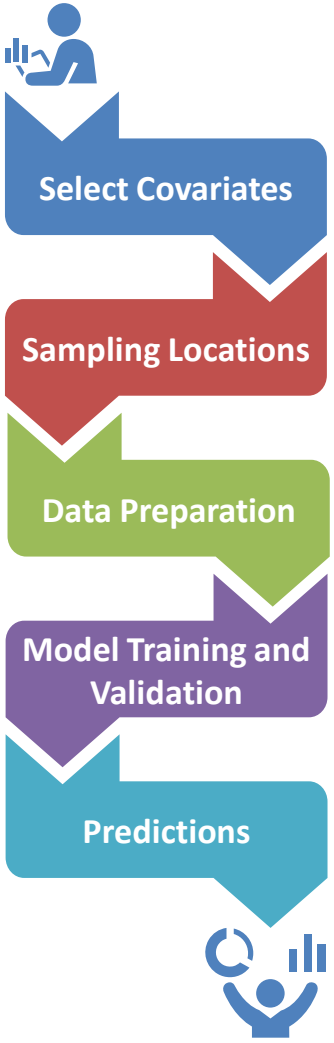
Content

- Part 1
  - Conventional Soil Mapping
  - Digital Soil Mapping
  - Environmental Covariates
  - Interpolation (Geostatistical Analysis)

- **Part 2**
  - **Machine Learning Methods**
  - **Decision Tree and Random Forest**
  - **Model Validation and Uncertainty Analysis**
  - **Soil Depth Functions and 3D Maps**

Sequence of DSM Steps



1. Environmental covariates, relevant as predictors of soil property/class, are derived from remote sensing, digital elevation, climatic datasets, …
2. Soil samples are collected at the specified locations (e.g., Latin hypercube sampling) and soil property is measured in the laboratory.
3. Intersecting the covariates with the soil point observations.
4. Machine learning models (e.g., random forest) are trained using training data, and accuracy assessment is carried out using the test data set.
5. The ML models are applied to the entire study area in order to produce a soil property/class map.

Technical and Practical Notes

s
c
o
r
p
a
n

Types of covariates

- Climate related covariates (e.g., temperature map)
- Vegetation and living organisms (e.g., vegetation index map)
- Relief and topography-related covariates (e.g., slope map)
- Parent material covariates (e.g., geology map)
- Spatial position or spatial context (e.g., easting — distance to east)
- Human or Anthropogenic Influences (e.g., land use/land management maps)

Choosing covariates

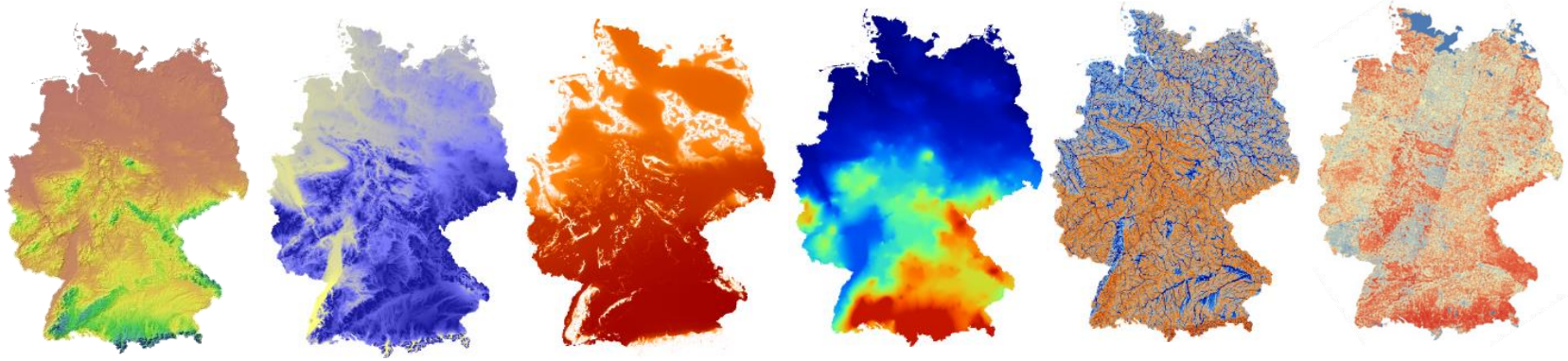
- What factors influence soil formation?
- What data do I have?

Technical and Practical Notes

s c o r p a n

Covariate data sources

- SRTM digital elevation model (30 m resolution)
- WorldDEM digital elevation model (12 m resolution)
- Landsat-8 satellite images (30 m resolution)
- Sentinel-2 satellite images (10, 20, and 60 m resolution)
- MODIS satellite images (250, 500, and 1000 m resolution)
- Global Land Cover maps (30 m resolution)
- JAXA's ALOS radar images (20 m resolution)
- Monthly precipitation images (1000 m resolution)
- Geology maps (polygon)



Technical and Practical Notes

s
c
o
r
p
a
n

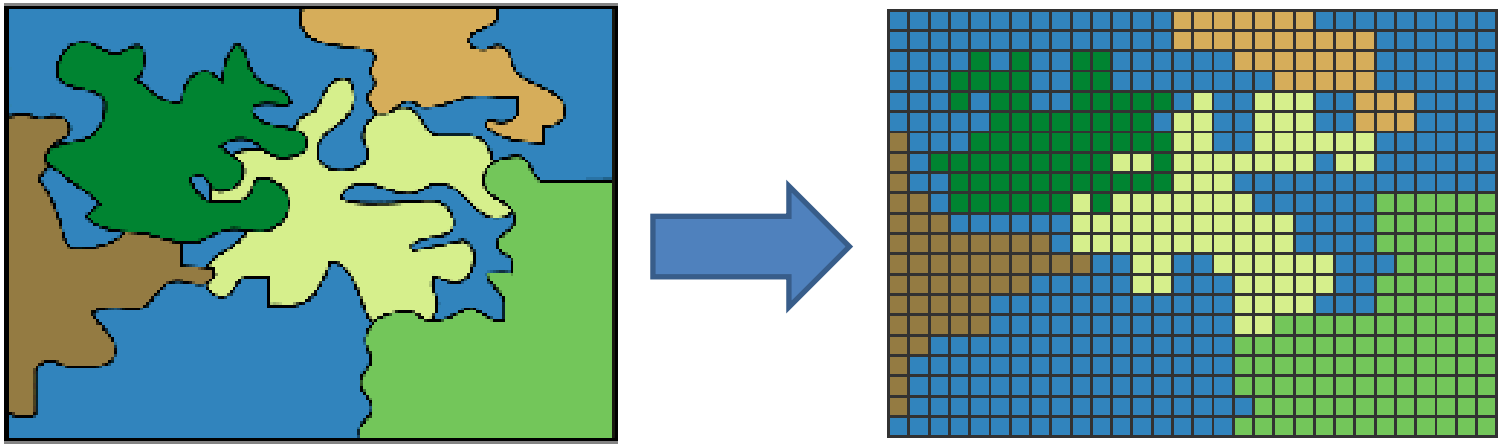
Preparing covariate layers

1. Converting polygon maps to rasters
2. Downscaling or upscaling rasters to a common resolution,
3. Filtering out missing pixels/reducing noise and multicolinearity problems,
4. Overlaying raster stacks and points

Technical and Practical Notes

s
c
o
r
p
a
n

1. Converting polygon maps to rasters



Polygon features

Raster polygon features

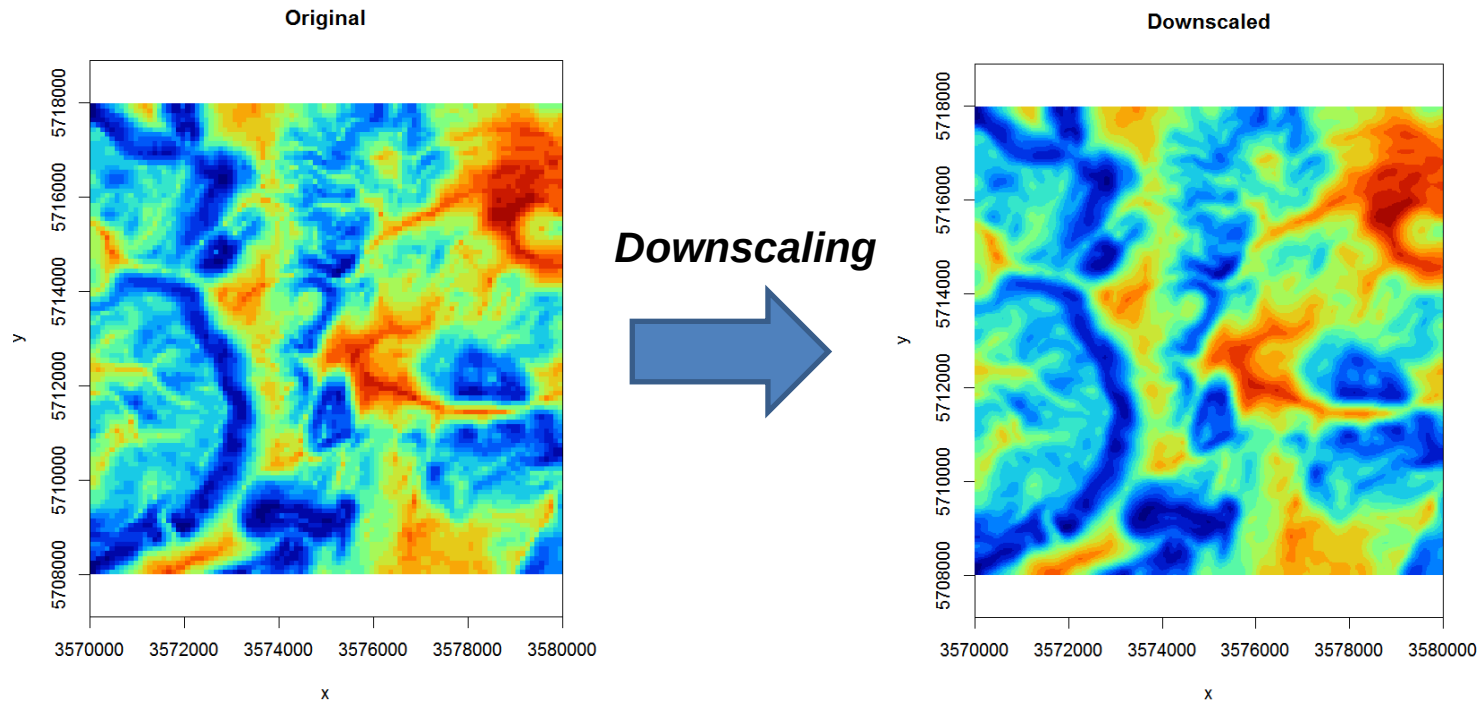
soilmapper.org
desktop.arcgis.com

Technical and Practical Notes

s c o r p a n

2. Downscaling or upscaling rasters

- To adjust the resolution of some covariates that have either too coarse or too fine a resolution compared to the target resolution
- The process of bringing raster layers to a common grid resolution is also known as resampling

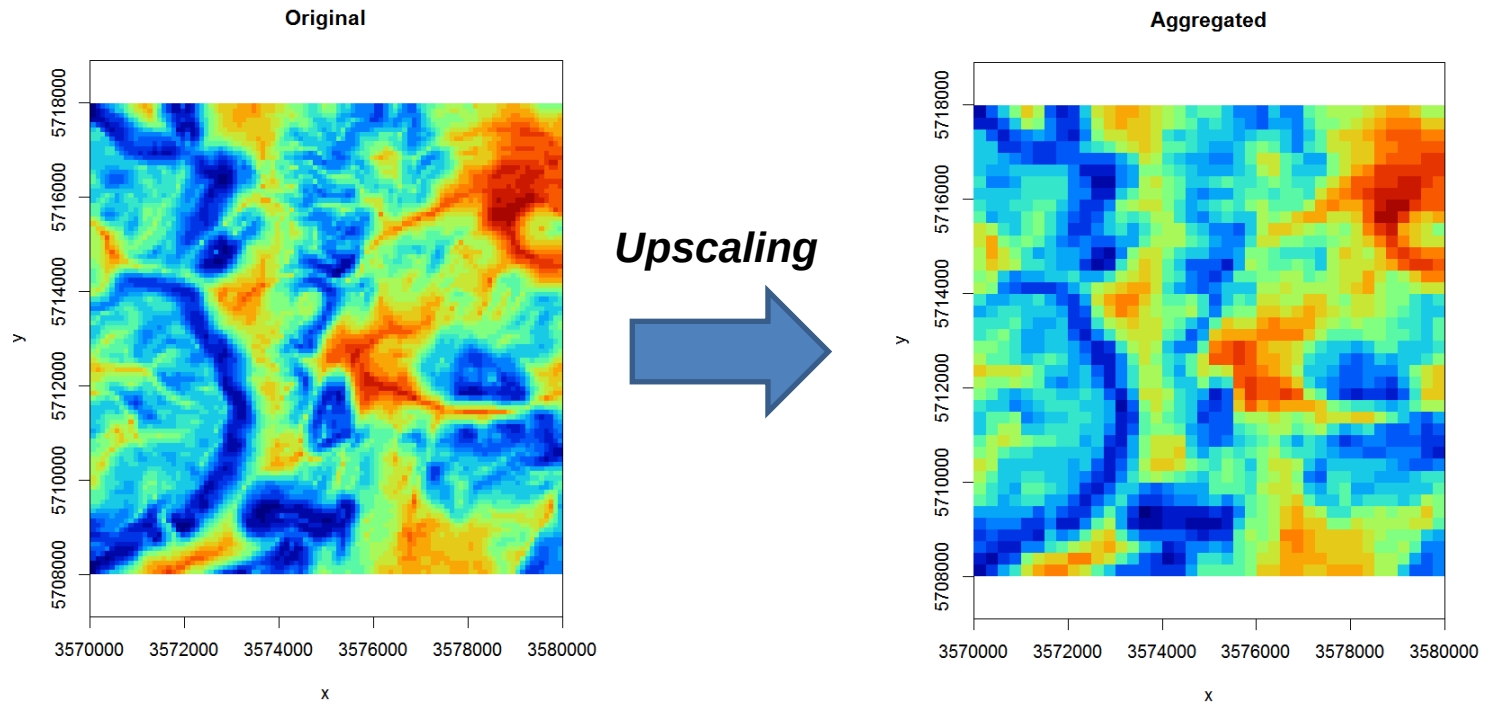


Technical and Practical Notes

s c o r p a n

2. Downscaling or upscaling rasters

- To adjust the resolution of some covariates that have either too coarse or too fine a resolution compared to the target resolution
- The process of bringing raster layers to a common grid resolution is also known as resampling



Technical and Practical Notes

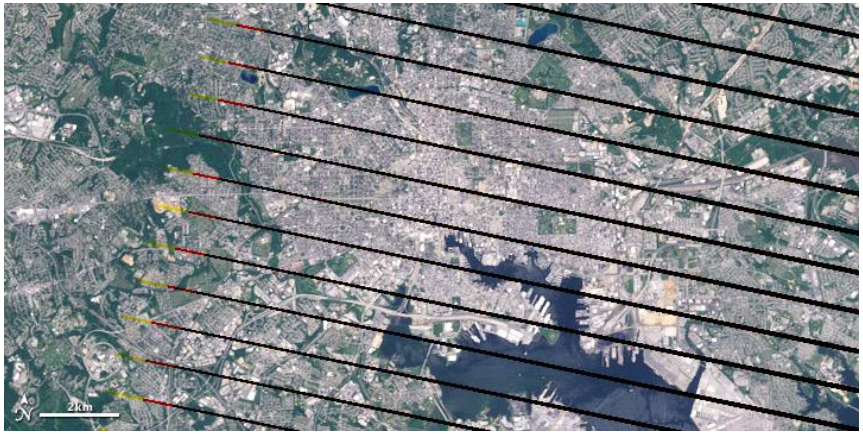
s c o r p a n

3. Filtering out missing pixels and artifacts

- could cause serious problems for producing soil maps as the missing pixels and artifacts would propagate to predictions: if only one layer in the raster stack misses values then predictive models might drop whole rows in the predictions even though data is available for 95% of rows.

- Missing pixels can be efficiently filtered by using for example the gap filling functionality available in the SAGA GIS

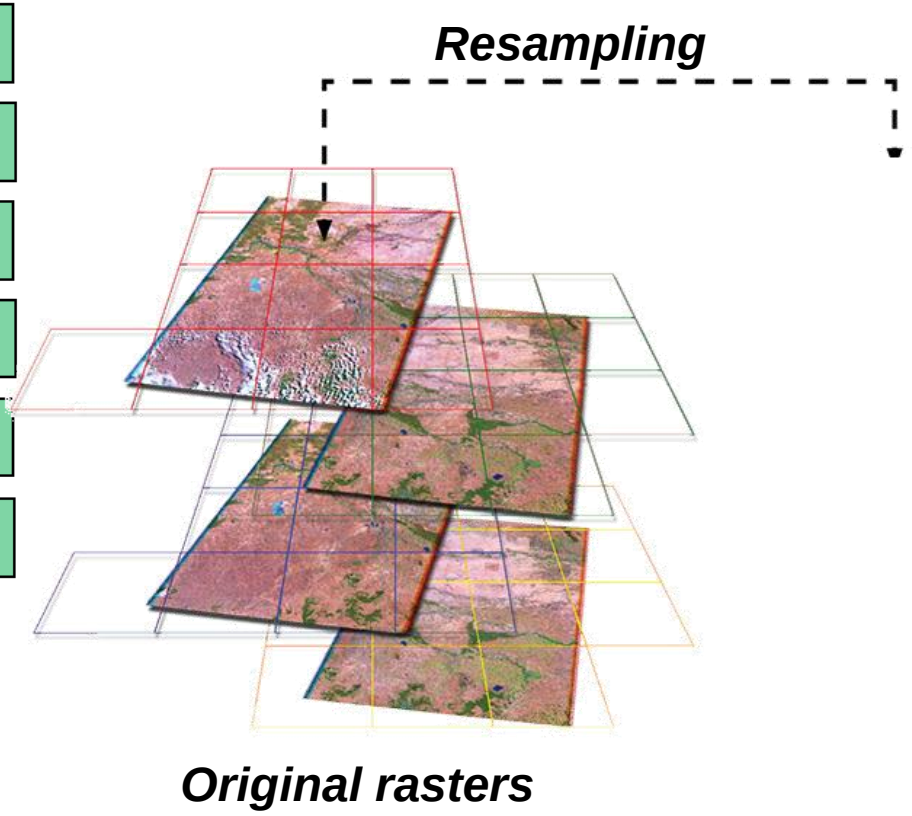
- Another way to filter the missing pixels, to reduce noise and to reduce data overlap is to use Principal Components transformation of original data



Technical and Practical Notes

s c o r p a n

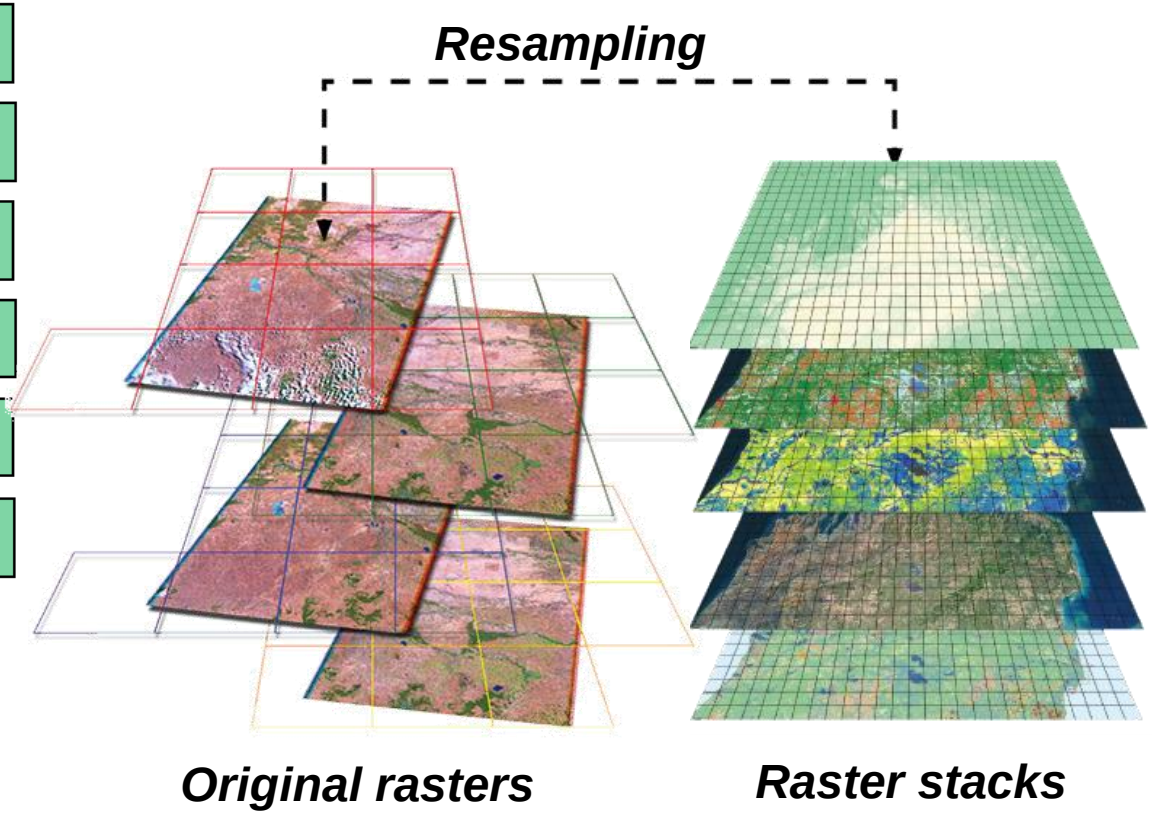
Raster stacks



Technical and Practical Notes

s
c
o
r
p
a
n

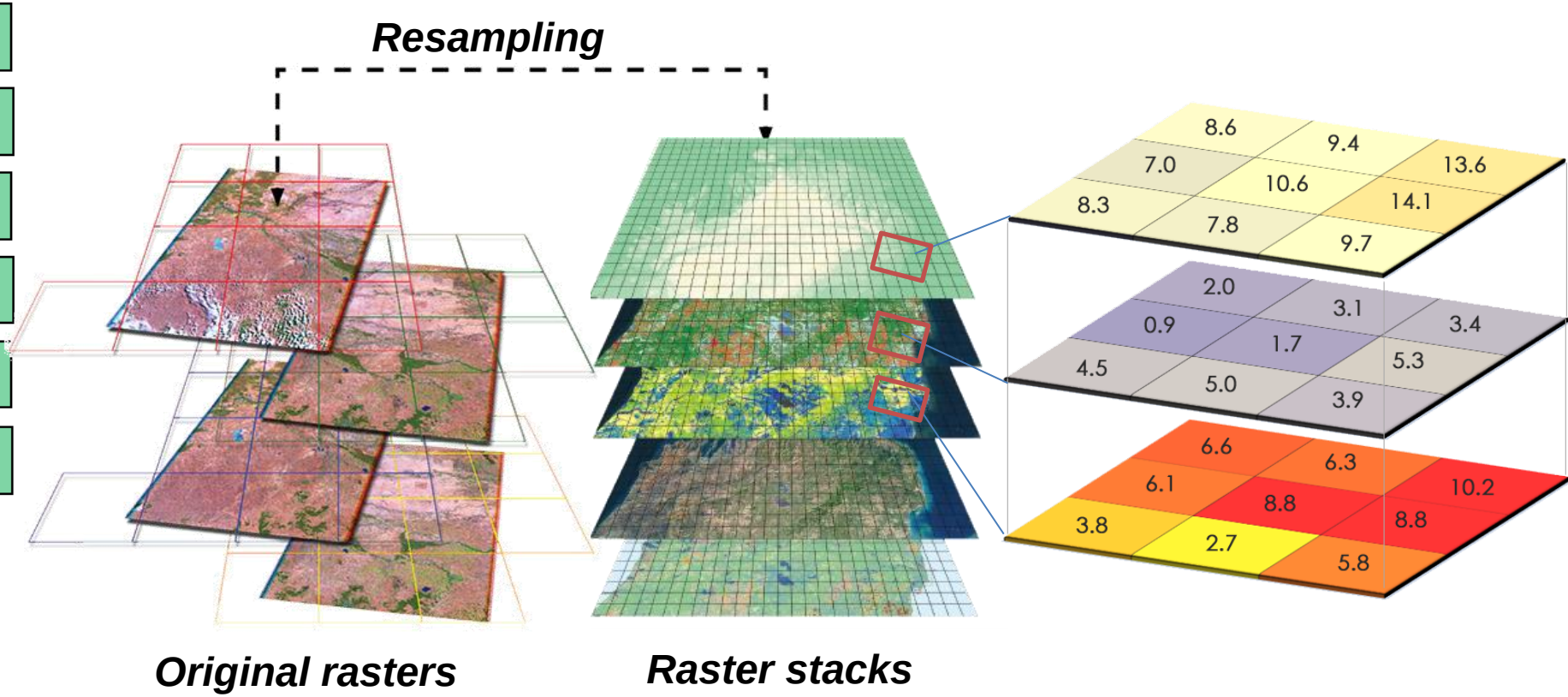
Raster stacks



Technical and Practical Notes

s c o r p a n

Raster stacks

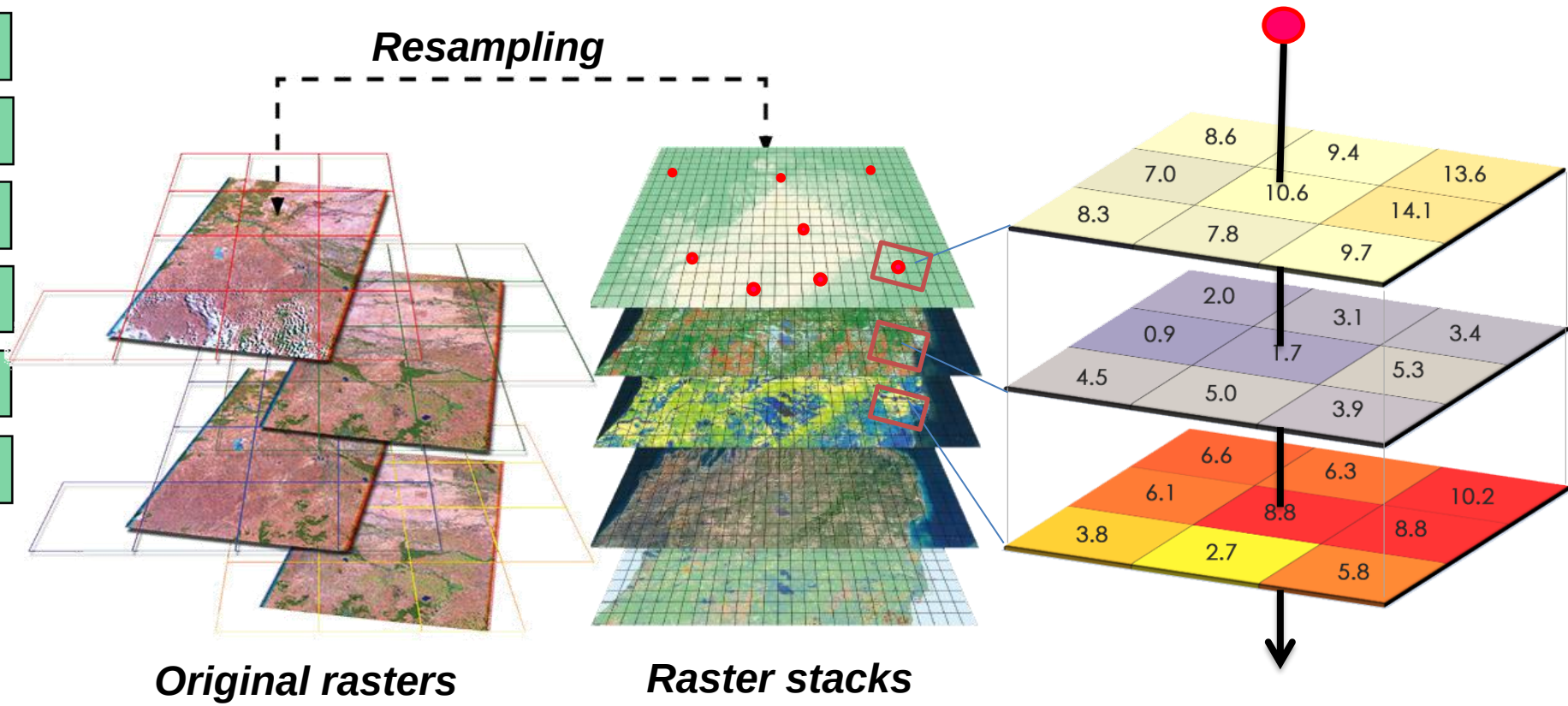


Original rasters ***Raster stacks***

Technical and Practical Notes

s c o r p a n

4. Overlaying raster stacks and points



Original rasters ***Raster stacks***

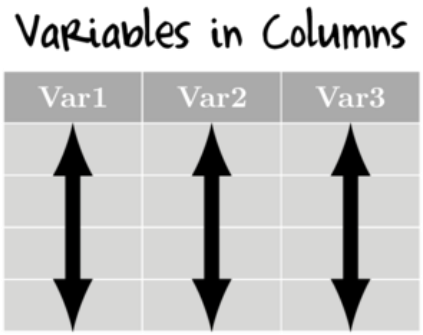
Technical and Practical Notes

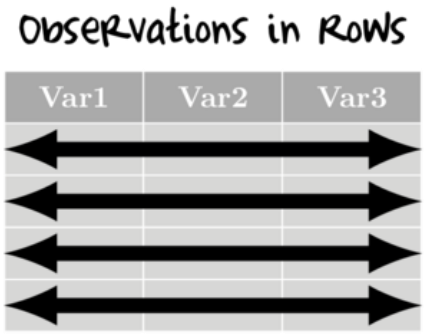
Geo-database
Tidy data
(data.frame)

Predictors
Covariates
Independent variables
Features

20 soil samples
10 covariates

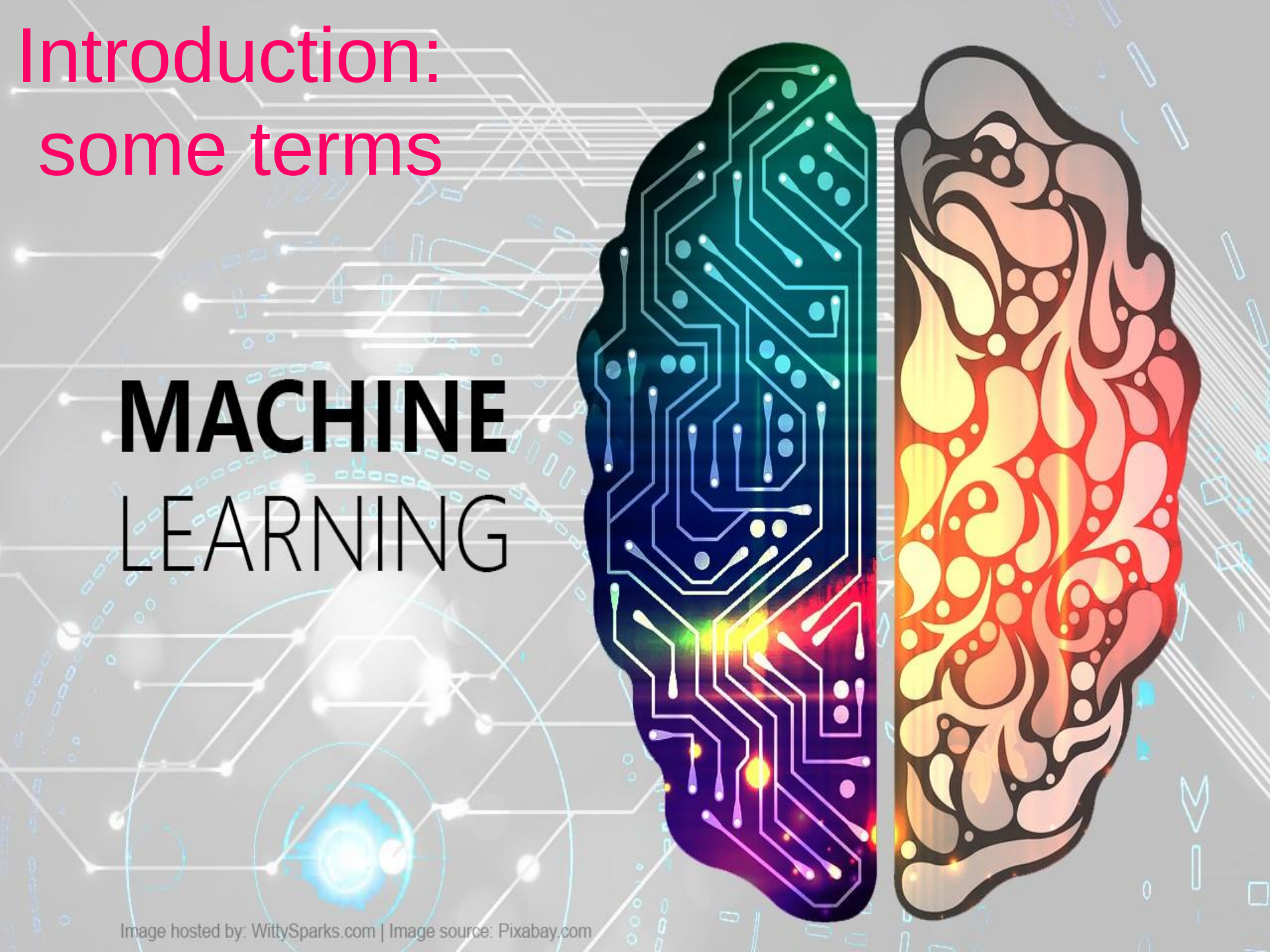
X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	Class
-0.60	3.06	-5.77	0.69	1.46	1.22	1.94	0.61	-0.07	-2.02	-0.87	0.71	0.89	-0.91	0.21	a
0.61	1.61	0.21	2.74	0.46	1.29	0.13	0.55	-0.60	-0.53	0.70	-0.41	-0.62	1.17	-0.32	b
3.81	0.99	-1.21	0.30	0.11	-0.34	-0.29	0.15	-0.09	1.30	0.08	0.66	-0.44	0.45	-0.13	a
0.50	0.14	-0.30	0.64	-1.28	0.28	0.76	-0.61	0.10	-0.07	0.41	-0.53	-0.62	0.06	-0.52	c
1.46	0.23	0.17	0.53	-0.94	0.37	0.64	-0.83	0.20	-0.06	0.18	-0.35	-0.64	0.39	-0.22	a
3.07	-0.19	1.31	0.84	-0.66	1.05	0.94	-1.12	-0.23	-0.31	0.45	-0.54	-0.16	0.38	0.53	a



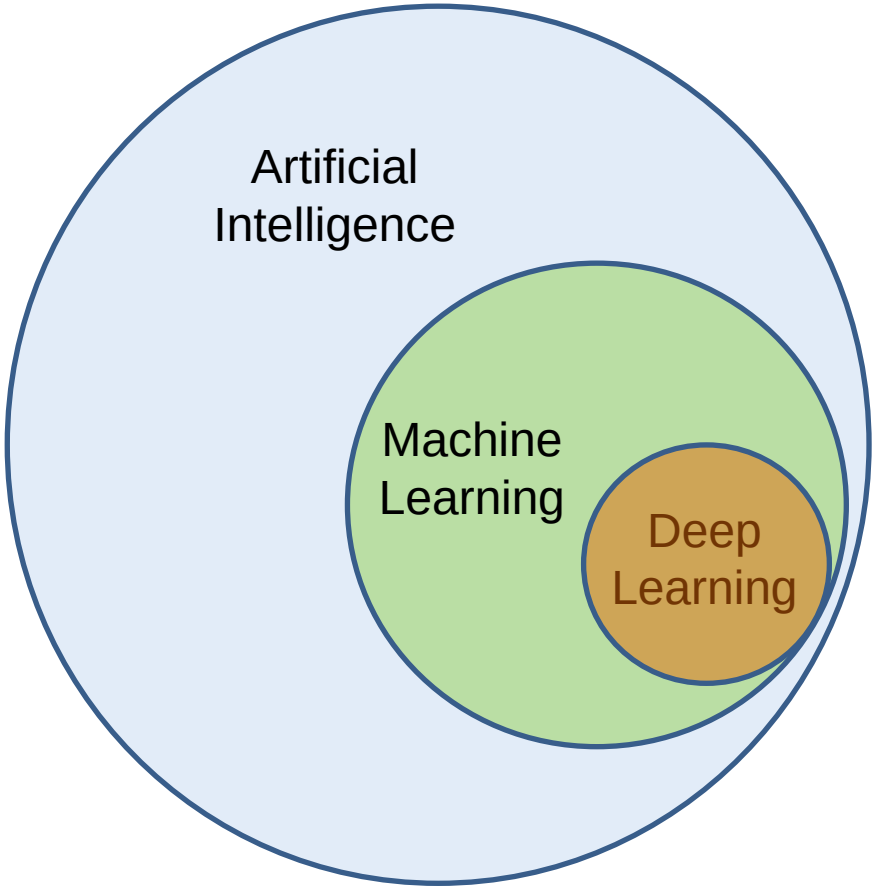


Response variable
Target variable
Dependent variable
(Soil clay or soil types)

Introduction: some terms



AI, ML, and DL



Artificial Intelligence

A technique which enables machines to mimic human behavior

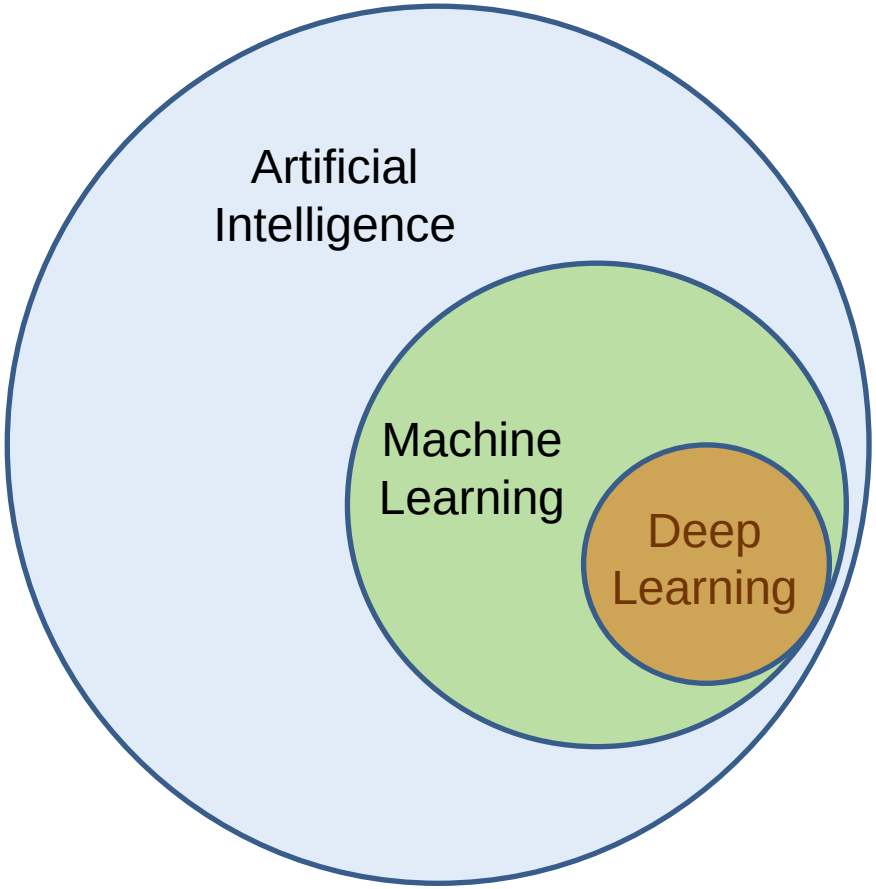
Machine Learning

Subset of AI technique which use statistical methods to enable machines to improve with training and experience

Deep learning

Subset of ML technique where a system can train itself to perform tasks

AI, ML, and DL



Artificial Intelligence

A technique which enables machines to mimic human behavior

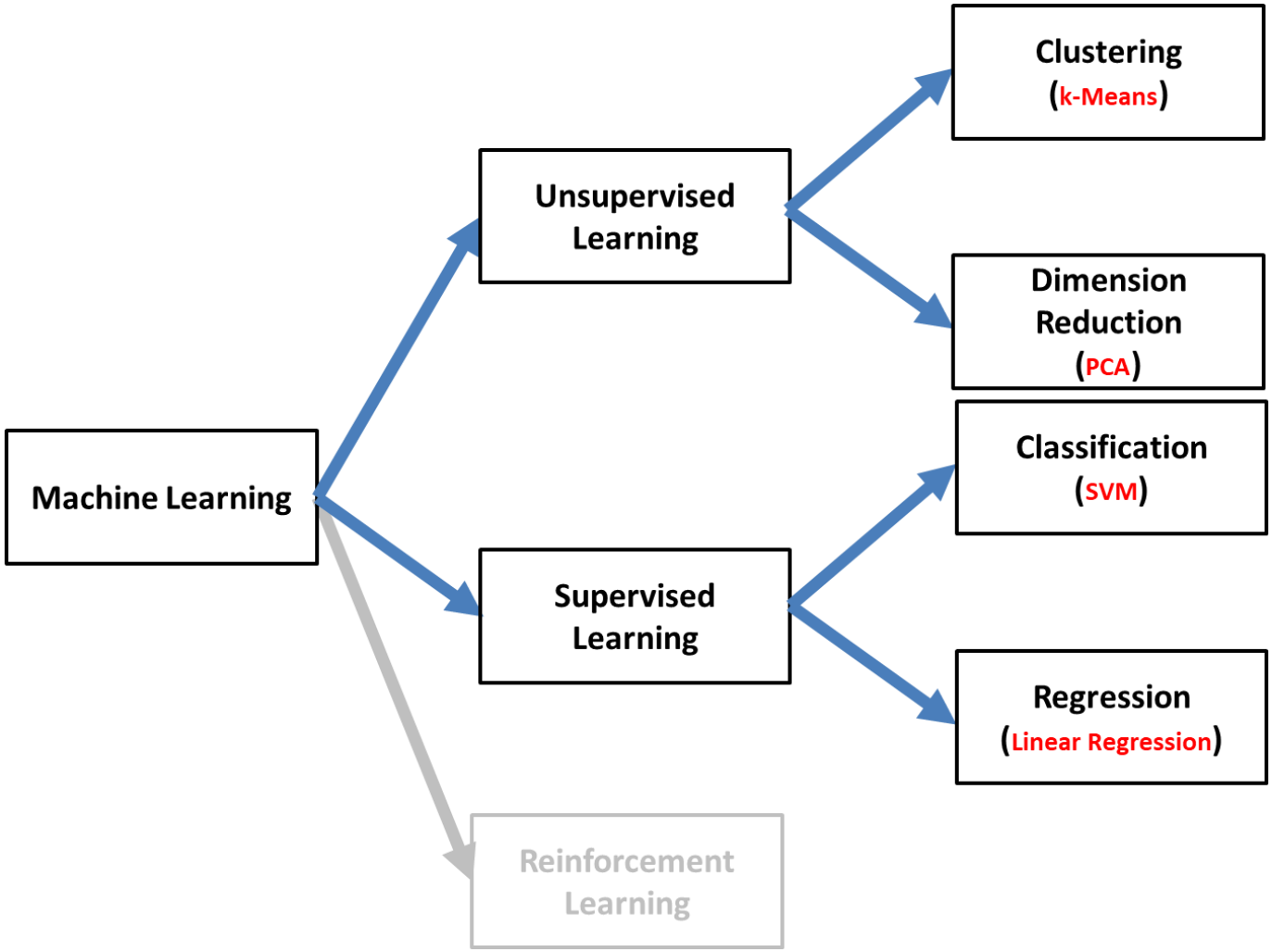
Machine Learning

Subset of AI technique which use statistical methods to enable machines to improve with training and experience

Deep learning

Subset of ML technique where a system can train itself to perform tasks

Types of Machine Learning



Unsupervised Learning

Unsupervised learning

- No response, **just** "*covariates*", (x_i)
- Algorithm identifies clusters
- No training data required
- e.g., clustering of satellite image by similar values

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	Class
-0.60	3.06	-5.77	0.69	1.46	1.22	1.94	0.61	-0.07	-2.02	-0.87	0.71	0.89	-0.91	0.21	a
0.61	1.61	0.21	2.74	0.46	1.29	0.13	0.55	-0.60	-0.53	0.70	-0.41	-0.62	1.17	-0.32	b
3.81	0.99	-1.21	0.30	0.11	-0.34	-0.29	0.15	-0.09	1.30	0.08	0.66	-0.44	0.45	-0.13	a
0.50	0.14	-0.30	0.64	-1.28	0.28	0.76	-0.61	0.10	-0.07	0.41	-0.53	-0.62	0.06	-0.52	c
1.46	0.23	0.17	0.53	-0.94	0.37	0.64	-0.83	0.20	-0.06	0.18	-0.35	-0.64	0.39	-0.22	a
3.07	-0.19	1.31	0.84	-0.66	1.05	0.94	-1.12	-0.23	-0.31	0.45	-0.54	-0.16	0.38	0.53	a

Advantages

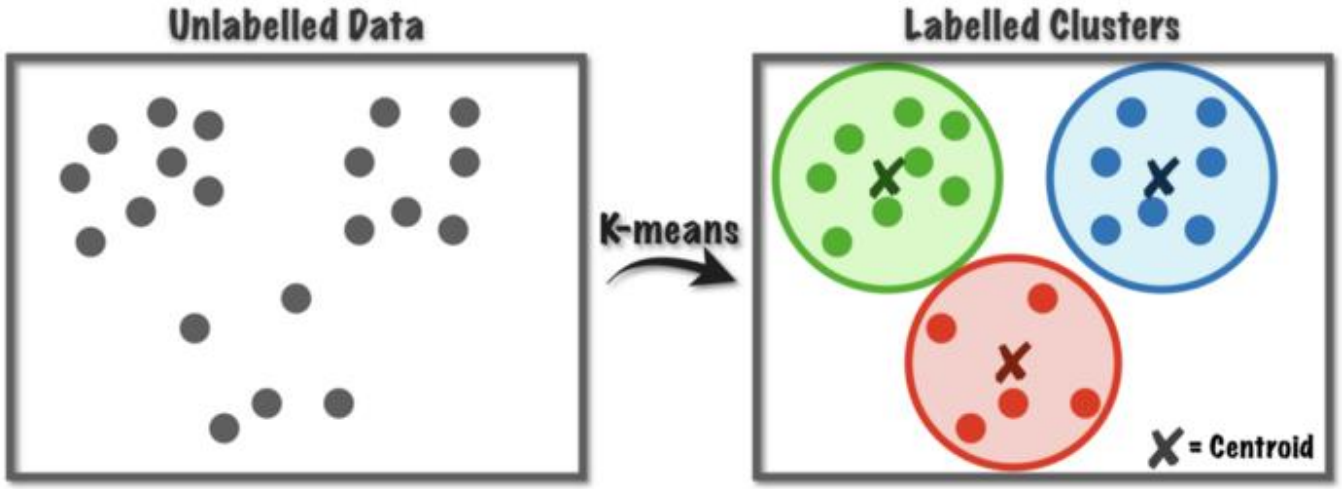
- Unsupervised classification is fairly quick and easy to run
- There is no extensive prior knowledge of area required

