
Automatische Schadensdiagnostik

In der Werkstoff- und Prüftechnik

PD Stefan Bosse

Universität Bremen - FB Mathematik und Informatik / AG 0

Universität Siegen - FB Maschinenbau / LMW

Überblick



Wie können wir Zusammenhänge in der Werkstofftechnik mit Prüfungen bestimmen?

Überblick



Wie können wir Zusammenhänge in der Werkstofftechnik mit Prüfungen bestimmen?



Wie können wir Zusammenhänge aus Daten ableiten?

Überblick



Wie können wir Zusammenhänge in der Werkstofftechnik mit Prüfungen bestimmen?



Wie können wir Zusammenhänge aus Daten ableiten?



Wie können wir automatisiert Schäden aus Daten erkennen und charakterisieren?

Überblick



Wie können wir Zusammenhänge in der Werkstofftechnik mit Prüfungen bestimmen?



Wie können wir Zusammenhänge aus Daten ableiten?



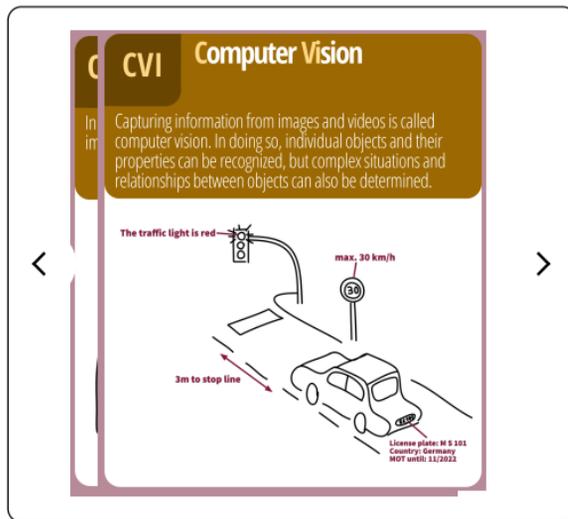
Wie können wir automatisiert Schäden aus Daten erkennen und charakterisieren?



Verfahren, Methoden, Algorithmen, Modelle

Anwendungen

Anwendungsklassen von Maschinellen Lernen



Motivation

Dieser Kurs mit interaktiven Übungen soll:

- Einen **anwendungsorientierten Einstieg** in die Datenanalyse und Modellierung mit Daten in der Werkstoff- und Prüftechnik sowie Interpretation mit Verfahren des **Maschinellen Lernens** bieten;
- Einen **Überblick** über gängige und weniger gängige **Verfahren** geben;
- **Interaktive Tutorials und Übungen mit zielgruppenorientierten Fallbeispielen** sollen Verfahren begreifbar und erfahrbar machen!

Beware of the Dog

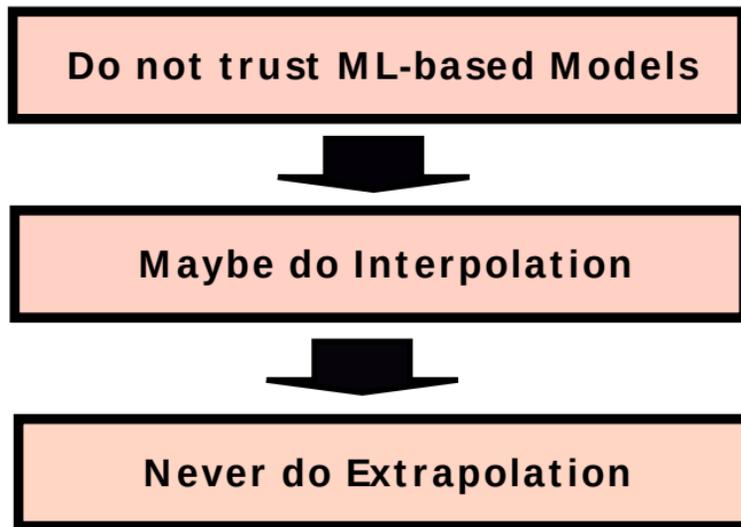


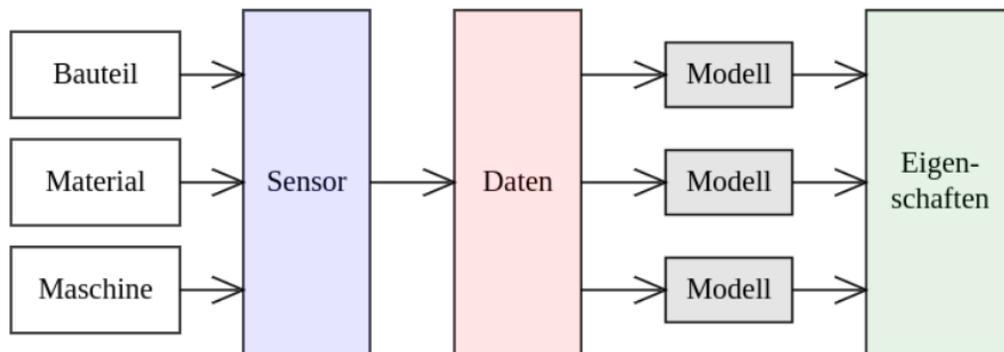
Abb. 1. ML Modelle sind häufig hoch nichtlineare und teils unstetige Funktionen mit unvorhersehbaren Verhalten außerhalb des Trainingsparameterraums

Inhalte

- Die Ontologie des Kurses besteht aus den **Bausteinklassen**:
 - **Modelle** (Datenstrukturen)
 - **Verfahren** (Algorithmen: Training, Test, Inferenz)
 - Überwachtes Training
 - Nichtüberwachtes Training
- Weiterhin aus den Anwendungs- und **Datenklassen**:
 - Sensorische und experimentelle Daten (Mess- und Prüftechnik)
 - Schadensmerkmale und Eigenschaften von Werkstoffen
 - Metrische und Kategorische Variablen



Die Grenzen der Datenklassen sind fließend! Material, Maschine und Bauteil als Sensoren!



