Zeitreihen: fortschreiben oder klassifizieren

Kapitel online

Aus einer Kurve werden Trainingsbeispiele — und dann zwei Fragen an dieselben Daten: Wie geht es weiter (Zahl)? Oder: Endet es im Plus (Etikett)? Mit Zeitreihen-Labor auf drei Datensätzen — Stromverbrauch, Maschinensensor und Börse — in dem du misst, warum Kursvorhersage scheitert und Wartungsvorhersage funktioniert.

Fortgeschritten · Kapitel 27 von 50 · ca. 25 Min. · 250 XP

Einfach gesagt: Ob der Strom morgen um 19 Uhr reicht oder eine Maschine bald ausfällt — dein Netz liest die letzten Werte und schätzt, wie es weitergeht.

Bisher war jede Zeile unserer Tabellen ein *Ding*: ein Kunde, eine Blume, ein Bild — die Reihenfolge war egal, in Kapitel 235 haben wir sie sogar absichtlich gemischt. Heute ist die Reihenfolge **die ganze Information**.

Du kennst das längst: Die Wetter-App sagt dir, wie warm es um 18 Uhr wird. Der Netzbetreiber muss heute wissen, wie viel Strom morgen um 19 Uhr fließt — sonst wird es dunkel. Und in der Werkstatt leuchtet ein Lämpchen auf, *bevor* das Lager kaputtgeht. Drei Mal dieselbe Sorte Daten: **ein Wert, gemessen immer wieder, in gleichen Abständen.** Eine Zeitreihe.

Und dann ist da die Frage, die mir der Betreiber dieser Seite gestellt hat: **„Wenn ich mittags auf den Vormittagsverlauf schaue — kann ich vorhersagen, ob der Tag im Plus endet?“** Die ehrliche Antwort ist nicht ja und nicht nein, sondern: **Das messen wir gleich, statt es zu behaupten.**

HerrKI-Tipp: Ein Satz für dieses Kapitel — **„Ein Modell findet nur, was drin ist.“**

Was du hier lernst

- Wie aus **einer Kurve Trainingsbeispiele** werden — mit dem gleitenden Fenster
- Die zwei Fragen an dieselben Daten: **fortschreiben** (Zahl) oder **klassifizieren** (Etikett)
- Warum du Zeitreihen **chronologisch teilen** musst und niemals mischen darfst
- Dass du zuerst die **Baseline schlagen** musst — sonst hast du nichts gelernt
- Warum die **Konfusionsmatrix** mehr verrät als jede Genauigkeitszahl

Für den Unterricht

Einstieg (5 Min): Kapitel gemeinsam lesen, Alltagsbeispiel aufgreifen.

Erkunden am Gerät (15 Min): die interaktiven Module auf herrki.de/270_zeitreihen ausprobieren.

Sichern mit Quiz (5 Min): „Kurzer Check" besprechen, „Das nimmst du mit" festhalten.

Aus einer Kurve werden Beispiele: das Fenster

Ein Netz kann mit einer Kurve nichts anfangen. Es kennt nur das aus Kapitel 030: eine Tabelle, jede Zeile ein Beispiel. Der Trick, aus der Kurve eine zu machen:

1Leg ein **Fenster** der Länge *n* über die ersten *n* Messwerte. Diese *n* Zahlen sind die **Eingabe** — eine Zeile.

2Der **nächste** Wert dahinter ist das **Ziel**. Fertig ist Beispiel 1.

3**Einen Schritt weiterschieben.** Das gibt Beispiel 2. Und so weiter bis zum Ende der Reihe.

Aus 672 Messwerten und n = 24 werden so 648 Beispiele mit je 24 Spalten. Die Spalten heißen **Lag-Features** (verzögerte Merkmale; *lag* = Versatz), ein Modell, das nur davon lebt, heißt **autoregressiv**. Ab jetzt ist es eine ganz normale Tabelle — und damit frisst jedes Werkzeug ab Kapitel 110 plötzlich Zeitreihen: die multiple Regression aus Kapitel 130 wie die binäre Klassifikation aus Kapitel 220.

