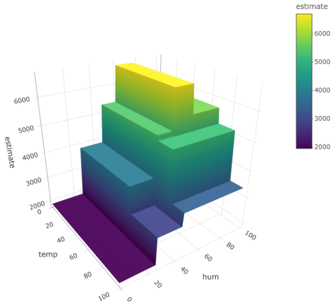
Interpretable Machine Learning

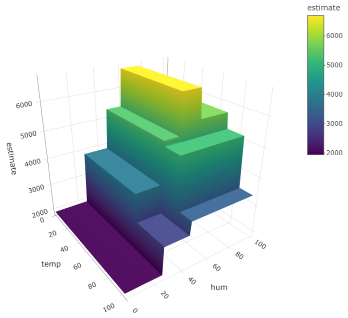
Rule-based Models



Learning goals

- Decision trees
- RuleFit
- Decision rules

Interpretable Machine Learning
Rule-based Models



Learning goals
- Decision trees
- RuleFit
- Decision rules

DECISION TREES ▸ Breiman et al. (1984)

Idea: Partition data into axis-aligned regions via greedy search for feature cut points (minimizing a split criterion), then predict a constant mean c_m in each leaf region $\mathcal{R}_m$:

$$\hat{f}(x) = \sum_{m=1}^{M} c_m \mathbb{1}_{\{x \in \mathcal{R}_m\}}$$

DECISION TREES ▸ BREIMAN

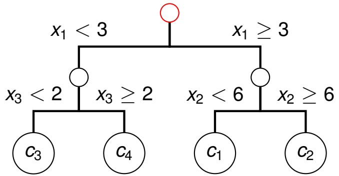
Idea: Partition data into axis-aligned regions via greedy search for feature cut points (minimizing a split criterion), then predict a constant mean c_m in each leaf region $\mathcal{R}_m$:

$$\hat{f}(x) = \sum_{m=1}^{M} c_m \mathbb{1}_{\{x \in \mathcal{R}_m\}}$$

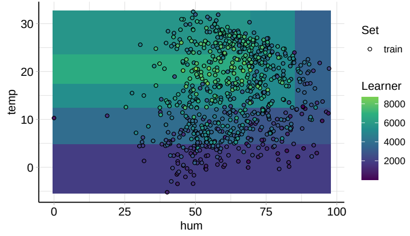
DECISION TREES ▸ Breiman et al. (1984)

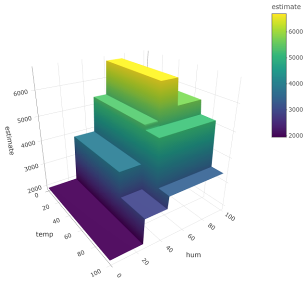
Idea: Partition data into axis-aligned regions via greedy search for feature cut points (minimizing a split criterion), then predict a constant mean c_m in each leaf region $\mathcal{R}_m$:

$$\hat{f}(x) = \sum_{m=1}^{M} c_m \mathbb{1}_{\{x \in \mathcal{R}_m\}}$$



- Applicable to regression and classification
- Models interactions and non-linear effects
- Handles mixed feature spaces & missing values

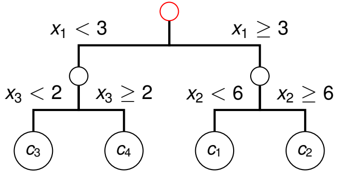




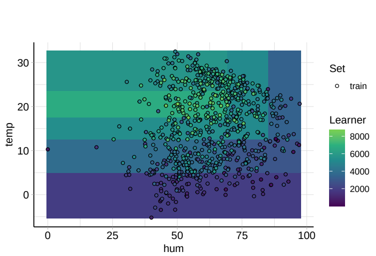
DECISION TREES ▸ BREIMAN

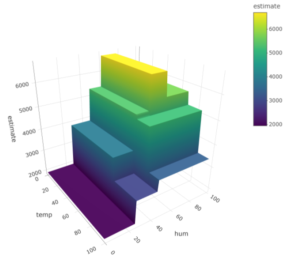
Idea: Partition data into axis-aligned regions via greedy search for feature cut points (minimizing a split criterion), then predict a constant mean c_m in each leaf region $\mathcal{R}_m$:

$$\hat{f}(x) = \sum_{m=1}^{M} c_m \mathbb{1}_{\{x \in \mathcal{R}_m\}}$$

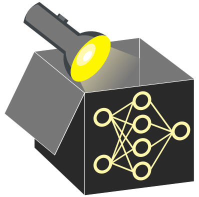


- Applicable to regression and classification
- Models interactions and non-linear effects
- Handles mixed feat, spaces & missing values

