

# **Big Data & Data Science**

## **Lösung zu diversen Aufgaben**

WS 2025/26

**Prof. Dr. Klemens Waldhör**

**© FOM Hochschule für Oekonomie & Management  
gemeinnützige Gesellschaft mbH (FOM), Leimkugelstraße 6, 45141 Essen**

Dieses Werk ist urheberrechtlich geschützt und nur für den persönlichen Gebrauch im Rahmen der Veranstaltungen der FOM bestimmt.

Die durch die Urheberschaft begründeten Rechte (u.a. Vervielfältigung, Verbreitung, Übersetzung, Nachdruck) bleiben dem Urheber vorbehalten.

Das Werk oder Teile daraus dürfen nicht ohne schriftliche Genehmigung der FOM reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

## Aufgabe 1

Analysieren und präsentieren Sie **Titanic- und Iris-Daten!**

- Welche Variablen kommen vor?
- Welche Datenqualität? Maßzahlen? (sh. Übung 1 und 2)
- Ermitteln Sie wichtigsten Kennzahlen für Daten!
- Visualisieren Sie die Daten!
- Welche Schlüsse können Sie ziehen?
- Was kann man den Daten entnehmen?
- Wofür könnte man sie einsetzen?

## Aufgabe 2

Suchen und verwenden Sie einen interessanten, neuen Visualisierungsansatz

# Übung 1a: Maßskalen - Titanic Dataset

Variable	Art	Skala
Passenger Class		
Name		
Sex		
Age		
No. of Siblings or Spouses on Board		
No. of Parents or Children on Board		
Ticket Number		
Passenger Fare		
Cabin		
Port of embarkation		
Life boat		
Survived		

# Übung 1a: Maßskalen - Titanic Dataset - Lösung

Variable	Art	Skala
Passenger Class	Qualitativ	Ordinal
Name	Qualitativ	Nominal
Sex	Qualitativ	Nominal
Age	Quantitativ	Metrisch diskret
No. of Siblings or Spouses on Board	Quantitativ	Metrisch diskret
No. of Parents or Children on Board	Quantitativ	Metrisch diskret
Ticket Number	Qualitativ	Nominal
Passenger Fare	Quantitativ	Metrisch stetig
Cabin	Qualitativ	Nominal
Port of embarkation	Qualitativ	Nominal
Life boat	Qualitativ	Nominal
Survived	Qualitativ	Nominal

## Übung 1b: Maßskalen – Iris Datensatz

Variable	Art	Skala
Länge des Kelchblatts		
Breite des Kelchblatts		
Länge des Kronblatts		
Breite des Kronblatts		
Art		

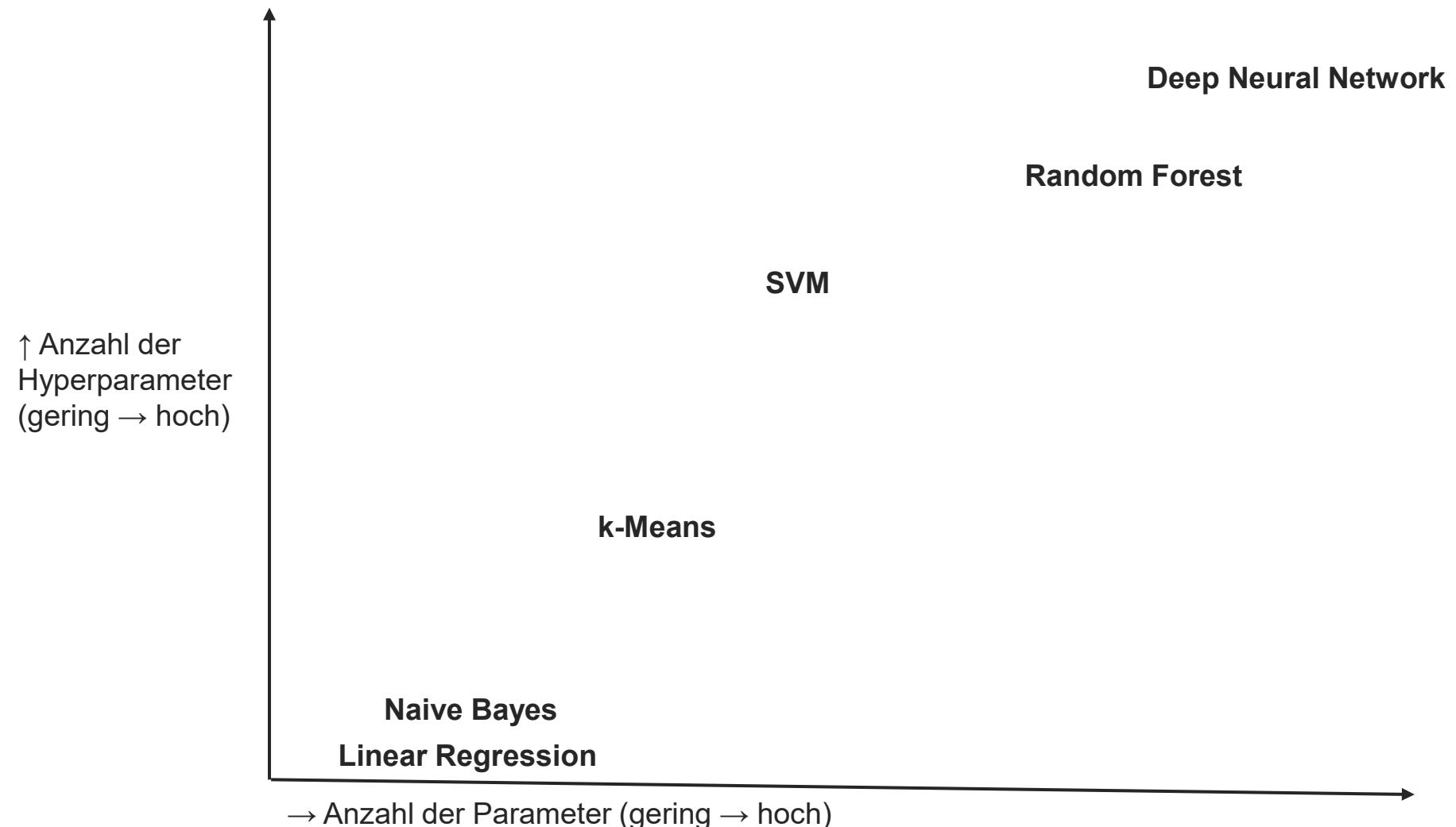
## Übung 1b: Maßskalen – Iris Datensatz Lösung

Variable	Art	Skala
Länge des Kelchblatts	Quantitativ	Metrisch
Breite des Kelchblatts	Quantitativ	Metrisch
Länge des Kronblatts	Quantitativ	Metrisch
Breite des Kronblatts	Quantitativ	Metrisch
Art	Qualitativ	Nominal

# Übung Parameter / Hyperparameter

Verfahren	Lernart	Typische Parameter	Typische Hyperparameter
Lineare Regression	Supervised	Regressionsgewichte, Bias	Lernrate (falls GD), Regularisierungsstärke ( $\lambda$ )
Logistische Regression	Supervised	Gewichte, Bias	Regularisierungsparameter (C), Lernrate
k-Nearest Neighbors (kNN)	Supervised	– (keine trainierten Parameter, speichert nur Daten)	k-Wert, Distanzmaß
Decision Tree	Supervised	Schwellenwerte an Knoten	Maximale Tiefe, min. Samples pro Split
Random Forest	Supervised	Parameter der einzelnen Bäume	Anzahl der Bäume, max. Tiefe, max. Features
Support Vector Machine (SVM)	Supervised	Support-Vektoren, Gewichte	C-Parameter, Kerneltyp, $\gamma$ (bei RBF)
Naive Bayes	Supervised	Klassenwahrscheinlichkeiten, bedingte Wahrscheinlichkeiten	– (meist kaum Hyperparameter)
k-Means	Unsupervised	Clusterzentren ( $\mu_1, \dots, \mu_k$ )	Anzahl Cluster k, max. Iterationen
Principal Component Analysis (PCA)	Unsupervised	Hauptachsen (Eigenvektoren)	Anzahl der Komponenten
Neural Network (MLP, CNN, etc.)	Supervised	Gewichte und Biases aller Layer	Lernrate, Anzahl Layer, Neuronen, Aktivierungsfunktion, Batchgröße, Epochen
Autoencoder	Unsupervised	Encoder-/Decoder-Gewichte	Lernrate, Layergröße, Regularisierung
Q-Learning / Reinforcement Agent	Reinforcement	Q-Werte (state-action values)	Lernrate $\alpha$ , Diskontfaktor $\gamma$ , Explorationsrate $\epsilon$

## Lösung



## Aufgabe 1

- a) Trainingsdaten: zum Lernen der Modellparameter.
- b) Validierungsdaten: zur Auswahl der Hyperparameter und Kontrolle von Overfitting.
- c) Testdaten: zur finalen Beurteilung der Modellleistung.

## Aufgabe 2

Nutzung der Testdaten beim Tuning führt zu Data Leakage  
das Modell 'kennt' dann Testdaten.

## Aufgabe 3

Beispielaufteilung (10.000 Beobachtungen):

70 % Train = 7.000

15 % Validierung = 1.500

15 % Test = 1.500

## Aufgabe 4: k-Fold Cross-Validation (z. B. k=5):

Daten in 5 gleich große Folds aufteilen.

