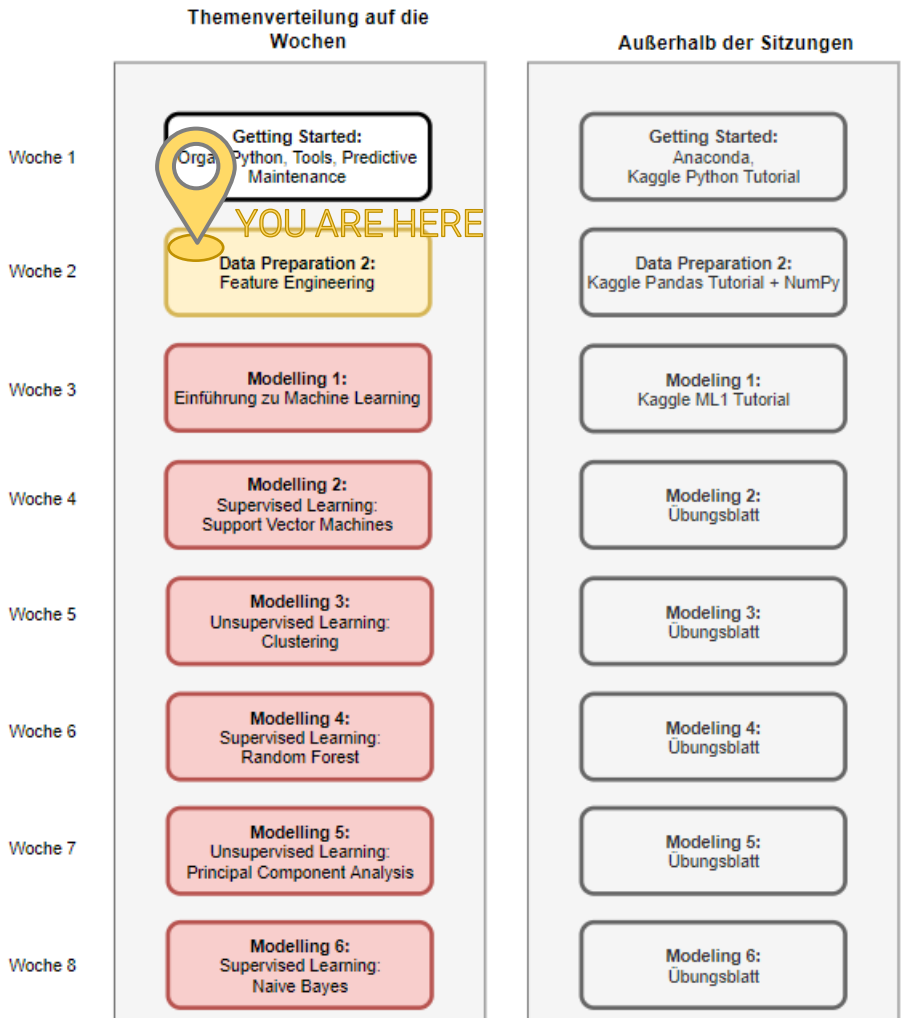
Data Preparation

Feature Engineering: Making data available to models

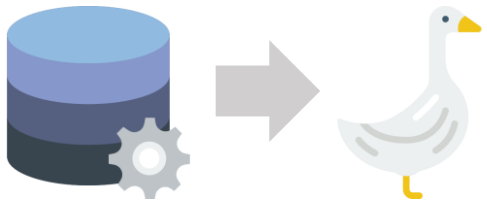
Wo sind wir?


Themenverteilung auf die Wochen
Außerhalb der Sitzungen
Woche 1
Woche 2
Woche 3
Woche 4
Woche 5
Woche 6
Woche 7
Woche 8
Getting Started: Orga, Python, Tools, Predictive Maintenance
YOU ARE HERE
Data Preparation 2: Feature Engineering
Modelling 1: Einführung zu Machine Learning
Modelling 2: Supervised Learning: Support Vector Machines
Modelling 3: Unsupervised Learning: Clustering
Modelling 4: Supervised Learning: Random Forest
Modelling 5: Unsupervised Learning: Principal Component Analysis
Modelling 6: Supervised Learning: Naive Bayes
Getting Started: Anaconda, Kaggle Python Tutorial
Data Preparation 2: Kaggle Pandas Tutorial + NumPy
Modeling 1: Kaggle ML1 Tutorial
Modeling 2: Übungsblatt
Modeling 3: Übungsblatt
Modeling 4: Übungsblatt
Modeling 5: Übungsblatt
Modeling 6: Übungsblatt

