

Machine Learning mit Python

EXXETA
CONSULTING AND TECHNOLOGIES

Schulung der GfU

Timm Gieger

12.-14. Juli 2021





Timm Gieger
Project Assistant

Biographie

- Project Assistant @ EXXETA, Mannheim (aktuell)
- Dozent für das Fach Data Science, Hochschule Ludwigshafen (aktuell)
- M. Sc. Data Science & Consulting, Hochschule Ludwigshafen (aktuell)
- B. Sc. Wirtschaftsinformatik, Hochschule Ludwigshafen

Beratungskompetenz

- Python
- Data Science & Deep Learning:Keras, Tensorflow, Numpy, Jupyter Notebook, Pandas, ScikitLearn
- SAP BW, SAP AC, SAP DWC

Sprachen

- Deutsch
- Englisch

Lernziele

- Prozessschritte zur Entwicklung von Predictive Applications in unterschiedlichen Geschäftskontexten wiedergeben und anwenden
- Grundlegende Konzepte und Methoden aus dem Bereich Data Science und Predictive Analytics
- Anwendungsfallspezifische Auswahl und Anwendung der passenden Methoden und Algorithmen der explorativen Datenanalyse und des maschinellen Lernens
- Modelle des maschinellen Lernens wiedergeben und ggf. Erstellung und Optimierung mit Toolunterstützung (Python)
- Moderne Werkzeuge aus dem Bereich Data Science bedienen und darin Predictive Applications umsetzen (Libraries)



1. Tag

- Begriff Data Science & Machine Learning
- Vorgehensmodelle und Beispielcases
- Einführung Python/Pandas/Numpy
- Lesen und schreiben von Daten aus unterschiedlichen Formaten (SQL,CSV...)
- Explorative Datenanalyse
- Pre-Processing

2. Tag

- Algorithmen & Konzepte Machine Learning
- Einführung scikit-learn
- Regression
- Klassifikation
- Clustering
- Evaluierung
- Hyperparameteroptimierung

AGENDA



AGENDA

3. Tag

- Ausstehende Themen von Tag2
- Anomaly Detection
- REST via Python
- NLP (Tokenizer, Bag of Words) → falls möglich
- Deployment (speichern und laden von Modellen)
- Fragen & Feedback



Vorstellungsrunde

- Persönliche Vorstellung (Name, Firma, etc...)
- Haben Sie bereits Erfahrungen mit Data Science/Machine Learning gemacht?
- Was erwarten Sie sich von dieser Schulung und welche Themen interessieren Sie besonders?
- Wie würden Sie Ihre Kenntnisse mit Python/SQL einschätzen? (Kenner, Könner, Experte)
- Hatten Sie bereits eine Schulung in diesem Gebiet?



Themengebiete

Supervised Learning
(Classification and Regression)

Unsupervised Learning (Clustering)

Model
Evaluation
and
Optimization

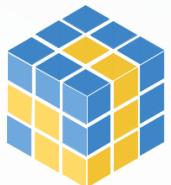
Data Pre-Processing

Explorative Data Analyses

Foundations of Data Science and Machine Learning



Toolkit



NumPy

matplotlib



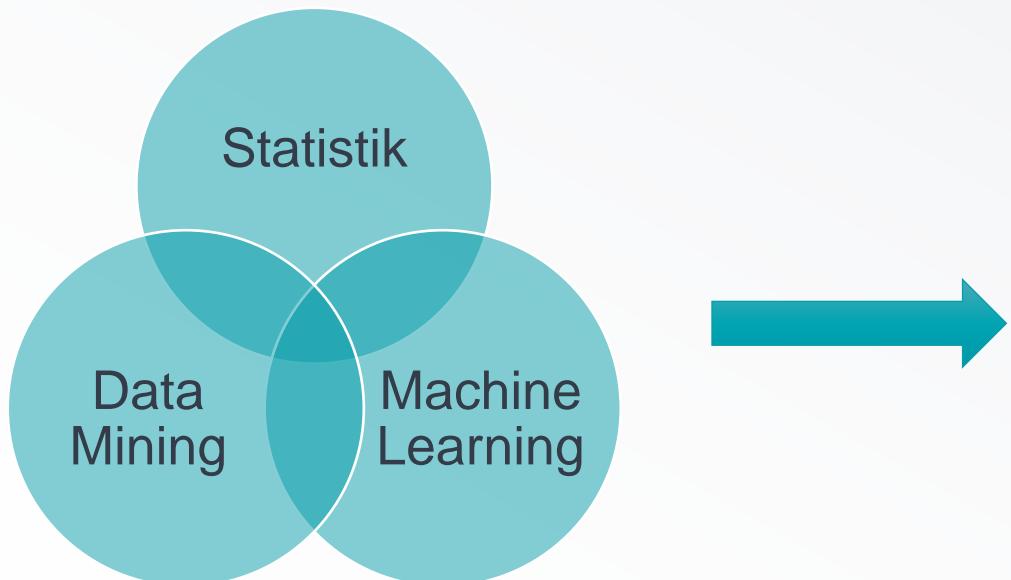
SQLAlchemy



Was ist Data Science?

Data Science im Rahmen der Schulung

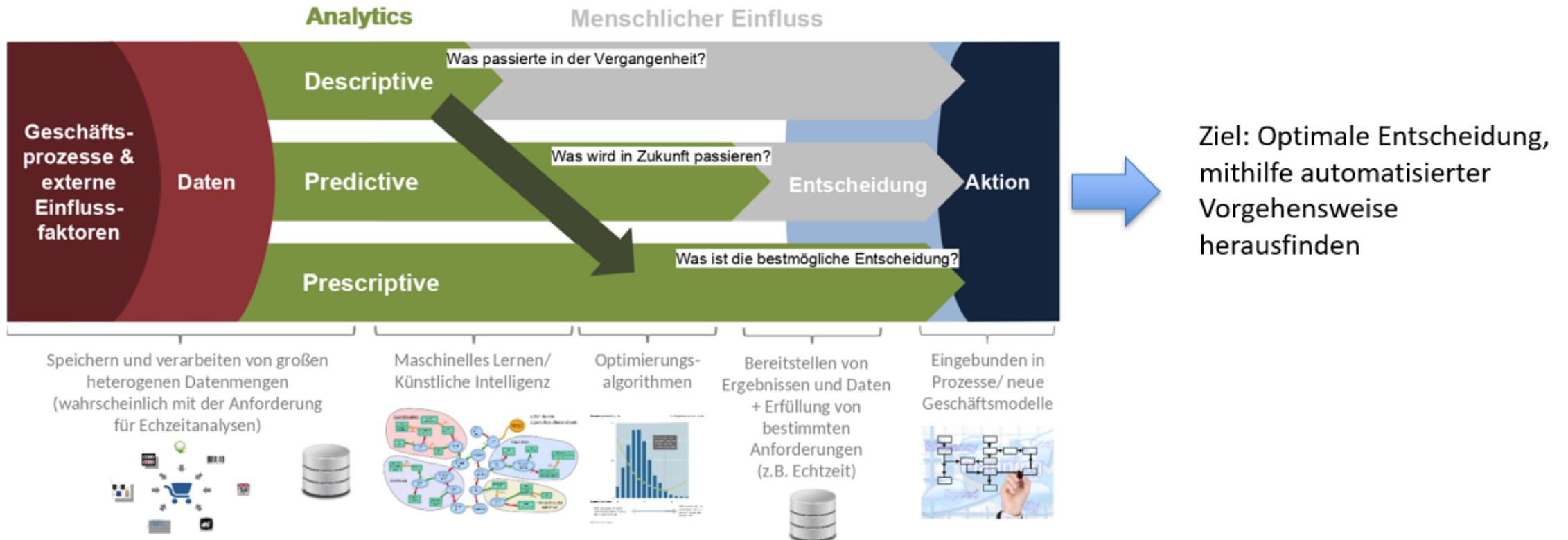
„...Extraktion nützlichen Wissens und aussagekräftiger Informationen aus großen Datenmengen, um geschäftliche Entscheidungsfindungen zu verbessern.“¹



Hauptgebiete von Data Science, welche nicht eindeutig voneinander separierbar sind und sich teils überlappen.



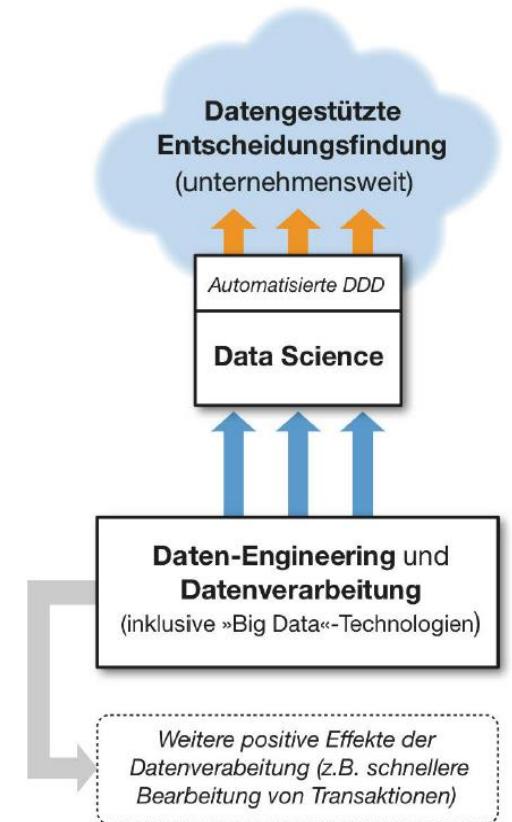
Arten von Analytics und deren Ziel



Data-Driven Decision-Making (DDD)

Arten von Entscheidungen, die von Interesse sind:

1. Entscheidungen, welche auf Muster in den Daten zurückzuführen sind.
 - Beispiel: Bevorratung eines bestimmten Artikels vor einem signifikanten Ereignis, welches das Verkaufsvolumen beeinflusst (Naturkatastrophe etc.)
2. Sich wiederholende Entscheidungen, welche durch Datenanalysen exakter werden.
 - schon kleine Verbesserungen können ausschlaggebend sein
 - Beispiel: Kundenabwanderung verhindern, indem durch Datenanalyse die am meisten profitablen Kundenprofile gefunden werden können. (Telekommunikationsanbieter).



Basis für DDD = Data Science



Automatisierte Entscheidungsfindung

Anwendungen

- Fraud Detection
- Banken- und Verbraucherkreditbranche
- Telekommunikationsdienstleistung
- Anzeigenwerbung

...

Fallen Ihnen noch weitere Gebiete ein?



Data Science als „Strategisches Gut“

„Daten und die Fähigkeit, ihnen nützliches Wissen zu entnehmen, sollten als wichtiges strategisches Gut betrachtet werden.“¹

Fragestellungen:

- Ist eine ausreichende Datengrundlage vorhanden um bestmögliche Entscheidung zu finden?
- Gibt es das benötigte Talent/Fähigkeit/Skill im Unternehmen um Daten auszuwerten überhaupt? (Data Science Team)
- Wie viel kann/muss investiert werden, damit Kosten/Nutzen Verhältnis stimmt?



Sowohl Investition in Daten als auch Fähigkeiten muss gewährleistet sein um Wettbewerbsvorteil zu etablieren



Data Science als „Strategisches Gut“

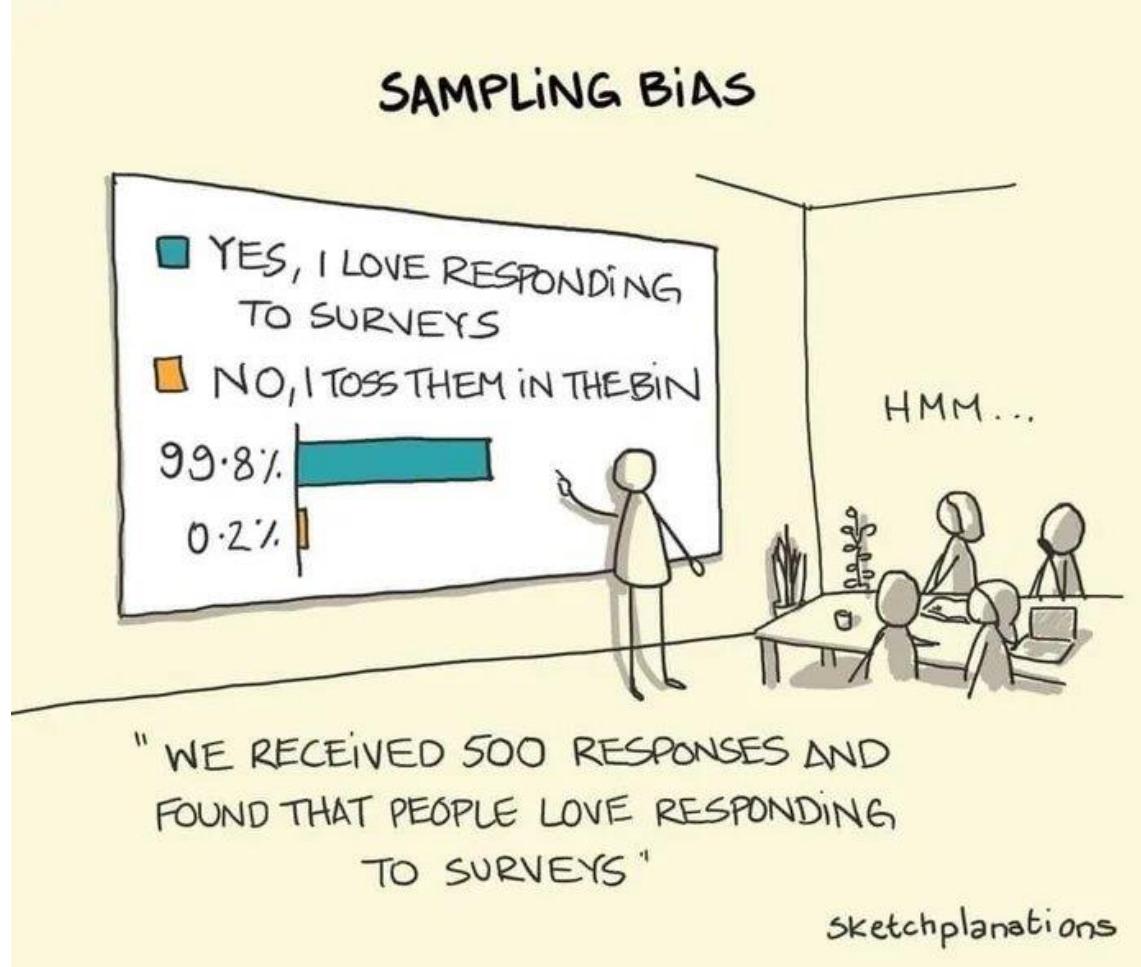
Herausforderungen:

- Daten müssen repräsentativ sein (Sampling-Bias)
- Hoher Initialaufwand
- Hohe Kosten
- Zeitaufwendig
- Nutzen schwer zu schätzen, da Daten erst nach Sammlung benutzt werden können und dann evtl. schon an Wert verloren haben

Welche Art von Geschäftsmodell fällt Ihnen spontan ein bei dem Daten und speziell deren Fähigkeit diese auszuwerten einen erheblichen Wettbewerbsvorteil darstellen?



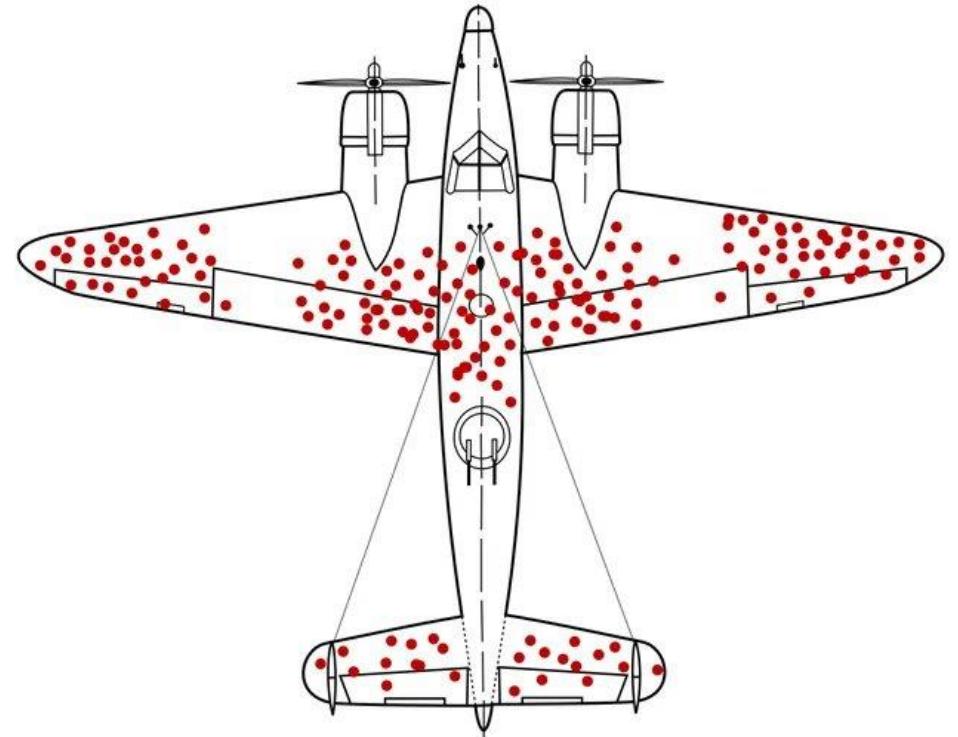
Sampling Bias Beispiele



Sampling Bias Beispiele

Statistische Visualisierung von Einschusslöchern
heimgekehrter British Army Bomber aus WWII um
herauszufinden wo die Panzerung von Flugzeugen
verstärkt werden müsste.

Was wäre Ihre Lösung?



Anwendungsbeispiel Joghurt

- Können wir so etwas verhindern?



Anwendungsbeispiel Joghurt

Machine Learning für Bestandsoptimierung

Einflussfaktoren für die Nachfrage

Problemstellung: Vorhersage der Nachfrage so genau wie möglich



Anwendungsbeispiel Joghurt

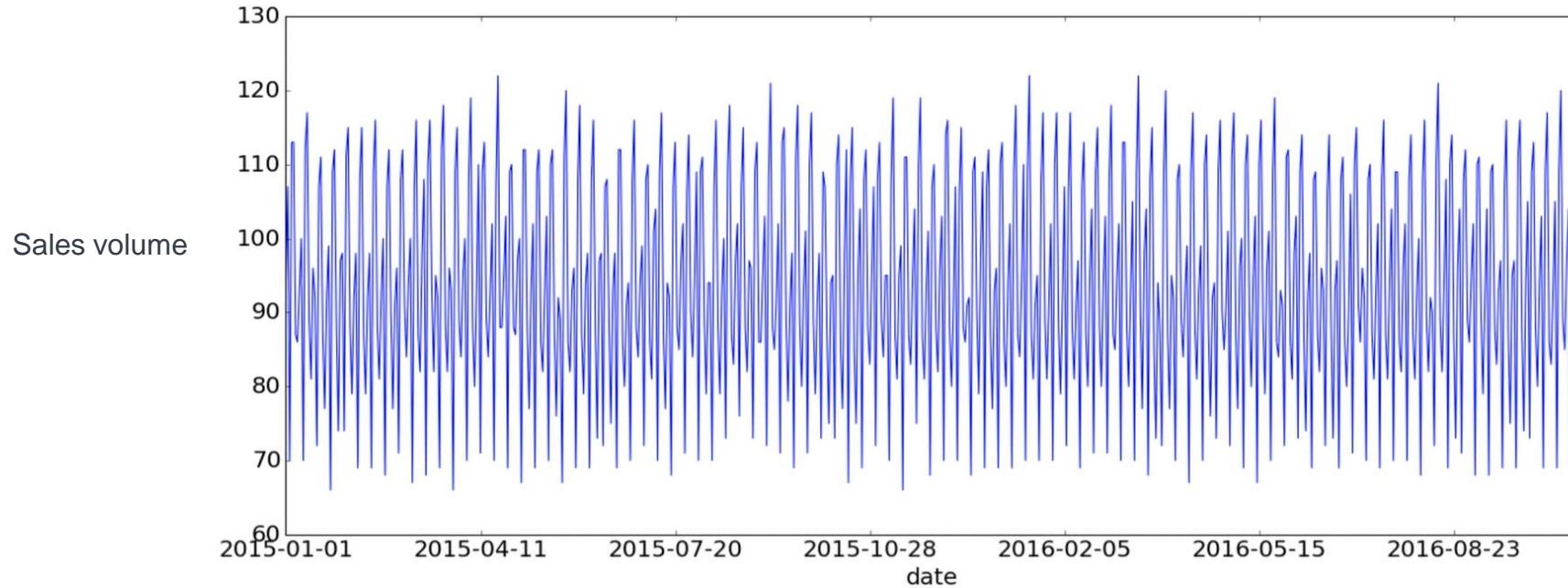
Machine Learning für Bestandsoptimierung



Anwendungsbeispiel Joghurt

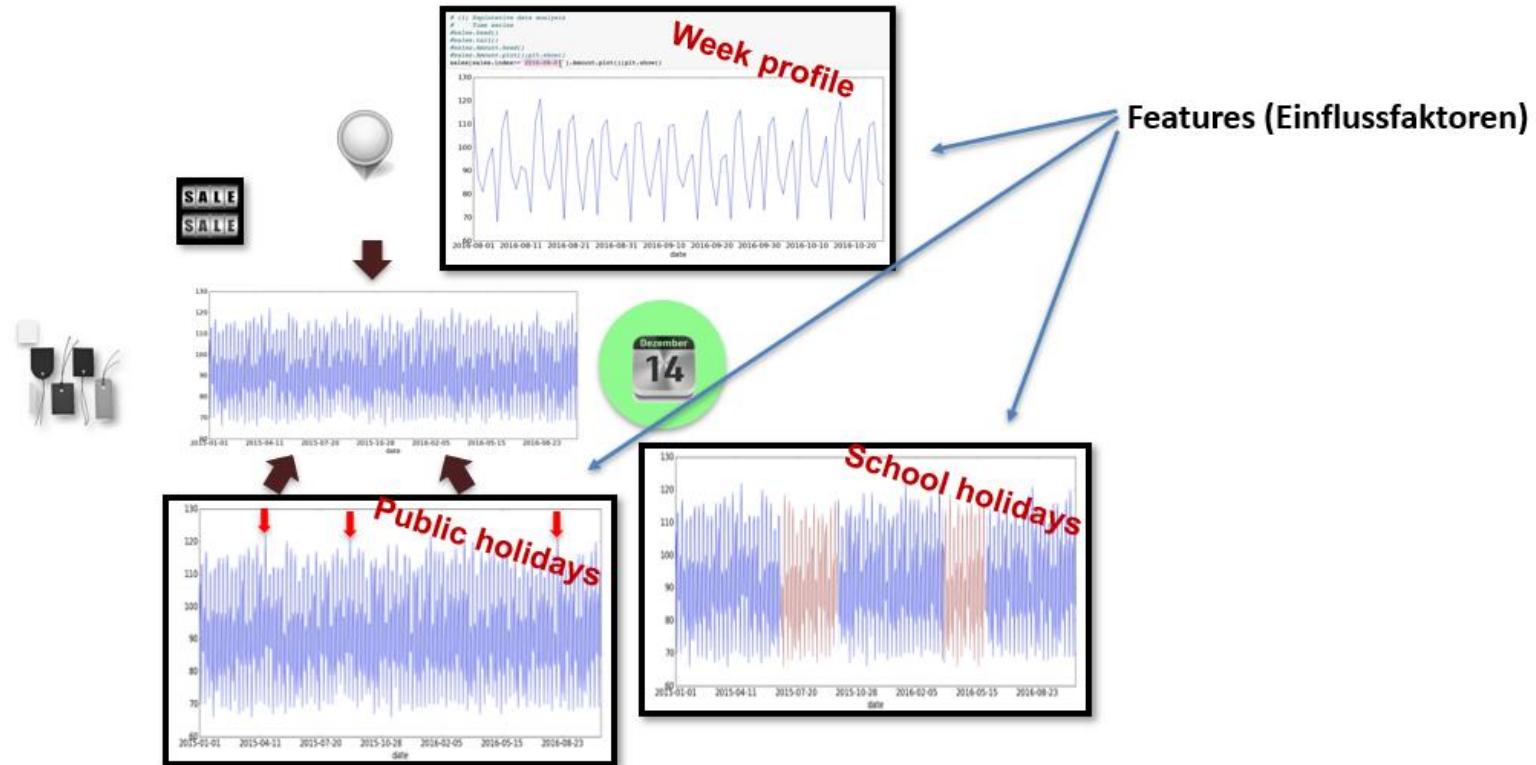
Machine Learning für Bestandsoptimierung