4 Folds trainieren, 1 Fold validieren.

Wiederhole 5x, jeder Fold einmal Validierung.

## Aufgabe 5: Was unterscheidet Kreuzvalidierung von einfacher Aufteilung?

Vorteil: robustere Schätzung als eine einfache Split-Aufteilung.

## Aufgabe 6: Bei 1.000 Beispielen, 5-Fold:

800 Train, 200 Validierung je Fold.

## Aufgabe 7: Warum ist Kreuzvalidierung bei kleinen Datensätzen besonders sinnvoll?

Sinnvoll bei kleinen Datensätzen → nutzt alle Daten mehrfach effizient.

## Aufgabe 8: Genauigkeiten über 5 Folds

Genauigkeiten: 0.88, 0.86, 0.89, 0.85, 0.90

$$\text{Mittelwert} = (0.88+0.86+0.89+0.85+0.90)/5 = 0.876$$

Standardabweichung  $\approx 0.018$

Bewertung: geringe Varianz  $\rightarrow$  Modell stabil.

## Aufgabe 9: Warum darf das Testset erst nach der Kreuzvalidierung verwendet werden?

Testset erst am Schluss verwenden, um unverzerrte Schätzung der Generalisierungsleistung zu erhalten.

Nutzung vorher  $\rightarrow$  Overfitting auf Testdaten möglich.

	Vorhersage positiv	Vorhersage negativ
Tatsächlich positiv	<input checked="" type="checkbox"/> True Positive (TP)	<input type="checkbox"/> False Negative (FN)
Tatsächlich negativ	<input type="checkbox"/> False Positive (FP)	<input checked="" type="checkbox"/> True Negative (TN)

## Konfusionsmatrizen & Qualitätsmaße

Berechnen Sie für folgenden Tabellen die angeführten Gütemaße und interpretieren sie das Ergebnis.

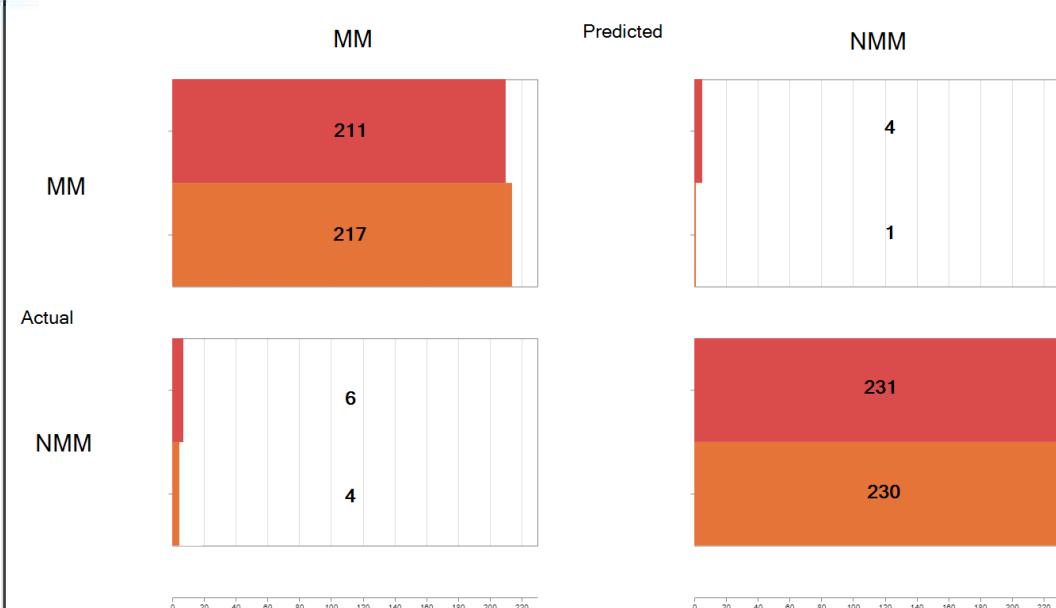
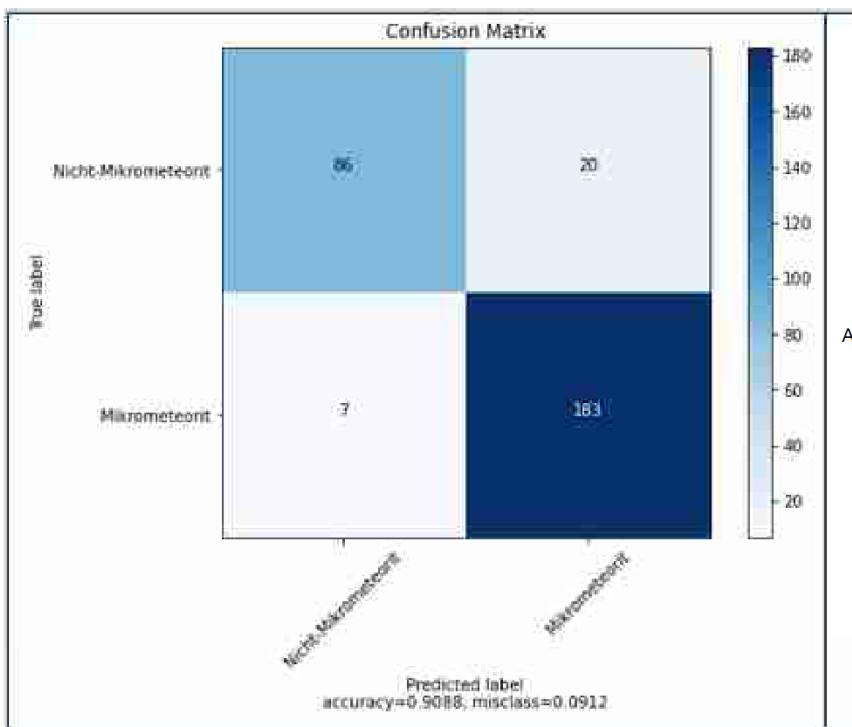


Abbildung 18: Confusion Matrix zur binären Klassifikation des besten Modells (Orange) und eines durchschnittlichen Modells (Rot) (Quelle: Eigene Darstellung)

Merkel M, Voigt R (2021) Bildverarbeitung/analyse - Mikrometeoriten erkennen. Seminararbeit, Nürnberg. Abruf am 2024-02-29.

Voigt R (2023) Automatische Erkennung von Mikrometeoriten mittels künstlicher Intelligenz: Prototyping einer State-of-the-Art Architektur. Masterthesis, Nürnberg. Abruf am 2024-02-29.

**Wahr / Vorhergesagt**

**Nicht-Mikrometeorit**

**Mikrometeorit**

**Nicht-Mikrometeorit**

86 (TN)

7 (FN)

**Mikrometeorit**

20 (FP)

183 (TP)

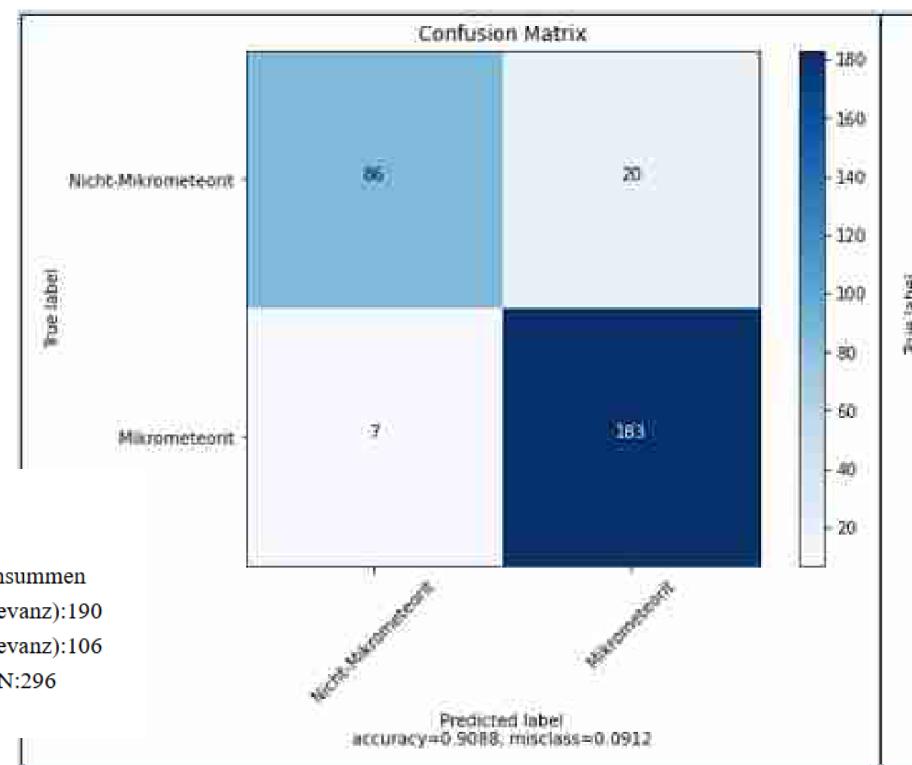
True Positives (TP) = 183

True Negatives (TN) = 86

False Positives (FP) = 20

False Negatives (FN) = 7

Gesamt =  $86 + 20 + 7 + 183 = 296$



# Binäre Klassifikatoren – Gütemaße - Lösung

## Gütemaße für binäre Klassifikatoren

	Vorhergesagte Klasse	Spaltensummen	
Wahre Klasse	TP: 183 FP: 20	FN: 7 TN: 86	R (Relevanz):190 I (Irrelevanz):106
Zeilensummen	P (Positivität):203	N (Negativität):93	NN:296
Gütemaße für binären Klassifikator bestimmten			

