Landslide Susceptibility Modeling with R

Alexander Brenning
Department of Geography, Friedrich Schiller University Jena

Romanian Association of Geomorphologists
Workshop at AGR 2024, 19 Sep 2024, Iasi

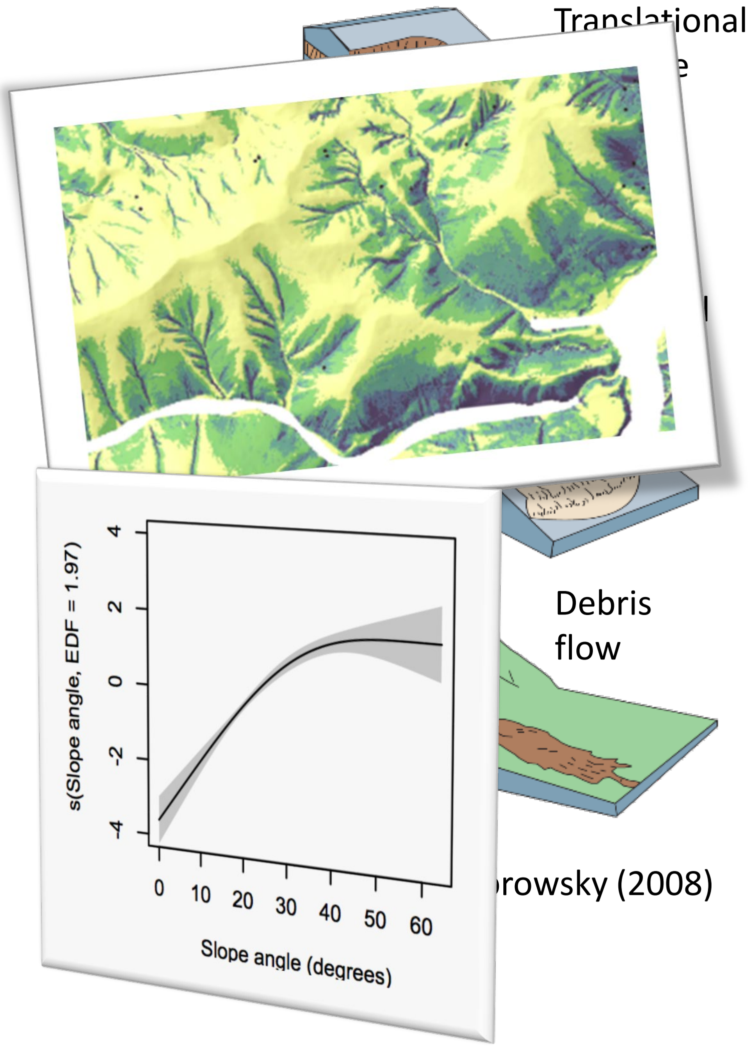
Modeling Landslides: What, Why?

Risk = Hazard x Exposure x Vulnerability

- Regional-scale modelling of susceptibility

Main scenarios:

- **Susceptibility mapping (=predictive modeling):** Identify hazardous areas for spatial planning
- **Explanation and attribution (=estimation and inference):** Identify and quantify preparatory and triggering factors



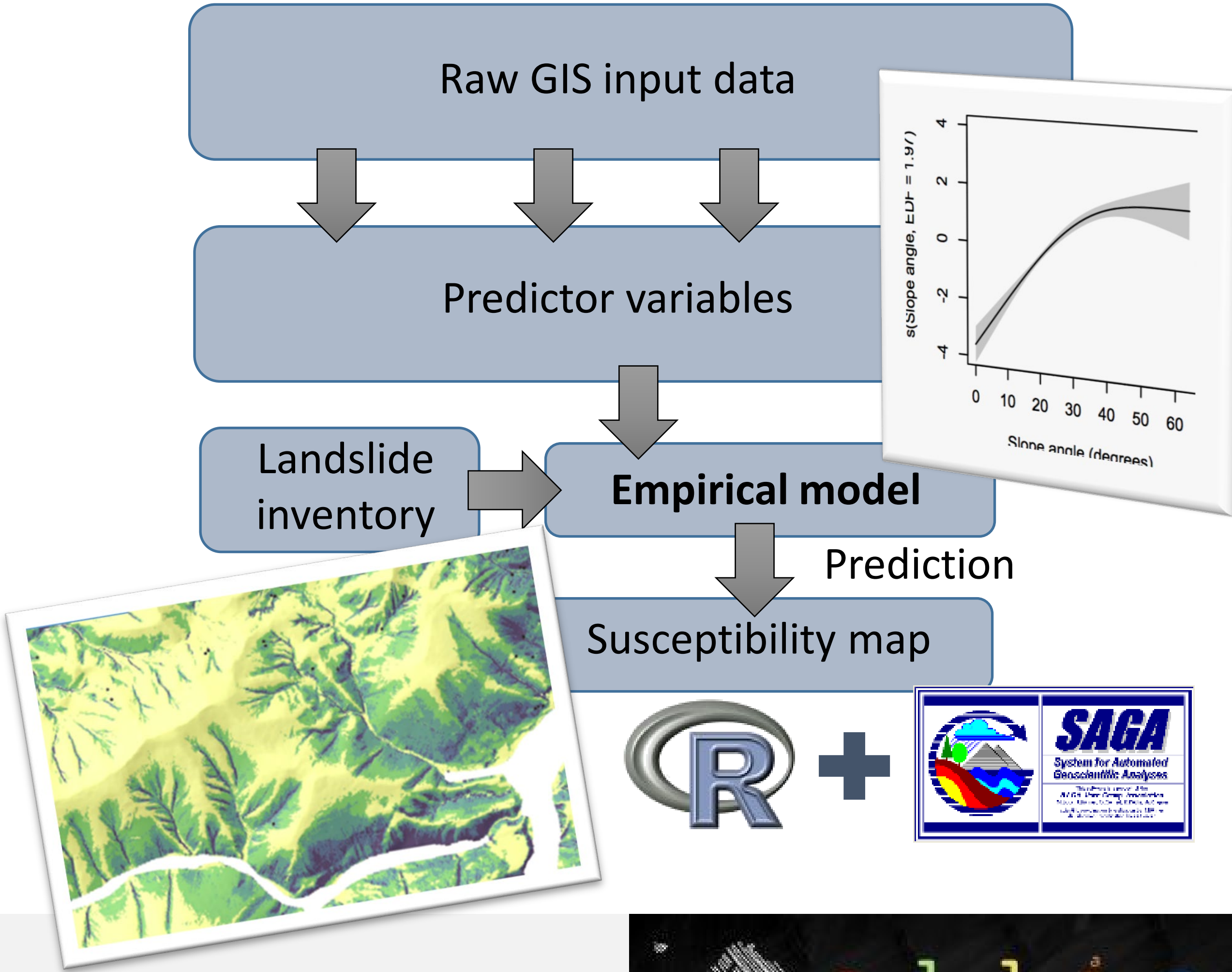
Modeling Landslides: What, Why?

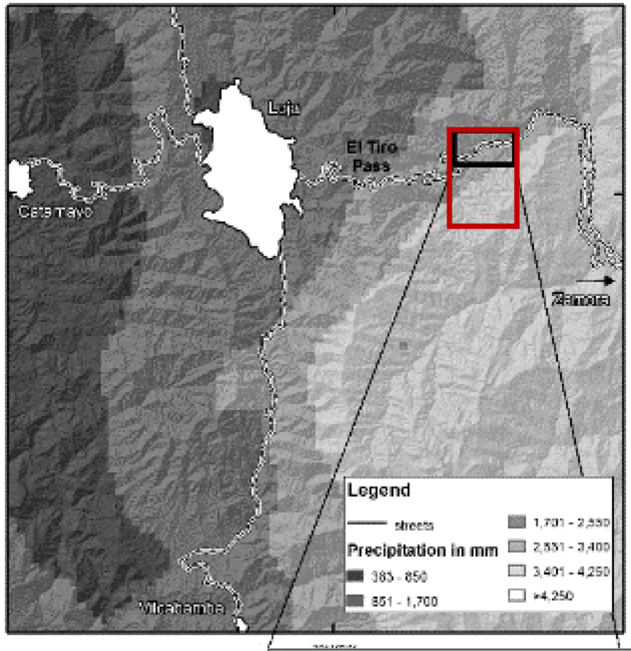
Model structure:

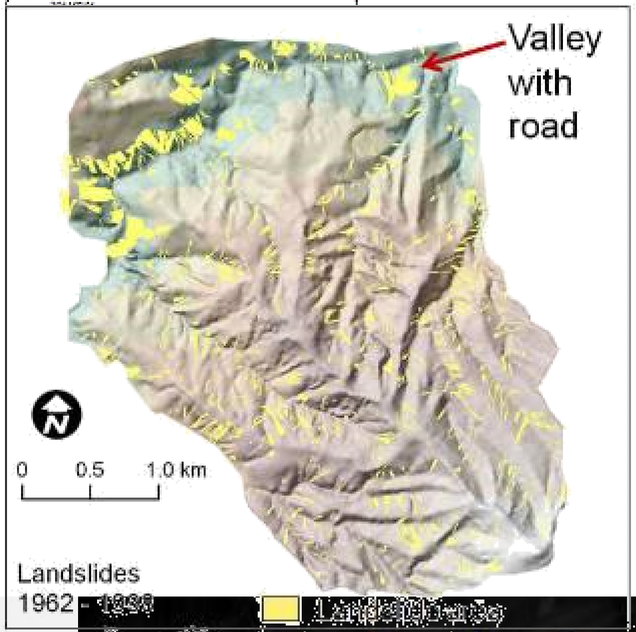
- **Response variable**: Landslide inventory
- **Predictor variables**:
    - Preparatory factors
    - Possibly triggering factors

Model types:

- Process-based / physically based
- Empirical: statistical
- Empirical: black box
- Hybrid







- Step-by-step tutorials: **RSAGA** package vignette and **GeocompR** book Chapter 12 ; data: R. Stoyan / M. Richter

