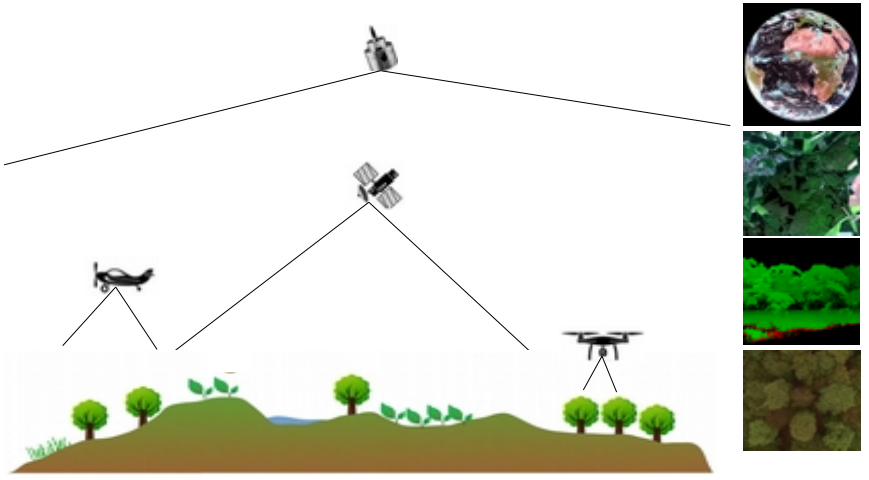
Introduction to machine learning for spatial environmental modelling

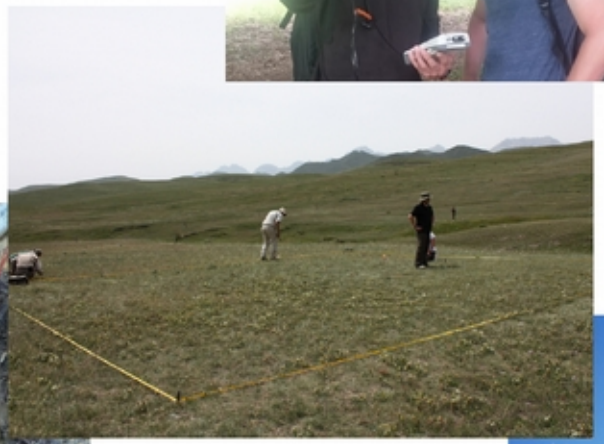
Hanna Meyer

ifgi
Institute for Geoinformatics
University of Münster



→ Slides and material: https://github.com/HannaMeyer/ML4GenTree

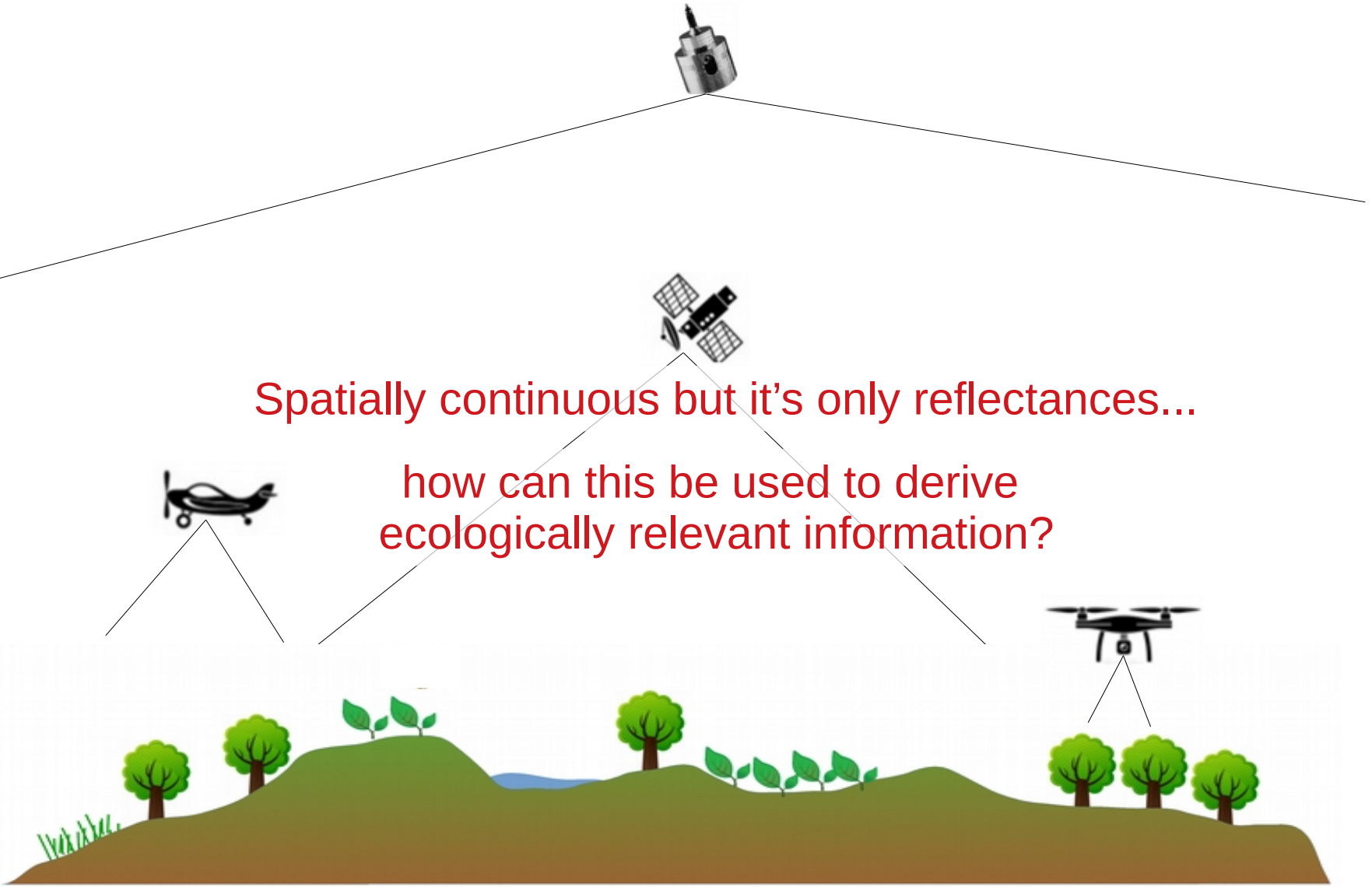
Problem: From field observations to maps of ecosystem variables



Problem: From field observations to maps of ecosystem variables

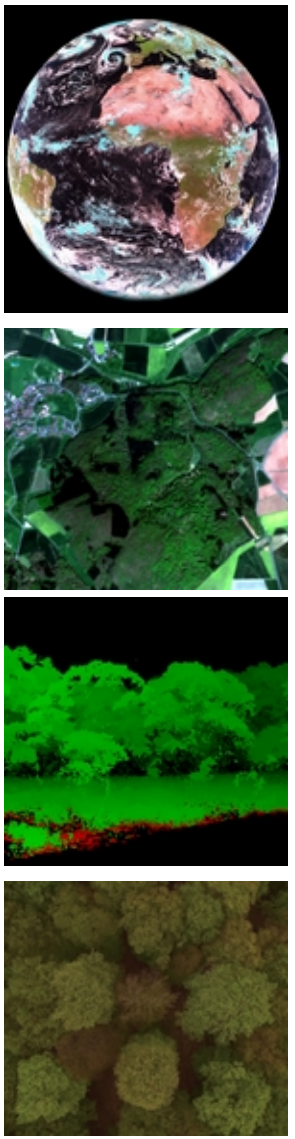
Nature 4.0 | Sensing Biodiversity

Remote Sensing of landscapes



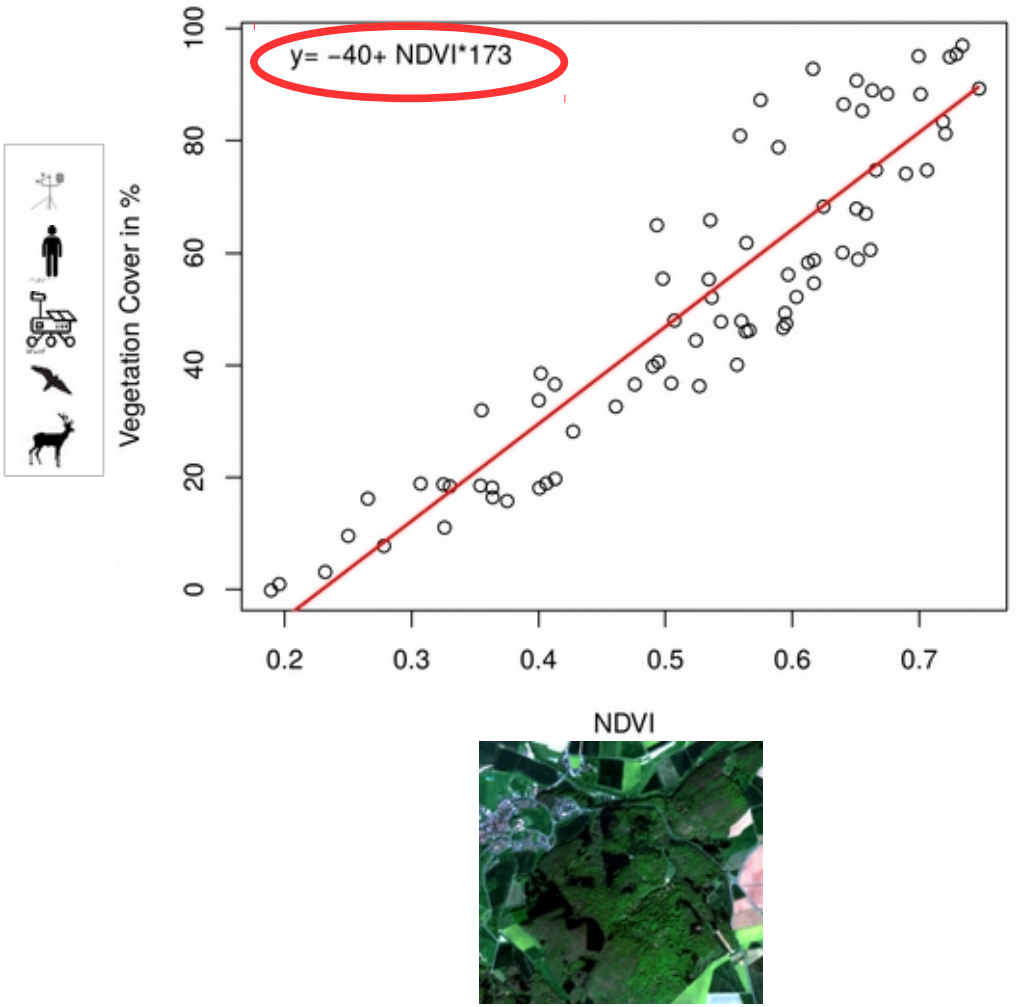
Spatially continuous but it's only reflectances...

how can this be used to derive ecologically relevant information?

