Block 3: Komplexität, Kausalität & Generalisierung

Data Science and Strategy for Business – Eine Lernreise mit Lena

March 31, 2026

Lernziele Block 3:

- Multikollinearität erkennen, BLUE einordnen
- Overfitting vs. Out-of-Sample Performance
- Kreuzvalidierung und Regularisierung anwenden
- Korrelation von Kausalität unterscheiden
- A/B-Tests und logistische Regression verstehen

Dieser Block bereitet auf fortgeschrittene ML-Methoden vor.

BLUE: Kann ich meinen Zahlen vertrauen?

Lena bei DataCo: 50.000 Kunden, Auftrag: *„Wer kündigt?"* `lm(churn ~ .)` liefert Koeffizienten – aber stimmen sie?

Was macht eine gute Waage aus?

BLUE-Komponenten:

E Estimator: $(X, y) \to \hat{\beta}$

U Unbiased: $E[\hat{\beta}] = \beta$

L Linear: $\hat{\beta} = \sum_i w_i \cdot y_i$

B Best: Kleinste Varianz aller lin. erw.-treuen Schätzer

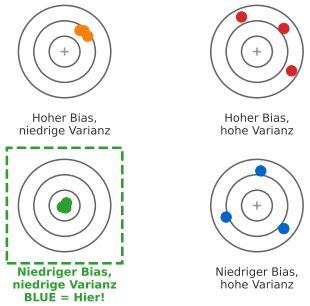
Waage-Analogie:

- **Genau** (Unbiased): $E[\hat{\beta}] = \beta$
- **Stabil** (Low Var.): Messungen eng beieinander
- **Beste**: Unter allen genauen die stabilste

BLUE = Best Linear Unbiased Estimator: genau UND so präzise wie möglich.

Gauss-Markov-Annahmen und Zielscheibe

BLUE: Bias und Varianz als Zielscheibe



Genau + Präzise = Ideal. Genau + Unpräzise = instabil. Ungenau + Präzise = Bias.

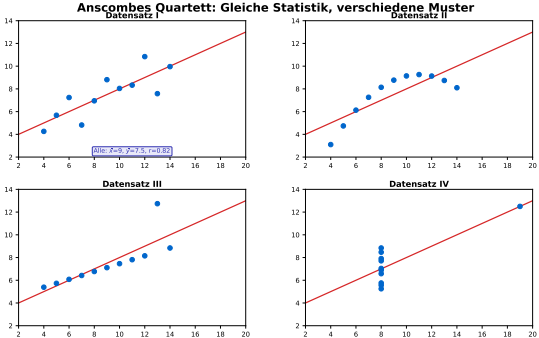
5 Gauss-Markov-Annahmen (alle nötig für BLUE):

Nr.	Annahme
1	Linearität in β
2	Zufällige Stichprobe
3	Keine perfekte Multikollinearität
4	$E[\varepsilon_i \mid X] = 0$
5	$\text{Var}(\varepsilon_i \mid X) = \sigma^2$

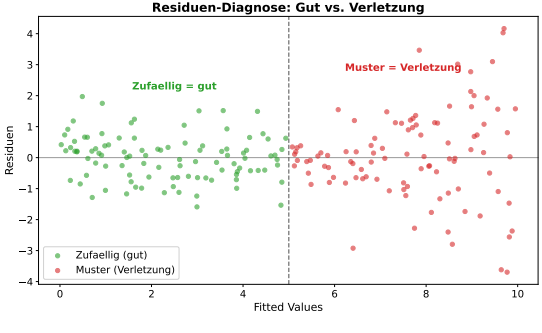
Wenn erfüllt $\rightarrow$ `lm()` ist optimal. MSE = Bias^2 + Varianz.
Merke: Normalverteilung NICHT Teil von GM – erst für Inferenz.

Annahmen 1–3: Datenstruktur (prüfbar). Annahmen 4–5: Fehlerstruktur (Residuenanalyse).

Diagnostik: Anscombe-Quartett und Residuenanalyse



Anscombe (1973): 4 Datensätze, gleiche Statistik, verschiedene Muster. Kennzahlen reichen nicht!



Residuenplots: U-Form → Linearität verletzt. Trichter → Heterosked. Kein Muster → OK.

Annahme	Diagnostik	R-Funktion	Verletzung
Linearität	Resid. vs. Fitted	`plot(model, 1)`	U-Form
Homoskedastizität	Scale-Location	`bptest()`	Trichter
Normalität	Q-Q Plot	`shapiro.test()`	Abweichung
Multikollinearität	VIF	`car::vif()`	VIF > 5

Faustregel: `plot(model)` **in R erzeugt 4 Diagnose-Grafiken automatisch.**

Zahlenbeispiel, Geschichte und häufige Fehler

Heteroskedastizität – Lenas Zahlen:

Szenario	$\hat{\beta}_1$	*SE*	*t*	*p*
Korrekt	2,3	0,4	5,75	$< 0{,}001$
Heterosked.	2,3	0,8	2,88	0,004

- $\hat{\beta}_1$ bleibt gleich (erwartungstreu)
- SE verdoppelt $\rightarrow$ t halbiert
- Kleiner Effekt: Signifikanz weg!

Gauss und Ceres (1801): 22 Messungen, Gauss (24 J.) entwickelt Methode der kleinsten Quadrate. Vorhersage so präzise, dass Ceres gefunden wird. $\sim$100 Jahre später: Markov formalisiert $\rightarrow$ Gauss-Markov-Theorem.

Drei Fehler über BLUE:

1. „Immer optimal" – Nur unter 5 Annahmen
2. „Normalverteilung nötig" – Erst für Inferenz
3. „Bei Verletzung nutzlos" – Robuste SE helfen

Verletzte Annahmen ändern nicht die Schätzung, aber die Inferenz.

R-Code und Übung: BLUE und Gauss-Markov

```
model <- lm(churn_score ~ alter + vertragslaufzeit
            + monatl_kosten + nutzung_gb, data = df)
par(mfrow = c(2, 2)); plot(model)
library(lmtest); bptest(model)  # p < 0.05 -> Heterosked.
library(car); vif(model)        # VIF > 5: problematisch
```

Übungsaufgaben:

1. **Waage:** 10 Messungen (100 g): 98, 102, 99, 101, 100, 103, 97, 101, 99, 100. Mittelwert? Erwartungstreu?
2. **Annahmen:** Welche GM-Annahme verletzt? (a) Exp. Zusammenhang, linear modelliert. (b) Monatl. + Jahreskosten im Modell. (c) Residuenstreuung wächst mit Umsatz.
3. **Heterosked.:** $\hat{\beta}_1 = 3{,}5$, $SE = 0{,}7$. Heterosked. $\rightarrow$ $SE = 2{,}1$. Beide t-Werte? Noch signifikant?

Drei Checks (Plot, Breusch-Pagan, VIF) decken die wichtigsten Probleme ab.

Multikollinearität: Die Zwillings-Analogie

Lena: Fügt Vertragslaufzeit + Kosten hinzu. Vorzeichen wechseln, SE explodieren – aber R^2 steigt!

Variable kopieren und beide ins Modell?

