Workshop

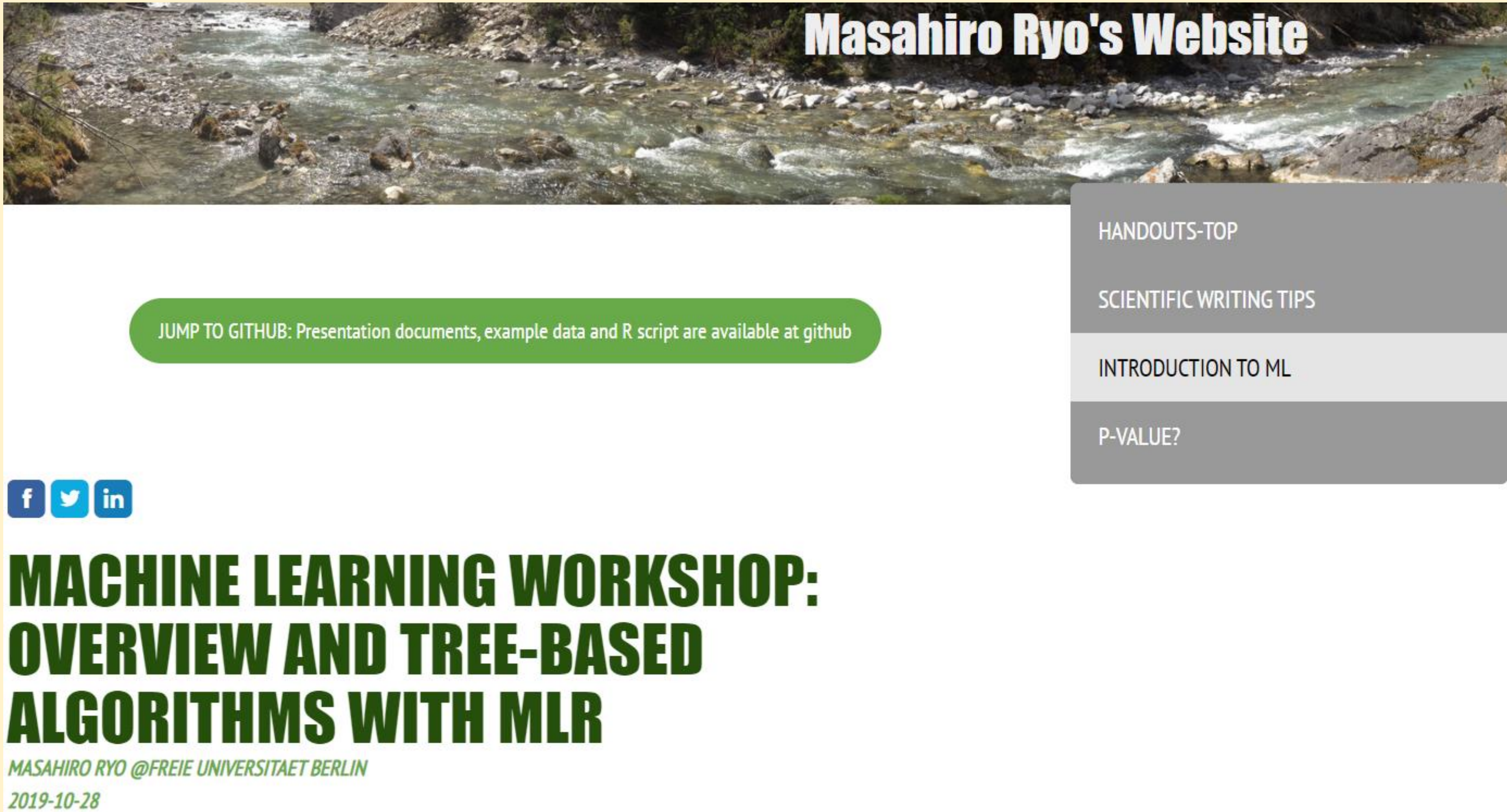
Introduction to Machine Learning in R

Masahiro Ryo

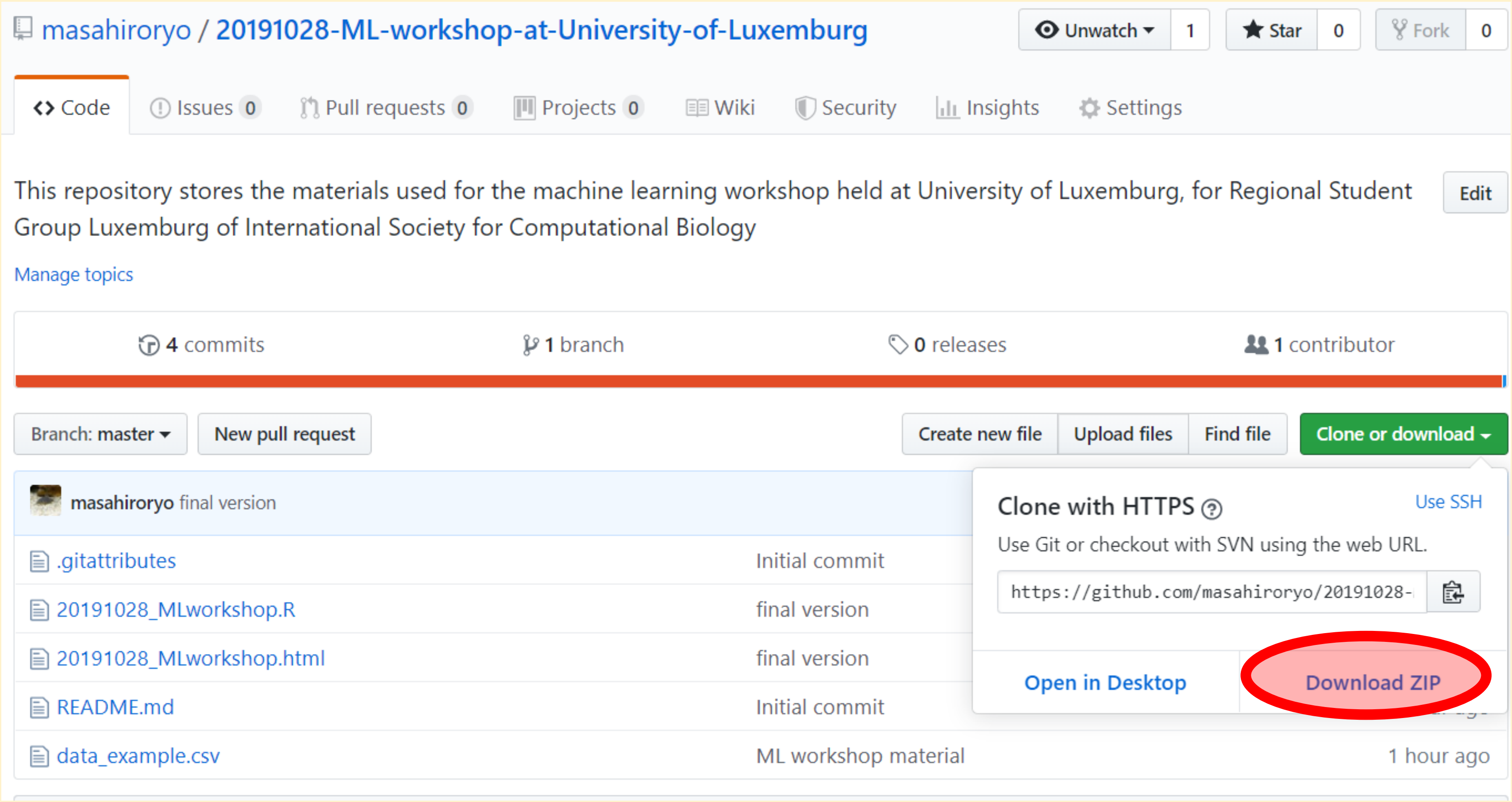
Free University of Berlin

Berlin-Brandenburg Institute of Advanced Biodiversity Research

https://masahiroryo.jimdo.com/introduction-to-ml/



https://github.com/masahiroryo/20191028-ML-workshop-at-University-of-Luxemburg



An overview of a practical ML workflow

1. Task, Learner, Training, & Prediction

   What to do? Which algorithm to use?

2. Performance assessment

   Good enough? How to assess it?

