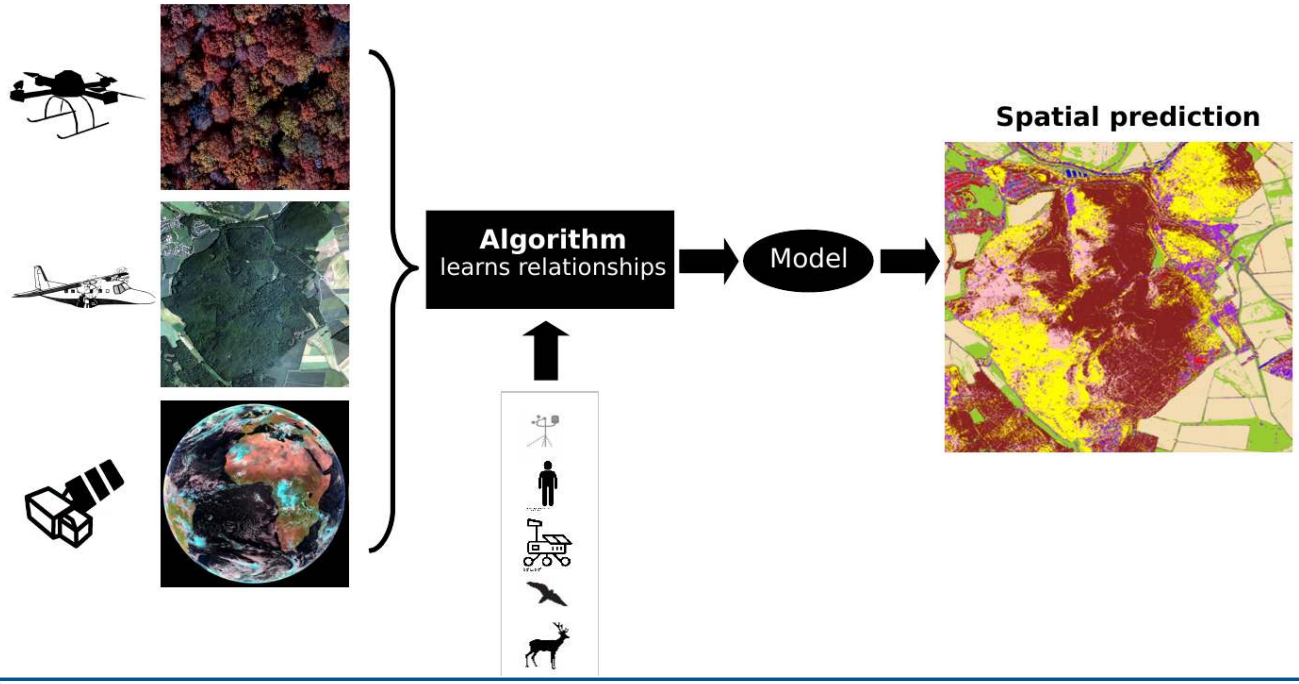
Machine learning for earth observation:
Mapping the „Area of Applicability“ of spatial prediction models

Part II: Practice

Hanna Meyer
Remote Sensing & Spatial Modelling,
Institute of Landscape Ecology, WWU Münster



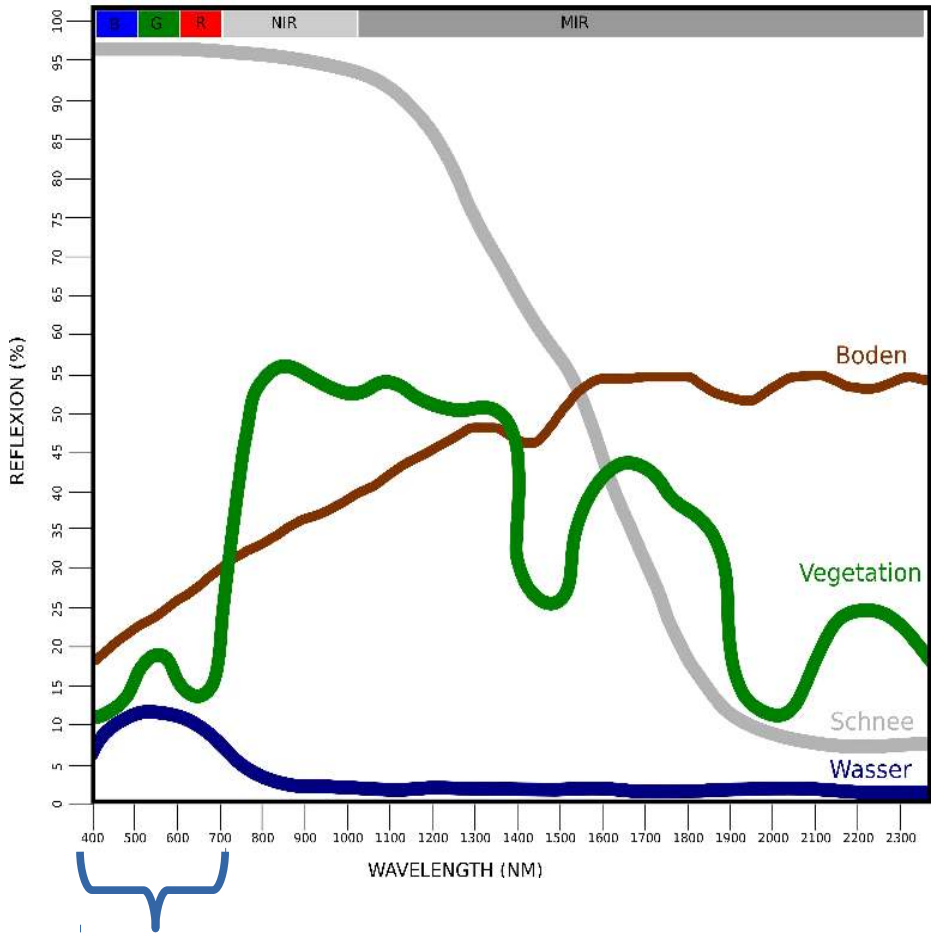
Aim of the workshop

- Getting to know the basic workflow: Machine learning for remote sensing applications
- Example: land cover classification
- Analysing the area of applicability of a prediction model
- Assessing the transferability of a prediction model

Next ~30min: Introduction, then guided hands-on

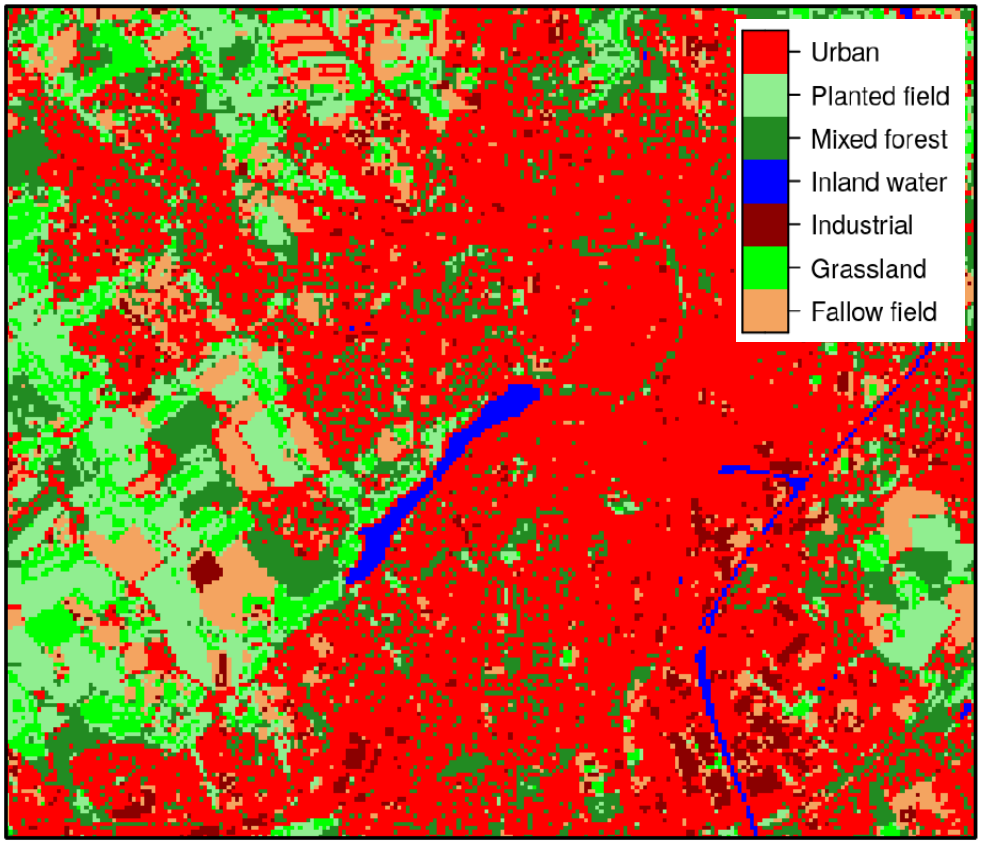
Idea of a land cover classification

- Land cover types differ in their reflection properties
- Apply Statistical methods to infer land use classes from reflection properties
- Idea: Learning from known data and applying the model to the entire landscape



Visible part

Case Study for the tutorial



Satellite data (selection)

Platform/Sensor	Spatial resolution (m)	Temporal resolution	Availability
Landsat MSS	79	16 days	since 1972
Landsat TM	30	16 days	since 1982
Landsat ETM+	30	16 days	since 1999
Landsat 8 (OLI)	30	16 days	since 2013
Sentinel-2	10	5/10 days	since 2014
MODIS Terra/Aqua	250-1000	4 per day	since 2000
Meteosat Second Generation	3000	15 minutes	since 2002

Sentinel-2 data

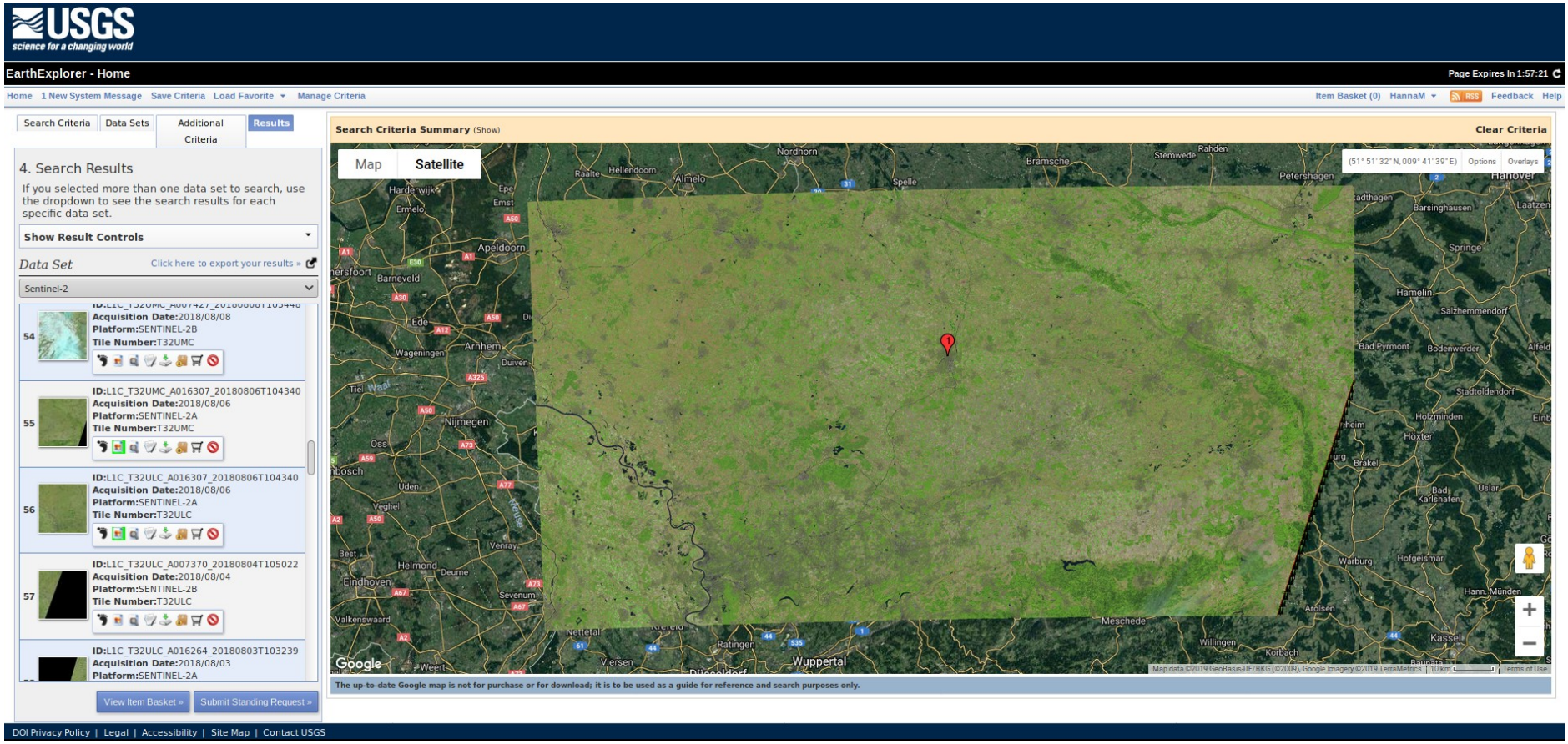
Spectral bands for the Sentinel-2 sensors[10]

