ALGORITHMEN

Sortieralgorithmen

- **Bubblesort** $O(n^2), O(n^2), O(n)$
  Tauscht benachbarte Elemente
- **Insertionsort** $O(n^2), O(n^2), O(n)$
  Verschiebt/Fügt Element in richtigem Platz ein
- **Mergesort** $O(n \log n), O(n \log n), O(n \log n)$
  Teilt rekursiv in der Mitte
  und fügt aufsteigend zusammen
- **Quicksort** $O(n^2), O(n \log n), O(n \log n)$
  Teilt am Pivot Element und schiebt alle kleinere davor

Optimierungsalgorithmen

Anlagerungsverfahren (Bin-Packing):
First Fit: Objekt in ersten passenden Platz
Best Fit: Objekt in engsten passenden Platz
Next Fit: Objekt in zuletzt verwendeten Platz

Random Interchange:
Tauscht zufällig Elemente und prüft auf verbesserung

Kernighan-Lin:
Tauscht Knotenpaare und prüft auf verbesserung

Greedy:
Nimmt größten wert unter Zielwert und zieht diesen davon ab.
wiederholt bis nichts übrig bleibt.

Evolutionäre Algorithmen:
Biologisch Inspiriert (Selektion, Rekombination, Mutation)

Suchalgorithmen

- **Lineare Suche**
  Geht jedes Element durch, bis Lösung
- **Binäre Suche**
  Sucht von der Mitte in die richtige Richtung
- **Interpolationssuche**
  Interpolirt ungefähren Index mit
  erstem und letzten Wert vorsortierter Liste
- **Breitensuche** (Graphen)
  (kürzester Pfad) Besucht immer die nächsten Knoten
- **Tiefensuche** (Graphen)
  Folgt einem Pfad bis zum Ziel
- **Dijkstra-Algorithmus**
  Wählt Nachbar mit geringster Distanz zum Start
  und aktualisiert Distanzen aller Nachbarn. $O(n^2)$
- **Kritischer Pfad**
  Längster Pfad im Ablauf

Laufzeit

$O(n)$ Notation

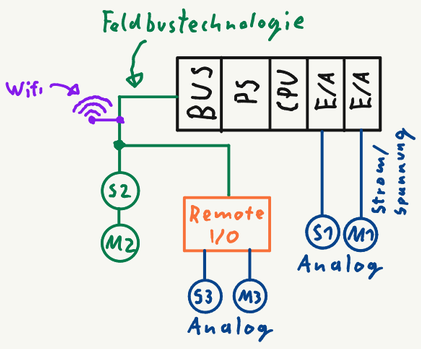
1-50 Regel:
1% des Codes verursacht 50% der Laufzeit

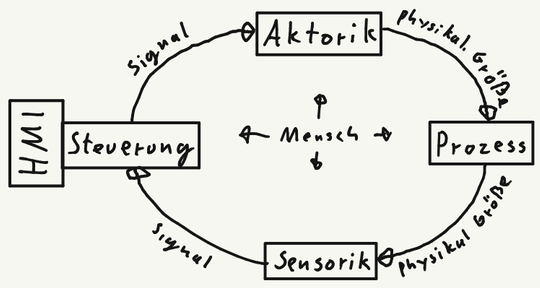
90:10 Regel:
90% der Laufzeit ist durch 10% des Codes verursacht

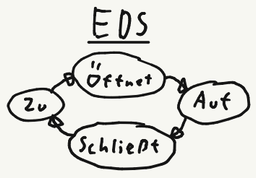
Eigenschaften

Stabil: Hält Reihenfolge von Gleichen
Abbrechbar: kann frühzeitig beendet werden
In-Place: Braucht kein extra Speicher
Iterativ: Nutzt Schleife für Ergebnisberechnung
Rekursiv: Ruft sich selbst mit Teilproblemen auf
Heuristisch: Findet Näherungslösung
Inkrementiell: Aktualisiert Ergebnis schrittweise
Determiniert: Bei gleicher Eingabe immer gleiches Ergebnis
Deterministisch: Schritte eindeutig (ohne Zufallsfaktor)
Probabilistisch: Nutzt Zufallselemente
(Las-Vegas-Algorithmus: Ergebnis korrekt, Laufzeit zufällig
Monte-Carlo-Algorithmus: Ergebnis zufällig, Laufzeit vorhersehbar)