Machine-Learning Modeling

$\mathbf{x}$

$\hat{f}$

$\hat{y}$

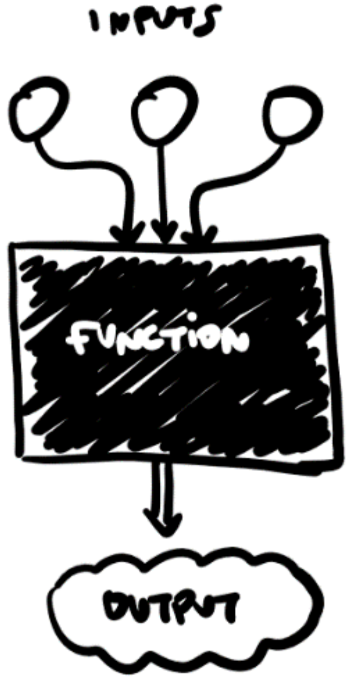
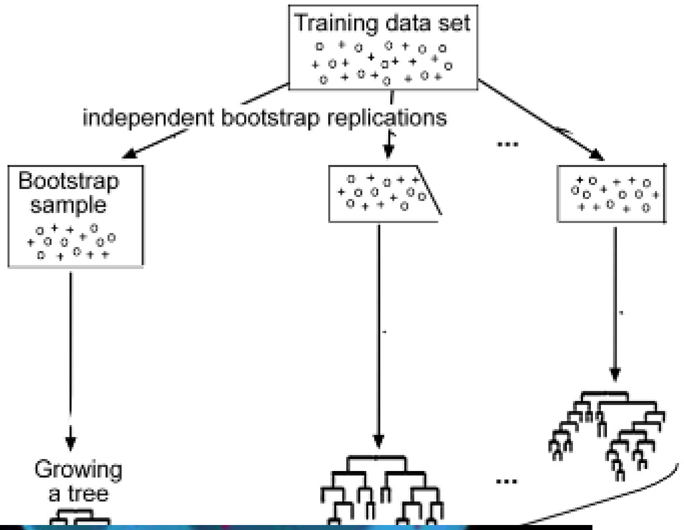
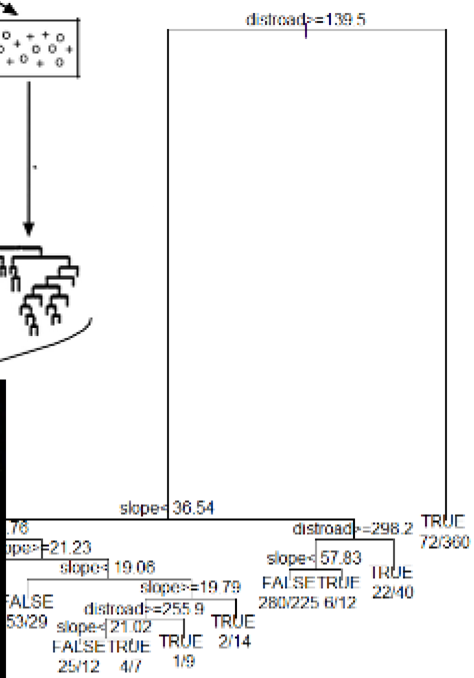
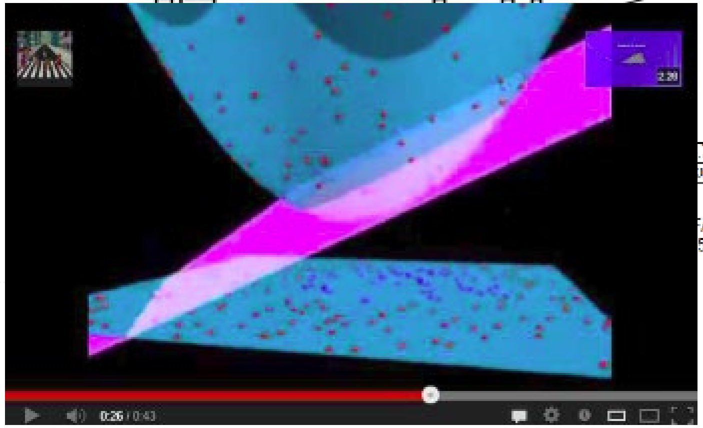
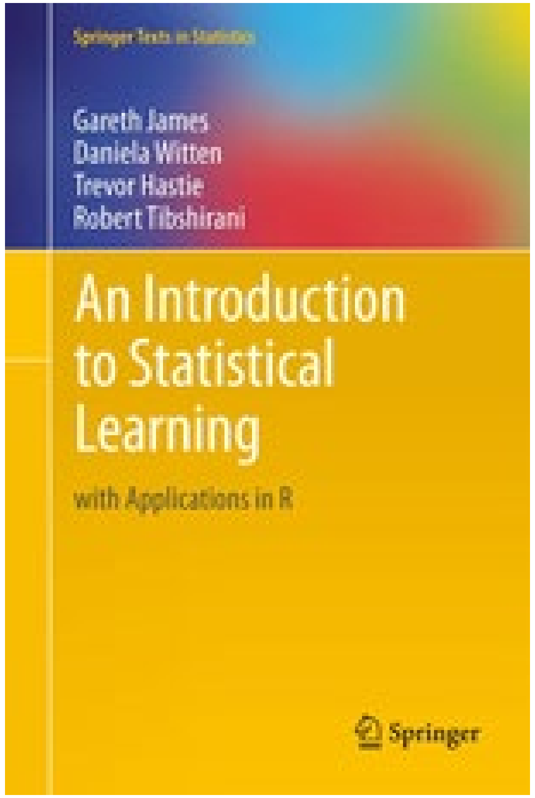


Image source: thatsoftwaredude.com

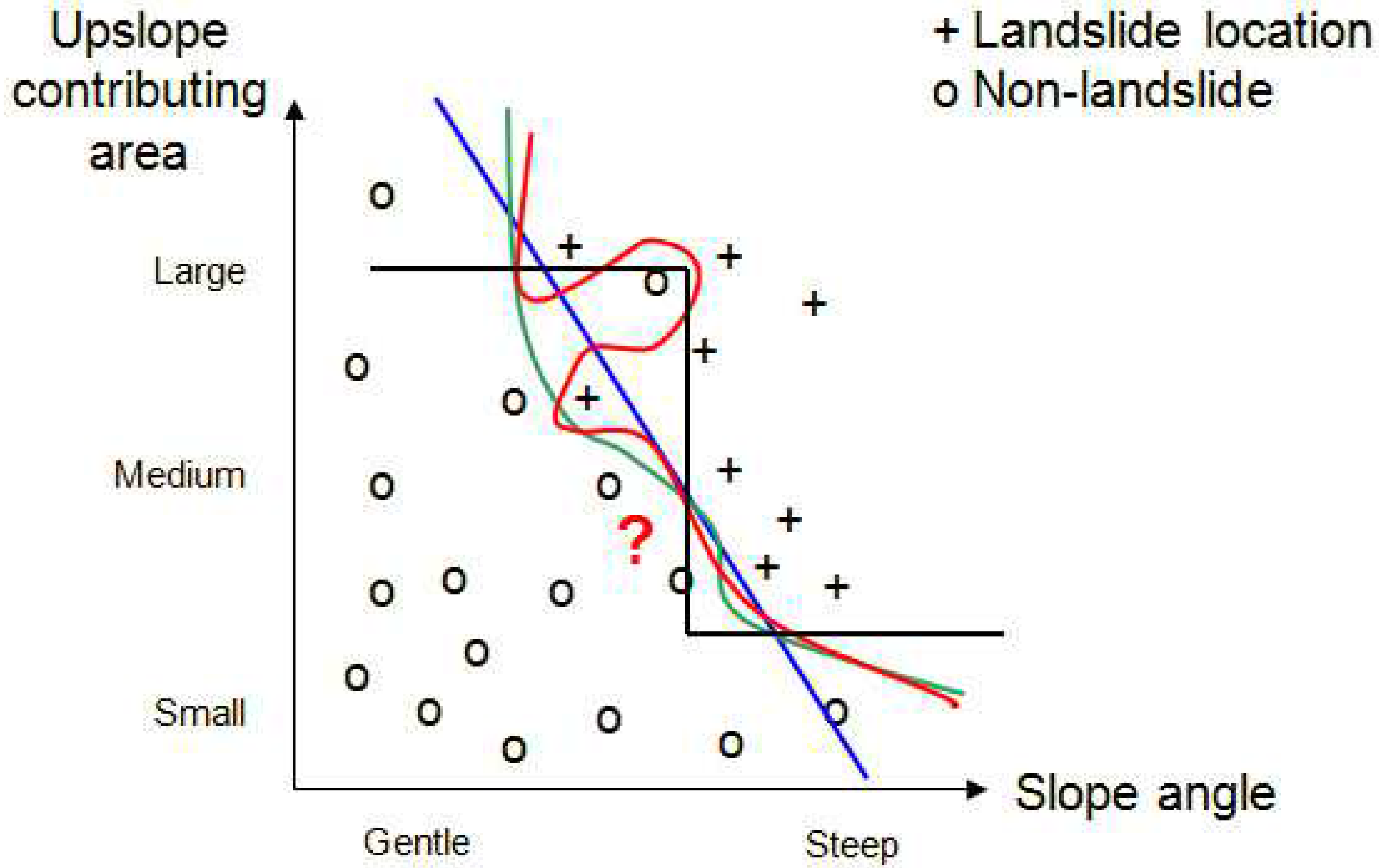






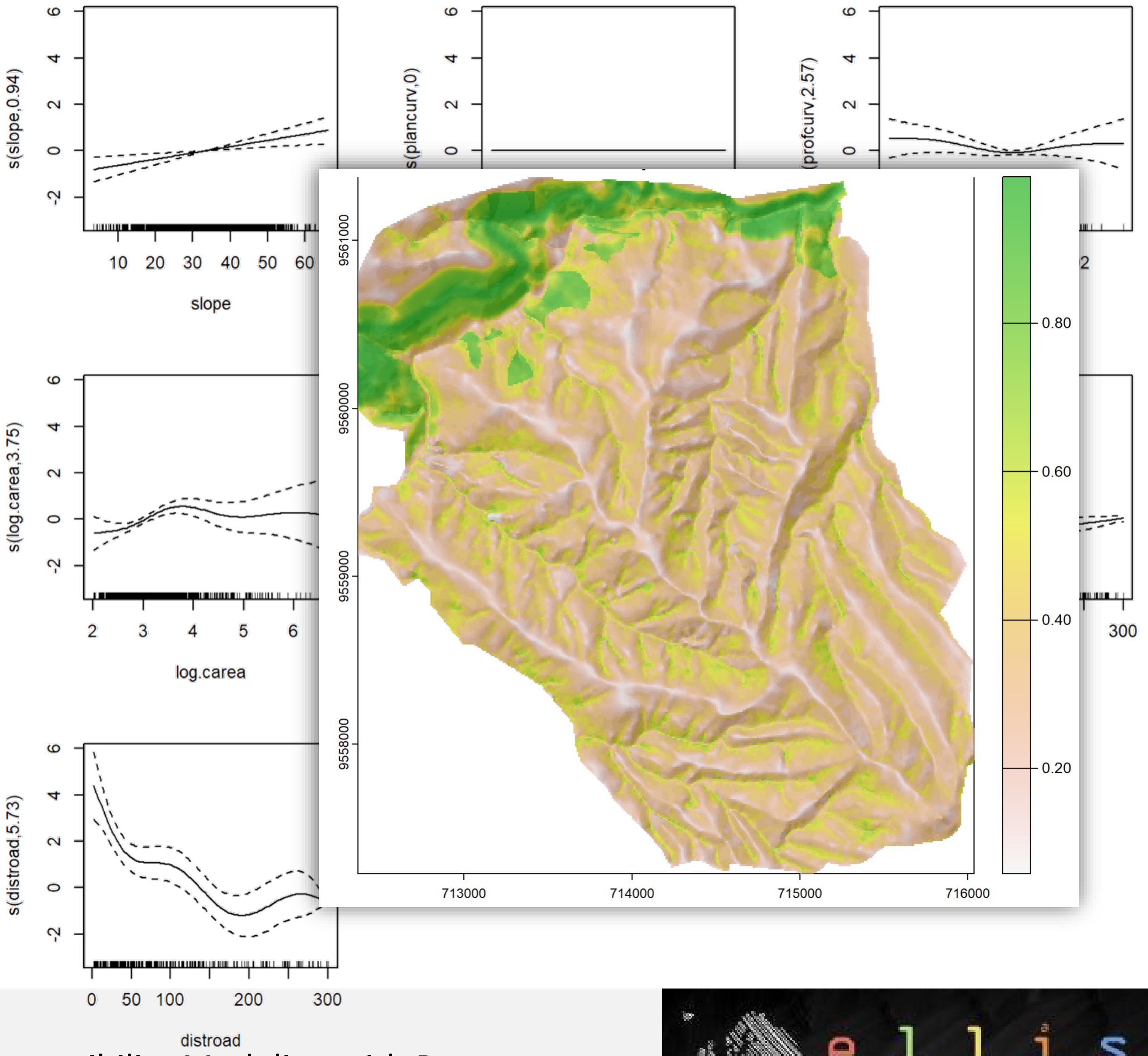


How Classifiers Work (Basically...)