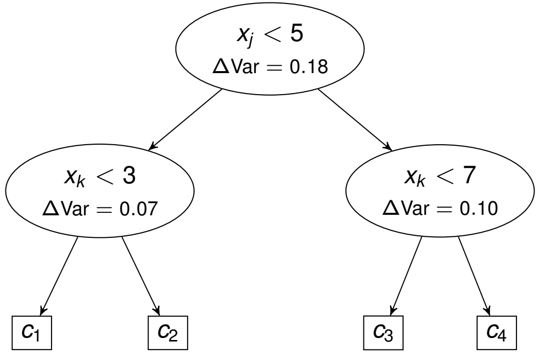




INTERPRETATION OF TREE-BASED MODELS



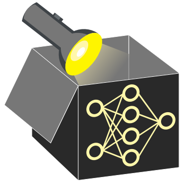
- Interpretation via path of decision rules along tree branches
- **Feature importance** (quantifies how often and how usefully x_j is used):
  - For each split on feature x_j, record the decrease in the split criterion
  - Aggregate this over the tree: sum or average over all splits involving x_j
  - Split criterion: variance (regression), Gini index / entropy (classification)



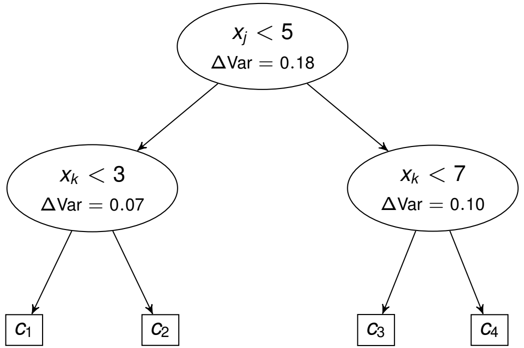
- Each ΔVar is assigned to the splitting feature
- Feature importance = sum of all ΔVar for that feature:

  x_j: 0.18
  x_k: $0.07 + 0.10 = 0.17$

INTERPRETATION OF TREE-BASED MODELS



- Interpretation via path of decision rules along tree branches
- **Feature importance** (quantifies how often and how usefully x_j is used):
  - For each split on feature x_j, record the decrease in the split criterion
  - Aggregate this over the tree: sum or avg. over all splits involving x_j
  - Split criterion: variance (regression), Gini index / entropy (classif.)



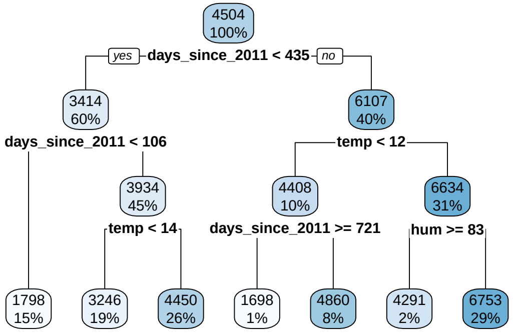
- Each ΔVar is assigned to the splitting feature
- Feature importance = sum of all ΔVar for that feat.:

  x_j: 0.18
  x_k: $0.07 + 0.10 = 0.17$

DECISION TREES - EXAMPLE

- Fit decision tree with tree depth of 3 on bike data
- E.g., mean prediction for the first 105 days since 2011 is 1798
  $\rightsquigarrow$ Applies to $\widehat{=}$15% of the data (leftmost branch)
- `days_since_2011`: highest feature importance (explains most of variance)

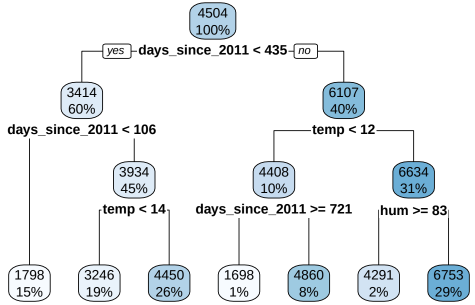
Feature	Importance
days_since_2011	79.53
temp	17.55
hum	2.92