Disadvantages

- The classes do not always correspond to informational classes
- The user also has to spend time interpreting and label the classes following the classification

Clustering

- Clustering can be considered the most important unsupervised learning problem, it deals with finding a structure in a collection of unlabeled data.
- Clustering algorithms group the samples of a set such that two samples in the same cluster are more similar to one another than two samples from different clusters
- One option: small Euclidean Distance (squared)



A data set with clear cluster structure

k-means Clustering

Basic Algorithm:

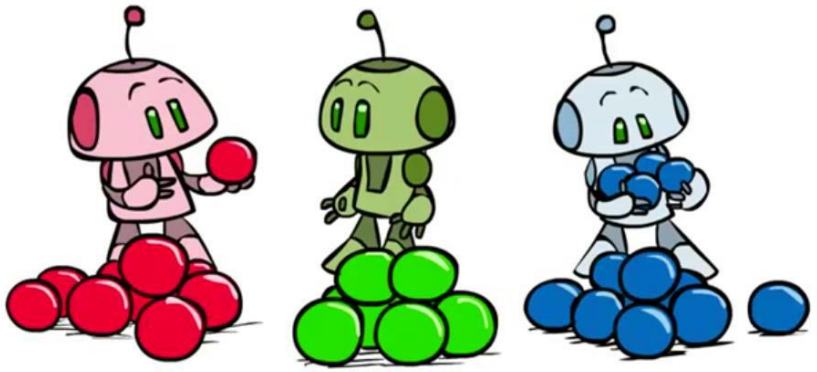
- Step 0: select K
- Step 1: randomly select k initial cluster centers
- Step 2: calculate distance from each object to each cluster centers.
- Step 3: assign each object to the closest cluster
- Step 4: compute the new centroid for each cluster

Iterate:

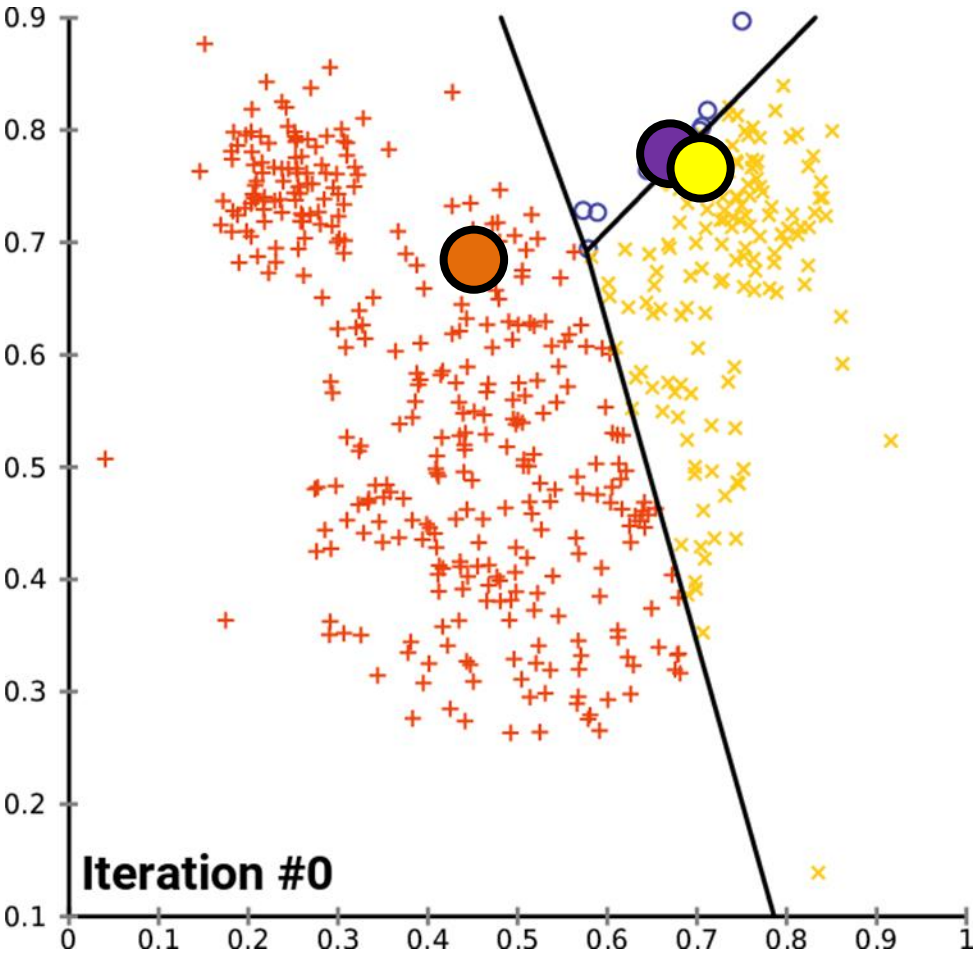
- calculate distance from objects to cluster centroids.
- assign objects to closest cluster
- recalculate new centroids

Stop based on convergence criteria

- no change in clusters
- max iterations

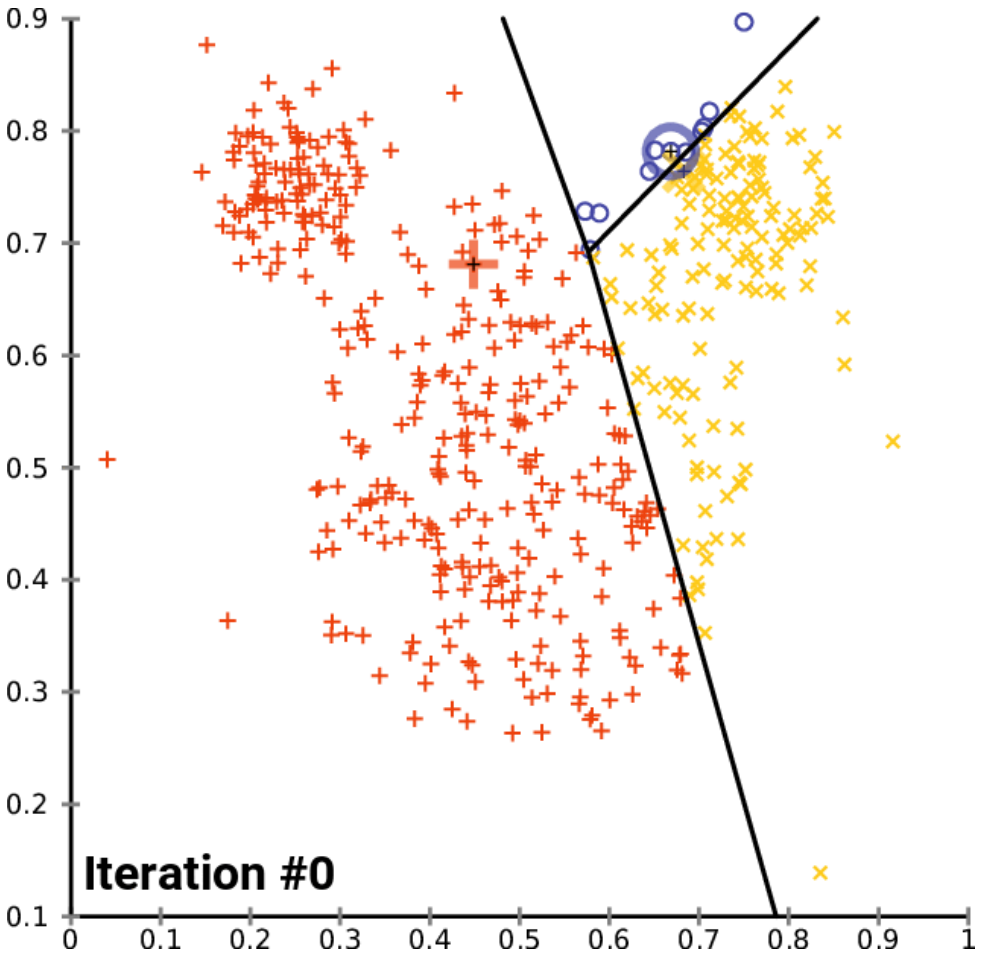


k-means Clustering



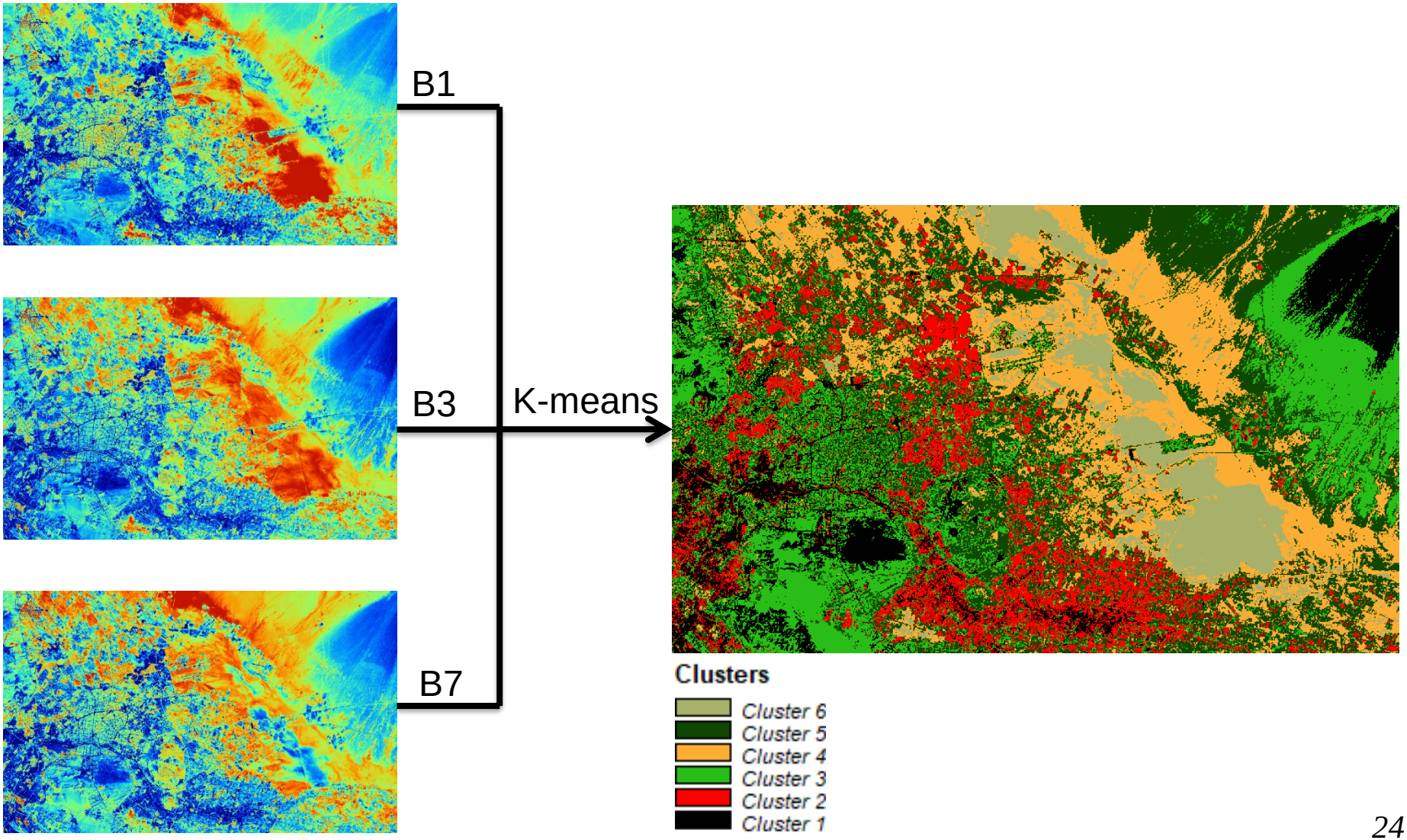
https://commons.wikimedia.org/wiki/File:K-means_convergence.gif

k-means Clustering



https://commons.wikimedia.org/wiki/File:K-means_convergence.gif

Clustering: example



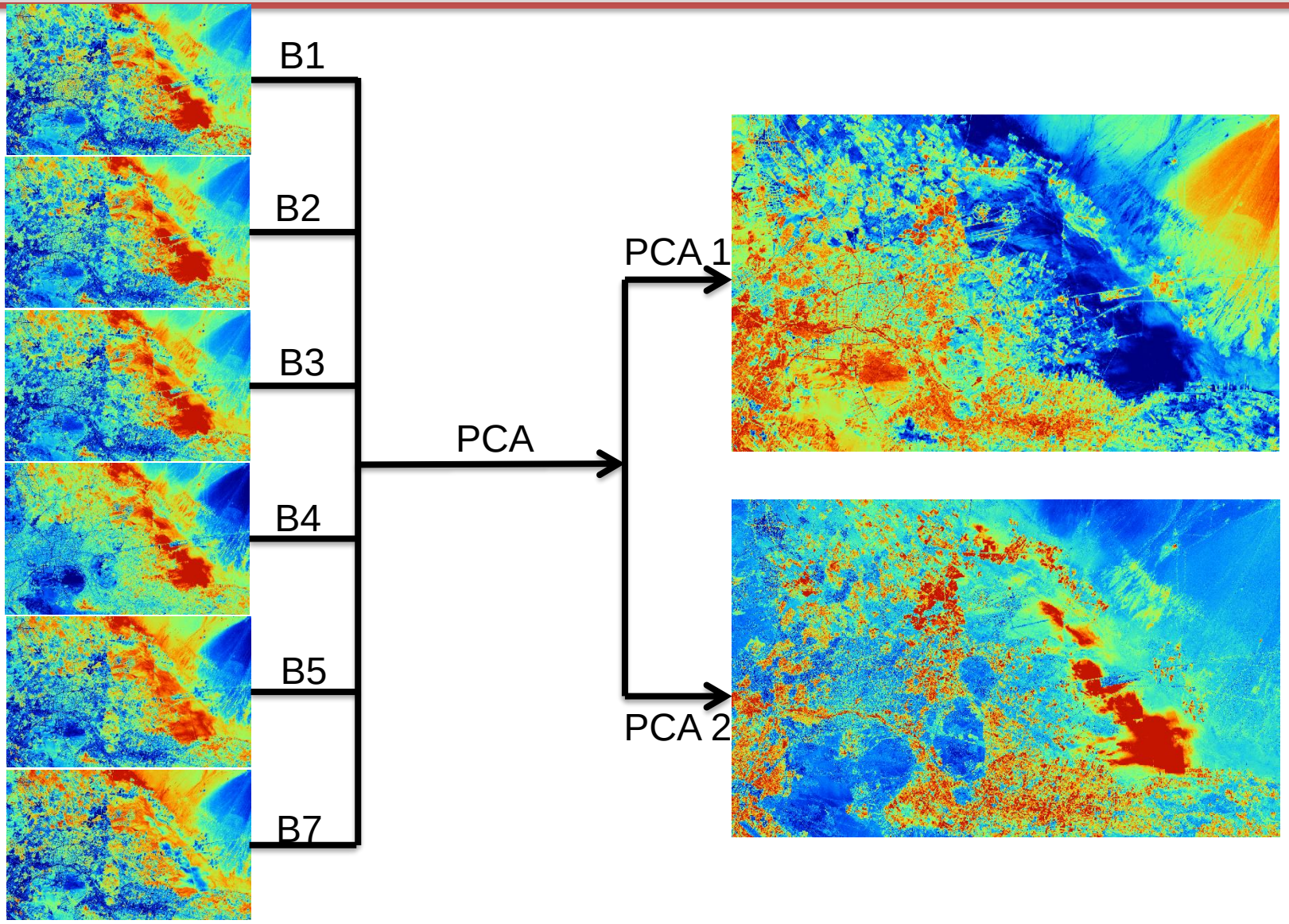
Dimension Reduction

Principal Components Analysis Ideas (PCA)

- It is often necessary to reduce the dimension of a dataset, a large number of covariates
- Transform some large number of covariates into a smaller number of uncorrelated covariates called principal components (PCs).
- Developed to capture as much of the variation in data as possible (minimal loss of information)

▪See online tutorials such as
http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

PCA: example



Supervised Learning

Supervised learning

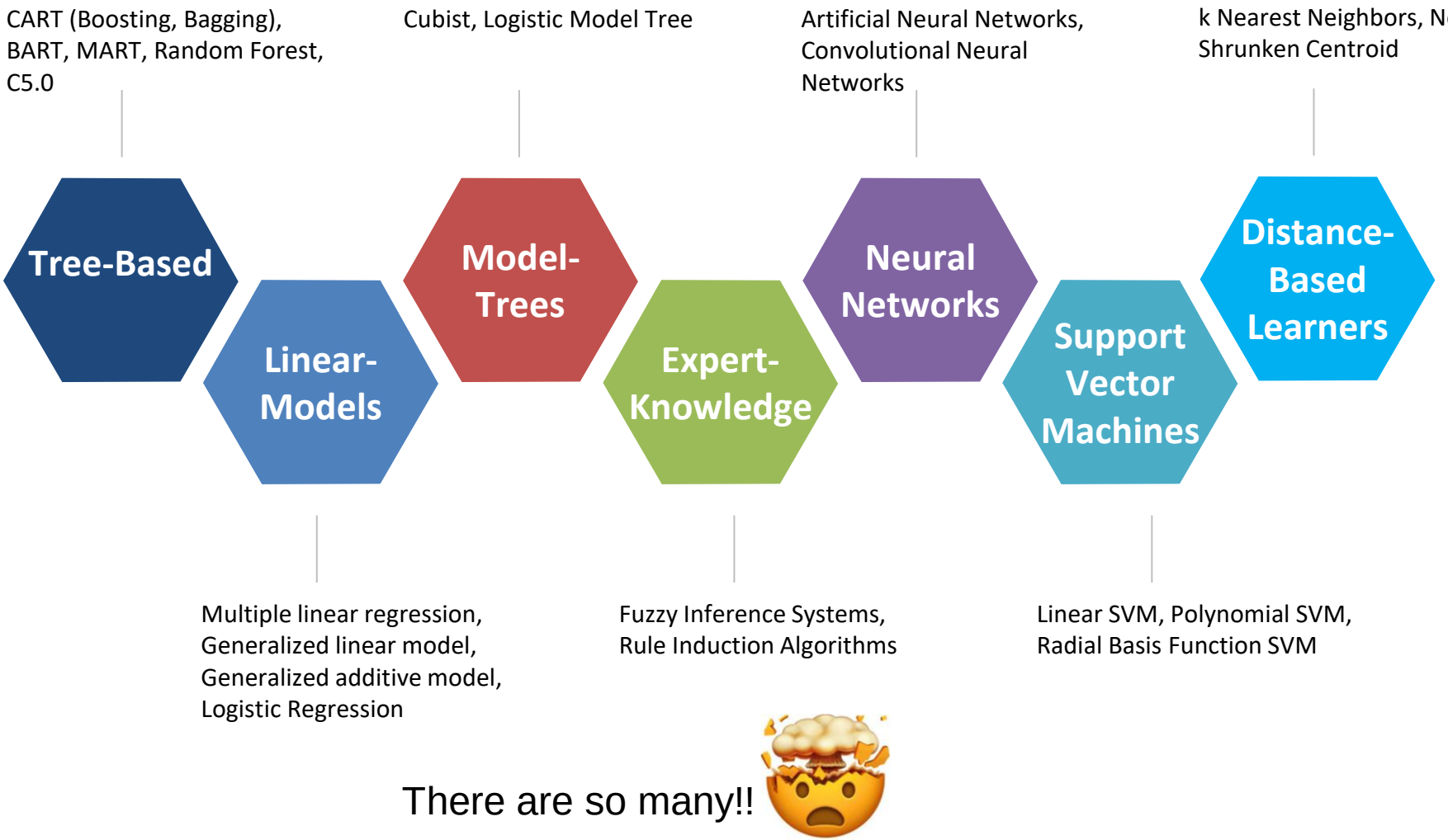
- For each covariate value $\mathbf{x}_i$ there is also a response value $\mathbf{y}_i$
- → e.g. random forest
- → what we usually do for soil mapping

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	Class
-0.60	3.06	-5.77	0.69	1.46	1.22	1.94	0.61	-0.07	-2.02	-0.87	0.71	0.89	-0.91	0.21	a
0.61	1.61	0.21	2.74	0.46	1.29	0.13	0.55	-0.60	-0.53	0.70	-0.41	-0.62	1.17	-0.32	b
3.81	0.99	-1.21	0.30	0.11	-0.34	-0.29	0.15	-0.09	1.30	0.08	0.66	-0.44	0.45	-0.13	a
0.50	0.14	-0.30	0.64	-1.28	0.28	0.76	-0.61	0.10	-0.07	0.41	-0.53	-0.62	0.06	-0.52	c
1.46	0.23	0.17	0.53	-0.94	0.37	0.64	-0.83	0.20	-0.06	0.18	-0.35	-0.64	0.39	-0.22	a
3.07	-0.19	1.31	0.84	-0.66	1.05	0.94	-1.12	-0.23	-0.31	0.45	-0.54	-0.16	0.38	0.53	a

Supervised learning

- – Regression: continuous responses, e.g. soil clay content, rainfall
- – Classification: categorical responses (binary or multinomial), e.g. soil type

Available Models



There are so many!! 🤯

Available Models in R

- The caret package is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for:

1. data splitting
2. pre-processing
3. feature selection
4. model tuning using resampling
5. variable importance estimation

6 Available Models

The models below are available in `train`. The code behind these protocols can be obtained using the function `getModelInfo` or by going to the github repository.

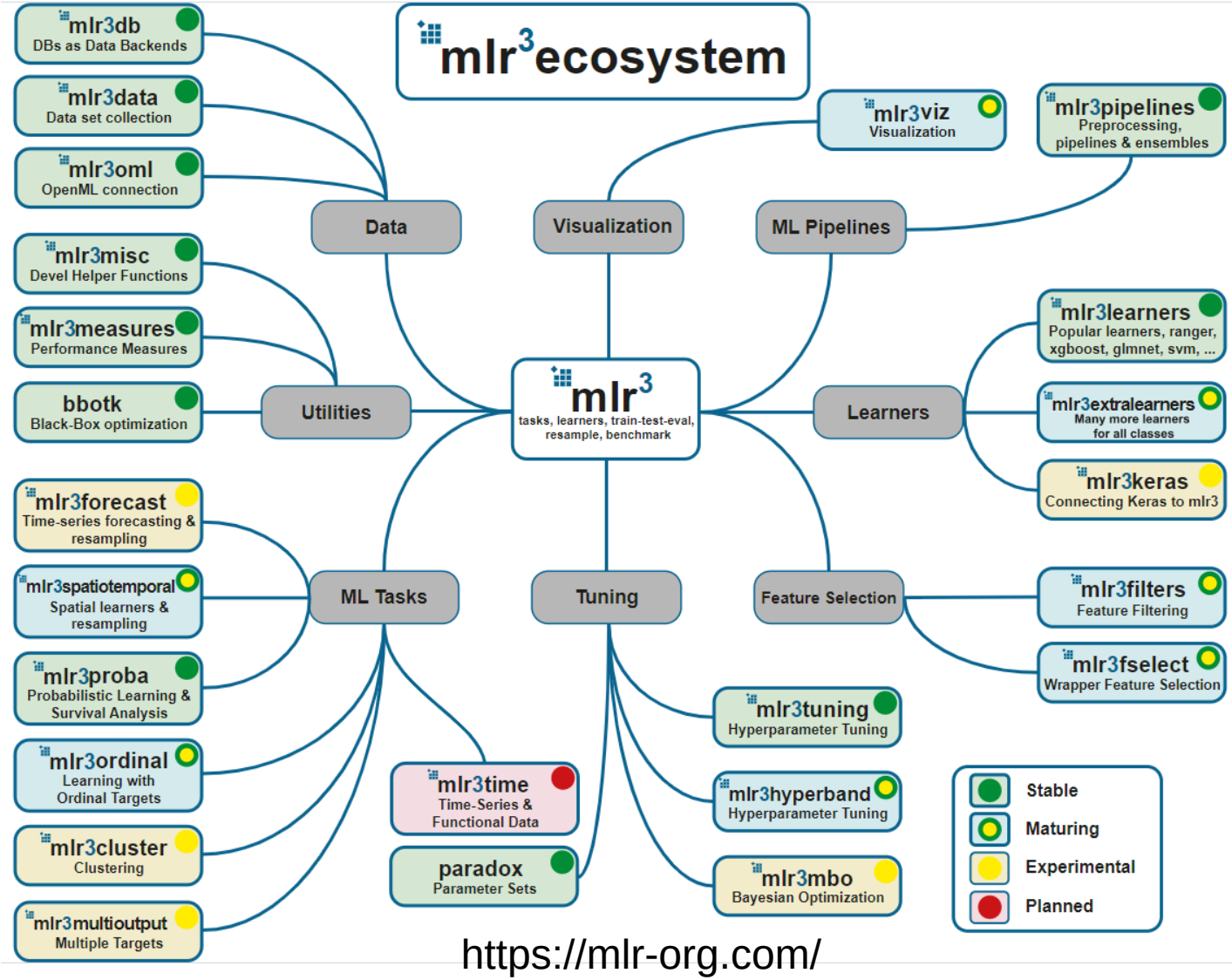
Show 238 entries

Search:

Model	*method* Value	Type	Libraries	Tuning Parameters
AdaBoost Classification Trees	adaboost	Classification	fastAdaboost	nIter, method
AdaBoost.M1	AdaBoost.M1	Classification	adabag, plyr	mfinal, maxdepth, coeflearn
Adaptive Mixture Discriminant Analysis	amdai	Classification	adaptDA	model
Adaptive-Network-Based Fuzzy Inference System	ANFIS	Regression	frbs	num.labels, max.iter
Adjacent Categories Probability Model for Ordinal Data	vglmAdjCat	Classification	VGAM	parallel, link

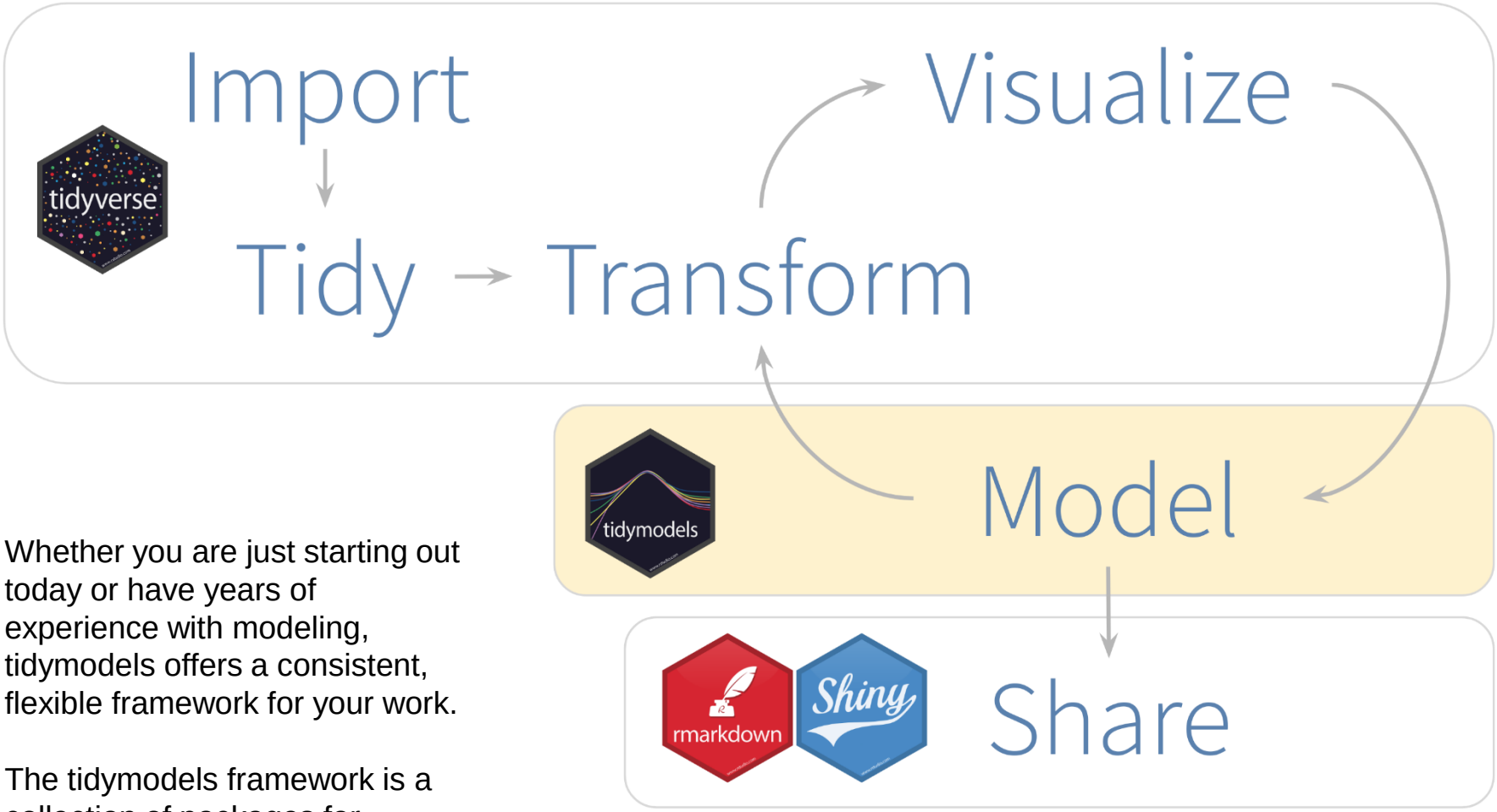
http://topepo.github.io/caret/index.html

Available Models in R



https://mlr-org.com/

Available Models in R



- Whether you are just starting out today or have years of experience with modeling, tidymodels offers a consistent, flexible framework for your work.
- The tidymodels framework is a collection of packages for modeling and machine learning using tidyverse principles.

https://www.tidymodels.org/

Things to Keep in Mind

- Each model predicts one realization of a DSM

- Effectiveness of the model are dependent on:
  - → Study Area (Extent and Resolution)
  - → Target Variable
  - → Environmental Covariates
  - → Model Structure (e.g. Linear vs. Hierarchical)

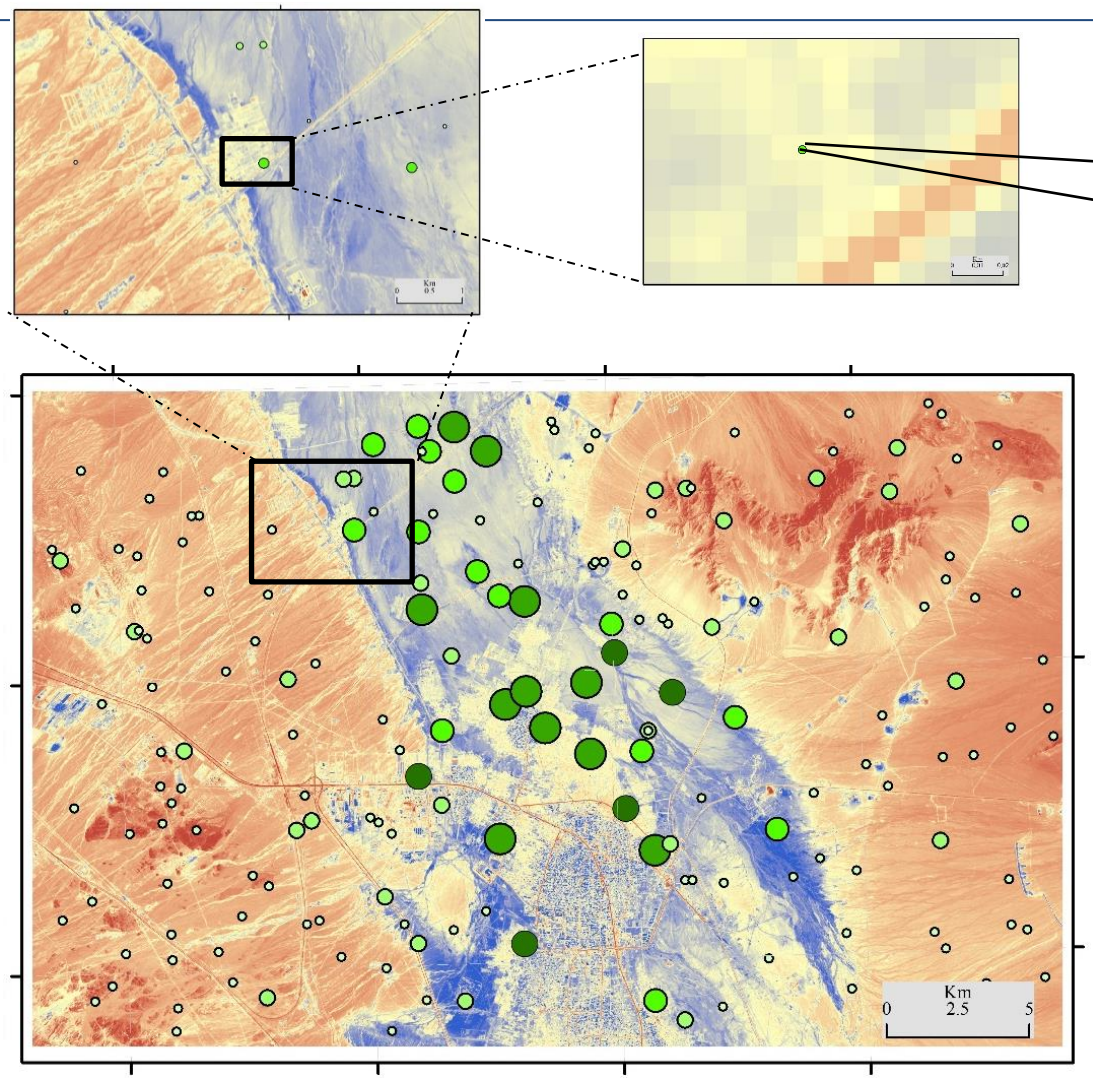
- Each model has its own set of hyperparameters (i.e. model settings) that need to be optimized.

- Some models are more computationally efficient than others.

- Each model may produce drastically different results given the same inputs.

- Some models tend to overfit.

❑ Digital soil mapping

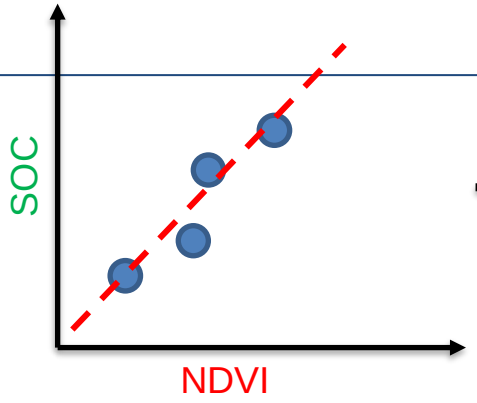


SOC	NDVI
1.2	0.7
0.5	0.2
0.8	0.6
1.5	0.8

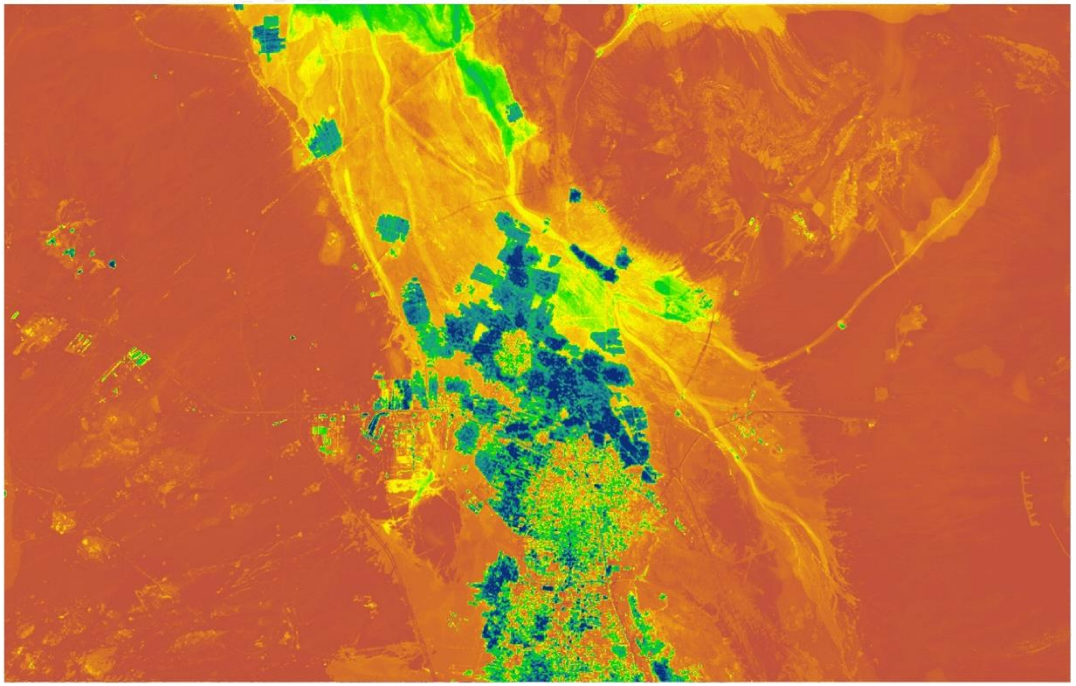
NDVI (covariate)

❑ Digital soil mapping

SOC	NDVI
1.2	0.7
0.5	0.2
0.8	0.6
1.5	0.8



SOC = 0.65 NDVI



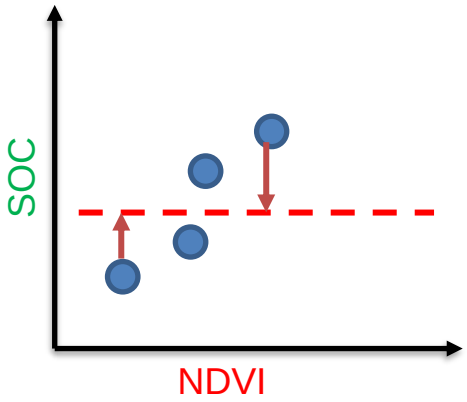
SOC map

❑ Machine learning

SOC = **0.65** NDVI

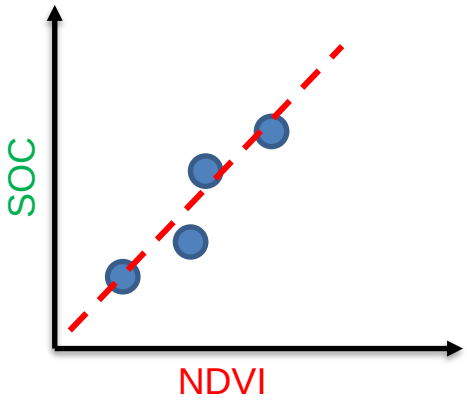
$y = \theta x$

θ = 0.02



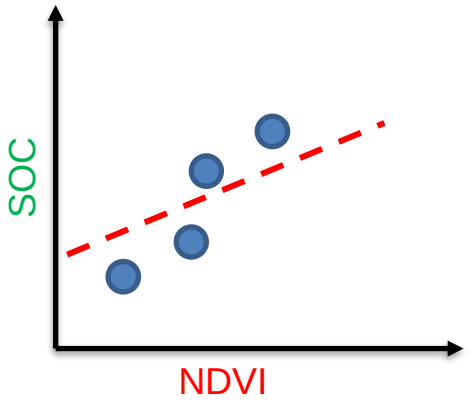
Error = 1

θ = 0.65



Error = 0.2

θ = 0.42

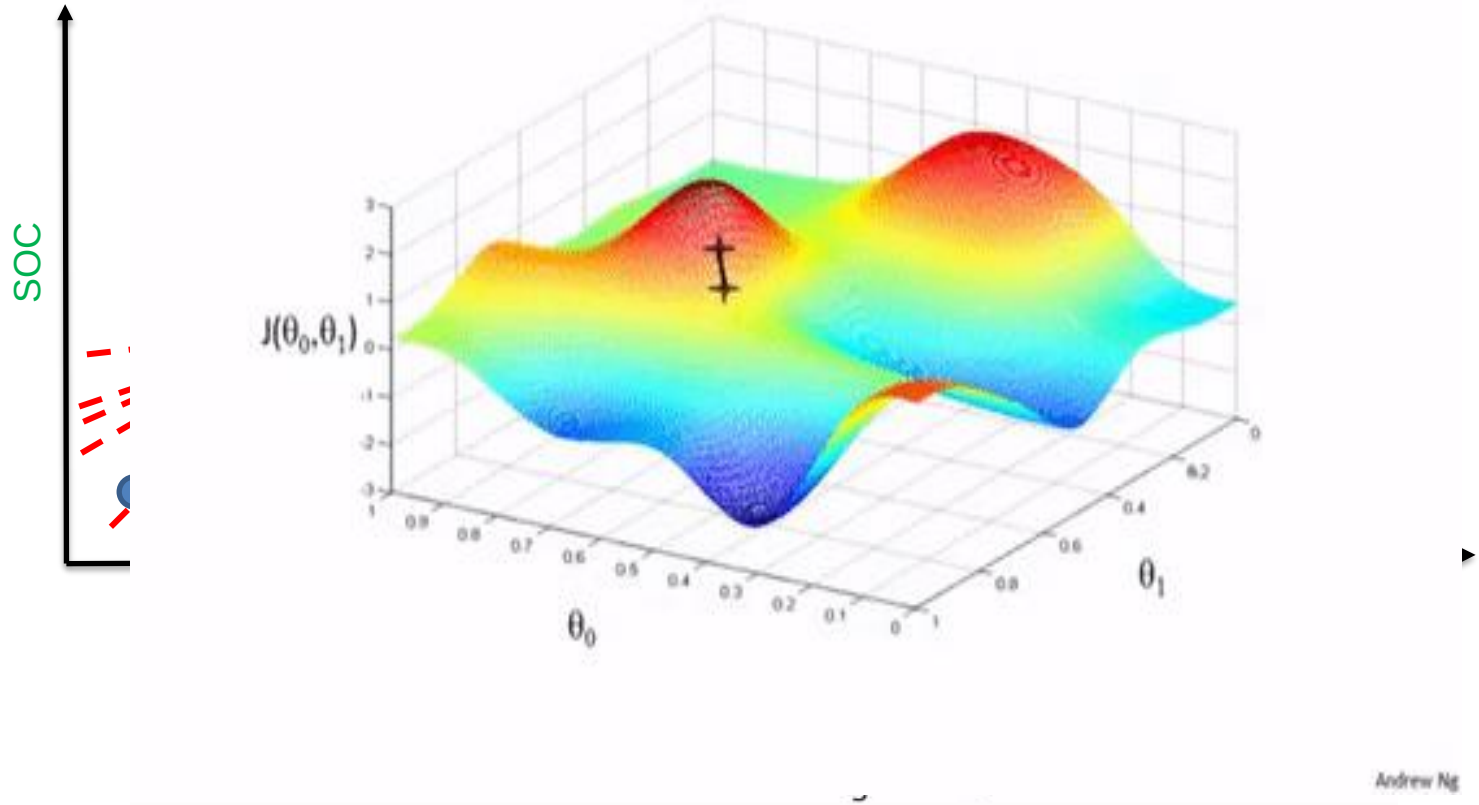


Error = 0.6

❑ Machine learning

SOC = **0.65** NDVI

$y = \theta x$



Things to Keep in Mind

- Each model predicts one realization of a DSM

- Effectiveness of the model are dependent on:
  - → Study Area (Extent and Resolution)
  - → Target Variable
  - → Environmental Covariates
  - → Model Structure (e.g. Linear vs. Hierarchical)

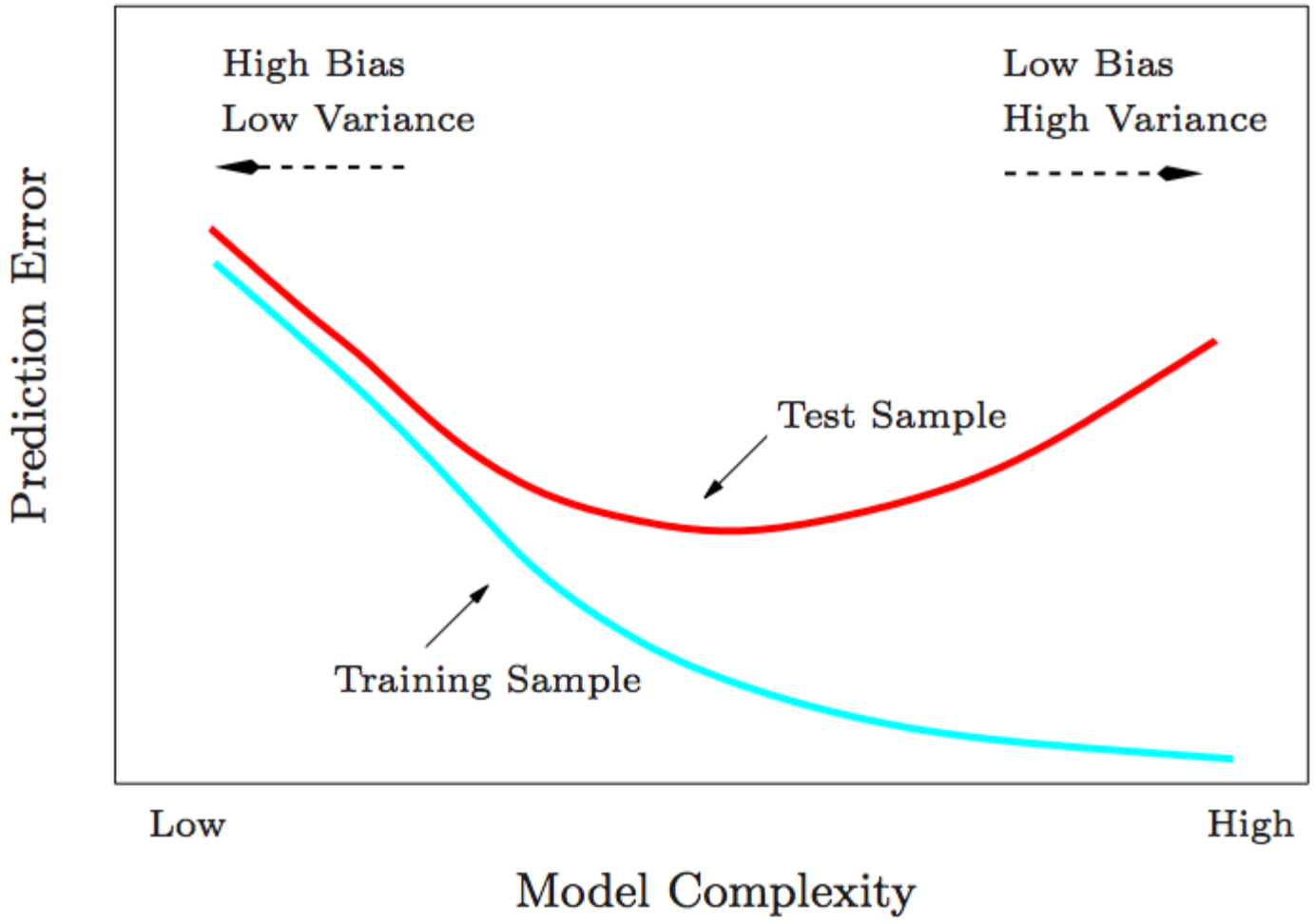
- Each model has its own set of hyperparameters (i.e. model settings) that need to be optimized.