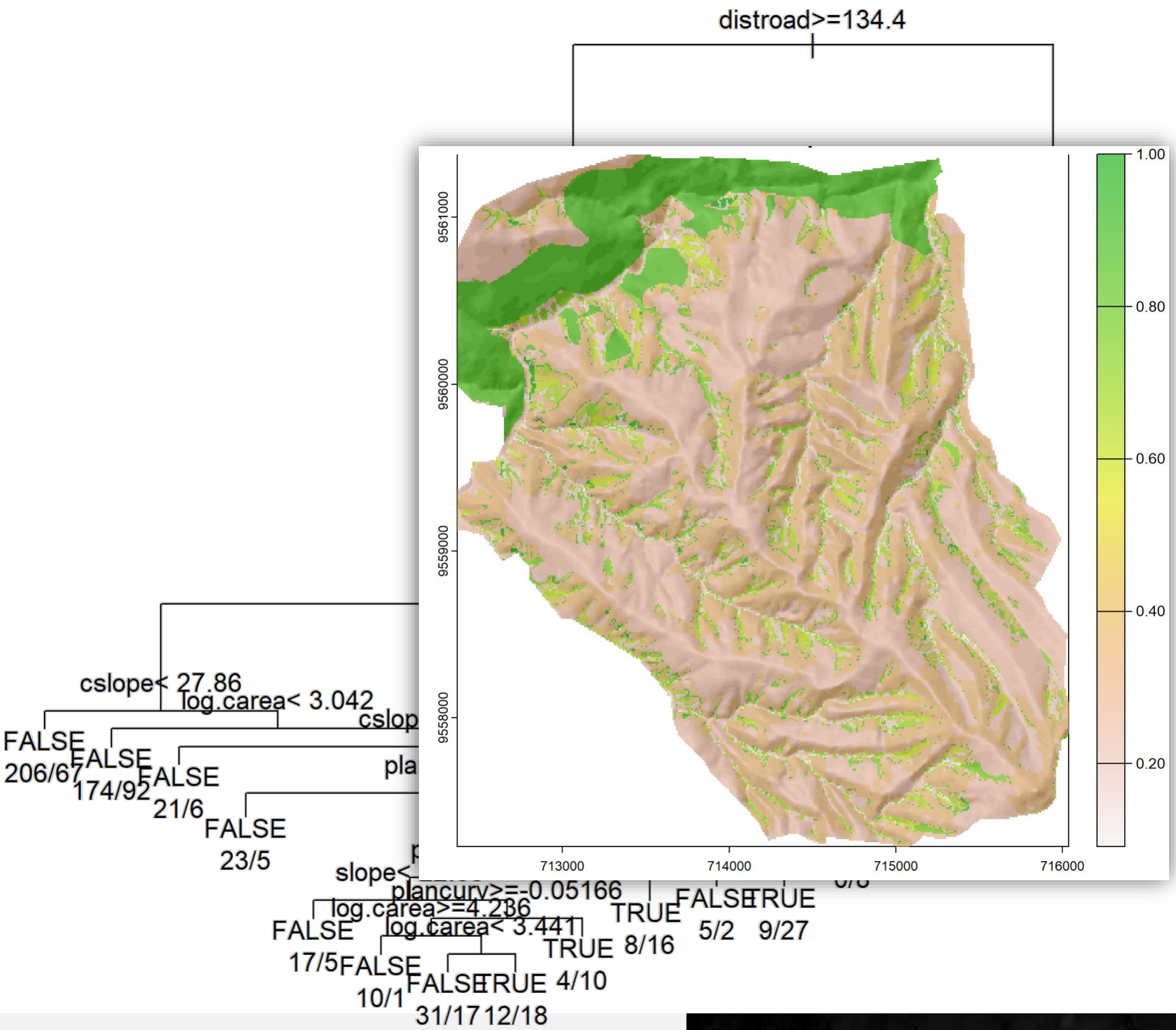
Generalized Additive Model (GAM)

- GAM is a nonlinear extension of the generalized linear model
- Flexible, but can be used for statistical inference
- Additive → fully interpretable



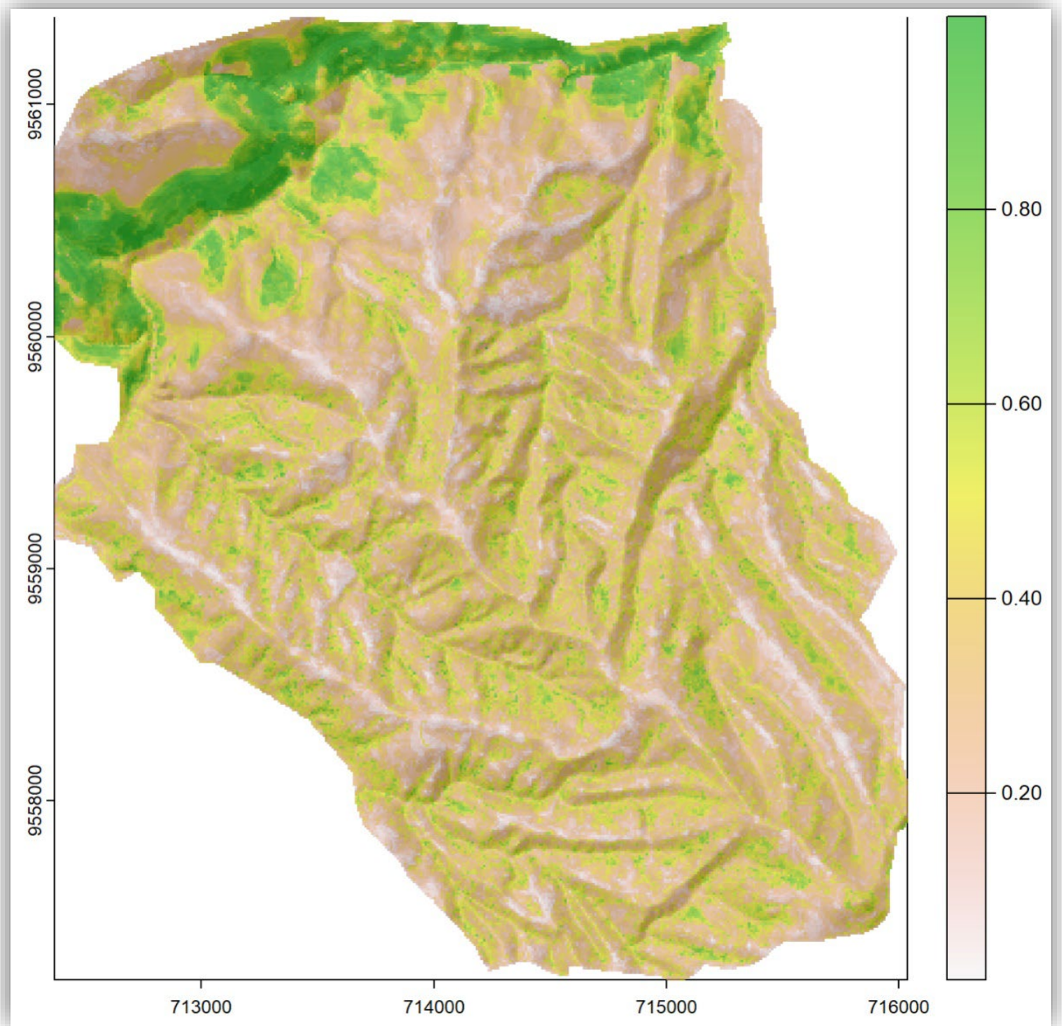
Classification Tree

- Insensitive to outliers, but still somewhat fragile
- Nonlinear, discontinuous relationships
- Statistical interactions of variables → *not* additive
- No statistical inference
- Not really interpretable in practice
- Discontinuities in prediction map



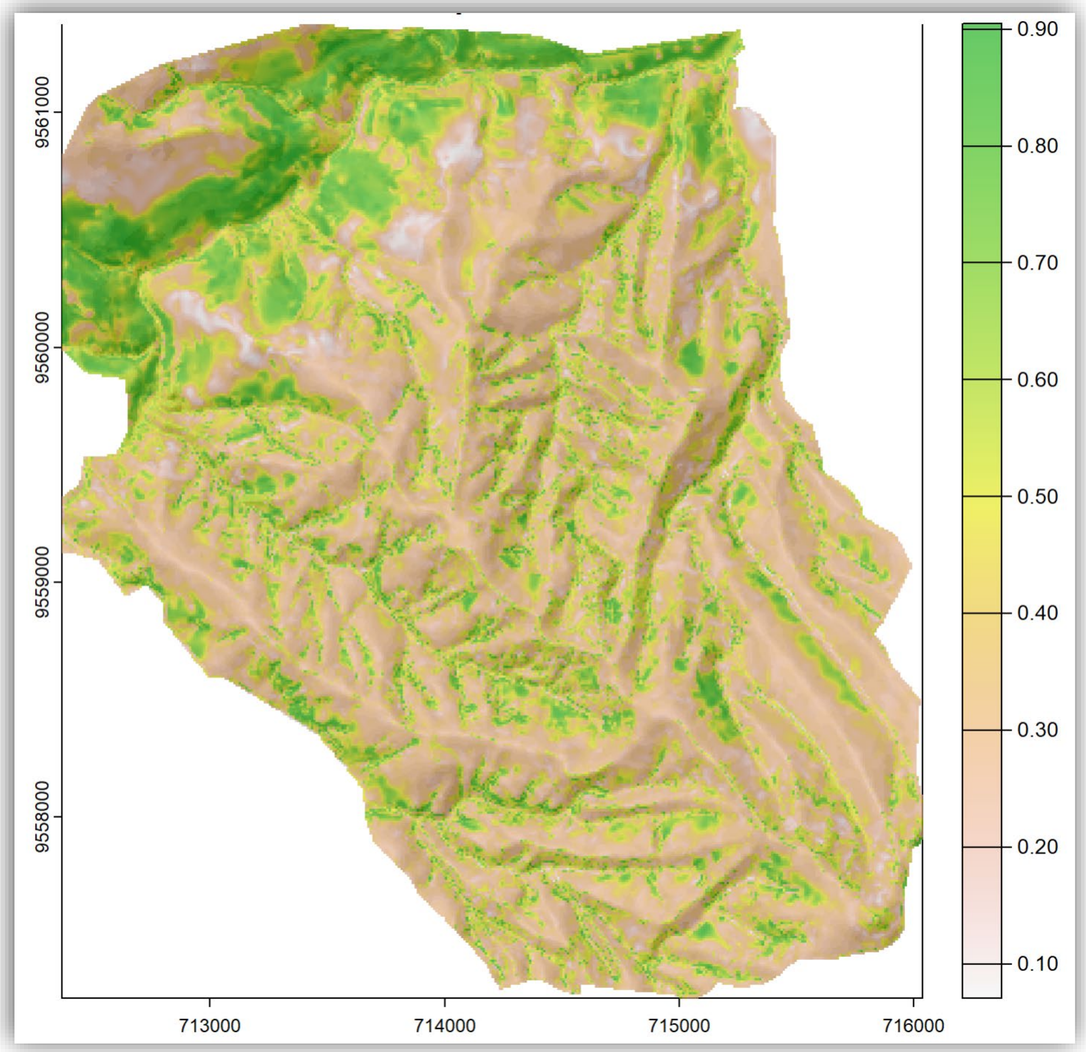
Random Forest

- Ensemble method based on hundreds of classification trees
- Highly robust and flexible
- Often works well with default settings
- Interpretation tools needed to understand model structure
- Prediction map may look noisy or discontinuous



Support Vector Machine

- Models smooth relationships
- Accounts for interactions among variables
- Sensitive to hyperparameters; default settings may be unsuitable



Words of Wisdom

*All models are wrong,
but some are useful*

George E. P. Box (1919-2013)

Machine-Learning Model Assessment

What do we need to assess a model‘s accuracy?