Agenda

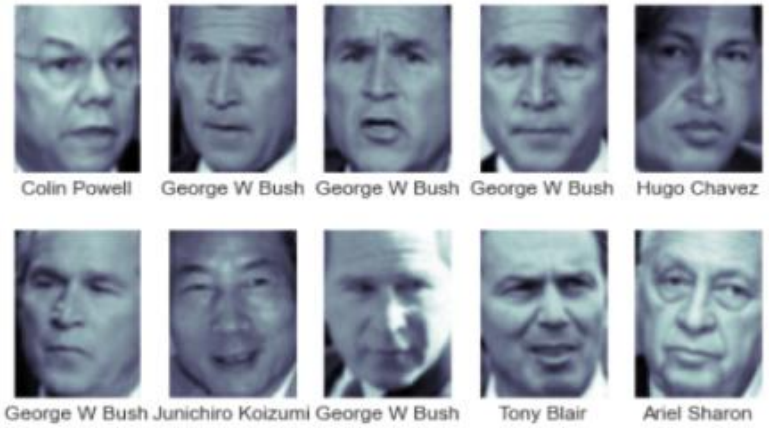
1. Was ist ein Feature? Vom Daten- in den Feature-Raum
2. Feature-Extraktion
3. Feature-Selektion
4. Feature Scaling und Normalisierung

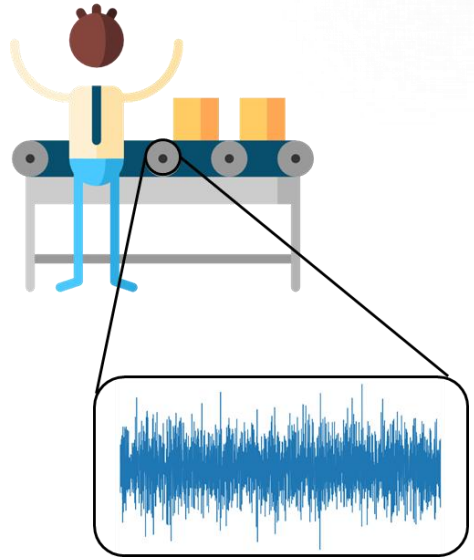
Von Daten zu Features



- Wenn wir in das Gebiet Machine Learning einsteigen, dann werden wir den Begriff „Feature“ sehr häufig hören
- Sie können sich schon jetzt merken: ein Machine Learning Modell nutzt **Features**, um etwas – die sog. **Labels** – vorherzusagen
- Sie werden jetzt dann sehen, dass Features im Grunde nichts anderes sind als **Arrays bzw. DataFrames**

Ihre Intuition: was sind hier „Features“?

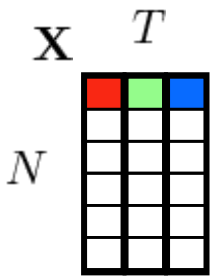




So what?

Features sind also (auch übersetzt) Merkmale – etwas, das zusammenfassend und charakterisierend bzgl. der zugrundeliegenden Daten ist

Was ist ein Feature?



Was denken Sie?

Welche Eigenschaften müssen Features haben?

In Python würden wir für die Dimensionen unserer Feature-Matrix schreiben:
`[n_samples, n_features]`

So what?
- Den meisten Aufwand im Data Science Prozess hat man mit Datenbereinigung und Feature-Engineering
- Modellierung macht gefühlte 20% aus

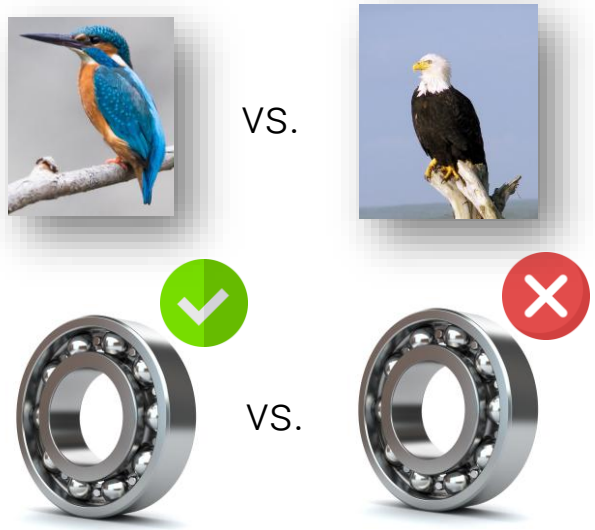
Definition

"In machine learning and pattern recognition, a **feature** is an **individual measurable property or characteristic of a phenomenon** being observed."
→ **Operationalisierung!**

- Features können also sowohl die **Daten an sich** sein, als auch davon **abgeleitete Größen**
- Meist spricht man aber von Features, wenn die Daten auf eine bestimmte Weise **aggregiert** wurden → in der Regel **verringert** man durch Feature-Engineering schon die **Dimensionalität** des Problems!
- Ganz strikt formuliert: Feature-Engineering bildet eine messbare Eigenschaft auf einen **Skalar** ab
- Daher ist der Begriff Feature oft nicht scharf abzugrenzen
- Man könnte auch sagen: aus den Rohdaten erzeugt man im Prozess des **Feature-Engineering** eine **Feature-Matrix** mit sauberem Input für unser Modell
- Der Vorgang des Feature-Engineering ist kritisch für die anschließende **Modellgüte**

Welche Eigenschaften müssen Features haben?

