

# **Seminar Grundlagen Machine Learning**

Methoden und Algorithmen zur praktischen  
Umsetzung mit Python

## 01: Introduction to Machine Learning

© Prof. Dr. Andreas Theissler

[www.ml-and-vis.org](http://www.ml-and-vis.org)

Montag, 11. und Dienstag, 12. November 2024  
9.00 bis 12.15 und 13.45 bis 17.00 Uhr

## Grundlagen

- Einführung Machine Learning
- Extraktion von Merkmalen aus Daten
- Datenvorverarbeitung
- Distanzmetriken

## Clustering

- Finden von Gruppen in Datensätzen
- partitionierende Verfahren (k-means)
- Programmierung mit Python sklearn

## Classification

- automatische Zuordnung von Daten, zum Beispiel Erkennung eines Objekts auf Basis von Eigenschaften
- Verfahren: zum Beispiel k-nearest neighbors, decision trees, support vector machines, artificial neural networks u.a.
- Programmierungen mit Python sklearn und tensorflow/keras

**=> ggf. weitere oder abweichende Punkte: to be discussed**

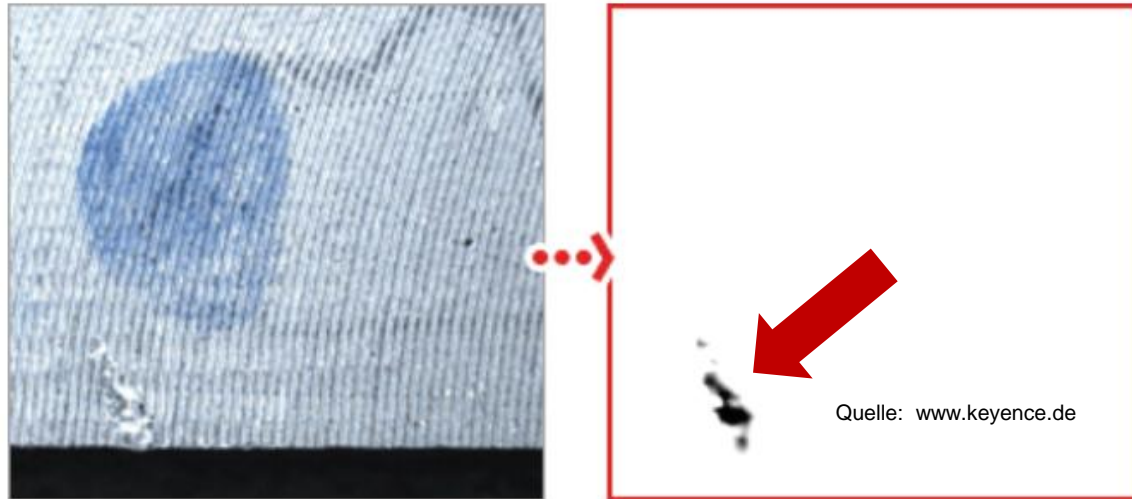
# Machine Learning –

## An introduction with examples and case studies

---

### **Example: Quality control in manufacturing**

- detection of quality issues based on images



# Case studies

## Example: OCR (optical character recognition)

- scanning of printed books or hand-written text
- the task is to classify image segments to pre-defined letters, digits, etc.

PROFESSIONAL DETAILS

Employed ☒ Self-employed ☐ Retired ☐ Other ☐ Checkmarks

Name of Employer or if Self-Employed, Trading Name **Sunset Tan**

Occupation **Manager**

Country **USA** ZIP Code **90210**

State **CA** City **Hollywood** Hand-printed text

Street **West Hollywood Ave**

House Building **2701** Apartment

Work Phone **310 221 0111** Mobile Phone

Gross Annual Income, USD **90000.00** Employment Date (mm dd yyyy) **10/15/2003**

I Declare the Correctness of the Information above

Barcode  
6 529141 677079

Document Date (mm dd yyyy) **7/10/2007**

Signature **Robert Gray**

Sunset Tan Stamp  
(Employer HR Officer Signature)



Sunset Tan  
Manager  
USA  
CA  
West Hollywood Ave  
...

taken from: [www.abbyy.com/Examples.jpg](http://www.abbyy.com/Examples.jpg)

# Case studies

## Example: driver assistant systems

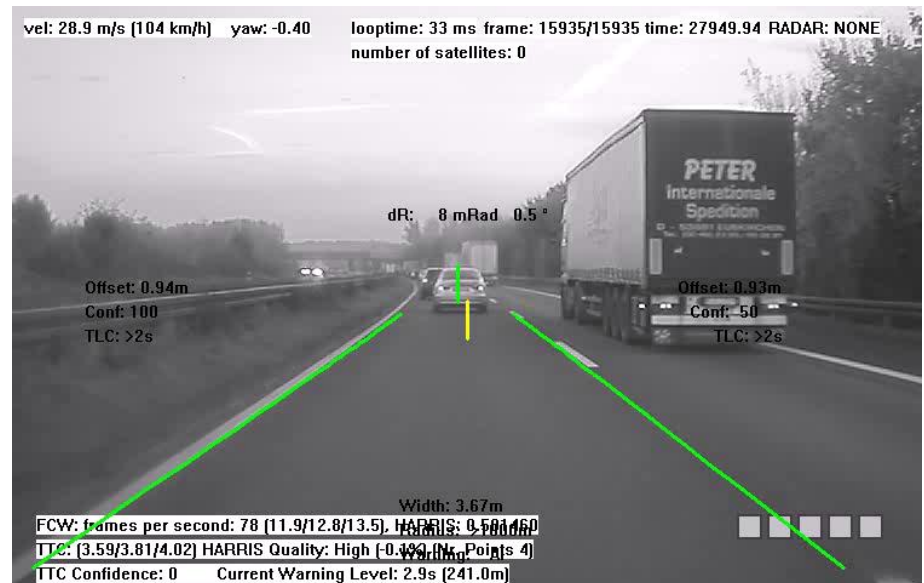
- detection of objects based on sensor data (e.g. camera, radar, laser)

pedestrian detection:



taken from:  
[www.kfztech.de/kfztechnik/fas](http://www.kfztech.de/kfztechnik/fas)

lane keeping, adaptive cruise control:



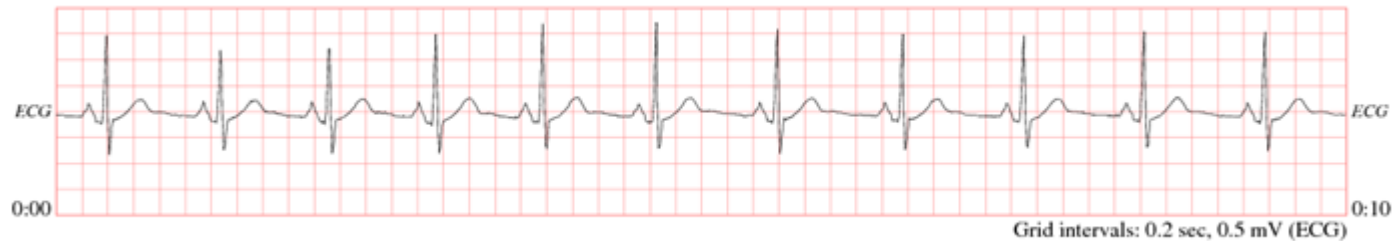
taken from:  
<http://mux.fh-oow.de/ti/archives/kamerabasierte-fahrerassistenzsysteme>

# Case studies

---

## Example: medical data (ECG)

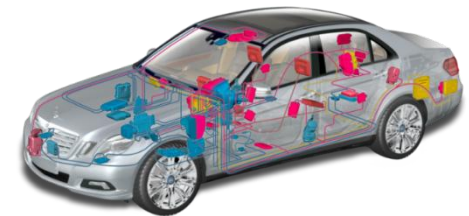
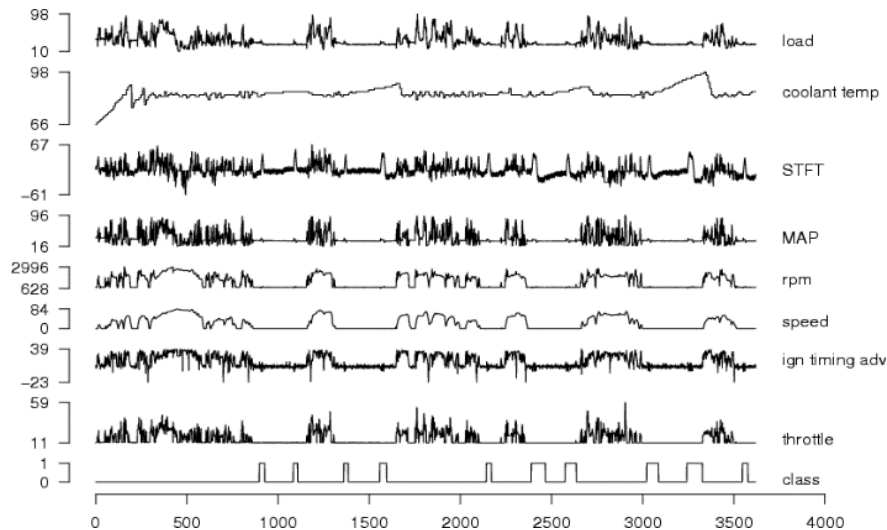
- classification of ECG data (heart beats)
- automatic detection of problems, e.g. arrhythmia, sleep apnea



taken from: [www.physionet.org](http://www.physionet.org)

## Example: Automotive data

- anomaly detection in recordings from automotive systems, e.g. road trials or lab tests
- anomalies may point to faults



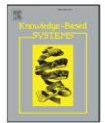
Knowledge-Based Systems 123 (2017) 163–173



Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: [www.elsevier.com/locate/knosys](http://www.elsevier.com/locate/knosys)



### Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection

Andreas Theissler

Faculty of Information Technology, Esslingen University of Applied Sciences, 73732 Esslingen, Germany



#### ARTICLE INFO

**Article history:**  
Received 27 October 2016  
Revised 16 February 2017  
Accepted 17 February 2017  
Available online 20 February 2017

**Keywords:**  
Anomaly detection  
Fault detection  
One-class classification  
Ensemble methods  
Vehicle electronics

#### ABSTRACT

The massive growth of data produced in the automotive industry by acquiring data during production and test of vehicles requires effective and intelligent ways of analysing these recordings. In order to detect potential faults, data from the in-vehicle network interconnecting vehicle subsystems is recorded during road trials. The complexity and volume of this data keeps increasing since the degree of interconnection between the vehicle subsystems and the amount of data transmitted over the in-vehicle network is augmented with each functionality added to modern vehicles. In this paper, an anomaly detection approach is proposed that (a) is capable of detecting faults of known and previously unknown fault types, (b) functions as an out-of-the-box approach not requiring the setting of expert-parameters and (c) is robust against different driving scenarios and fault types. To achieve this, an ensemble classifier is used consisting of two-class and one-class classifiers. Without modelling effort and user parameterisation the approach reports anomalies in the multivariate time series which point the expert to potential faults. The approach is validated on recordings from road trials and it could be shown that the ensemble-anomaly detector is robust against different driving scenarios and fault types.

© 2017 Elsevier B.V. All rights reserved.

# Case studies

## ► translation (deepl.com)

The screenshot shows the DeepL translator website. The browser address bar displays <https://www.deepl.com/translator>. The page has a search bar with the placeholder text "Suchen". The DeepL logo is in the top left, and a hamburger menu icon is in the top right. The main content area is divided into two panels. The left panel is titled "Translate from **ENGLISH** (detected) ▼" and contains three paragraphs of English text about Twitter's tweet length limit. The right panel is titled "Translate into **GERMAN** ▼" and contains the German translation of the same text. A yellow arrow points from the English text to the German text. At the bottom of the left panel, there is a note: "To look up words in the dictionary, just click on them." At the bottom of the right panel, there is a blue box with white text: "Click on a word to get alternative formulations."

DeepL

Translate from **ENGLISH** (detected) ▼

Twitter is testing doubling the length of its tweets to 280 characters, a move that overhauls the social network's defining feature.

A small percentage of Twitter's 328 million users will find they can post longer tweets from Tuesday evening, with all other users will be able to see them.

The test applies across Twitter apart from in Japanese, Korean and Chinese, which use scripts instead of letters, meaning tweets in those languages are rarely constrained by the existing limit.

To look up words in the dictionary, just click on them.

Translate into **GERMAN** ▼

Twitter testet die Verdoppelung der Länge seiner Tweets auf 280 Zeichen, eine Bewegung, die das Social Network's Definitions-Feature überarbeitet.

Ein kleiner Prozentsatz der 328 Millionen Twitter-Nutzer wird feststellen, dass sie ab Dienstagabend längere Tweets posten können, die alle anderen Nutzer sehen können.