Herangehensweise: Lernen aus historischen Daten



Anwendungsbeispiel Joghurt

Machine Learning für Bestandsoptimierung

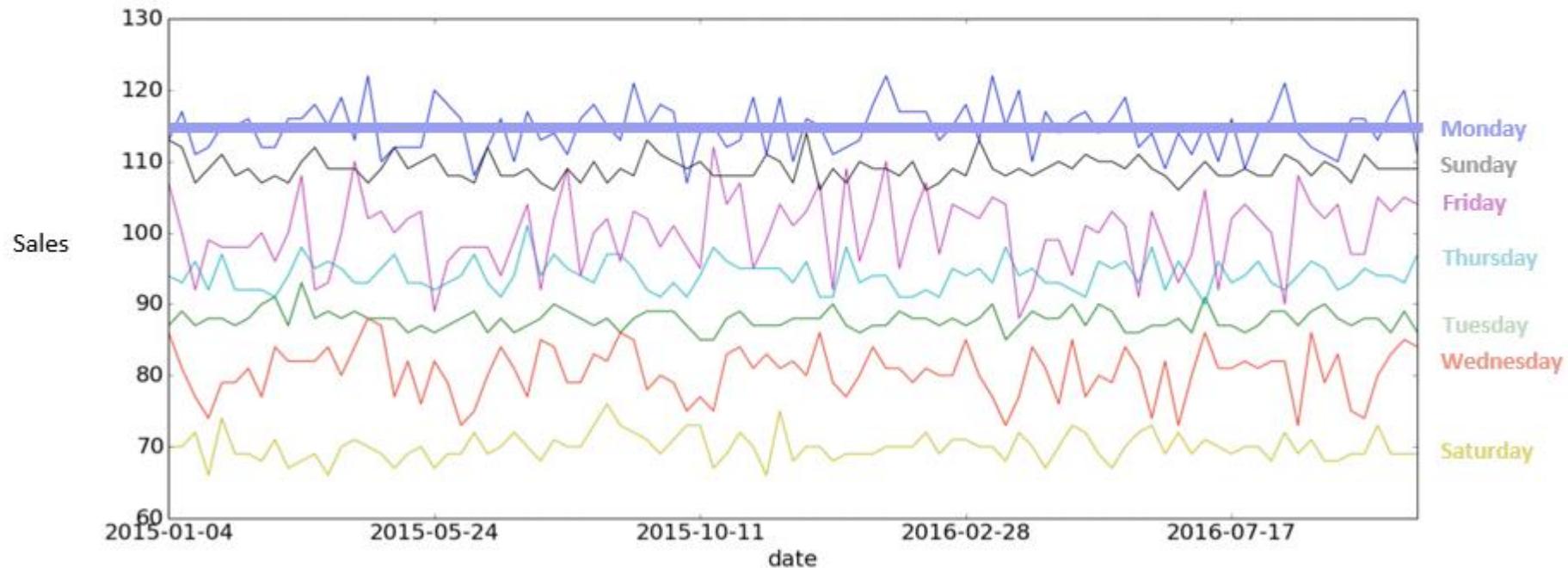


Anwendungsbeispiel Joghurt

Machine Learning für Bestandsoptimierung

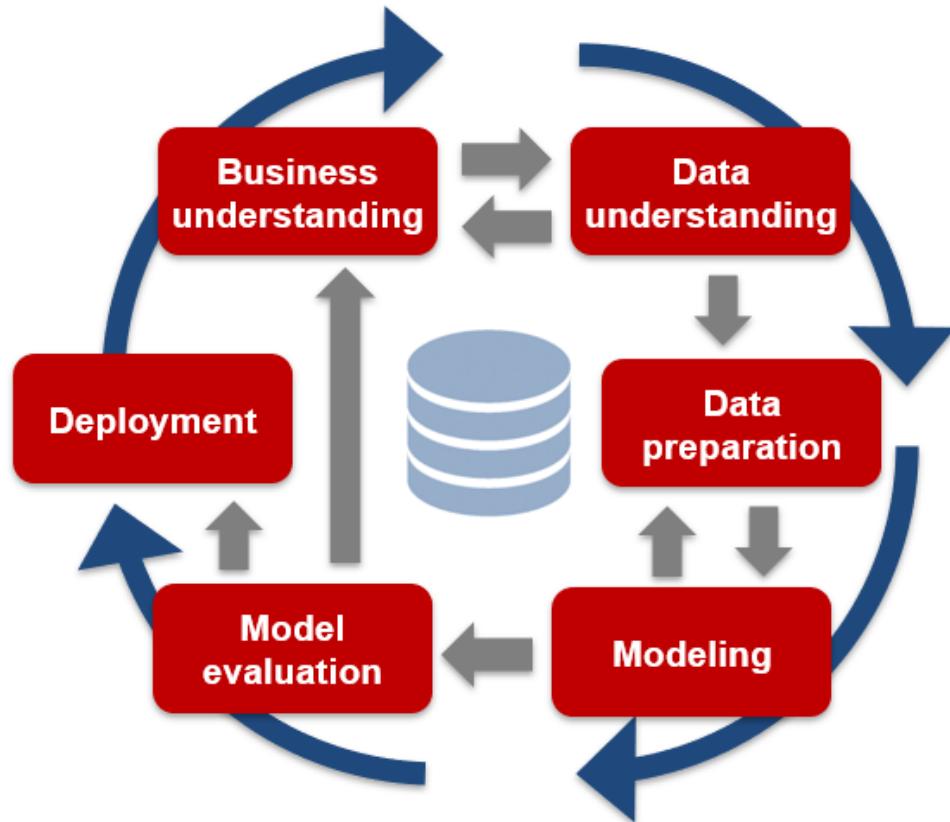
Wochenprofil (Wochensaisonalität): Die meisten Joghurts werden Montags gekauft.

Entscheidung: Mehr Joghurts für Montags bestellen! (DDD)



Cross-Industry Standard Process for Data Mining (CRISP-DM)

Wie ist die Vorgehensweise bei einer solchen Problemstellung?



Live Demo

Joghurtbeispiel

Cross-Industry Standard Process for Data Mining (CRISP-DM)

Aufgabenverständnis

- Verständnis über Aufgabenstellung erlangen
- Initiale Problemformulierung womöglich nicht optimal oder unvollständig (mehrere Iterationen nötig)

Datenverständnis

- Relevante Daten extrahieren → oftmals wurden Daten nicht direkt für eigentliche Problemstellung gesammelt Bsp.: Vertriebsdaten, Transaktionsdaten, Stammdaten etc.
- Kosten für Daten variiert → manche kostenlos, manche müssen gekauft oder erhoben werden
- Explorative Datenanalyse: Wie sind die Daten verteilt?
- Evtl. visuelle/graphische Aufbereitung der Daten zum besseren Verständnis



Cross-Industry Standard Process for Data Mining (CRISP-DM)

Datenaufbereitung

- Data Quality – Probleme (Fehlwerte, Noise, Outliers...)
- Umwandlung von Datenformaten
- Normalisierung/Standardisierung
- Feature Extraction

Modellbildung

- Anwenden von Machine-Learning Algorithmen auf Daten
- Hyperparameter-Tuning (Grid-Search)
- Regularisierung



Cross-Industry Standard Process for Data Mining (CRISP-DM)

Beurteilung/Evaluierung

- Ergebnisse des Modells bewerten (Test-/Validation- Set)
- Cross-Validation
- Evaluierung anhand spezifischer Metriken Bsp.: „Root-Mean-Squared-Error“ , „Accuracy“
- Evaluierungsverfahren wie beispielsweise: „Confusion-Matrix“, „ROC-Analyse“
- Overfitting/Underfitting erkennen

→ Problemstellung erfüllt?

Deployment

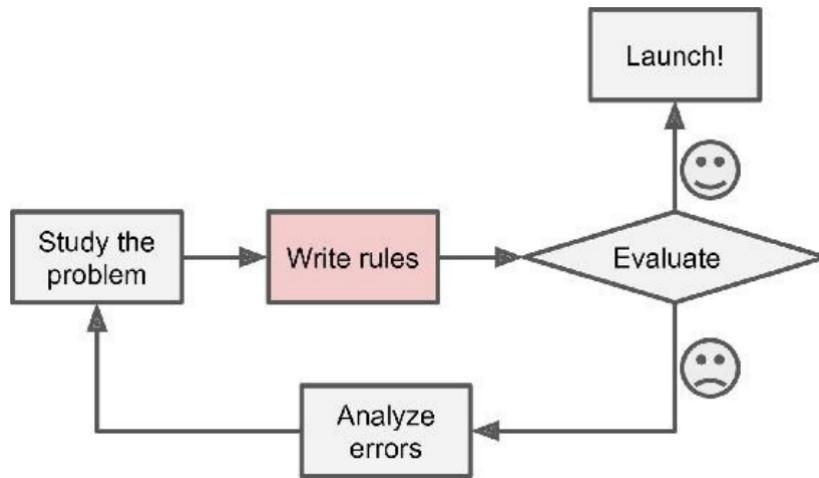
- Überführung des Modells in die Produktivumgebung
- Grundlegende Themen des Software-Engineerings wie beispielsweise: Kompatibilität, Skalierung, Wartung etc. kommen zum tragen.



Warum Machine Learning?

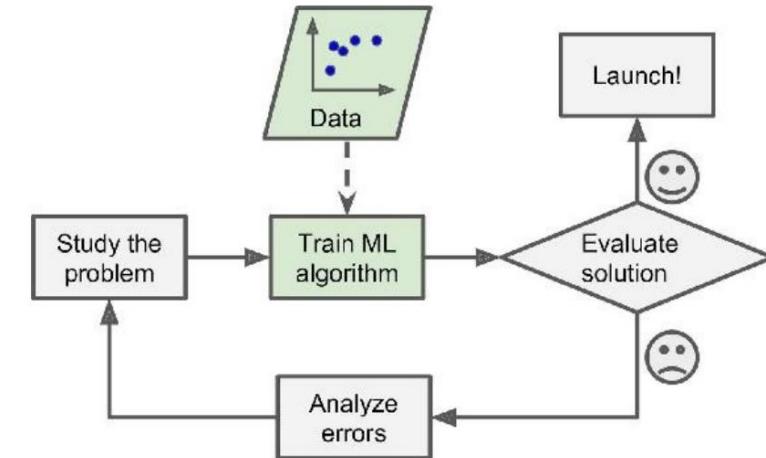
Motivation – traditioneller Ansatz vs. Lernen aus Daten

- Beispiel: Spam Filter



Traditioneller Ansatz

- komplex, hard-codiert
- Schwer zu warten



Machine Learning Ansatz

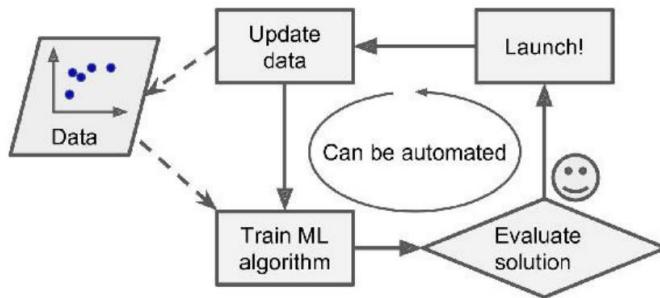
- Automatisches lernen aus Daten
- Automatisches re-trainieren



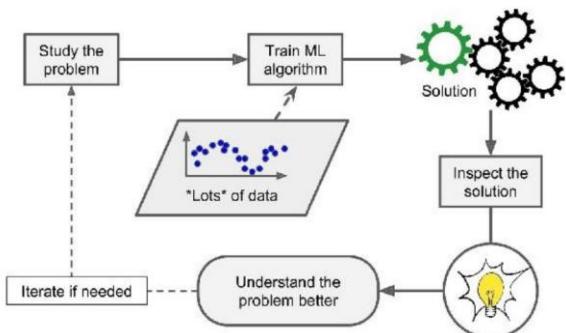
Warum Machine Learning?

Motivation – traditioneller Ansatz vs. Lernen aus Daten

- Machine Learning kann automatisiert werden – **flexibel für Änderungen!**

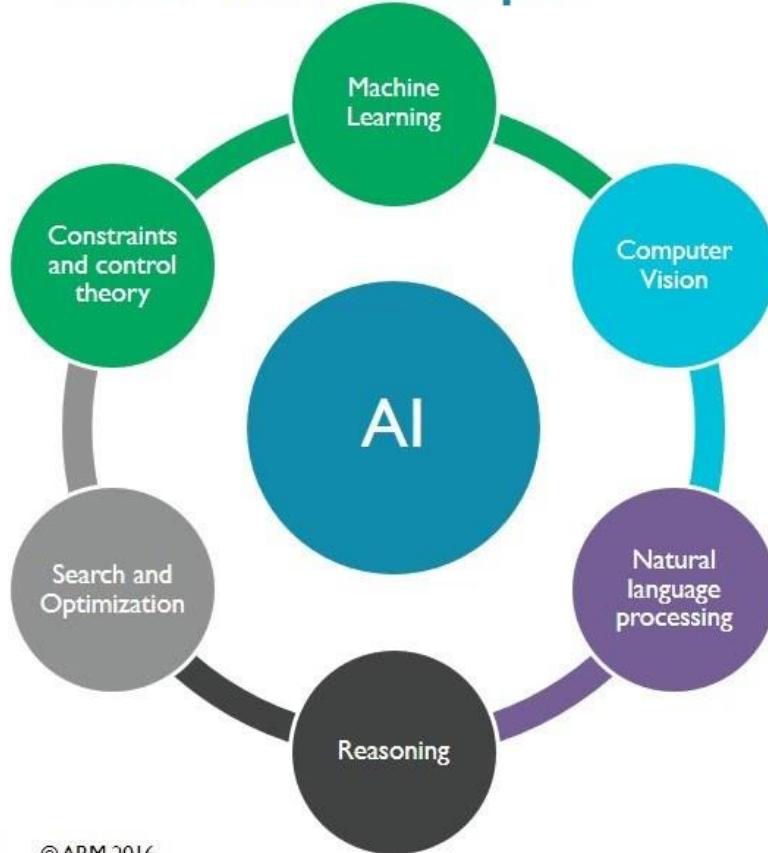


- Machine Learning und Data Mining können zum besseren Verständnis des Problems beitragen



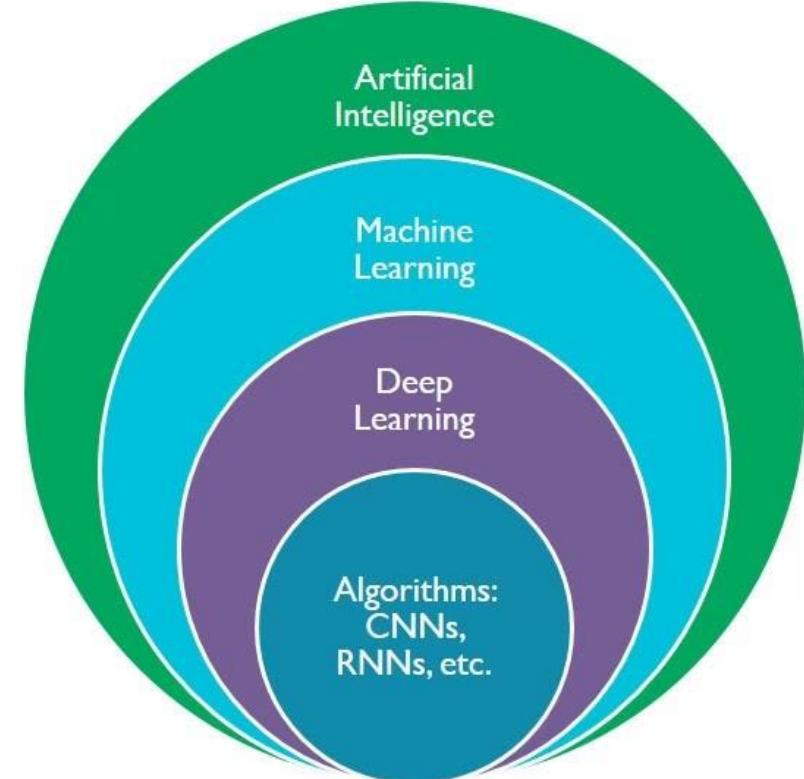
Artificial Intelligence and Machine Learning

The AI landscape



6

©ARM 2016



ARM

Live Demo

Vader Sentiment-Analyse

Herausforderungen Machine Learning

- Machine Learning und Predictive Applications bringen im Vergleich zu traditionellen Software Systemen **neue Herausforderungen** mit sich
 - Unzureichende Trainingsdaten
 - Trainingsdaten sind nicht repräsentative (sampling bias)
 - Schlechte Datenqualität (Fehlwerte → oder Fehlwerte, die nicht als solche erkennbar sind Bsp.: als „unknown“ gekennzeichnet)
 - Irrelevante oder unzureichende Features
 - Overfitting → grundlegendes Problem ist Bias/Variance Trade-Off
- „**Garbage in, garbage out**“ ist zu beachten !
→ Besonders bei der Automatisierung von Geschäftsentscheidungen (DDD)



Live Demo

Teachable Machine

Anlegen eines virtual Environments & Installation nötiger Packages

Anlegen einer venv

1. Anaconda Prompt öffnen
2. „conda create –n myenv python=3.8“
3. Navigieren in Verzeichnis von requirements.txt
4. „pip install –r requirements.txt“
5. Aktivieren der venv mit „conda activate myenv“



Importieren der Notebooks aus GitHub

1. Aktivieren der venv
2. Git installieren → unter Windows leider nur durch exe möglich
3. “git clone https://github.com/tigi-coder/ml_with_python.git”



Einführung Python/Pandas/Numpy

Typen von Data Science Problemen

- Data Science Probleme sind in ihrer Form einzigartig, es gibt keine Allgemeinlösung für jede Aufgabenstellung → No free lunch theorem
- Kombination mehrerer Faktoren macht Problem einzigartig (Ziele, Wünsche, Einschränkungen, Persönlichkeiten...)
- Aufteilung der Aufgabe in Teilprobleme → werden separat voneinander gelöst und anschließend wieder zu einer Komplettlösung zusammengesetzt
 - für gängige Teilaufgaben gibt es eventuell bereits vorhandene Lösungen/Tools
 - „Rad“ wird nicht neu erfunden → Kostenaspekt
 - Fokussierung auf Teilprobleme ermöglicht genauere Betrachtung von Problem → mehr Spielraum für Optimierung

Wie sehen diese Teilprobleme aus?



Typen von Data Science Problemen

1. Klassifikation und Wahrscheinlichkeitsabschätzung der Klassenzugehörigkeit
 - Unterteilung der Daten in Klassen z.B.: männlich, weiblich
 - Ziel ist es jeden Datenpunkt einer Klasse zuzuordnen
 - Zugehörigkeit zu mehreren Klassen meist ausgeschlossen
 - Fragestellung: Welche Kunden haben hohes Abwanderungspotenzial?
2. Regression
 - Vorhersagen eines numerischen Wertes
 - Fragestellung: Wieviel Umsatz werden wir im nächsten Jahr voraussichtlich machen?



Typen von Data Science Problemen

3. Ähnlichkeitstest

- Anhand vorhandener Daten ähnliche Datenpunkte finden
- Fragestellung: Wie kann ich Leute finden, die ähnliche Produkte gekauft haben/mögen?

4. Clustering

- Zusammenfassen von Datenpunkten zu Gruppen anhand ihrer Ähnlichkeit
- Es wird vorrangig kein richtiges „Ziel“ verfolgt
- Fragestellung: Finden wir ähnliche Gruppen in unseren Daten wieder?



Typen von Data Science Problemen

5. Assoziationsregeln oder Warenkorbanalyse

- Zusammenhänge zwischen Daten finden, welche an denselben Transaktionen beteiligt sind. → Häufig zusammengekaufte Produkte. Bsp.: Bier und Windeln (Empfehlungssysteme)
- Statt Gruppierung nach Merkmalen von Objekten wird nach gleichzeitigem Auftreten in Transaktionen gesucht.

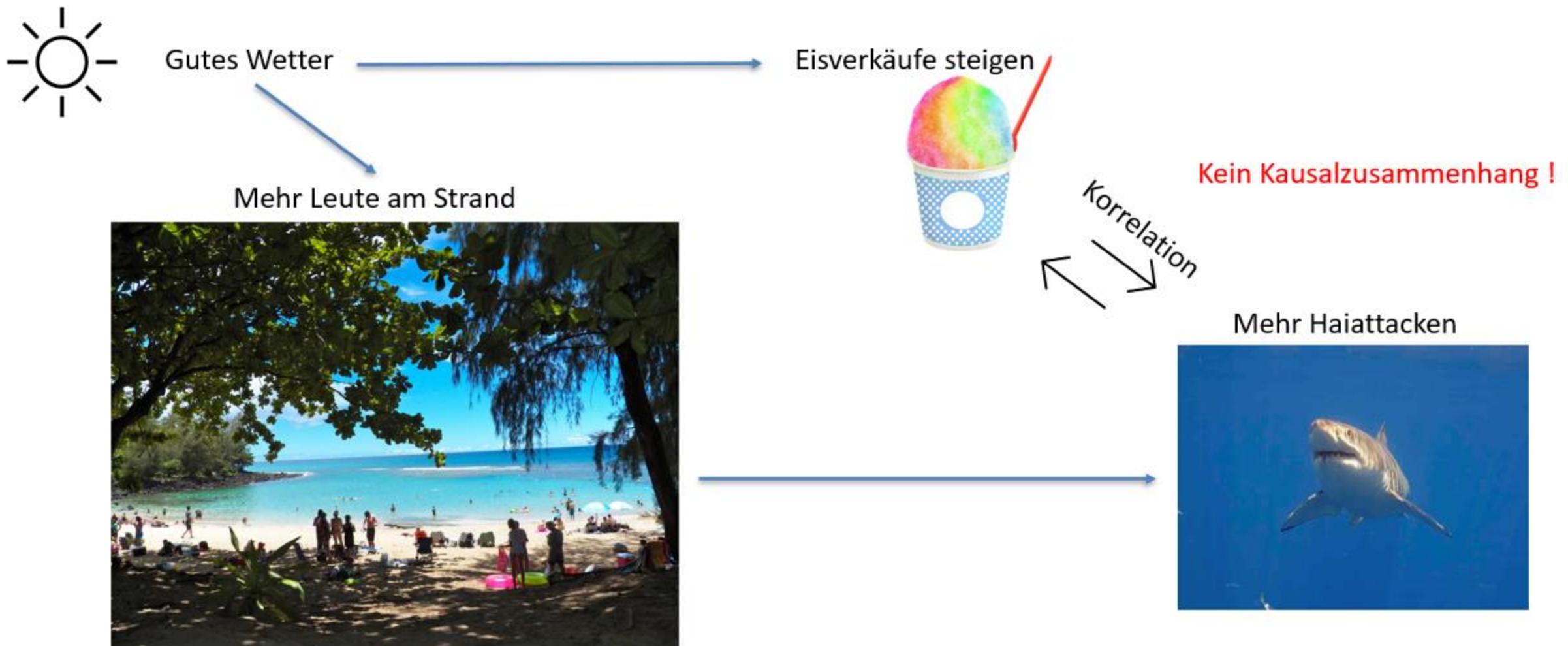
6. Kausalmodellbildung

- Ursache-Wirkungs- Zusammenhänge werden untersucht → welche Aktionen/Ereignissen werden von anderen beeinflusst?
- Bsp.: Mehr Nachfrage nach Eis bei heißem Wetter
- **Achtung! Korrelation ≠ Kausalität**



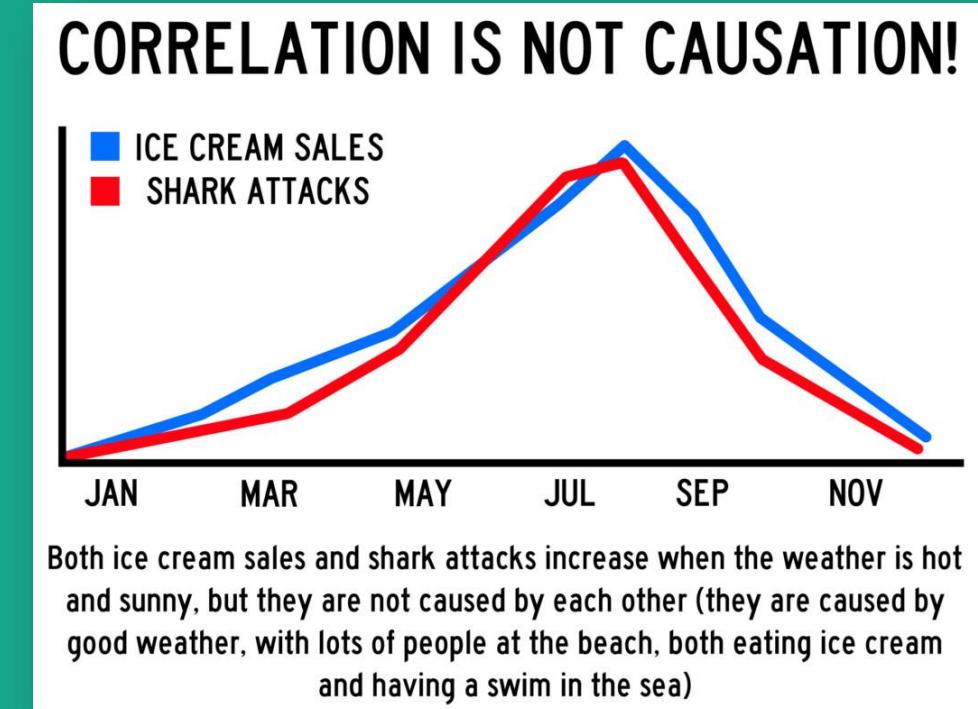
Korrelation ≠ Kausalität

Beispiel Hawaii



Korrelation ≠ Kausalität

- Wenn ein Feature A und B korrelieren, kann man nicht zwangsläufig davon ausgehen, dass A der Auslöser/Treiber für B ist.
- Bsp.: Bei der Untersuchung einer demografischen Datenbank, wird eventuell ein Zusammenhang zwischen der „Anzahl von Krankenhäusern“ und der „Anzahl von Autodiebstählen in der Region“ gefunden, welche korrelieren.
 - Dies bedeutet nicht, dass das eine die Ursache für das andere ist.
 - Beide sind offensichtlich verbunden, jedoch durch ein drittes Attribut, nämlich „Bevölkerung“





Bevor wir mit Machine Learning beginnen können,
müssen wir zuerst unsere Daten besser verstehen !

Explorative Datenanalyse

Was ist Explorative Datenanalyse (EDA) ?

„Exploratory data analysis is detective work“¹

*„Die Arbeit eines guten Ermittlers zeichnet sich dadurch aus,
dass er weiß, wonach es sich an einem Tatort zu suchen lohnt
und welche Hilfsmittel er dazu benötigt“²*



Arten von Features

Nominal skaliert

- In der Regel kategorisch
- Keine Reihenfolge ersichtlich
- Modus als einziges Lagemaß sinnvoll

Ordinal skaliert

- In der Regel kategorisch
- Können in Reihenfolge gebracht werden (Schulnoten,Dienstränge)
- Trotz Reihenfolge machen Rechenoperationen keinen Sinn (Kein arithmetisches Mittel möglich)
- Ebenfalls schwer quantifizierbar wie groß die Klassenunterschiede sind.