Zwei andere Wege nur zum Einordnen: Die **1D-Faltung** benutzt denselben Kernel wie Kapitel 250, lässt ihn aber über Zeitschritte statt Bildzeilen wandern — was im Bild ein *Kantenfilter* ist, ist auf einer Zeitreihe ein **Trendwechsel-Detektor**. Und **rekurrente Netze** (RNN, LSTM) durchlaufen die Reihe mit einem Gedächtnis — ein eigenes Kapitel wert.

Zwei Fragen an dieselben Daten

Du hast deine Tabelle und darfst dir aussuchen, **was in der Zielspalte steht** — und mit dieser einen Entscheidung wechselst du zwischen zwei Werkzeugkästen:

	Fortschreiben	Klassifizieren
Fachbegriff	Zeitreihenprognose (*Forecasting*)	Zeitreihenklassifikation (*Time Series Classification*)
Die Frage	Wie geht es weiter?	Was für ein Verlauf ist das?
Ziel	eine Zahl (143,7 MW)	ein Etikett („Plus“/„Minus“)
Ausgabe	1 Neuron, **linear** (120)	1 Neuron, **Sigmoid** (220)
Loss	MSE (115)	Cross-Entropy (220)
Bewertung	MAE / RMSE — **immer gegen eine Baseline**	Konfusionsmatrix — **nie nur die Genauigkeit**

Bei „Plus oder Minus?“ heißt das auch **Richtungsvorhersage**: nicht mehr *wie viel*, nur noch *wohin*. Klingt nach Vereinfachung — ist aber keine, die die Aufgabe leichter macht.

> **Sonderfall Nowcasting.** Eine Periode läuft gerade, ein Teil ist vorbei — du schätzt aus dem angefangenen Stück, wie sie *endet*. Genau der Vormittags-Fall von oben. Der Begriff kommt aus der **Meteorologie** (Regen in den nächsten 90 Minuten) und der **Konjunkturforschung** (Wachstum des laufenden Quartals vor der amtlichen Zahl).

Baselines: die Latte, die du erst mal reißen wirst

Der wichtigste Absatz dieses Kapitels: **Eine Fehlerzahl allein sagt gar nichts.** „MAE 6,7“ wird erst zur Aussage, wenn daneben steht, was der *dümmste denkbare Tipp* erreicht hätte. Drei brauchst du:

Baseline	Regel	Wofür
naiv	„Morgen ist wie heute.“ Vorhersage = letzter bekannter Wert.	Fortschreiben
saisonal-naiv	„Wie gestern zur selben Uhrzeit.“ Vorhersage = Wert vor 24 Schritten.	Reihen mit Rhythmus
Basisrate	„Immer Plus.“ Ohne hinzusehen.	Klassifizieren

Sie kosten null Rechenzeit und sind gemein gut — auf Finanzkursen ist der naive sogar *optimal*. Wer sein Netz nicht gegen sie antreten lässt, misst nur seinen eigenen Optimismus.

Runter zum Tensor

Was das Fenster wirklich anrichtet — klein genug fürs Kopfrechnen. Acht Messwerte, **n = 3**:

```
# die rohe Reihe: 8 Messwerte, einer pro Zeitschritt
reihe = [12, 15, 14, 18, 21, 20, 24, 27]
n = 3

# gleitendes Fenster: Eingabe = n Werte, Ziel = der Wert direkt dahinter
X = [reihe[i:i+n] for i in range(len(reihe) - n)]
y = [reihe[i+n] for i in range(len(reihe) - n)]

# 8 Werte - 3 Fensterplaetze = 5 Zeilen
np.array(X).shape -> (5, 3)
np.array(y).shape -> (5,) bzw. (5, 1) als Spalte
```

Ausgeschrieben: links das Fenster (drei Lag-Spalten), rechts in Rot das Ziel.