A performance measure

- An overall numerical measure of the goodness of our predictions
- E.g., in classification: overall accuracy, kappa coefficient, **AUROC**, sensitivity, specificity, …
- In probability estimation: Brier score
- Performance measure should be user-centric

An estimation procedure

- We don‘t just “calculate” our performance measure – we estimate it (in the statistical sense of “estimation”)
- We need to start thinking about bias and precision of our *estimates*.

…and of course **suitably sampled data**….

- Complete inventory / random sampling

Training vs. Test-set Estimation

- Performance estimation on **training set** is overoptimistic.
- **Validation**: Randomly split the data set into two disjoint sets: a training set and a **test set**, or **hold-out set**.
  - Yields **unbiased** error estimates
  - But: trade-off between sizes of training vs. test set
  - And what if training and test points are close together?

ibm.com

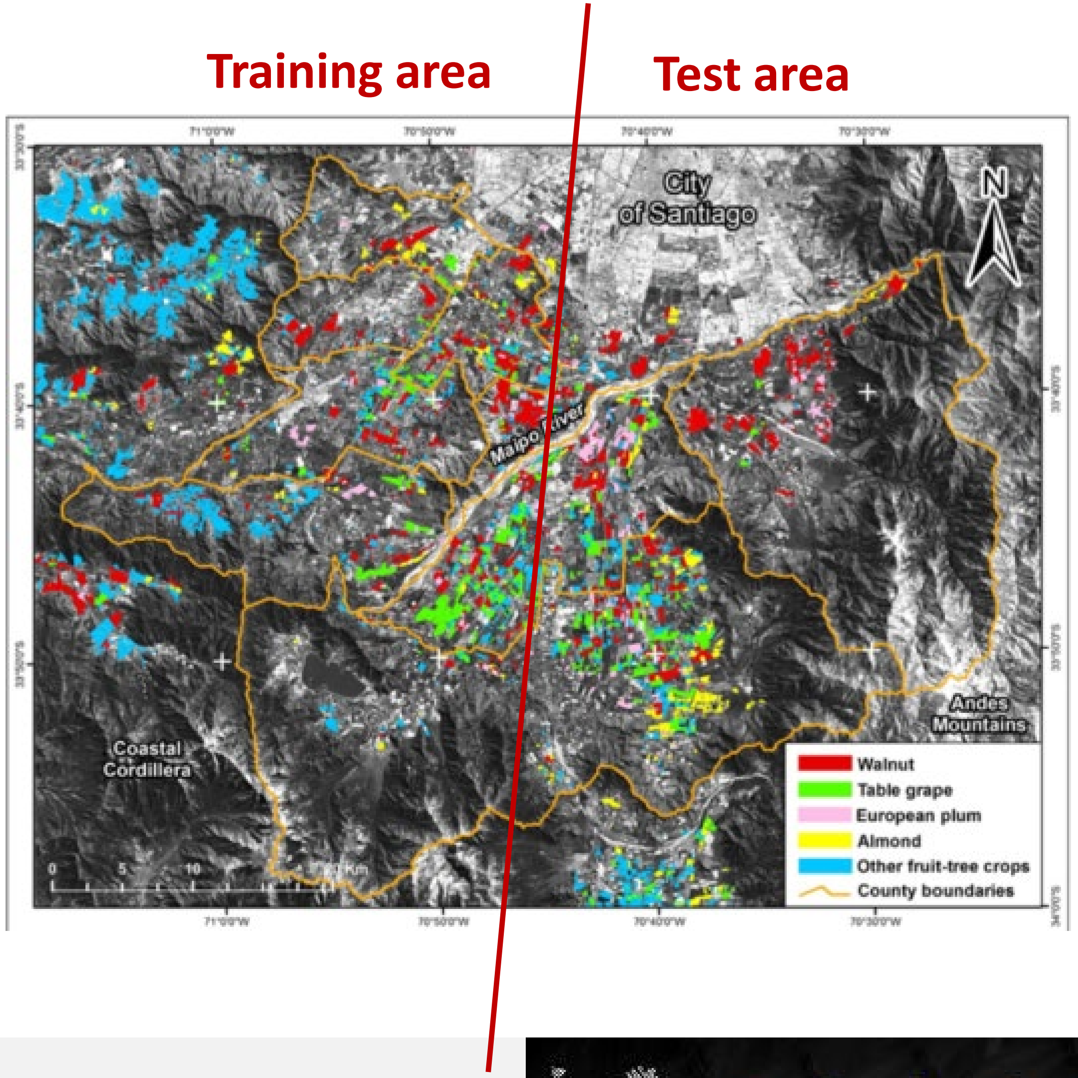
Test *Area* Approach

- Split the study area into spatially disjoint training and test areas.

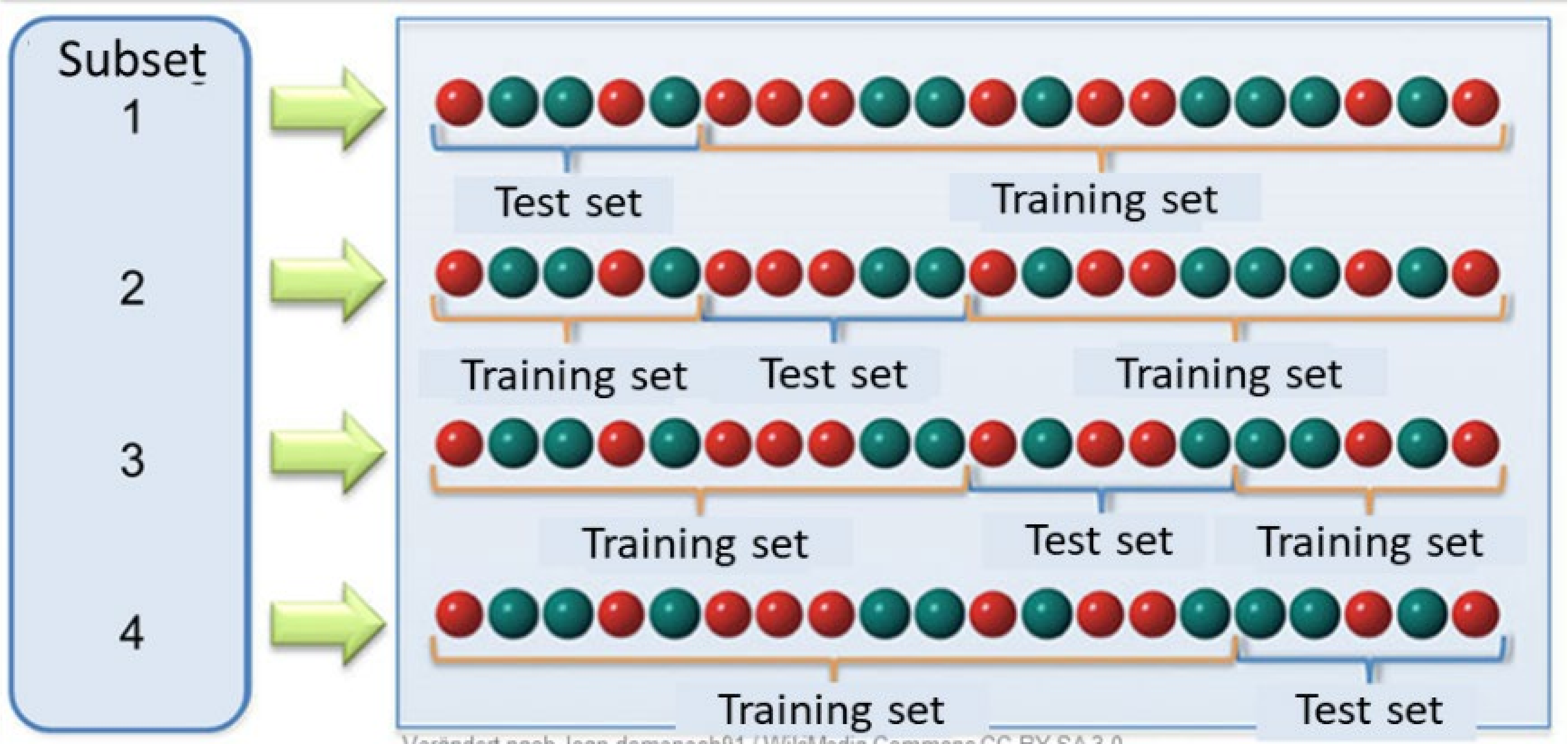
Problem:

- Data distributions may differ, e.g. geological background, topography

→ Test-area error estimates may be biased, not representative for study region.



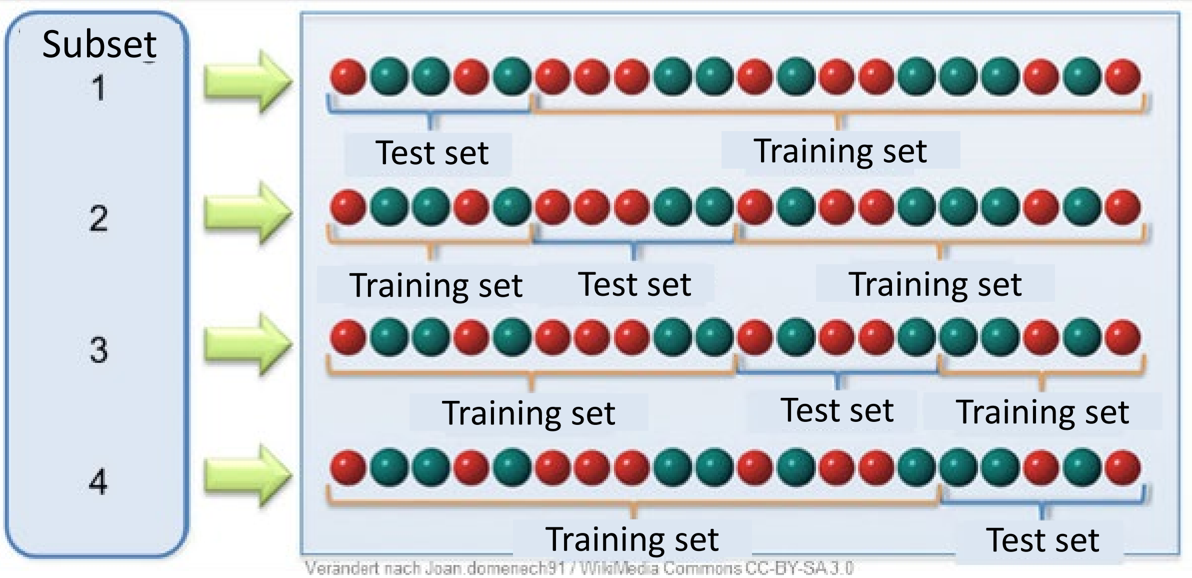
k-fold Cross-Validation



k-fold Cross-Validation

- Randomly partition the sample into *k* equally-sized disjoint subsets.
  - Usually $k = 10$ or $k = 5$.
- Train the classifier on the data from all but one of these subsets,