- Some models are more computationally efficient than others.

- Each model may produce drastically different results given the same inputs.

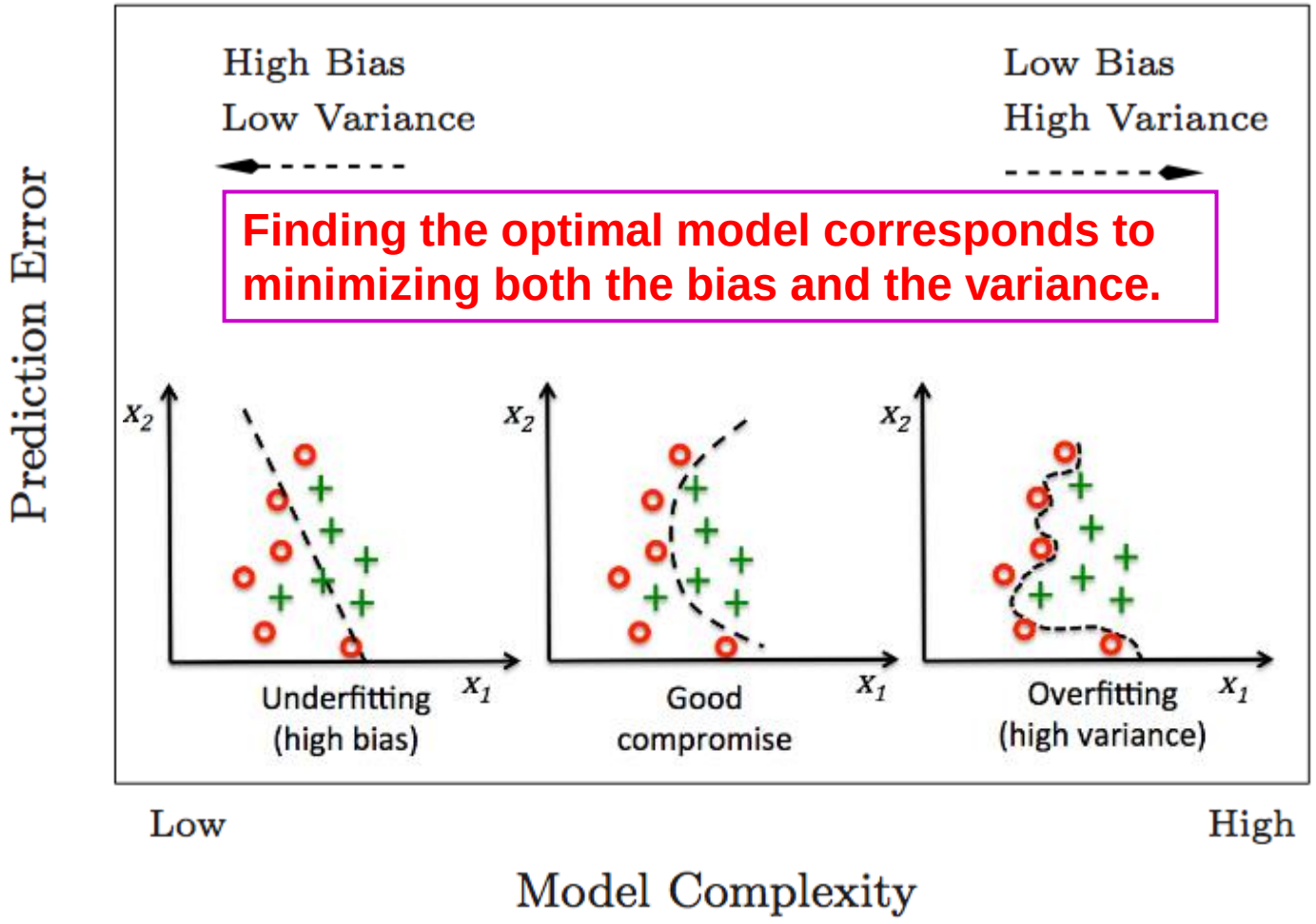
- Some models tend to overfit.

Overfitting? Bias-Variance trade-off



Test and training error as a function of model complexity. (Hastie et al. 2009)

Overfitting? Bias-Variance trade-off



Test and training error as a function of model complexity. (Hastie et al. 2009)

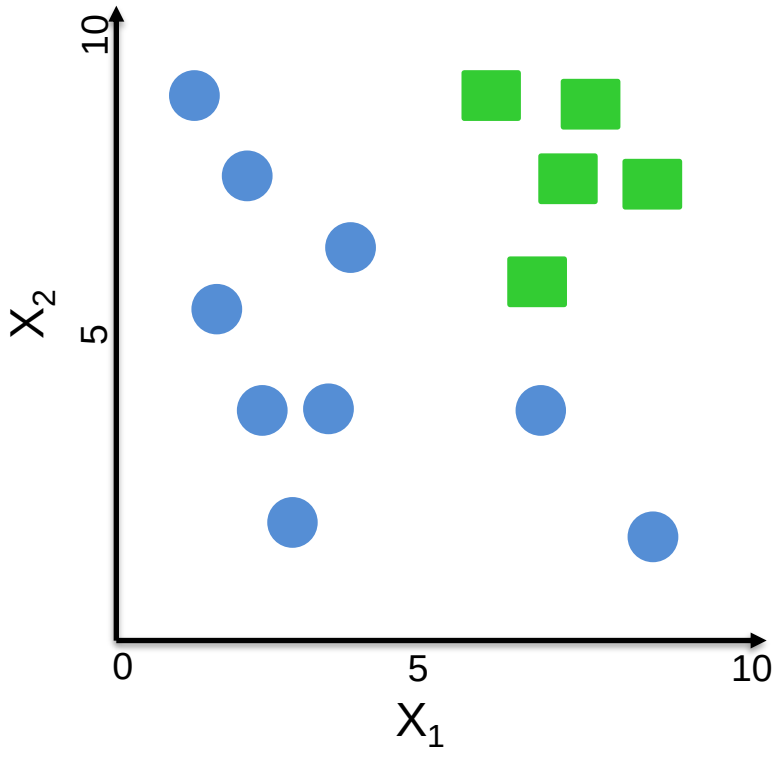
From Available Models

Decision Tree

Decision Tree: Definition

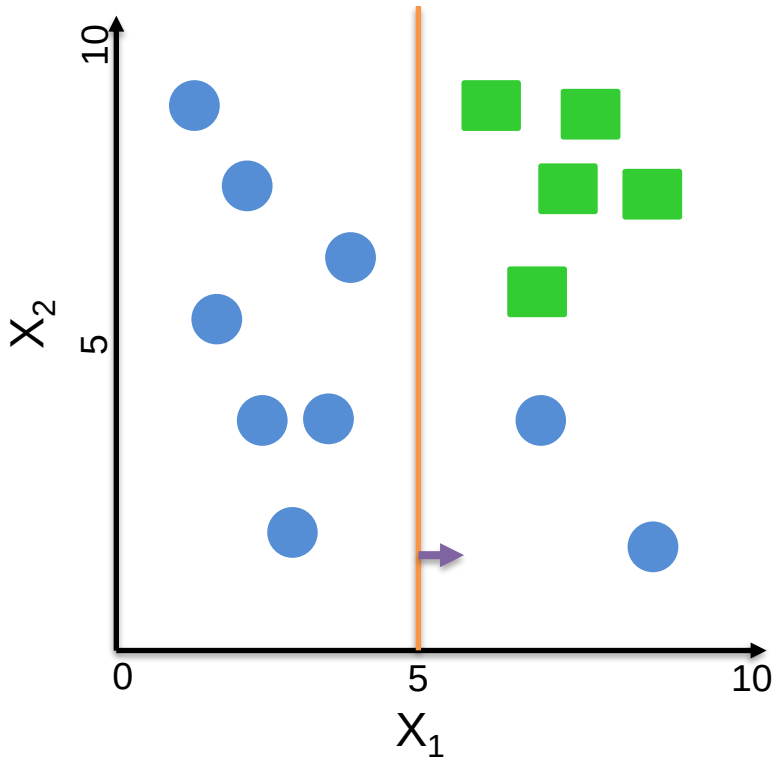
- Decision Tree is a supervised learning approach.
- It can be used for solving regression (e.g., clay content) and classification (e.g., soil types) problems.
- It partitions the data into subsets.
- The partitioning process starts with a binary split (Yes-No) and continues until no further splits can be made.
- It predicts the outcome by asking a set of if-else questions.
- It resembles a tree-like graph, easy to understand

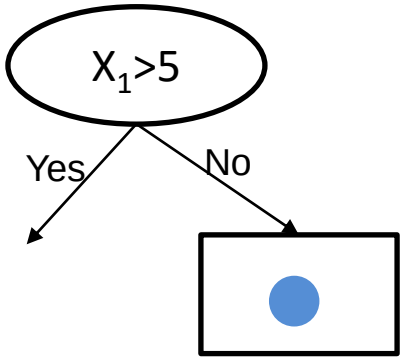
Decision Tree: example I



Example of a dataset and the corresponding decision tree. (Alpaydin, 2010)

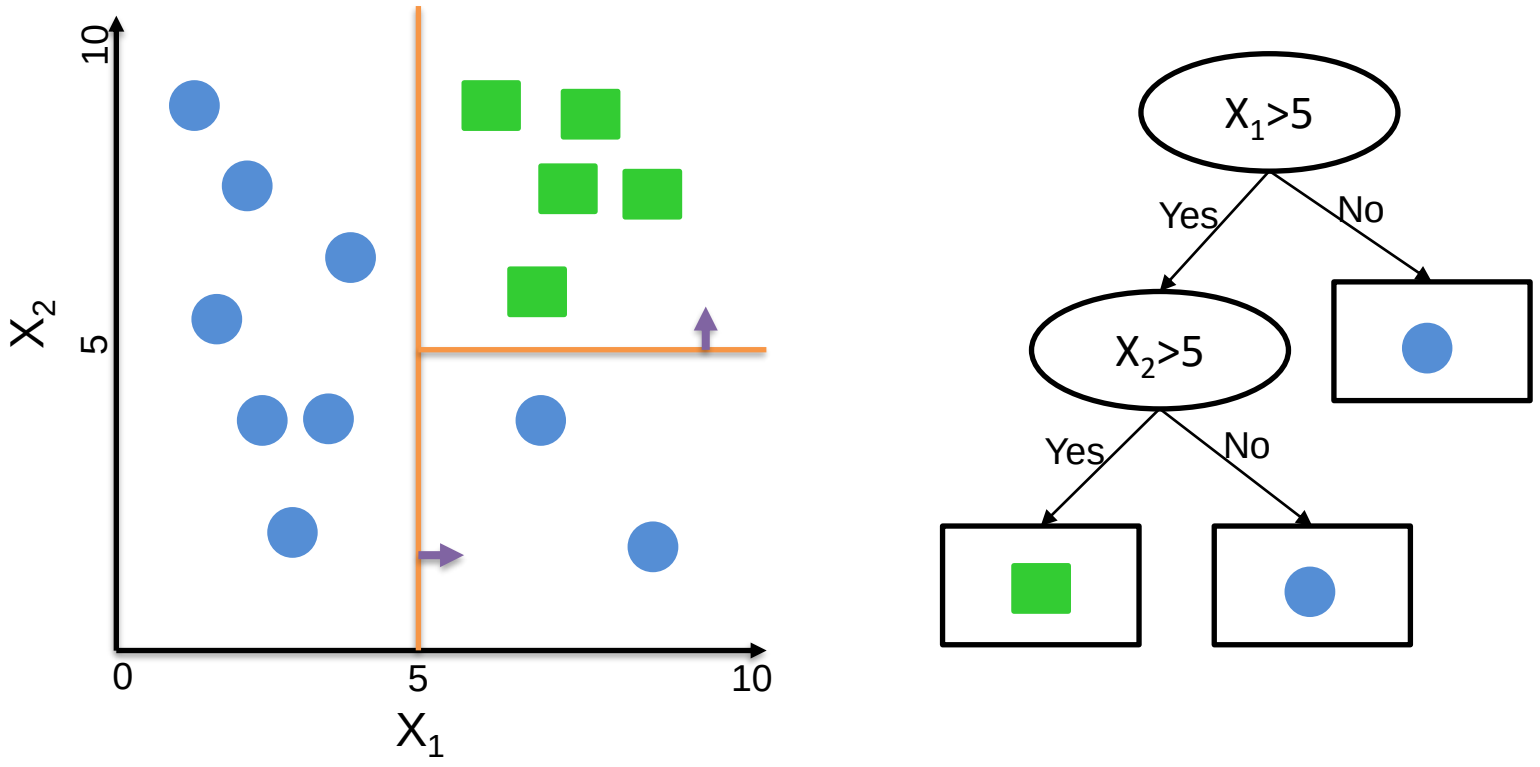
Decision Tree: example I





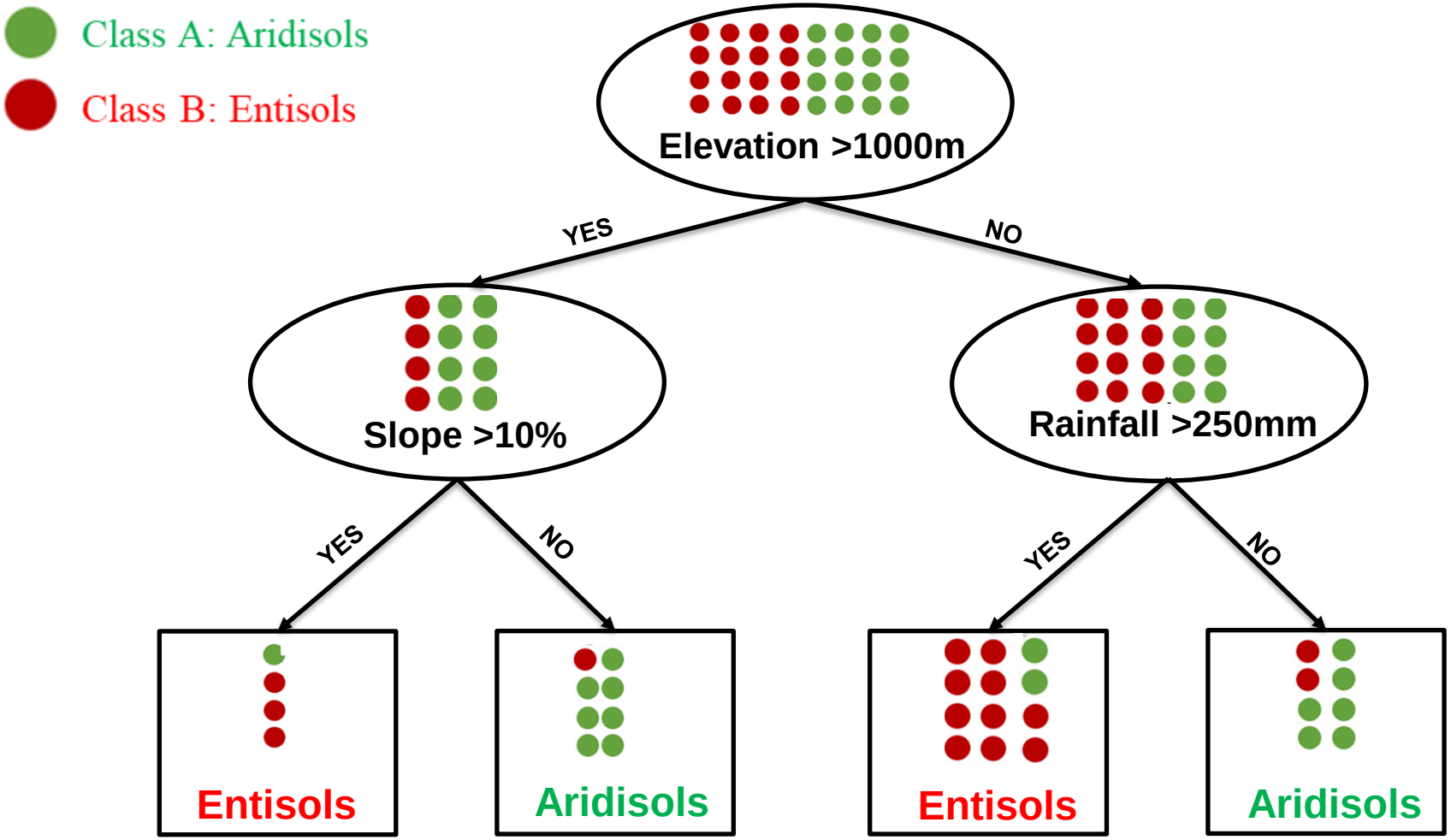
Example of a dataset and the corresponding decision tree. (Alpaydin, 2010)

Decision Tree: example I



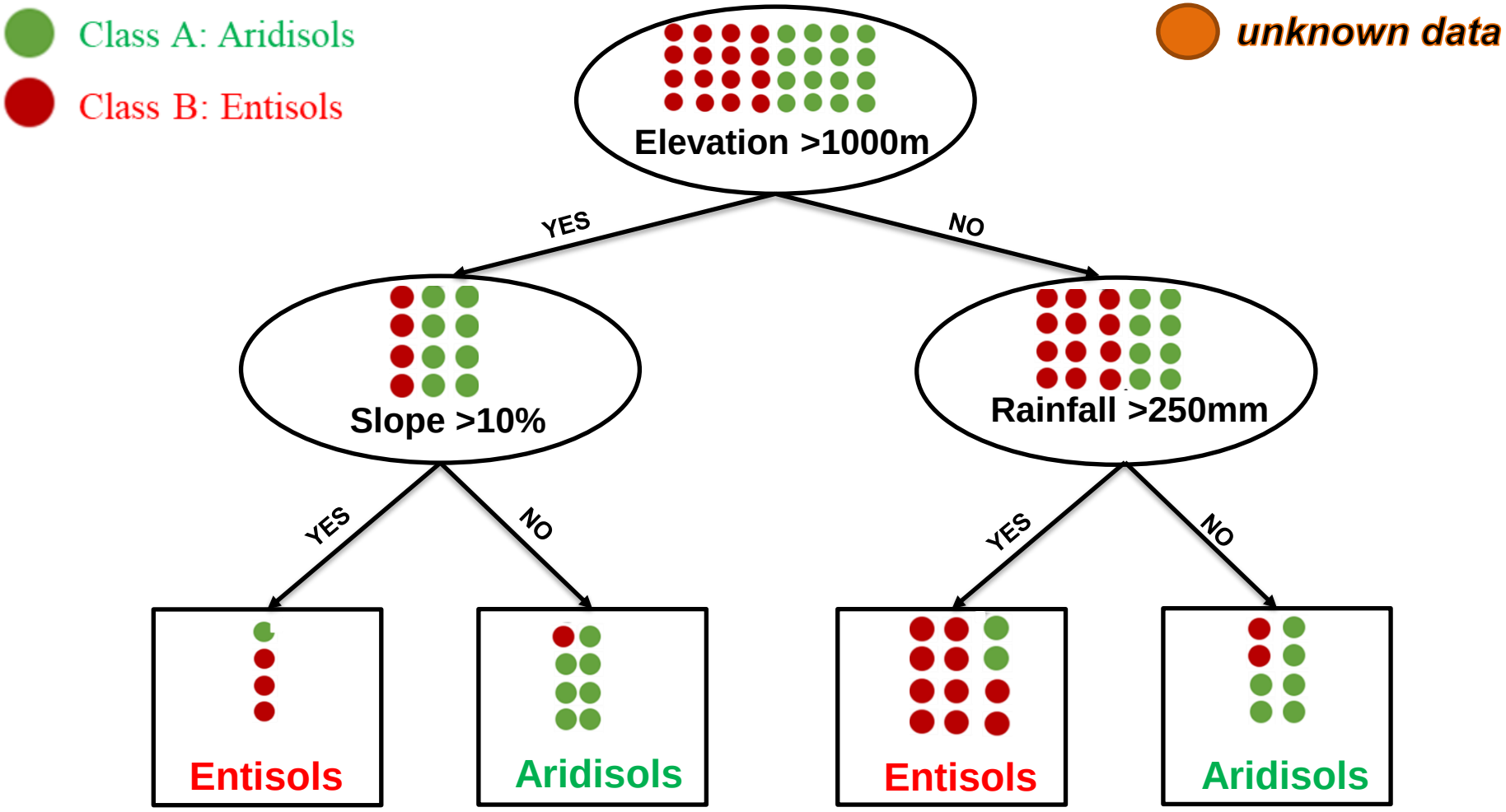
Example of a dataset and the corresponding decision tree. (Alpaydin, 2010)

Decision Tree: example II



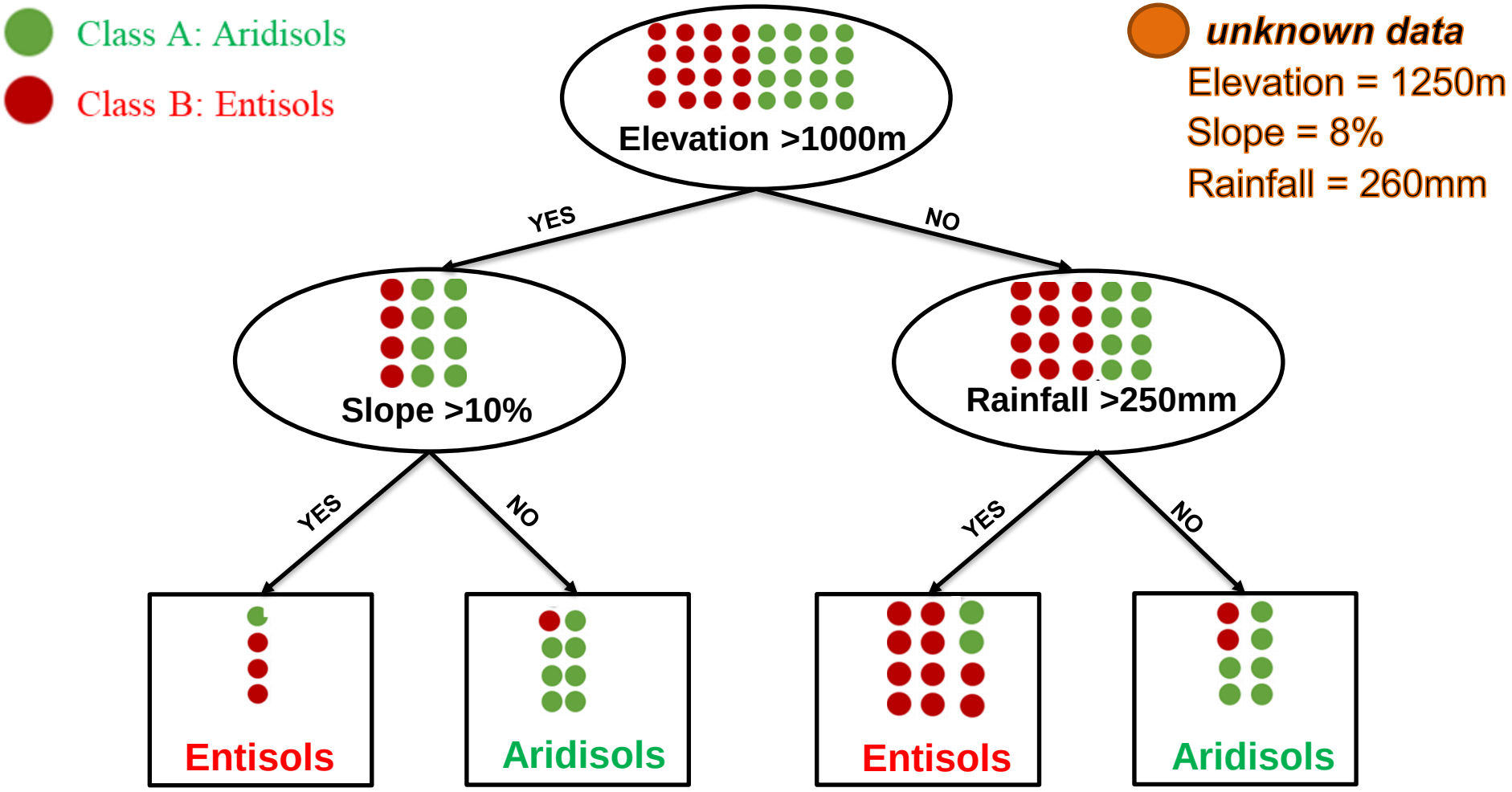
A decision tree model to predict Entisols and Aridisols

Decision Tree: example II



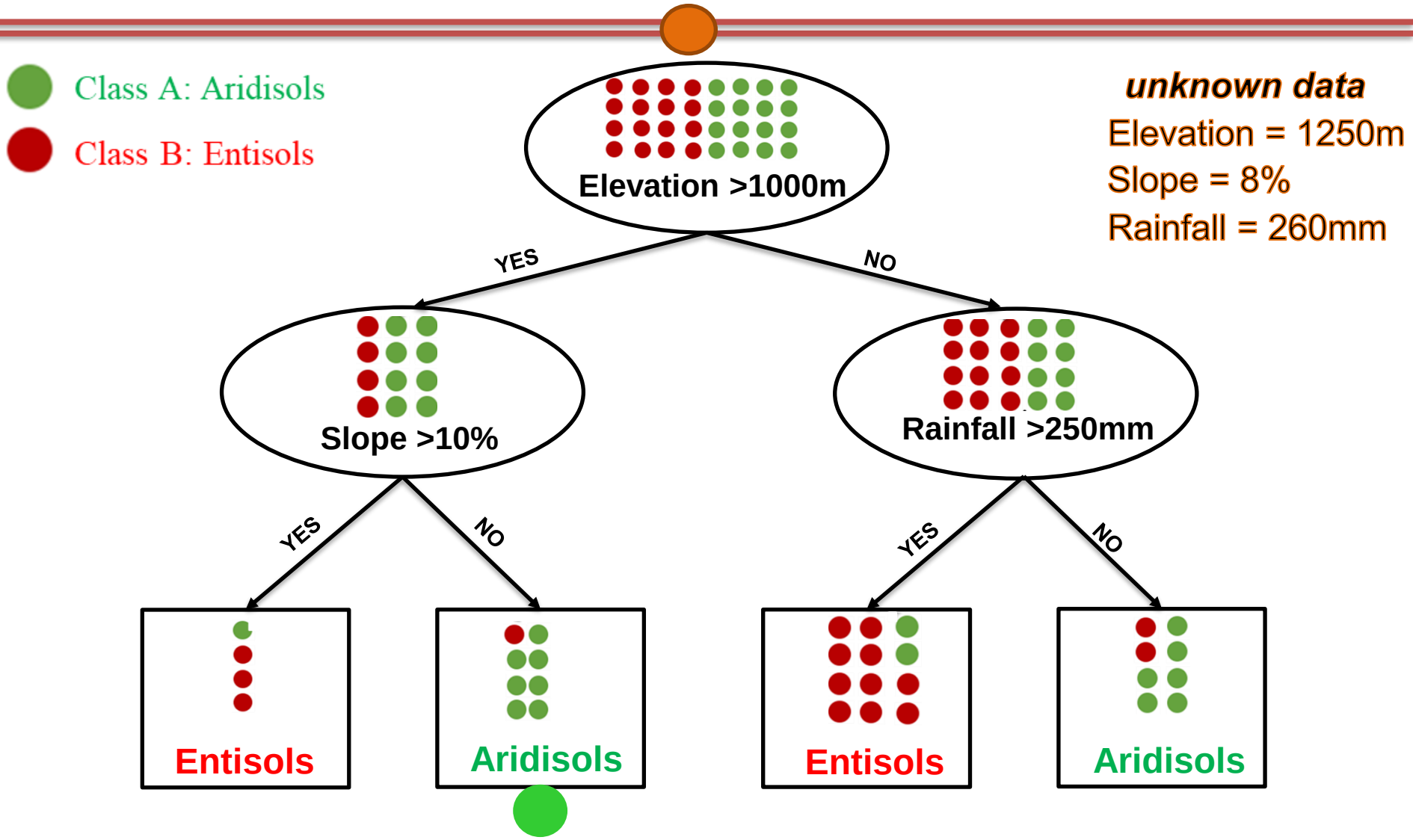
A decision tree model to predict Entisols and Aridisols

Decision Tree: example II



A decision tree model to predict Entisols and Aridisols

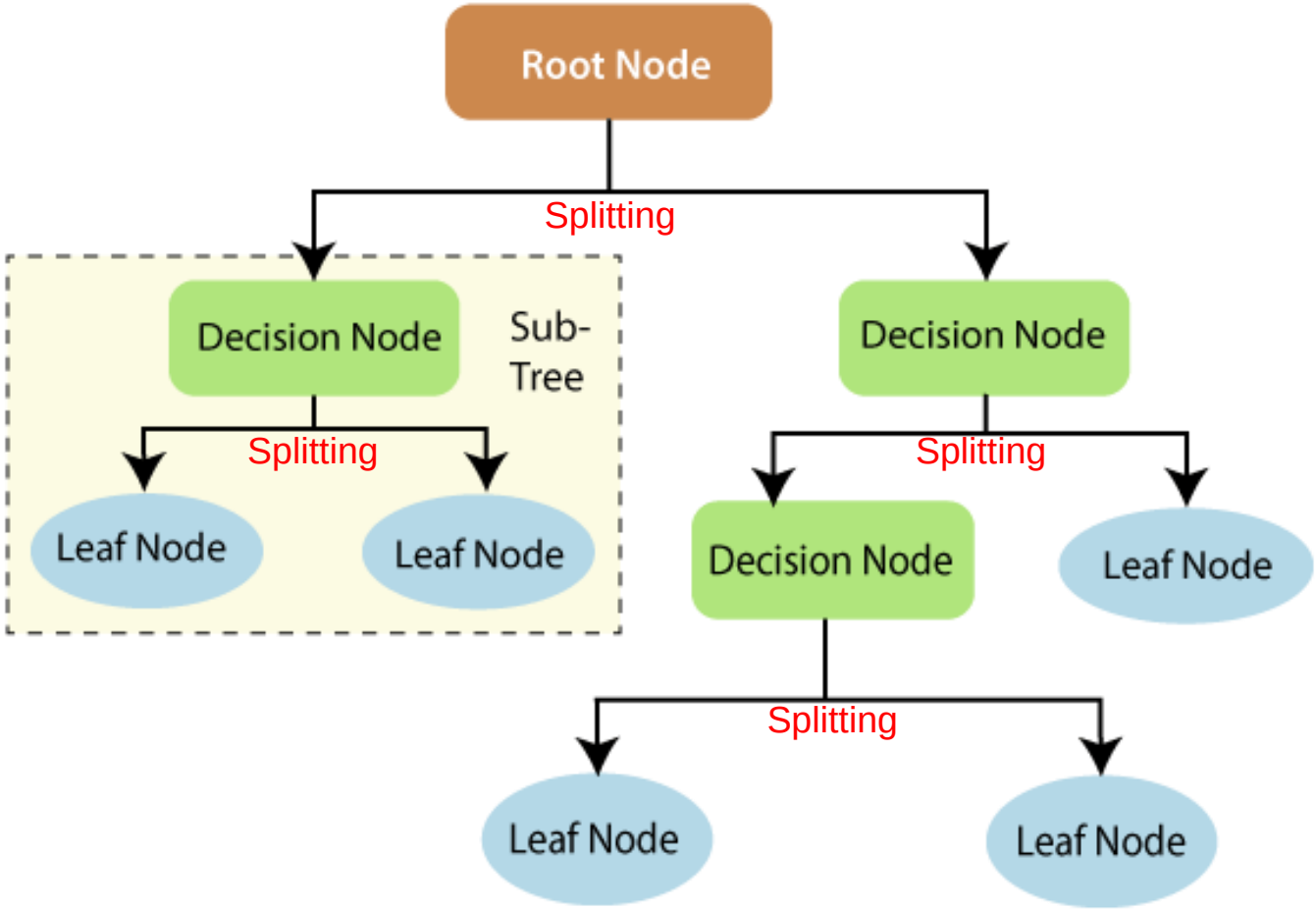
Decision Tree: example II



A decision tree model to predict Entisols and Aridisols

https://www.kdnuggets.com/

Decision Tree: graphic



https://www.kdnuggets.com/

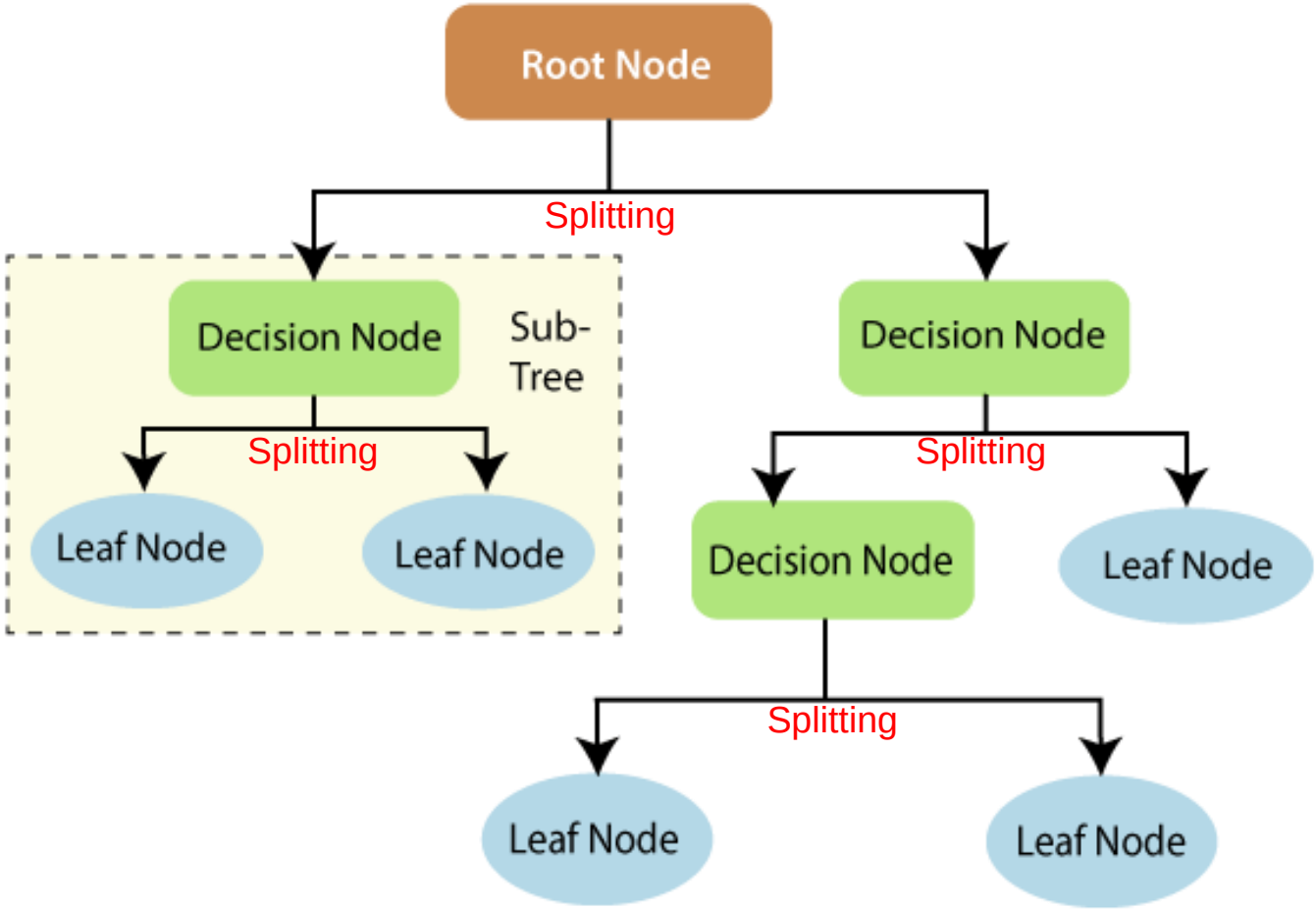
Decision Tree: Terminology

- Root Node: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node.
- Leaf/Terminal Node: Nodes do not split is called Leaf or Terminal node.
- Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.
- Splitting: It is a process of dividing a node into two or more sub-nodes.
- Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
- Branch/Sub-Tree: A subsection of the entire tree is called branch or sub-tree.

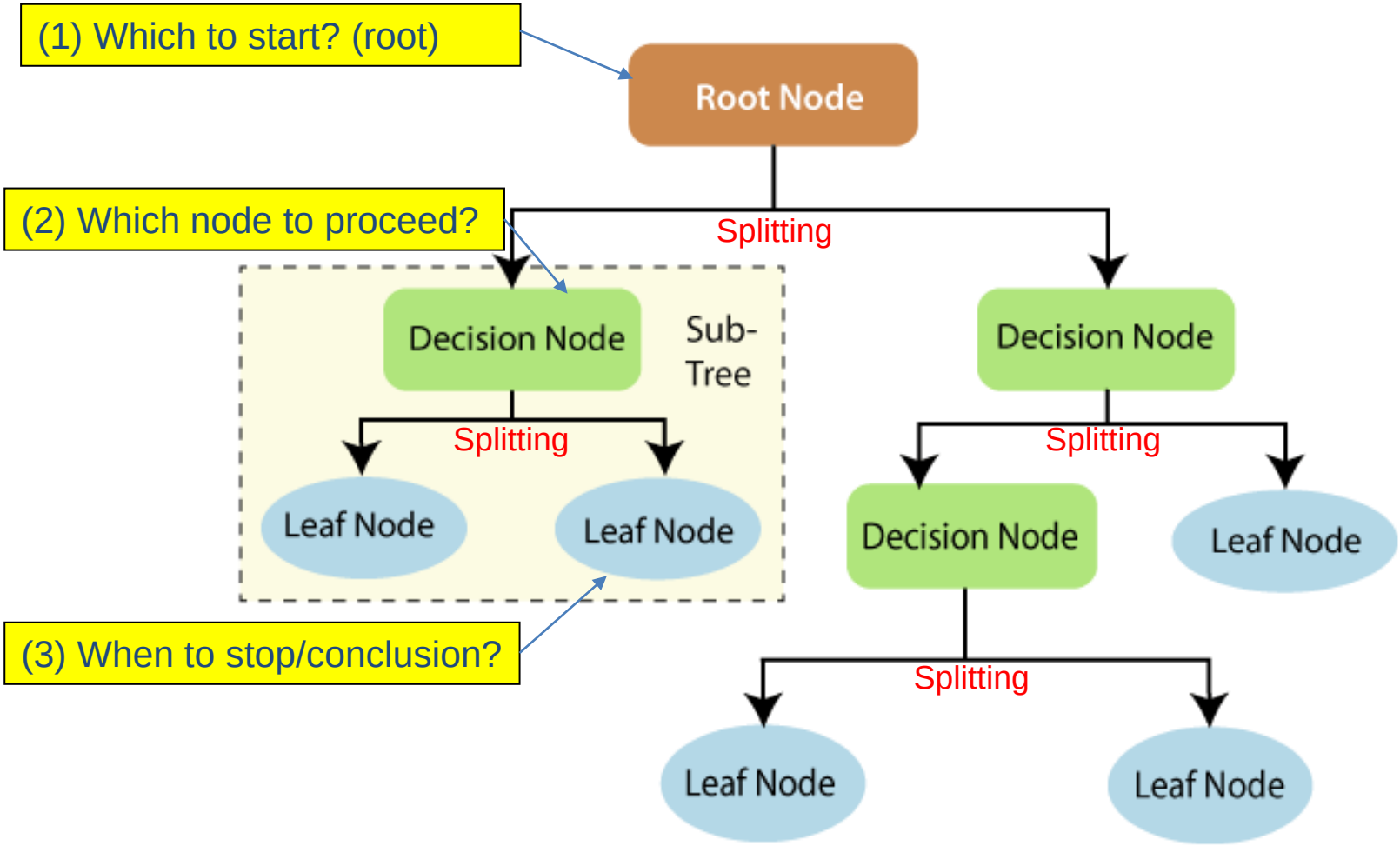
Decision Tree: Terminology

- Root Node: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node.
- Leaf/Terminal Node: Nodes do not split is called Leaf or Terminal node.
- Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.
- **Splitting**: It is a process of dividing a node into two or more sub-nodes.
- Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
- Branch/Sub-Tree: A subsection of the entire tree is called branch or sub-tree.

Splitting Issues



Splitting Issues



https://www.kdnuggets.com/

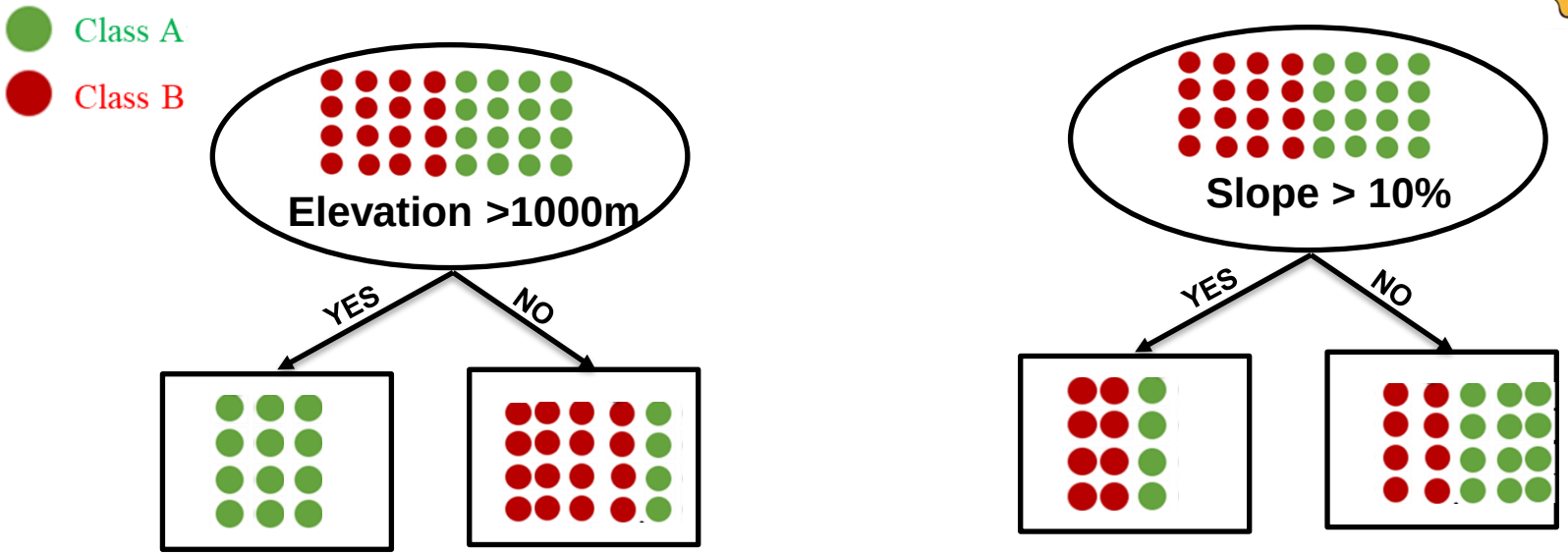
Splitting Issues

Issues

- How to determine the Best Split
- Determine when to stop splitting

How to determine the Best Split

- at each step, we look for the best split (Greedy Strategy).

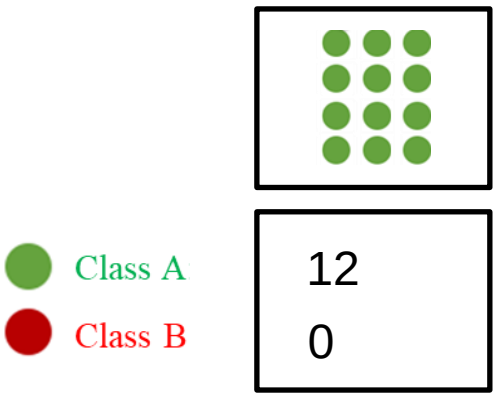


Which test split is the best?

How to determine the Best Split

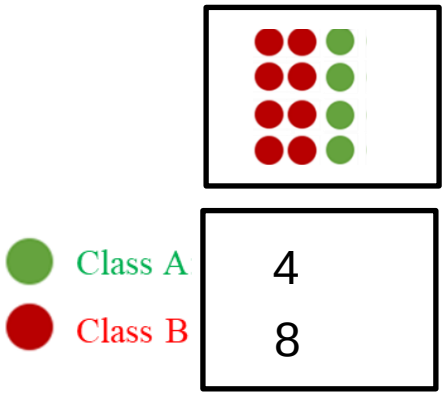
- Nodes with homogeneous class distribution are preferred

- Need a measure of node impurity:



Homogeneous,

Low degree of impurity



Non-homogeneous,

High degree of impurity

Measures of Node Impurity

- Gini Index or Gini ratio
- This measure provides a ranking to the attributes for splitting the data.

- It stores sum of squared probabilities of each class. We can formulate it as illustrated below.