Frequenzspektrum



Features müssen **informativ** sein bzw. genügend Informationen über das zugrundeliegende Phänomen tragen

Features müssen **diskriminativ** sein und somit Unterschiede bzw. Strukturen in den Daten erklären können

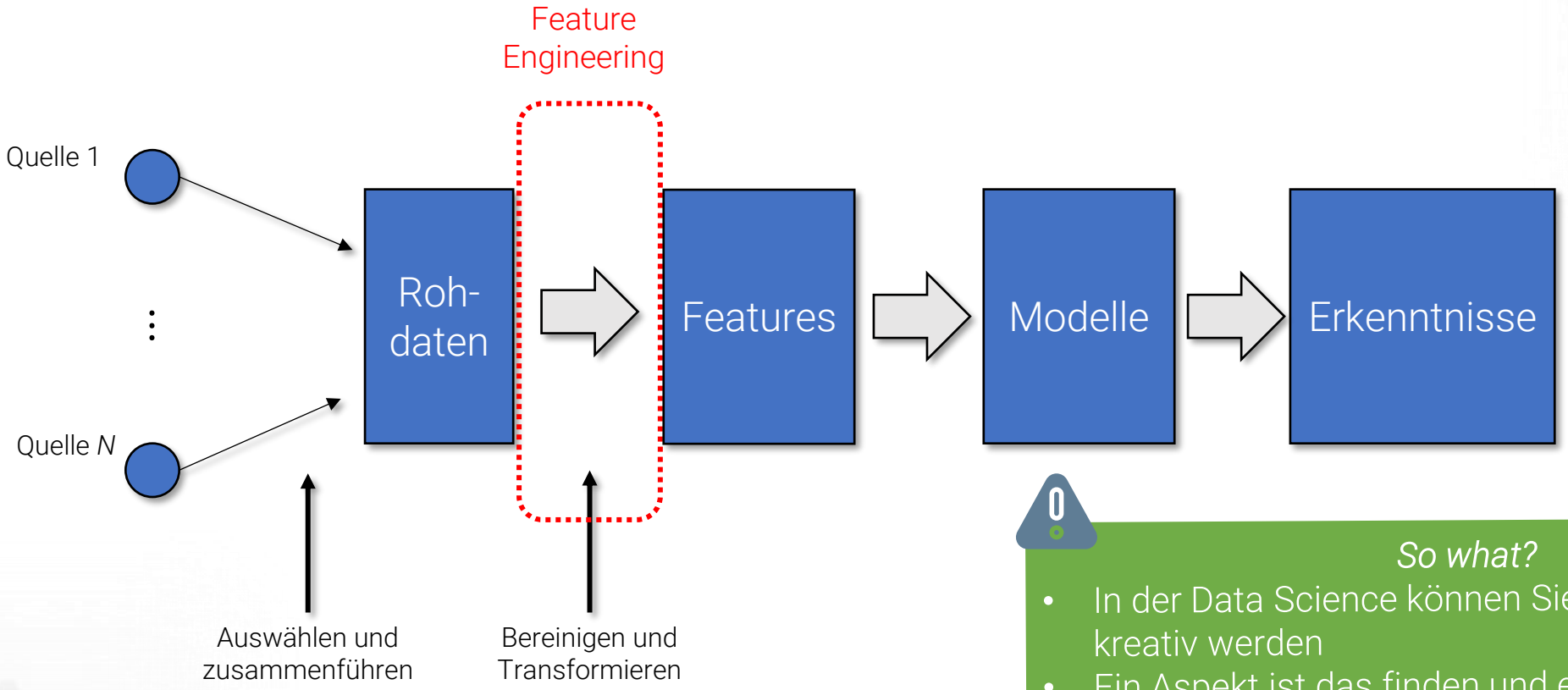
Features dürfen **nicht** zu stark miteinander **korrelieren**, um Redundanz zu vermeiden

So what?