Unabhängige Zeugen:

- Jeder trägt *eigene* Information bei – Richter nutzt beide

Eineiige Zwillinge (= korrelierte Prädiktoren):

- Aussagen praktisch identisch – OLS kann Beiträge nicht trennen

Übertragung: Zwillinge = korrelierte Prädiktoren, Richter = OLS

Vorhersage: Modell funktioniert trotzdem!
Interpretation: Einzelne Koeffizienten unbrauchbar.
→ Ob Problem, hängt vom Ziel ab.

Multikollinearität schadet der Interpretation, weniger der Vorhersage.

Definition und vier Symptome

Exakte Multikollinearität

$x_3 = 2x_1 + x_2$ – OLS kann β **nicht berechnen**. R: `NA`. Selten.

Approximative Multikollinearität

Stark korreliert ($r = 0{,}92$). OLS läuft, aber Schätzungen **unzuverlässig**. Tückisch.

Vier Symptome:

- **Hohes R^2, keine Sign.** – geteilte Erklärungskraft
- **Aufgeblähte SE** – Typ-II-Fehler
- **Instabile Koeff.** – 1 Punkt weg → Änderung
- **Vorzeichenwechsel** – OLS überkompensiert

Alle 4 zusammen = Multikollinearität → VIF berechnen!

Der approximative Fall ist gefährlich – Modell läuft, aber falsche Schlüsse.

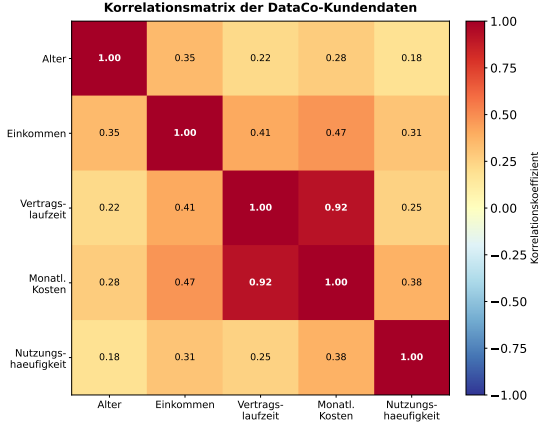
VIF: Formel, Schwellenwerte und Heatmap

VIF-Formel: X_j auf alle anderen regressieren:

$$VIF_j = \frac{1}{1 - R_j^2}$$

- $R_j^2 = 0$: $VIF = 1$ (keine Inflation)
- $R_j^2 = 0{,}9$: $VIF = 10$ (10-fach)
- $R_j^2 \to 1$: $VIF \to \infty$

VIF	$\sqrt{VIF}$	Bewertung
1	1,0×	ideal
5	2,2×	problematisch
10	3,2×	Handlung nötig



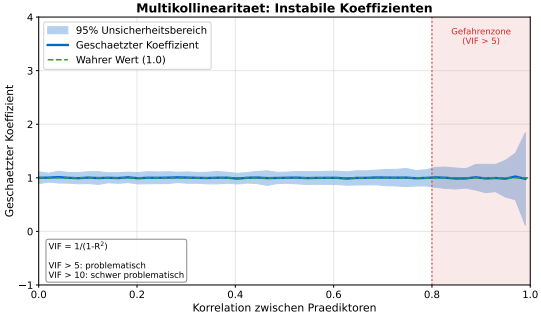
Heatmap: Dunkelrot = hohe Korrelation. Heatmap zeigt *welche* Paare, VIF quantifiziert die Auswirkung.

$\sqrt{VIF}$ zeigt den SE-Aufblähungsfaktor. VIF > 5 verdient Aufmerksamkeit.

VIF-Beispiele, Lösungen und Geschichte

Lenas VIF-Tabelle:

Feature	VIF	Bewertung
Vertragslaufzeit	5,6	problematisch
Monatl. Kosten	4,5	grenzwertig
Alter	1,14	OK
Nutzung (GB)	1,33	OK



Fünf Lösungen:

1. **Variable entfernen** – inhaltlich begründet
2. **Kombinieren** – Index bilden
3. **Regularisierung** (Ridge/Lasso)
4. **PCA** – unkorrelierte Komponenten
5. **Mehr Daten** – kleinere SE

Geschichte: Hoerl & Kennard (1970) – DuPont, korrelierte Messdaten → Ridge.

Fehler 1: „Hohe r = Problem" – hängt von n/Ziel ab.
Fehler 2: „Höchsten VIF entfernen" – kann kausal relevanteste Variable sein.

Vorhersage: Multikollinearität oft tolerierbar. Interpretation: muss behandelt werden.

R-Code und Übung: Multikollinearität und VIF

```
library(car)
vif_werte <- vif(model)
print(vif_werte)  # alter:1.14 vertragslfz:5.60 kosten:4.50
library(corrplot)
cor_matrix <- cor(df[, c("alter","vertragslaufzeit",
                         "monatl_kosten","nutzung_gb")])
corrplot(cor_matrix, method="color", type="upper",
         addCoef.col="black")
model_red <- lm(churn_score ~ alter + vertragslaufzeit
                + nutzung_gb, data = df)
vif(model_red)   # Alle VIF jetzt unter 2
```

Übungsaufgaben:

1. **VIF:** $R_1^2 = 0{,}75$. (a) VIF_1? (b) SE-Faktor? (c) Problematisch?
2. **Symptome:** Modell A: $R^2 = 0{,}35$, beide sign. Modell B (+2 Var.): $R^2 = 0{,}38$, nichts sign. Welches Symptom?
3. **Vorhersage vs. Interpretation:** Warum schadet Multikollinearität der Interpretation, aber weniger der Vorhersage?

Zwei Befehle reichen: `vif()` **für Diagnose,** `corrplot()` **für Visualisierung.**

Bias-Varianz: Das starre Navi

Lena: Training **99%**, Test **52%** – kaum besser als Münze!

47 Pp. Lücke = Overfitting!

Die Navi-Analogie:

Das starre Navi:

- Speichert Routen, nicht Logik
- Bekannte Strecke: perfekt
- Umleitung: verloren

Lenas Modell:

- „Merkt sich" Trainingsbeobachtungen
- Lernt auch Rauschen mit

Routen speichern = reproduzieren
Netz verstehen = Muster erkennen

Kern: **Bias-Varianz-Tradeoff** – zu viel Varianz (starre Routen) vs. zu viel Bias („Fahr einfach nach Norden")

Ein gutes Modell „versteht" das Straßennetz – es speichert nicht nur einzelne Routen.

Overfitting vs. Underfitting: Symptome und Gegenmassnahmen

Overfitting: Modell zu komplex.
Symptome: Train >95%, Test 20–40 Pp. darunter, β sehr groß, viele Parameter.
Underfitting: Modell zu einfach.
Symptome: Beide Fehler hoch, kein Gap.

