

## Klassifikations und Regressionsbäume (CART)

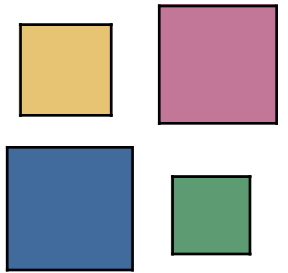
CART ist eine der einfachsten und meist benutzten Techniken in der Klassifikation.

Der CART Algorithmus generiert einen Klassifikationsbaum (Entscheidungsbaum) über binäre Aufteilungen der Daten.

Üblicher Weise werden diese Splits jeweils nur über eine Variable vorgenommen.

Bei Klassifikationsbäumen werden die Splits derart gewählt, dass die beiden Untermengen eines Splits bzgl. der Klassifikationsvariablen möglichst homogen sind.

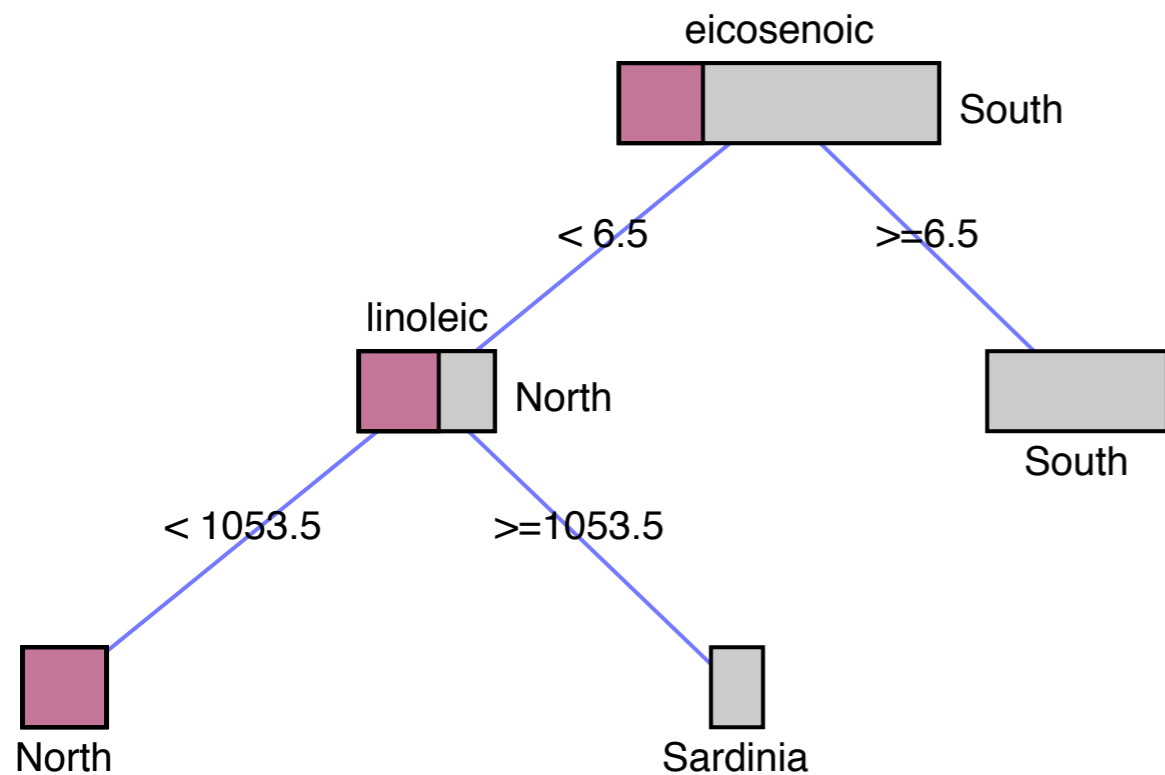
Bei Regressionsbäumen wird meist die quadratische Abweichung der Vorhersage zur Zielvariablen minimiert.



