Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Validierung

CAS practical machine learning: Validierung

▸ Violeta Vogel, TI BFH

Agenda

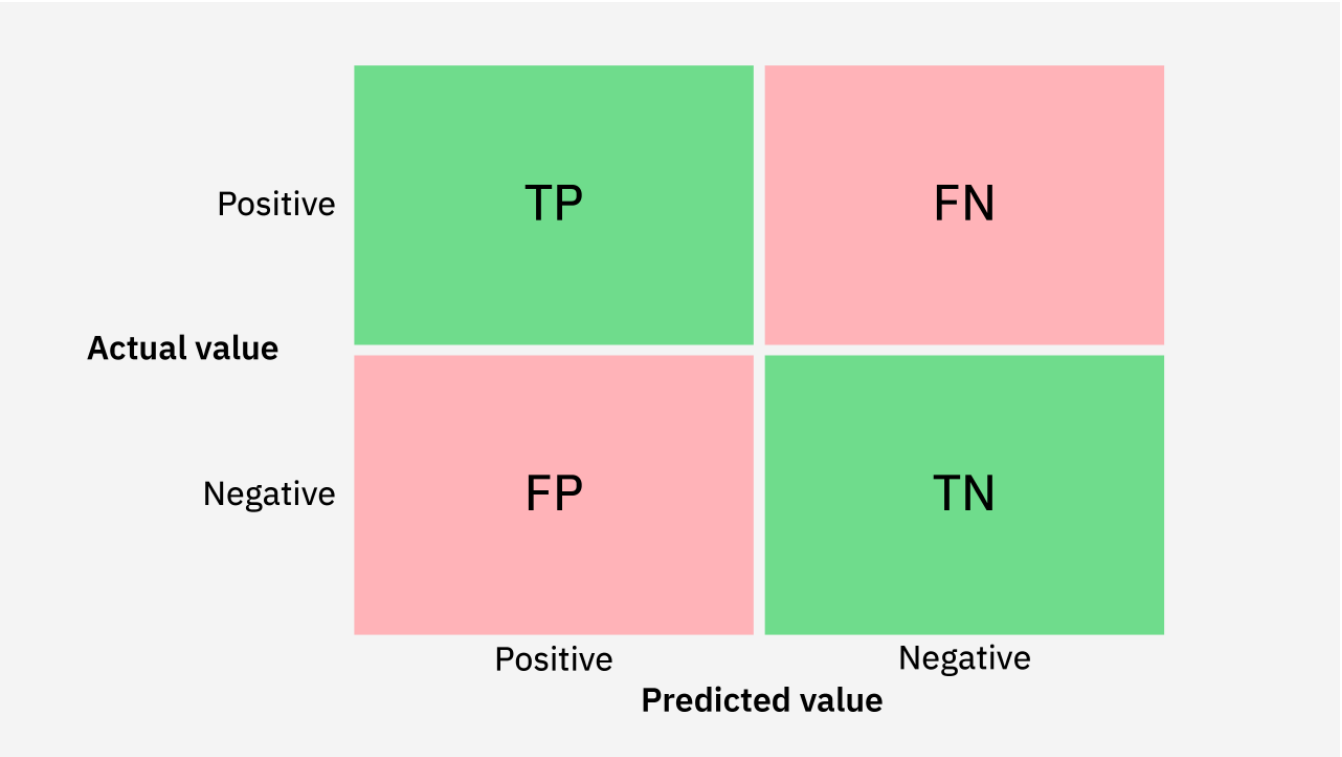
- Konfusionsmatrix
- Precision
- Recal
- F-Score

Konfusionsmatrix

- **True Positives (TP):** Korrekt vorhergesagte positive Fälle
- **True Negatives (TN):** Korrekt vorhergesagte negative Fälle
- **False Positives (FP):** Negativfälle, die fälschlicherweise als positiv vorhergesagt wurden (Fehler vom Typ I)
- **False Negatives (FN):** Positivfälle, die fälschlicherweise als negativ vorhergesagt wurden (Fehler vom Typ II)

Konfusionsmatrix

- Zeilen: Repräsentieren die tatsächlichen Klassen (oder Labels) der Daten
- Spalten: Repräsentieren die vorhergesagten Klassen (oder Labels) durch das Modell



Konfusionsmatrix Vorteile

- Leistungsbewertung:
  - Detaillierte Analyse der Modellleistung, indem sie nicht nur die Gesamtgenauigkeit, sondern auch die Fehlerarten (TP, TN, FP, FN) zeigt
- Modelloptimierung:
  - Durch die Analyse der Fehlerverteilung können Schwachstellen im Modell identifiziert und gezielt verbessert werden
- Vergleich von Modellen:
  - Leistung verschiedener Klassifizierungsmodelle zu vergleichen und das am besten geeignete Modell für eine bestimmte Aufgabe auszuwählen
- Anwendungsspezifische Metriken:
  - Basierend auf der Konfusionsmatrix können verschiedene Metriken berechnet werden, wie z.B. Präzision, Recall, F1-Score

Precision

- Angibt, wie viele der vom Modell als positiv identifizierten Fälle tatsächlich positiv sind
- Verwenden, wenn es sehr wichtig ist, dass positive Vorhersagen genau sind.