Arten von Features

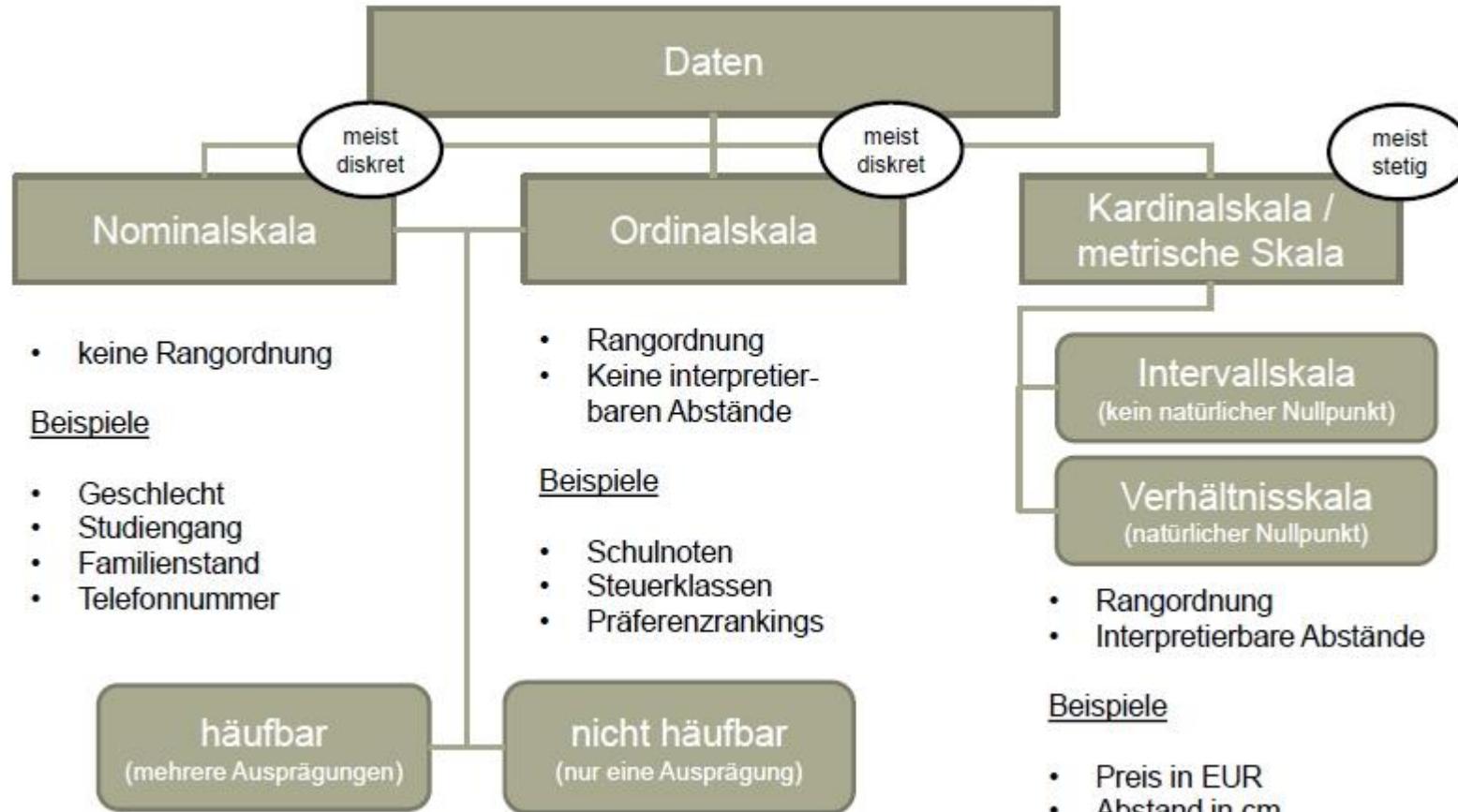
Metrisch skaliert (Kardinalskala)

- Meistens stetig
 - Intervallskala
 - Kein natürlicher Nullpunkt (Temperatur in Celsius)
 - Verhältnisskala
 - Natürlicher Nullpunkt (Temperatur in Kelvin, Körpergröße)



Arten von Features

Merkblatt



Ziele von EDA

- Explorative Datenanalyse ist ein wichtiger erster Punkt bei der Analyse von Daten und Erstellung von Predictive Applications
 - Daten näher kennenlernen (Muster erkennen)
 - Verteilung (symmetrisch, normal, schief), Datenqualitätsprobleme, Outlier Korrelationen/Beziehungen
 - Aufstellen und prüfen von Thesen/Annahmen
- Wichtig um relevante Features für eine Vorhersage zu finden
- Ziel ist es folgende Aspekte so früh wie möglich zu addressieren
 - Feststellen von Fehlern (evtl. in der Datensammlung/Verteilung)
 - Zutreffen von Annahmen
 - Grobe Untersuchung der Beziehung zwischen unabhängigen Variablen (möglichen Features) und abhängiger Variable (target)
 - Erste Modellselektion → was wäre geeignet



Statistische Kennzahlen - Lagemaße

Mean/Mittelwert

Einfaches Arithmetisches Mittel μ : $\mu = \frac{1}{N} \sum_{i=1}^N x_i$

Gewogenes Arithmetisches Mittel \bar{x} (Häufigkeitsverteilung): $\bar{x} = \frac{1}{n} \sum_{i=1}^k x_i n_i = \sum_{i=1}^k x_i \frac{n_i}{n}$

→ Falls absolute oder relative Häufigkeit gegeben

Getrimmtes Mittel: Berechnung des Arithmetischen Mittels auf einen Teilausschnitt der Daten (z.B.: Mit Ausschluss von Outliern)

Geometrisches Mittel (prozentuale Veränderungsrate):

$$\bar{x}_{geom} = \sqrt[n]{x_1 * x_2 * \dots * x_n} = \sqrt[n]{\prod_{i=1}^n x_i}$$

Harmonisches Mittel (Mittelwert von Verhältniszahlen Bsp.: 100km/h)

$$\bar{x}_{harm} = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}} \quad \longrightarrow \text{Division der Anzahl der Beobachtungswerte } n \text{ durch Summe der Kehrwerte}$$



Statistische Kennzahlen - Lagemaße

Zentralwert/Median

Stellt den exakten mittleren Wert dar (50 % der Daten liegen ober- und unterhalb des Medians)

→ Daten müssen vorher sortiert werden!

Berechnung:

Gerade-Anzahl an Datenpunkten:

$$ZW_1 = \frac{n}{2}$$

$$ZW_2 = \frac{n}{2} + 1$$

Ungerade Anzahl an Datenpunkten:

$$ZW = \frac{n+1}{2}$$



Statistische Kennzahlen - Lagemaße

Beispiel Median

Ungerade Anzahl an Datenpunkten:

- Age: 17 19 21 **22** 23 23 38

Gerade Anzahl an Datenpunkten:

- Age: 17 19 21 **22** **23** 23 23 38 → $[ZW_1 = 22 ; ZW_2 = 23] \ (22+23)/ 2 = 22.5$



Statistische Kennzahlen - Lagemaße

Quantile

Weitere Lagemaße für Daten sind Quantile:

25%-Quantil (unteres Quantil):

- 25% der Daten liegen unter diesem Wert und 75% der Daten oberhalb dieses Wertes

50%-Quantil = Median

75%-Quantil (oberes Quantil):

- 75% der Daten liegen unter diesem Wert und 25% der Daten oberhalb dieses Wertes



Statistische Kennzahlen - Lagemaße

Quantile

Berechnung:

1. Sortieren der Daten
2. Berechne $n * p \rightarrow n = \text{Anzahl Elemente}, p = \text{gewünschtes Perzentil}$ (Bsp.: 0,25 für 25%)
3. Falls Ergebnis ganzzahlig $\rightarrow x_p = \frac{1}{2} (x_{(k)} + x_{(k+1)})$

Falls Ergebnis nicht ganzzahlig $\rightarrow x_p = x_{(k)}$ $\rightarrow k = \text{nächste auf das Ergebnis folgende ganze Zahl}$ (Bsp.: $n * p = 2.5 \rightarrow x_p = 3$)



Statistische Kennzahlen - Lagemaße

Five Number Summary

Five Number Summary:

- Überblick über die Beschaffenheit einer Variable anhand fünf statistischer Kennzahlen
- Minimum, 25%-Quantil, Median, 75%-Quantil, Maximum

Five-Number-Summary für die Variable „horsepower“ im mpg-Dataset:

```
min      46.000000
25%     75.000000
50%    93.500000
75%   126.000000
max   230.000000
```



Statistische Kennzahlen - Streuungsmaße

Interquartilsabstand

- Interquartilsabstand oder Spannweite gibt die Differenz zwischen dem 75%-Quantil und 25%-Quantil an

Bsp.: Berechnung der IQR von Variable „weight“ des mpg-Datasets

```
df.weight.describe()  
count    398.000000  
mean    2970.424623  
std     846.841774  
min    1613.000000  
25%    2223.750000  
50%    2803.500000  
75%    3608.000000  
max    5140.000000
```

$$\text{IQR} = 3608 - 2223.75 = 1384.25$$



Statistische Kennzahlen - Streuungsmaße

Varianz und Standardabweichung

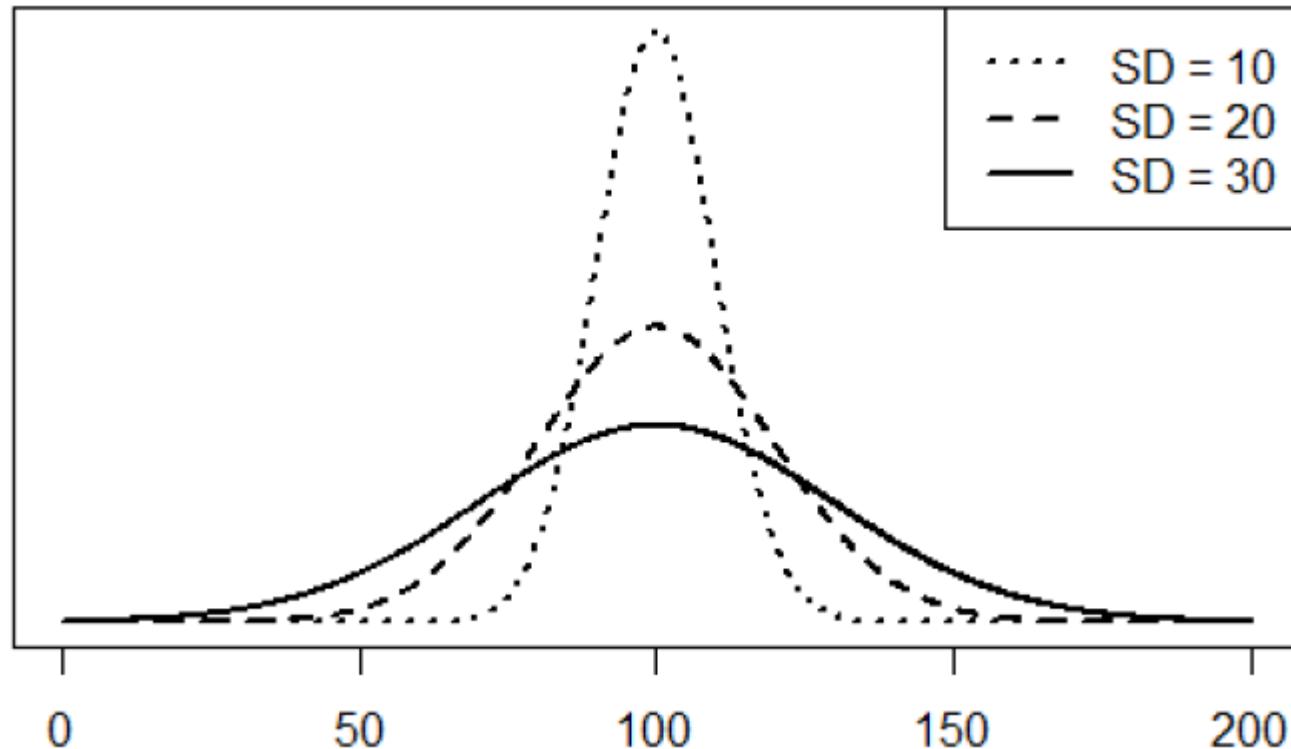
- Varianz wird häufig für Intervallskalierte Variablen berechnet
- Drückt aus wie weit die einzelnen Werte im Durchschnitt vom Mittelwert entfernt sind
→ Wird berechnet durch Durchschnitt der quadrierten Abweichungen
- Zieht man die Wurzel der Varianz bekommt man die Standardabweichung



Statistische Kennzahlen

Beispiel Varianz

- 3 Verteilungen mit gleichem Mittelwert und unterschiedlicher Varianz.



Statistische Kennzahlen - Streuungsmaße

z-Standardisierung

- Um zu prüfen ob aus den Daten eine allgemeine Aussage getroffen werden kann hilft die z-Standardisierung
 - Attribute müssen dafür Normalverteilt sein
 - Mittelwert = 0
 - Standardabweichung = 1
- Beispiel Körpergrößen von Frauen und Männern

Frage: Ist eine Frau mit 183 cm Körpergröße relativ zu den anderen Frauen größer als ein Mann mit der Größe von 200cm relativ zu den anderen Männern?

- Berechnung:
$$z_i = \frac{x_i - \bar{x}}{s}$$



Statistische Kennzahlen - Streuungsmaße

Kovarianz

- Eng verwandt mit der Korrelation
- Maß für den linearen Zusammenhang zweier Variablen (→ wie Korrelation auch)
- Nicht standardisiert → schwer Rückschlüsse aus den Werten zu schließen
- Positives Vorzeichen: Beide Variablen bewegen sich in die gleiche Richtung
- Negatives Vorzeichen: Beide Variablen bewegen sich in entgegengesetzte Richtungen
- Standardisierte Kovarianz = Korrelation

Berechnung:

$$\text{cov}_{x,y} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



Statistische Kennzahlen - Streuungsmaße

Korrelation

- Mithilfe der Korrelation lassen sich Beziehungen zwischen den vorliegenden Variablen finden
- Der Wert der Korrelation (Korrelationskoeffizient) wie stark eine Variable x, eine andere Variable y beeinflusst.
- Misst den linearen Zusammenhang zweier Variablen
- Korrelationskoeffizient variiert von -1 bis +1
 - +1 = Wenn x um 1 Einheit steigt, steigt y ebenfalls um 1 Einheit
 - 0 = keine Korrelation
 - -1 = Wenn x um 1 Einheit steigt, sinkt y um 1 Einheit

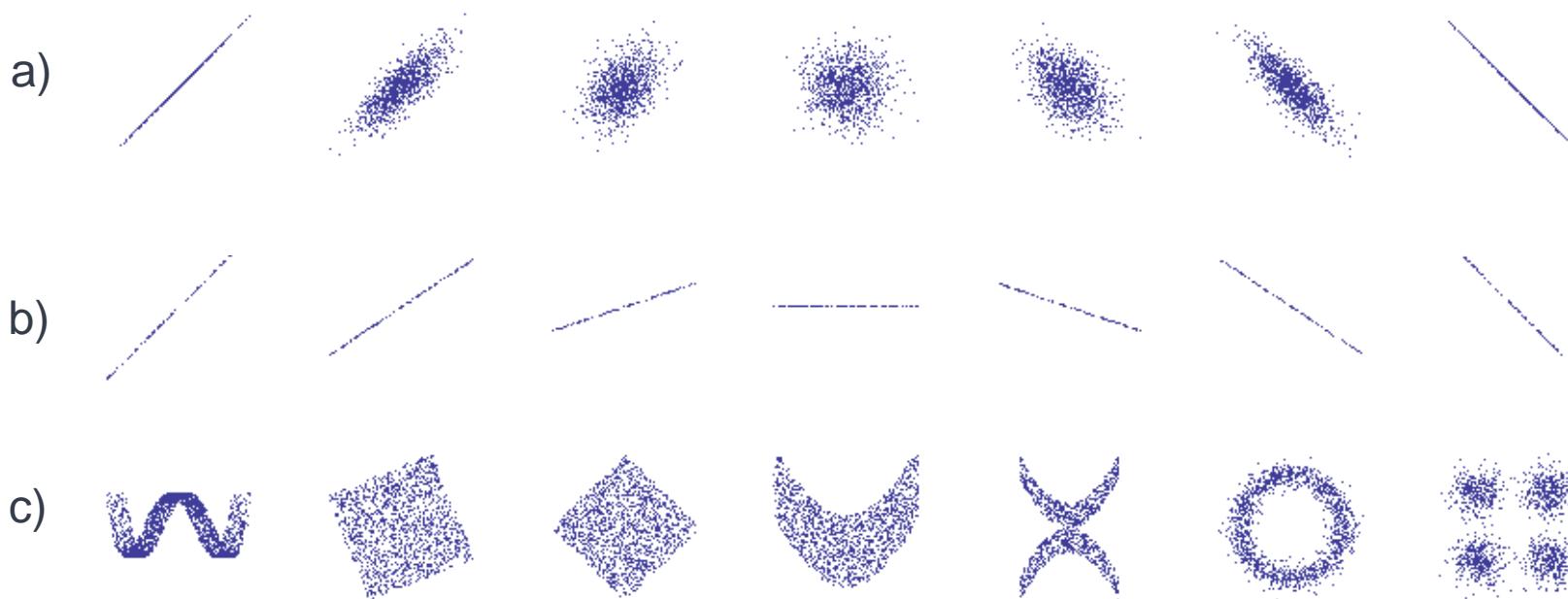
Berechnung beispielsweise durch Pearson Korrelationskoeffizienten:

$$\text{cor}_{x,y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})} \sqrt{\sum_{i=1}^n (y_i - \bar{y})}} = \frac{\text{cov}_{x,y}}{\sigma_x \sigma_y}$$



Übung

Korrelation erkennen



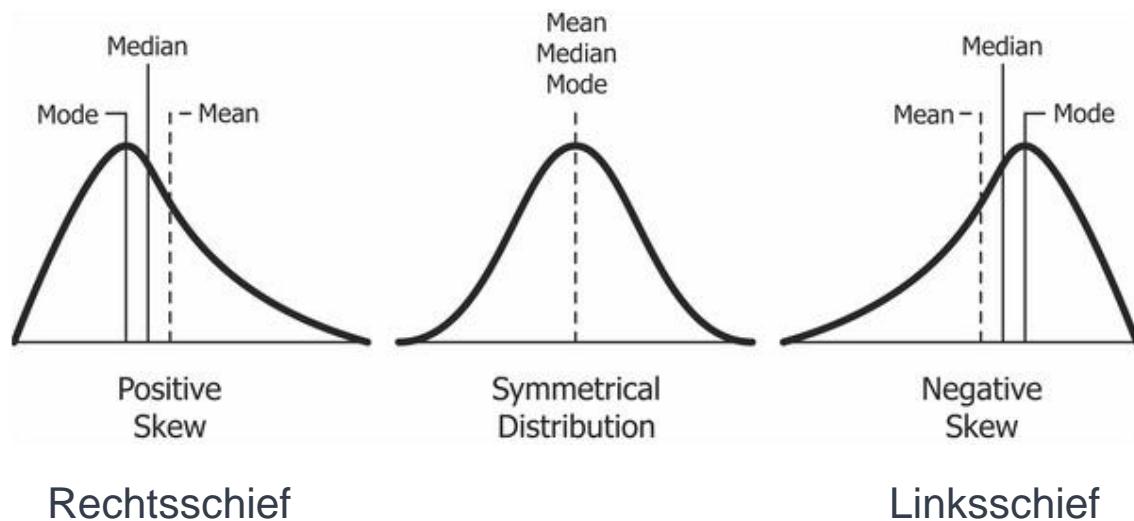
Live Demo

Pearson Korrelation

Statistische Kennzahlen - Streuungsmaße

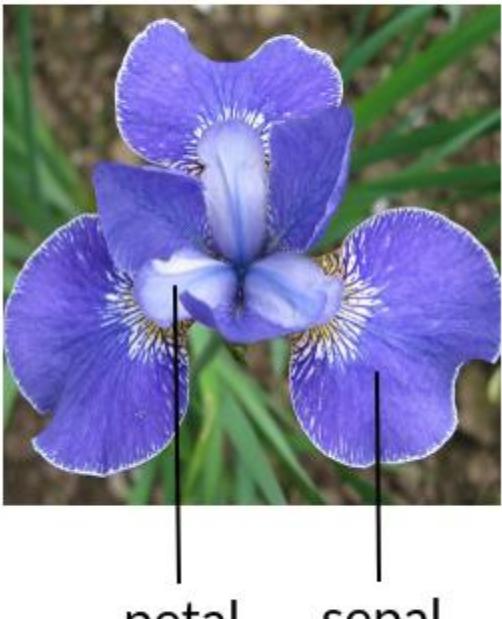
Schiefe

- Schiefe: Maß für die Symmetrie einer Verteilung
- Höchster Punkt der Verteilung zeigt den Modus → am häufigsten vorkommender Wert
- Verteilung ist „schief“ wenn sie auf einer Seite des Modus schneller ansteigt als auf der anderen → häufig durch Extremwerte verursacht

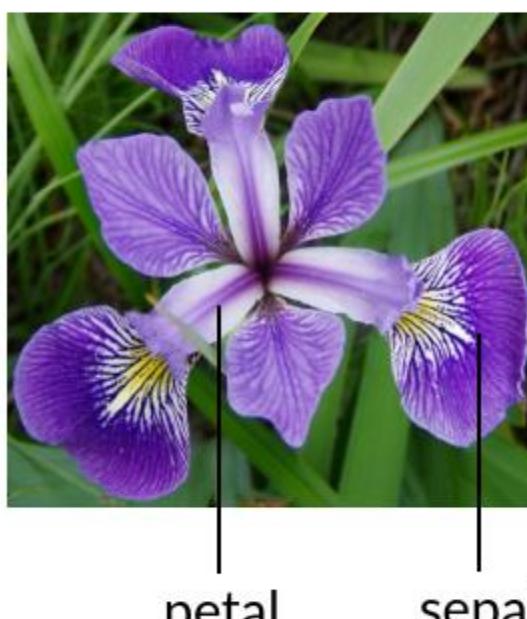


Iris Dataset

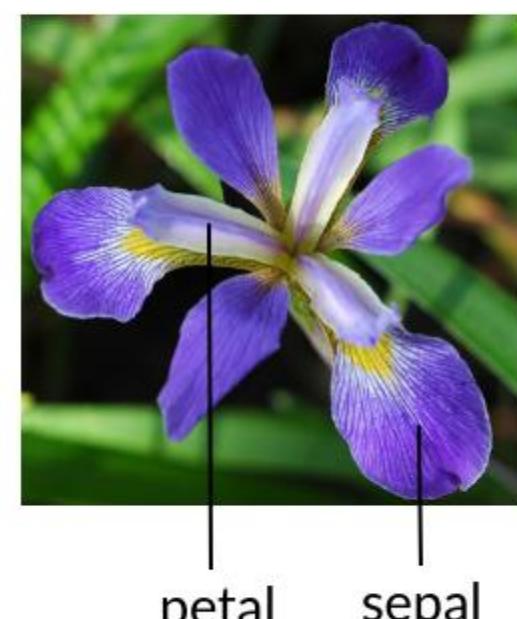
iris setosa



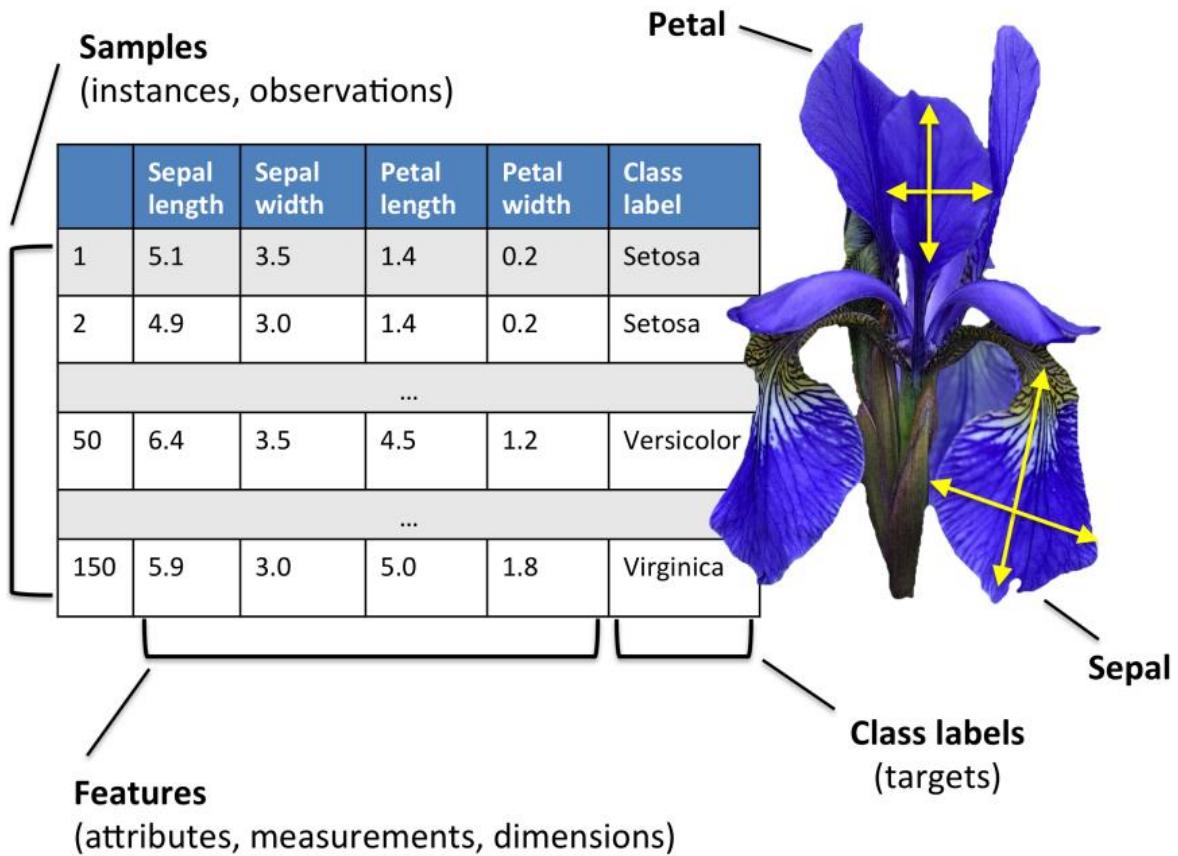
iris versicolor



iris virginica



Iris Dataset



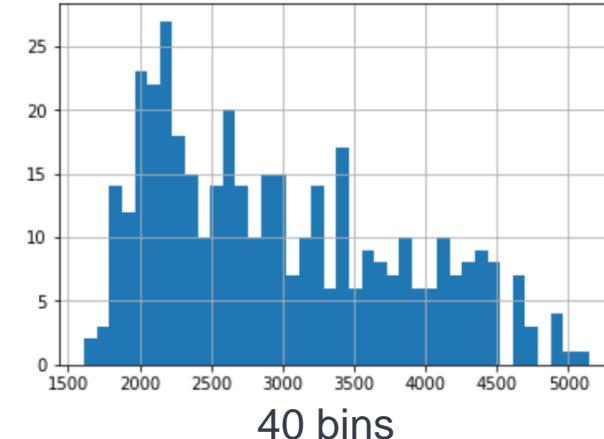
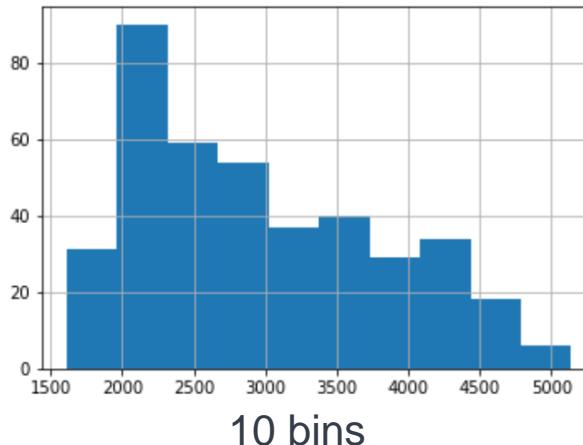
Statistische Kennzahlen mit Pandas/Numpy