3. Fine tuning (preferred)

   Data preprocessing (e.g. transformation), feature selection, hyperparameter tuning

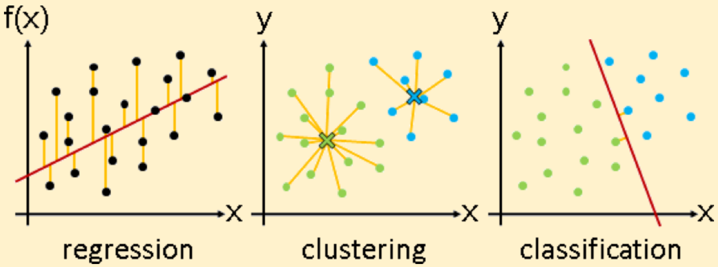
4. Interpretation (advanced)

   Effect size: Variable importance measure
   Effect pattern: Partial dependence plot, ICE plot, ACE plot ...

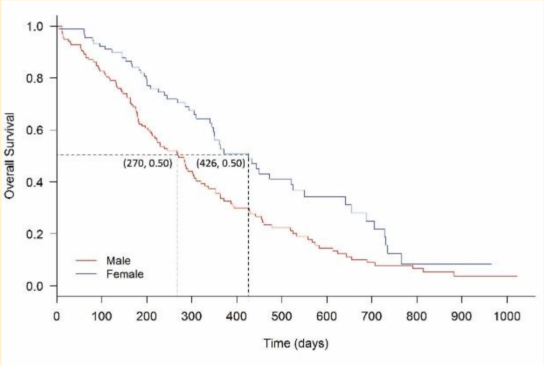
5. Careful attention (advanced)

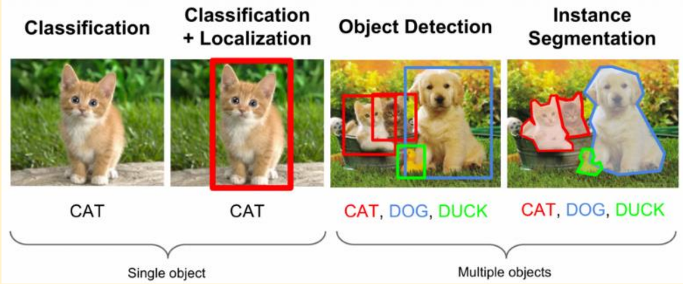
   Variable interaction: Friedman's H-statistic, Basu et al. 2018, Ryo et al. 2018 ...
   Reliability check: LIME (local interpretable model-agnostic explanations)
   Stability check: Confidence interval estimate
   Data pattern: Imbalance dataset, spatial/temporal data

1.1. Task   What to do?



1. Regression
2. Classification
3. Clustering
4. Dimension reduction
5. Survival analysis
6. Multilabel classification
7. Reinforcement learning





1.2. Learner Which algorithm to use?

1. Regression (59)
2. Classification (82)
3. Clustering (10)
4. Dimension reduction
5. Survival analysis (12)
6. Multilabel classification (3)
7. Reinforcement learning

Classification (82)

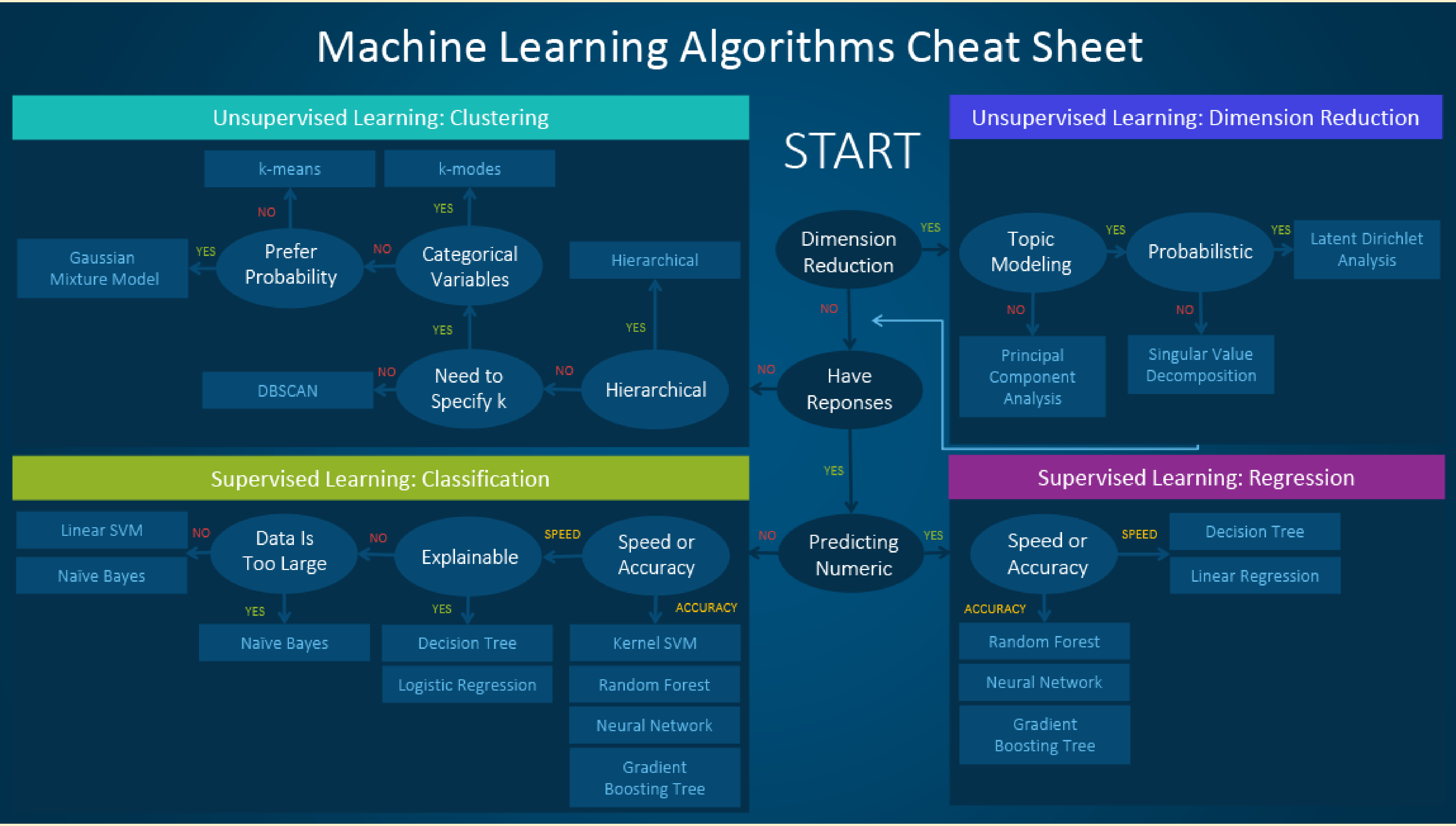
For classification the following additional learner properties are relevant and shown in column **Props**:

- *prob*: The method can predict probabilities,
- *oneclass, twoclass, multiclass*: One-class, two-class (binary) or multi-class classification problems be handled,
- *class.weights*: Class weights can be handled.

Class / Short Name / Name	Packages	Num.	Fac.	Ord.	NAs	Weights	Props	Note
classif.ada *ada* ada Boosting	ada rpart	X	X				prob twoclass	`xval` has bee default for spe
classif.adaboostm1 *adaboostm1* ada Boosting M1	RWeka	X	X				prob twoclass multiclass	NAs are direct WEKA with `na` `na.pass`.
classif.bartMachine *bartmachine* Bayesian Additive Regression Trees	bartMachine	X	X		X		prob twoclass	`use_missing` set to `TRUE` b missing data s
classif.binomial *binomial* Binomial Regression	stats	X	X			X	prob twoclass	Delegates to choosable bin function via le `link`. We set by default to s
classif.boosting *adabag* Adabag Boosting	adabag rpart	X	X		X		prob twoclass multiclass featimp	`xval` has bee default for spe
classif.bst *bst*	bst rpart	X					twoclass	Renamed para to `Learner` d

https://mlr.mlr-org.com/articles/tutorial/integrated_learners.html

1.2. Learner Which algorithm to use?



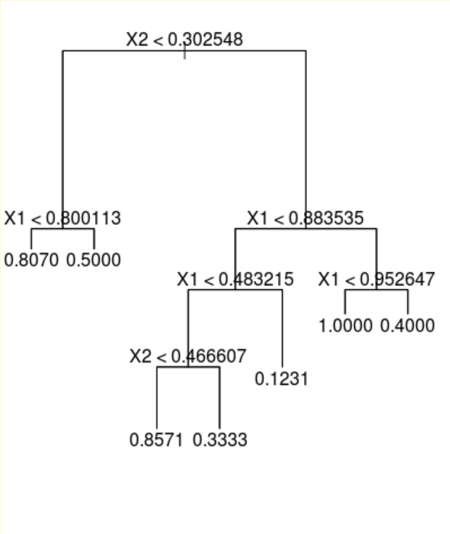
https://www.kdnuggets.com/2017/06/which-machine-learning-algorithm.html