Fünf Gegenmassnahmen:

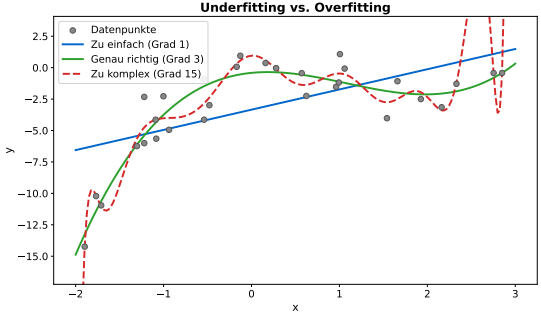
1. **Mehr Daten**
2. **Feature Selection**
3. **Regularisierung**
4. **Kreuzvalidierung**
5. **Einfacheres Modell**

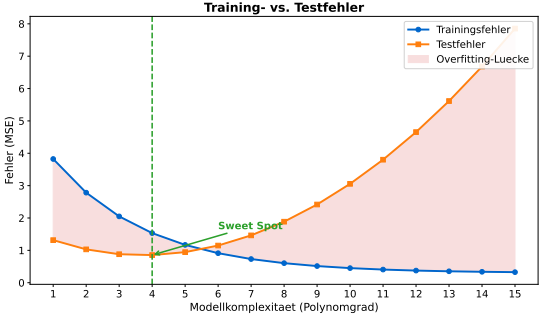
Faustregel: Mind. **10–20 Beob. pro Prädiktor**!

Features	n/p	**Status**
$p = 4$	125	sicher
$p = 50$	10	kritisch
$p = 200$	2,5	gefährlich

Kleine Lücke + beide schlecht = Underfitting. Große Lücke = Overfitting.

In-Sample vs. Out-of-Sample: Overfitting-Demo





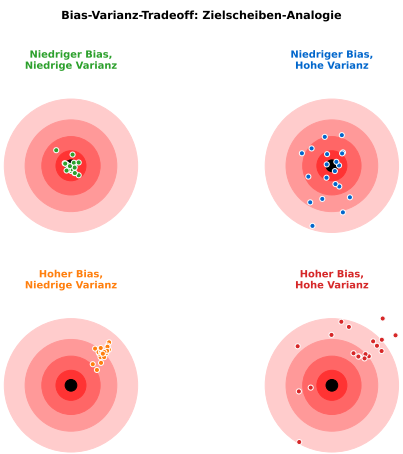
Polynomgrade:

	Grad 1	Grad 3	Grad 15
Train-MSE	5,1	3,8	2,5
Test-MSE	5,3	4,1	28,7
Diagnose	Under	Sweet	Over

- Train: sinkt monoton
- Test: U-Form
- Lücke wächst bei Overfitting
- Flexibilität $\neq$ Qualität!

Was zählt, ist ausschliesslich die Out-of-Sample-Performance.

Bias-Varianz-Zerlegung und Zielscheiben



$$E[(y - \hat{y})^2] = \underbrace{\text{Bias}^2}_{\text{Syst.}} + \underbrace{\text{Varianz}}_{\text{Instab.}} + \underbrace{\sigma^2}_{\text{Irreduz.}}$$

Lenas Zahlen: $\text{Bias}^2 = 0{,}16$, $\text{Var} = 0{,}30$, $\sigma^2 = 0{,}04$. **Gesamt: 0,50.**
Varianz dominiert (60%)!

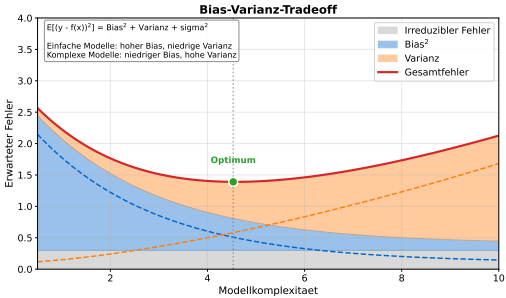
Modell	Bias	Varianz
Grad-15	niedrig	enorm
Linear f. quadr.	hoch	niedrig
Grad-3 (Sweet)	niedrig	niedrig

Einfacher → weniger Varianz, mehr Bias.

Ideal: eng + zentral. Overfit: instabil. Underfit: daneben.

Das optimale Modell minimiert Bias^2 + Varianz – nicht einen allein.

Bias-Varianz-Tradeoff, Geschichte und Irrtümer



Links = Underfitting. Rechts = Overfitting. **Sweet Spot** = min. Gesamtfehler.
Netflix (2009): 100+ Modelle, nie in Produktion – zu komplex!
Challenger (1986): Nur Flüge mit Schäden analysiert → Selection Bias.

Drei Irrtümer:
1. „Mehr Features = besser" – Fluch der Dimensionalität
2. „Train-Accuracy = Qualität" – 75%/73% besser als 99%/52%
3. „Overfitting nur bei kleinen Daten" – n/p entscheidend, nicht n

Das theoretisch beste Modell ist nicht immer das praktisch nützlichste.

R-Code, Übung und Zwischenfazit: Bias-Varianz

```
set.seed(123)
x_train <- runif(50,0,10)
y_train <- 2 + 0.5*x_train + 0.2*x_train^2 + rnorm(50,0,2)
x_test <- runif(100,0,10)
y_test <- 2 + 0.5*x_test + 0.2*x_test^2 + rnorm(100,0,2)
train_mse <- test_mse <- numeric(15)
for (d in 1:15) {
  fit <- lm(y_train ~ poly(x_train, d, raw=TRUE))
  train_mse[d] <- mean((y_train - predict(fit))^2)
  test_mse[d] <- mean((y_test - predict(fit,
    newdata=data.frame(x_train=x_test)))^2)
}
cat("Optimaler Grad:", which.min(test_mse))  # 2-3
```

Übungen:

1. $R^2_{\text{Train}} = 0{,}98$, $R^2_{\text{Test}} = 0{,}41$. Problem? Gegenmassnahmen?
2. Bias $= 0{,}6$, Var $= 0{,}1$, $\sigma = 0{,}3$. Gesamtfehler? Einfacher oder komplexer?

Zwischenfazit: Fehler = Bias^2 + Varianz + σ^2. 10–20 Beob./Prädiktor. **Nächster Schritt:** Regularisierung.

Lösung 2: $0{,}6^2 + 0{,}1 + 0{,}3^2 = 0{,}55$ **– Bias dominiert, komplexer machen!**

Regularisierung: Die Leine für wilde Koeffizienten

Lena: β: 42, -87, 63. Riesig, instabil, Vorzeichen wechseln.

Was wenn große β Kosten verursachen?

Leine-Analogie:

- **Ohne** ($\lambda = 0$): β rennen frei, passen sich an Rauschen an
- **Mit** ($\lambda > 0$): Zurückziehen bei Ausscheren
- Kürzer ($\lambda \uparrow$) $\rightarrow$ weniger Freiheit

Türsteher – Ridge vs. Lasso:

- **Ridge**: Alle rein, aber geschrumpft
- **Lasso**: Unwichtige abgewiesen ($= 0$)

$\lambda = 0$: OLS λ mittel: Sweet Spot
λ klein: Lange Leine $\lambda \rightarrow \infty$: Alle $\beta \rightarrow 0$

Regularisierung opfert etwas Bias, gewinnt Stabilität (weniger Varianz).