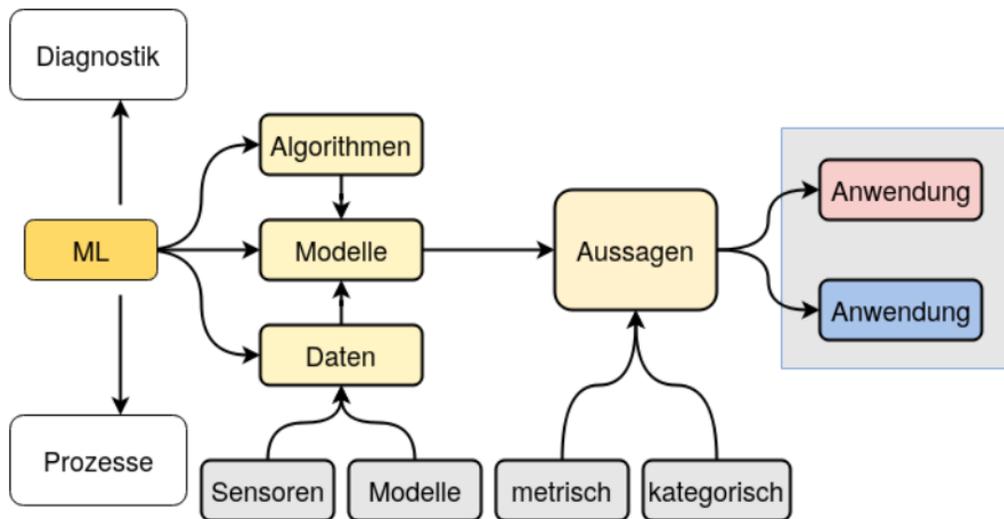


Abb. 2. Gemeinsame Verfahren und Modelle → Unterschiedliche Daten, Aussagen, Anwendungen

Organisation der Veranstaltung

1. Vorlesungen mit integrierten Übungen

- Vermittlung der Grundlagen
- Unmittelbare Übung und Anwendung der Grundlagen mit einfachen Übungen

2. Asynchrone Videos und Tutorials

- Auch offline seh- und hörbar

3. Gemeinsame Treffen mit Videokonferenz (Zoom, falls erforderlich) sowie vor Ort (Blöcke)

4. Interaktive Tutorials und Übungen mit *NoteBook* und *WorkBook* (NoteBook-2) im Web Browser!

- Offline ausführbar (evtl. werden Daten von einem Server geladen)
- Programmiersprache: R

Organisation der Veranstaltung

5. **Lernfelder:** Forschendes Lernen

- Nach Einführung der Grundlagen und einigen interaktiven Übungen gibt es konkrete zu bearbeitende Beispiele
- Die Lernfelder sind die Verknüpfung zu den Grundlagen von ML und Datenanalyse

6. Texte und Folien

- Vorlesungsskript (am Anfang: für jedes Modul/jede Einheit) als Ebook
- Das Vorlesungsskript gibt die Folieninhalte 1:1 wieder (nur anderes Layout und kompaktiert)
- Alle Folien im HTML Format (auch offline lesbar) und PDF
- Begleitende Literatur (Bücher im PDF)

Lernfelder

Orientierung an den Daten

1. Skalare Daten in Tabellen

- Skalare Materialeigenschaften
- Prozess- und Materialdaten
- Daten aus der Prüftechnik, z.B. Messung der Rauheit oder Härte

2. Vektorielle Daten

- Daten- und Zeitserien
- Messkurven, z.B. Dehnungstest
- Zeitaufgelöste Signale, wie z.B. akustische Maschinendaten

Services

1. Web Service: Informationen, Dokumente, Folien, Videos:
<https://edu-9.de/Lehre/asd3k>
2. Videos: <https://edu-9.de/Lehre/asd3k>

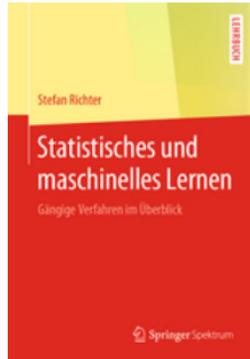
Prüfungsleistungen

1. Bearbeitung der Aufgaben (Programmierung und Dokumentation, jeweils 1 Punkt)
2. Mündliche Prüfung
3. Alternativ: Schriftliche Seminararbeit (Umfang ca. 20-30 Seiten)

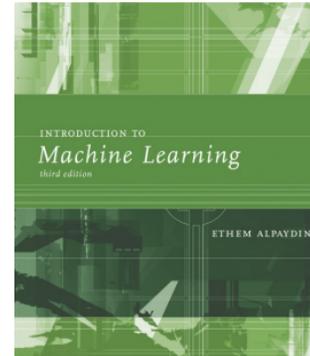
Literatur

- Zur Vertiefung!