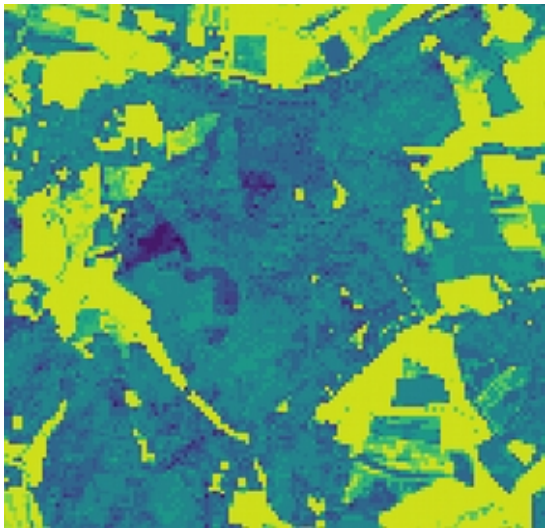


Statistical modelling

E.g. vegetation cover from remote sensing

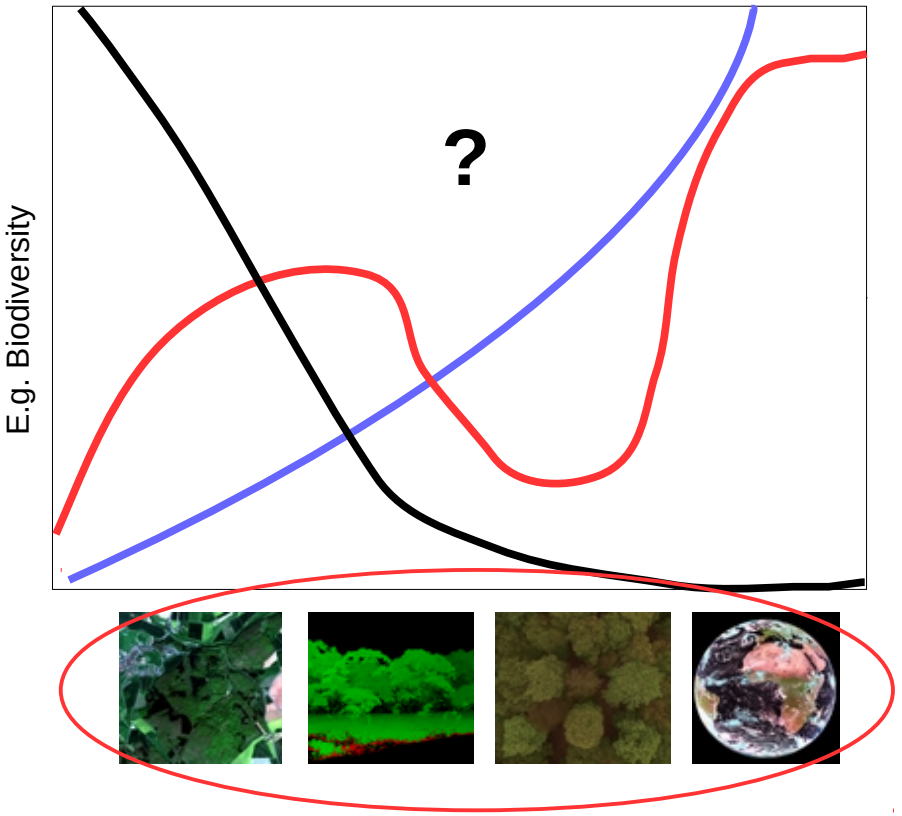


Modelled vegetation cover



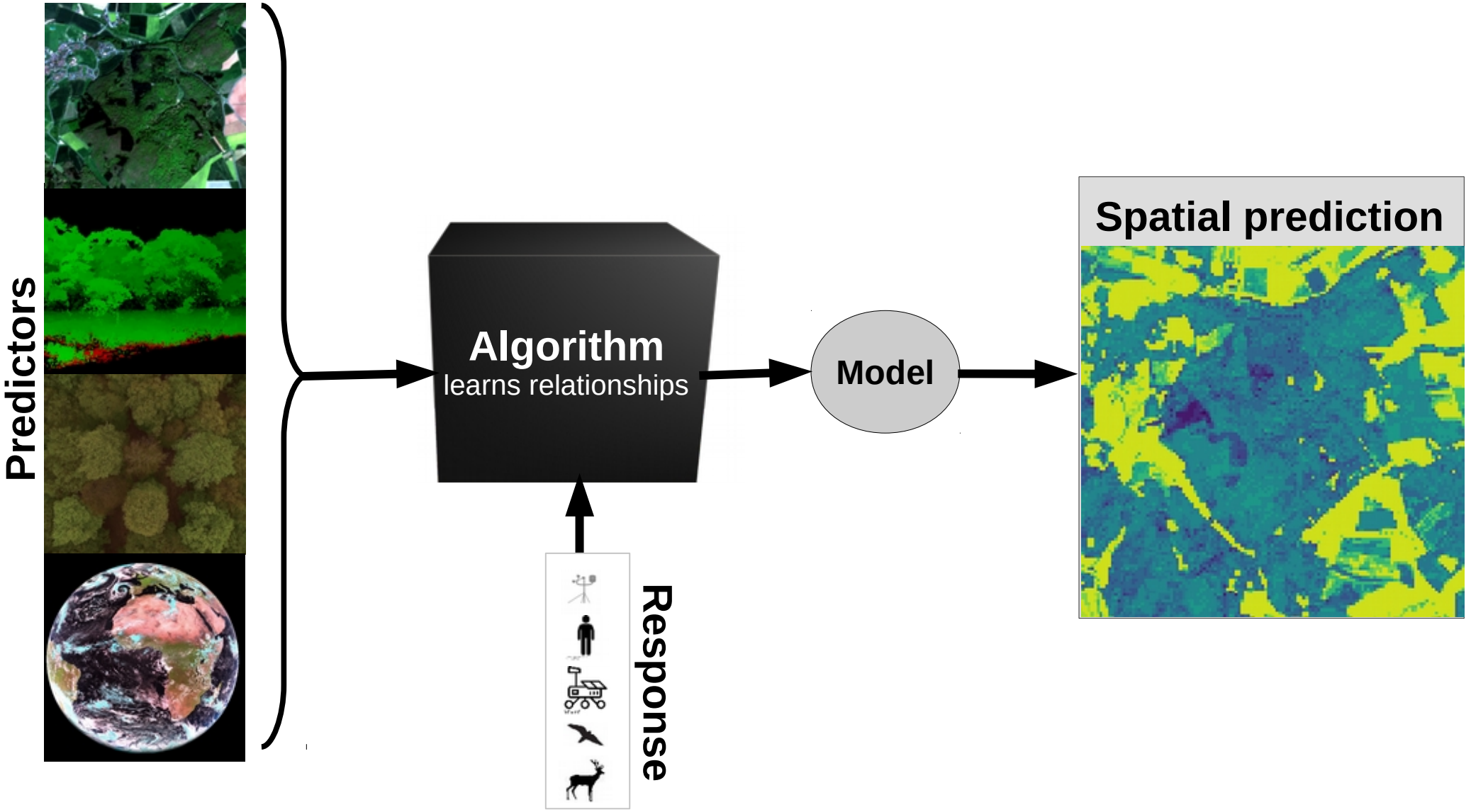
Statistical modelling

Typical ecological variables from satellite?



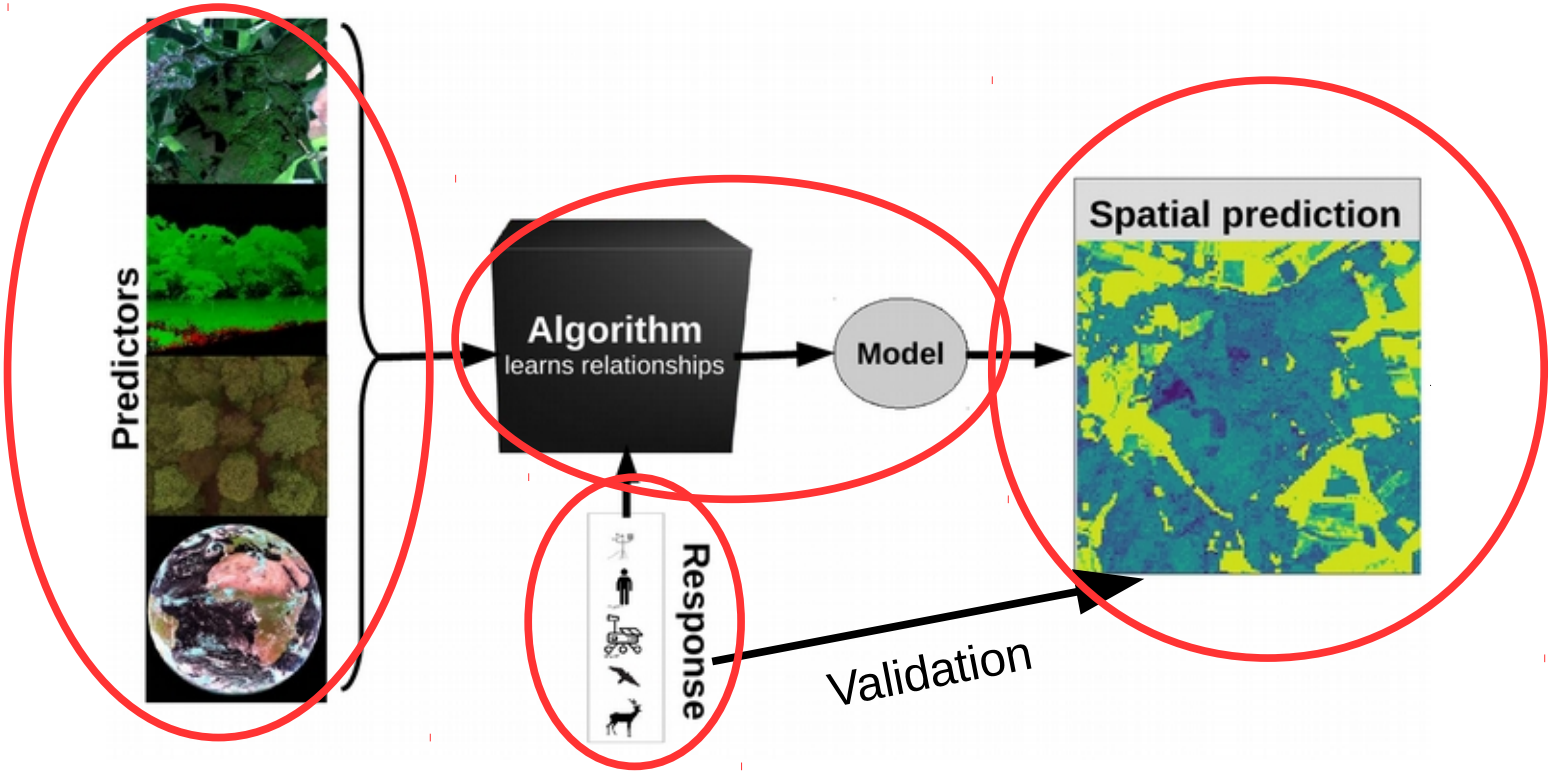
Models that can deal with complex nonlinear relationships are required!

Remote-sensing based monitoring of the environment: The machine learning way



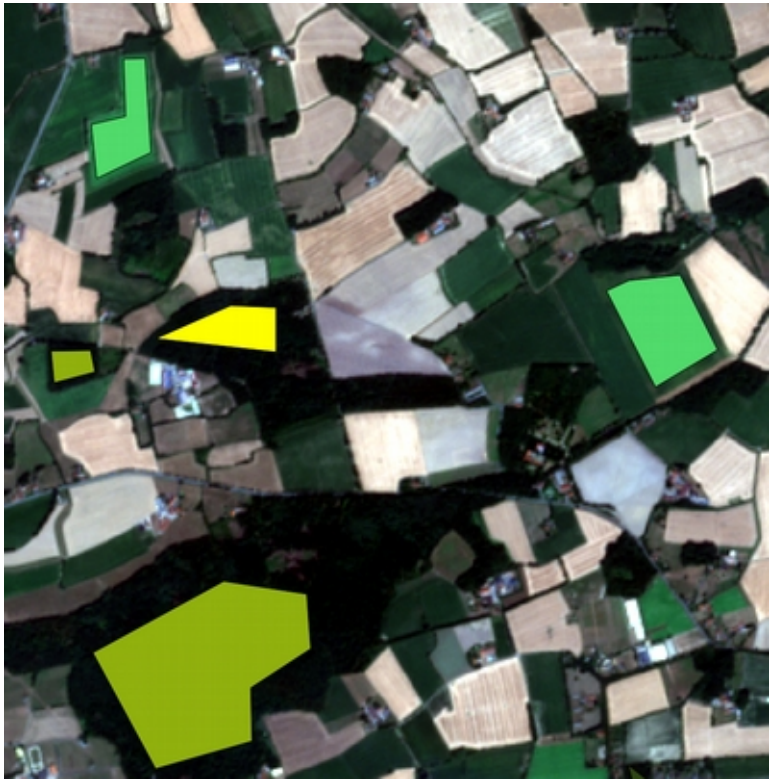
A basic machine learning workflow in 5 steps

1) Select training data
2) Choose predictor variables
3) Train a model
4) Make spatial predictions
5) Model validation

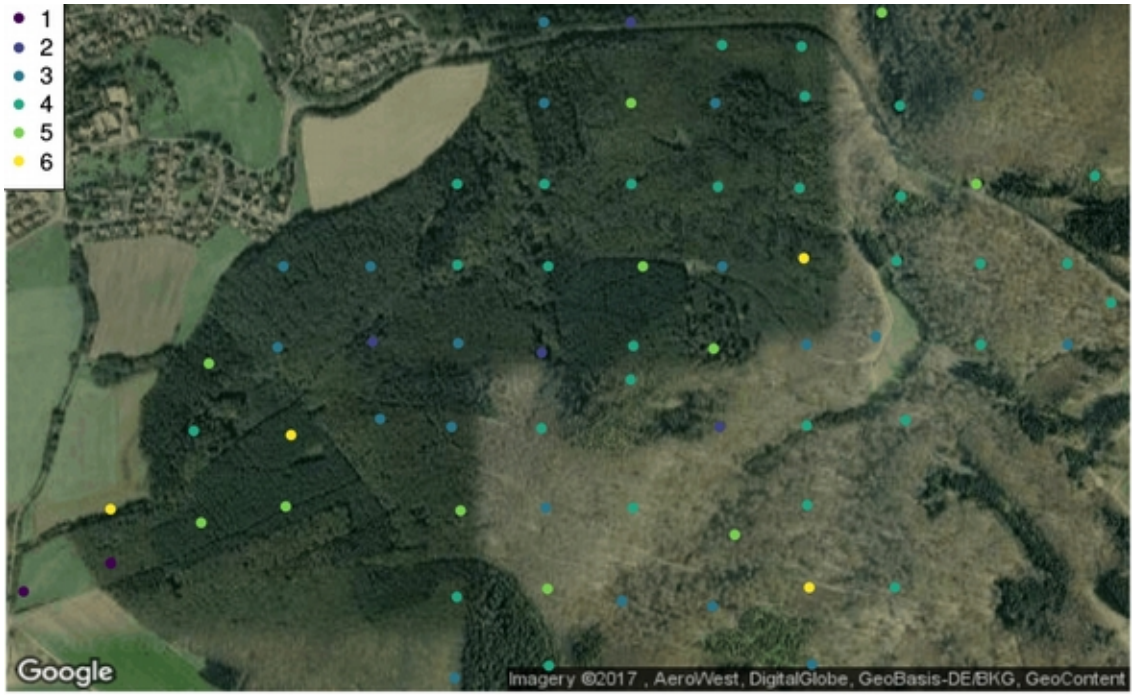


Step 1: Training data (ground truth/reference/response)