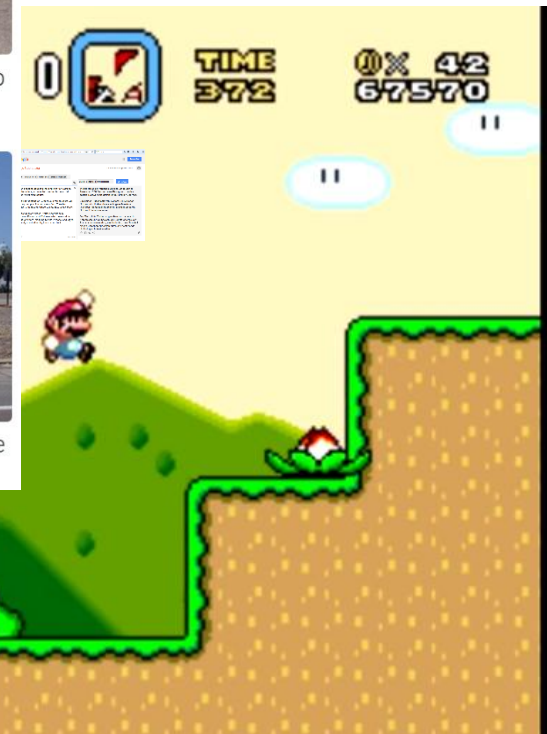
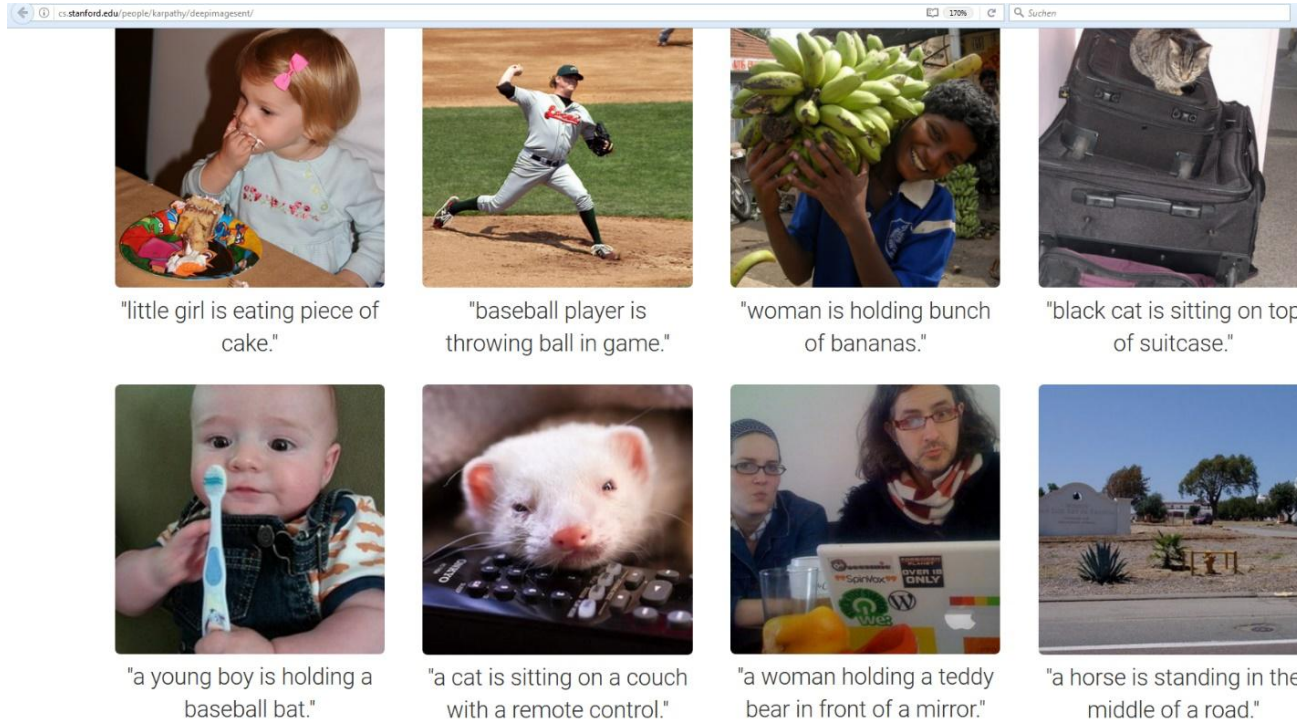
Der Test gilt für Twitter mit Ausnahme von Japanisch, Koreanisch und Chinesisch, die Skripte anstelle von Buchstaben verwenden, d. h. Tweets in diesen Sprachen werden selten durch die bestehende Grenze eingeschränkt.

Click on a word to get alternative formulations.



# Case studies

- Understanding images



- Playing computer games

# Case studies

---

- ▶ Denoising images

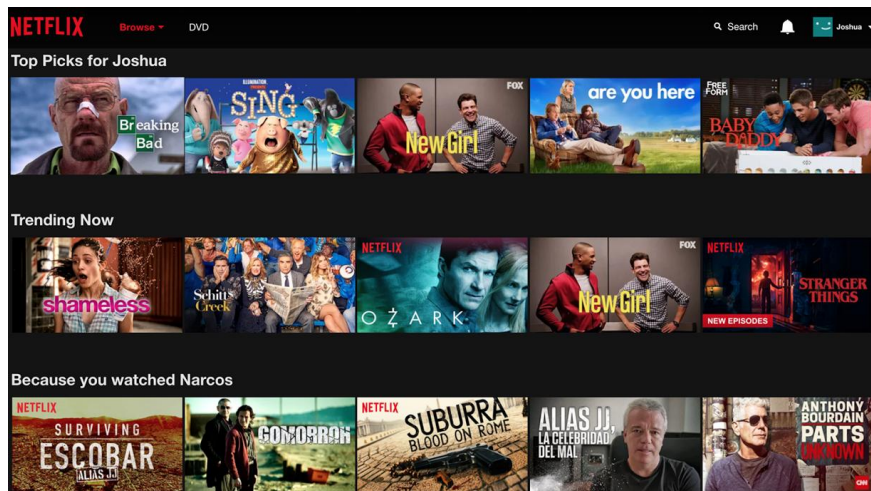
ground-truth / noisy observation / prediction



