Block 1: Grundlagen & Empirisches Fitting

Data Science and Strategy for Business

March 12, 2026

Lernziele Block 1

- Data Science als Schnittmenge von Statistik, Informatik und Domaenenwissen einordnen
- Projekte entlang des CRISP-DM-Zyklus strukturieren
- Daten als strategisches Asset einordnen und Big-Data-Grenzen reflektieren
- R-Projekte anlegen und relative Pfadangaben nutzen
- Tidyverse-Pakete interpretieren und modifizieren
- EDA zur Mustererkennung einsetzen
- Einfache und multiple Regressionen interpretieren
- Overfitting konzeptionell verstehen

Nach diesem Block können Sie Data-Science-Projekte strukturieren und erste Modelle interpretieren.

Was ist Data Science?

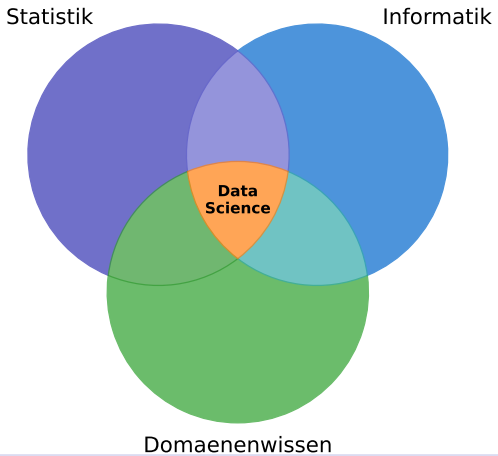
Definition
Data Science ist die modellbasierte Abstraktion der Realität durch:

- Statistische Methoden
- Informatik-Werkzeuge
- Domaenen-Expertise

Abgrenzung

- Keine "Black Box" ohne Verstaendnis
- Keine reine Automatisierung
- Kritisches Denken erforderlich

Data Science als Schnittmenge



Data Science ist mehr als Programmieren – es erfordert statistisches und fachliches Verstaendnis.

Die drei Säulen der Data Science

Statistik

- Wahrscheinlichkeit
- Inferenz
- Modellierung
- Unsicherheit

Informatik

- Programmierung
- Datenbanken
- Algorithmen
- Skalierbarkeit

Domaenenwissen

- Geschaeftslogik
- Branchenkenntnis
- Problemverstaendnis
- Interpretation

Kernkompetenz: Die Fähigkeit, alle drei Bereiche zu verbinden und Ergebnisse zu kommunizieren.

Fehlende Domaenenkenntnis führt zu technisch korrekten, aber inhaltlich unsinnigen Analysen.

"AI is not Magic"

Realistische Erwartungen

- Datenqualität bestimmt Modellqualität
- "Garbage in, garbage out"
- Modelle extrapolieren nicht zuverlaessig
- Korrelation $\neq$ Kausalität

Häufige Missverstaendnisse

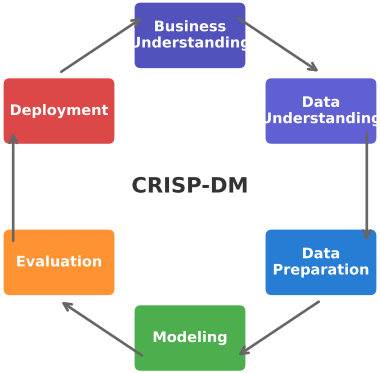
- "Mehr Daten loesen alle Probleme"
- "Das Modell findet die Antwort"
- "AI ersetzt menschliches Urteil"

Was funktioniert:

1. Klare Fragestellung
2. Gute Datenqualität
3. Passende Methode
4. Kritische Interpretation
5. Iterative Verbesserung

Erfolgreiche Data-Science-Projekte erfordern klare Ziele und realistische Erwartungen.

CRISP-DM: Der Projektstandard



CRISP-DM: Phasen 1–3

1. **Business Understanding**
   - Geschaeftsziel definieren
   - Erfolgskriterien festlegen
   - Ressourcen planen
   - Risiken identifizieren
2. **Data Understanding**
   - Datenqüllen erfassen
   - Datenqualität prüfen
   - Erste Exploration
   - Hypothesen bilden
3. **Data Preparation**
   - Daten bereinigen
   - Features ableiten
   - Datensatz aufteilen
   - Dokumentieren

Zeitaufwand: Phasen 2–3 beanspruchen oft 60–80% der Projektzeit!

