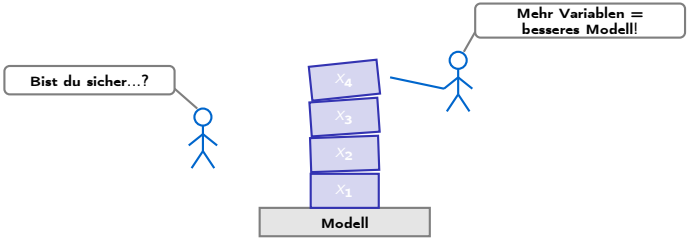
Block 3: Komplexität, Kausalität & Generalisierung

Data Science and Strategy for Business

March 31, 2026

Warum einfach, wenn's auch kompliziert geht?



Block 3 zeigt: Komplexitaet kontrollieren ist genauso wichtig wie sie aufzubauen.

Lernziele Block 3

- Multikollinearität erkennen und Auswirkungen erklären
- BLUE-Eigenschaften konzeptionell einordnen
- Overfitting und In-/Out-of-Sample Performance unterscheiden
- Kreuzvalidierung und Regularisierung erklären
- Korrelation von Kausalität unterscheiden
- A/B-Tests als kausale Inferenz verstehen
- Logistische Regression von OLS abgrenzen

Dieser Block bereitet auf fortgeschrittene ML-Methoden vor (Modul ER017).

BLUE-Eigenschaften

OLS-Schätzer sind BLUE:

- **B**est – kleinste Varianz
- **L**inear – lineare Funktion der Daten
- **U**nbiased – erwartungstreu
- **E**stimator – Schätzer

Voraussetzungen:

1. Linearität
2. Keine perfekte Multikollinearität
3. Exogenität (keine Korrelation X mit ε)
4. Homoskedastizität
5. Keine Autokorrelation

Bei Verletzung:

- Verzerrte Schätzungen
- Falsche Standardfehler
- Ungültige Tests
- Schlechte Vorhersagen

Diagnostik:

- Residuen-Plots
- VIF für Multikollinearität
- Breusch-Pagan Test

BLUE gilt nur unter idealen Bedingungen – in der Praxis oft verletzt.

BLUE: Best erklärt

Was bedeutet "Best"?
Unter allen linearen, erwartungstreuen Schätzern hat OLS die kleinste Varianz.

Gauss-Markov-Theorem
Wenn Annahmen erfuellt, gibt es keinen besseren linearen Schätzer als OLS.

Wichtig:
"Best" bezieht sich nur auf Varianz, nicht auf Bias!

Einschraenkungen

- Gilt nur für lineare Schätzer
- Non-lineare können besser sein
- Bei kleinem n oft irrelevant
- Regularisierte Schätzer haben niedrigeren MSE

Praxis:
In vielen Faellen ist ein Schätzer mit etwas Bias aber weniger Varianz vorzuziehen (Bias-Variance-Tradeoff)!

BLUE ist theoretisch schoen, praktisch aber nicht immer optimal.

Annahmen prüfen

1. Linearität
Residuen vs. Fitted Values Plot
Sollte kein Muster zeigen!

2. Homoskedastizität
Konstante Varianz der Residuen
Breusch-Pagan Test:
`bptest(model)`

3. Normalität
Q-Q Plot der Residuen
Shapiro-Wilk Test

4. Keine Autokorrelation
Durbin-Watson Test:
`dwtest(model)`
Wichtig bei Zeitreihen!

5. Exogenität
Schwer zu testen
Theoretische Ueberlegungen nötig
Instrumentvariablen bei Verletzung

In R:
`plot(model)` gibt 4 Diagnose-Plots

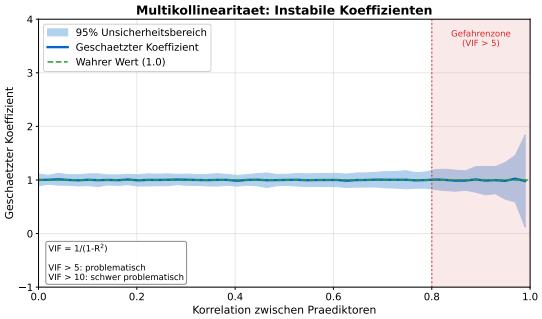
Immer Annahmen prüfen – blinde Anwendung von OLS ist gefaehrlich.