Strafidee: Ridge, Lasso und Elastic Net

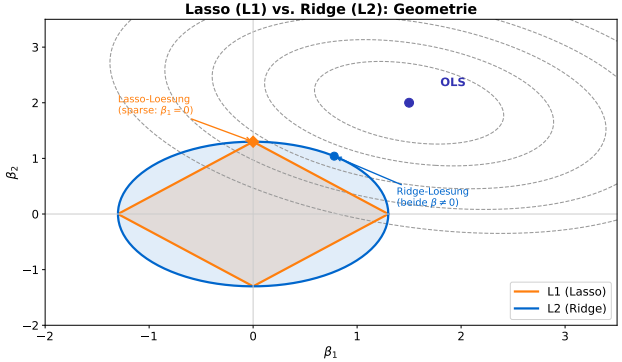
Kernidee: Strafterm bestraft große Koeffizienten.

Methode	Formel	Eigenschaft
Ridge (L2)	$\min_\beta \sum(y_i - \hat{y}_i)^2 + \lambda \sum \beta_j^2$	Schrumpft alle
Lasso (L1)	$\min_\beta \sum(y_i - \hat{y}_i)^2 + \lambda \sum \lvert\beta_j\rvert$	Schrumpft + selektiert
Elastic Net	$\min_\beta \text{RSS} + \lambda[\alpha \sum \lvert\beta_j\rvert + (1-\alpha) \sum \beta_j^2]$	Kombination L1+L2

RSS: Wie gut passt das Modell? Allein $\rightarrow$ OLS. **Strafe:** Große β „teuer". $\lambda = 0$: OLS. $\lambda \rightarrow \infty$: alle $\beta \rightarrow 0$.
Elastic Net: L1+L2. Korrelierte Feature-Gruppen $\rightarrow$ gemeinsam ausgewählt. `glmnet`: $\alpha \in (0, 1)$.

Ohne Budget kauft man alles (auch Rauschen). Mit Budget muss man priorisieren.

Geometrie: Warum Lasso Nullen erzeugt



Ridge (Kreis):

- Keine Ecken $\rightarrow$ Berührpunkt nie auf Achse
- Kein β_j exakt Null
- Behält *alle* Features

Lasso (Diamant):

- Ecken auf Achsen!
- Ellipsen treffen fast immer Ecke
- $\rightarrow$ β_j exakt Null
- Automatische Selektion

L2 = Kreis = keine Selektion. L1 = Diamant = automatische Variablenselektion.

Vergleich und Lambda-Wahl

Eigenschaft	Ridge	Lasso	Elastic Net
Koeff. auf 0?	Nein	Ja	Ja
Selektion?	Nein	Ja	Ja
Multikollinearität	Sehr gut	Problem.	Gut
`glmnet` α	0	1	$(0, 1)$

Lambda: Kreuzvalidierung – nie per Hand!

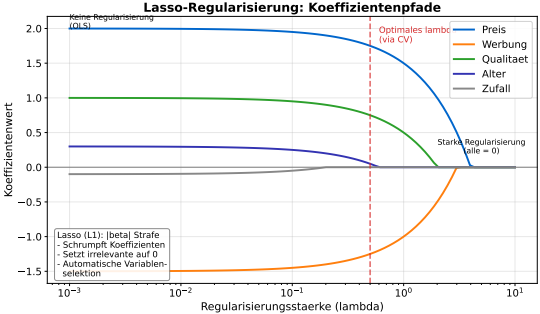
`cv.glmnet`:

1. Daten in k Folds
2. Für viele λ: Train auf $k-1$, Test auf letztem
3. Mittlerer CV-Fehler pro λ
4. Optimales λ wählen

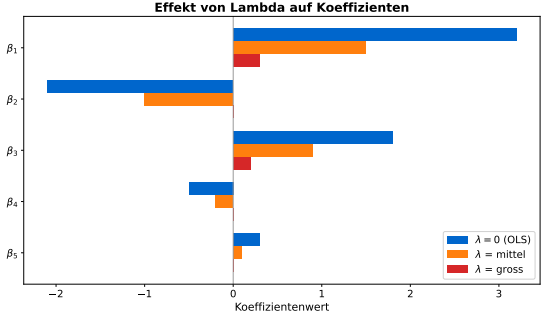
Zwei λ-Werte:
lambda.min: Bestes MSE.
lambda.1se: Innerhalb 1 SE – einfacher, kaum schlechter.
Lena: λ_{min}: 9/12 Features. $\lambda_{\texttt{1se}}$: 5/12 Features.

Ridge (alle relevant), Lasso (wenige relevant), Elastic Net (korrelierte Gruppen).

Koeffizientenpfade: Lasso und Lambda-Effekt



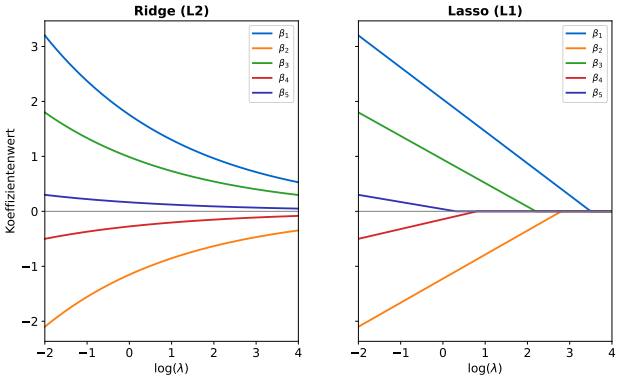
Lasso-Pfade: Koeffizienten fallen mit λ nacheinander auf Null. Unwichtige zuerst.



Lambda-Effekt: Feature, das am längsten überlebt = wichtigstes.

Koeffizientenpfade zeigen die Rangfolge der Feature-Wichtigkeit.

Ridge- vs. Lasso-Pfade im Vergleich



Ridge-Pfade:

- Sanfte, stetige Schrumpfung
- Kein β exakt Null
- Alle Features bleiben

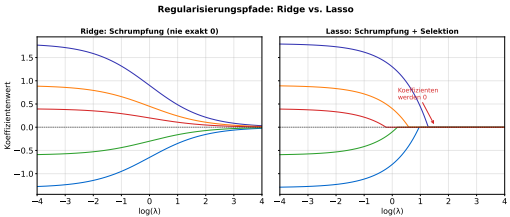
Lasso-Pfade:

- Koeffizienten springen auf Null
- Unwichtige zuerst
- Scharfe Knicke = Selektion

Diagnose: Ridge $\rightarrow$ stetig. Lasso $\rightarrow$ welche Features bei welchem λ eliminiert.

Ridge: stetige Schrumpfung. Lasso: scharfe Selektion an den Knicken.

