Wirtschaftsinformatik – Informationsmanagement und Data Science

datasciencelab

HOCHSCHULE HEILBRONN

› Data Science Grundlagen
Data Preparation

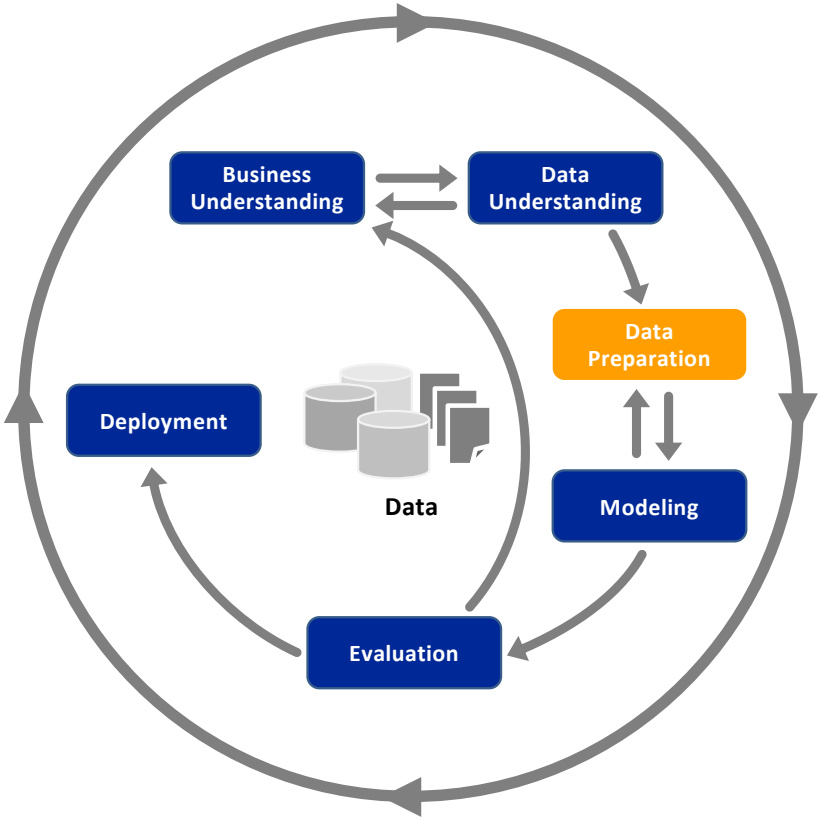
Prof. Dr. Carsten Lanquillon | Fakultät Wirtschaft und Verkehr | Wirtschaftsinformatik

CRISP-DM Phase 3: Data Preparation

Ziel: Aufbereitung der Daten (oft als Datenmatrix) für die Modellierungsphase

Aufgaben

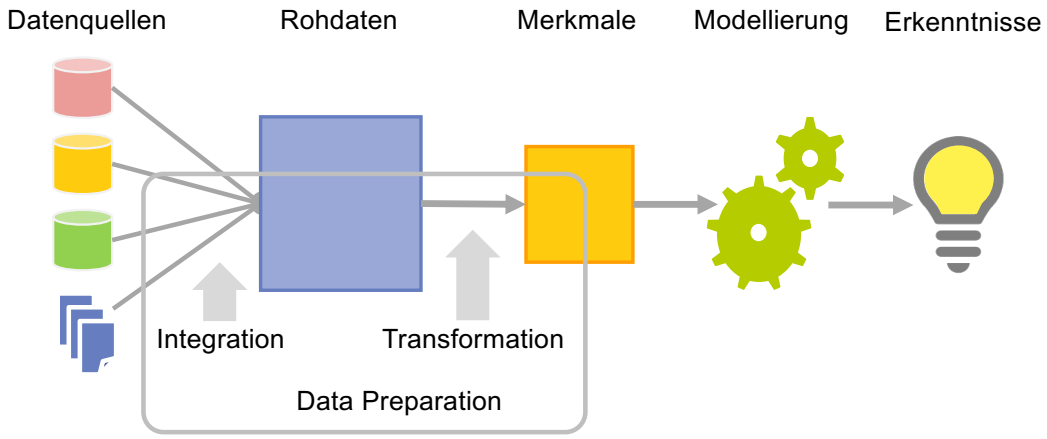
- Integration unterschiedlicher Datenquellen
- Datenbereinigung (Fehler, fehlende Werte)
- Formatierung, Codierung, Skalierung
- Auswahl relevanter Merkmale (Feature Selection)
- Konstruktion neuer Merkmale (Feature Engineering)



