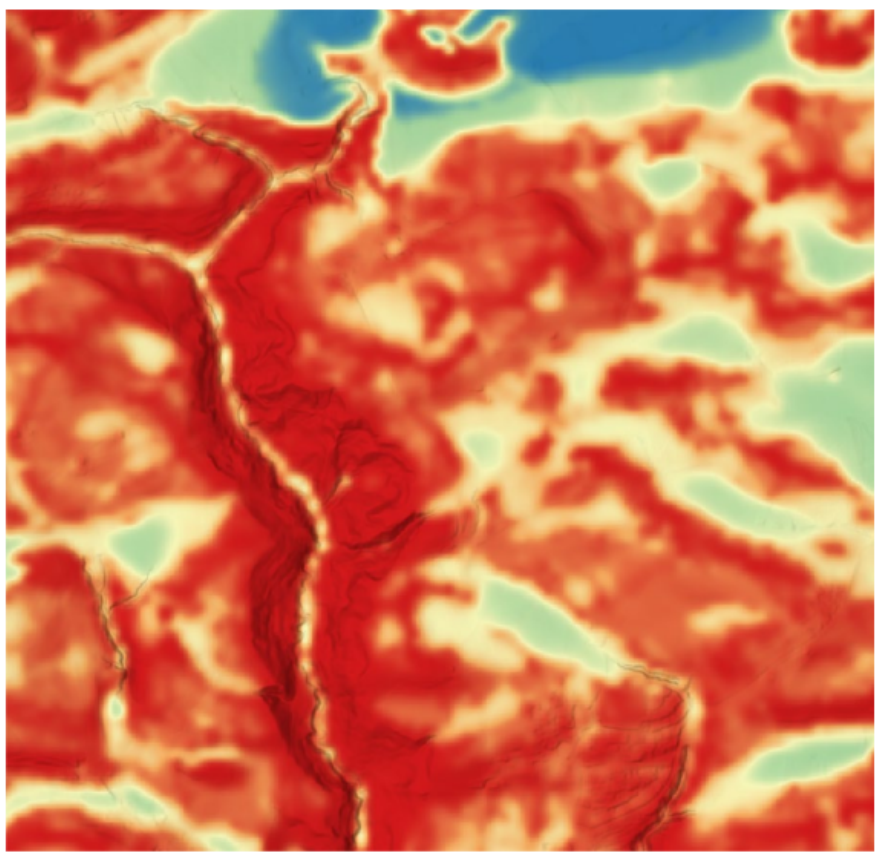


European wide air pollution mapping

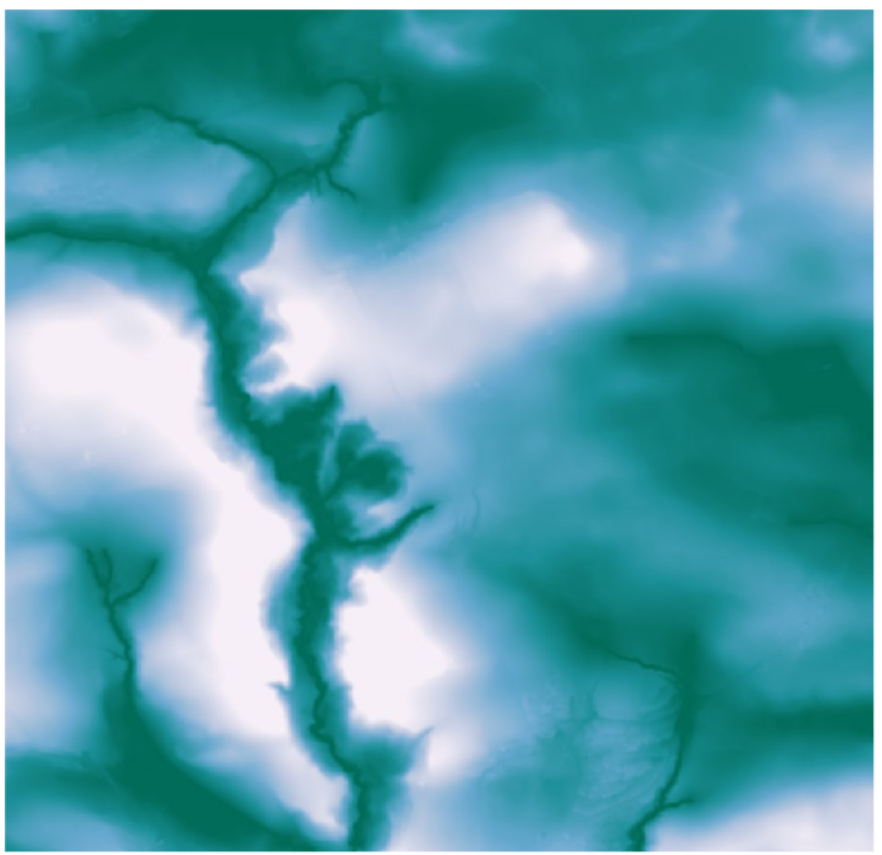


Feature engineering - algorithms

Erosionsakkumulation
MRVBF



Höhe über Gewässer
berechnet mit Gewässernetz
swissTLM3d



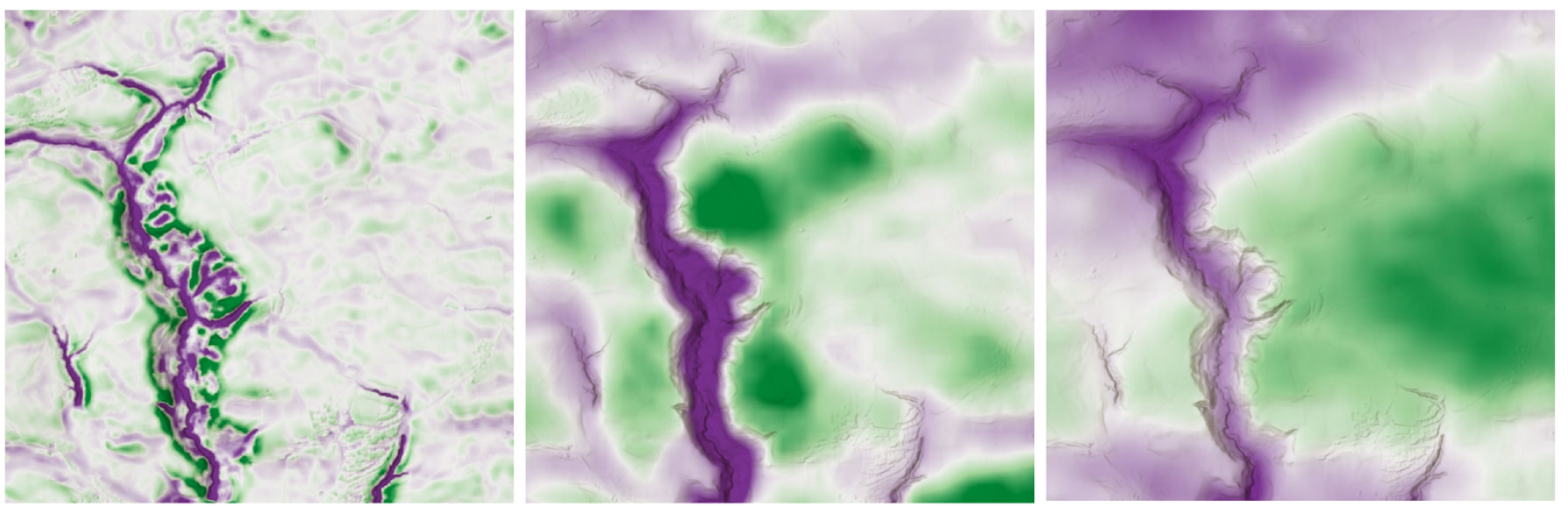
Feature engineering - settings

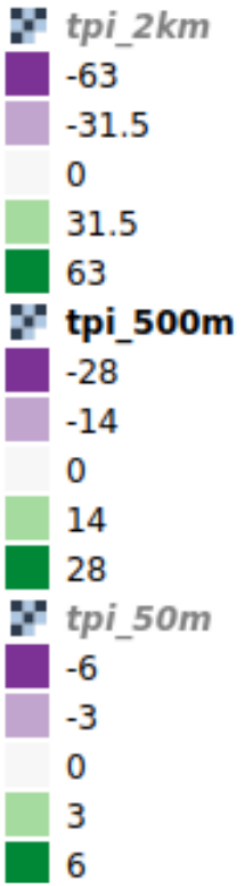
Topographic Position Index (TPI)

TPI Radius 50 m

TPI Radius 500 m

TPI Radius 2 km



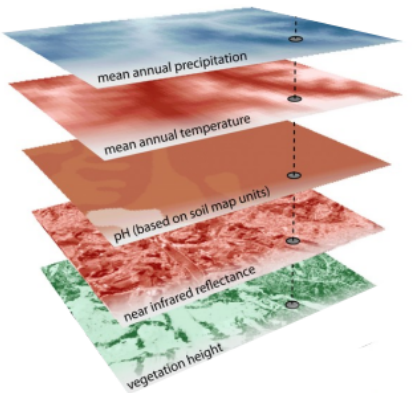


Multi-attribute soil mapping

texture
density
gravel
soil depth
drainage
pH, ECEC
SOC

300-1400
locations with soil properties in

2–4 soil depth
3 study areas



300-500
environmental covariates

48
statistical models

1.2 Drawbacks of classical geostatistical (kriging) approaches

1. What “properties” does a spatial prediction method need to have to handle today’s spatial data problems?
2. What are the challenges with classical geostatistical approaches regarding these “properties”?

