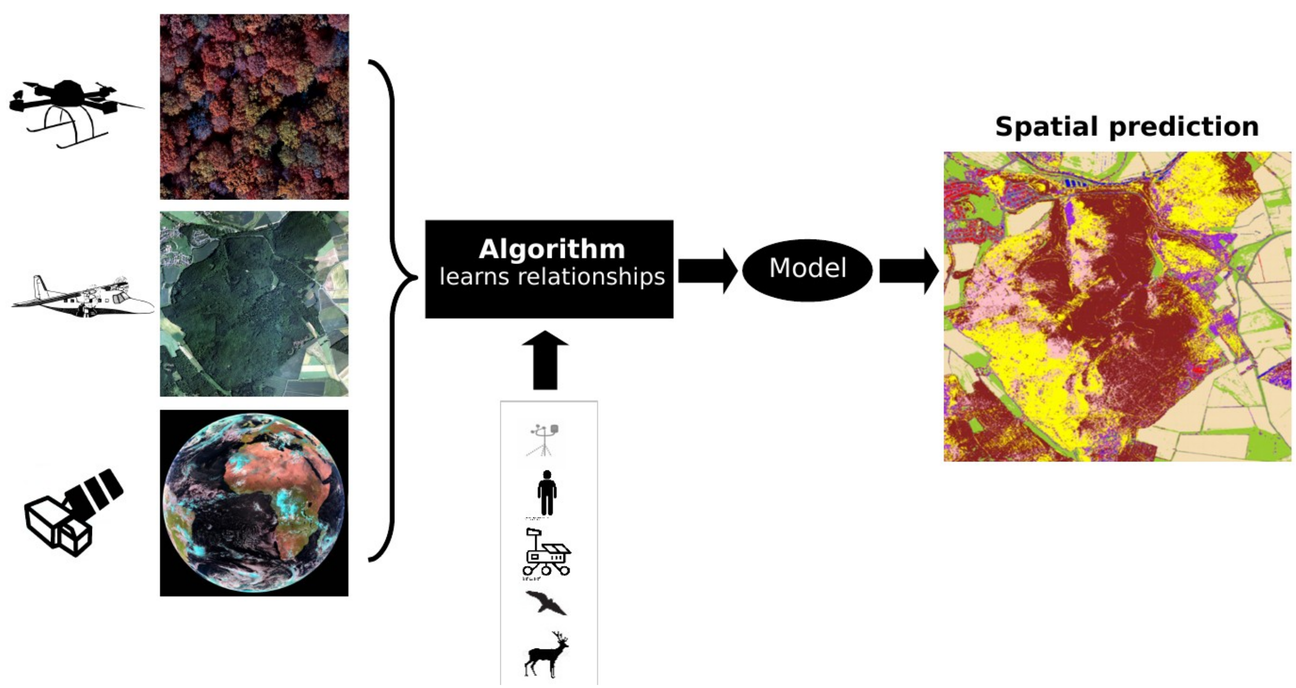
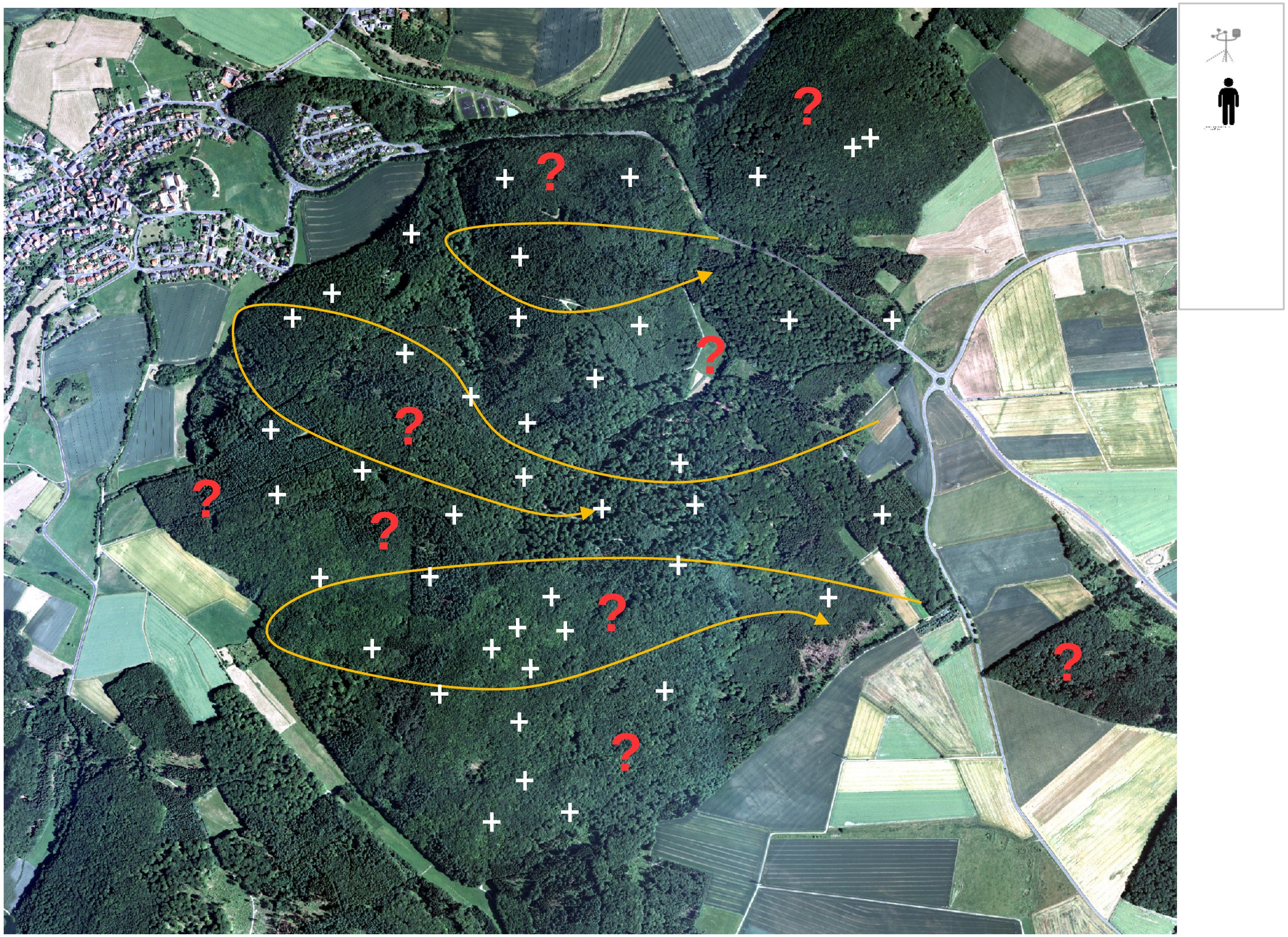
UAV images, satellite data and machine learning:
Towards a spatio-temporal continuous monitoring