Auch die **Anzahl** an genutzten Features ist wichtig:

- Zu wenig Features: das Modell kann das Phänomen nicht beschreiben
- Zu viele Features: das Modell ist schwierig anzulernen

Feature Engineering: Bindeglied zwischen Daten und Erkenntnissen

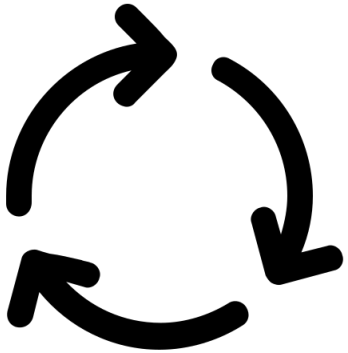


So what?

- In der Data Science können Sie an vielen Stellen kreativ werden
- Ein Aspekt ist das finden und erzeugen von geeigneten Features
- Im Erzeugen von Features steckt Domänenwissen → Wir unterstützen das Modell durch unser Wissen

Iterativer Prozess: Modell vs. Features

Iterativer Prozess



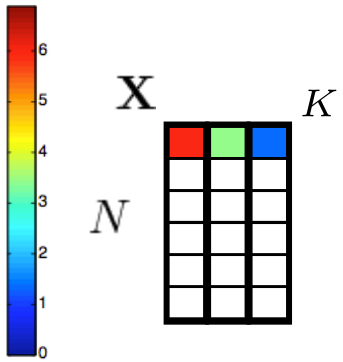
Features

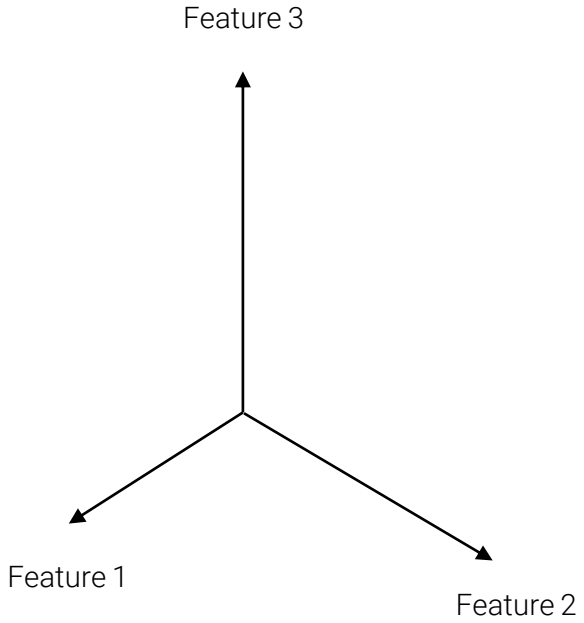
- Features bzw. Feature-Engineering sind nicht leicht zu generalisieren
- Geeignete Features hängen sowohl vom zu beschreibenden Phänomen als auch vom gewählten Modell ab

Modell

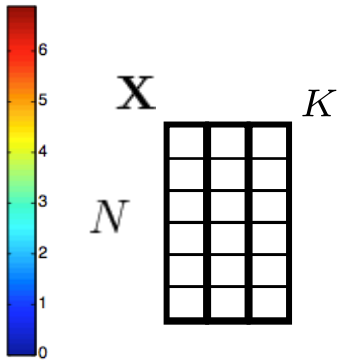
- Die Güte des gewählten Modells wiederum hängt maßgeblich von den gewählten Features und deren Anzahl ab
- Modell und Features beeinflussen sich also gegenseitig

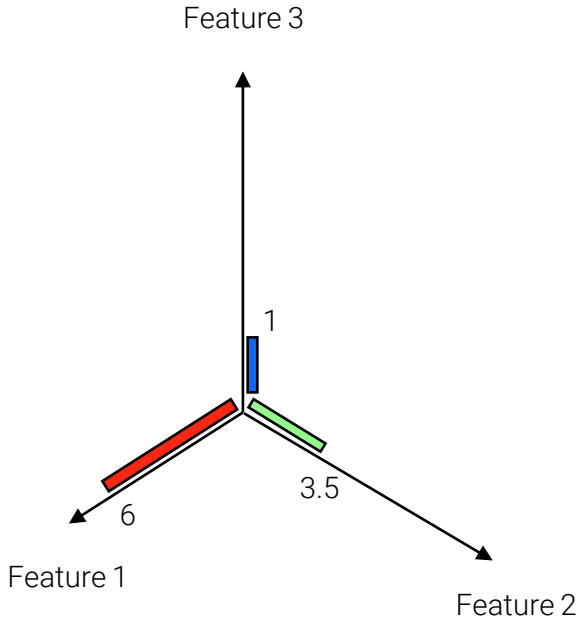
Von der Feature-Matrix in den Feature-Raum – und zurück