Data Preparation oder Data Preprocessing

Ziel: Erzeugung der Datenmatrix für die Modellierungsphase

- Bereinigung der Daten
  - Umgang mit fehlenden Werten
  - Umgang mit Ausreißern
  - Berücksichtigung Datentypen und Skalenniveaus
- Feature Selection (Dimensionsreduktion)
  - Filter-Ansätze
  - Wrapper-Ansätze
- Feature Engineering
  - Konstruktion neuer Merkmale
  - Dimensionsreduktion durch neue Merkmale



Datenbereinigung

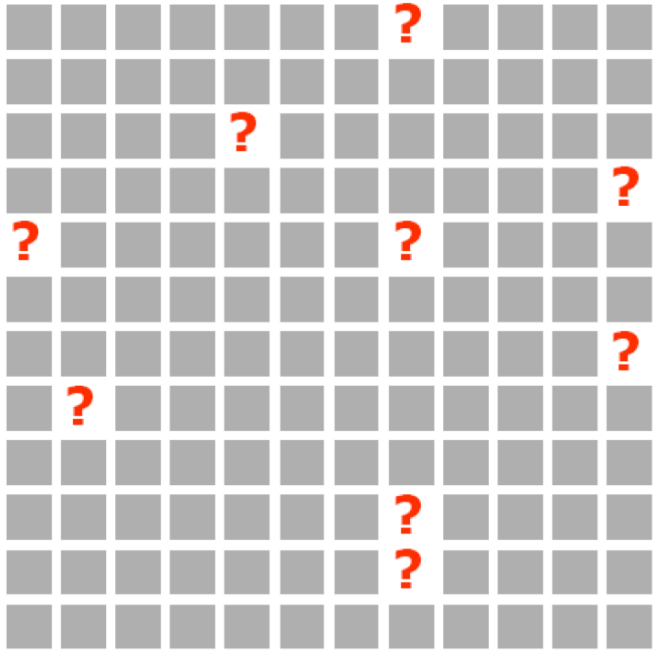
Ziel: Behebung von Datenqualitätsproblemen, korrekte Formatierung und richtiges Skalenniveau

- Erinnerung: Datenmatrix
  - Eine Zeile pro Objekt
  - Eine Spalte pro Merkmal
  - Atomare Werte (Ausprägungen)
- Behandlung fehlender Werte
- Behandlung Ausreißer
- Werte korrekt Formatiert
- Datentypen und Skalenniveaus anpassen

Merkmal 1	Merkmal 2	Merkmal 3	...	Merkmal m
$Wert_{11}$	$Wert_{12}$	$Wert_{13}$	...	$Wert_{1m}$
$Wert_{21}$	$Wert_{22}$	$Wert_{23}$	...	$Wert_{2m}$
...	...	...	...	...
$Wert_{n1}$	$Wert_{n2}$	$Wert_{n3}$	...	$Wert_{nm}$

Datenbereinigung: Umgang mit fehlenden Werte

- Fehlende Werte nicht verändern
- Zeilen mit zu vielen fehlenden Werten entfernen
- Spalten mit zu vielen fehlenden Werten entfernen
- Fehlende Werte sinnvoll ersetzen
  - Konstanter Wert
    - Vorgabe durch Experten
    - Als eigenen Wert eines nominalen Merkmals betrachten
  - Modus (für nominale oder rangskalierte Daten)
  - Mittelwert (nur für metrische Merkmale)
  - Vorgänger, Nachfolgewert, Mittelwert (nur für Zeitreihen)
  - Werte zufällig erzeugen