1.3. Train   Data + algorithm = a model

	y1 <dbl>	x1 <int>	x2 <int>	x3 <int>	x4 <fctr>	x5 <fctr>
1	2.795473	0	0	0	low	A
2	6.366427	1	1	1	high	*NA*
3	2.452364	0	1	1	moderate	B

INPUT: S, **where** S = *set of classified instances*
OUTPUT: *Decision Tree*
Require: $S \neq \emptyset$, $num\_attributes > 0$

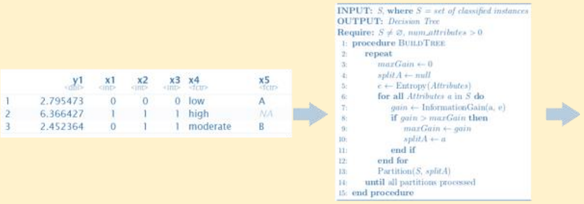
```
procedure BuildTree
    repeat
        maxGain ← 0
        splitA ← null
        e ← Entropy(Attributes)
        for all Attributes a in S do
            gain ← InformationGain(a, e)
            if gain > maxGain then
                maxGain ← gain
                splitA ← a
            end if
        end for
        Partition(S, splitA)
    until all partitions processed
end procedure
```

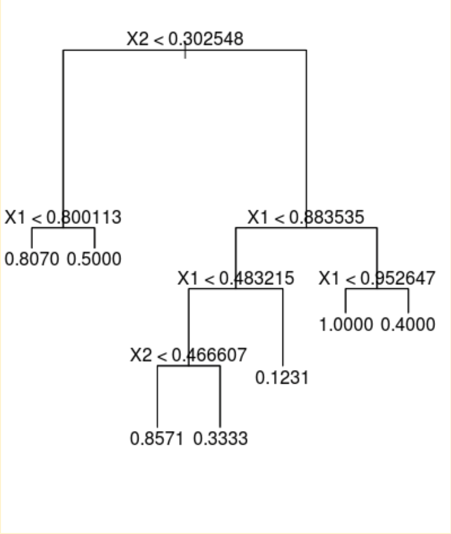


1.4. Prediction Newdata -> a model = prediction

New data (y1 is unknown)

y1 <dbl>	x1 <int>	x2 <int>	x3 <int>	x4 <fctr>	x5 <fctr>	x6 <fctr>
?	0	0	1	moderate	A	C
	1	0	1	moderate	A	NA
	1	0	1	high	A	B
	1	0	1	high	D	B
	0	1	1	low	A	B





Prediction

y1 <dbl>
-2.9829087
7.0826056
6.4154060
-0.1391601
6.2386937

Data Preprocessing

Source: `vignettes/tutorial/preproc.Rmd`

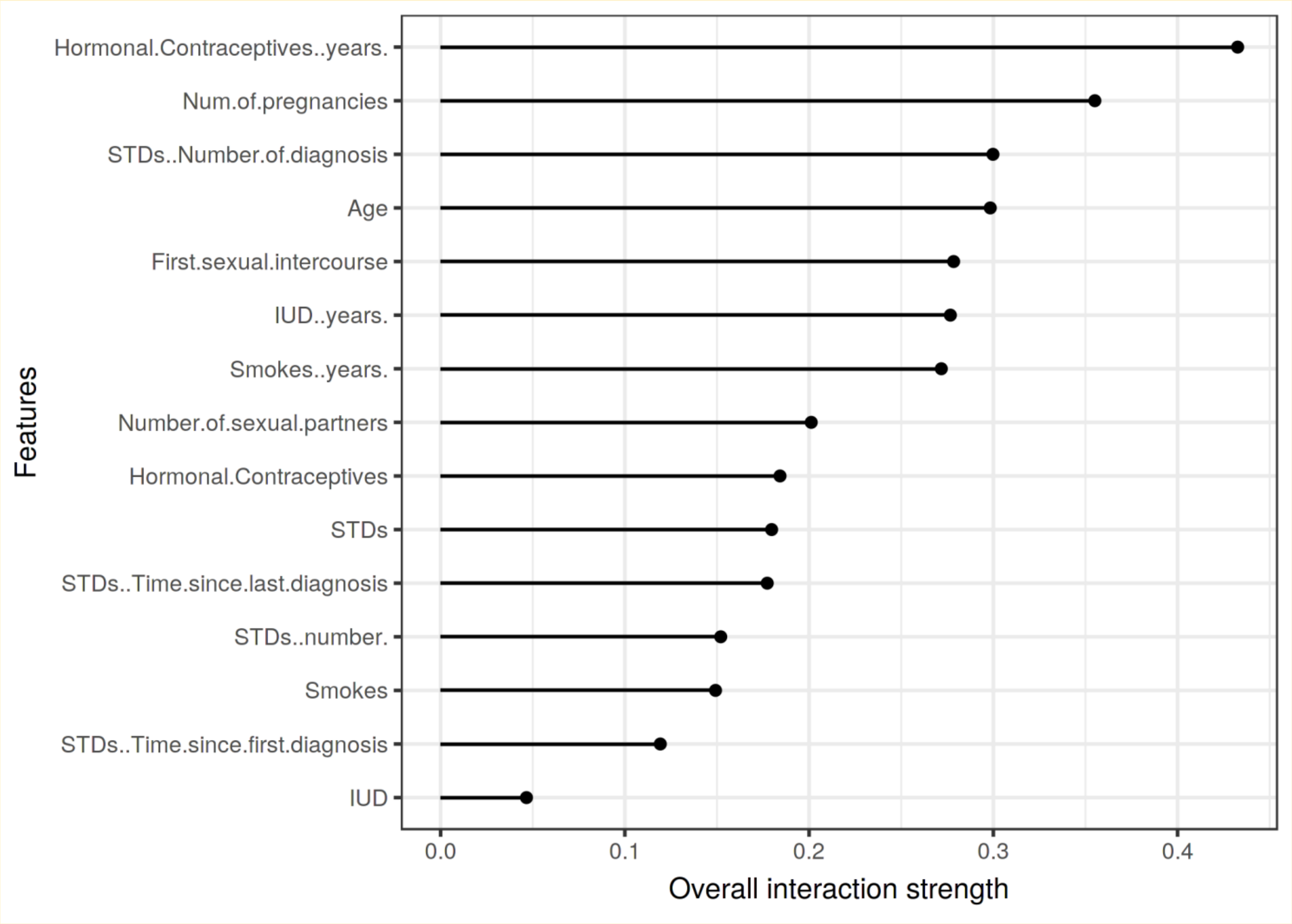
Data preprocessing refers to any transformation of the data done before applying a learning algorithm. This comprises for example finding and resolving inconsistencies, imputation of missing values, identifying, removing or replacing outliers, discretizing numerical data or generating numerical dummy variables for categorical data, any kind of transformation like standardization of predictors or Box-Cox, dimensionality reduction and feature extraction and/or selection.

`mlr` offers several options for data preprocessing. Some of the following simple methods to change a `Task()` (or `data.frame`) were already mentioned on the page about learning tasks:

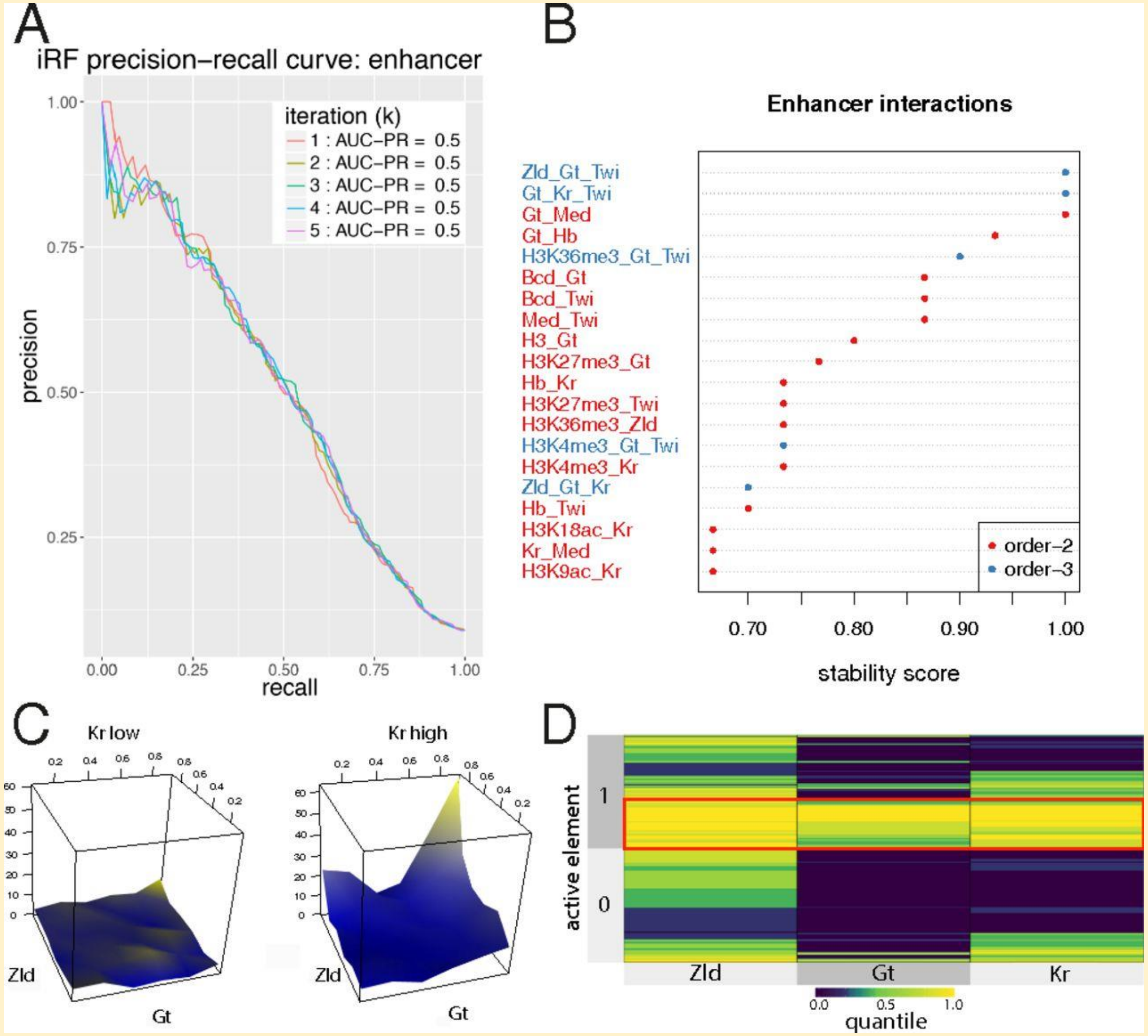
- `capLargeValues()` : Convert large/infinite numeric values.
- `createDummyFeatures()` : Generate dummy variables for factor features.
- `dropFeatures()` : Remove selected features.
- `joinClassLevels()` : Only for classification: Merge existing classes to new, larger classes.
- `mergeSmallFactorLevels()` : Merge infrequent levels of factor features.
- `normalizeFeatures()` : Normalize features by different methods, e.g., standardization or scaling to a certain range.
- `removeConstantFeatures()` : Remove constant features.
- `subsetTask()` : Remove observations and/or features from a `Task()` .

Moreover, there are tutorial pages devoted to

- Feature selection and
- Imputation of missing values.



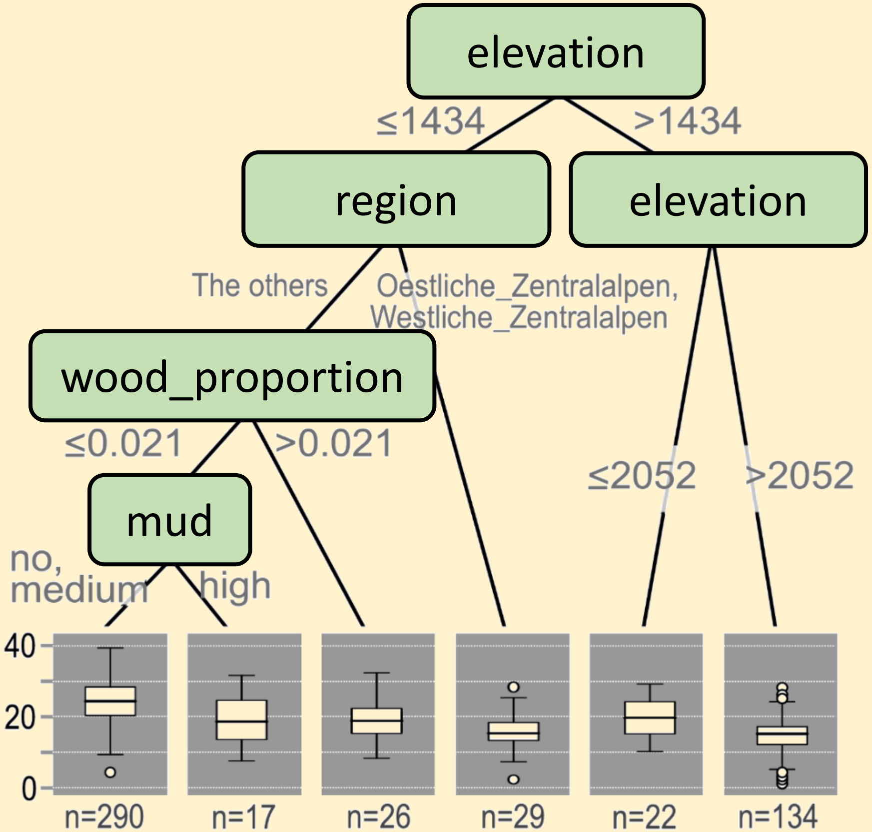
https://christophm.github.io/interpretable-ml-book/interaction.html



A
iRF precision−recall curve: enhancer
iteration (k)
1 : AUC−PR = 0.5
2 : AUC−PR = 0.5
3 : AUC−PR = 0.5
4 : AUC−PR = 0.5
5 : AUC−PR = 0.5
precision
recall
B
Enhancer interactions
Zld_Gt_Twi
Gt_Kr_Twi
Gt_Med
Gt_Hb
H3K36me3_Gt_Twi
Bcd_Gt
Bcd_Twi
Med_Twi
H3_Gt
H3K27me3_Gt
Hb_Kr
H3K27me3_Twi
H3K36me3_Zld
H3K4me3_Gt_Twi
H3K4me3_Kr
Zld_Gt_Kr
Hb_Twi
H3K18ac_Kr
Kr_Med
H3K9ac_Kr
order−2
order−3
stability score
C
Kr low
Kr high
Zld
Gt
D
active element
Zld
Gt
Kr
quantile

Decision tree Classification and Regression Tree (CART)

(Breiman et al. 1984)

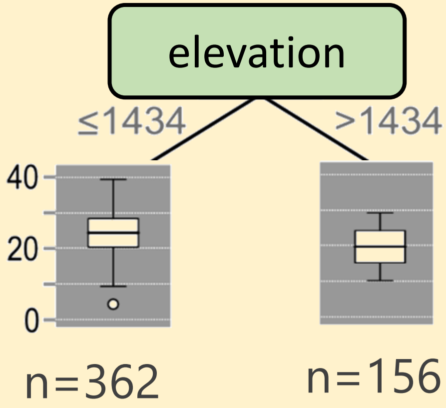


Points

1) No need for *a priori* selection of data & statistical assumptions (just run)
2) Missing values allowed
3) Nonlinearity
4) Indication for variable interactions

Decision tree Classification and Regression Tree (CART)

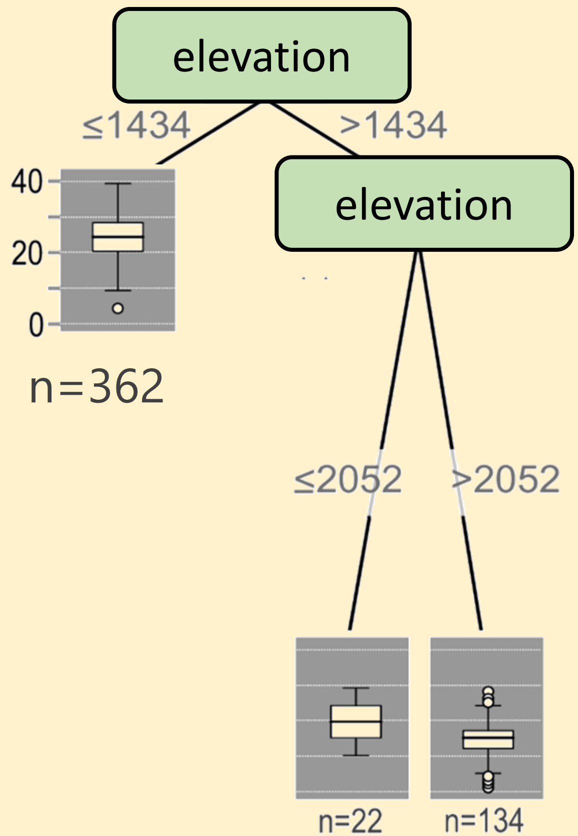
(Breiman et al. 1984)