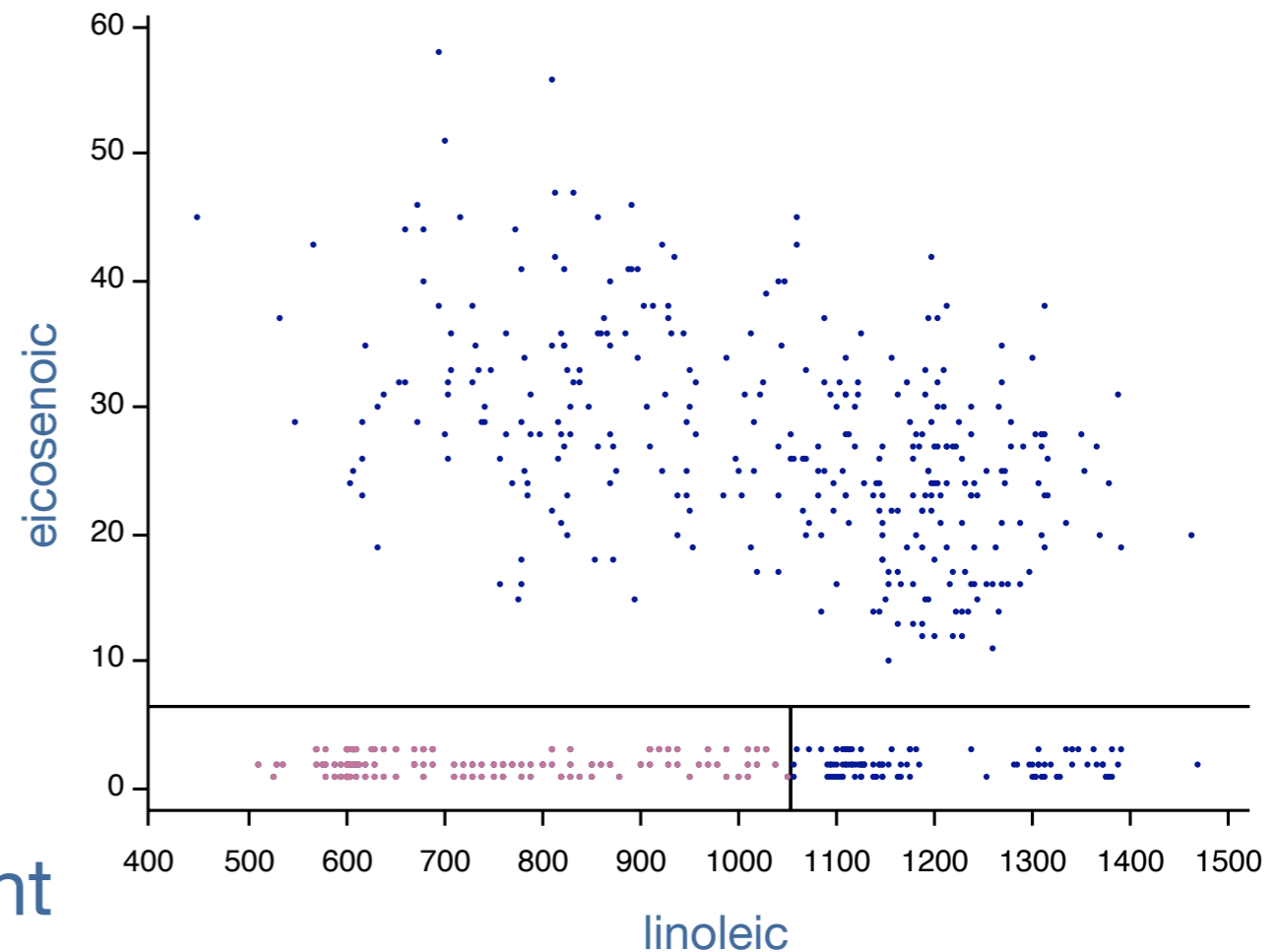
# Beispiel: Olive Oils

- Zielvariable: Region, alle 8 Fettsäuren

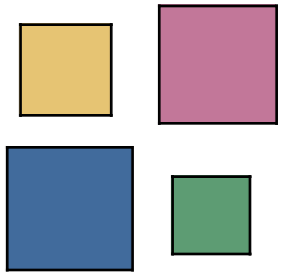
**Klassifikationsbaum**



**(sectioned) Scatterplot**

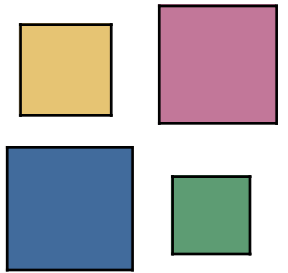


Die Beschriftung der Blätter entspricht der “Mehrheit” der Fälle in dem Blatt.



## Classification vs. Regression

- Klassifikations- und Regessionsbäume unterscheiden sich von der Baumstruktur nicht.
- Während bei Klassifikationsbäumen die Zielvariable **kategoriiell** ist, sind Zielvariablen von Regressionsbäumen **stetige** Größen.
- Somit kommt es bei Klassifikationsbäumen darauf an in einem Blatt möglichst nur noch Werte einer Gruppe der Zielvariablen zu versammeln, und bei Regressionsbäumen die Abweichung zwischen den Werten im Blatt und dem geschätzten Wert des Blattes zu minimieren.
- Das übliche Maß ist die Summe der quadratischen Abweichungen.



## Split Kriterium bei Regressionsbäumen

Für die Zielvariable  $y$ , eine potentielle Split Variable  $j$  aus  $1, \dots, p$  Variablen, und einen potentiellen Split Punkt  $s$ , definieren sich zwei Halbebenen im  $\mathbb{R}^p$ :

$$R_1(j, s) = \{X | X_j \leq s\} \quad \text{und} \quad R_2(j, s) = \{X | X_j > s\}$$

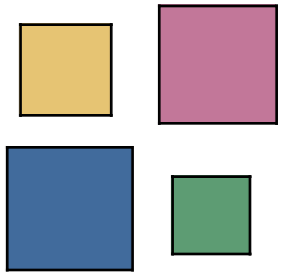
Gesuchte wird nun die Split Variable  $j$  und der Split Punkt  $s$ , die

$$\min_{j,s} \left[ \min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right]$$

erfüllen. Für beliebige  $j$  und  $s$  wird die innere Minimierung durch

$$\hat{c}_1 = \text{ave}(y_i | x_i \in R_1(j, s)) \quad \text{und} \quad \hat{c}_2 = \text{ave}(y_i | x_i \in R_2(j, s))$$

gelöst. In jedem Schritt wird also das optimale Paar  $(j, s)$  gesucht.



## Split Kriterium bei Klassifikationsbäumen

Sei nun die Zielvariable kategoriell mit Werten  $1, 2, \dots, K$ . Für eine Region  $R_m$  mit  $N_m$  Beobachtungen ist

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

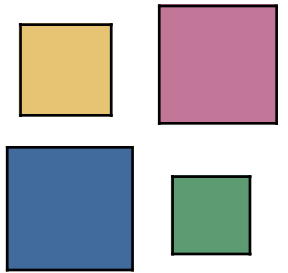
der Anteil Beobachtungen von Klasse  $k$  im Knoten  $m$ . Die Zuordnung des Knotens zur Klasse geschieht über  $k(m) = \arg \max_k \hat{p}_{mk}$ .

Übliche Maße der Unreinheit  $Q_m(T)$  der Knoten  $m$  eines Baums  $T$ :

- Missklassifikation:  $\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)}$

- Gini Index:  $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^K \hat{p}_{mk} (1 - \hat{p}_{mk})$

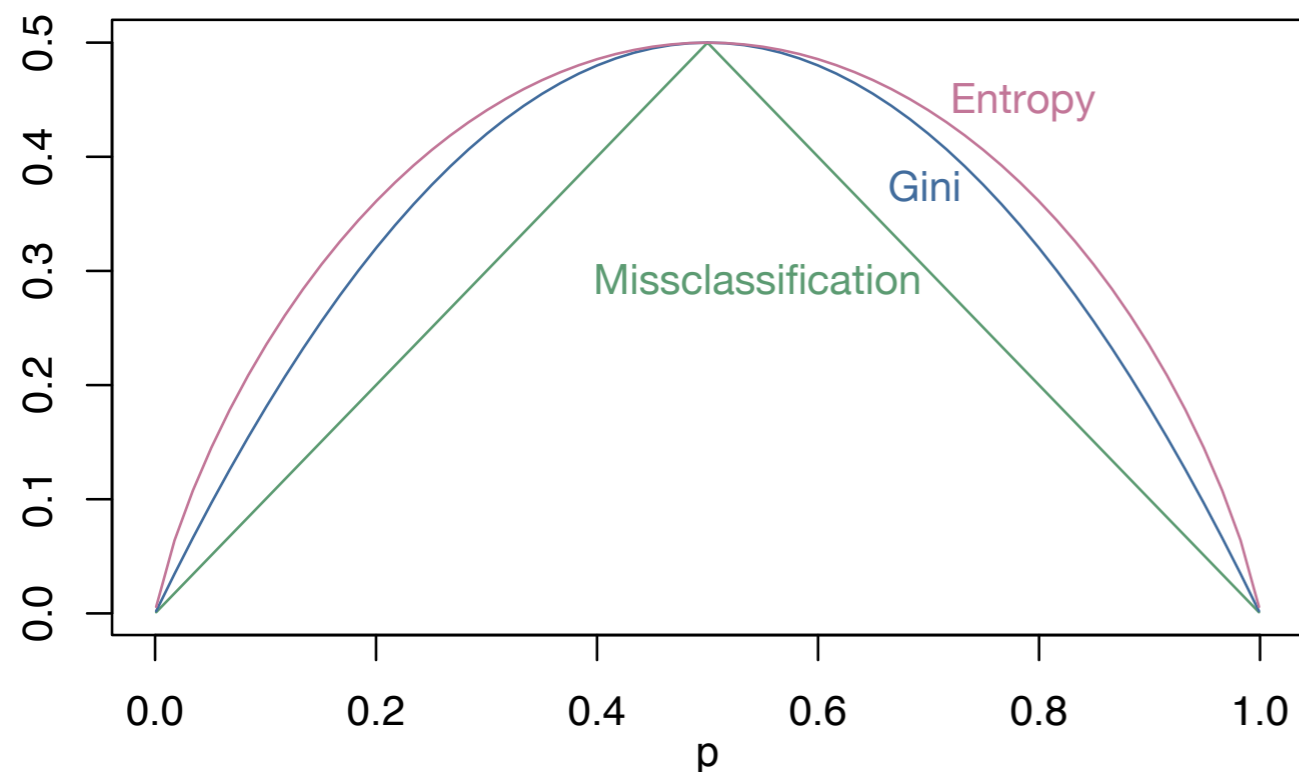
- Entropy bzw. Devianz:  $\sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk}$