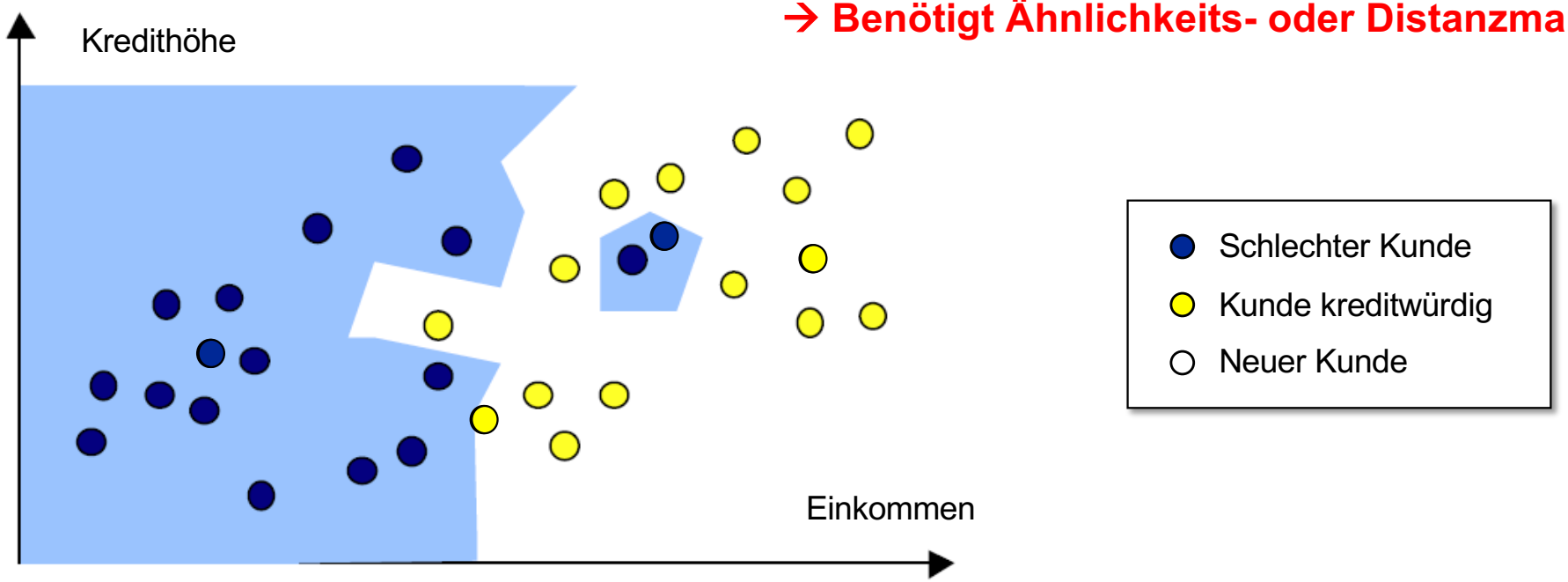
Abhängigkeit Datenaufbereitung und Modellierung

- Die erforderlichen Vorverarbeitungsschritte hängen teilweise vom Lernverfahren ab
- Es gibt nicht die eine richtige oder beste Art der Datenvorverarbeitung
- Mögliche relevante Fragen
  - Welche Skalenniveaus werden erwartet?
  - Kann das Lernverfahren mit fehlenden Werten umgehen?
  - Ist das Lernverfahren robust gegenüber Ausreißern?
  - Ist das Lernverfahren robust gegenüber Unterschieden in den Wertebereichen?
  - Ist das Lernverfahren robust gegenüber irrelevanten oder redundanten Merkmalen?

Beispiel: Klassifikation mit dem nächsten Nachbarn

Einfache Lösung: Ordne neue Objekte der Klasse zu, die der nächste Nachbar hat!