Gütemaß	Wert
TP	183
TN	86
FP	20
FN	7
NN	296
T	269
F	27
R	190
I	106
P	203
N	93
Accuracy	0.9087837837837838
ACC	0.9087837837837838
Inkorrektheitsrate	0.09121621621621621
TPR	0.9631578947368421
Sensitivitaet	0.9631578947368421
Recall	0.9631578947368421
TNR	0.8113207547169812
FPR	0.18867924528301888
FNR	0.03684210526315789
PR	0.9014778325123153
Precision	0.9014778325123153
Negativer Vorhersagewert	0.9247311827956989
Negative Falschklassifikationsrate	0.07526881720430108
Positive Falschklassifikationsrate	0.09852216748768473
Fehlerrate	0.09121621621621621
Erfolgsrate	0.9087837837837838
Relevanz Geron	0.9014778325123153
Spezifitaet	0.8113207547169812
Praevalenz	0.6418918918918919
Positives Likelihood Verhaeltnis	5.104736842105265
Negatives Likelihood Verhaeltnis	0.04541003671970625
F1 Score	0.9312977099236641
ACCO	0.5
KappaKoeffizient	0.8175675675675675
Distance ROC Kurve	0.1922425507550064
AUC	0.8872393247269117

**Tatsächlich /  
Vorhergesagt**

**MM (tatsächlich)**

**NMM (tatsächlich)**

**MM (Mikrometeorit)**

211 (TP)

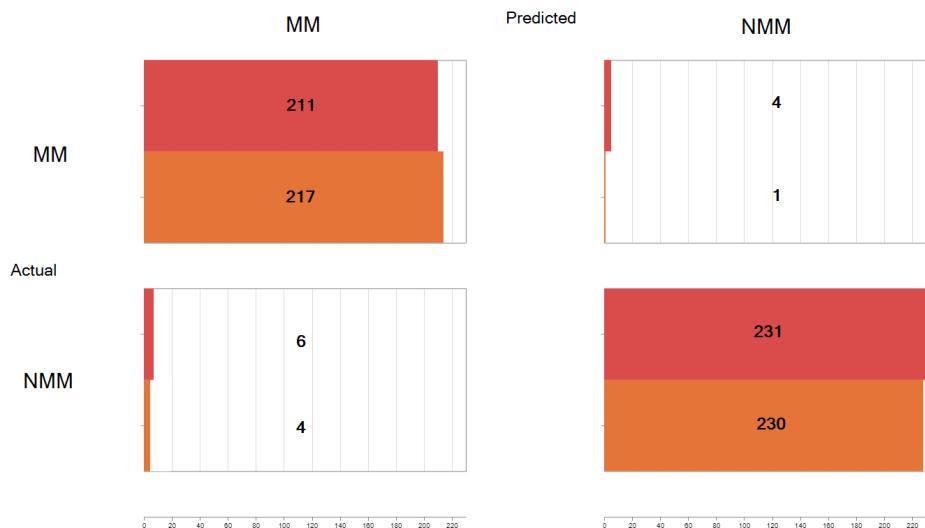
6 (FP)

**NMM (Nicht-  
Mikrometeorit)**

4 (FN)

231 (TN)

$$\text{Gesamt} = 211 + 4 + 6 + 231 = 452$$



## Gütemaße für binäre Klassifikatoren

	Vorhergesagte Klasse		Spaltensummen
	Wahre Klasse	Falsche Klasse	
Zeilensummen	TP: 211	FN: 4	R (Relevanz): 215
	FP: 6	TN: 231	I (Irrelevanz): 237
P (Positivität): 217	N (Negativität): 235	NN: 452	
Gütemaße für binären Klassifikator bestimmten			

# Binäre Klassifikatoren – Gütemaße - Lösung

## Gütemaße für binäre Klassifikatoren

	Vorhergesagte Klasse		Spaltensummen
Wahre Klasse	TP: 211	FN: 4	R (Relevanz): 215
Zeilensummen	FP: 6	TN: 231	I (Irrelevanz): 237
P (Positivität): 217	N (Negativität): 235	NN: 452	

Gütemaße für binären Klassifikator bestimmten

Gütemaß	Wert
TP	211
TN	231
FP	6
FN	4
NN	452
T	442
F	10
R	215
I	237
P	217
N	235
Accuracy	0.9778761061946902
ACC	0.9778761061946902
Inkorrektheitsrate	0.022123893805309734
TPR	0.9813953488372092
Sensitivitaet	0.9813953488372092
Recall	0.9813953488372092
TNR	0.9746835443037974
FPR	0.02531645569620253
FNR	0.018604651162790697
PR	0.9723502304147466
Precision	0.9723502304147466
Negativer Vorhersagewert	0.9829787234042553
Negative Falschklassifikationsrate	0.01702127659574468
Positive Falschklassifikationsrate	0.027649769585253458
Fehlerrate	0.022123893805309734
Erfolgsrate	0.9778761061946902
Relevanz Geron	0.9723502304147466
Spezifitaet	0.9746835443037974
Praevalenz	0.4756637168141593
Positives Likelihood Verhaeltnis	38.76511627906973
Negatives Likelihood Verhaeltnis	0.019087888855330772
F1 Score	0.9768518518518517
ACC0	0.5
KappaKoeffizient	0.9557522123893805
Distance ROC Kurve	0.03141744696672403
AUC	0.9780394465705033

**Tatsächlich /  
Vorhergesagt**

**MM (Mikrometeorit)**

211 (TP)

**NMM (Nicht-  
Mikrometeorit)**

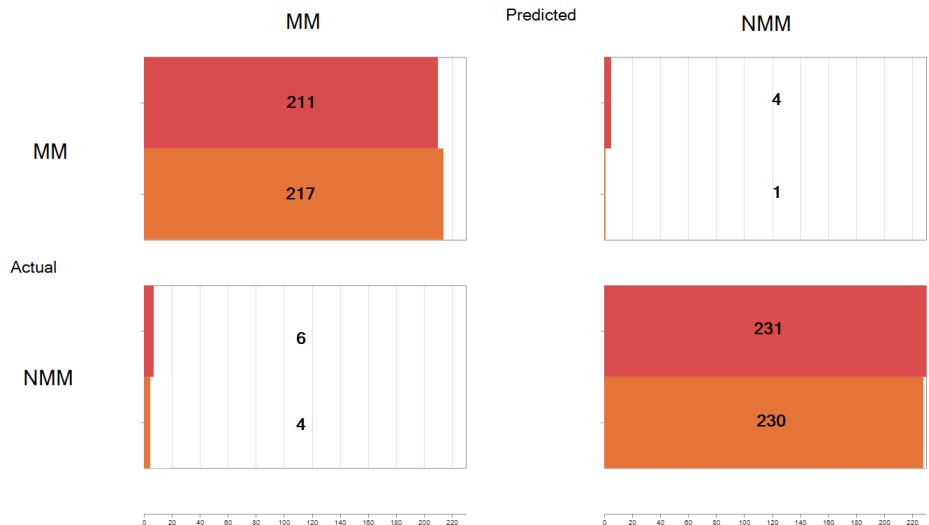
4 (FN)

**NMM (tatsächlich)**

6 (FP)

231 (TN)

Gesamt =  $217 + 1 + 4 + 230 = 452$



## Gütemaße für binäre Klassifikatoren

	Vorhergesagte Klasse		Spaltensummen
Wahre	TP: 217	FN: 1	R (Relevanz): 218
Klasse	FP: 4	TN: 230	I (Irrelevanz): 234
Zeilensummen	P (Positivität): 221	N (Negativität): 231	NN: 452
Gütemaße für binären Klassifikator bestimmten			