Multikollinearität

Definition
Prädiktoren sind stark miteinander korreliert.

Symptome

- Hohe R^2, aber keine signifikanten Koeffizienten
- Instabile Koeffizienten
- Grosse Standardfehler
- Vorzeichenwechsel bei Koeffizienten



VIF $>$ 5 ist problematisch, VIF $>$ 10 ist schwer problematisch.

Multikollinearitaet: Wer war's?

Das Zwillingsproblem

- Wenn X_1 und X_2 stark korreliert sind, kann OLS ihre Effekte nicht trennen
- Standardfehler werden gross
- Koeffizienten instabil
- R^2 bleibt hoch, aber einzelne p-Werte nicht signifikant

Diagnose: VIF (Variance Inflation Factor)

Wenn Praediktoren korreliert sind, kann das Modell ihre Beitraege nicht trennen.

Multikollinearität: Lösungen

Diagnose: VIF

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

$R_j^2 = R^2$ wenn Variable j auf alle anderen regressiert wird.

In R:

```
library(car)
vif(model)
```

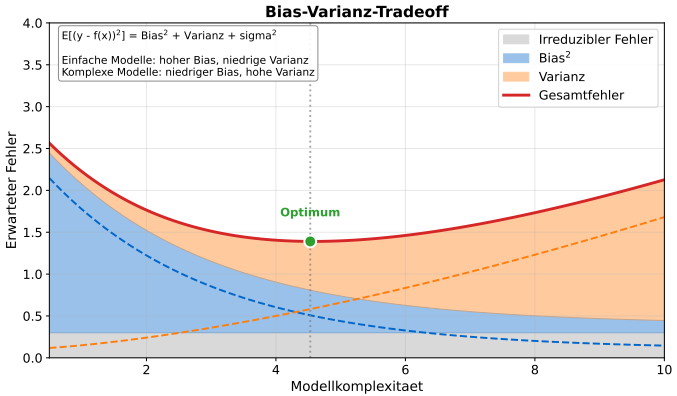
Lösungen

1. Variable entfernen
2. Variablen kombinieren (Index)
3. Regularisierung (Lasso/Ridge)
4. Hauptkomponentenanalyse
5. Mehr Daten sammeln

Wichtig:
Multikollinearität ist für Interpretation problematisch, weniger für Vorhersage!

Bei reiner Vorhersage kann Multikollinearität toleriert werden.

Bias-Varianz-Tradeoff



Das Ziel ist minimaler Gesamtfehler, nicht minimaler Bias oder minimale Varianz.

Bias vs. Varianz

Bias (Verzerrung)

- Systematischer Fehler
- Modell zu einfach
- Wichtige Zusammenhaenge fehlen
- "Underfitting"

Beispiel:
Lineare Regression für quadratischen Zusammenhang.

Gesamtfehler: $E[(y - \hat{y})^2] = Bias^2 + Varianz + \sigma^2$

Varianz

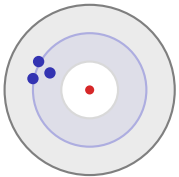
- Zufällige Schwankung
- Modell zu komplex
- Lernt Rauschen mit
- "Overfitting"

Beispiel:
Polynom Grad 20 für 10 Datenpunkte.

Optimale Modellkomplexität balanciert Bias und Varianz.

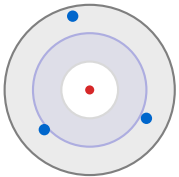
Bias vs. Varianz: Ein Bild sagt mehr

Hoher Bias



Konsistent daneben

Hohe Varianz



Verstreut um die Mitte

Ziel: Niedrig in beidem – das ist der Sweet Spot.

Optimale Modellkomplexitaet minimiert den Gesamtfehler aus Bias und Varianz.