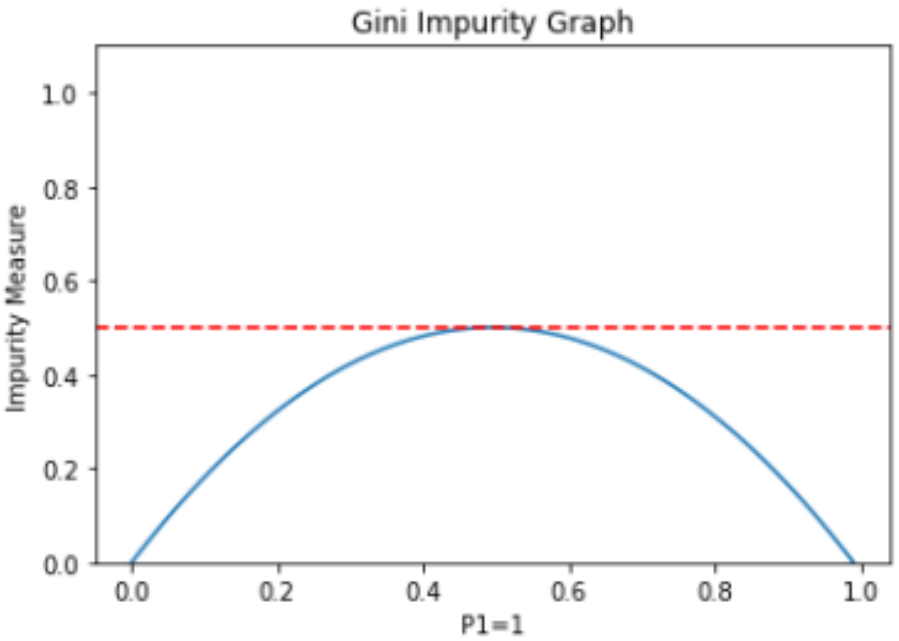
$$Gini = 1 - \sum(P_i)^2$$

i =1 to number of classes

P = probabilities of each class

A Gini Impurity of **0** is the lowest.

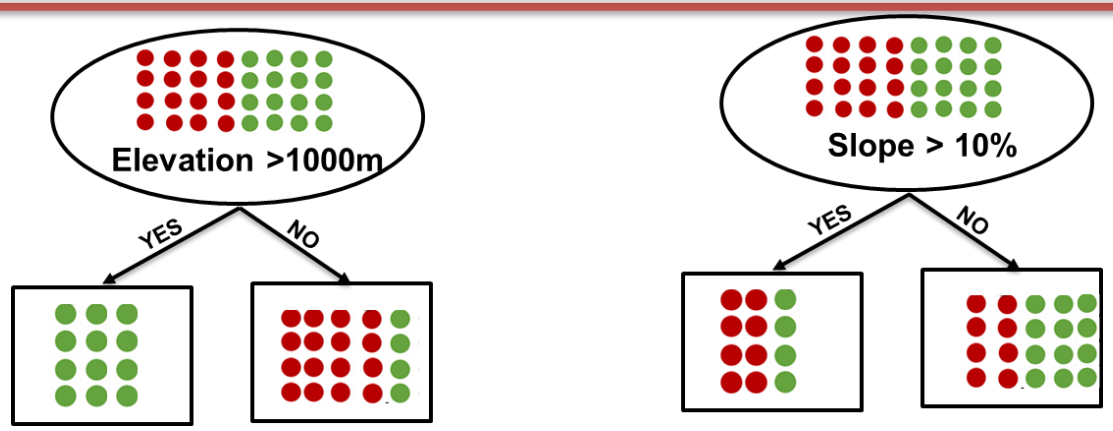
It can only be achieved when everything is the same class



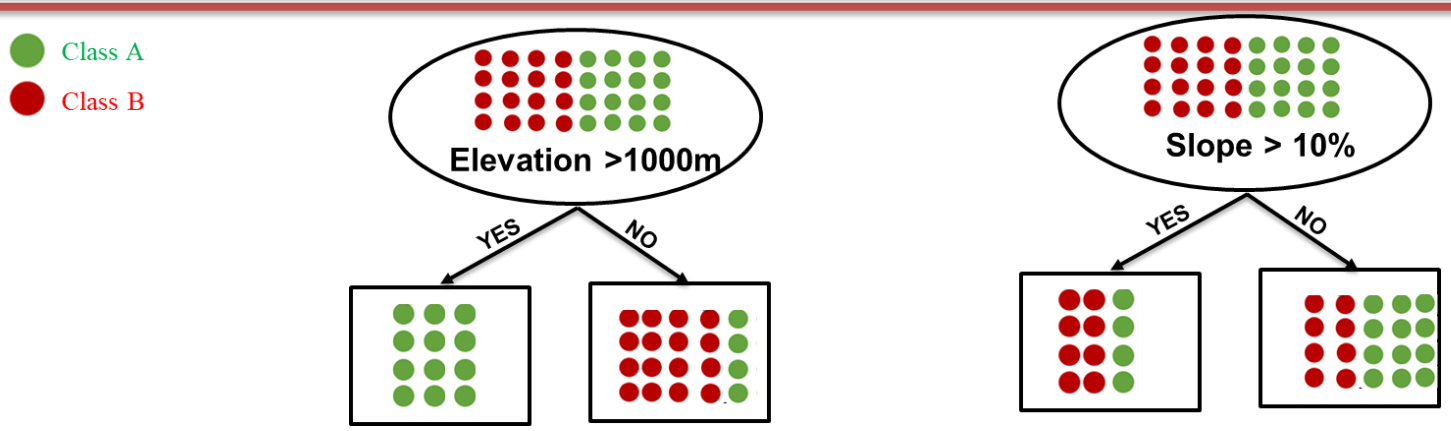
$Gini = 1 - \sum(P_i)^2$ Gini Index

Class A

Class B

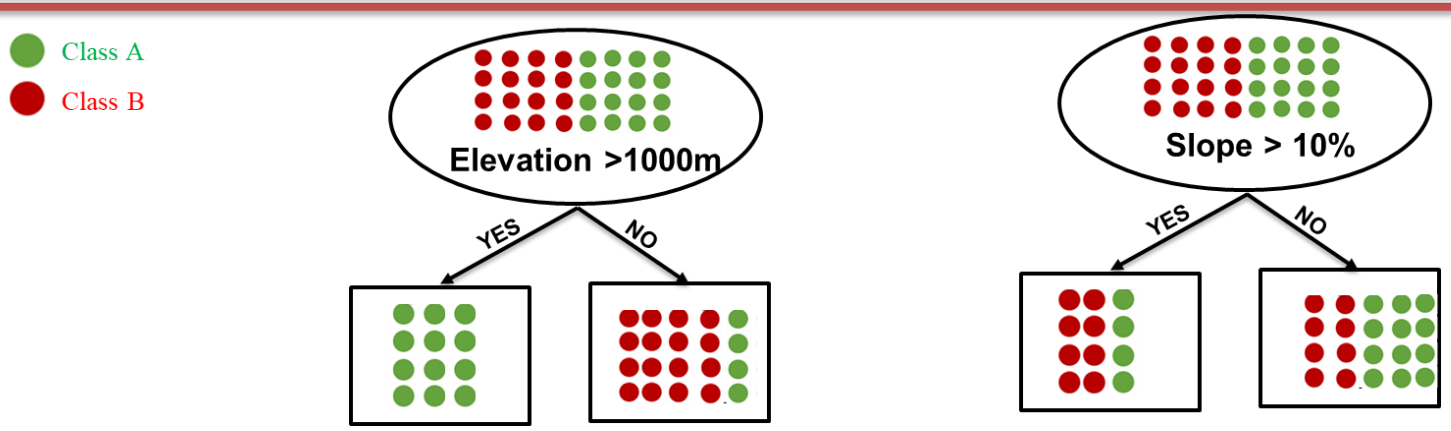


$Gini = 1 - \sum(P_i)^2$ Gini Index



Elevation	A	B	P (A)	P (B)	Gini
>1000 m	12	0	12÷12=1	0÷12=0	$1-[(1)^2+(0)^2]=0$
<1000 m	4	16	4÷20=0.2	16÷20=0.8	$1-[(0.2)^2+(0.8)^2]=0.32$

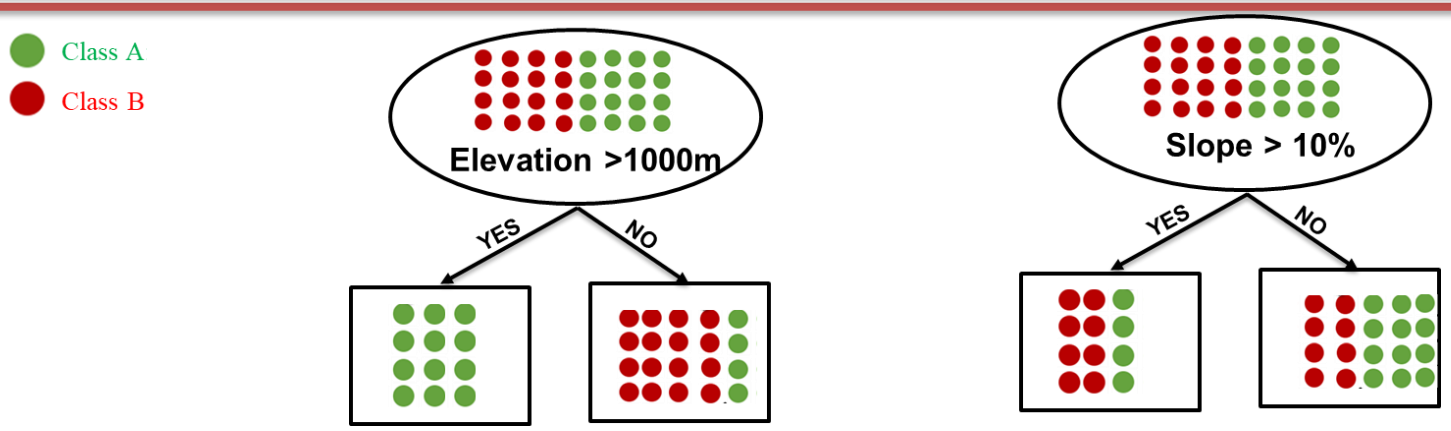
$Gini = 1 - \sum(P_i)^2$ Gini Index



Elevation	A	B	P (A)	P (B)	Gini
>1000 m	12	0	12÷12=1	0÷12=0	$1-[(1)^2+(0)^2]=0$
<1000 m	4	16	4÷20=0.2	16÷20=0.8	$1-[(0.2)^2+(0.8)^2]=0.32$

$$Gini(Elevation) = \left(\frac{12}{32}\right) \times 0 + \left(\frac{20}{32}\right) \times 0.32 = 0.20$$

$Gini = 1 - \sum(P_i)^2$ Gini Index



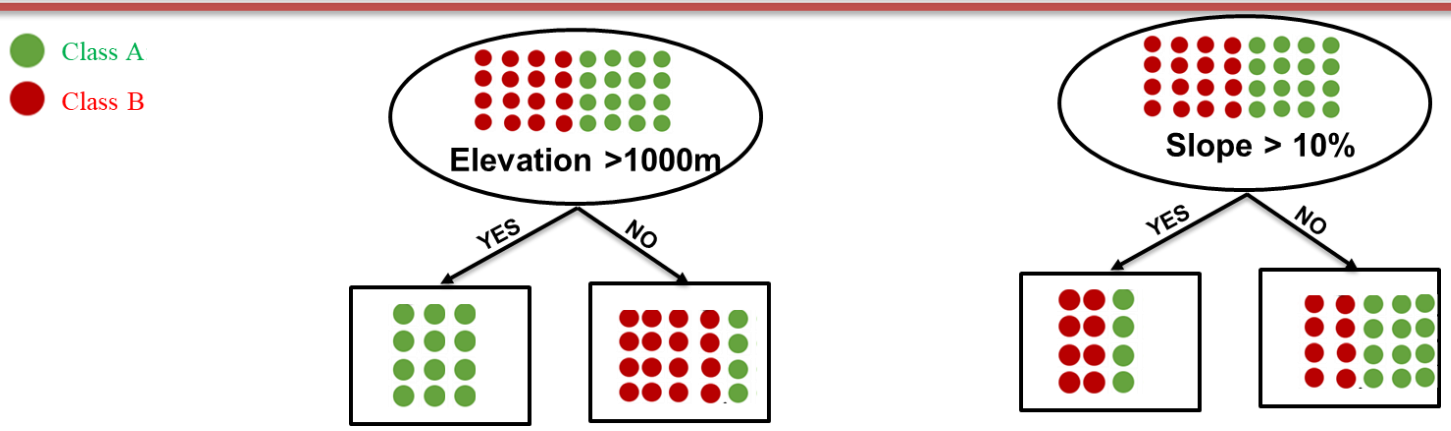
Elevation	A	B	P (A)	P (B)	Gini
>1000 m	12	0	12÷12=1	0÷12=0	$1-[(1)^2+(0)^2]=0$
<1000 m	4	16	4÷20=0.2	16÷20=0.8	$1-[(0.2)^2+(0.8)^2]=0.32$

$$Gini(Elevation) = \left(\frac{12}{32}\right) \times 0 + \left(\frac{20}{32}\right) \times 0.32 = 0.20$$

Slope	A	B	P (A)	P (B)	Gini
>10 %	4	8	4÷12=1	8÷12=0	$1-[(0.22)^2+(0.66)^2]=0.44$
<10 %	12	8	12÷20=0.2	8÷20=0.8	$1-[(0.6)^2+(0.4)^2]=0.48$

$$Gini(Slope) = \left(\frac{12}{32}\right) \times 0.44 + \left(\frac{20}{32}\right) \times 0.48 = 0.46$$

$Gini = 1 - \sum(P_i)^2$ Gini Index



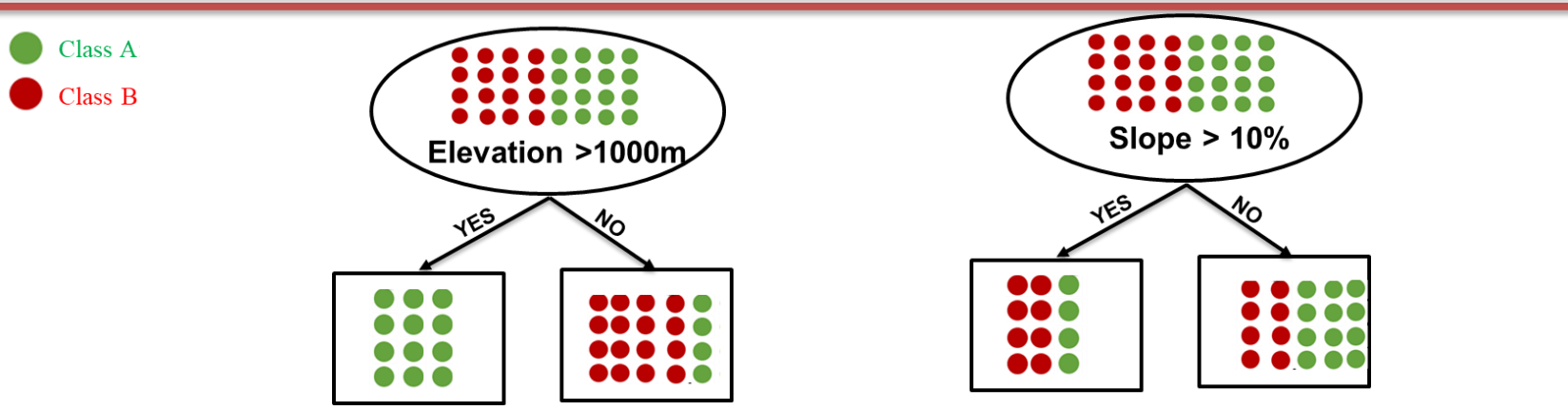
Elevation	A	B	P (A)	P (B)	Gini
>1000 m	12	0	12÷12=1	0÷12=0	1-[(1)²+(0)²]=0
<1000 m	4	16	4÷20=0.2	16÷20=0.8	1-[(0.2)²+(0.8)²]=0.32

$$Gini(Elevation) = \left(\frac{12}{32}\right) \times 0 + \left(\frac{20}{32}\right) \times 0.32 = 0.20$$

Slope	A	B	P (A)	P (B)	Gini
>10 %	4	8	4÷12=1	8÷12=0	1-[(0.22)²+(0.66)²]=0.44
<10 %	12	8	12÷20=0.2	8÷20=0.8	1-[(0.6)²+(0.4)²]=0.48

$$Gini(Slope) = \left(\frac{12}{32}\right) \times 0.44 + \left(\frac{20}{32}\right) \times 0.48 = 0.46$$

$Gini = 1 - \sum(P_i)^2$ Gini Index



Elevation	A	B	P (A)	P (B)	Gini
>1000 m	12	0	12÷12=1	0÷12=0	1-[(1)²+(0)²]=0
<1000 m	4	16	4÷20=0.2	16÷20=0.8	1-[(0.2)²+(0.8)²]=0.32

$$Gini(Elevation) = \left(\frac{12}{32}\right) \times 0 + \left(\frac{20}{32}\right) \times 0.32 = 0.20$$

Slope	A	B	P (A)	P (B)	Gini
>10 %	4	8	4÷12=1	8÷12=0	1-[(0.22)²+(0.66)²]=0.44
<10 %	12	8	12÷20=0.2	8÷20=0.8	1-[(0.6)²+(0.4)²]=0.48

$$Gini(Slope) = \left(\frac{12}{32}\right) \times 0.44 + \left(\frac{20}{32}\right) \times 0.48 = 0.46$$

Splitting Issues

Issues

- How to determine the Best Split
- Determine when to stop splitting

Determine when to stop splitting

- Stopping Rule
  - Typical stopping conditions for a node:
    - Stop if all instances belong to the same class
    - Stop if all the attribute values are the same
  - More restrictive conditions:
    - Stop if number of instances is less than some user-specified threshold
    - Stop if class distribution of instances are independent of the available features
    - Stop if expanding the current node does not improve impurity measures (e.g., Gini index).

Stop the algorithm before it becomes a fully-grown tree (Pre-Pruning)

Overfitting In Decision Trees

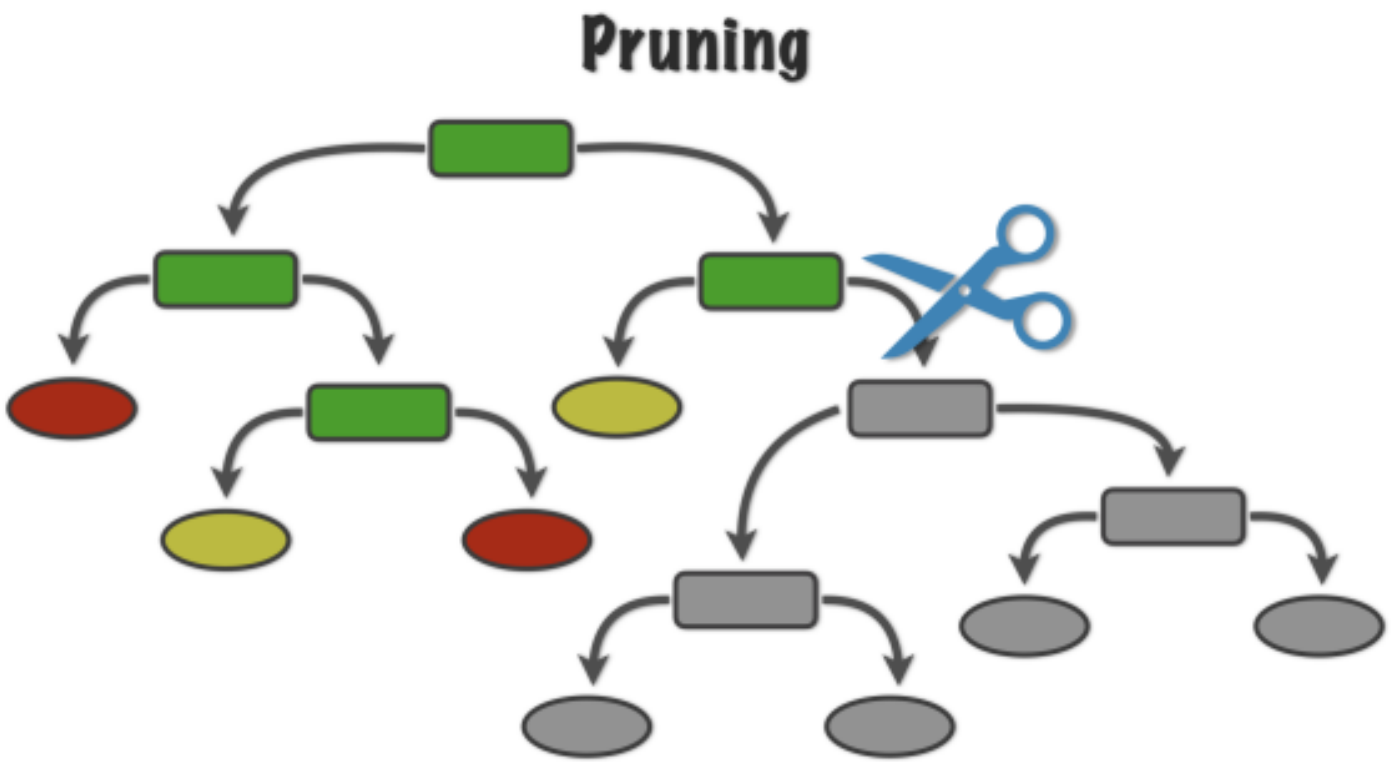
Overfitting happens when a decision tree tries to be as perfect as possible by increasing the depth of tests and thereby reduces the error.

- Post-pruning
  - Grow decision tree to its entirety
  - Trim the nodes of the decision tree in a bottom-up fashion
  - If generalization error improves after trimming, replace sub-tree by a leaf node.
  - Class label of leaf node is determined from majority class of instances in the sub-tree

Pruning

Comparing pre-pruning and post-pruning,

- Pre-pruning is faster but
- Post-pruning generally leads to more accurate trees.



https://alanjeffares.wordpress.com/tutorials/decision-tree/

Regression Trees

- Regression tree:
  - is constructed in almost the same manner as a classification tree,
  - except that the impurity measure that is appropriate for classification is replaced by a measure appropriate for regression.
  - the goodness of a split is measured by the Mean Square Error from the estimated value.

$$MSE = sum(y - prediction)^2$$

Where y is the output for the training sample and prediction is the predicted output for the node.

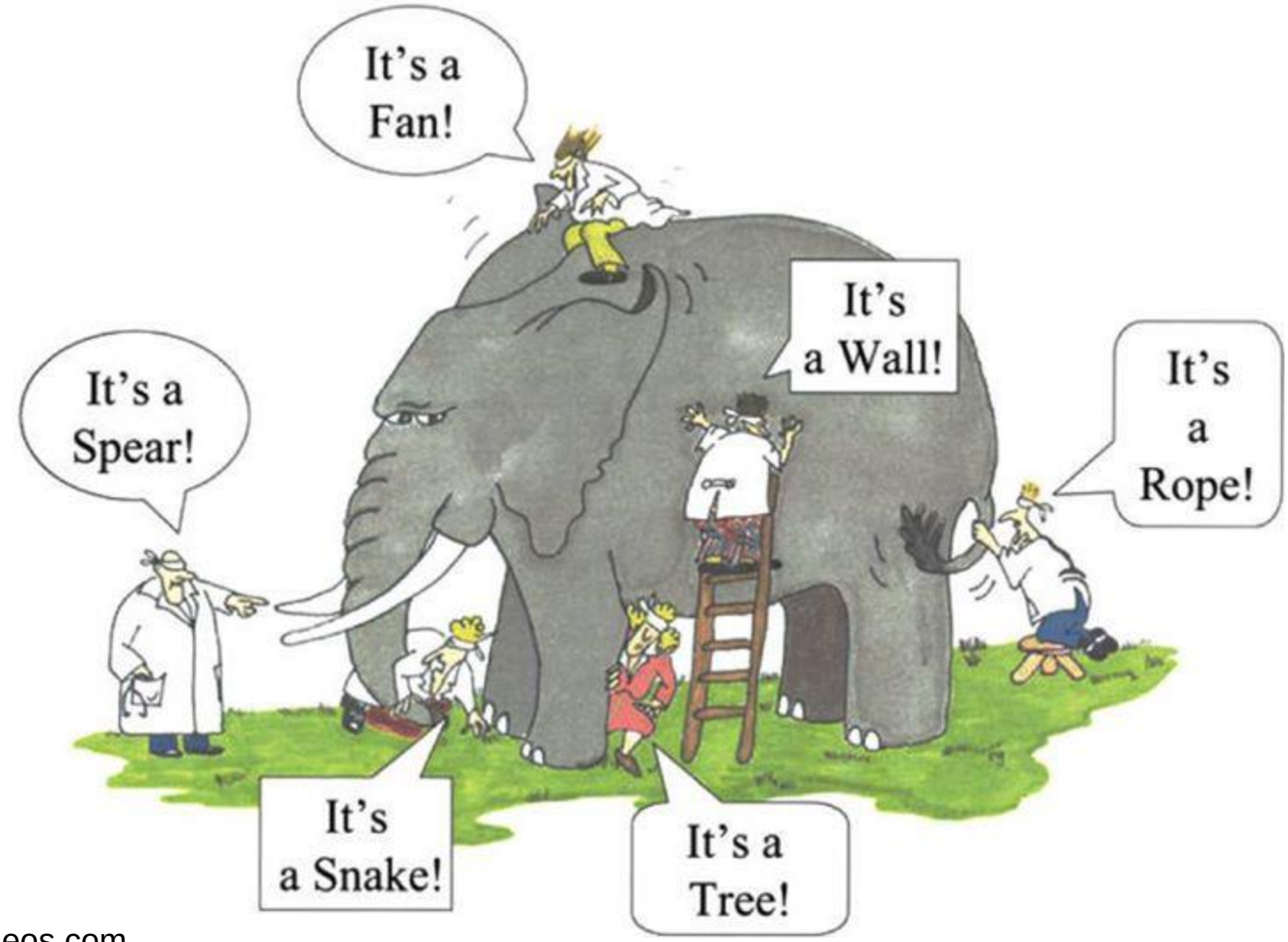
Advantages and Disadvantages

- Advantages:
  - Extremely fast at classifying unknown records.
  - Simple to understand and to interpret for small-sized trees
  - Trees can be visualized.
  - Accuracy comparable to other classification techniques for many simple data sets.
  - Excludes unimportant features.
  - Requires little data preparation. Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed.
  - Able to handle both numerical and categorical data. Other techniques are usually specialised in analysing datasets that have only one type of variable.

Advantages and Disadvantages

- Disadvantages:
  - Easy to overfit.
  - Large trees can be difficult to interpret
  - Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an **ensemble**.

Ensemble Methods



Source: Patheos.com

What Are Ensemble Methods?

- Ensemble learning is a machine learning paradigm where multiple models (e.g., multiple decision trees) (often called “weak learners”) are trained to solve the same problem (e.g., prediction of soil types) and combined to get better results.

- The main hypothesis is that when weak models are correctly combined we can obtain more accurate models.

Question

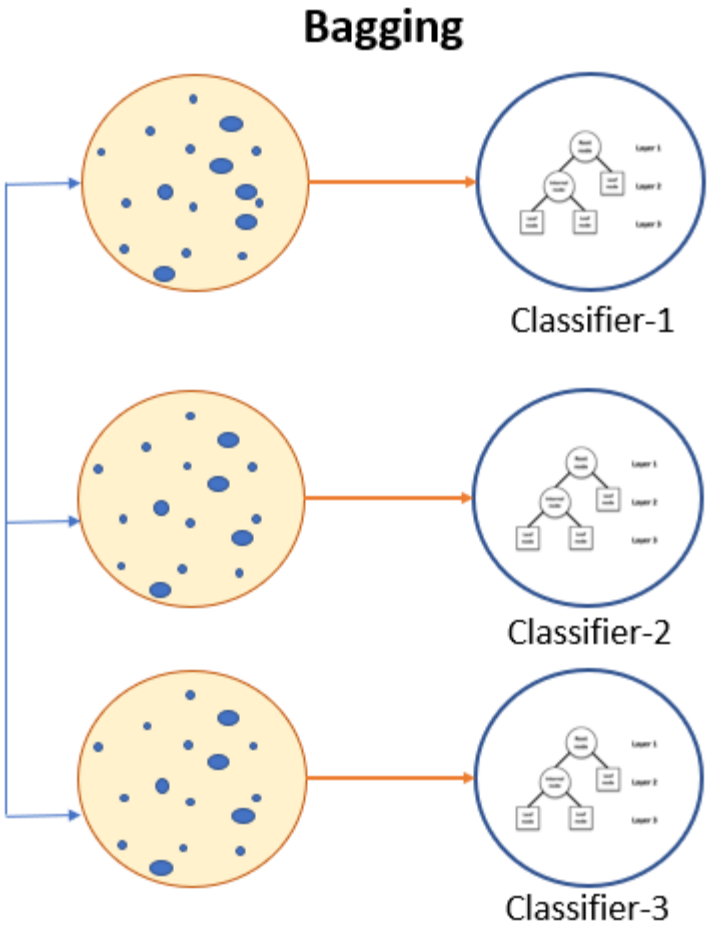
How to combine these models

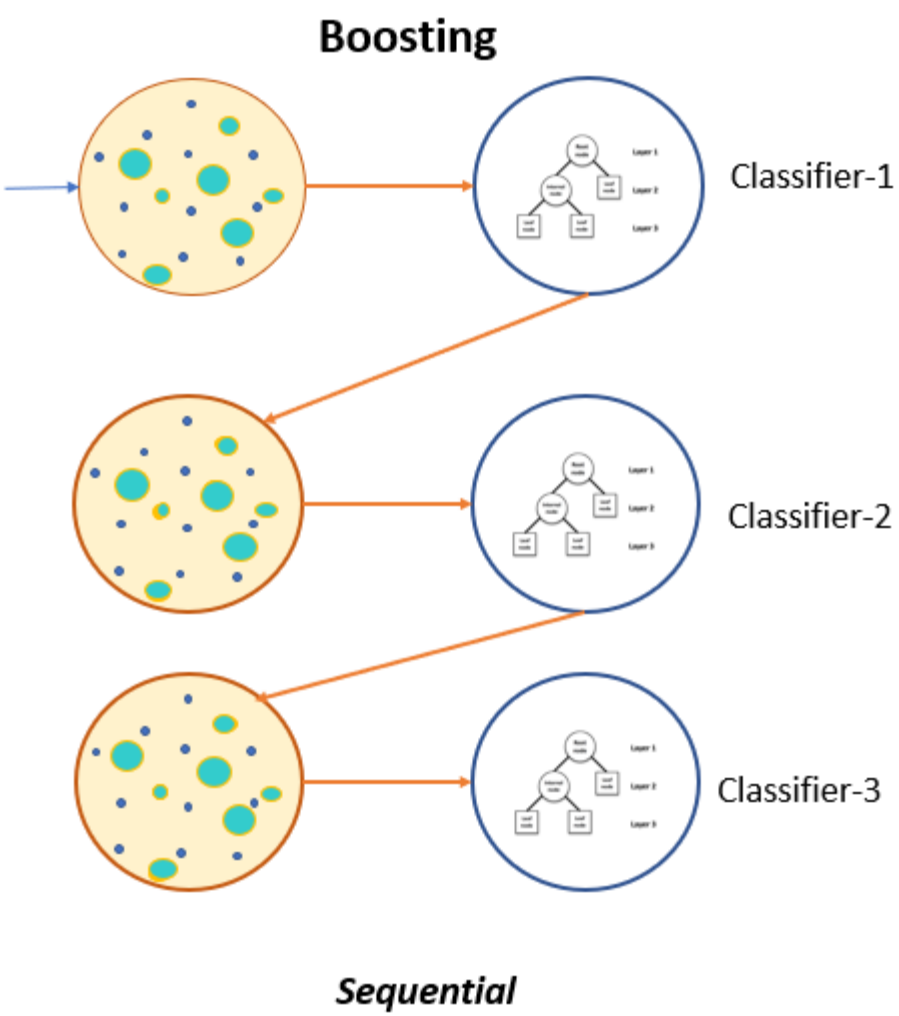
(models do not all just learn the same)

Combining Weak Learners

- Bagging, Boosting, Stacking Models
  - bagging, that often considers homogeneous weak learners, learns them independently from each other in parallel and combines them following some kind of deterministic averaging process
  - boosting, that often considers homogeneous weak learners, learns them sequentially in a very adaptative way (a base model depends on the previous ones) and combines them following a deterministic strategy
  - stacking, that often considers heterogeneous weak learners, learns them in parallel and combines them by training a meta-model to output a prediction based on the different weak models predictions

Bagging, Boosting, Stacking Models

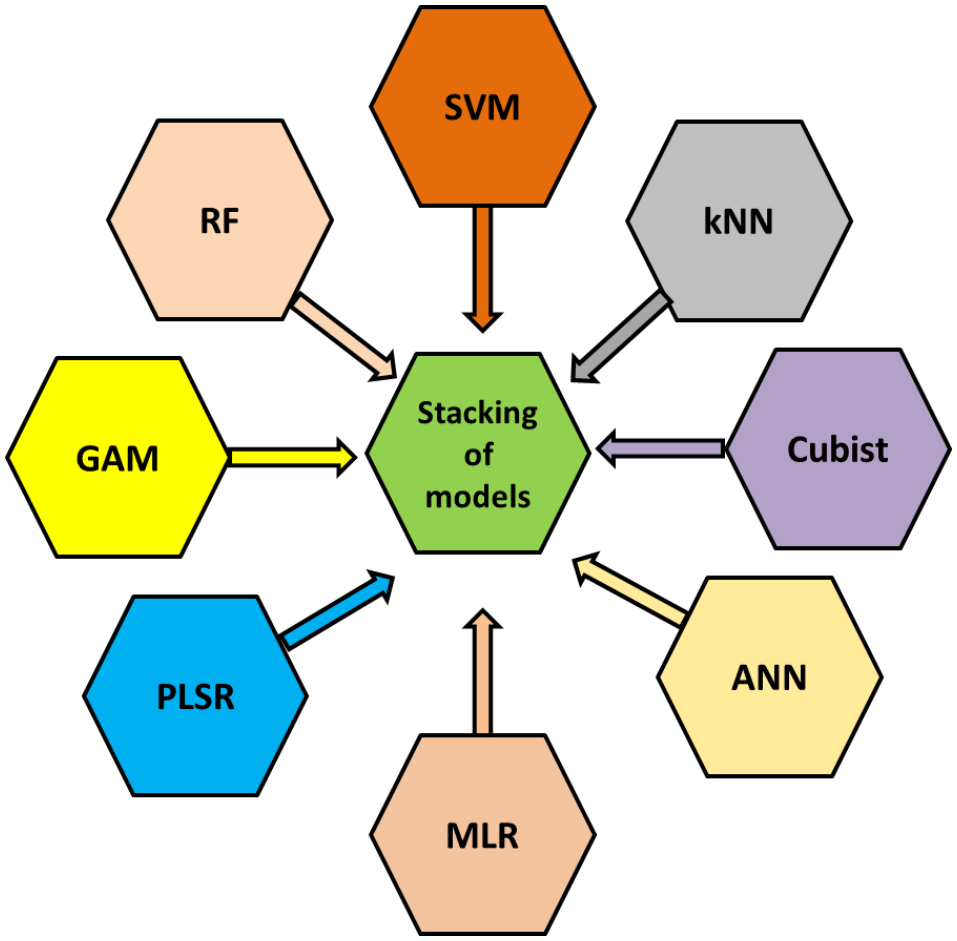




https://www.pluralsight.com/guides/ensemble-methods:-bagging-versus-boosting

Bagging, Boosting, Stacking Models

- Stacking



Bagging

- Bagging or Bootstrap aggregating (Breiman 1994):

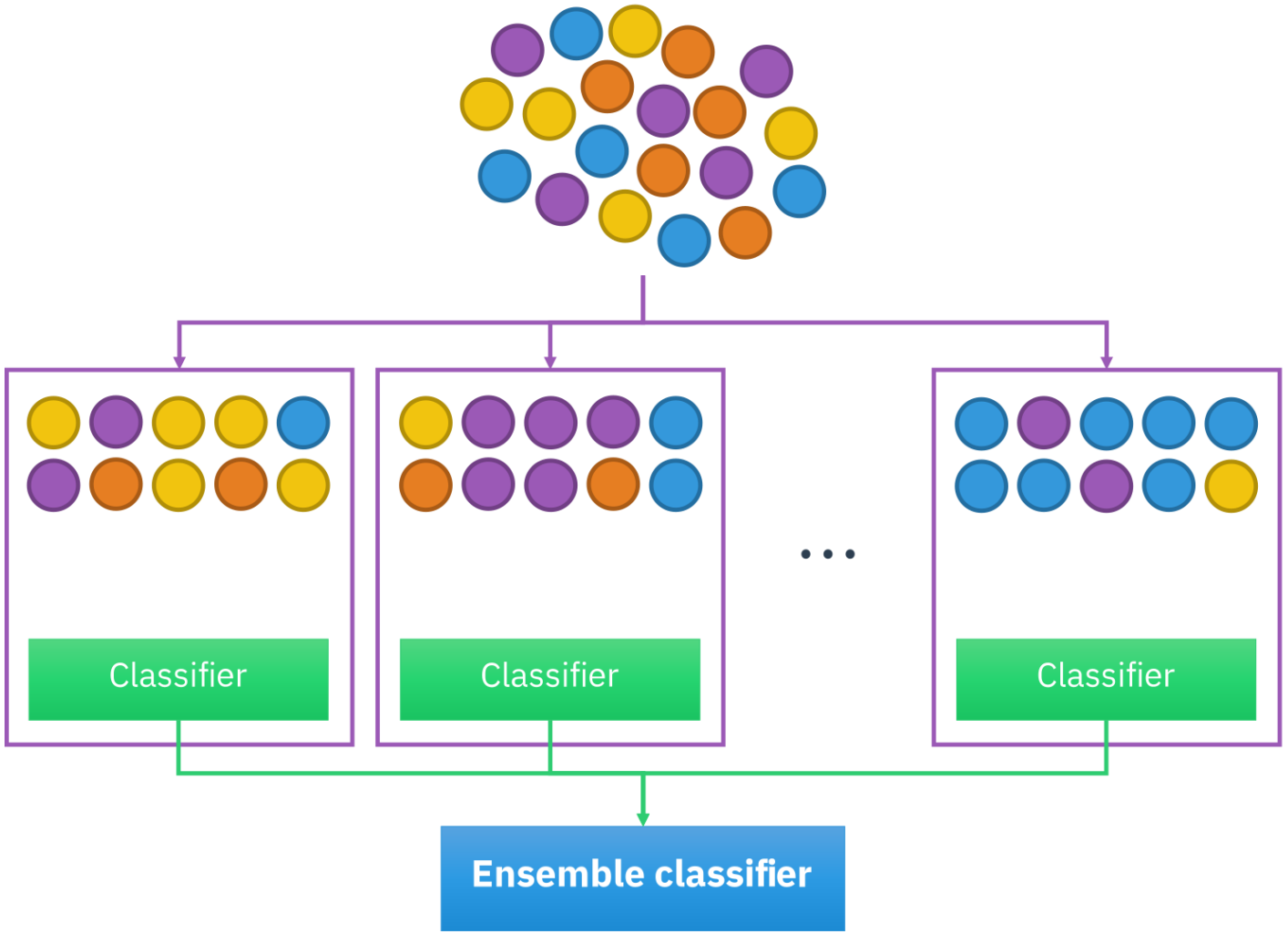
- First, we **create multiple bootstrap samples** so that each new bootstrap sample will act as another (almost) independent dataset drawn from true distribution.

- Then, we can fit a weak learner (**decision tree**) for each of these samples

- Finally **aggregate them** (simple average for regression or simple majority vote for classification)

Bootstrap: a statistical technique, for generating samples of **size B** (called bootstrap samples) from an initial dataset of **size N** by randomly drawing with replacement B observations.

