

Seminar Advances in Deep Learning for Time Series (ADLTS)

Lecture 1: Introduction

Dr. Dario Zanca

Machine Learning and Data Analytics (MaD) Lab
Friedrich-Alexander-Universität Erlangen-Nürnberg
17.10.2024

Team: FAU & PUCV



Dr. Dario Zanca (FAU)
dario.zanca@fau.de



Naga Venkata Sai Jitin Jami, M. Sc. (FAU)
jitin.jami@fau.de



Dr. Christoffer Loeffler (PUCV)
christoffer.loeffler@pucv.cl



&



PONTIFICIA
UNIVERSIDAD
CATÓLICA DE
VALPARAÍSO

Organisational Information

Seminar Advances of Deep Learning for Time Series (MLTS)

- 5 ECTS
 - Team-based project *(more details on the second lecture)*
 - Evaluation:
 - FAU students: 60% written report, 40% oral presentation
 - PUCV students: 20% code, 40% written report, 40% presentation
-

- I. Introduction
 - II. The Tool Tracking dataset
 - III. DL for Time Series
 - IV. Time-aware models
 - V. XAI for Time Series - part 1
 - VI. Active Learning for Time Series - part 1
 - VII. Semi-supervised Learning
 - VIII. Domain-shifts, Ethics, and Bias
 - IX. XAI for Time Series - part 2
 - X. Active Learning for Time Series - part 2
-

I. Introduction

II. The Tool Tracking dataset

III. DL for Time Series

IV. Time-aware models

V. XAI for Time Series - part 1

VI. Active Learning for Time Series - part 1

VII. Semi-supervised Learning

VIII. Domain-shifts, Ethics, and Bias

IX. XAI for Time Series - part 2

X. Active Learning for Time Series - part 2

Lectures

- Recordings (one new lecture every week)
- Consultation hours or other activities

Project

- Work in groups (with support of a supervisor)
 - Work on a challenging real-world dataset
 - *(more details on the second lecture)*
-

StudOn Sose 2025:

https://www.studon.fau.de/studon/goto.php?target=lcode_PVu9Hkrc

Google group:

soon

Lecture outline

1. Motivations and real-world examples
2. Definitions and basic properties
3. Types of ML
4. ML Pipeline
5. ML Tasks for time series



ADLTS \ Introduction \ Motivations



An old history of time series analysis: Babylonian astronomical diaries

VII century B.C.

“[...] Night of the 5th, beginning of the night, the moon was 2 ½ cubits behind Leonis [...] Night of the 17th, last part of the night, the moon stood 1 ½ cubits behind Mars, Venus was below.”

- Babylonians collected the earliest evidence of periodic planetary phenomena
- Applied their mathematics for systematic astronomic predictions



An old history of time series analysis: Babylonian astronomical diaries

Nowadays, thousands of ground-based and space-based telescopes^(a) generate new knowledge every night.

- The Vera C. Rubin Observatory in Chile is geared up to collect 20 terabytes per night from 2022^(b).
- The Square Kilometre Array, the world's largest radio telescope, will generate up to 2 petabytes daily, starting in 2028.
- The Very Large Array (ngVLA) will generate hundreds of petabytes annually.



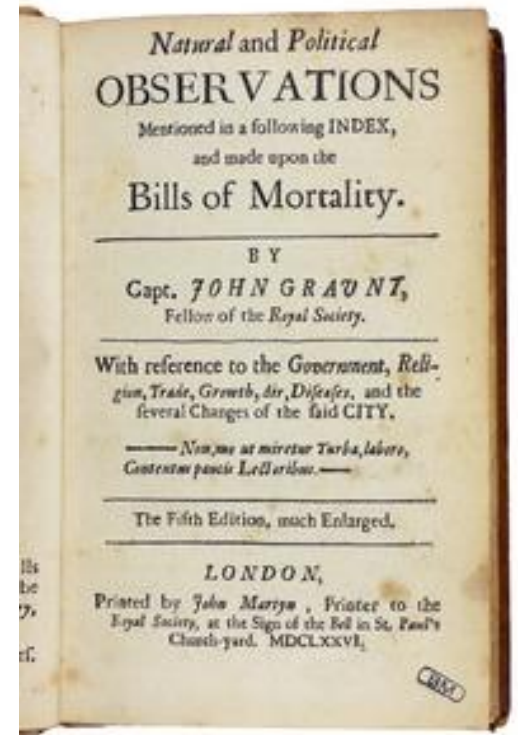
^(a) <https://research.arizona.edu/stories/space-versus-ground-telescopes>

^(b) <https://www.nature.com/articles/d41586-020-02284-7>

An old history of time series analysis: The Birth of Epidemiology

1662, John Graunt describes the data collection:

"When anyone dies, [...] the same is known to the Searchers, corresponding with the said Sexton. The Searchers hereupon...examine by what Disease, or Casualty the corps died. Hereupon they make their Report to the Parish-Clerk, and he, every Tuesday night, carries in an Accompt of all the Burials, and Christnings, hapning that Week, to the Clerk of the Hall."

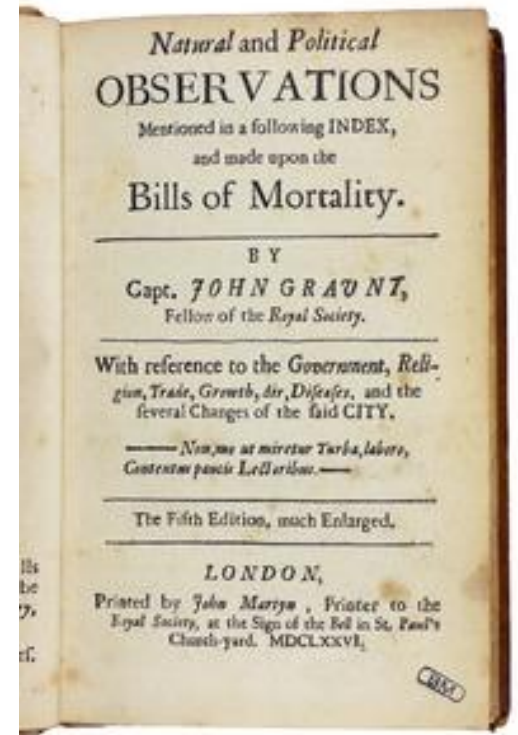


An old history of time series analysis: The Birth of Epidemiology

1662, John Graunt describes the data collection:

"When anyone dies, [...] the same is known to the Searchers, corresponding with the said Sexton. The Searchers hereupon...examine by what Disease, or Casualty the corps died. Hereupon they make their Report to the Parish-Clerk, and he, every Tuesday night, carries in an Accompt of all the Burials, and Christnings, hapning that Week, to the Clerk of the Hall."

- Rudimentary conclusions about the mortality and morbidity of certain diseases
- Graunt's work is still used today to study population trends and mortality



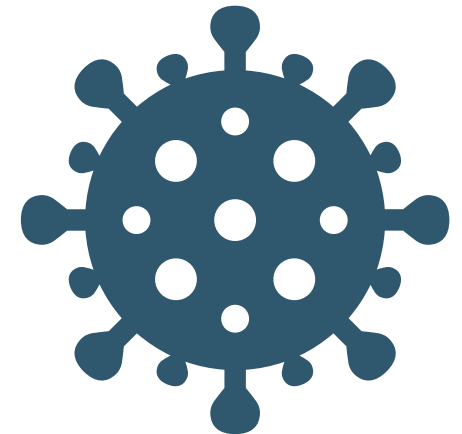
Epidemiology nowadays

Overview of Modern Epidemiology:

- **Data-Driven:** Utilizes large health datasets.
- **Infectious Disease Tracking:** Focus on emerging infections.
- **Genetic and Global Health:** Incorporates genetics and global health issues.

Importance of Time Series Processing:

- **Trend Analysis:** Identifies patterns and seasonality.
- **Prediction & Forecasting:** Models future disease spread and resource needs.
- **Surveillance:** Early detection and intervention monitoring.



Epidemiology

Overview

- Data
- Infection
- infection
- Genes
- global

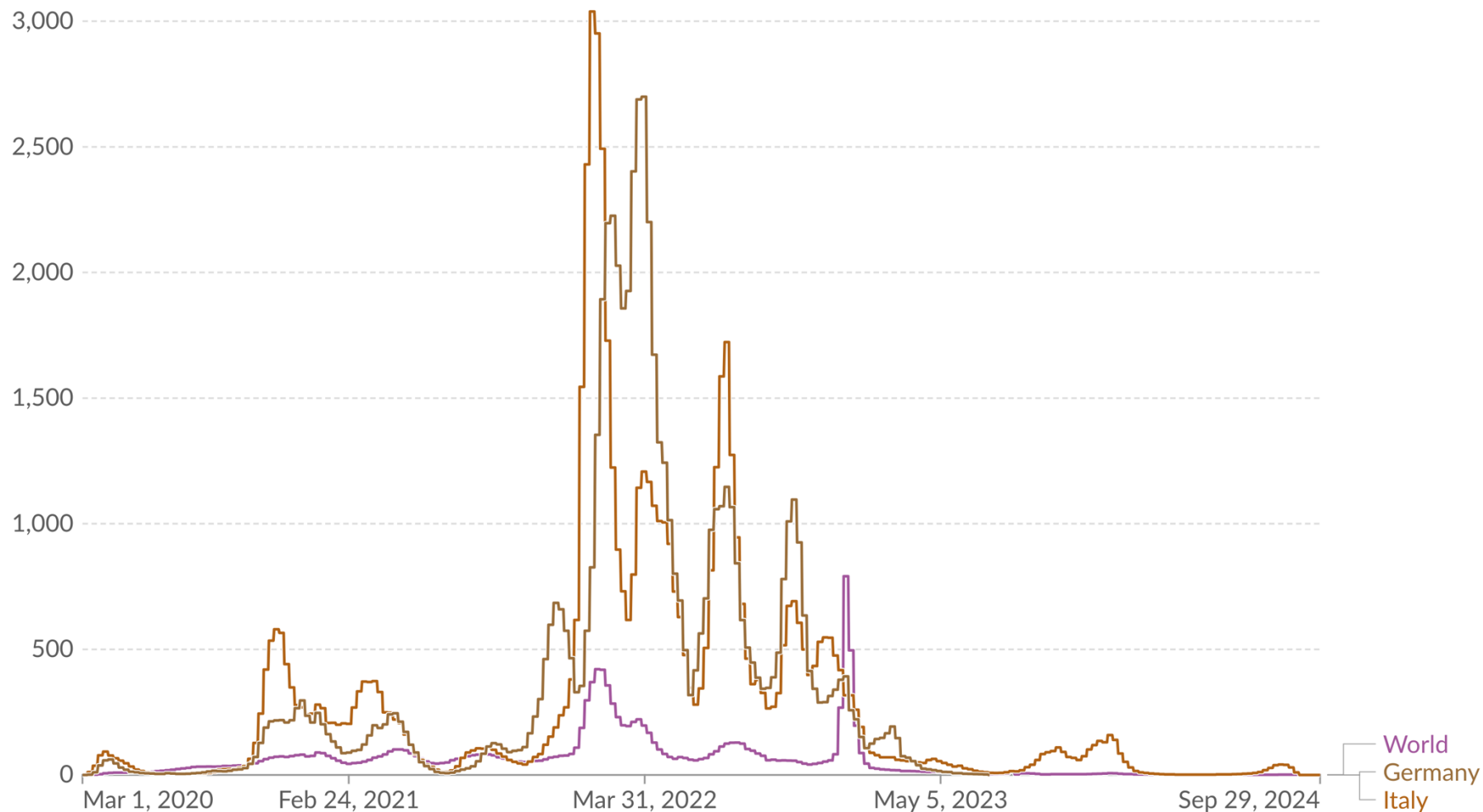
Importance

- Trends
- Predictions
- and risk
- Surveys

Daily new confirmed COVID-19 cases per million people

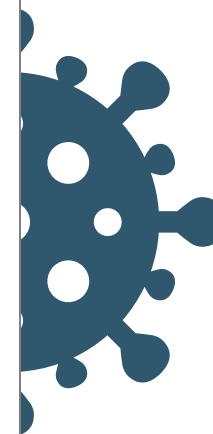
7-day rolling average. Due to limited testing, the number of confirmed cases is lower than the true number of infections.