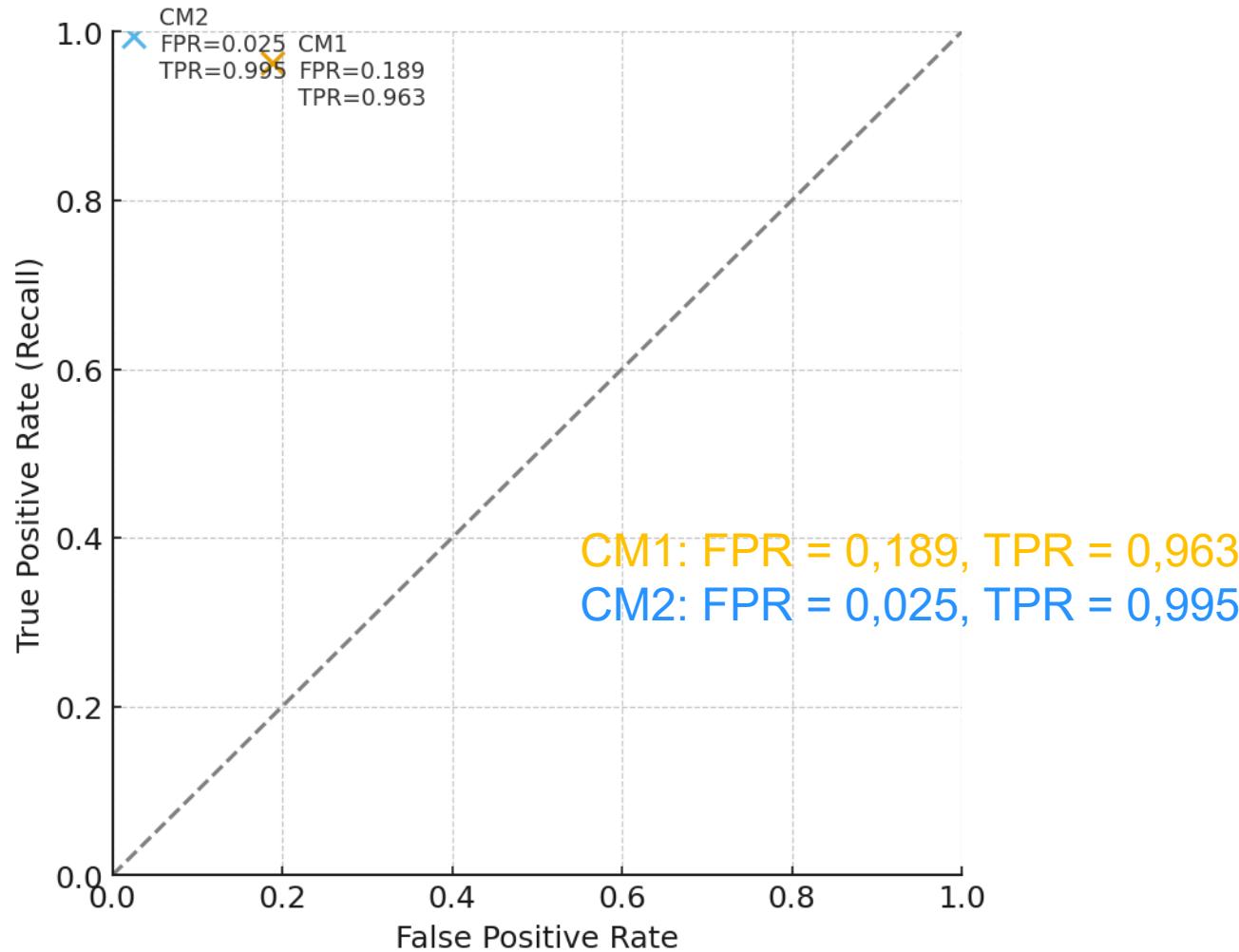
# Binäre Klassifikatoren – Gütemaße - Lösung

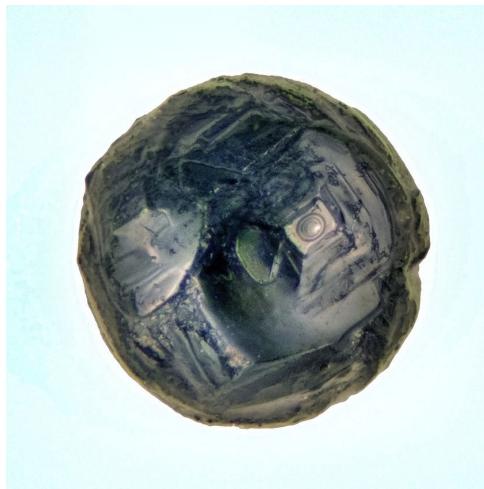
## Gütemaße für binäre Klassifikatoren

	Vorhergesagte Klasse		Spaltensummen
Wahre Klasse	TP: 217	FN: 1	R (Relevanz): 218
	FP: 4	TN: 230	I (Irrelevanz): 234
Zeilensummen	P (Positivität): 221	N (Negativität): 231	NN: 452
Gütemaße für binären Klassifikator bestimmten			

Gütemaß	Wert
TP	217
TN	230
FP	4
FN	1
NN	452
T	447
F	5
R	218
I	234
P	221
N	231
Accuracy	0.9889380530973452
ACC	0.9889380530973452
Inkorrektheitsrate	0.011061946902654867
TPR	0.9954128440366973
Sensitivitaet	0.9954128440366973
Recall	0.9954128440366973
TNR	0.9829059829059829
FPR	0.017094017094017096
FNR	0.0045871559633027525
PR	0.9819004524886877
Precision	0.9819004524886877
Negativer Vorhersagewert	0.9956709956709957
Negative Falschklassifikationsrate	0.004329004329004329
Positive Falschklassifikationsrate	0.01809954751131222
Fehlerrate	0.011061946902654867
Erfolgsrate	0.9889380530973452
Relevanz Geron	0.9819004524886877
Spezifitaet	0.9829059829059829
Praevalenz	0.4823008849557522
Positives Likelihood Verhaeltnis	58.23165137614662
Negatives Likelihood Verhaeltnis	0.004666932588751468
F1 Score	0.9886104783599088
ACCO	0.5
KappaKoeffizient	0.9778761061946903
Distance ROC Kurve	0.017698797141111382
AUC	0.98915941347134

ROC: Punkte aus gegebenen Konfusionsmatrizen





## **Exkurs: Konfusionsmatrizen & Qualitätsmaße am Beispiel Mikrometeoriten vs. Nicht- Mikrometeoriten**

**Datenbasis**

- 1.000 Bilder je Szenario
- Spalten = wahre Klasse
- Zeilen = vorhergesagte Klasse

	<b>Wahre Klasse: MM</b>	<b>Wahre Klasse: NichtMM</b>
<b>Vorhergesagt: MM</b>	TP	FP
<b>Vorhergesagt: NichtMM</b>	FN	TN
Summen		

## Szenario A: Gutes, balanciertes Modell (500 Mikro / 500 Nicht)

	Wahre Klasse: MM	Wahre Klasse: NichtMM
Vorhergesagt: MM	TP = 460	FP = 40
Vorhergesagt: NichtMM	FN = 40	TN = 460
Summen	500	500

TP = True Positives • TN = True Negatives • FP = False Positives • FN = False Negatives

### Qualitätsmaße (Formel + eingesetzte Zahlen)

- Accuracy:  $(TP+TN)/N = (460+460)/1000 = 0.9200$
- Precision (PPV):  $TP/(TP+FP) = 460/(460+40) = 0.9200$
- Recall (Sensitivity/TPR):  $TP/(TP+FN) = 460/(460+40) = 0.9200$
- Specificity (TNR):  $TN/(TN+FP) = 460/(460+40) = 0.9200$
- F1-Score:  $2 \cdot TP/(2 \cdot TP+FP+FN) = 920/(920+40+40) = 0.9200$
- FPR:  $FP/(FP+TN) = 40/(40+460) = 0.0800$
- FNR:  $FN/(FN+TP) = 40/(40+460) = 0.0800$
- Balanced Accuracy:  $(TPR+TNR)/2 = (0.9200+0.9200)/2 = 0.9200$
- NPV:  $TN/(TN+FN) = 460/(460+40) = 0.9200$
- MCC:  $((TP \cdot TN) - (FP \cdot FN)) / \sqrt{((TP+FP)(TP+FN)(TN+FP)(TN+FN))} = ((460 \cdot 460) - (40 \cdot 40)) / \sqrt{(460+40)(460+40)(460+40)(460+40)} = 0.8400$

## Szenario B: Konservativ (hohe Präzision, geringere Sensitivität), 200/800

	Wahre Klasse: MM	Wahre Klasse: NichtMM
Vorhergesagt: MM	TP = 140	FP = 20
Vorhergesagt: NichtMM	FN = 60	TN = 780
Summen	200	800

TP = True Positives • TN = True Negatives • FP = False Positives • FN = False Negatives

### Qualitätsmaße (Formel + eingesetzte Zahlen)

- Accuracy:  $(TP+TN)/N = (140+780)/1000 = 0.9200$
- Precision (PPV):  $TP/(TP+FP) = 140/(140+20) = 0.8750$
- Recall (Sensitivity/TPR):  $TP/(TP+FN) = 140/(140+60) = 0.7000$
- Specificity (TNR):  $TN/(TN+FP) = 780/(780+20) = 0.9750$
- F1-Score:  $2 \cdot TP / (2 \cdot TP + FP + FN) = 280 / (280 + 20 + 60) = 0.7778$
- FPR:  $FP / (FP + TN) = 20 / (20 + 780) = 0.0250$
- FNR:  $FN / (FN + TP) = 60 / (60 + 140) = 0.3000$
- Balanced Accuracy:  $(TPR + TNR) / 2 = (0.7000 + 0.9750) / 2 = 0.8375$
- NPV:  $TN / (TN + FN) = 780 / (780 + 60) = 0.9286$
- MCC:  $((TP \cdot TN) - (FP \cdot FN)) / \sqrt{((TP+FP)(TP+FN)(TN+FP)(TN+FN))} = ((140 \cdot 780) - (20 \cdot 60)) / \sqrt{(140+20)(140+60)(780+20)(780+60)} = 0.7365$

## Szenario C: Aggressiv (hohe Sensitivität, geringere Präzision), 200/800

	Wahre Klasse: MM	Wahre Klasse: NichtMM
Vorhergesagt: MM	TP = 180	FP = 220
Vorhergesagt: NichtMM	FN = 20	TN = 580
Summen	200	800

TP = True Positives • TN = True Negatives • FP = False Positives • FN = False Negatives