# Case studies

## ANN für Recommender Systems (Zhang et al. 2019)

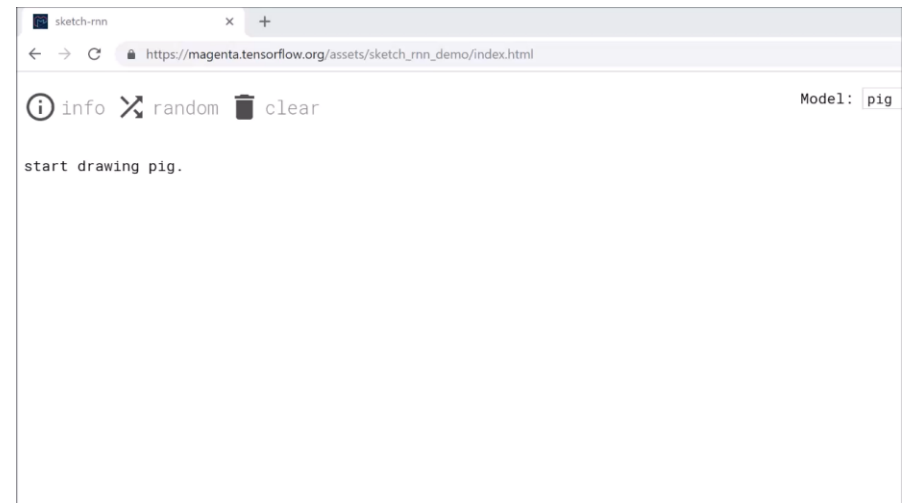
HCI zeigt Empfehlungen an



<https://medium.com/netflix-techblog/interleaving-in-online-experiments-at-netflix-a04ee392ec55>

## ANN zur Planung: Ermitteln der Benutzeraktion

Vorschlag zur Automatisierung /  
Vervollständigung



[https://magenta.tensorflow.org/assets/sketch\\_rnn\\_demo/index.html](https://magenta.tensorflow.org/assets/sketch_rnn_demo/index.html)

# Case studies

---

Some fun stuff... turning a horse into a zebra with Artificial Neural Networks

<https://www.youtube.com/watch?v=JzgOfISLNjk>





# Playing Games: AlphaGo – AI defeats world-champion

Making history

AlphaGo is the first computer program to defeat a professional human Go player, the first to defeat a Go world champion, and is arguably the strongest Go player in history.

To capture the intuitive aspect of the game, we needed a new approach.

We created AlphaGo, a computer program that combines advanced search tree with deep neural networks. These neural networks take a description of the Go board as an input and process it through a number of different network layers containing millions of neuron-like connections.

One neural network, the “policy network”, selects the next move to play. The other neural network, the “value network”, predicts the winner of the game. We introduced AlphaGo to numerous amateur games to help it develop an understanding of reasonable human play. Then we had it play against different versions of itself thousands of times, each time learning from its mistakes.

Over time, AlphaGo improved and became increasingly stronger and better at learning and decision-making. This process is known as reinforcement learning. AlphaGo went on to defeat Go world champions in different global arenas and arguably became the greatest Go player of all time.

Source: DeepMind

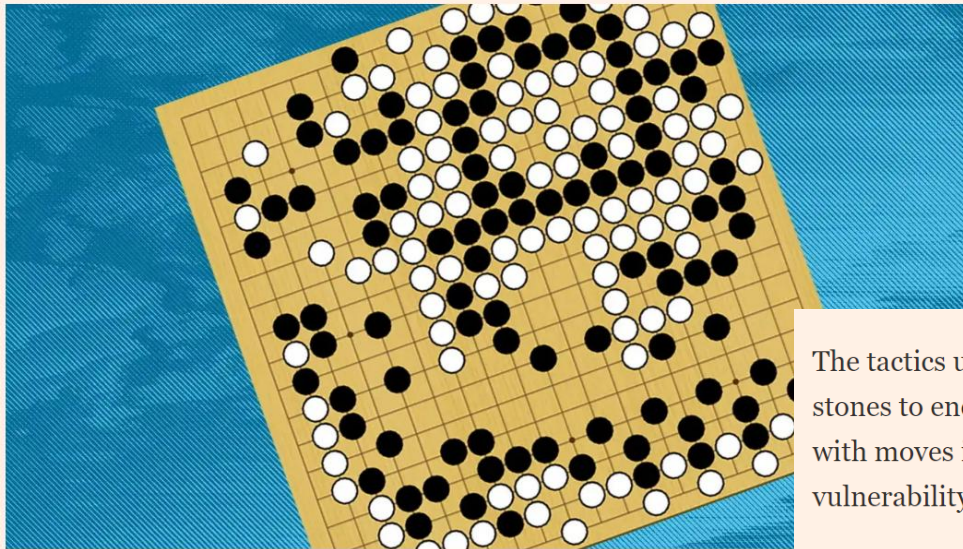
<https://www.deepmind.com/research/highlighted-research/alphago>

# Playing Games: Go – Mankind striking back

## Amateur defeats AI

### Man beats machine at Go in human victory over AI

Amateur Kellin Pelrine exploited weakness in systems that have otherwise dominated board game's grandmasters



Kellin Pelrine won 14 of 15 games against one of the top Go-playing systems in January © FT montage

Richard Waters in San Francisco FEBRUARY 17 2023

The tactics used by Pelrine involved slowly stringing together a large “loop” of stones to encircle one of his opponent’s own groups, while distracting the AI with moves in other corners of the board. The Go-playing bot did not notice its vulnerability, even when the encirclement was nearly complete, Pelrine said.

“As a human it would be quite easy to spot,” he added.

The discovery of a weakness in some of the most advanced Go-playing machines points to a fundamental flaw in the deep learning systems that underpin today’s most advanced AI, said Stuart Russell, a computer science professor at the University of California, Berkeley.

Source: Financial Times

<https://www.ft.com/content/175e5314-a7f7-4741-a786-273219f433a1>

# Generative KI: Large Language Models (LLM), z.B. als Basis von ChatGPT

ChatGPT ▾

🔗 Gemeinsam nutzen

Ich halte eine zweitägige Schulung zu Grundlagen des Machine Learning. Schreibe 4-5 einführende und motivierende Sätze für die Teilnehmerinnen und Teilnehmer.



📅 Erinnerung aktualisiert

Herzlich willkommen zur zweitägigen Schulung über die Grundlagen des Machine Learning! In den nächsten Tagen werden wir gemeinsam in die faszinierende Welt der Künstlichen Intelligenz eintauchen und Schritt für Schritt verstehen, wie Maschinen Muster erkennen und Entscheidungen treffen können. Machine Learning ist längst nicht mehr nur Theorie, sondern findet in zahlreichen Bereichen unseres Alltags Anwendung – von der Sprachübersetzung bis zur Bildanalyse. Unser Ziel ist es, dass Sie am Ende der Schulung die Grundprinzipien verstehen und eine solide Basis haben, um selbst weiter zu experimentieren und zu lernen. Lassen Sie uns neugierig sein und gemeinsam diese spannende Reise beginnen!