Visualisierungsmethoden

Univariate Visualisierungsmethoden

Histogramm

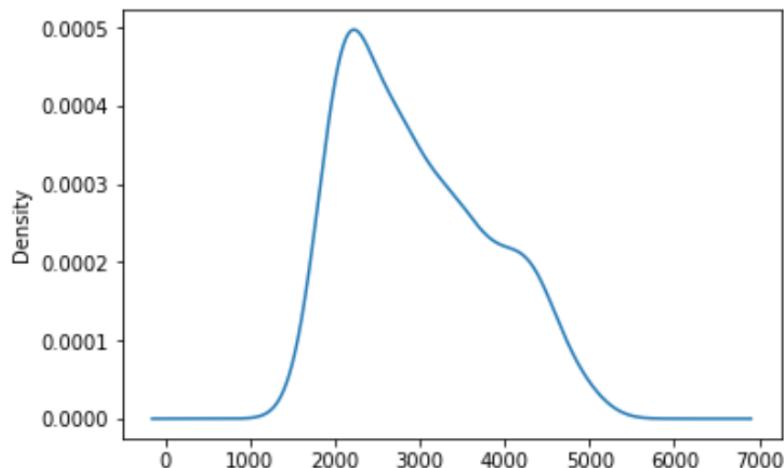
- Stellt dar wie häufig ein Wert einer Variable auftritt (Häufigkeit)
- Ermöglicht ersten Eindruck über Verteilung der Daten
 - Variable wird auf X-Achse abgetragen
 - Auftrittshäufigkeit wird auf Y-Achse dargestellt
- Modus lässt sich ableiten → am häufigsten vorkommender Wert oder Wert mit der höchsten Auftrittswahrscheinlichkeit
- Binning: Durch erhöhen oder reduzieren von bins kann ein Histogramm geringere oder höhere Genauigkeit (mehr Noise) aufweisen



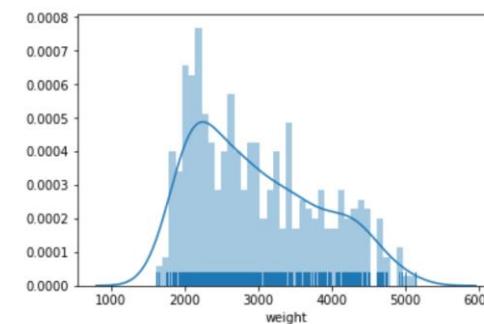
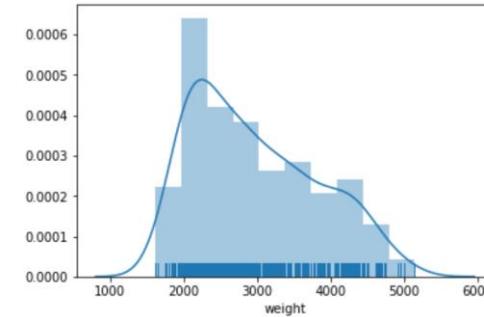
Univariate Visualisierungsmethoden

Density Plot

- Variation des Histogramms
- Werte werden stark geglättet dargestellt → Spitzen werden reißen nicht so stark aus
- Peaks zeigen an wo sich die Werte konzentrieren



Related to histogram



Univariate Visualisierungsmethoden

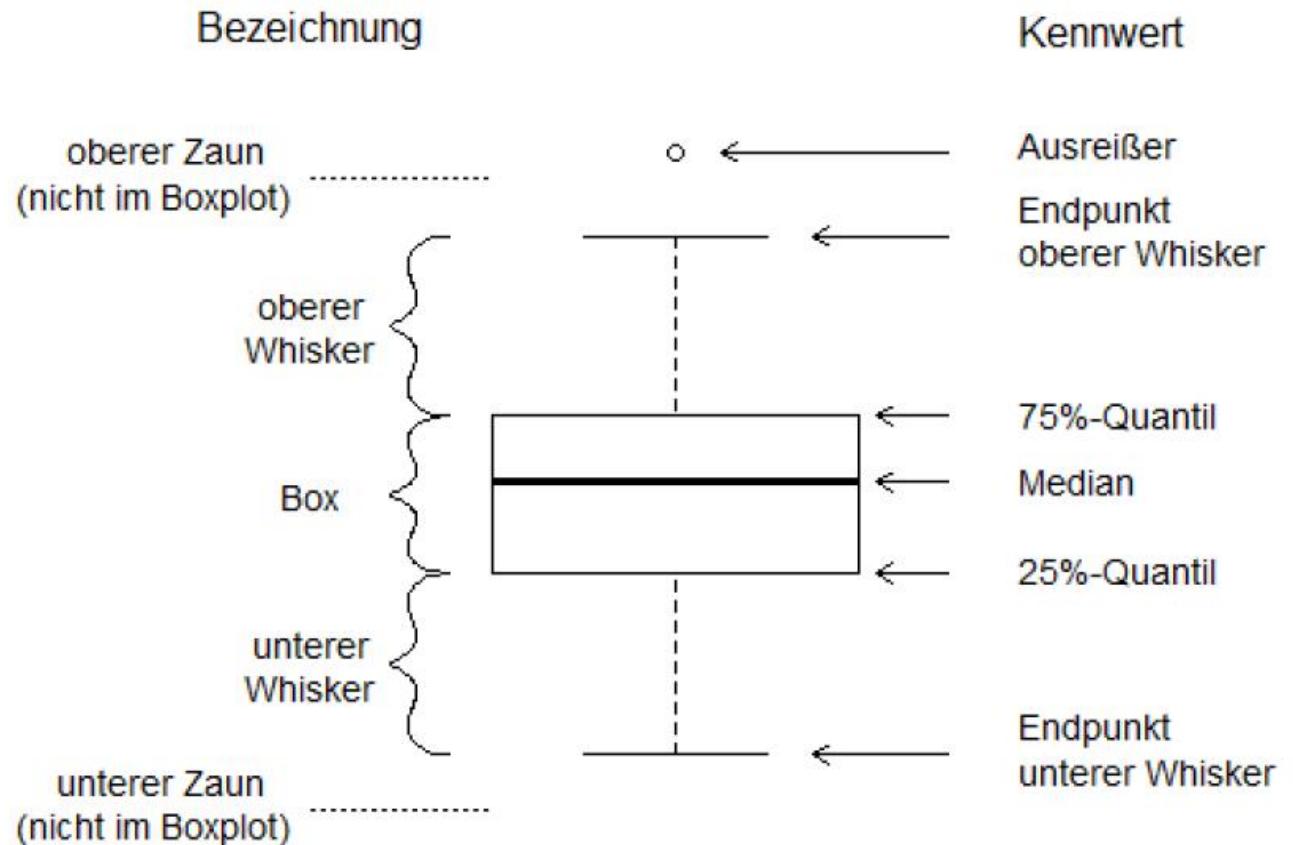
Boxplot

$$\text{Oberer Zaun} = Q_{75\%} + 1,5 * \text{IQR}$$

$$\text{Unterer Zaun} = Q_{25\%} - 1,5 * \text{IQR}$$

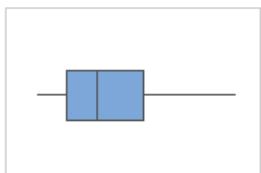
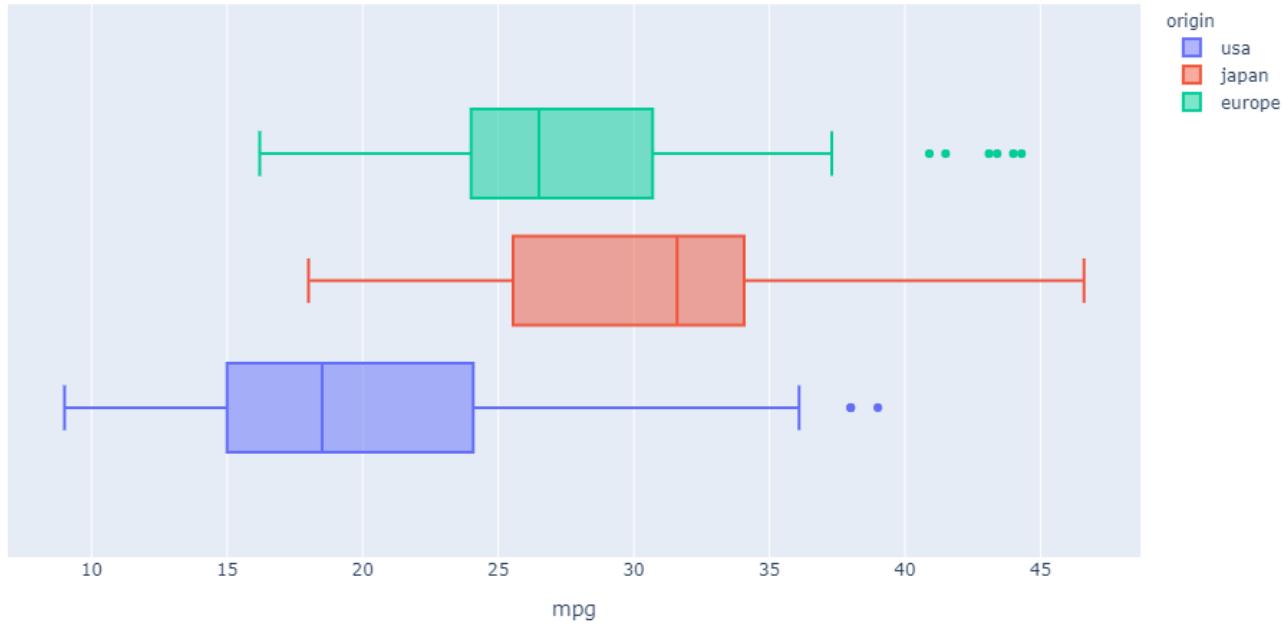
Vorteile:

- Schneller Aufschluss über Verteilung der Daten
- Grobe Ausreißer Erkennung möglich
- Schiefe kann ungefähr bestimmt werden

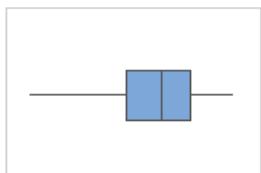


Univariate Visualisierungsmethoden

Boxplot



Rechtsschief



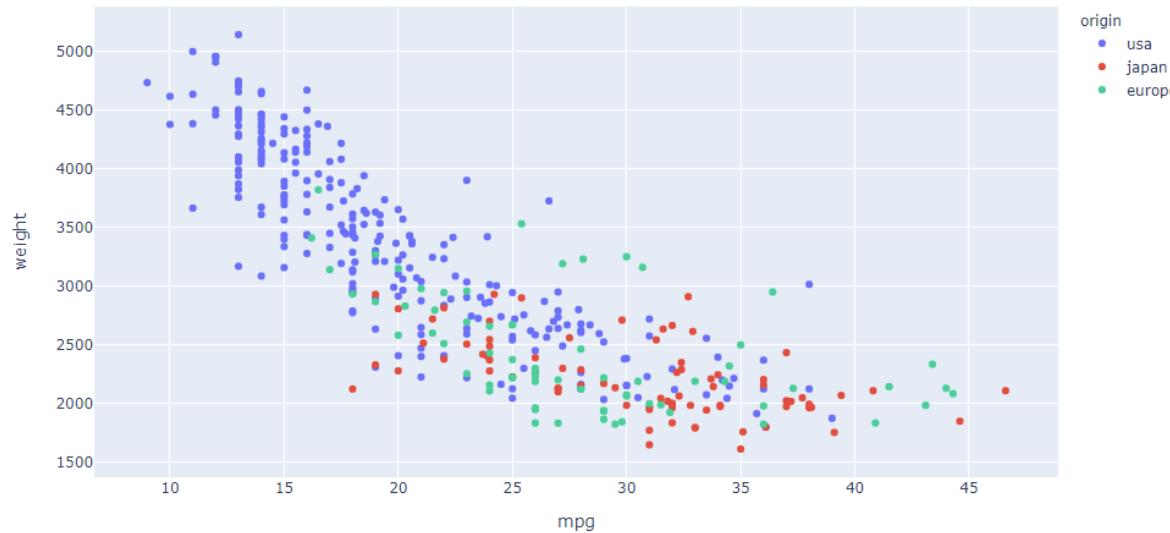
Linksschief



Multivariate Visualisierungsmethoden

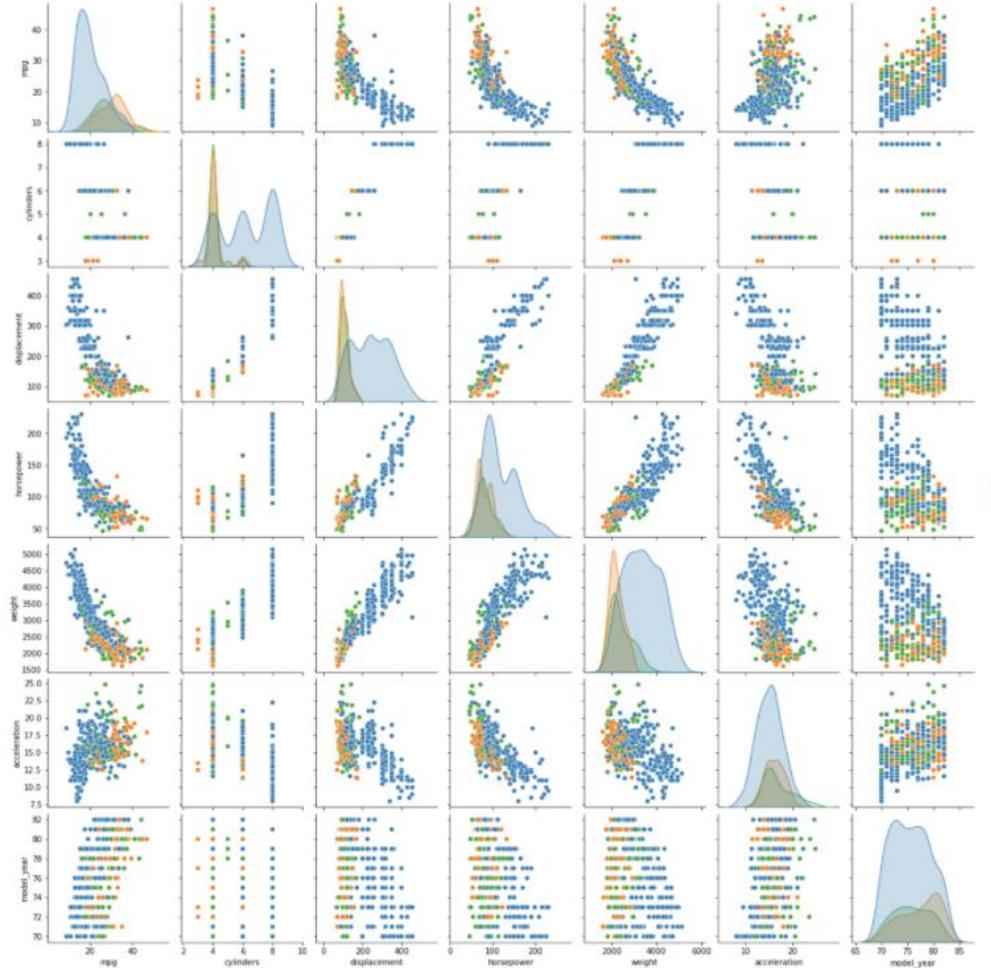
Scatterplot

- Jeder Datenpunkt wird einzeln auf Graph abgetragen
- Durch Kombination von x-Achse,y-Achse, color, size etc. können mehrere Features gleichzeitig betrachtet werden
- Bietet sich zur Untersuchung von Beziehung zwischen numerischen Features an (Beispielsweise Korrelation)



Multivariate Visualisierungsmethoden

Pairplot



- Jedes Attribut wird in jeder Kombination mit anderen Attributen verglichen
- Vergleich mit eigenem Attribut zeigt eigene Verteilung (Density Plot) → ähnlich Histogramm (roter Pfeil)
- Kombinationen bilden sich aus einem Attribut aus vertikaler Achse (blauer Pfeil) und horizontaler Achse (grüner Pfeil)
- Darstellungsform ist ein Scatterplot



EDA mit Python

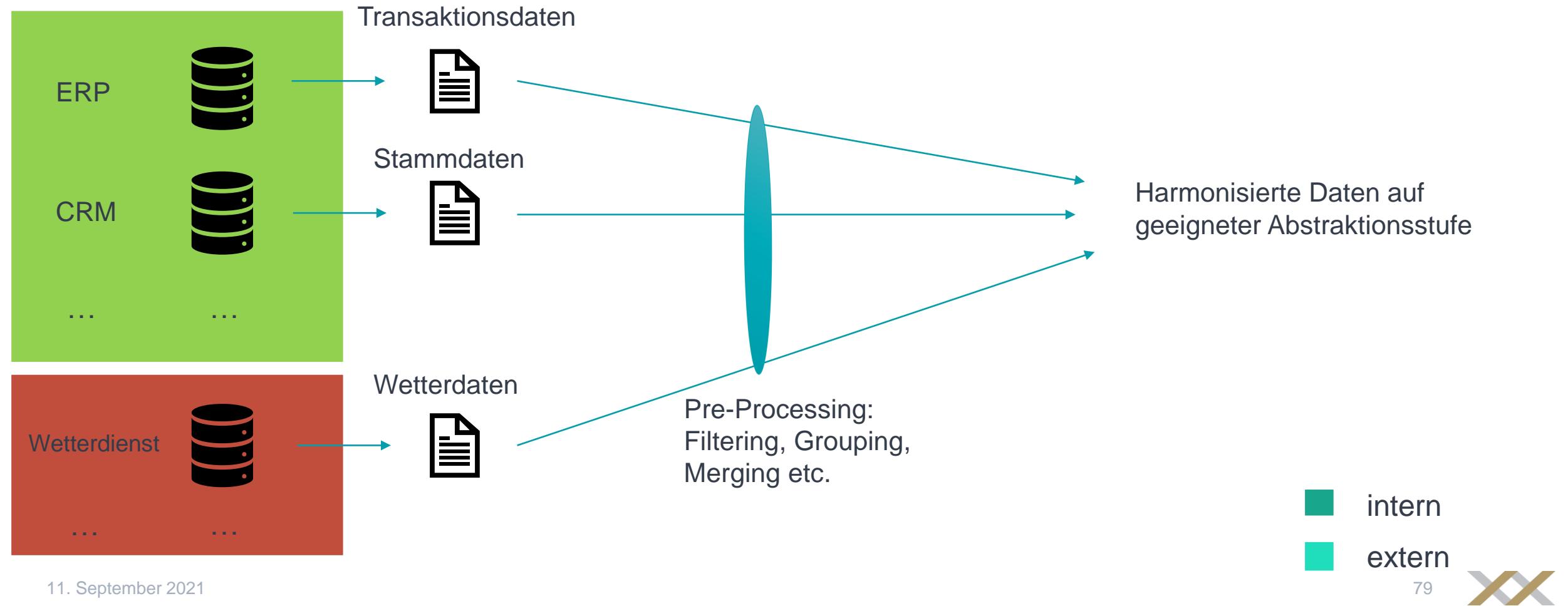
Querying/Merging/Filtering/Grouping

Warum Merging/Filtering/Grouping... ?

- Daten werden selten (eher nie) in geeigneter Abstraktionsstufe ausgeliefert
- Aufgabe des Data Scientist ist mit Auswahl geeigneter Methoden um passende Datenform für späteres Modell zu wählen
- Auch für EDA kann eine höhere/niedrigere Abstraktionsstufe größeren Einblick in die vorliegenden Daten geben
- Teil des Pre-Processing neben anderen notwendigen Schritten wie beispielsweise der Umgang mit Fehlwerten.



Warum Merging/Filtering/Grouping... ?



Querying/Merging/Filtering/Grouping mit Python

Pre-Processing

Pre-Processing

Überblick

- Daten sind in Wirklichkeit „unrein“: Möglicherweise sehr viel inkorrekte Daten, z.B.: Messfehler, Menschliches Versagen oder Computerfehler, Übertragungsfehler, etc.
 - Unvollständig: Attribute fehlen
 - Noisy: verzerrte Daten, Error, Outliers (z.B.: Größe = -100cm)
 - Inkonsistenzen: Diskrepanzen zwischen verschiedenen Einträgen (Age vs. Birthday)
 - Intentional Errors: Vesteckte Fehlwerte
 - z.B.: jeder fehlende Eintrag für „birthday“ = „January 1st“

→ Ursprung für Fehlwerte kann vielerlei sein.



Source: <https://www.realisedatasystems.com/3-reasons-why-data-quality-should-be-your-top-priority-this-year/>



Warum Pre-Processing?

- Schlechte Datenqualität → geringe Qualität der darauf aufbauenden Data Mining und Machine Learning Ergebnisse
 - Verbessert die Performance von Predictive Applications → z.B.: Accuracy
 - Modellierung setzt gute Datenqualität voraus → Klassifikationsalgorithmen können grundsätzlich nicht mit Fehlwerten umgehen
- Datenaufbereitung, -säuberung und –transformation beanspruchen den Hauptteil der Arbeit bei Predictive Applications und Data Science Projekten.



Hauptaufgabe des Pre-Processing

- Datenbereinigung
 - Füllen von Fehlwerten, glätten von „noisy“ data, identifizieren und entfernen von Outliern und „noisy“ data, auflösen von Inkonsistenzen.
- Datenintegration
 - Integration von mehreren Datenbanken oder Files (Bsp.: Merging)
- Datentransformation
 - Normalisierung und Aggregation
 - Datendiskretisierung
- Datenreduktion
 - Reduzierung des Datenvolumens mit Beibehaltung der selben analytischen Ergebnisse



Datenbereinigung

Fehlwerte

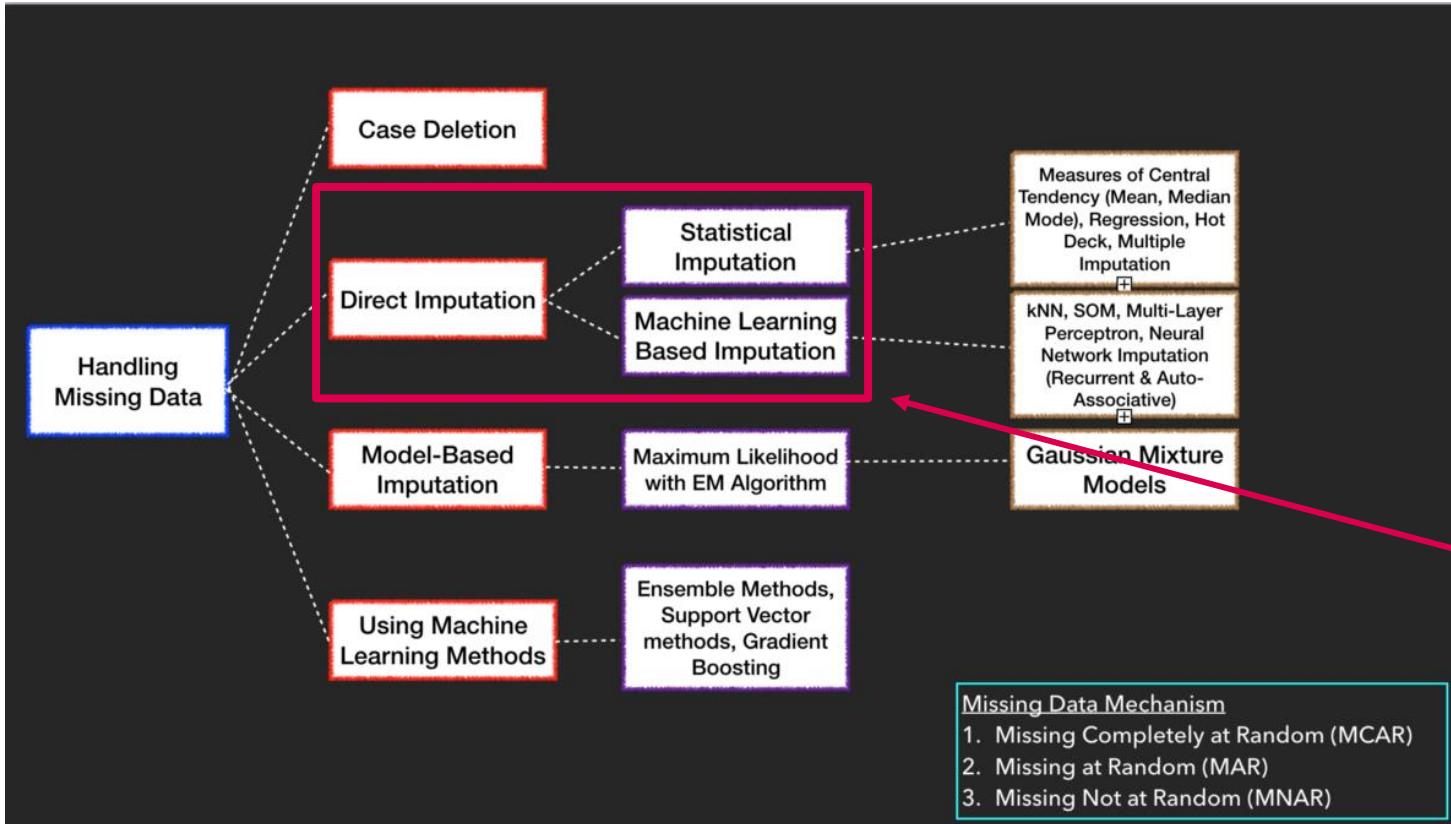
- Aus der Statistik werden grundsätzlich 3 verschiedenen Klassen unterschieden
 - MCAR: Missing Completely at Random
 - Die Wahrscheinlichkeit, dass ein Wert fehlt ist nicht abhängig von den vorliegenden Datenwerten, noch von den fehlenden Datenwerten.
 - Beispiel: Ausfall eines Temperatursensors aufgrund technischer Probleme tritt zufällig auf
 - MAR: Missing at Random
 - Die Wahrscheinlichkeit, dass ein Wert fehlt ist teilweise von den anderen vorliegenden Daten abhängig aber nicht von irgendwelchen anderen Fehlwerten.
 - Beispiel: Ausfall eines Temperatursensors eher bei Nacht, trotzdem noch zufällig.
 - MNAR: Missing Not at Random
 - Die Wahrscheinlichkeit, dass eine Wert fehlt liegt an den Fehlwerten selbst.
 - Beispiel: Ausfall eines Temperatursensors wahrscheinlicher bei extremen Temperaturen.