Our World
in Data



Data source: World Health Organization (2024); Population based on various sources (2024)

CC BY



Importance of time series

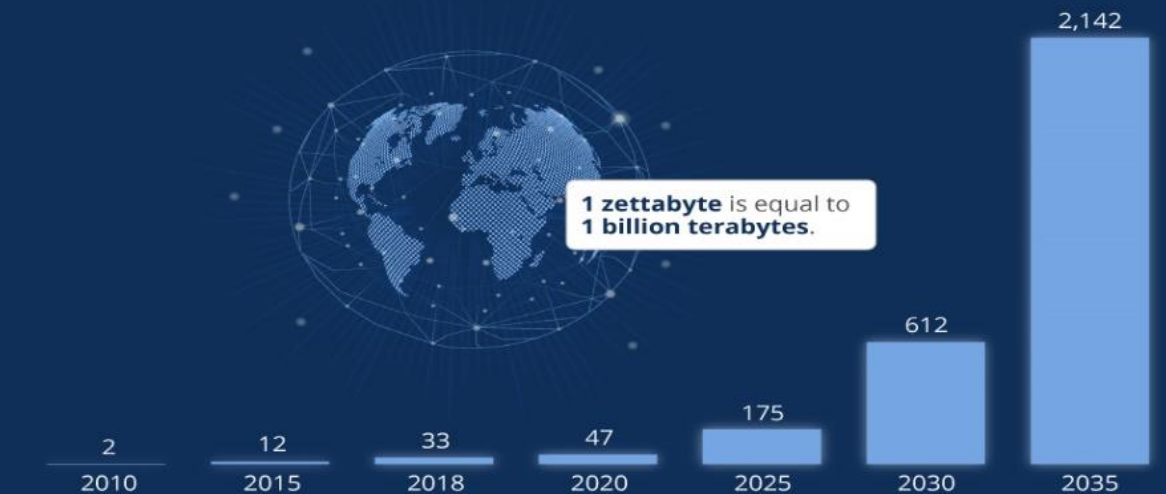
Machine learning on time series is becoming increasingly important because of the massive production of time series data from diverse sources, e.g.,

- Digitalization in healthcare
- Internet of things
- Smart cities
- Process monitoring

The amount of created data increased from two zettabytes in 2010 to 47 zettabytes in 2020

Global Data Creation is About to Explode

Actual and forecast amount of data created worldwide 2010-2035 (in zettabytes)



CC BY ND
@StatistaCharts

Source: Statista Digital Economy Compass 2019

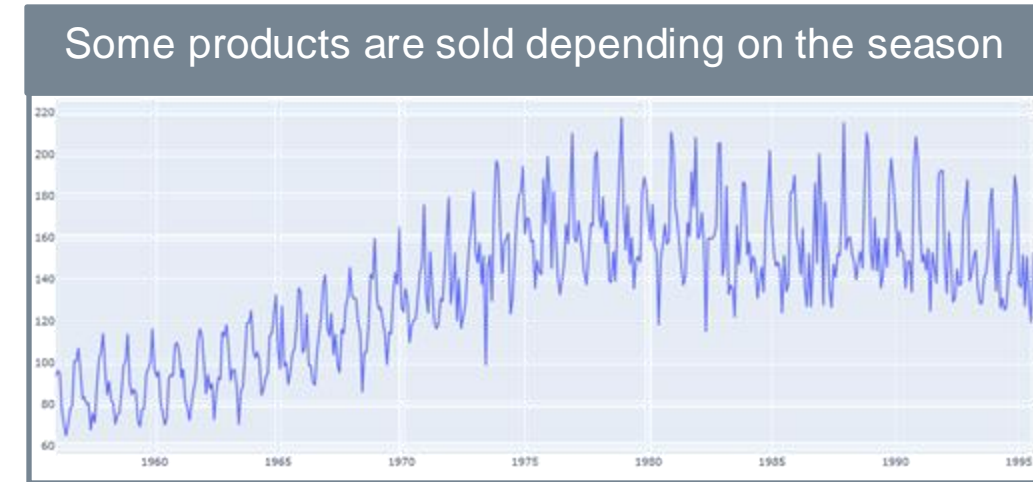
statista

<https://www.statista.com>

Example: Predicting demand of products

Amazon sells 400 million products in over 185 countries^(a).

- Maintaining surplus inventory levels for every product is cost-prohibitive.
- Predict future demand of products

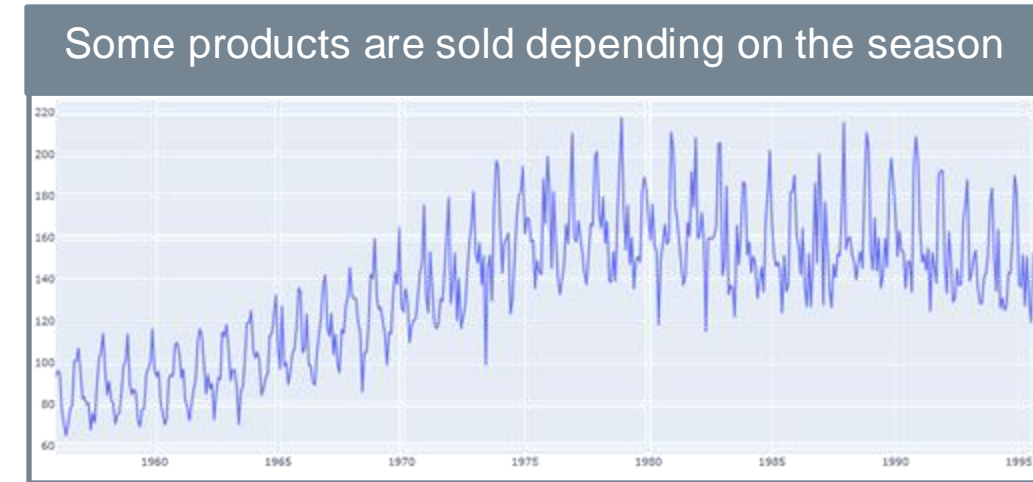


^(a) <https://www.amazon.science/latest-news/the-history-of-amazons-forecasting-algorithm>

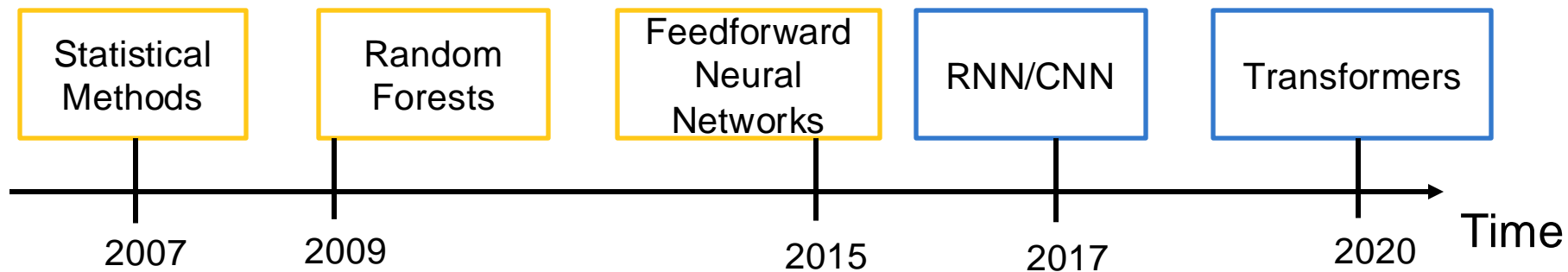
Example: Predicting demand of products

Amazon sells 400 million products in over 185 countries^(a).

- Maintaining surplus inventory levels for every product is cost-prohibitive.
- Predict future demand of products



Methods:



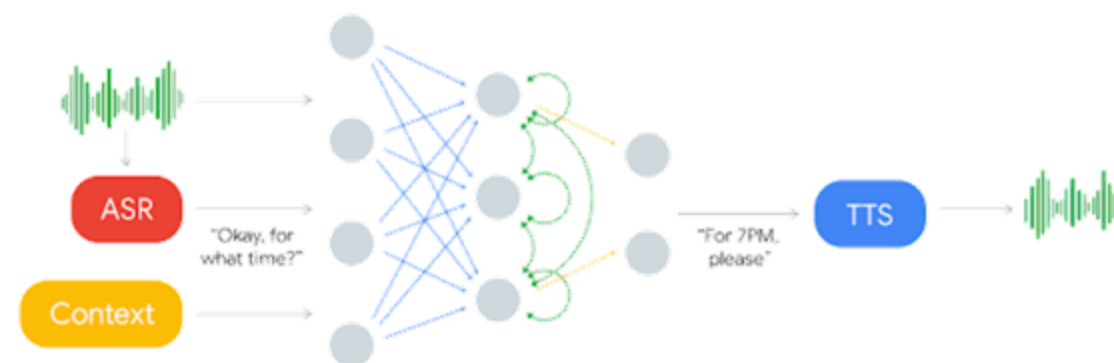
□ First models required manual feature engineering

(a) <https://www.amazon.science/latest-news/the-history-of-amazons-forecasting-algorithm>

Example: Google Duplex makes tedious phone calls

Long standing goal of making humans having a natural conversation with machines, as they would with each other.

- Carry out real-world tasks over the phone



- Additional audio features
- Automatic speech recognition
- Desired service, time/day



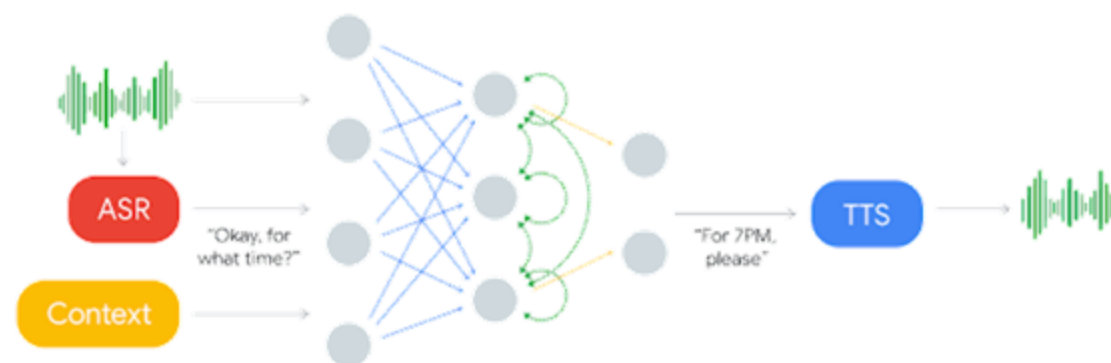
E.g., Duplex calling a restaurant.

Example. **Google**

Duplex makes tedious phone calls

Method: An RNNs with several features. We use a combination of text to speech (TTS) engine and a synthesis TTS engine to control intonation (e.g., “hmm”s and “uh”s).

Limitations: trained on specific tasks.
Cannot deal general conversations.