Von der Feature-Matrix in den Feature-Raum – und zurück





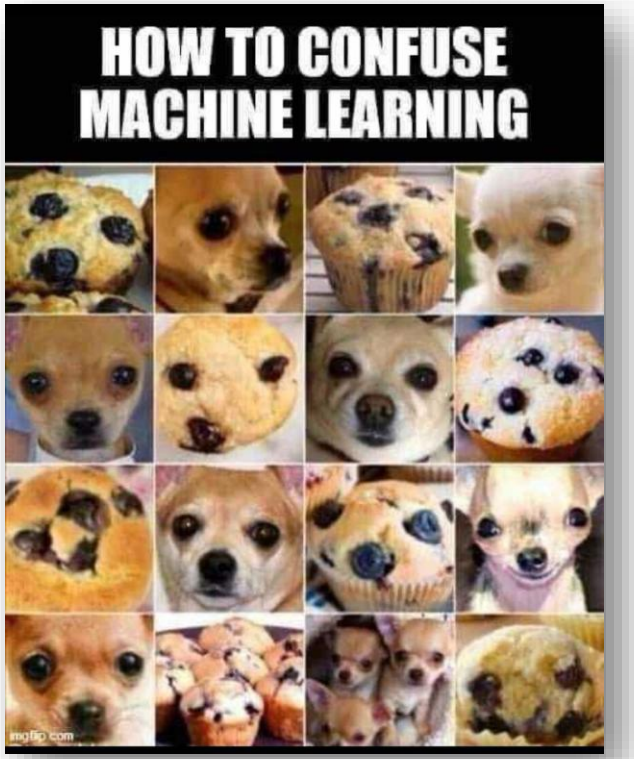
So what?

Vorsicht Verwechslungsgefahr: jetzt wird aus einer Datenmatrix eine Feature-Matrix – oft verwendet man für beides in der Schreibweise ein großes, dick gedrucktes X!