### Qualitätsmaße (Formel + eingesetzte Zahlen)

- Accuracy:  $(TP+TN)/N = (180+580)/1000 = 0.7600$
- Precision (PPV):  $TP/(TP+FP) = 180/(180+220) = 0.4500$
- Recall (Sensitivity/TPR):  $TP/(TP+FN) = 180/(180+20) = 0.9000$
- Specificity (TNR):  $TN/(TN+FP) = 580/(580+220) = 0.7250$
- F1-Score:  $2 \cdot TP / (2 \cdot TP + FP + FN) = 360 / (360 + 220 + 20) = 0.6000$
- FPR:  $FP / (FP + TN) = 220 / (220 + 580) = 0.2750$
- FNR:  $FN / (FN + TP) = 20 / (20 + 180) = 0.1000$
- Balanced Accuracy:  $(TPR + TNR) / 2 = (0.9000 + 0.7250) / 2 = 0.8125$
- NPV:  $TN / (TN + FN) = 580 / (580 + 20) = 0.9667$
- MCC:  $((TP \cdot TN) - (FP \cdot FN)) / \sqrt{((TP+FP)(TP+FN)(TN+FP)(TN+FN))} = ((180 \cdot 580) - (220 \cdot 20)) / \sqrt{(180+220)(180+20)(580+220)(580+20)} = 0.5103$

	Wahre Klasse: MM	Wahre Klasse: NichtMM
Vorhergesagt: MM	TP = 0	FP = 0
Vorhergesagt: NichtMM	FN = 200	TN = 800
Summen	200	800

TP = True Positives • TN = True Negatives • FP = False Positives • FN = False Negatives

## Qualitätsmaße (Formel + eingesetzte Zahlen)

- Accuracy:  $(TP+TN)/N = (0+800)/1000 = 0.8000$
- Precision (PPV):  $TP/(TP+FP) = \text{nicht definiert}$  ( $TP+FP=0$ )
- Recall (Sensitivity/TPR):  $TP/(TP+FN) = 0/(0+200) = 0.0000$
- Specificity (TNR):  $TN/(TN+FP) = 800/(800+0) = 1.0000$
- F1-Score:  $2 \cdot TP / (2 \cdot TP + FP + FN) = 0/(0+0+200) = 0.0000$
- FPR:  $FP/(FP+TN) = 0/(0+800) = 0.0000$
- FNR:  $FN/(FN+TP) = 200/(200+0) = 1.0000$
- Balanced Accuracy:  $(TPR+TNR)/2 = (0.0000+1.0000)/2 = 0.5000$
- NPV:  $TN/(TN+FN) = 800/(800+200) = 0.8000$
- MCC:  $((TP \cdot TN) - (FP \cdot FN)) / \sqrt{((TP+FP)(TP+FN)(TN+FP)(TN+FN))} = \text{nicht definiert}$

**Szenario E: Zufall auf balancierten Klassen (500/500)**

	Wahre Klasse: MM	Wahre Klasse: NichtMM
Vorhergesagt: MM	TP = 250	FP = 250
Vorhergesagt: NichtMM	FN = 250	TN = 250
Summen	500	500

TP = True Positives • TN = True Negatives • FP = False Positives • FN = False Negatives

**Qualitätsmaße (Formel + eingesetzte Zahlen)**

- Accuracy:  $(TP+TN)/N = (250+250)/1000 = 0.5000$
- Precision (PPV):  $TP/(TP+FP) = 250/(250+250) = 0.5000$
- Recall (Sensitivity/TPR):  $TP/(TP+FN) = 250/(250+250) = 0.5000$
- Specificity (TNR):  $TN/(TN+FP) = 250/(250+250) = 0.5000$
- F1-Score:  $2 \cdot TP / (2 \cdot TP + FP + FN) = 500 / (500 + 250 + 250) = 0.5000$
- FPR:  $FP / (FP + TN) = 250 / (250 + 250) = 0.5000$
- FNR:  $FN / (FN + TP) = 250 / (250 + 250) = 0.5000$
- Balanced Accuracy:  $(TPR + TNR) / 2 = (0.5000 + 0.5000) / 2 = 0.5000$
- NPV:  $TN / (TN + FN) = 250 / (250 + 250) = 0.5000$
- MCC:  $((TP \cdot TN) - (FP \cdot FN)) / \sqrt{((TP+FP)(TP+FN)(TN+FP)(TN+FN))} = ((250 \cdot 250) - (250 \cdot 250)) / \sqrt{(250+250)(250+250)(250+250)(250+250)} = 0.0000$

## Hinweise zur Interpretation

---

- Positive Klasse ist „Mikrometeorit“.
- Spalten = wahre Klasse, Zeilen = vorhergesagte Klasse (einheitlich umgesetzt).
- Szenario B minimiert False Positives (hohe Präzision), übersieht aber mehr Mikrometeoriten (niedriger Recall).
- Szenario C maximiert Recall, erzeugt dafür mehr False Positives (niedrige Präzision).
- Szenario D zeigt die Tücke unausgewogener Klassenverteilungen: hohe scheinbare Accuracy, aber Recall = 0.
- Balanced Accuracy und MCC sind robustere Maße bei Klassenungleichgewicht.

## Wende Deltaregel auf xor an

A	$\neg A$	A	B	A	B	$A \rightarrow B$	$A \leftrightarrow B$
T	F	T	T	T	T	T	T
F	T	T	F	F	T	F	F
		F	T	F	T	T	F
		F	F	F	F	T	T

A	B	A	B	$A \mid B$	A	B
T	T	F		F	F	
T	F	T		T	F	
F	T	T		T	F	
F	F	F		T	T	

$$w_i' = w_i + \lambda \cdot (t_x - o_x) \cdot o_i$$

Implementiere in Excel, Python, ...

$w'$  = neues Gewicht,  $w$  = altes Gewicht,  $\lambda$  (lambda) = Lernrate,  $t_x$  = Trainingslabel (Soll- bzw. Lehrwert)  
 $o_x$  = Ausgabewert (berechneter Wert),  $o_i$  = Eingabewert

## Auswertung Mikrometeoriten-Klassifikation

- Quelle: N=517
- Spalten: wahre Klasse (Spalten), vorhergesagt (Zeilen)

### Definitionen & Formeln

- TPR/Recall =  $TP / (TP+FN)$
- FPR =  $FP / (FP+TN)$
- Precision (PPV) =  $TP / (TP+FP)$
- $F1 = 2 \cdot TP / (2 \cdot TP + FP + FN)$
- $MCC = (TP \cdot TN - FP \cdot FN) / \sqrt{((TP+FP)(TP+FN)(TN+FP)(TN+FN))}$
- Youden J =  $TPR - FPR$  (balanciert Sensitivität/Spezifität).

# Aufgabe: Grenzen (Limitierungen) einfacher Perzeptrons

Anpassung der Gewichte nach der Regel:  $w' = w + \lambda \cdot o_i \cdot (t_x - o_x)$

Lernrate Lambda	Delta Regel: Logisches xor, Activation ox: Aktivierungsfunktion = 0 wenn Net < 0.5, sonst 1							
	oi	Input-neuronen	w	Output-neuron	tx training label	Net = wi*oi	ox	Delta tx-ox
Startphase - Gewicht 0	1	1	0,000	1	0	0,000	0	0,000
	1	2	0,000					
Gewichte nach 1. Trainingphase	1	1	0,000	1	1	0,000	0	-1,000
	0	2	0,000					
Gewichte nach 2. Trainingsatz	1	1	-0,200	1	1	-0,200	0	-1,000
	0	2	0,000					
Gewichte nach 3. Trainingsatz	1	1	-0,400	1	1	-0,400	0	-1,000
	0	2	0,000					
Gewichte nach 4. Trainingsatz	1	1	-0,600	1	1	-0,600	0	-1,000
	0	2	0,000					
Gewichte nach 5. Trainingsatz	0	1	-0,800	1	1	0,000	0	-1,000
	1	2	0,000					
Gewichte nach 6. Trainingsatz	0	1	-0,800	1	1	-0,200	0	-1,000
	1	2	-0,200					
Gewichte nach 7. Trainingsatz	0	1	-0,800	1	1	-0,400	0	-1,000
	1	2	-0,400					
Gewichte nach 8. Trainingsatz	0	1	-0,800	1	1	-0,600	0	-1,000
	1	2	-0,600					
Gewichte nach 9. Trainingsatz	0	1	-0,800	1	0	0,000	0	0,000
	0	2	-0,800					
Test		1	-0,800	1	0	-1,600	0	0,000
		1	-0,800					

$w'$  = neues Gewicht,  $w$  = altes Gewicht,  $\lambda$  (lambda) = Lernrate,  $t_x$  = Trainingslabel (Soll- bzw. Lehrwert)