Sentinel-2 bands	Sentinel-2A		Sentinel-2B		
	Central wavelength (nm)	Bandwidth (nm)	Central wavelength (nm)	Bandwidth (nm)	Spatial resolution (m)
Band 1 – Coastal aerosol	442.7	21	442.2	21	60
Band 2 – Blue	492.4	66	492.1	66	10
Band 3 – Green	559.8	36	559.0	36	10
Band 4 – Red	664.6	31	664.9	31	10
Band 5 – Vegetation red edge	704.1	15	703.8	16	20
Band 6 – Vegetation red edge	740.5	15	739.1	15	20
Band 7 – Vegetation red edge	782.8	20	779.7	20	20
Band 8 – NIR	832.8	106	832.9	106	10
Band 8A – Narrow NIR	864.7	21	864.0	22	20
Band 9 – Water vapour	945.1	20	943.2	21	60
Band 10 – SWIR – Cirrus	1373.5	31	1376.9	30	60
Band 11 – SWIR	1613.7	91	1610.4	94	20
Band 12 – SWIR	2202.4	175	2185.7	185	20

https://en.wikipedia.org/wiki/Sentinel-2

Getting satellite data

- E.g. using the Earth Explorer
- Or automatic download via getSpatialData package



→ https://earthexplorer.usgs.gov/

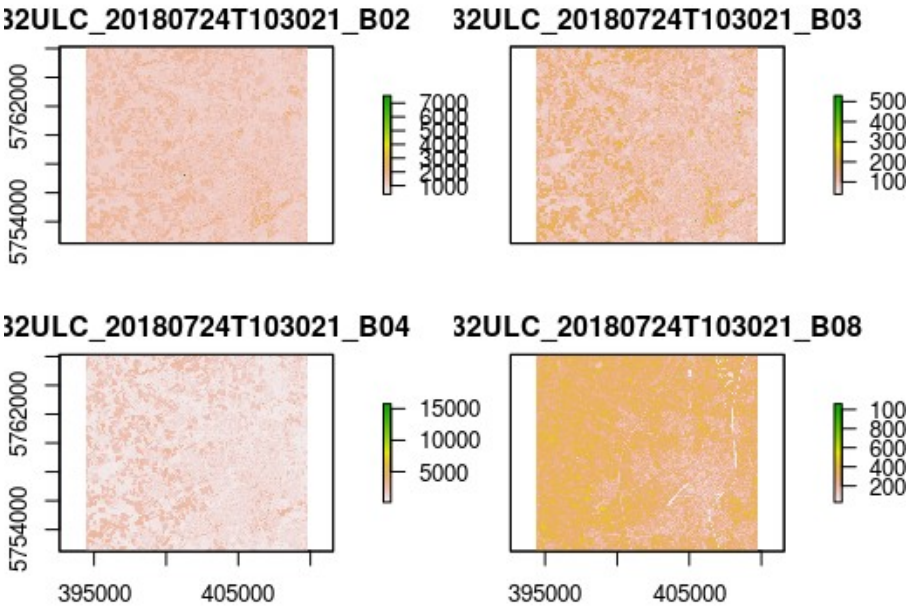
Load satellite data in R

- Packages for raster data: raster, stars, terra
- Here we will use the raster package but stars version available (see Edzer's PR)

How to import satellite data into R

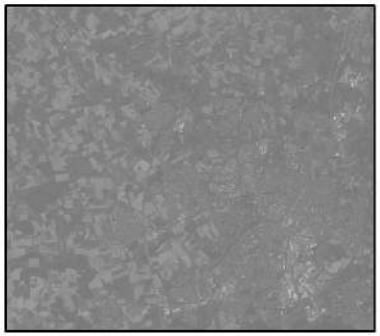
```
library(raster)
sen <- stack("blue.tif","green.tif",
             "red.tif","NIR.tif")
plot(sen)
```

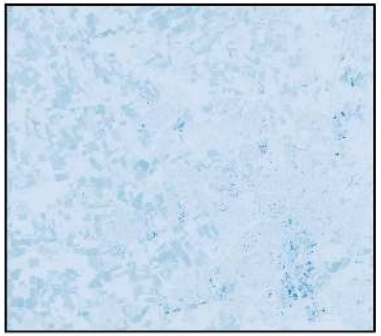
And more channels!



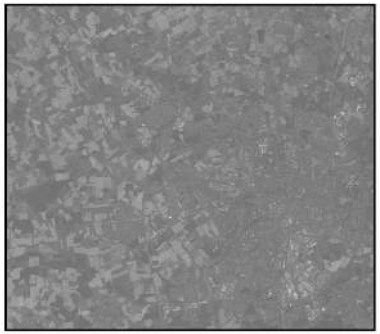
How do we get the “color” in the satellite data?