Overfitting erkennen

Symptome

- Training-Fehler sehr klein
- Test-Fehler viel größer
- Komplexes Modell mit vielen Parametern
- Koeffizienten sehr gross

Ursachen

- Zu wenig Daten
- Zu viele Features
- Zu flexibles Modell
- Kein separates Testset

Gegenmassnahmen

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

Faustregel:
Mindestens 10-20 Beobachtungen pro Prädiktor!

Overfitting ist das Hauptproblem in Machine Learning – immer validieren!

Regularisierung: Lasso und Ridge

Idee
Koeffizienten "bestrafen" um Overfitting zu reduzieren.

Ridge (L2):

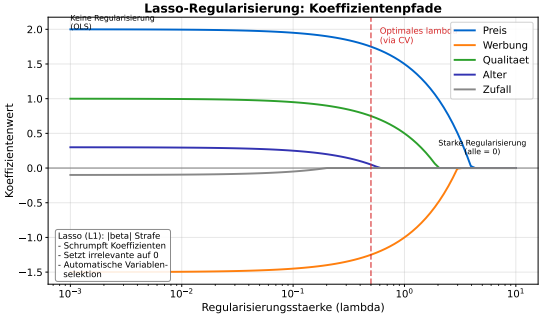
$$\min \sum (y_i - \hat{y}_i)^2 + \lambda \sum \beta_j^2$$

Schrumpft Koeffizienten, entfernt keine.

Lasso (L1):

$$\min \sum (y_i - \hat{y}_i)^2 + \lambda \sum |\beta_j|$$

Schrumpft und setzt auf 0 (Variablenselektion!).



λ **wird per Kreuzvalidierung gewählt.**

Lasso vs. Ridge: Wann was?

Ridge bevorzugen wenn:

- Viele kleine Effekte
- Alle Variablen wichtig
- Multikollinearität
- Stabilere Koeffizienten gewuenscht

Lasso bevorzugen wenn:

- Sparsity erwartet (wenige wichtige)
- Variablenselektion gewuenscht
- Interpretierbarkeit wichtig
- Sehr viele Features

Elastic Net
Kombination aus beiden:

$$\lambda_1 \sum |\beta_j| + \lambda_2 \sum \beta_j^2$$

In R (glmnet): `alpha = 1`: Lasso
`alpha = 0`: Ridge
`alpha = 0.5`: Elastic Net

Praxis:
Oft Elastic Net mit CV für alpha und λ.

Elastic Net kombiniert Vorteile von Lasso und Ridge.

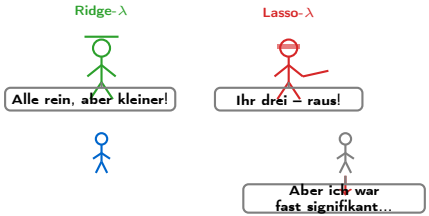
Was macht Lambda an der Tuer?

Regularisierung als Tuersteher

- Ridge (λ): Laesst alle Koeffizienten rein, schrumpft sie aber
- Lasso (λ): Weist unwichtige Koeffizienten ab (setzt auf 0)
- Je groesser λ, desto strenger der Tuersteher

Praxis:

- λ per Kreuzvalidierung waehlen
- Elastic Net kombiniert beide



Ridge schrumpft alle Koeffizienten, Lasso setzt unwichtige auf exakt Null.