🔊 📄 👍 🗨️ ↺ ▾



Sende eine Nachricht an ChatGPT



ChatGPT kann Fehler machen. Überprüfe wichtige Informationen.

# Generative KI: Text-to-image models (Fortführung von vorheriger Folie...)

ChatGPT ▾

🔗 Gemeinsam nutzen

Lassen Sie uns neugierig sein und gemeinsam diese spannende Reise beginnen!

Thanks. Now create a corresponding image.



Here's an image to set the scene for your Machine Learning training. Let me know if there's anything else you'd like to adjust!





---

# FUNDAMENTALS MACHINE LEARNING

# Introduction

## Data types

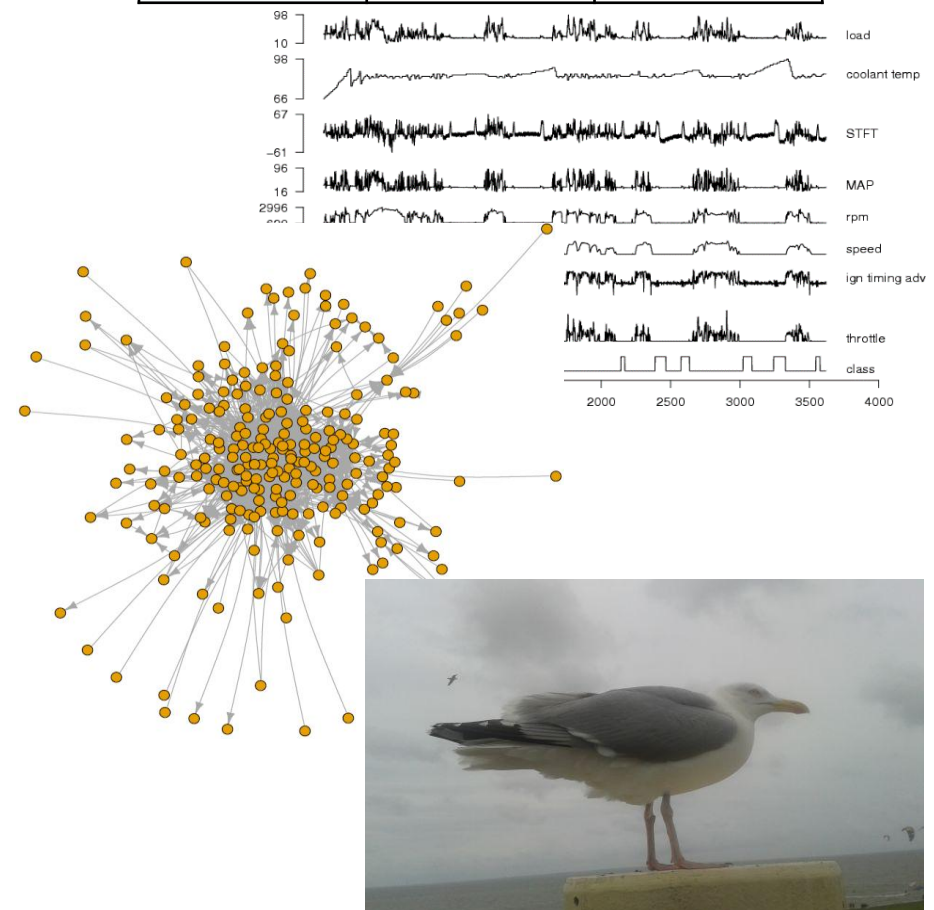
What kind of data can Machine Learning be applied on?

- ▶ Any kind of data that is meaningful for an application.

Some examples:

- ▶ relational databases, data warehouses
  - ▶ *e.g. customer database, spare parts, products in production line*
- ▶ time-related data
  - ▶ *e.g. technical measurements, stock exchange data*
- ▶ multimedia data
  - ▶ *e.g. text, audio, image, video data*
- ▶ graph and network data
  - ▶ *e.g. social networks*

Name	hair	department
Peter	brown	research
Stephan	black	development
Jens	brown	development



# Introduction

## Applications

---

### Some applications

- ▶ Fault detection
  - ▶ e.g. in production lines or the automotive industry
- ▶ Production lines
  - ▶ optimization of production times
- ▶ Mobility solutions
  - ▶ e.g. in driver assistant systems, autonomous driving, connected vehicles (predictive maintenance)
- ▶ Customer relationship management (CRM):
  - ▶ find customers to be addressed by a campaign
  - ▶ approach customers with specific, goal-oriented offers
- ▶ Medical engineering
  - ▶ e.g. cancer diagnosis, analysis of ECG data, ...
- ▶ Personalized Online Advertising
  - ▶ e.g. Facebook, Amazon, ...
- ▶ Speech processing
  - ▶ e.g. App “Shazam”
- ▶ Web search
  - ▶ find documents, movies, etc... (google, ...)

# Fundamentals

## Terms and definitions

---

### Data objects or instances


- ▶ data sets are made up of so-called “**instances**” or “**data objects**”
- ▶ other names are: samples, records, tuples, data instances, ...
- ▶ examples of data instances:
  - ▶ *vehicles* in an automotive database
  - ▶ *patients* in a medical database
  - ▶ *customers* in a sales database

# Fundamentals

## Terms and definitions

### Attributes or features

- ▶ instances are described by “**attributes**” or “**features**”
- ▶ features/attributes describe an instance, e.g.
  - ▶ *number cylinders, maximum power, ...* for a vehicle
  - ▶ *blood type, age, ...* for a patient
  - ▶ *age, gender, size, ...* for a customer
- ▶ **a set of features describing one instance is called a “feature vector”**
- ▶ e.g. in a database
  - ▶ data instance = row
  - ▶ feature = column



	feature <sub>1</sub>	feature <sub>2</sub>	...	feature <sub>d</sub>
instance <sub>1</sub>				
⋮				
instance <sub>N</sub>				