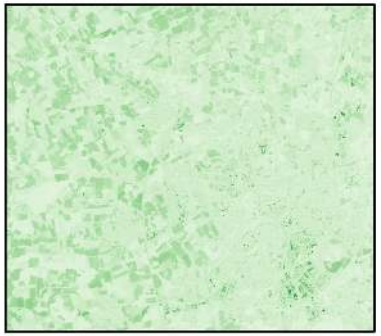
blue



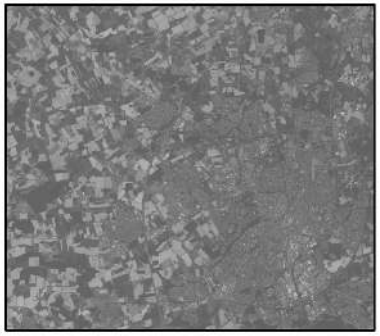


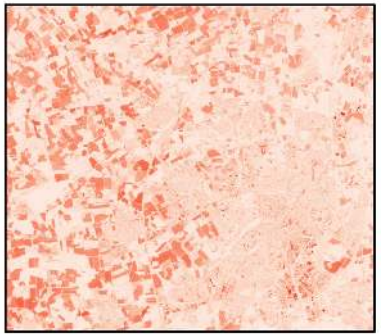
green





red

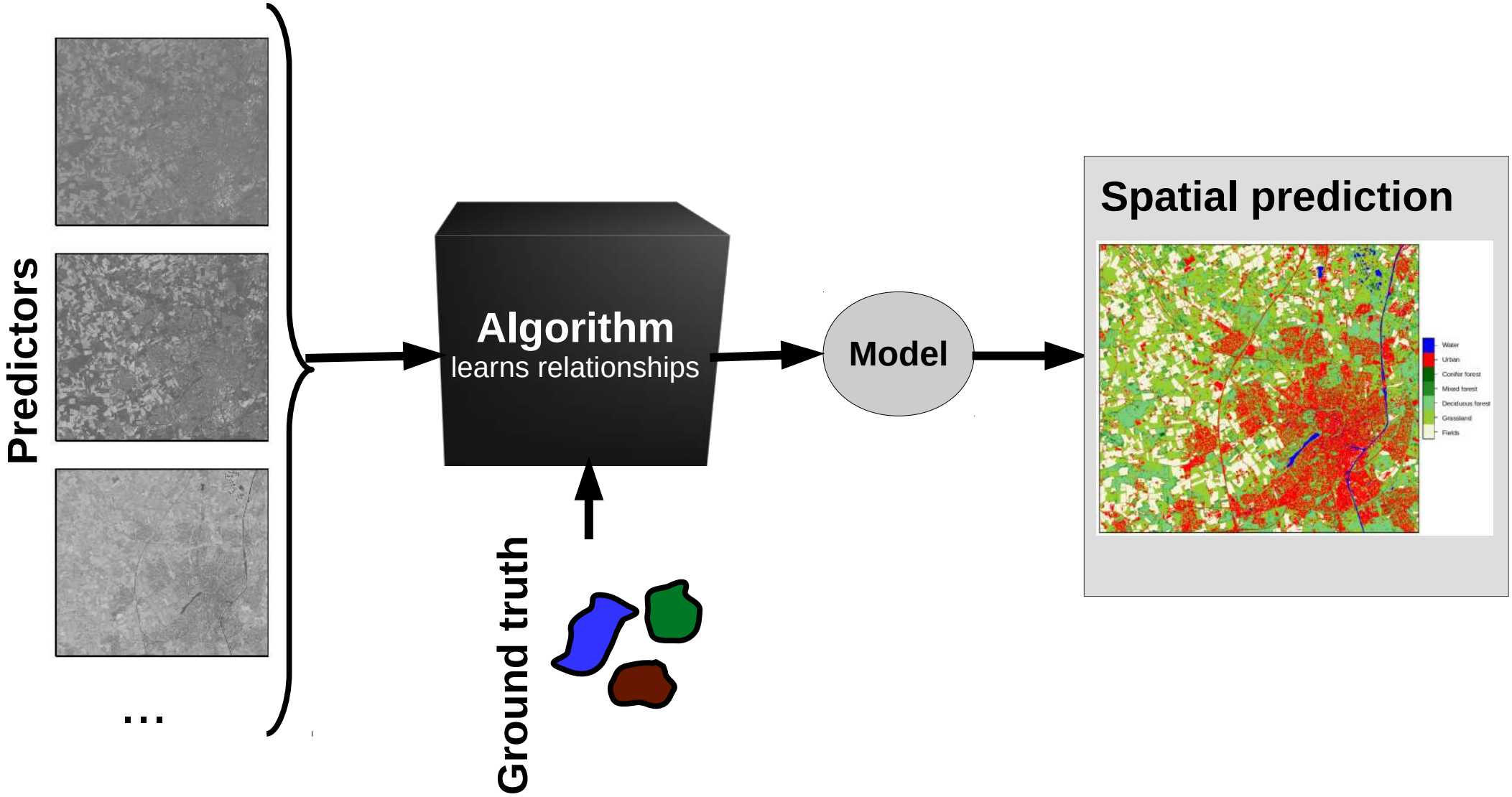




How to do a color composite in R

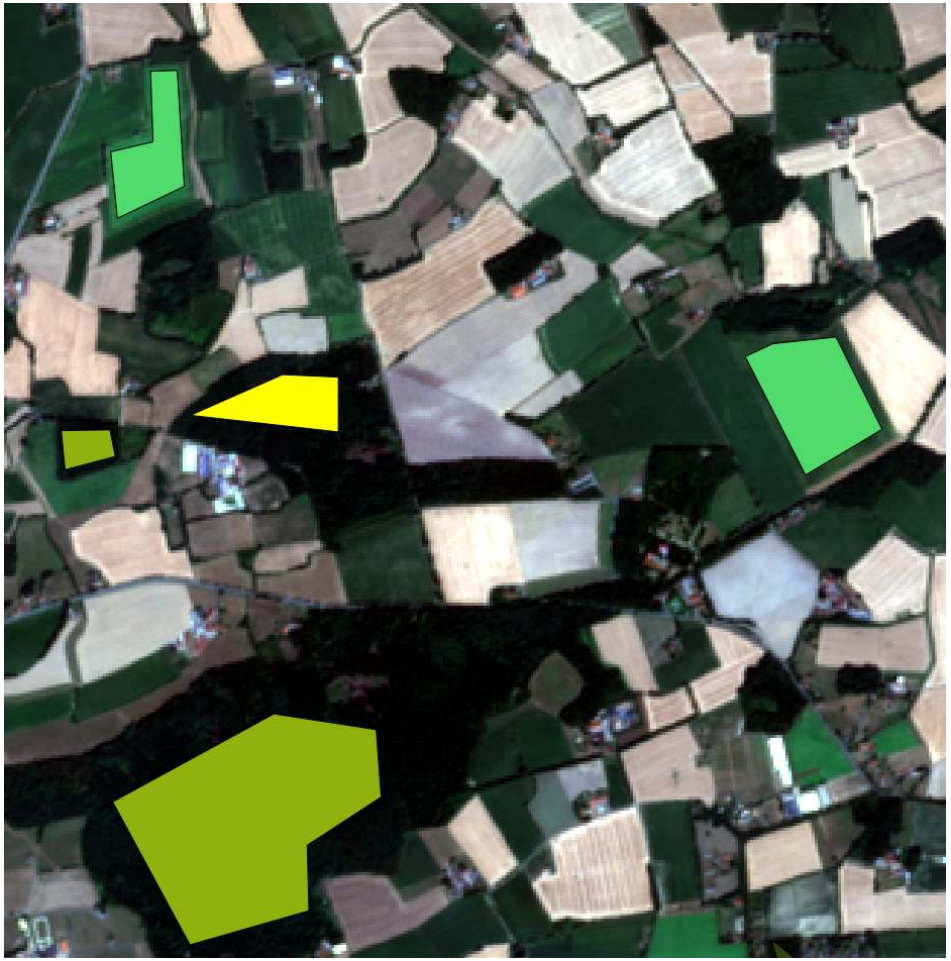
```
sen <- stack("blue.tif","green.tif",
             "red.tif","NIR.tif")
plotRGB(sen,r=3,g=2,b=1,stretch="lin")
```

How to use the spectral properties to classify land cover?



How to use the spectral properties to classify land cover?

Reference data are required

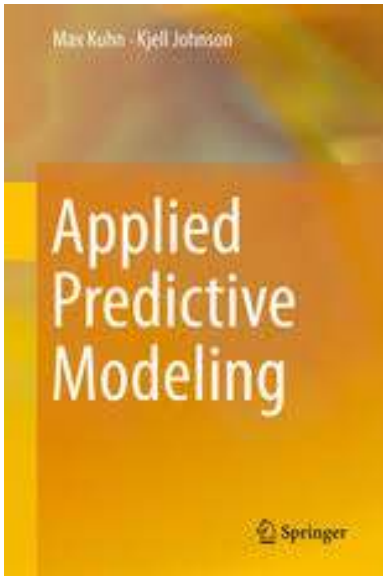


- Training data from field work, expert knowledge, existing databases,…
- Usually polygons that are taken using a GPS or manually digitized
- In this case study: digitized (in QGIS)

Machine learning 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

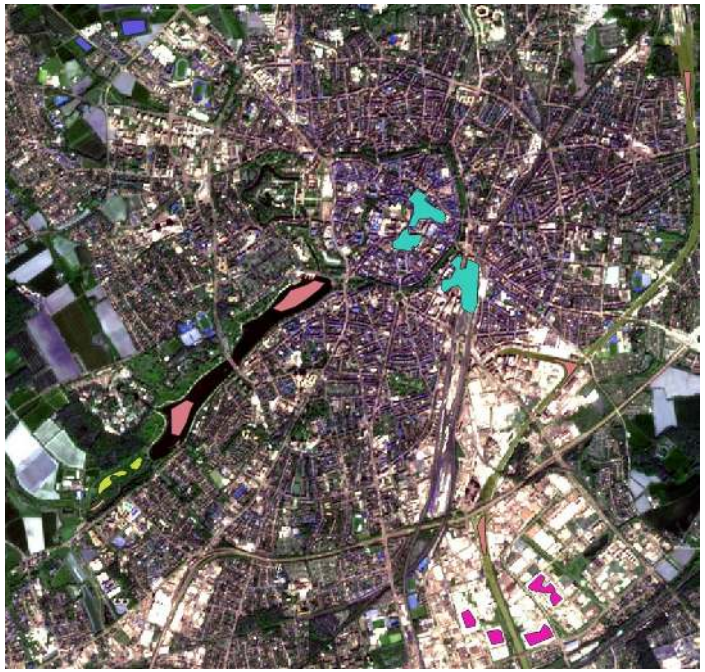
For today's session



Step 1: Model training in R
”Default model”

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")
```

Random Forest used here as Machine learning algorithm

Step 1: Model training in R
”Default model”: Example of results

Variables	Validation	Accuracy	Kappa
all	random	>0.99	>0.99

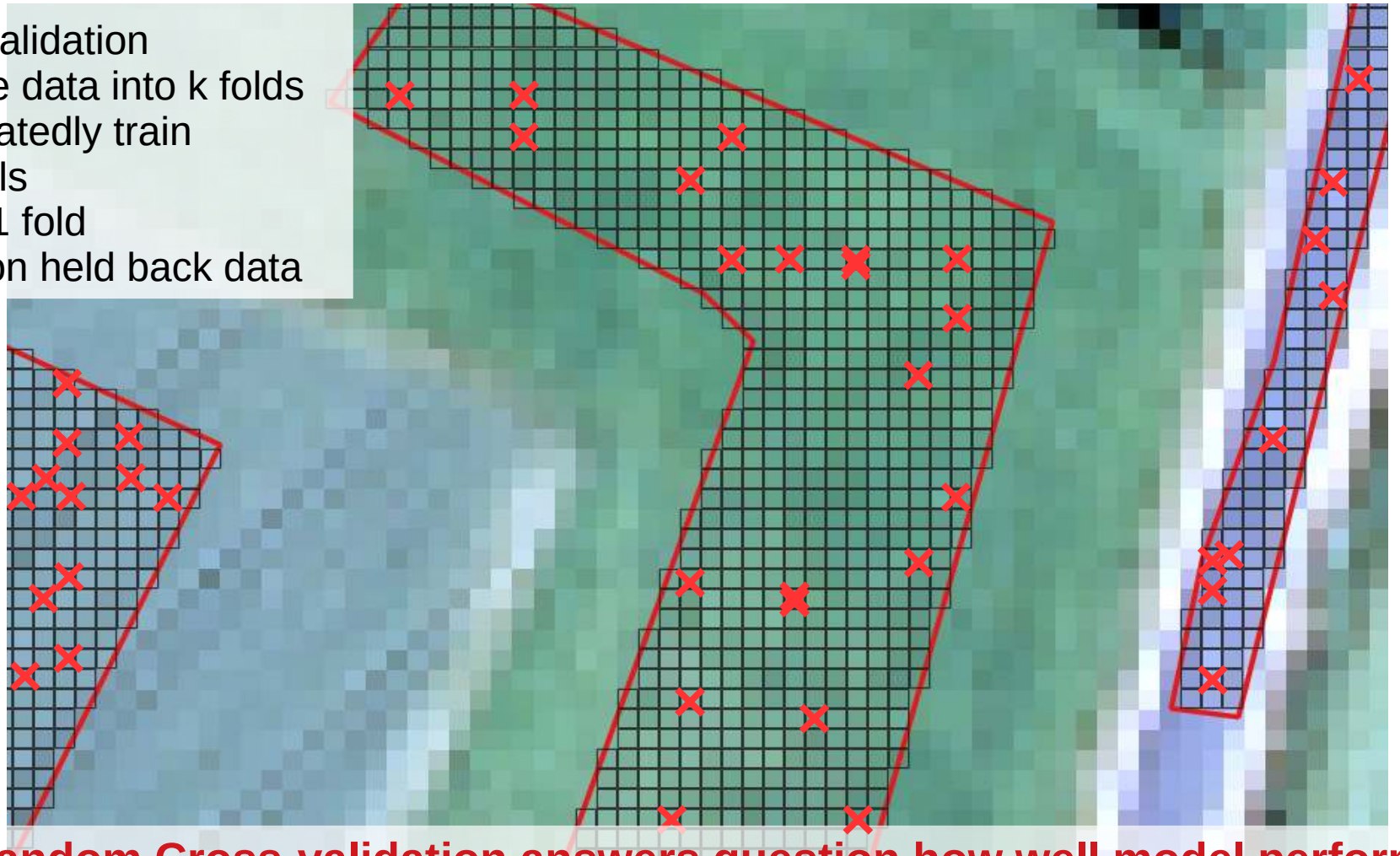
Perfect prediction?

Step 1: Model training in R

...however spatial dependencies need to be taken into account

Cross-validation

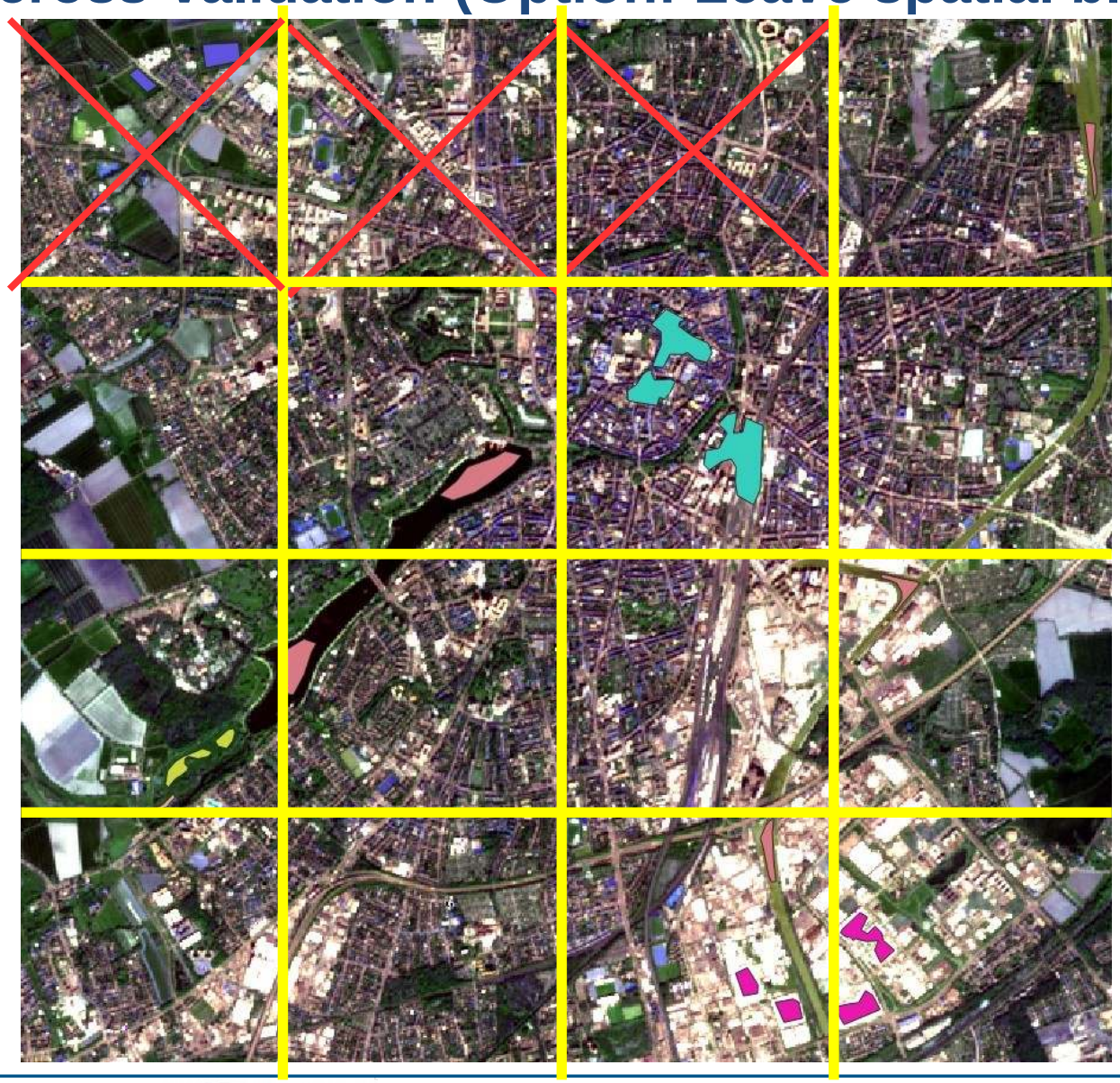
- Divide data into k folds
- Repeatedly train models on k-1 fold
- Test on held back data



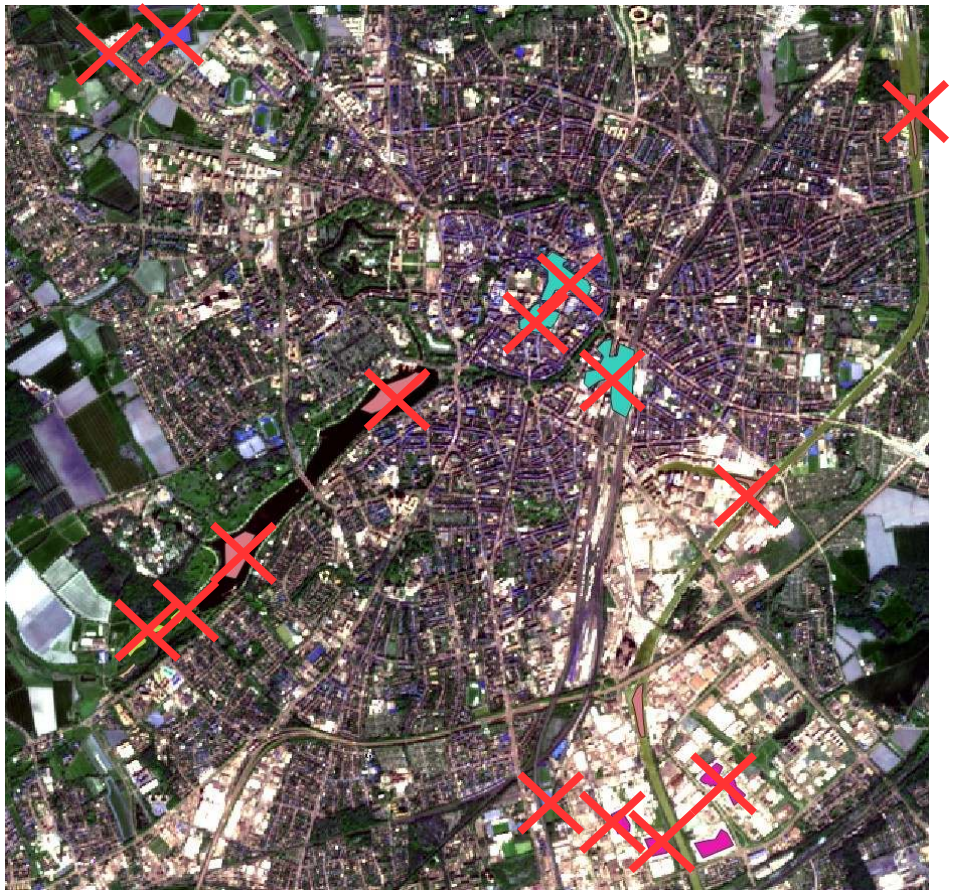
Random Cross-validation answers question how well model performs on very similar locations

Step 1: Model training in R

Spatial cross-validation (Option: Leave spatial block out)



Step 1: Model training in R
Spatial cross-validation (Option: Leave group of polygons out)



How to do it in R

"**C**aret **A**pplications for **S**tatio-**t**emporal models"