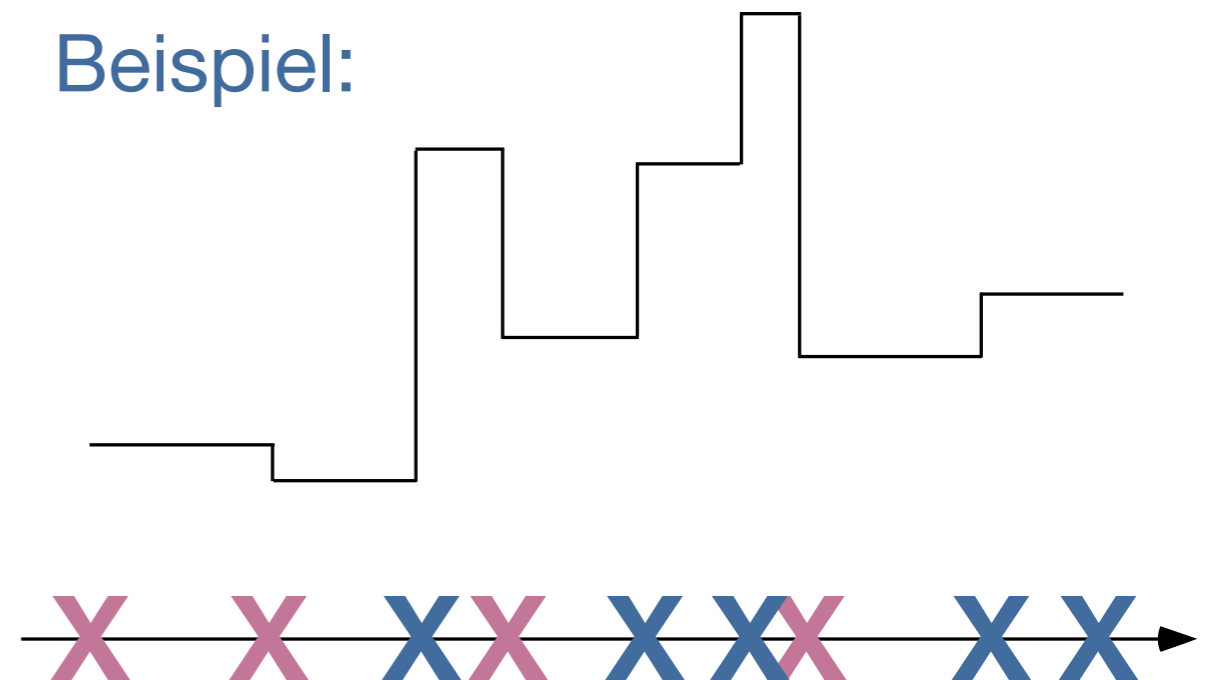
## Impurity Maße

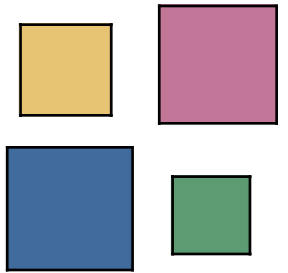
Im Fall von nur 2 Klassen ( $p$  sei der Anteil der 2. Klasse) vereinfachen sich die Maße zu:

- Missklassifikation:  $1 - \max(p, 1 - p)$
- Gini Index:  $2p(1 - p)$
- Entropy:  $-p \log p - (1 - p) \log(1 - p)$