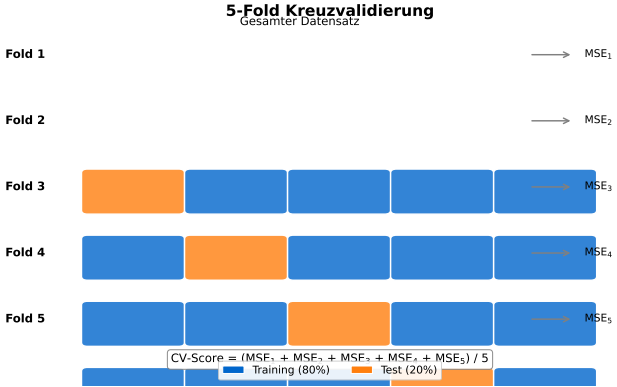
Regularisierung in R

Mit glmnet:

```
library(glmnet)

# Daten vorbereiten
X <- model.matrix(y ~ ., data)[,-1]  # ohne Intercept
y <- data$y

# Lasso mit Kreuzvalidierung
cv_lasso <- cv.glmnet(X, y, alpha = 1)  # alpha=1 fuer Lasso
plot(cv_lasso)  # Lambda vs MSE

# Optimales Lambda
best_lambda <- cv_lasso$lambda.min

# Finales Modell
final_model <- glmnet(X, y, alpha = 1, lambda = best_lambda)
coef(final_model)  # Koeffizienten (viele = 0 bei Lasso)
```

cv.glmnet waehlt automatisch das beste Lambda per Kreuzvalidierung.

Kreuzvalidierung



K-Fold CV: Jeder Datenpunkt ist einmal im Testset.

Kreuzvalidierung: Varianten

K-Fold CV
Daten in K Teile teilen
Standard: K = 5 oder K = 10
Guter Kompromiss Bias/Varianz

Leave-One-Out (LOOCV)
K = n (jeder Punkt einmal raus)
Wenig Bias, hohe Varianz
Rechenintensiv

Stratified CV
Klassenverteilung erhalten
Wichtig bei unbalancierten Daten

Nested CV
CV in CV für Modellselektion
Aussen: Performance schätzen
Innen: Hyperparameter tunen

Time Series CV
Nur vergangene Daten zum Training
Wichtig: Keine Zukunft "leaken"!

Die richtige CV-Strategie hängt vom Problem ab.

Kreuzvalidierung: Details

Warum CV?

- Robustere Schätzung als einzelner Split
- Nutzt alle Daten für Training UND Test
- Standard für Modellwahl

Typische Werte

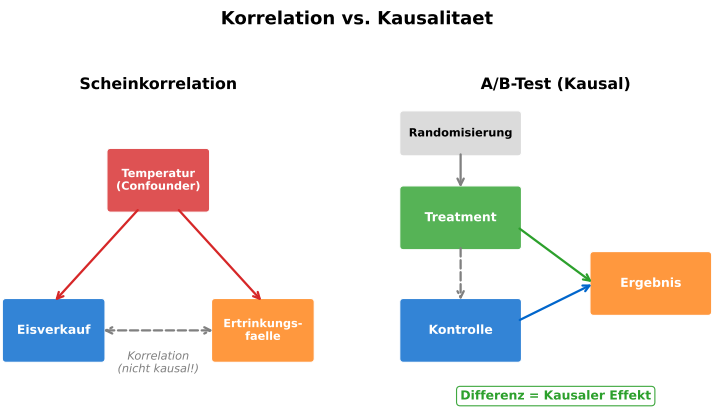
- K = 5 oder K = 10
- Leave-One-Out: K = n

In R (caret):

```
library(caret)
ctrl <- trainControl(
  method = "cv",
  number = 10
)
model <- train(
  y ~ .,
  data = train_data,
  method = "lm",
  trControl = ctrl
)
```

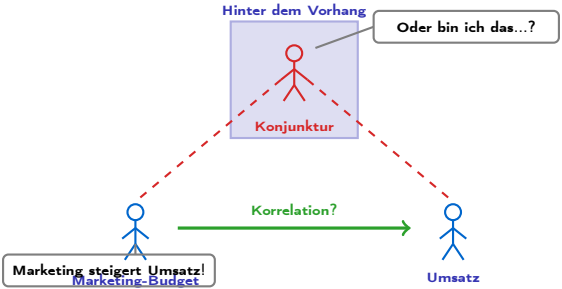
CV-Fehler ist die beste Schätzung für Out-of-Sample Performance.

Korrelation vs. Kausalität



Korrelation kann durch Confounding, Zufall oder umgekehrte Kausalität entstehen.

Warum Korrelation nicht Kausalitaet ist



Ohne Experiment (A/B-Test) koennen versteckte Confounders die wahre Ursache sein.

Wann ist es kausal?