DECISION TREES - EXAMPLE

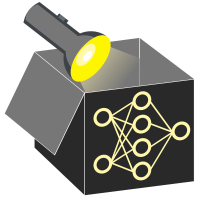
- Fit decision tree with tree depth of 3 on bike data
- E.g., mean prediction for the first 105 days since 2011 is 1798
  $\rightsquigarrow$ Applies to $\widehat{=}$15% of the data (leftmost branch)
- `days_since_2011`: highest feat. importance (explains most of variance)

Feature	Importance
days_since_2011	79.53
temp	17.55
hum	2.92



UNBIASED RECURSIVE PARTITIONING

▸ Hothorn et al. (2006) ▸ Zeileis et al. (2008) ▸ Strobl et al. (2007)

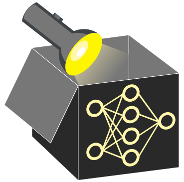


Problems with CART (Classification and Regression Trees):

1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance $\rightsquigarrow$ prone to overfitting

UNBIASED RECURSIVE PARTITIONING

▸ Hothorn 2006 ▸ Zeileis 2008 ▸ Strobl 2007

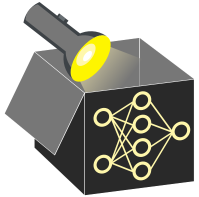


Problems with CART (Classification and Regression Trees):

1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance $\rightsquigarrow$ prone to overfitting

UNBIASED RECURSIVE PARTITIONING

▸ Hothorn et al. (2006) ▸ Zeileis et al. (2008) ▸ Strobl et al. (2007)



Problems with CART (Classification and Regression Trees):

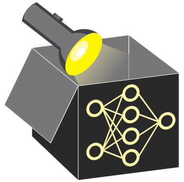
1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance ⇝ prone to overfitting

Unbiased recursive partitioning via conditional inference trees (`ctree`) or model-based recursive partitioning (`mob`):

1. Separate selection of **feature used for splitting** and **split point**
2. Hypothesis test as stopping criteria

UNBIASED RECURSIVE PARTITIONING

▸ Hothorn 2006 ▸ Zeileis 2008 ▸ Strobl 2007



Problems with CART (Classification and Regression Trees):

1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance ⇝ prone to overfitting

Unbiased recursive partitioning via conditional inference trees (`ctree`) or model-based recursive partitioning (`mob`):

1. Separate selection of **feature used for splitting** and **split point**
2. Hypothesis test as stopping criteria

UNBIASED RECURSIVE PARTITIONING

▸ Hothorn et al. (2006) ▸ Zeileis et al. (2008) ▸ Strobl et al. (2007)

Problems with CART (Classification and Regression Trees):

1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance $\rightsquigarrow$ prone to overfitting

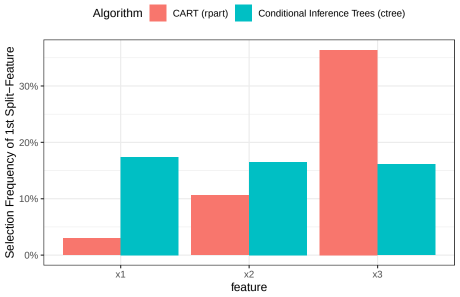
Unbiased recursive partitioning via conditional inference trees (`ctree`) or model-based recursive partitioning (`mob`):

1. Separate selection of **feature used for splitting** and **split point**
2. Hypothesis test as stopping criteria

Example (selection bias):
Simulate data ($n = 200$) with $Y \sim N(0, 1)$ and 3 features of different cardinality independent from Y (repeat 500 times):

- $X_1 \sim Binom(n, \frac{1}{2})$
- $X_2 \sim M(n, (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}))$
- $X_3 \sim M(n, (\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}))$

Which feature is selected in the first split?



UNBIASED RECURSIVE PARTITIONING

▸ Hothorn 2006 ▸ Zeileis 2008 ▸ Strobl 2007

Problems with CART (Classification and Regression Trees):

1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance $\rightsquigarrow$ prone to overfitting

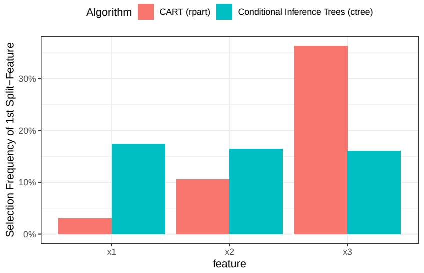
Unbiased recursive partitioning via conditional inference trees (`ctree`) or model-based recursive partitioning (`mob`):

1. Separate selection of **feature used for splitting** and **split point**
2. Hypothesis test as stopping criteria

Example (selection bias):
Simulate data ($n = 200$), $Y \sim N(0, 1)$ and 3 features of different cardinality indep. from Y (repeat 500 times):

- $X_1 \sim Binom(n, \frac{1}{2})$
- $X_2 \sim M(n, (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}))$
- $X_3 \sim M(n, (\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}))$

Which feature is selected in the first split?