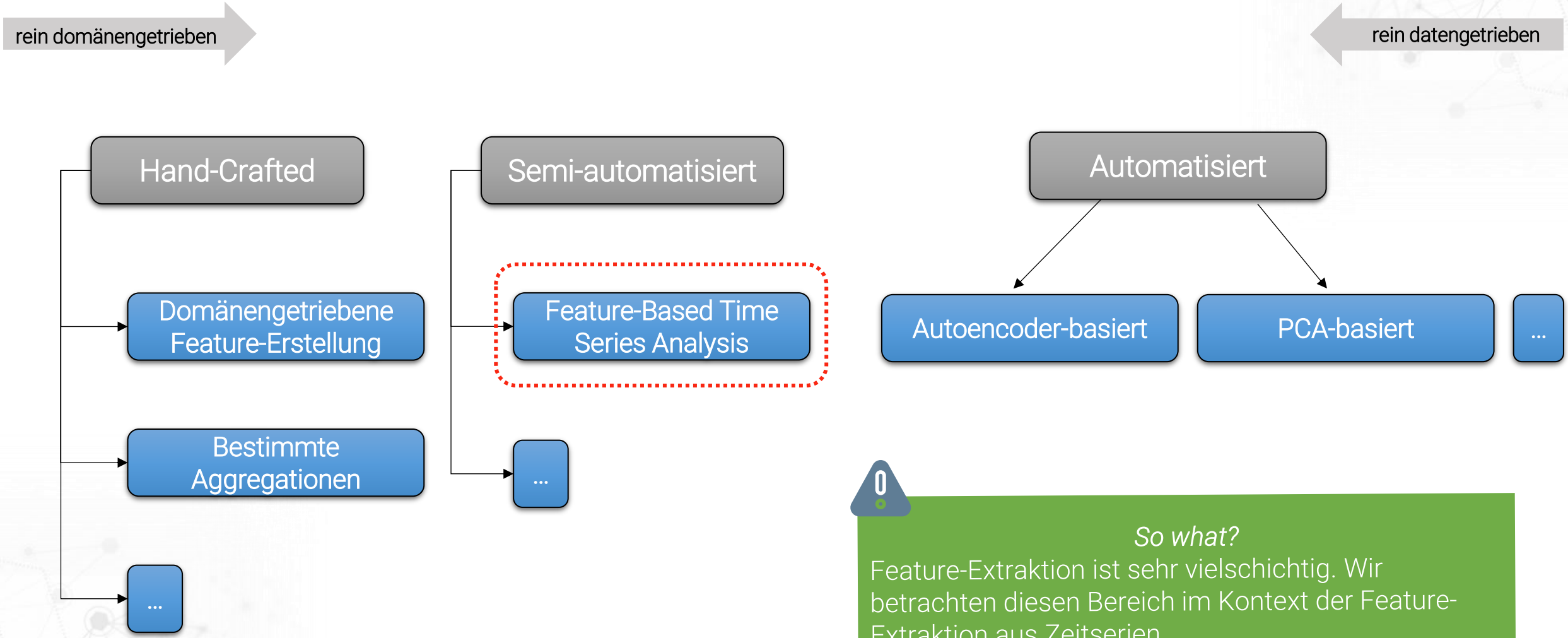
Feature Extraction

Phänomen → Skalar

Daten → Skalar



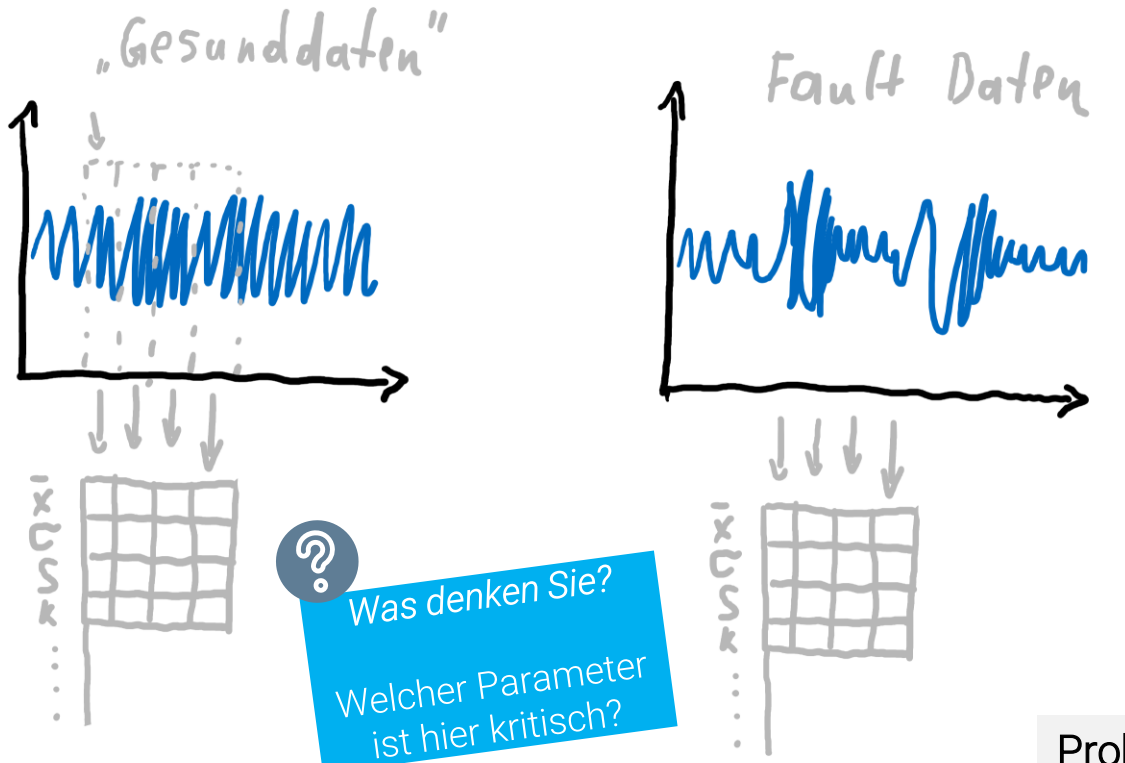
Arten der Feature-Extraktion und Features



So what?

Feature-Extraktion ist sehr vielschichtig. Wir betrachten diesen Bereich im Kontext der Feature-Extraktion aus Zeitserien.

Feature-Extraktion aus Zeitserien



- Stellen Sie sich vor, Sie wollen unterscheiden, ob ein Motor einen Schaden hat oder nicht
- Das tun Sie z.B. anhand von aufgezeichneten Stromdaten
- Wir könnten zum einen Charakteristiken der **gesamten** Zeitserien extrahieren
- Oder von **Intervallen** der jeweiligen Zeitserien

Was denken Sie?

Welcher Parameter ist hier kritisch?

Was denken Sie?

Wie könnten wir nun Charakteristiken dieser Zeitserien erzeugen?

So what?

Achtung: es gibt (mind.) zwei große Gruppen an Zeitseriendaten.

- Event-related/triggered: es existieren ausgezeichnete Zeitpunkte anhand derer referenziert werden kann
- Kontinuierliche: fehlen solcher Zeitpunkte

Probleme, die dadurch behoben werden:

- Domänenwissen fehlt → nach was soll ich suchen?
- Datenumfang zu groß
- In welchen **Segmenten** der Zeitserien befinden sich die **Charakteristika**?
- Bei Zeitserien: ein Startpunkt liegt in den wenigsten Fällen vor! → die extrahierten Charakteristika sollen zu **beliebigen Zeitpunkten diskriminativ** sein!