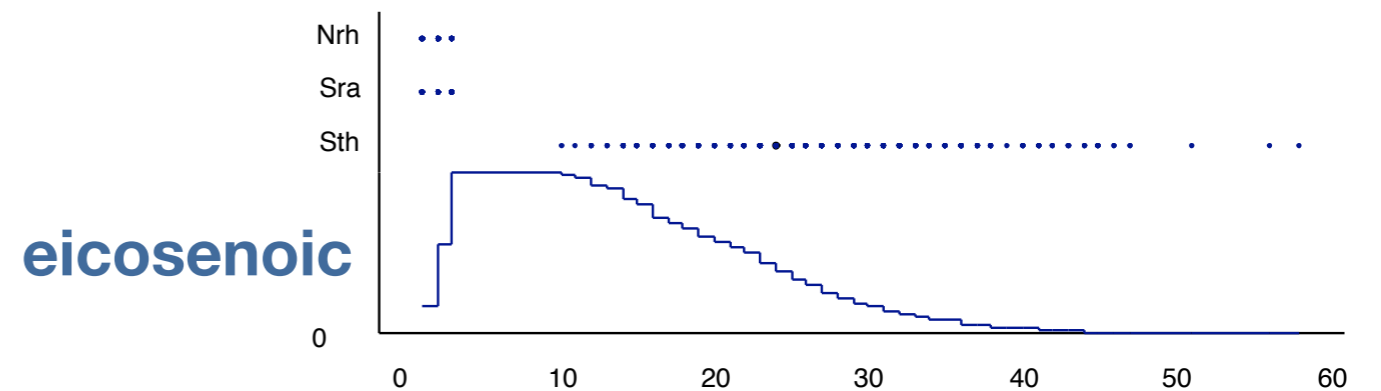
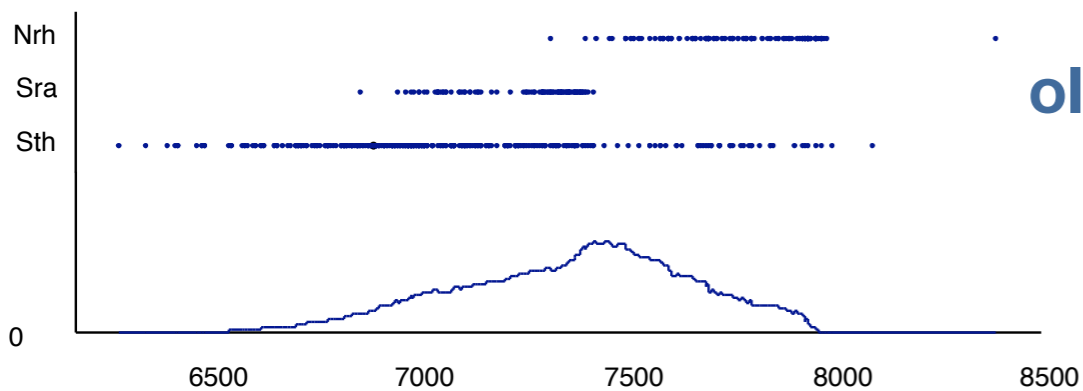
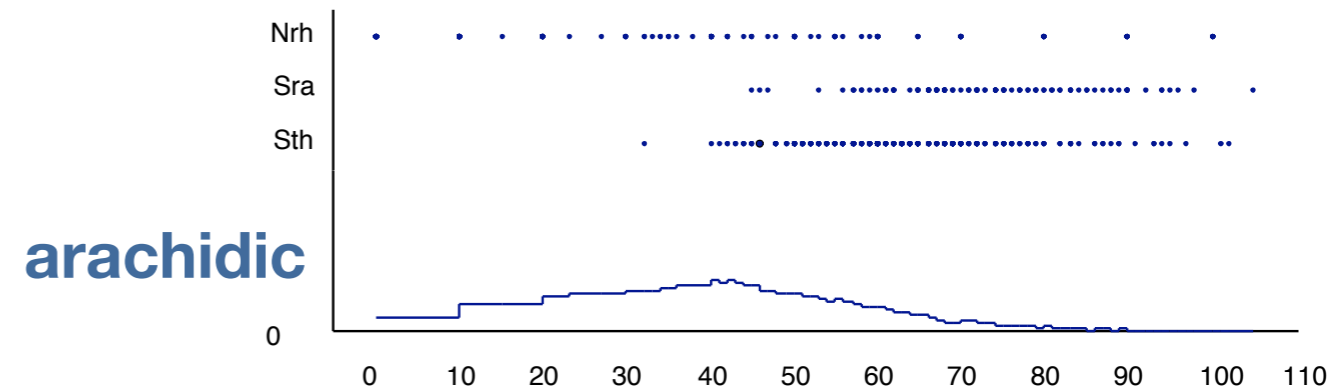
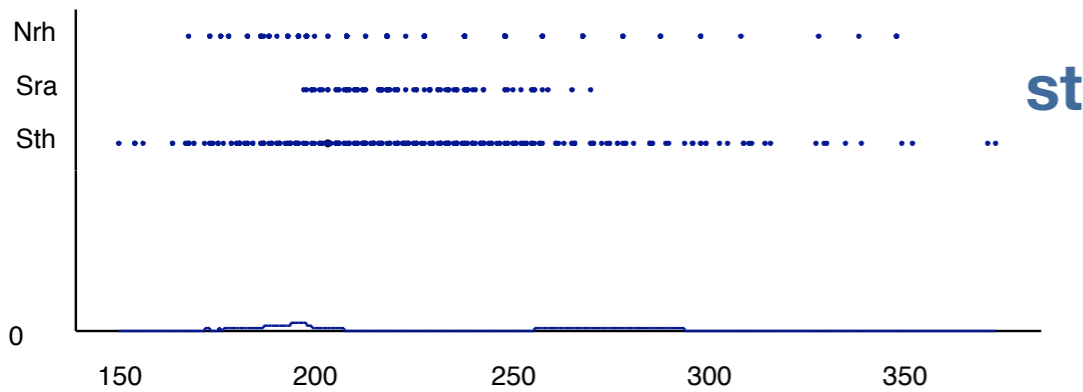