X — die Fenster
↓
[5, 3]

→

y — das Ziel
↓
[5, 1]

Und jetzt fällt alles zusammen: **Das ist exakt die Tabelle aus Kapitel 130.** Drei Merkmale, ein Zielwert — das Netz rechnet nichts Zeitreihen-Spezifisches, sondern eine multiple Regression auf drei Spalten, die zufällig aus derselben Messreihe stammen:

```
# Fortschreiben — die Formel aus Kapitel 130, nur heissen die Merkmale jetzt Lags
ŷ = w₁·x(t−3) + w₂·x(t−2) + w₃·x(t−1) + b -> Ausgabe linear, Loss MSE

# Klassifizieren — dieselben Spalten, nur ein anderer Hut obendrauf (Kapitel 220)
p = sigmoid(w₁·x(t−3) + w₂·x(t−2) + w₃·x(t−1) + b) -> p("Plus"), Loss Cross-Entropy
```

Das Etikett entsteht aus derselben Reihe: „Ist der nächste Wert höher als der letzte im Fenster?“ Also 18>14 → **1**, 21>18 → **1**, 20>21 → **0**, 24>20 → **1**, 27>24 → **1**. Vier von fünf sind Plus — die **Basisrate liegt bei 80 %**. Ein Modell mit 78 % Genauigkeit wäre hier schlechter als raten mit geschlossenen Augen. Merk dir dieses Gefühl.

Skalieren, aber je Fenster — und weiter als ein Schritt

Die Regel aus Kapitel 135 gilt weiter, die aus Kapitel 235 auch: **Minimum und Maximum nur aus dem Trainingsteil**. Dazu ein Kniff, den das Labor benutzt: Wir ziehen von jedem Fenster **seinen eigenen letzten Wert** ab und teilen erst dann durch die Trainings-Spannweite. Das Netz sieht nicht „141,2 · 138,7 · 143,9“, sondern „−2,7 · −5,2 · 0“ — die *Abstände zum Jetzt*. Das Niveau muss so nicht mitgelernt werden, und der naive Tipp wird zur Vorhersage **„Änderung = 0“**: Das Netz muss beweisen, dass es mehr weiß als „bleibt, wie es ist“.

Willst du weiter als einen Schritt, fütterst du die Vorhersage als nächste Eingabe zurück (**rekursives Fortschreiben**) — mit dem Haken aus Kapitel 380: Fehler verketten sich.

⚠ Trifft ein Schritt mit 95 % Zuverlässigkeit, dann sind zwanzig Schritte hintereinander nicht 95 % zuverlässig, sondern $0{,}95^{20} \approx$ **36 %**. Deshalb wird jede Prognose unschärfer, je weiter sie reicht — und deshalb zeichnen Wetterdienste ab Tag 5 einen Trichter statt einer Linie.

Labor I: Wie geht es weiter?

Vier Datensätze stehen bereit — **alle vier synthetisch**, im Browser mit festem Startwert (Seed) erzeugt. Kein echter Kurs. Das ist Absicht: Nur so wissen wir *genau*, welches Muster drinsteckt — und du siehst dieselben Zahlen wie ich.

Datensatz	Was drinsteckt	Erwartung
A — Stromverbrauch 24 Werte/Tag × 28 Tage	Tagesrhythmus mit Morgen- und Abendspitze, Wochenende −15 %, +5 % Trend, 5 % Rauschen	Fortschreiben funktioniert
B — Börse (Zufallspfad) 672 Schritte	$\text{Kurs}_{t+1} = \text{Kurs}_t \cdot (1 + \varepsilon)$, ε normalverteilt, σ = 0,8 %, Drift +0,02 %. **Sonst nichts.**	nichts schlägt den naiven Tipp
C — Börse mit Momentum 672 Schritte	wie B, aber ε enthält **70 % der Durchschnittsrendite der letzten 12 Schritte**	jetzt gibt es etwas zu finden
D — Maschinensensor 672 Schritte	Grundschwingung mit Rauschen; vor jedem Ausfall baut sich über 24 Schritte ein Vorwarnzeichen auf (+40 % Amplitude, zusätzlicher Frequenzanteil)	Klassifikation ist hier spannend

Am Gerät ausprobieren: herrki.de/270_zeitreihen#abschnitt-5

Direkt zum Modul

🔮 Zeitreihen-Labor I: Fortschreiben

Ein Netz (Fenster → 16 → 1 linear, Adam 0,01, 60 Epochen) tritt gegen die naiven Tipps an. Geteilt wird **chronologisch 80 / 20**: Das Netz sieht nur links der roten Linie, bewertet wird rechts davon.