  …and test it on the held out set.
- Repeat this for all *k* partitions in order to use the entire data set for testing.
  - Also repeat this procedure *r* times using different random partitionings.
- Special case $k = N$: **Leave-one-out cross-validation** (LOO-CV)



Applicable when the goal is

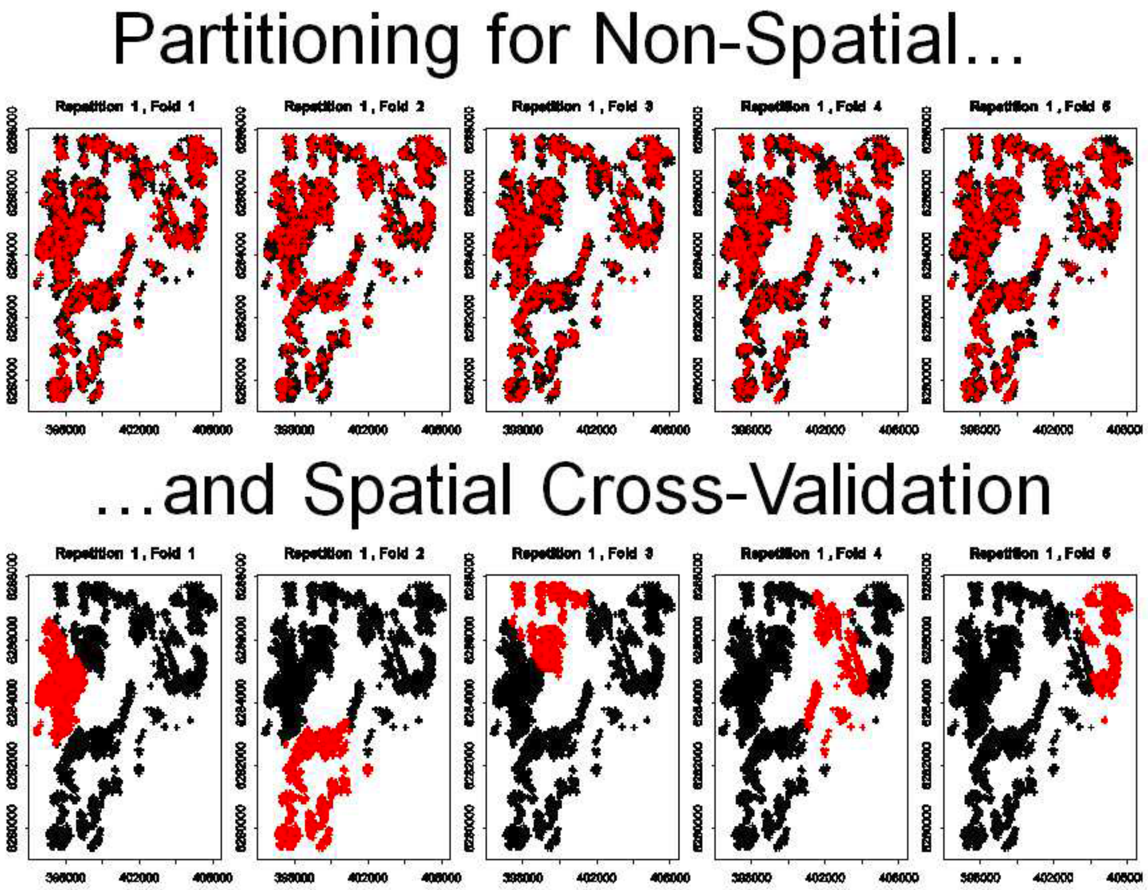
- to predict *within* an area,
- *not* to generalize to new realizations of the random field or to transfer the model to another area.

Spatial Cross-Validation:
Assessing Model Transferability and Generalization

- Divide the study area into disjoint subregions (**blocks**)
  - E.g. using *k*-means clustering of coordinates (Ruß & Brenning, 2010)
- Or: use existing blocks, e.g. agricultural fields

Distinguish between:

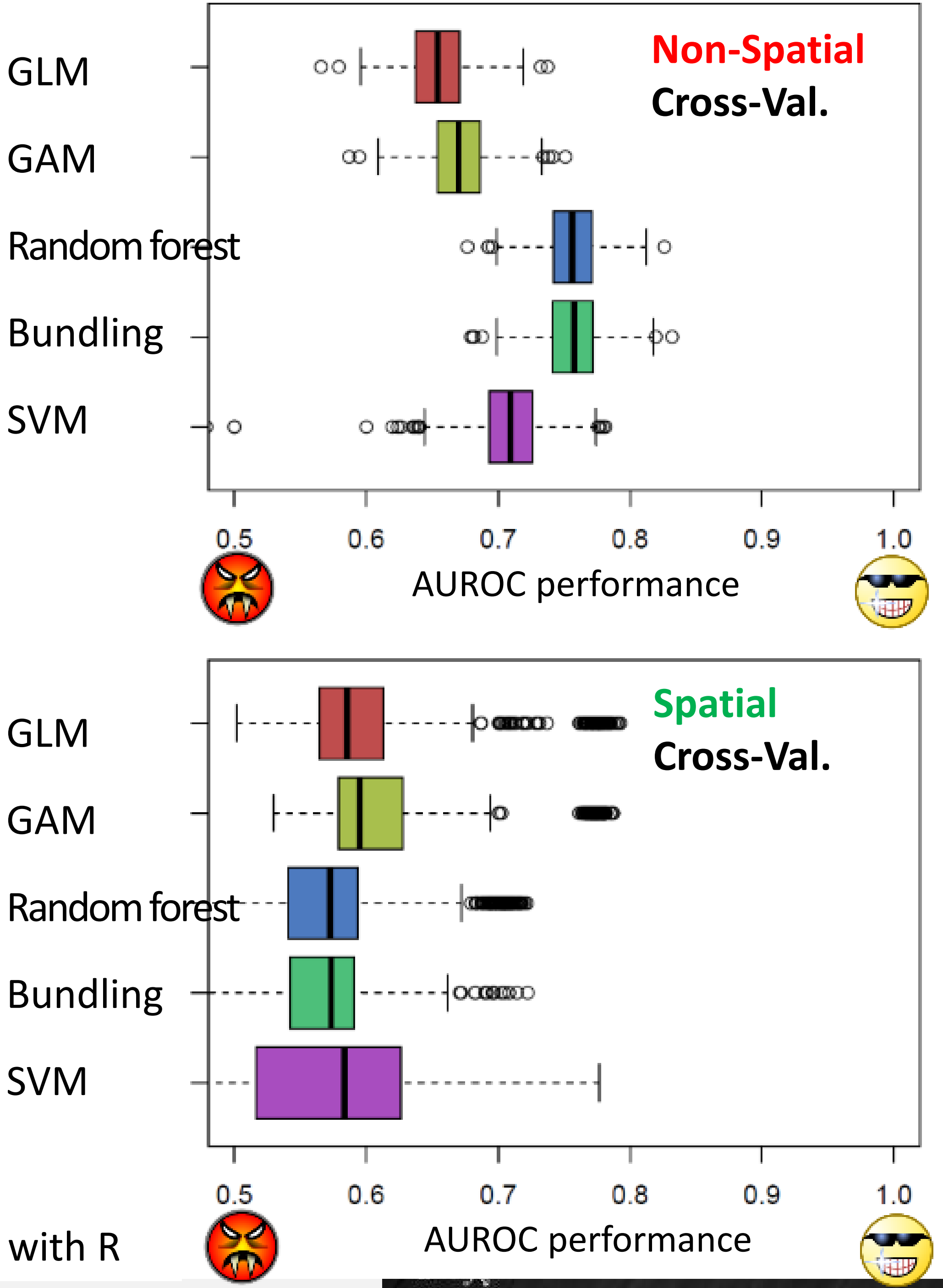
- Leave *one* block out at a time (**leave-one-block-out CV**)
- Partition the the blocks, and leave partition out (**CV at the block level**)



Model Performance: Landslide Susceptibility

- Case study from Ecuadorian Andes
- Non-spatial CV results are over-optimistic
- Spatial CV reveals overfitting to training data
  - Here: leave-one-block out, using k-means clustering to define blocks
- Simpler ML models are more transferable, better able to generalize from training sample

Compare Brenning (2005) in NHESS



Example: Ecuador Landslides
AUROC Estimates

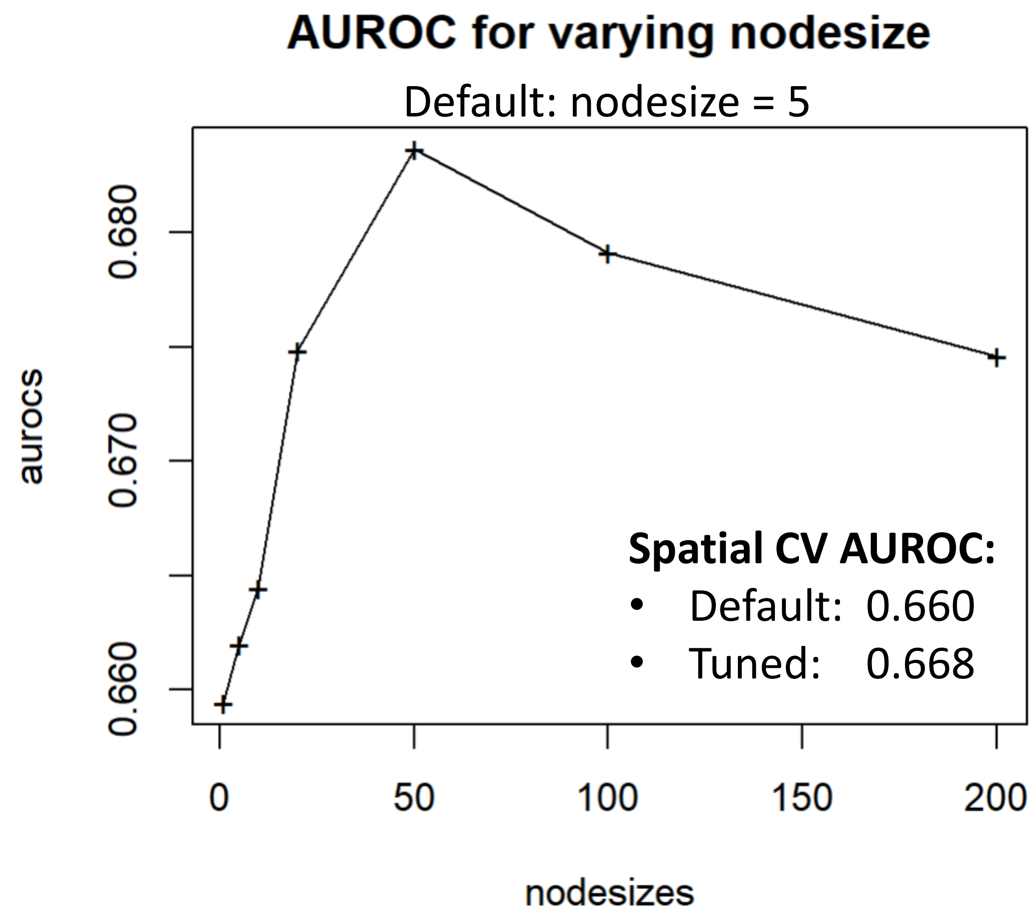
Model	Training set	Random CV	Spatial LOBO CV
GAM	0.784	0.755	**0.687**
CART	0.826	0.704	0.640
Random Forest	1.000	0.749	0.660

Results will vary from one run to another due to random variability → use larger number of repetitions!
10-fold cross-validation
LOBO: leave one spatial block out at a time; blocks defined by k-means clustering of coordinates

Hyperparameter Tuning

- The behaviour of some models depends on so-called **hyperparameters**
  - E.g. random forest: minimum node size of terminal nodes
- Can we improve model performance by choosing the optimal hyperparameter value?
  - Yes, sometimes…
- Computationally expensive optimization task
- Requires a second “layer” of CV (**nested cross-validation**)

Ecuador Landslides
Random Forest



Lessons Learned

- In *predictive* modelling, we can be pragmatic about the type of model used – as long as it provides good predictions.
- CV helps us to reduce bias in model assessments.
- Spatial CV allows us to assess how well a model generalizes from a training set, how transferable it is.