S. Richter, Statistisches und Maschinelles Lernen. Springer Spektrum, 2019.

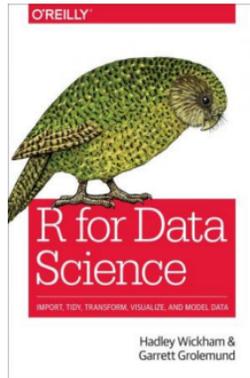


E. Alpaydm, Introduction to Machine Learning. MIT Press, 2010.



Programmierung

Hadley Wickham, Garrett Golemund, R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. 2017

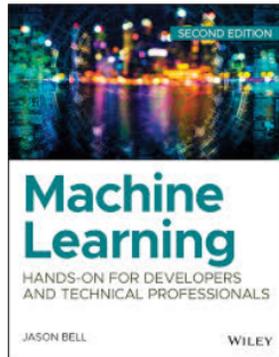


Uwe Ligges, Programmieren mit R. 2006.

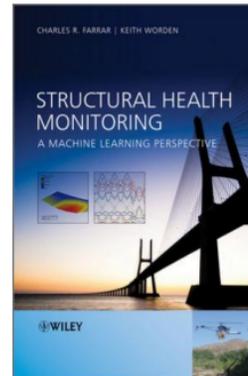


Domainspezifische Literatur

J. Bell, Machine Learning - Hands-On for Developers and Technical Professionals. John Wiley & Sons, Ltd, 2015.



C. R. Farrar and K. Worden, Structural Health Monitoring: A Machine Learning Perspective. Wiley-Interscience, 2013.



Software

Anforderungen und Eigenschaften



Textuelle oder grafische Programmierung?

Software

Anforderungen und Eigenschaften



Textuelle oder grafische Programmierung?



Effiziente Berechnung (Performanz)?

Software

Anforderungen und Eigenschaften



Textuelle oder grafische Programmierung?



Effiziente Berechnung (Performanz)?



Lernkurve und Einfachheit bei gleichzeitig hoher Ausdrucksfähigkeit (Abstraktion)?

Programmiersprachen

Python

- Prozedurale und objektorientierte Programmiersprache
- Mittlere Ausdrucksstärke
- Polymorphie
- "Unhandliche" Syntax durch Einrückung

R

- Prozedurale, funktionale, und teils deklarative Sprache
- Arrays
- Teils sehr mächtige Funktionen und Ausdrücke
- Abstraktion von Ausdrücken, deklarativ
- Polymorphie

JavaScript

- Prozedurale, funktionale, und teils objektorientierte Sprache
- Gepackte Arrays (C Layout)
- Polymorphie

Programmiersprachen: Implementierung

Python

- Virtuelle Maschine
 - Bytecode Interpreter (langsam!)
- Terminal
- Browser: Jupyter mit Server
- Numerische Berechnungen nicht direkt in Python möglich (nur C/C++)

R

- Virtuelle Maschine
 - Bytecode Interpreter (mittel!)
 - Just-in-time native code Compiler (?, JIT, schnell)
- Terminal
- Browser: R Studio mit Server
- Numerische Berechnungen eventuell direkt in R möglich (sonst C/C++)

JavaScript

- Virtuelle Maschine
 - Bytecode Interpreter (schnell!) und teils
 - Just-in-time native code Compiler (JIT, sehr schnell)
- Terminal und Browser
- Numerische Berechnungen direkt in JavaScript möglich

NoteBook

- Interaktive vorwiegend praktische Übungen werden rein digital im Web Browser mit den *NoteBooks* durchgeführt
- Ein digitale Übung (oder Tutorial) besteht aus:
 - Textabschnitten
 - Informationsblöcken
 - Aufgaben (mit Lösungen)
 - Editoren für Programmcode
 - Ausführungsterminals für Programmcode
 - uvm.



↑ ↓ ☰ ☒ WEB Umfragen (Stefan Bosse) [7.2020] ☰ ☒ ☑ ☒ ☒

WEB Umfragen: Analyse der Umfrage (3)

- Hinweis: Grundlegende JavaScript Kenntnisse sind erforderlich. Lese auch das Tutorial [JavaScript](#) ☒ !!!!

In diesem Tutorial wird die Umfrage aus dem vorherigen Tutorial auf einem WEB Server aus diesen Notebook heraus veröffentlicht und die Daten der Erhebung wieder eingesammelt.

- Mittels des *Math.statistics* Moduls können nun Daten statistisch analysiert werden
 - Eine Dokumentation der Statistikfunktionen findet sich hier (es gibt noch einige zusätzliche Funktionen, siehe unten) [STAT](#) ☒

Der WEB Server

- Der WEB Umfrageserver ist über die URL <http://ag-0.de:22222> ☒ für alle öffentlich erreichbar
- Also hier [WEB Server](#) ☒

Auswertung

Die Umfrage

- Hier die eigene Umfrage aus dem vorherigen Tutorial übertragen

Umfrage

```
1 survey = {
2   author: 'Stefan Bosse',
3   url: 'ag-0.de:22222',
4   label: 'mysurvl',
5   title: 'Meine Beispielsumfrage',
6   start: 'now',
7   time: 600, // seconds
8   cinema: true,
```

Abb. 3. Ein NoteBook im WEB Browser

NoteBook Konzept

- Top-down Bearbeitungsfluss
- Statische Struktur mit dynamischen Inhalten
- Alle dynamischen Inhalte können in einer JSON Datei gespeichert und wieder geladen werden
- Es können Notizzettel überall im Notebook angeheftet werden (werden auch gespeichert)
- Musterlösungen (dynamische Inhalte) können eingebettet und mit einem Schlüssel freigeschaltet werden