Bagging



Original Data

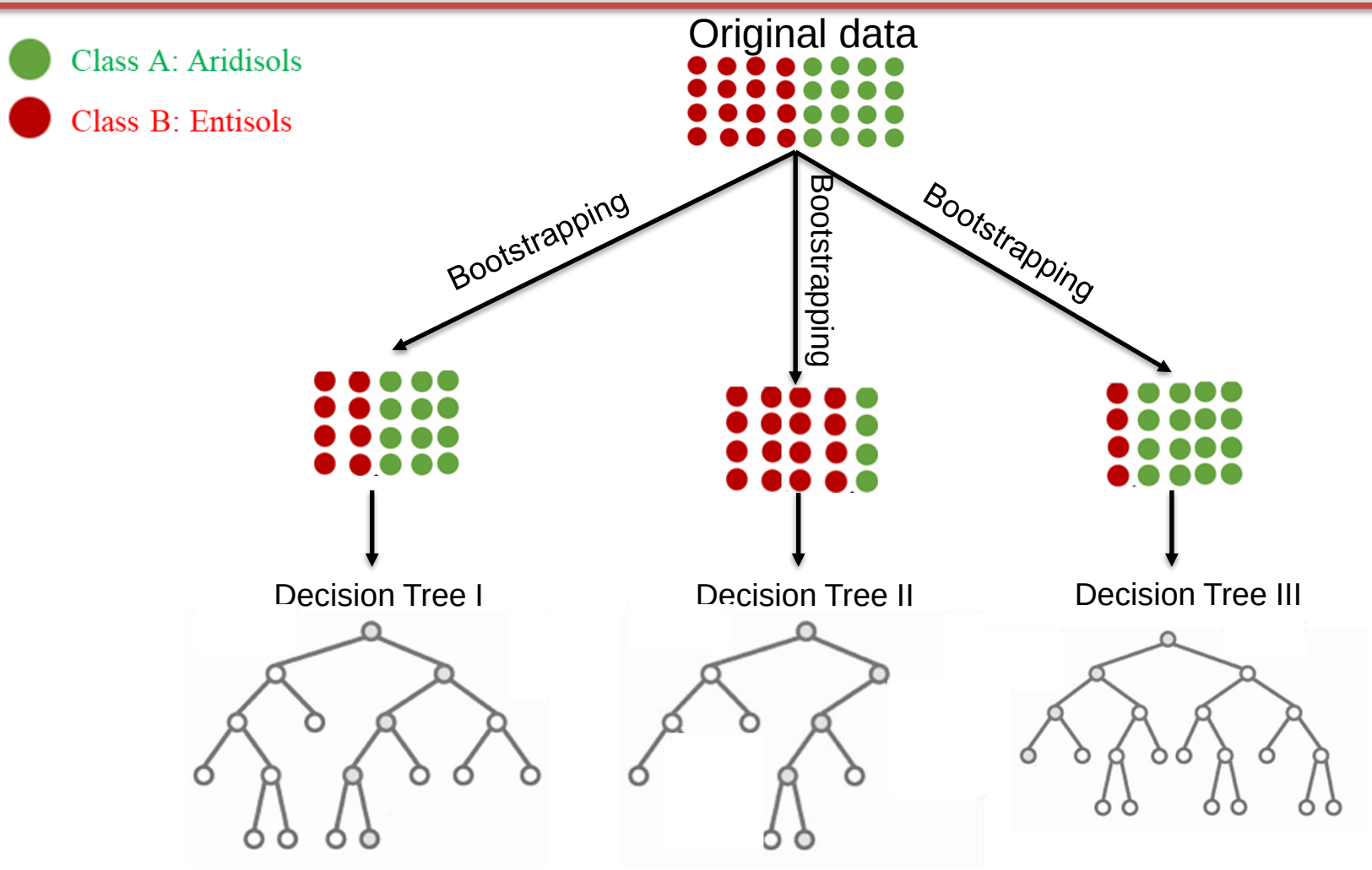
Bootstrapping

Aggregating

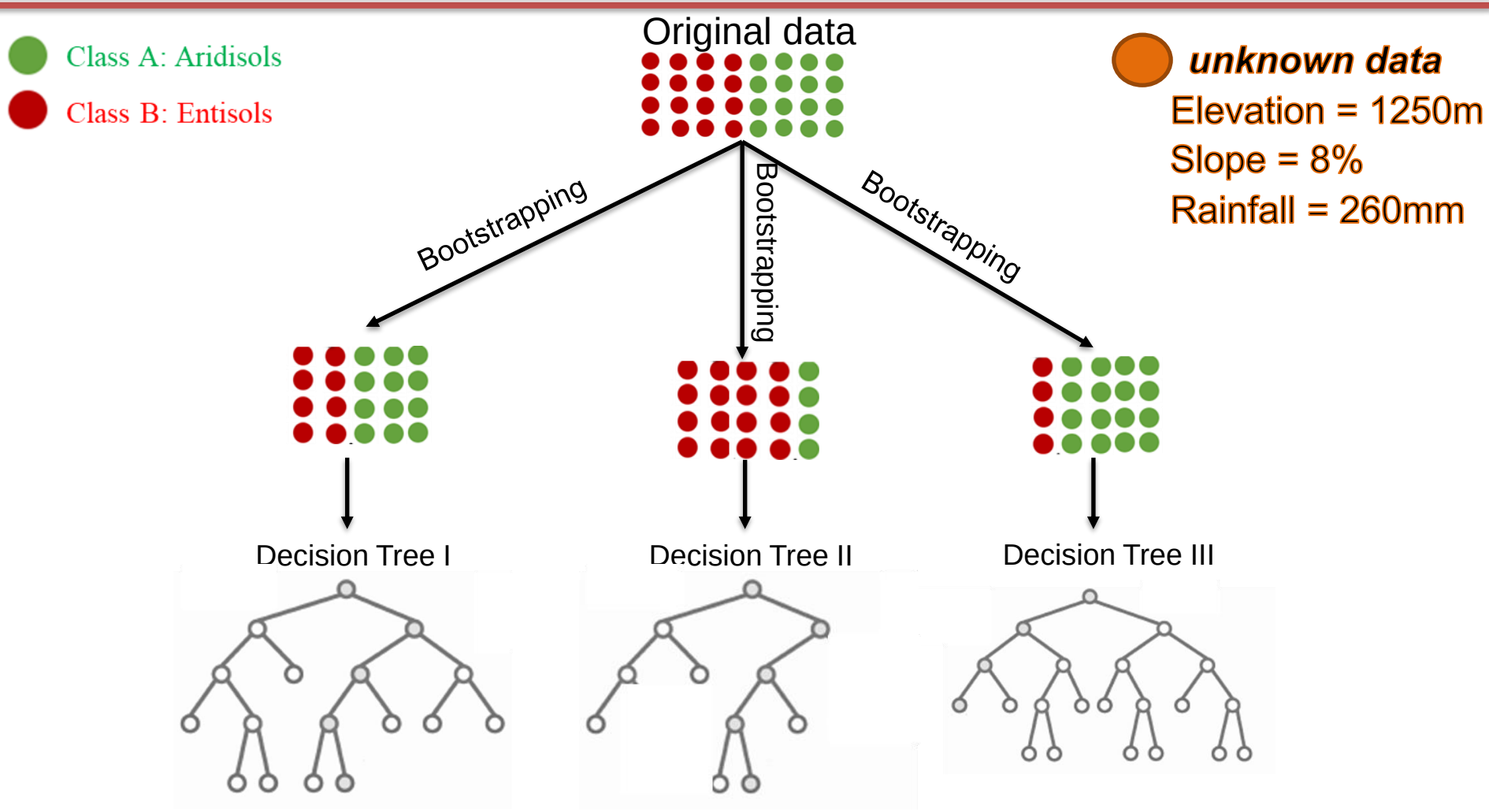
Bagging

https://www.pluralsight.com/guides/ensemble-methods:-bagging-versus-boosting

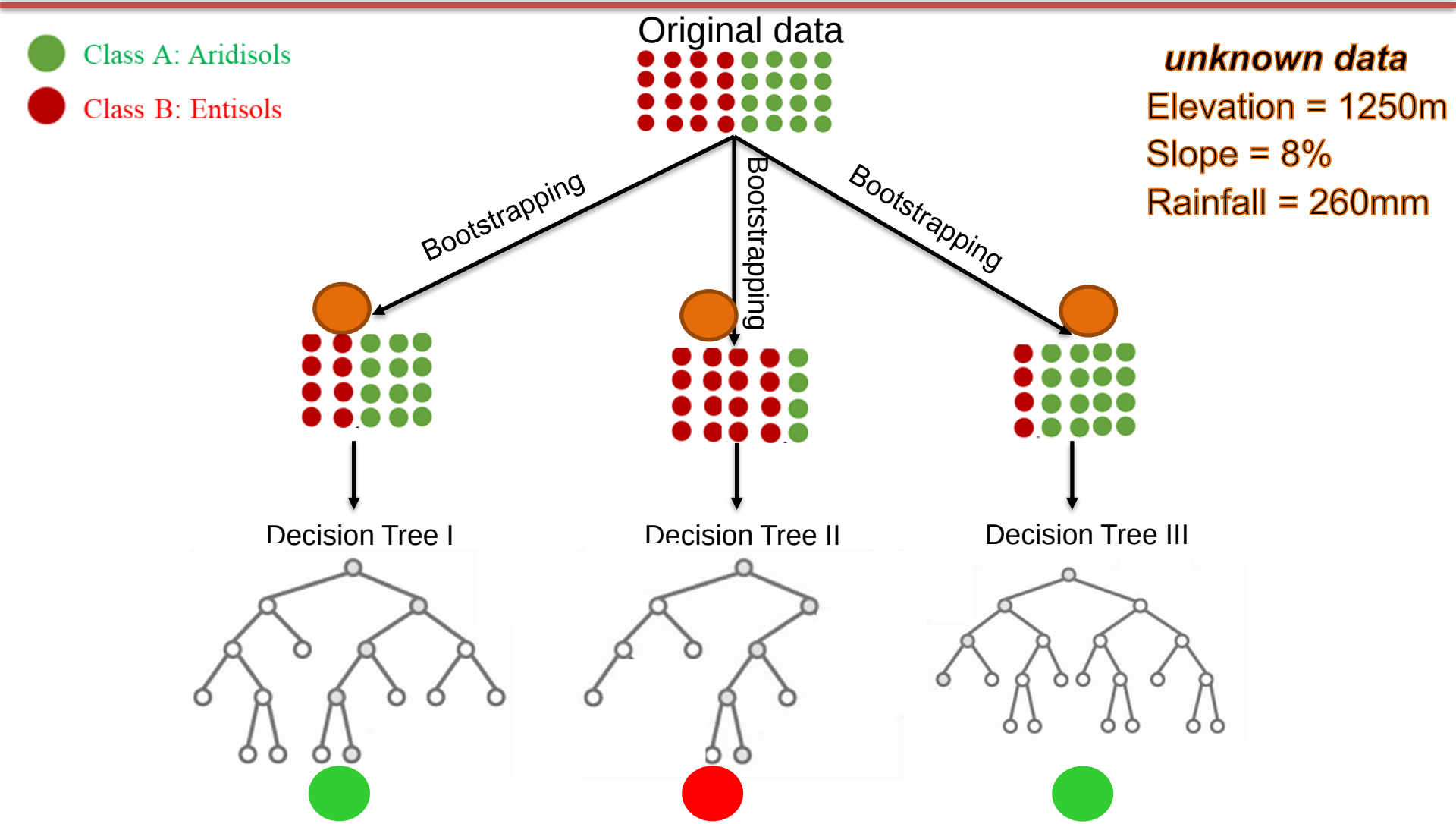
Bagging: example I



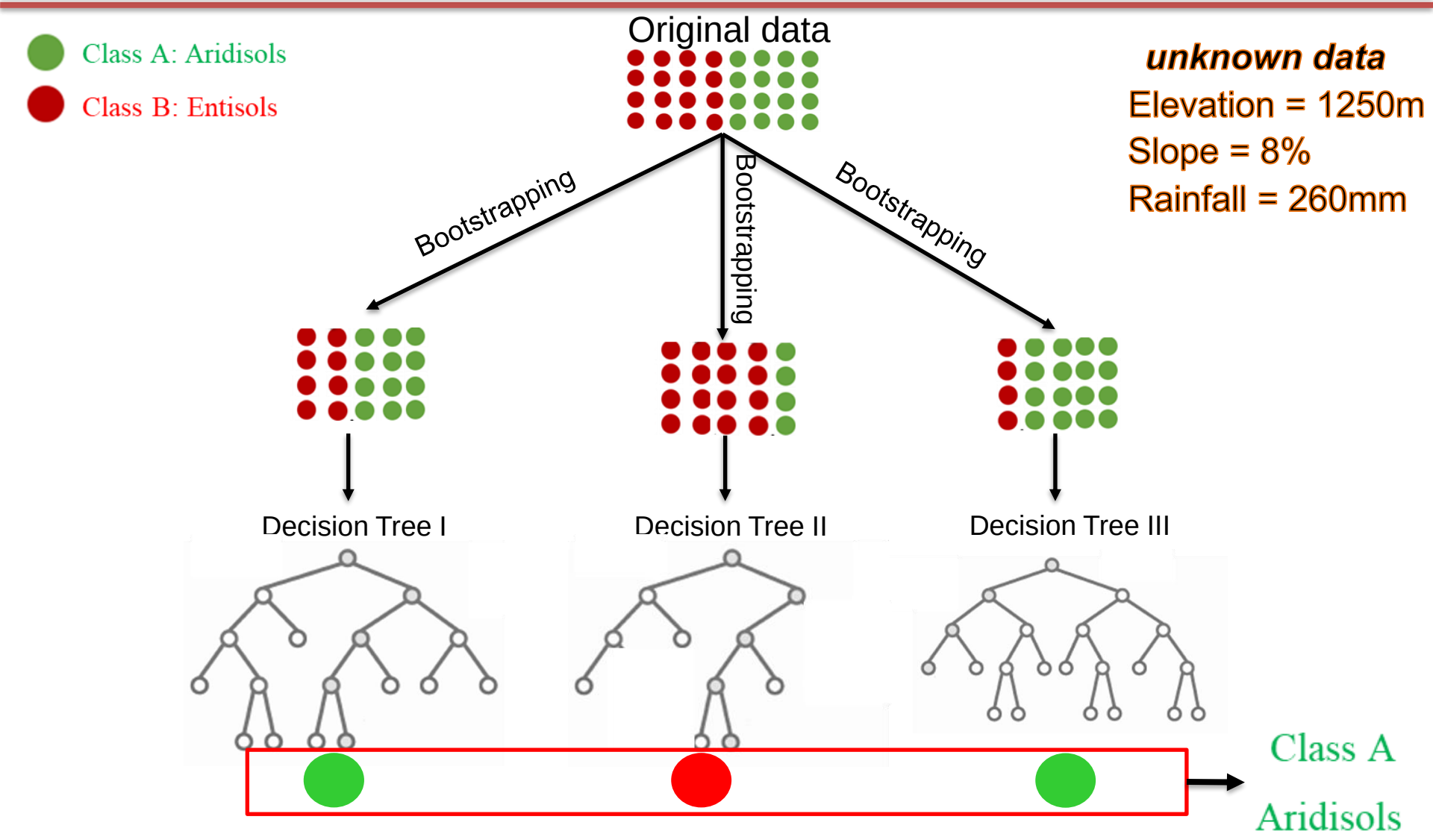
Bagging: example I



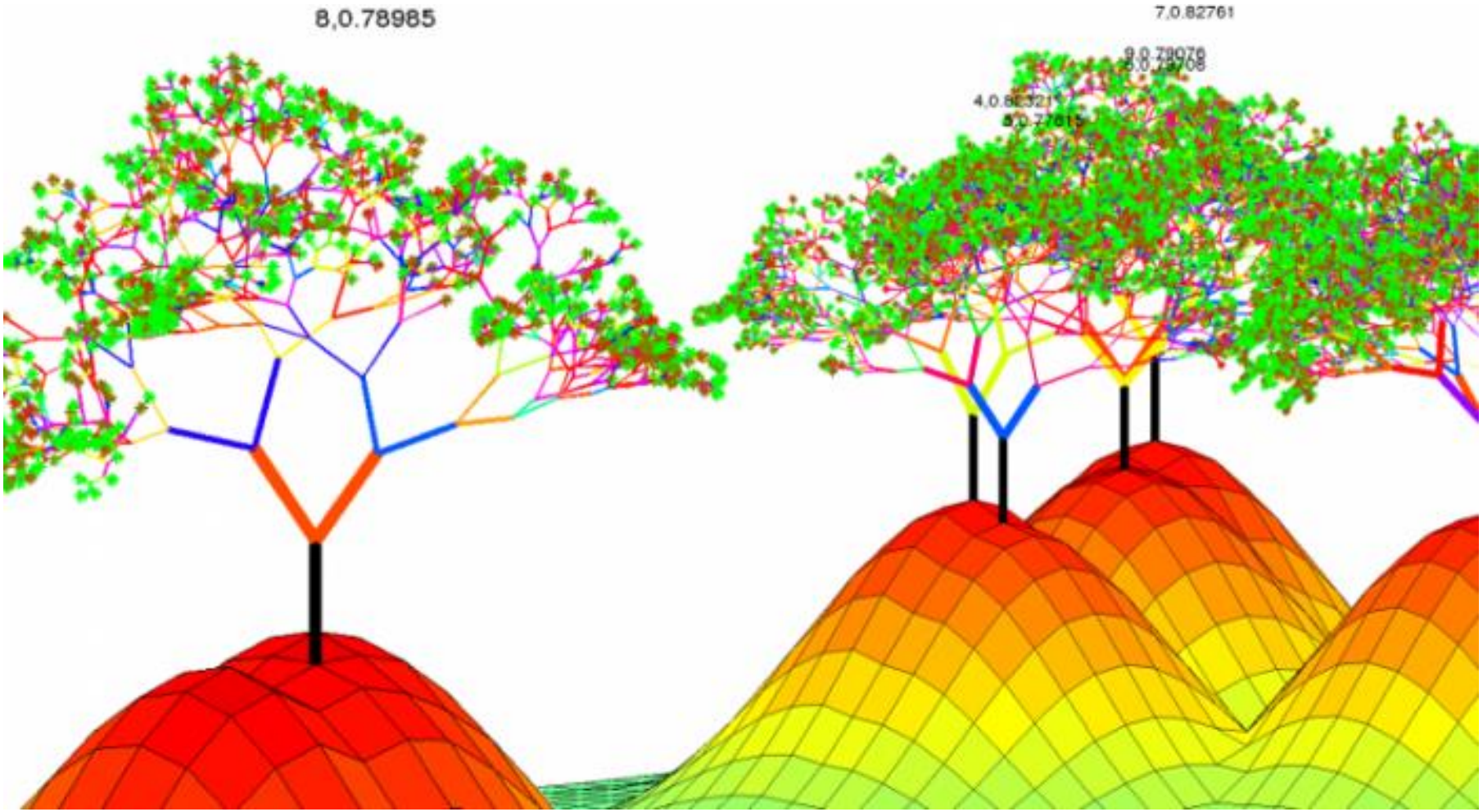
Bagging: example I



Bagging: example I



Random Forest



https://towardsdatascience.com/random-forest-learning-essential-understanding-1ca856a963cb

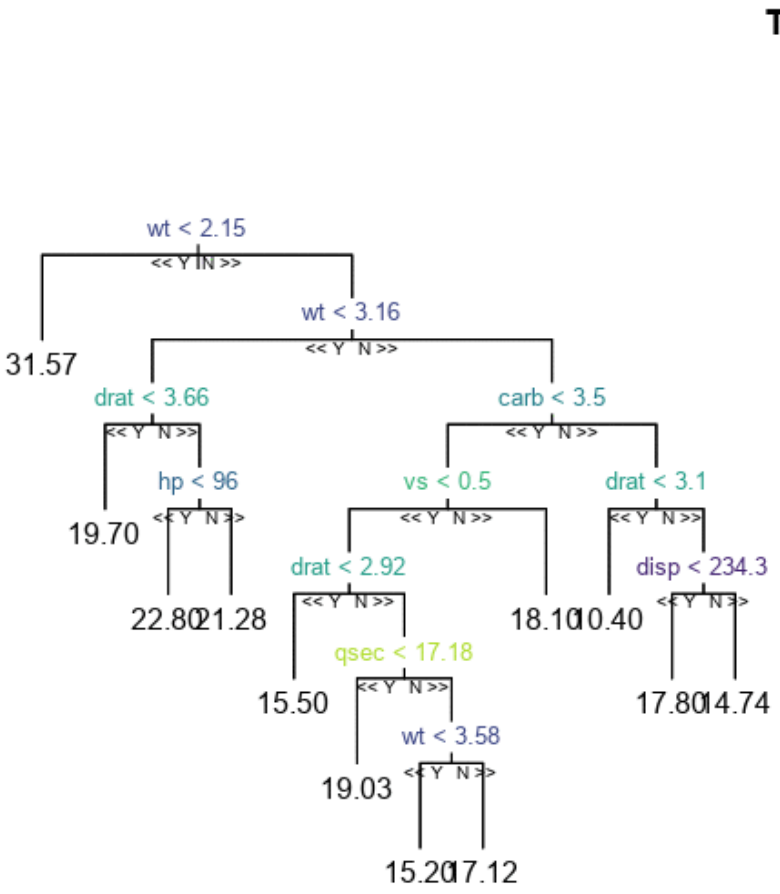
Random Forest

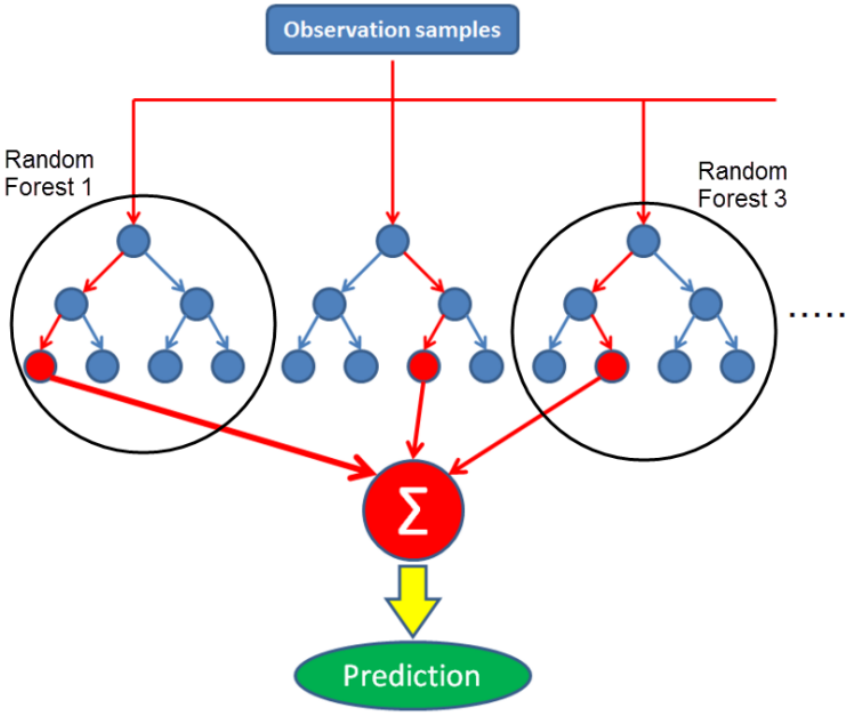
- Random Forest:
  - The random forest approach is a **bagging method** for classification and regression that operate by constructing multiple decision trees at training phase and outputting the class that is the mode of the classes or mean/average prediction of the individual trees.
  - Random forests also use another trick to make the multiple decision trees a bit less correlated with each others:
  - when growing each tree, instead of **only** sampling over the observations in the dataset to generate a bootstrap sample, we also sample over **covariates** and keep only a random subset of them to build the tree.

How Does It Work?

- Random Forest:
  - Bagging
    - Draw a bootstrap sample:
      - Random selection of 2/3 of the training data; repeat *n* times.
      - Grow an unpruned tree to each bootstrap sample
    - Random predictor selection: for each split in each tree a random subset is selected from the predictor variables. The best split is chosen from among the selected predictors.
    - Predict new data by aggregating the predictions of the n trees.
      - Average for continuous variables
      - Majority vote for categorical variables

Random Forest





Out Of Bag Accuracy Assessment

- Random Forest:
  - Random Forest comes with an internal accuracy assessment (based on cross validation).
  - Bootstrap sampling: the algorithm sets aside ~36% of the training data for each tree grown: out of bag data (OOB).
  - OOB data can be used to asses prediction accuracy:
    - Predict the data not in the bootstrap sample for each tree
    - Aggregate the OOB predictions: mean (continuous data), majority vote (categorical data).

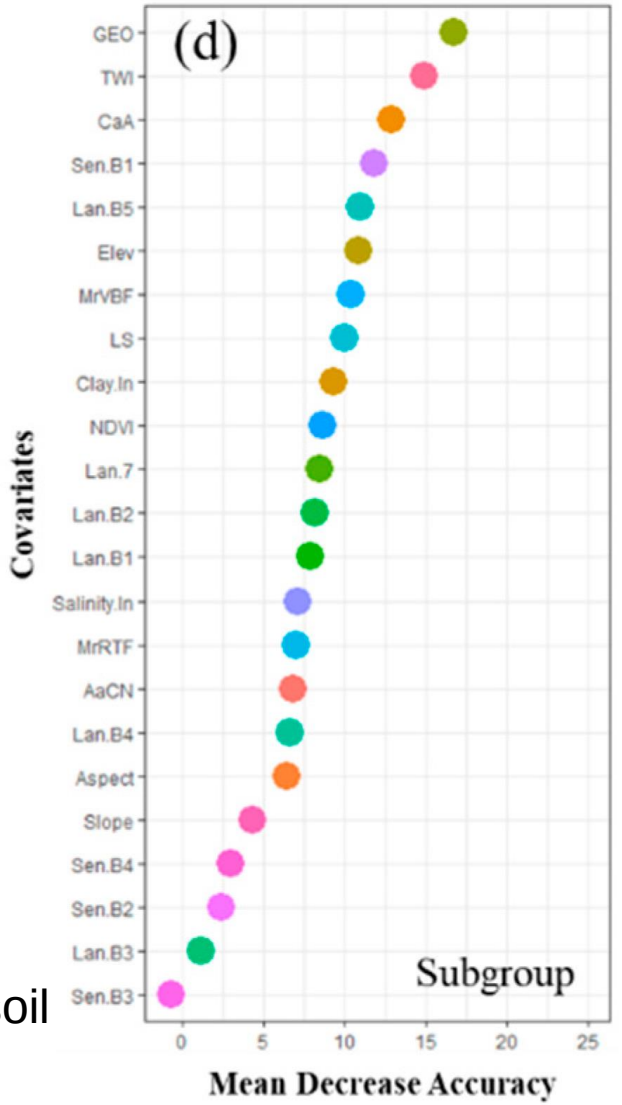
Variable Importance

- Random Forest:
  - Ensembles of trees are not easy to interpret.
  - Ensemble can reflect the potentially (complex) effect of a variable on the response
  - Variable importance plot: shows how much prediction error increases when the values of one predictor are permuted (break association with response) while all others are left unchanged.
  - Permuted variable is used together with other variables to predict the response, prediction accuracy will decrease.

Variable Importance Plot

Table 2. Environmental variables used as soil formation factors in the study area.

Definition	Code
Topographic attributes	
Slope	SLOPE
Aspect	ASPECT
LS factor	LS
Elevation	Elev
Catchments area	CaA
Catchment network base level	CNBL
Topographic Wetness Index	TWI
Altitude above channel network	AaCN
Multi-resolution Valley Bottom Flatness Index	MrVBF
Multi-resolution of ridge top flatness index	MrRTF
Remote sensing attributes	
Clay Index: (shortwave IR-1/shortwave IR-2)	Clay.In
Blue, green, red, near infrared, shortwave IR-1, shortwave IR-2,	Lan.B1-B7
Normalized Difference Vegetation Index: (Shortwave IR-1—Near infrared)/(Shortwave IR-1+ Near infrared)	NDVI
Salinity Index: (Red—Near infrared)/(Red + Near infrared)	Salinity.In
Sentinel Bands 1, 2, 3, and 4	Sen.B1-B4
Geomorphology map	
Hierarchical four level classification (geomorphic surfaces)	GEO



The importance of variables used for predicting soil classes in Subgroup (d).

Advantages and Disadvantages

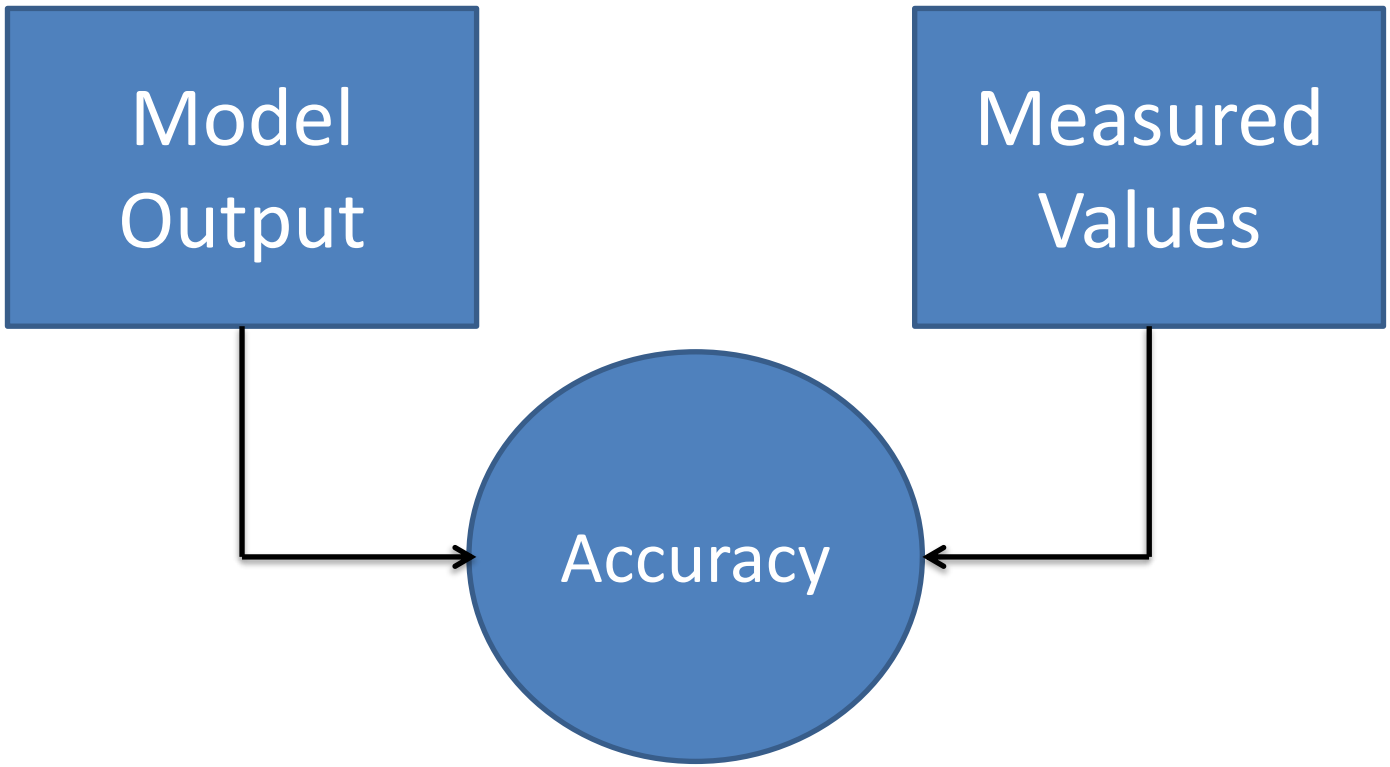
- Advantages:
  - Quite fast
  - Models interactions in the data / non linear relationships
  - Very good predictive power expected
  - Yields covariate importance, makes covariate selection possible

- Disadvantages:
  - Difficult to interpret

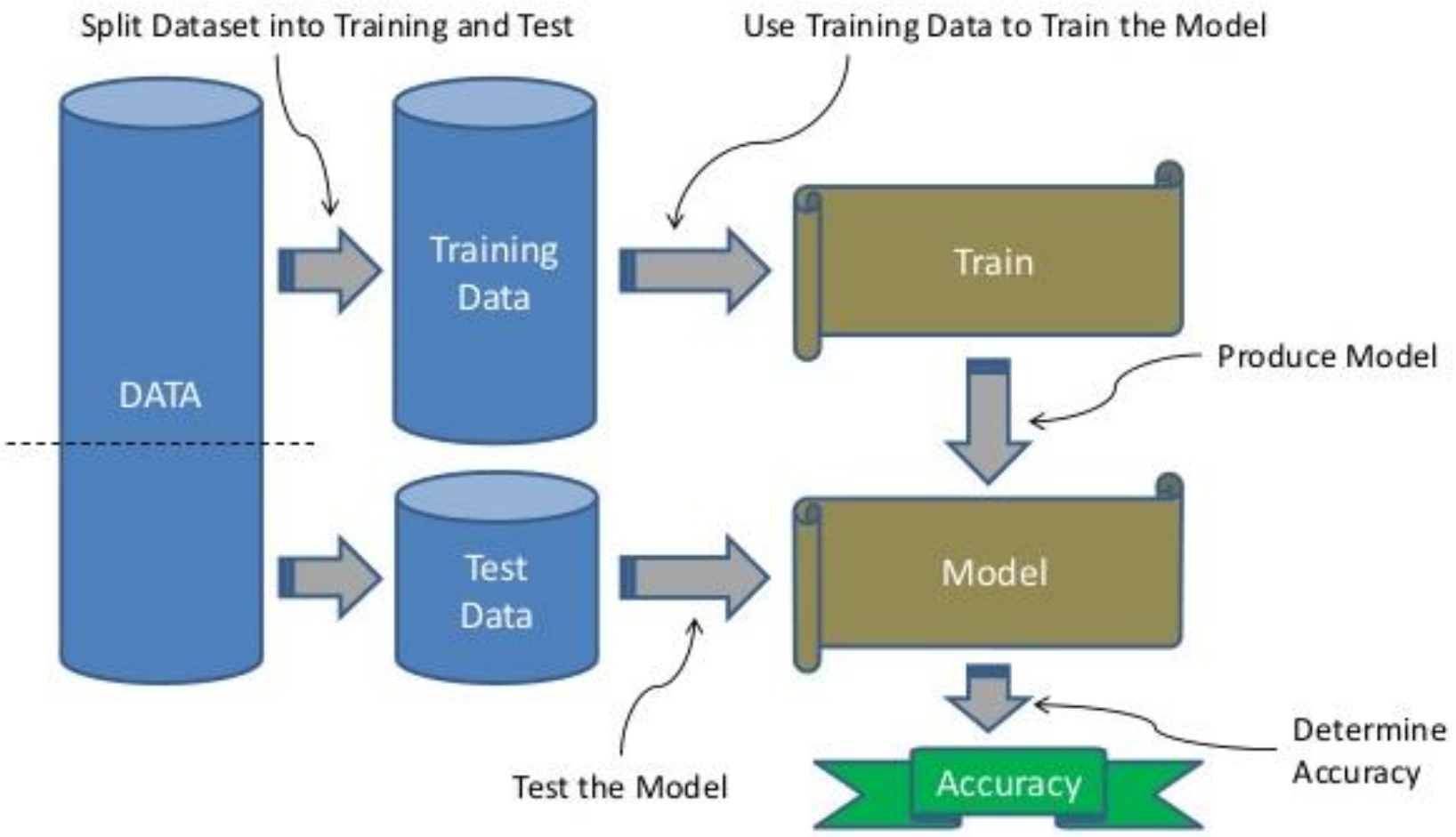
Different Models

Accuracy Assessment of Models

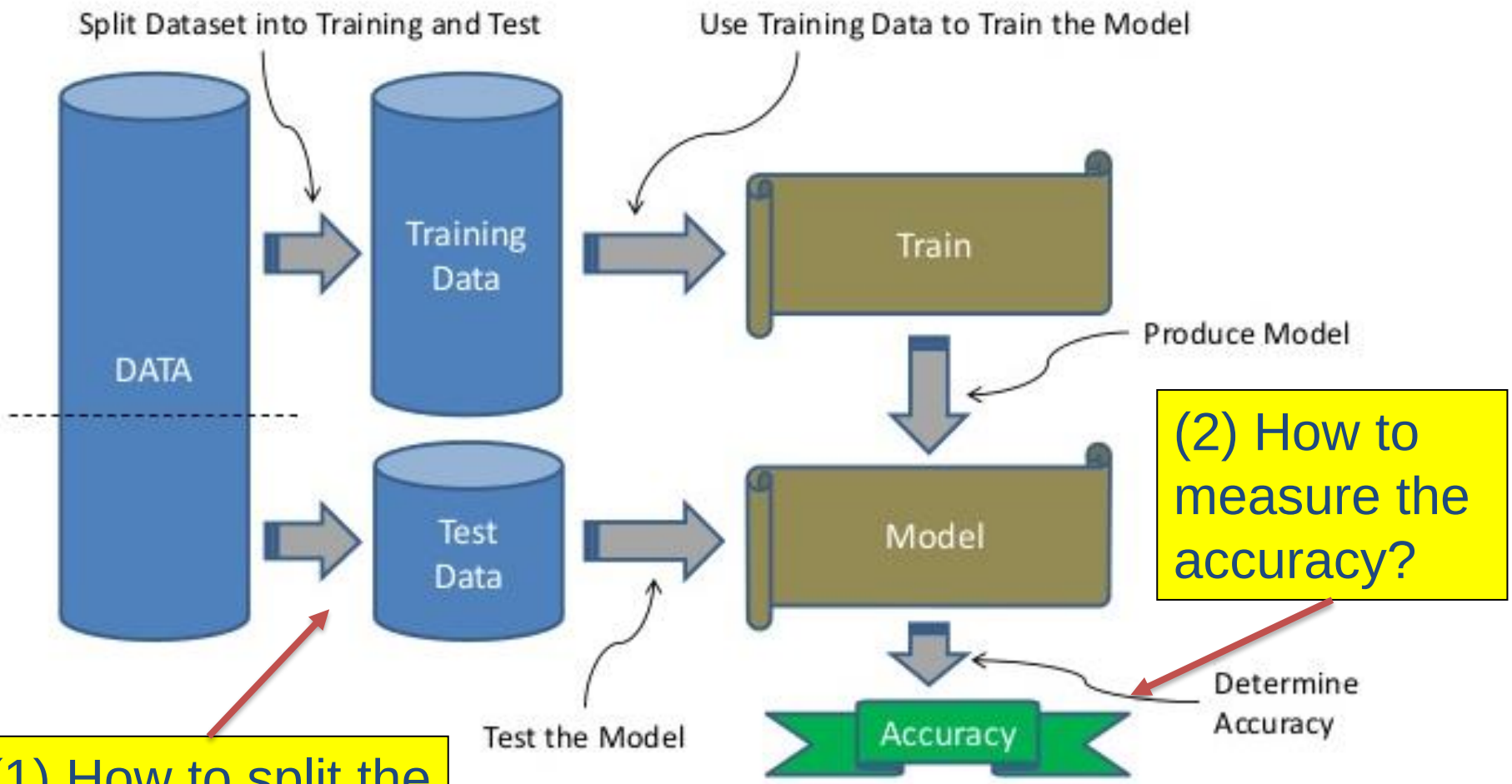
- Assess accuracy of a model output is one of the most important steps in digital soil mapping



Training and Testing the Models



Accuracy Assessment of Models



How to split the data?

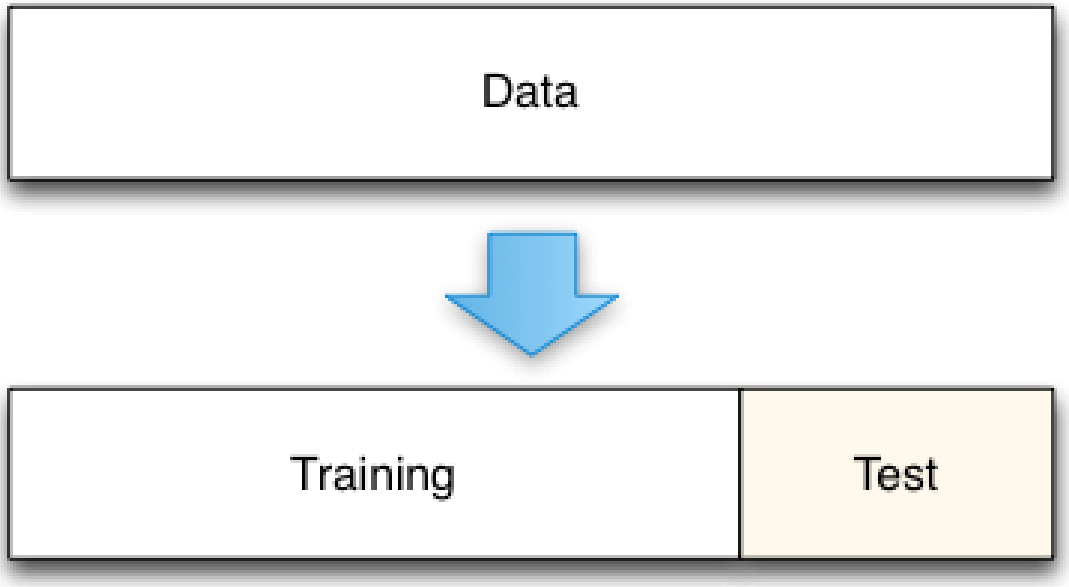
- Holdout method
- K-Folds Cross Validation
- Leave-one-out cross-validation
- Spatial K-Folds Cross Validation

Holdout method

- Holdout method:
  - In this approach we randomly split the complete data into training and test sets.
  - Then perform the model training on the training set and use the test set for validation purpose.
  - Ideally split the data into 70:30 or 80:20.
  - With this approach there is a possibility of high bias if we have limited data, because we would miss some information about the data which we have not used for training.
  - If our data is huge and our test sample and train sample has the same distribution then this approach is acceptable.

Holdout method

- Holdout method:



K-Folds Cross Validation

- K-Folds Cross Validation:
  - K-Folds technique is a popular method.
  - It generally results in a less biased model compare to other methods. Because it ensures that every observation from the original dataset has the chance of appearing in training and test set.
  - This is one among the best approach if we have a limited input data.

K-Folds Cross Validation

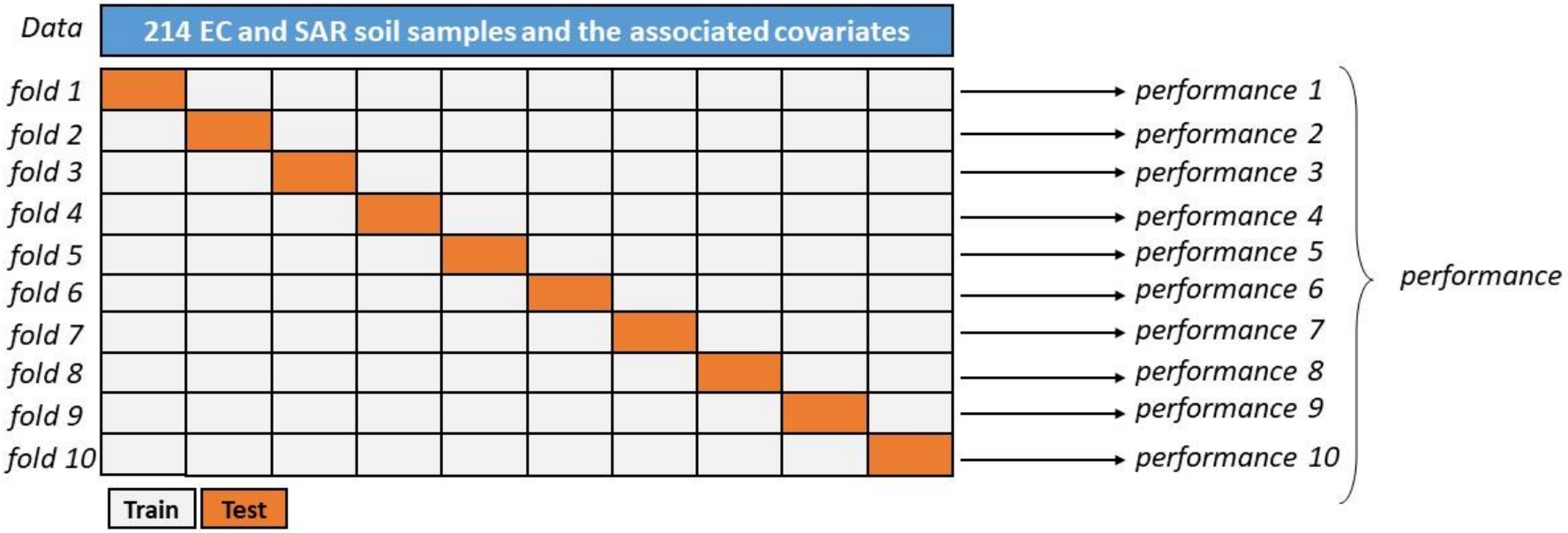
- K-Folds Cross Validation:
  1. Split the entire data randomly into K folds (value of K shouldn't be too small or too high, ideally we choose 5 to 10 depending on the data size).

  2. Then fit the model using the K-1 (K minus 1) folds and validate the model using the remaining Kth fold. Note down the scores/errors.

  3. Repeat this process until every K-fold serve as the test set. Then take the average of your recorded scores. That will be the performance metric for the model.

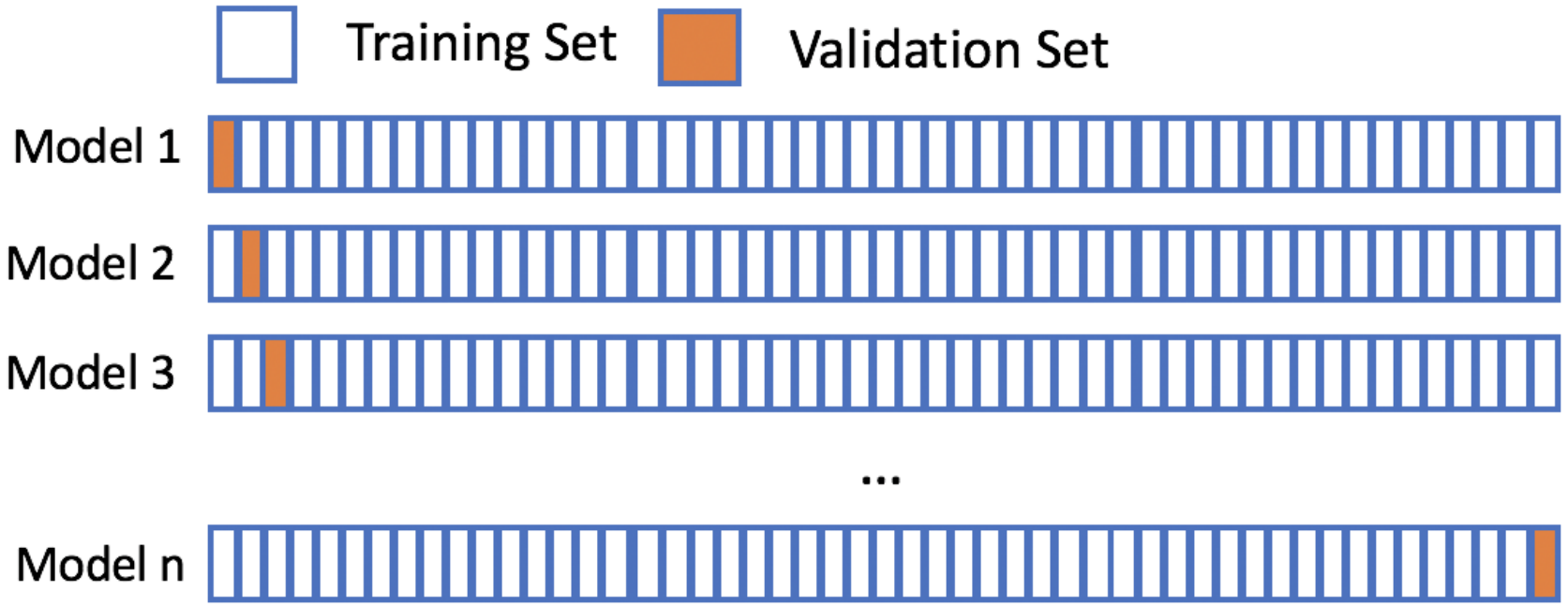
K-Folds Cross Validation

- K-Folds Cross Validation: k = 10



Leave-one-out cross-validation

- Leave-one-out cross-validation: *k=n*



Spatial K-Folds Cross Validation

- Spatial K-Folds Cross Validation:

- Spatial data exhibit a few properties that make it difficult to apply standard statistical methods to them.
- Spatial data exhibit spatial autocorrelation, where observations close to each other in space have related values
- K-fold CV, however, does not resample spatial groups of data. Instead, it uses random resampling, which produces training and validation sets whose points are distinct but come from overlapping spatial regions.
- Spatial cross-validation involves modifications of CV for spatial data. Instead of defining folds of points at random, folds are defined by spatial boundaries.

Spatial K-Folds Cross Validation

- Spatial K-Folds Cross Validation:

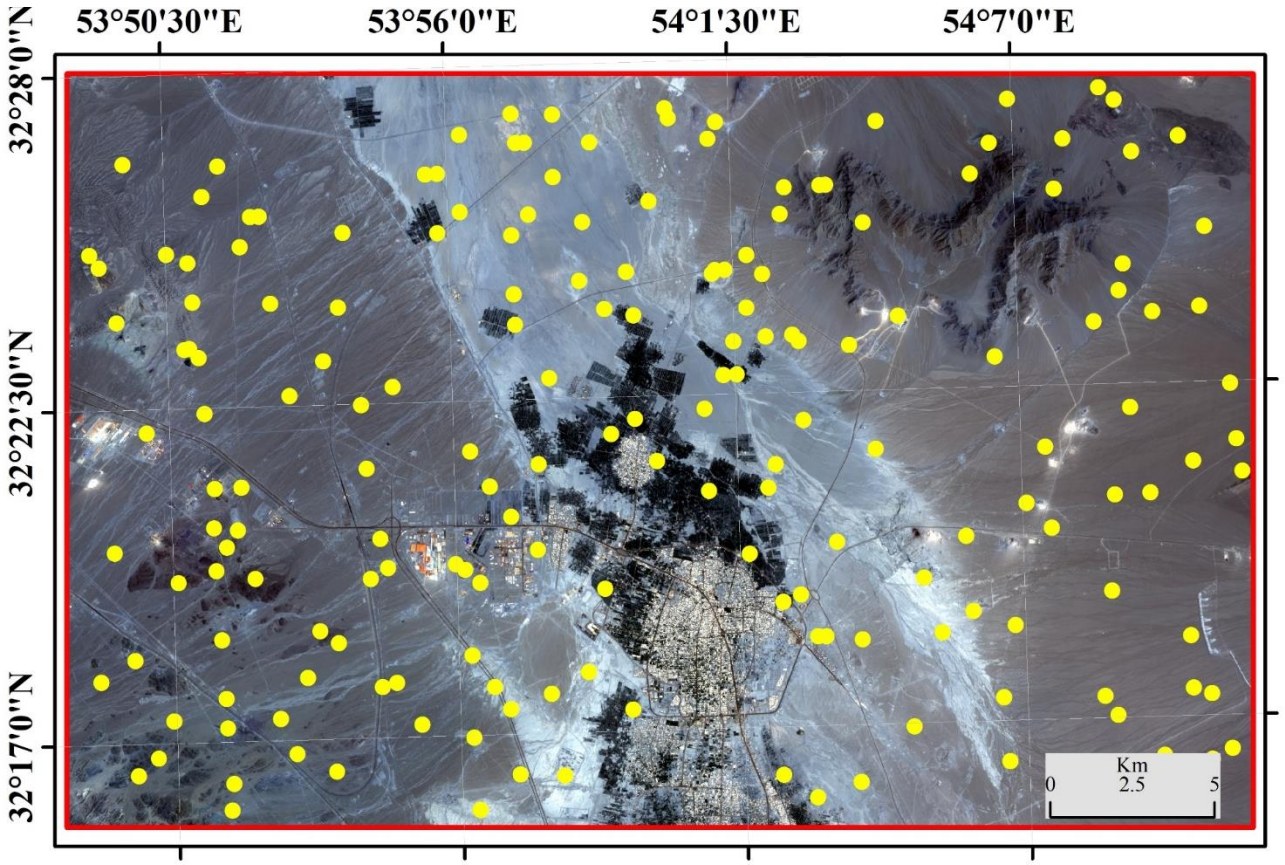


Illustration of the spatial blocking strategy. The numbers in the blocks are fold numbers, showing allocation of blocks to folds.