UNBIASED RECURSIVE PARTITIONING

Differences to CART:

- Two-step approach (1. find most significant split feature, 2. find best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leave nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point

UNBIASED RECURSIVE PARTITIONING

Differences to CART:

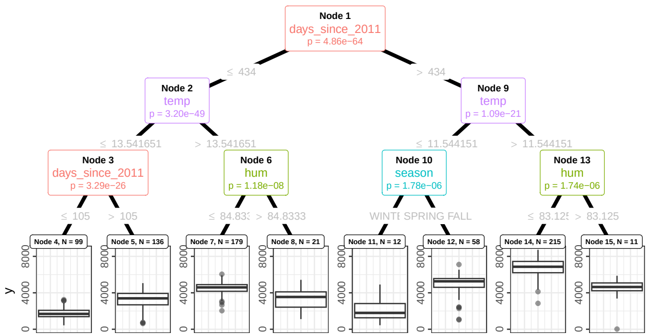
- Two-step approach (finds 1. most significant split feat., 2. best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leaf nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point

UNBIASED RECURSIVE PARTITIONING

Differences to CART:

- Two-step approach (1. find most significant split feature, 2. find best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leave nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point

Example (`ctree`): Bike data (constant model in final nodes)



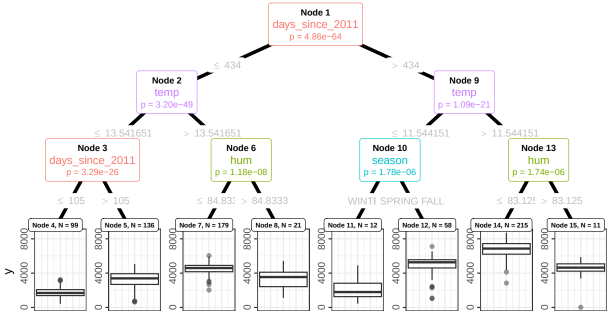
Train error (MSE):
758,844.0 (`ctree`)
742,244.4 (`mob`)

UNBIASED RECURSIVE PARTITIONING

Differences to CART:

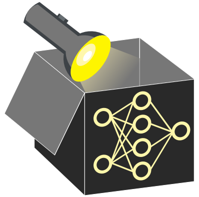
- Two-step approach (finds 1. most significant split feat., 2. best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leaf nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point

Example (`ctree`): Bike data (constant model in final nodes)



Train MSE:
758,844 (`ctree`)
742,244 (`mob`)

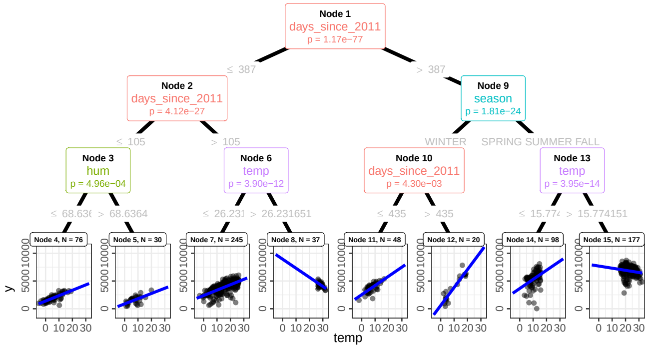
UNBIASED RECURSIVE PARTITIONING



Differences to CART:

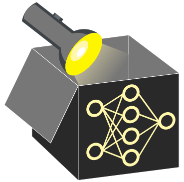
- Two-step approach (1. find most significant split feature, 2. find best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leave nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point

Example (`mob`): Bike data (linear model with `temp` in final nodes)



Train error (MSE):
758,844.0 (`ctree`)
742,244.4 (`mob`)