for classification



… or regression

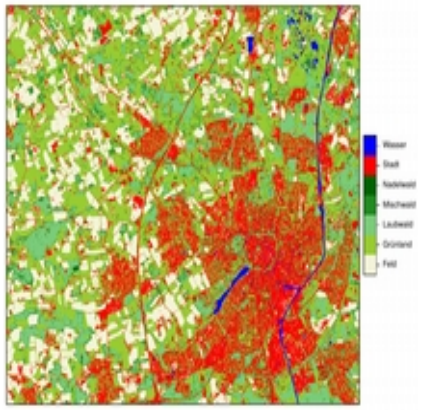


Step 2: Selecting predictor data

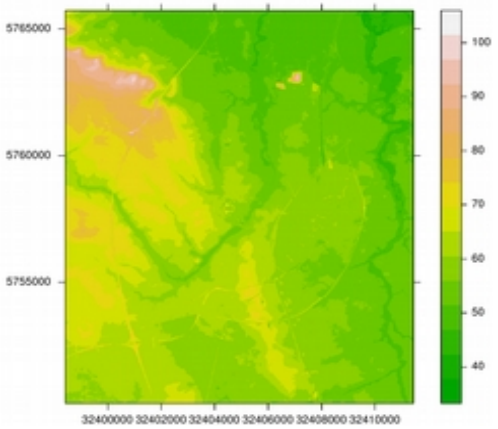
- Variables that we assume have a relationship with the target variable
- Numeric or categorical
- Need to be available in a spatial continuous way

e.g. some ideas for predictors for soil organic carbon modelling

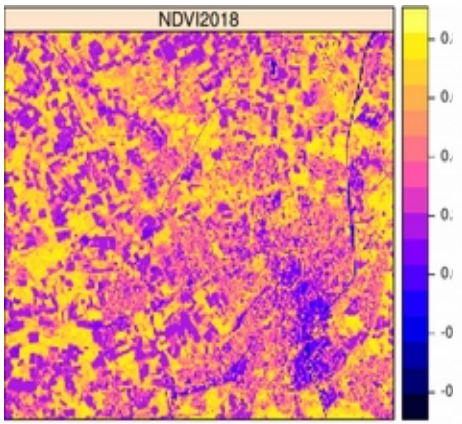
Land cover



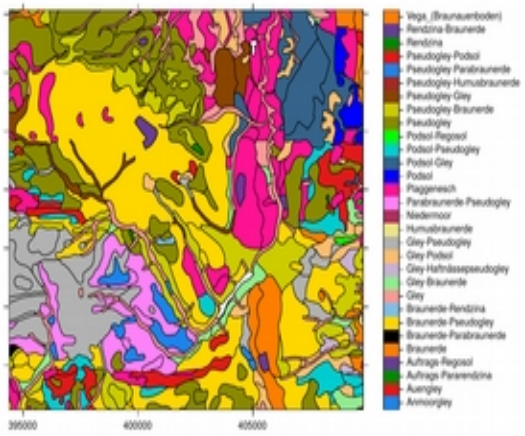
Elevation



NDVI

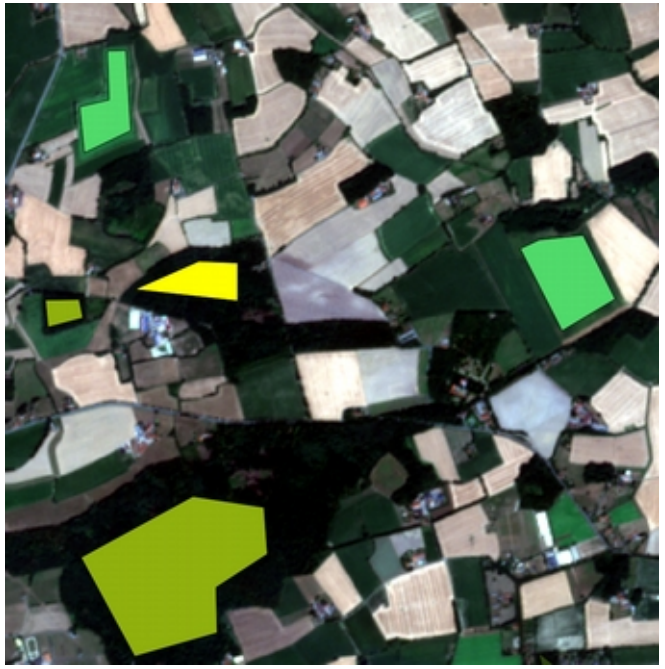


Soil types



Step 2b: Combine predictors and response

Extract predictors at the location of the training locations

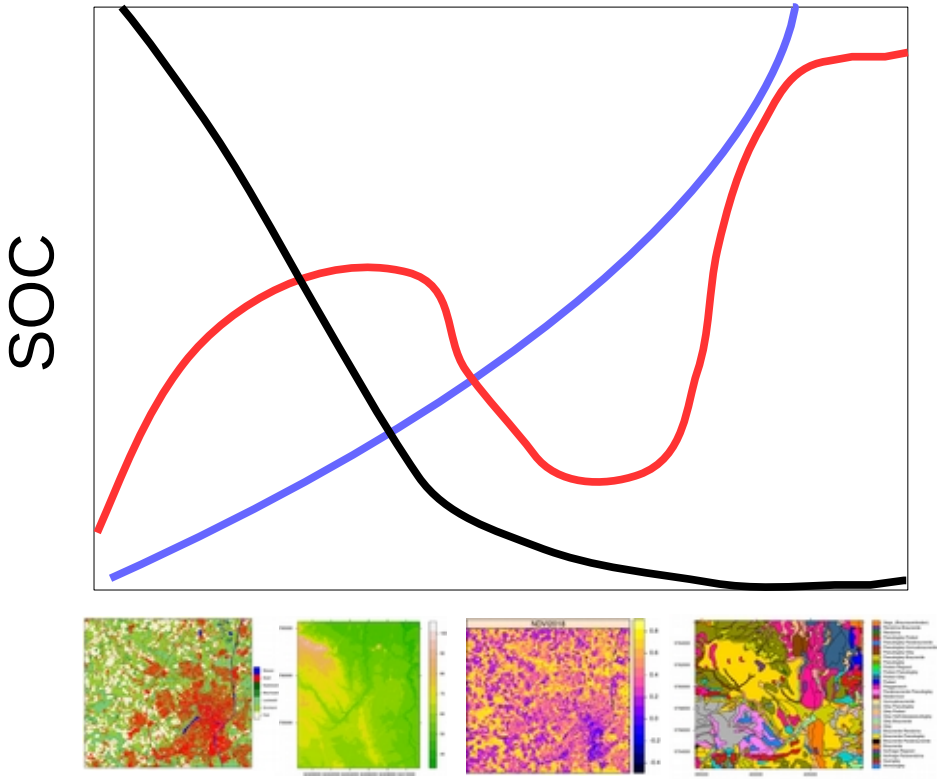


Predictors | Response

	B02	B03	B04	B08	...	Class
1	857	632	387	308		Water
2	848	633	389	312		Water
3	843	624	357	343		Water
4	854	630	360	333		Water
5	854	628	376	302		Water
6	859	615	364	350		Water

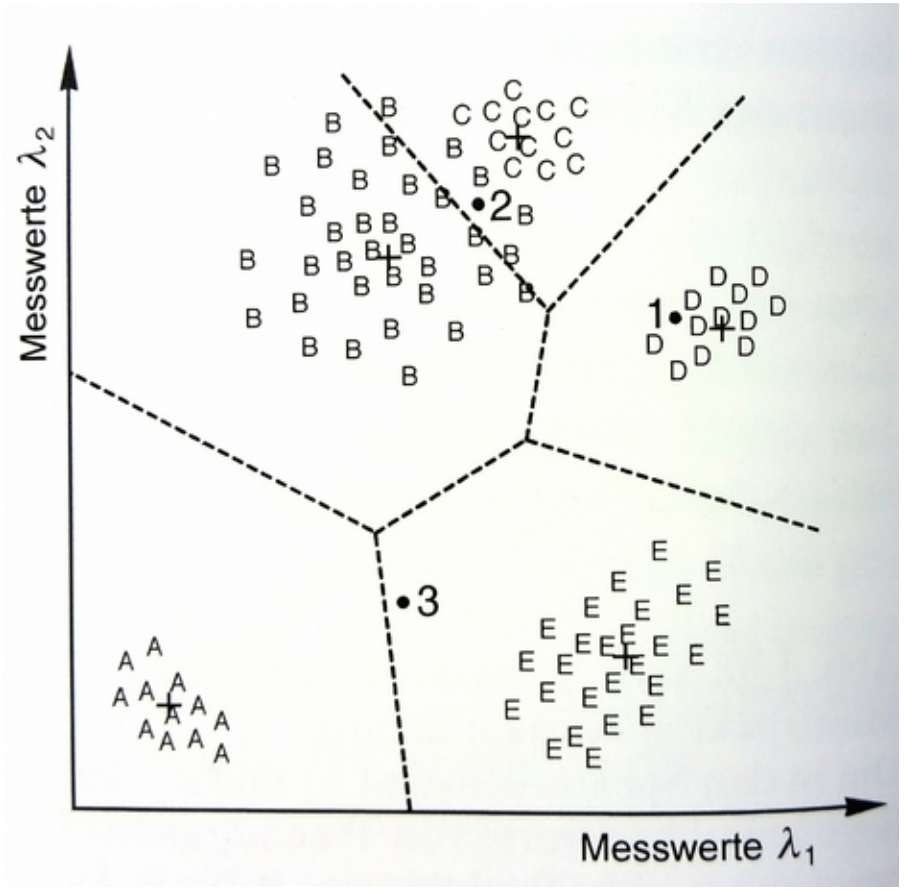
Step 3: Model training

Apply a machine learning algorithm to learn relationships between predictors and response