Vorher/Nachher, Vergleich und Geschichte



Geschichte:

- **1970:** Hoerl – Ridge
- **1996:** Tibshirani – Lasso
- **2005:** Zou/Hastie – Elastic Net

OLS vs. Lasso (500 Kunden, 12 Features):

Feature	OLS	Lasso	
Beschwerden	0,35	0,31	geschrumpft
Nutzung	−0,33	−0,28	geschrumpft
PLZ	3,20	0,00	entfernt
Browser	−7,80	0,00	entfernt
Wochentag	1,50	0,00	entfernt

Drei Irrtümer:

1. „Verbessert immer" – nur bei Overfitting
2. „Lasso immer besser" – Ridge bei vielen kleinen Effekten
3. „Großes λ" – zu groß $\rightarrow$ alle $\beta \rightarrow 0$

35 Jahre: vom Chemieproblem zur ML-Standardmethode.

R-Code: glmnet-Pipeline

```
library(glmnet); set.seed(42)
n <- 200; p <- 20
X <- matrix(rnorm(n*p), n, p)
true_beta <- c(3, -2, 1.5, -1, 0.5, rep(0, 15))
y <- X %*% true_beta + rnorm(n, 0, 2)
cv_lasso <- cv.glmnet(X, y, alpha=1, nfolds=10)
cat("lambda.min:", round(cv_lasso$lambda.min, 4))
coef(cv_lasso, s="lambda.min")  # x1-x5 erkannt
cv_ridge <- cv.glmnet(X, y, alpha=0, nfolds=10)
cat("Ridge Nullen:",
    sum(coef(cv_ridge, s="lambda.min")[-1]==0)) # 0
cv_enet <- cv.glmnet(X, y, alpha=0.5, nfolds=10)
cat("ENet Nullen:",
    sum(coef(cv_enet, s="lambda.min")[-1]==0))
lasso_fit <- glmnet(X, y, alpha=1)
plot(lasso_fit, xvar="lambda", label=TRUE)
```

Ridge: 0 Nullen. Lasso: viele Nullen. Elastic Net: dazwischen. cv.glmnet wählt λ per CV.

Übung und Zwischenfazit: Regularisierung

Übungsaufgaben:

1. Warum setzt L1 β auf exakt Null, L2 nicht? (Diamant vs. Kreis)
2. λ steigt von 0 $\rightarrow$ groß. Welches Feature überlebt am längsten?
3. Ridge, Lasso, oder Elastic Net? (a) 8 Features, alle relevant. (b) 200 Features, meist irrelevant. (c) 15 Features, 5 korrelierte Gruppen.

Zwischenfazit: Reg. opfert Bias, gewinnt Stabilität. λ per CV. **Ridge:** Schrumpft alle. **Lasso:** Selektiert. **Elastic Net:** Kombiniert. **Weiter:** $\rightarrow$ Kreuzvalidierung

Ridge schrumpft alle, Lasso selektiert, Elastic Net kombiniert. Lambda per CV.

Drei Splits, drei Zahlen – Rotierende Richter

Lena: "78%, 71%, 82% – welche stimmt?"

Problem: Ein Split = eine Klausuraufgabe.

Train/Test-Split: 80/20-Aufteilung. Zufall bestimmt Testset – Spanne 11 Pp.!

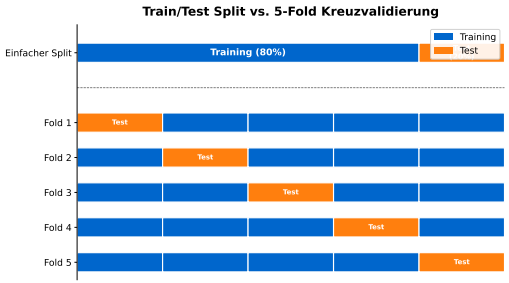
Lösung: Rotierende Richter

- 5 Richter, jede Runde urteilt **einer**
- Nach 5 Runden hat **jeder** einmal geurteilt
- Mittel der 5 Urteile ist fairer

Übertragung: Jeder **Fold** = einmal Testset. Jeder Datenpunkt **genau einmal** getestet.

Ein Split = eine Aufgabe. Kreuzvalidierung = die ganze Klausur. Varianz sinkt.

K-Fold Cross-Validation – Algorithmus & Formel



K-Fold CV: Daten in K Folds. Jeder einmal Testset.

Algorithmus:

1. Mische, teile in K Folds
2. Train auf $K-1$, Test auf k, speichere e_k
3. $\widehat{\mathrm{CV}} = \frac{1}{K} \sum_{k=1}^{K} e_k$

$$\mathrm{SD}_{\mathrm{CV}} = \sqrt{\frac{1}{K-1} \sum_{k=1}^{K} (e_k - \widehat{\mathrm{CV}})^2}$$

Wahl von K:

- $K = 5$: Weniger Aufwand, etwas mehr Bias
- $K = 10$: Standard (Breiman 1992, Kohavi 1995)
- $K = n$ (LOOCV): Min. Bias, max. Varianz

Standard: K=10. **Kleine** $\mathrm{SD}_{\mathrm{CV}}$ $\rightarrow$ **robust.**

CV-Varianten im Überblick

Variante	K	Wann?	Details
K-Fold	5 / 10	Standard	Jeder Punkt $1\times$ im Test
LOOCV	n	$n < 50$	n Modelle; hohe Varianz
Stratified	5 / 10	Unbalanciert	Klassenverteilung beibehalten
Nested	a+i	Hyperpar.-Tuning	Äußere: Performance; innere: λ
Time Series	var.	Zeitlich	Nur Vergangenheit trainiert

Entscheidungsregel:

- Unbalanciert? $\rightarrow$ Stratified
- Zeitreihen? $\rightarrow$ Time Series CV
- Hyperparameter? $\rightarrow$ Nested
- Sonst: K=10

Die richtige CV-Strategie hängt vom Datentyp und der Fragestellung ab.

CV + Regularisierung: Das Dream-Team

cv.glmnet: 100 λ-Werte $\rightarrow$ K-Fold CV $\rightarrow$ `lambda.min` oder `lambda.1se`.
Lenas 5-Fold CV (logistisches Lasso):

	F1	F2	F3	F4	F5	**Mittel (SD)**
Accuracy	.74	.78	.77	.73	.78	**.760 (.024)**
AUC	.81	.84	.83	.80	.82	**.820 (.015)**

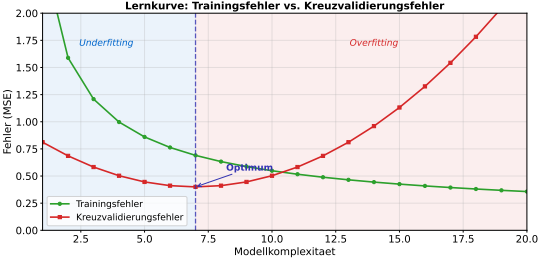
Modellvergleich (gleiche 5 Folds):

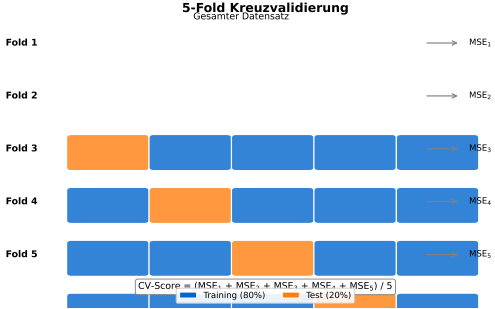
Modell	**CV-Acc (SD)**	**CV-AUC (SD)**
OLS (alle)	0.72 (0.04)	0.78 (0.03)
Ridge	0.75 (0.03)	0.81 (0.02)
Lasso	**0.76 (0.02)**	**0.82 (0.02)**

Lasso gewinnt: Höchste AUC, kleinste Streuung, 3/12 Prädiktoren eliminiert.

CV-Modellvergleich: Mittelwert UND Streuung beachten!