Datensatz A — Stromverbrauch

Fenster n 24

Schritte voraus 1

▶ Trainieren ↻ Zurücksetzen

Datensatz wählen und „Trainieren“ drücken.

— Ist · — naiv · — saisonal-naiv · — Netz — *gezeigt wird das Ende des Trainingsteils plus der komplette Testteil.*

▶ Ausgabe erscheint hier …

Was du hier siehst. Bei **A** hat das Netz eine Chance: Der Tagesrhythmus steckt im Fenster, es muss ihn nur auslesen. Bei **B** gewinnt der naive Tipp — keine Panne, sondern Mathematik: In einem Zufallspfad *ist* der letzte Wert die beste Schätzung für den nächsten. Der saisonal-naive Tipp liegt bei B und C weit daneben, weil ein Kurs keinen 24-Schritte-Rhythmus hat. Und schieb „Schritte voraus“ hoch: Die Kurven werden flacher — die Fehlerverkettung, sichtbar gemacht.

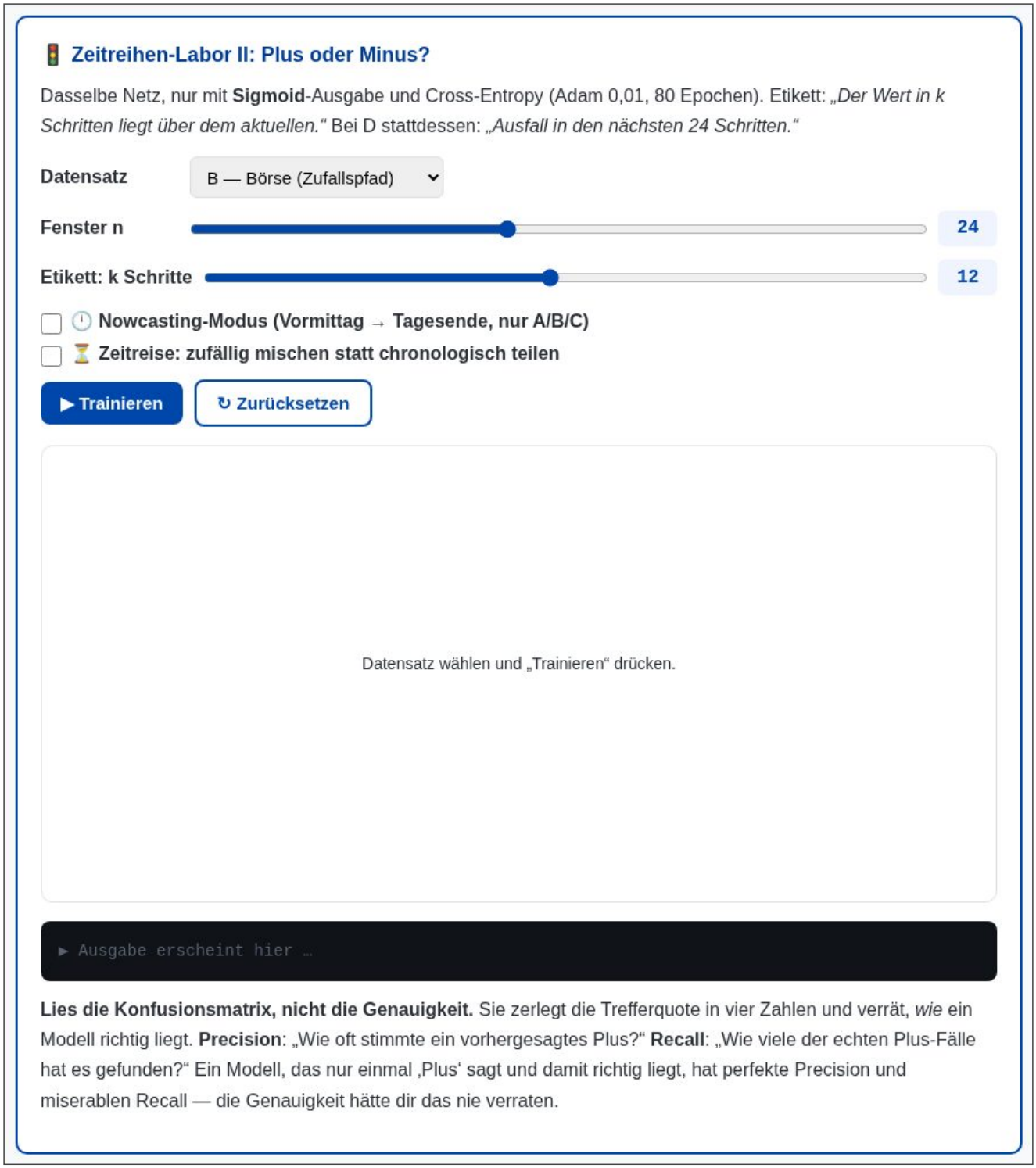
Interaktives Modul – am Gerät ausprobieren