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

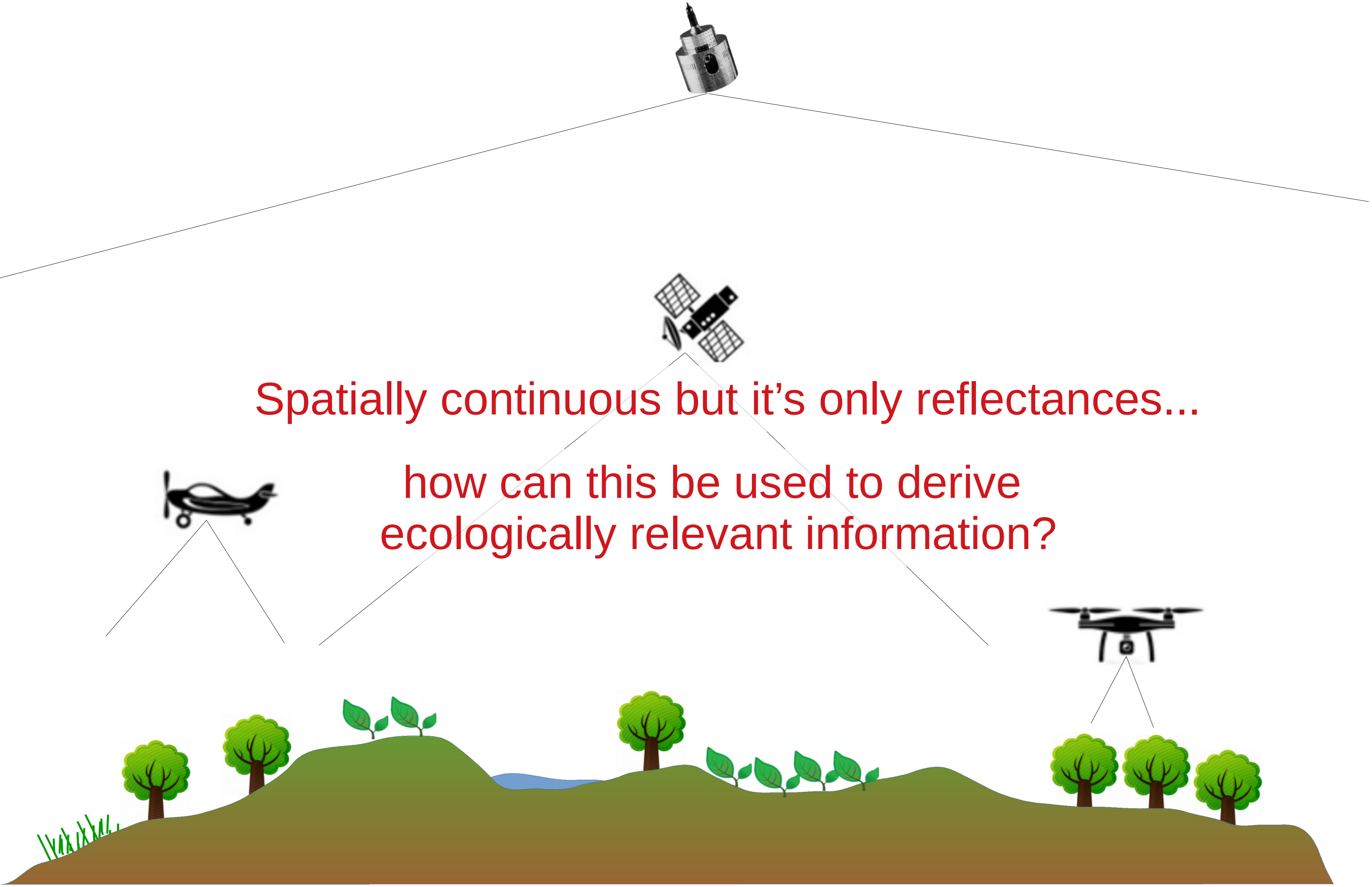


Problem: From field observations to maps of ecosystem variables

Nature 4.0 | Sensing Biodiversity

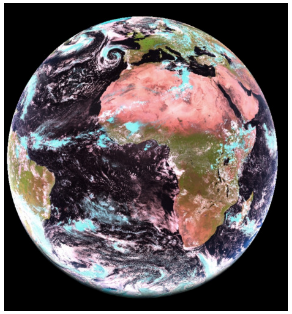


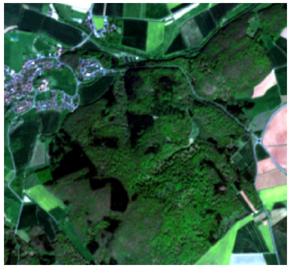
Remote Sensing of landscapes

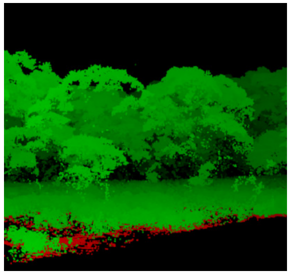


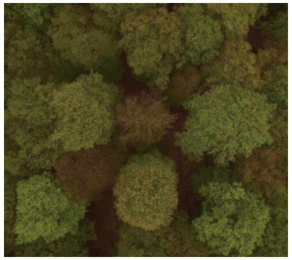
Spatially continuous but it's only reflectances...

how can this be used to derive ecologically relevant information?





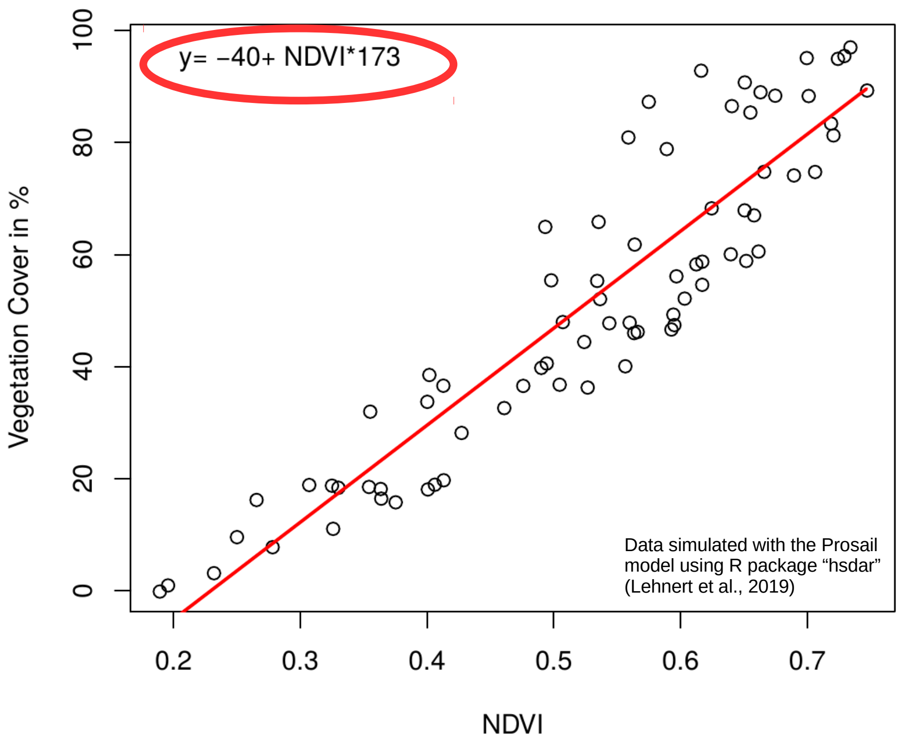




Predictive modelling of the environment

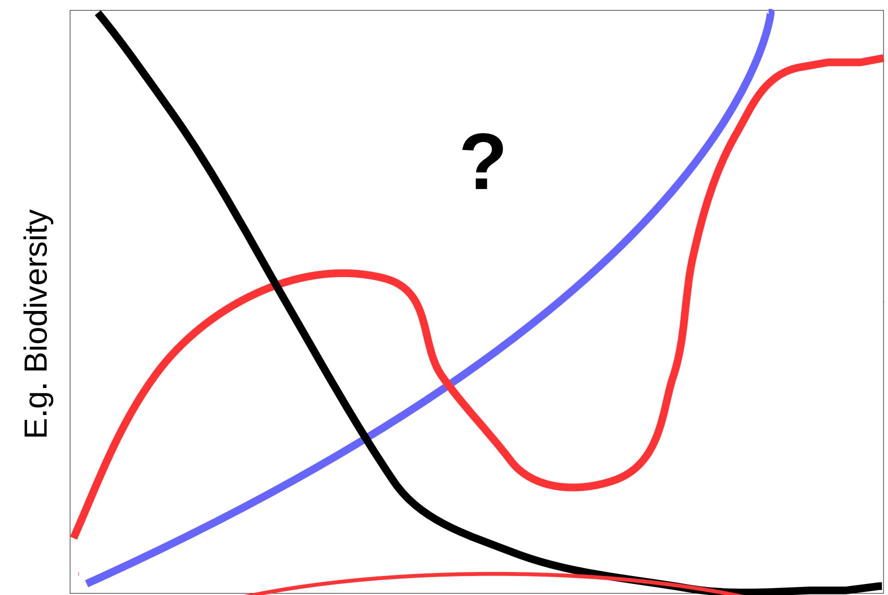
e.g. vegetation cover from satellite (VIS/NIR)