2 Introduction to random forest

Why random forest here?

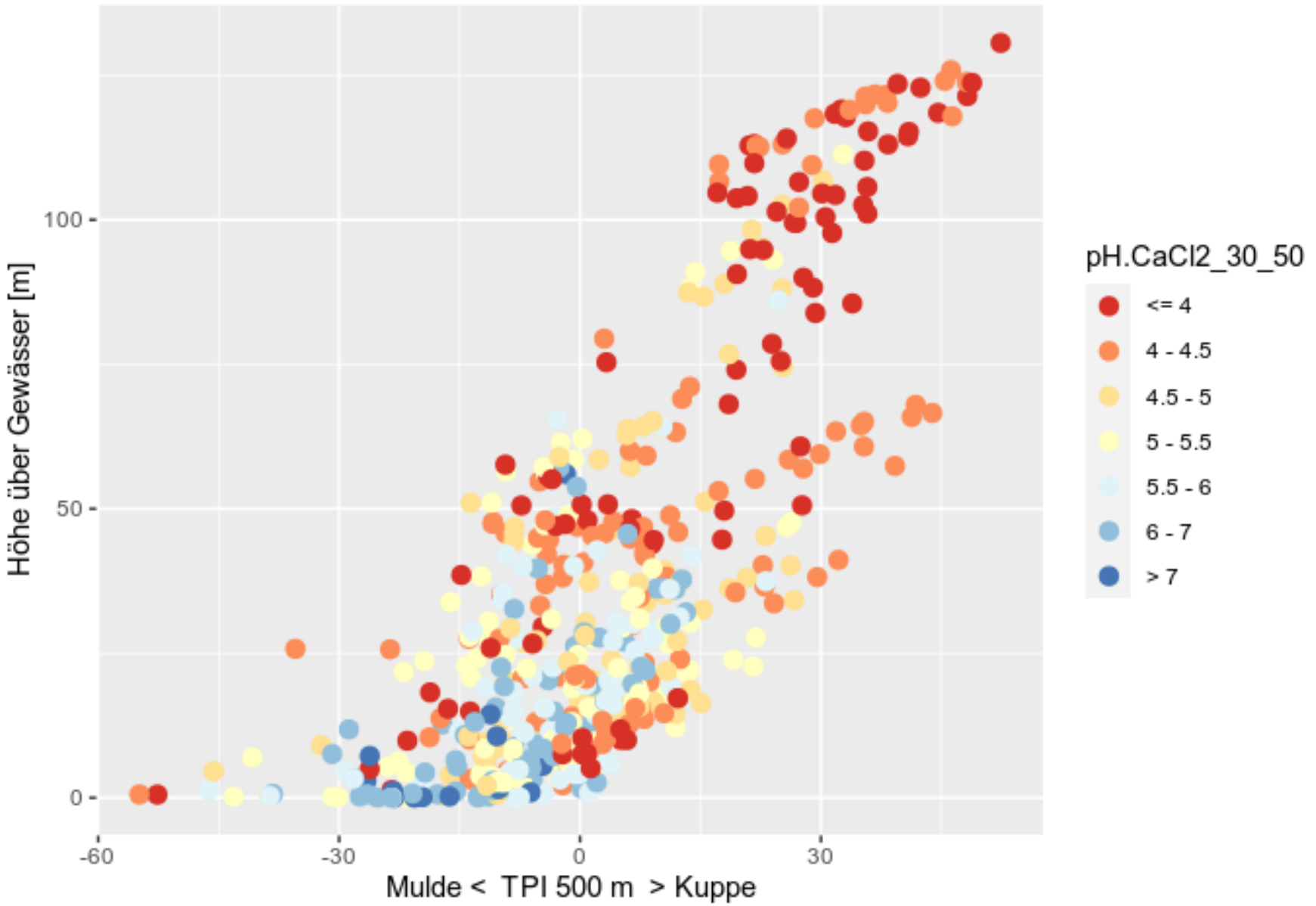
- Quite fast model fitting and prediction
- Models interaction in the data, models non-linear relationships
- Very good predictive power expected, therefore often used for spatial prediction
- No extrapolation over value range of the response
- Prediction uncertainty for continuous responses (at point location)

2.1 Random forest - Overview

- **Ensemble** machine learning method
- Classification and regression trees (CART) as base element
- **Bagging** (Bootstrap aggregation)

- Large number of trees are fully grown
- Trees are decorrelated with
  - Resampling of original dataset with replacement
  - Only a random subset of covariates are tested to split tree nodes
- Prediction is the average of all trees (continouous response) or the majority vote (binary or multinomial response).

