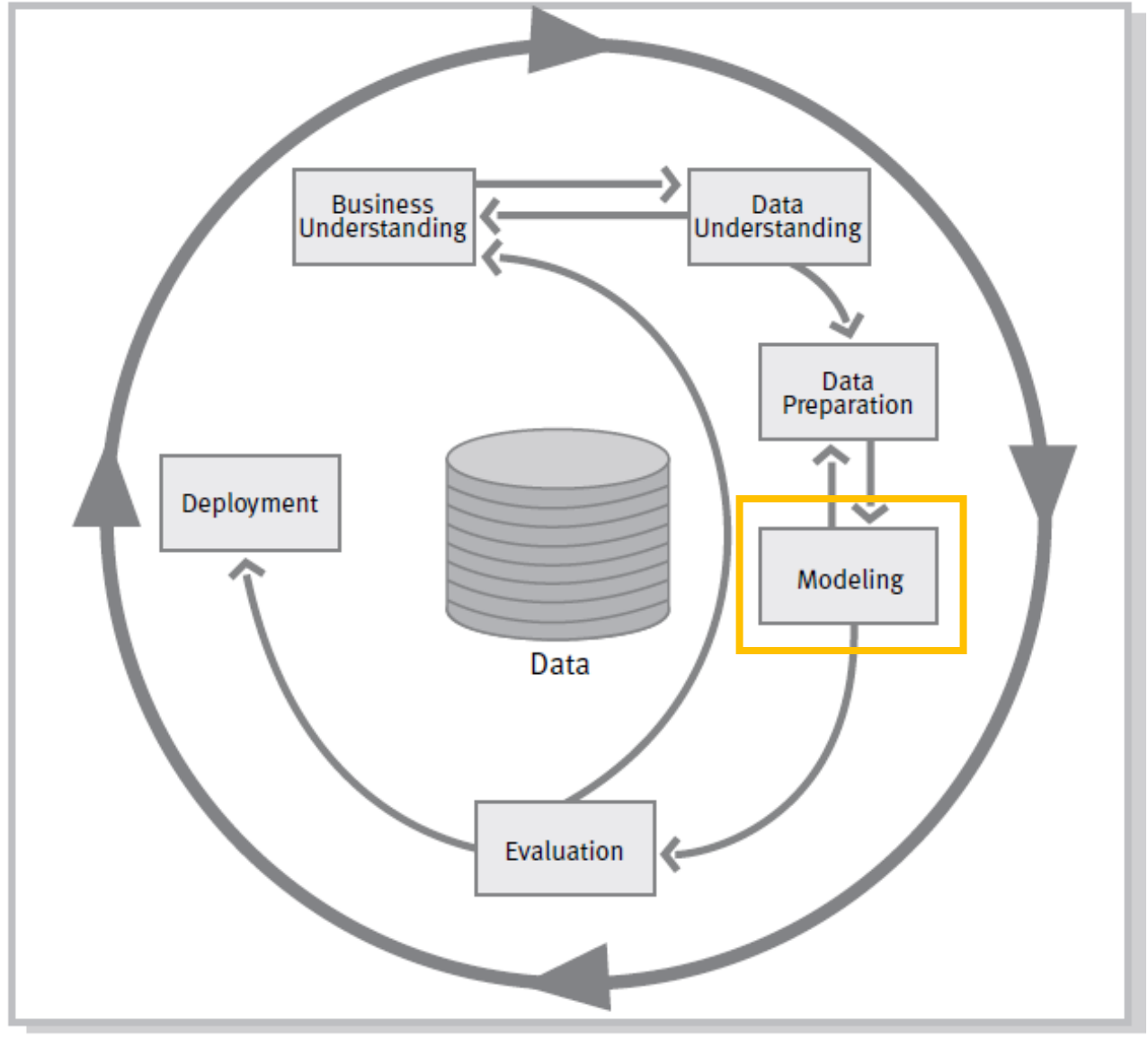
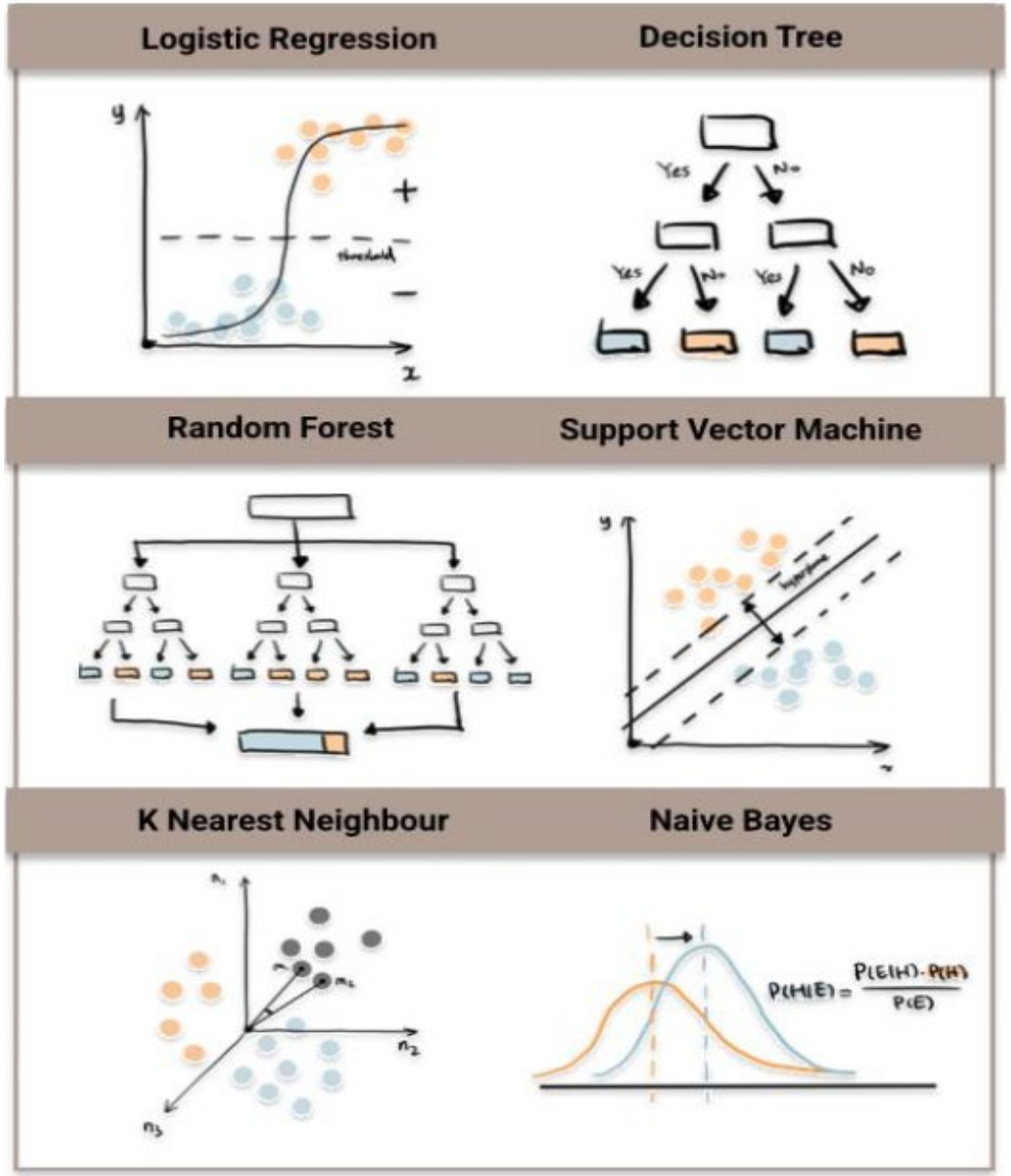
Fachhochschule Nordwestschweiz
Hochschule für Wirtschaft