```
library(CAST)

indices <- CreateSpacetimeFolds(trainDat,
                                spacevar = "PolygonID",
                                k=3,
                                class="Label")

ctrl <- trainControl(method="cv",
                     index = indices$index)

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

Step 1: Model training in R
Spatial cross-validation (Leave group of polygons out)

Variables	Validation	Accuracy	Kappa
all	random	>0.99	>0.99
all	spatial	**0.68**	**0.61**

- Random cross-validation performance tells how well we can reproduce training data
- Spatial cross-validation tells us how well we can make predictions
- The spatial performance is usually lower, but this no reason to use the random performance!!!

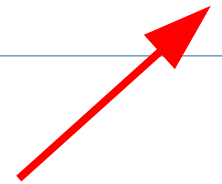
Step 1: Model training in R
Use spatial cross-validation for the entire modelling process

Spatial cross-validation should be used for the entire modelling process

- For variable selection
- Hyperparameter tuning
- Error assessment

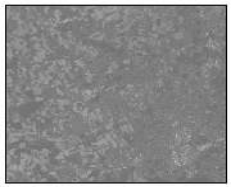
How to do it in R

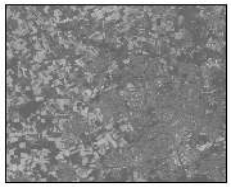
```
model <- ffs(predictors,
             response,
             method="rf",
             tuneLength=5,
             trControl=ctrl)
```

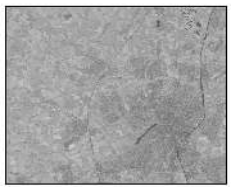


Important!!!
Use spatial folds here!!!