$o_x$  = Ausgabewert (berechneter Wert),  $o_i$  = Eingabewert

## Auswertung Mikrometeoriten-Klassifikation

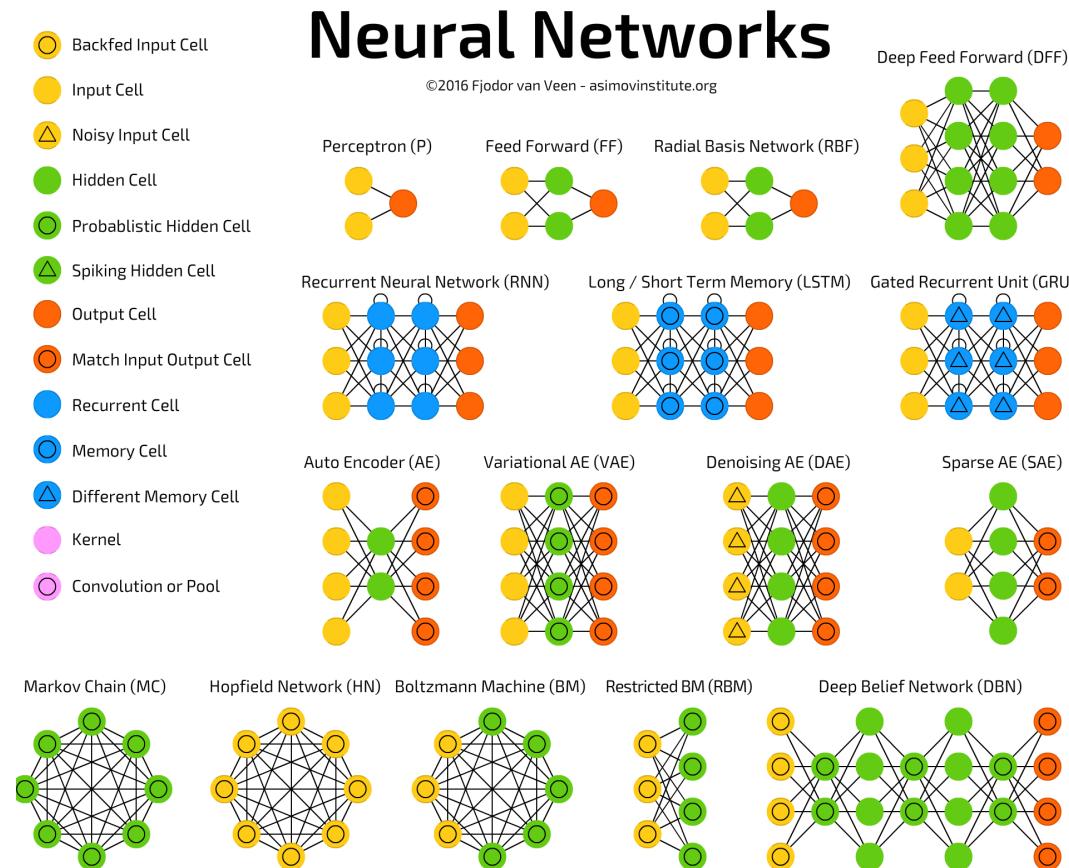
- Quelle: N=517
- Spalten: wahre Klasse (Spalten), vorhergesagt (Zeilen)

### Definitionen & Formeln

- TPR/Recall =  $TP / (TP+FN)$
- FPR =  $FP / (FP+TN)$
- Precision (PPV) =  $TP / (TP+FP)$
- $F1 = 2 \cdot TP / (2 \cdot TP + FP + FN)$
- $MCC = (TP \cdot TN - FP \cdot FN) / \sqrt{((TP+FP)(TP+FN)(TN+FP)(TN+FN))}$
- Youden J =  $TPR - FPR$  (balanciert Sensitivität/Spezifität).

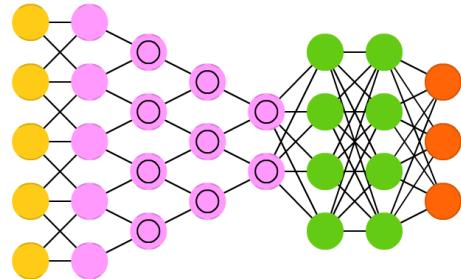
**Wählen Sie zwei der folgenden NN-Varianten und bereiten Sie zwei Folien vor, die die Grundlagen dieser Ansätze erklären**

- <https://www.asimovinstitute.org/neural-network-zoo/>

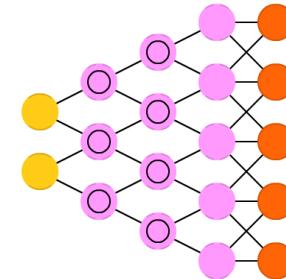


**Wählen Sie zwei der folgenden NN-Varianten und bereiten Sie zwei Folien vor, die die Grundlagen dieser Ansätze erklären**

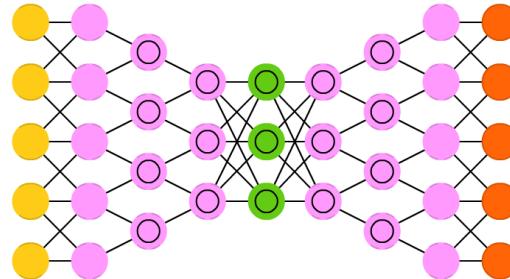
Deep Convolutional Network (DCN)



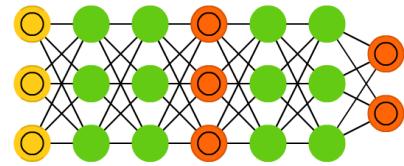
Deconvolutional Network (DN)



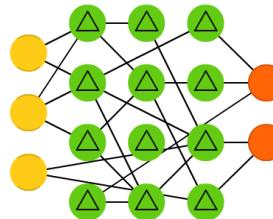
Deep Convolutional Inverse Graphics Network (DCIGN)



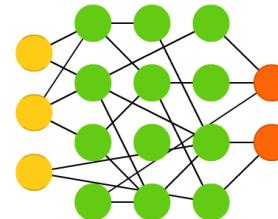
Generative Adversarial Network (GAN)



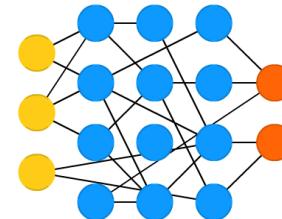
Liquid State Machine (LSM)



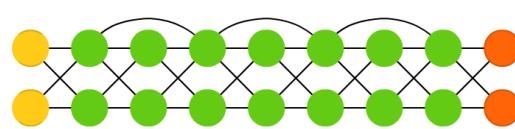
Extreme Learning Machine (ELM)



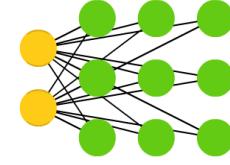
Echo State Network (ESN)



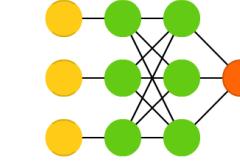
Deep Residual Network (DRN)



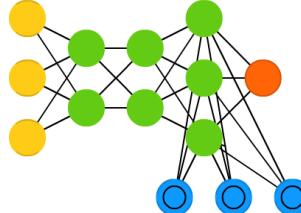
Kohonen Network (KN)



Support Vector Machine (SVM)



Neural Turing Machine (NTM)



# Übung: Boltzmannmaschine – Neural Net Zoo

Eine Boltzmannmaschine (RBM = Restricted Boltzmann Machine) ist ein generatives neuronales Netzwerk, das lernt, wie typische Muster in Daten aussehen.

Es besteht aus:

- sichtbaren Neuronen → enthalten die Eingangsdaten (z. B. Sensormessungen)
- versteckten Neuronen → lernen verborgene Strukturen und Muster
- bidirektionalen Gewichten → modellieren Zusammenhänge zwischen Variablen

Beim Training lernt die Boltzmannmaschine Wahrscheinlichkeitsverteilungen:

- Welche Kombinationen von Sensordaten kommen häufig gemeinsam vor?
- Welche sind typisch für eine bestimmte Aktivität?

Eine RBM kann danach:

- neue Daten rekonstruieren
- ungewöhnliche oder fremde Muster erkennen
- Ähnlichkeiten zwischen Aktivitäten messen (über Rekonstruktionsfehler)

## Schichten RBM

Sie besteht aus genau zwei Schichten

- einer sichtbaren Schicht und
- einer versteckten Schicht
- Ausgabe ist rekonstruierter Input (keine Ausgabeschicht)
- Das Beispielmodell umfasst 26 Eingabeneuronen und 20 Neuronen in der versteckten Schicht.

## Konnektivität

Das wichtigste Merkmal ist die Beschränkung (Restricted): Jede Einheit der sichtbaren Schicht ist mit jeder Einheit der versteckten Schicht verbunden. Es gibt keine Verbindungen innerhalb der sichtbaren Schicht (visible-visible) und keine Verbindungen innerhalb der versteckten Schicht (hidden-hidden).

## Vorteil

Diese Beschränkung vereinfacht den Trainingsprozess drastisch, da die Aktivierungen der Neuronen in jeder Schicht, gegeben die Aktivierungen der anderen Schicht, unabhängig voneinander werden. Sie dienen oft als Bausteine für tiefere Architekturen.

Für die folgenden Beschreibungen werden Sensordaten aus der Erfassung von Aktivitäten, die mit einer Smartwatch erhoben wurden, verwendet.