1) Find a predictor & a threshold value which separate the data into two the most distinctively.
   ⇒ If "elevation" is > 1434 m or not

Decision tree Classification and Regression Tree (CART)

(Breiman et al. 1984)

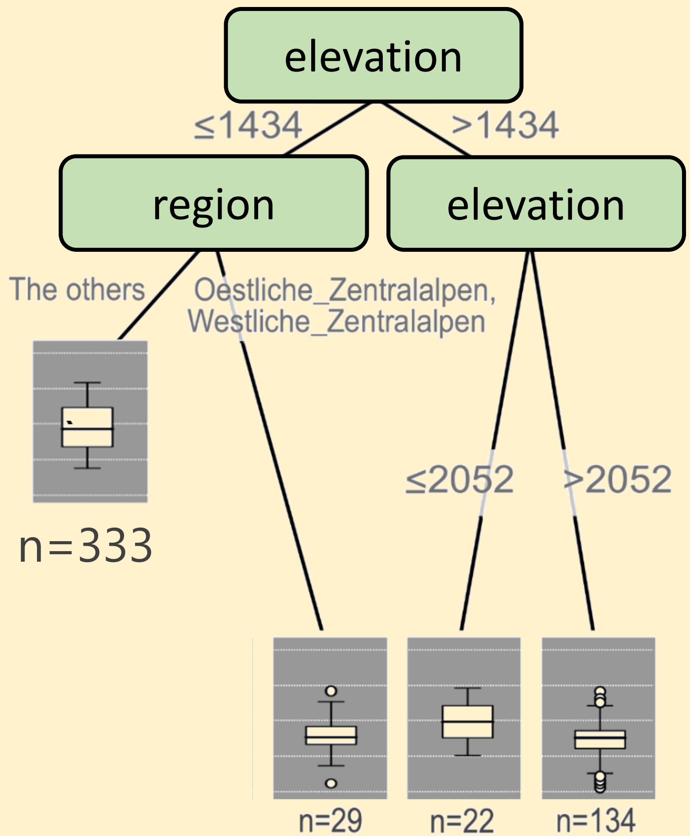


1) Find a predictor & a threshold value which separate the data into two the most distinctively.
   ⇒ *If "elevation" is > 1434 m or not*

2) For each of the separated data, repeat.
   ⇒ *If "elevation" is > 2052 m or not (for right-hand side)*

Decision tree Classification and Regression Tree (CART)

(Breiman et al. 1984)



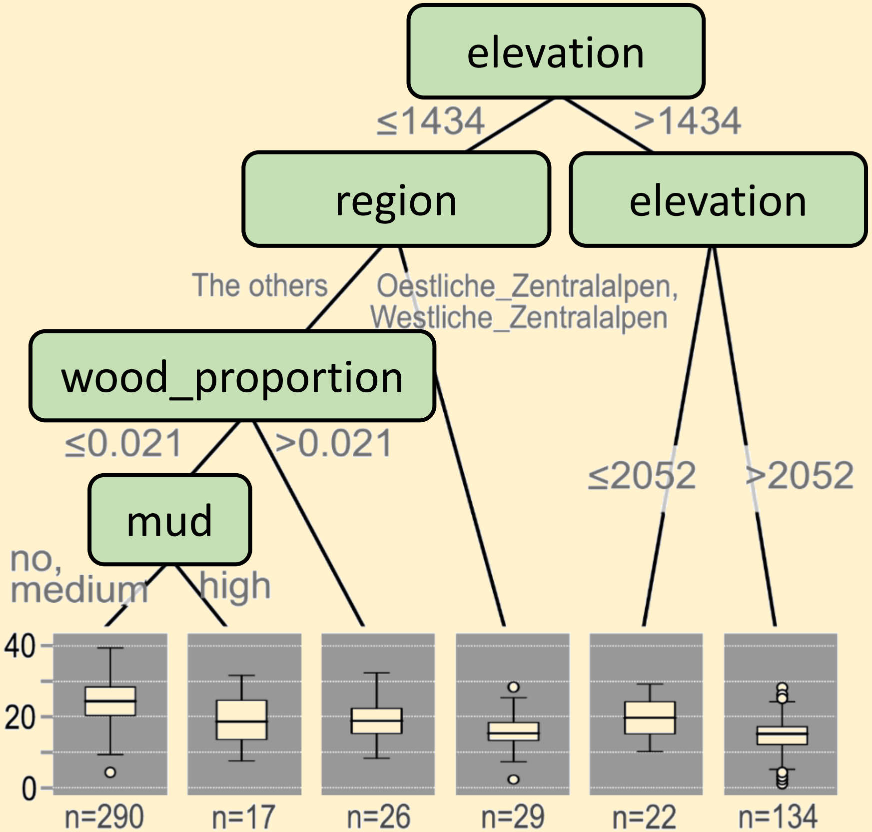
1) Find a predictor & a threshold value which separate the data into two the most distinctively.
   ⇒ *If "elevation" is > 1434 m or not*

2) For each of the separated data, repeat.
   ⇒ *If "elevation" is > 2052 m or not*
   (for right-hand side)

3) Stop separation when a set of rules are achieved (i.e. no more improvement).

Decision tree Classification and Regression Tree (CART)

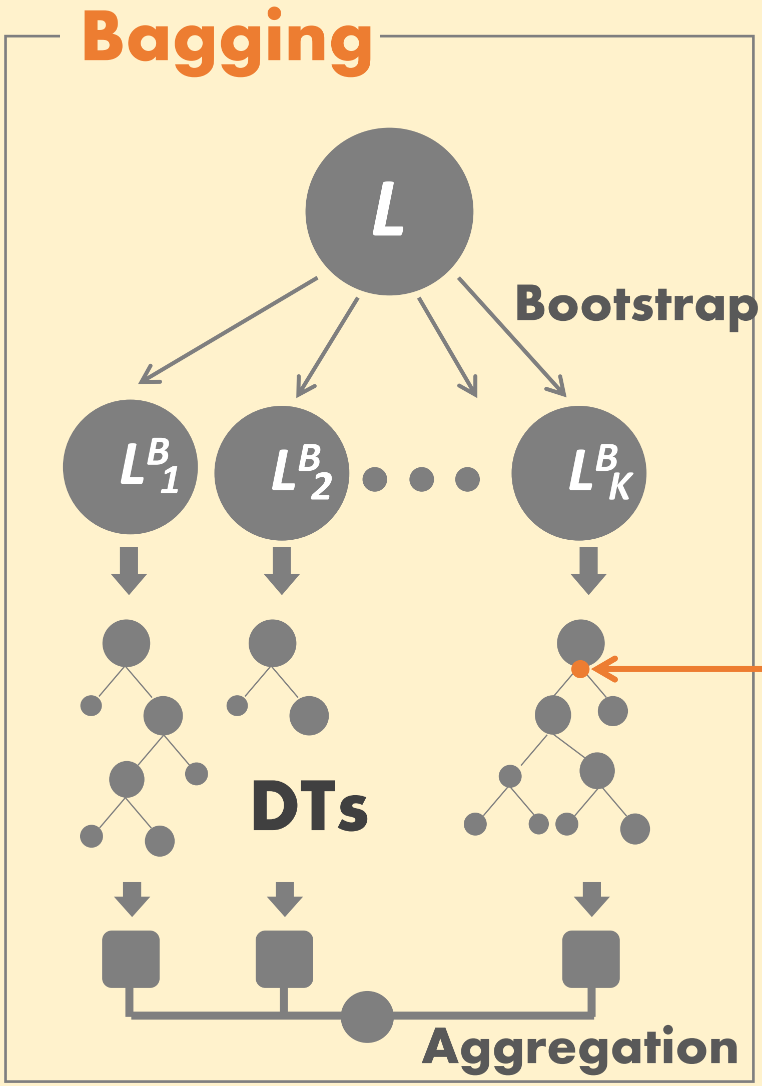
(Breiman et al. 1984)

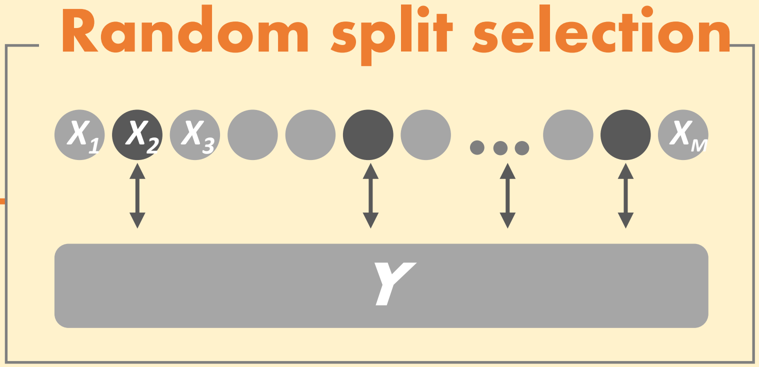


From decision tree to random forests

Random forests model ensemble approach

(Breiman 2001)