Maschinelles Lernen – Klassifikation



– Hans Friedrich Witschel, Andreas Martin

Klassifikations-verfahren



Die absolute Baseline – «Constant»

- **Vorhersage**: der Modus (=häufigste Wert) des Klassenattributs auf den Trainingsdaten
- Orange: «Constant»

Noch eine Baseline: One Rule

- Idee: finde das Attribut, das am besten mit dem Klassenattribut «korreliert»

Für jedes Attribut A

- **Erstelle für jeden möglichen Attributwert A_i eine Regel:**
  - **zähle, wie oft jeder Wert der Klasse C zusammen mit A_i auftritt**
  - **Finde den häufigsten dieser Werte C_j**
  - **Kandidat für eine Regel: wenn $A = A_i$, dann $C = C_j$**
- **berechne die Gesamtzahl der Fehler des Regelsatzes (= der Regelkanidaten für alle A_i) auf der Trainingsmenge**
- **Wähle das Attribut mit dem niedrigsten Gesamtfehler**

Beispielregelsatz (Kreditkartenbetrug)

A=Kartentyp (Standard/Gold/Premium)
C=Betrug (ja/nein)

WENN Kartentyp = «Standard» DANN Betrug=«nein»
WENN Kartentyp = «Gold» DANN Betrug=«ja»
WENN Kartentyp = «Premium» DANN Betrug = «nein»

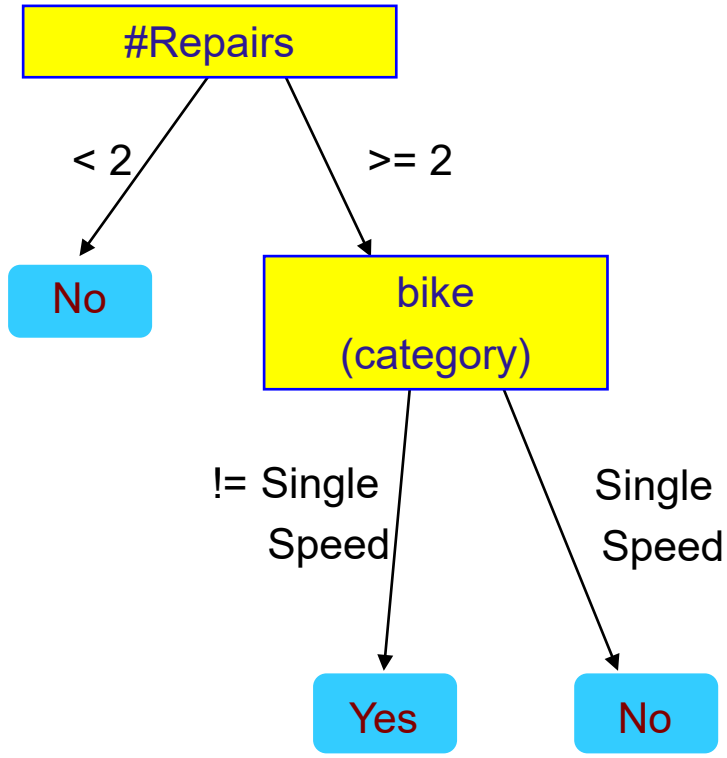
Übung: One Rule von Hand berechnen

Monate seit letzer Reparatur	# Reparaturen letzte 3 Jahre	Response
<= 20	< 2	No
<= 20	>= 2	Yes
> 20	< 2	No
> 20	< 2	No
> 20	< 2	No
<= 20	>= 2	Yes
<= 20	>= 2	Yes
<= 20	< 2	No
> 20	>= 2	No
<= 20	>= 2	Yes

Wir haben unser erstes Modell gelernt!