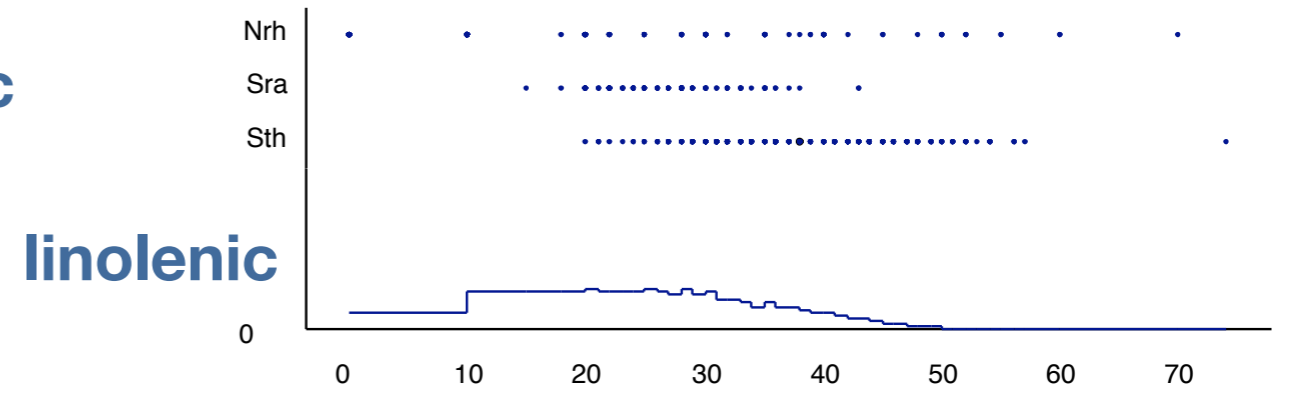
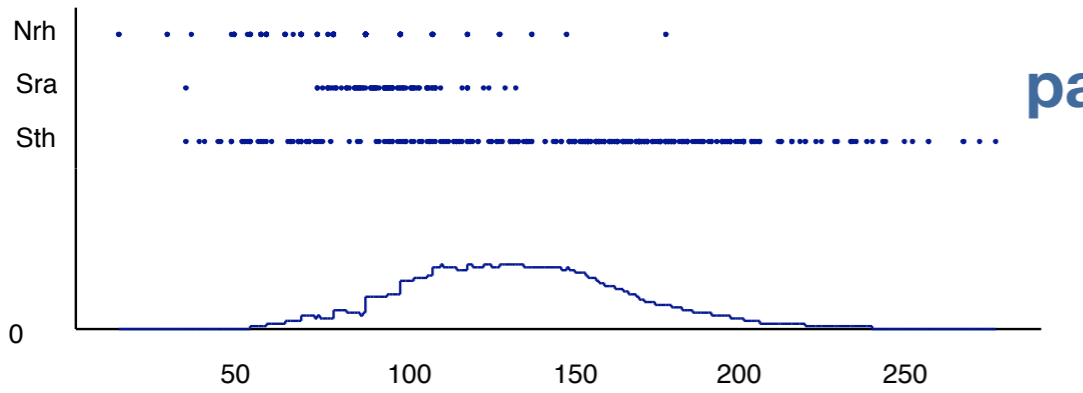
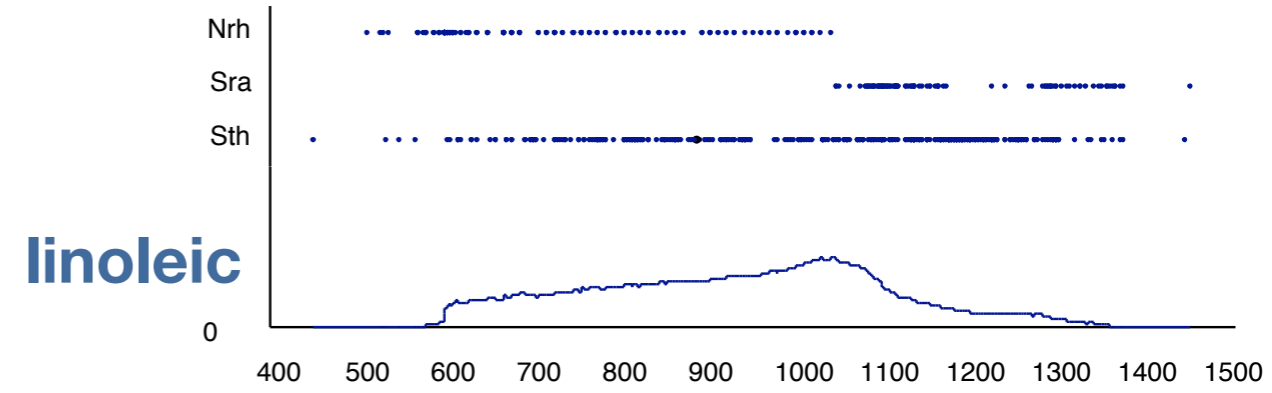
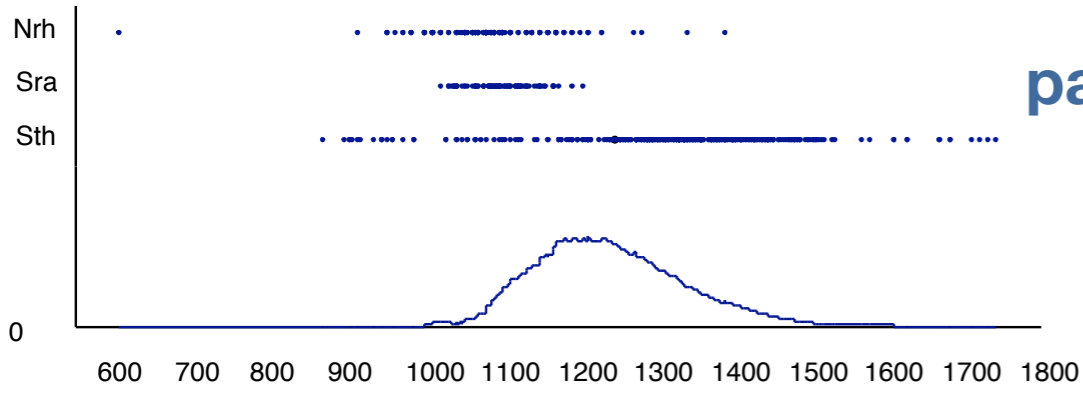