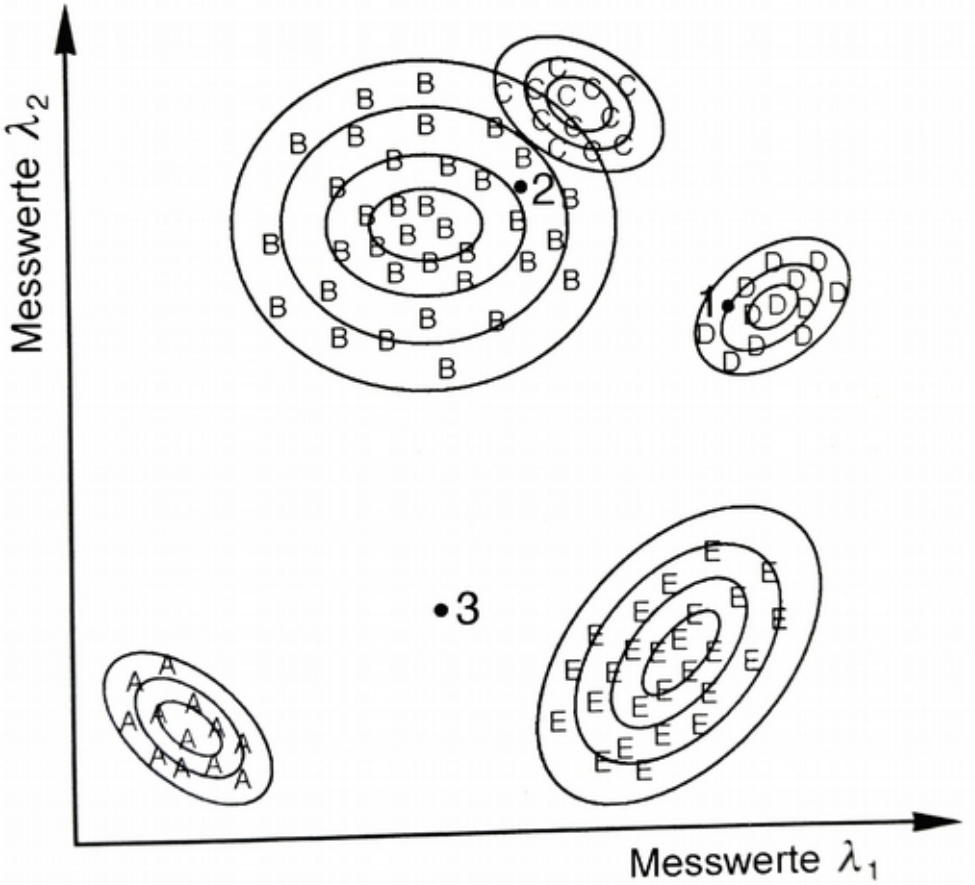


Step 3: Model training - Traditional Supervised Classifiers

Minimum distance classifier

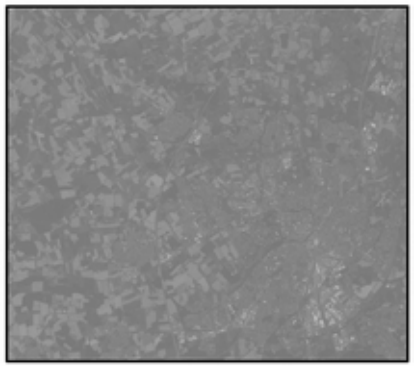


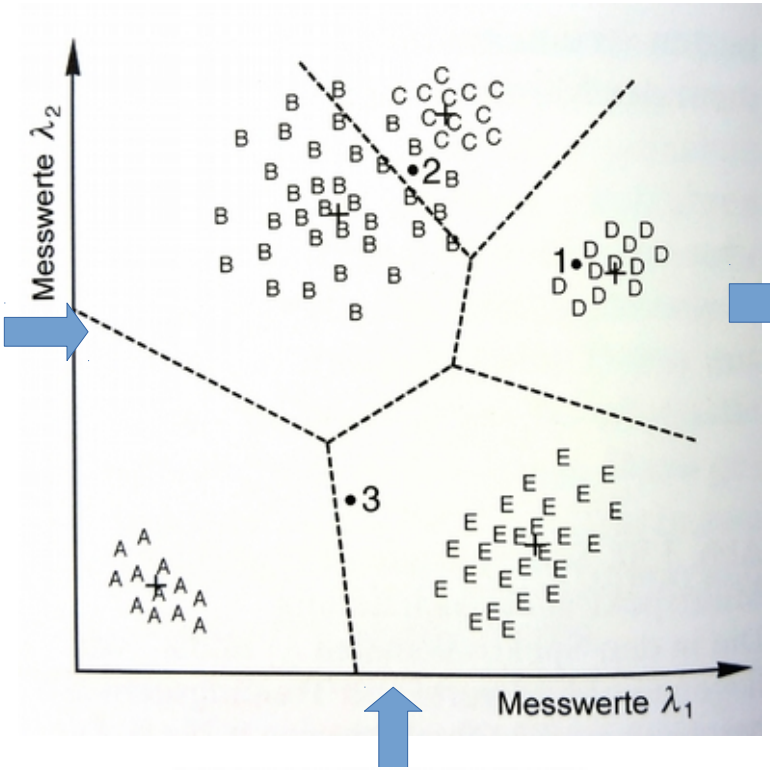
Maximum likelihood classifier

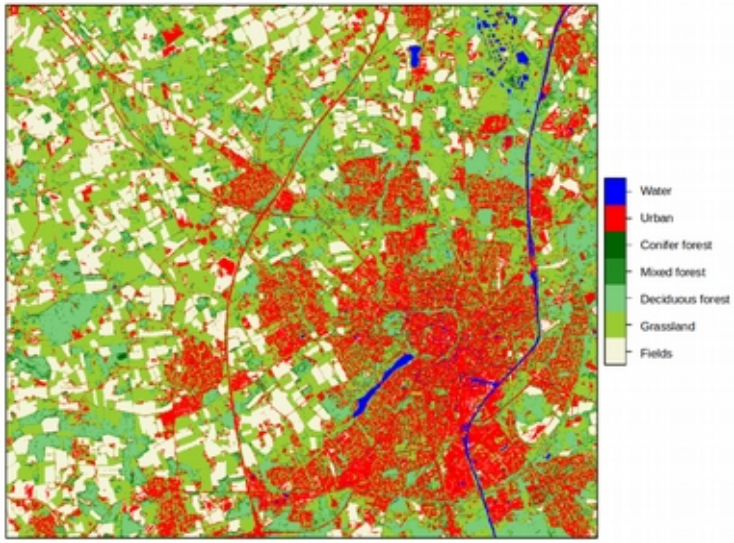


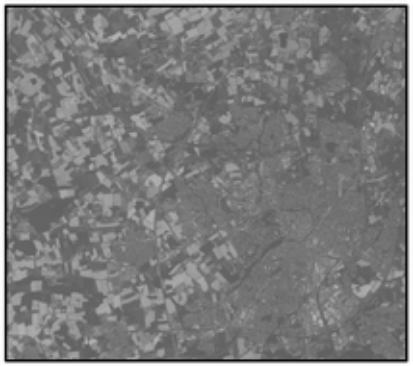
Albertz (2009): Einführung in die Fernerkundung. WBG, Darmstadt

Step 3: Model training - Traditional Supervised Classifiers









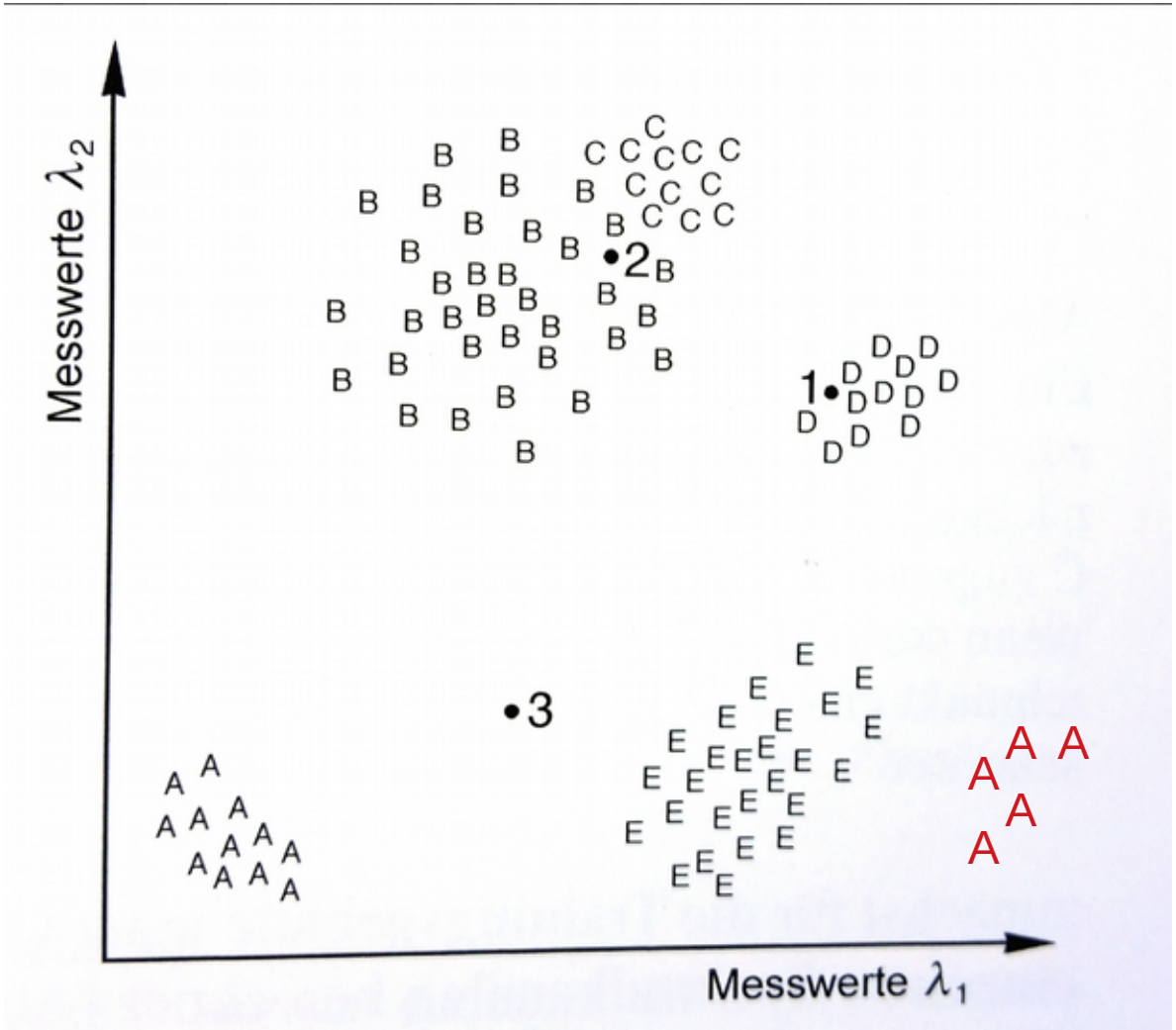
Classification of each Pixel (1-3,…) according to its location in the feature space

Step 3: Model training: Traditional Supervised Classifiers

Complex patterns cannot be accounted for by these algorithms!

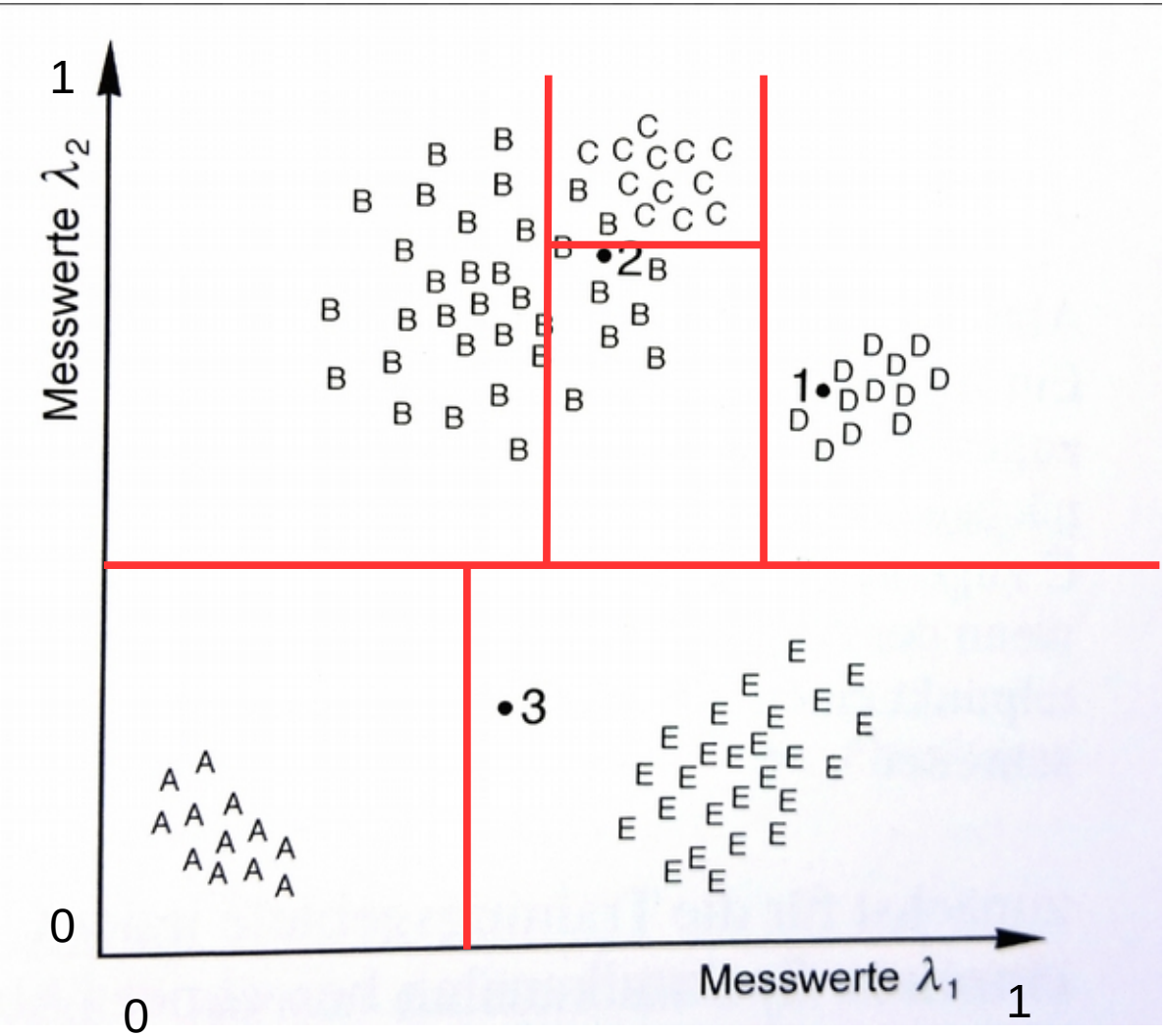
E.g. variations in the spectral characteristics depending on

- Forest on slopes with different aspect
- Crops that are irrigated/non irrigated
- Water in streams/lakes
- ...