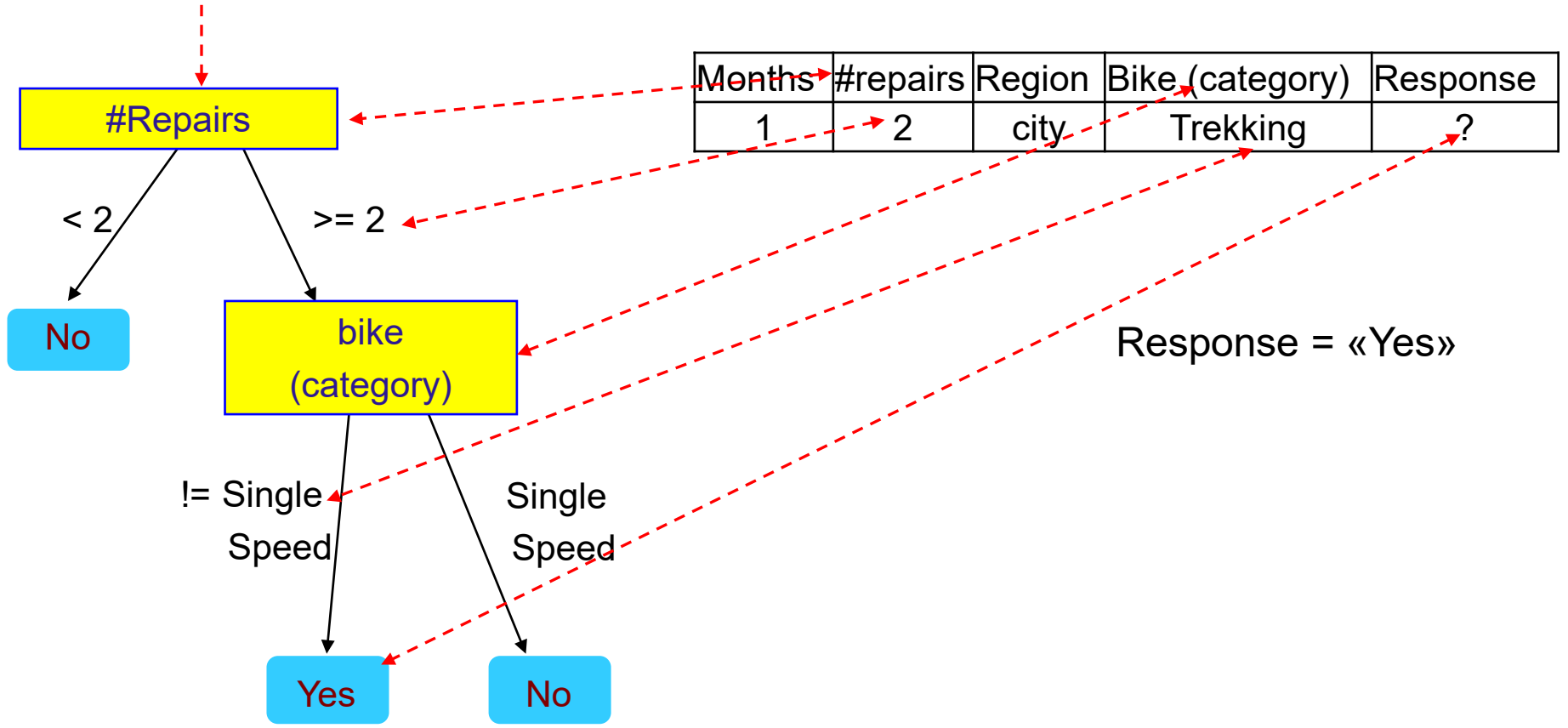
Entscheidungsbäume

months since last repair	#repairs last 3 years	Region	last bike bought (category)	Response
<= 20	< 2	city	Trekking	No
<= 20	>= 2	city	Racing	Yes
> 20	< 2	city	Single Speed	No
> 20	< 2	rural	Single Speed	No
> 20	< 2	rural	Mountain	No
<= 20	>= 2	rural	Trekking	Yes
<= 20	>= 2	city	Mountain	Yes
<=20	< 2	rural	Mountain	No
> 20	>= 2	city	Single Speed	No
<= 20	>= 2	rural	Trekking	Yes



Entscheidungsbäume – Anwendung

Wir starten an der Baumwurzel



Neuer Kunde:

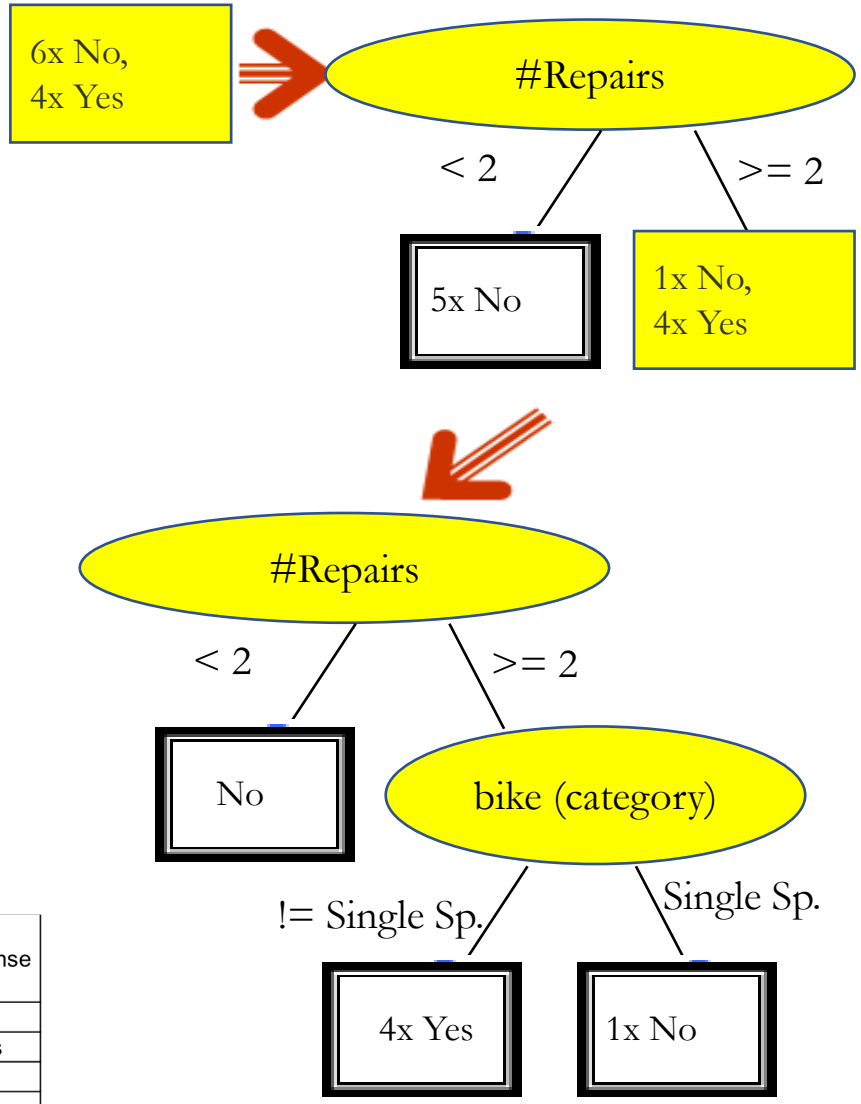
Months	#repairs	Region	Bike (category)	Response
1	2	city	Trekking	?