Classification and regression trees – binary splitting

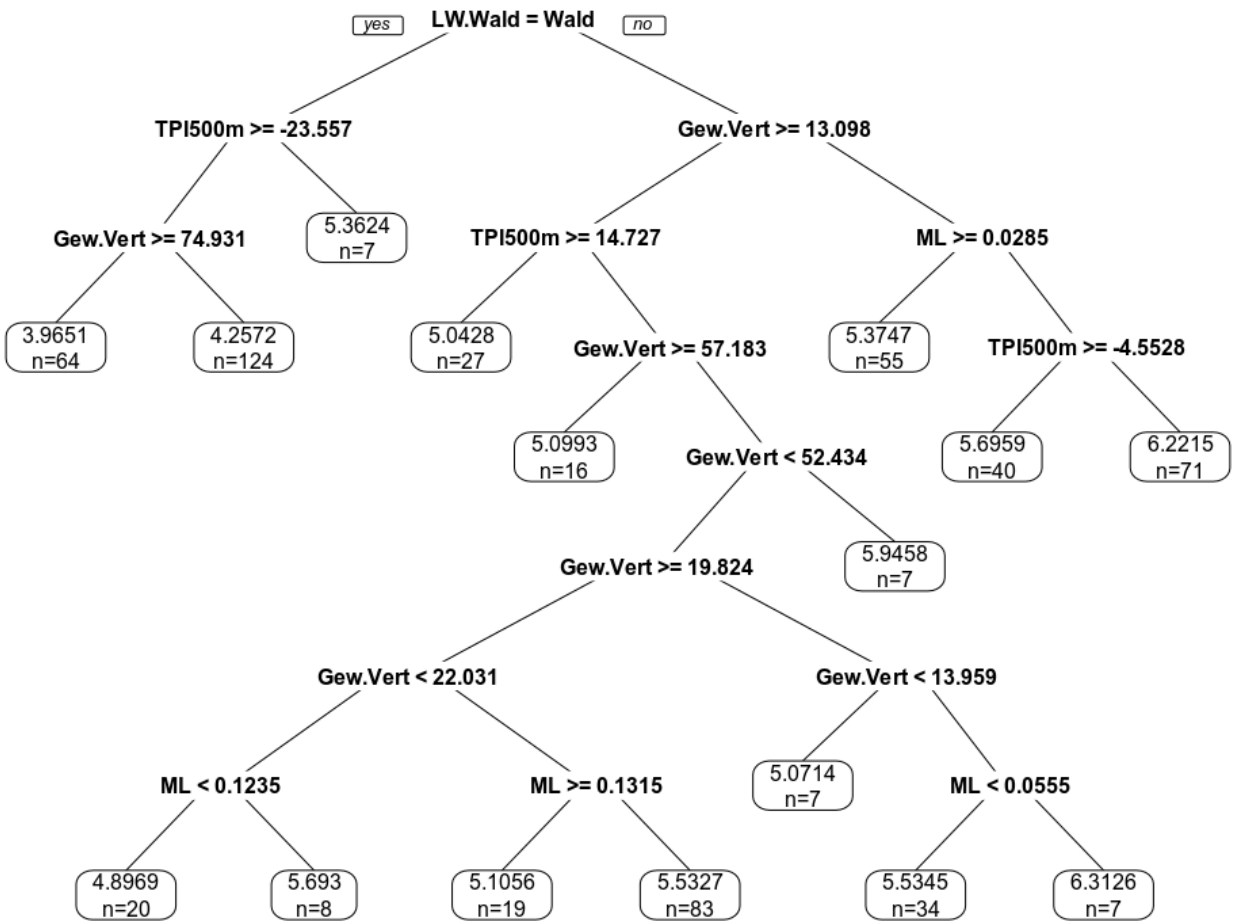


Classification and regression trees – tree

Response: soil pH

Covariates: forest/no-forest, ML: molasse geological unit, topographic index, vertical distance above river.

Value in box: mean pH of all locations falling in this tree node.



Classification and regression trees – formally

Technique: Recursive binary splitting

Consider all covariates $X_1, \ldots, X_p$, and all possible values s for splitting for each of the covariates, and then choose a covariate X_j and splitpoint s such that the resulting tree has the lowest residual sum of squares.

For any j and s a half-plane is defined by

$$R_1(j, s) = \{X | X_j < s\} \quad \text{and} \quad R_2(j, s) = \{X | X_j \geq s\}$$

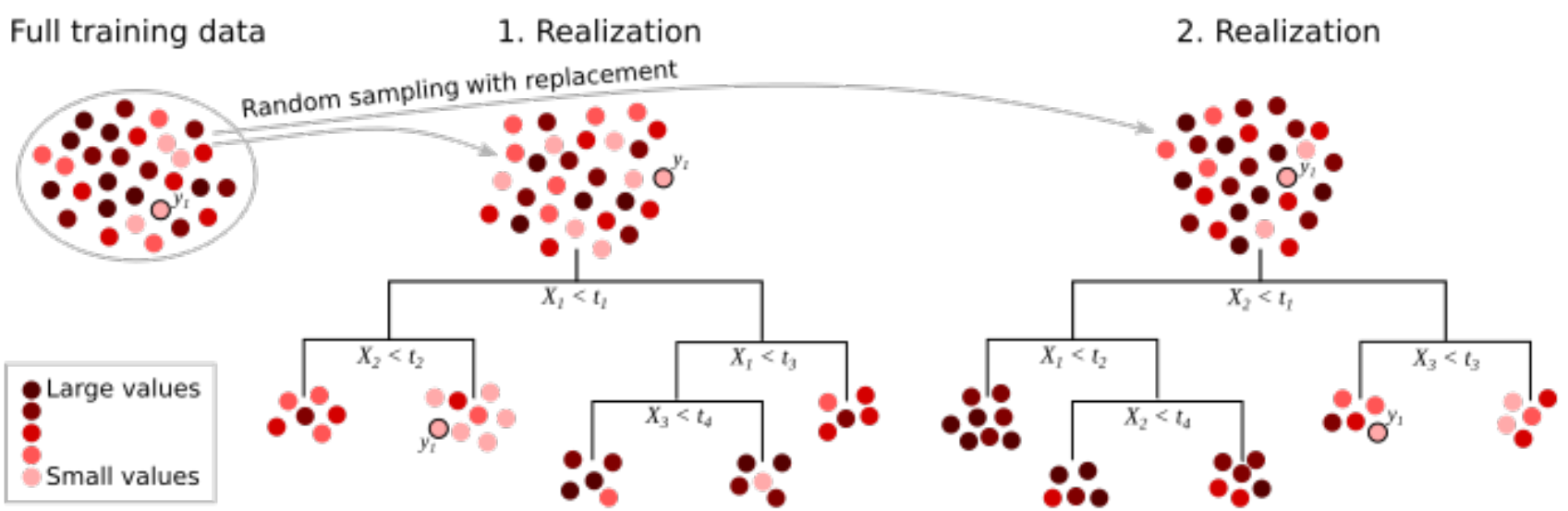
where j and s are chosen to minimize

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \overline{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \overline{y}_{R_2})^2$$

with $\overline{y}_{R_1}$ and $\overline{y}_{R_2}$ being the mean of the observations values in tree node $R_1(j, s)$ and $R_2(j, s)$, respectively.