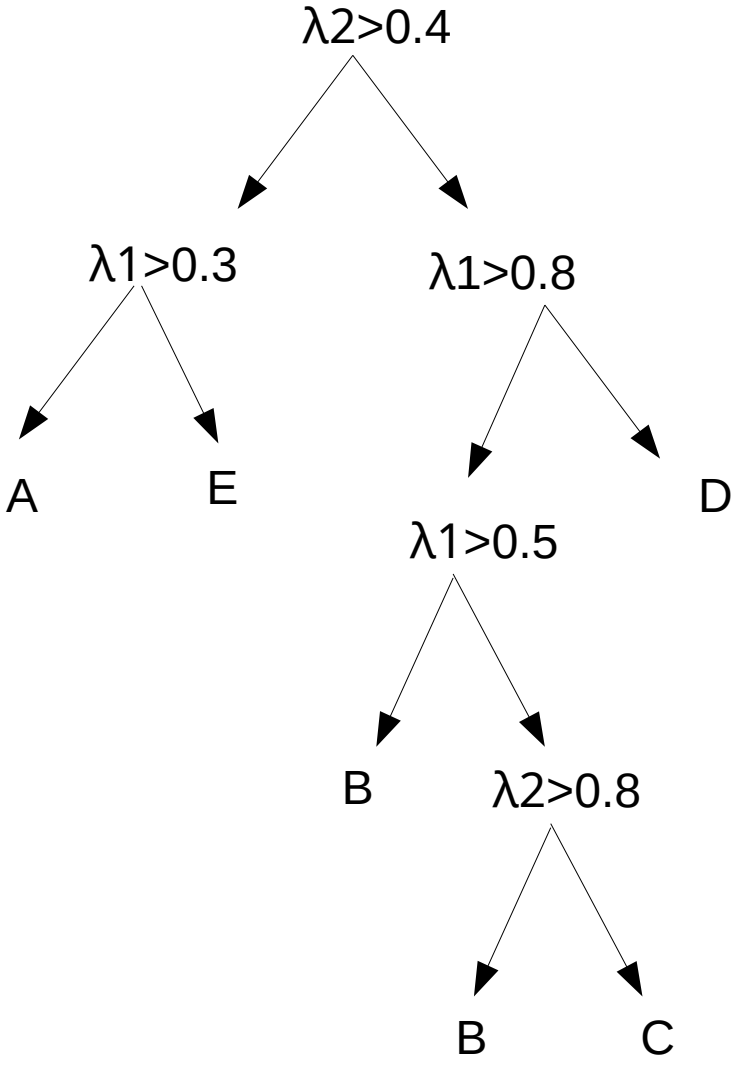


→ we need more complex models

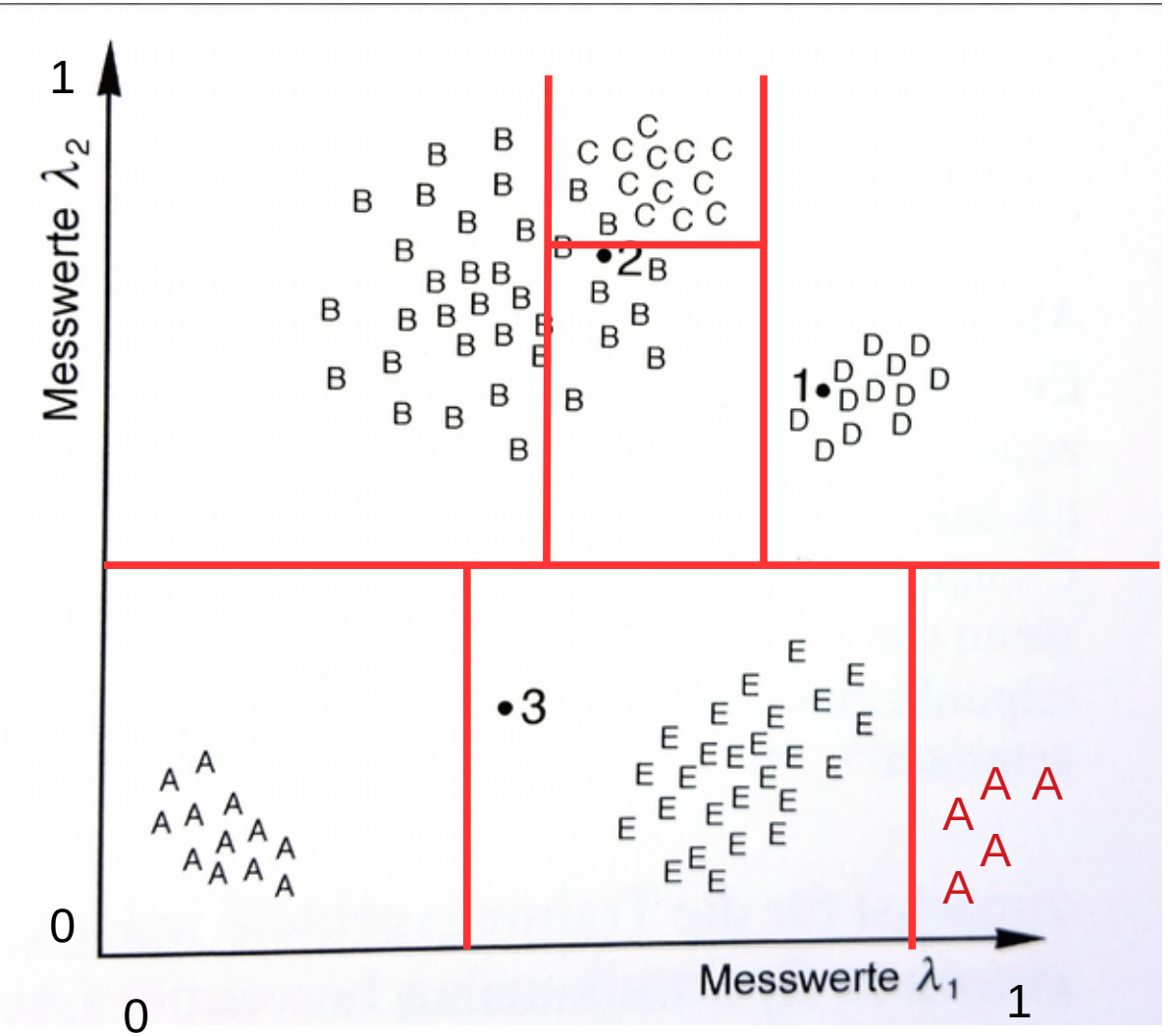
Step 3: Model training – Example Random forest



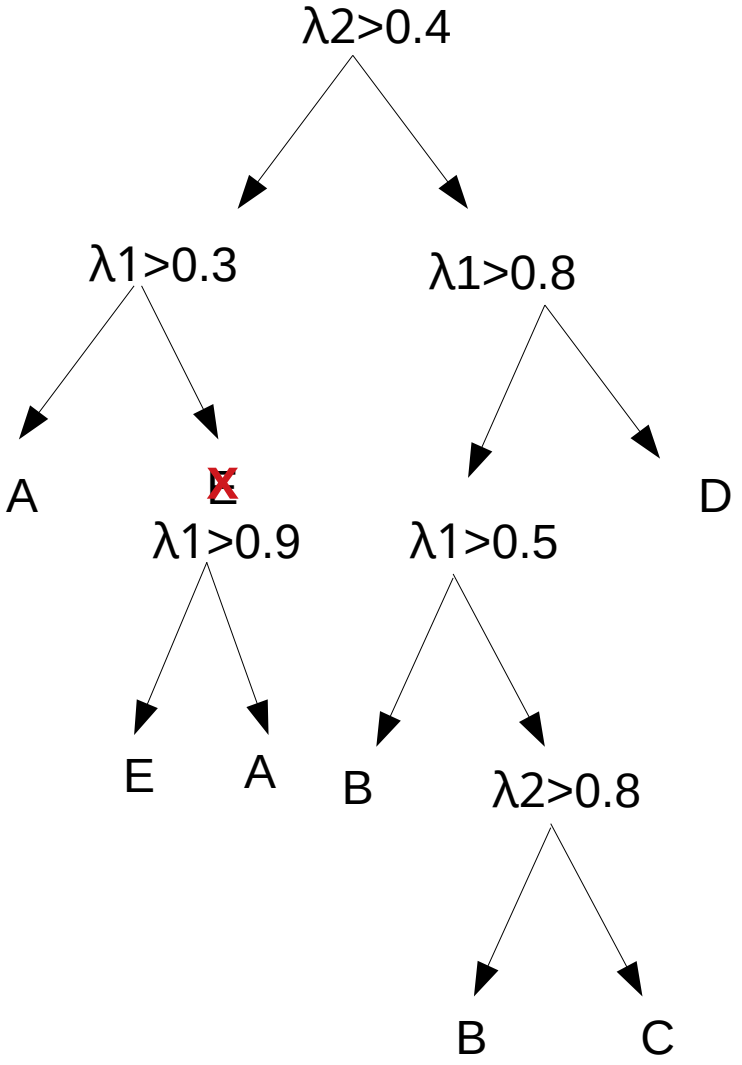
changed from Albertz (2009): Einführung in die Fernerkundung. WBG, Darmstadt



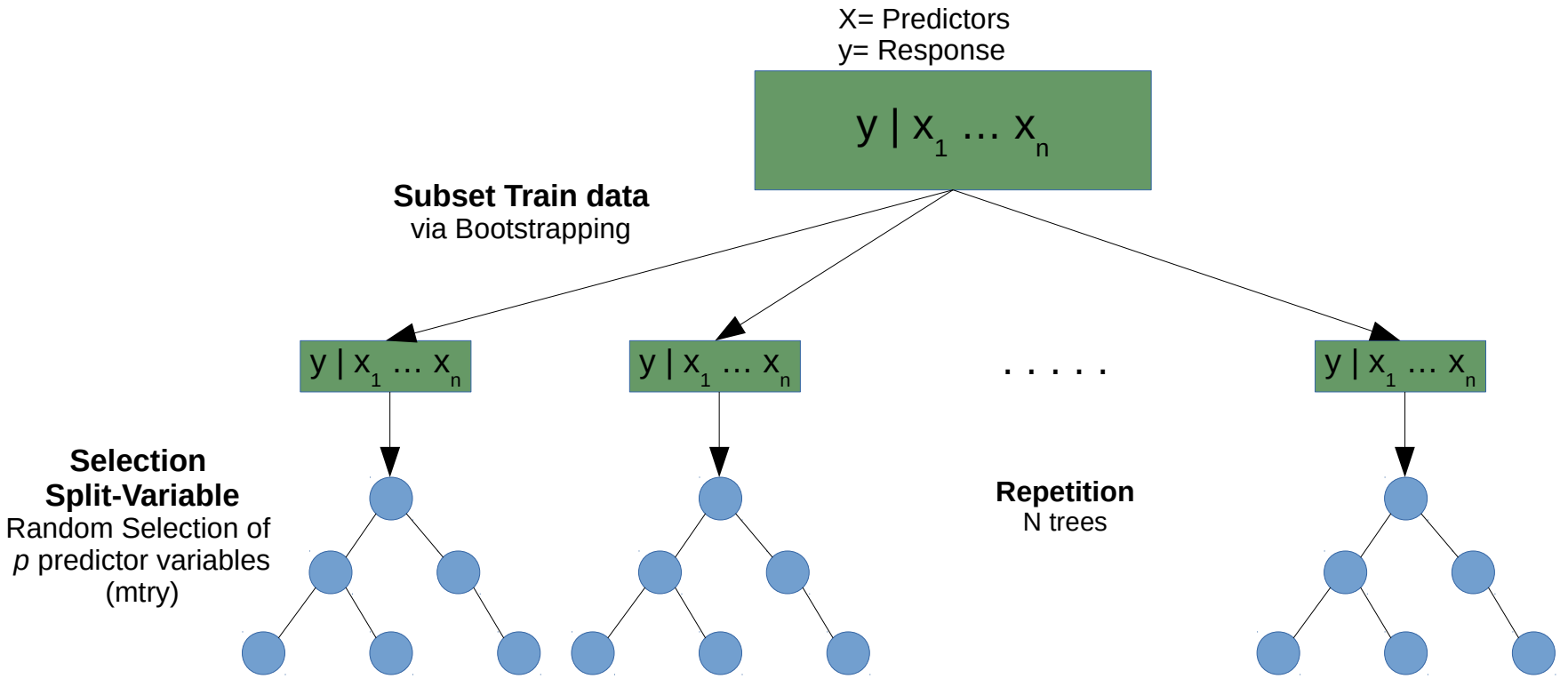
Step 3: Model training – Example Random forest



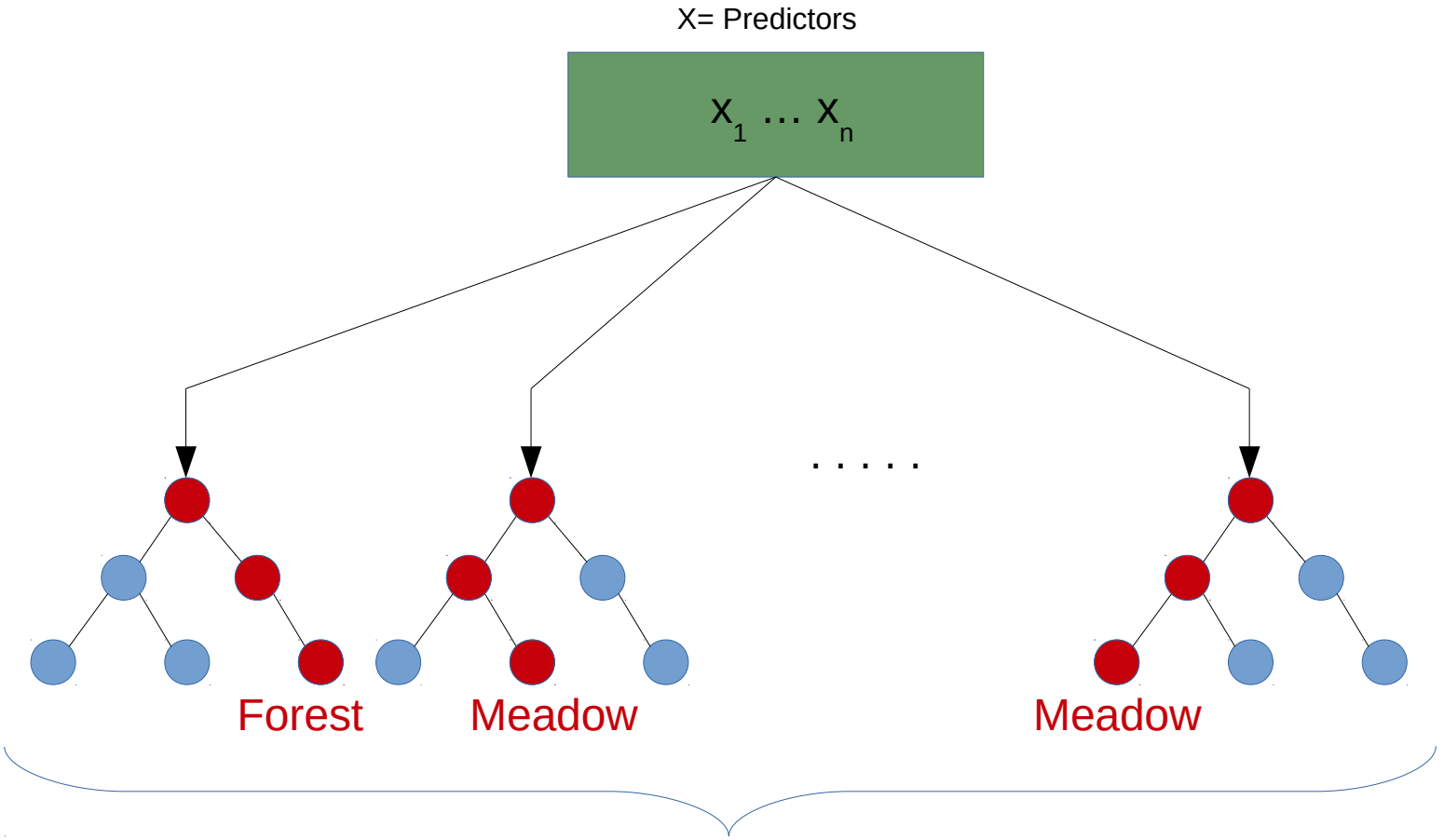
changed from Albertz (2009): Einführung in die Fernerkundung. WBG, Darmstadt