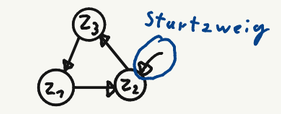
AUTOMATISIERUNGSTECHNIK







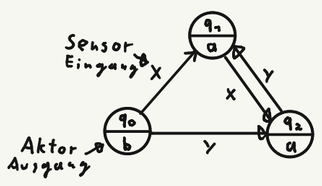
Autonomer Automat



	z_1	z_2	z_3
z_1	0	0	1
z_2	1	0	0
z_3	0	1	0
Σ	1	1	1

Determiniert

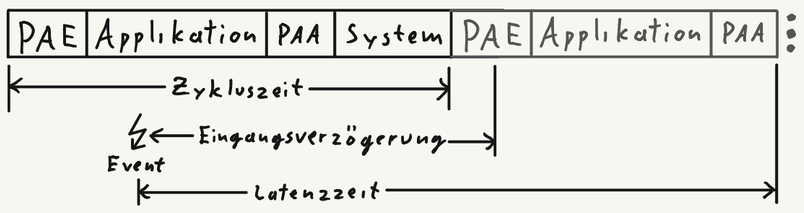
Nicht autonomer Automat

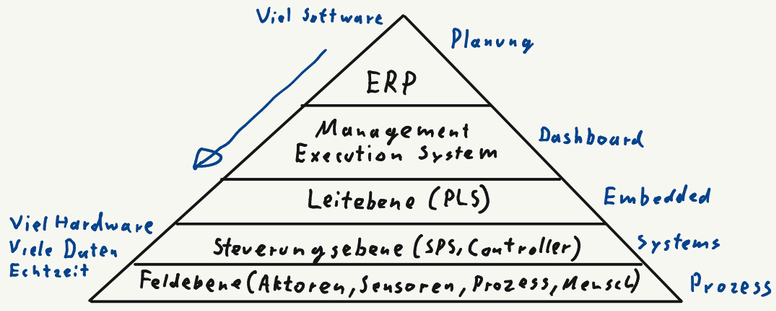


F(x)	q_0	q_1	q_2
q_0	0	0	0
q_1	1	0	0
q_2	0	1	0

F(y)	q_0	q_1	q_2
q_0	0	0	0
q_1	0	0	1
q_2	1	0	0

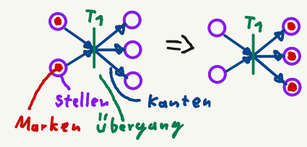
Nicht Determiniert





Petri-Netze

Sind alle Stellen zu einem Übergang markiert, werden alle Stellen danach markiert und die Markierungen davor gelöscht. Und Kanten sind halt Verbindungen.



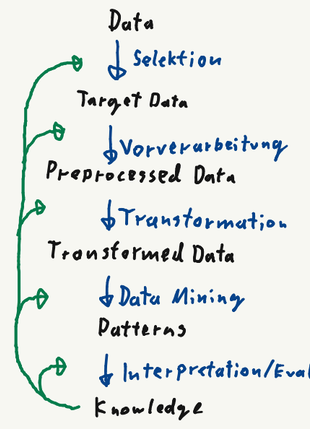
BIG DATA UND ML

Die 5 V's: **Volumen**, **Variety**, **Velocity**, **Veracity**, **Value**

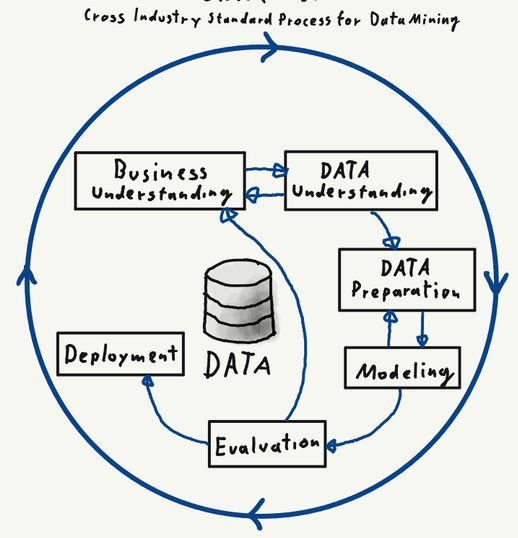
- Volumen: Menge der Daten
- Variety: Vielfalt der Daten
- Velocity: Geschwindigkeit der Datengenerierung
- Veracity: Zuverlässigkeit, Sinnhaftigkeit
- Value: Wirtschaftlicher Wert

Prozesse

KDD
Knowledge Discovery in Databases



CRISP-DM
Cross Industry Standard Process for Data Mining



Branchen:

- Personalservice
- Produktionsentwicklung (Marktlücken & Trends)
- Distribution & Logistik
- Gesundheitswesen
- Produktion (Wartung)
- Marketing & Vertrieb (Werbung)
- Versicherung (Betrugserkennung)
- Finanzen (Vorhersagen)
- Wirtschaft & Forschung