"ACC.MEAN\_X","ACC.MEAN\_Y","ACC.MEAN\_Z","ACC.SD\_X","ACC.SD\_Y","ACC.SD\_Z","ACC.IQR\_X","ACC.IQR\_Y","ACC.IQR\_Z","ACC.ZC\_X","ACC.ZC\_Z","ACC.QMW","ACC.FFT.DC\_COMP\_X","ACC.FFT.MAGNITUDE\_RATIO\_X","ACC.FFT.MAGNITUDE\_MEDIAN\_X","ACC.FFT.X1","ACC.FFT.X2","ACC.FFT.X3","ACC.FFT.X4","ACC.FFT.X5","ACC.FFT.X6","ACC.FFT.X7","ACC.FFT.X8","ACC.FFT.X9","ACC.FFT.X10","ACC.FFT.DC\_COMP\_Y","ACC.FFT.MAGNITUDE\_RATIO\_Y","ACC.FFT.MAGNITUDE\_MEDIAN\_Y","ACC.FFT.Y1","ACC.FFT.Y2","ACC.FFT.Y3","ACC.FFT.Y4","ACC.FFT.Y5","ACC.FFT.Y6","ACC.FFT.Y7","ACC.FFT.Y8","ACC.FFT.Y9","ACC.FFT.Y10","ACC.FFT.DC\_COMP\_Z","ACC.FFT.MAGNITUDE\_RATIO\_Z","ACC.FFT.MAGNITUDE\_MEDIAN\_Z","ACC.FFT.Z1","ACC.FFT.Z2","ACC.FFT.Z3","ACC.FFT.Z4","ACC.FFT.Z5","ACC.FFT.Z6","ACC.FFT.Z7","ACC.FFT.Z8","ACC.FFT.Z9","ACC.FFT.Z10","GYR.MEAN\_X","GYR.MEAN\_Y","GYR.MEAN\_Z","GYR.SD\_X","GYR.SD\_Y","GYR.SD\_Z","GYR.IQR\_X","GYR.IQR\_Y","GYR.IQR\_Z","GYR.ZC\_X","GYR.ZC\_Y","GYR.ZC\_Z","GYR.QMW","GYR.FFT.DC\_COMP\_X","GYR.FFT.MAGNITUDE\_RATIO\_X","GYR.FFT.MAGNITUDE\_MEDIAN\_X","GYR.FFT.X1","GYR.FFT.X2","GYR.FFT.X3","GYR.FFT.X4","GYR.FFT.X5","GYR.FFT.X6","GYR.FFT.X7","GYR.FFT.X8","GYR.FFT.X9","GYR.FFT.X10","GYR.FFT.DC\_COMP\_Y","GYR.FFT.MAGNITUDE\_RATIO\_Y","GYR.FFT.MAGNITUDE\_MEDIAN\_Y","GYR.FFT.Y1","GYR.FFT.Y2","GYR.FFT.Y3","GYR.FFT.Y4","GYR.FFT.Y5","GYR.FFT.Y6","GYR.FFT.Y7","GYR.FFT.Y8","GYR.FFT.Y9","GYR.FFT.Y10","GYR.FFT.DC\_COMP\_Z","GYR.FFT.MAGNITUDE\_RATIO\_Z","GYR.FFT.MAGNITUDE\_MEDIAN\_Z","GYR.FFT.Z1","GYR.FFT.Z2","GYR.FFT.Z3","GYR.FFT.Z4","GYR.FFT.Z5","GYR.FFT.Z6","GYR.FFT.Z7","GYR.FFT.Z8","GYR.FFT.Z9","GYR.FFT.Z10","MAG.MEAN\_X","MAG.MEAN\_Y","MAG.MEAN\_Z","MAG.SD\_X","MAG.SD\_Y","MAG.SD\_Z","MAG.IQR\_X","MAG.IQR\_Y","MAG.IQR\_Z","MAG.ZC\_X","MAG.ZC\_Y","MAG.ZC\_Z","MAG.QMW","MAG.FFT.DC\_COMP\_X","MAG.FFT.MAGNITUDE\_RATIO\_X","MAG.FFT.MAGNITUDE\_MEDIAN\_X","MAG.FFT.X1","MAG.FFT.X2","MAG.FFT.X3","MAG.FFT.X4","MAG.FFT.X5","MAG.FFT.X6","MAG.FFT.X7","MAG.FFT.X8","MAG.FFT.X9","MAG.FFT.X10","MAG.FFT.DC\_COMP\_Y","MAG.FFT.MAGNITUDE\_RATIO\_Y","MAG.FFT.MAGNITUDE\_MEDIAN\_Y","MAG.FFT.Y1","MAG.FFT.Y2","MAG.FFT.Y3","MAG.FFT.Y4","MAG.FFT.Y5","MAG.FFT.Y6","MAG.FFT.Y7","MAG.FFT.Y8","MAG.FFT.Y9","MAG.FFT.Y10","MAG.FFT.DC\_COMP\_Z","MAG.FFT.MAGNITUDE\_RATIO\_Z","MAG.FFT.MAGNITUDE\_MEDIAN\_Z","MAG.FFT.Z1","MAG.FFT.Z2","MAG.FFT.Z3","MAG.FFT.Z4","MAG.FFT.Z5","MAG.FFT.Z6","MAG.FFT.Z7","MAG.FFT.Z8","MAG.FFT.Z9","MAG.FFT.Z10","BARO.MEAN\_GPX","BARO.SD\_GPX","BARO.IQR\_GPX","BARO.ZC\_GPX","BARO.MAX\_GPX","BARO.MIN\_GPX","BARO.DIFF\_GPX","ACTIVITY","-2.70411949211672",  
7.65867160894374,"4.34083573194055","2.27780897237077","2.55596738618775","5.4527854529081","2.88215464353561","2.51844620704661",7.58823879063129,"56","8","59","11.2334704373903","3.79980278015137",0.0148849753775352,"27.753881661472",380,"390","1160","1560","1150","350",370,"1550","1480","2450",7.86281591863823,0.00915072439083257,41.4408221495126,"1980",740,"1580",780,"1560",1550,810,"1010",1870,"1140",13.3352008673813,"0.0201823455682048",49.5338477717013,380,"390","1150",350,"1160",370,"1560",1140,"1540",1550,3.1784892581883,-0.339521420489856,-1.06787421983862,"89.1203604101504","73.7233183049615",28.1785776818284,"120.843935966492",125.958037521362,37.4327793121337,109,79,"81",118.988250298747,"135.081516265869",0.00888953627416838,"1009.38816257015",380,"1150",4440,"4950",840,"850",4940,"390",350,"340",83.1975212097168,0.0132897392505055,"701.721434340368",380,"390",350,"370",3210,"2380",400,"1150",4440,"440",52.4333591461181,"0.00948918964461963",290.636523261283,"350",380,"400",390,"1150",4440,"780",4810,"760",4830,"15.2909009009009",15.3387387387387,-,33.1495495495495,"3.60459896888041",7.19804840405437,"3.9807449683539",5.285,"7.585",6.33,"0",5,"0",40.5921176056597,14.22,0.0404687292571002,"31.3771141890212",700,"380",340,"330",370,"710",440,"640",230,"870",33.23,0.0329755885570842,"36.4397860003034",330,"120",340,"150",310,"140",300,"270",60,"180",15.97,0.0425989542804785,"28.2334859318786",340,"710",370,"330",380,"700",680,"400",250,"390",980.12383333333,"0.059577372188062",0.0549999999999995,"0",980.42,"980.03",0.3899999999999986,"Cardmix"

## Ziel der Implementierung Boltzmannmaschine

Die implementierte Lösung nutzt eine RBM pro Aktivität, um typische Muster dieser Aktivität zu lernen und Aktivitäten zu unterscheiden.

## Ziele

- Pro-Activity-Modell  
Jede Aktivität (Drink, EShave, ETooth ...) erhält ihr eigenes RBM-Modell, das nur auf deren Daten trainiert wird.
- Erkennung der Activity  
Ein neues Sensormuster wird an alle RBMs gegeben.  
Das Modell, das es am besten rekonstruieren kann, ist die vorhergesagte Aktivität.

## Evaluation

- Konfusionsmatrix (Testdaten)
- Gesamt-Konfusionsmatrix (Train + Test)
- Precision, Recall, F1, Accuracy
- Rekonstruktionsfehler-Heatmaps
- Logging & Speicherung der Modelle

## Klassifikation im Modell

- Funktionale Ausgabe sind der rekonstruierten Input und der Rekonstruktionsfehler.
- Die „Ausgabe der RBM“ ist die Rekonstruktion des Inputs aus den versteckten Neuronen:  
 $\text{Input-Feature} \rightarrow \text{RBM} \rightarrow \text{Rekonstruiertes Feature}$   
↓  
Fehlermaß als Output
  - Input → Visible Layer
  - Berechne Aktivierung Hidden Layer
  - Rekonstruiere wieder Visible Layer → aus den Hidden-Neuronen
- Für jede Aktivität wird ein RBM-Modell trainiert.  
In diesem Beispiel vier Modelle: 'Drink': 'EShave': 'ETooth': 'Other'

Für ein unbekanntes Sensorsample:

- Alle RBMs versuchen das Sensorsample zu rekonstruieren.
- Diejenige RBM gewinnt, die den kleinsten Rekonstruktionsfehler oder die niedrigste Energie liefert.
- Das ist dann die vorhergesagte Aktivität.