Lernkurve, CV-Visualisierung & Geschichte





Geschichte:

- **1974 Stone:** Datenaufteilung
- **1992 Breiman:** K=10 optimal
- **1995 Kohavi:** Stratified Standard

Drei CV-Irrtümer:

- „LOOCV immer best" – hohe Varianz
- „CV ersetzt Holdout" – nur Selektion
- „Mehr Folds besser" – ab K=10 minimal

Sweet Spot der Lernkurve: wo CV-Fehler minimal. LOOCV nur bei $n < 50$.

R-Code: Kreuzvalidierung (caret, cv.glmnet, manuell)

```
# --- caret ---
library(caret)
ctrl <- trainControl(method="cv", number=5,
  classProbs=TRUE, summaryFunction=twoClassSummary)
model <- train(churn~., data=dataco, method="glm",
  family="binomial", trControl=ctrl, metric="ROC")
# --- cv.glmnet ---
library(glmnet)
x <- model.matrix(churn~., data=dataco)[,-1]
cv_fit <- cv.glmnet(x, dataco$churn, alpha=1,
  family="binomial", nfolds=5)
coef(cv_fit, s="lambda.min")  # 3 von 12 = 0
# --- Manuell ---
set.seed(42); K <- 5
folds <- sample(rep(1:K, length.out=nrow(dataco)))
errors <- numeric(K)
for (k in 1:K) {
  fit_k <- glm(churn~., dataco[folds!=k,], binomial)
  pred_k <- predict(fit_k, dataco[folds==k,], "resp")
  errors[k] <- mean((pred_k>0.5)==dataco$churn[folds==k])
}
cat("Acc:", mean(errors), "SD:", sd(errors))
```

caret: Stratified automatisch. cv.glmnet: 100 λ-Werte. Manuell: innere Logik.

Übung & Zwischenfazit: Kreuzvalidierung

Übungsaufgaben:

1. **Folds:** $n = 120$, $K = 10$. Pro Fold? Pro Training? Wie oft trainiert?
2. **Modellwahl:** A: 0.80 (SD 0.08). B: 0.77 (SD 0.02). Welches?
3. **Opt. Bias:** Warum entsteht er bei λ-Wahl auf denselben Daten?

Zwischenfazit:

- K-Fold nutzt **alle** Daten
- Beste Out-of-Sample-Schätzung
- Standard: K=10
- Stratified bei Unbalance
- Nested bei Hyperpar.-Tuning
- CV + Regularisierung = Dream-Team

Von unsicherem Split zu robuster Schätzung – CV macht Modellvergleich fair.

Korrelation $\neq$ Kausalität – Der Puppenspieler

Lena: Marketing $\leftrightarrow$ Bindung: $r = 0{,}45$. Chef will Budget verdreifachen – kausal?

Eis & Ertrinken: **Temperatur** treibt beides.

Kausalität: Änderung in X (ceteris paribus) $\rightarrow$ Änderung in Y.
Confounder: Z beeinflusst X und Y $\rightarrow$ Scheinkorrelation.

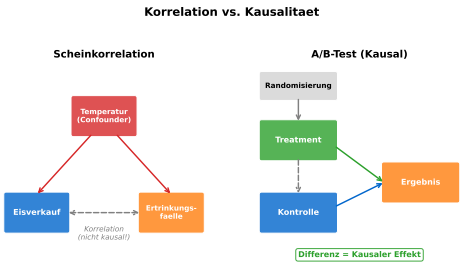
Drei Mechanismen hinter Korrelation:

Mechanismus	Beispiel
Confounding	Z treibt X und Y. *Temperatur* $\rightarrow$ *Eis & Ertrinken.*
Reverse	$Y \rightarrow X$. *Kranke nehmen Medikamente.*
Spurious	Tyler Vigen: Maine-Scheidung $\sim$ Margarine (r=0,99).

In keinem Fall würde Intervention wirken!

Hinter jeder Korrelation: Gibt es einen versteckten Dritten?

Fünf Kriterien & der Goldstandard



Fünf Kausalitätskriterien (Bradford Hill 1965):

1. **Korrelation** – notwendig, nicht hinreichend
2. **Zeitfolge** – Ursache vor Wirkung
3. **Kein Confounding** – härteste Bedingung
4. **Mechanismus** – plausible Erklärung
5. **Konsistenz** – wiederholbar

Goldstandard: Randomisiertes Experiment

- Eliminiert **alle** Confounders
- Kein Experiment möglich? → 5 Kriterien als Ersatz

Korrelation kann durch Confounding, Reverse Causation oder Zufall entstehen.

Pearls Leiter, do-Operator & DAGs

Pearls Leiter (Turing 2011):

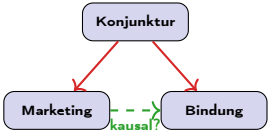
Stufe	Frage
1: Sehen	Mails öffnen $\leftrightarrow$ seltener kündigen
2: Tun	Mehr Mails $\rightarrow$ weniger Kündigung?
3: Vorstellen	Hätte Kunde ohne Mail gekündigt?

do-Operator:

$$P(Y \mid X{=}x) \neq P(Y \mid \mathrm{do}(X{=}x))$$

$\mathrm{do}()$ kappt kausale Einflüsse auf X.

DAGs:

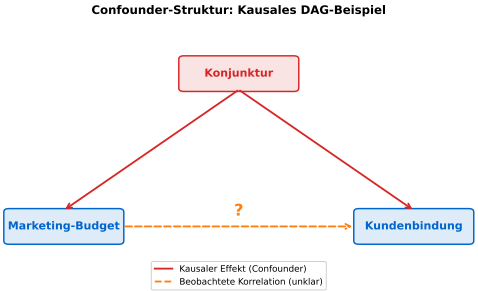


Drei DAG-Bausteine:

- **Fork**: $X \leftarrow Z \rightarrow Y$ – Z kontrollieren
- **Kette**: $X \rightarrow M \rightarrow Y$ – M **nicht** kontrollieren
- **Collider**: $X \rightarrow C \leftarrow Y$ – C **nicht** kontrollieren!

DAGs machen kausale Annahmen explizit. do-Operator: "Was wenn ich eingreife?"

Confounding, Lenas Puzzle & Kausale Methoden



Lenas Puzzle:

1. Roh: Marketing $\leftrightarrow$ Retention: $r = 0{,}45$
2. Konjunktur korreliert mit beiden ($r > 0{,}5$)
3. Partiell: $r_{\mathbf{Mkt,Ret|Konj}} = 0{,}12$ (n.s.!)

Fazit: Scheinkorrelation. $\rightarrow$ A/B-Test nötig.

Kausale Methoden (ohne Experiment):

Methode	Kernidee	Annahme
Kontrollvar.	Confounder ins Modell	Alle bekannt
Matching	Vergleichbare Paare	Gute Matches
IV	Exogene Variation	Gültiges Instrument
DiD	Vorher/Nachher + Kontrolle	Parallele Trends

Partielle Korrelation: einfachstes Confounder-Werkzeug. Jede Methode hat eine kritische Annahme.