		Predicted: Negative	Predicted: Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ Predicted\ Positive}$$

Recall

- angibt, wie viele der tatsächlich positiven Fälle vom Modell als positiv erkannt wurden
- Verwenden, wenn falsch negative Ergebnisse teurer sind als falsch positive Ergebnisse

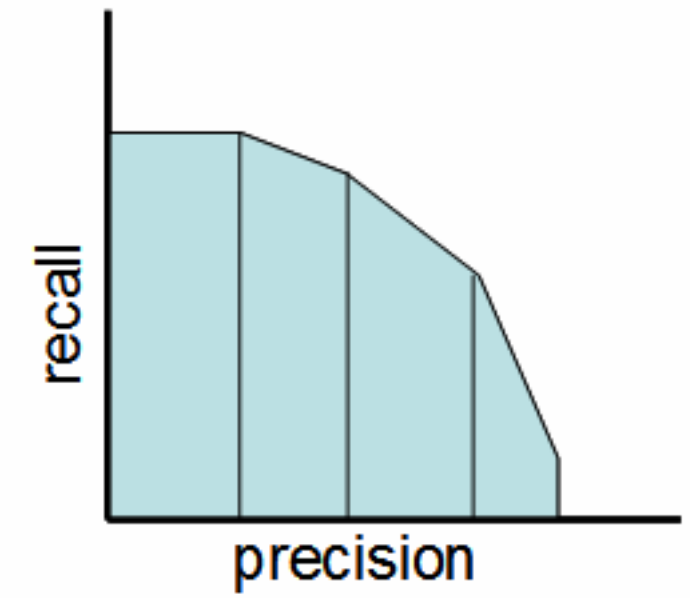
		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

Recall versus Precision

- Die Precision steigt, wenn die Anzahl der falsch positiven Ergebnisse sinkt
- Der Recall steigt, wenn die Anzahl der falsch negativen Ergebnisse sinkt
- Zielkonflikt: Mit steigender Precision sinkt i.d.R. der Recall und umgekehrt



F-Score

- Balance zwischen Precision und Recall
- Insbesondere bei unausgeglichenen Datensätzen wertvoll
- Der F1-Score ist besonders nützlich, wenn sowohl Präzision als auch Rückruf wichtig sind und eine gleichmäßige Gewichtung erhalten sollen.
-

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Praxis: Kreuzvalidierung

- Vorgehen mit scikit-learn (vgl. [ipynb])

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score

model = DecisionTreeClassifier(random_state=1234)

## cross validation
kfold = 10 ## default: 5
scores = cross_val_score(model, X, y, cv=kfold)
```

- cross_val_score gibt einen Array mit den Scores der einzelnen Iterationen zurück
- die beiden Anweisungen können auch kombiniert werden (vgl. [ipynb])

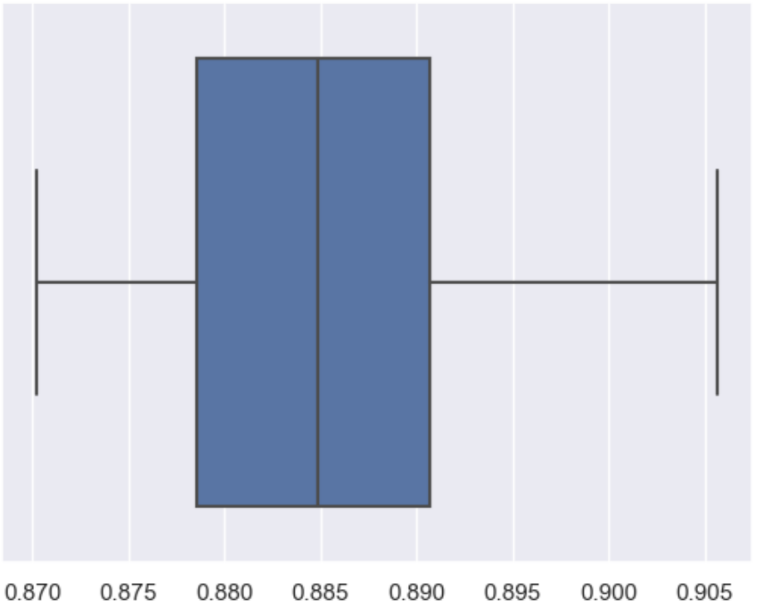
Praxis: Kreuzvalidierung

- sichten der Ergebnisse

```
print('mean:', np.mean(scores))
print('std: ', np.std(scores))
sns.boxplot(x=scores);
```

```
mean: 0.8859026369168358
std:  0.010953816382393995
```

- diese Untersuchung liefert uns ein weiteres wichtiges Ergebnis: die **Stabilität** des untersuchten Learners, d.h.:
  - die Abhängigkeit der Performance von random_state beim Train - Test - Split
- **je kleiner die Standardabweichung (std) umso stabiler das Resultat der jeweiligen Methode**



Praxis: **Workshop 13**

- Zeit: 30'
- vergleichen Sie alle bisher bekannten Klassifikatoren in Bezug auf deren Stabilität unter Anwendung von Kreuzvalidierung
  - verwenden Sie für die Klassifikatoren jeweils Default-Parametrisierung
  - setzen Sie für die Kreuzvalidierung folgende Funktion ein:
  - sklearn.model_selection.cross_val_score