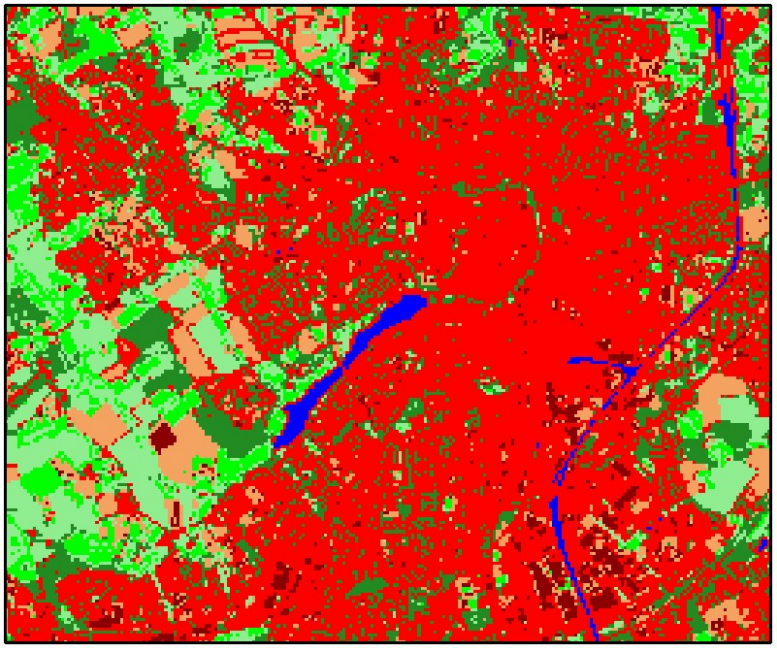
Step 2: Model prediction in R







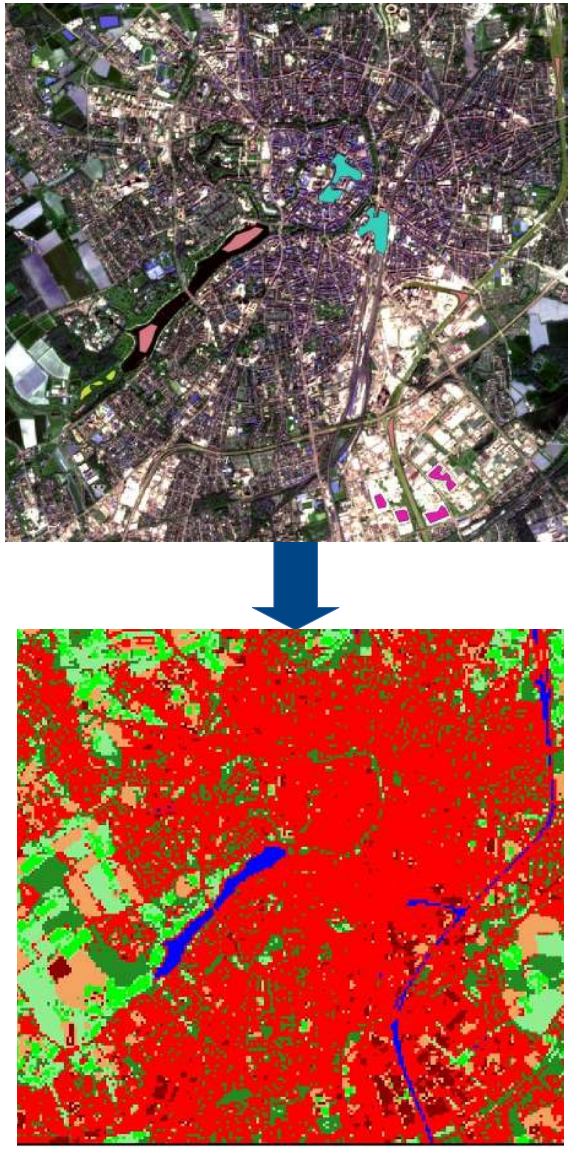
... + trained model =



How to do it in R

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

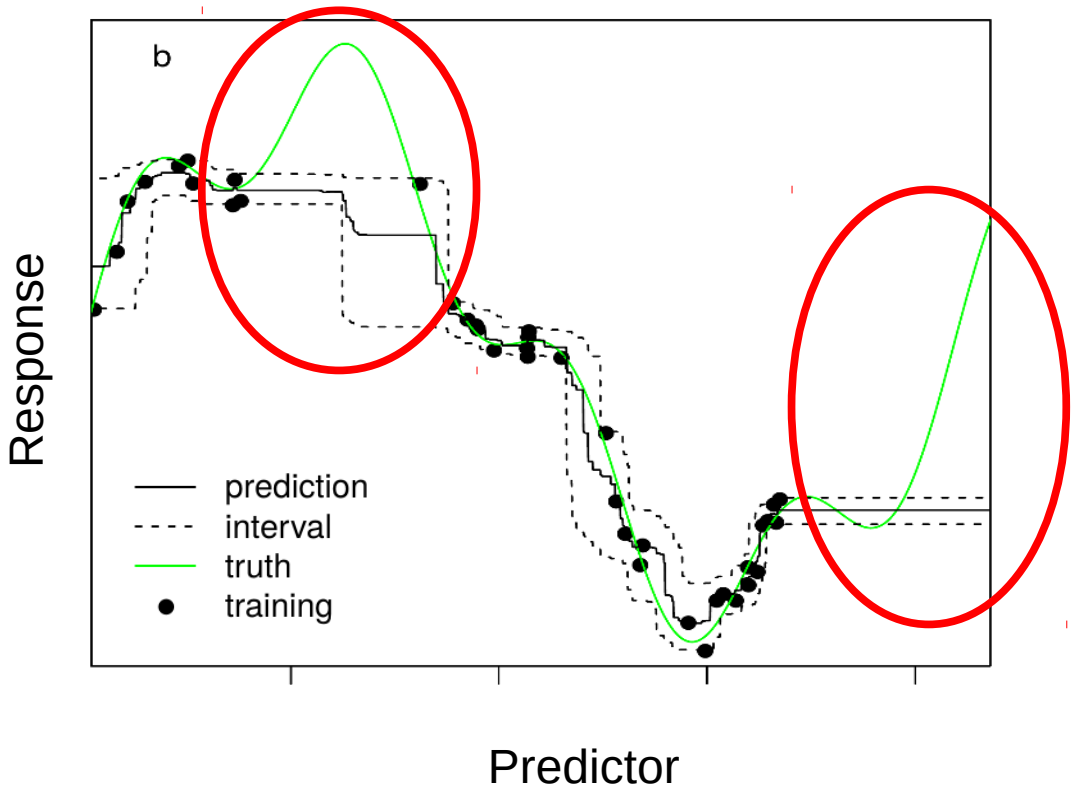
Step 3: Estimating the area of applicability



Can we really apply the model to the entire area?

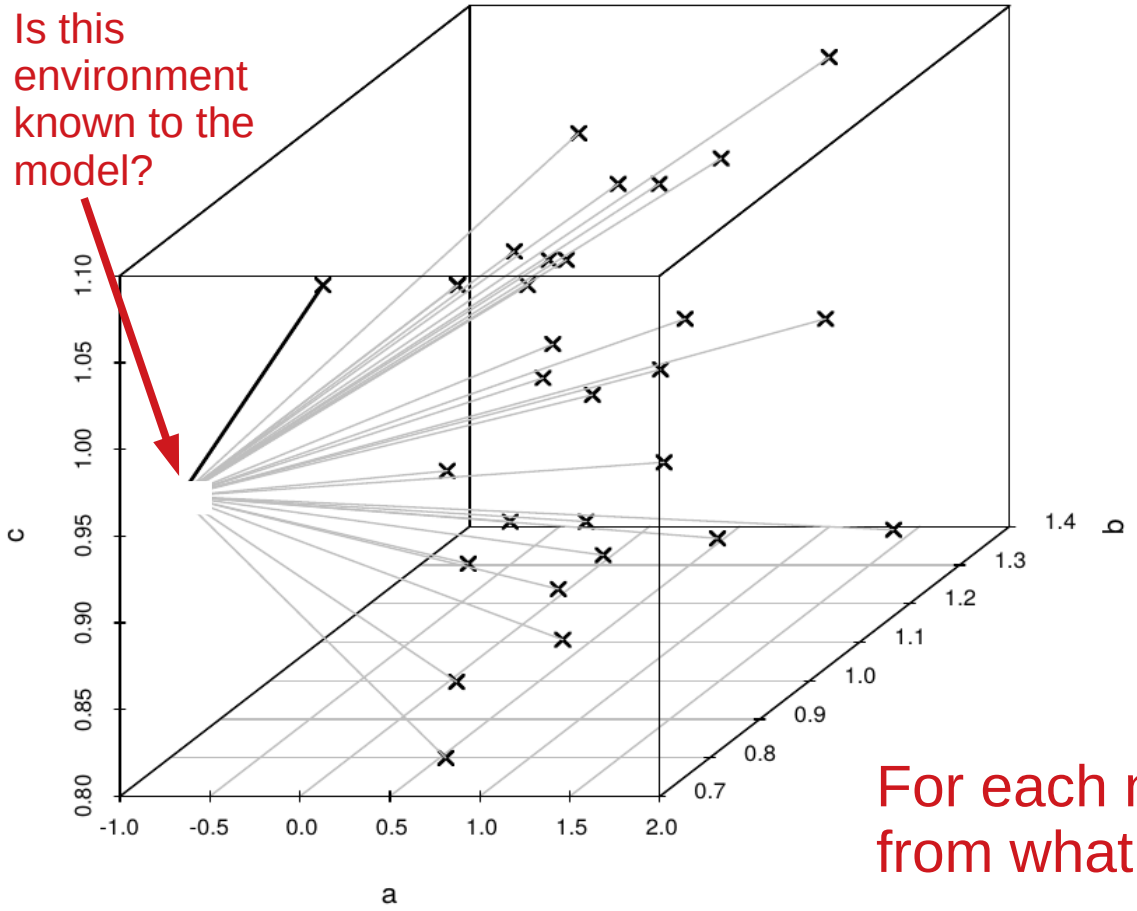
- Transfer to new space required
- New space might differ in environmental properties
- But what if the algorithm has never seen such properties?

Step 3: Estimating the area of applicability



- Machine learning can fit very complex relationships.
- But gaps in predictor space are problematic (the model has no knowledge about these areas!)
- A measure for “unknown space” is needed to estimate the area of applicability of a model!

Step 3: Estimating the area of applicability



Meyer&Pebesma (under review)

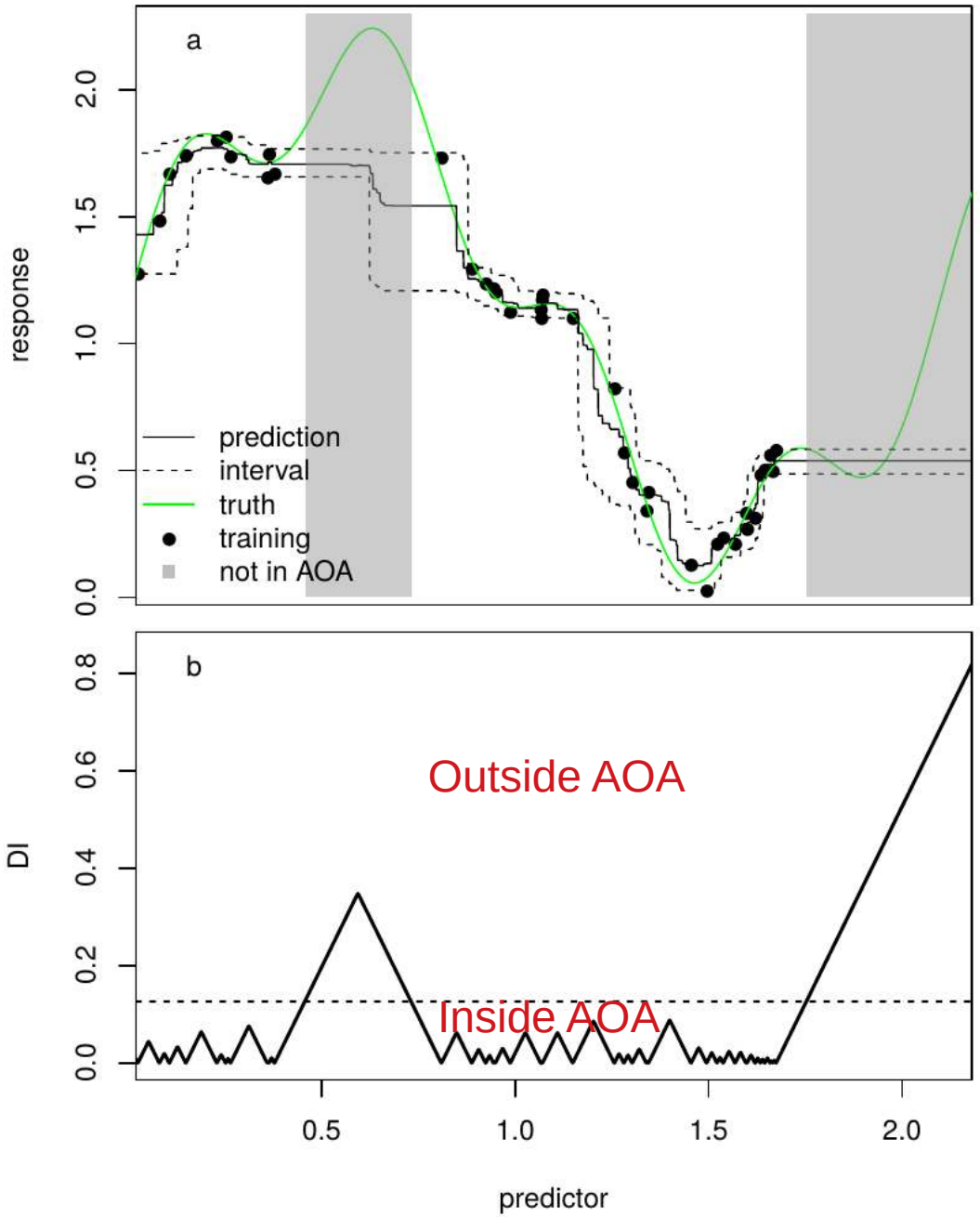
- Unknown space: Environmental conditions that are very different from the training locations
- Suggestion: Distance in (weighted) predictor space as measure for unknown space

For each new location/pixel: how distant is it from what the algorithms has seen?

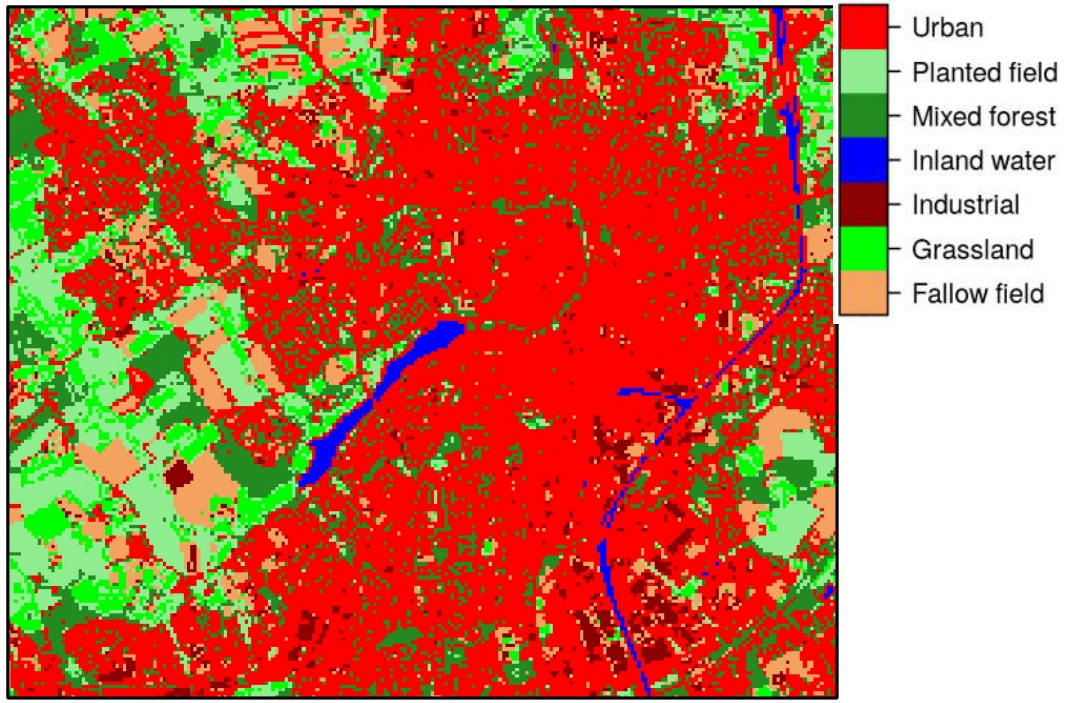
Step 3: Estimating the area of applicability