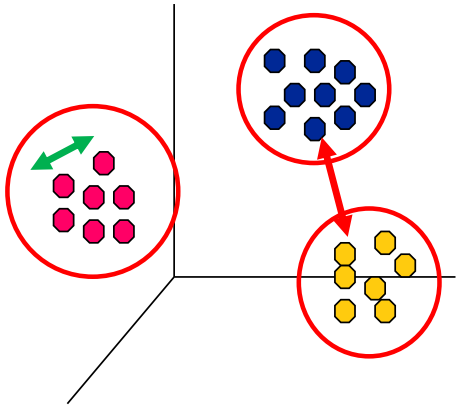
→ **Benötigt Ähnlichkeits- oder Distanzmaße!**



Beispiel: Clusteranalyse

Erzeuge Gruppen (Cluster, Klassen, Segmente)
von Objekten mit folgenden Eigenschaften:

- **“Within-Cluster Homogeneity”**
  Objekte in einer Gruppe sind ähnlich zueinander
- **“Between-Cluster Heterogeneity”**
  Objects in verschiedenen Gruppen sind unähnlich



Wan sind Objekte ähnlich?

→ **Benötigt Ähnlichkeits- oder Distanzmaße!**

Exkurs: Abstandsberechnung (1)

Zerlegung der Abstandsberechnung zwischen zwei Objekten in zwei Schritte:

1. **Merkmalsebene**
   - Definiere geeignetes Abstandsmaß für jedes Merkmal unter Berücksichtigung des Skalenniveaus
   - Stelle Vergleichbarkeit der Größenordnungen der resultierenden Abstandswerte aller Merkmale sicher
2. **Objektebene**
   - Aggregation der Abstandswerte auf Merkmalsebene zu einem kombinierten Abstandswert

Exkurs: Abstandsberechnung – Merkmalsebene

- Der Abstand bei **nominalen Merkmalen** ist 0, wenn die Werte gleich sind, sonst 1
  - → $d_{nominal}(x,y) \in \{0,1\}$
- Der Abstand zwischen zwei Ausprägungen eines **ordinalen Merkmals** sollte proportional zur Anzahl der dazwischenliegenden Ränge sein
  - → $0 \leq d_{ordinal}(x,y) \leq k-1$ mit Anzahl k der verschiedenen Ausprägungen
- Bei **metrischen Merkmalen** ist der Abstand bereits inhärent als absolute Differenz definiert
  - → $d_{metric}(x,y) = |x-y| \in [0,\infty)$

→ **Beobachtung: Sehr unterschiedliche Größenordnungen bei den Wertebereiche möglich!**

Exkurs: Abstandsberechnung – Wertebereiche

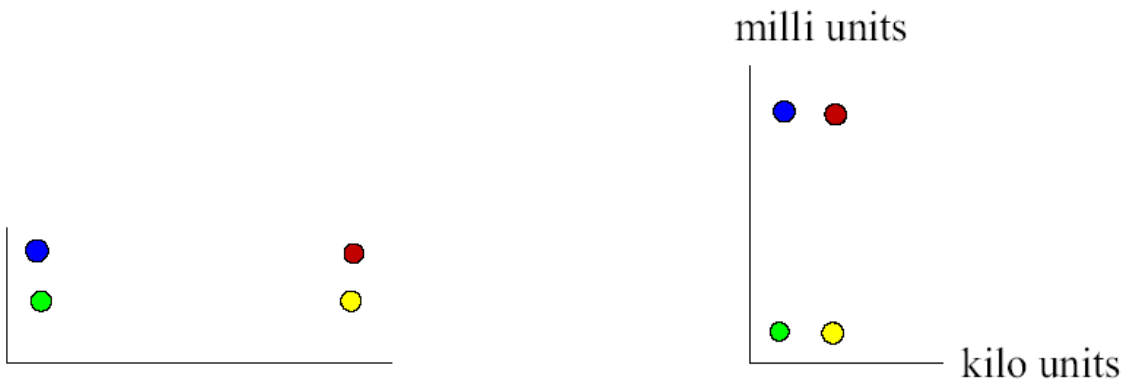
Beispiel: Körpergröße und Gewicht eines Patienten