R+ WorkBook Konzept

- Dynamische Struktur mit dynamischen Inhalten \Rightarrow Freies Experimentieren (optional), ähnlich Jupyter
- Ein R+ WorkBook besteht aus
 - Textabschnitten (Markdown)
 - Codesnippets mit Editoren und Ausgabekonsolen
 - Speziellen Snippets wie editierbare Tabellen oder allg. Formulare
- Programmierung in R(+),
- Alle dynamischen Inhalte und Daten können im JSON Format gespeichert und wieder geladen werden

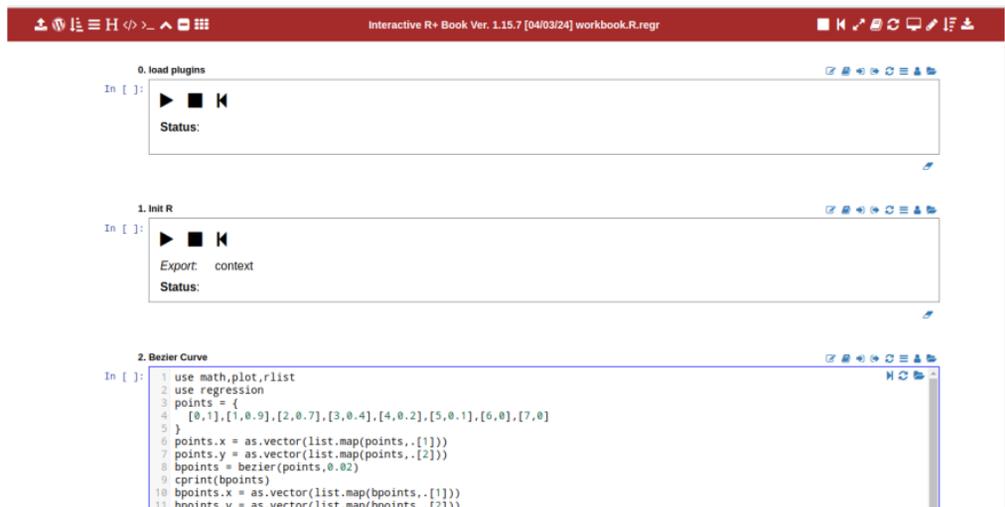


Abb. 4. Ein R+ WorkBook Beispiel

Programmiersprache?

Python

```
import numpy as np
l=1 r=1 y=l+t
v=np.array([1,2,3])
w=np.array([4,5,6])
u=np.add(v, w)
```

R

```
l=1 r=1 y=l+r
v=c(1,2,3)
w=c(4,5,6)
u=v+w
```

JS

```
l=1 r=1 y=l+r
v=[1,2,3]
w=[4,5,6]
u=v.map((x,i) =>
  (x+w[i]))
v=new Float32Array([1,2,3])
w=new Float32Array([4,5,6])
u=new Float32Array(3)
for(var i=0;i<v.length;i++)
  u[i]=v[i]+w[i];
```

Programmiersprache?

- R:
 - Weit verbreitet,
 - Läuft nicht im Browser oder nw.js direkt
 - Umfrangreiche Softwareinstallation und DevTools erforderlich
- WorkBook/JS:
 - Nicht weit verbreitet,
 - Läuft im Browser und nw.js direkt,
 - Keine Softwareinstallation notwendig (außer nw.js)



R+ ist die Synthese: R in JS mit Compiler R → JS und Unterstützung gängiger Bibliotheken (Math, ML, Statistics, Tensorflow, C5.0)

Programmiersprache?

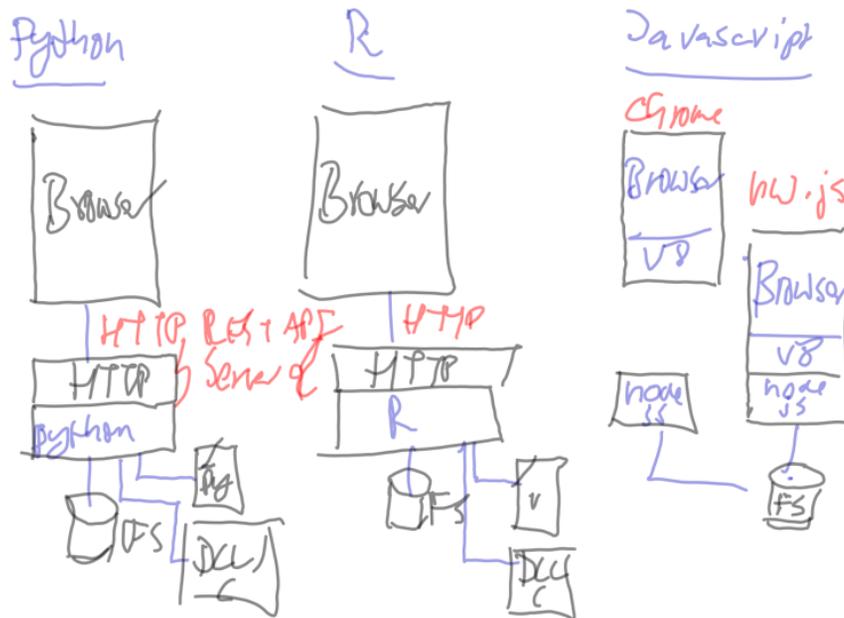


Abb. 5. Vergleich der Softwarearchitekturen für verschiedene Programmiersprachen

Programmiersprache R+!



Wir verwenden im Kurs R+, ein R Dialekt.