Labor II: Plus oder Minus?

Dieselben Reihen, andere Zielspalte — und plötzlich reicht die Genauigkeit als Maßstab nicht mehr. Labor II erzeugt dieselben Kurven mit demselben Startwert, aber über **2688 Schritte** statt 672 (die ersten 672 sind exakt die Reihe aus Labor I): Eine Genauigkeit aus 130 Testfällen schwankt allein durch Zufall um mehrere Prozentpunkte. Wer „57 %“ meldet, ohne zu sagen, aus wie vielen Fällen, meldet nichts.

Am Gerät ausprobieren: herrki.de/270_zeitreihen#abschnitt-6

Direkt zum Modul



Interaktives Modul – am Gerät ausprobieren

Das Häkchen „Zeitreise“ ist eine Falle zum Anfassen. Es tut das, was in Kapitel 235 für gewöhnliche Tabellen richtig war: mischen, dann teilen. Bei einer Zeitreihe ist das **Data Leakage**, und zwar doppelt. Erstens rutscht Zukunft ins Training. Zweitens überlappen Nachbarfenster fast vollständig: Bei n = 24 teilen sich zwei aufeinanderfolgende Fenster 23

von 24 Werten und meist auch das Etikett — der Testfall steht praktisch schon im Trainingssatz. Die Genauigkeit steigt und lügt.

Was die Fachwelt dazu nimmt

Der Klassiker der **Prognose**: die monatlichen Fluggastzahlen internationaler Airlines von 1949 bis 1960, 144 Messpunkte, bekannt durch George Box und Gwilym Jenkins — Trend, Jahresrhythmus, wachsende Schwankungsbreite, dieselbe Struktur wie unser Datensatz A. 2020 kam die kalte Dusche: Im **M4-Wettbewerb** traten 61 Verfahren auf 100.000 echten Zeitreihen an, und **reine Machine-Learning-Verfahren landeten hinter einfachen statistischen Methoden**; vorn lagen Hybride. Keine Absage an Netze, sondern eine Erinnerung: **Baselines sind kein Spielzeug, sie sind Gegner.**

Bei der **Klassifikation** sieht es anders aus: **EKG** („normal oder auffällig?“, einer der Standarddatensätze im **UCR Time Series Archive**), **vorausschauende Wartung** (Vibration, Temperatur, Stromaufnahme — unser Datensatz D) und Bewegungssensoren, die aus dem Beschleunigungssignal deines Handys lesen, ob du gehst oder gestürzt bist. Und **Nowcasting** zeigte 2021 in *Nature*, was geht, wenn das Muster physikalisch da ist: Niederschlag der nächsten **90 Minuten** aus Radarbildern, von Fachleuten den bisherigen Verfahren vorgezogen. Regenfronten sind träge — sie haben Masse und Richtung.

Und die Börse? Der Vormittags-Gedanke von oben ist kein Sonderweg: **Gao, Han, Li und Zhou** zeigten 2018 ein **Intraday Momentum** — die Rendite der ersten halben Handelsstunde sagt die der letzten halben Stunde statistisch *schwach* vorher. Messbar, aber winzig, und genau darin liegt die Pointe: **Was messbar und groß genug ist, wird weggehandelt.** Auch Kursfenster als *Bilder* zu behandeln und ein CNN daraufzusetzen, ist erforscht (**Wang und Oates**, 2015) — nur macht auch ein CNN aus einem Zufallspfad kein Muster.