Random forest algorithm

1. Resample dataset (with replacement)
2. Take a random sample of covariates of size `mtry`
3. Test all selected covariates: find optimal covariate value to split data into 2 portions
4. Chose covariate with lowest error (e.g. MSE) and split data
5. Continue to split the data with (3)-(4) until you are left with a small number of data points (`min.node.size`) in each leaf of the tree
6. Repeat (1)-(5) `ntree` times



Random forest tuning

Main tuning parameters:

- `mtry`: number of randomly selected covariates to test at each split
- `ntree`: number of trees (mostly not sensitive, if large enough)
- `min.node.size`: size of remaining dataset in tree leaf, when it stops to split (mostly not sensitive, if small enough)

Other options:

- `splitrule`: loss function to split data
- `max.depth`: limit tree depth (interaction depth), otherwise fully grown
- `case.weights`: increase/decrease weights of to be sampled for a tree (e.g. because of data quality)
- … many more (in general not changing the big picture)

Random forest – Default split rules

Regression – Mean squared error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}\,(x_i))^2$$

with y_i being the observed value, $\hat{f}\,(x_i)$ the prediction that prediction function $\hat{f}$ gives for the ith observation and n the total number of observations. For random forest, $\hat{f}$ is the mean prediction of all fitted regression trees.

Classification – Gini index

$$G = \sum_{k=1}^{K}\hat{p}_{mk}(1 - \hat{p}_{mk})$$

with $\hat{p}_{mk}$ being the proportion of training observations in the mth tree leaf from the kth class (response category), across all K classes.

Random forest – Compute predictions

CART

For each location, decide in which tree end node it falls ("send" each location with its covariate values down the trees). Take the mean (regression) or proportion/majority class (classification) of the observations within this end node.

Random forest

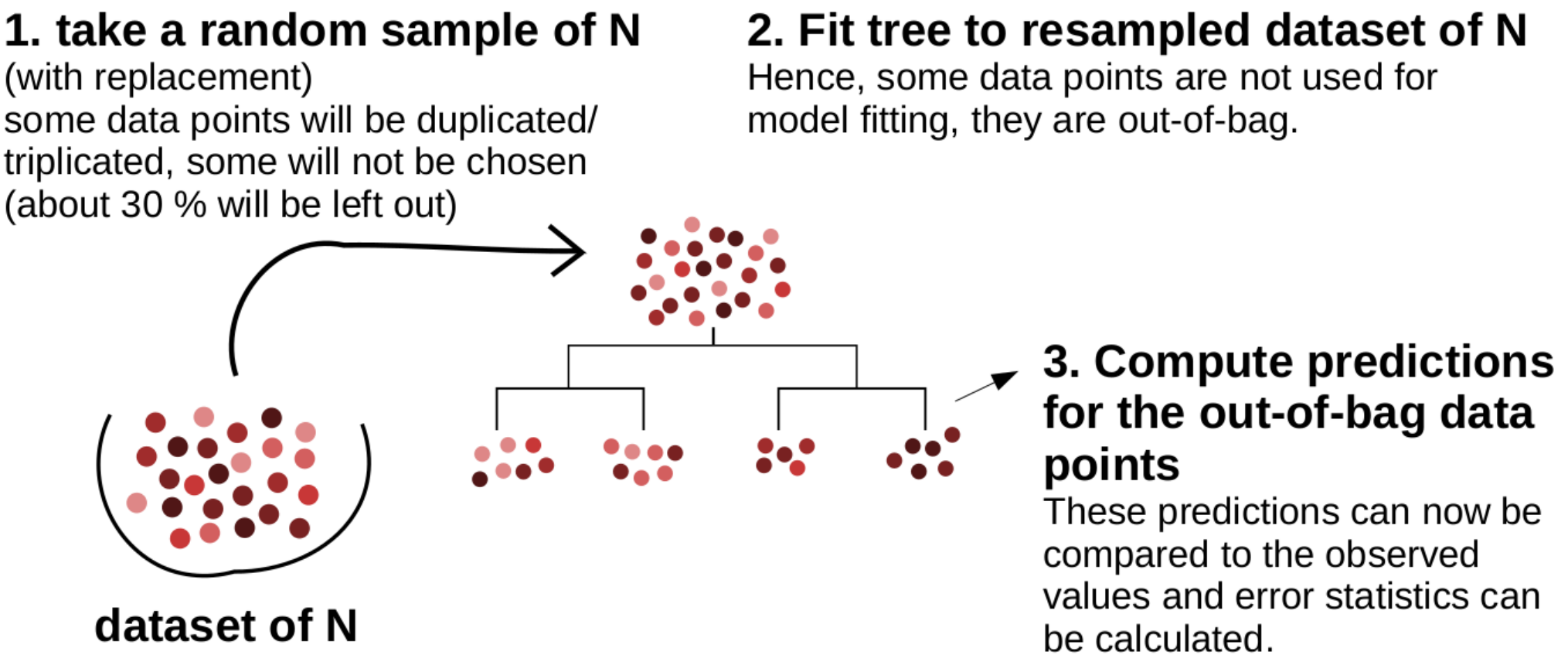
With $\hat{f}^1(x), \hat{f}^2(x), \ldots, \hat{f}^B(x)$ being predictions of one single CART, random forest predictions are averaged over B different bootstraped (bagged) samples b used for the training of $\hat{f}^b(x)$

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x).$$

For regression, this results in a weighted mean of all the observations. For classification, we obtain proportion of classes "voting" for an categorical outcome.