- R+ ist selber in JavaScript implementiert, erweiterbar, und läuft daher in jedem Web Browser und nw.js
- Alle digitalen Übungen beinhalten R+
- R+ gleicht einige syntaktische und operationale Schwächen von R aus, führt Vereinfachungen ein, ist aber daher nicht vollständig kompatibel zu R



Wichtig: In diesem Kurs soll nicht programmiert sondern beobachtet und evaluiert werden.

Daher dürften die meisten programmatischen Abschnitte weniger als 10 Zeilen R+ umfassen.

Programmiersprache R+!

Web R+ WorkShell Live

CLEAR

LOAD

+

-

Machinelles Lernen

Schlüsselwörter und Begriffe



Welche Begriffe werden häufig bei ML genannt:

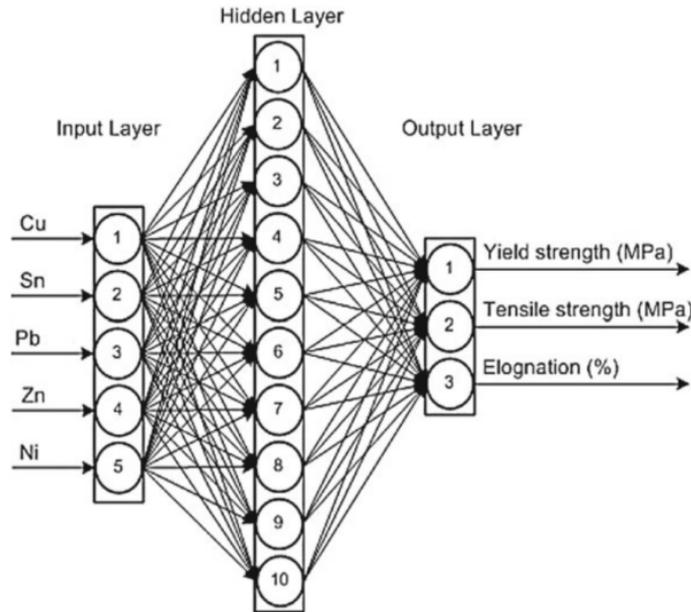
Anwendungsgebiete



Welche Anwendungsgebiete gibt es:

"Material Informatik"

[1] Y. Liu, T. Zhao, W. Ju, and S. Shi, "Materials discovery and design using machine learning," J Materiomics, vol. 3, 2017.



[Ozerdem MS, Kolkusa S (2009)]

Abb. 6. Vorhersage von Materialeigenschaften von kombinierbaren/parametrisierbaren Legierungen

Fragestellungen



Welche Fragestellungen (zu lösende Probleme) gibt es:

Inhalte

1. **Eingabe** x : Daten (Attribute) und Eigenschaften (Analyse)
2. **Sensoren**: Erfassung von Daten, $S(welt): welt \rightarrow x$
3. **Ausgabe** y : Numerische und kategorische Werte
4. **Metriken und Taxonomie**: Grundlagen des Maschinellen Lernens
5. **Algorithmen und Modelle**: $f(x): x \rightarrow y$
6. **Training, Lernen, Prädiktion, Test** $M(\langle x,y \rangle): \langle x,y \rangle \rightarrow f$
7. **Anwendungen**

Geschichte

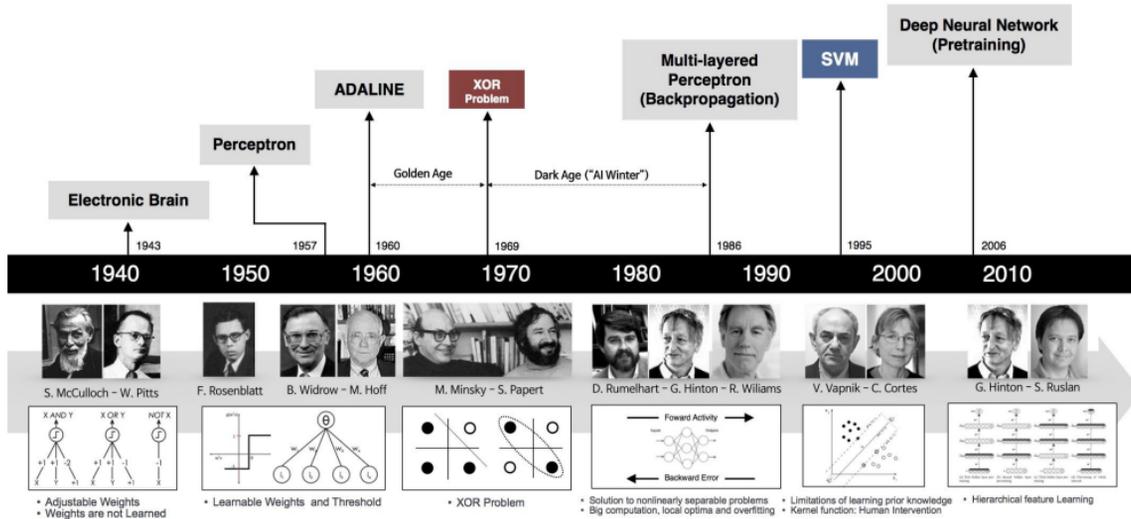


Abb. 7. Die Geschichte fokussiert auf Neuronale Netze. Es gibt mehr.