Datenbereinigung

Umgang mit Fehlwerten im Fall von MAR

- **Überblick I**



Teil dieser Schulung

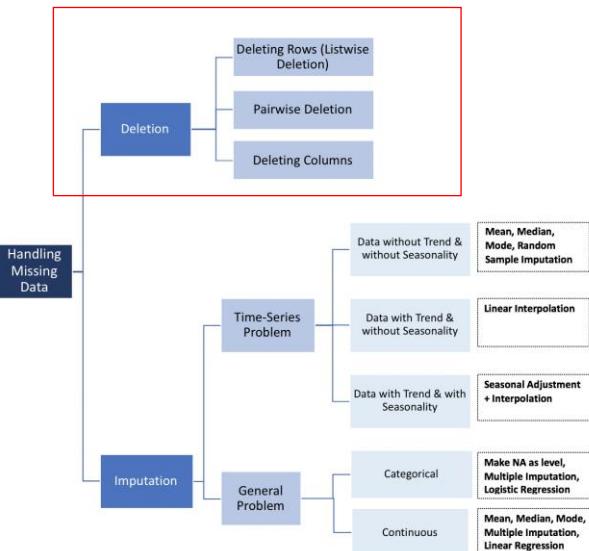
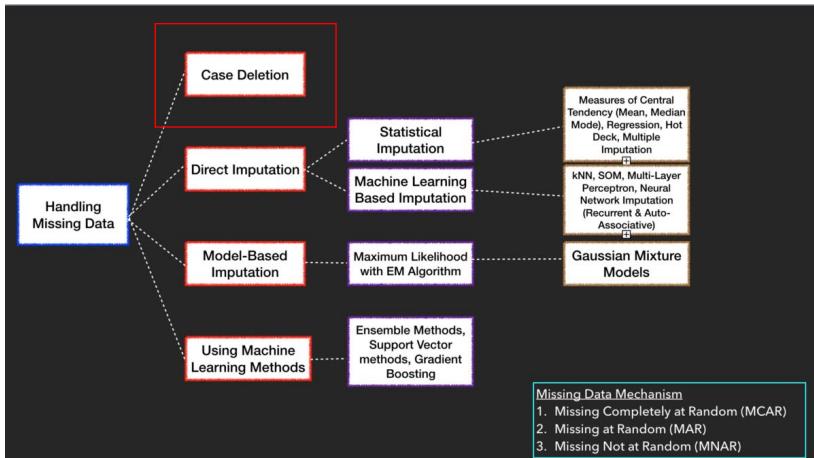
Source: <https://medium.com/ibm-data-science-experience/missing-data-conundrum-exploration-and-imputation-techniques-9f40abe0fd87>



Datenbereinigung

Umgang mit Fehlwerten im Fall von MCAR

- Umgang mit Fehlwerten im Fall von MCAR: Case Deletion
 - Nur die Werte die komplett verfügbar sind werden benutzt
 - Es wird hier kein BIAS produziert, da Daten zufällig fehlen
 - Aber: Datenfelder zu löschen in denen sich Fehlwerte befinden resultiert in einer kleineren Datenmenge als zuvor.
 - Diese Strategie kann dazu führen, dass ein Großteil der Daten ignoriert wird.



Sources: <https://medium.com/ibm-data-science-experience/missing-data-conundrum-exploration-and-imputation-techniques-9f40abe0fd87>

Source: <https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4>



Datenbereinigung

MCAR

- Arten von „Deletion“ (Lösung/Entfernung)

- **Listwise deletion**

- Entfernen aller Datensätze die einen oder mehrere

Fehlwerte enthalten

- Reduziert die Anzahl Datenpunkte drastisch
 - Annahmen dass MCAR vorliegt nicht immer richtig (Eventuell gibt es einen versteckten Grund für die Fehlwerte). → führt zu Verzerrungen in den Daten

User	Device	OS	Transactions
A	Mobile	NA	5
B	Mobile	Android	3
C	NA	iOS	2
D	Tablet	Android	1
E	Mobile	iOS	4

- **Pairwise deletion**

- Nur vorhandene samples werden zur Analyse herangezogen.
 - (1) Kombination aus User, Device, Transaction
 - (2) Kombination aus User, OS, Transaction
 - Keine Datenpunkte werden gelöscht
 - Ignorieren von Variablen mit Fehlwerten – Beispiel Berechnung Covarianz:
 - Cov(ageNA, DV1) benutzt sample 3 und 4
 - Cov(ageNA, DV2) benutzt sample 1,3 und 4
 - Cov(DV1, DV2) benutzt sample 2, 3 und 4

→ Schwer zu interpretieren; setzt ebenfalls MCAR voraus

User	Device	OS	Transactions
A	Mobile	NA	5
B	Mobile	Android	3
C	NA	iOS	2
D	Tablet	Android	1
E	Mobile	iOS	4

	ageNA	DV1	DV2
[1,]	18	NA	9
[2,]	NA	1	4
[3,]	27	5	2
[4,]	22	-3	7



Datenbereinigung

MCAR

- **Arten von „Deletion“ (Lösung/Entfernung)**
 - **Löschen von Spalten (nicht empfohlen)**
 - Nur falls >60% der Daten fehlen oder das Attribut nicht von Bedeutung ist !



Datenaufbereitung

MNAR

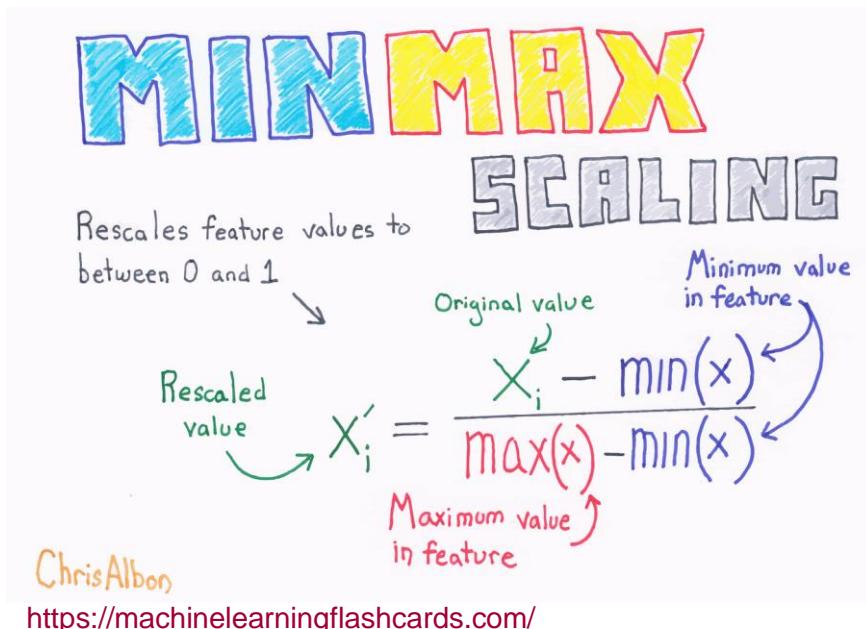
- Missing not at Random (MNAR)
 - Fehlwerte einer Variable stehen möglicherweise in Verbindung zu vorliegenden oder nicht vorliegenden Variablen.
 - (1) Fehlwert hängt von hypothetischem Wert ab (Bsp.: Menschen mit hohen Einkommen wollen ihren Gehalt meistens nicht preisgeben in Befragungen). → Fehlwert möglicherweise hohes Einkommen
 - (2) Fehlwert hängt vom Wert einer anderen Variable ab (Bsp.: Angenommen Frauen geben generell ihr Alter nicht gerne an! Hier wäre der Fehlwert beeinflusst von der Variable gender.)
- Wird als “non-ignorable” bezeichnet da der Grund des Fehlens nicht ignoriert werden kann.
 - Kein Entfernen oder einfaches Ersetzen von Daten durch Imputation (siehe unten) möglich!
- MCAR und MAR: Daten mit Fehlwerten können ohne Probleme gelöscht werden.
- MNAR: Das Entfernen von Fehlwerten könnte eventuell einen Bias im Modell erzeugen.



Datentransformation

Normalisierung

- Normalisierung durch min-max Skalierung.
 - Normalisieren aller Werte zu einem Wertebereich von 0 bis 1.



Körpergröße (unskaliert)	Körpergröße (Min-Max skaliert)
190	1
175	0.57
155	0
176	0.6
164	0.257

Datentransformation

Normalisierung

- Normalisierung durch Standard Skalierung (z-Standardisierung)

STANDARDIZATION

$$\text{Standardized feature value } \tilde{x}_i = \frac{x_i - \bar{x}}{\sigma}$$

Value of the i th observation
Mean of the feature vector
Standard deviation of the feature vector

Standardization is a common scaling method.
 \tilde{x}_i represents the number of standard deviations each value is from the mean value. It rescales a feature to have a mean of 0 and unit variance.

Chris Albon

<https://machinelearningflashcards.com/>

Standardization:

$$z = \frac{x - \mu}{\sigma}$$

with mean:

$$\mu = \frac{1}{N} \sum_{i=1}^N (x_i)$$

and standard deviation:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

Unit variance bedeutet, dass die Standardabweichung der Variable gegen 1 geht wenn die sample size gegen unendlich geht.



Datentransformation

Encoding

- Viele Machine Learning Algorithmen können nur mit numerischen Attributen als Feature/Target umgehen.
 - grundsätzlich befinden sich mehrere kategorische Variablen in unseren Daten
 - Encoding zielt darauf ab kategorische in numerische Variablen zu konvertieren
 - Es gibt mehrere Optionen → Label Encoding oder One-Hot Encoding



Datentransformation

Encoding

Beispiel:

Sample	Category
1	Human
2	Human
3	Penguin
4	Octopus
5	Alien
6	Octopus
7	Alien

LabelEncoding



Sample	Category	Numerical
1	Human	1
2	Human	1
3	Penguin	2
4	Octopus	3
5	Alien	4
6	Octopus	3
7	Alien	4

Achtung: Wir müssen vorsichtig sein mit Label Encoding! Machine Learning Algorithmen könnten das Attribut als Ordinal skaliert interpretieren.

OneHotEncoding



Sample	Human	Penguin	Octopus	Alien
1	1	0	0	0
2	1	0	0	0
3	0	1	0	0
4	0	0	1	0
5	0	0	0	1
6	0	0	1	0
7	0	0	0	1

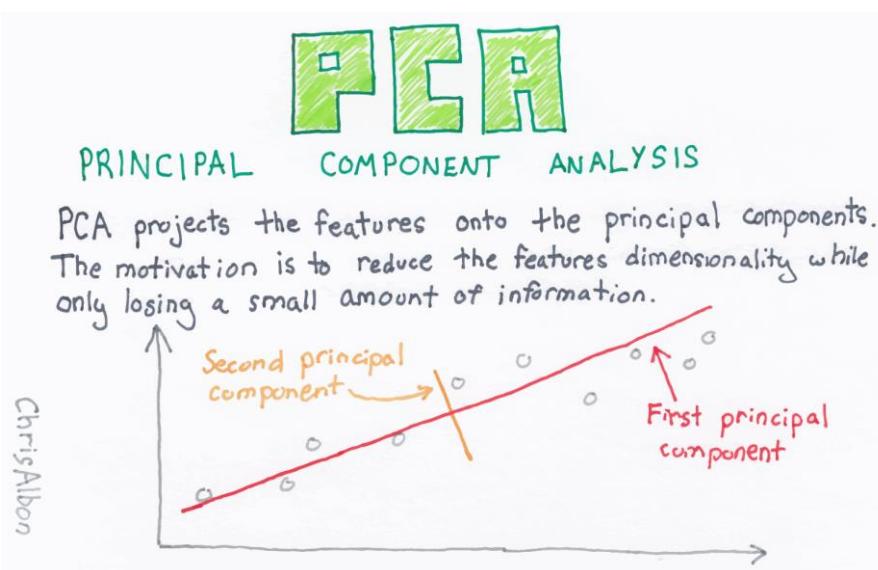
One-Hot Encoding funktioniert besser mit normal nominalen/kategorischen Attributen für Machine Learning



Datenreduktion

PCA

- Ziel: Reduzieren der Anzahl von Features durch Identifikation von Korrelationen zwischen mehreren Features und reduzieren dieser zu einem (oder mehreren) Features, den sogenannten „Principal Components“
 - Reduzierter Rechenaufwand und Feature Dimensionen mit nur geringem Verlust an Informationen



<https://machinelearningflashcards.com/>



Fehlwerte & Pre-Processing mit Python +Vorschau PCA

Die richtigen Features finden

Korrelation vs. PPS (Predictive Power Score)

- Nicht jedes Feature ist hilfreich um Target vorherzusagen
- Extraktion der relevanten Features ist ausschlaggebend für resultierende Performance
- Verfahren wird benötigt, welches auf kleinen und großen Datensätzen funktioniert
- Mögliche Methoden: Korrelation, PPS



Die richtigen Features finden

PPS – Predictive Power Score

- Hat wie Korrelation einen Wert von 0 bis 1 welcher aussagt wie gut ein einzelnes feature ein Attribut vorhersagen kann
- Deckt im Vergleich zu Korrelation auch nicht-lineare Zusammenhänge auf
- Korrelation geht von synchroner Abhängigkeit aus, jedoch liegen oft asynchrone Abhängigkeiten vor.
Bsp.: Postleitzahlen erklären das Attribut Stadt eindeutig, jedoch kann man von der Stadt nicht unbedingt auf die Postleitzahl schließen (z.B.: 50676 = Stadt Köln, aber Köln = ? PLZ)
- Kann auch kategorische Variablen in Betracht ziehen



Die richtigen Features finden

PPS – Predictive Power Score

Nachteile:

- Berechnung dauert bei größeren Datasets deutlich länger als die Correlation Matrix
 - Score sagt nichts über die Art des Zusammenhangs zwischen den Attributen aus (linear, oder nicht-linear)
 - Synergien zwischen mehreren Features werden nicht beachtet (wie bei Korrelation), müssen also noch untersucht werden
- Aussagekräftigsten Features nach und nach zu Modell hinzufügen und dabei Modellperformance monitoren (forward/backward selection)



PPS in Python

Supervised & Unsupervised Learning

Typen von Machine Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- ...



Supervised Learning

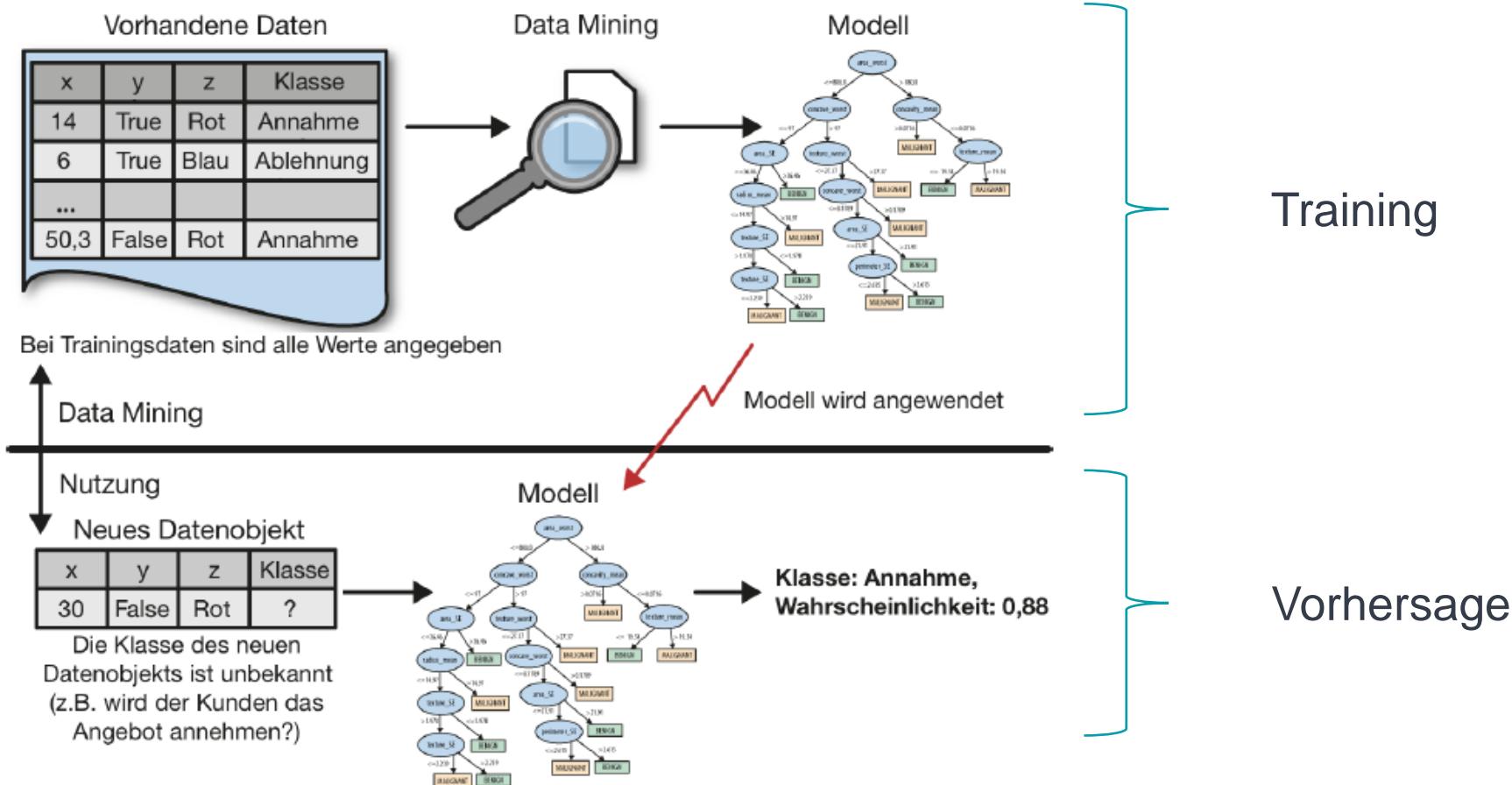
Fragestellung Supervised Learning:

z.B.: Gibt es Kunden, welche eine hohe Wahrscheinlichkeit besitzen ihren Vertrag zu kündigen?

- Training erfolgt auf historischen Daten, welche alle gelabelt sind.
- Ziel ist es durch die vorliegenden Daten (Features), die Zielvariable so genau wie möglich vorhersagen zu können und ggf. eine Wahrscheinlichkeit der Klassenzugehörigkeit zu ermitteln.
- Vorhersage erfolgt mit Daten bei der das Target oder die Klasse fehlt.

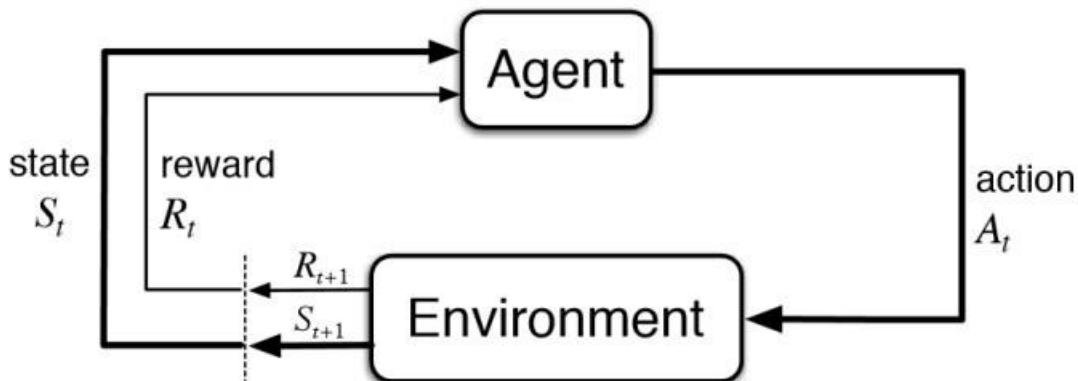


Supervised Learning



Reinforcement Learning

- Sogenannter „Agent“ wird trainiert
- „Agent“ führt Aktionen aus und bekommt Rückmeldung von Umgebung → z.B.: Belohnung bei Bewegung in die richtige Richtung.
- Verwendung: Robotics, Gaming-Bots...



Source: https://www.google.com/search?q=reinforcement+learning&source=lnms&tbo=isch&sa=X&ved=2ahUKEwiwfG11IrwAhXUgf0HHdJxAwQQ_AUoAноECAEQBA&biw=1920&bih=937#imgrc=ivmsD3gf5LO7WM

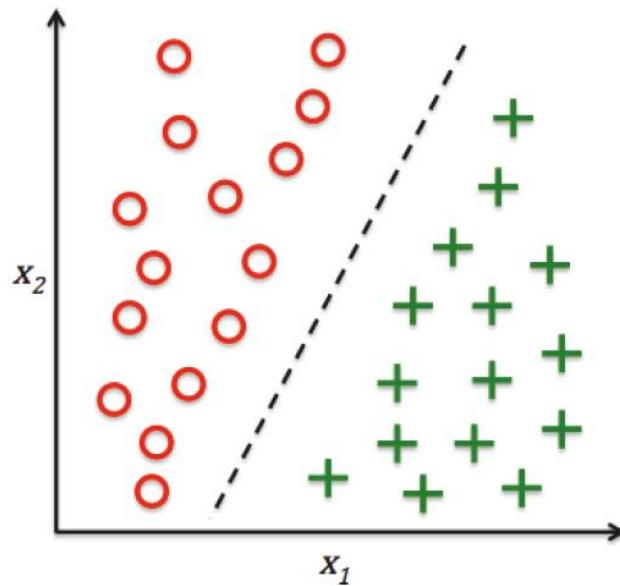


Supervised Learning

Klassifikation vs. Regression

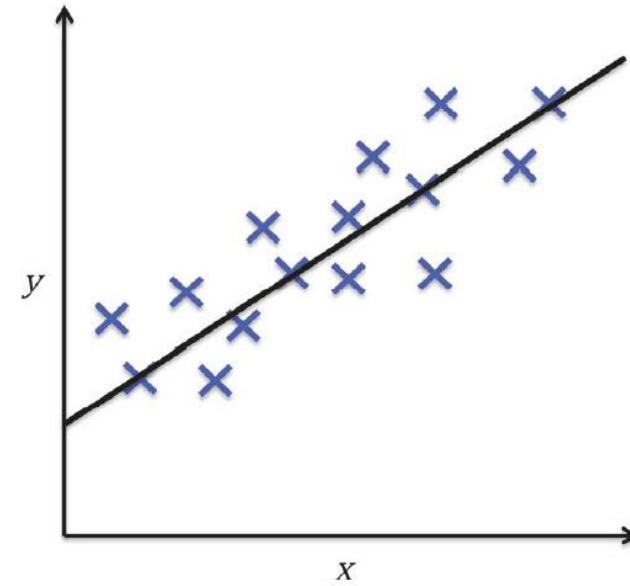
Klassifikation:

- Vorhersagen von Klassenzugehörigkeit kategorischer Werte durch Trennung des Entscheidungsraums



Regression:

- Vorhersagen von numerischen Werten durch Aufstellung einer Funktion, welche eine Art Approximation darstellt