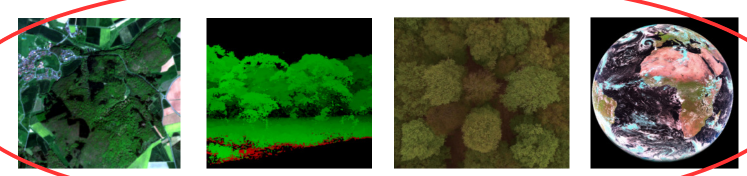
Field data



Remote sensing data

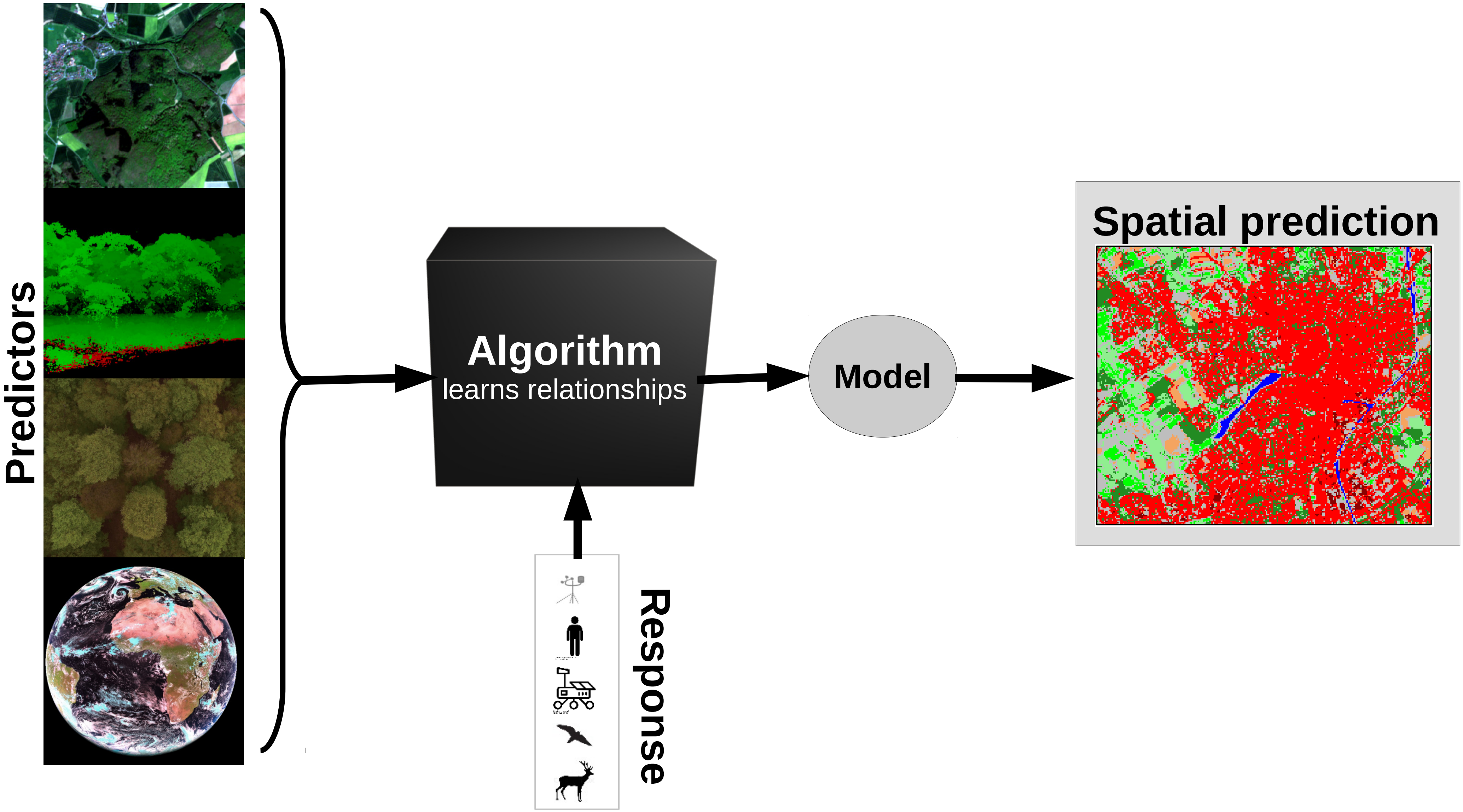
Typical ecological variables from satellite?



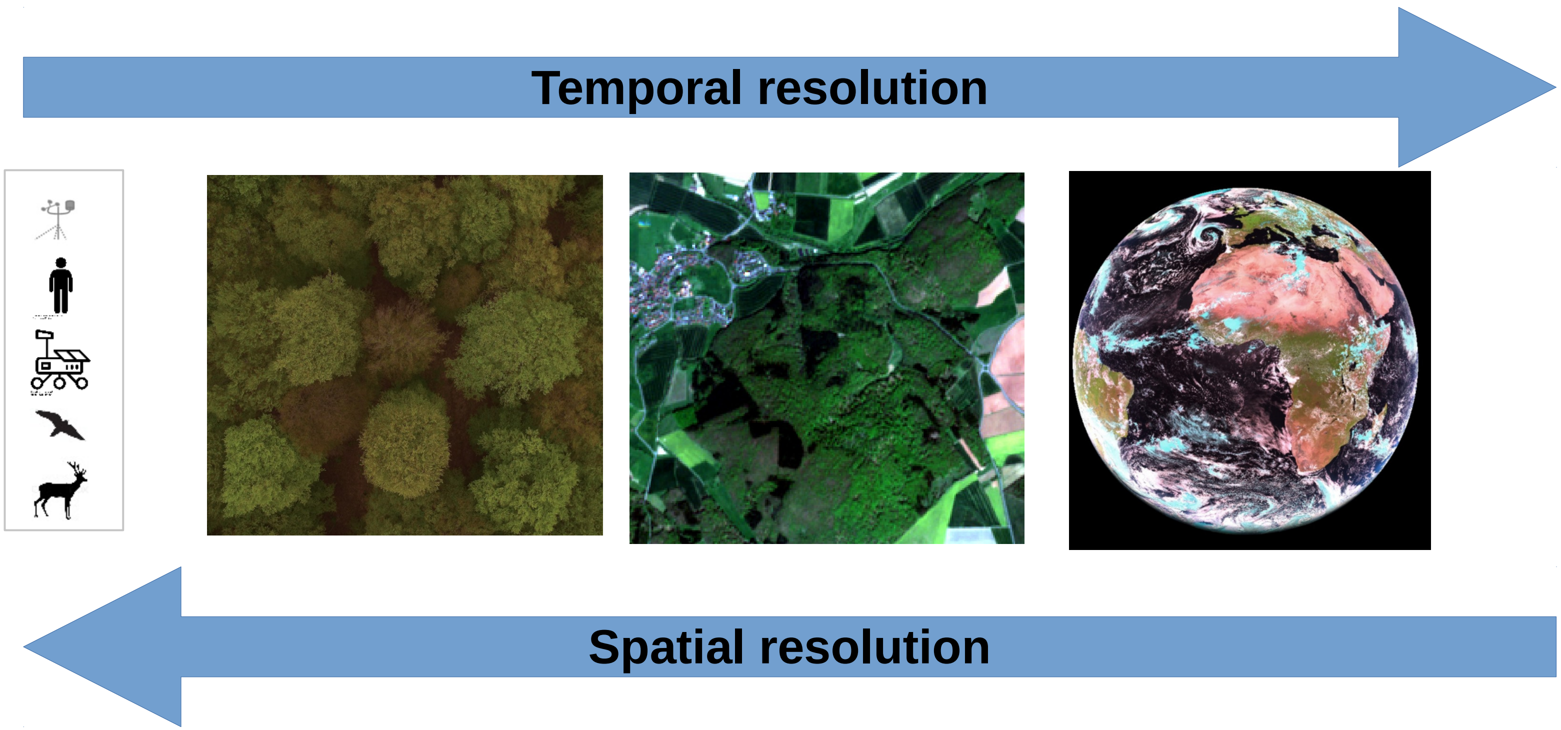


Models that can deal with complex nonlinear relationships are required!