- Definition: The area for which, on average, the cross-validation error of the model applies

- Estimated using a threshold on the DI
- Threshold = DI of cross-validated training data (95% quantile)
- DI < threshold = inside AOA
  DI > threshold = outside AOA



Step 3: Estimating the area of applicability



Technically we can make predictions for the entire area. But they only make sense if the predictor properties are known to the model

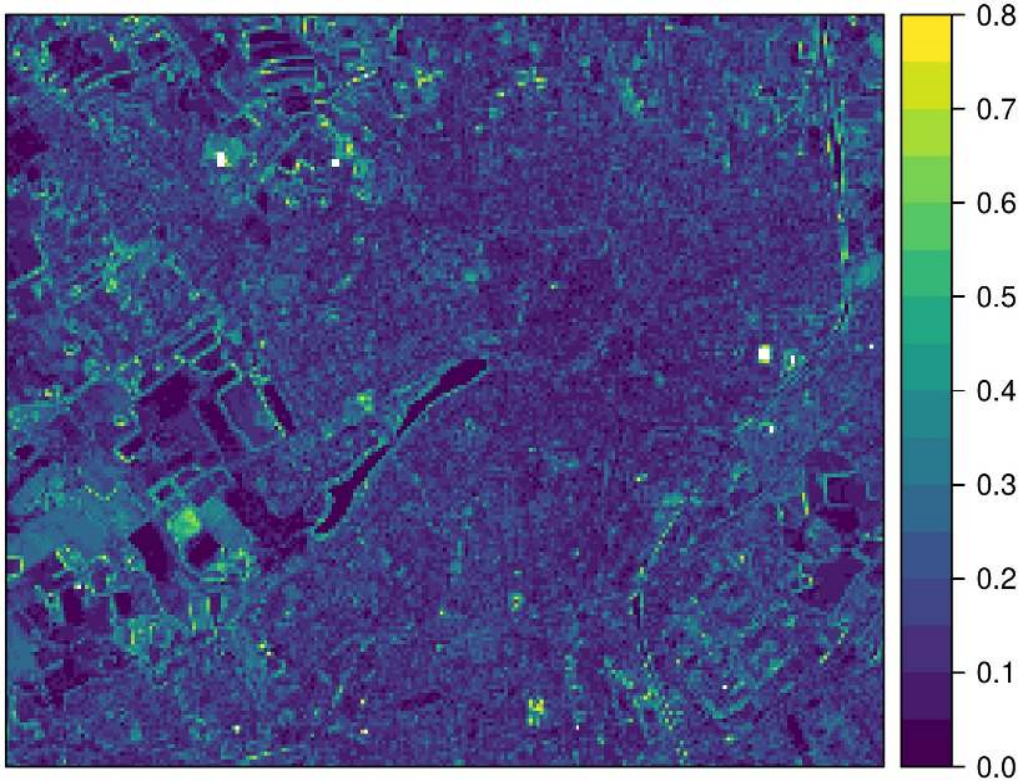
→ Estimate the area of applicability of the model

How to do it in R

```
AOA <- aoa(pred_sp,model)
```

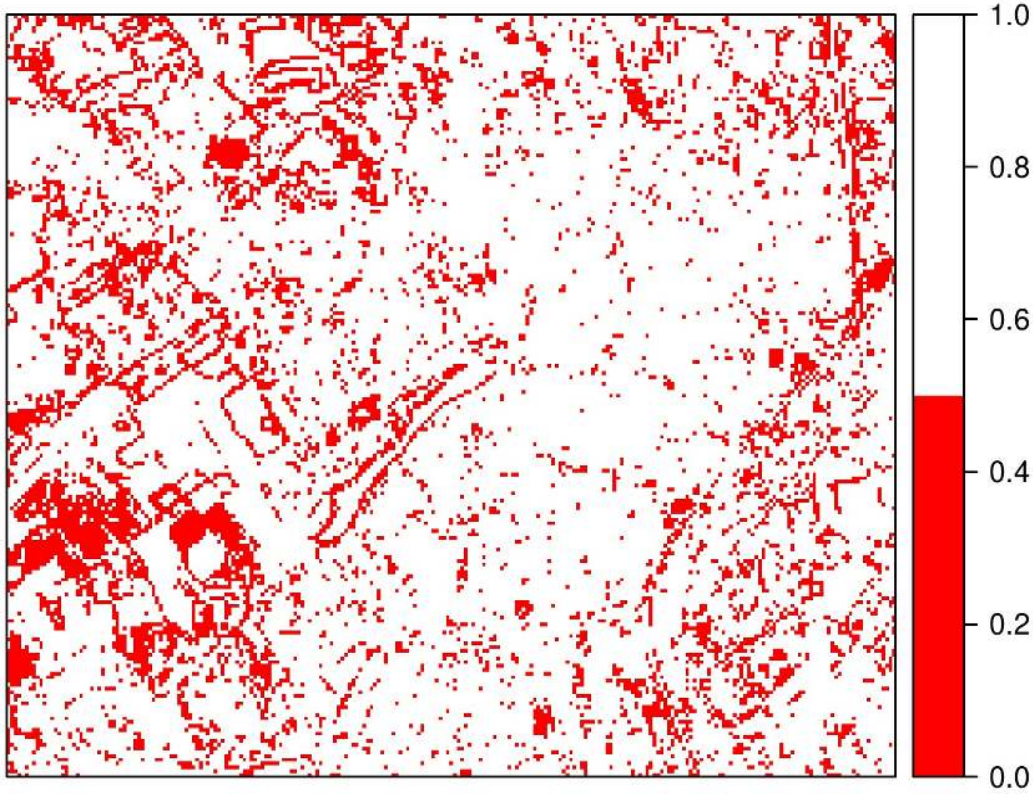
Step 3: Estimating the area of applicability
Interpretation of the AOA

Dissimilarity Index



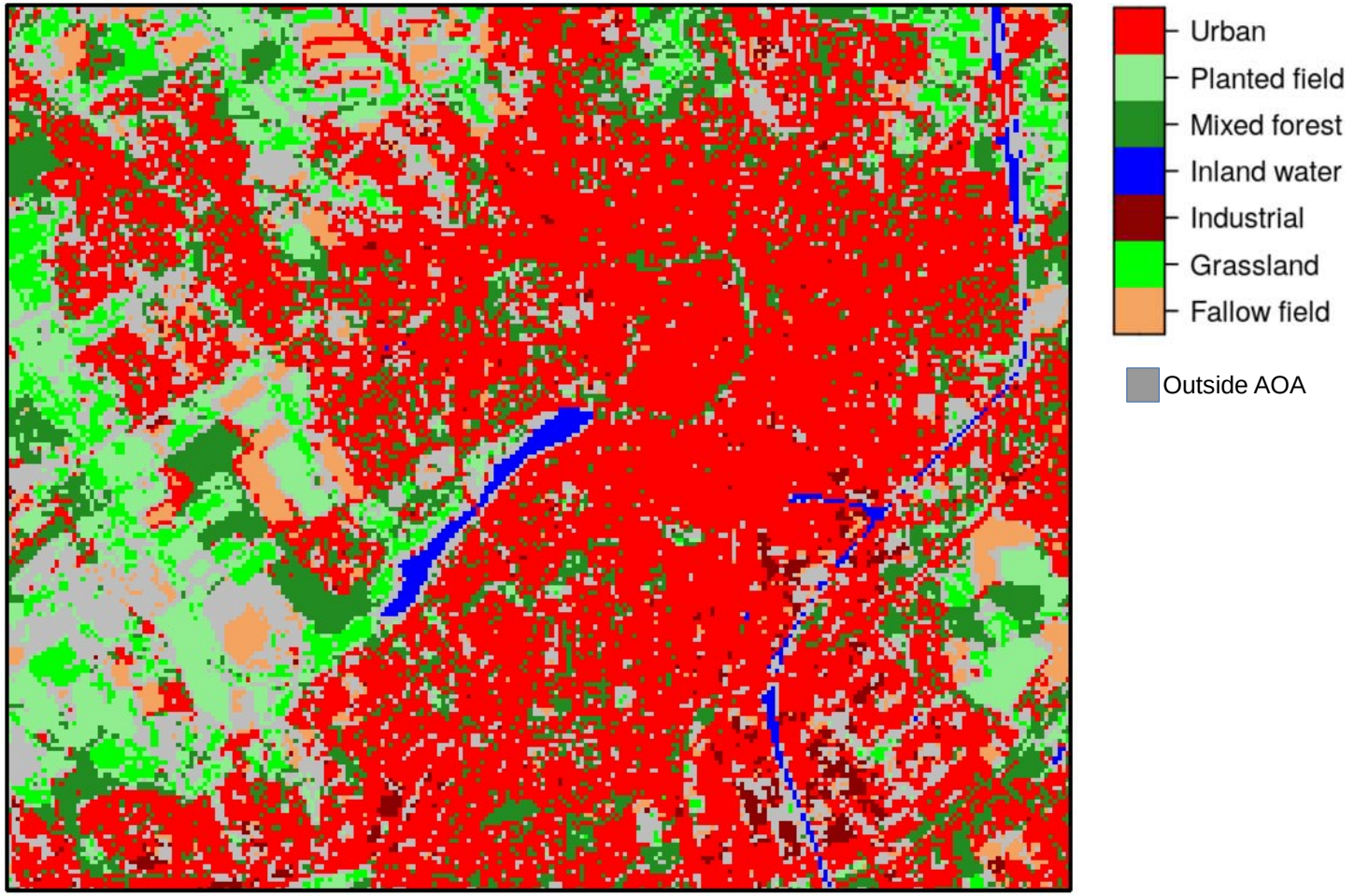
Can take values 0 to ∞

Area of Applicability



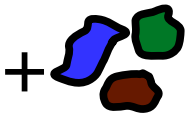
Can take values 0 = not in AOA
and 1= inside AOA

Step 3: Estimating the area of applicability
Interpretation of the AOA



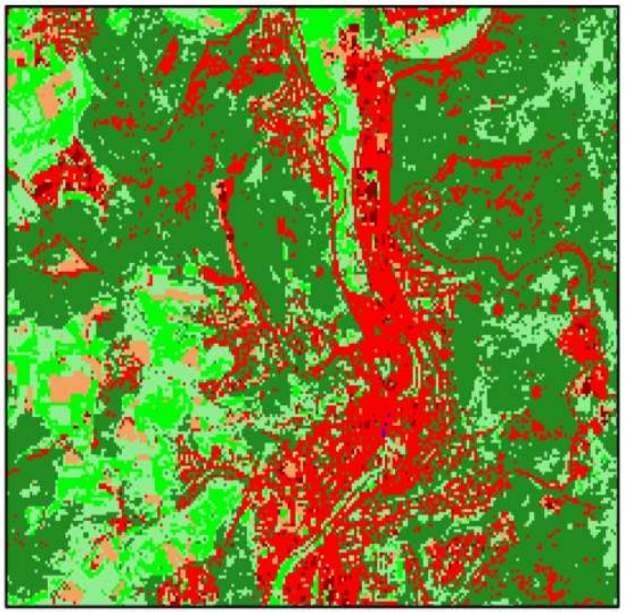
Step 4 (optional): Model transfer

Can we apply the trained model to make predictions for an other area?