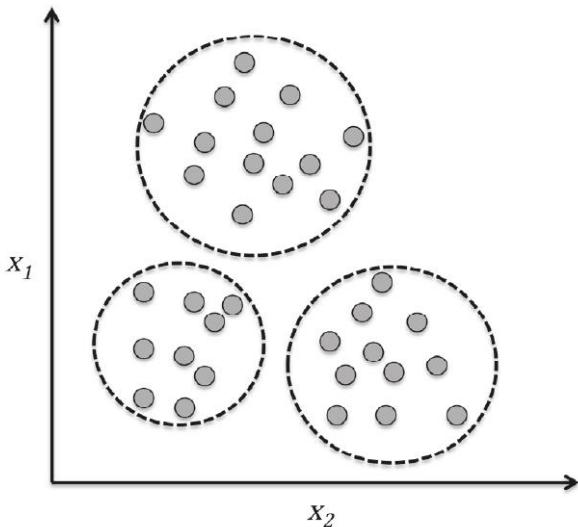


Unsupervised Learning

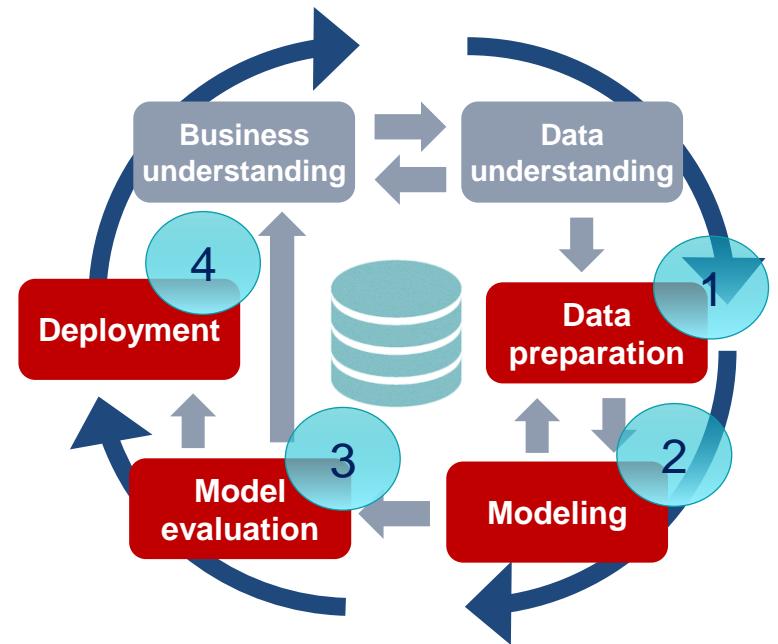
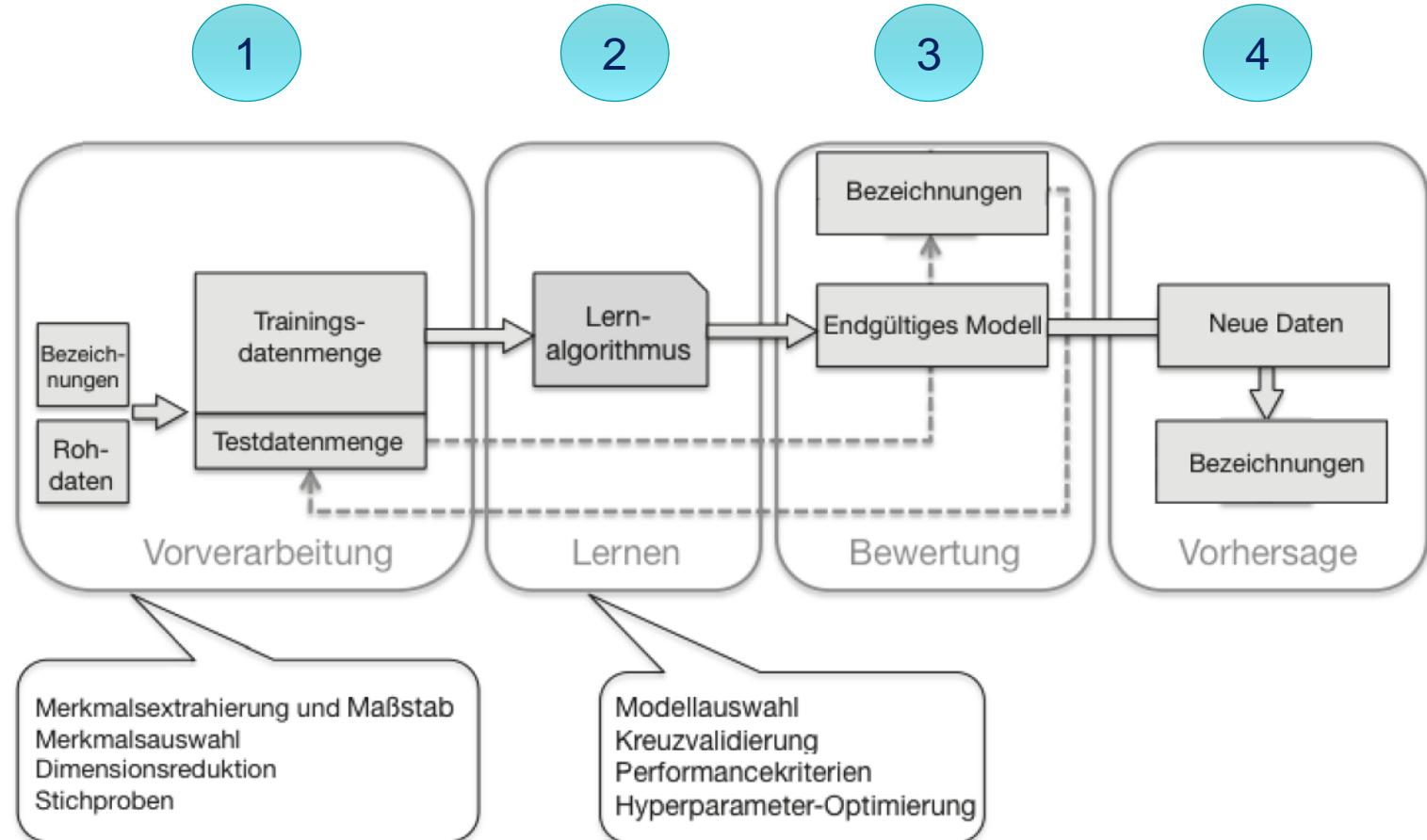
Clustering

Clustering:

- Aufteilung von nicht gelabelten Daten in ähnliche Gruppen



Machine Learning Pipeline



Machine-Learning Methoden & Konzepte

Erstes Machine Learning Modell mit Python

Machine Learning Methoden

- Lineare Regression
- Multiple Lineare Regression
- Polynomiale Regression
- Decision Trees
- Support Vector Machines
- Logistic Regression
- K-Means
- DBSCAN

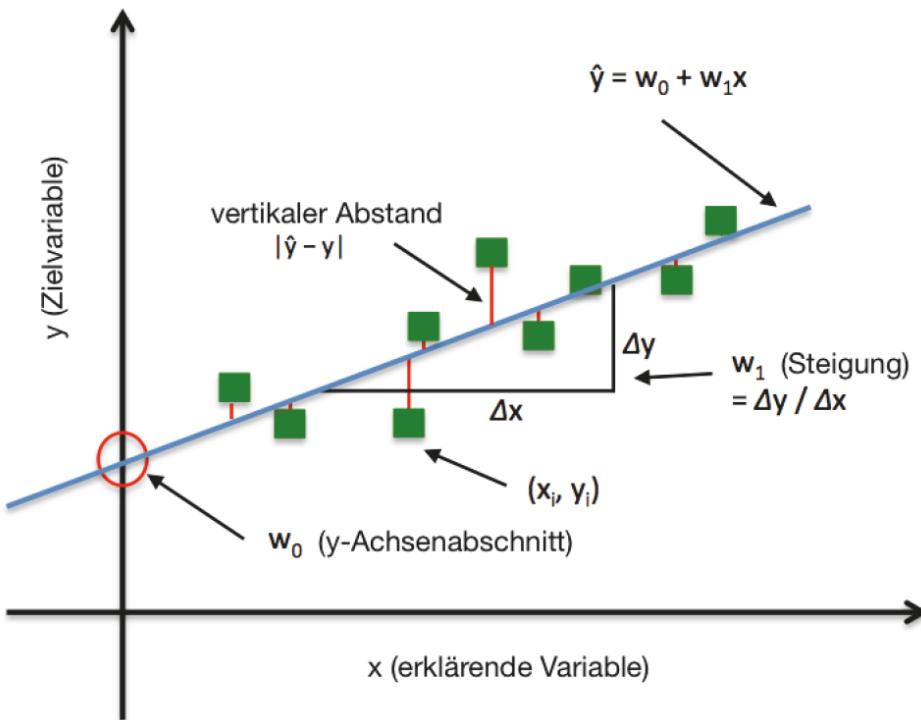


Regression

Einfache Lineare Regression

- Dient zur Vorhersage eines Zielwerts y anhand eines Merkmals/Features x

$$y = w_0 + w_1 x$$



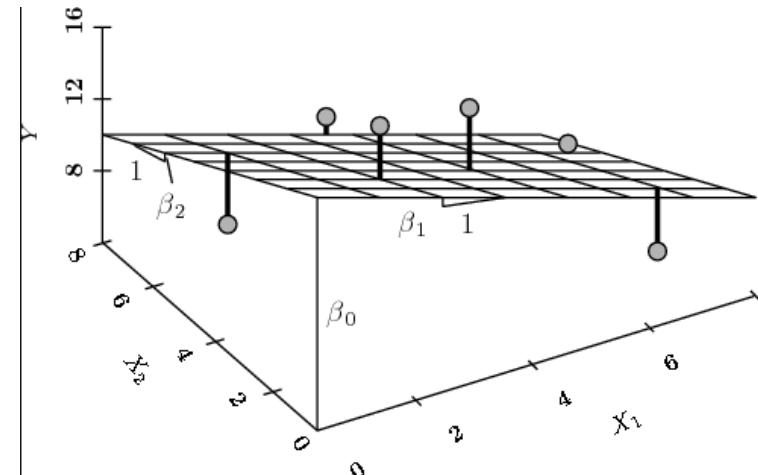
Regression

Multiple Lineare Regression

- Es können mehrere Features zur Vorhersage der Zielvariable hinzugezogen werden

$$y = w_0x_0 + w_1x_1 + \dots + w_mx_m = \sum_{i=0}^m w_i x_i = w^T x$$

- Jedes Merkmal x hat seinen eigenen Koeffizienten und somit auch einen individuellen Einfluss auf die Zielvariable
- Aus Gerade wird eine Ebene:



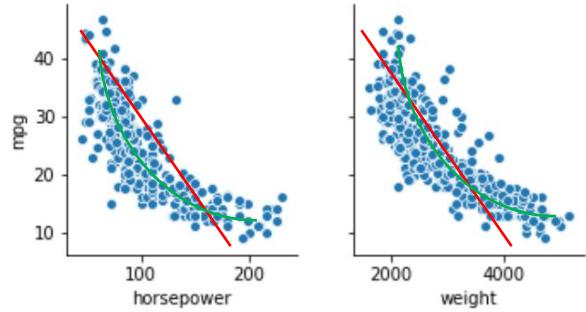
Source: https://www.google.com/search?q=Regressionsebene&source=lnms&tbo=isch&sa=X&ved=2ahUKEwiBn4GPv43wAhU4hf0HHVw6Av0Q_AUoA3oECAEQBQ&biw=1920&bih=937#imgrc=ICJ4cfQrU2z-FM



Regression

Polynomiale Regression

- Was wenn die Beziehung zwischen Target „y“ und unseren Features „komplexer“ ist?



- Polynomiale Regression erlaubt es die optimalen Koeffizienten für ein polynomial n-ten Grades zu ermitteln:
- $\hat{y} = w_0 + w_1x + w_2x^2 + \dots + w_nx^n$

Regression

Polynomiale Regression

$$\hat{y} = w_0 + w_1x + w_2x^2 + \dots + w_nx^n$$

- Um die Koeffizienten w zu ermitteln, können wir die Werte für x^i berechnen und diese als normale numerische Features behandeln.
 - Es müssen keine weiteren Änderungen vorgenommen werden !

	horsepower	weight	horsepower ²	weight ²
0	130.0	3504	16900.0	12278016
1	165.0	3693	27225.0	13638249
2	150.0	3436	22500.0	11806096
3	150.0	3433	22500.0	11785489
4	140.0	3449	19600.0	11895601

- Bei der Anwendung einer „normalen“ Linearen Regression werden diese Features dann verwendet und es werden bessere Resultate erzielt (falls eine nicht-lineare Beziehung besteht)

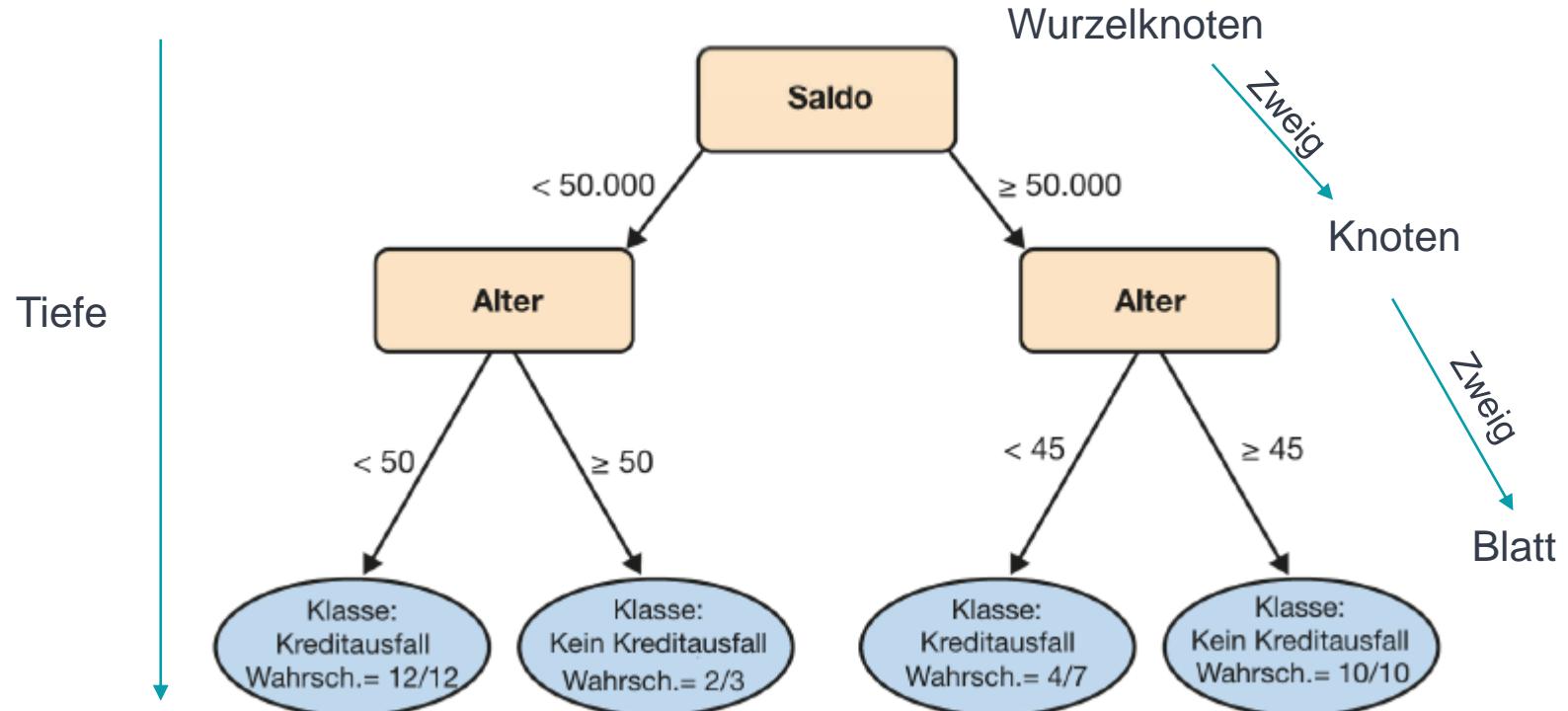


Regression mit Python

Klassifikation

Entscheidungsbaum

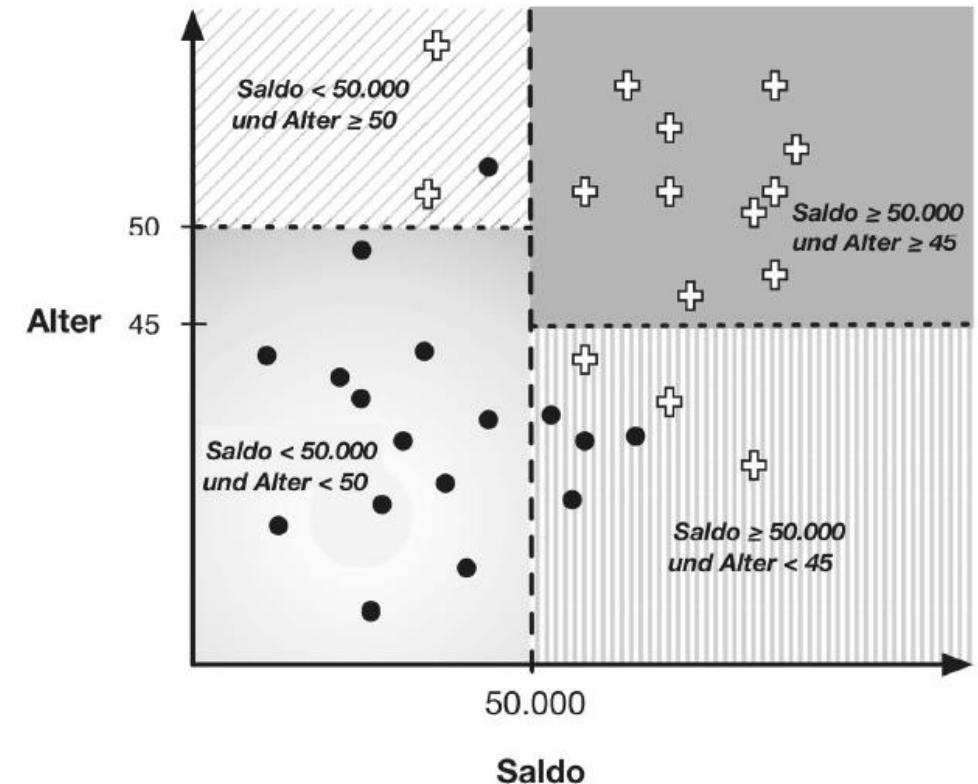
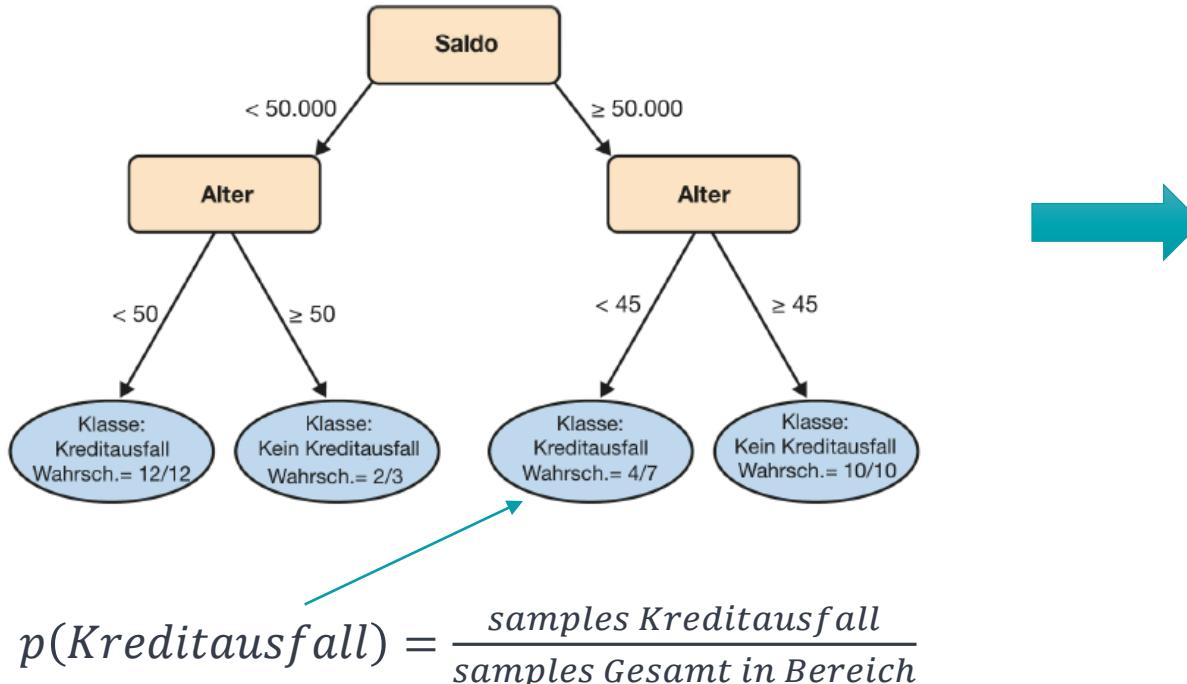
- Entscheidungsbaum zur Beurteilung der Kreditwürdigkeit von Kunden anhand von Merkmal Saldo und Alter



Klassifikation

Entscheidungsbaum

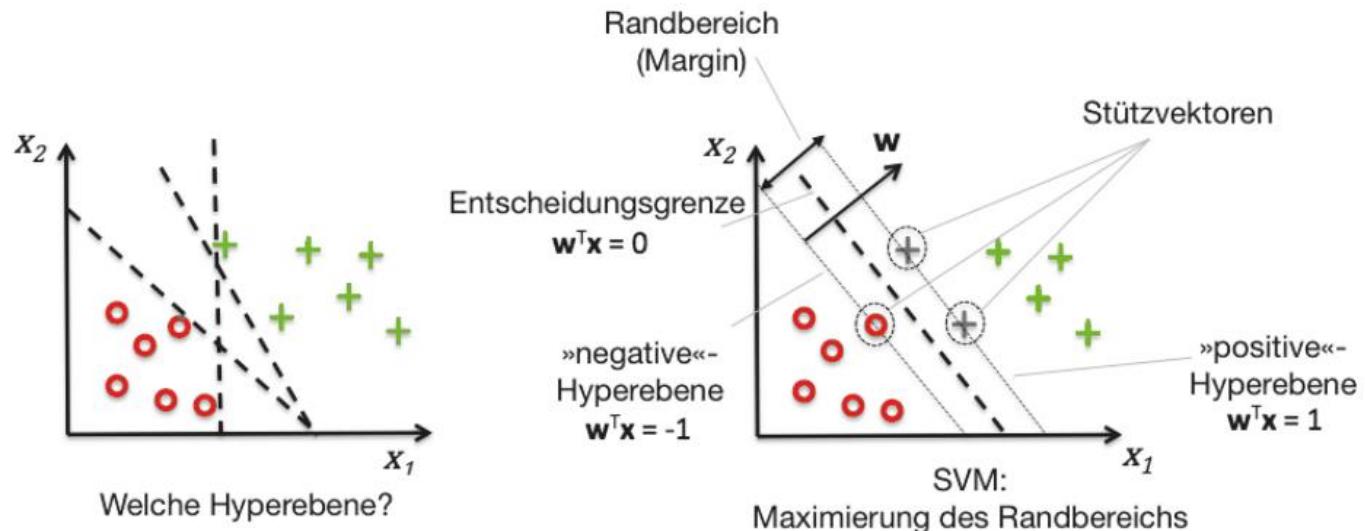
- Entscheidungsgrenzen sind keine Geraden oder Ebenen sondern Stufen/Bereiche



Klassifikation

Support Vector Machines

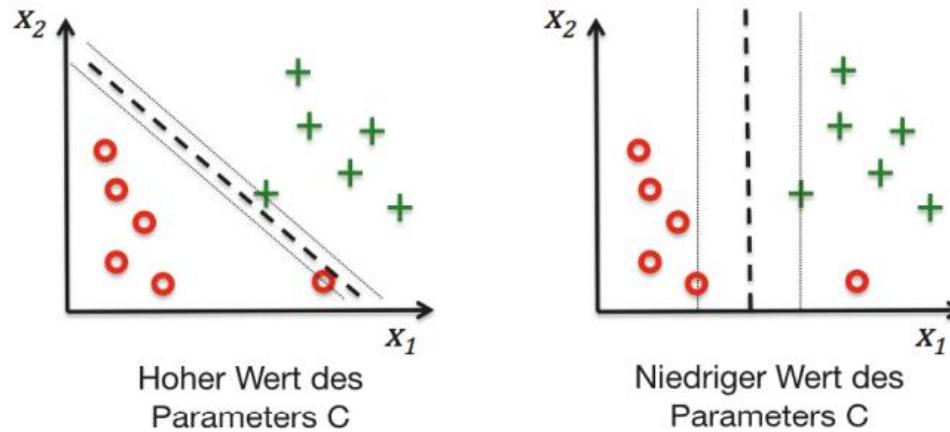
- Benutzt „Support Vectors“ um Entscheidungsgrenze zu ziehen
- Ziel ist es einen möglichst breiten Raum zwischen den Daten der verschiedenen Klassen zu bilden (Margin)



Klassifikation

Support Vector Machines für lineare Probleme

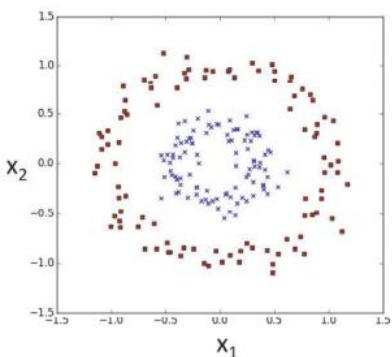
- Schlupfvariable wird eingeführt für Daten mit verschmelzenden Entscheidungsgrenzen
- Parameter C bestimmt den Einfluss der Schlupfvariablen, umso größer C, umso höher die Bestrafung für Fehlklassifizierungen (Regularisierung).



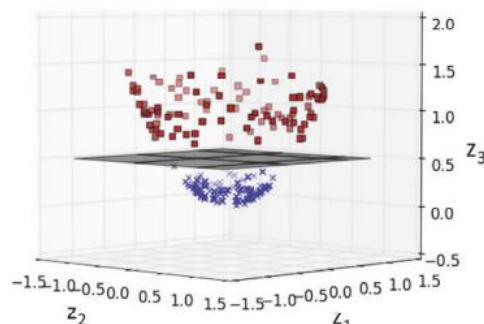
Klassifikation

Support Vector Machines für nicht-lineare Probleme

- Nicht-lineare Probleme lassen sich mit dem sogenannten „Kernel“-Trick lösen
- Die ursprüngliche Funktion wird dabei in einen höherdimensionalen Raum überführt.