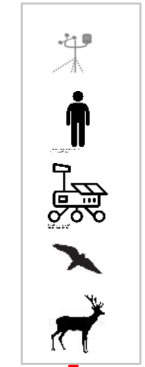
Predictive modelling of the environment: The machine learning way



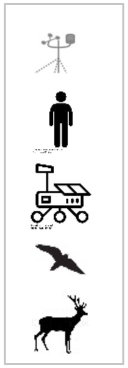
...but there are large differences in scale



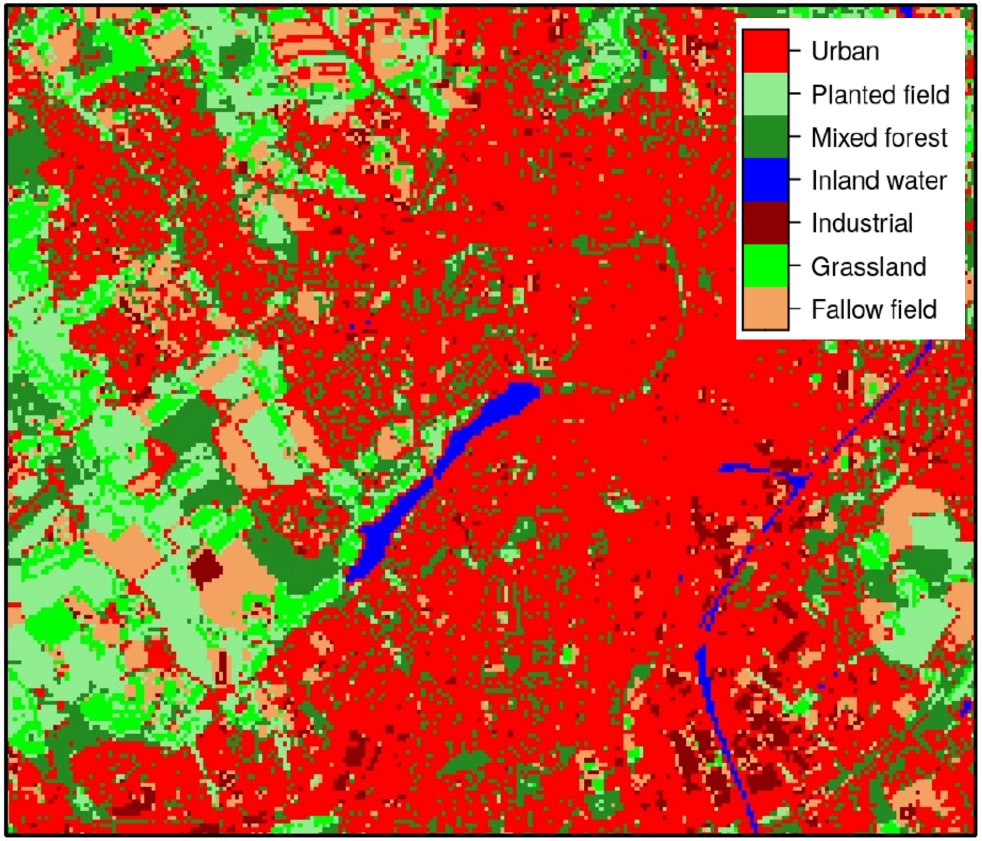
Suggestion:
Multi-scale generation of training sites



Aim of this workshop



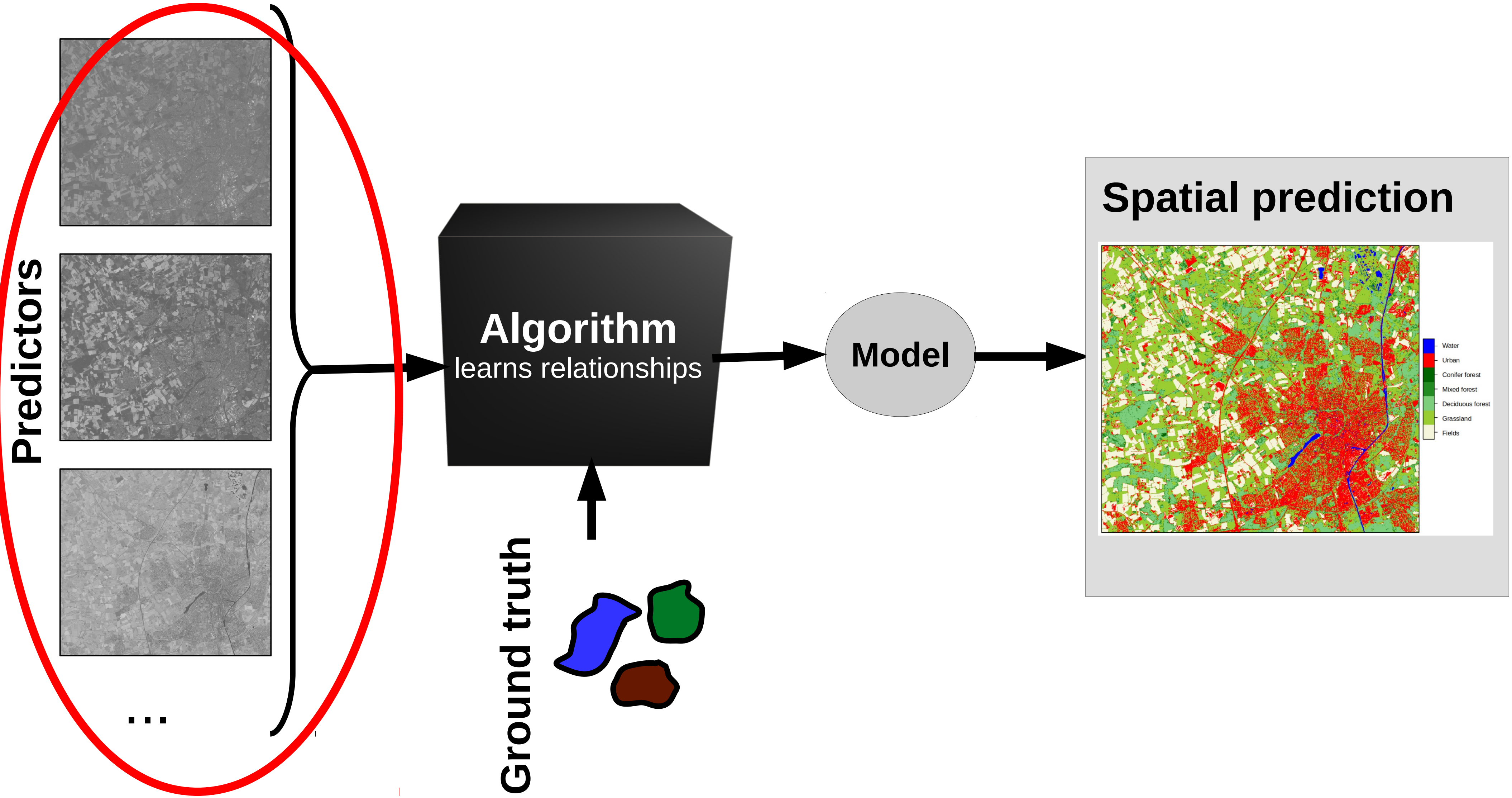
?



Why in R?

- Option to develop reproducible workflows
- Flexible functionalities for data handling, modelling, validation, visualization
- Open Source
- Large and helpful user community

How to use the spectral properties to classify land cover?