[www.pinterest.com]

Modelle

1. Entscheidungsbäume (gerichtete Graphen)
2. Funktionen (z.B. Polynome)
3. Funktionsgraphen (z.B. künstliche neuronale Netzwerke)
4. Ungerichtete Graphen (Gruppenbildung)
5. Zyklische und azyklische Graphen

Algorithmen

1. Entscheidungsbäume: C4.5, ID3, IDT, C5.0 (R!), Regressionsbäume
 - Teilungsverfahren (mit Entropie, Informationsgewinn, usw.)
2. Lineare und nichtlineare Regression, Support Vector Machines (SVM)
 - Least Square Fit (iterativ, mehrschrittig)
 - Lineare Algebra (numerisch, einschrittig)
3. Datenanalyse!: Hauptkomponentenanalyse, statistische Methoden
4. Bayesian Netzwerke mit statistischem Methoden (probabilistische Verfahren)
5. Rückwärtspropagation von Fehlergradienten (vor allem KNN) aus Vorwärtsberechnung
6. Überwachte und nichtüberwachte Trainingsverfahren
7. Zustandsbasierte Funktionen (LSTM) für Datenserien

Datenanalyse und Eigenschaftsselektion

Wir unterscheiden folgende Klassen von Eigenschaften in der Datenanalyse und Prädiktion (Merkmale, Features):

1. Eigenschaften der Eingabedaten, vor allem dominante Eigenschaften abgeleitet aus den Eingabedaten x mit starker y Korrelation
 - Beispiel: Charakteristische Signalfrequenz einer Betriebsschwingung die auf einen Schaden hindeutet (NDT/SHM)
2. Zieleigenschaften, also Werte der Zielvariable y
 - Numerische Eigenschaften (kontinuierlich oder diskret), z.B. Materialdichte, Schadensposition, Bruchdehnung
 - Kategorische Eigenschaften, Z.B. Farbe, Tierart, Schadensklasse, Entscheidungen

Datenanalyse und Eigenschaftsselektion

Häufig sind die rohen sensorischen Daten(variablen) zu hochdimensional und noch abhängig voneinander (schwache Korrelation mit y)

Datenanalyse und Eigenschaftsselektion

Häufig sind die rohen sensorischen Daten(variablen) zu hochdimensional und noch abhängig voneinander (schwache Korrelation mit y)

Reduktion auf wesentliche Merkmale kann ML Qualität deutlich verbessern!

Datenanalyse und Eigenschaftsselektion

Häufig sind die rohen sensorischen Daten(variablen) zu hochdimensional und noch abhängig voneinander (schwache Korrelation mit y)

Reduktion auf wesentliche Merkmale kann ML Qualität deutlich verbessern!

Häufig besitzen einzelne Sensorvariablen keine oder nur geringe Aussagekraft (geringe Entscheidbarkeitsqualität) → geringe bis keine Korrelation mit y oder sogar Antikorrelation (Störung)

Datenverarbeitung

- Die Daten die als Grundlage für die Induktion (Lernen) und die Deduktion (Applikation/Inferenz der Zielvariablen) müssen i.A. vorverarbeitet werden → **Merkmalsselektion**

Datenverarbeitung

- Die Daten die als Grundlage für die Induktion (Lernen) und die Deduktion (Applikation/Inferenz der Zielvariablen) müssen i.A. vorverarbeitet werden → **Merkmalsselektion**

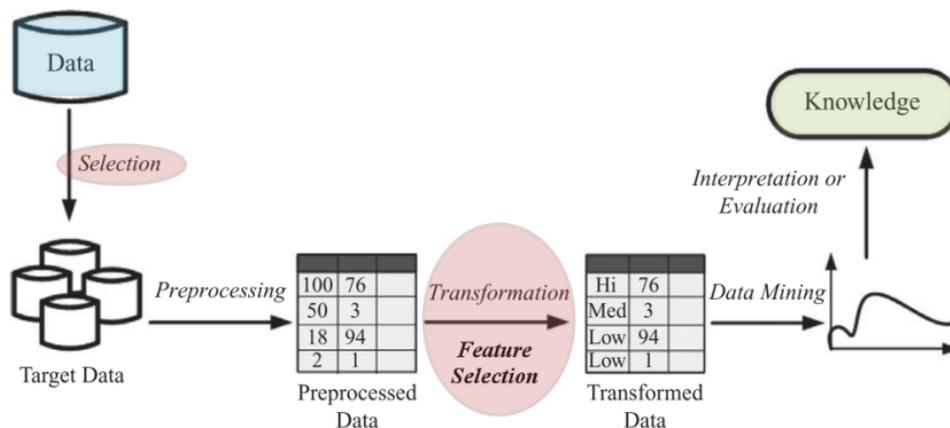


Abb. 8. Maschinelles Lernen ist ein Werkzeug der Datenanalyse und des Data Minings