- Additional audio features
- Automatic speech recognition
- Desired service, time/day

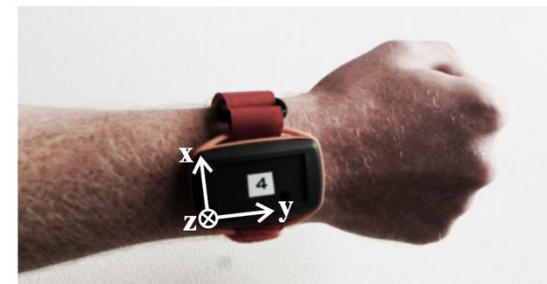


E.g., Duplex calling a restaurant.

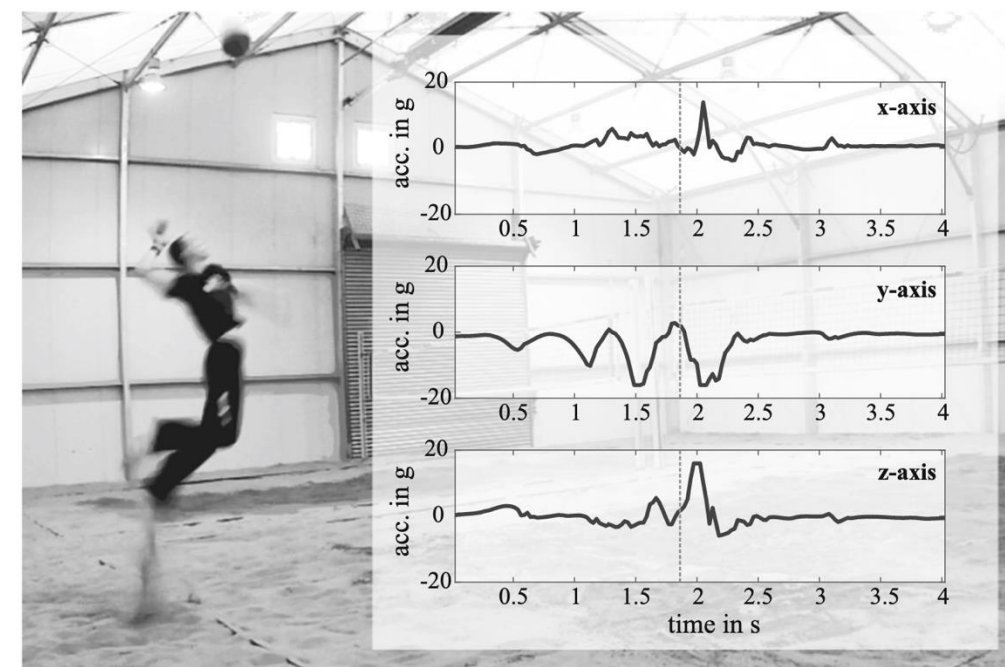
Example: Activity recognition in sports (FAU Erlangen)

Many injuries in sports are caused by overuse.

- These injuries are a major cause for reduced performance of professional and non-professional beach volleyball players.
- Monitoring of player actions could help identifying and understanding risk factors and prevent such injuries.



Sensor attachment at the wrist of the dominant hand with a soft, thin wristband

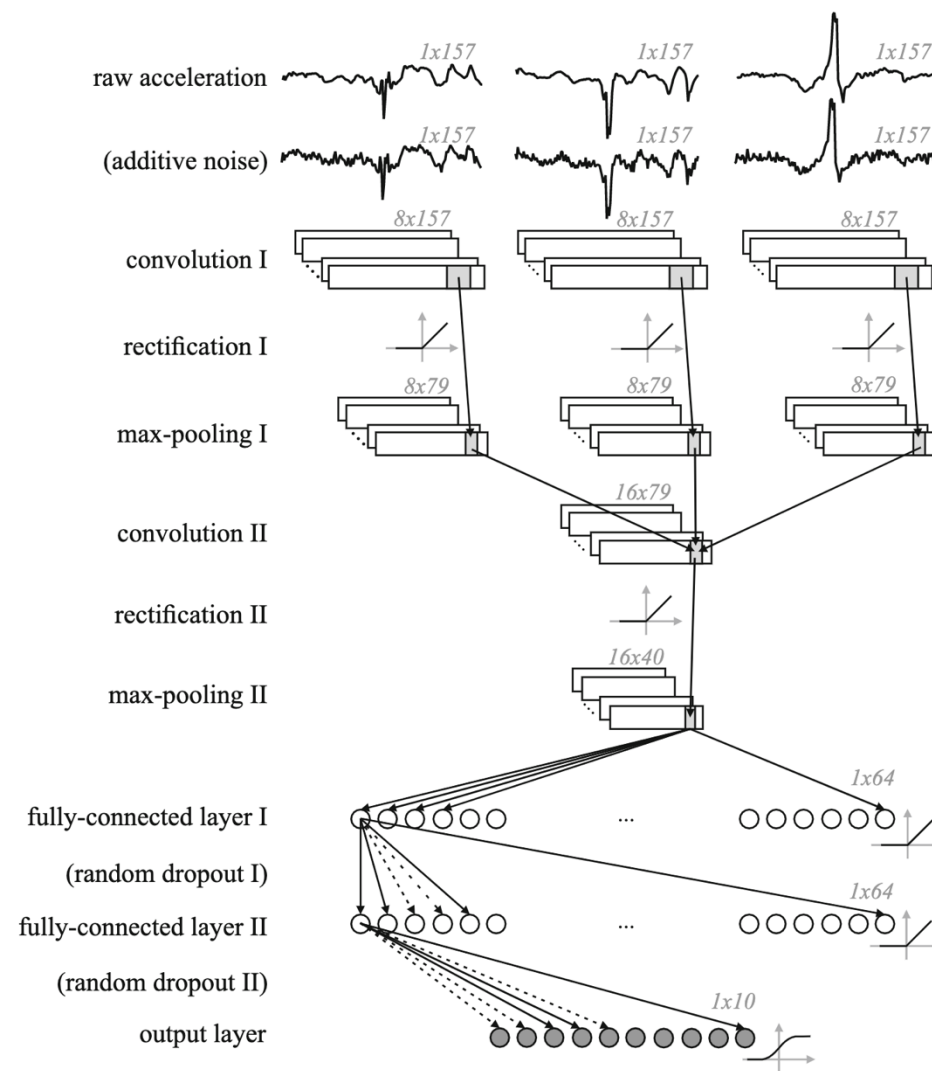


Example: Activity recognition in sports (FAU Erlangen)

Method: A CNN is used to classify players' activities. Classifications allow to create players' profiles.

Actions:

- Underhand serve
- Overhand serve
- Jump serve
- Underarm set
- Overhead set
- Shot attack
- Spike
- Block
- Dig
- Null class.





ADLTS \ Introduction \

Definitions and basic properties



What is a time series?

A time series can be described as a set of observations, taken sequentially in time,

$$S = \{s_1, \dots, s_T\}$$

where $s_i \in \mathbb{R}^d$ is the measured state of the observed process at time t_i .

Typically, observations are generally *dependent*

- Studying the nature of this dependency is of particular interest
 - Time series analysis is concerned with techniques for the analysis of these dependencies
-

Examples of time series

❖ Monthly Goods Shipped from a Factory



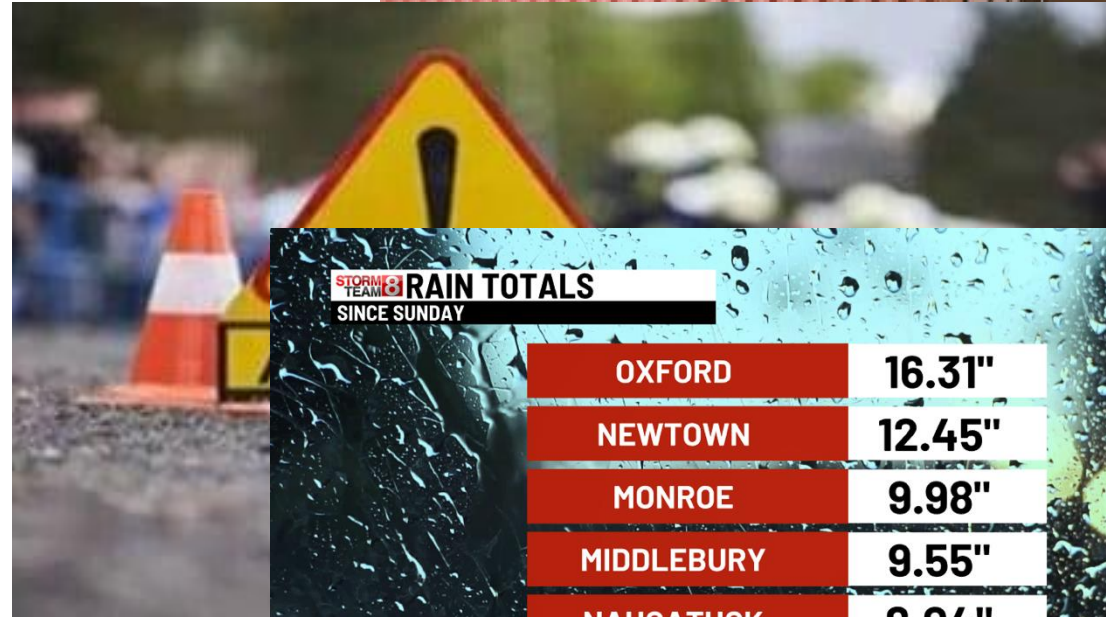
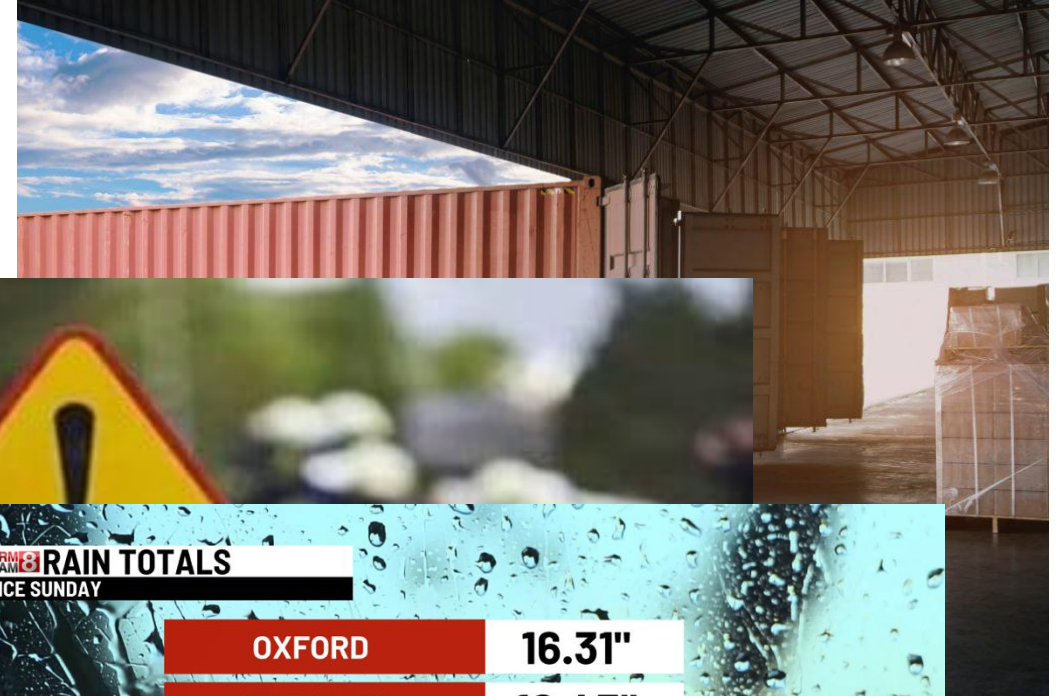
Examples of time series

- ❖ **Monthly Goods Shipped from a Factory**
- ❖ **Weekly Road Accidents**



Examples of time series

- ❖ Monthly Goods Shipped from a Factory
- ❖ Weekly Road Accidents
- ❖ Daily Rainfall Amounts
- ❖ ...