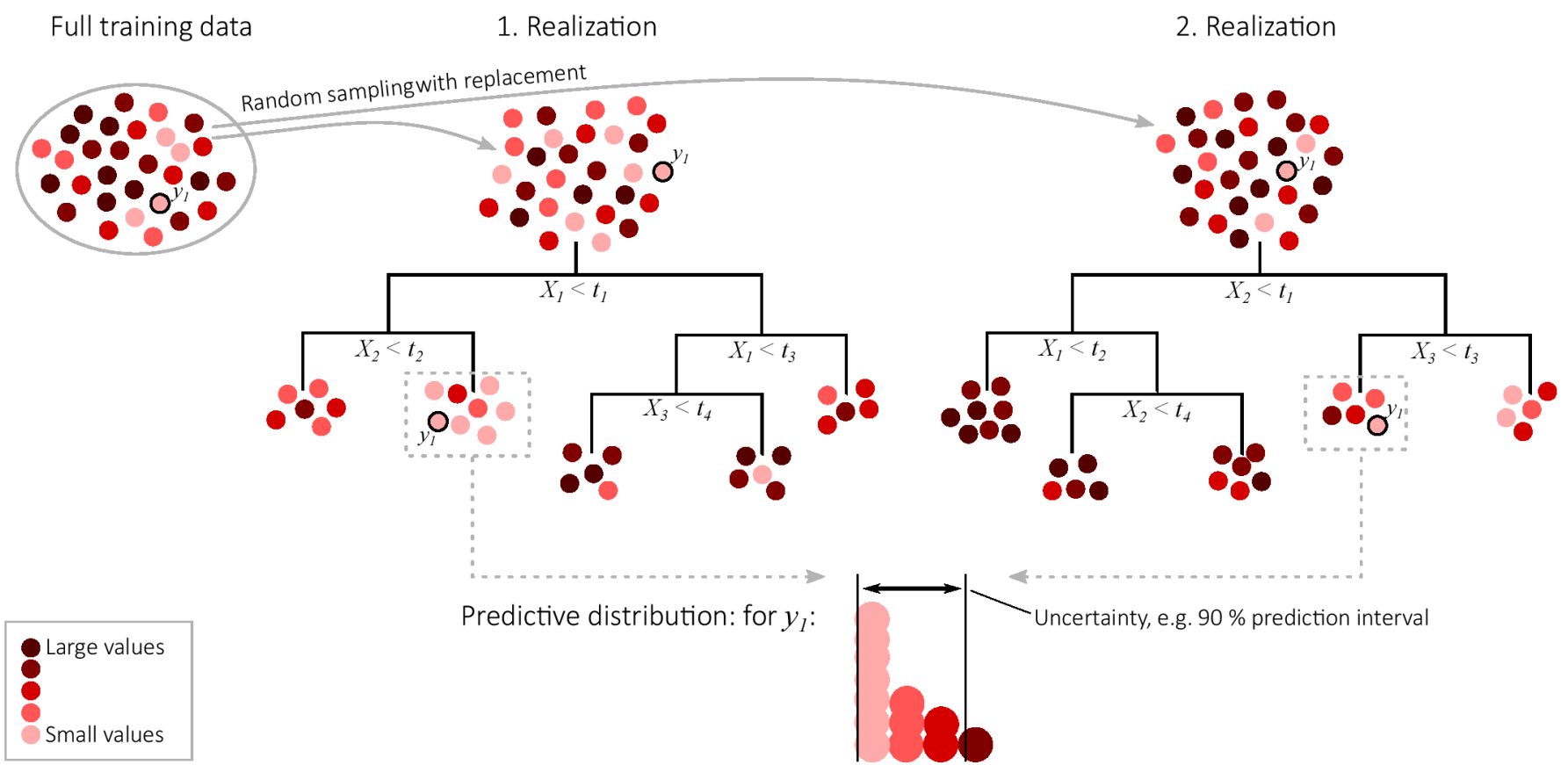
Random forest – Out-of-bag predictions

- Due to resampling with replacement, ~1/3 of observations is not used for one single tree. They are “out-of-bag” (OOB).



2.2 Quantile regression forest – Uncertainty

- Keep all observations in the final tree leaves
- Get distribution D from all observations that were in tree leaf with observation y_i
- Compute required quantities from D like 90 % prediction intervals or standard errors



3 Spatial auto-correlation in ML models

3.1 Apply geostatistics on residuals

Often termed “regression kriging”, even if combined with machine learning. Two step approach:

1. Fit non-spatial model, compute its residuals.
2. Fit ordinary kriging to residuals.

Predictions are a sum of model outputs from (1) and (2).

Non-spatial model can also be a linear regression (see e.g. analysis of Wolfcamp data).

Difficulty:

- quantification of prediction uncertainty
- some challenges with kriging are re-introduced

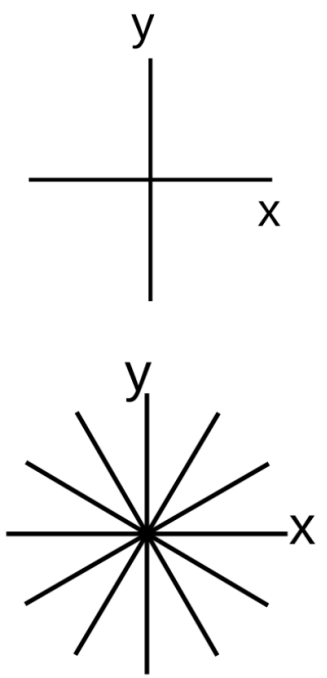
3.2 Spatial coordinates as covariates

Add x- and y-coordinate axis as additional covariates.

Allows random forest to partition the study area into smaller sections and fit local models.

Todel trend not only North-South and East-West, rotate coordinate axis by α

(in figure: 30° and 60°)



3.3 Relative spatial distances as covariates

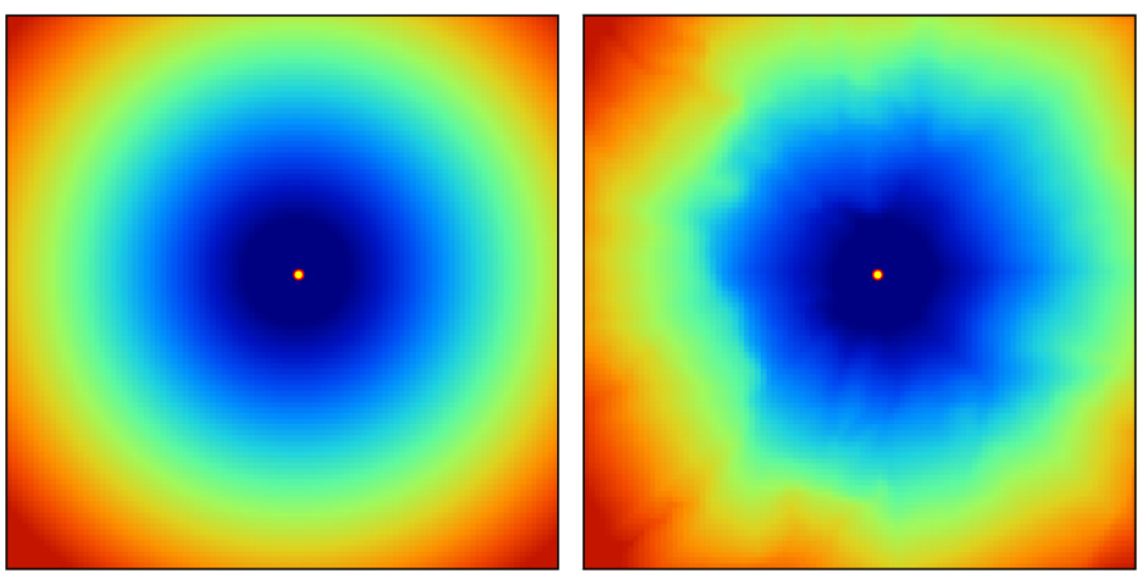
An additional set of covariates is defined based on the relative distance to each observation:

$$X_G = \{d_{p_1}, d_{p_1}, \ldots, d_{p_n}\}$$

where d_{p_i} is the euclidean distance (or any other more complex proximity distance) to the observed location p_i and n is the total number of training observations.

For each n observations one covariate is added (R packages `GSIF`, `spatialRF`).

Example covariate layer for one observation by euclidean distance (left) or travel time distance (right).



3.4 Spatial neighbors as covariates

Include observed values from n nearest locations and distances to these locations.

For each n nearest neighbor 2 colums are added as covariates:

- value at nth nearest observation
- distance corresponding to the location contributing the observed value

n is a tuning parameter.

R package `RFSI`

3.5 Smooth surface of coordinates

Works for Generalized Additive Models (GAM), i.e. fitted by boosting algorithm (R packages `mgcv`, `mboost`, `geoGAM`).

- Spatial auto-correlation can be modeled by including a “smooth spatial surface”
- Non-stationary covariate effects by interactions with surface (tensor splines)
- Difficulty: choice of degrees of freedom for splines surfaces.

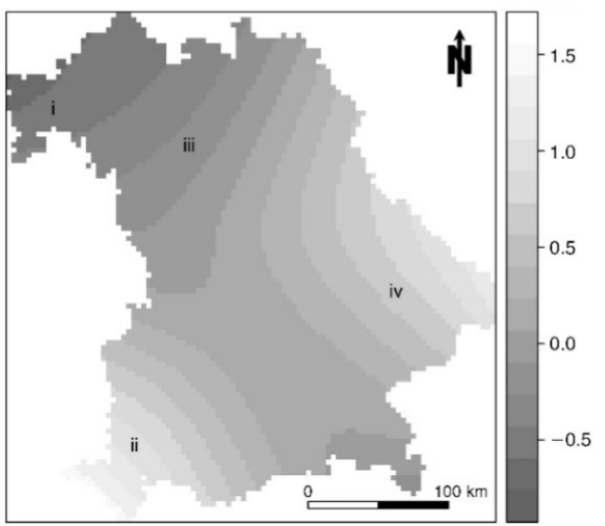


FIG. 6. Spatial difference in Red Kite breeding between 1979–1983 and 1996–1999 for model (add/vary). The breeding probabilities in the northwestern part decreased, while the southwestern part goes with increased breeding probabilities. For the four selected areas [(i) Unterfranken, (ii) Schwaben, (iii) Mittelfranken, and (iv) Niederbayern], the variability of the estimated spatial difference is shown in Fig. 7. Spatial differences can be interpreted as difference in log-odds ratios.