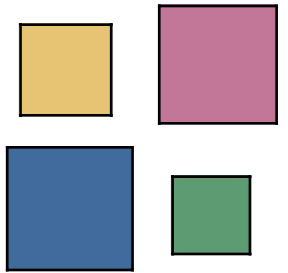
Beispiel:





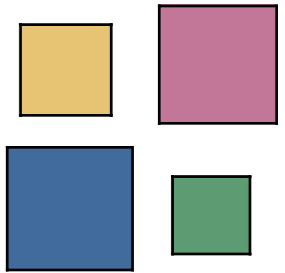
# Impurity: Olive Oils bzgl. Region





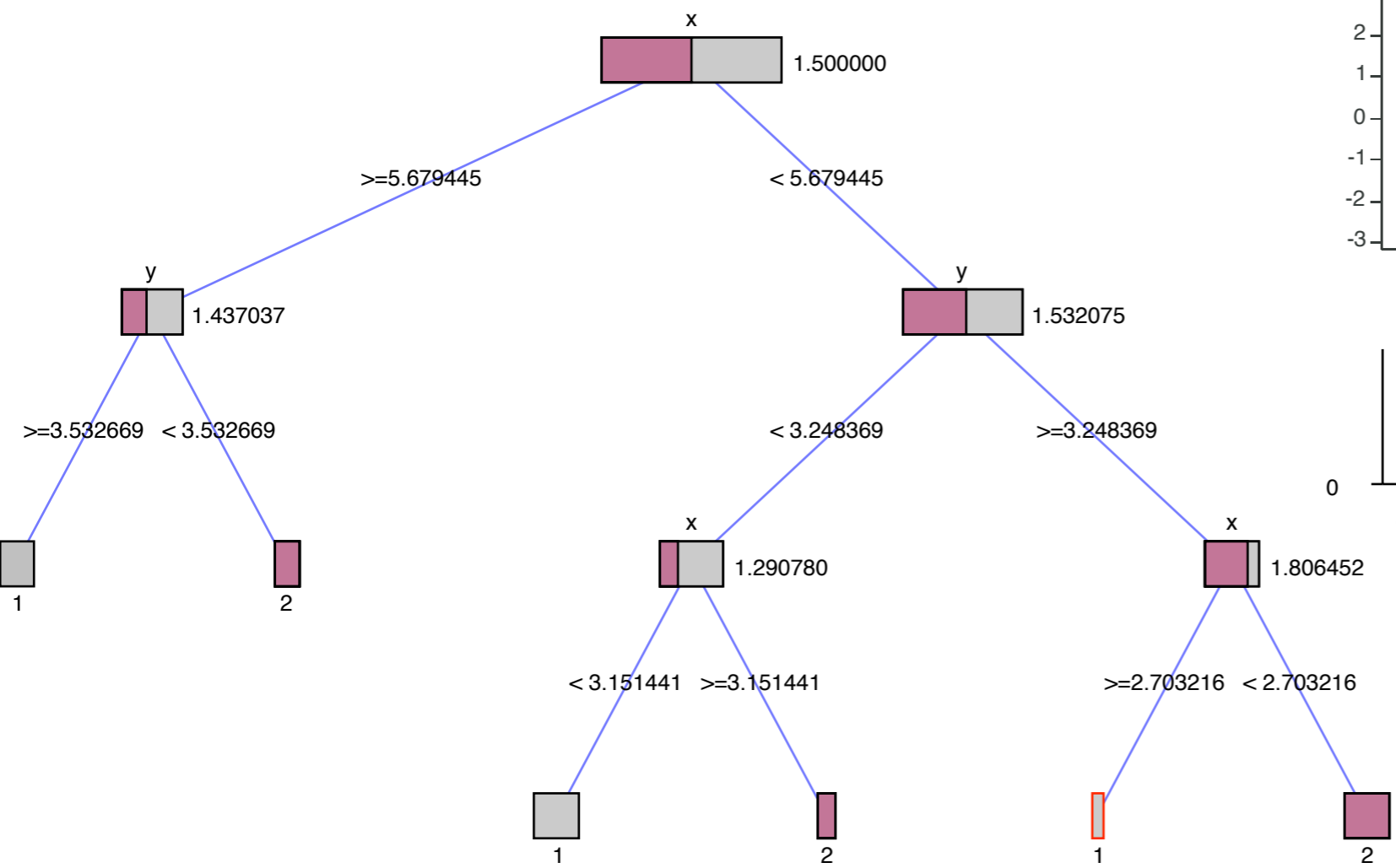
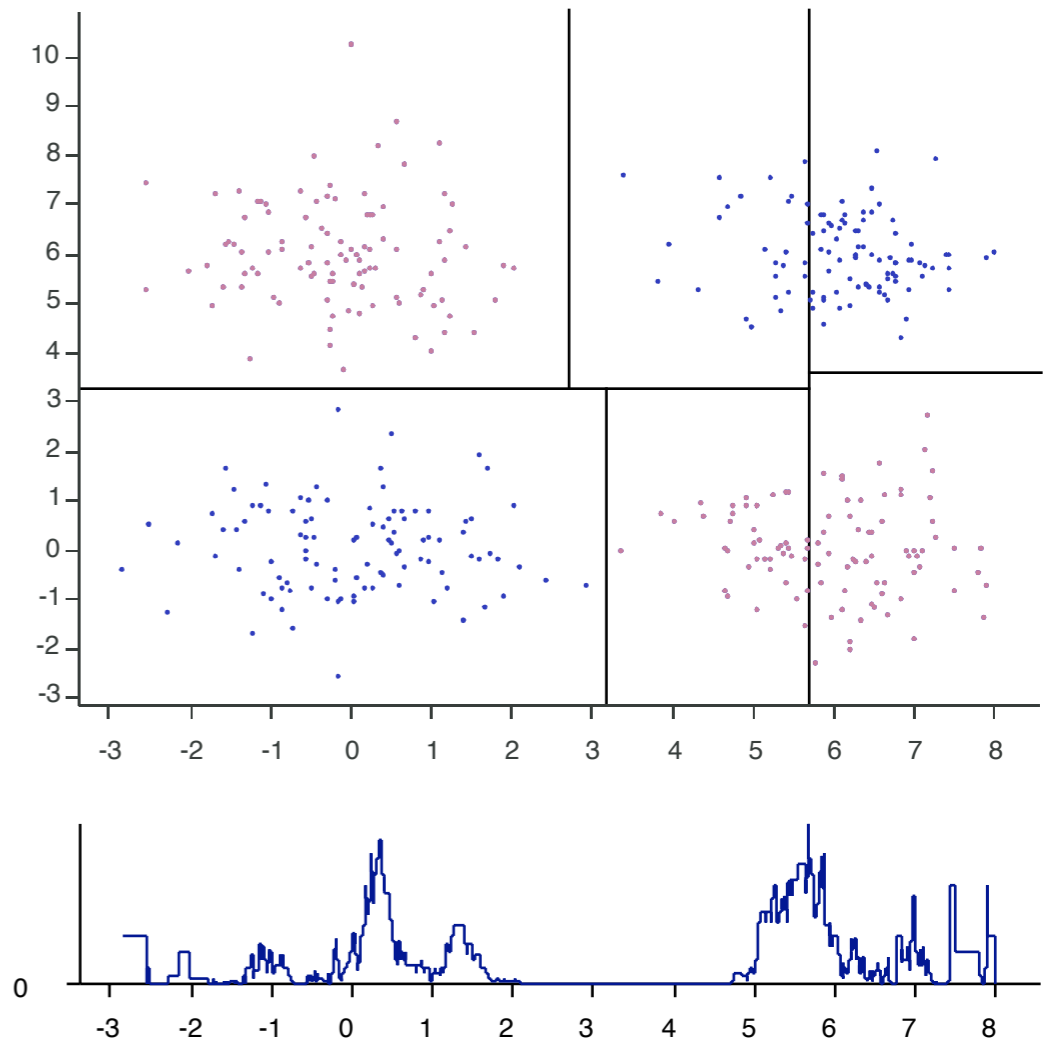
## Bemerkungen

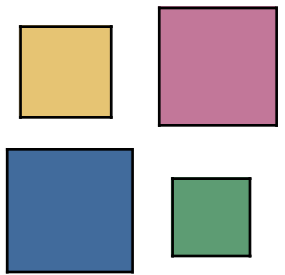
- Für einen Knoten mit z.B. jeweils 400 Beobachtungen pro Gruppe bewertet die Missklassifikation den Split (300, 100) und (100, 300) identisch mit (200, 400) und (200, 0) — mit je 0.25 Fehlklassifikationsrate. Gini und Entropy bevorzugen den zweiten Split, da er einen reinen Knoten generiert.
- Für kategorielle Daten existieren bei  $k$  Ausprägungen einer Variable  $2^k - 1$  verschiedene Splits.
- Für eine binäre Zielvariable vereinfacht sich das Problem, da die Klassen nach dem Anteil des positiven (oder negativen) Ausgangs sortiert werden können, und darauf dann der optimale Split mittels Gini berechnet werden kann. Dies funktioniert auch für quantitative Zielvariablen über die Sortierung nach dem Mittelwert der Gruppen.