Kausale Kriterien

1. Korrelation vorhanden
2. Zeitliche Reihenfolge
3. Kein Confounding
4. Mechanismus plausibel
5. Konsistenz über Studien

Goldstandard:
Randomisiertes Experiment (A/B-Test)

Observationsdaten:

- Keine Randomisierung
- Confounding möglich
- Nur Assoziation, keine Kausalität

Lösung:

- Kontrollvariablen
- Matching
- Instrumentvariablen
- Difference-in-Differences

Ohne Experiment ist Kausalität schwer nachweisbar.

Confounding: Das Problem

Definition
Eine dritte Variable beeinflusst sowohl X als auch Y.

Klassisches Beispiel:
Eiscremeverkauf korreliert mit Ertrinken.
Confounder: Temperatur!

Business-Beispiel:
Werbung korreliert mit Umsatz.
Confounder: Saison, Events, Budget.

Lösungsansaetze

1. Randomisierung (eliminiert alle Confounder)
2. Kontrolle (statistisch adjustieren)
3. Matching (aehnliche Gruppen)
4. Instrumentvariablen

Problem:
Unbekannte Confounder können nicht kontrolliert werden!
"No unmeasured confounding" ist kritische Annahme.

Confounding ist der Hauptgrund, warum Korrelation nicht Kausalität impliziert.

A/B-Testing

Prinzip

1. Zufällige Zuteilung zu A oder B
2. Intervention nur in Gruppe B
3. Messen des Ergebnisses
4. Differenz = Kausaler Effekt

Warum funktioniert es?
Randomisierung macht Gruppen vergleichbar – alle Confounders sind balanciert!

Beispiele:

- Website-Design
- Preisstrategien
- E-Mail-Kampagnen
- Feature-Tests

Wichtig:

- Genügend Power
- Richtige Metrik
- Keine Interferenz

A/B-Tests sind das Standardtool für kausale Fragen im digitalen Business.

A/B-Test Design

Vor dem Test

1. Hypothese formulieren
2. Primaere Metrik festlegen
3. Minimaler Effekt definieren
4. Sample Size berechnen
5. Laufzeit planen

Häufige Fehler:

- Test zu frueh stoppen
- Viele Metriken ohne Korrektur
- Unklare Hypothese

Während des Tests

- Nicht reinschauen (Peeking Problem!)
- Randomisierung prüfen
- Technische Probleme monitoren

Nach dem Test

- Primaere Metrik auswerten
- Effektgröße berichten
- Heterogenität prüfen
- Dokumentieren!

Gutes A/B-Test Design ist entscheidend für valide Schlussfolgerungen.

Cohen's d im A/B-Test

Definition

$$d = \frac{\bar{X}_B - \bar{X}_A}{s_{pooled}}$$

Interpretation
Unterschied in Standardabweichungen:

- d = 0.2: klein
- d = 0.5: mittel
- d = 0.8: gross

Vorteile

- Unabhängig von n
- Vergleichbar zwischen Studien
- Sagt mehr als p-Wert

Sample Size Calculation:
Mit d und gewuenschter Power laesst sich nötige Stichprobengröße berechnen.

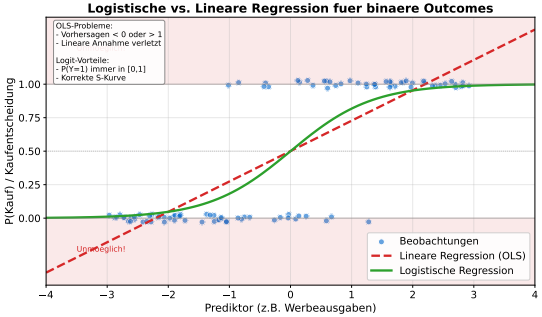
Effektstaerke vor dem Experiment festlegen – wie gross muss der Effekt sein, um relevant zu sein?

Logistische Regression: Motivation

Problem mit OLS

Bei binaerer Zielvariable (0/1):

- Vorhersagen ausserhalb [0,1]
- Linearität verletzt
- Heteroskedastizität



Logit modelliert P(Y=1) und garantiert Werte in [0,1].