Model

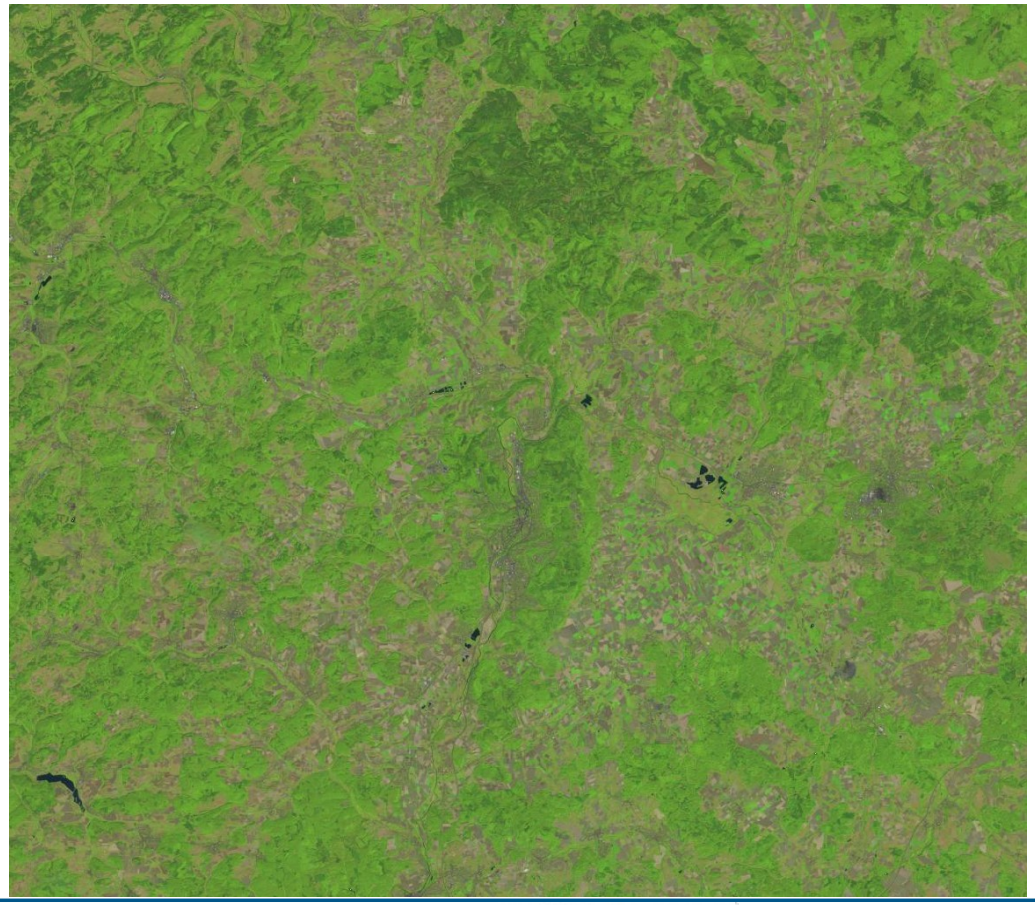
Prediction



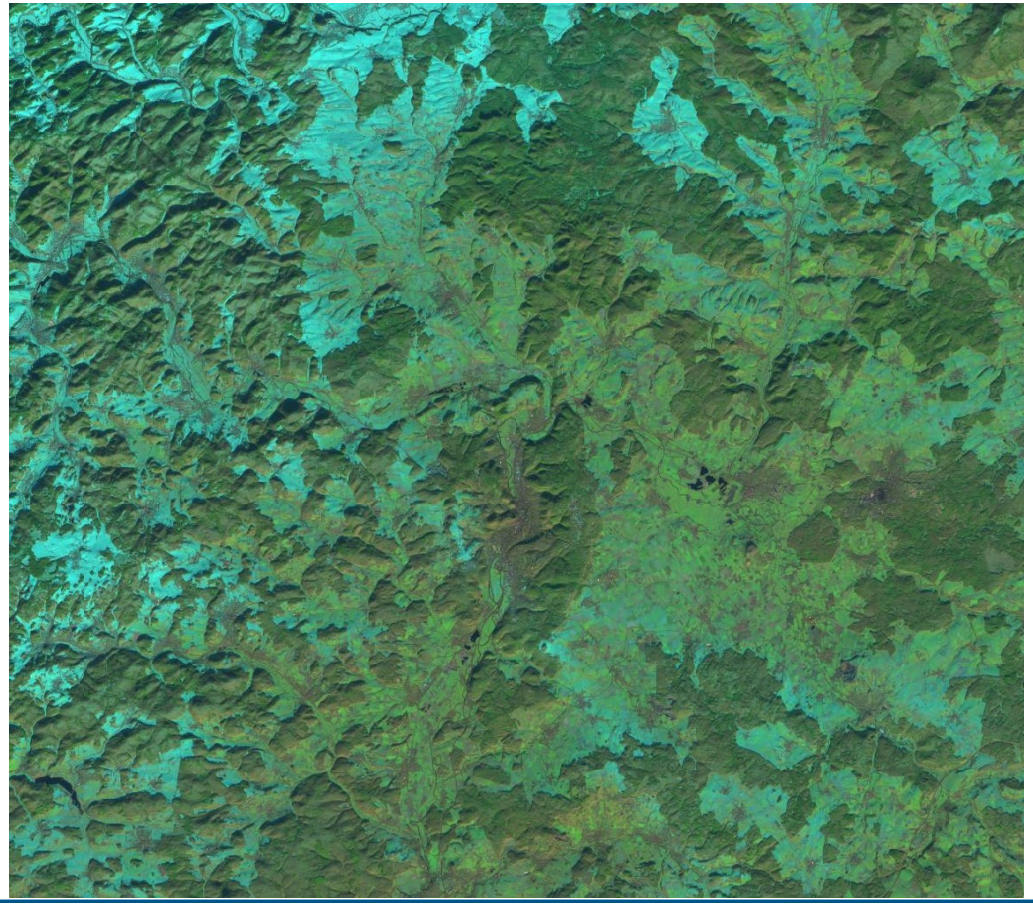
Step 4 (optional): Model transfer
When can a model be transferred?

Model can only make reliable predictions if the environment (defined by predictors) is known!

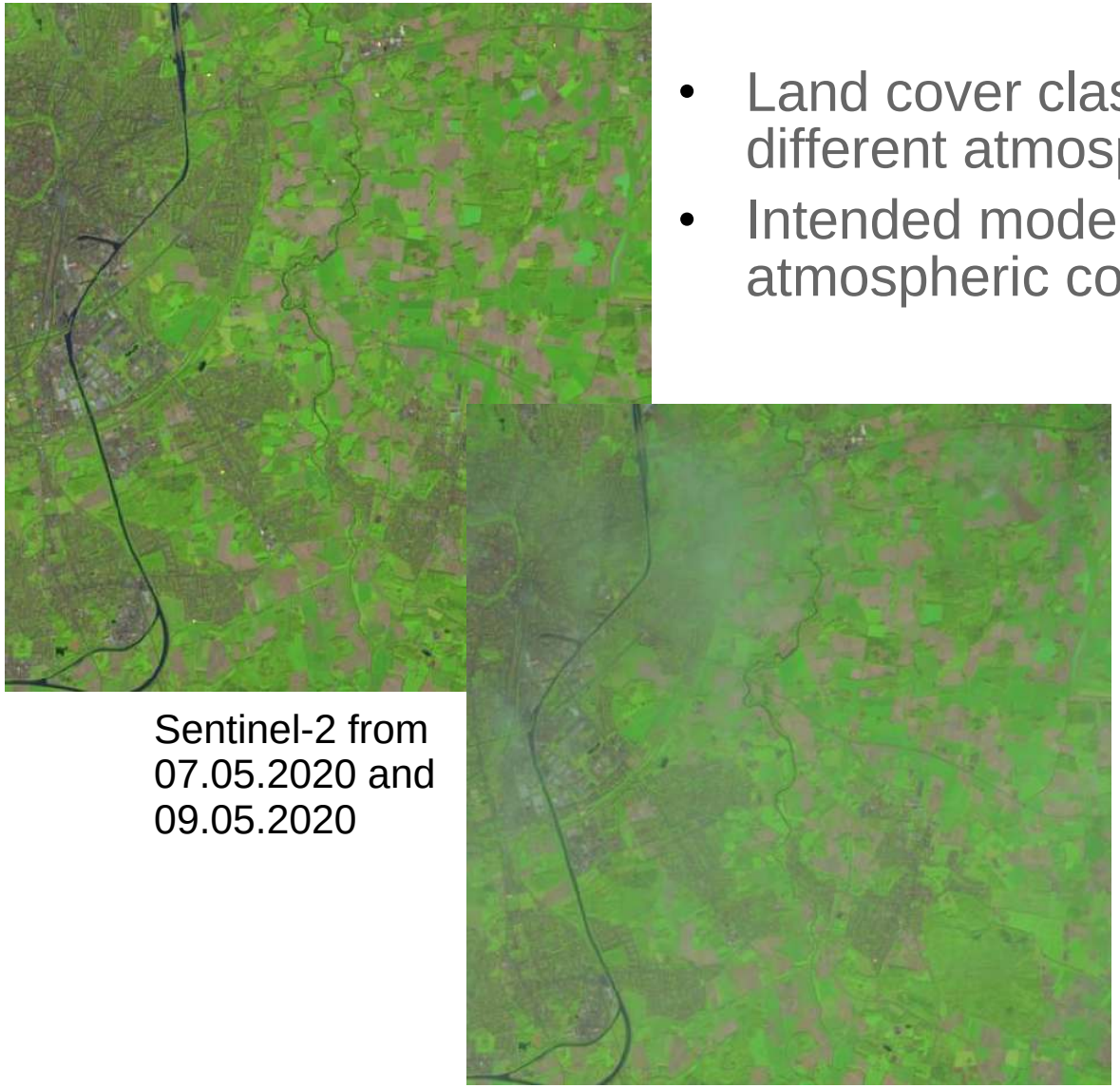
Marburg in summer



Marburg in winter



Step 4 (optional): Model transfer
When can a model be transferred?

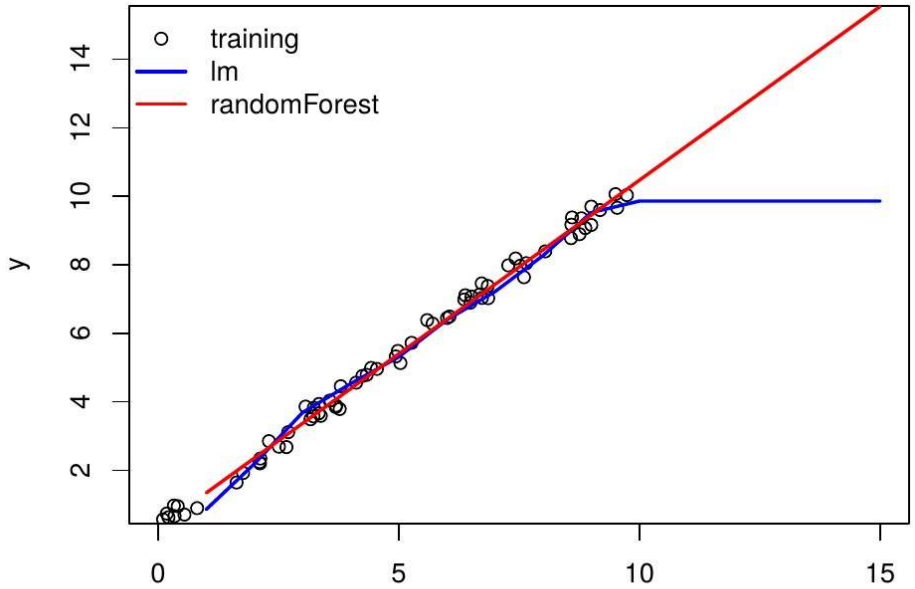


Sentinel-2 from 07.05.2020 and 09.05.2020

- Land cover classes might “look” different under different atmospheric conditions
- Intended model transfer would either require atmospheric corrections...
  - ...or assumption that we learn from data taken under variable conditions
  - Similar problem: images taken by different sensors

Step 4 (optional): Model transfer
When can a model be transferred?