# CART: Greedy Algorithmus

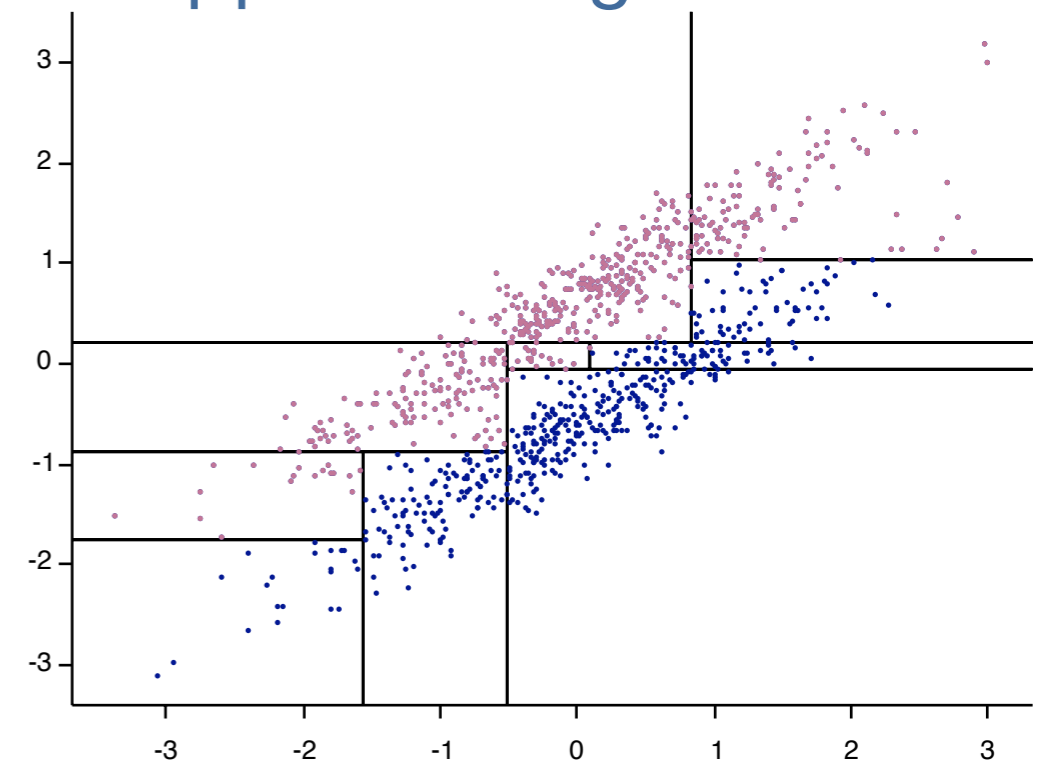
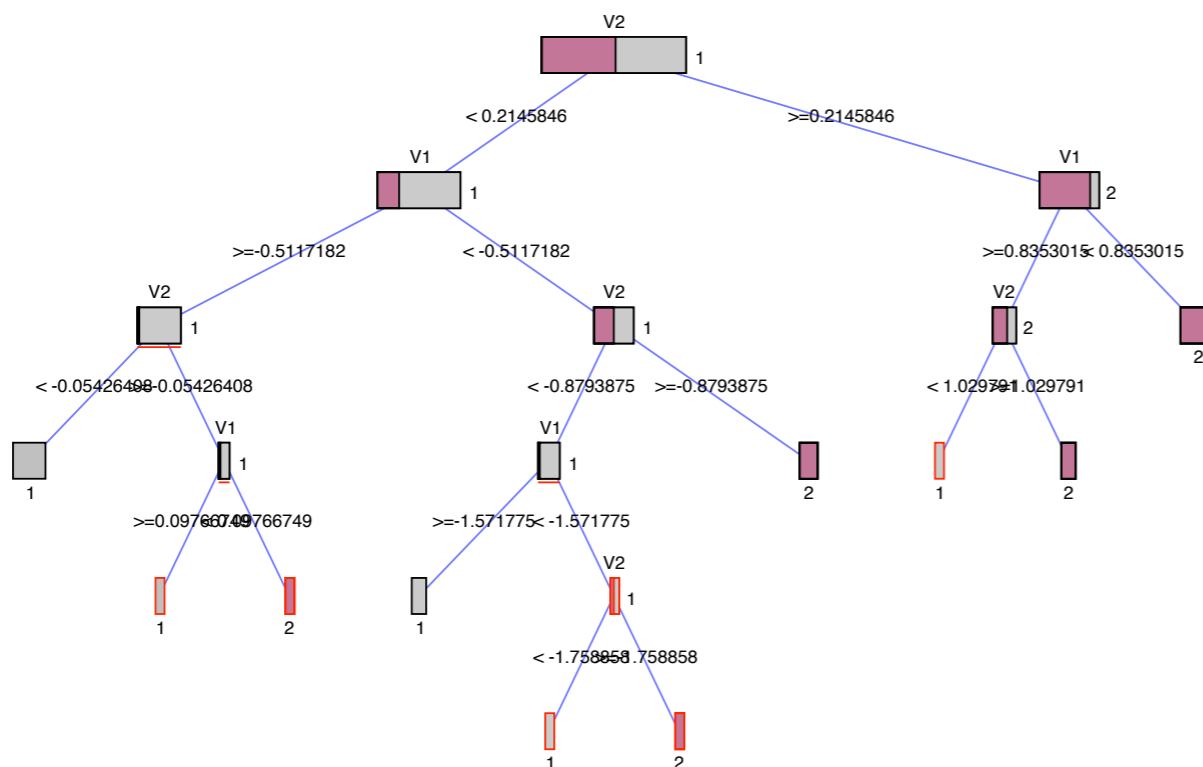
CART sucht sich in jedem Schritt die Variable die den optimalen Gewinn liefert, ohne die Splits in nächsten Schritten zu beachten.