Zurück zur Anwendung: Und die Börse?

Was du selbst gemessen hast: Auf **B** landet die Genauigkeit auf der Basisrate — das Netz hat nichts gelernt, was „immer Plus“ nicht auch geliefert hätte. Auf **C** liegt sie deutlich darüber, aber nur, weil wir das Momentum mit der Hand eingebaut haben: Das Netz hat nichts entdeckt, es hat gefunden, was wir versteckt hatten. **Echte Märkte liegen zwischen B und C — deutlich näher an B.** Und selbst der messbare Rest überlebt den Weg zum Konto meistens nicht:

- **Transaktionskosten und Spread.** 53 % Trefferquote klingt nach Vorsprung. Bei Gebühren und Geld-Brief-Spanne ist sie es nicht — jeder Trade kostet, egal ob er richtig war.
- **Nicht-Stationarität.** Die Regel aus dem Trainingszeitraum gilt im Testzeitraum vielleicht nicht mehr. Bei Datensatz A ändert sich der Tagesrhythmus nie; bei einem Markt ändert sich alles — gerade dann, wenn viele dieselbe Regel entdeckt haben.
- **Überlebens-Verzerrung** (Survivorship Bias). Historische Kurslisten enthalten oft nur Unternehmen, die es bis heute gibt — die Pleiten fehlen, jede Strategie sieht im Rückblick besser aus.

Mein Fazit: Die Börse ist ein **hervorragendes Lernbeispiel** — gerade weil sie ehrlich scheitert und dabei alle Fallen auf einmal vorführt: die Baseline, die man nicht schlägt; die Genauigkeit, die nur die Basisrate ist; das Leakage, das plötzlich tolle Zahlen macht; und die Rückschau-Optimierung, die auf jedem Testzeitraum irgendeine glänzende Kombination findet (Kapitel 240). Als **Geldanlage** ist sie mit diesen Mitteln **nichts**. Dieselben Werkzeuge glänzen dagegen bei A und D — Energie, Wartung, Medizin, Wetter. Dort gibt es Physik, Gewohnheit und Verschleiß. Also Muster.

Was heißt das für dich?

- **Frag zuerst nach der Baseline, nicht nach der Architektur.** „Was macht der naive Tipp?“ Kommt darauf keine Zahl, ist das Modell nicht bewertet, sondern nur gebaut.
- **Bei Trefferquoten immer die Basisrate danebenlegen.** „87 % Genauigkeit“ ist wertlos, wenn 87 % aller Fälle ohnehin zur Mehrheitsklasse gehören — das gilt für Kursprognosen wie für Screenings und Betrugserkennung.
- **Bei allem mit Zeitstempel: nicht mischen.** Chronologisch teilen, Skalierung nur aus dem Trainingsteil. Die eine Regel, die dich vor selbstbetrügerisch guten Zahlen schützt.
- **Such dir Aufgaben mit Trägheit.** Stromverbrauch, Temperatur, Verschleiß, Füllstände — wo Physik oder Gewohnheit den nächsten Wert an den letzten fesseln, sind diese Werkzeuge stark. Wo alle aktiv gegen die Vorhersagbarkeit arbeiten, sind sie es nicht.

Zum Weiterlesen

- **Standardwerk, kostenlos:** Hyndman & Athanasopoulos, *Forecasting: Principles and Practice* (3. Aufl.) — otexts.com/fpp3
- **Die kalte Dusche:** Makridakis u. a., *„The M4 Competition“*, Int. Journal of Forecasting 36(1), 2020 — doi.org/10.1016/j.ijforecast.2019.04.014
- **Datensammlung Klassifikation:** UCR Time Series Archive — cs.ucr.edu/~eamonn/time_series_data_2018
- **Nowcasting in Bestform:** Ravuri u. a., *„Skilful precipitation nowcasting…“*, Nature 597, 2021 — nature.com/articles/s41586-021-03854-z
- **Der Vormittags-Effekt:** Gao, Han, Li & Zhou, *„Market intraday momentum“*, J. of Financial Economics 129(2), 2018 — doi.org/10.1016/j.jfineco.2018.05.009
- **Zeitreihen als Bilder:** Wang & Oates, *„Imaging Time-Series…“*, 2015 — arxiv.org/abs/1506.00327
- **Auf HerrKI:** 235, 250 und 220

Kurzer Check – mit Lösung

Lösungen für Lehrkräfte – für Lernende die Online-Version nutzen.