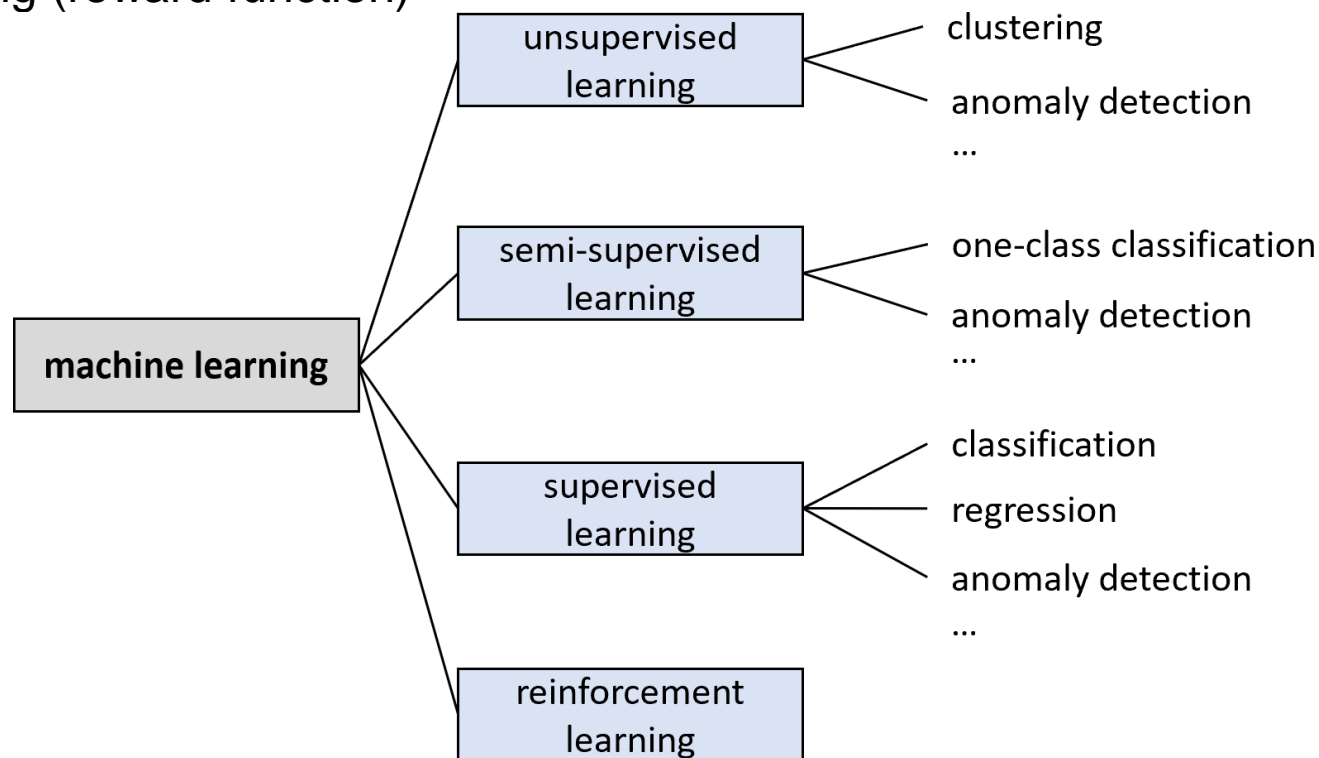
# Fundamentals

## Types of machine learning

---

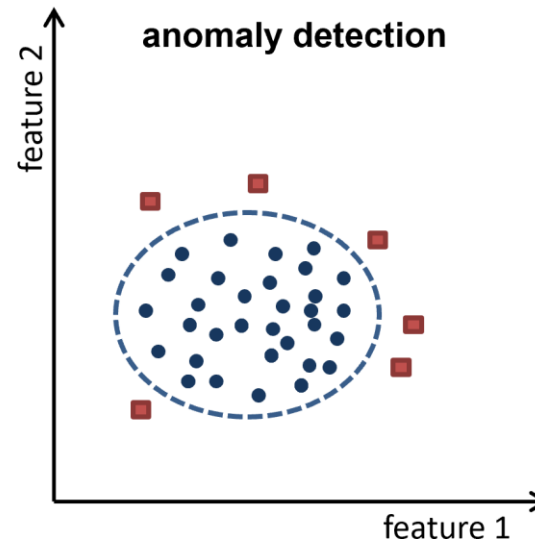
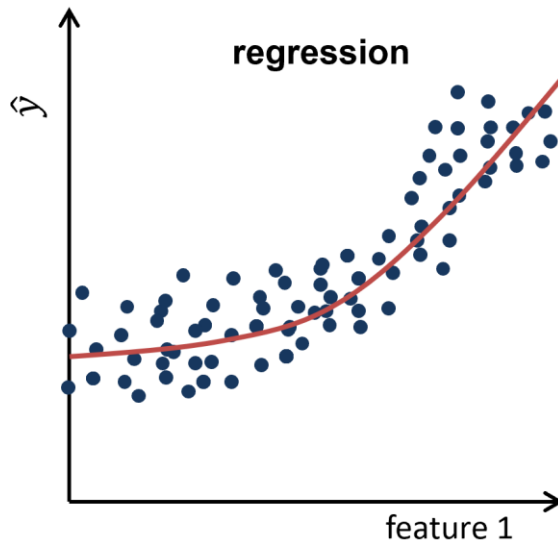
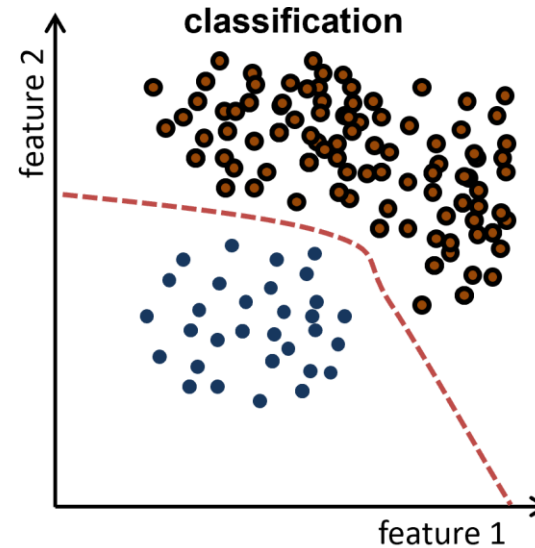
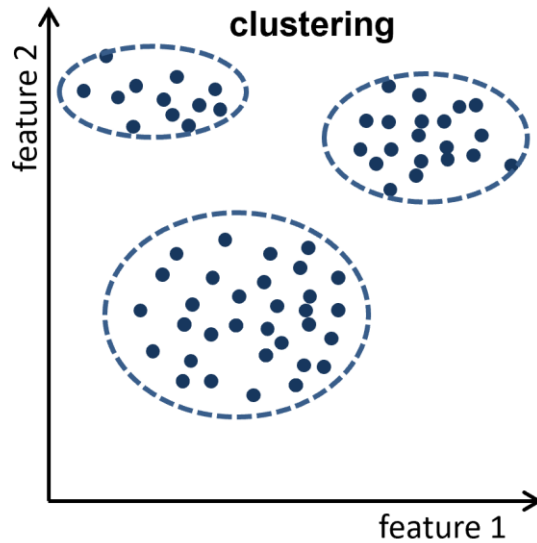
### General types of machine learning:

- ▶ unsupervised learning (no labels/target values)
- ▶ semi-supervised learning (some labels/target values)
- ▶ supervised learning (labels/target values are known)
- ▶ reinforcement learning (reward function)



# Fundamentals

## Machine Learning: Summary



### Machine learning: learning from sample data

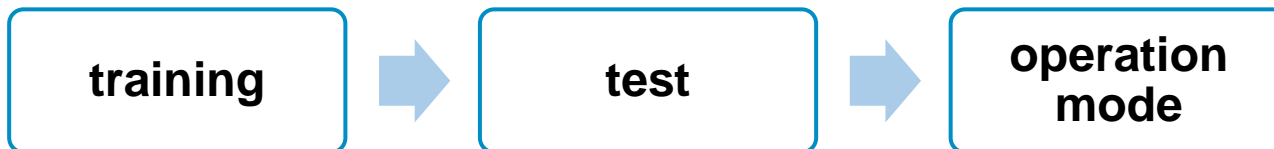
➤ **Definition “learning system”:**

A system is said to have the ability to learn if for a given task  $T$ , the performance  $P$  improves with experience  $E$  (Mitchell, 1997).

**For classification this general definition becomes:**

- **task  $T$ :** classification of instances
- **performance measure  $P$ :** expresses a system's ability to classify unseen data instances correctly
- **experience  $E$ :** the ability for generalization from a training set

➤ **Machine learning systems have three phases/stages:**





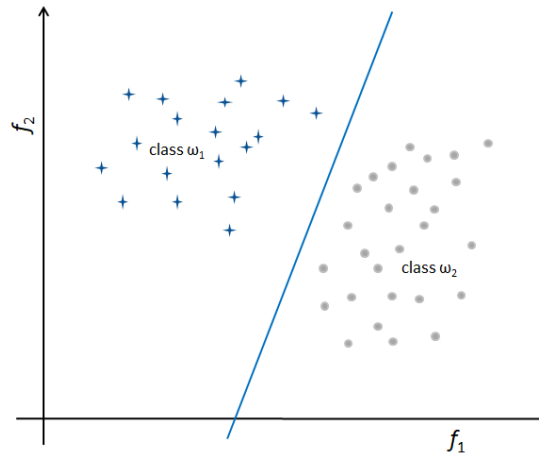
# Fundamentals

## Machine learning: training a model

---

- ▶ Machine Learning is about fitting a model to a training data set (train set)
- ▶ the model has parameters, these are changed until the model describes the data set
- ▶ e.g. a straight line could be used to separate a two-dimensional data set of two classes („two-class classification“) or a line could be used as a linear regression:

Remember: a line is given by  $y = mx + b$  or  $y = w_0 + w_1x$



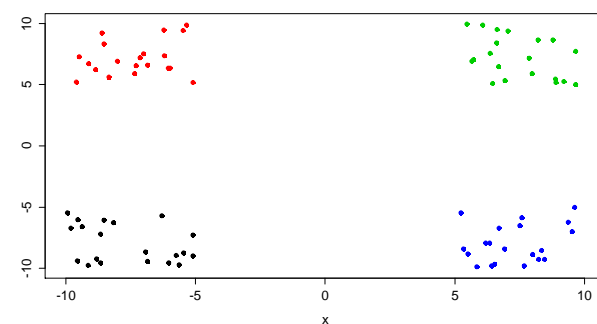
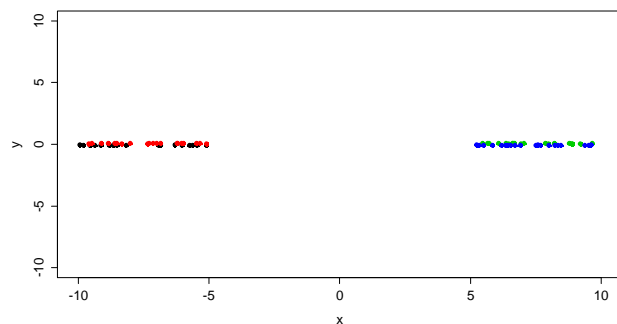
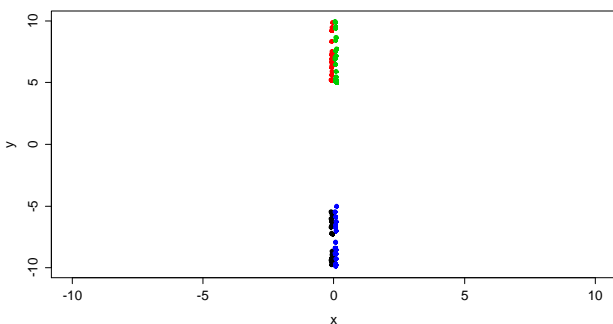
The underlying principle: During training of the machine learning model, the parameters  $w_0$  and  $w$  would be adapted until the line separates the data set

# Fundamentals

## Data preparation

---

- ▶ Usually, numeric features should be normalized/scaled, in order to equally weight all features (e.g. to a common scale of  $[0,1]$  or using z-score) if you apply algorithms
  - ▶ some algorithms do not require normalization
  - ▶ if you do not normalize, there should be a reason for that
- 
- ▶ see for example the effect of not scaling before clustering:
    - ▶ the same data set scaled differently:
    - ▶ How many groups in the data do you see in plot (1), (2), (3) ?



# Fundamentals

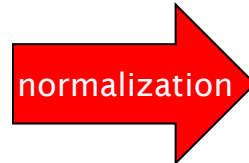
## Data preparation

- ▶ **Scaling:** Scale the attributes such that they fall within a smaller range and such that each attribute becomes a priori equally important.

*E.g.: scale the attribute values to  $[0; 1]$ , or such that the attribute has zero mean and standard deviation equal to one.*

- ▶ **min-max normalization:**  $\text{domain}(X) \rightarrow [0,1]$ ,  $x_i \mapsto \frac{x_i - \min_X}{\max_X - \min_X}$
- ▶ **z-score standardization:**  $\text{domain}(X) \rightarrow \mathbb{R}$ ,  $x_i \mapsto \frac{x_i - \text{mean}(X)}{\text{sd}(X)}$

Name	gender	height [cm]	weight [kg]
Peter	M	185	82.5
Maria	F	161	53.0
Emma	F	172	61.3



Name	gender	height [cm]	weight [kg]
Peter	0	1.00	1.00
Maria	1	0.00	0.00
Emma	1	0.46	0.28

# Fundamentals

## Data preparation

---

### Data Preparation: Feature extraction

Often it is better to not work with the raw data but instead extract features.