Spatial K-Folds Cross Validation

- Spatial K-Folds Cross Validation:

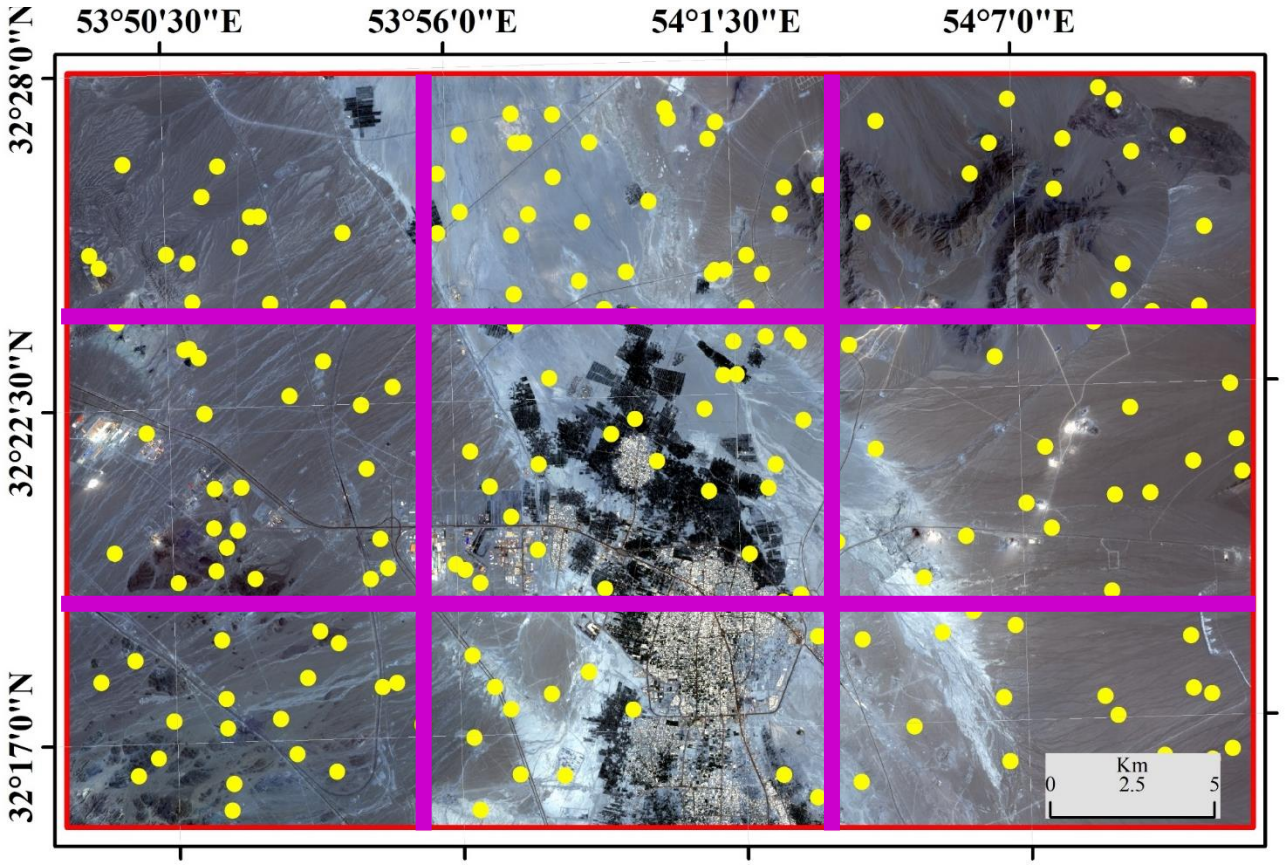


Illustration of the spatial blocking strategy. The numbers in the blocks are fold numbers, showing allocation of blocks to folds.

Spatial K-Folds Cross Validation

- Spatial K-Folds Cross Validation:

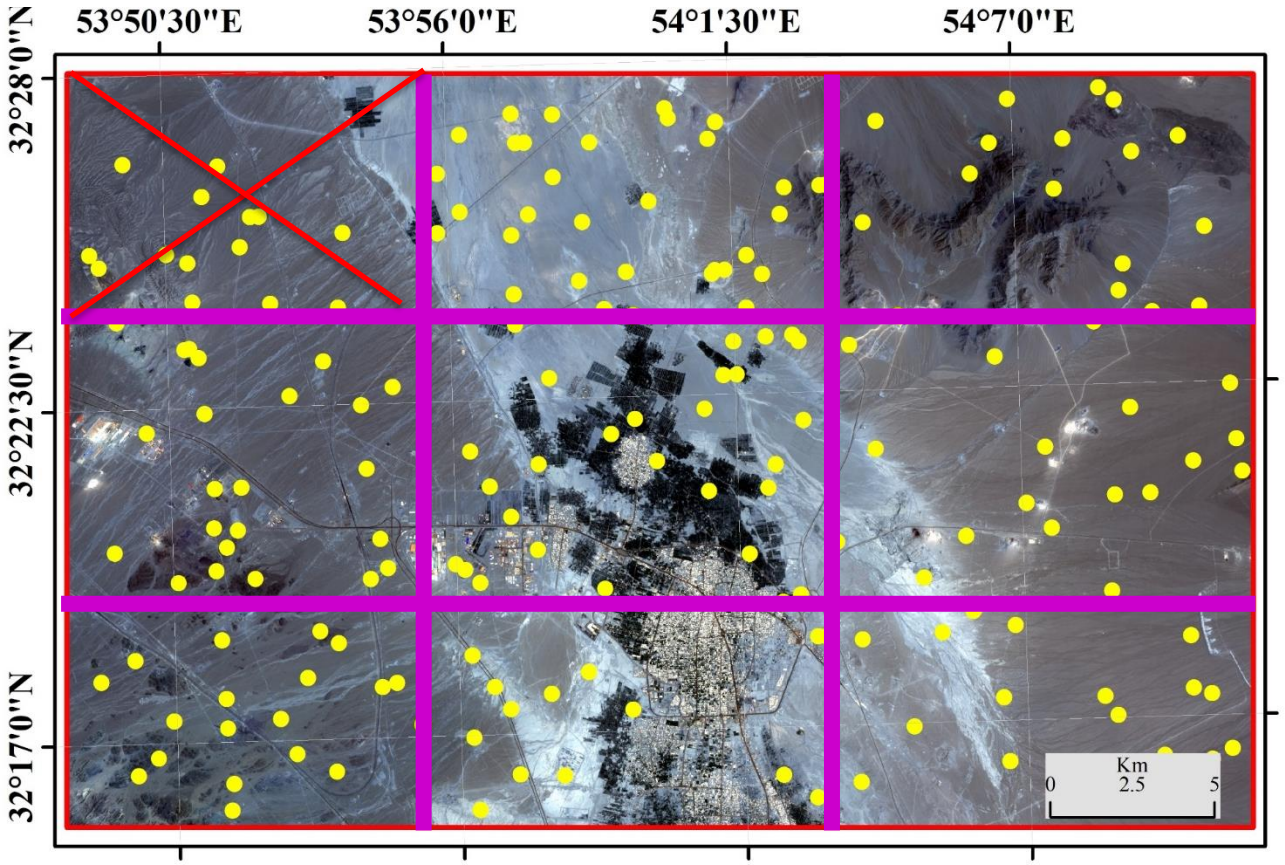


Illustration of the spatial blocking strategy. The numbers in the blocks are fold numbers, showing allocation of blocks to folds.

Spatial K-Folds Cross Validation

- Spatial K-Folds Cross Validation:

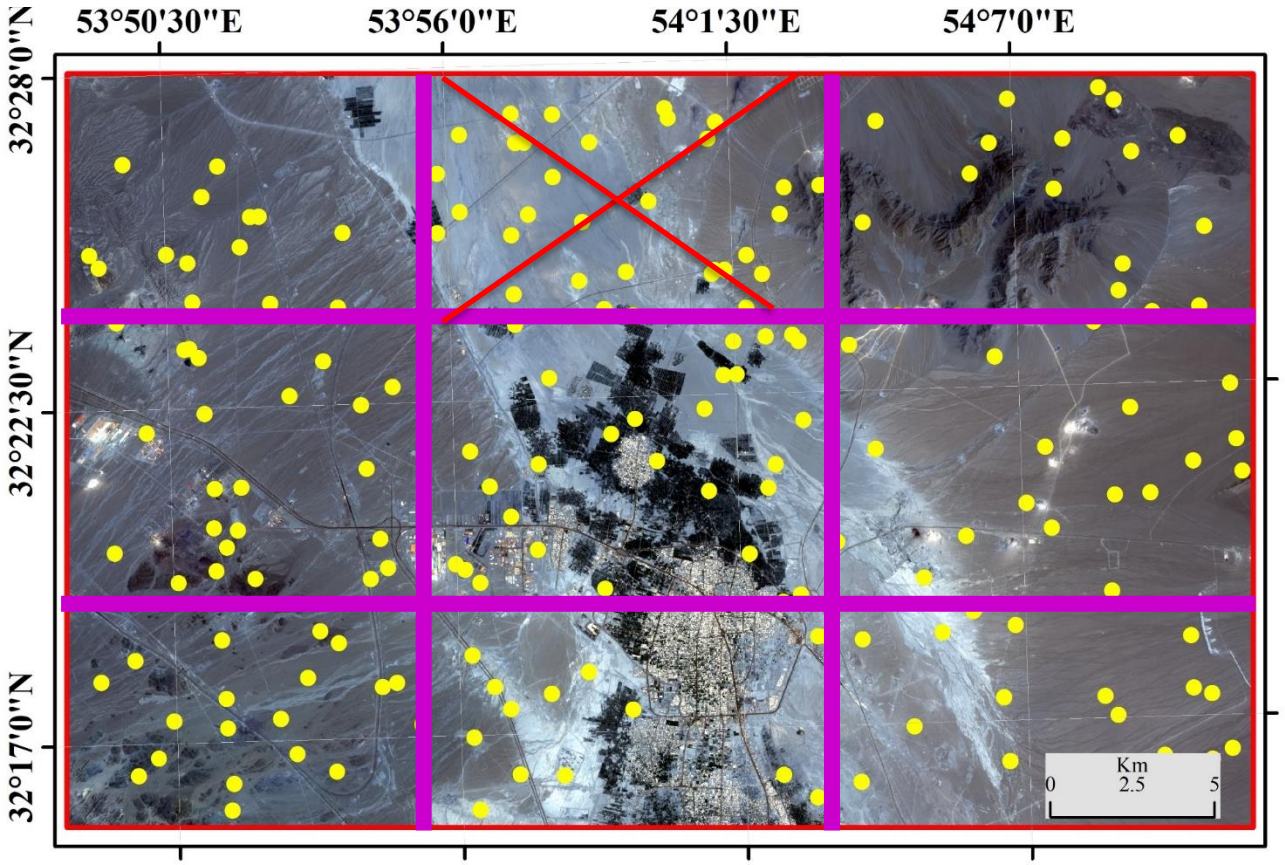


Illustration of the spatial blocking strategy. The numbers in the blocks are fold numbers, showing allocation of blocks to folds.

Spatial K-Folds Cross Validation

- Spatial K-Folds Cross Validation:

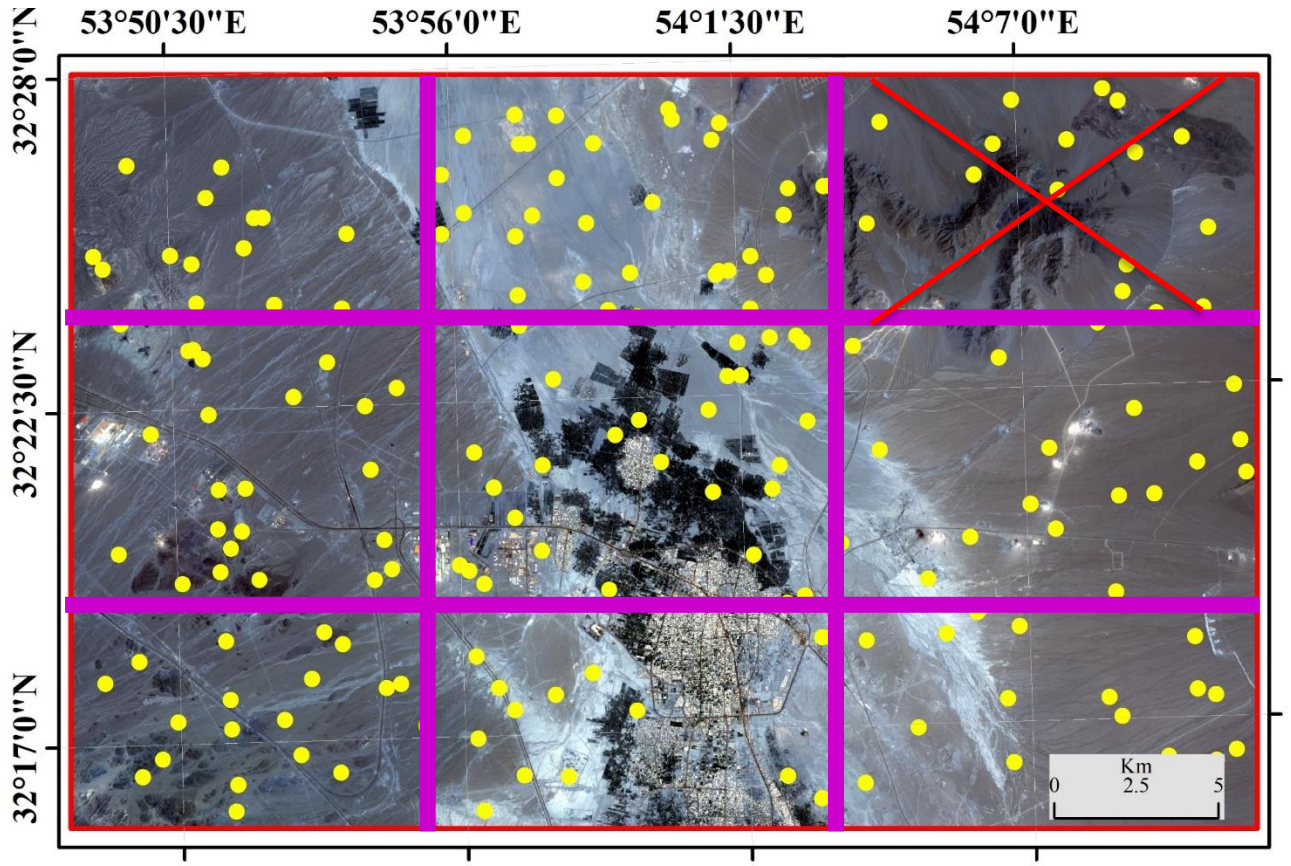


Illustration of the spatial blocking strategy. The numbers in the blocks are fold numbers, showing allocation of blocks to folds.

Spatial K-Folds Cross Validation

- Spatial K-Folds Cross Validation:

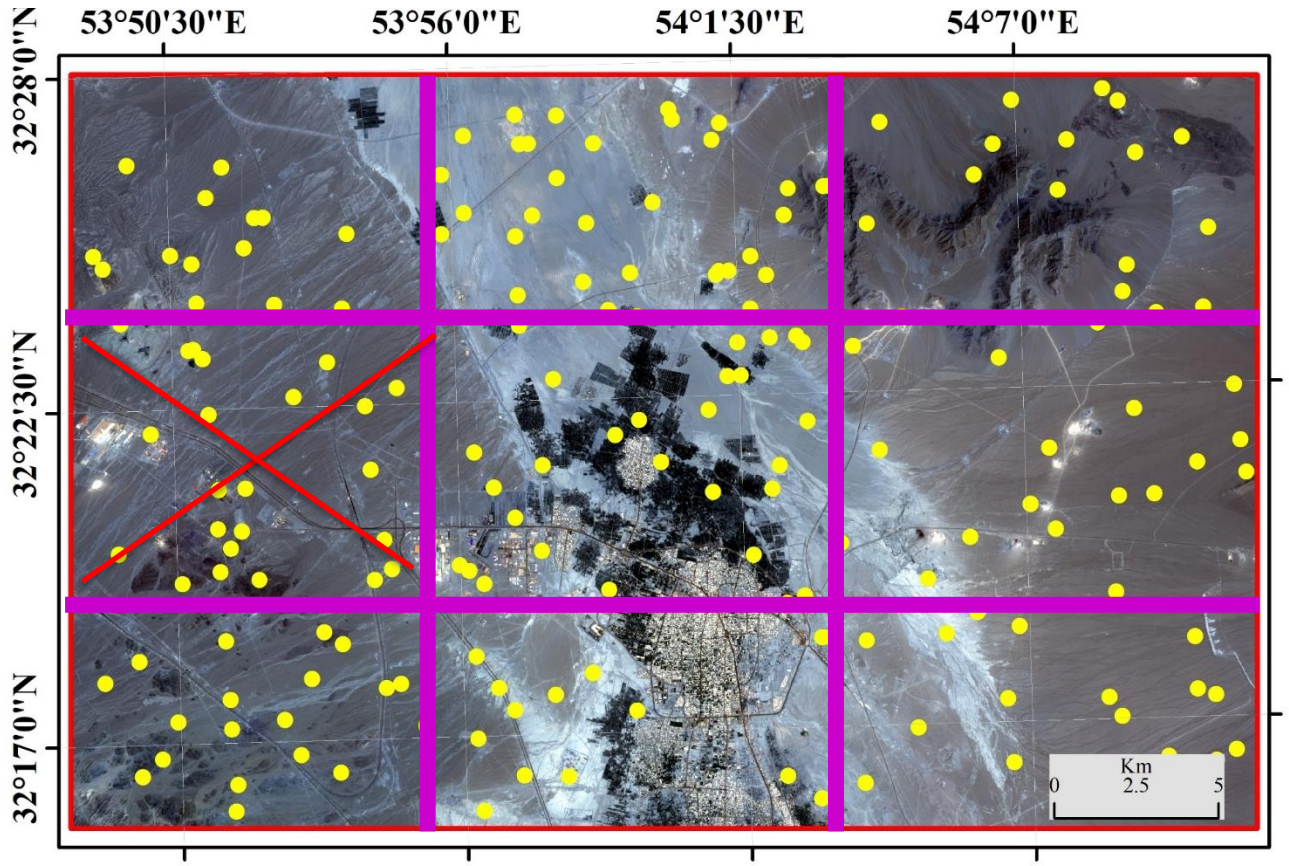
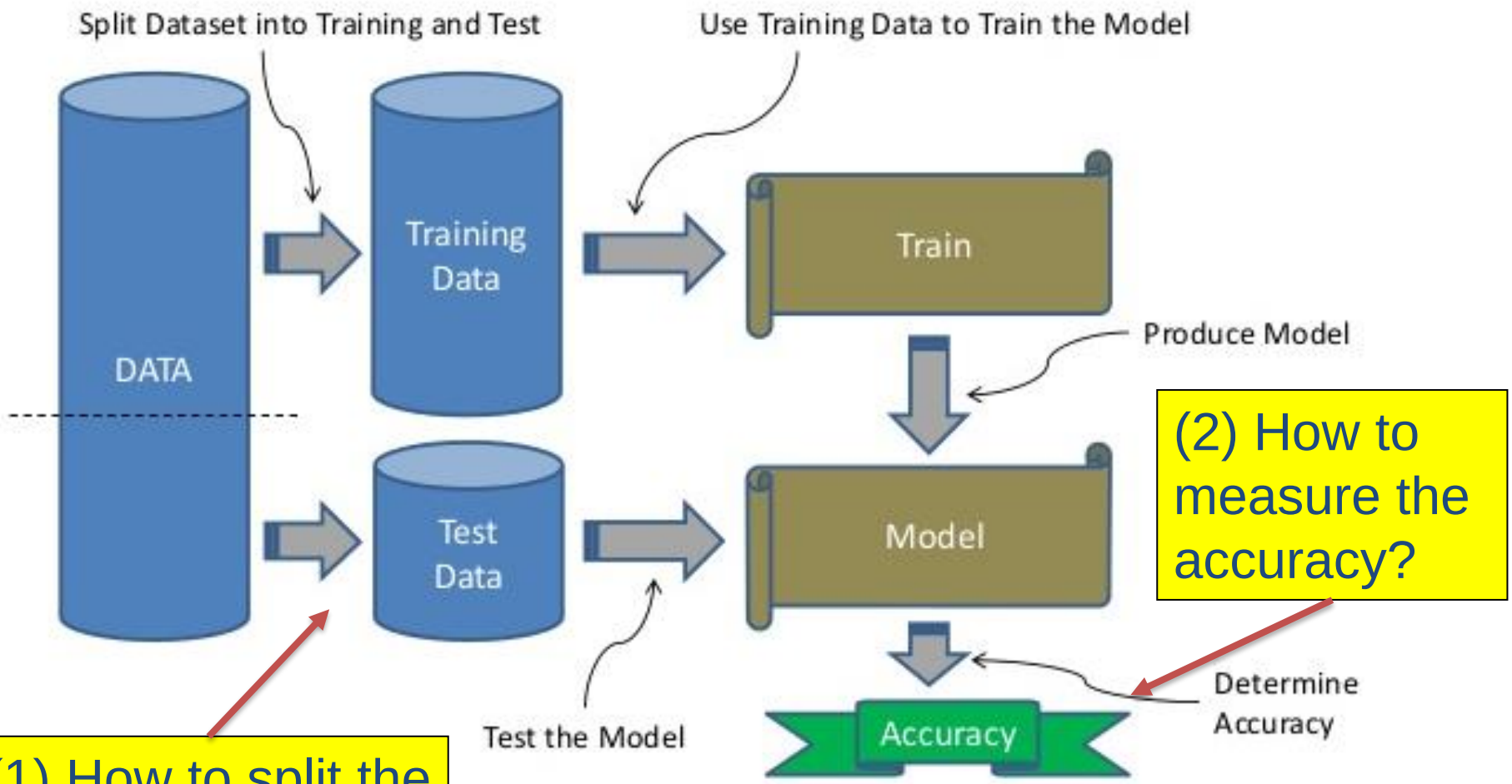


Illustration of the spatial blocking strategy. The numbers in the blocks are fold numbers, showing allocation of blocks to folds.

Accuracy Assessment of Models



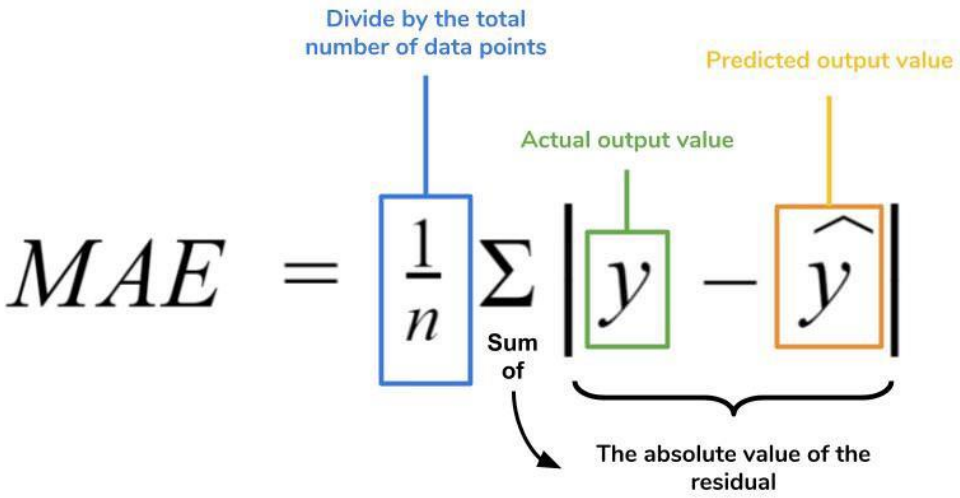
How to measure the accuracy?

- Performance Metrics:

Regression	Classification
• Mean Absolute Error (MAE) • Root Mean Squared Error (RMSE) • R-Squared and Adjusted R-Squared	• Recall • Precision • F1-Score • Accuracy • Area Under the Curve (AUC)

Performance Metrics for Regression

- Mean Absolute Error:

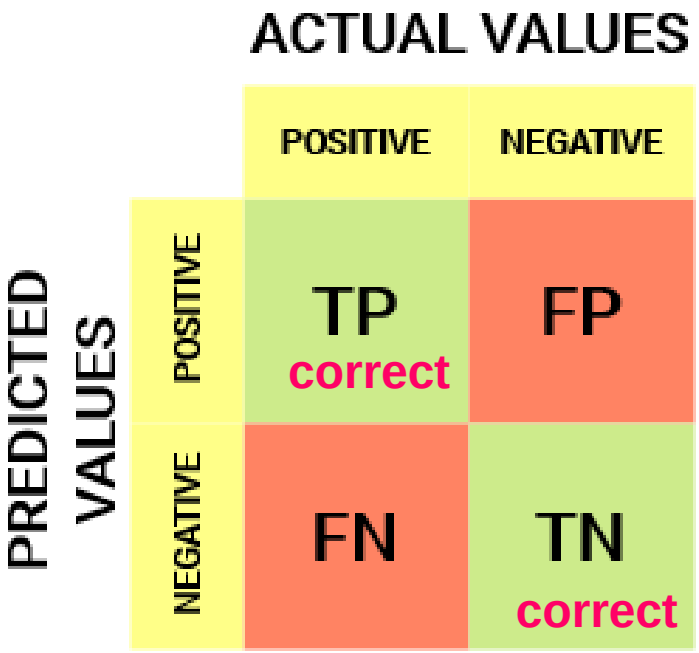


- Root Mean Square Error:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Performance Metrics for Classification

- Confusion Matrix: is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of the model.



where TP, TN, FP and FN are true positive, true negative, false positive and false negative, respectively

Performance Metrics for Classification

- Confusion Matrix:

Class A: Aridisols

Class B: Entisols

	Class A: Aridisoils	Class B: Entisols
Class A: Aridisoils		
Class B: Entisols		

Performance Metrics for Classification

- Confusion Matrix:

Class A: Aridisols

Class B: Entisols

Actual Test data

Predicted Test data

	Class A: Aridisoils	**Class B: Entisols**
Class A: Aridisoils		
Class B: Entisols		

Performance Metrics for Classification

- Confusion Matrix:

Class A: Aridisols

Class B: Entisols

Actual Test data

1 2 3 4 5

Predicted Test data

1 2 3 4 5

	Class A: Aridisols	**Class B: Entisols**
Class A: Aridisols		
Class B: Entisols		

Performance Metrics for Classification

- Confusion Matrix:

Class A: Aridisols

Class B: Entisols

Actual Test data

1 2 3 4 5

Predicted Test data

1 2 3 4 5

	Class A: Aridisols	**Class B: Entisols**
Class A: Aridisols	| |	
Class B: Entisols		

Performance Metrics for Classification

- Confusion Matrix:

Class A: Aridisols

Class B: Entisols

Actual Test data

Predicted Test data

	Class A: Aridisoils	Class B: Entisols
Class A: Aridisoils	| |	
Class B: Entisols	|	

Performance Metrics for Classification

- Confusion Matrix:

Class A: Aridisols

Class B: Entisols

Actual Test data

1 2 3 4 5

Predicted Test data

1 2 3 4 5

	Class A: Aridisols	**Class B: Entisols**
Class A: Aridisols	| |	
Class B: Entisols	|	|

Performance Metrics for Classification

- Confusion Matrix:

Class A: Aridisols

Class B: Entisols

Actual Test data

1 2 3 4 5

Predicted Test data

1 2 3 4 5

	Class A: Aridisols	**Class B: Entisols**
Class A: Aridisols	| |	|
Class B: Entisols	|	|

Performance Metrics for Classification

- Confusion Matrix:

Class A: Aridisols

Class B: Entisols

Actual Test data

1 2 3 4 5

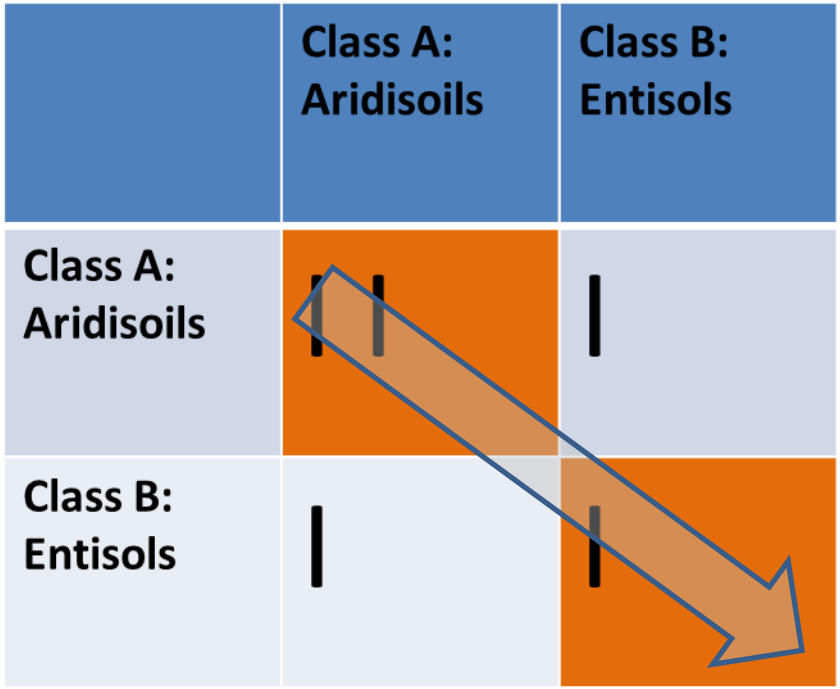
Predicted Test data

1 2 3 4 5

	Class A: Aridisols	**Class B: Entisols**
Class A: Aridisols	2	1
Class B: Entisols	1	1

Performance Metrics for Classification

- Overall Accuracy: is a metric calculating the classifier overall accuracy

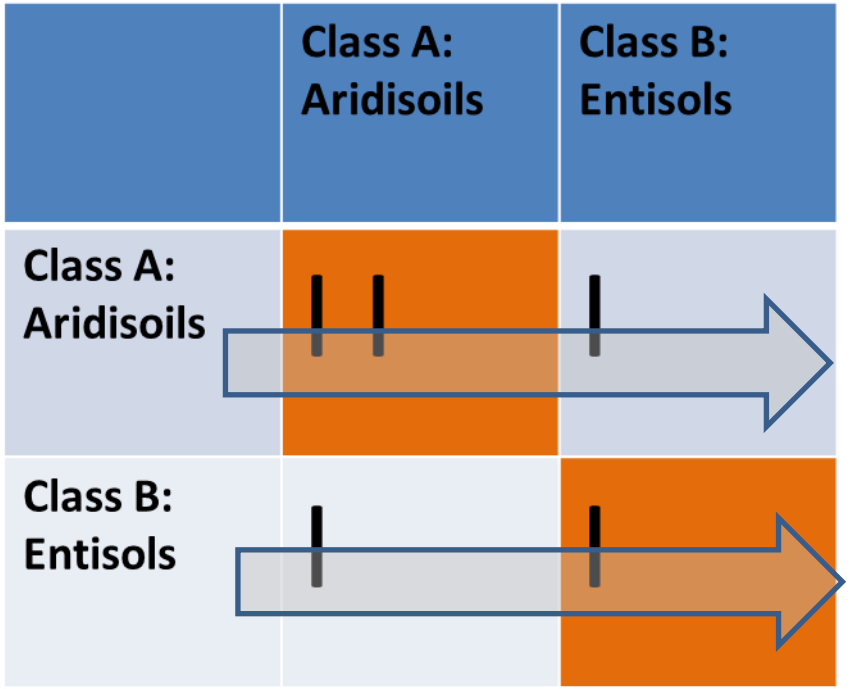


$$OA = \left(\frac{\text{Correctly classified}}{\text{Total number of test data}}\right) * 100$$

$$OA = \left(\frac{2+1}{5}\right) * 100 = 60\%$$

Performance Metrics for Classification

- Precision: is the proportion of those predicted instances that are correctly classified



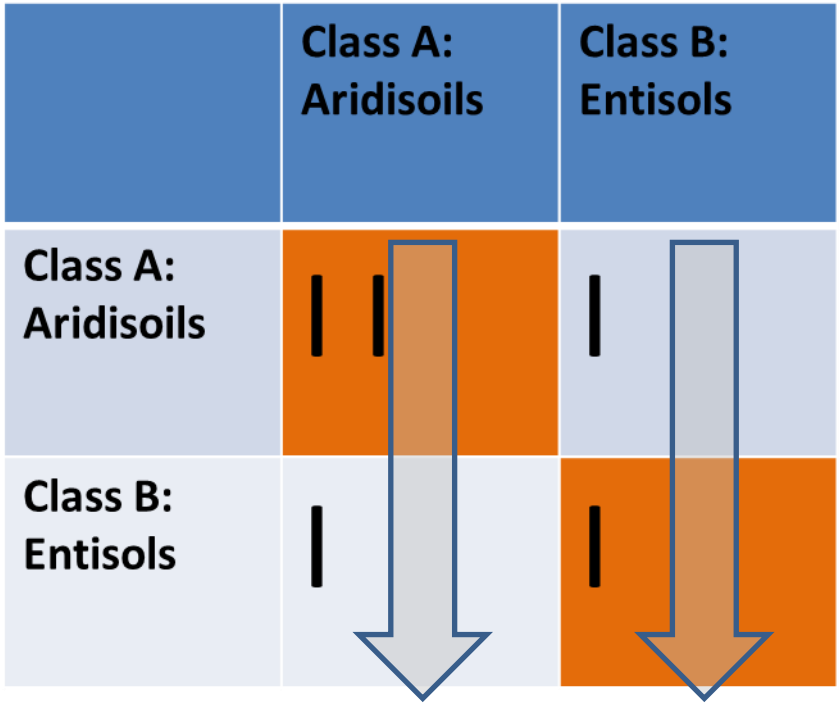
$$Pr = \left(\frac{\text{Correctly classified for each class}}{\text{Total number of row data}}\right) * 100$$

$$Pr(class\ A) = \left(\frac{2}{2+1}\right) * 100 = 66\%$$

$$Pr(class\ B) = \left(\frac{1}{1+1}\right) * 100 = 50\%$$

Performance Metrics for Classification

- Recall: is the proportion of those instances that are correctly classified



$$Re = \left(\frac{\text{Correctly classified for each class}}{\text{Total number of coloum data}}\right) * 100$$

$$Re(class\ A) = \left(\frac{2}{2+1}\right) * 100 = 66\%$$

$$Re(class\ B) = \left(\frac{1}{1+1}\right) * 100 = 50\%$$

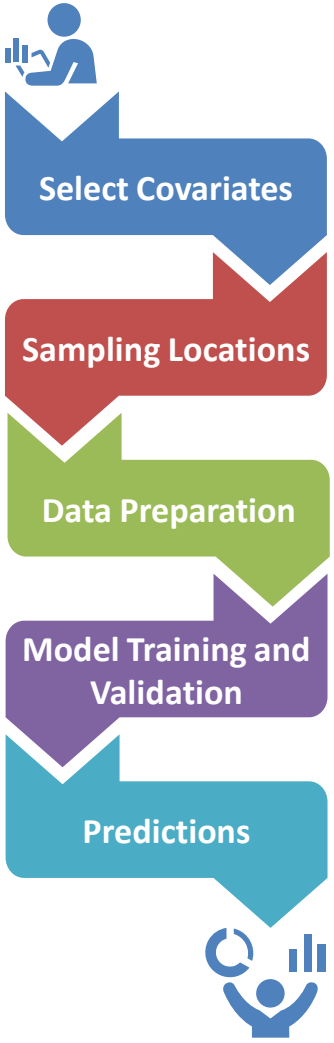
Performance Metrics for Classification

- F-score: the F-score is the harmonic mean of precision and recall

$$F - score = \left(\frac{2 \times Preciosion \times Recall}{Preciosion + Recall} \right) * 100$$

Metric	Formula
Accuracy	$ACC = \frac{TP+TN}{TP+TN+FP+FN}$
Error rate	$ERR = \frac{FP+FN}{TP+TN+FP+FN}$
Precision	$PRC = \frac{TP}{TP+FP}$
Sensitivity Specificity	$SNS = \frac{TP}{TP+FN}$ $SPC = \frac{TN}{TN+FP}$
ROC	$ROC = \frac{\sqrt{SNS^2+SPC^2}}{\sqrt{2}}$
F_1 score	$F_1 = 2\frac{PRC \cdot SNS}{PRC+SNS}$
Geometric Mean	$GM = \sqrt{SNS \cdot SPC}$

Sequence of DSM Steps



1. Environmental covariates, relevant as predictors of soil property/class, are derived from remote sensing, digital elevation, climatic datasets, …
2. Soil samples are collected at the specified locations (e.g., Latin hypercube sampling) and soil property is measured in the laboratory.
3. Intersecting the covariates with the soil point observations.
4. Machine learning models (e.g., random forest) are trained using training data, and accuracy assessment is carried out using the test data set.
5. The ML models are applied to the entire study area in order to produce a soil property/class map.

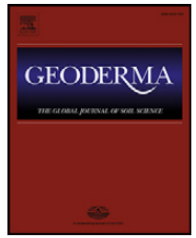
DSM Examples using Machine Learning

Geoderma 265 (2016) 62–77

Contents lists available at ScienceDirect

Geoderma

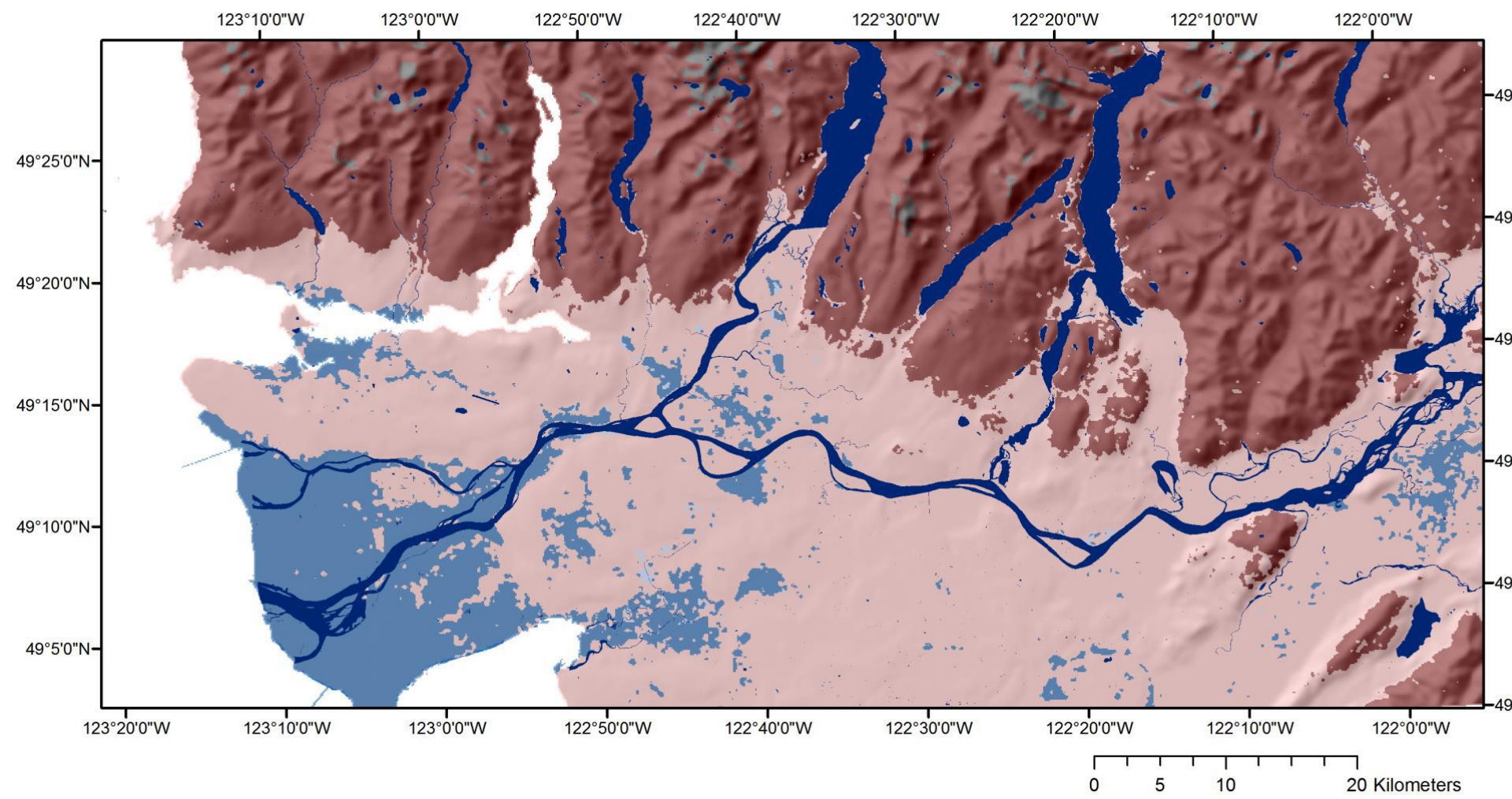
journal homepage: www.elsevier.com/locate/geoderma



An overview and comparison of machine-learning techniques for classification purposes in digital soil mapping

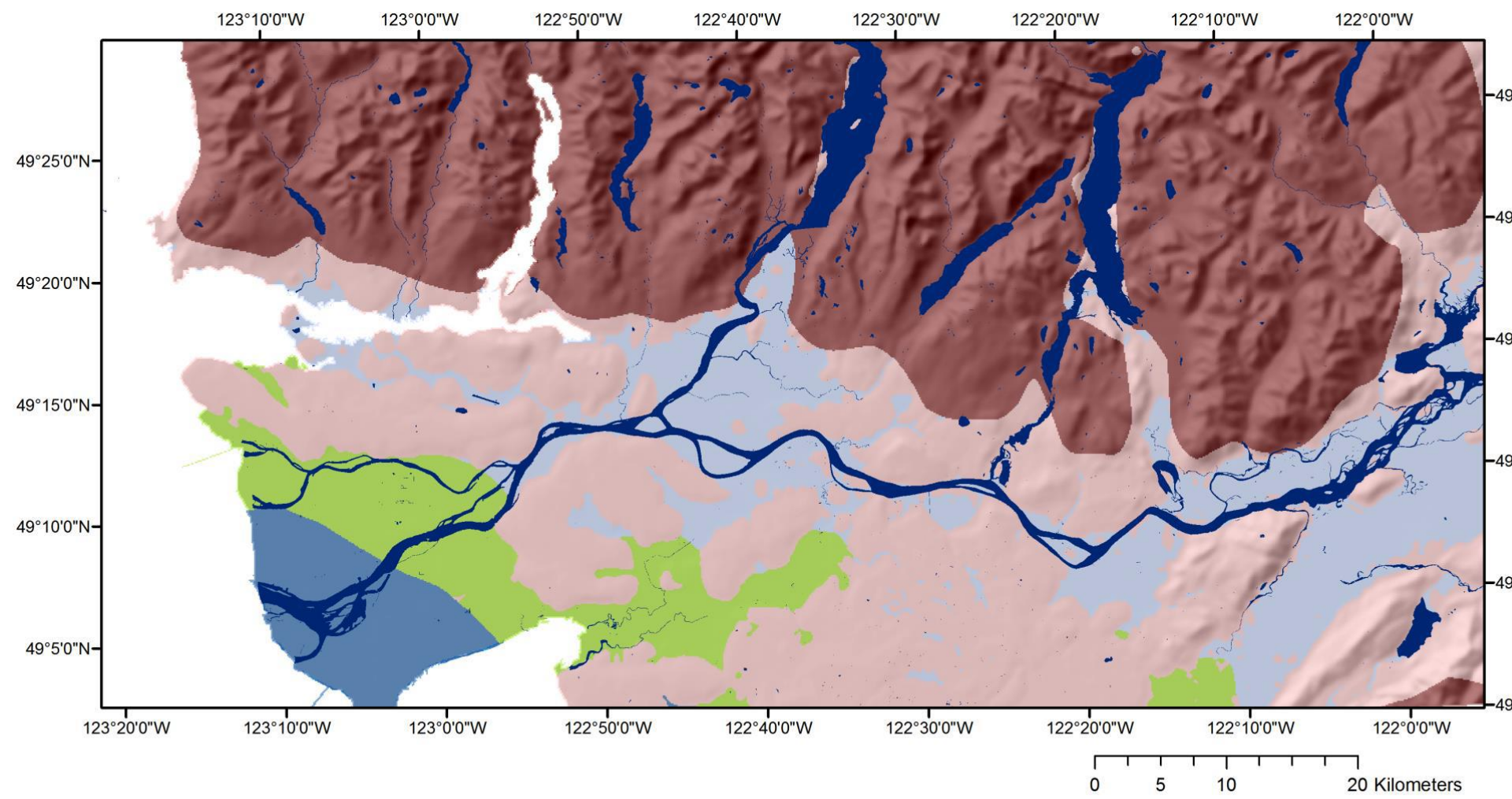
Brandon Heung [a,*], Hung Chak Ho [b], Jin Zhang [a], Anders Knudby [c], Chuck E. Bulmer [d], Margaret G. Schmidt [a]

[a] Soil Science Lab, Department of Geography, Simon Fraser University, 8888 University Drive, Burnaby, BC, V5A 1S6, Canada
[b] Remote Sensing and Spatial Predictive Modeling Lab, Department of Geography, Simon Fraser University, 8888 University Drive, Burnaby, BC, V5A 1S6, Canada
[c] Department of Geography, University of Ottawa, 60 University, Ottawa, ON, K1N 6N5, Canada
[d] British Columbia Ministry of Forests Lands and Natural Resources Operations, Natural Resource Sciences Section, Vernon, BC, V1B 2C7, Canada