Frage 1: Wie wird aus einer Zeitreihe ein Trainingsbeispiel?

- Jeder Zeitpunkt wird einzeln als Beispiel genommen, ohne seine Vergangenheit.
- **✓ Ein Fenster der letzten n Werte ist die Eingabe; der nächste Wert (oder ein Etikett) ist das Ziel — dann rutscht das Fenster einen Schritt weiter.**
- Die komplette Reihe wird als ein einziges großes Beispiel übergeben.

Genau — das gleitende Fenster. Aus 672 Messwerten und n = 24 werden 648 Zeilen mit je 24 Lag-Spalten. Ohne Vergangenheit gäbe es nichts zu lernen; die ganze Reihe auf einmal wäre genau ein Beispiel — zu wenig zum Trainieren.

Frage 2: Warum darf man Zeitreihen-Daten vor dem Aufteilen nicht mischen?

- Weil das Training dann deutlich länger dauert.
- Weil sich beim Mischen die Reihenfolge der Spalten ändert.
- **✓ Weil sonst Zukunft ins Training rutscht — und weil überlappende Nachbarfenster den Testfall praktisch schon im Training stehen haben. Die Testgenauigkeit wird geschönt.**

Richtig: Data Leakage (Kapitel 235) in Zeitreihen-Form. Zwei benachbarte Fenster teilen sich bei n = 24 ganze 23 Werte und meist auch das Etikett — landet eines im Training und eines im Test, misst du dein Gedächtnis, nicht dein Modell.

Frage 3: Ein Klassifikator trifft an 53 % der Tage richtig — der Markt stieg an 53 % der Tage. Was hat er gelernt?

- Er hat den Markt verstanden: 53 % sind schließlich mehr als der Zufall mit 50 %.
- **✓ Nichts, was über die Basisrate hinausgeht. Erst die Konfusionsmatrix zeigt, ob er mehr kann als „immer Plus“.**
- Die Genauigkeit ist gut, weil sie über 50 % liegt.

Genau. „Immer Plus“ hätte dieselben 53 % geliefert — ohne Modell, ohne Training, ohne Strom. Erst die Aufteilung in richtig/falsch Plus und richtig/falsch Minus zeigt, ob das Modell unterscheidet oder nur eine Taste drückt.

Das nimmst du mit

- **Das Fenster macht aus einer Kurve eine Tabelle.** Ab da ist es normales Machine Learning — dieselbe Regression, dieselbe Klassifikation wie in 130 und 220.
- **Zwei Fragen, dieselben Daten.** Zahl = lineare Ausgabe + MSE; Etikett = Sigmoid + Cross-Entropy. Die Zielspalte entscheidet.
- **Chronologisch teilen, niemals mischen.** Skalierung nur aus dem Trainingsteil — sonst misst du Data Leakage statt Können.
- **Erst die Baseline schlagen.** Naiv, saisonal-naiv, Basisrate. Wer diese drei nicht schlägt, hat kein Modell, sondern eine Rechnung.
- **Ein Modell findet nur, was drin ist.** In C steckte Momentum — also fand es Momentum. In B steckte nichts — also fand es nichts.

Weiter geht's: Bisher hatte jedes Netz die richtige Antwort *daneben* stehen — Regression, Klassifikation, Faltung, Autoencoder, Zeitreihen. Im nächsten Kapitel „Lernen durch Belohnung: Wie Robi laufen lernt“ gibt es keine Lösung mehr, nur Punkte — Reinforcement Learning. Danach beginnt der Pfad *Aktuell* mit „Latent Space — die geheime Landkarte“.

So zitierst du dieses Kapitel

HerrKI: Zeitreihen: fortschreiben oder klassifizieren. herrki.de/270_zeitreihen, ein privates Hobbyprojekt von Terragon.de, abgerufen am 09.09.2026.