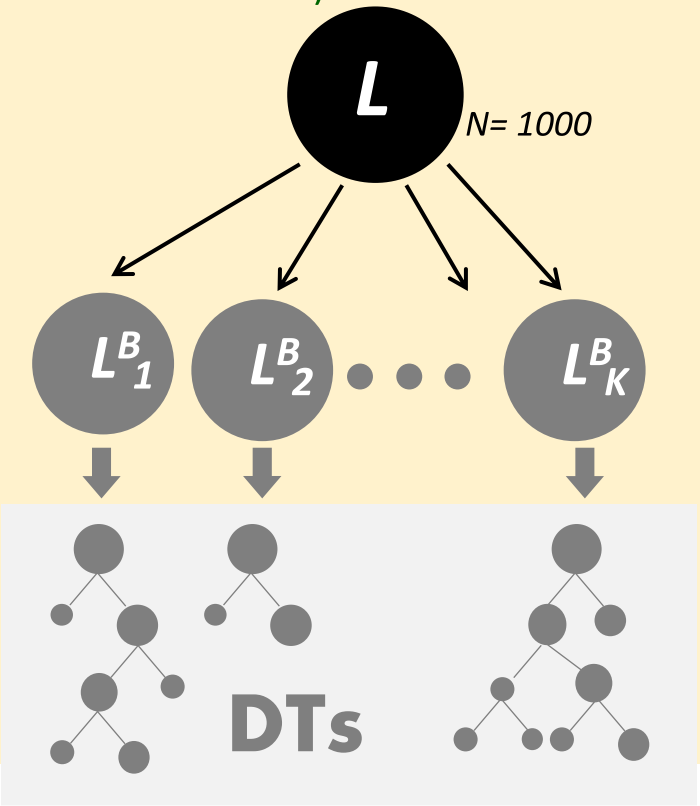


Random forests model ensemble approach

(Breiman 2001)

Bootstrap

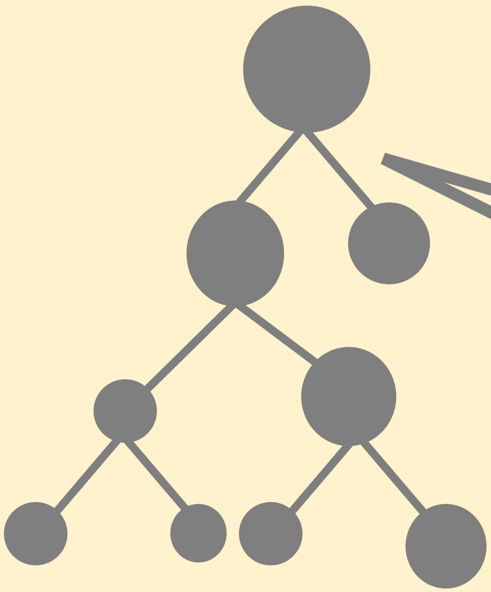
- To generate many replicates ***L**^{B}* from the original dataset ***L***
- Each consisting of *N* cases, drawn at **RANDOM**, but with replacement
  (ca. **63.2%** of the original data is chosen)



Efron and Tibshirani, 1993

Random forests model ensemble approach

(Breiman 2001)



At each node…

- Do not compare all predictor variables
- But RANDOMLY pick up some and then compare

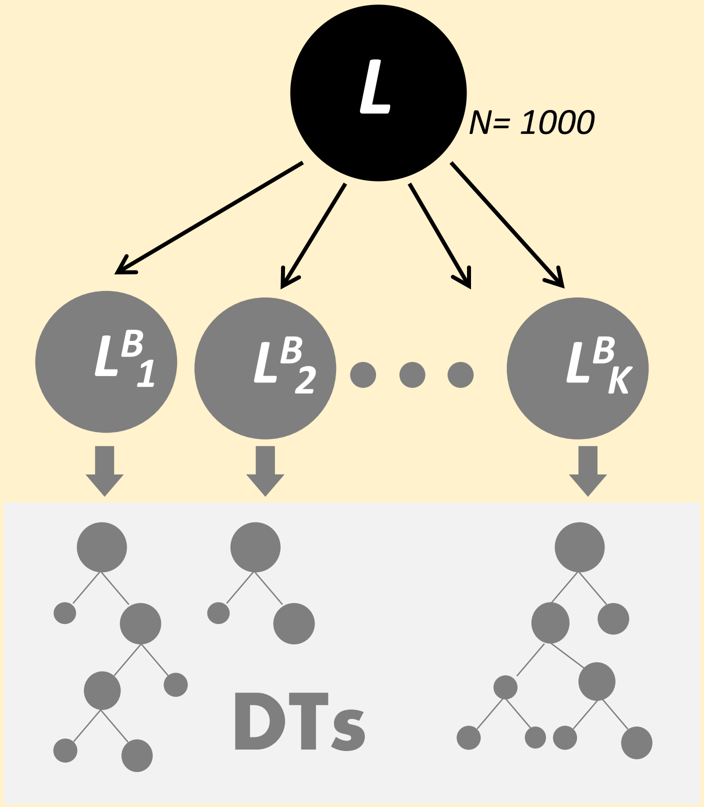
e.g.
Even though you prepare 80 predictor variables, only a handful of those are compared.

Dietterich, 2000

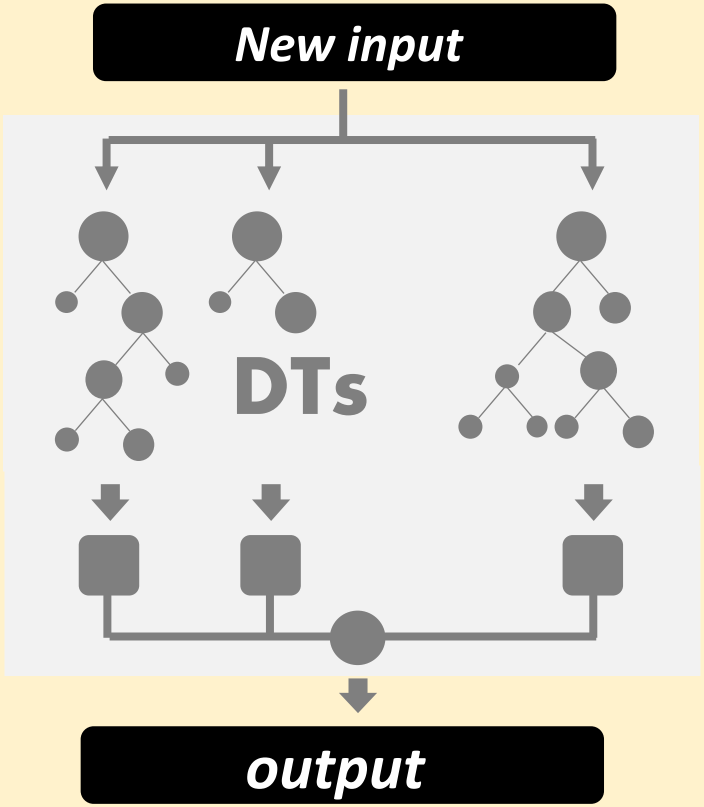
Random forests model ensemble approach

(Breiman 2001)