Logistische Regression: Modell

Das Modell

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

Log-Odds:

$$\log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X$$

Interpretation:
e^{β_1} = Odds Ratio bei 1 Einheit X mehr

In R:

```
model <- glm(
  y ~ x1 + x2,
  data = data,
  family = binomial
)
summary(model)

# Odds Ratios
exp(coef(model))
```

Logit-Koeffizienten sind log-Odds – exp() gibt Odds Ratios.

Odds Ratios interpretieren

Was ist ein Odds Ratio?

$$OR = \frac{P(Y=1)/(1-P(Y=1))}{P(Y=0)/(1-P(Y=0))}$$

Interpretation

- OR $= 1$: kein Effekt
- OR > 1: erhöhte Chance
- OR < 1: verringerte Chance

Beispiel:
OR = 2.5 bedeutet: 2.5-fache Chance bei 1 Einheit mehr X

Achtung bei Interpretation

- OR ist nicht Relatives Risiko!
- Bei häufigen Ereignissen ueberschätzt OR
- Immer KI mit angeben

Typische Darstellung:
"Kunden mit Premium-Abo haben eine 2.5-fach hoeheres Odds zu konvertieren (OR=2.5, 95%-KI: [1.8, 3.5])"

Odds Ratios sind der Standard in der Medizin und Sozialwissenschaften.

Logistische Regression: Modellguete

Confusion Matrix

	Pred. 0	Pred. 1
Actual 0	TN	FP
Actual 1	FN	TP

Metriken:

- Accuracy = (TP+TN) / N
- Precision = TP / (TP+FP)
- Recall = TP / (TP+FN)
- F1 = 2 * Prec * Rec / (Prec + Rec)

AUC-ROC

Area Under the Receiver Operating Characteristic Curve

- AUC = 0.5: Zufall
- AUC = 0.7-0.8: akzeptabel
- AUC = 0.8-0.9: gut
- AUC > 0.9: exzellent

Wichtig:

Kein R^2 wie bei OLS! Pseudo-R^2 oft irreführend.

Welche Metrik relevant ist, hängt vom Business-Problem ab.

Logistische Regression: Anwendung

Typische Anwendungen:

- **Churn Prediction:** Wird der Kunde kündigen?
- **Conversion:** Wird der Besucher kaufen?
- **Credit Scoring:** Wird der Kredit zurückgezahlt?
- **Fraud Detection:** Ist die Transaktion betrügerisch?

Modellguete:

- Confusion Matrix: TP, FP, TN, FN
- Accuracy, Precision, Recall, F1
- AUC-ROC Kurve
- Kein R^2 wie bei OLS!

Logistische Regression ist die Basis für viele Klassifikationsaufgaben.

Block 3: Zusammenfassung

Konzepte

- BLUE und Annahmen
- Multikollinearität (VIF)
- Bias-Varianz-Tradeoff
- Regularisierung (Lasso)
- Kreuzvalidierung
- Korrelation vs. Kausalität
- A/B-Testing
- Logistische Regression

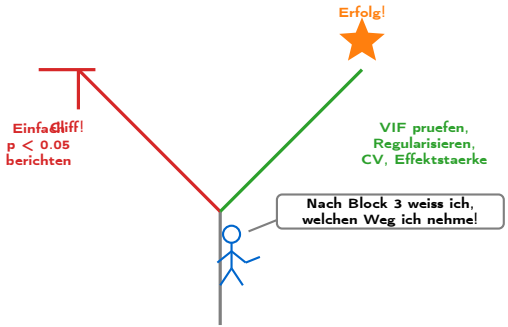
Praktische Skills

- VIF berechnen
- glmnet für Lasso
- caret für CV
- A/B-Test Design
- Cohen's d berechnen
- glm() für Logit
- Odds Ratios interpretieren

Ausblick: Modul ER017 vertieft ML-Algorithmen (Random Forest, Boosting, Neural Networks).

Diese Grundlagen sind essentiell für fortgeschrittene Data Science.

Bereit fuer die Praxis?



Data Science ist mehr als p-Werte – es geht um robuste, validierte Modelle.

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?