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

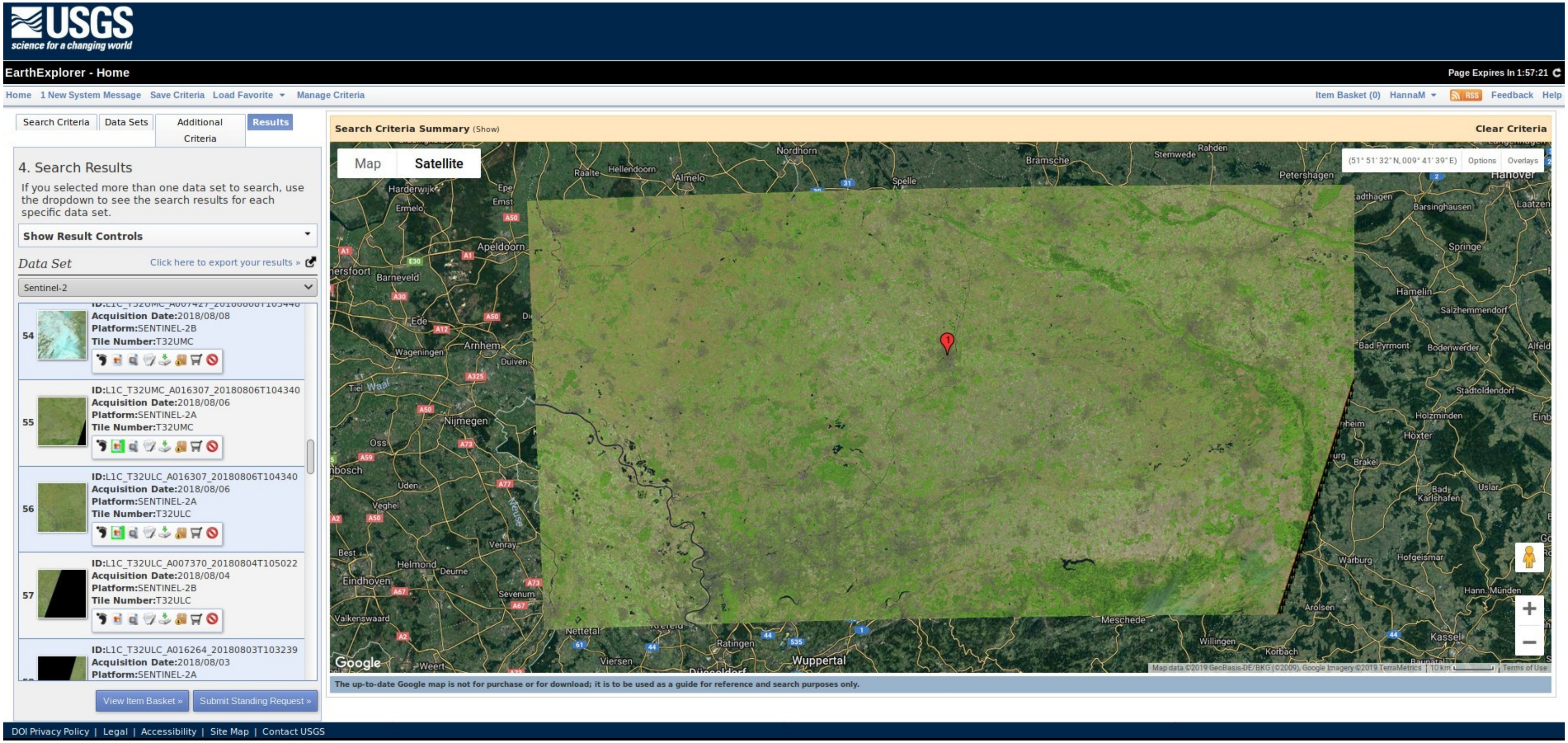
Spectral bands for the Sentinel-2 sensors[10]

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/

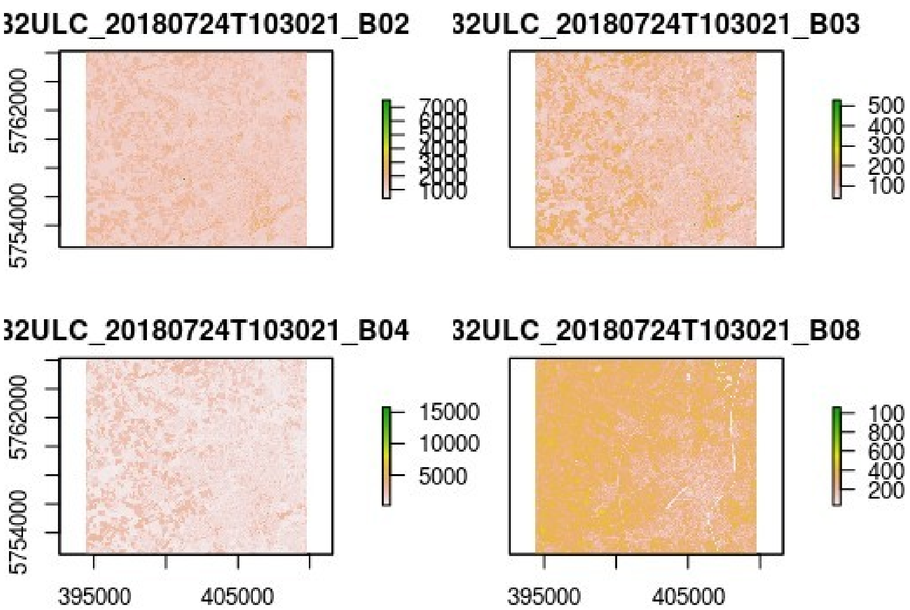
Load satellite data in R

- 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

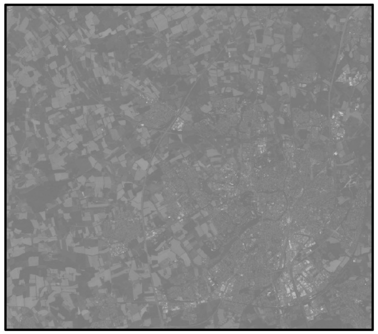
```
library(raster)
sen <- stack("blue.tif","green.tif",
             "red.tif","NIR.tif")
plot(sen)
```

And more channels!

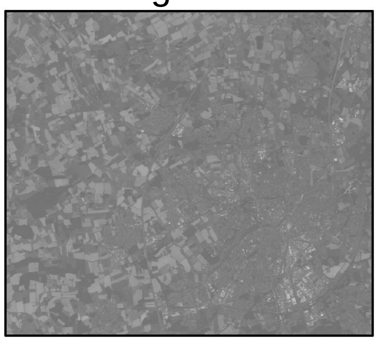


How do we get the “color” in the satellite data?