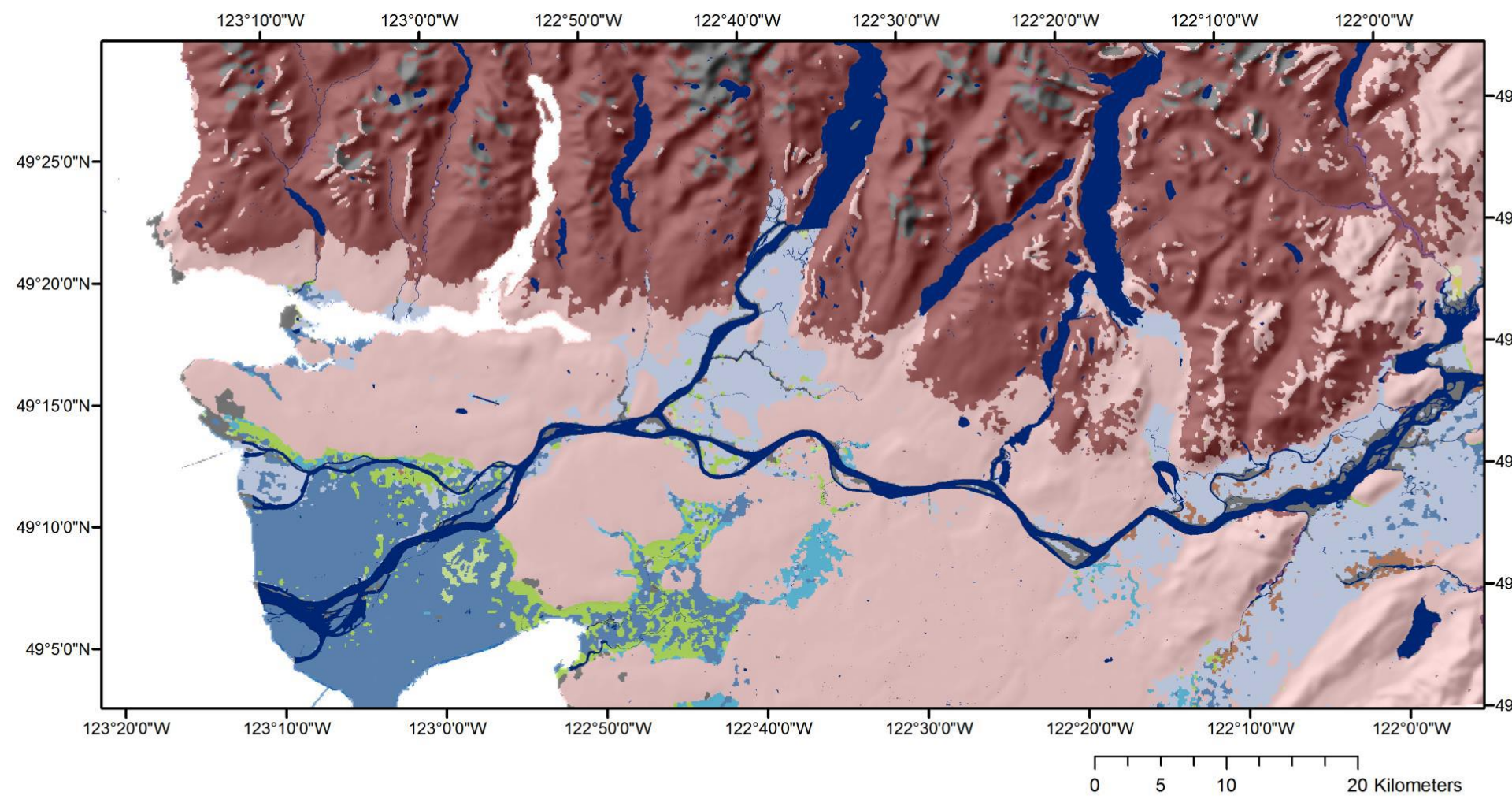
C= 40%

Area-Weighted: Nearest Shrunken Centroid



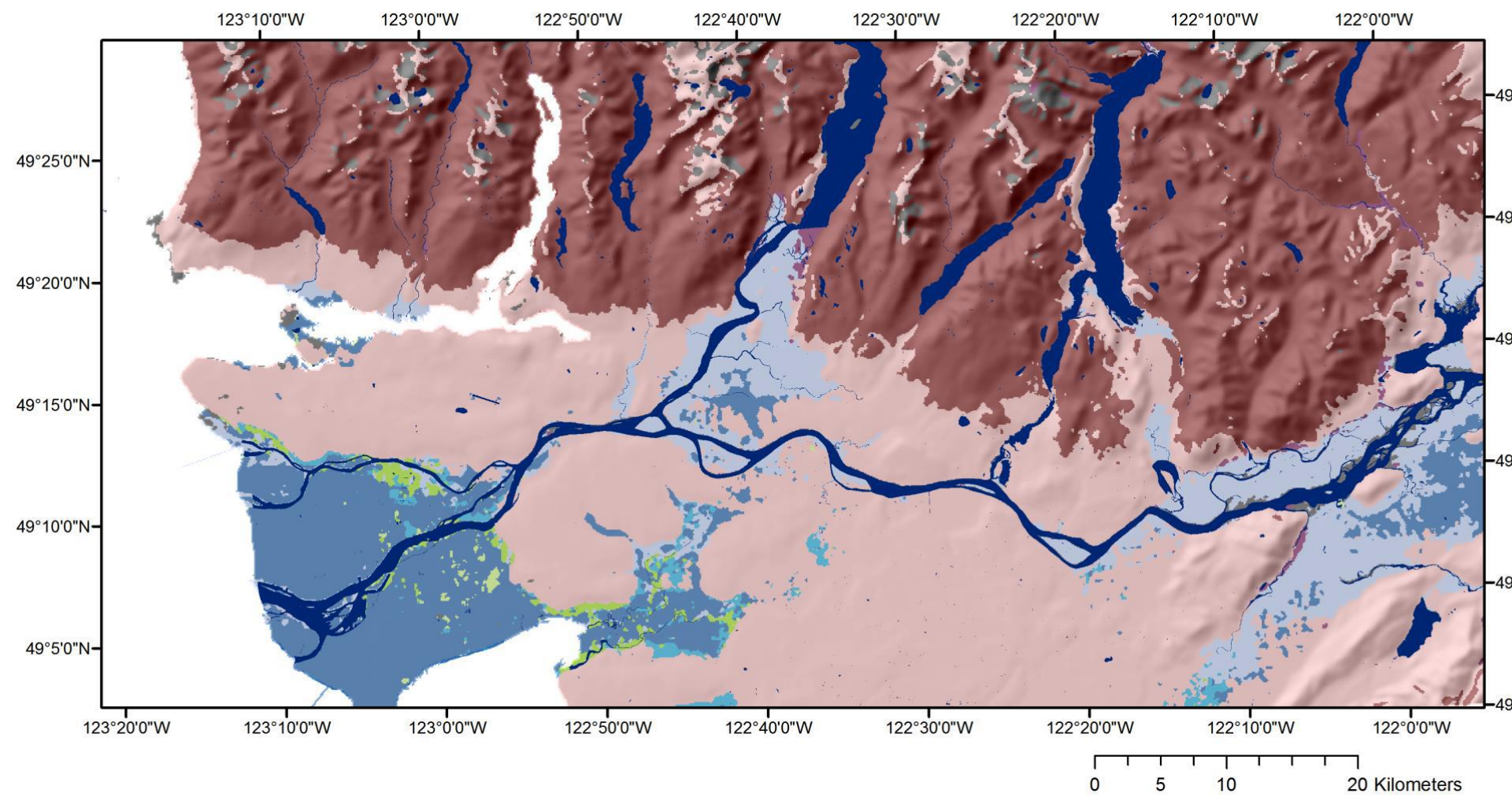
C= 42%

Area-Weighted: CART



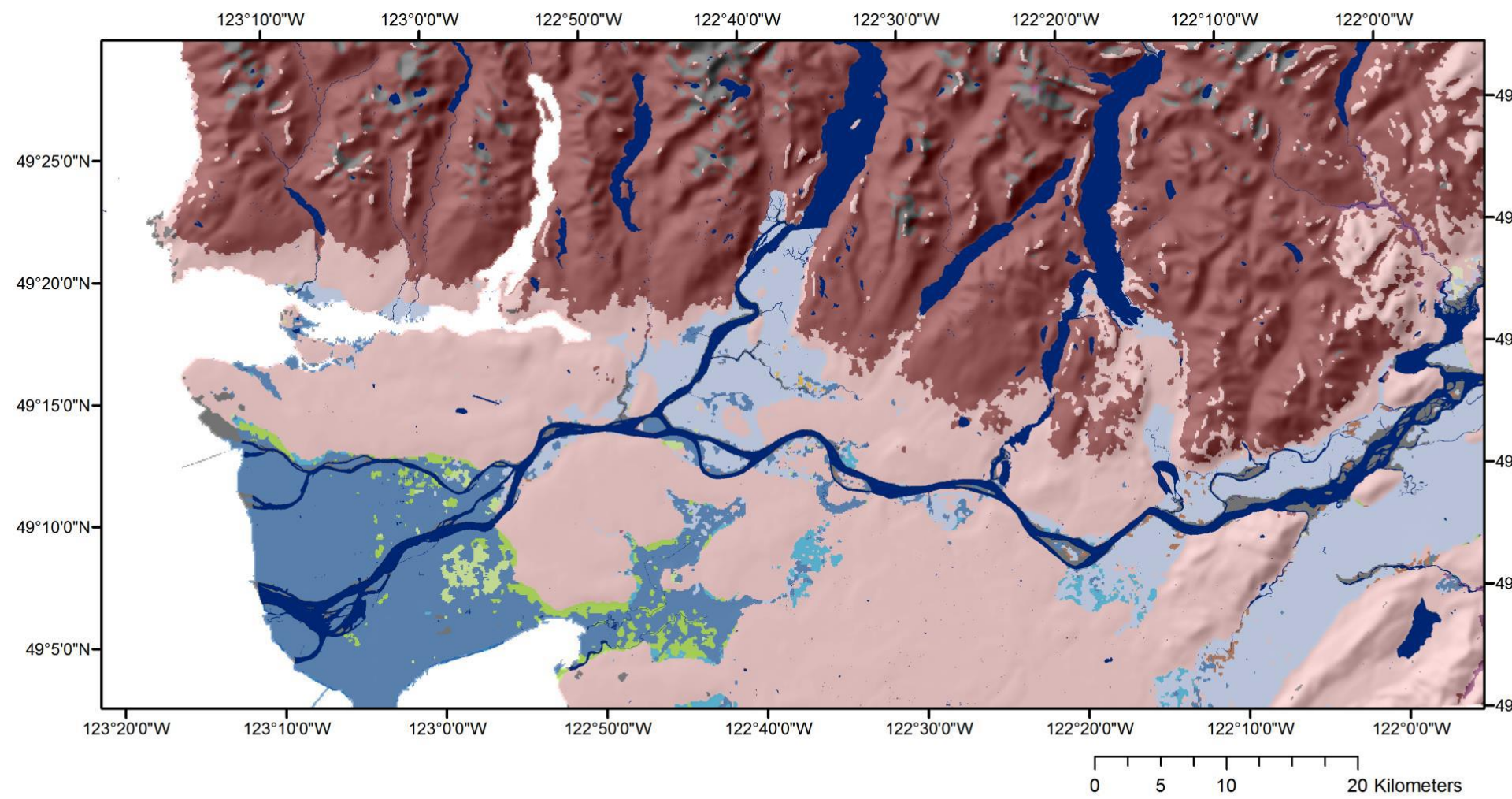
C= 48%

Area-Weighted: Multinomial Logistic Regression



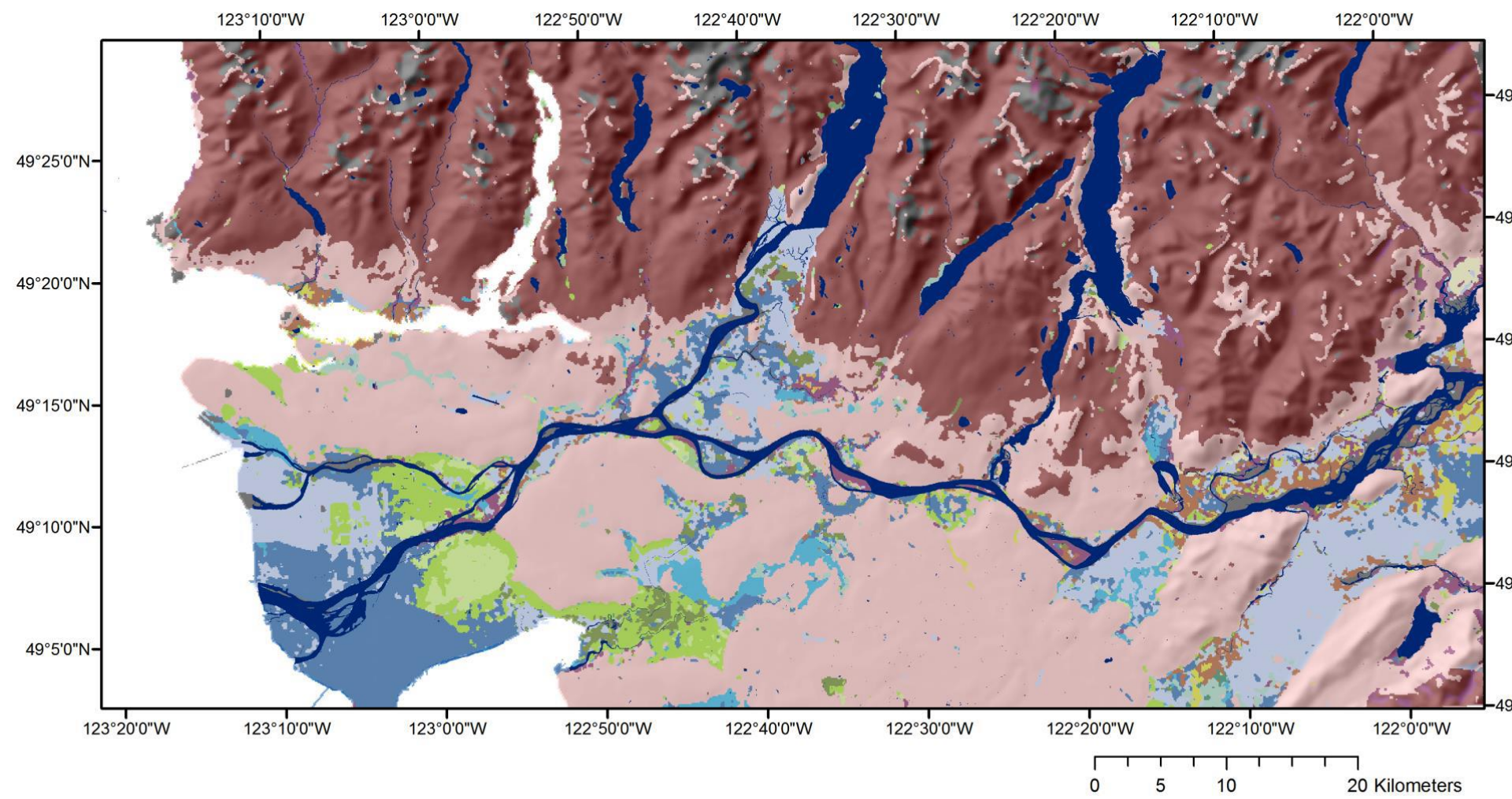
C= 49%

Area-Weighted: Artificial Neural Network



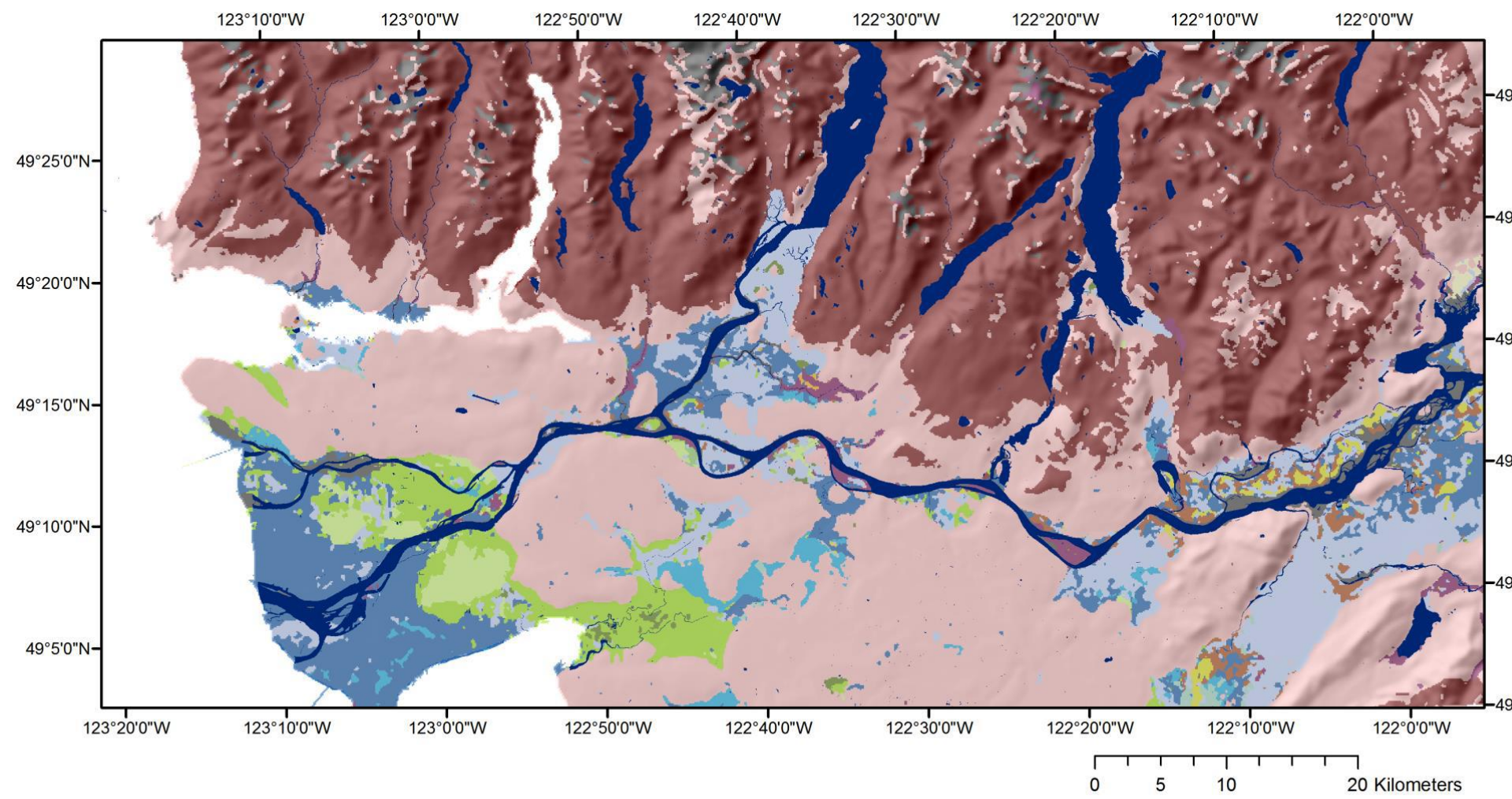
C= 50%

Area-Weighted: Support Vector Machine - Linear



C= 65%

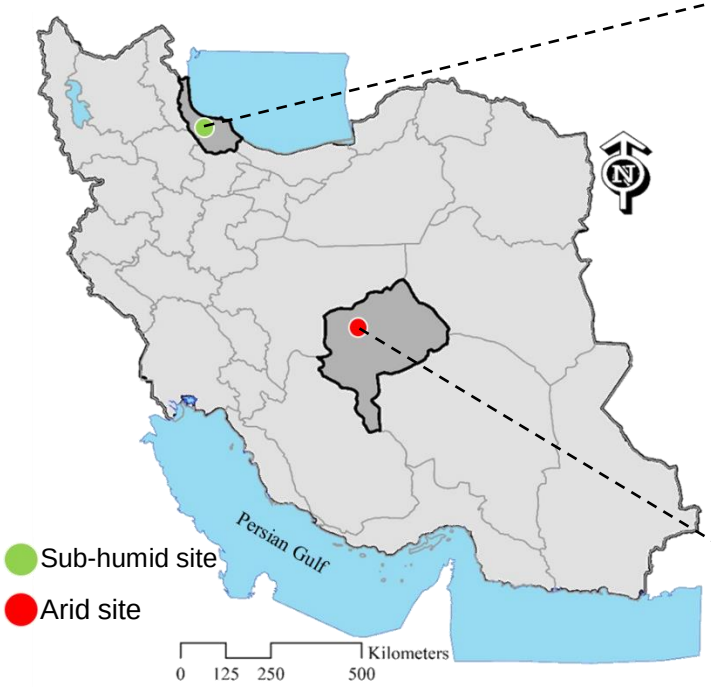
Area-Weighted: Logistic Model Tree

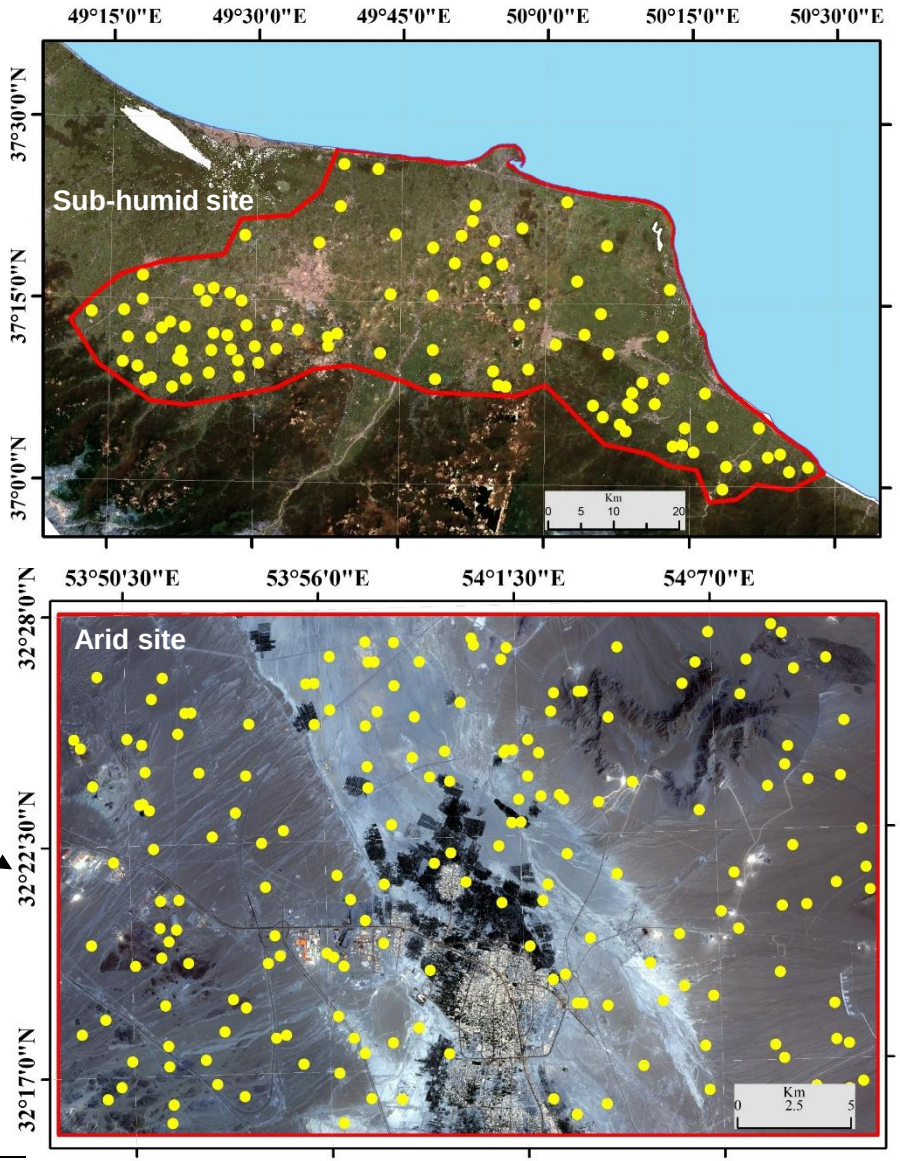


C= 66%

Area-Weighted: Random Forest

❑ **Methods**: Study area





Study sites and data collection details.

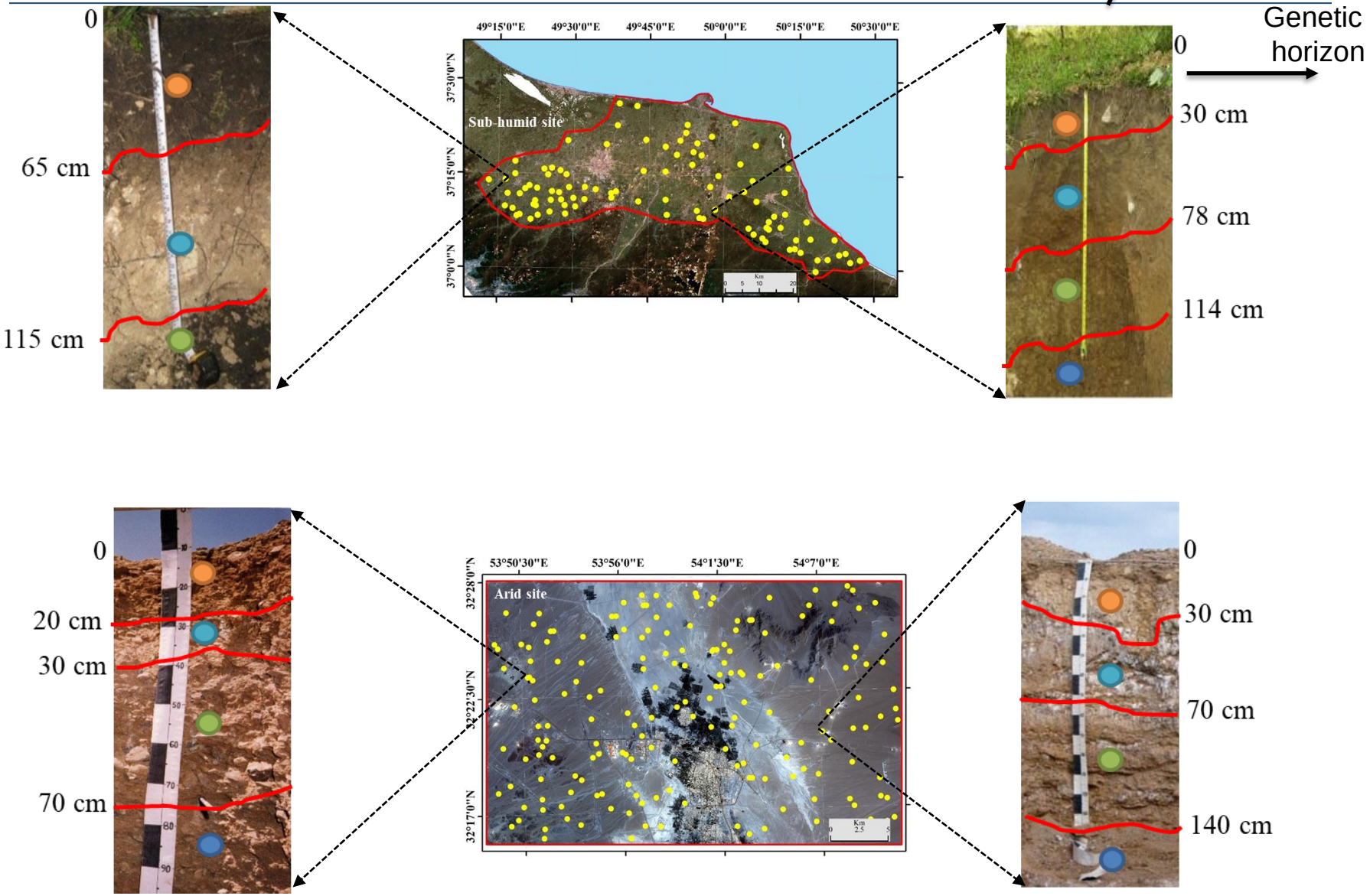
Site names	Area (km²)	Soil types	Climate conditions	Precipitation (mm/year)	Elevation (m)	Samples (no.)
Arid site	720	Solonchaks, Gypsisols and Regosols	Arid	75	944–1944	154
Sub-Humid site	3000	Kastanozems, Cambisols and Chernozems	Sub-Humid	1200	-26–700	99

❑ **Methods**: Soil sampling

$S = f(S, C, O, R, P, A, N) + e$

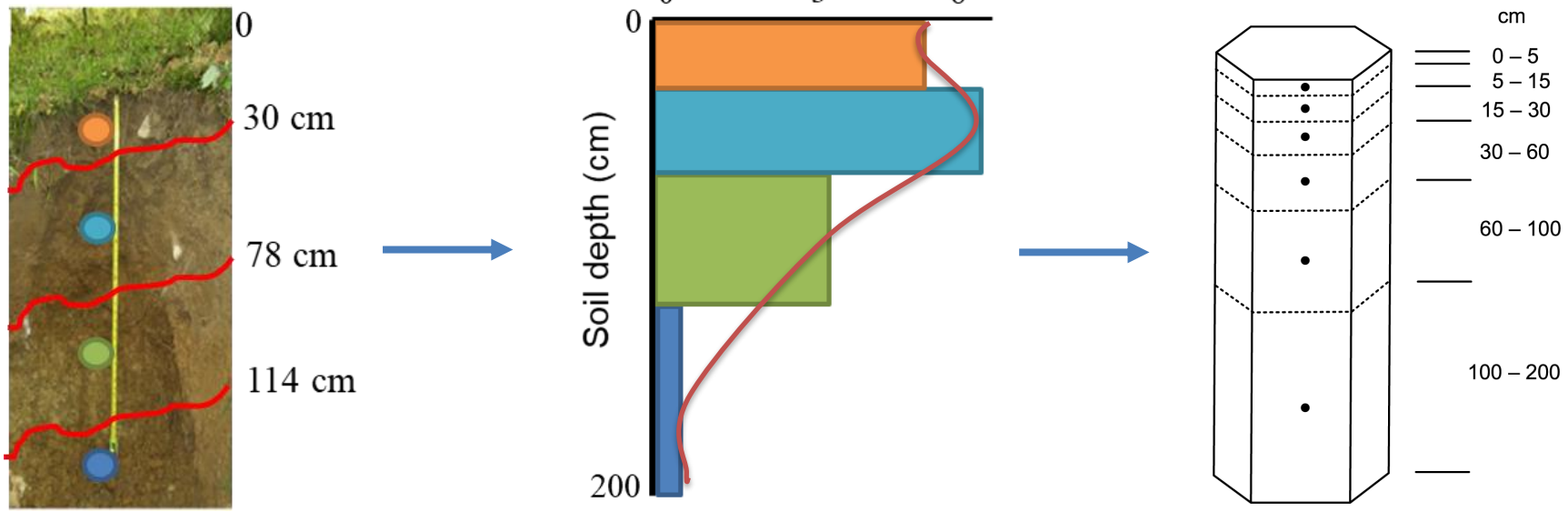
soil profile

Genetic horizon



❑ **Methods**: Soil depth function

$S = f(S, C, O, R, P, A, N) + e$



Soil profile

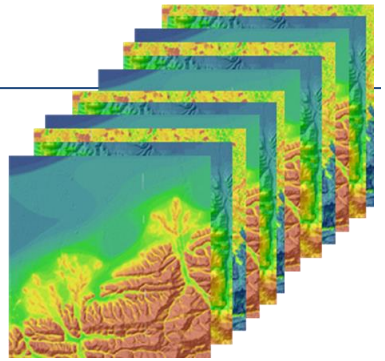
Soil depth function

Soil standard depth

❑ **Methods**: Environmental covariates

$$S = f(S, C, O, R, P, A, N) + e$$

covariates



Digital elevation model

Wetness index
MrVBF
Slope
Curvature

Remote sensing data

Landsat-8 images

Sentinel-2 images

RS indices such as NDVI

Feature Selection

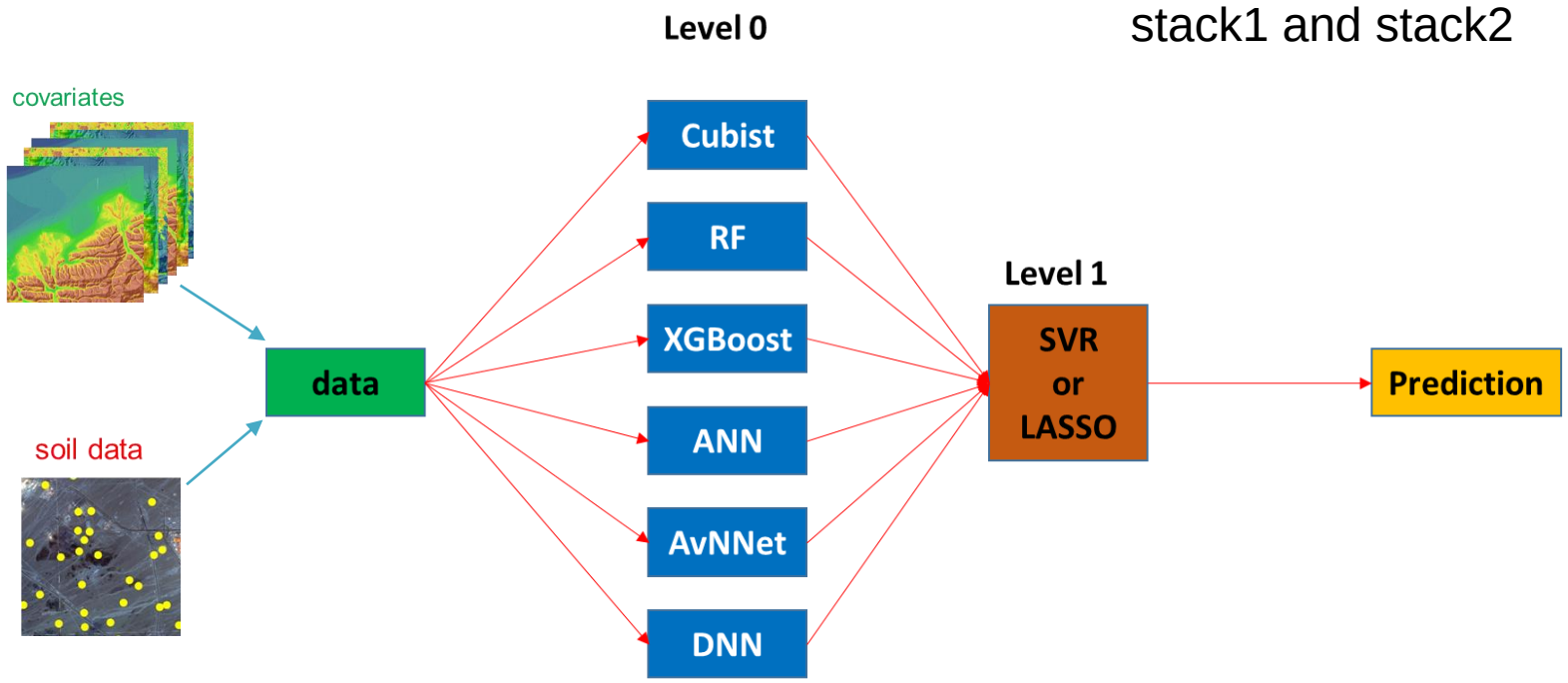
Boruta algorithm

	Terrain-based covariates
1	Elevation
2	Wetness Index
3	Catchments area
4	Catchment Slope
5	Multi-resolution Valley Bottom Flatness Index
6	Valley Depth
7	Plane Curvature
8	Profile Curvature
9	General Curvature
10	Total Insolation
	RS-based covariates
11	Blue band of Landsat-8 [0.482 µm]
12	Green band of Landsat-8 [0.561 µm]
13	Red band of Landsat-8 [0.654 µm]
14	Near infrared band of Landsat-8 [0.864 µm]
15	Shortwave IR-1 band of Landsat-8 [1.608 µm]
16	Shortwave IR-2 band of Landsat-8 [2.200 µm]
17	Blue band of Sentinel-2 [0.490 µm]
18	Green band of Sentinel-2 [0.560 µm]
19	Red band of Sentinel-2 [0.665 µm]
20	Vegetation Red Edge of Sentinel-2 [0.705 µm]
21	Vegetation Red Edge of Sentinel-2 [0.740 µm]
22	Vegetation Red Edge of Sentinel-2 [0.783 µm]
23	Near infrared band of Sentinel-2 [0.842 µm]
24	Vegetation Red Edge of Sentinel-2 [0.865 µm]
25	Shortwave IR-1 band of Sentinel-2 [1.610 µm]
26	Shortwave IR-2 band of Sentinel-2 [2.190 µm]
27	Normalized difference vegetation index (Landsat-8 based)
28	Normalized difference vegetation index (Sentinel-2 based)

Google Earth Engine
ArcGIS
SAGA

❑ **Methods**: Machine learning

$$S = f(S, C, O, R, P, A, N) + e$$

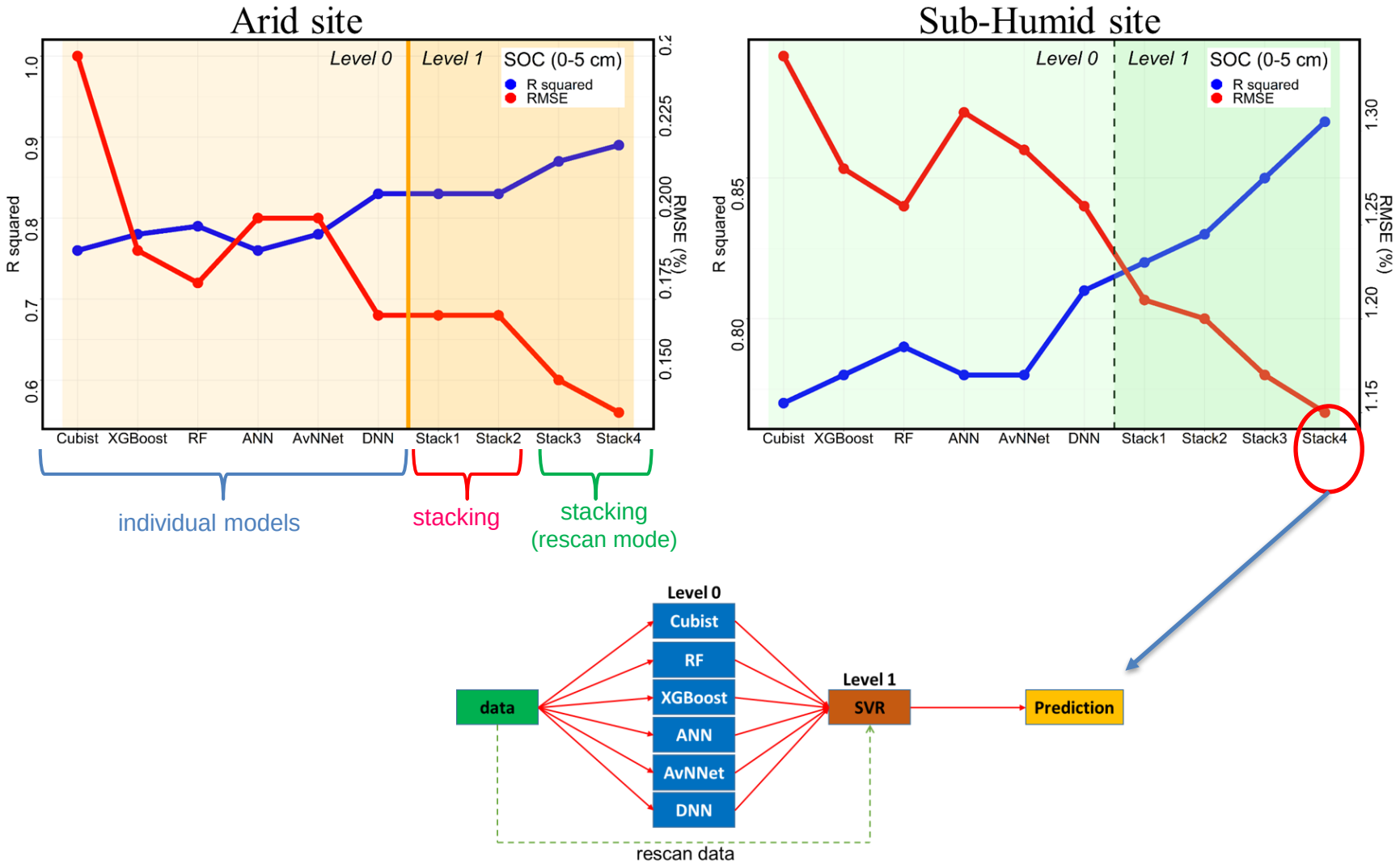


General framework of stacking approaches used in this study.

Cubist, Random Forests (RF), extreme gradient boosting (XGBoost), classical artificial neural network models (ANN), neural network ensemble based on model averaging (AvNNet), deep learning neural networks (DNN), least absolute shrinkage and selection operator (LASSO), support vector regression (SVR)

❑ **Results**: Machine learning

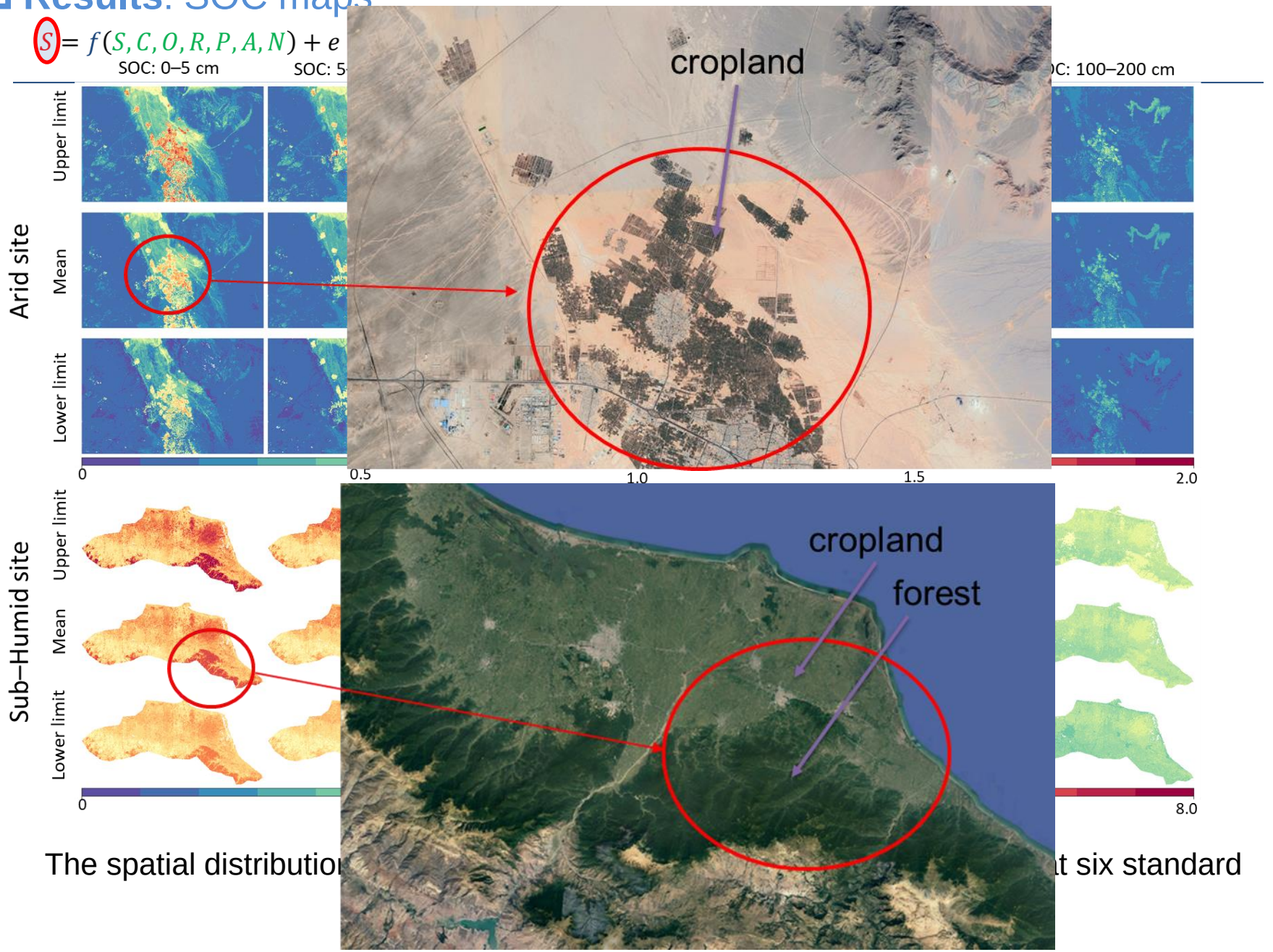
$$S = f(S, C, O, R, P, A, N) + e$$



Coefficient of determination (R2) and RMSE values of the individual models (Level 0) and stacking models (Level 1) for SOC content at the first standard depth in two regions