Datenverarbeitung und Programmausführung

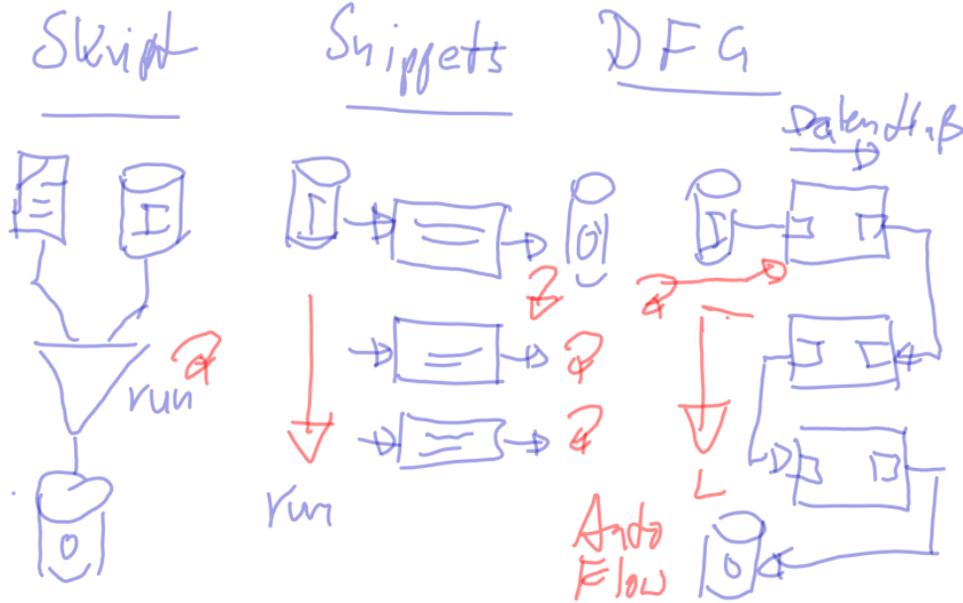


Abb. 9. Vergleich verschiedener Ausführungs und Datenverarbeitungskonzepten

Modellbildung

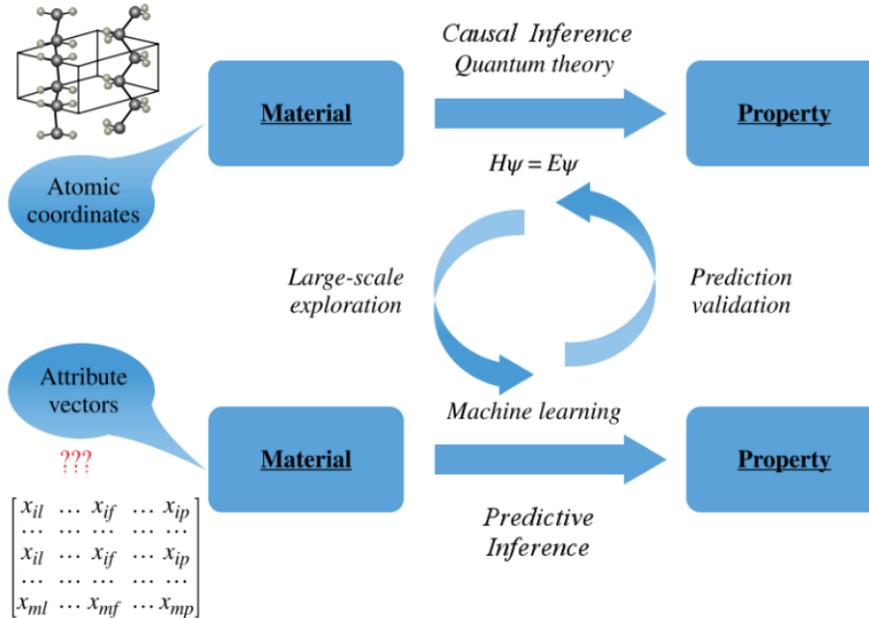


Abb. 10. Kausale vs. Prädiktive Modellbildung und Physikalische Modelle versa algorithmisch bestimmte Modelle (Hypothesen)