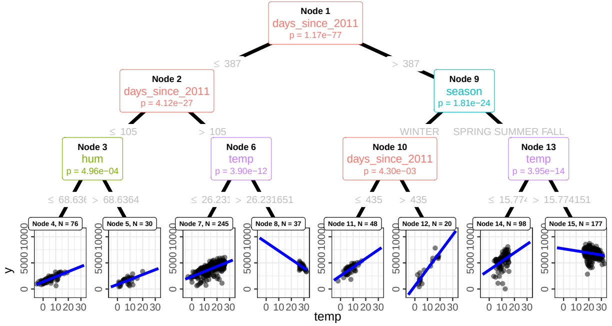
UNBIASED RECURSIVE PARTITIONING



Differences to CART:

- Two-step approach (finds 1. most significant split feat., 2. best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leaf nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point

Example (`mob`): Bike data (linear model with `temp` in final nodes)



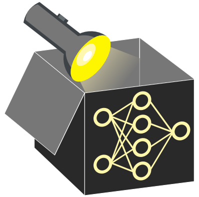
Train MSE:
758,844 (`ctree`)
742,244 (`mob`)

OTHER RULE-BASED MODELS

Decision Rules ‣ Holte 1993

- Flat list of simple "if – then" statements
  ⇝ very intuitive and easy-to-interpret
- Mainly devised for classification
  (support for regression is limited)
- Numeric features are typically discretised

IF $x_1 \leq 2.3$ AND $x_4 =$ "A"	THEN	y = 1
ELSE IF $x_2 > 5.0$	THEN	y = 2
ELSE		y = 3

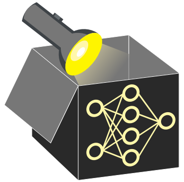


OTHER RULE-BASED MODELS

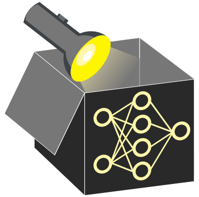
Decision Rules ‣ Holte 1993

- Flat list of simple "if – then" statements
  ⇝ very intuitive and easy-to-interpret
- Mainly devised for classification
  (support for regression is limited)
- Numeric features are typically discretised

IF $x_1 \leq 2.3$ AND $x_4 =$ "A"	THEN	y = 1
ELSE IF $x_2 > 5.0$	THEN	y = 2
ELSE		y = 3



OTHER RULE-BASED MODELS



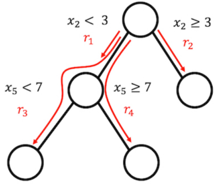
Decision Rules ▸ Holte 1993

- Flat list of simple "if – then" statements
  ⇝ very intuitive and easy-to-interpret
- Mainly devised for classification
  (support for regression is limited)
- Numeric features are typically discretised

IF $x_1 \leq 2.3$ AND $x_4 =$ "A" THEN y = 1
ELSE IF $x_2 > 5.0$ THEN y = 2
ELSE y = 3

RuleFit ▸ Friedman & Popescu 2008

- Extract binary rules $r_m(\mathbf{x}) \in \{0, 1\}$ from many shallow trees (one per root-to-leaf path)
- Fit an L_1-regularized LM
  $\hat{f}(\mathbf{x}) = \beta_0 + \sum_m \beta_m r_m(\mathbf{x}) + \sum_j \gamma_j x_j$
- Regularization retains only a few rules
  ⇒ sparse, non-linear, interaction-aware
- Coefficients relate to rule/feature importance



▸ Molnar 2022

OTHER RULE-BASED MODELS

Decision Rules ▸ Holte 1993

- Flat list of simple "if – then" statements
  ⇝ very intuitive and easy-to-interpret
- Mainly devised for classification
  (support for regression is limited)
- Numeric features are typically discretised

IF $x_1 \leq 2.3$ AND $x_4 =$ "A" THEN y = 1
ELSE IF $x_2 > 5.0$ THEN y = 2
ELSE y = 3

RuleFit ▸ Friedman and Popescu 2008

- Extract binary rules $r_m(\mathbf{x}) \in \{0, 1\}$ from many shallow trees (one per root-to-leaf path)
- Fit an L_1-regularized LM
  $\hat{f}(\mathbf{x}) = \beta_0 + \sum_m \beta_m r_m(\mathbf{x}) + \sum_j \gamma_j x_j$
- Regularization retains only a few rules
  ⇒ sparse, non-linear, interaction-aware
- Coefficients relate to rule/feature importance

▸ Molnar 2022