What Can Go Wrong?

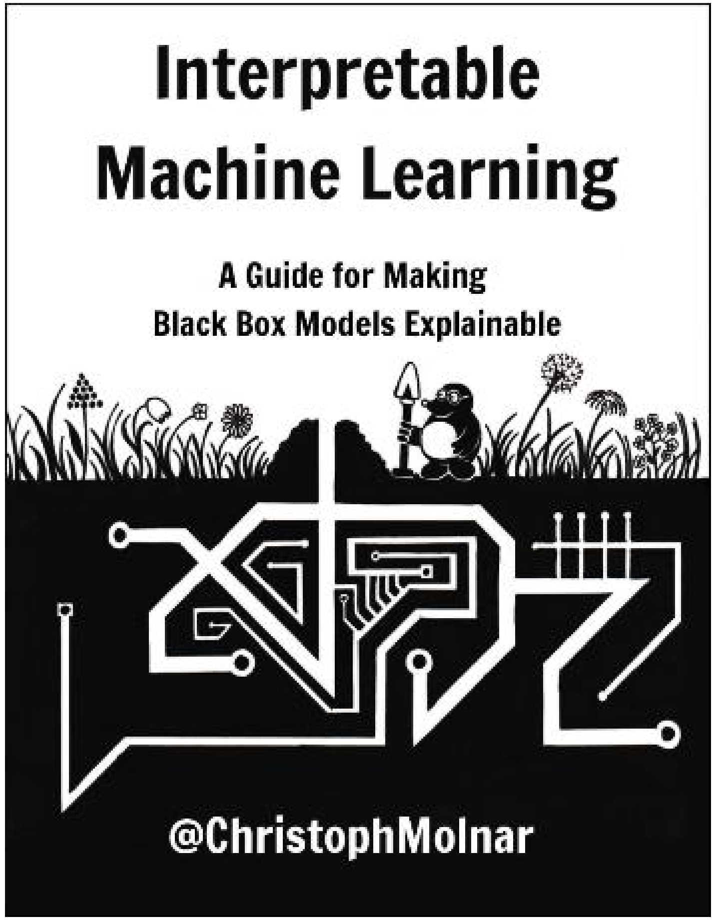
- The type of (re-) sampling used for model assessment must be consistent with the prediction task at hand.
  - E.g., range of prediction distances in spatial model application
  - E.g., forecasting vs. hindcasting for a given prediction horizon
- Never use the same test set for hyperparameter tuning and model assessment.
  - This will lead to an over-optimistic model assessment.
  - Use nested CV.
- Use appropriate error measures that match the prediction task.
  - Different types of misclassification can have different consequences (‘costs’).
- It’s not just about the numbers.
  - Consider qualitative aspects.

Interpretable Machine Learning

- What relationships are represented in my model?
- Which variables really contributed to the model, to its predictions?

- Intrinsically interpretable models versus black-box models
- Model-specific versus **model-agnostic** interpretation tools
- Feature summary statistics versus visualization
- Dataset-level versus **instance-level** (or local) interpretation

For a comprehensive overview, please consult this book:



https://christophm.github.io/interpretable-ml-book/

Permutation-based Variable Importance (PVI)

- PVI measures a predictor's overall contribution to the model's predictive skill.
- It is based on **how the performance measure changes when a predictor is permuted**.
- Criticism: PVI uses nonsensical combinations of feature values.
- Popular now (but with similar issues): SHAP feature importance

Algorithm:

Input: Trained model.

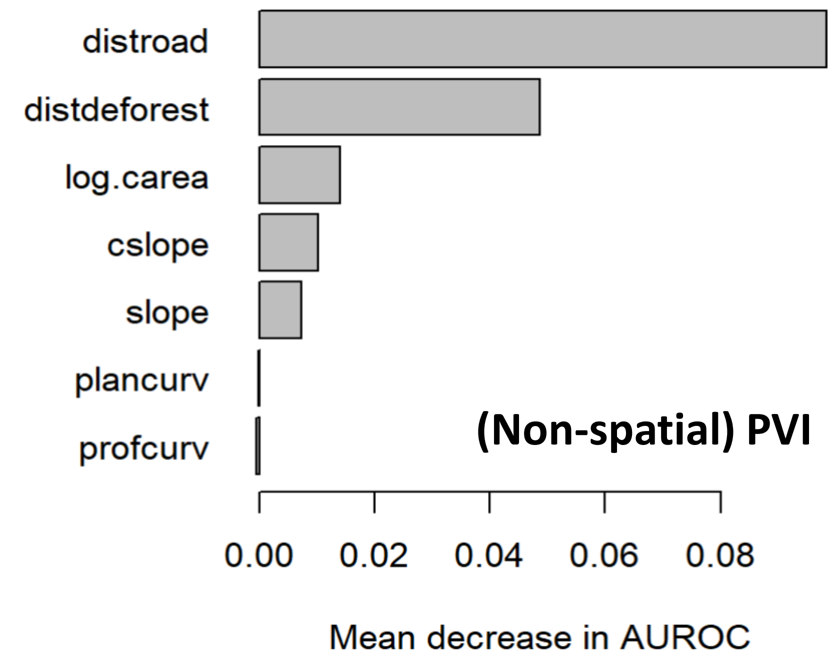
1. Assess its accuracy on the test set.
2. Permute a predictor on the test set, and use this partly "messed-up" data for prediction and accuracy assessment. Repeat this many times, using different random permutations of the predictor.
3. Calculate the mean difference between "regular" and "messed-up" accuracy.

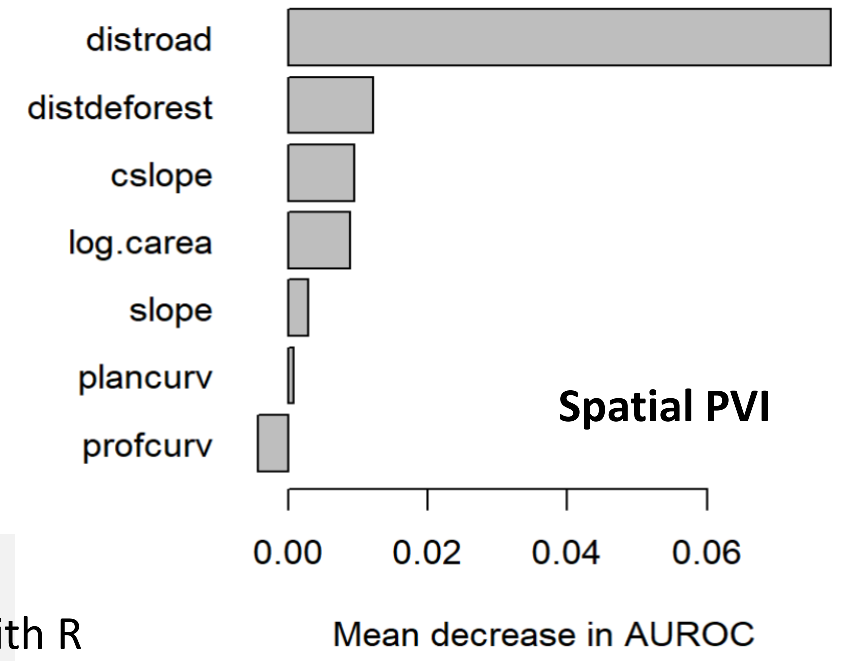
Repeat this for each variable, and for each cross-validation training / test set combination.

Spatial PVI

- “Standard” PVI ignores spatial dependence or grouping as well as the prediction horizon in the model’s application

- Embed PVI assessment within a spatial CV to assess a variable’s ability to contribute to *generalizable* or *transferable* predictive capabilities.
  - `sperrorest` package





Partial Dependence Plot (PDP)

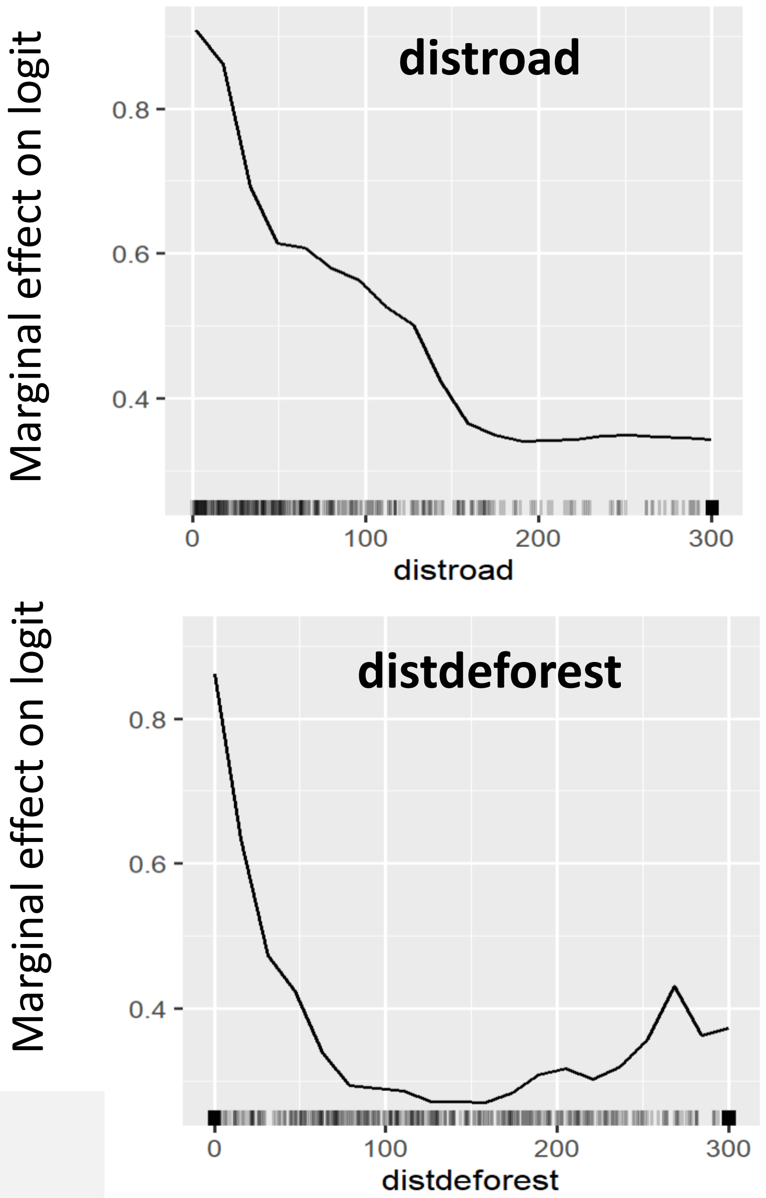
- In GAM modelling, we were able to plot the transformation functions. Can we create similar plots for more complex models?
- **Partial dependence function**:

$$\hat{f}_{x_S}(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}\left(x_S, x_C^{(i)}\right)$$

where $\hat{f}$ is the fitted model, x_s the selected variable, and x_c all other predictors. The sum is over all observations.