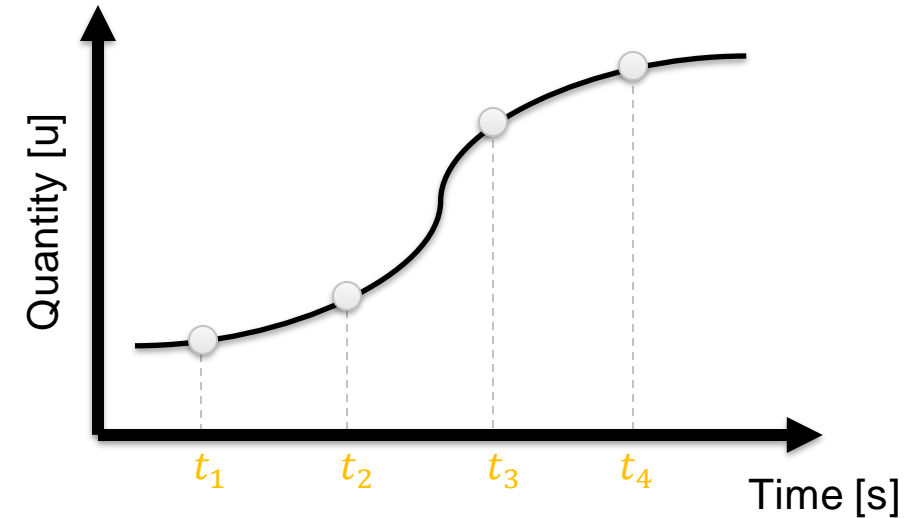


STORM TEAM 3 RAIN TOTALS SINCE SUNDAY	
OXFORD	16.31"
NEWTOWN	12.45"
MONROE	9.98"
MIDDLEBURY	9.55"
NAUGATUCK	8.04"
SEYMOUR	6.80"

Terminology: Regularly Sampled vs Irregularly Sampled

Discrete time series are **regularly sampled** if their observations are equally spaced in time.

$$\forall i \in \{1, \dots, T - 1\},$$
$$\Delta_{t_i} = t_{i+1} - t_i = \text{const.}$$



Terminology: Regularly Sampled vs Irregularly Sampled

Discrete time series are **regularly sampled** if their observations are equally spaced in time.

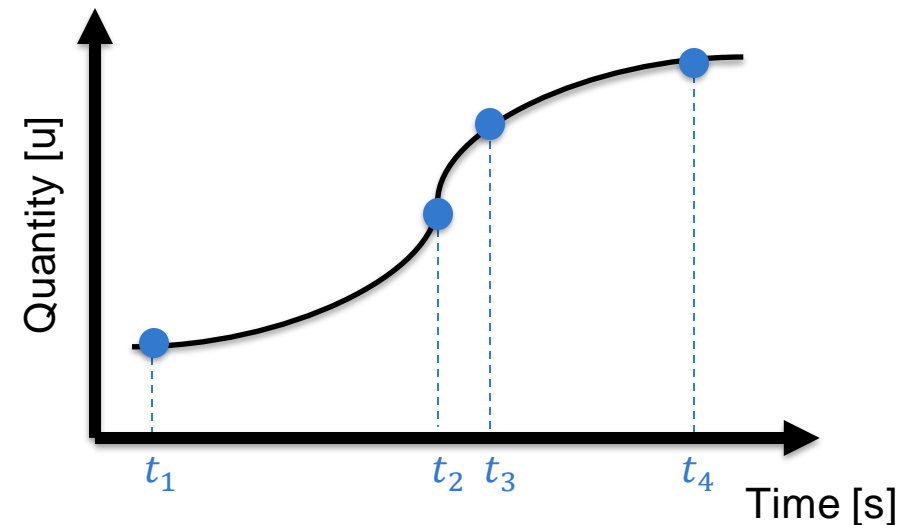
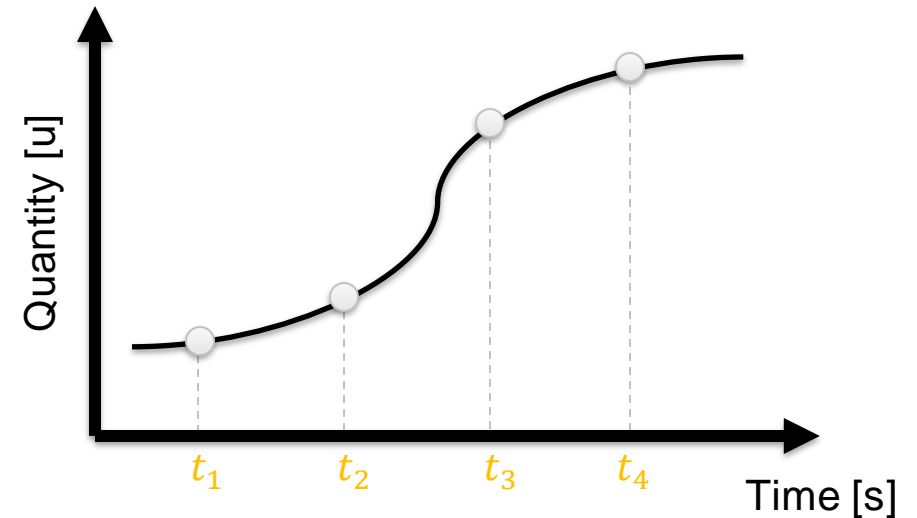
$$\forall i \in \{1, \dots, T-1\},$$

$$\Delta t_i = t_{i+1} - t_i = \text{const.}$$

In contrast, for **irregularly sampled** time sequences, the observations are not equally spaced.

- They are generally defined as a collection of pairs

$$S = \{(s_1, t_1), \dots, (s_T, t_T)\}$$

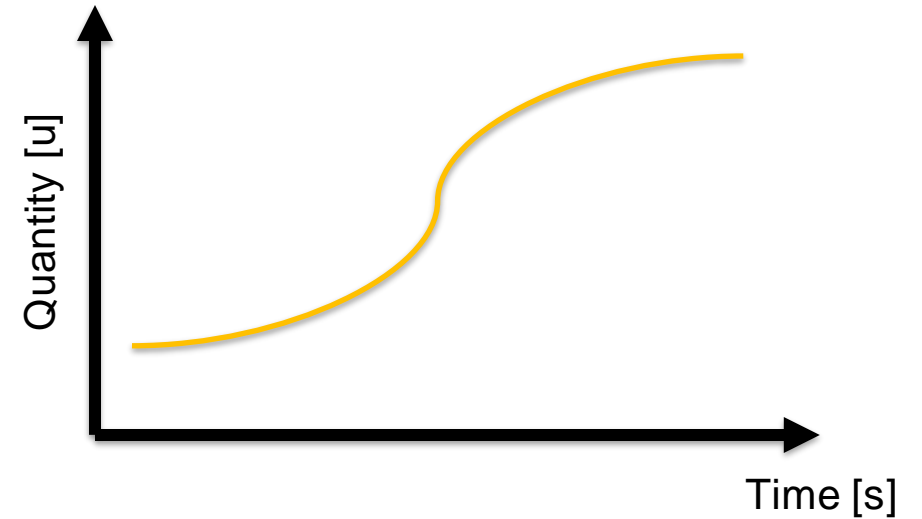


Terminology: Univariate vs Multivariate

Let $S = (s_1, \dots, s_T)$ be a time series,
where $s_i \in \mathbb{R}^d, \forall i \in \{1, \dots, T\}$.

If $d = 1$, S is said **univariate**.

- Only one variable is varying over time.



Terminology: Univariate vs Multivariate

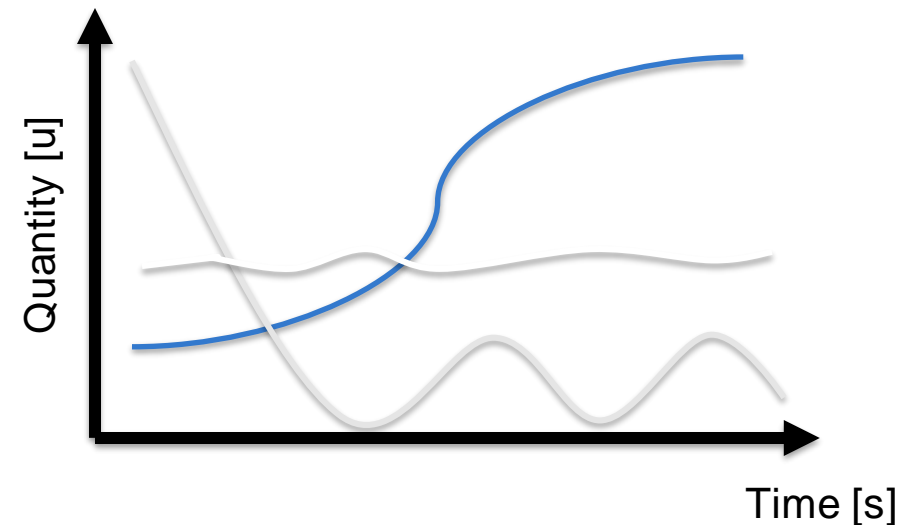
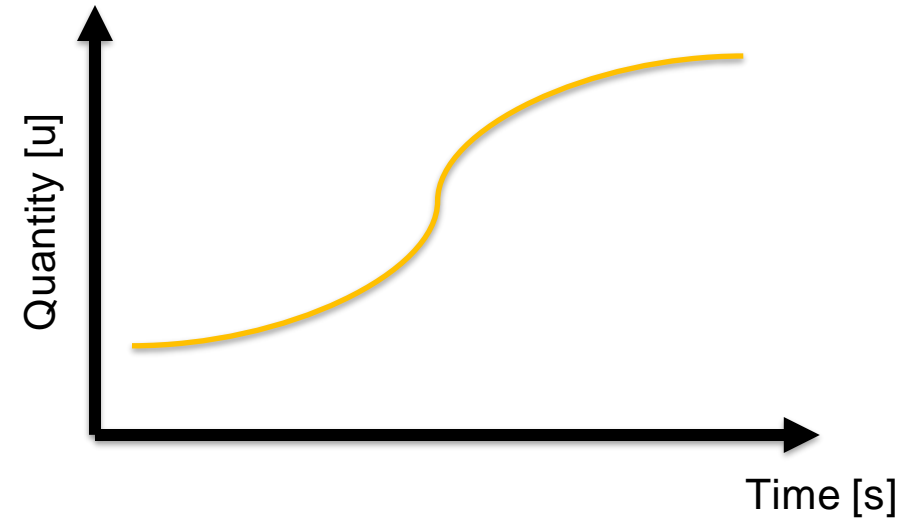
Let $S = (s_1, \dots, s_T)$ be a time series,
where $s_i \in \mathbb{R}^d, \forall i \in \{1, \dots, T\}$.

If $d = 1$, S is said **univariate**.

- Only one variable is varying over time.

If $d > 1$, S is said **multivariate**.

- Multiple variables are varying over time
 - E.g., tri-axial accelerometer measurements



Terminology: Discrete vs Continuous

A time series is **continuous** when observations are made continuously through time. The term continuous is used for series of this type even when the measured variables can take discrete set of values.

- E.g., the number of people in a room.

A time series is **discrete** when observations are taken only at specific times. The term discrete is used for series of this type even when the measured variables is a continuous variable.