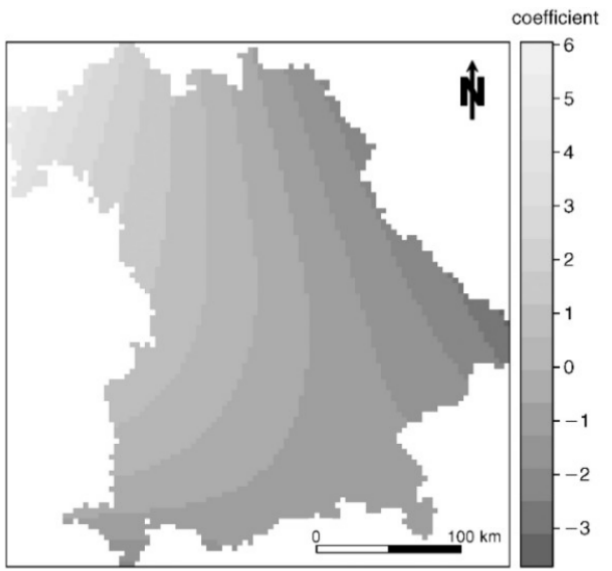


FIG. 8. Spatially varying coefficients for altitude in Red Kite breeding model (add/vary); here altitude was standardized to the unit interval. Altitude has a positive effect in the western and northwestern part, while its effect is zero or even negative in the rest of Bavaria.

Hothorn et al. 2011

4 Select relevant covariates

4.1 Model selection

Model building

Select structure of the model, i.e. relevant covariates and relevant response-covariate relationships.

Model selection

Select relevant covariates, drop non-relevant or correlated covariates.

Model selection – reasoning

Model selection, why:

- Decrease computing time (e.g. for prediction).
- More insight what is important. Improve future covariate preparation.
- Large number of strongly correlated/nearly identical covariates might lead to overvitting.

Why not:

- Potential loss of predictive performance.
- Chosen ML algorithm should be able to deal with large number of correlated covariates.
- Correlated covariates might improve predictions locally, at tail of distributions.
- Control overfitting by method (e.g. bagging, regularization).
- Removal of many covariates might lead to jumpy/uneven final map.

Model selection with random forest

Variable importance

2 types of covariate importance:

- Sum of decrease in goodness-of-fit error by adding splits of this covariate (impurity), oriented on fitting the data. How much do we reduce error by using this covariate at this split?
- Mean decrease in OOB error by randomly permuting a covariate, oriented on predictions. How much worse do OOB predictions get if we randomly shuffle a covariate?

4.2 Approach 1: Recursive backward elimination

1. Remove covariate(s) with lowest importance.
2. Refit random forest with remaining.
3. Repeat (1)-(2) until all covariates are removed.
4. Find optimum number of covariates with minimal OOB error.

4.3 Approach 2: Boruta algorithm

1. All covariates are shuffled randomly. Shuffled covariates are appended to original covariates (shadow covariates).
2. Refit random forest with “real” and “shadow” covariates
3. Repeat (1)-(2) multiple times with different random seed.
4. Keep only covariates that are on average of larger importance than the shadow covariates.

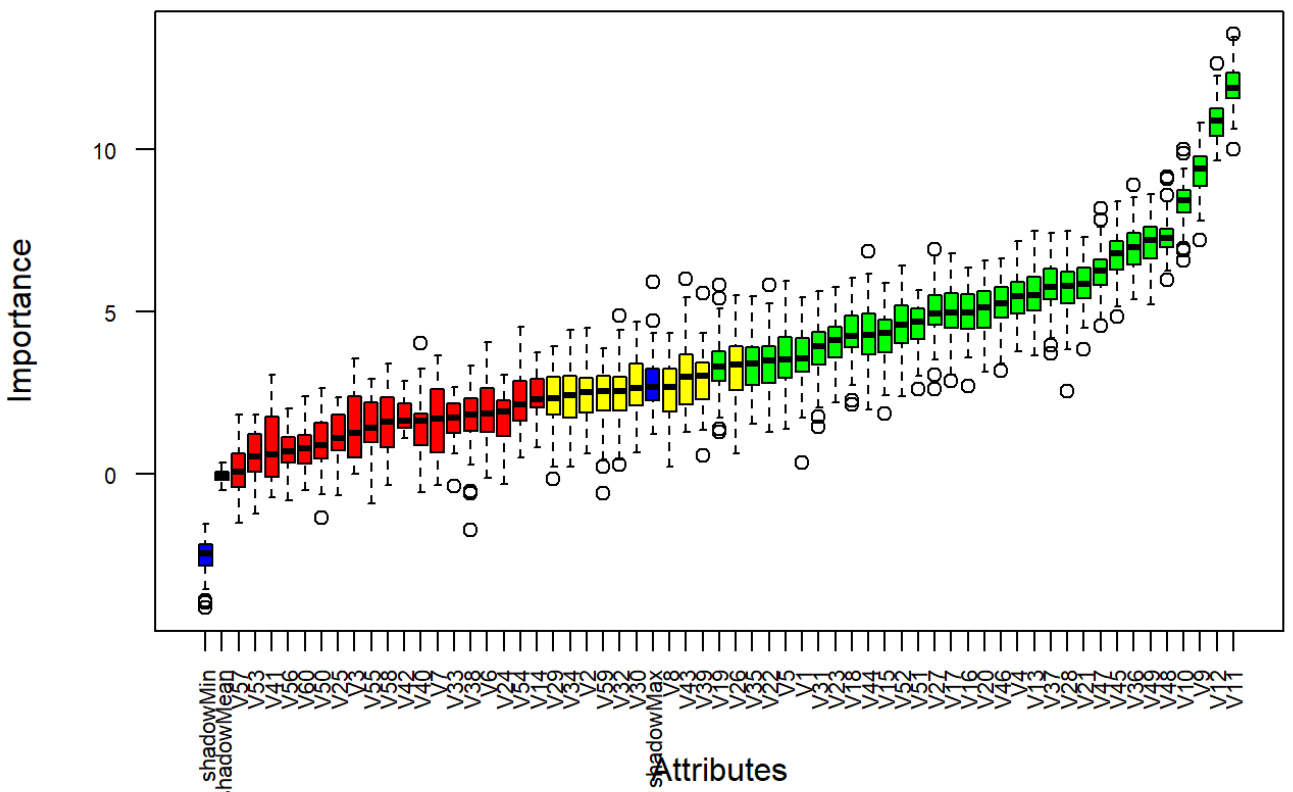
Boruta algorithm result

`blue`: shadow covariates, minimum, mean, maximum

`green`: larger importance than shadows

`yellow`: unclear decision (within variation of shadow max)