- Fall 1
  - Körpergröße in **cm**: Werte zwischen 100 und 200
  - Gewicht in **kg**: Werte zwischen 30 und 130
  - → Ergibt vergleichbare Größenordnungen bei beiden Abstandswerten
- Fall 2
  - Körpergröße in **m**: Werte zwischen 1 und 2
  - Gewicht in **g**: Werte zwischen 30.000 und 130.000
  - → Ergibt stark unterschiedliche Größenordnungen bei beiden Abstandswerten

→ **Beobachtung: Größenordnungen hängen von den verwendeten Einheiten ab!**

Exkurs: Abstandsberechnung – Skalierungseffekte

→ Unterschiedliche Ergebnisse in Abhängigkeit der Skalierung



→ Durch **Normalisierung** oder **Standardisierung** kann eine Vergleichbarkeit der Wertebereiche bei Abstandsberechnung erreicht werden!

Exkurs: Abstandsberechnung – Min-Max-Normalisierung

- Lineare Transformation der ursprünglichen Werte in einen vorgegebenen Bereich (meist das Einheitsintervall [0,1])
- Es seien a und b die untere und obere Grenze des ursprünglichen Wertebereichs:

$$x^{new} = \frac{x - a}{b - a}$$

- Verwendung von natürlichen Grenzen oder beobachtetes Minimum und Maximum
- **Achtung:**
  - Ausreißer verzerren der Ergebnis stark: Die meisten Objekte sind ähnlich!
  - Zukünftige Werte können außerhalb der bekannten Grenzen liegen!

Exkurs: Abstandsberechnung – Standardisierung

- Transformation der ursprünglichen Werte, so dass die neuen Werte Mittelwert 0 und Standardabweichung 1 haben:

$$x^{new} = \frac{x - \bar{x}}{s}$$

- Mit empirischem arithmetischen Mittelwert und empirischer Varianz:

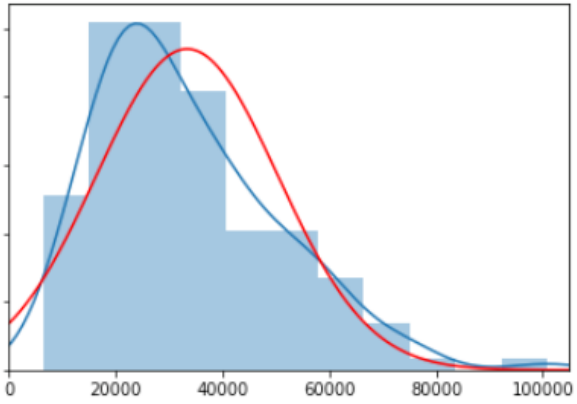
$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \qquad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

- Normalverteilte Merkmale sind danach standardnormalverteilt!

Exkurs: Abstandsberechnung – log-Normalisierung

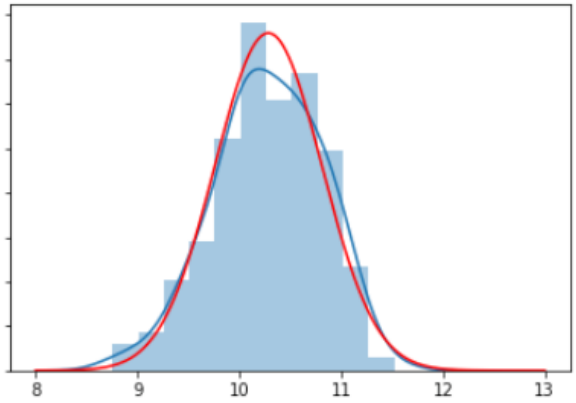
- Transformation der ursprünglichen Werte durch Logarithmierung
- Meist Verwendung des natürlichen Logarithmus (Basis e)
- Berücksichtigt relative Veränderungen (Größenordnung der Änderung)
- Weitere Eigenschaft: Neuer Wertebereich ist positiv

x	x^{neu}
10	2,3
100	4,6
1.000	6,9
10.000	9,2



Laufleistung von Fahrzeugen

log



Logarithmierte Laufleistung

Exkurs: Abstandsberechnung – Objektebene (1)