Response = «Yes»

Entscheidungsbäume – Lernen

- D_t = alle Instanzen, die den aktuellen Knoten t "erreicht" haben

- **Hunt's algorithm**:
  a. D_t enthält Instanzen, die mehr als einer Klasse angehören: wähle ein Attribut A, um die Instanzen aufzuteilen, bezeichne t mit dem Namen von A. Erstelle für jeden Wert von A einen Kindknoten von t, verfahre mit diesen so wie mit t
  b. D_t enthält Instanzen, die alle zur gleichen Klasse y_t gehören: t ist ein Blattknoten mit Label y_t



months since last repair	#repairs last 3 years	Region	last bike bought (category)	Response
<= 20	< 2	city	Trekking	No
<= 20	>= 2	city	Racing	Yes
> 20	< 2	city	Single Speed	No
> 20	< 2	rural	Single Speed	No
> 20	< 2	rural	Mountain	No
<= 20	>= 2	rural	Trekking	Yes
<= 20	>= 2	city	Mountain	Yes
<=20	< 2	rural	Mountain	No
> 20	>= 2	city	Single Speed	No
<= 20	>= 2	rural	Trekking	Yes

Entscheidungsbäume – Split-Attribut wählen

- Idee: Attribute auswählen, deren Werte die Menge D_t in Teilmengen trennen, deren Instanzen möglichst alle der gleichen Klasse angehören, d.h. die hinsichtlich der Klasse homogen sind

Heterogen

Homogen

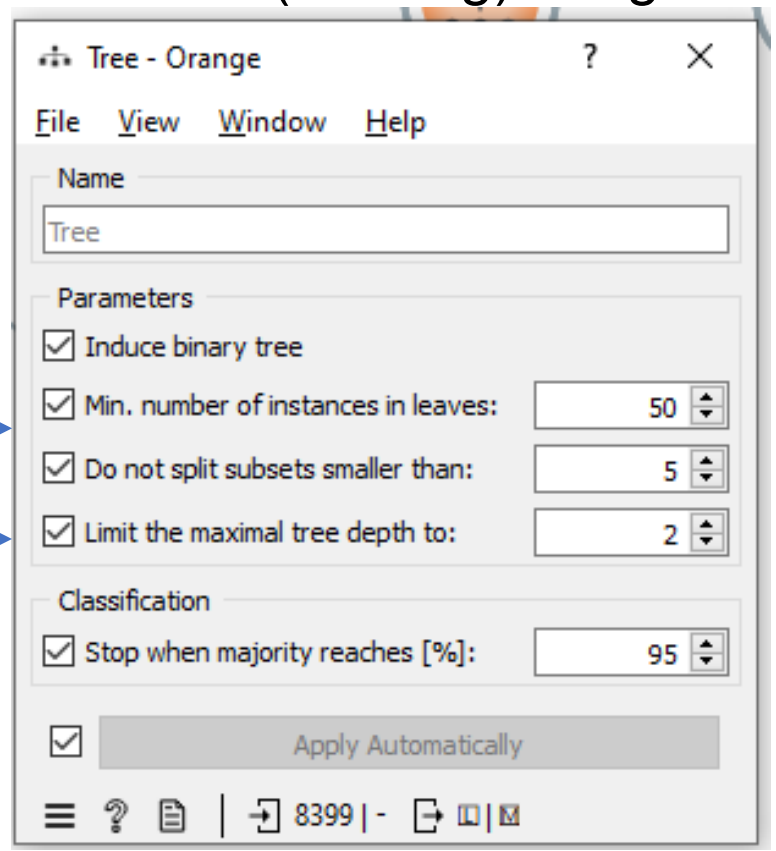
- Z.B.: Auswahl basierend auf der Fehlerrate, d.h. Erstellung eines One Rule-Klassifikators für jedes D_t

Entscheidungsbäume – wichtige Parameter

- Meist werden Entscheidungsbäume «abgeschnitten» (Pruning). Mögliche Kriterien:

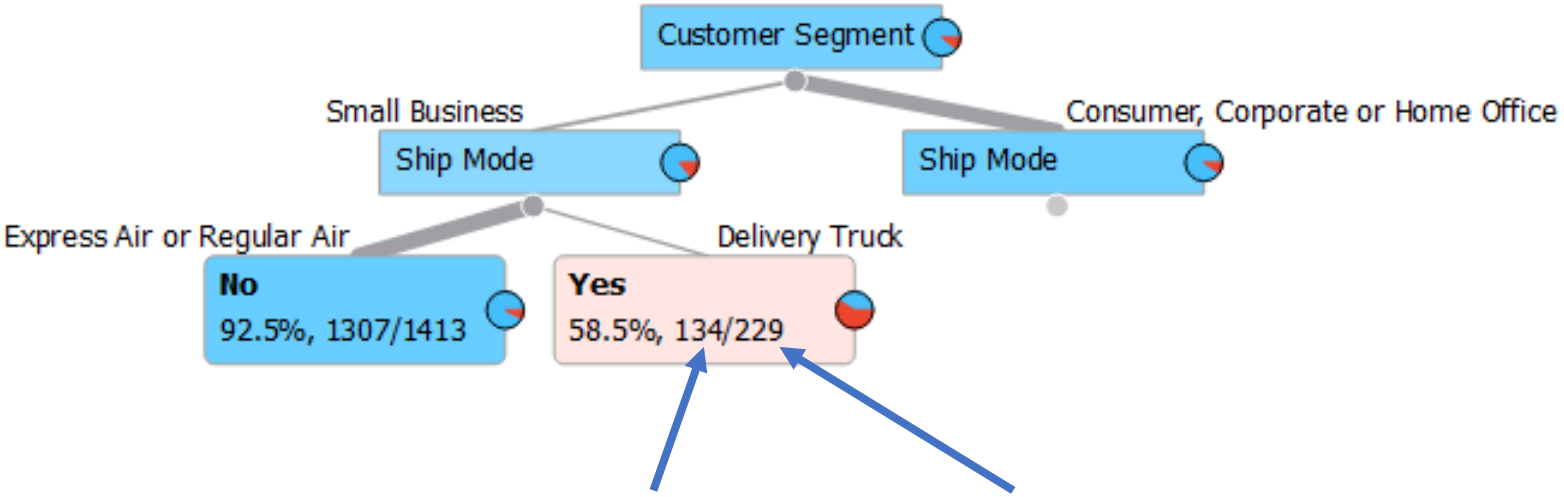
a. Mindestgrösse der Menge D_t

b. Tiefe einschränken



Entscheidungsbäume – Zahlen interpretieren

- Interpretation von Kennzahlen:

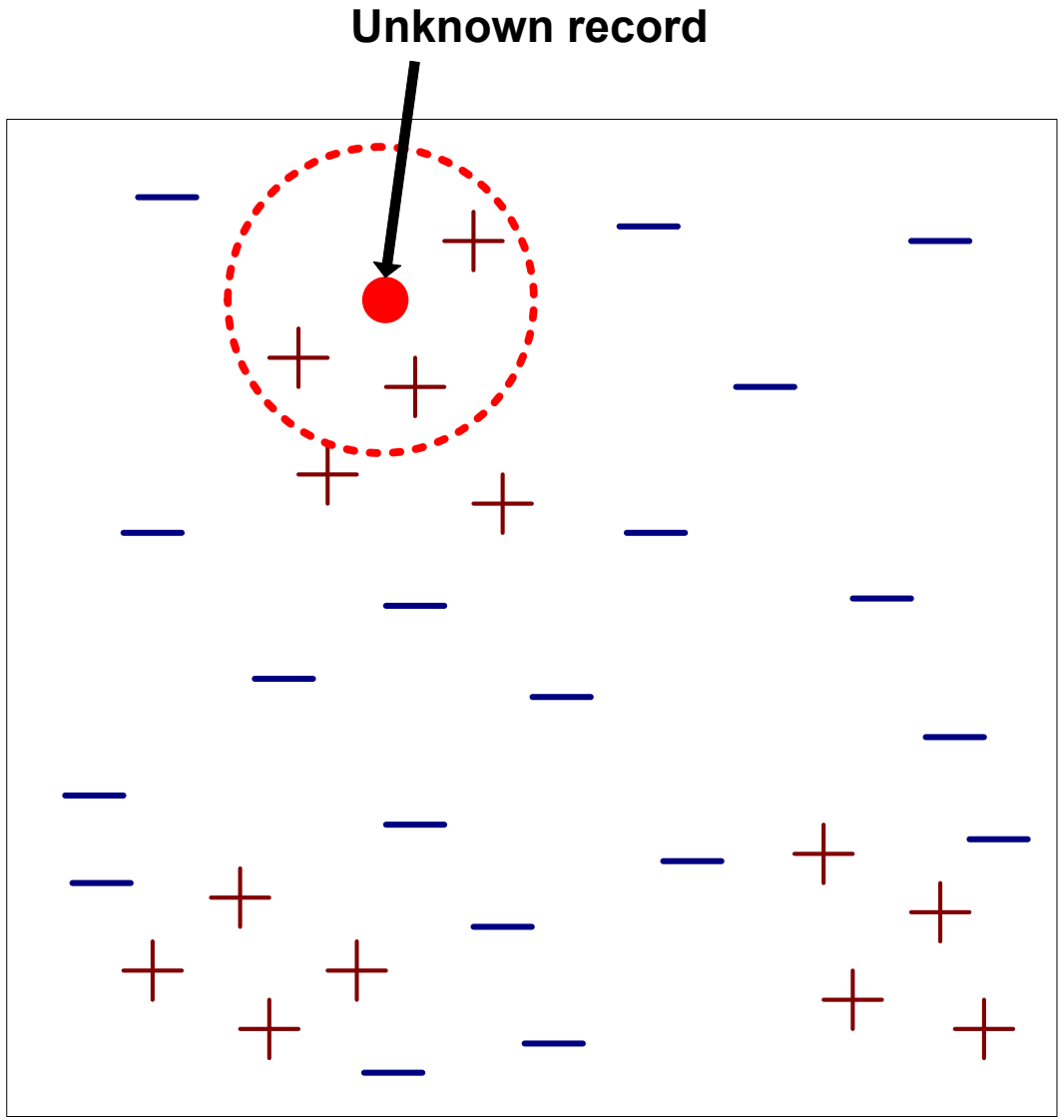


134 Bestellungen in diesem Knoten haben Returned =«Yes»

229 Bestellungen «landen» in diesem Knoten

K Nearest Neighbour

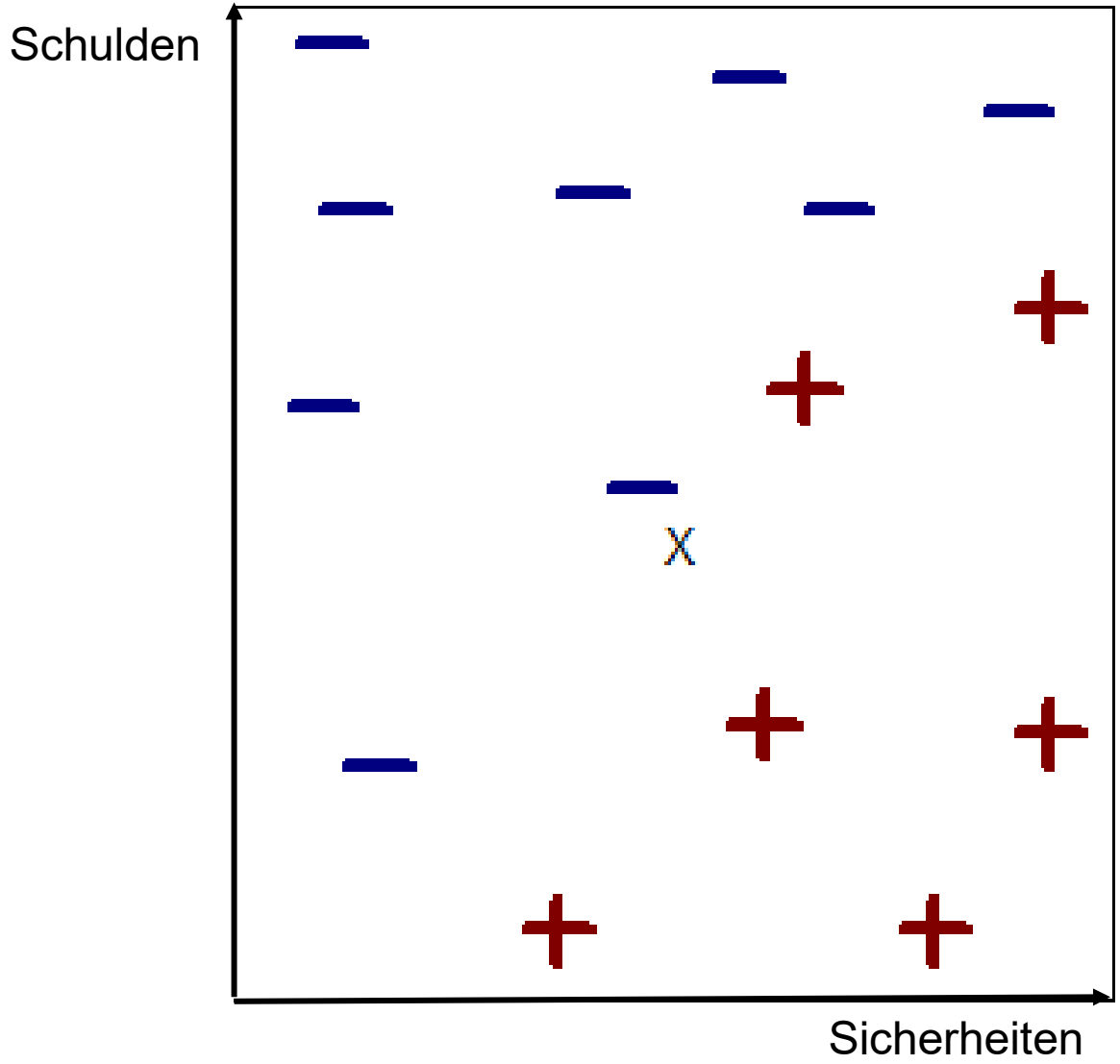
- Inputs:
  a. Eine Trainingsmenge
  b. Ein Mass für Ähnlichkeit oder Abstand zwischen Instanzen
  c. Der Wert von k = Anzahl der nächstgelegenen Nachbarn, die gefunden werden sollen
- Vorhersage für eine neue Instanz:
  a. Berechne Ähnlichkeit zu allen Instanzen der Trainingsmenge
  b. Die k ähnlichsten identifizieren
  c. «Vererbung» der Klasse z.B. durch (nach Ähnlichkeit gewichteten) Mehrheitsentscheid



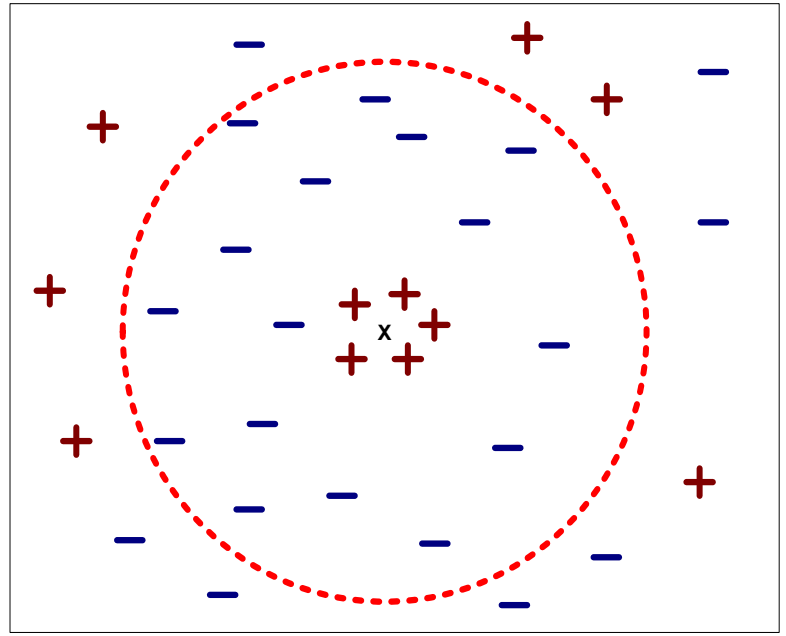
kNN: Übung

Wie wird der graue Punkt von einem kNN für k=1,2,3 klassifiziert? Für jedes k: mit Abstand ähnlichkeitsgewichtete oder nicht gewichtete Mehrheitsabstimmung?