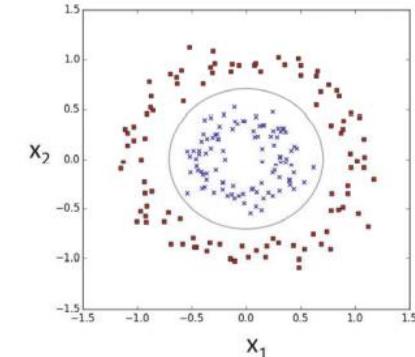


ϕ



ϕ^{-1}

Dimension wird hinzugefügt und
Entscheidungsgrenze gefittet



Dimension wird wieder weggenommen,
Entscheidungsgrenze bleibt dabei
erhalten



Klassifikation

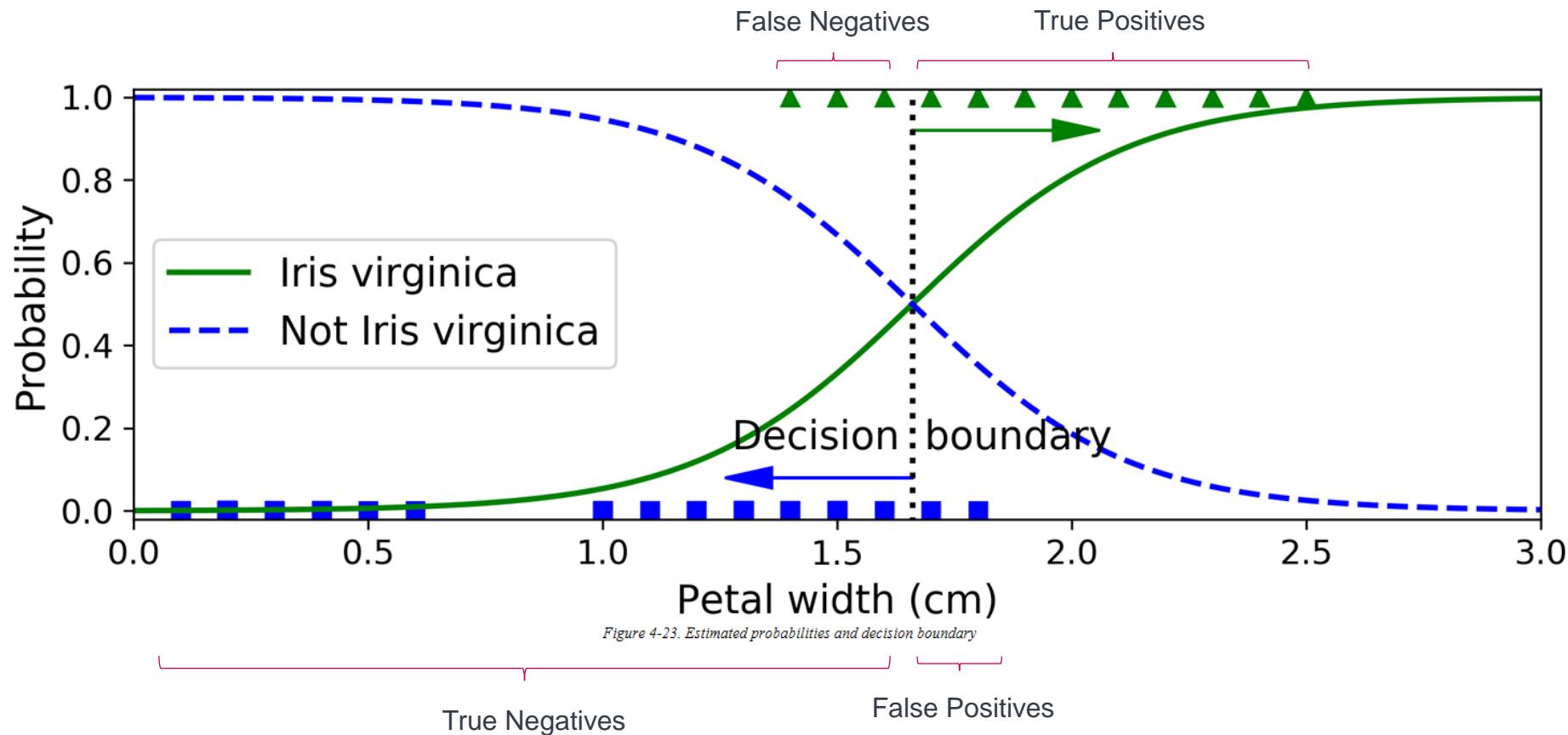
Logistic Regression

- Klassifizierung anhand Regressionsalgorithmus → Ermittlung der Wahrscheinlichkeit für Klassenzugehörigkeit
- Wenn Wahrscheinlichkeit für Zugehörigkeit zu einer Klasse größer als 50% wird positive Klasse ausgegeben, sonst negative
- Ergebnis ist eine Sigmoid Funktion mit Werten zwischen 0 und 1
- Binärer Klassifizierer



Klassifikation

Logistic Regression



Klassifikation mit Python

Clustering

K-Means

- Prototype-based Clustering
 - Jeder Cluster wird von einem zentralen Datenpunkt („centroid“) repräsentiert
- Leichte Implementierung und gute Effizienz
- Anzahl der Cluster(k) müssen vorher definiert werden

→ Beurteilung der Güte beispielsweise mit „Ellenbogenkriterium“ oder Silhouettendiagramm



Clustering

K-Means

Algorithmus:

1. Auswahl aus Objekten k Zentroide als anfängliche Clusterzentren.
2. Alle Objekte dem nächsten Zentroiden zuweisen.
3. Neuberechnung des Zentroiden mit den aus Schritt 2 zugewiesenen Objekten.
4. Wiederholung von 2+3, bis sich die Zuordnung nicht mehr ändert (entweder Schwellenwert oder maximale Iterationen werden vorgegeben).

Aber was ist unser Ähnlichkeitsmaß ?



Clustering

K-Means

Euklidische Distanz:

- Datenobjekte die sich ähnlich sind nahe beieinander platziert

Formel: $d(x, y)^2 = \sum_{j=1}^m (x_j - y_j)^2 = \|x - y\|_2^2$

- Durch euklidische Distanz kann k-Means Algorithmus als Optimierungsaufgabe formuliert werden
- Summe der quadrierten Abweichungen innerhalb eines Clusters soll minimiert werden

Formel: SSE = $\sum_{i=1}^n \sum_{j=1}^k w^{(i,j)} \|x^{(i)} - \mu^{(j)}\|_2^2$

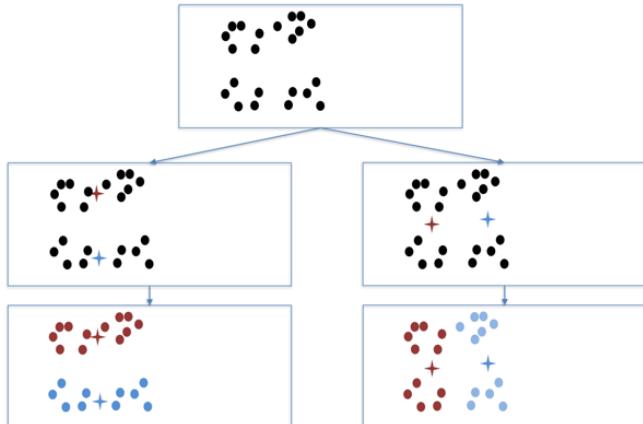


Clustering

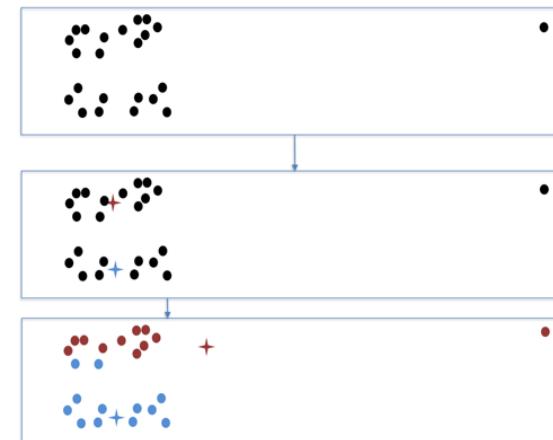
K-Means

- Pro's:
 - Leicht zu implementieren
 - Äußerst effektiv
- Con's
 - Initiale Platzierung der Zentroide ist wichtig
 - Einfluss von Outliern

Platzierung:



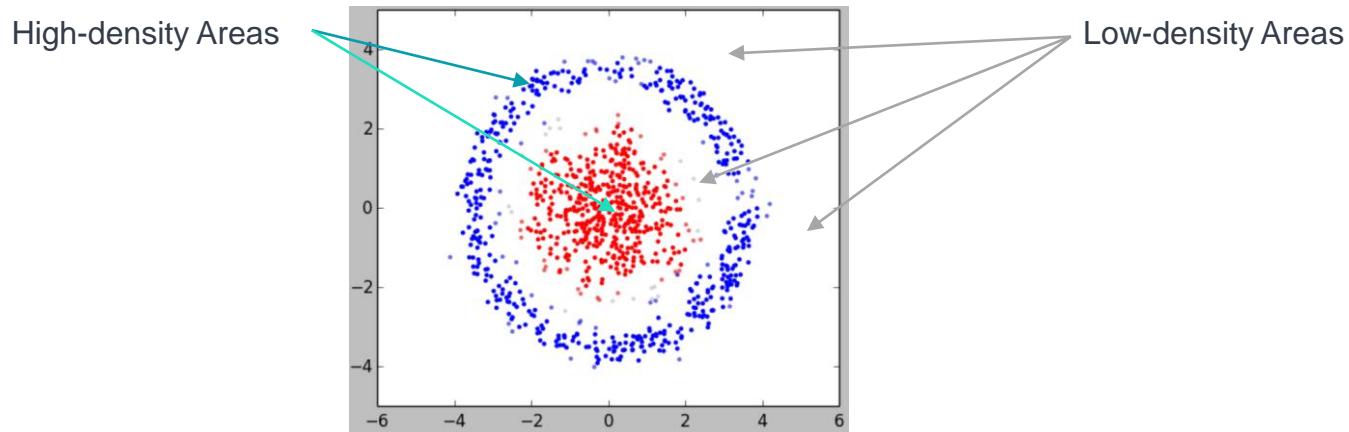
Outlier:



Clustering

DBSCAN

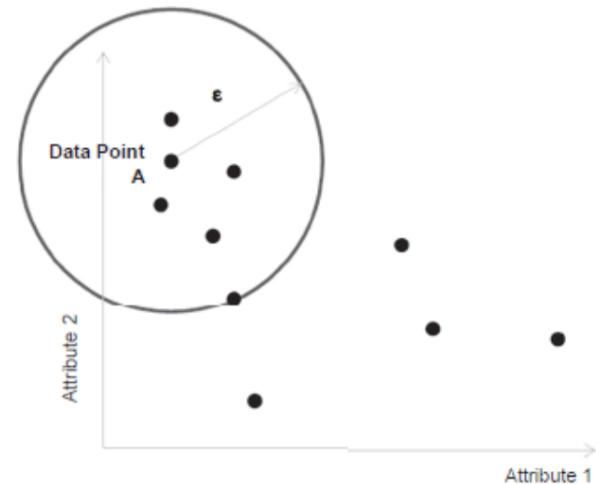
- In vielen Anwendungsfällen sind die Anzahl an Clustern vorher nicht bekannt
→ Problem bei k-Means
- Außerdem können komplexe Datenstrukturen für k-Means ein Problem werden.
- Für solche Situationen, kann man auf Density-Based Clustering zurückgreifen:



Clustering

DBSCAN

- Dichte messen
 - Dichte = Anzahl von Punkten innerhalb eines gewissen Bereichs mit dem Radius ε (epsilon)
 - Beispiel: Dichte um den Datenpunkt A ist 6
- Basierend auf dieser Idee identifiziert der DBSCAN Algorithmus Dichteregionen mit den Hyperparametern radius (ε) und den „minimum number of points“



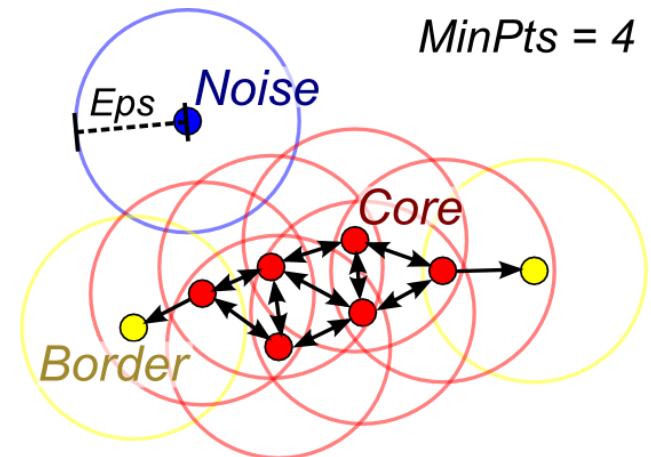
Clustering

DBSCAN

DBSCAN

(Density-Based Spatial Clustering of Applications with Noise)

1. Berechnen der Density für jeden Datenpunkt abhängig von epsilon → Density > MinPts = High-Density Area
2. Wenn Datenpunkt = High-Density Area → Core Point
Falls Datenpunkt ≠ High-Density Area, aber Core Point innerhalb Radius (epsilon) → Border Point
3. Falls Datenpunkt ≠ High-Density Area und kein Core Point innerhalb Radius (epsilon) → Noise



Clustering

DBSCAN

Probleme

- Wenn ein Dataset Regionen mit variierender Dichte enthält kann DBSCAN diese nicht identifizieren

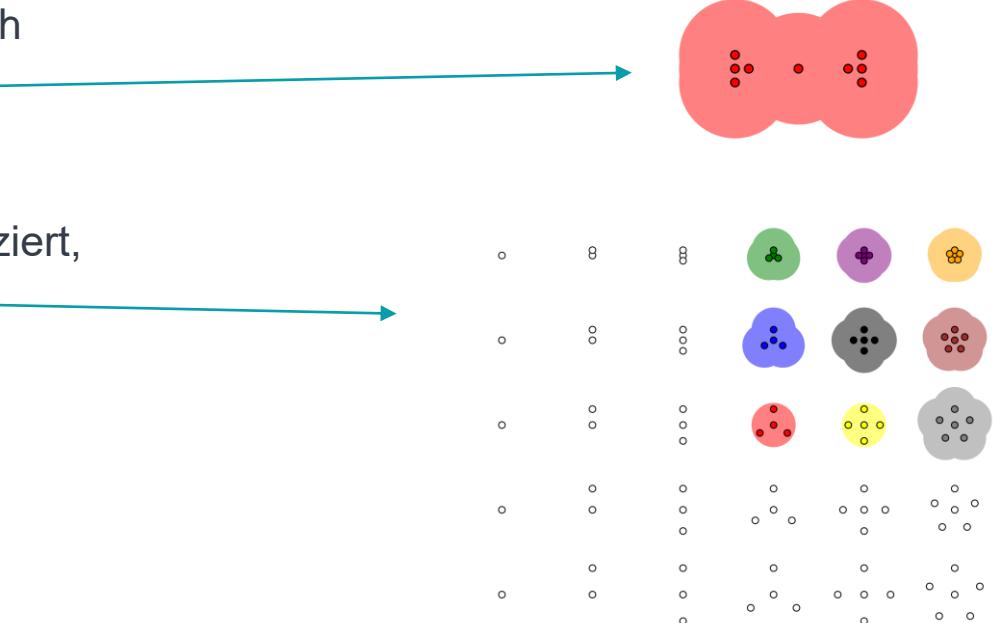
– Entweder setzen wir MinPoints zu klein/ epsilon zu hoch

→ worst case: Es wird nur 1 Cluster identifiziert

– Oder wir setzen MinPoints zu hoch / epsilon zu klein

→ worst case: Alle Punkte werden als „Noise“ identifiziert, low-density areas werden übergangen.

→ K-Means Clustering ist passender in diesem Fall



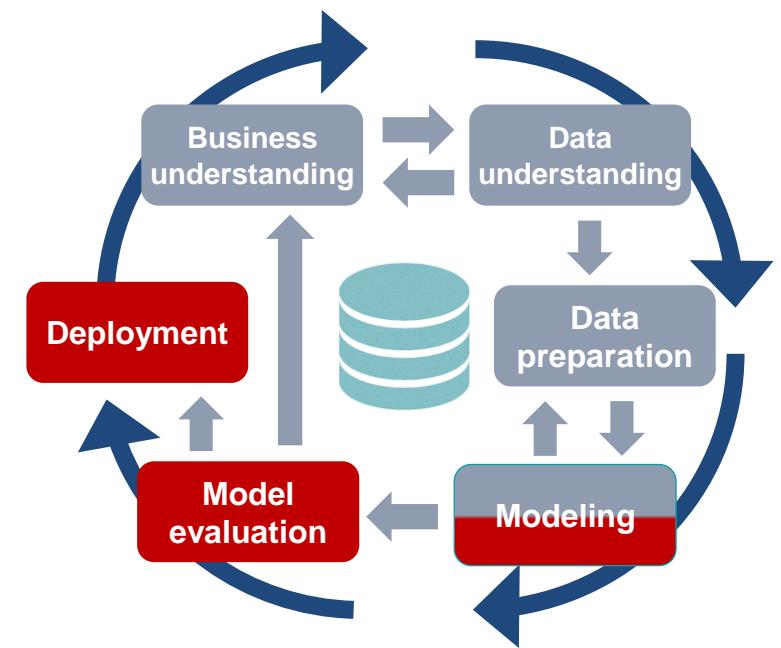
Clustering mit Python

Evaluierung

Modellevaluierung

- Wichtiger Schritt nach/während dem Modelltraining bei dem die Güte des Modells anhand verschiedener Metriken bewertet wird.
- Ist teilweise in Modelltraining eingebunden
- Kann sowohl rein numerisch, als auch graphisch erfolgen
- Kann automatisiert werden (nicht immer zu empfehlen)

Ziel: erfüllt das Modell die Problemstellung?



Methoden

Train-/Test-Split

- Zurückhalten von Daten um „neue Daten“ zu simulieren, welche nicht zur Modellbildung verwendet wurden
- Ziel:
 - Verallgemeinerungsfähigkeit von Modell beurteilen
 - Overfitting erkennen
- Nachteil:
 - Testdaten müssen von Trainingsdaten abgezogen werden (schlecht bei niedriger Datenmenge)
 - Ausschneiden von „wichtigen Mustern“ kann Modellperformance beeinflussen



Methoden

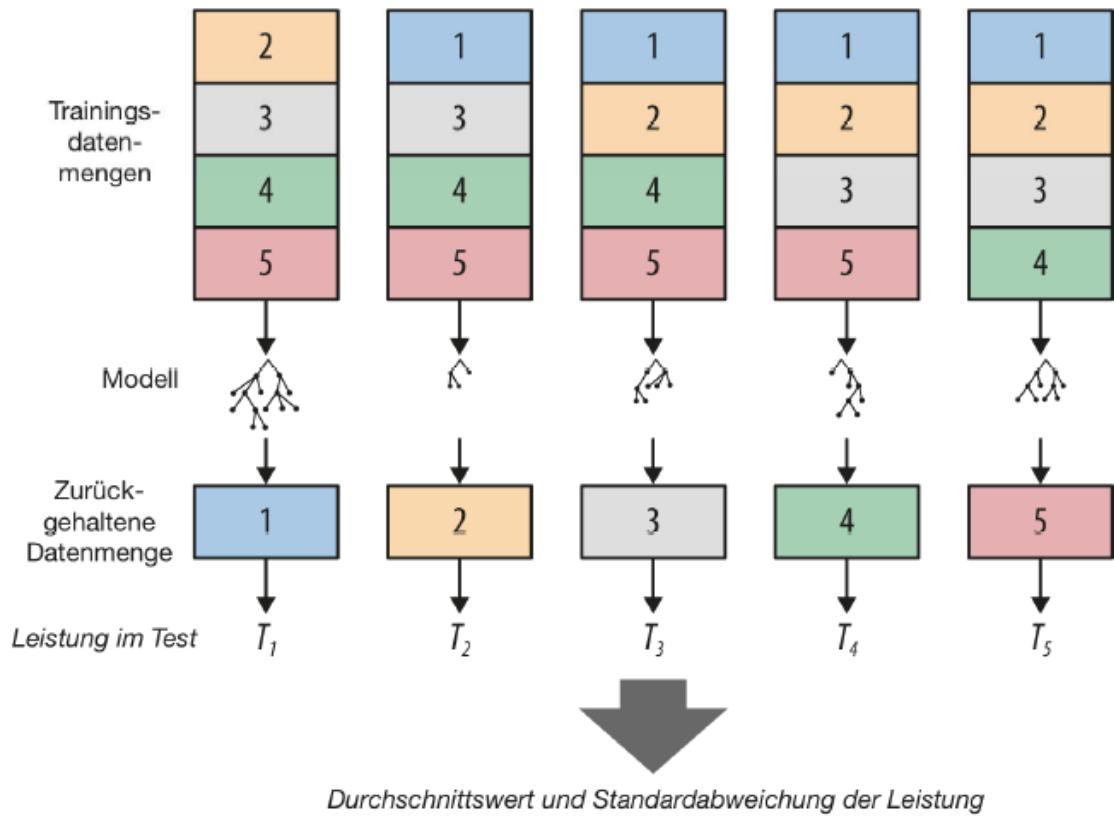
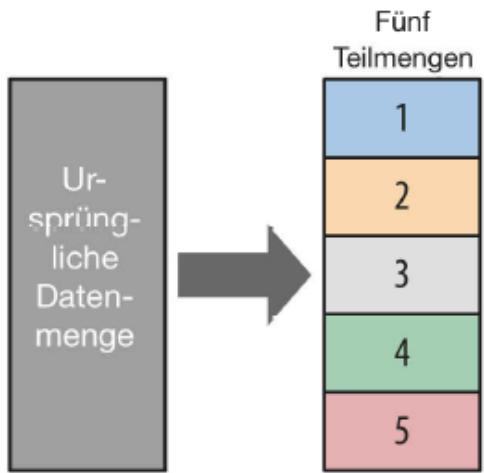
K-fold Cross Validation

- Mehrere Kombinationen von Trainings- und Testdaten werden geprüft
 - Möglichkeit für „unglückliche“ Auswahl von Trainings-/Testdaten wird geprüft
 - Mehrere Werte fließen in finale Beurteilung ein
 - Durchschnittswert und Standardabweichungen bestimmen finales Ergebnis
- Vorgehen
 - Gesamte Datenmenge wird nach Zufallsprinzip in k gleich große Teilmengen unterteilt (meist 5 oder 10)
 - $k-1$ Teilmengen werden zum Training benutzt und 1 Teilmenge zum testen
 - Bei 5-facher Kreuzvalidierung werden 5 verschiedene Kombinationen geprüft



Methoden

K-fold Cross Validation



Evaluationsmetriken

Accuracy

- Einfach messbar
- Fasst Leistung eines Classifiers in einer Zahl zusammen
- Anteil korrekter Entscheidungen
- Problem → oft zu einfach für Business Probleme

$$\text{Korrektklassifizierungsrate} = \frac{\text{Anzahl korrekter Entscheidungen}}{\text{Gesamtzahl der Entscheidungen}}$$



Confusion Matrix

False Positives/ False Negatives

- Klasse 0 (negativ) meist erwünschtes Ergebnis
- Klasse 1 (positiv) meist unerwünschtes Ergebnis

Bsp.: Sars-Cov-2

- Klasse 1 = positiv = erkrankt
- Klasse 0 = negativ = nicht erkrankt

Konsequenz

- Positive Klassen kommen seltener vor
- Anzahl Fehlklassifizierung bei Klasse negativ (False positive) häufiger
- Auswirkungen von Fehlklassifizierung der Klasse positiv (False Negative) jedoch größer
 - Bsp.: Person wird als nicht erkrankt klassifiziert obwohl erkrankt (False Negative)



Confusion Matrix

- Oftmals muss ein Blick auf die falsch-klassifizierten Daten geworfen werden um sicherzustellen ob Business Problem gelöst wurde
- Beispiel Betrugserkennung:
 - Klassen oft unausgewogen: auf 1 Betrugsfall kommen evtl. mehr als 999 „nicht“-Betrugsfälle
 - Keine False Positives = Accuracy von 99,9%

→ Business Problem nicht gelöst, da Betrugsfälle nicht erkannt werden

	p	n
Predicted p	Richtig Positive	Falsch Positive
Predicted n	Falsch Negative	Richtig Negative



F1 Score

- Misst den Anteil von False Positives und False Negatives in einem Modell
- Score von 1 indiziert perfektes Modell und Score von 0 schlechtestes Modell
- Besser als Accuracy, da “Fehlalarm” minimiert wird.

$$\begin{aligned} F_1 &= \frac{2}{\frac{1}{\text{recall}} \times \frac{1}{\text{precision}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \\ &= \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})} \end{aligned}$$

	p	n
J	56	7
N	5	42



Regression

SSE, MSE, RMSE

- SSE: Summe der quadrierten Abstände zur Evaluierung wie gut die Regressionsgerade die Daten approximiert.

$$SSE = \sum_{i=1}^n (x_i - \hat{x}_i)^2$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$$

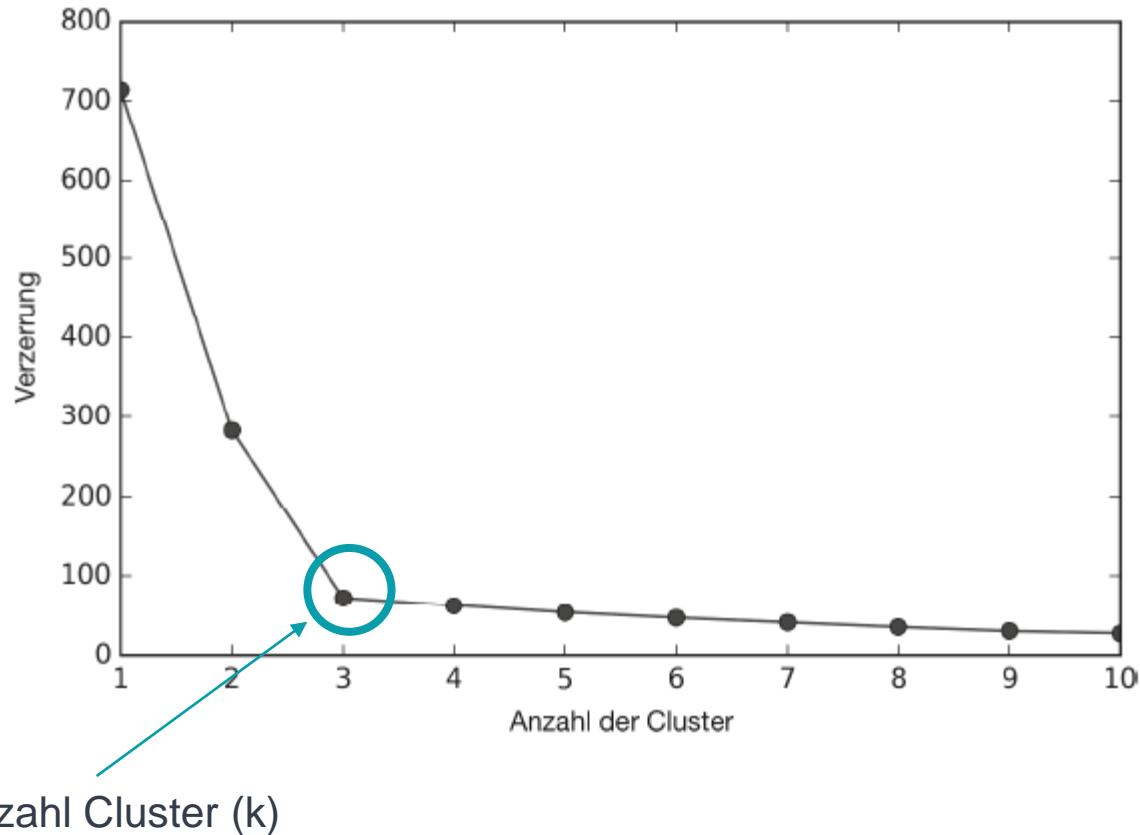
$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2}$$



Clustering

Elbow-Method

Beispiel: Bewertung der Güte eines K-Means Clusterings. Was ist das optimale k?