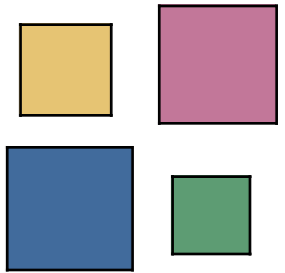


## Bemerkungen

- Wegen des Greedy Algorithmus kann keine Aussage über die globale Optimalität eines Baums gemacht werden.
- Haupt Schwäche von Bäumen ist die Tatsache, dass sie nur orthogonale Splits erzeugen können  $\Rightarrow$  Treppenbildung:



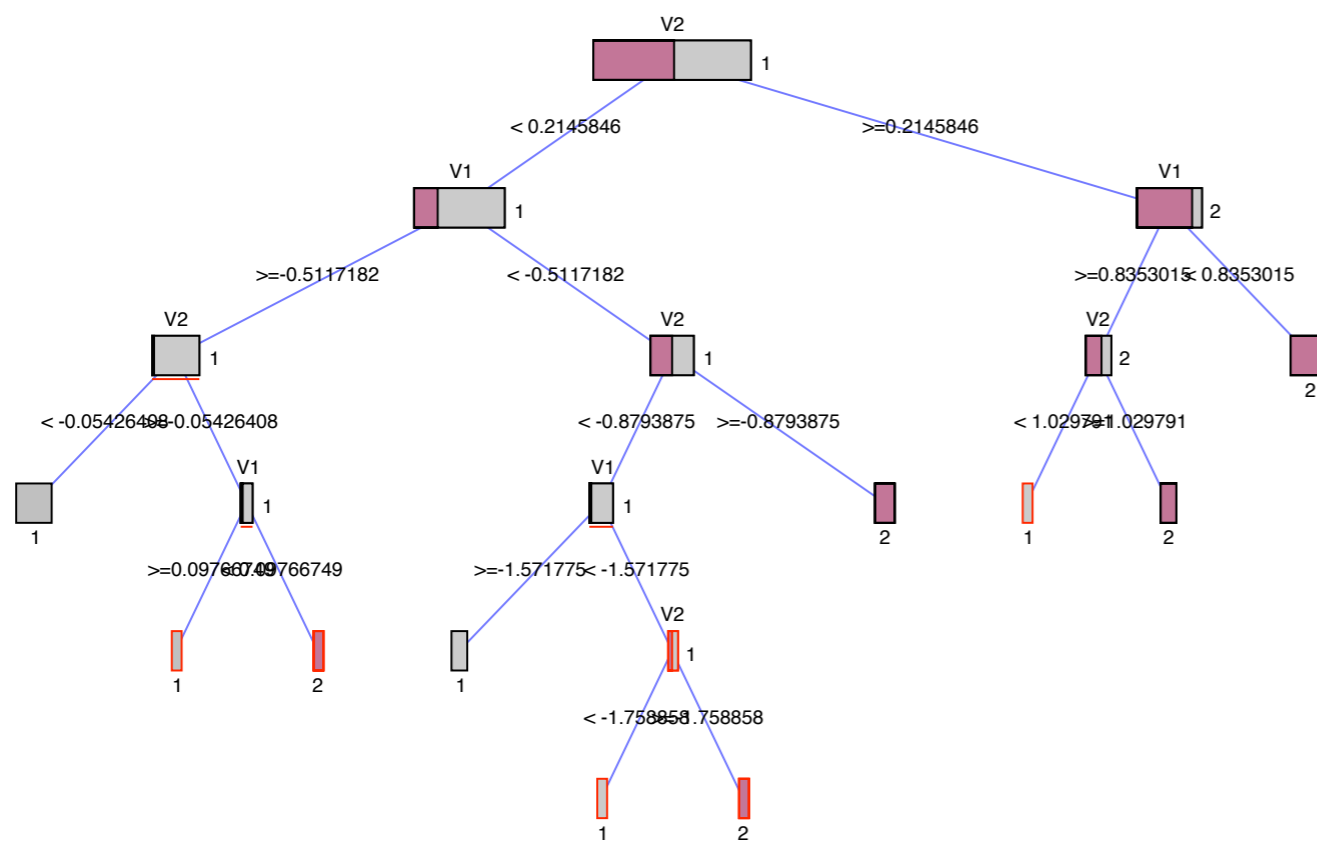
- Bäume sind nur sehr begrenzt stabil gegenüber Änderungen der Inputs.
- Bäume sind auf Grund ihrer einfachen Interpretation sehr populär.



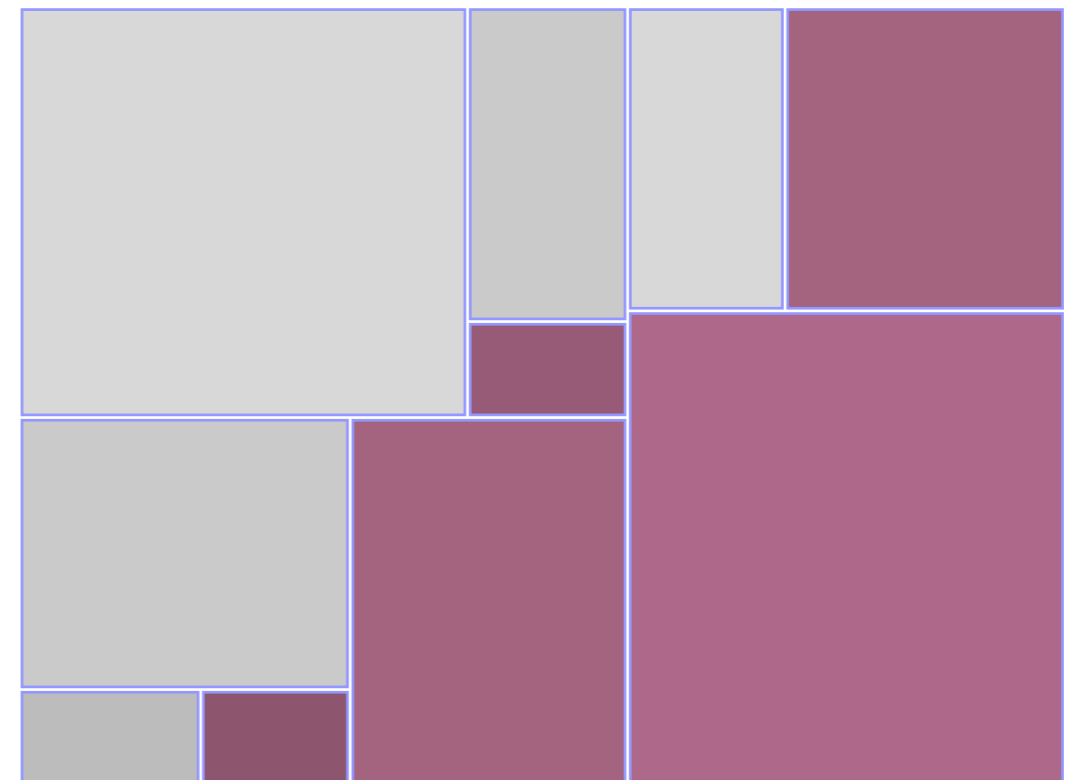
## Alternative Darstellung: Treemap

Zu jedem Baum existiert die Duale Darstellung über eine Treemap.

- Treemaps sind wesentlich kompakter als Bäume
- Baumstruktur schlechter zu erkennen  $\Rightarrow$  Depth Shading
- Treemaps können ggf. weitere Variablen darstellen und lassen sich gut in einem interaktiven Kontext verwenden.



Baum



Treemap