Modellbildung

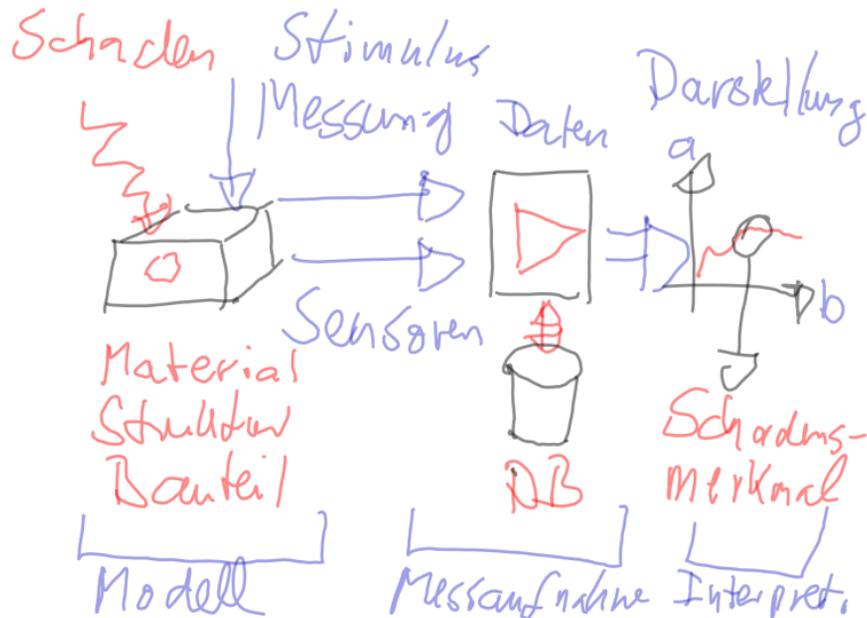


Abb. 11. Am Anfang stehen Messdaten aus Experimenten

Modellbildung

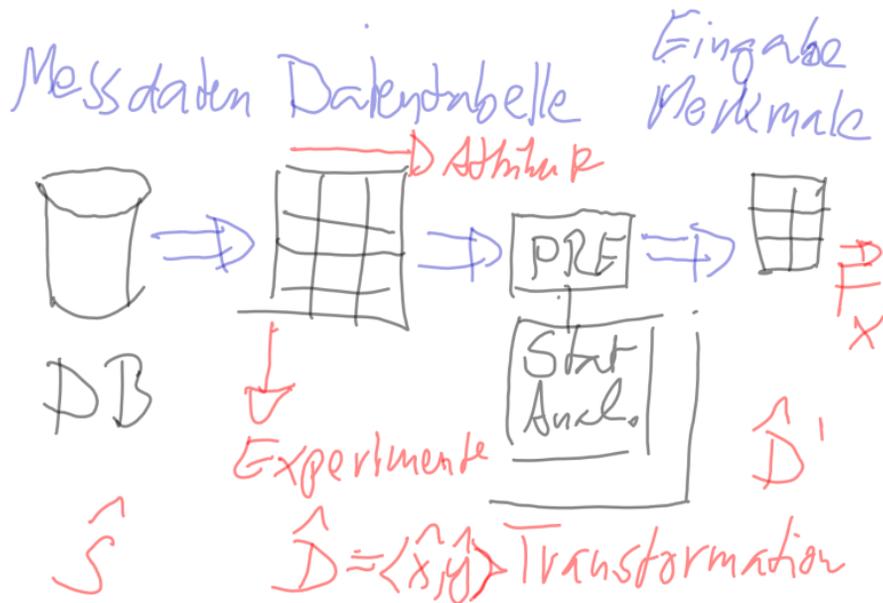


Abb. 12. Ableitung von Eingabemerkmale aus den Messdaten

Modellbildung

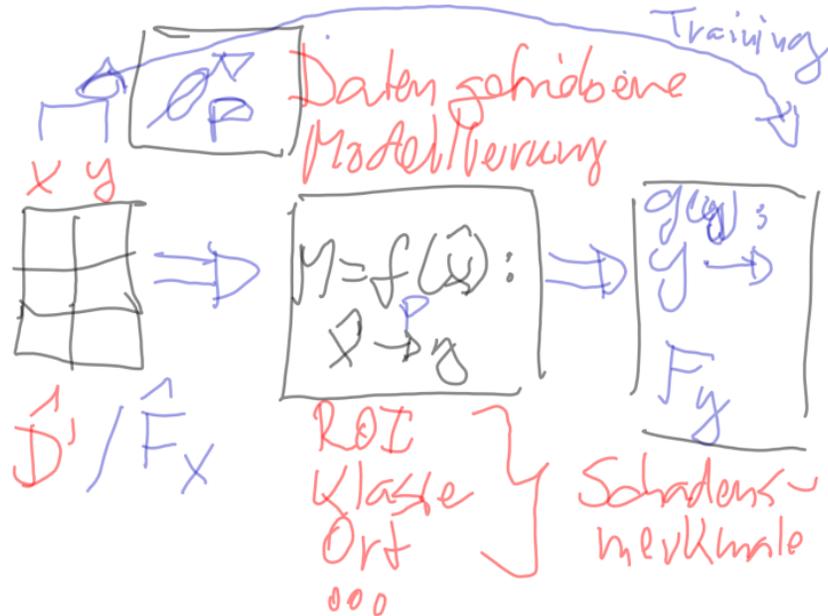
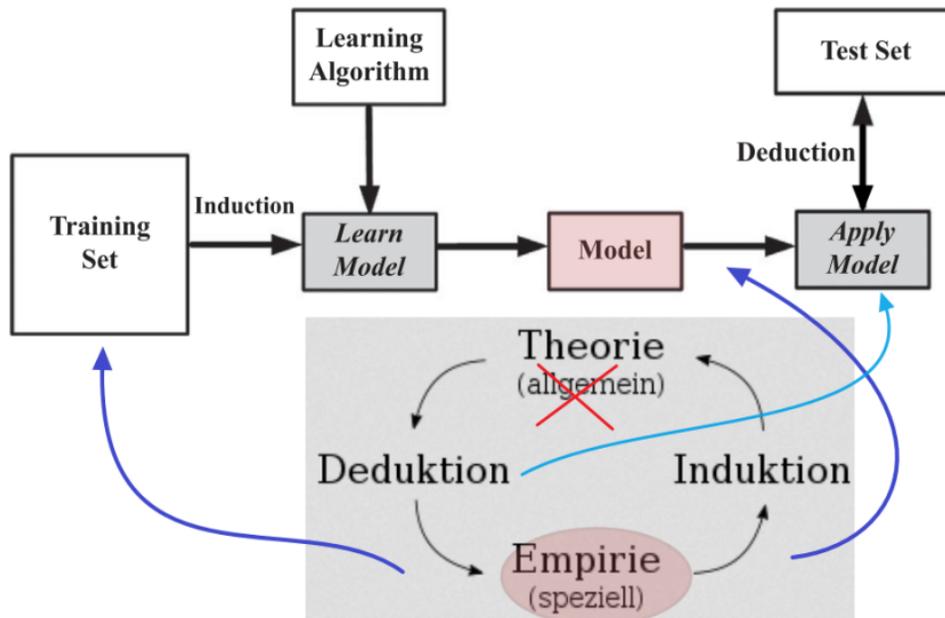


Abb. 13. Training passt Modellparameter derart an dass $f(x); x \rightarrow y$ die Eingabemerkmale mit geringsten Fehler auf die Ausgabemerkmale anpasst

Induktion und Deduktion



[6]

Abb. 14. Ablauf Überwachtes Lernen mit Trainings- (Induktion) und Applikationsphasen (Deduktion). Aber: Meistens keine Verallgemeinerung!