Die Vorbereitung der Daten ist der zeitaufwendigste Teil eines Data-Science-Projekts.

CRISP-DM: Phasen 4–6

4. Modeling
- Methode wählen
- Modell trainieren
- Parameter tunen
- Validieren

5. Evaluation
- Ergebnisse prüfen
- Geschaeftsziel erfuellt?
- Stakeholder-Review
- Entscheidung treffen

6. Deployment
- Produktivsetzung
- Monitoring einrichten
- Dokumentation
- Wartungsplan

Iteration: Der Prozess ist zyklisch – neue Erkenntnisse führen zu Anpassungen.

Nach dem Deployment beginnt oft ein neuer Zyklus mit verbesserten Daten und Modellen.

Daten als strategisches Asset

Wert von Daten

- Wettbewerbsvorteil durch Insights
- Automatisierung von Entscheidungen
- Personalisierung für Kunden
- Risikominimierung

Data Governance

- Qualitätsstandards definieren
- Zugriffsrechte regeln
- Compliance sicherstellen (DSGVO)
- Lebenszyklus verwalten

Herausforderungen:

1. Datensilos aufbrechen
2. Qualität sichern
3. Talent gewinnen
4. Kultur entwickeln
5. ROI nachweisen

Daten sind nur wertvoll, wenn sie zugaenglich, qualitativ hochwertig und nutzbar sind.

Managing Data Teams

Realistische Einschätzung

- Datenbereinigung: 40–60% der Zeit
- Feature Engineering: 20–30%
- Modellierung: 10–20%
- Deployment: 10–20%

Typische Unterschätzungen

- Datenqualitätsprobleme
- Infrastruktur-Setup
- Stakeholder-Kommunikation
- Iterationen und Debugging

Ihre Rolle als Manager:

1. Realistische Zeitplaene
2. Klare Prioritäten setzen
3. Ressourcen bereitstellen
4. Erwartungen managen
5. Erfolge kommunizieren

“Uebersetzer”-Funktion:
Zwischen Fachbereich und Data Scientists vermitteln.

Als Manager müssen Sie die Komplexität verstehen, ohne selbst zu programmieren.

R: Die Arbeitsumgebung

Warum R?

- Spezialisiert auf Statistik
- Exzellente Visualisierung
- Grosse Community
- Kostenlos und Open Source
- Reproduzierbare Analysen

RStudio IDE

- Integrierte Entwicklung
- Script-Editor
- Konsole
- Plots und Viewer

R-Projekt-Struktur:

- `projekt.Rproj` – Projektdatei
- `data/` – Rohdaten
- `scripts/` – R-Skripte
- `output/` – Ergebnisse
- `docs/` – Dokumentation

Vorteile:

- Relative Pfade
- Reproduzierbarkeit
- Versionskontrolle (Git)

R-Projekte sind der Schluessel zu reproduzierbarer Forschung und Zusammenarbeit.

Relative Pfade in R

Absolute Pfade (vermeiden!)

`C:/Users/Max/Dokumente/Projekt/data/sales.csv`

Probleme:

- Funktioniert nur auf einem PC
- Keine Zusammenarbeit möglich
- Pfade ändern sich

Best Practice:

1. Immer mit R-Projekt arbeiten
2. `here::here()` für robuste Pfade nutzen
3. Alle Daten im Projektordner speichern

Relative Pfade (richtig!)

`data/sales.csv`

Vorteile:

- Funktioniert ueberall
- Einfache Zusammenarbeit
- Versionskontrolle möglich

Relative Pfade sind die Grundlage für reproduzierbare und teilbare Analysen.

Das Tidyverse: Modernes R

Kernpakete

- `dplyr` – Datentransformation
- `ggplot2` – Visualisierung
- `tidyr` – Datenbereinigung
- `readr` – Datenimport
- `purrr` – Funktionales Programmieren

Philosophie

- Tidy Data: Jede Variable eine Spalte
- Pipe-Operator: %>%
- Konsistente Syntax
- Lesbare Code-Ketten

Typische Pipeline:

```
daten %>%
  filter(jahr >= 2020) %>%
  group_by(region) %>%
  summarize(
    umsatz = sum(sales)
  ) %>%
  arrange(desc(umsatz))
```

→ Filtere, gruppiere, aggregiere, sortiere

Das Tidyverse macht R-Code lesbar, wartbar und effizient.

dplyr: Die wichtigsten Verben

Zeilen bearbeiten

- `filter()` – Zeilen auswählen
- `arrange()` – Sortieren
- `distinct()` – Duplikate entfernen
- `slice()` – Nach Position