- E.g, event logs.
-

Terminology: Discrete vs Continuous

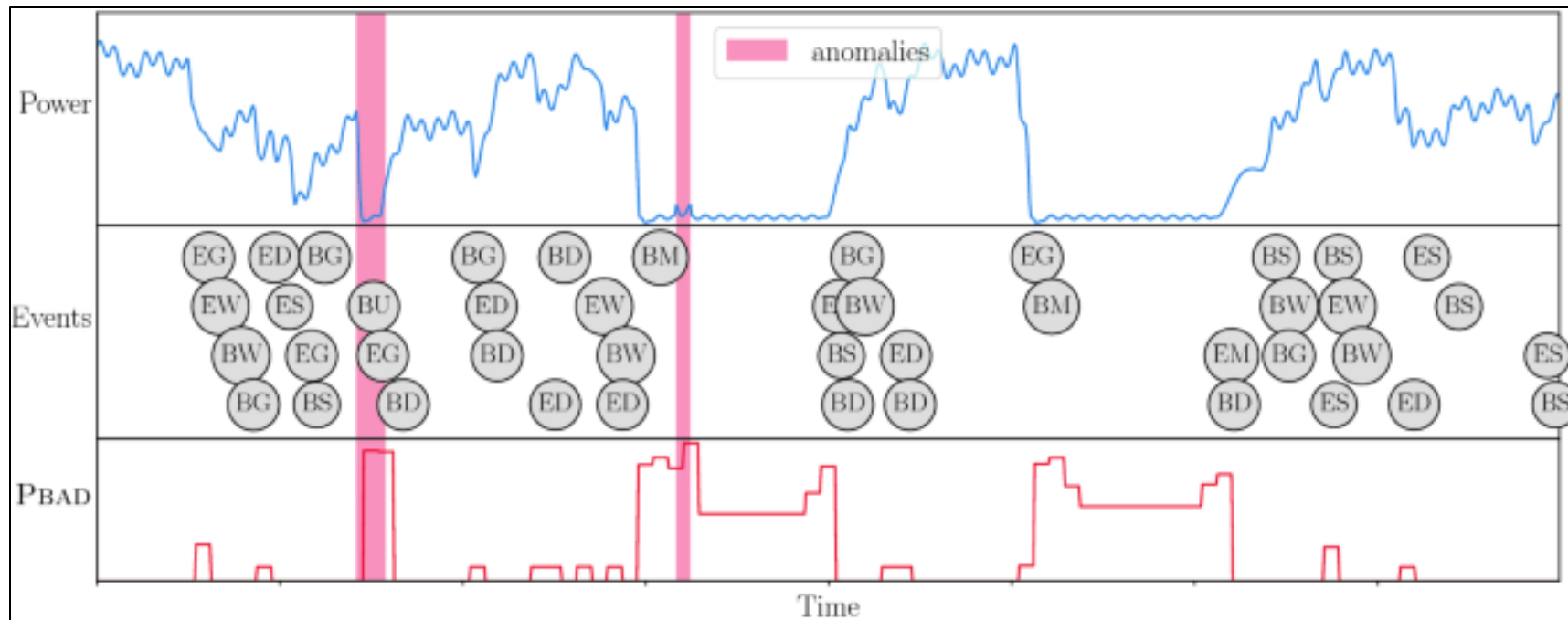
We will denote as **mixed-type** a multivariate time series consisting of both continuous and discrete observations

- E.g., a time series consisting of continuous sensor values and discrete event log for the monitoring of an industrial machine

Terminology: Discrete vs Continuous

We will denote as **mixed-type** a multivariate time series consisting of both continuous and discrete observations

- E.g., a time series consisting of continuous sensor values and discrete event log for the monitoring of an industrial machine



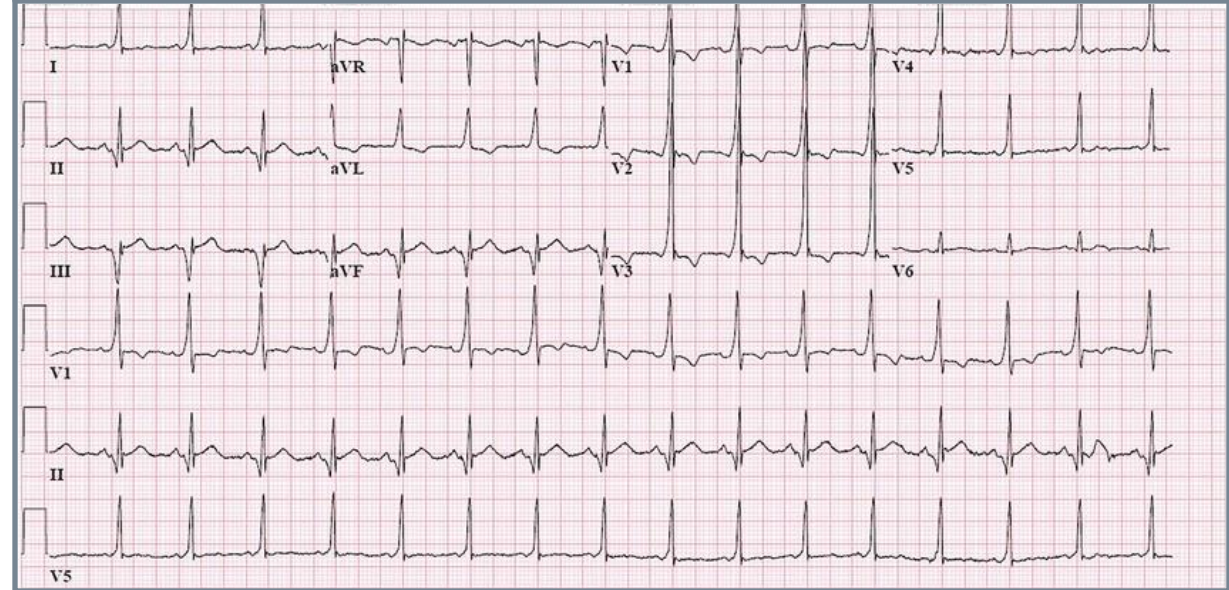
Terminology: Periodic

A time series is said **periodic** if there exists a number $\tau \in \mathbb{R}$, called *period*, such that

$$s_i = s_{i+\tau}, \forall i \in \{1, \dots, T - \tau\}$$

E.g., the continuous time series defined by the trigonometric function $f(x) = \sin(x)$

Is the biological signal of an heartbeat a periodic function?



Terminology: Deterministic vs Non-Deterministic

A **deterministic** time series is one that can be fully described by a known analytical expression or a set of rules. Observations are generated from a system that behaves predictably, with no element of randomness.

In contrast, a **non-deterministic** time series cannot be fully described by an analytical expression. A time series may be non-deterministic for the following reasons:

- The information necessary to describe the process is not fully observable, or
 - The process generating the time series involves inherent randomness.
-

Stochastic Process

Non-deterministic time series can be regarded as manifestations (equiv., realization) of a **stochastic process**, which is defined as a set of random variables $\{X_t\}_{t \in \{1, \dots, T\}}$

Even if we were to imagine having observed the process for an infinite period T of time, the infinite sequence

$$S = \{\dots, s_{t-1}, s_t, s_{t+1}, \dots\} = \{s_t\}_{t=-\infty}^{+\infty}$$

would still be a single **realization** from that process.

Stochastic Process

Still, if we had a battery of N computers generating series $S^{(1)}, \dots, S^{(N)}$, and considering selecting the observation at time t from each series,

$$\{s_t^{(1)}, \dots, s_t^{(N)}\}$$

this would be described as a sample of N realizations of the random variable X_t

Stochastic Process

Still, if we had a battery of N computers generating series $S^{(1)}, \dots, S^{(N)}$, and considering selecting the observation at time t from each series,

$$\{s_t^{(1)}, \dots, s_t^{(N)}\}$$

this would be described as a sample of N realizations of the random variable X_t

This random variable X_t is associated with an **unconditional density**, denoted by

$$f_{X_t}(s_t)$$

- E.g., for the Gaussian white noise process $f_{X_t}(s_t) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-s_t^2}{2\sigma^2}}$

Stochastic Process

The **unconditional mean** is the expectation, provided it exists, of the t -th observation, i.e.,

$$E(X_t) = \int_{-\infty}^{+\infty} s_t f_{X_t}(s_t) ds_t = \mu_t$$

Similarly, the **variance** of the random variable X_t is defined as

$$E(X_t - \mu_t)^2 = \int_{-\infty}^{+\infty} (s_t - \mu_t)^2 f_{X_t}(s_t) ds_t$$

Stochastic Process

Given any particular realization $S^{(i)}$ of a stochastic process (i.e., a time series), we can define the vector of the $j + 1$ most recent observations

$$x_t^i = [s_{t-j}^{(i)}, \dots, s_t^{(i)}]$$

We want to know the probability distribution of this vector x_t^i across realizations. We can calculate the **j -th autocovariance**

$$\gamma_{jt} = E(X_t - \mu_t)(X_{t-j} - \mu_{t-j})$$

Stationarity

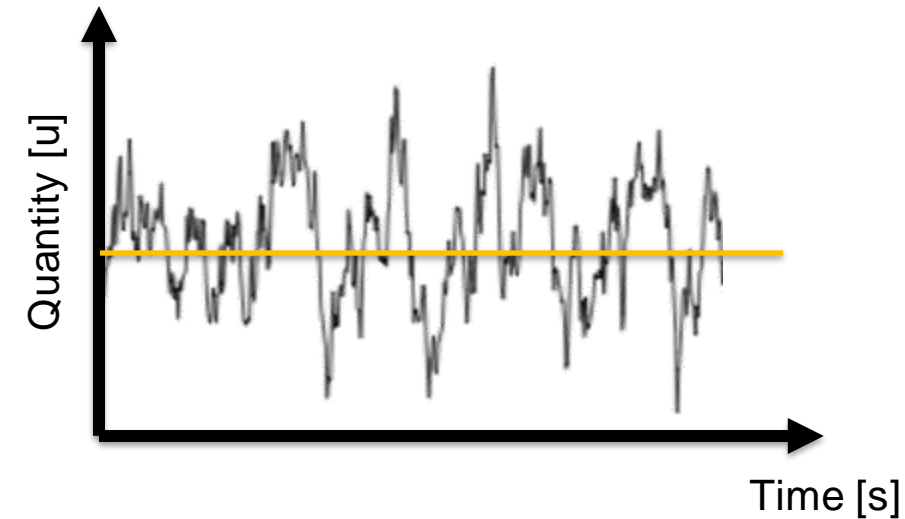
If neither the mean μ_t or the autocovariance γ_{jt} depend on the temporal variable t , then the process is said to be (weakly) **stationary**.

E.g., let the stochastic process $\{X_t\}_{t=-\infty}^{+\infty}$ represent the sum of a constant μ with a Gaussian white noise process $\{\epsilon_t\}_{t=-\infty}^{+\infty}$, such that

$$X_t = \mu + \epsilon_t$$

Then, its mean is constant: $E(X_t) = \mu + E(\epsilon_t) = \mu$

and its j -th autocovariance: $E(X_t - \mu)(X_{t-j} - \mu) = \gamma_j$



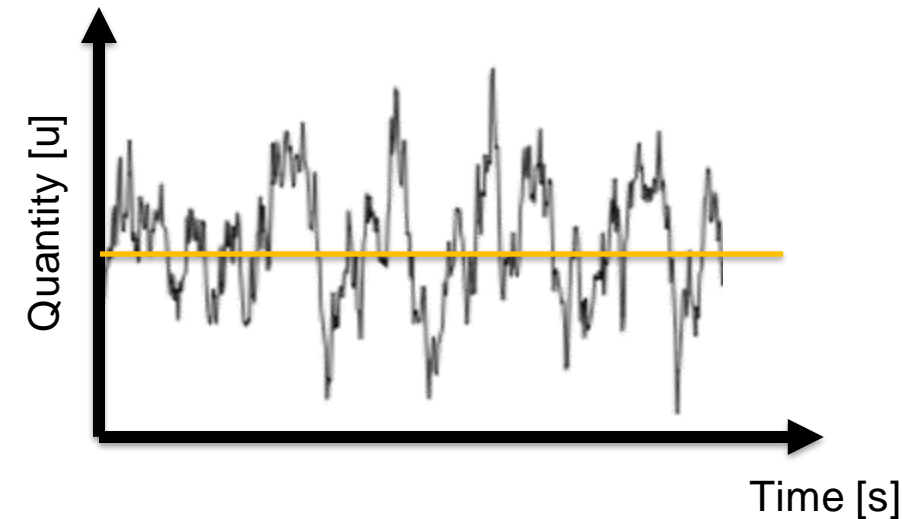
Stationarity

If neither the mean μ_t or the autocovariance γ_{jt} depend on the temporal variable t , then the process is said to be (weakly) **stationary**.