Mathematische Formulierung

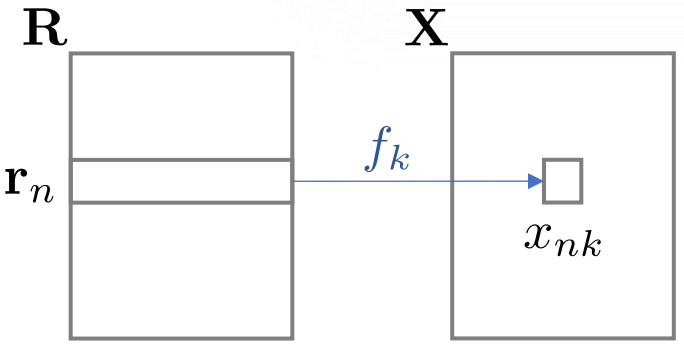
Was denken Sie?

Wie könnte man die Transformation von Rohdaten in Features generisch formulieren?

Rohdaten $\mathcal{R} := \{\mathbf{r}_1, \ldots, \mathbf{r}_N\}$ $\mathbf{r}_n \in \mathbb{R}^T$ $\mathbf{R} \in \mathbb{R}^{N \times T}$

Abbildung $\mathbf{f} : \mathbb{R}^T \to \mathbb{R}^K$ $f_k : \mathbb{R}^T \to \mathbb{R}$

Featurevektoren $\mathcal{X} := \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ $\mathbf{x}_n \in \mathbb{R}^K$ $\mathbf{X} \in \mathbb{X}^{N \times K}$

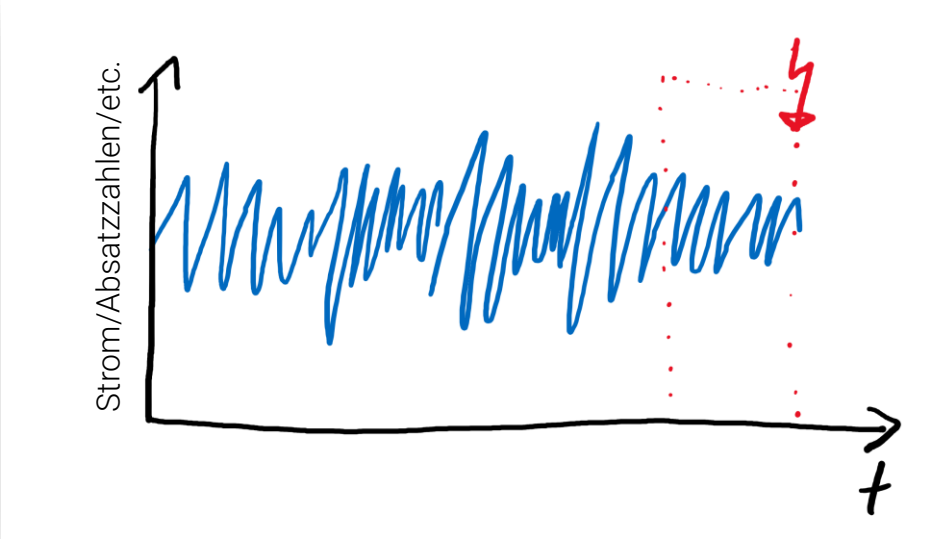


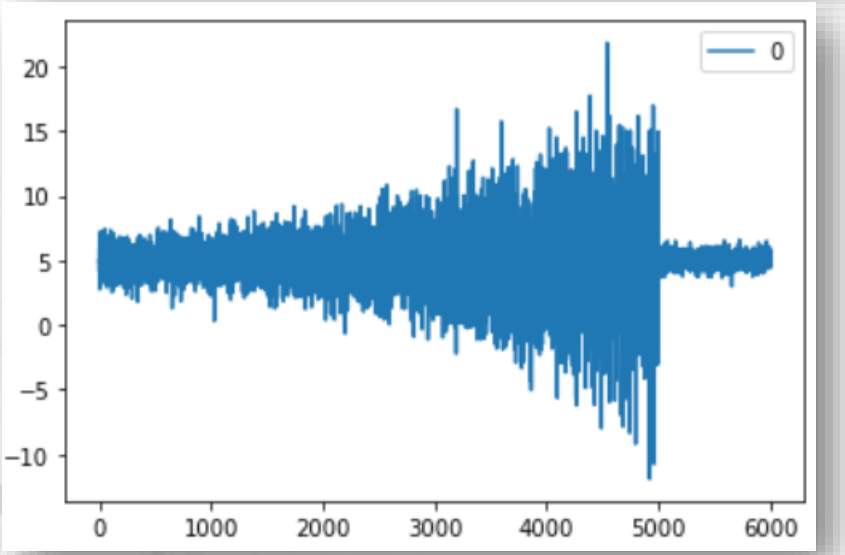
So what?

Hinter Feature Extraction steht also eine **vektorwertige Funktion**, die vom Raum der Rohdaten in den Feature-Raum abbildet.