Spalten bearbeiten

- `select()` – Spalten auswählen
- `mutate()` – Neue Spalten
- `rename()` – Umbenennen
- `relocate()` – Umordnen

Gruppieren & Aggregieren

- `group_by()` – Gruppieren
- `summarize()` – Aggregieren
- `count()` – Zaehlen
- `ungroup()` – Gruppierung aufheben

Tabellen verbinden

- `left_join()` – Linker Join
- `inner_join()` – Innerer Join
- `bind_rows()` – Zeilen anfuegen

Mit diesen Verben loesen Sie 90% aller Datentransformationsaufgaben.

ggplot2: Grammatik der Grafiken

Grundprinzip
Grafiken aus Schichten aufbauen:

1. **Daten** – Was darstellen?
2. **Aesthetics** – Wie abbilden?
3. **Geometries** – Welche Form?
4. **Facets** – Aufteilen?
5. **Theme** – Design?

Beispiel:

```
ggplot(daten, aes(x, y)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~gruppe) +
  theme_minimal()
```

Häufige geoms:

- `geom_point()`, `geom_line()`
- `geom_bar()`, `geom_histogram()`
- `geom_boxplot()`, `geom_smooth()`

ggplot2 folgt einer konsistenten Grammatik – einmal gelernt, vielfach anwendbar.

Explorative Datenanalyse (EDA)

Ziele der EDA

- Daten verstehen und "fuehlen"
- Verteilungen erkunden
- Muster und Anomalien finden
- Hypothesen generieren
- Datenqualität prüfen

Typische Fragen

- Wie sind die Variablen verteilt?
- Gibt es Ausreißer?
- Welche Korrelationen existieren?
- Fehlen Werte? Wo?

EDA-Checkliste:

1. Dimensionen: `dim()`
2. Struktur: `str()`
3. Zusammenfassung: `summary()`
4. Fehlende Werte zaehlen
5. Histogramme aller Variablen
6. Scatterplots wichtiger Paare
7. Korrelationsmatrix

Nie modellieren ohne vorherige EDA – Sie müssen Ihre Daten kennen!

EDA: Visualisierungen

Univariat

- Histogramm
- Dichtekurve
- Boxplot
- Balkendiagramm

→ Eine Variable verstehen

Bivariat

- Scatterplot
- Liniendiagramm
- Gruppierte Boxplots
- Heatmap

→ Zusammenhaenge finden

Multivariat

- Faceted Plots
- Paarweise Scatterplots
- Korrelationsmatrix
- Parallel Coordinates

→ Komplexe Muster

Goldene Regel: Immer die richtige Visualisierung für die Fragestellung wählen!

Gute Visualisierungen kommunizieren Insights – schlechte verschleiern sie.

Einfache Lineare Regression: Konzept

Das Modell

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- Y – Abhängige Variable
- X – Unabhängige Variable
- β_0 – Achsenabschnitt (Intercept)
- β_1 – Steigung (Effekt von X)
- ε – Fehlerterm (Residuum)

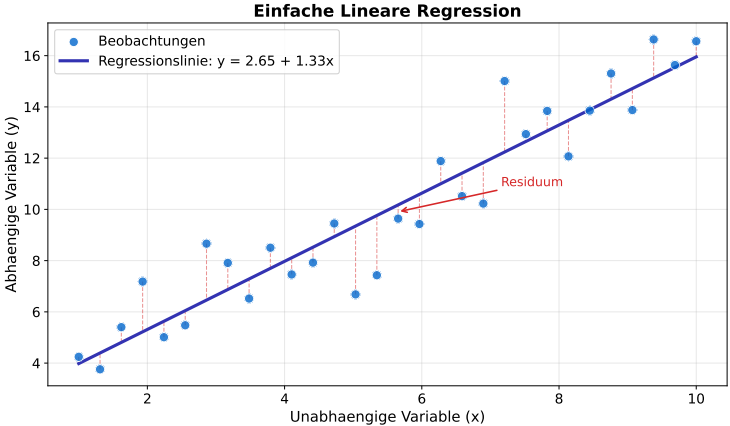
Interpretation
β_1 = Um wie viel aendert sich Y, wenn X um 1 Einheit steigt?

Beispiel:
Umsatz = 1000 + 50 $\times$ Werbung

$\rightarrow$ Jeder zusaetzliche CHF Werbung bringt 50 CHF Umsatz.