E.g., let the stochastic process $\{X_t\}_{t=-\infty}^{+\infty}$ represent the sum of a constant μ with a Gaussian white noise process $\{\epsilon_t\}_{t=-\infty}^{+\infty}$, such that

$$X_t = \mu + \epsilon_t$$

Then, its mean is constant: $E(X_t) = \mu + E(\epsilon_t) = \mu$
and its j -th autocovariance: $E(X_t - \mu)(X_{t-j} - \mu) = \gamma_j$



In other words: A process is said to be stationary if the process statistics do not depend on time.

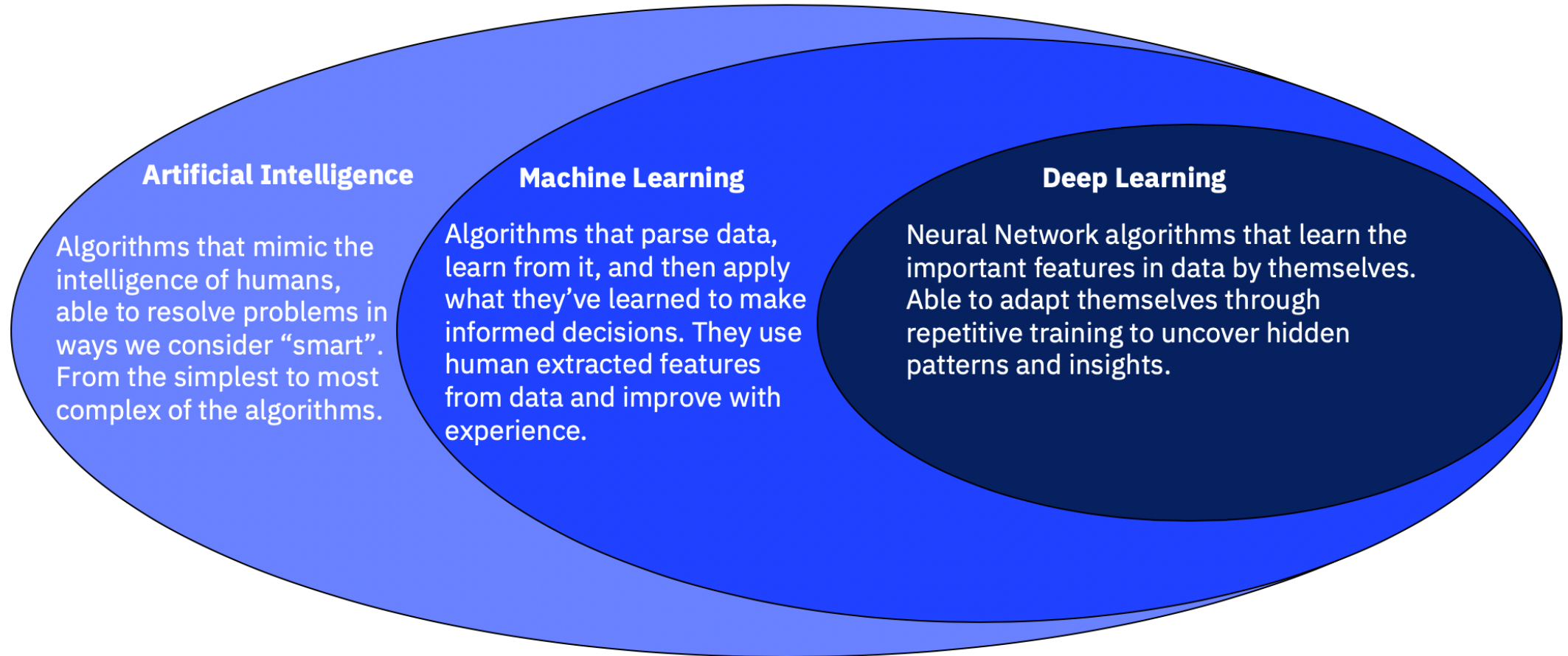


ADLTS \ Introduction \

Types of machine learning



What is Machine Learning (ML)?



Supervised, unsupervised and reinforcement learning

Supervised Learning

- Learning using a teacher
- Makes machine learning explicitly
- Labeled data

Unsupervised Learning

- Learning using an “abstract” metric
- Machine understands the data (Identifies patterns/structures)
- Evaluation is qualitative or indirect

Reinforcement Learning

- Reward based learning
 - Learning from positive and negative reinforcement
 - Machine learns how to act in a certain environment to maximize rewards
-

Supervised, unsupervised and reinforcement learning

Supervised Learning

- Learning using a teacher
- Makes machine learning explicitly
- Labeled data

Unsupervised Learning

- Learning using an “abstract” metric
- Machine understands the data (Identifies patterns/structures)
- Evaluation is qualitative or indirect

Reinforcement Learning

- Reward based learning
- Learning from positive and negative reinforcement
- Machine learns how to act in a certain environment to maximize rewards

Supervised Learning

The agent observes some example **Input (Features)** and **Output (Label) pairs** and learns a **function that maps input to output**.

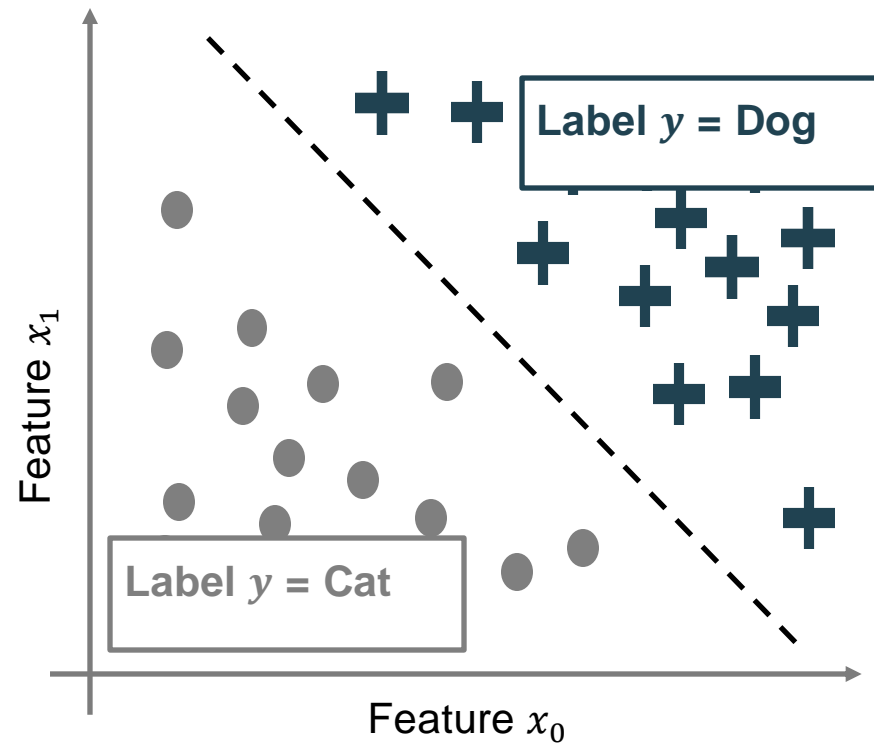
Key Aspects:

- Learning is **explicit**
- Learning using **direct feedback**
- Data with **labeled output**

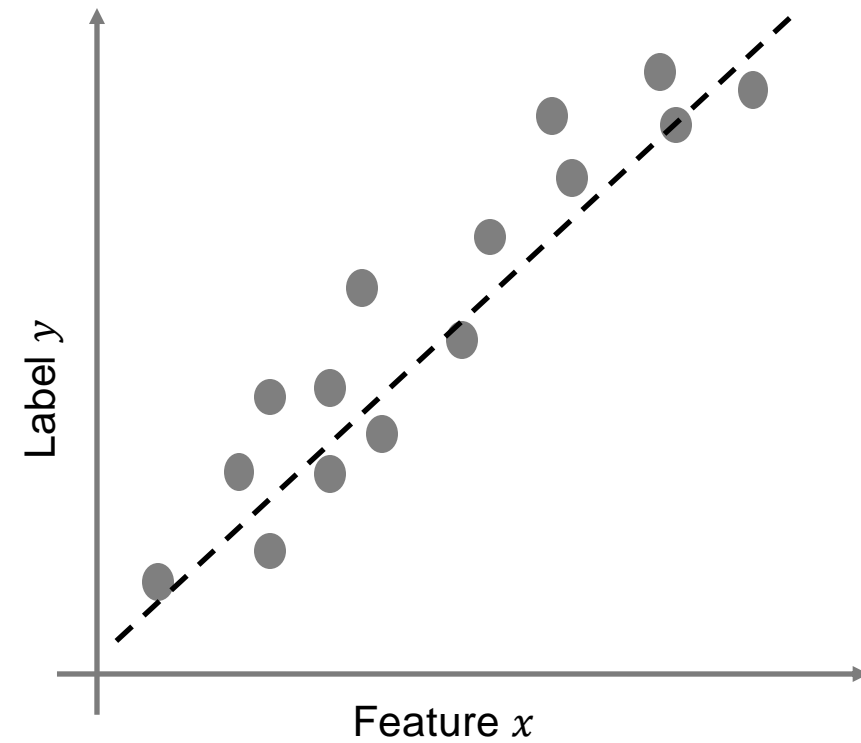
→ Resolves classification and regression problems