`red`: smaller importance than shadows



5 Model interpretation

5.1 Effects plots

Partial dependence plots

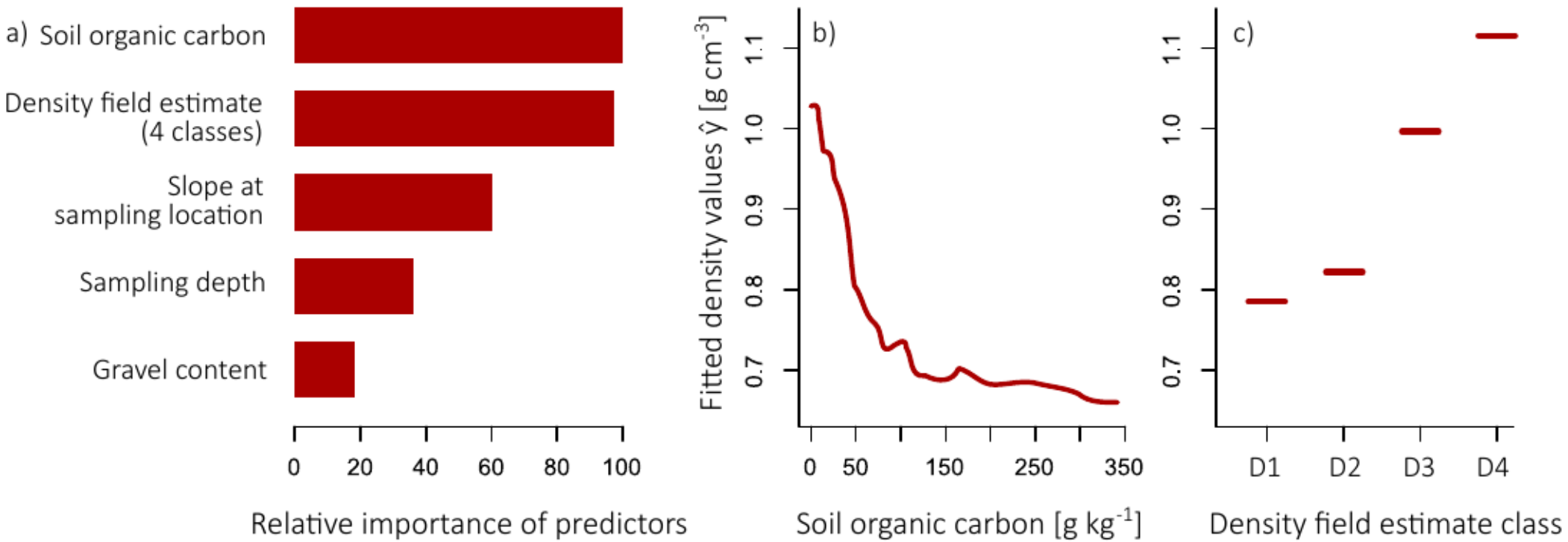
Partial depencende function $\hat{f}_s$ for the covariate s evaluated by calculating averages over the data used for model calibration:

$$\hat{f}_s(x_s) = \frac{1}{n}\sum_{i=1}^{n}\hat{f}(x_s, x_C^{(i)})$$

Partial dependece function reports for given value of covariate s the average marginal effect on the prediction. $x_C^{(i)}$ are actual covariate values for remaining covariates in which we are not interested in. n is the number of observations in the dataset.

Assumption: no correlation between covariate s and remaining covariates C.

Partial dependence plots – example

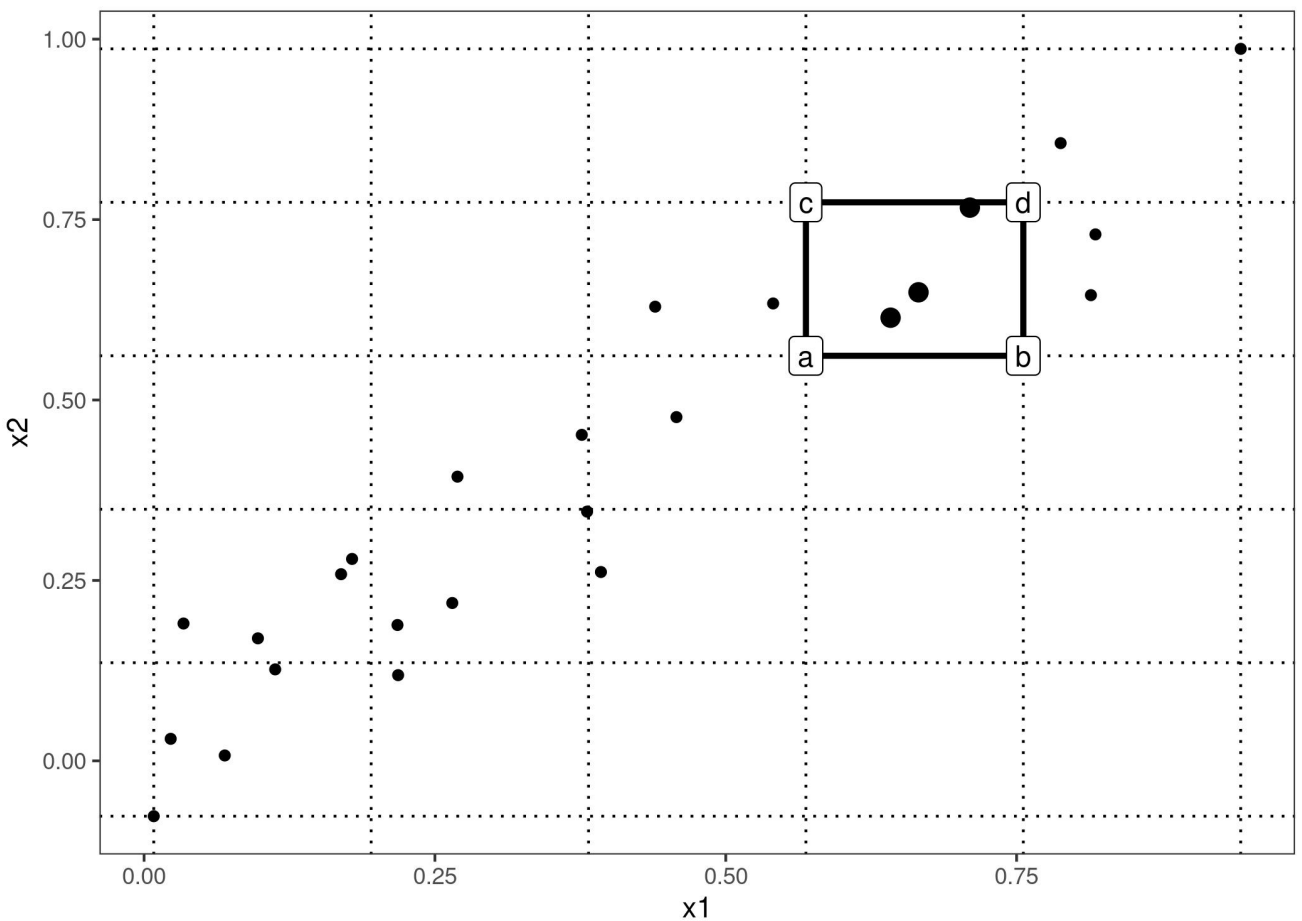


Importance and depenence plots for a model to predict soil density (expensive to measure) from other soil properties (cheaper to observe).