Skalentypen:

Nominal: Kategorien
Ordinal: Reihenfolge mit Rangordnung
Intervall: Reihenfolge + Gleichabständigkeit
Ratioskala: Intervall + Echte Null
Numerisch (Diskret): Ganze Zahlen
Kardinal: Quantitative Vergleiche

Arten der Datenanalyse

Descriptive Analytics:
Beschreibt Vergangenheit

Diagnostic Analytics:
Diagnostiziert Ursachen

Predictive Analytics:
Prognostiziert Zukunft

Prescriptive Analytics:
Empfielt Handlung

Statistikwerte

Mean: $\bar{x} = \frac{1}{n} \sum_1^n x_i$

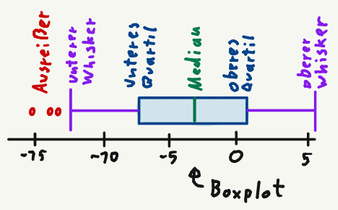
Median: $\begin{cases} \text{even: } \tilde{x} = \frac{X_{(n/2)+1} + X_{(n/2)}}{2} \\ \text{uneven: } \tilde{x} = X_{(n+1)/2} \end{cases}$

Modus: $\hat{x}$ = Häufigster Wert von X

Std-Abweichung: $std = \sqrt{\frac{1}{n-1} \sum_1^n (x_i - \bar{x})}$

Varianz: std^2

25%-Percentil = 1. Quartil (= Q1)
50%-Percentil = 2. Quartil = Median
75%-Percentil = 3. Quartil (= Q3)
Unterer Whisker = Q1 − 1,5 · (Q3 − Q1)
Oberer Whisker = Q3 − 1,5 · (Q3 − Q1)



Data Preparation

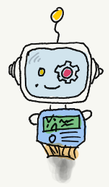
Bereinigung fehlender Werte: Interpolation

Ausreißererkennung: $\begin{cases} \text{Obere Grenze} = mean + k \cdot std \\ \text{Untere Grenze} = mean - k \cdot std \end{cases}$

Lineare Interpolation: $y = y_1 + ((x - x_1) \cdot (y_2 - y_1) / (x_2 - x_1))$

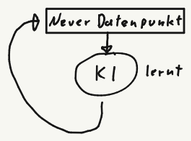
Normierung: $\begin{cases} \text{Z-Score: } z = \frac{(x - mean)}{std} \\ \text{Min-Max: } x_{neu} = \frac{(x - x_{min})}{(x_{max} - x_{min})} \end{cases}$

MASCHINELLES LERNEN



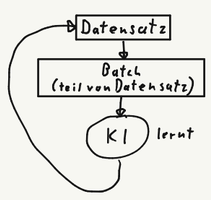
Incremential (Online) Learning

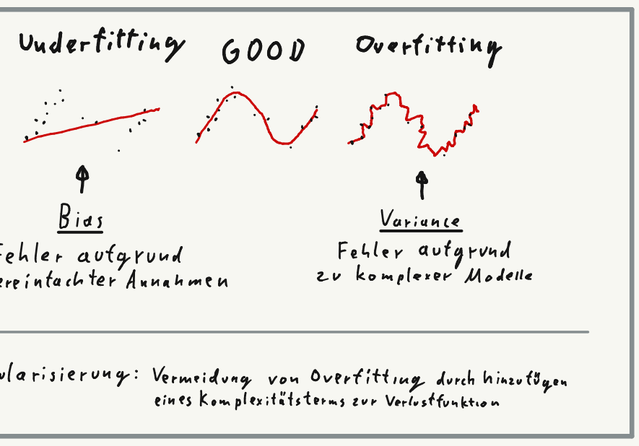
Kontinuierliches, echtzeit-lernen, flexibel, erfordert stetige Daten und schnelle Anpassung.



Batch (Offline) Learning

Lernen mit festen Daten, tiefergehend, erfordert umfassende Datenverarbeitung, kann nicht sofort auf neue Daten reagieren





Validierungs- und Evaluierungsmethoden

CONFUSION MATRIX 2.1

$$TPR = \frac{\Sigma \text{True Positives}}{\Sigma \text{Condition Positives}}$$

$$FPR = \frac{\Sigma \text{False Positives}}{\Sigma \text{Condition Negatives}}$$

$$FNR = \frac{\Sigma \text{False Negatives}}{\Sigma \text{Condition Positives}}$$

$$TNR = \frac{\Sigma \text{True Negatives}}{\Sigma \text{Condition Negatives}}$$

Hold-out cross validation
Teil von Dataset wird nicht zum training sondern nur zum testen verwendet