Supervised Learning Problems

Classification



Regression



Regression

Regression is used to predict
a **continuous value**

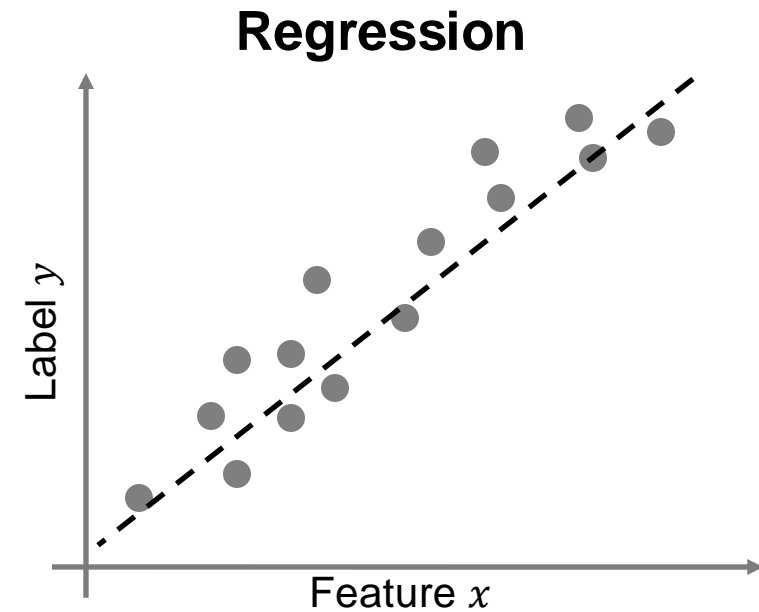
Training is based on a set of
input – output pairs (**samples**)

$$\mathcal{D} = \{(\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$$

Sample : $(\mathbf{x}_i, \mathbf{y}_i)$

$\mathbf{x}_i \in \mathbb{R}^m$ is the **feature vector** of sample i

$\mathbf{y}_i \in \mathbb{R}$ is the **label value** of sample i



Regression

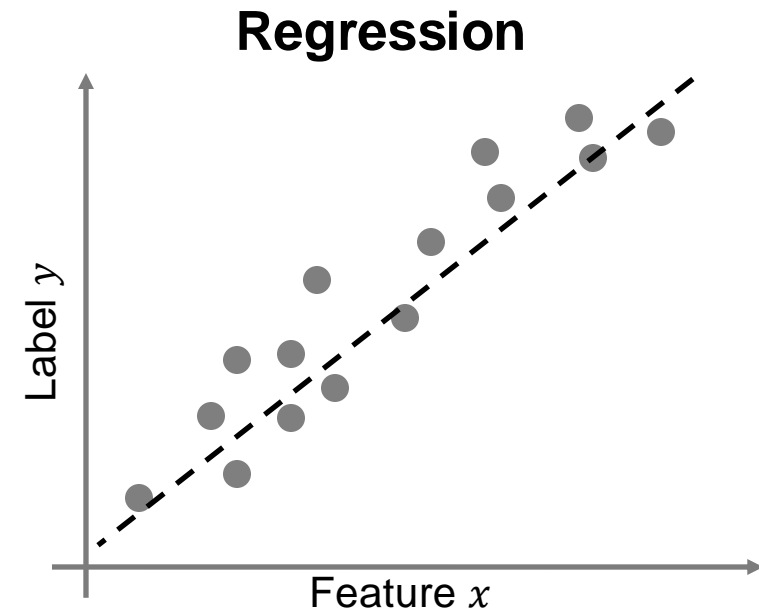
Goal:

Find a relationship (function), which expresses the input and output the best!

That means, we **fit a regression model** f to all samples:

$$f(x_i) = y_i, \forall (x_i, y_i) \in \mathcal{D}$$

In this case f is a **linear regression model!** (Black Line)



Classification

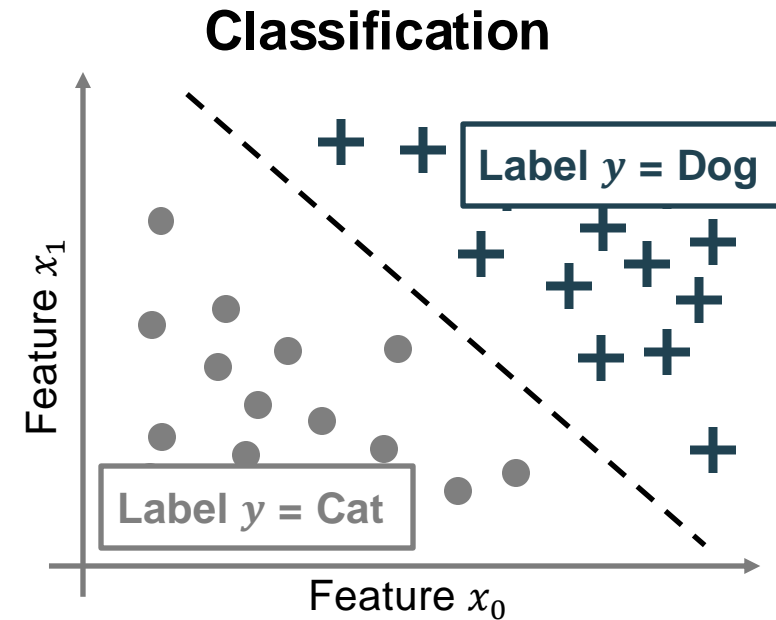
Classification is used to predict the **class** of the input

Sample : $s_i = (\vec{x}_i, \vec{y}_i)$

$\vec{y}_i \in L$ is the **label** of sample i

In this example:

- $L = \{„Cat“ := -1, „Dog“ := 1\}$
- Binary classification problem
- The output belongs to only one class!



Classification

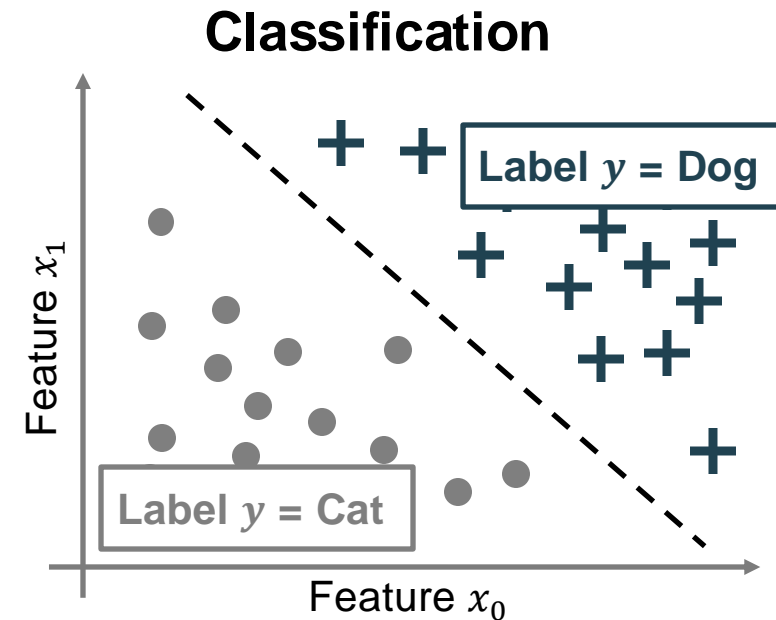
Goal:

Find a way to divide the input into the output classes!

That means, we **find a decision function** f for all samples:

$$f(\vec{x}_i) = \vec{y}_i, \forall (\vec{x}_i, \vec{y}_i) \in \mathcal{D}$$

In this case $f(x) = 0$ is a **decision boundary!** (Black Line)



Supervised Learning Problems

Regression	Classification
<ul style="list-style-type: none">• The output are continuous or real values	<ul style="list-style-type: none">• The output variable must be a discrete value (class)
<ul style="list-style-type: none">• We try to fit a regression model, which can predict the output more accurately	<ul style="list-style-type: none">• We try to find a decision boundary, which can divide the dataset into different classes.
<ul style="list-style-type: none">• Regression algorithms can be used to solve the regression problems such as Weather Prediction, House price prediction, Stock market prediction etc.	<ul style="list-style-type: none">• Classification Algorithms can be used to solve classification problems such as Hand-written digits(MNIST), Identification of cancer cells, Defected or Undefected solar cells etc.

Supervised, unsupervised and reinforcement learning

Supervised Learning

- Learning using a teacher
- Makes machine learning explicitly
- Labeled data

Unsupervised Learning

- Learning using an “abstract” metric
- Machine understands the data (Identifies patterns/structures)
- Evaluation is qualitative or indirect

Reinforcement Learning

- Reward based learning
- Learning from positive and negative reinforcement
- Machine learns how to act in a certain environment to maximize rewards

Unsupervised Learning

Unsupervised learning observes some example Input (Features) – No Labels! - and finds patterns based on a metric

Key Aspects:

- Learning is **implicit**
- Learning using **indirect feedback**
- Methods are **self-organizing**

Resolves **clustering** and **dimensionality reduction** problems

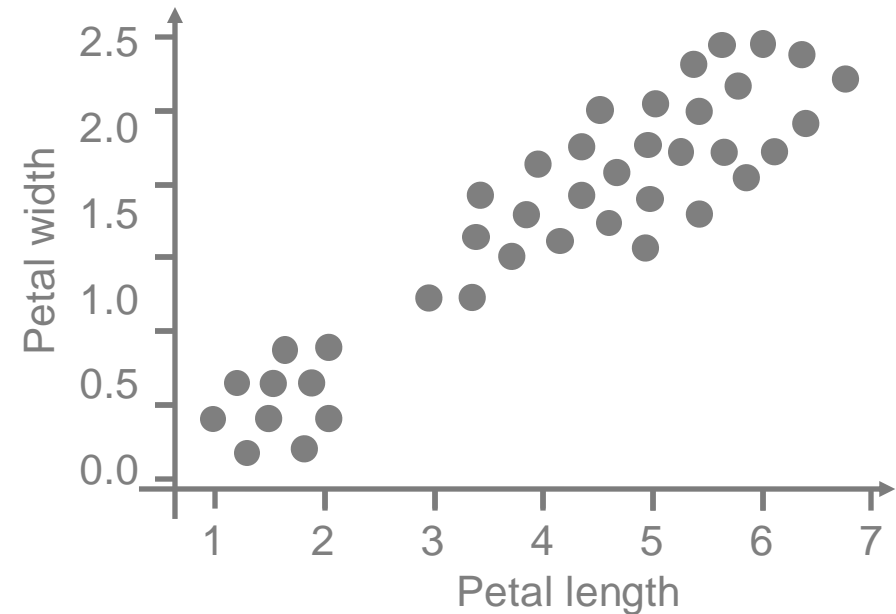
Clustering

Goal: Identify similar samples and assign them the same label

Mostly used for data analysis,
data exploration, and/or
data preprocessing

Clustering basic principles:

- Homogeneous data in the cluster
(Intra-cluster distance)
- Heterogeneous data between the cluster
(Inter-cluster distance)



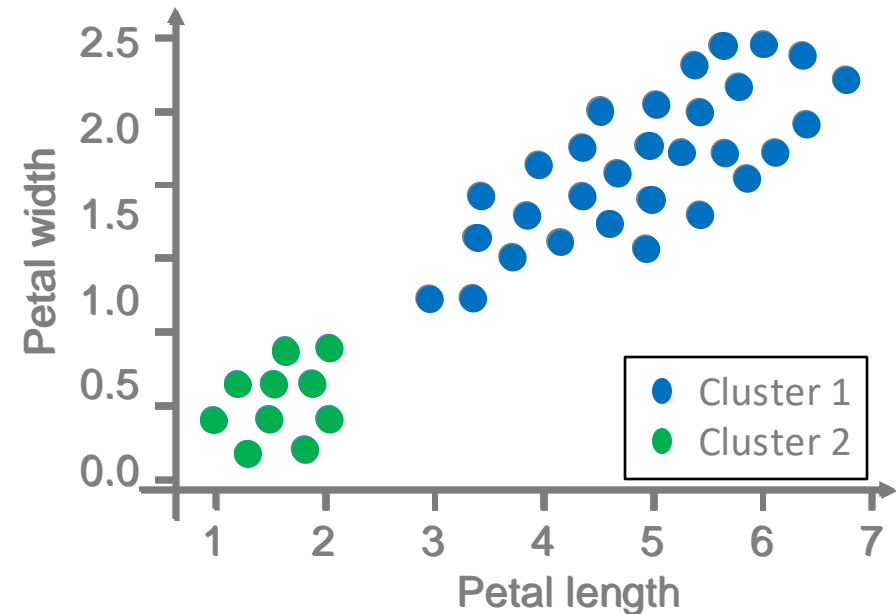
Clustering

Goal: Identify similar samples and assign them the same label

Mostly used for data analysis,
data exploration, and/or
data preprocessing

Clustering basic principles:

- Homogeneous data in the cluster
(Intra-cluster distance)
- Heterogeneous data between the cluster
(Inter-cluster distance)



Curse of dimensionality

As the number of features or dimensions grows, the amount of data we need to generalize accurately grows exponentially.” – Charles Isbell

The intuition in lower dimensions does not hold in higher dimensions:

- Almost all samples are close to at least one boundary
- Distances (e.g., Euclidean) between all samples are similar
- Features might be wrongly correlated with outputs
- Finding decision boundaries becomes more complex

→ Problems become much more difficult to solve!