Clustering

Davies-Bouldin-Score

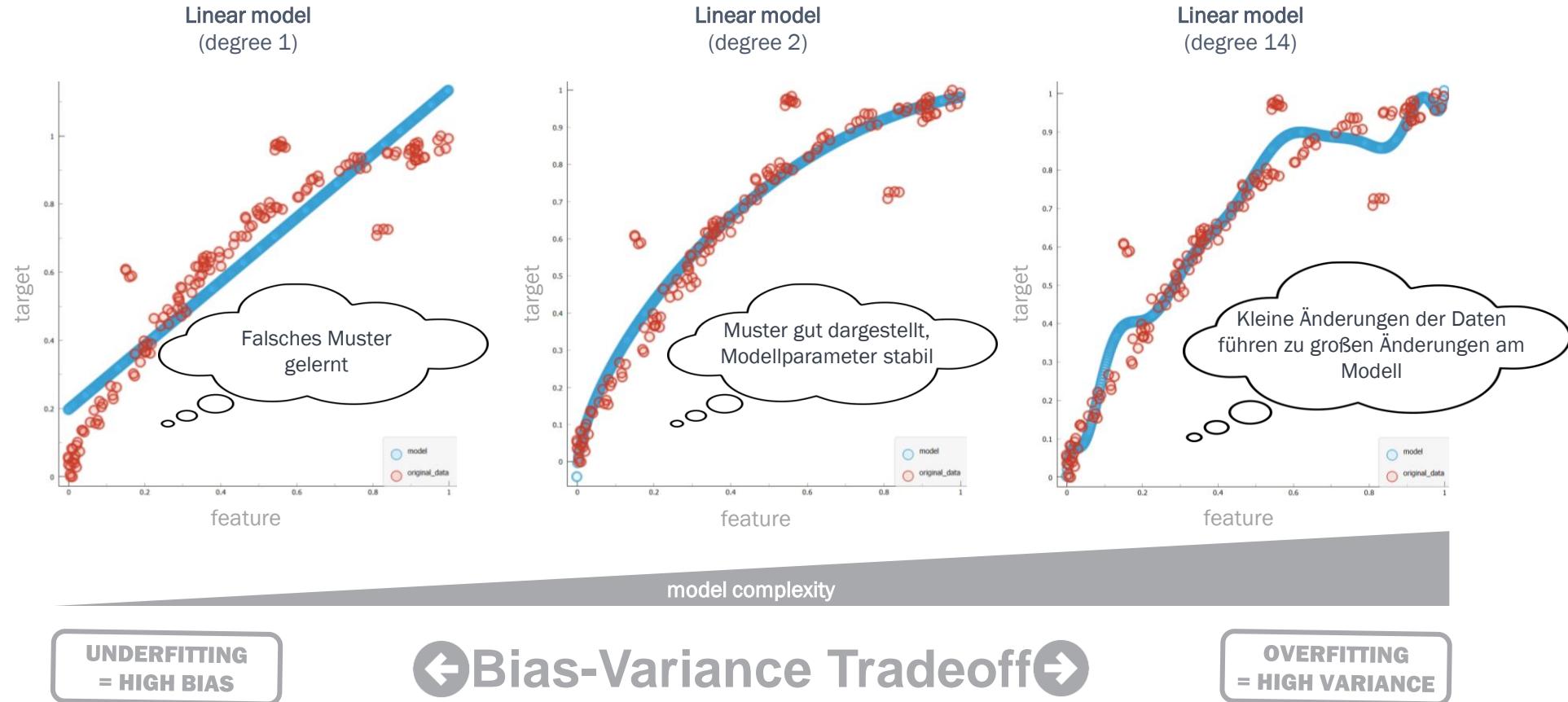
- Bewertet wie ähnlich sich Datenpunkte des gleichen Clusters durchschnittlich sind
- Score hat minimum Value von 0 aber maximum value geht über 1
- Kleine Werte sagen ein gutes Clustering voraus



Evaluierung mit Python

Bias-Variance Tradeoff

Bias-Variance Tradeoff



Bias-Variance Tradeoff

- Low Bias, Low Variance (so gut wie unmöglich):
 - Modell hat informative/relevante Features
 - Modellparameter sind stabil
 - Modell performt gut auf unbekannten Daten (Generalisierung)
- Low Bias, High Variance:
 - Overfitting
 - Sprunghafte Modellparameter bei der Verwendung unterschiedlicher Trainingssets (inkonsistent)
 - Modell lernt eventuell zu viel Noise oder Daten auswendig (keine Generalisierung möglich)
- High Bias, Low Variance:
 - Underfitting
 - Vorhersagen sind konsistent, aber inakkurat im Durchschnitt
- High Bias, High Variance:
 - Modell hat weder informative Features, noch ist es richtig dimensioniert.
 - Vorhersagen sind inkonsistent und inakkurat im Durchschnitt
 - Keine Vorhersagen auf das Target möglich

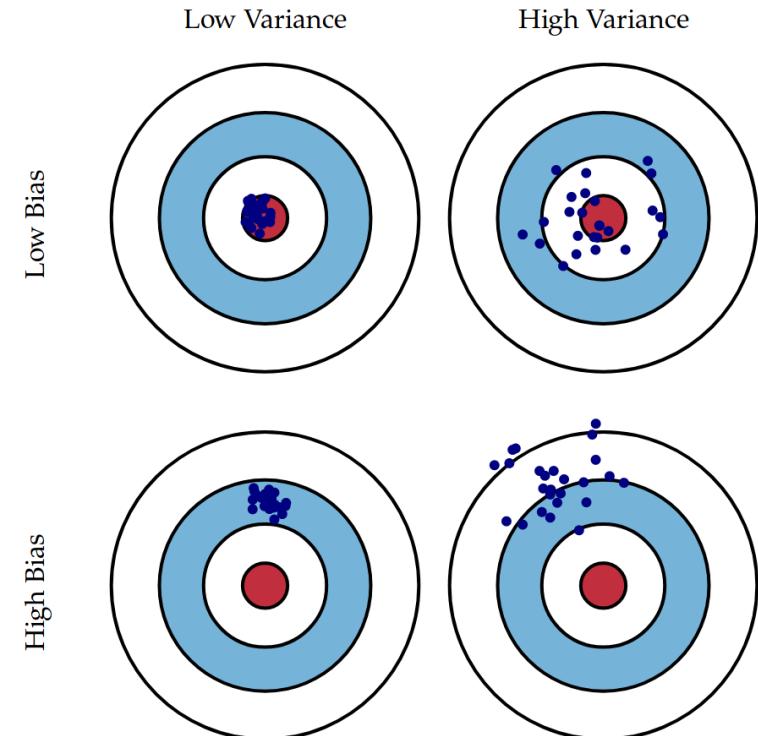


Fig. 1 Graphical illustration of bias and variance.

Sources: Fortmann-Roe, Scott. 2012. "Understanding the Bias-Variance Tradeoff."



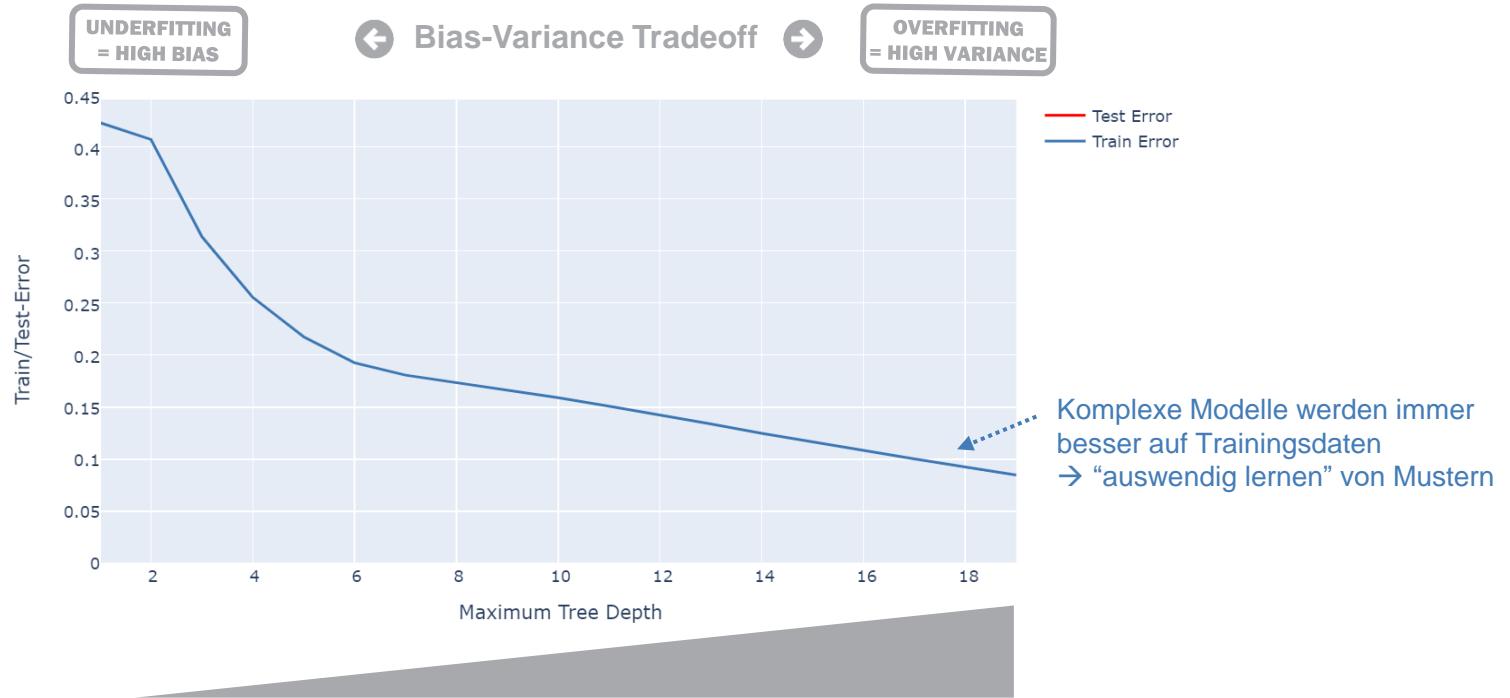
Live Demo Over- / Underfitting



Bias- Variance Tradeoff

Train/Test-Error

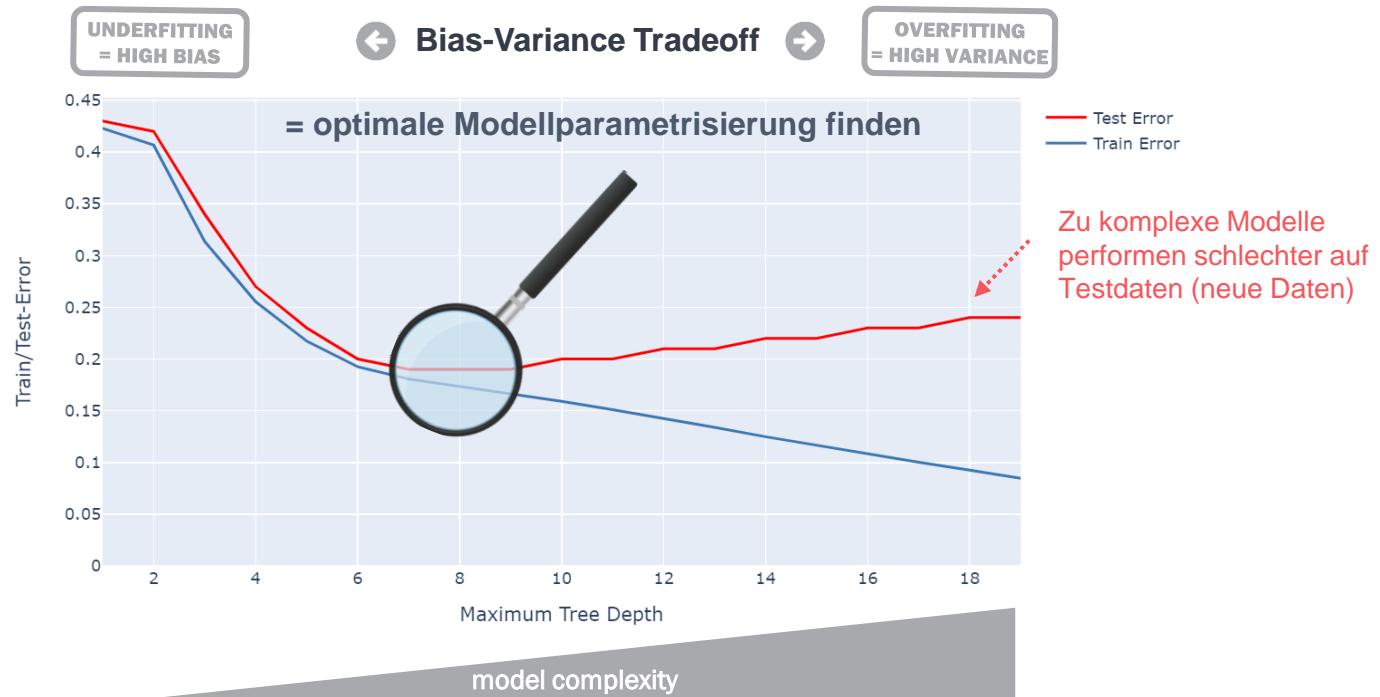
- Eine numerische Betrachtung des Bias-Variance Tradeoffs
 - Variieren von Modellparametern & messen des train-test-errors



Bias- Variance Tradeoff

Train/ Test- Error

- Eine numerische Betrachtung des Bias-Variance Tradeoffs
 - Variieren von Modellparametern & messen des train-test-errors



Anomaly Detection

Anomaly Detection

Definition: „Something that deviates from what is standard, normal, or expected“ -*Oxford Dictionary*

- Anomalie = Outlier
- Abweichung bekannter KPI's von Normwert (Threshold value)
- Kann sowohl mit Supervised- als auch Unsupervised Learning gelöst werden (auch hybrider Ansatz möglich)



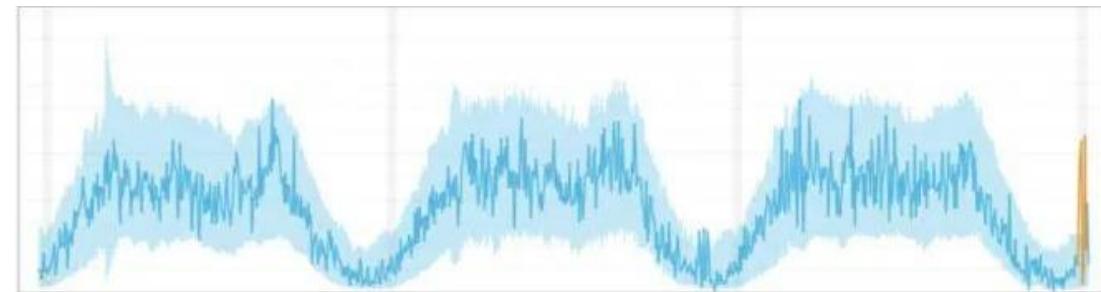
Anomaly Detection

Arten von Outliern

- Globale Outlier
 - deutlich zu erkennen
 - weicht von allen Normwerten ab
 - meist einzelne Werte



- Kontextuelle Outlier
 - Weicht von Normwert innerhalb eines Kontexts ab
z.B.: Saisonalität, Ereignisse etc.



Anomaly Detection

Arten von Outliern

- Kollektive Outlier
 - Mehrere Datenpunkte nacheinander weisen Starke Anomalie zu allen bisher bekannten Datenpunkten auf.
Bsp.: Börsencrash, Defekter Sensor



Anomaly Detection

Anwendungsgebiete

- Sequence data:
 - Web-logs
 - Kundentransaktionen
 - Abweichungen von Zwischensequenzen von Aminosäuren
 - Netzwerkaktivitäten
- Spatiotemporal:
 - Verkehrsdaten (unnatürliches Verhalten von bewegten Objekten)
 - Meteorologiedaten (Wettermuster)



Anomaly Detection

Anwendungsgebiete

- Graphdaten:
 - Unnatürliche Komponenten in einer Datenbank von chemischen Komponenten
- Zeitreihen:
 - Erkennen von abrupten Abweichungen oder Neuerungen



Anomaly Detection

Nearest Neighbour for Anomaly Detection

- Einer der simpelsten Klassifizierungsalgorithmen (Supervised und Unsupervised)
- Neue Datenpunkte werden anhand von k-nächsten Nachbarn klassifiziert
- Vorteile:
 - Es muss kein Modell trainiert werden
 - Auch mit wenigen Daten kann gearbeitet werden
- Hypothese für Anomaly Detection:
 - Anomalien sind Datenpunkte, welche keine nächsten Nachbarn besitzen oder deren durchschnittliche Distanz zu deren Klasse stark von der Norm abweicht.



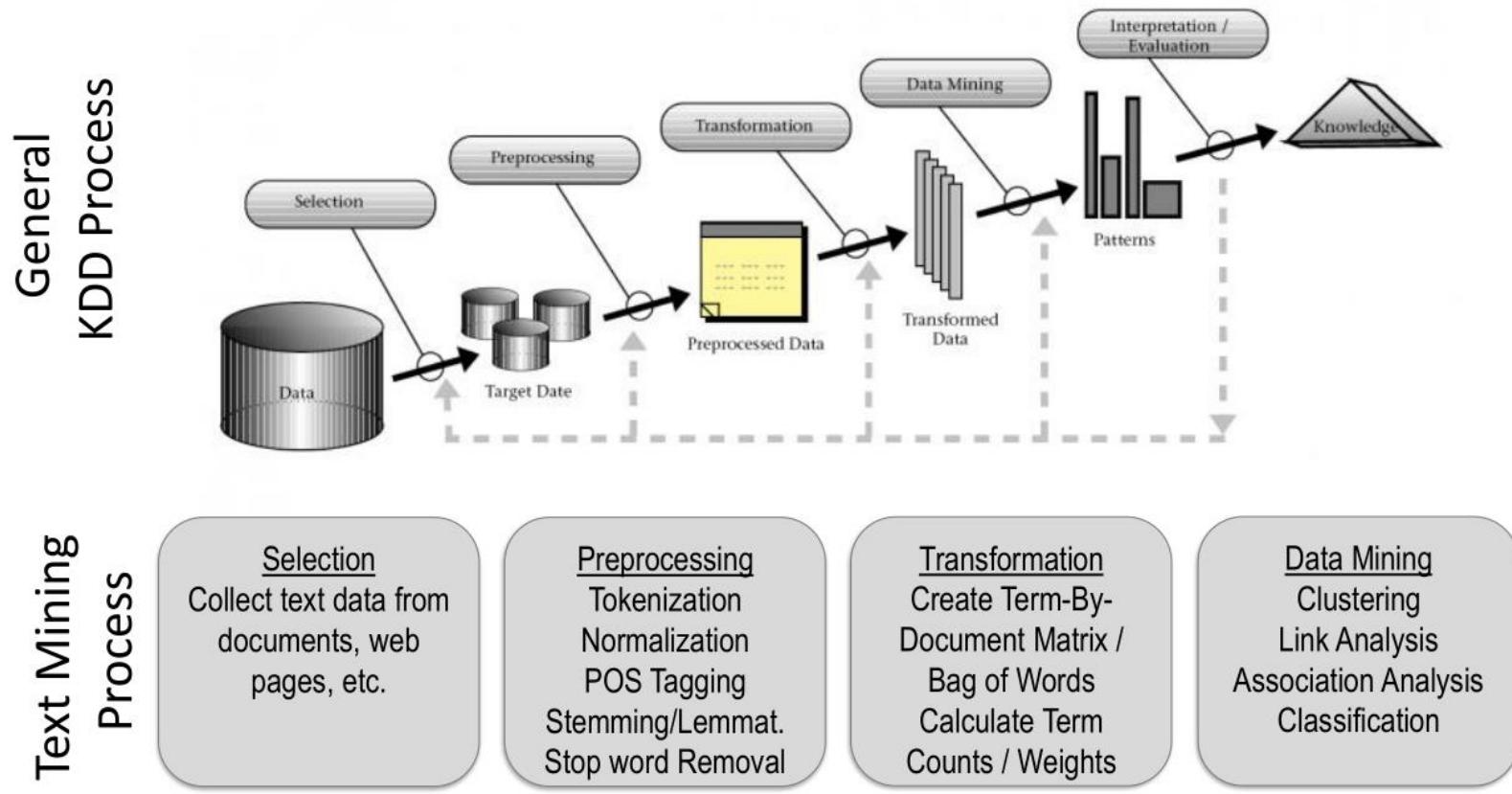
Anomaly Detection with Python

REST via Python

NLP

Wissen aus unstrukturierten Textdaten gewinnen

- Umgang mit Textdaten und Anwendung von Text Mining Methoden
 - aus unstrukturierten Daten strukturierte Daten machen und bekannte Methoden anwenden



Tokenization

- Aufteilen eines Textes in einzelne Wortabschnitte:

simples Beispiel:

Satz: „I want to go to the mall !“

```
= ['I', 'want', 'to', 'go', 'to', 'the', 'mall', '!']
```

- Mehrere Arten von Tokenizern verfügbar



Stemming/Lemmatisierung

Stemming mit Porter Stemmer

- Überführen von Tokens in eine Ursprungsform oder abschneiden von Pref-/Suffixen

Beispiel:

Satz: „Runners like running and thus they run“

= ['runner', 'like', 'run', 'and', 'thu', 'they', 'run'] → nicht alle Transformationen machen Sinn

- Weiterer Ansatz wäre beispielsweise das Nachschlagen der Ursprungsform in Lexikon



Bag-of-Words Modell

- Aufbau eines Vokabulars, bei dem jedes Wort einen Index zugeordnet bekommt
- Auf Index aufbauend werden One-Hot-Codierte Vektoren gebildet, welche Word-Frequency in jedem Sample zählen
- Ergebnis ist Merkmalsvektor, welcher für Klassifikation benutzt werden kann → Achtung: Es müssen vorher einige Iterationen des Business- und Data Understandings durchlaufen werden um optimales Vokabular zu finden !



Bag-of-Words Modell

Beispiel

- „Today it is raining“
- „It is raining outside“
- „The weather today is not very good“

Vocabulary:

```
'today': 7, 'it': 2, 'is': 1, 'raining': 5, 'outside': 4, 'the': 6, 'weather': 9, 'not': 3, 'very': 8, 'good': 0
```

Merkmalsvektoren:

```
[[0 1 1 0 0 1 0 1 0 0]
 [0 1 1 0 1 1 0 0 0 0]
 [1 1 0 1 0 0 1 1 1 1]]
```



N-Gramme

Bigramme

- Ermöglicht Einbeziehen von Wortzusammenhängen

Vocabulary:

```
{'today it': 7, 'it is': 2, 'is raining': 1, 'raining outside': 4, 'the weather': 5, 'weather today': 9, 'today is': 6, 'is not': 0, 'not very': 3, 'very good': 8}
```

Merkmalsvektoren:

```
[[0 1 1 0 0 0 1 0 0]
 [0 1 1 0 1 0 0 0 0]
 [1 0 0 1 0 1 1 0 1]]]
```



Tf-idf-Maß zur Beurteilung von Wortrelevanz

Term frequency- inverse document frequency

- Worte die oft in Dokumenten auftauchen wie beispielsweise „is“, „that“ etc... aber keine Bedeutung für den eigentlichen Kontext haben, bekommen eine niedrigere Bewertung

Berechnung:

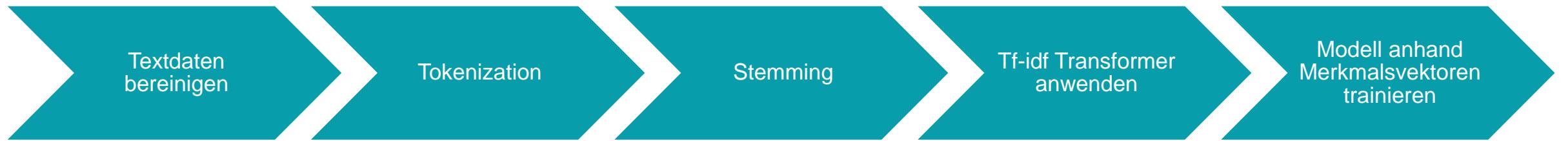
$$\text{idf}(t,d) = \log \frac{n_d}{1 + \text{df}(d,t)}$$

n_d = Gesamtzahl der Dokumente

$\text{df}(d,t)$ = Anzahl der Dokument d, welche Wort t enthalten (umso größer, umso weniger Gewichtung)



Beispelpipeline



NLP mit Python

Speichern und Laden von Modellen

Fragen und Feedbackrunde

- Haben Sie Fragen zu den behandelten Themen?
- Welche Themen haben Sie vermisst?
- Feedback an mich?

Feedback zum Training:

<https://ratings.gfu.cloud/form.html?h=70def7be8194563a4c26df8a22456b408dfe512141b18eb6fcf0c58bf10911f9&type=trainer>



Ihre Ansprechpartner



DAVID GOEBEL

MANAGER

+49 1573 6180503

david.goebel@exxeta.com



DR. GEORG WALTER

PRINCIPAL CONSULTANT

+49 1522 2690102

georg.walther@exxeta.com



WE ❤ DATA

data.exxeta.com