Examples:

- ▶ statistical features such as
  - ▶ „central tendency“: mean value, median, ...
  - ▶ measures of „dispersion“: standard deviation / variance
- ▶ from an image we could extract the colours, the shapes, etc.  
(not commonly done nowadays => more likely Deep learning models are used for images)
- ▶ from text we could extract the number of words, find the most frequent words, etc.
- ▶ from a time series we could extract shapes like „rising“, „falling“ etc.
- ▶ this process is called „feature engineering“
- ▶ feature engineering is an important step in machine learning, but:
- ▶ In Deep Learning the step of feature engineering is completely or partially done by the model

# Distance measures

## Similarity and dissimilarity

---

For a data set of  $N$  instances, where the instance are denoted by  $o_1, o_2, o_3, \dots, o_N$  the dissimilarities between all instances can be expressed using a **dissimilarity matrix**

$$\begin{pmatrix} d(o_1, o_1) & d(o_1, o_2) & d(o_1, o_3) & \cdots & d(o_1, o_N) \\ d(o_2, o_1) & d(o_2, o_2) & d(o_2, o_3) & \cdots & d(o_2, o_N) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d(o_N, o_1) & d(o_N, o_2) & d(o_N, o_3) & \cdots & d(o_N, o_N) \end{pmatrix}$$

where  $d(o_1, o_2)$  is the distance between the instance  $o_1$  and  $o_2$   
(note: the matrix is symmetric)

# Distance measures

## Dissimilarity of numeric attributes

### Dissimilarity of numeric features

**Euclidean** distance:

$$d_E(o_i, o_k) = \sqrt{\sum_{j=1}^p (x_{ij} - x_{kj})^2}$$

in two dimensions this corresponds to the  
pythagoras:

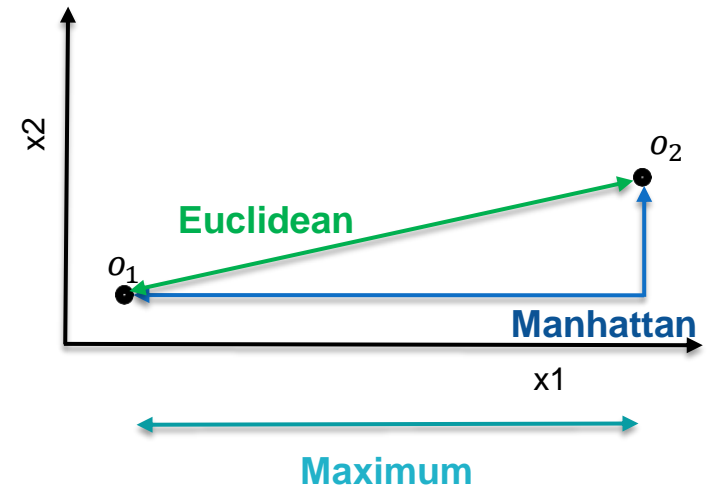
$$d_E(o_1, o_2) = \sqrt{(o2_{x1} - o1_{x1})^2 + (o2_{x2} - o1_{x2})^2}$$

**Manhattan**/City-Block distance:

$$d_M(o_i, o_k) = \sum_{j=1}^p |x_{ij} - x_{kj}|$$

where  $x_{ij}$  denotes feature  $j$  of instance  $i$ , and  $p$  is the number of features

**Example of distance measures:**



**Maximum**/Chebychev distance:

$$d_{Max}(o_i, o_k) = \max(x_{ij} - x_{kj}) \text{ over all } j$$

# Distance measures

## Time series distance measures

### Example: time series distance measure „Euclidean distance“

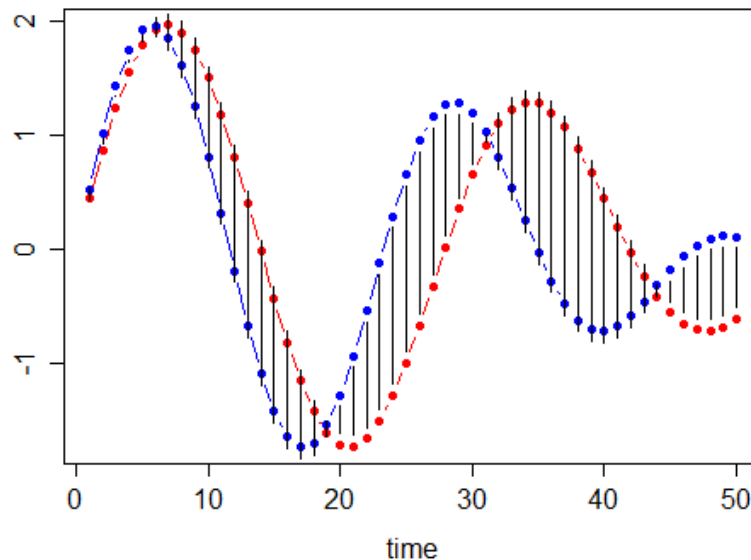
The Euclidean distance between two time series  $X_{T1}$  and  $X_{T2}$  is calculated with this equation:

$$\text{dist}(X_{T1}, X_{T2}) = \sqrt{\sum_{j=1}^T (X_{T1i} - X_{T2i})^2}$$

where  $X_{T1i}$  is one data point of time series  $X_{T1}$ .

The distances at each time point are squared and summed up, followed by the square root.

- ▶ time series  $X_{T1}$  and  $X_{T2}$  need to have the same length
- ▶ Euclidean distance is sensitive to misalignments of the two time series



# Distance measures

## Time series distance measures

---

### **Example: time series distance measure „Dynamic time warping“ (DTW)**

For each data point in time series  $X_{T1}$  DTW finds the closest data point in the time series  $X_{T2}$ .

This is done by:

1. creating a matrix of size  $T \times T$  with all distances between the data points
  2. trying to find the path through the matrix, that has the minimal distance („warping path“)
- due to the computational costs, typically not the full matrix is created but rather a subset around the diagonal and not necessarily the absolute minimum is searched
  - DTW can handle misaligned time series
  - DTW can handle time series that are not of equal length
-



# Distance measures

## Time series distance measures

### Dynamic time warping (DTW)