R-Code, Geschichte & Übung: Kausalität

```
cor.test(dataco$marketing, dataco$retention)
# r = 0.45, p = 0.027
library(ppcor)
pcor.test(dataco$marketing, dataco$retention,
          dataco$konjunktur)  # r=0.12, p=0.38
library(ggdag)
dag <- dagify(Retention ~ Marketing + Konjunktur,
  Marketing ~ Konjunktur,
  exposure="Marketing", outcome="Retention")
adjustmentSets(dag)  # => { Konjunktur }
```

Geschichte:

- Pearl (Turing 2011): do-Kalkül
- Rauchen/Krebs: 50 J. Debatte; Hill 1965
- Vigen (2014): Spurious Correlations

Übung:

1. Kausal? (a) Frühstück → Noten? (b) Feuerwehr → Brände?
2. Pearls Leiter: (a) "Leser kaufen mehr" (b) "Newsletter → Umsatz" (c) "Hätte Kunde X?"

Von Fishers Irrtum bis Pearls Revolution – Kausalität hat eine reiche Geschichte.

A/B-Test: Vier Schritte & Testdesign

Lena: Korrelation war Confounding → A/B-Test! 1000 Kunden, 30 Tage.

Bei n=20: Power nur 12%!

Vier Schritte:

1. **Zufällige Zuteilung** – keine Selbstselektion
2. **Intervention nur in B** – A = Kontrolle
3. **Ergebnis messen** – primäre Metrik
4. $\Delta = \bar{Y}_B - \bar{Y}_A$ = kausaler Effekt

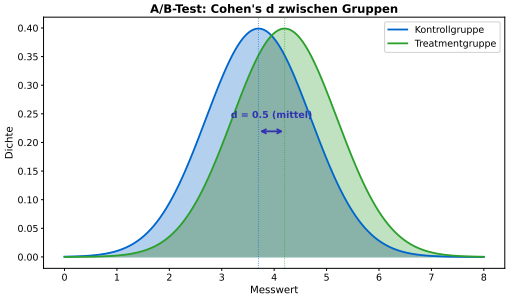
Randomisierung → Gruppen identisch in **allen** Variablen.

Fünf Vorab-Entscheidungen:

1. **Hypothese:** Neues Onboarding erhöht Retention
2. **Metrik:** Retention (binär) – nur *eine*
3. **MDE:** $d = 0{,}3$
4. n: Basierend auf MDE, Power, α
5. **Laufzeit + Pre-Registration**

A/B-Tests = Goldstandard. Planung dauert oft länger als das Experiment.

Stichprobengröße, Cohen's d & Power



- **Blau:** H_0. **Orange:** H_1
- Power $= 1 - \beta$
- Mehr $n \rightarrow$ höhere Power

Sample-Size-Formel (pro Gruppe):

$$n \approx \frac{2\,(z_{\alpha/2} + z_{\beta})^2}{d^2}$$

Lena: d=0,3 $\rightarrow$ $n \approx 174$/Gruppe. Plant 500.

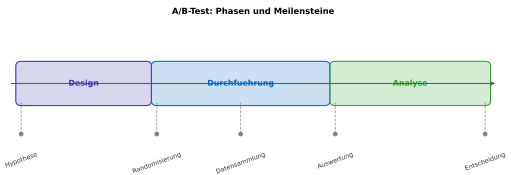
Cohen's d: $d = \frac{\bar{x}_T - \bar{x}_C}{s_p}$

d	Interpretation
0.2	Klein
0.5	Mittel
0.8	Groß

d unabhängig von n – bei 1 Mio. wird d=0,01 signifikant!

Power = P(Effekt erkennen). Effektstärke vorher festlegen!

Lenas A/B-Test: Ergebnis & Drei Fehler



Drei Phasen:

1. **Design:** H_0, Metrik, MDE, n
2. **Durchführung:** 30 Tage, kein Peeking
3. **Analyse:** Effektstärke, Subgruppen

Ergebnis: A: 68%, B: 76%. $d = 0{,}45$, $p = 0{,}003$. Signifikant + relevant.
Bei 10k Neukunden/Monat: **+400k CHF/Monat**.

Drei Fehler:

1. **Peeking:** Tägliches Prüfen → FP-Rate bis 30%
2. **Multiple Testing:** 10 Metriken → 40% FP. Lösung: eine primäre Metrik
3. **Simpson:** Gesamt positiv, Subgruppen negativ

Lena überzeugt Board mit Zahlen und kausalem Beweis.

275 Jahre kontrollierte Experimente & Irrtümer

Geschichte:

- **1747 Lind:** 6 Behandlungen, 12 Matrosen. Zitrus hilft. Erste kontrollierte Studie.
- **1920er Fisher:** Randomisierung formalisiert. Lady Tasting Tea.
- **2008 Obama:** 24 Versionen, **+60 Mio. $**.
- **Heute:** Google 10k+ Tests/Jahr.

Drei Irrtümer:

1. **"Immer klare Antworten"** – Nicht-signifikant $\neq$ kein Effekt (Power!)
2. **"Signifikant = wichtig"** – Bei großem n wird d=0,02 signifikant
3. **"Peeking ist okay"** – Jedes Prüfen erhöht FP-Rate

Von Skorbut (1747) bis Silicon Valley – A/B-Tests sind aktueller denn je.

R-Code & Übung: A/B-Testing

```
t.test(retention ~ gruppe, data = ab_daten)
library(effsize)
cohen.d(retention ~ gruppe, data = ab_daten)
power.t.test(delta=0.3, sd=1, sig.level=0.05,
             power=0.80, type="two.sample") # n=176
```

Übungsaufgaben:

1. **Power:** $d = 0{,}4$, $\alpha = 0{,}05$, Power $= 0{,}80$. $n \approx 2(1{,}96 + 0{,}84)^2/d^2 = ?$
2. **Cohen's d:** $\bar{X}_C = 48$, $s_C = 12$; $\bar{X}_T = 52$, $s_T = 14$ (je $n = 200$). s_p? d?
3. **Peeking:** Stopp nach 8 Tagen bei $p = 0{,}04$ (Plan: 30). FP-Rate bei $k = 8$: $1 - 0{,}95^k$?

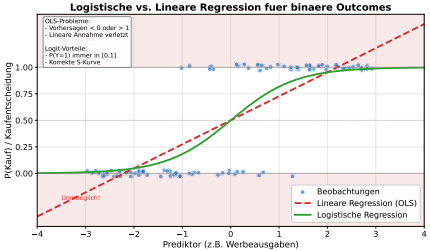
---

t.test: Signifikanz. cohen.d: Effektstärke. power.t.test: Planung.

Warum OLS versagt – Die logistische Funktion

Lena: $P = -0{,}12$ und $P = 1{,}35$? Unsinn!

Churn ist binär. Dimmer statt Schalter.