Accumulated local effects plots

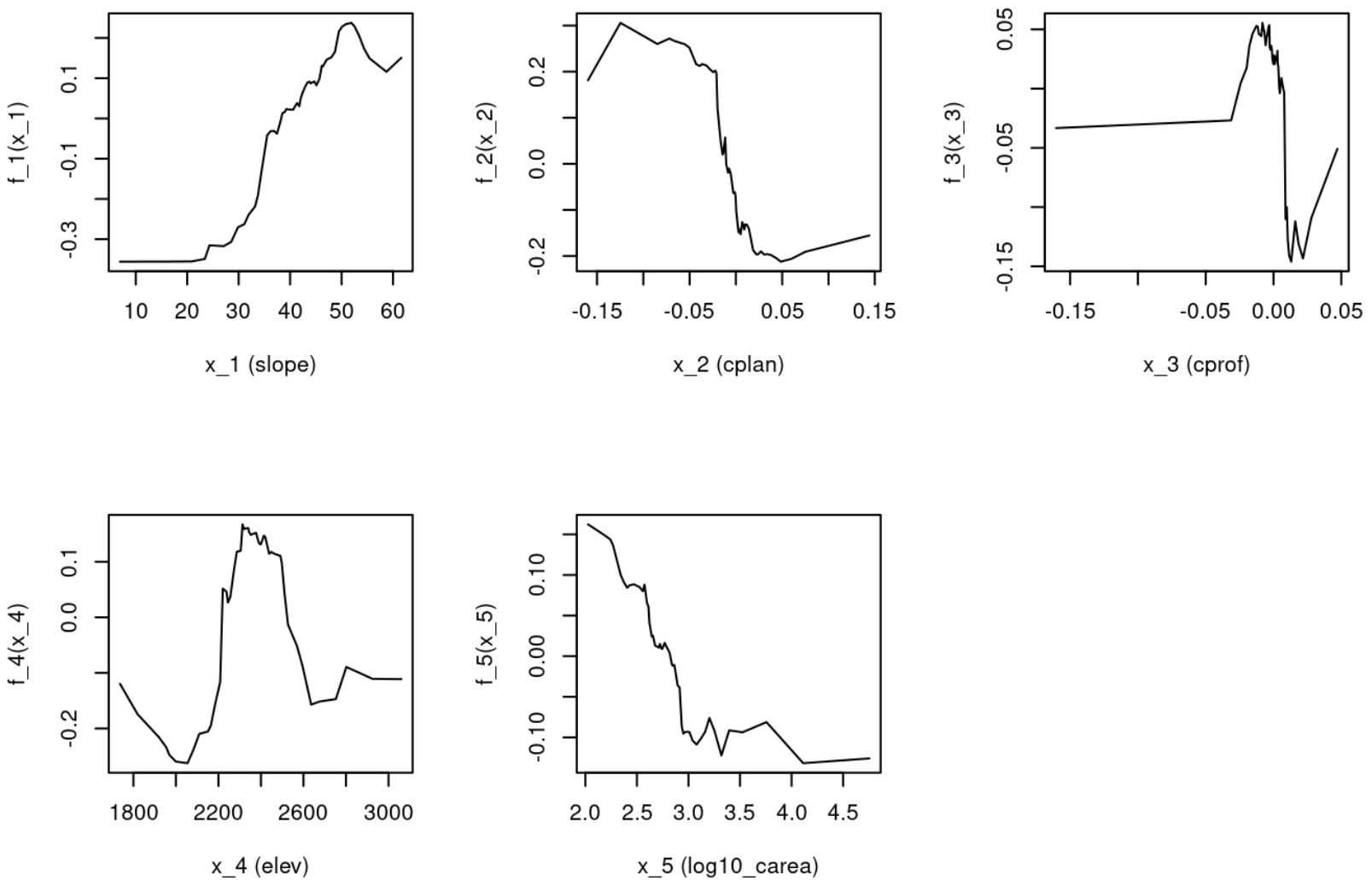
Accounting for correlation structure in covariates.

Instead of using mean of remaining covariates x_C a moving window approach is implemented.



Molnar, 2024. Fig. 8.8

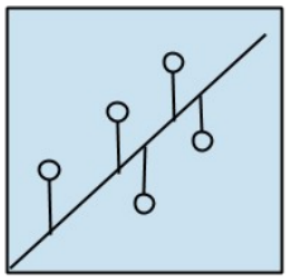
Accumulated local effects plots – example



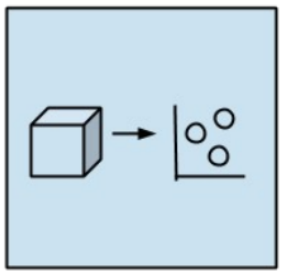
6 Machine learning methods

There is not just random forest

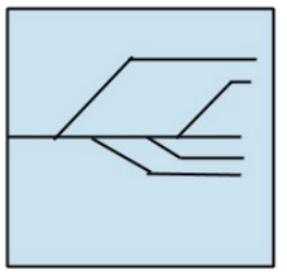
I tried to tidy up ...



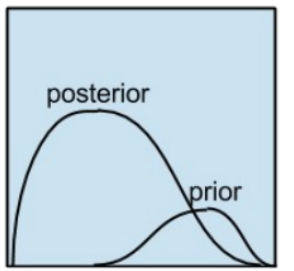
Regression
linear and non-linear models, geostatistics



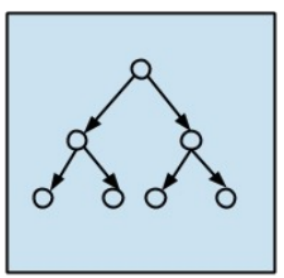
Dimension reduction
PCA, PLS



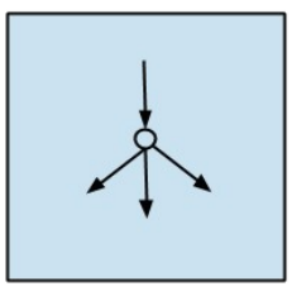
Regularisation Shrinkage
Lasso



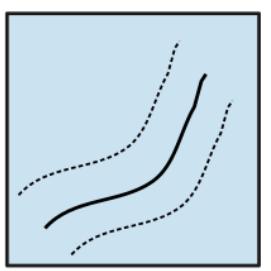
Bayes methods



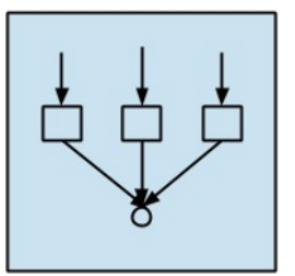
Decision trees
CART



Neuronal networks



Support vector machines
kernel methods



Ensembles
bootstrap, boosting, model averaging

ML for spatial prediction – summary

Advantages

- Machine learning (ML) handles todays requirements well
  - Large dataset sizes (*n* observations, *n* covariates)
  - Lowering computational demands
  - Automatic model selection and building of model structure
  - Classification tasks
  - Uncertainty for point locations

Disadvantages

- Spatial auto-correlation is only integrated in ad-hoc manner – currently no satisfactory solution
- Standard errors for spatial averages only via workaround (e.g. to report quantities per municipality or arable land parcel)
- Solved problems within classical geostatistics reoccur for ML