	k=1	k=2	k=3
Ungewichtet			
Gewichtet			



kNN – Nachteile



- Wahl von k:
  a. Zu klein: kann von «noise points» beeinflusst werden
  b. Zu gross: kann Entscheidungsgrenzen «übertreten»

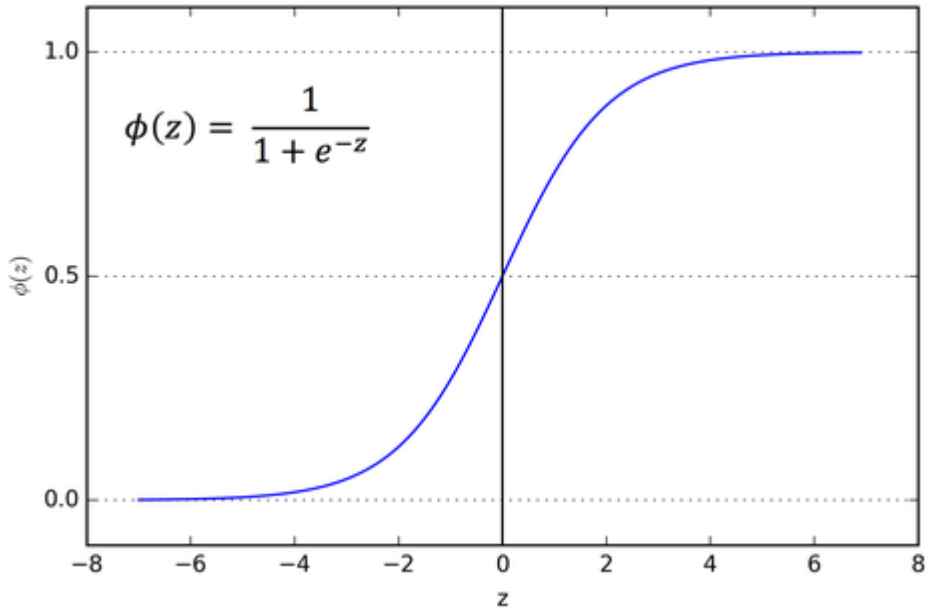
- Interpretierbarkeit:
  a. Es gibt kein Modell!
  b. Immerhin: Vorhersagen können mittels nächster Nachbarn erklärt werden

Logistische Regression

- **Grundidee**: alle Attribute werden in einer gewichteten Summe kombiniert

$$z = w_1 x_1 + \cdots w_n x_n + b$$

- Und das wird dann in die sigmoid-Funktion $\varphi(z) = \frac{1}{1+e^{-z}}$ eingesetzt
- Wenn $\varphi(z) > 0.5$, dann sagt der Klassifikator «ja», sonst «nein»



Logistische Regression – Beispiel

$$\varphi(z) = \frac{1}{1 + e^{-z}}$$

- Swiss Bikes Wintercheck – Koeffizienten:

intercept	0	← b						
months since last repair	-0.154346	← w_1	7	-1.08042		10	-1.54346	
repairs last 3 years	1.4031	← w_2	1	1.4031		1	1.4031	
region=city	0		1	0		1	0	
region=rural	0		0	0		0	0	
last bike bought (category)=Mountain	0		0	0		0	0	
last bike bought (category)=Rennrad	0		0	0		0	0	
last bike bought (category)=Single Speed	0		0	0		0	0	
last bike bought (category)=Trekking	0		1	0		1	0	
			0.322678	**gewichtete Summe**			**-0.14036**	**gewichtete Summe**
			0.579977	**Ergebnis (Sigmoid)**			**0.464967**	**Ergebnis (Sigmoid)**

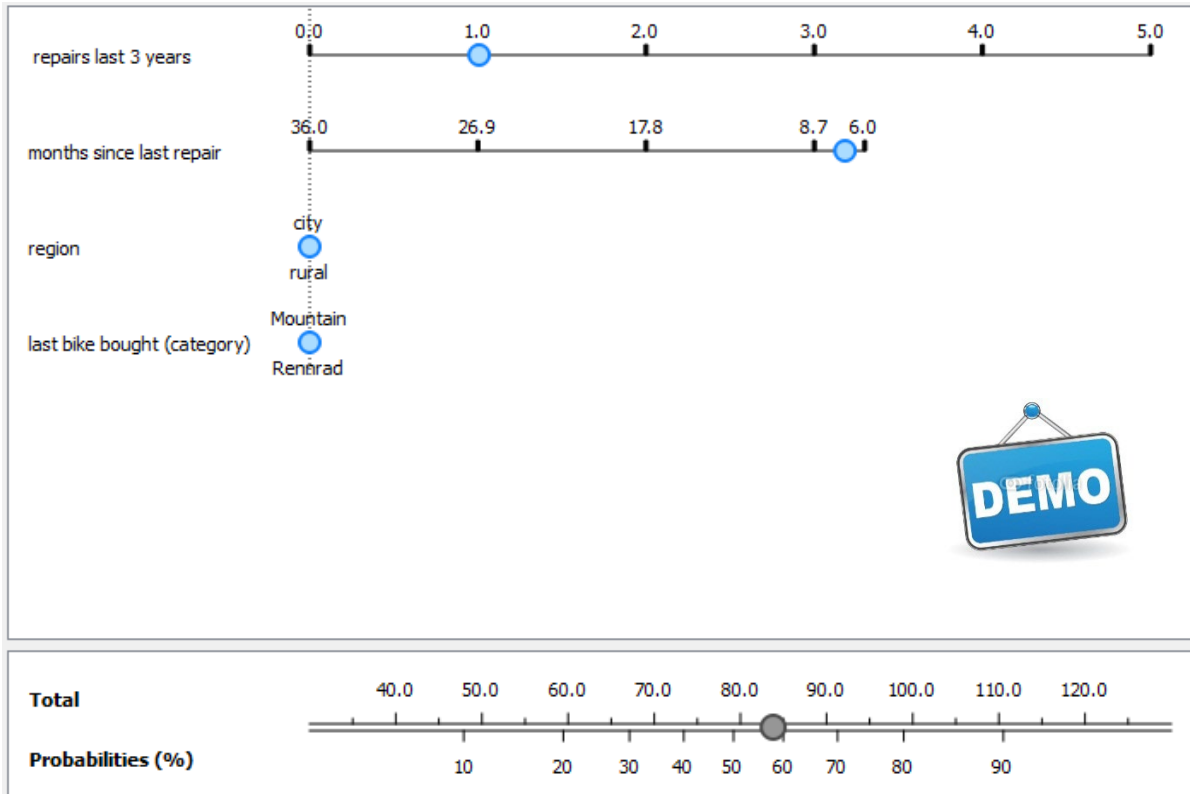
- Testkunde: hatte eine Reparatur in den letzten 3 Jahren (vor 7 Monaten), wohnt in der Stadt, hat ein Trekking-Rad