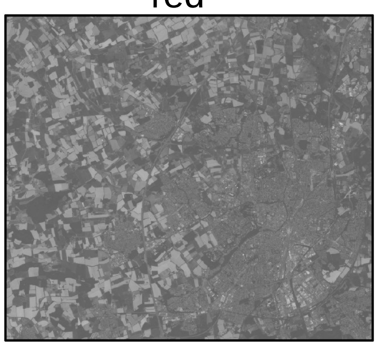
blue

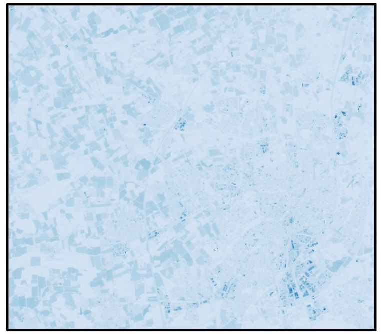


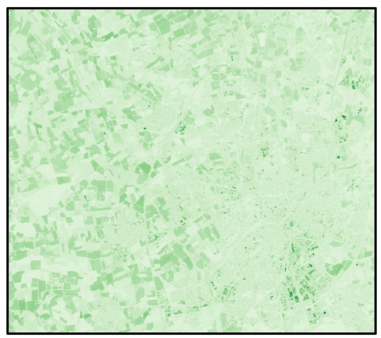
green

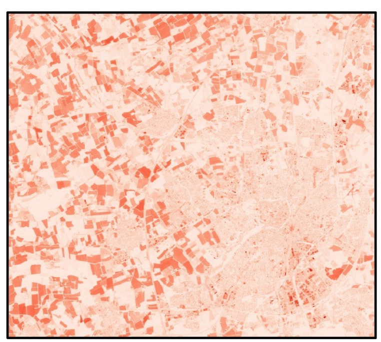


red





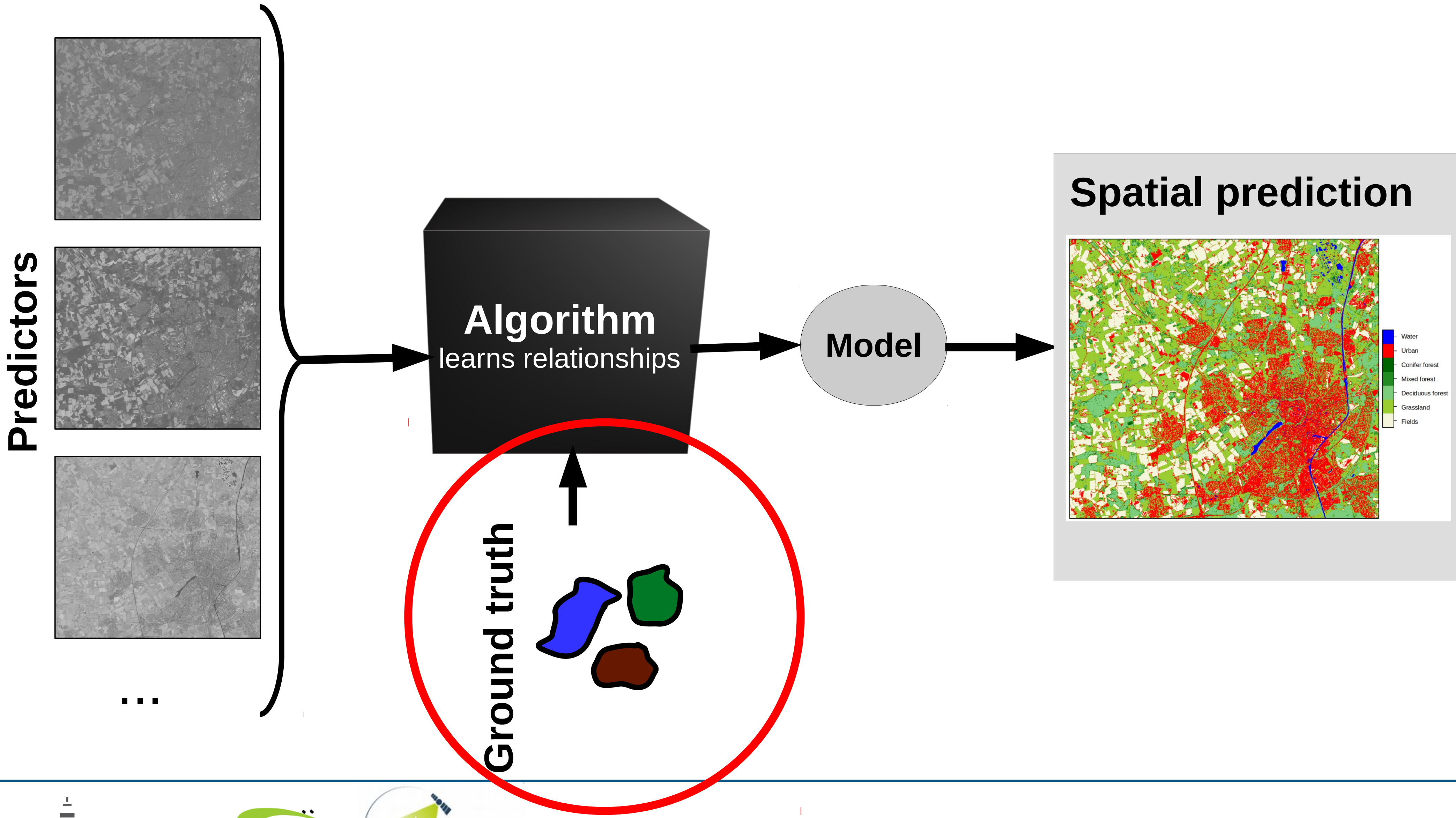




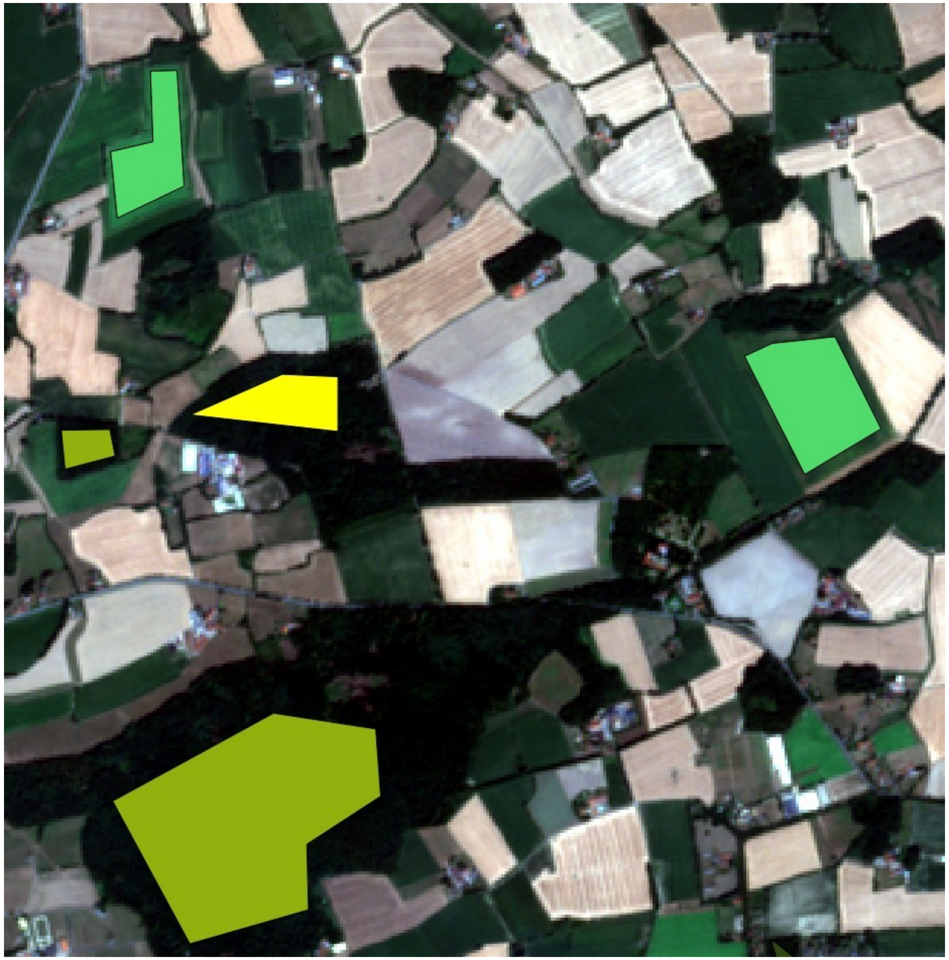
How to do a color composite in R

```
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?

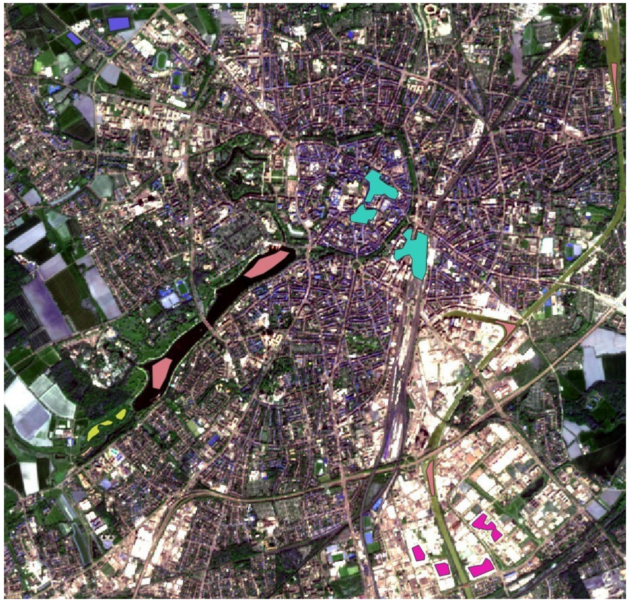


How to use the spectral properties to classify land cover?