- In R: function `partialPlot` in package `randomForest`, or package `pdp` more generally
- Problem: PDP ignore dependencies among predictors

Ecuador landslides: Random Forest PDPs



Accumulated Local Effects (ALE) Plot

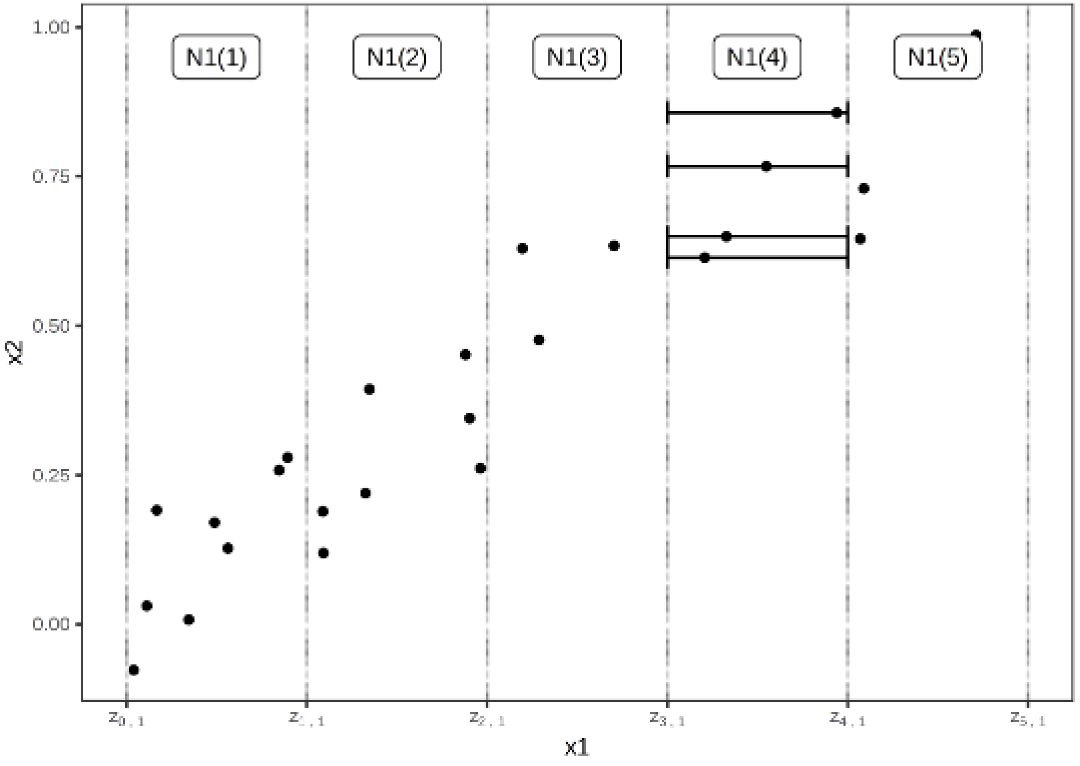
- The ALE plot looks at how the model predictions *change* in a small window of the predictor.
- It averages only over observations *in* that moving window, not over all observations.
- This solves two issues of the PDP and PVI:
  - Nonsensical, or highly unlikely, instances are avoided.
  - The effects of correlated predictors are separated.

$$\hat{f}_{j,ALE}(x) = \underbrace{\sum_{k=1}^{k_j(x)}}_{\text{accumulate}} \frac{1}{n_j(k)} \sum_{i:x_j^{(i)} \in N_j(k)} \underbrace{\left[f(z_{k,j}, x_{\setminus j}^{(i)}) - f(z_{k-1,j}, x_{\setminus j}^{(i)})\right]}_{\text{differences in predictions}}$$

differences in predictions
for different predictor values ("**effects**")

accumulate over all intervals, up to the one corresponding to x_j

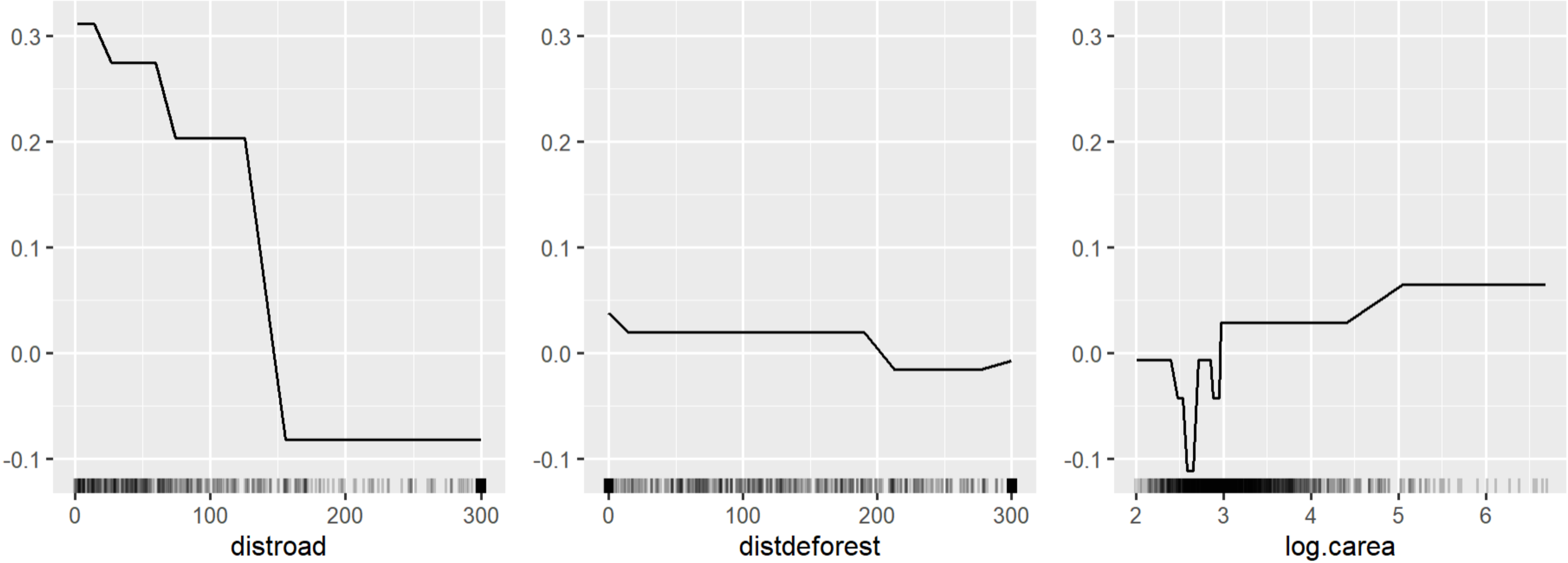
sum over all features within the **local** neighborhood in the variable of interest, x_j



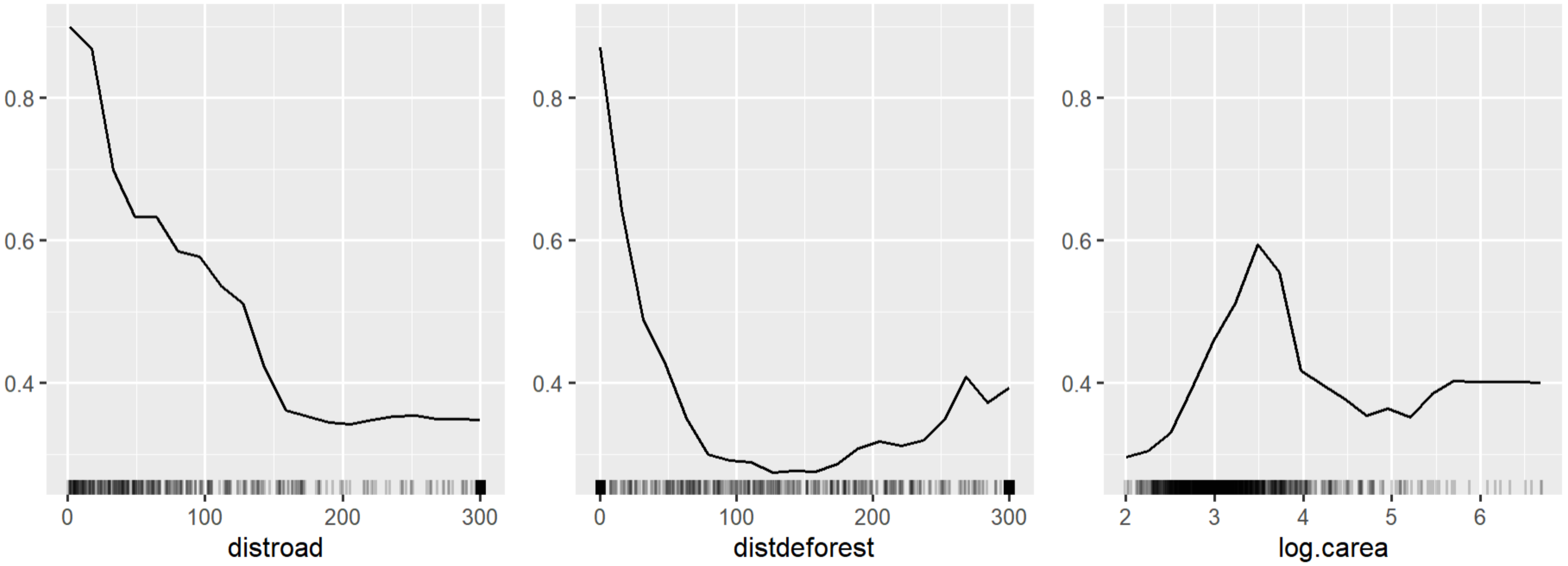
https://christophm.github.io/interpretable-ml-book/ale.html#ale