K-fold cross validation
Teilt in k-teile und verwendet jedes mal anderen Teil zum Testen

Leave-one-out cross validation

ROC-Kurve

Unsupervised Learning

Clustering

Hierachisch:	Dendogramm
Partitionierend:	k-Means
Dichtebasiert:	DBSCAN
Gitterbasiert:	Gitter lol

Dimension Reduction

Bsp. Temp + Luftfeuchtigkeit → Wohlfühlindex

Vorteile:
- Effizienz
- weniger Overfitting
- Bessere Visualisierung
- Rauschreduktion

DBSCAN

Parameter: Max Distanz, min. Samples per Cluster
Pro: Komplexe Clusterformen, Rauscherkennung, Dynamische Anzahl Cluster
Con: Sensible Parameter

K-Means

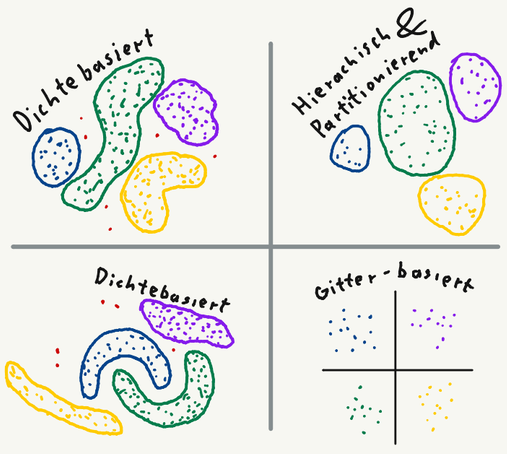
Parameter: Anzahl Cluster
Pro: Schnell, Datenunabhängig
Con: Anzahl Cluster muss gewählt werden

Dendogramm

Parameter: (Anzahl Cluster/Abstand)
Pro. Cluster Hierachie, keine Parameterwahl
Con: Keine Interation, keine Optimierung

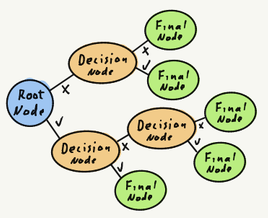
Gitterbasierend

Parameter: Gittergröße
Pro. Schnell, effizient, einfach
Con: Komplexe Cluster, Parameterwahl

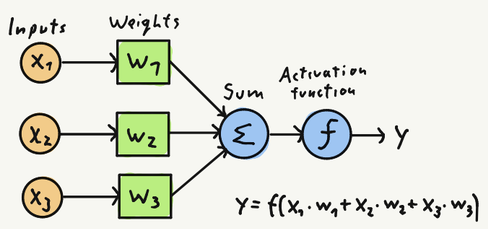


Supervised Learning

- Decision Tree Algorithm



- Perceptron Lernalgorithmus

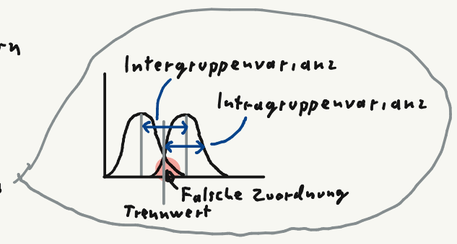


- Pattern Recognition
  Image → Feature extraction → Classifier / Regression
  (Deep Learning: Low-, mid-, high-Level Features)

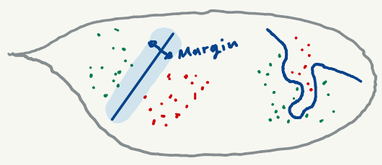
- K-nearest-neighbor
  Mehrheit aus K nächsten Nachbarn

Klassifizierung

- Diskriminanzanalyse
  Intergruppenvarianz möglichst groß
  Intragruppenvarianz möglichst klein

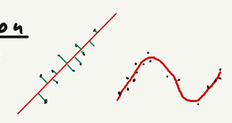


- Support Vector Machine
  Margin möglichst groß



Regression

- Gerade
- Kurve



Ziel: Minimierung des mittleren (quadratischen) Fehlers

Semi-Supervised Learning

Teilweise gelabelte Daten.
Modell wird mit gelableten Daten trainiert,
Ungelabelte Daten werden von ML-Modell gelablet und es wird erneut auf alle Daten trainiert.

Pro:
- Genauer als Unsupervised Learning
- Weniger Aufwand wegen labeln

Con:
- Ungenauer als Supervised Learning

Reinforcement Learning

Agent führt Aktionen in Umgebung aus um Belohnung zu maximieren

Anwendung: Q-Learning & Monte-Carlo-Methode