Random Forest: Model training



Random Forest: Prediction

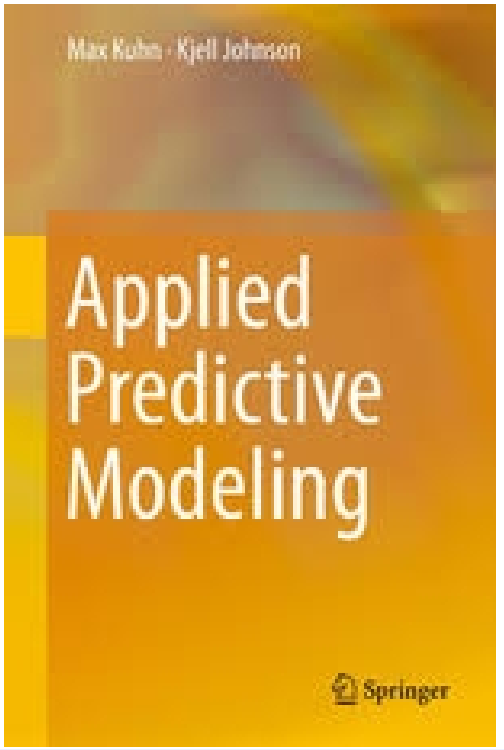


Decision made by combining the individual trees

(Classification: Majority; Regression: Mean)

Step 3: Model training in R

- Many packages for different ML algorithms (e.g. Random Forests, Neural Networks, Support Vector Machines, ...)
- For classification and regression problems
- Wrapper packages
  - allowing access to many algorithms via a unified syntax
  - Supporting functionality for cross-validation etc.
  - Caret (Classification And REgression Training)
  - Mlr (Machine Learning in R)
  - Tidymodels



Step 3: Model training (Random forests)

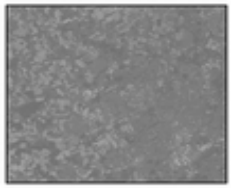
Predictors | Response

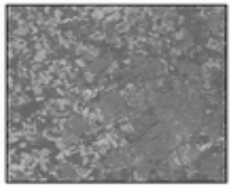
	B02	B03	B04	B08	...	Class
1	857	632	387	308		Water
2	848	633	389	312		Water
3	843	624	357	343		Water
4	854	630	360	333		Water
5	854	628	376	302		Water
6	859	615	364	350		Water

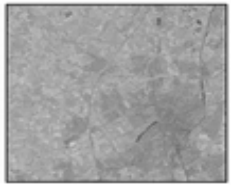
How to do it in R

```
library(caret)
model <- train(predictors,
                response,
                method="rf")
```

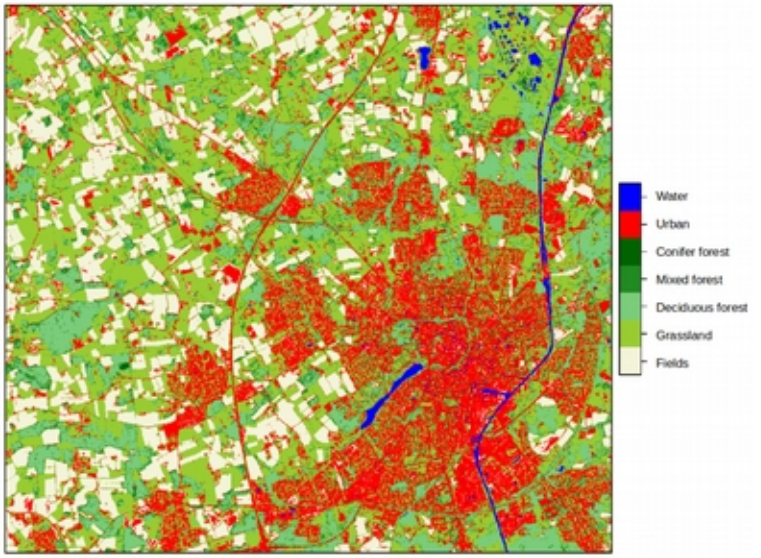
Step 4: Model prediction







+ trained model =



...

How to do it in R

```
library(raster)
pred_sp <- stack(predictors)
prediction <- predict(pred_sp,model)
```

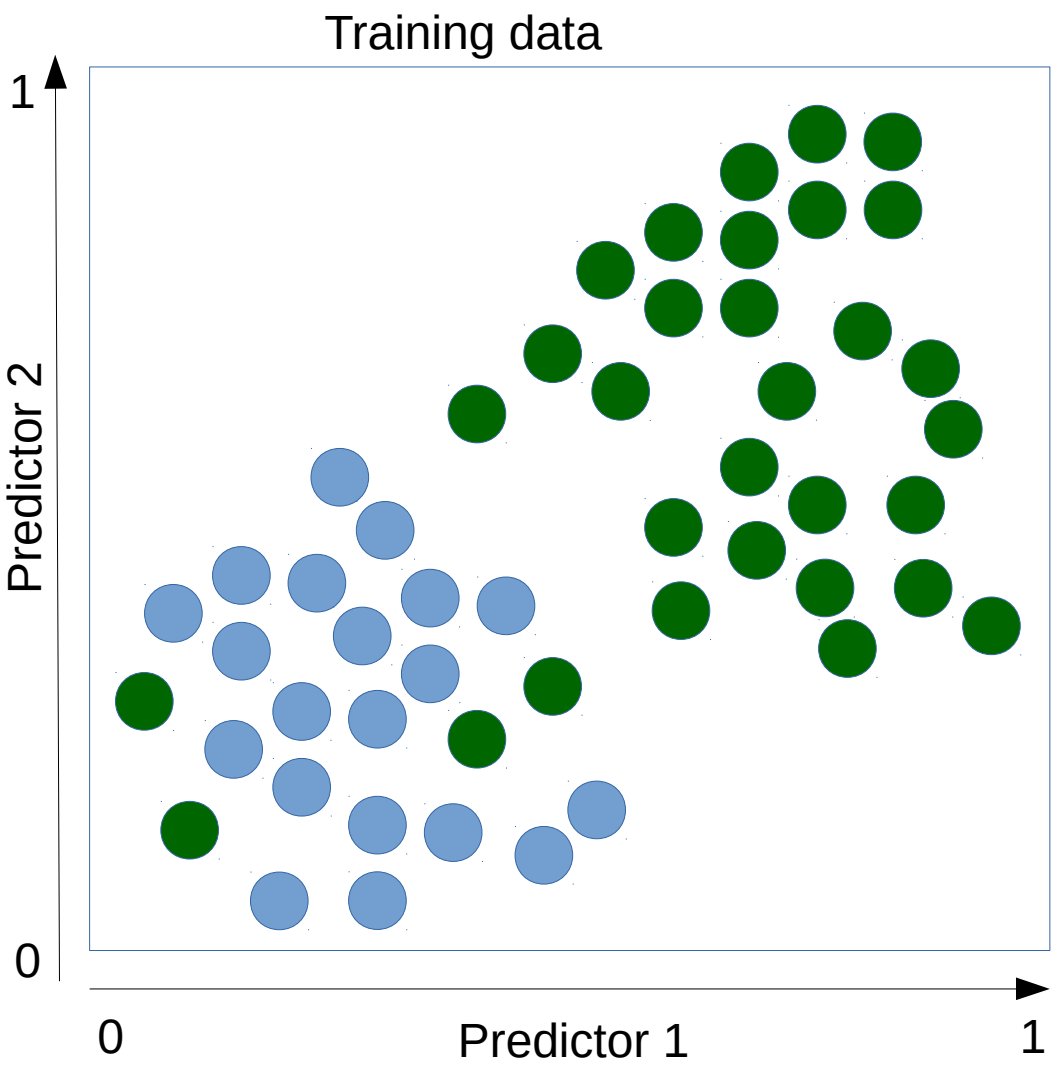
Step 5: Model validation

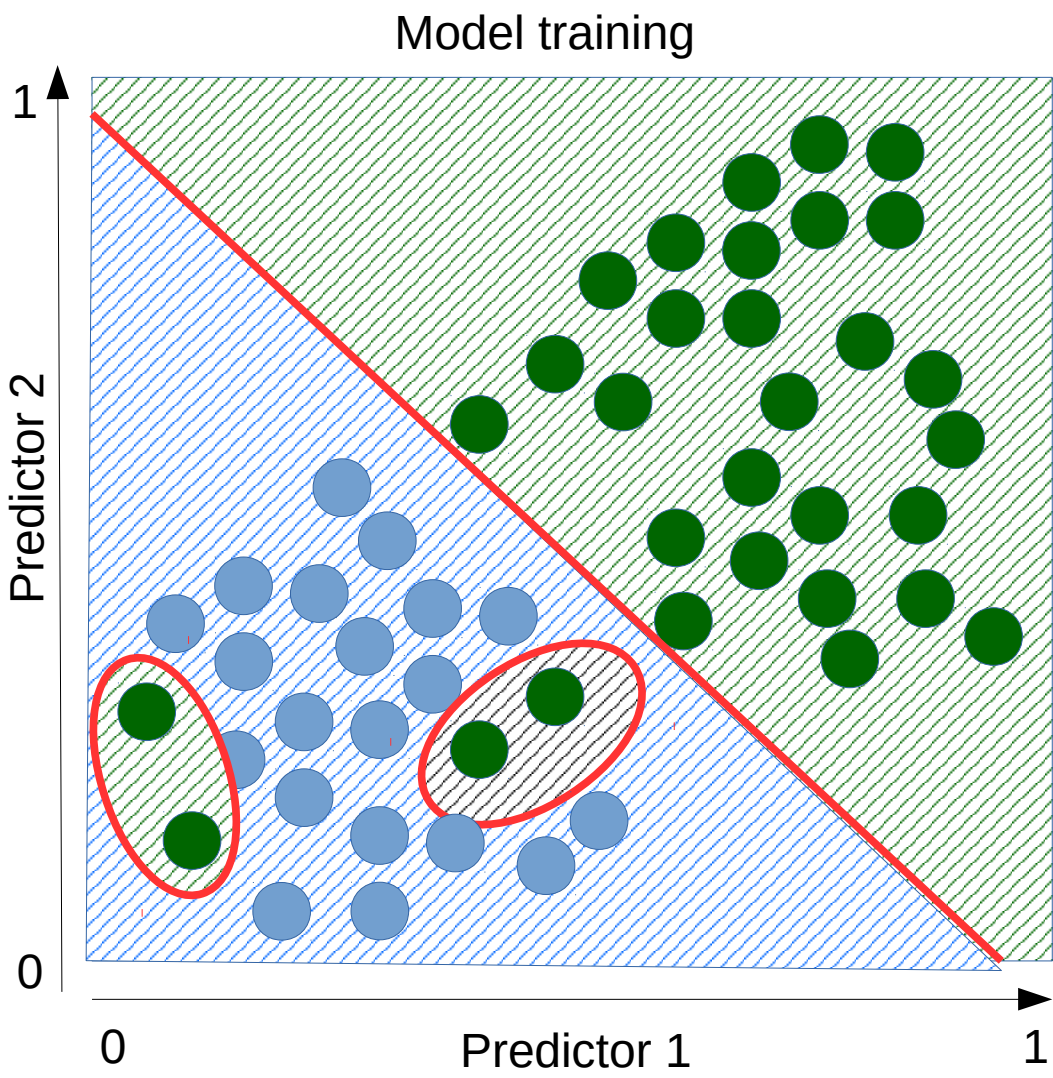
https://xkcd.com/1838/

We want to avoid this!!

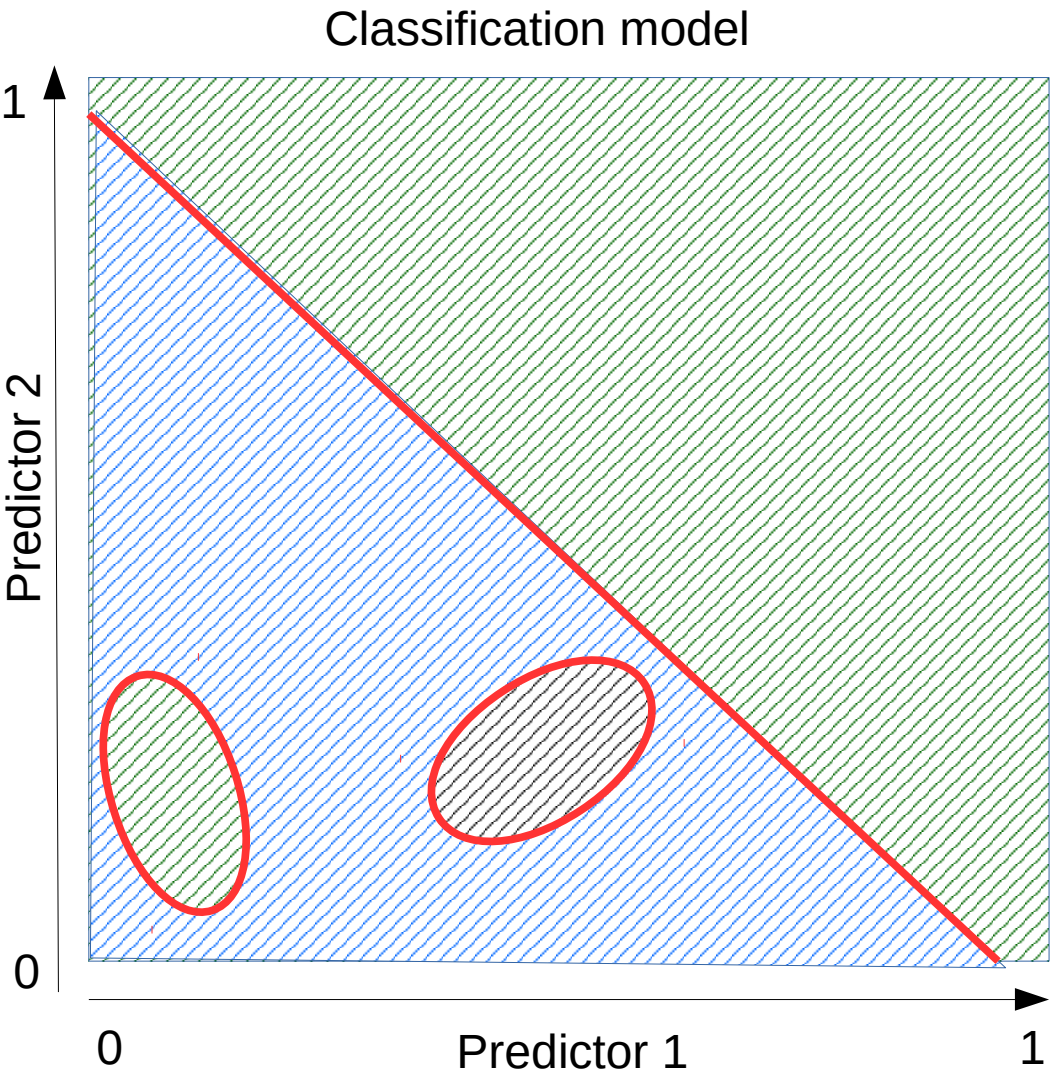
- Robustness needs to be ensured
- First important point: Meaningful error estimates are needed
- How good is the model?