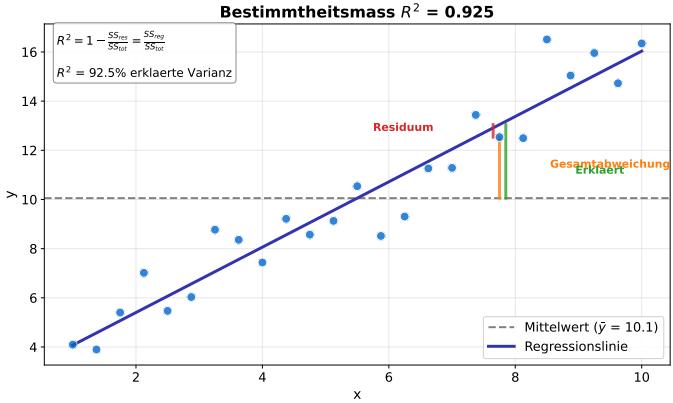
Die lineare Regression ist das Fundament vieler statistischer Methoden.

Einfache Lineare Regression: Visualisierung



Die Regressionslinie minimiert die Summe der quadrierten Residuen (OLS).

Bestimmtheitsmass R^2



R^2 misst den Anteil der Varianz in Y, der durch das Modell erklärt wird.

R^2: Interpretation und Grenzen

Was R^2 sagt

- Anteil erklärter Varianz
- Wert zwischen 0 und 1
- $R^2 = 0.7$: 70% erklärt
- Hoeher = bessere Anpassung

Richtwerte (kontextabhängig)

- Sozialwissenschaften: 0.3 gut
- Physik/Technik: 0.9+ erwartet
- Business: 0.2–0.5 typisch

Was R^2 NICHT sagt

- Kausalität
- Korrektheit des Modells
- Praktische Relevanz
- Out-of-Sample Performance

Warnung:
R^2 steigt immer mit mehr Variablen – auch bei irrelevanten!
$\rightarrow$ Adjusted R^2 nutzen

Ein hohes R^2 garantiert kein gutes Modell – und ein niedriges ist nicht immer schlecht.

Multiple Regression: Mehrere Prädiktoren

Erweitertes Modell

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \varepsilon$$

Warum mehrere Variablen?

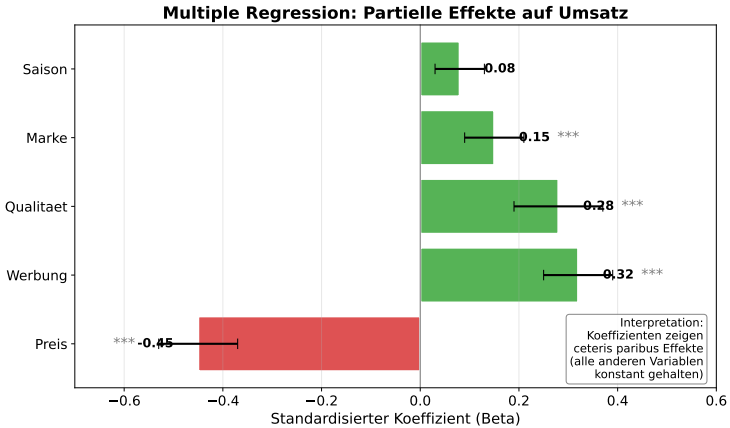
- Mehr Varianz erklären
- Stoervariablen kontrollieren
- Ceteris-paribus-Interpretation
- Realität ist multivariat

Beispiel:
Umsatz = β_0 + β_1 Preis + β_2 Werbung + β_3 Qualität + ε

Interpretation von β_1:
Effekt des Preises auf Umsatz, wenn Werbung und Qualität konstant gehalten werden.

Multiple Regression erlaubt "alles andere gleich"-Aussagen.

Multiple Regression: Partielle Effekte



Jeder Koeffizient zeigt den Effekt einer Variable, wenn alle anderen konstant sind.

Interaktionsterme

Idee
Der Effekt einer Variable hängt von einer anderen ab.

Modell mit Interaktion:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 (X_1 \times X_2) + \varepsilon$$

β_3 = Wie aendert sich der Effekt von X_1, wenn X_2 steigt?

Beispiel:
Umsatz = β_0 + β_1 Preis + β_2 Marke + β_3 (Preis $\times$ Marke) + ε

Interpretation:
Preiselastizität unterscheidet sich je nach Marke!