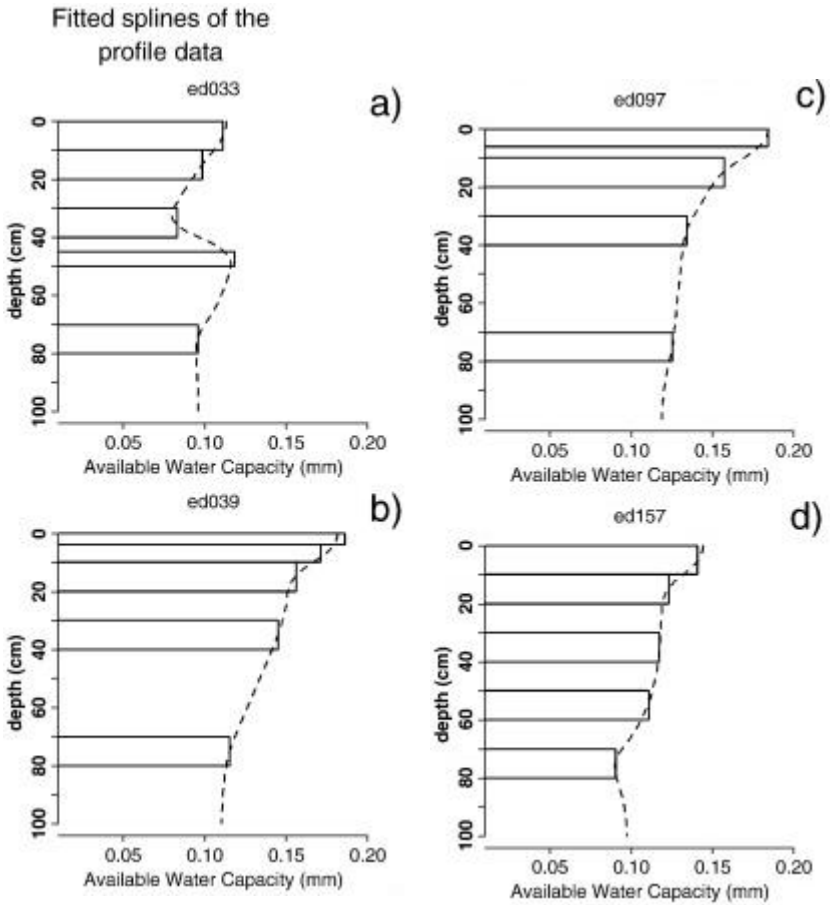
❑ **Results**: SOC maps

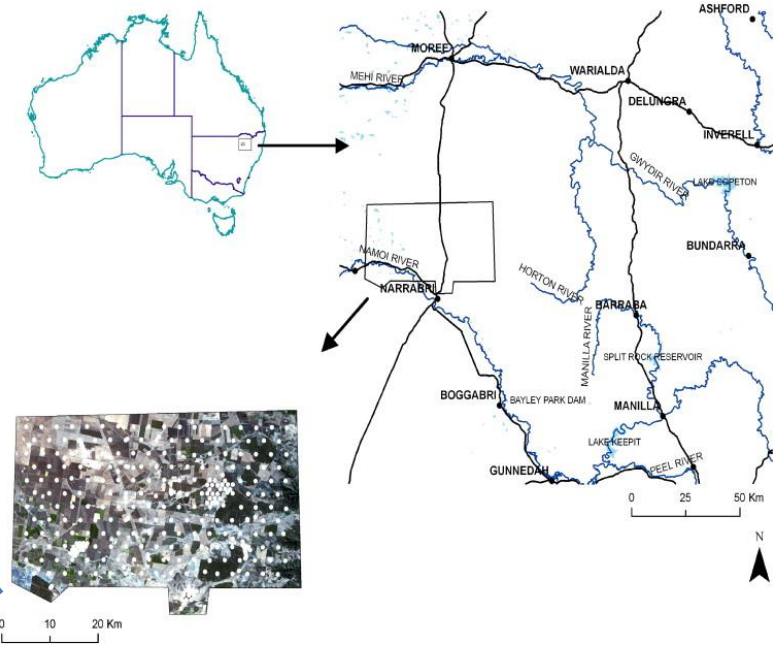
$$S = f(S, C, O, R, P, A, N) + e$$



The spatial distributio t six standard

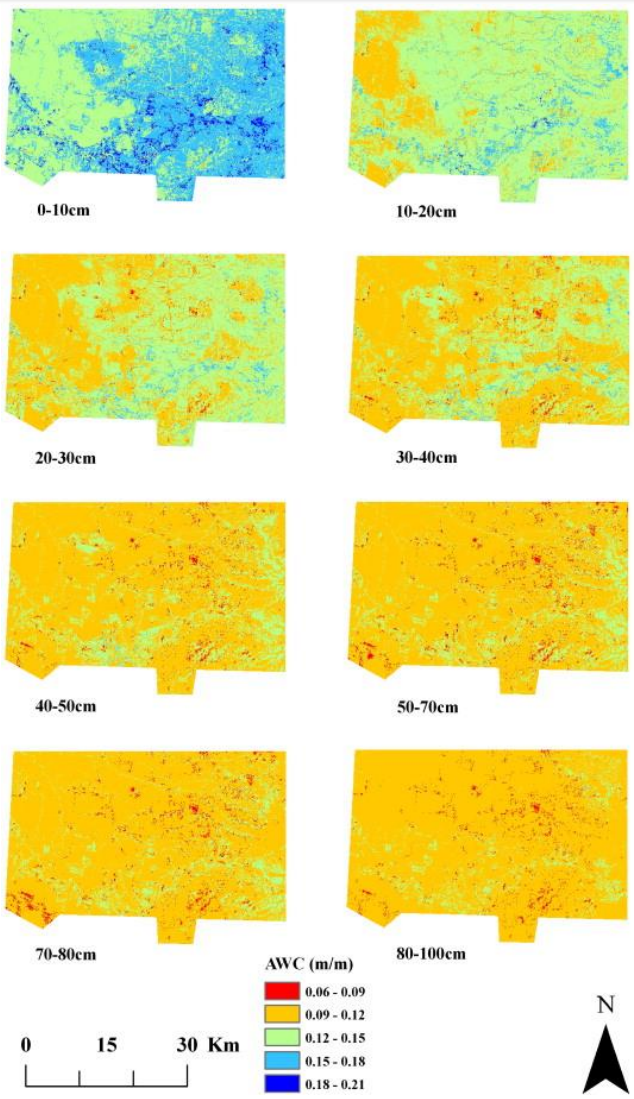
Soil Depth Functions → 3D Maps

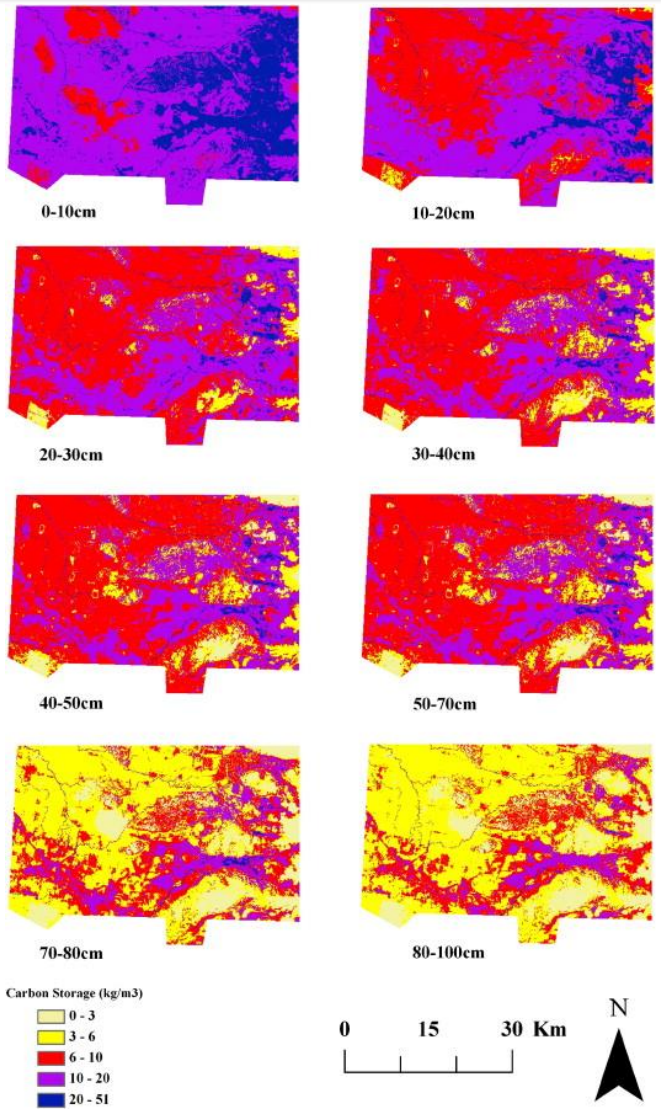




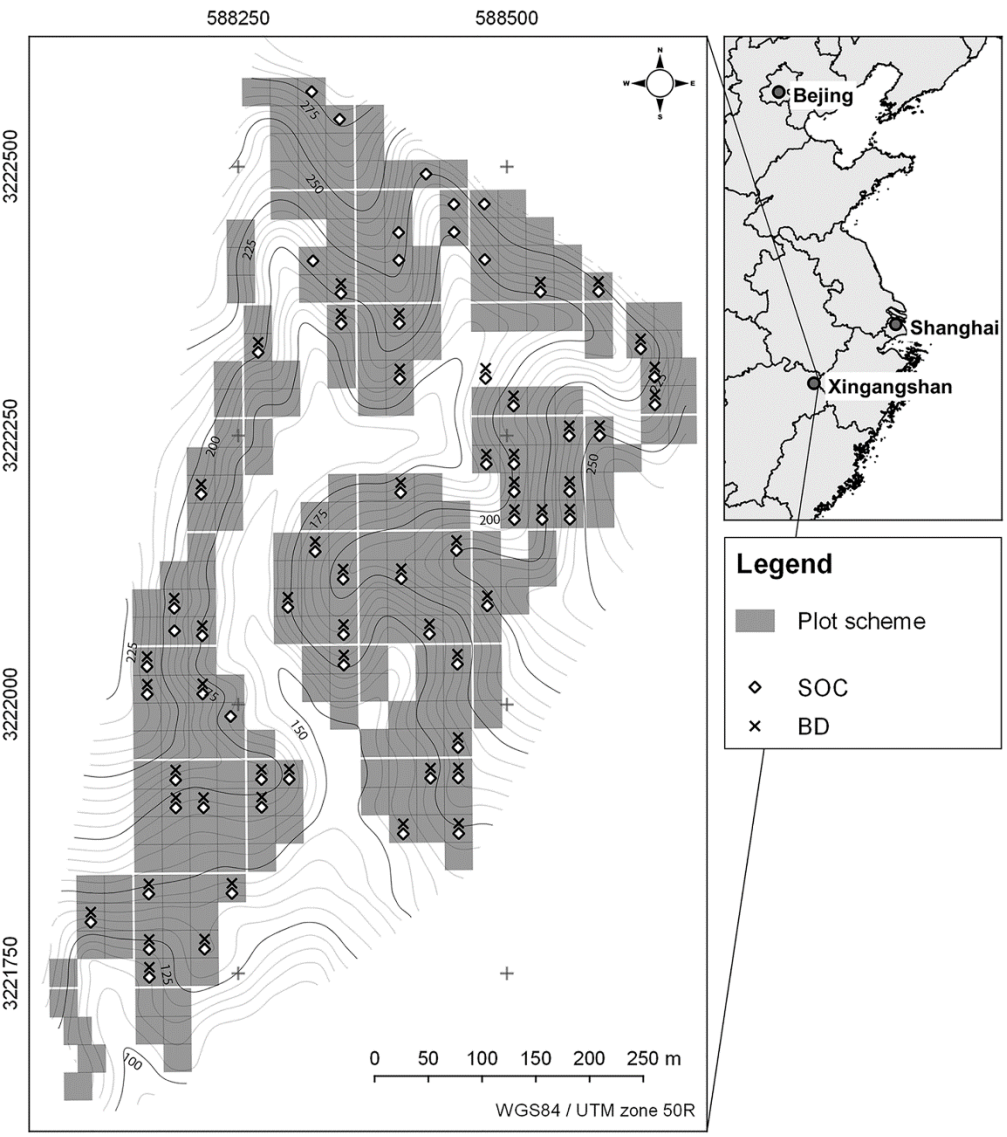
Malone et al., 2012

Soil Depth Functions → 3D Maps





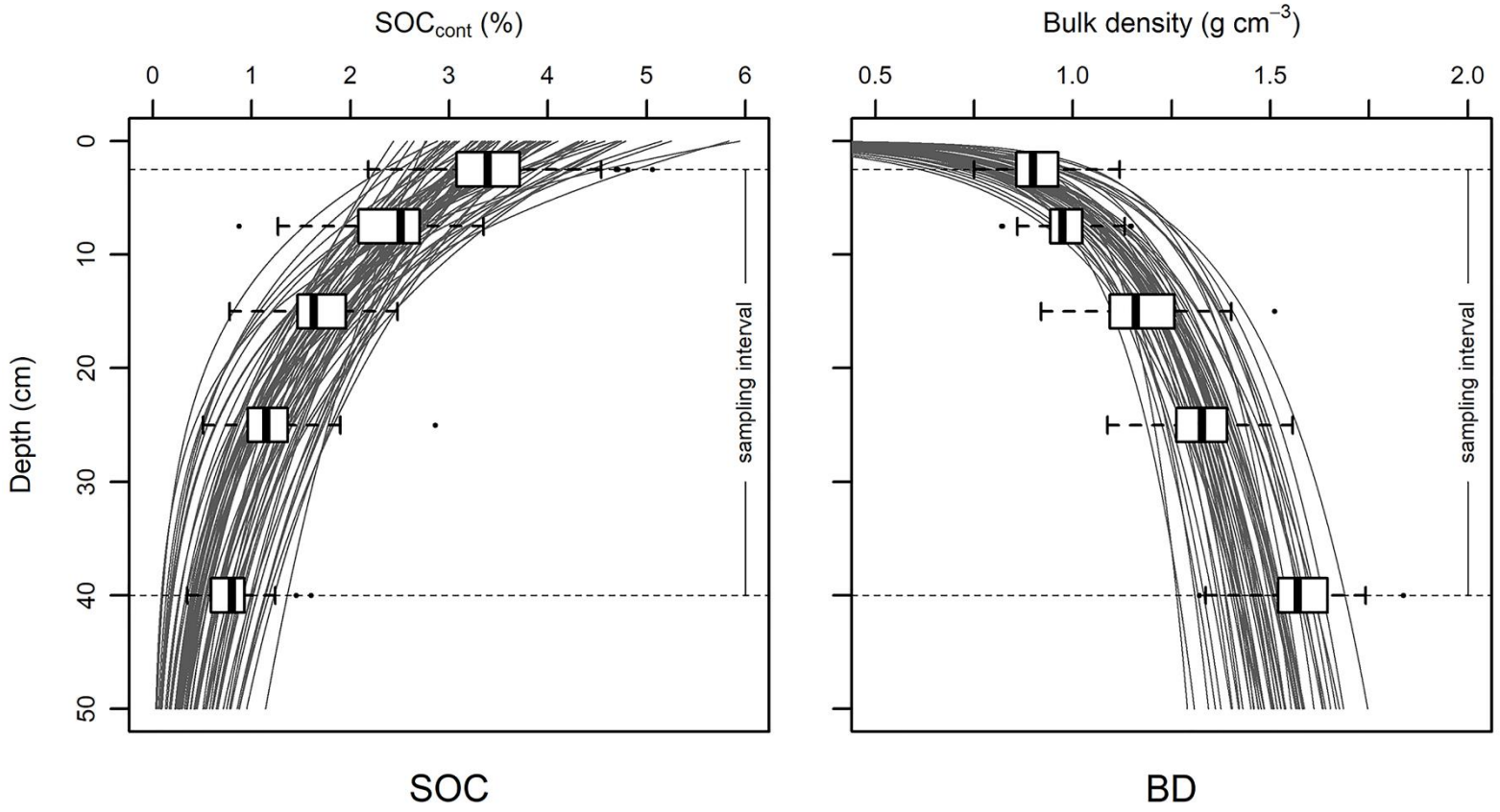
Soil Depth Functions → 3D Maps



Study area in mainland China and indication of sampled plots.

Rentschler et al., 2019

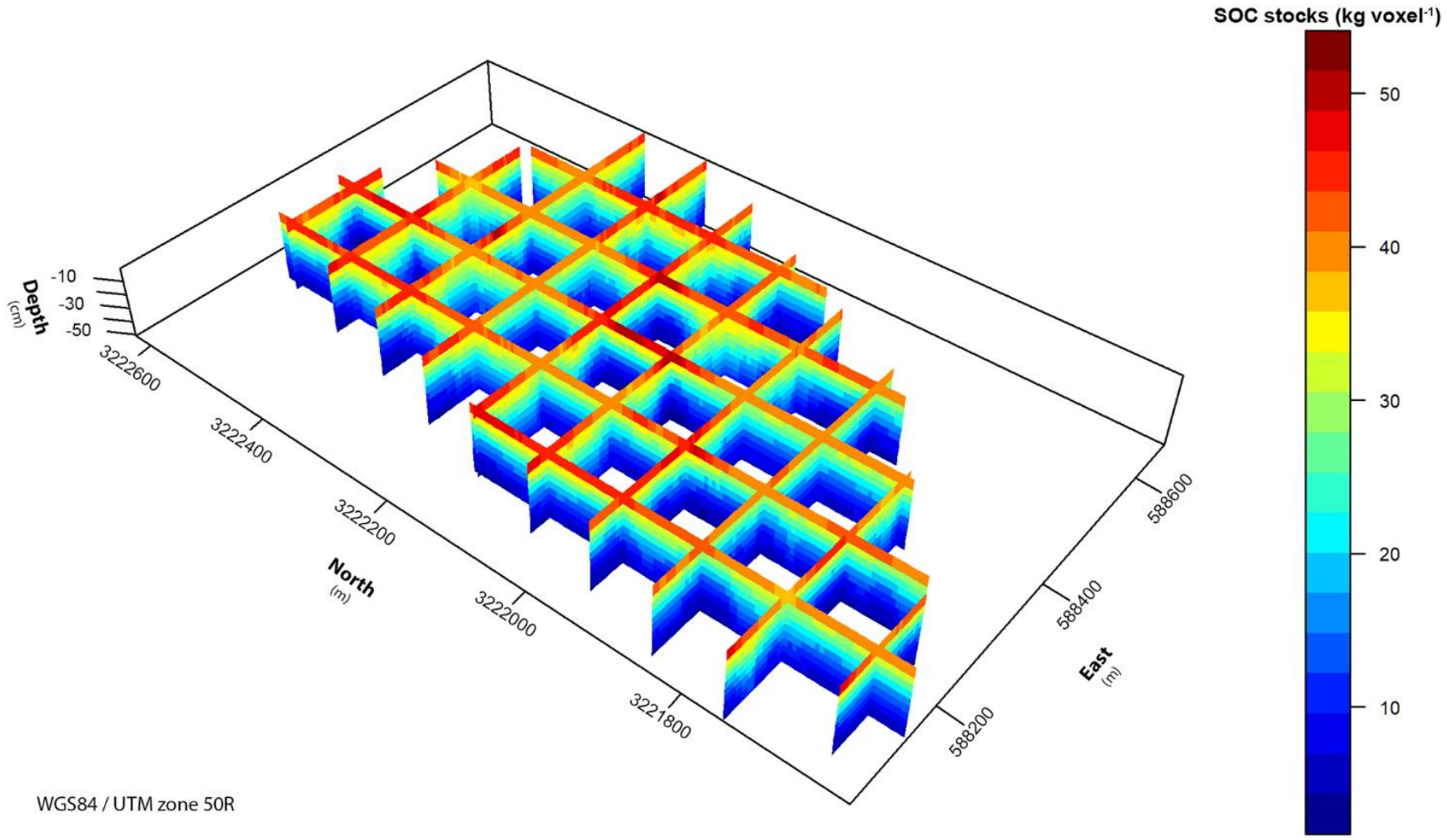
Soil Depth Functions → 3D Maps



Datasets for SOC and BD.

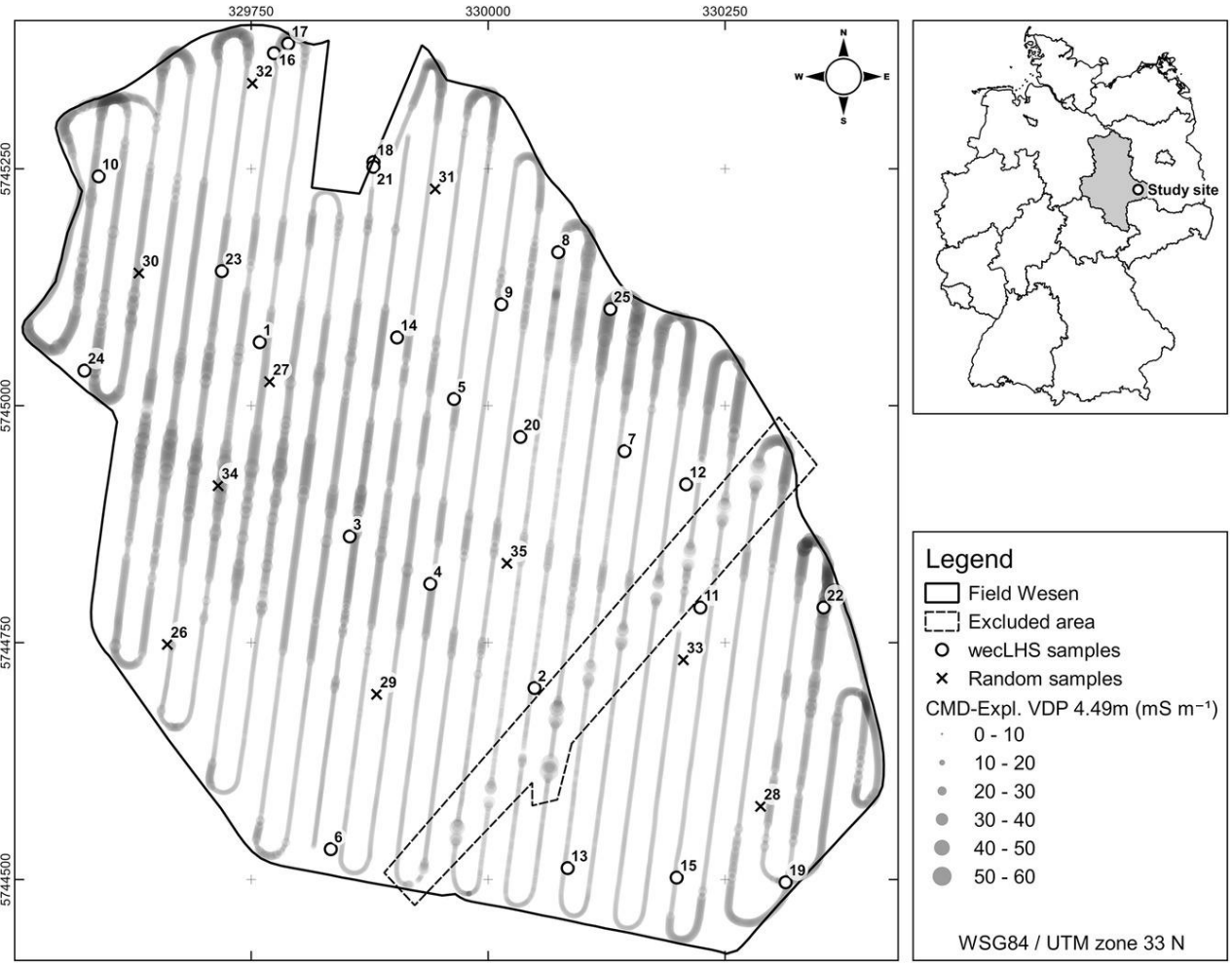
The boxplots show the variation of the SOC and BD values for each depth increment. SOC and BD samples were taken in five depth increments and 9 cores per plot were bulked. The grey lines show model depth functions (3rd degree polynomial for SOC and natural logarithmic function for BD

Soil Depth Functions → 3D Maps



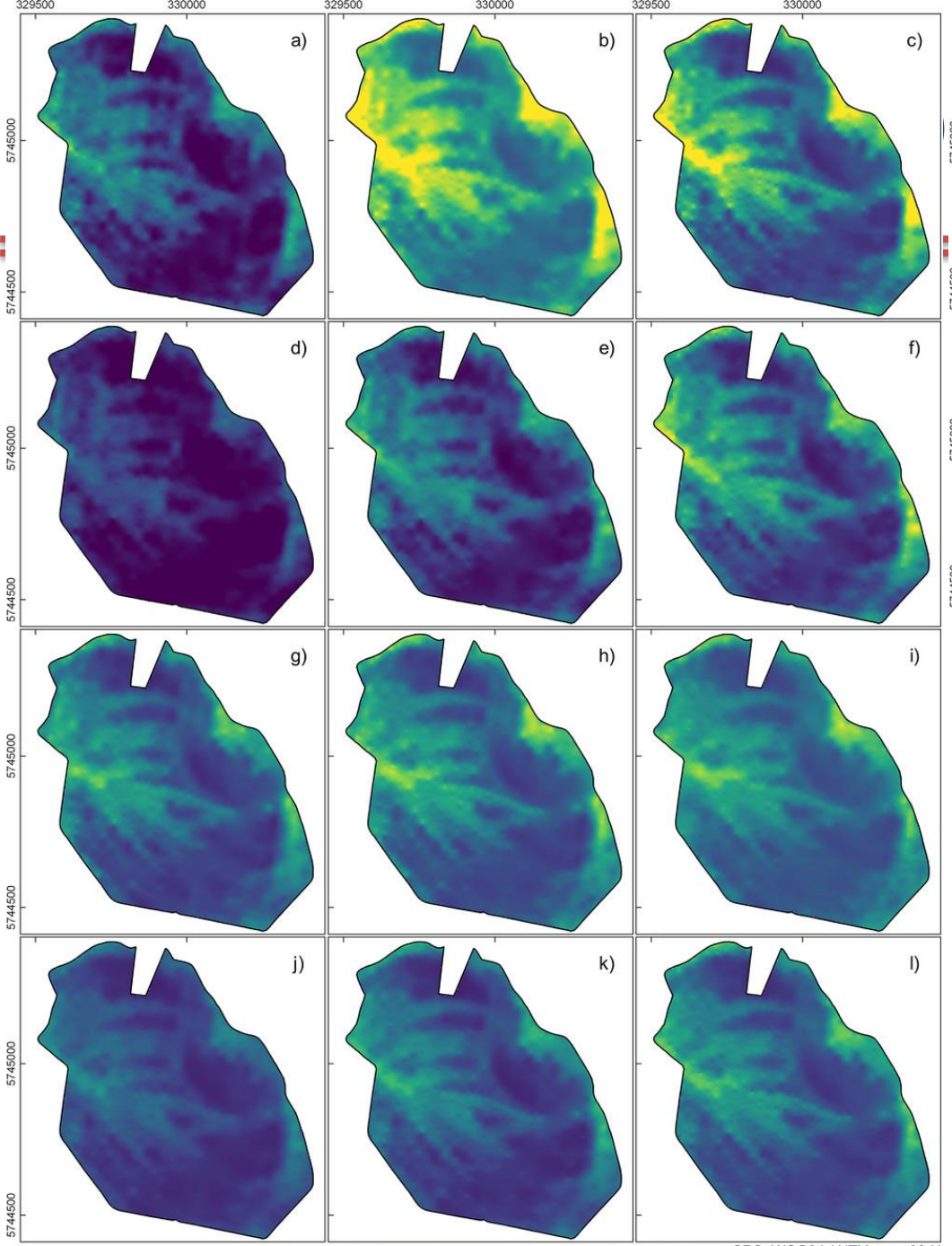
Three-dimensional prediction of SOC stocks for the whole catchment

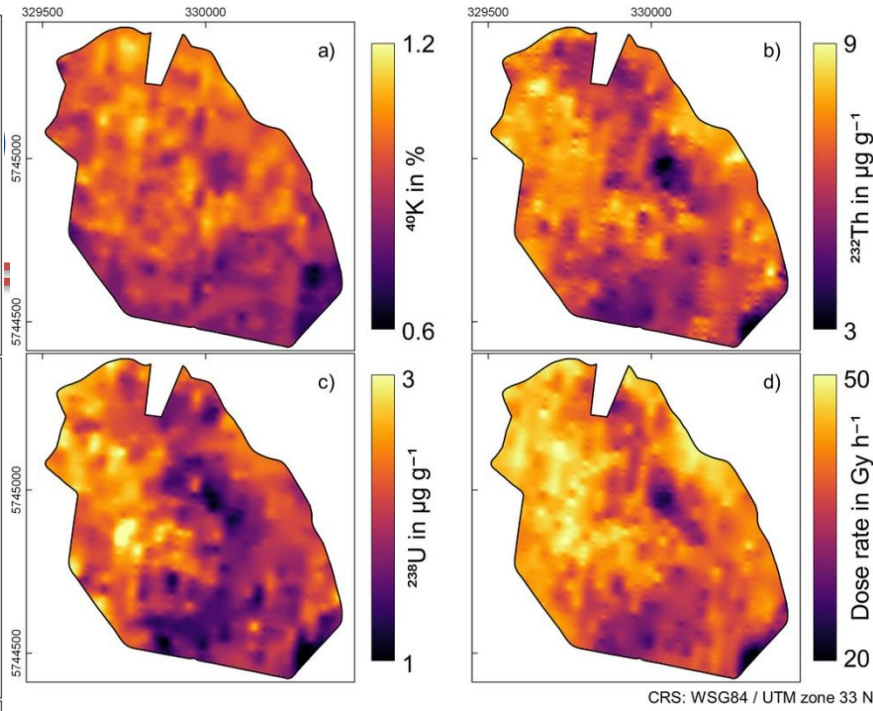
Soil Depth Functions → 3D Maps



Location of the field Wesen near Selbitz, Saxony-Anhalt, Germany, sampling scheme of the geophysical measurements with electromagnetic induction (EMI) and gamma-ray spectrometry, and sampled soil profiles

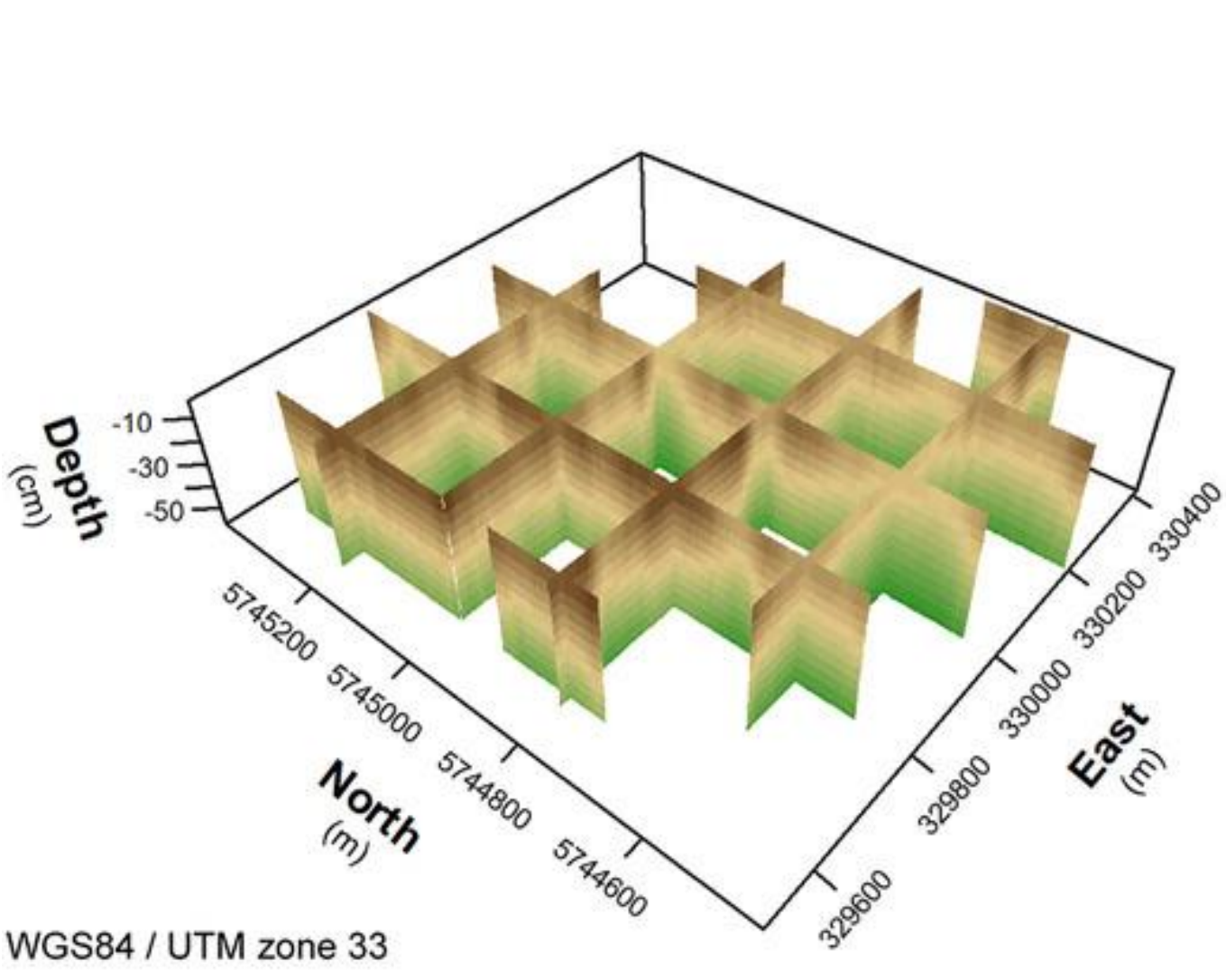
Rentschler et al., 2020

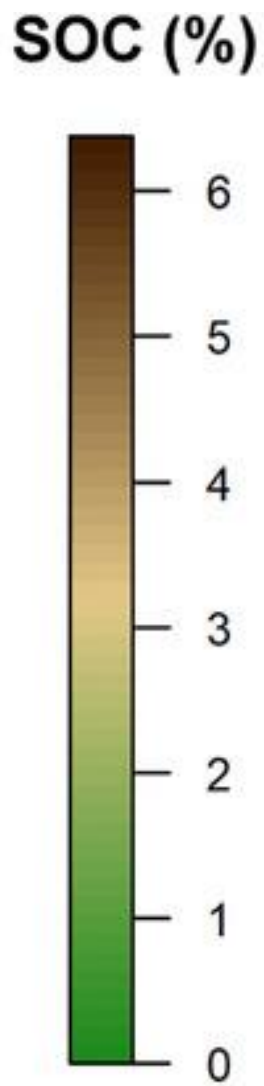




The spatial distribution of proximal soil sensors

Soil Depth Functions → 3D Maps

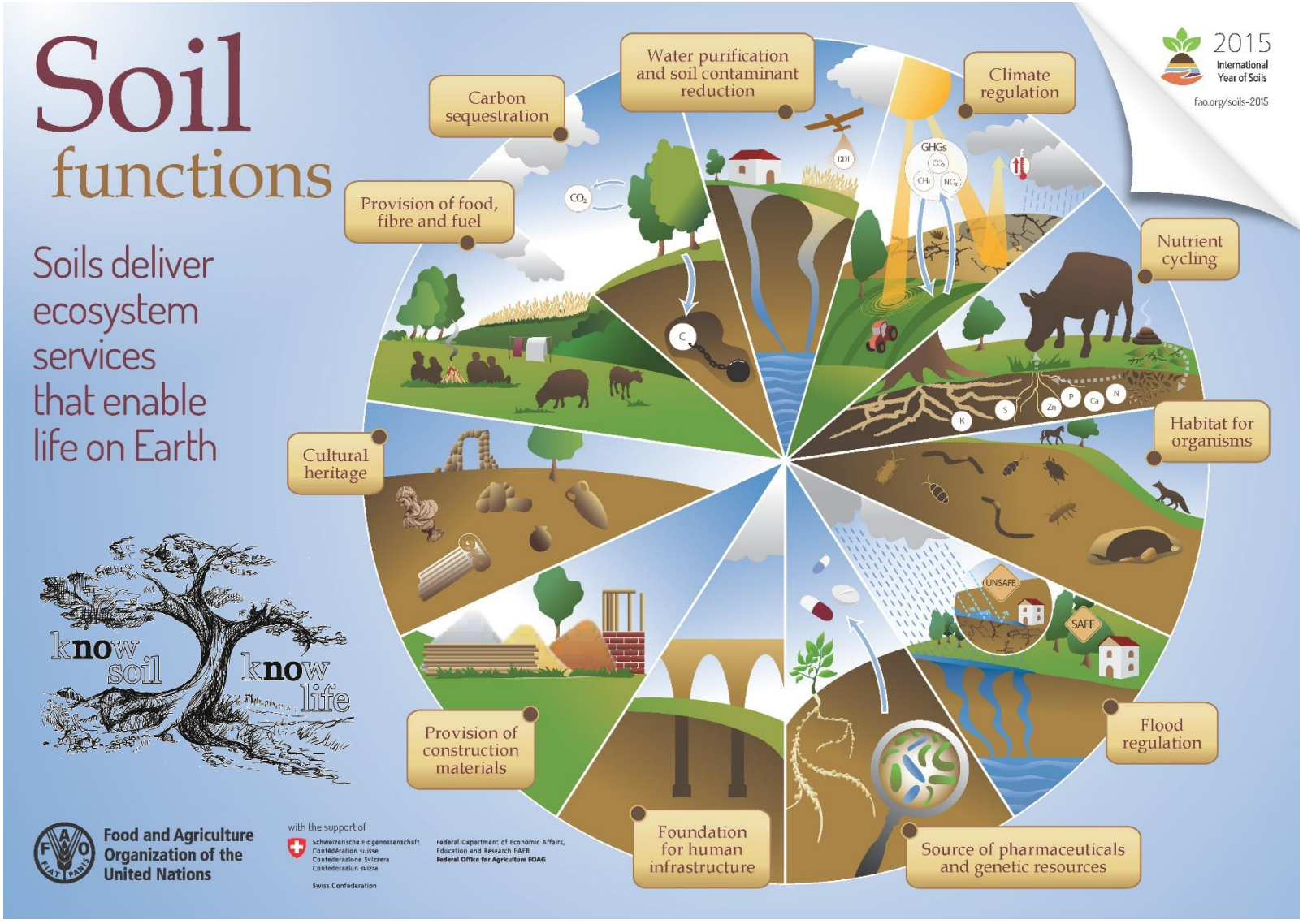




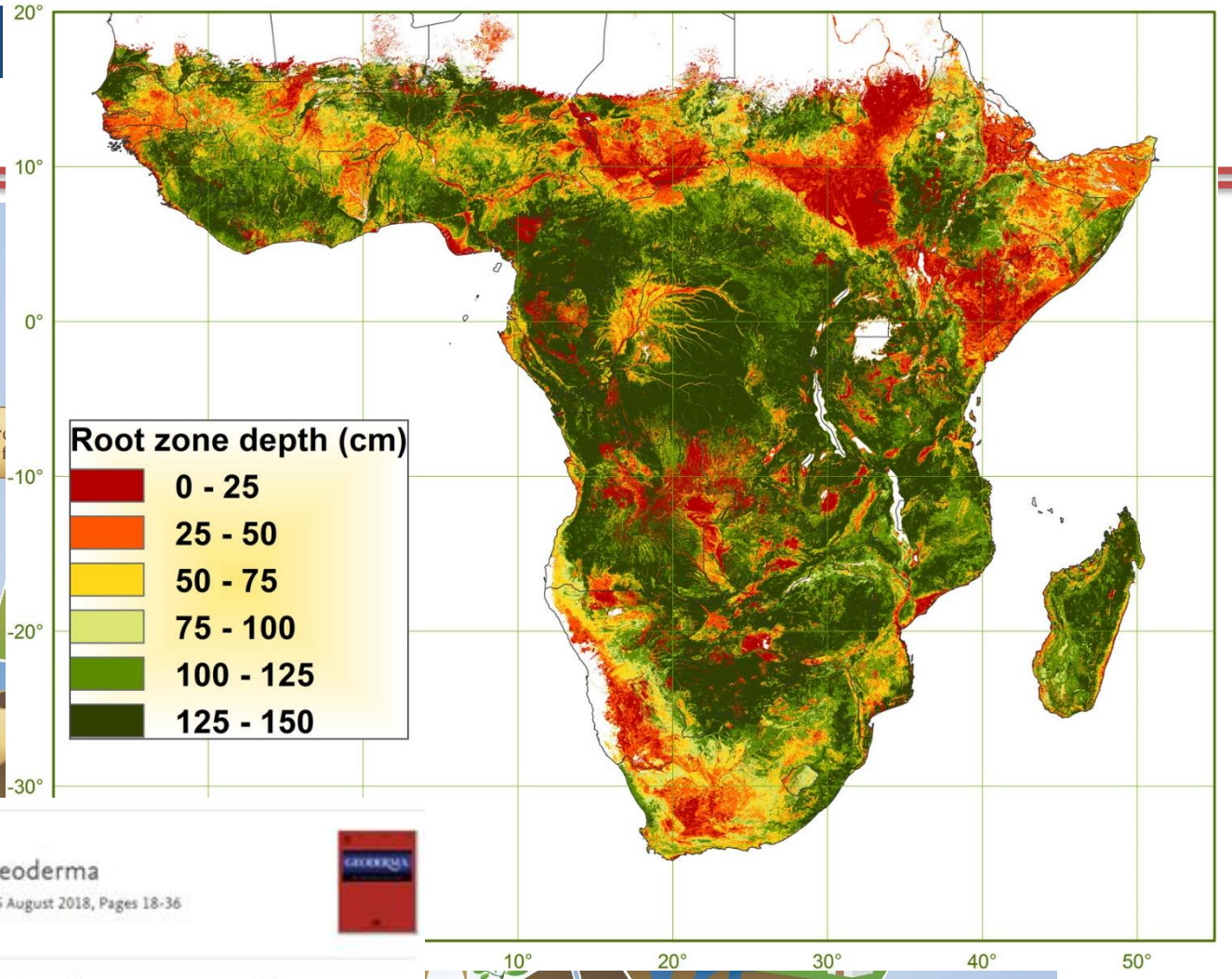
Take-Home Message

- ML has great potential for spatial and spatio-temporal predictions
- Each model predicts one realization of a DSM
- R allows for easy model training
- Accuracy assessment is important
- Avoid overfitting by careful variable selection
- Soil depth functions + ML = 3D soil mapping of soil properties

Extend Soil Mapping to Soil Functions



Extend Soil



Soil functions

Soils deliver ecosystem services that enable life on Earth

Cultural heritage

ELSEVIER

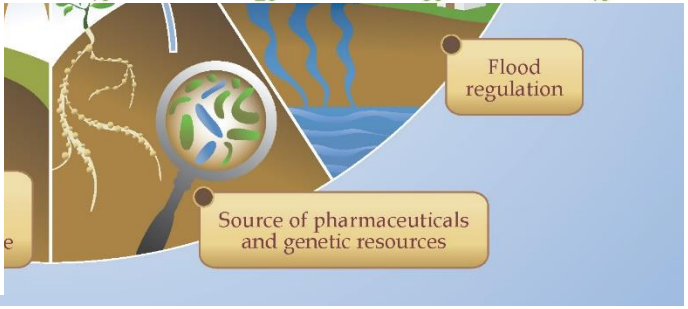
Geoderma

Volume 324, 15 August 2018, Pages 18-36

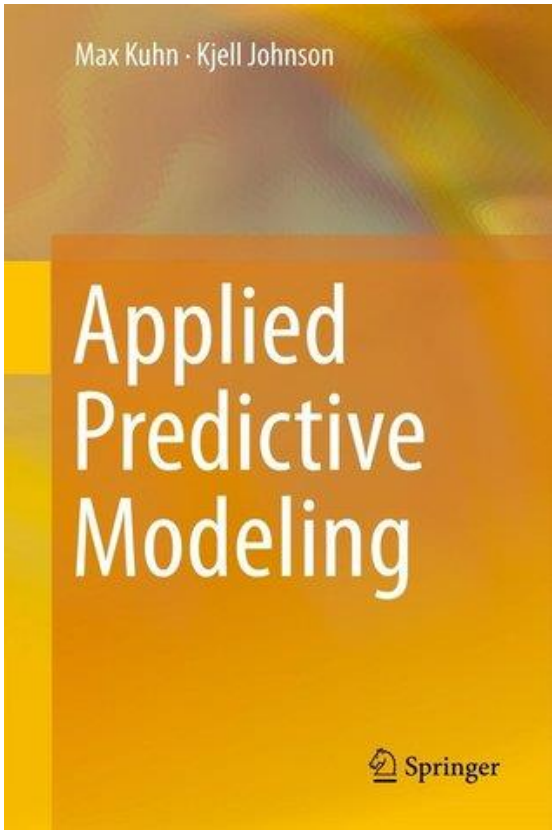
Mapping rootable depth and root zone plant-available water holding capacity of the soil of sub-Saharan Africa

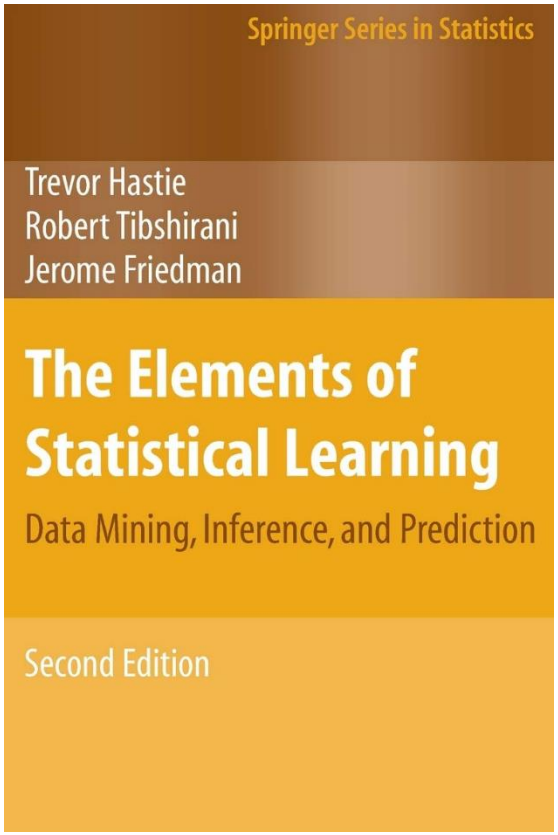
Johan G.B. Leenaars, Lieven Claessens, Gerard B.M. Heuvelink, Tom Hengl, María Ruiperez González, Lenny G.J. van Bussel, Nicolas Guilpart, Haishun Yang, Kenneth G. Cassman

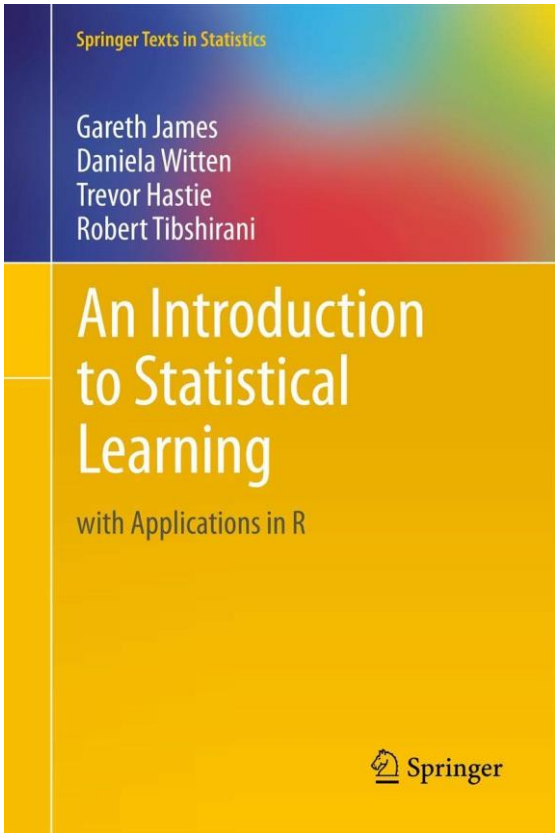
Show more



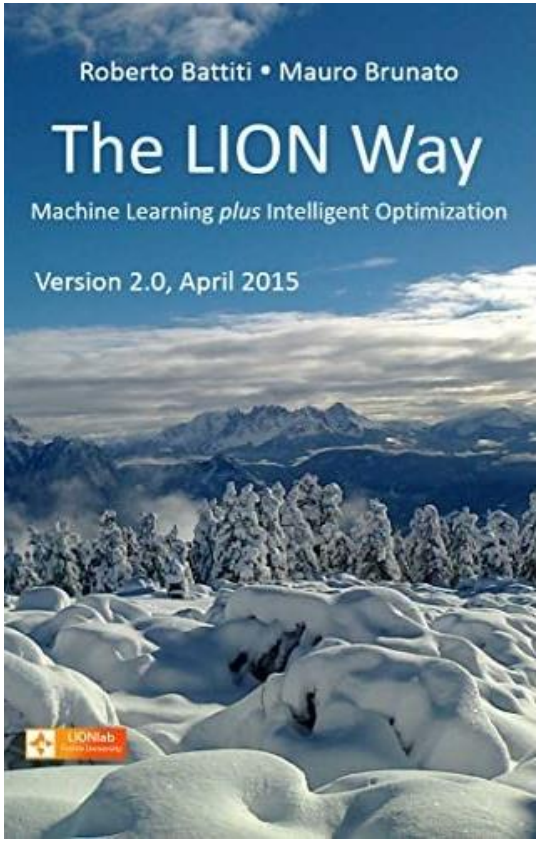
Useful Resources

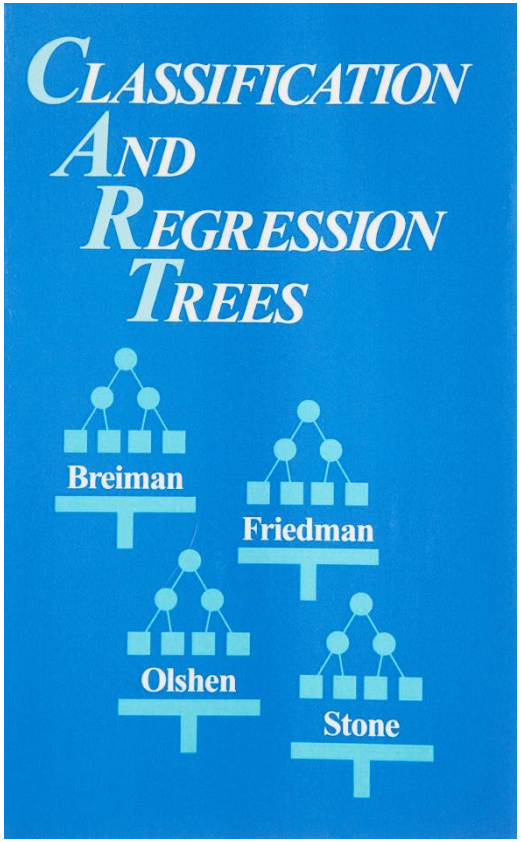






Useful Resources





Hothorn, Torsten. CRAN Task View: Machine Learning & Statistical Learning
https://CRAN.R-project.org/view=MachineLearning, 2020.