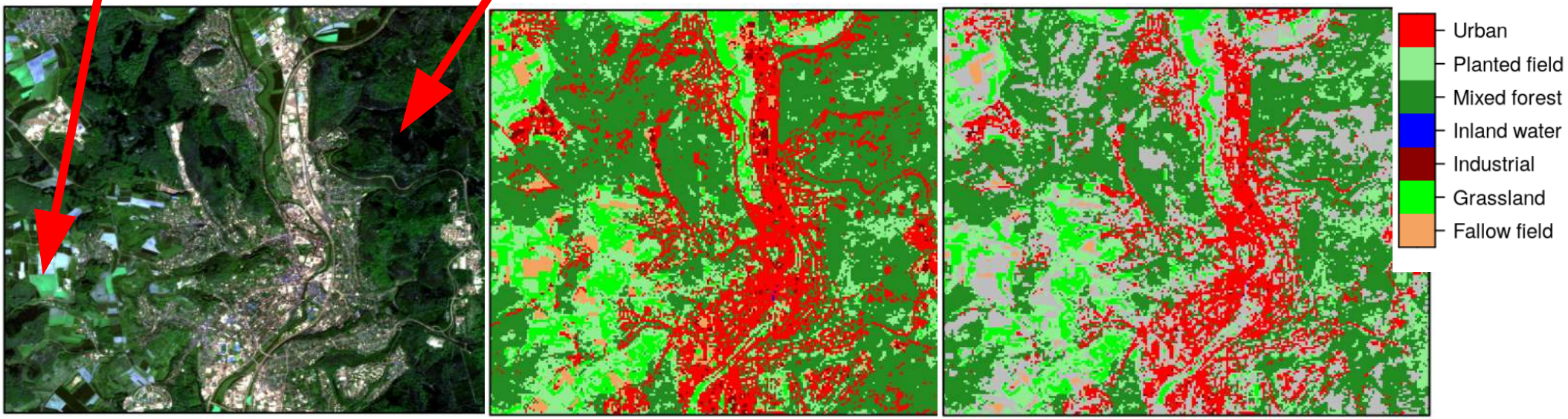
- ML cannot predict something new
- Land cover classes in new area must be part of the training data
- (Regression models: Possible range of response in new area must be observed in training)



Step 4 (optional): Model transfer
Estimate transferability by looking at the area of applicability

Fields with different spectral properties

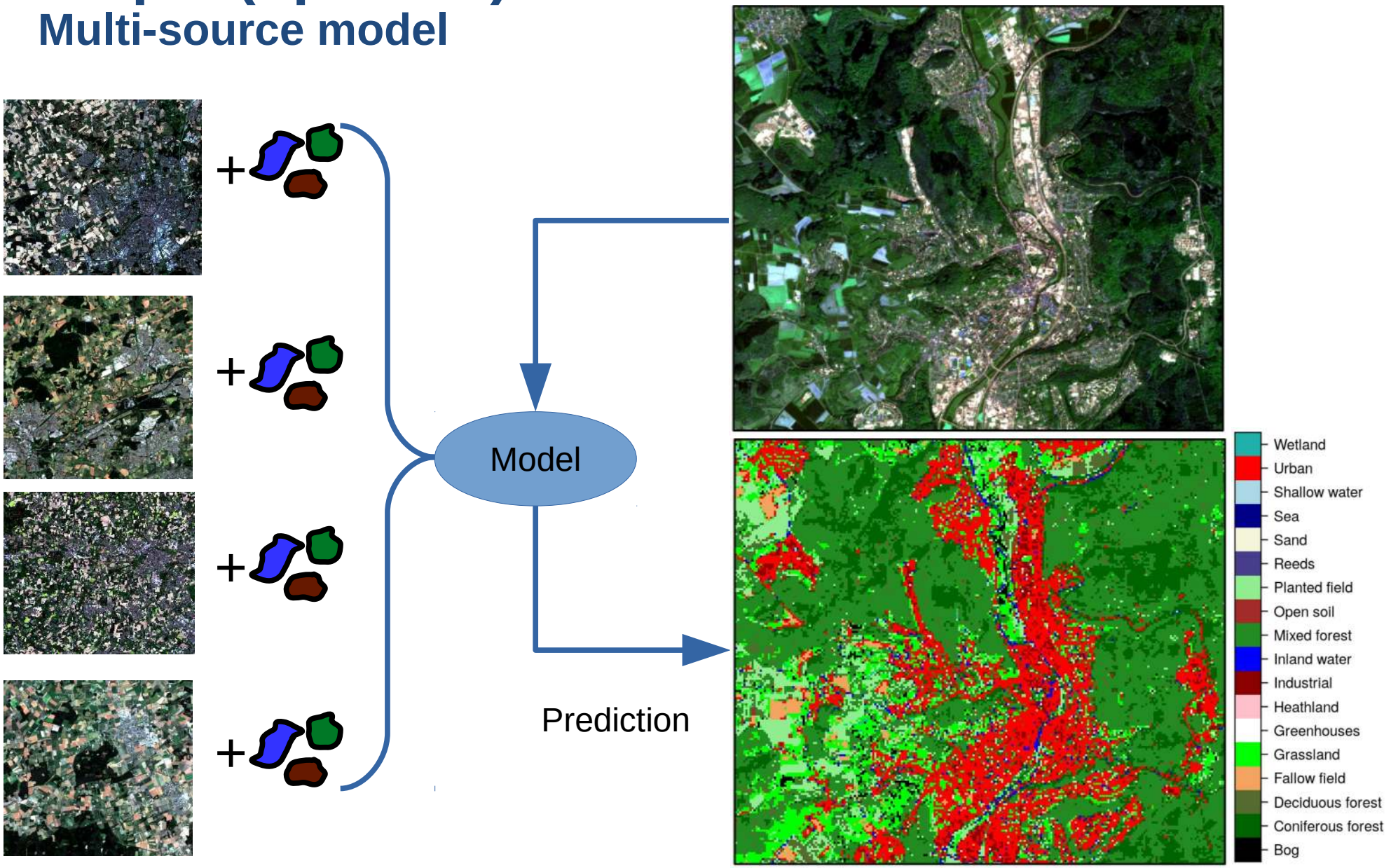
Coniferous forest not part of the training data



Large parts outside AOA.
How could we increase the AOA?

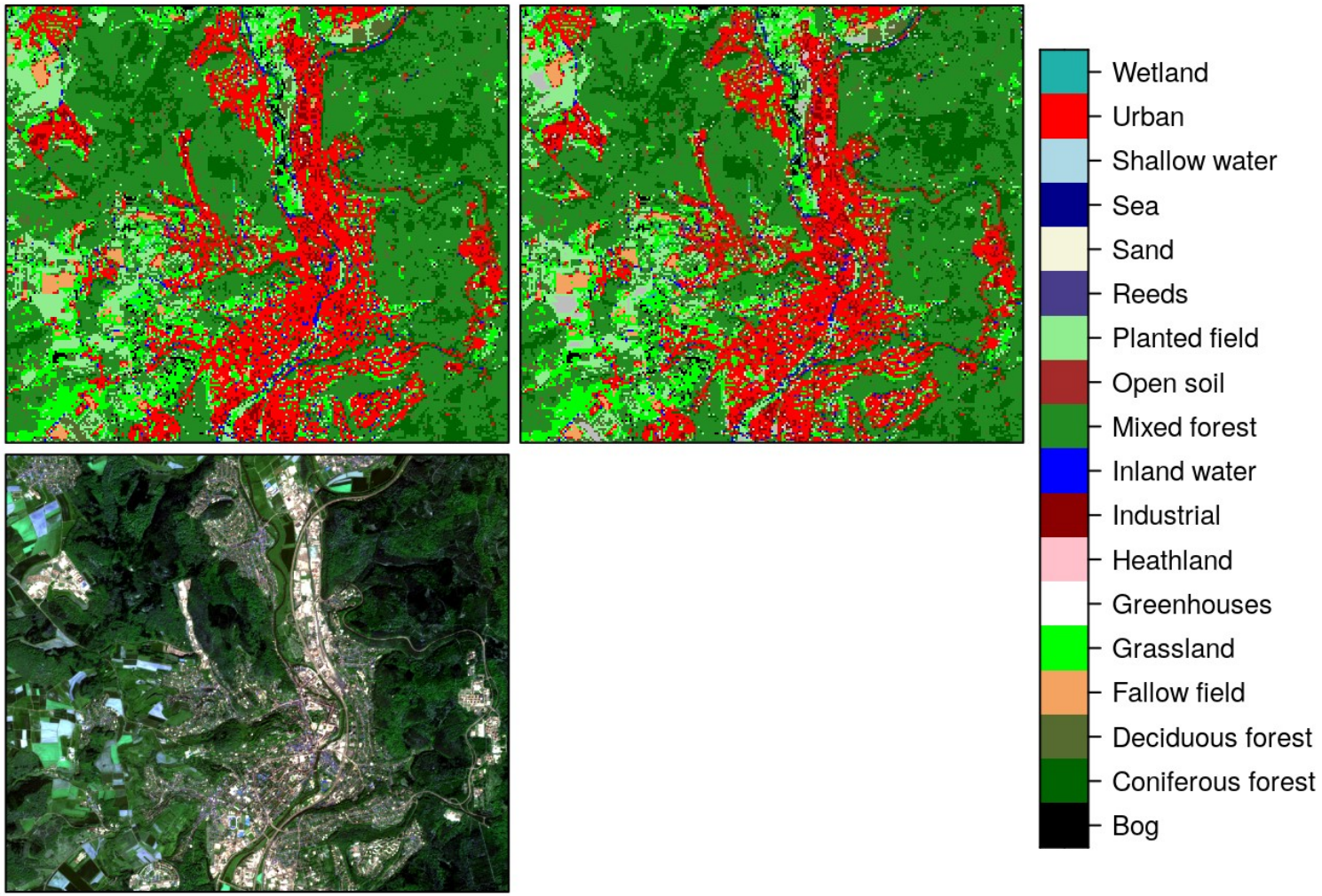
Step 4 (optional): Model transfer

Multi-source model



Step 4 (optional): Model transfer
Multi-source model

Prediction (left), prediction only for the AOA (right) and RGB composite (bottom)



Summary

- Spatial dependencies in the data require spatial cross-validation for model tuning, variable selection and validation
- But the cross-validation error is only valid for areas with predictor properties that are known to the model
- Use the AOA method to identify for which areas on average the cross-validation error of the model applies. Predictions outside AOA should not be presented!
- If model transfer is intended: use the AOA to estimate if the model can be transferred to the new area

- Note: the AOA methodology is still experimental (https://arxiv.org/abs/2005.07939). Please give us feedback :)
- Hands-on: www.github.com/HannaMeyer/OpenGeoHub_2020