Reference data are required



- Training data from field work, expert knowledge, existing databases,… OR FROM HIGHER RESOLUTION UAV DATA
- Usually polygons
- In this case study: digitized (in QGIS)

Combine predictors and response

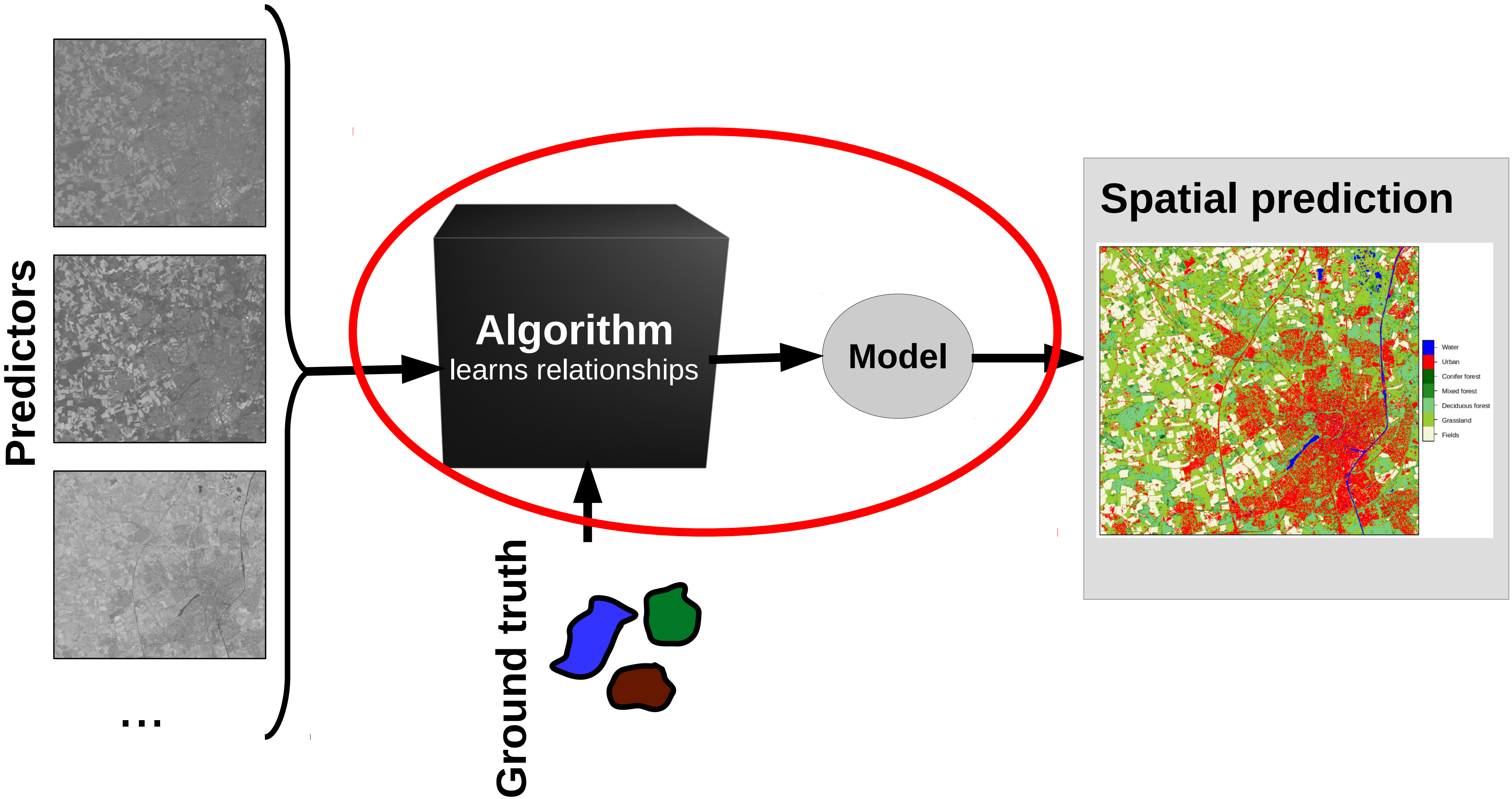


	B02	B03	B04	B08	…	Class
1	857	632	387	308		Water
2	848	633	389	312		Water
3	843	624	357	343		Water
4	854	630	360	333		Water
5	854	628	376	302		Water
6	859	615	364	350		Water

How to do it in R