Step 1: building models



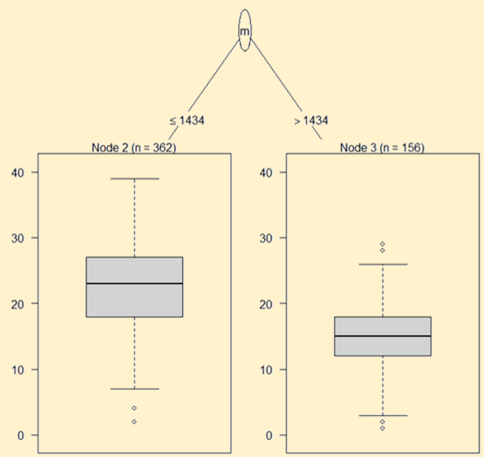
Step 2: running models

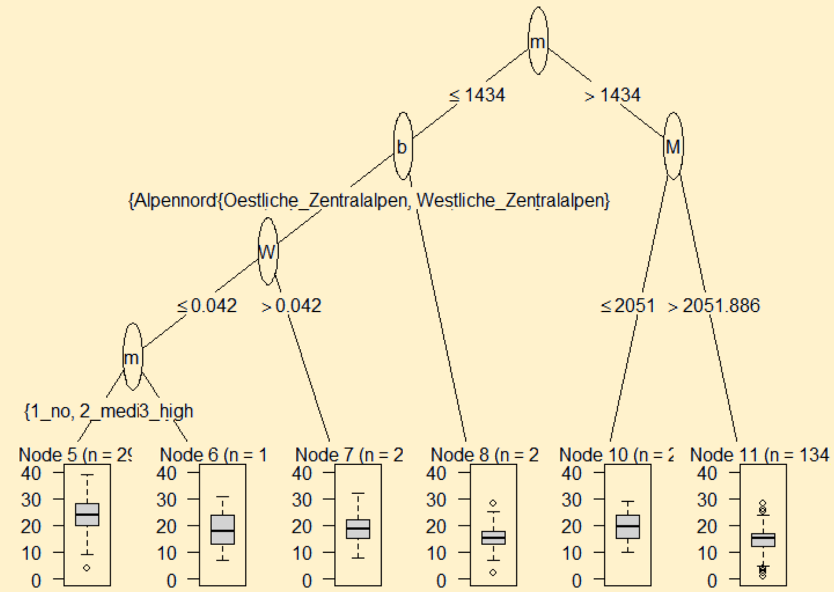


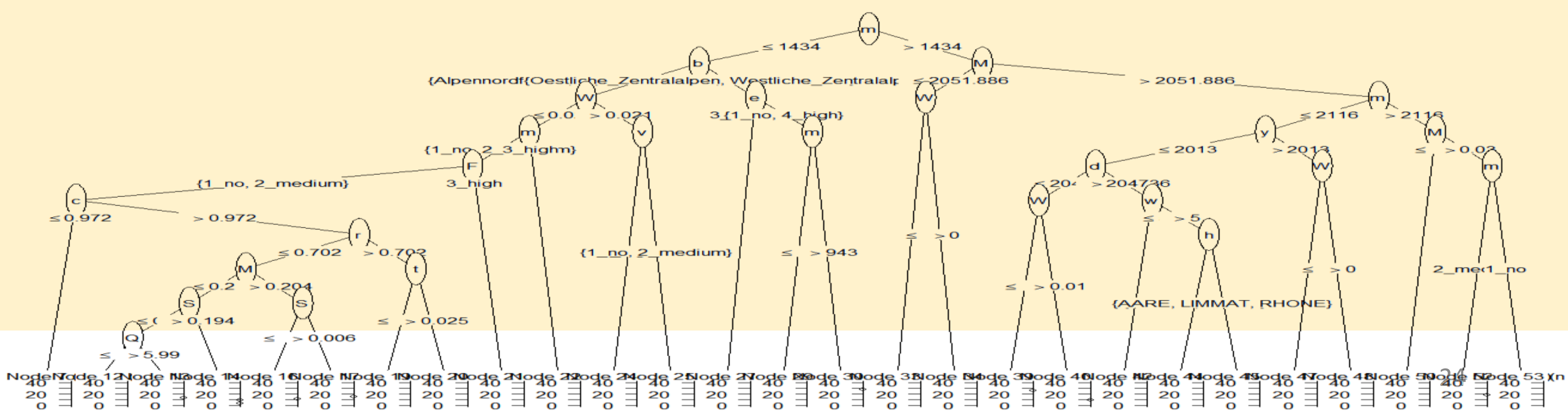
Breiman, *Machine Learning*, 1996

Two problems of decision tree algorithms

Over-fitting problem







Variable selection bias

Preferential order

Binary < categorical < continuous

Random forests?

Biased estimation on relative variable importance

History

1963: Morgan & Sonquist first developed the tree model protocol

1984: Breiman radically improved

1987: Mingers et al. reported **the two problems**

1994: White & Liu proposed statistical approach to solve

1999: Strasser & Weber proposed permutation test
(several attempts exist here)

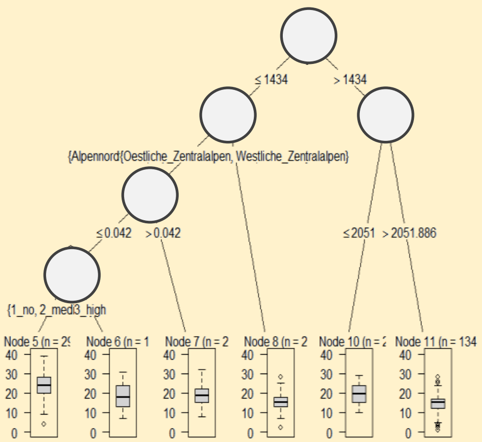
(De'ath et al. (2000) introduced it to ecology)

(2001: Breiman proposed Random forests)

2006: Hothorn et al. solved the problem

Statistically-reinforced decision trees

Conditional inference tree by Hothorn et al. (2006)



1. Estimate **p-values** for all covariates **x** based on permutation

(p-value of test statistic: χ^2 & t)

Test type & test statistic		**Covariate x**	
		categorical	numeric
Response y	categorial	CMH (χ^2)	KW (χ^2)
	numeric	KW (χ^2)	Pearson (t)

CMH: Cochran-Mantel-Haenszel, KW: Kruskal-Wallis

2. Choose the covariate **x_*** with minimum p-value;

stop if no covariates fall below significance level (α)

(with Bonferroni correction)

3. Find the value of the covariate x_* which best splits the sample into two subsamples and split

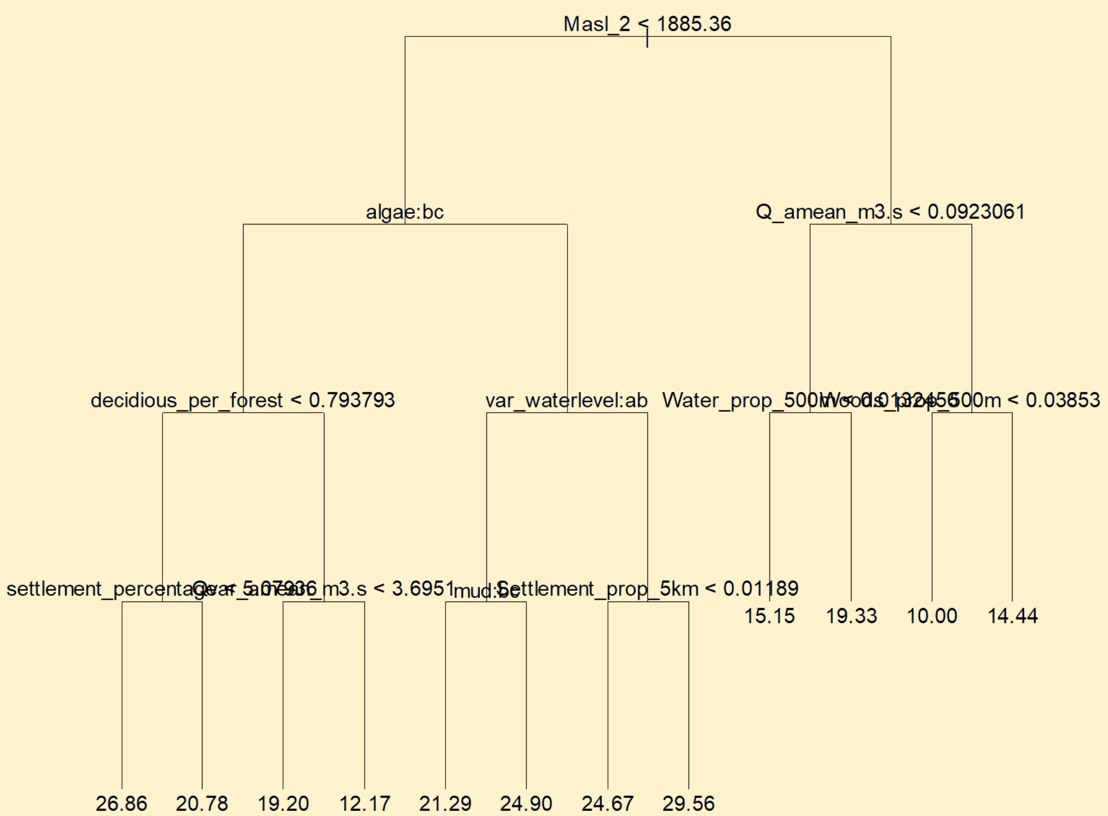
(entropy or MSE)