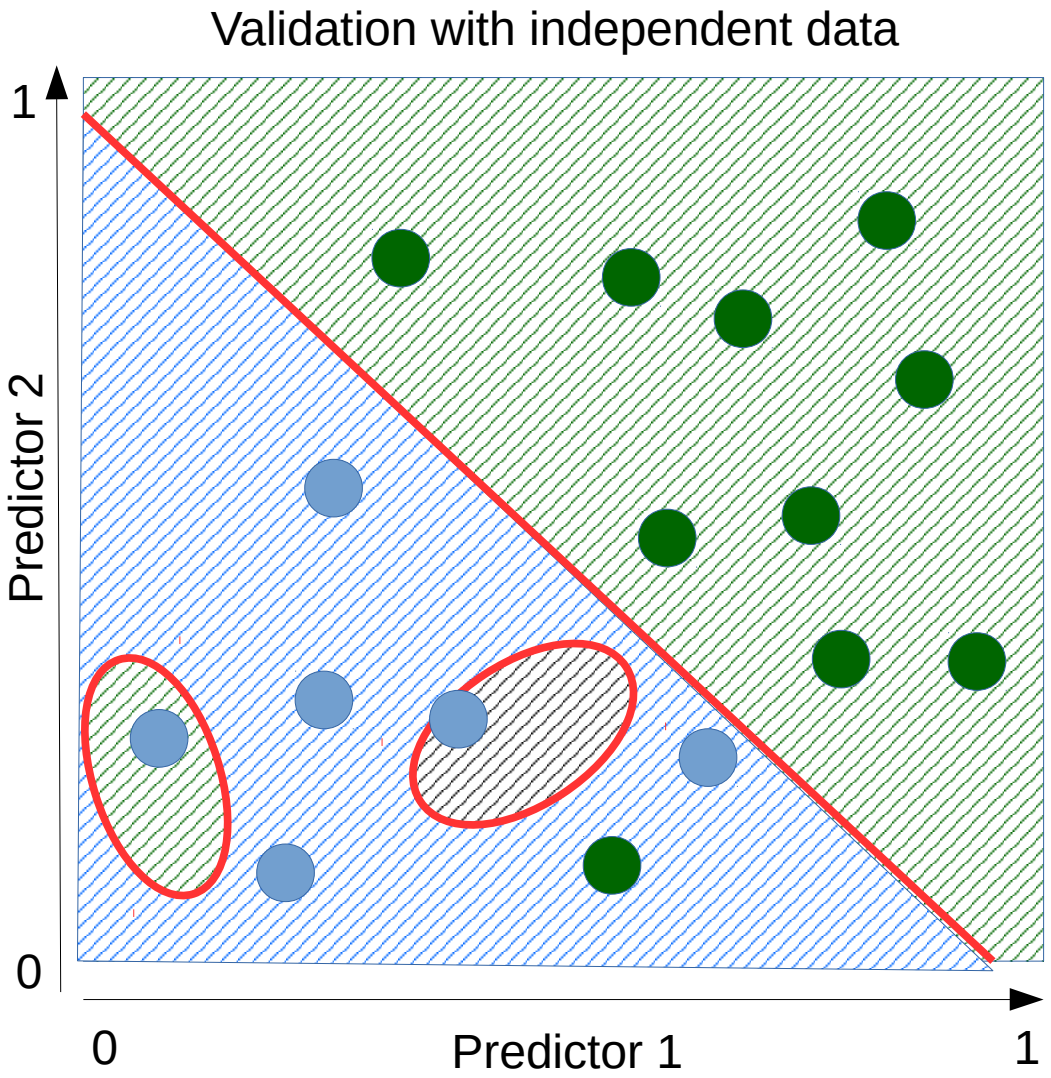
Step 5: Model validation – Problem of overfitting





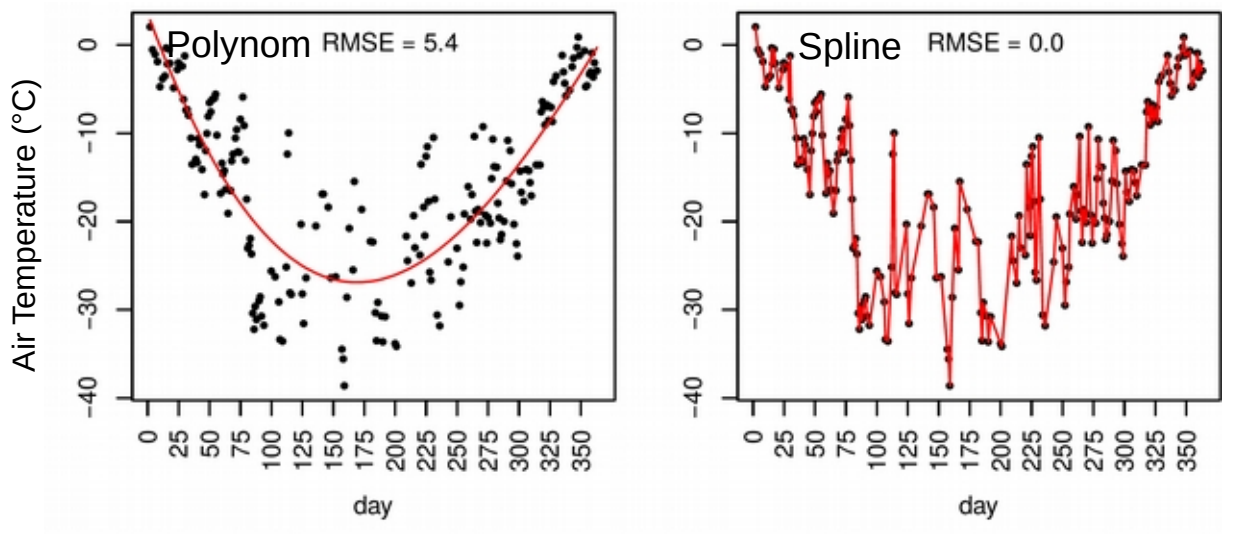
Step 5: Model validation – Problem of overfitting



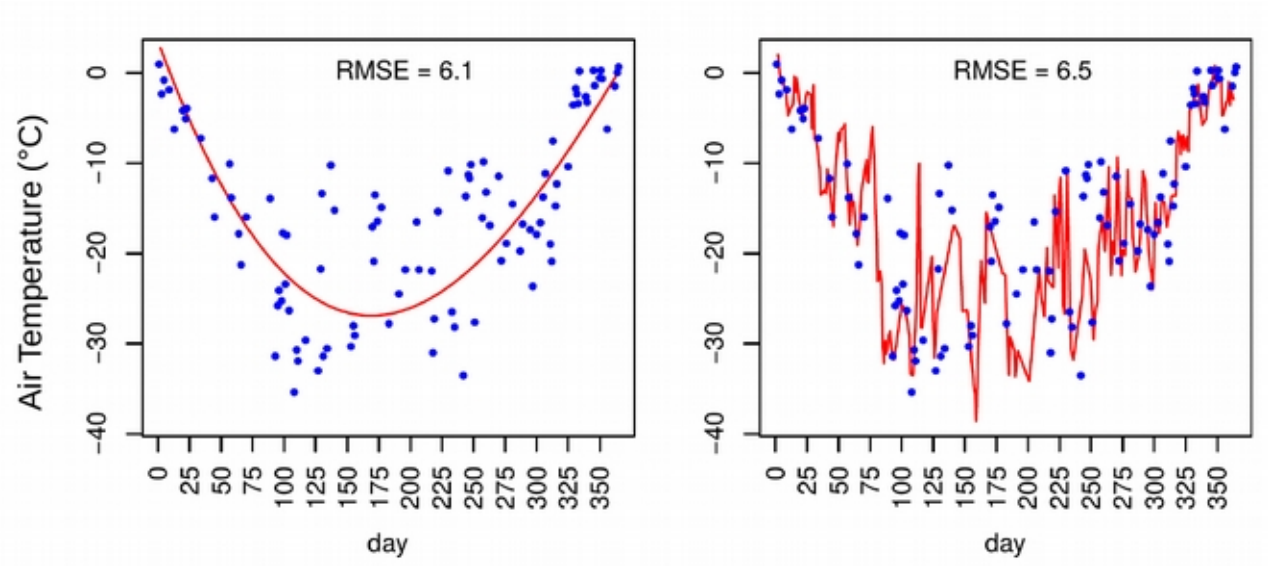


...But how good is the model?

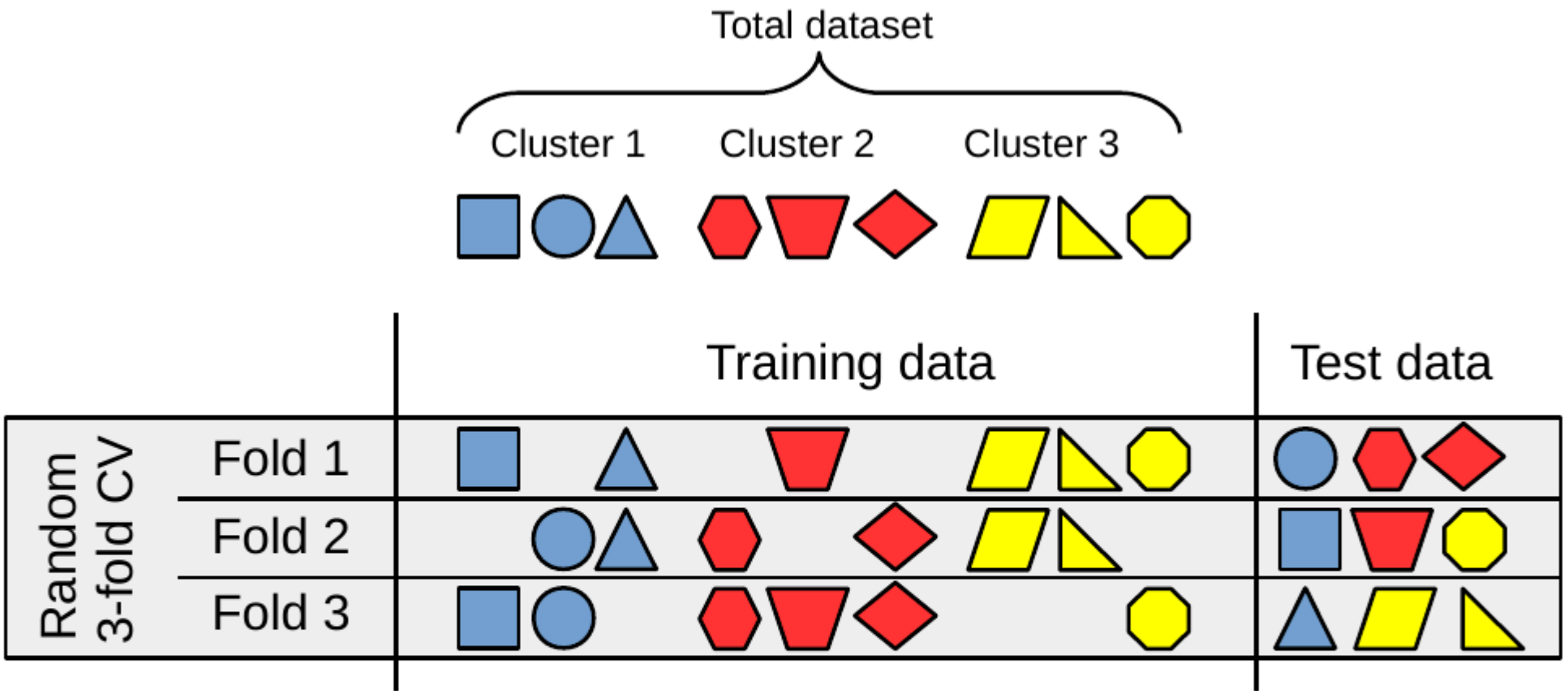
Model training



Model validation (1/3 of the data)