Stift, Zettel, kurz mit Nachbarn austauschen

Anwendungsfall: Vorhersage Motorausfall, Einbruch Absatz eines Produkts





- Neben dieser Klassifikationsaufgabe, können auch kontinuierliche Labels – also Regressionsprobleme mittels dieser Feature-Extraktion aus Zeitserien angegangen werden

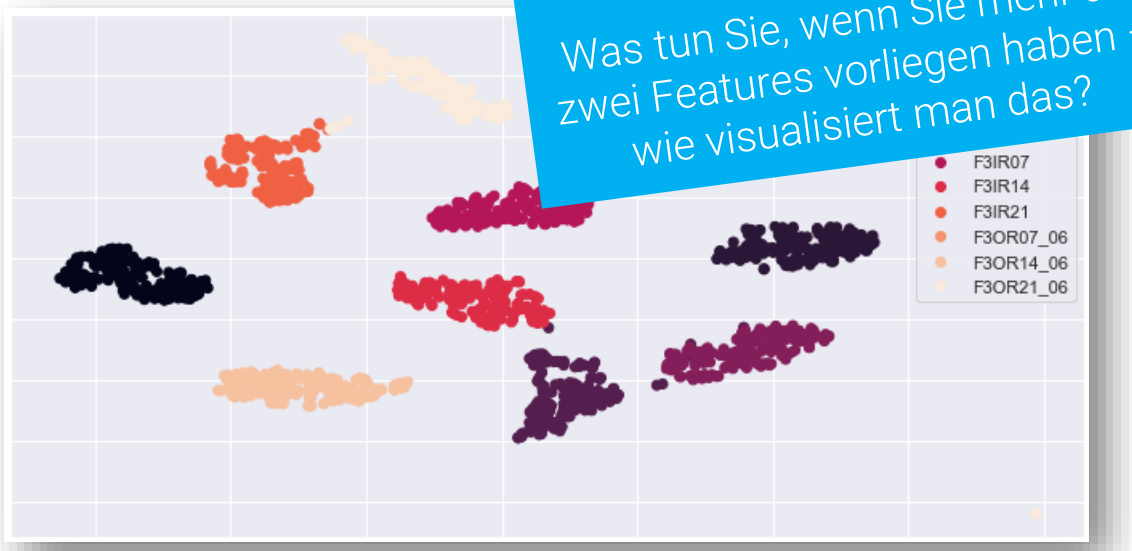
- Wenn man z.B. eine Zeitserie vor und nach einem Event (Motorausfall, Einbruch im Absatzverhalten, etc.) vorliegen hat → Extrahierte Features können Aufschluss auf das Event geben!

Demo BirdNET

Visualisierung anhand t-SNE

Was denken Sie?

Was tun Sie, wenn Sie mehr als zwei Features vorliegen haben – wie visualisiert man das?



Grafik erstellt durch ein Team der Projektarbeit „Predictive Maintenance“

- t-SNE: t-Distributed Stochastic Neighbor Embedding
- Nicht-lineare Methode zur Datenexploration
- Projektion hochdimensionaler Daten in den zwei- oder dreidimensionalen Raum
- Grobe Funktionsweise: die Dichten der Daten vom hochdimensionalen Raum müssen auch im niedrigdimensionalen erhalten bleiben
- Abbildung aus Projektarbeit Predictive Maintenance: → 30-dim. Feature-Raum von Vibrationsdaten an einem Motor

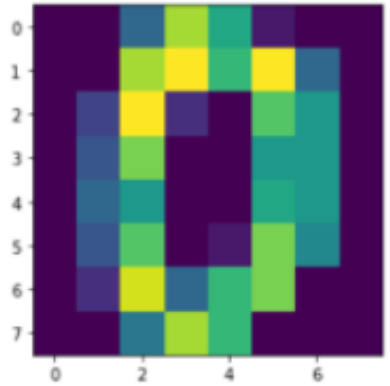
Demo
Tensorflow projector

https://projector.tensorflow.org/

https://distill.pub/2016/misread-tsne/

So what?
Eines der wichtigsten Verständnisse – bzw. die wichtigste Perspektive:
Daten im Feature-Raum!

Beispiel: Embedding Digits



In `sklearn` gibt es ein bekanntes Dataset - `load_digits()`. In ihm liegen uns handgeschriebene Zahlen vor. Jede Zahl wird ursprünglich durch ein Array der Dimension `(8, 8)` repräsentiert. Im Datensatz liegen die Zahlen als Zeilenvektoren in einer großen Datenmatrix vor. Wir wollen uns in diesem Beispiel diesen 64-dimensionalen Datensatz in den zweidimensionalen Raum mittels `TSNE` projizieren.