4. Repeat steps 1-3 until being stopped

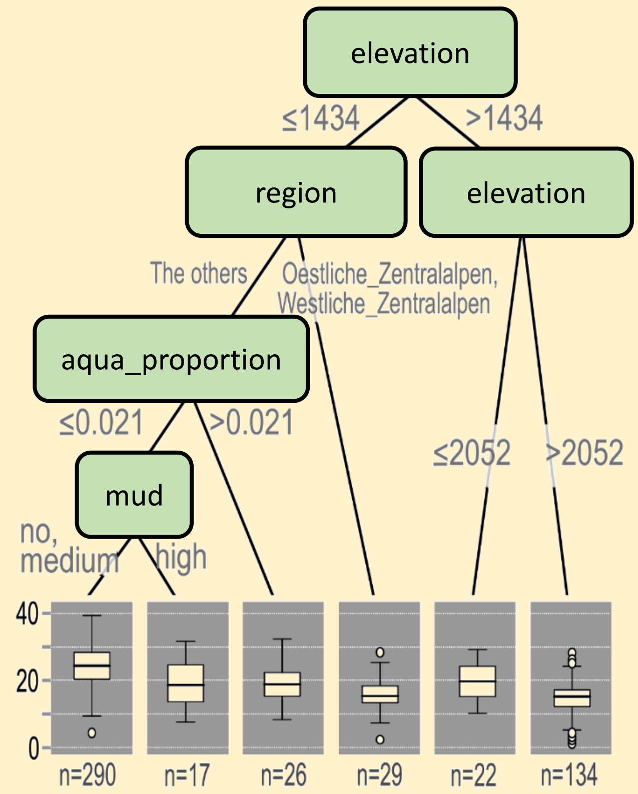
Statistically-reinforced decision trees

Conditional inference tree by Hothorn et al. (2006)

conventional tree

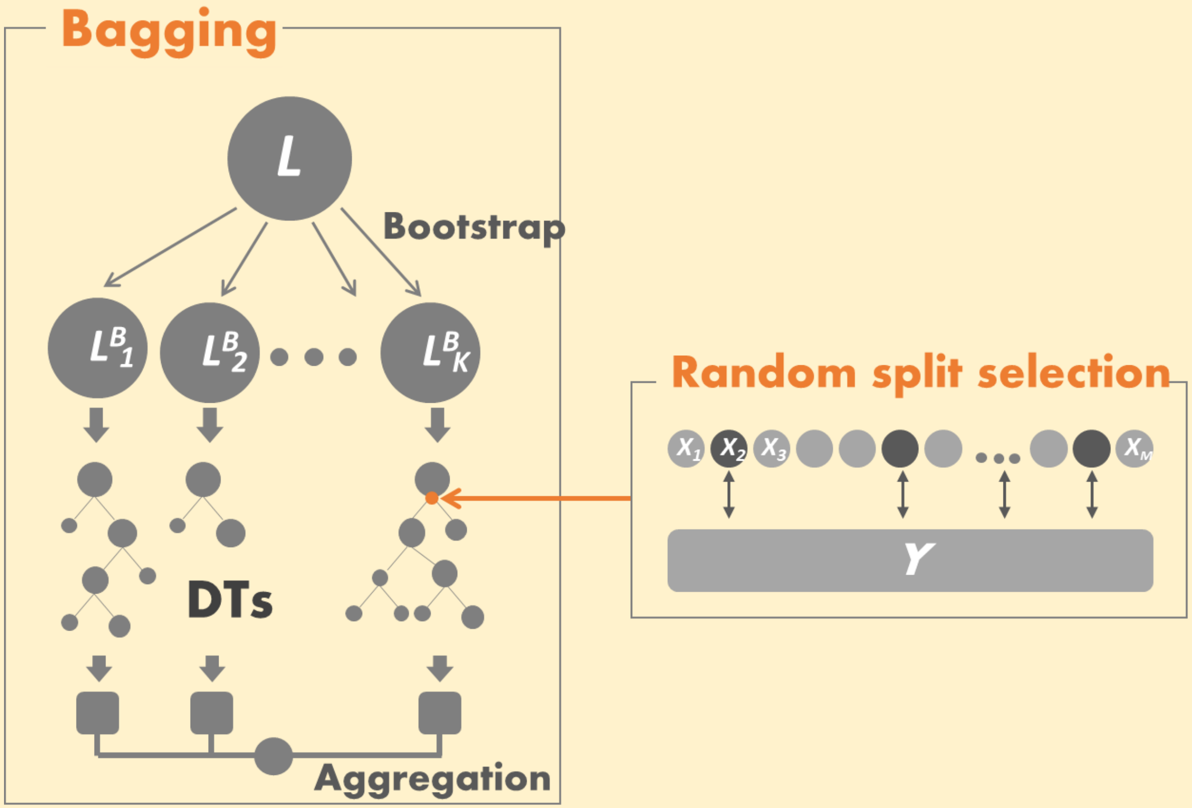


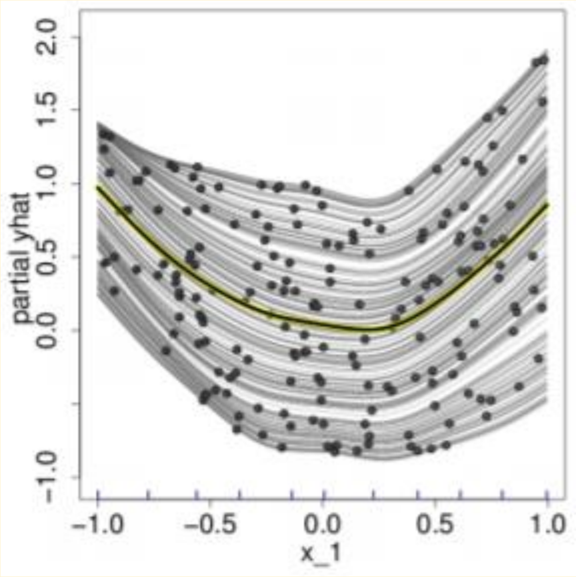
p < 0.01 tree

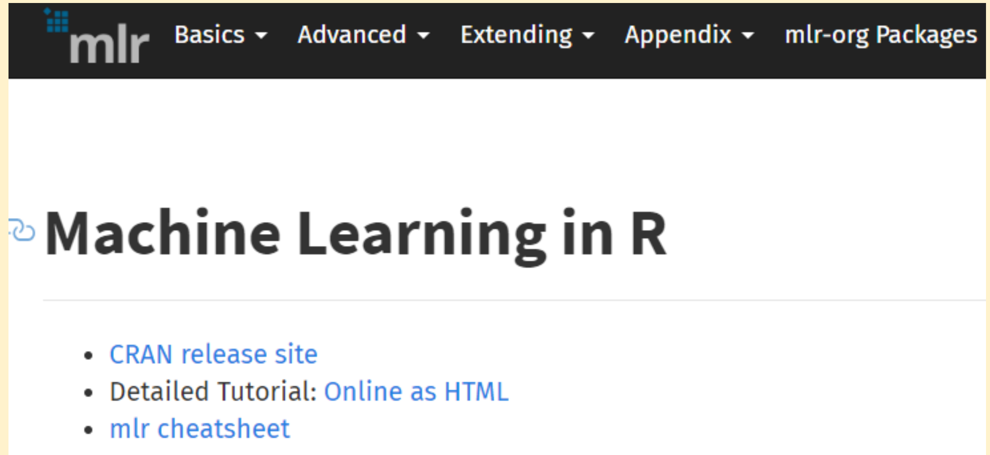


Statistically-reinforced random forests

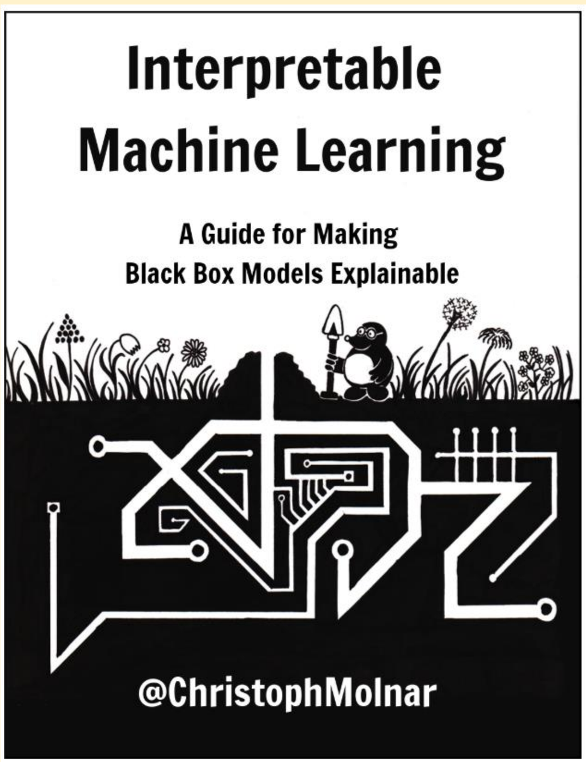
Conditional random forest by Strobl et al. (2008)







https://mlr.mlr-org.com/



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