Step 5: Model validation – “Default” random cross-validation



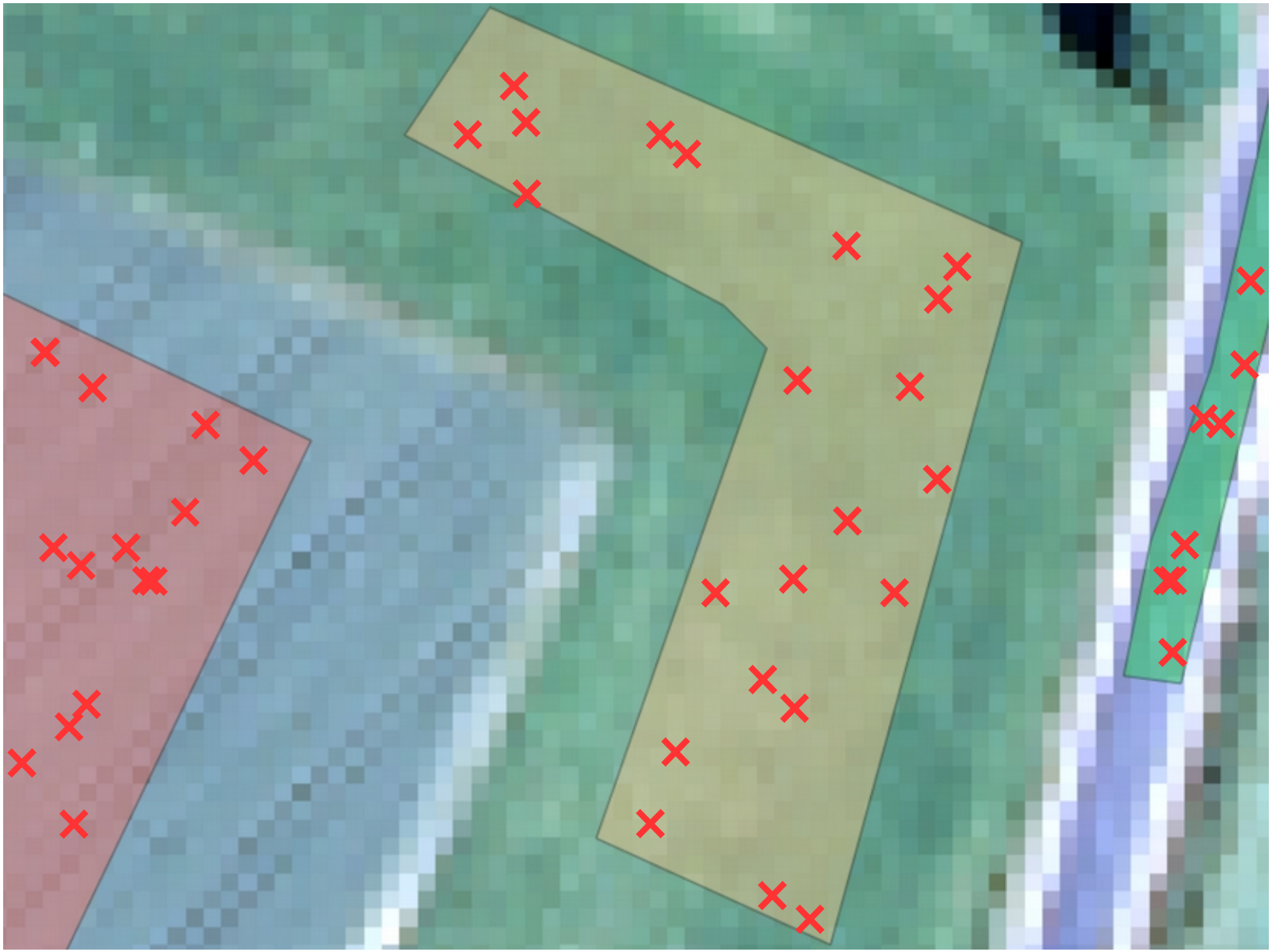
Step 5: Model validation – “Default” random cross-validation

How to do it in R

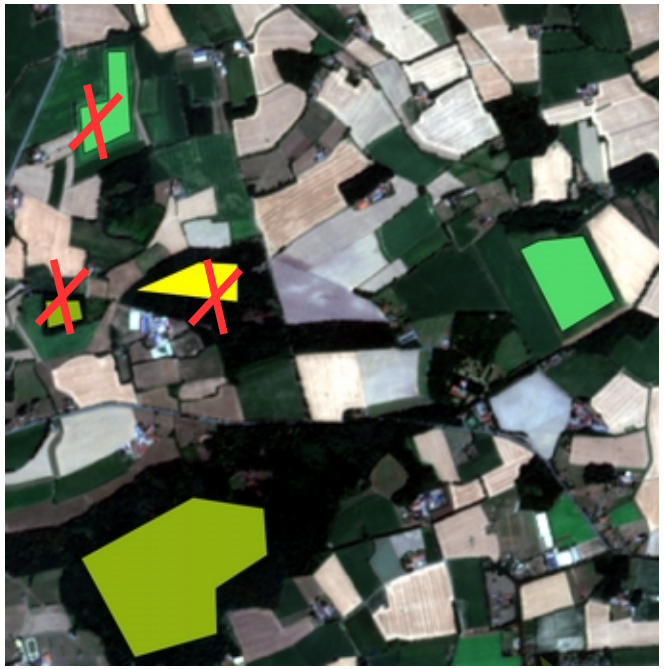
```
model <- train(predictors,
               response,
               method="rf",
               trControl=trainControl(method="cv"))
```

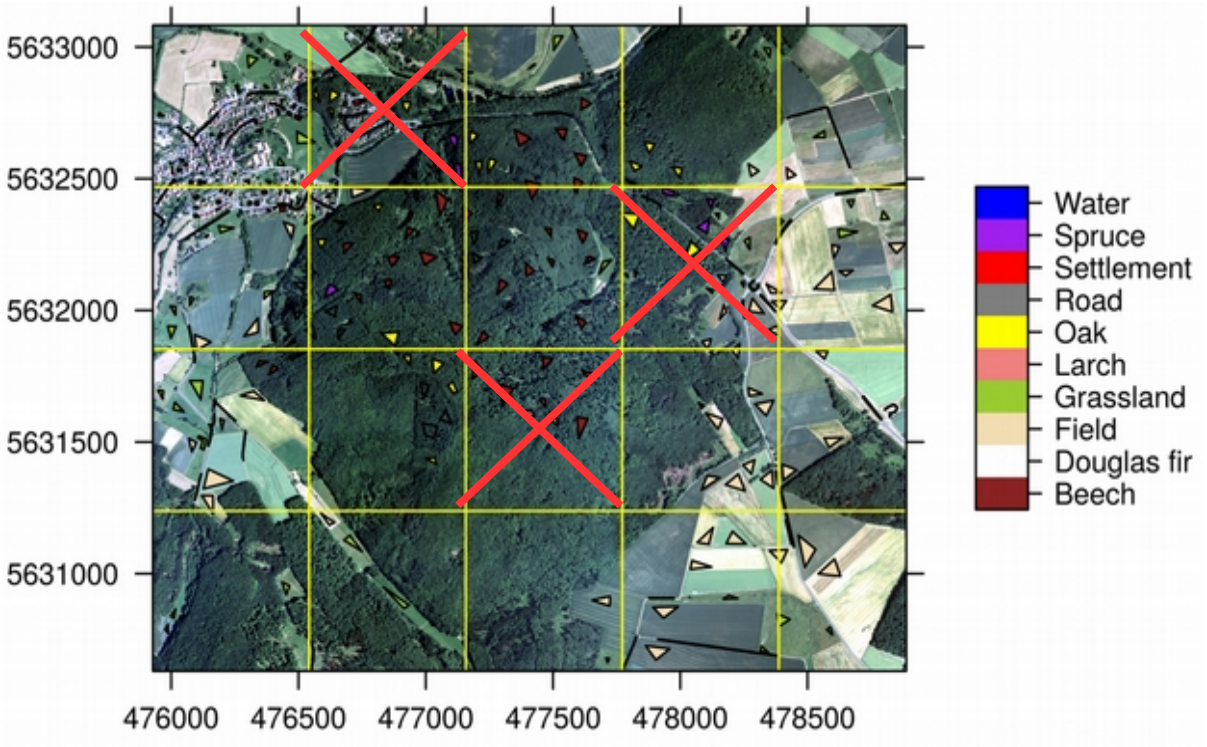
```
> model
Random Forest

3962 samples
  16 predictor
   9 classes: 'Feld_bepfl', 'Feld_unbepfl', 'Gewaesser', 'Laubwald', 'Mischwald', 'Na
nat_Feuchtw', 'Siedlungsgebiet', 'StrGruenland'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 3566, 3565, 3567, 3565, 3566, 3565, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.9512871  0.9426287
   9    0.9545617  0.9465151
  16    0.9507795  0.9420838

Kappa was used to select the optimal model using the largest value.
The final value used for the model was mtry = 9.
```

Step 5: Model validation – “Default” random cross-validation

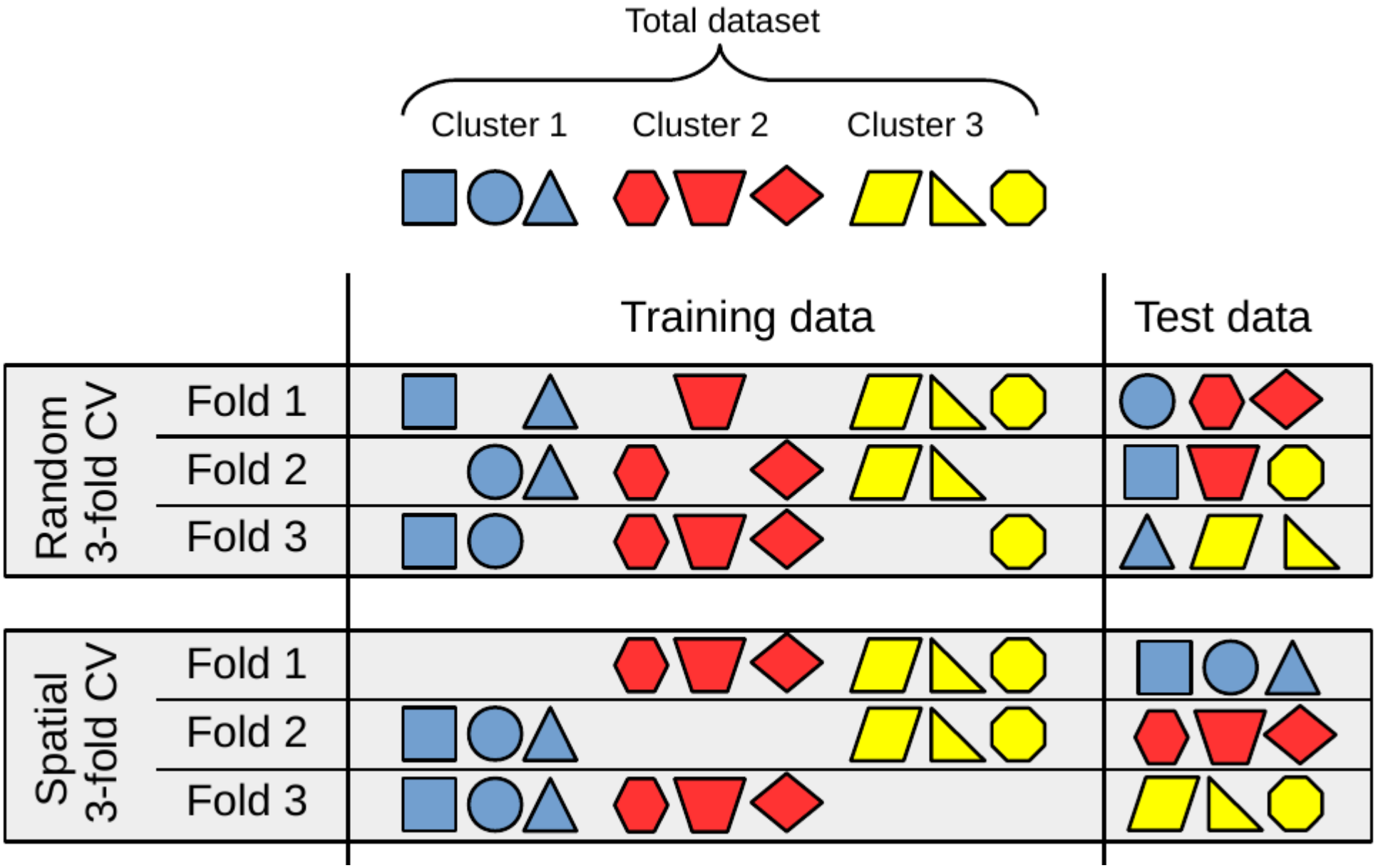


Step 5: Model validation – Spatial validation





Step 5: Model validation – Spatial validation



Step 5: Model validation – Spatial validation

How to do it in R

```
library(CAST)
indices <- CreateSpacetimeFolds(trainingData,
                                spacevar="Station")
model <- train(predictors,
               response,
               method="rf",
               trControl=trainControl(method="cv",
                                      index = indices$index))
```

PolygonID/
Spatial block/...

```
> model
Random Forest

3962 samples
  16 predictor
   9 classes: 'Feld_bepfl', 'Feld_unbepfl', 'Gewaesser', 'Laubwald', 'Mischwald', 'Nadelwald', 'Re
nat_Feuchtw', 'Siedlungsgebiet', 'StrGruenland'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 3650, 3626, 3483, 3785, 3768, 3791, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.6110833  0.4846005
   9    0.6590675  0.5467704
  16    0.6568454  0.5481304

Kappa was used to select the optimal model using the largest value.
The final value used for the model was mtry = 16.
```

Time for practice...

- Task: Identify the invasive gorse on the Banks Peninsula in New Zealand based on Sentinel satellite data
- Technical Focus: Random Forest model training and spatial prediction
- Material: LUCmodelling.html in the Summer School GitHub Repository HannaMeyer/ML4GenTree
- Data: Sentinel spectral channels (sentinel2017.grd) and polygon containing training sites: (trainingSites.shp)