OLS-Probleme:

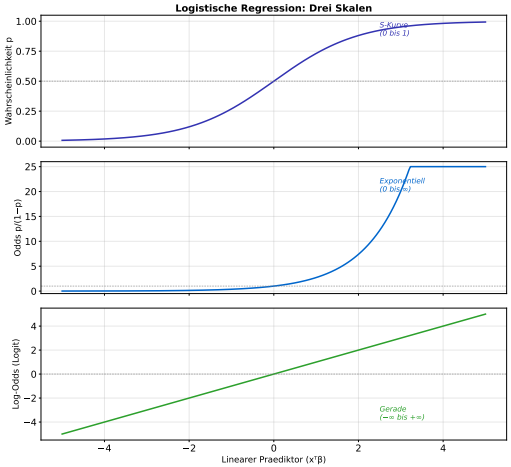
1. $P < 0$ oder $P > 1$
2. Beziehung S-förmig
3. Var $= P(1-P)$

$$P(Y{=}1|X) = \frac{\mathbf{1}}{\mathbf{1}+e^{-(\beta_\mathbf{0}+\beta_\mathbf{1}X)}}$$

- Beschränkt auf $(0, 1)$
- Symmetrisch um $P{=}0{,}5$
- Steilste Stelle bei 0,5

Sigmoid: jeder reelle Wert $\rightarrow (0, 1)$. OLS = Gerade, Logit = S-Kurve.

Drei Skalen: Wahrscheinlichkeit, Odds, Log-Odds



Drei Perspektiven:

- P: $(0, 1)$, S-förmig
- **Odds** $\frac{P}{1-P}$: $(0, \infty)$, multiplikativ
- **Log-Odds**: $(-\infty, +\infty)$ – hier **linear**!

Logit: $\ln \frac{P}{1-P} = \beta_0 + \beta_1 X$

Odds Ratio: $OR = e^{\beta_1}$

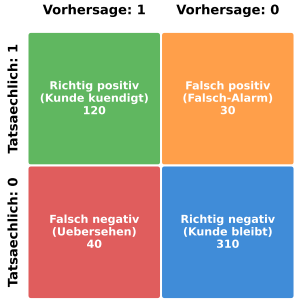
- e^{β_1} = Odds-Faktor bei +1 Einheit
- $OR = 1$: kein Effekt. > 1: erhöht. < 1: verringert

Achtung: OR $\neq$ RR! Nur bei $p < 0{,}10$: $OR \approx RR$.

Log-Odds-Skala: Regression linear. $\exp(\beta)$ **gibt interpretierbare Odds Ratios.**

Confusion Matrix & Klassifikationsmetriken

Confusion Matrix: Lenas DataCo-Modell



Accuracy = 86% | Precision = 80% | Recall = 75% | F1 = 77%

Vier Metriken:

- **Accuracy** $= 456/500 = 91{,}2\%$
- **Precision** $= 56/81 = 69{,}1\%$
- **Recall** $= 56/75 = 74{,}7\%$
- **F1** $= 71{,}8\%$

„Immer Bleibt" = 85% Acc – kein Kündiger!

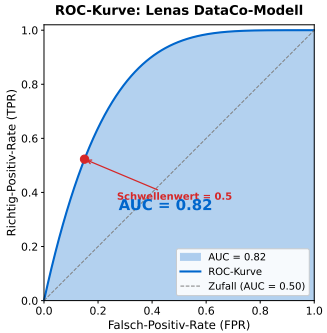
Premium: $OR = 2{,}5 \rightarrow 40\% \rightarrow 63\%$ Bleiben.

Lena (n=500, 75 Kündiger):

	Bleibt	Kündigt
Bleibt	400 (TN)	25 (FP)
Kündigt	19 (FN)	56 (TP)

Precision/Recall wichtiger als Accuracy bei Unbalance. FN = teuerster Fehler.

ROC/AUC & DataCo glm()-Output



AUC: 0,5 = Zufall. 0,8 = Gut. >0,9 = Exzellent.
Lena: AUC = 0,82.

glm()-Output:

Variable	β	OR	Bedeutung
Premium	−0,92	0.40	−60% Odds
Alter	0,02	1.02	+2%/Jahr
Nutzung	−0,05	0.95	−5%/Monat
Beschwerden	0,30	1.35	+35%/Stk.

Risikoprofil: Kein Premium, wenig Nutzung, 3+ Beschwerden.
Strategie: Premium-Konversion, Frühwarnung bei $P > 0{,}6$.

ROC: TPR vs. FPR bei jedem Schwellenwert. Modell erklärt wer kündigt und warum.

Titanic, Anwendungen, Berkson & Irrtümer

Titanic:

Feature	OR	Interpretation
Weiblich	10,5	10× Überlebens-Odds
1. Klasse	3,8	Reiche überlebten öfter
Kind	2,1	Frauen+Kinder zuerst

Anwendungen:

- Churn, Conversion, Credit Scoring
- Fraud Detection (AUC entscheidend)

Geschichte:

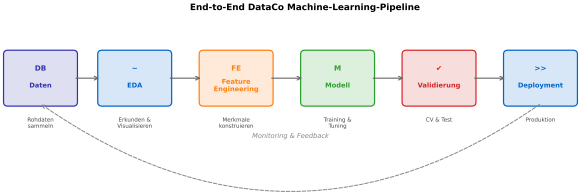
- 1838 Verhulst: Log. Funktion
- 1944 Berkson: „Logit" (Mayo Clinic)
- Heute: meistgenutztes Klassifikationsmodell

Irrtümer:

- „Keine Regression" – linear auf Log-Odds
- „Accuracy beste Metrik" – 5% Churn: 95% ohne Erkennung
- „OR = RR" – nur bei $p < 0{,}10$

Log. Regression: einfach, schnell, interpretierbar – Baseline für Klassifikation.

Pipeline, R-Code & Übung: Logistische Regression




```
logit_model <- glm(churn ~ premium + alter +
  nutzung_monate + beschwerden,
  data=dataco, family=binomial)
exp(cbind(OR=coef(logit_model), confint(logit_model)))
library(pROC)
auc(roc(dataco$churn,
    predict(logit_model, type="response")))  # 0.82
pred <- ifelse(predict(logit_model,"response")>0.5,1,0)
caret::confusionMatrix(factor(pred), factor(dataco$churn))
```

Übung: (1) $\beta = 0{,}47 \rightarrow$ OR? Bei p=0,30: neues p? (2) TN=9920, FP=30, FN=10, TP=40 $\rightarrow$ Acc, Prec, Rec, F1? (3) AUC 0,72 vs. 0,88?

Rohdaten → BLUE → VIF → Regularisierung → CV → Kausalität → A/B → Logit → Entscheidung.

Lenas Reise durch Block 3 – Zusammenfassung

Acht Erkenntnisse:

1. **BLUE:** OLS nur unter Annahmen optimal $\rightarrow$ Diagnostik
2. **Multikollinearität:** Instabile β $\rightarrow$ VIF prüfen
3. **Bias-Varianz:** Perfekte Anpassung $\neq$ gute Vorhersage
4. **Regularisierung:** Lasso/Ridge $\rightarrow$ robustere Modelle
5. **CV:** Out-of-Sample ist der wahre Maßstab
6. **Kausalität:** Korrelation $\neq$ Kausalität $\rightarrow$ DAGs
7. **A/B-Test:** Goldstandard $\rightarrow$ Power, Cohen's d
8. **Logit:** Sigmoid, OR, ROC/AUC, Confusion Matrix

Lena: AUC $= 0{,}82$, A/B-Test +8pp Retention, Board überzeugt.

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?

Von deskriptiver Analyse über Inferenz zur kausalen Schlussfolgerung – Data Science.