==== Klassenverteilung nach Gruppierung

====

```
'Drink': 110 Samples  
'EShave': 132 Samples  
'ETooth': 317 Samples  
'Other': 372 Samples
```

==== Konfusionsmatrix ===

	pred_Drink	pred_EShave	pred_ETooth	pred_Other
true_Drink	28	0	1	4
true_EShave	0	40	0	0
true_ETooth	7	36	26	26
true_Other	3	20	2	87

==== Klassifikationsbericht (Precision, Recall, F1) ===

	precision	recall	f1-score	support
Drink	0.74	0.85	0.79	33
EShave	0.42	1.00	0.59	40
ETooth	0.90	0.27	0.42	95
Other	0.74	0.78	0.76	112
accuracy			0.65	280
macro avg	0.70	0.72	0.64	280
weighted avg	0.75	0.65	0.62	280

==== Konfusionsmatrix (ALLE Fälle) ===

	pred_Drink	pred_EShave	pred_ETooth	pred_Other
true_Drink	100	0	2	8
true_EShave	0	132	0	0
true_ETooth	23	142	73	79
true_Other	9	71	20	272

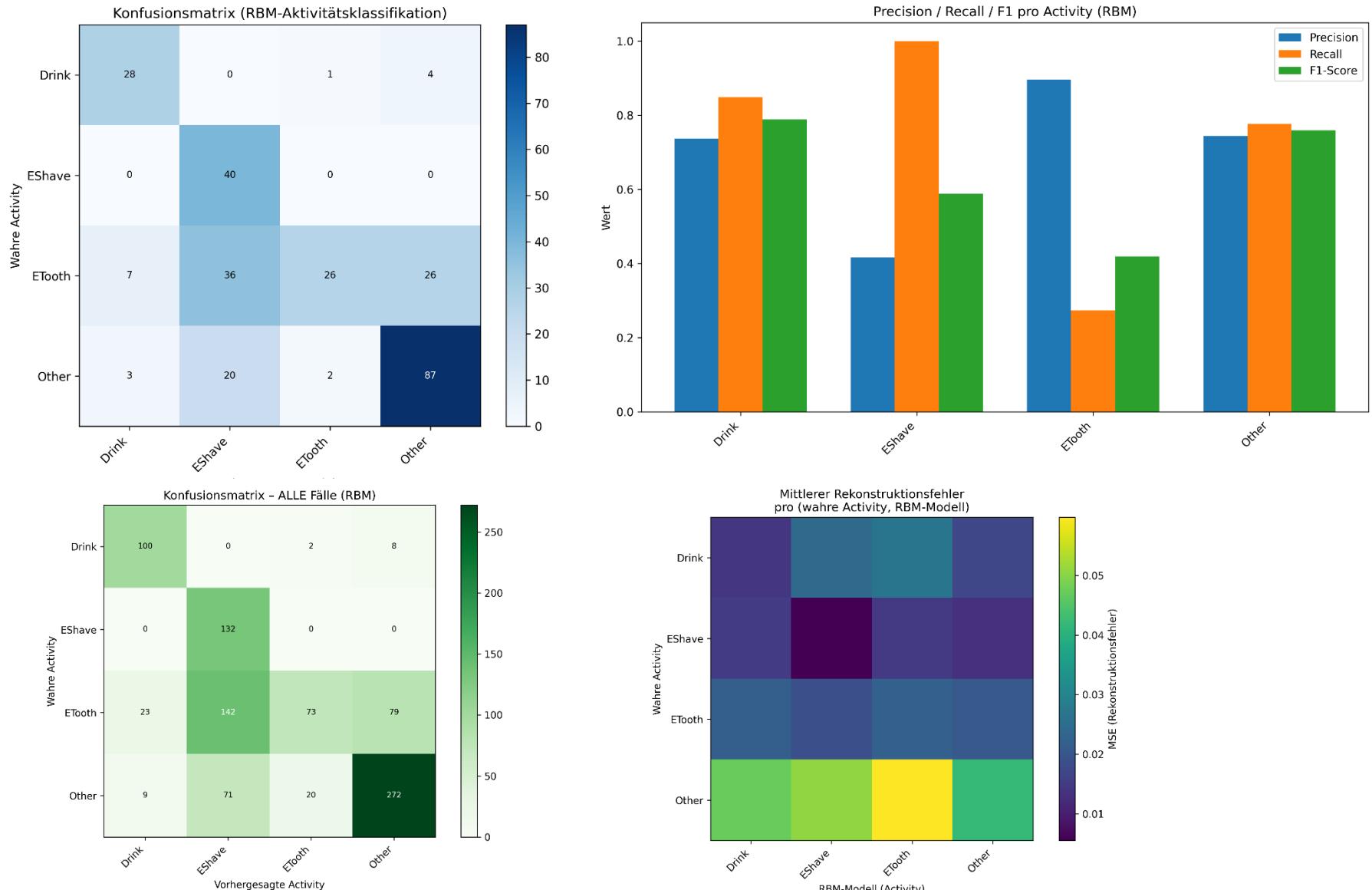
==== Klassifikationsbericht (ALLE Fälle) ===

	precision	recall	f1-score	support
Drink	0.76	0.91	0.83	110
EShave	0.38	1.00	0.55	132
ETooth	0.77	0.23	0.35	317
Other	0.76	0.73	0.74	372
accuracy			0.62	931
macro avg	0.67	0.72	0.62	931
weighted avg	0.71	0.62	0.59	931

Accuracy (ALLE Fälle): 0.6198

# Boltzmannmaschine

## Ergebnisse



# Übung: Autoencoder – Neural Net Zoo

Ein Autoencoder (AE) ist ein künstliches neuronales Netzwerk, das lernt, die wichtigsten Muster in Daten zu erfassen, indem es diese komprimiert und anschließend wieder rekonstruiert.

Es besteht aus:

- Encoder → reduziert die Eingangsdaten auf wenige, aussagekräftige Merkmale
- Latent Space → kompakte Repräsentation der Daten (z. B. Bewegungssignatur)
- Decoder → stellt aus dieser Repräsentation die ursprünglichen Daten wieder her

Beim Training lernt der Autoencoder:

- Welche Komponenten der Sensordaten sind wesentlich?
- Welche Strukturen definieren eine typische Aktivität?

Ein Autoencoder kann danach:

- Daten rekonstruiert zurückgeben
- Abweichungen erkennen (großer Rekonstruktionsfehler = ungewöhnliches Muster)
- typische Aktivitätsmuster voneinander trennen → zur Klassifikation über Rekonstruktionsqualität

## Ziel der Implementierung Autoencoder

Die implementierte Lösung nutzt einen Autoencoder pro Aktivität, um typische Muster dieser Aktivität zu lernen und Aktivitäten zu unterscheiden.

### Ziele

- Pro-Activity-Modell  
Jede Aktivität (Drink, EShave, ETooth ...) erhält ihr eigenes Autoencoder -Modell, das nur auf deren Daten trainiert wird. In diesem Beispiel werden alle Aktivitäten mit weniger als 100 Fällen zur Klasse „Other“ zusammengefasst.
- Das Beispielmodell umfasst 26 Eingabeneuronen und 16 Neuronen in der versteckten Schicht.
- Erkennung der Activity  
Ein neues Sensormuster wird an alle Autoencoder gegeben.  
Das Modell, das es am besten rekonstruieren kann, ist die vorhergesagte Aktivität.

### Evaluation

- Konfusionsmatrix (Testdaten)
- Gesamt-Konfusionsmatrix (Train + Test)
- Precision, Recall, F1, Accuracy
- Rekonstruktionsfehler-Heatmaps
- Logging & Speicherung der Modelle

### Klassifikation im Modell

- Auch beim Autoencoder gilt der rekonstruierte Input und der daraus berechnete Rekonstruktionsfehler als funktionale „Ausgabe“. Er zeigt, wie gut das Modell die Aktivität verstanden hat bzw. wie typisch der Sensorinput ist.
- Für jede Aktivität wird ein Autoencoder-Modell trainiert.  
In diesem Beispiel vier Modelle: 'Drink': 'EShave': 'ETooth': 'Other'

Für ein unbekanntes Sensorsample:

- Alle Autoencoder-Modell versuchen das Sensorsample zu rekonstruieren.
- Dasjenige Autoencoder-Modell gewinnt, das den kleinsten Rekonstruktionsfehler liefert.
- Das ist dann die vorhergesagte Aktivität.

Input → Rekonstruktion → Fehler → Aktivität  
kleiner Fehler → Aktivität passt  
großer Fehler → Aktivität untypisch (→ „Other“ möglich)

==== Klassenverteilung nach Gruppierung

====

```
'Drink': 110 Samples  
'EShave': 132 Samples  
'ETooth': 317 Samples  
'Other': 372 Samples
```

Klassifizierte TEST-Daten mit allen Autoencodern...

==== Konfusionsmatrix (TEST) ===

	pred_Drink	pred_EShave	pred_ETooth	pred_Other
true_Drink	25	0	2	6
true_EShave	0	38	0	2
true_ETooth	0	0	90	5
true_Other	1	0	3	108

==== Klassifikationsbericht (TEST) ===

	precision	recall	f1-score	support
Drink	0.96	0.76	0.85	33
EShave	1.00	0.95	0.97	40
ETooth	0.95	0.95	0.95	95
Other	0.89	0.96	0.93	112
accuracy			0.93	280
macro avg	0.95	0.90	0.92	280
weighted avg	0.93	0.93	0.93	280

Accuracy (TEST): 0.9321

==== Konfusionsmatrix (ALLE Fälle) ===

	pred_Drink	pred_EShave	pred_ETooth	pred_Other
true_Drink	84	0	3	23
true_EShave	0	129	0	3
true_ETooth	0	1	309	7
true_Other	1	0	7	364

==== Klassifikationsbericht (ALLE Fälle) ===

	precision	recall	f1-score	support
Drink	0.99	0.76	0.86	110
EShave	0.99	0.98	0.98	132
ETooth	0.97	0.97	0.97	317
Other	0.92	0.98	0.95	372
accuracy			0.95	931
macro avg	0.97	0.92	0.94	931
weighted avg	0.95	0.95	0.95	931

Accuracy (ALLE Fälle): 0.9517

# Autoencoder Ergebnisse