```
extr <- extract(sen, trainSites, df=TRUE)
extr <- merge(extr, trainSites, by.x="ID", by.y="PolygonID")
```

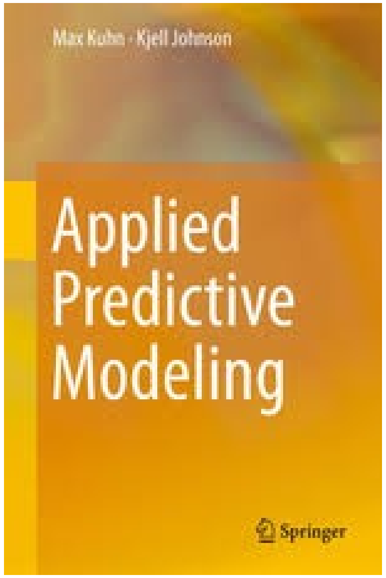
Model training



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

For today's session



Step 1: Model training in R

	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

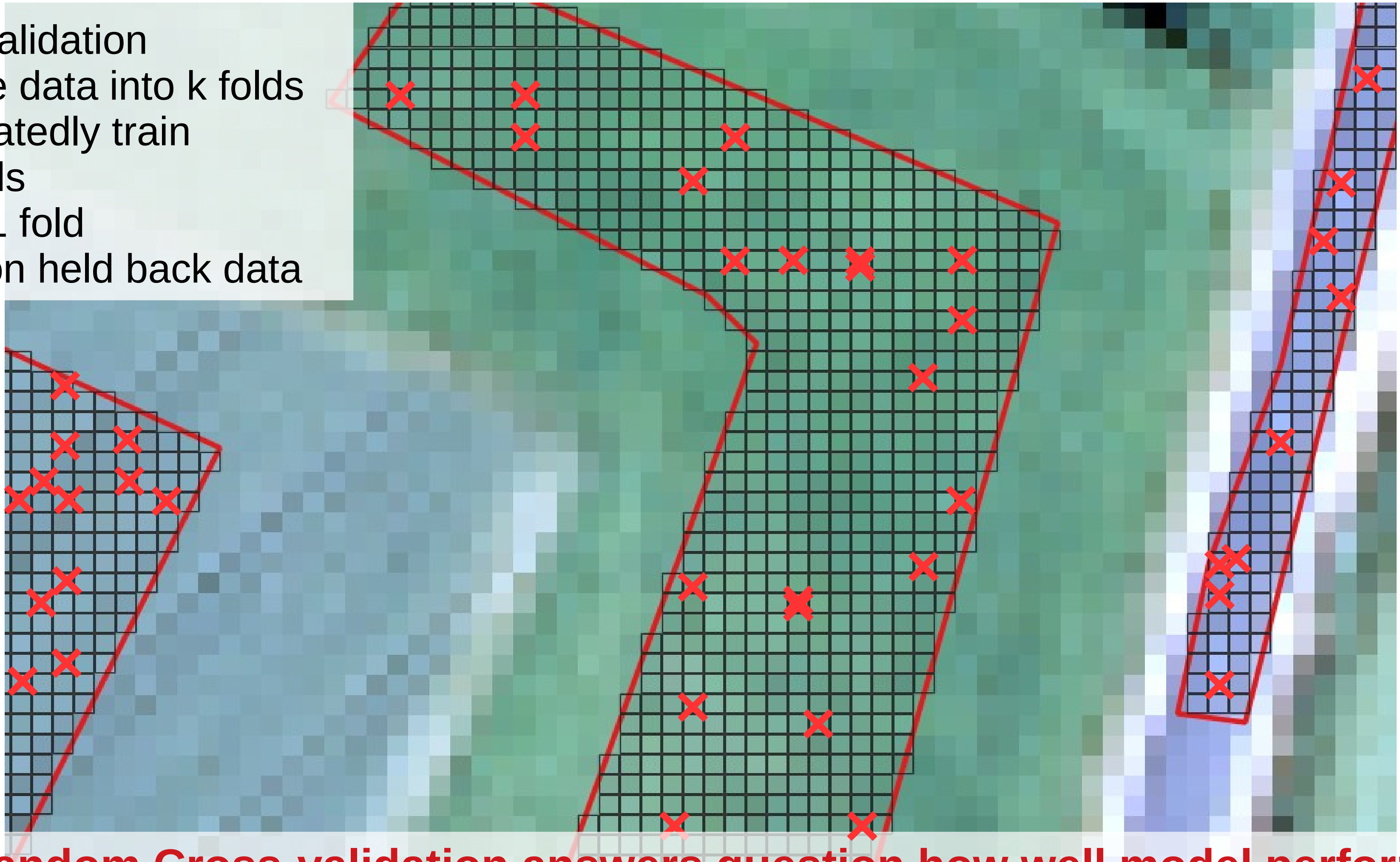
Perfect prediction?

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

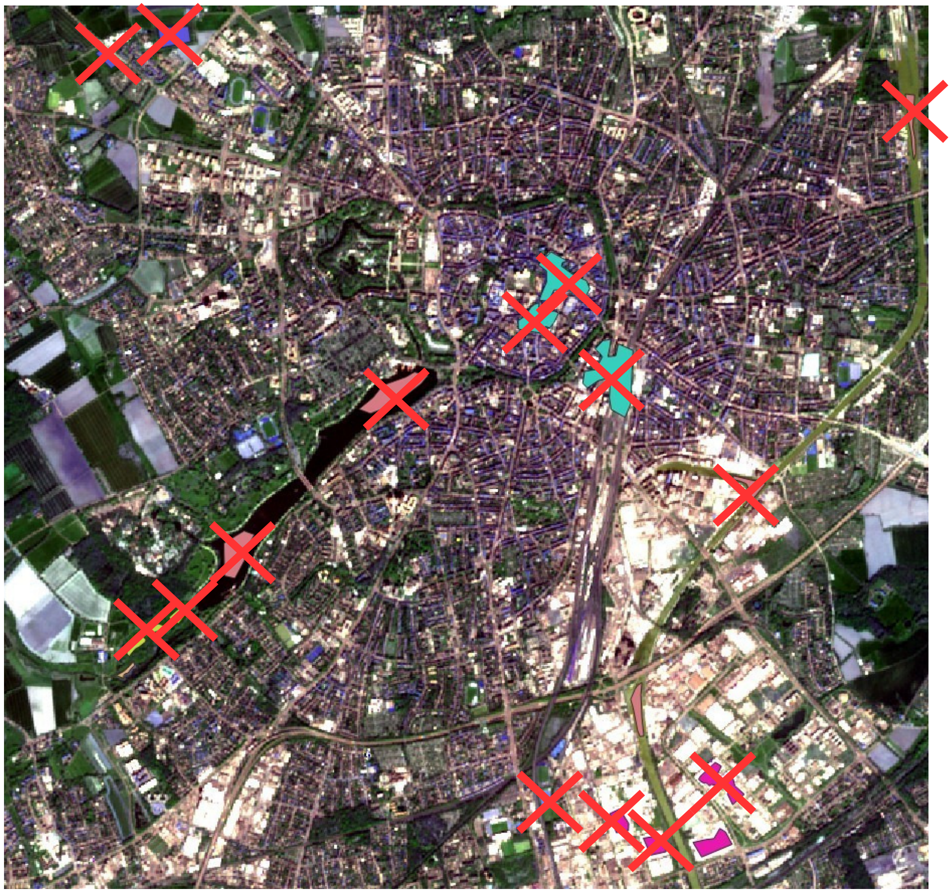


Random Cross-validation answers question how well model performs on very similar locations

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

```
library(CAST)
```
"**C**aret **A**pplications for **S**tatio-**t**emporal models"

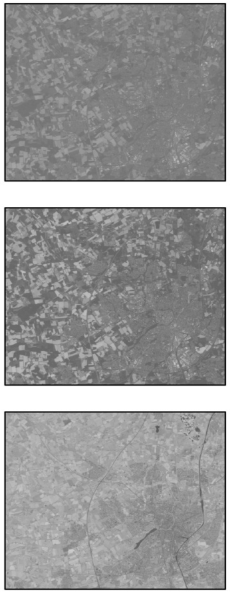
```
indices <- CreateSpacetimeFolds(trainDat,
                                spacevar = "PolygonID",
                                k=3,
                                class="Label")

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

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

Step 1: Model training in R
Spatial cross-validation: Example of results

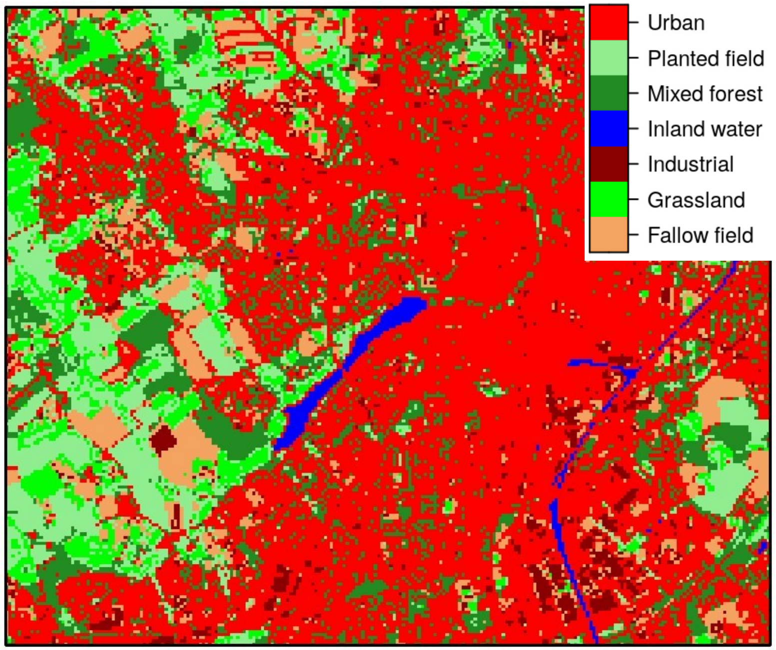
Variables	Validation	Accuracy	Kappa
all	random	>0.99	>0.99
all	spatial	**0.68**	**0.61**

- 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 2: Model prediction in R



+ trained model =



...

How to do it in R

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

Further information and hands-on

- Our initial motivation was to predict time series. For this, consider the "Area of Applicability. Taught at the OpenGeoHub Summer school: https://www.uni-muenster.de/RemoteSensing/lehre/summer_schools/
- Hands-on: www.github.com/HannaMeyer/OpenGeoHub_2020