In R: packages **ALEPlot**, **iml** and **DALEX**

Case study: Ecuador landslides
ALE plots



Case study: Ecuador landslides
PD plots for comparison



Model Interpretation in High Dimensions

$\mathbf{x}$

$\hat{f}$

$\hat{y}$

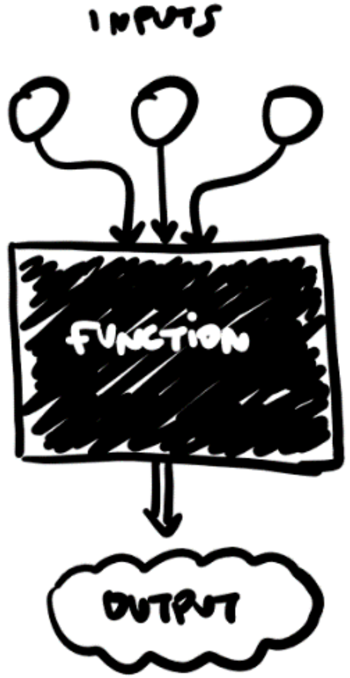
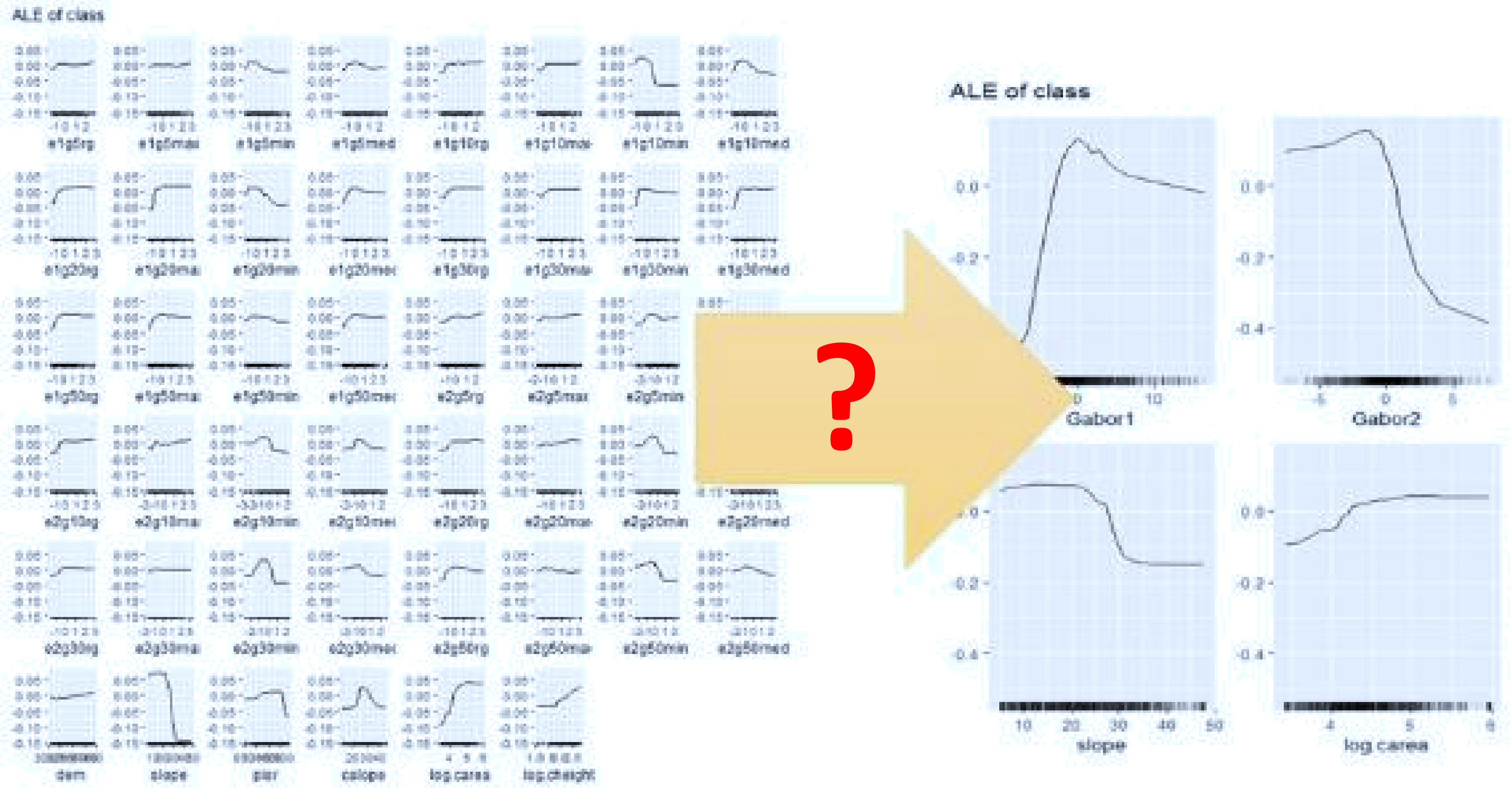


Image source: thatsoftwaredude.com



See Brenning (2023) in *Machine Learning* (and R package **`wiml`**) for a proposal!

Blog post: https://geods.netlify.app/post/interpretable-ml-with-a-twist/

Lessons Learned

- The interpretation of complex machine-learning models is limited, requires simplification.

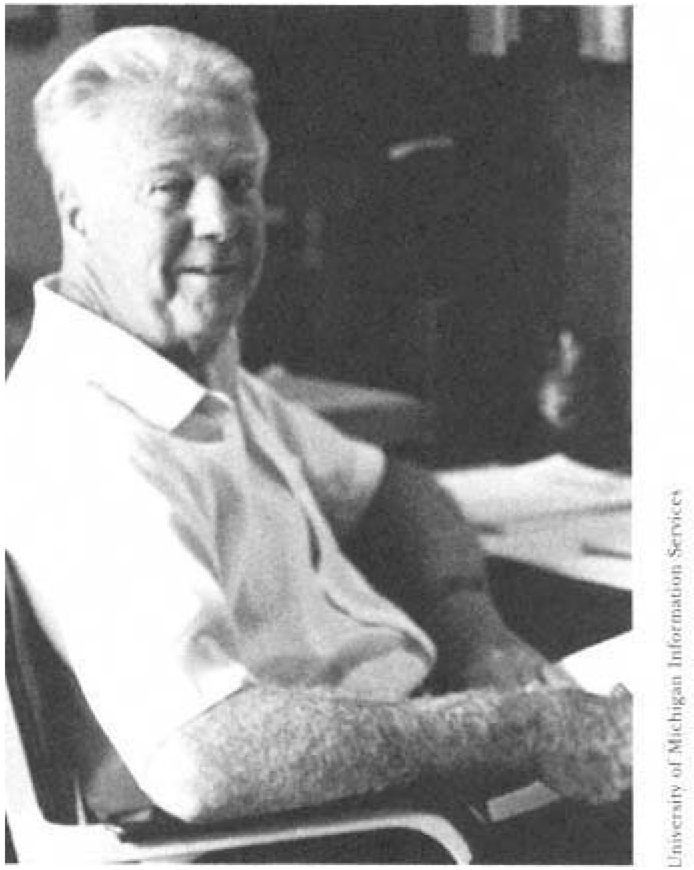
- Permutation-based feature importance provides a very rough measure of a variable’s contribution to the predictive performance.
  - A simpler approach is to compare performances achieved with different feature sets.

- Partial dependence and ALE plots allow us to visualize the relationships between response and predictors, one at a time.
  - Prefer ALE plots, but understand their limitations.

What Can Go Wrong?

- For complex ML model, interpretation tools never tell you the whole story.
  - They are models of models…
- If a key objective is to interpret your data, use an interpretable model.
  - Additive models are great.
- Related features will always “steal” importance from each other.
  - Importance means “importance while accounting for the other variables in the model”.
- Different features can be important in different predictive settings.

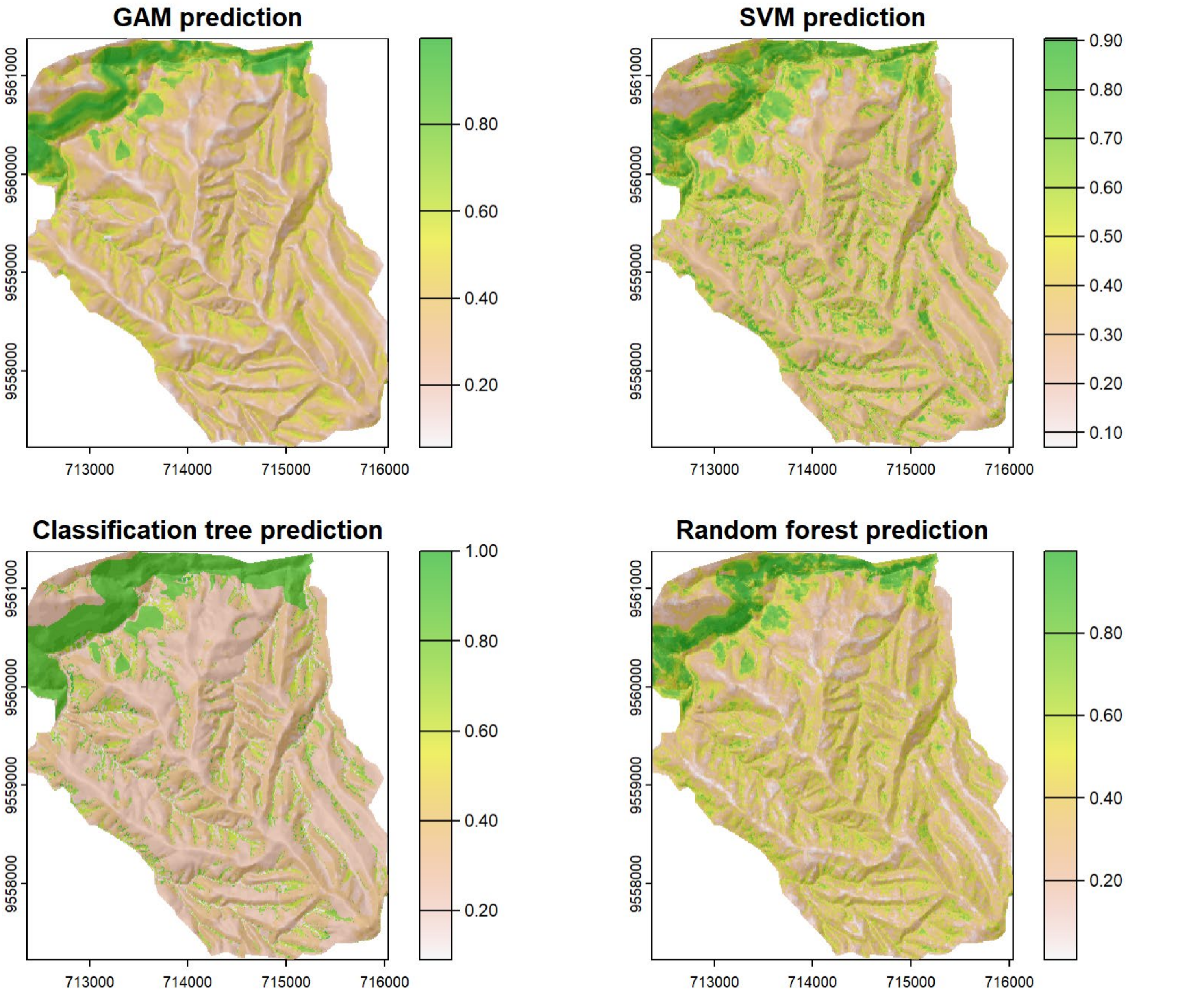
Words of Wisdom (Coombs 1964, *Theory of Data*)



"we buy information with assumptions"

Clyde Hamilton Coombs (1912-1988)

But let‘s finish with a map…



Links & References

@geobrenning

- R code & data related to this workshop: https://github.com/alexanderbrenning/agr24_workshop

- Brenning, A. (2012). Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: the R package 'sperrorest'. *Proceedings, 2012 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 23-27 July 2012, 5372-5375. [link] [package]
- Brenning, A. (2023). Spatial machine-learning model diagnostics: a model-agnostic distance-based approach. *International Journal of Geographical Information Science*. [link] [github]
  - Blog article: https://geods.netlify.app/post/spatial-ml-model-diagnostics/
- Brenning, A. (2023). Interpreting machine-learning models in transformed feature space with an application to remote-sensing classification. *Machine Learning*. [link] [package]
  - Blog article: https://geods.netlify.app/post/interpretable-ml-with-a-twist/
- Molnar, C. 2023. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. [link]
- Schratz, P., Becker, M., Lang, M., Brenning, A. (2022). Mlr3spatiotempcv: Spatiotemporal resampling methods for machine learning in R. *arXiv preprint*. [link]