> **Ziel:** Quantifizierung des Abstands $d(\mathbf{x},\mathbf{y})$ zwischen den Objekten **x** und **y** mit $\mathbf{x} = (x_1,...,x_m)$ und $\mathbf{y} = (y_1,...,y_m)$ mit m Merkmalswerten

> Anforderungen an die Abstandsfunktion (Metrik) d:

- Nicht-Negativität $d(\mathbf{x},\mathbf{y}) \geq 0$
- Identität $d(\mathbf{x},\mathbf{x}) = 0$
- Symmetrie $d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x})$
- Dreiecksungleichung $d(\mathbf{x},\mathbf{y}) \leq d(\mathbf{x},\mathbf{z}) + d(\mathbf{z},\mathbf{y})$

Exkurs: Abstandsberechnung – Objektebene (2)

> Aggregation der Abstände auf Merkmalsebene d_i zwischen den Objekten **x** und **y**

$$d(\mathbf{x},\mathbf{y}) = \sqrt[p]{\sum_i d_i(x_i, y_i)^p}$$

> Typische Werte für den Parameter *p*
  - *p = 1* → Manhatten-Abstand (city block metric)
  - *p = 2* → Euklidischer Abstand
  - *p = ∞* → Maxium (Supremum) Norm

Exkurs: Abstandsberechnung – Zusammenfassung

- Auswahl problemadäquate Abstandsfunktion oder Ähnlichkeitsfunktion
- Viele Werkzeuge erwarten einheitliche Skalenniveaus (meist ausschließlich numerisch)
- Anpassung Skalenniveaus
  → Feature Encoding
- Normalisierung oder Standardisierung zum Angleichen der Skalen
- Entfernung irrelevanter und redundanter Merkmale verbessert die Abstandsberechnung
  → Dimensionsreduktion

Feature Encoding – Repräsentation

Viele Lernverfahren oder Werkzeuge erwarten spezielle Format oder Skalenniveaus

> Nominal zu metrisch

Achtung: Die einfache ganzzahlige Codierung ohne Berücksichtigung der Zusammenhänge ist sehr problematisch, wenn Entfernungen zwischen Objekten berechnet werden!

Stattdessen:

– One-Hot-Encoding (Dummy-Encoding):
  → Ein binäres (0/1) Merkmal für jede Ausprägung
– Representation Learning, Feature Embeddings

Farbe		rot	gelb	grün
rot	→	1	0	0
gelb		0	1	0
grün		0	0	1

> Metrisch zu nominal bzw. ordinal

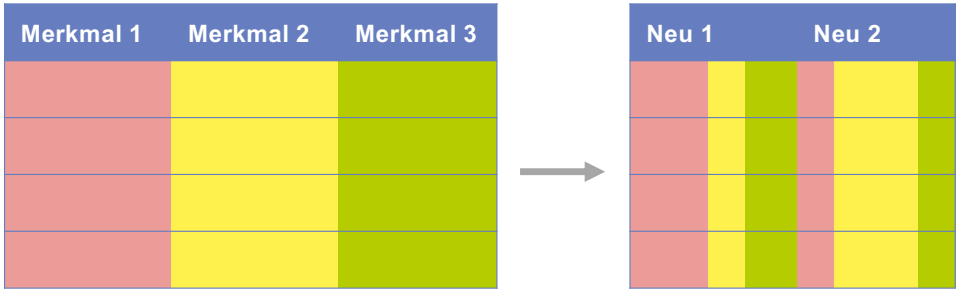
– Binning (Klassierung)
– Zusammenfassung durch Aggregationsfunktionen

Dimensionsreduktion
Feature Selection vs. Feature Engineering

- Feature Selection
  - → Dimensionsreduktion durch Auswahl bestehender Merkmale
- Feature Engineering
  - → Konstruktion neuer Merkmale
  - – Geeignetere Repräsentation
    → *Feature Encoding*
  - – Erzeugung neuer Merkmale, die Zusammenhänge besser erfassen
  - – Dimensionsreduktion
    → *Feature Extraction*

Dimensionsreduktion durch Feature Selection

Dimensionsreduktion durch Feature Extraction



Warum Dimensionsreduktion?