- Premium-Marke: geringere Preissensitivität
- Budget-Marke: hoehere Preissensitivität

Interaktionen erfassen, dass Effekte kontextabhängig sein können.

Regressionsoutput interpretieren

Wichtige Kennzahlen

- **Koeffizient**: Effektstaerke
- **Std. Error**: Unsicherheit
- **t-Wert**: Koeff. / SE
- **p-Wert**: Signifikanz
- R^2: Erklaerungskraft
- **F-Statistik**: Gesamtmodell

Typische Fragen

1. Ist der Effekt signifikant? ($p < 0.05$)
2. Wie gross ist der Effekt? (Koeff.)
3. Wie sicher sind wir? (SE, KI)
4. Erklärt das Modell viel? (R^2)

Warnung:
Signifikanz $\neq$ Relevanz!

Immer sowohl statistische Signifikanz als auch praktische Relevanz betrachten.

Overfitting: Das Problem

Definition
Das Modell lernt die Trainingsdaten zu gut – inklusive des Rauschens.

Symptome

- Sehr gute In-Sample-Performance
- Schlechte Out-of-Sample-Performance
- Komplexe, instabile Modelle
- Koeffizienten mit extremen Werten

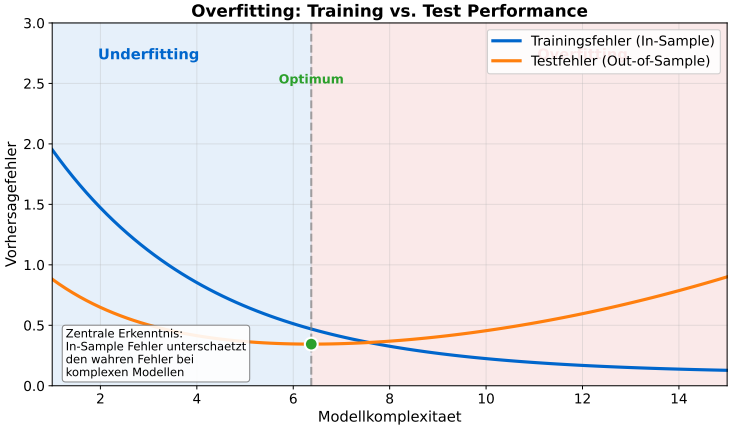
Ursachen

- Zu viele Variablen
- Zu komplexe Modelle
- Zu wenig Daten
- Keine Validierung

Kernproblem:
Wir wollen generalisieren, nicht memorieren!

Ein Modell, das Trainingsdaten perfekt erklärt, ist oft nutzlos für neue Daten.

Overfitting: Visualisierung



Das optimale Modell minimiert den Testfehler, nicht den Trainingsfehler.

In-Sample vs. Out-of-Sample

In-Sample

- Performance auf Trainingsdaten
- Immer optimistisch
- R^2 steigt mit Komplexität
- Nicht zur Modellwahl nutzen!

Problem:
Das Modell "kennt" die Daten bereits.

Out-of-Sample

- Performance auf neuen Daten
- Realistischere Schätzung
- Erfordert Datenteilung
- Basis für Modellwahl!

Methoden:

- Train/Test Split
- Kreuzvalidierung (Block 3)

Immer auf ungesehenen Daten evaluieren – sonst täuschen Sie sich selbst.

Block 1: Zusammenfassung

Konzepte

- Data Science = Statistik + CS + Domain
- CRISP-DM als Projektstandard
- Daten als strategisches Asset
- EDA vor Modellierung
- Regression als Grundmodell

Praktische Skills

- R-Projekte und relative Pfade
- Tidyverse für Datentransformation
- ggplot2 für Visualisierung
- lm() für Regression
- Koeffizienten interpretieren

Kernbotschaft:
Data Science ist modellbasiertes Denken mit dem Ziel der Generalisierung.

Block 2 behandelt Inferenz und Unsicherheit – wie sicher sind unsere Schätzungen?

Ausblick: Block 2

Naechste Themen

- Inferenz: Signal vs. Rauschen trennen
- Bootstrapping und Permutationstests
- Hypothesentests: t-Test, ANOVA
- Fehler 1. und 2. Art als Business-Risiken
- p-Hacking und Multiple Testing
- Effektstaerken statt nur p-Werte

Zentrale Frage:
Wie sicher können wir sein, dass unser Koeffizient nicht nur Zufall ist?

Statistische Inferenz quantifiziert Unsicherheit – essentiell für Geschaeftsentscheidungen.