  → Vorhersage: «ja» (Interesse)

- Was, wenn die Reparatur 10 Monate her ist?

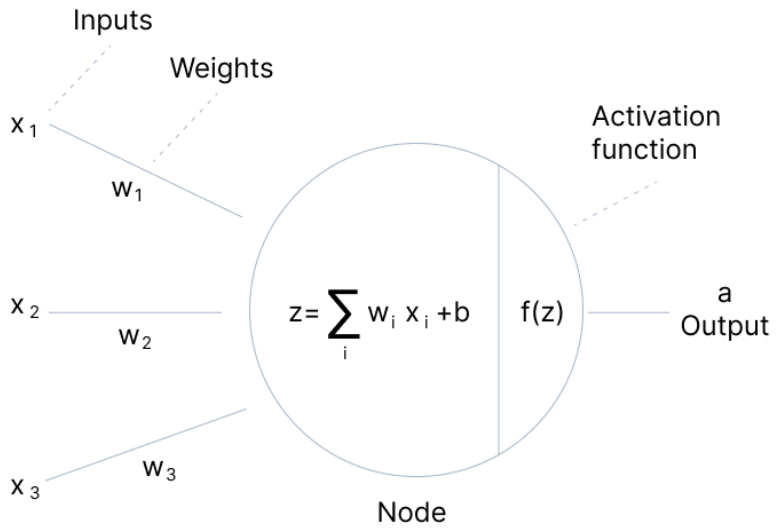
Logistische Regression – Interpretation

- **Interpretation**:
  a. Dieser Klassifikator sagt genau dann «ja», wenn die gewichtete Summe $z > 0$ ist
  b. $\varphi(z)$ kann als Wahrscheinlichkeit interpretiert werden
  c. Wenn man die Attributwerte normalisiert, können die Gewichte als «Wichtigkeit» des Attributs interpretiert werden
  d. Orange: Einfluss der Attribute kann per Nomogram exploriert werden



Logistische Regression – Beobachtungen

- **Nachteil**: alle Attribute werden als unabhängig voneinander betrachtet, d.h. multivariate Muster sind ausgeschlossen
  a. Bsp.: für ein multivariates Muster: ‘wenn Region=«city» und Monate < 8, dann «ja»’
  b. Bei multivariaten Mustern kann ein Attribut, das sonst eher unwichtig ist, eine wichtige Rolle spielen…!
- **Für später**: die Berechnung von $\varphi(z)$ ist genau das, was ein einzelnes Neuron in einem neuronalen Netz macht (nur dass manchmal statt sigmoid was anderes verwendet wird)



Bildquelle: https://www.v7labs.com/blog/neural-networks-activation-functions

V7 Labs

Gradient Boosting – Idee

- Grundidee: kombiniere mehrere «schwache» Algorithmen
- Vorgehen:
  a. Lerne ein Modell (z.B. Entscheidungsbaum)
  b. Identifiziere die Fehler des Modells
  c. Trainiere ein weiteres Modell darauf, die Fehler des ersten zu korrigieren
  d. Usw.
  e. Kombiniere die Vorhersagen aller Modelle (mit abnehmendem Einfluss)

- **Vorteil**: oft deutlich genauere Vorhersagen
- **Nachteil**: nicht bzw. sehr schwer interpretierbar

Auswahl des richtigen Klassifikators???

	Decision Trees	Neural Networks	Naïve Bayes	kNN	SVM	Rule-learners
Accuracy in general	**	***	*	**	****	**
Speed of learning with respect to number of attributes and the number of instances	***	*	****	****	*	**
Speed of classification	****	****	****	*	****	****
Tolerance to missing values	***	*	****	*	**	**
Tolerance to irrelevant attributes	***	*	**	**	****	**
Tolerance to redundant attributes	**	**	*	**	***	**
Tolerance to highly interdependent attributes (e.g. parity problems)	**	***	*	*	***	**
Dealing with discrete/binary/continuous attributes	****	***(not discrete)	***(not continuous)	***(not directly discrete)	**(not discrete)	***(not directly continuous)
Tolerance to noise	**	**	***	*	**	*
Dealing with danger of overfitting	**	*	***	***	**	**
Attempts for incremental learning	**	***	****	****	**	*
Explanation ability/transparency of knowledge/classifications	****	*	****	**	*	****
Model parameter handling	***	*	****	***	*	***

Table 4. Comparing learning algorithms (**** stars represent the best and * star the worst performance)

Aus: Kotsiantis, S.B., 2007. Supervised Machine Learning: A Review of Classification Techniques. Informatica, 31, pp.249–268.