- Datenmatrix
  - ist weniger komplex
  - benötigt weniger Speicherplatz
  - benötigt weniger Rechenleistung für die Verarbeitung
- Vermeidung des “Curse of Dimensionality“
  - Abstandsberechnungen in hochdimensionalen Räumen schwierig
  - Geringere Gefahr für Overfitting bei der Modellierung
- Visualisierung der wichtigsten Zusammenhänge

Feature Selection (Merkmalsauswahl)

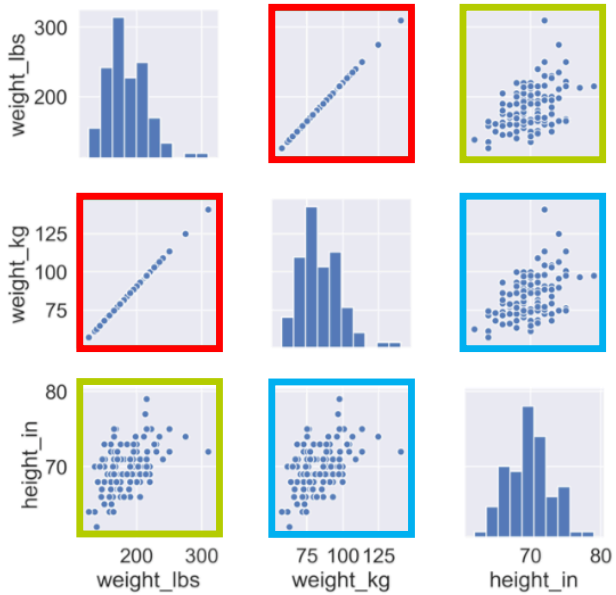
Ziel: Finde eine für das Analyseziel geeignete Teilmenge relevanter Merkmale

- Filter-Ansätze
  - Entfernung von Merkmalen mit vielen fehlenden Werten
  - Entfernung redundanter Merkmale (doppelte Merkmale, Zusammenhang untereinander)
  - Selektiere die besten *k* Merkmale nach vorgegebenem Bewertungskriterium

    → Bei überwachtem Lernen: Stärke des Zusammenhangs mit der Zielgröße!
- Wrapper-Ansätze in Verbindung mit einem Lernverfahren
  - Schrittweises Entfernen (stepwise backward elimination)
  - Schrittweises Hinzufügen (stepwise forward selection)

Feature Selection: Beispiel redundante Merkmale

Scatter Matrix (Matrix aus Streudiagrammen) und Korrelationsmatrix



	weight_lbs	weight_kg	height_in
weight_lbs	1.00	1.00	0.47
weight_kg	1.00	1.00	0.47
height_in	0.47	0.47	1.00

Feature Extraction – Dimensionsreduktion

Ziel: Bilde die gegebenen m Merkmale auf eine kleinere Anzahl an Merkmalen ab

- **Beispiel:** Hauptachsentransformation – Principal Component Analysis (PCA)
  - Zerlegt Eingaberaum in orthogonale Komponenten, die jeweils so viel Varianz wie möglich erklären
  - Dimensionsreduktion erfolgt durch Beschränkung auf die ersten k Komponenten ($k < m$)
- Nachteile
  - Neue Darstellung schwierig zu interpretieren
  - Nur für lineare Zusammenhänge geeignet
  - Andere Verfahren für nicht-lineare Zusammenhänge

Data Preparation – Zusammenfassung

- Vorbereitung der Datenmatrix für die Modellierungsphase
- Gute Datenaufbereitung essentiell für das erfolgreiche Lernen aus Daten
- Erfordert Fach- und Analyseexpertise
- Sehr zeitaufwendiger, meist (noch) manueller Prozess

- Ausnahmen für spezielle Anwendungen wie etwa Bilderkennung
  - Automatische Merkmalserzeugung möglich (*Deep Learning* bzw. *Representation Learning*)