Example: Curse of dimensionality

The production system has N sensors attached with either the input set to “On” or “Off”

Question: How many samples do we need, to have **all possible sensor states** in the dataset?

$$N = 1 \quad : |D| = 2^1 = 2$$

$$N = 10 \quad : |D| = 2^{10} = 1024$$

$$N = 100 \quad : |D| = 2^{100} = 1.2 \times 10^{30}$$

For $N = 100$, the number of points are even more than the number of atoms in the universe!

Dimensionality reduction

The goal:

Transform the samples from a high to a lower dimensional representation!


Ideally:

Find a representation, which solves your problem!

Typical Approaches:

- Feature Selection
- Feature Extraction

	S0	S1	S2	S3	S4	S5	S6	S7	S8
Sample0	0.2	0.1	11.1	2.2	Off	7	1.1	0	1.e-1
Sample1	1.2	-0.1	3.1	-0.1	On	9	2.3	-1	1.e-4
Sample2	2.7	1.1	0.1	0.1	Off	10	4.5	-1	1.e-9
Sample3	3.1	0.1	1.1	0.2	Off	1	6.6	-1	1.e-1



	T0	T1	T2	T3
Sample0	11.3	0.1	-1	7.8
Sample1	4.3	-0.1	1	6.8
Sample2	2.8	1.1	-1	7.1
Sample3	4.2	0.1	1	6.9

Dimensionality reduction

The goal:

Transform the samples from a high to a lower dimensional representation!

Ideally:

Find a representation, which solves your problem!

Typical Approaches:

- Feature Selection
- Feature Extraction

	S0	S1	S2	S3	S4	S5	S6	S7	S8
Sample0	0.2	0.1	11.1	2.2	Off	7	1.1	0	1.e-1
Sample1	1.2	-0.1	3.1	-0.1	On	9	2.3	-1	1.e-4
Sample2	2.7	1.1	0.1	0.1	Off	10	4.5	-1	1.e-9
Sample3	3.1	0.1	1.1	0.2	Off	1	6.6	-1	1.e-1

Identical

	T0	T1	T2	T3
Sample0	11.3	0.1	-1	7.8
Sample1	4.3	-0.1	1	6.8
Sample2	2.8	1.1	-1	7.1
Sample3	4.2	0.1	1	6.9

Dimensionality reduction

The goal:

Transform the samples from a high to a lower dimensional representation!

Ideally:

Find a representation, which solves your problem!

Typical Approaches:

- Feature Selection
- Feature Extraction

	S0	S1	S2	S3	S4	S5	S6	S7	S8
Sample0	0.2	0.1	11.1	2.2	Off	7	1.1	0	1.e-1
Sample1	1.2	-0.1	3.1	-0.1	On	9	2.3	-1	1.e-4
Sample2	2.7	1.1	0.1	0.1	Off	10	4.5	-1	1.e-9
Sample3	3.1	0.1	1.1	0.2	Off	1	6.6	-1	1.e-1

Applied a function f

	T0	T1	T2	T3
Sample0	11.3	0.1	-1	7.8
Sample1	4.3	-0.1	1	6.8
Sample2	2.8	1.1	-1	7.1
Sample3	4.2	0.1	1	6.9

Supervised, unsupervised and reinforcement learning

Supervised Learning

- Learning using a teacher
- Makes machine learning explicitly
- Labeled data

Unsupervised Learning

- Learning using an “abstract” metric
- Machine understands the data (Identifies patterns/structures)
- Evaluation is qualitative or indirect

Reinforcement Learning

- Reward based learning
- Learning from positive and negative reinforcement
- Machine learns how to act in a certain environment to maximize rewards

Reinforcement Learning

Reinforcement learning observes some example Input (Features) – No Labels! - and finds the **optimal action** i.e., maximizes its future reward

Key Aspects:

- Learning is **implicit**
- Learning using **indirect feedback** based on trials and reward signals
- Actions are **affecting future measurements (i.e., inputs)**

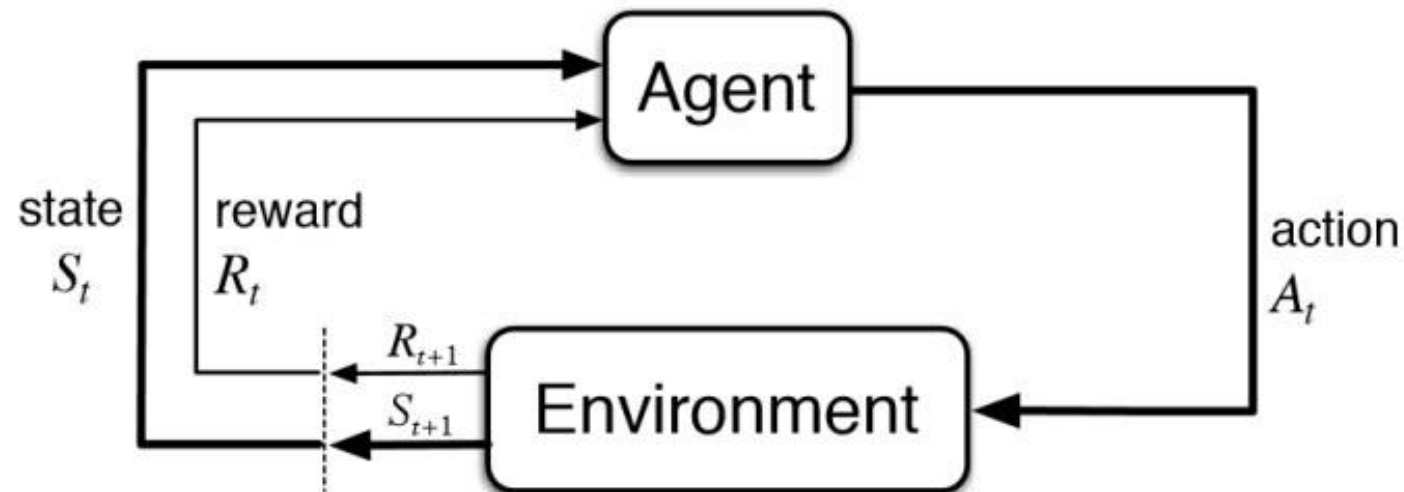
Resolves **control** and **decision** problems

- i.e., controlling agents in games or robots
-

Reinforcement Learning

Goal: Agents should take actions in an environment which maximize the cumulative reward.

To achieve this RL uses **reward** and **punishment** signals based on the previous actions to optimize the model.



Reinforcement Learning vs Unsupervised Learning

Unsupervised Learning	Reinforcement Learning
<ul style="list-style-type: none">• An indirect feedback is generated according to a metric	<ul style="list-style-type: none">• The feedback is given by a reward signal
<ul style="list-style-type: none">• Feedback is instantaneous	<ul style="list-style-type: none">• Feedback can be delayed (credit assignment problem)
<ul style="list-style-type: none">• Learning by using static data (no re-recording of data necessary)	<ul style="list-style-type: none">• Training is based on trials i.e. interaction between environment and agent (re-recording necessary)
<ul style="list-style-type: none">• Prediction does not affect future measurements – The data is assumed Independent Identically Distributed (i.i.d)	<ul style="list-style-type: none">• The prediction (actions) affect future measurements i.e. the measurements are no necessarily i.i.d!



ADLTS \ Introduction \ ML pipeline and good practices



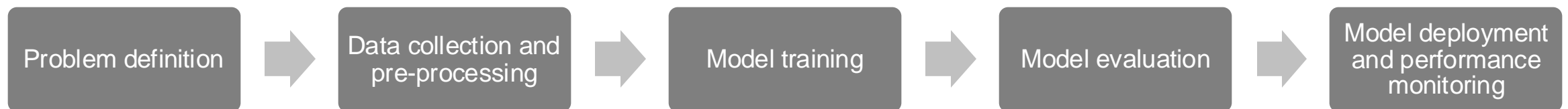
The ML pipeline

The concept of a pipeline guidance in a machine learning project.

- step-by-step process
- Each step has a **goal** and a **method**

There exist many pipelines proposals in the literature.

Here we propose a compact 5-steps pipeline:



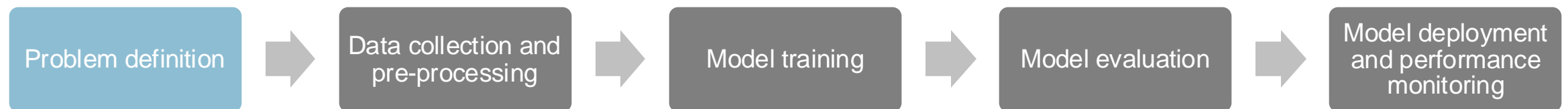
Step 1. Problem definition

In order to develop a satisfying solution, we need to define the problem.

- What goal (or **task**) we want to solve
- What kind of **data** we need

E.g., our goal is to monitor an industrial machine to predict failure and allow convenient scheduling of corrective maintenance.

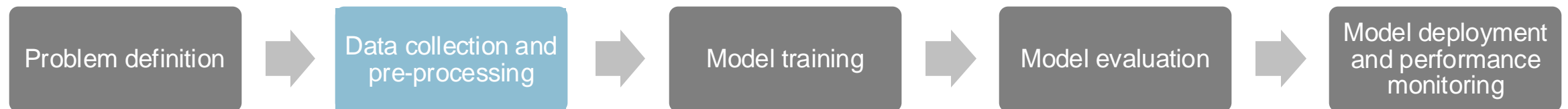
- Our goal is to predict failure one week in advance
 - Alternatively, predict the remaining useful life
- Prediction is based on data from the machine (sensors) and users (logs)



Step 2. Data collection and pre-processing

The phase of gathering the data and creating our dataset is called **data ingestion**.

- Data should **contain necessary information** to solve the task
- Data should **be enough** to describe all possible states



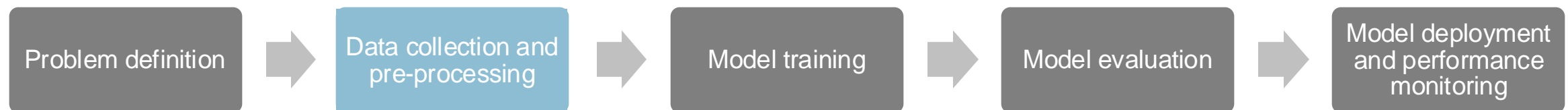
Step 2. Data collection and pre-processing

The phase of gathering the data and creating our dataset is called **data ingestion**.

- Data should **contain necessary information** to solve the task
- Data should **be enough** to describe all possible states

Then, a **data preparation** phase follows, with the goal of making data usable for our machine learning solution.

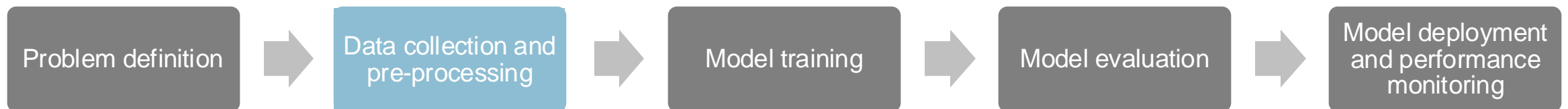
- Remove missing values and outliers, apply dimensionality reduction, normalize, rescale, ...



Step 2. Data collection and pre-processing

Finally, and before training any machine learning algorithm, we perform **data segregation**.

- We **separate the target** value from the input features
- We split the data collection into
 - Training set: used to fit our model parameters
 - Validation set: used to have an unbiased estimation of the model generalization capabilities during training, and tune hyperparameters of our model
 - Test set: used to evaluate performance of a final version of the model

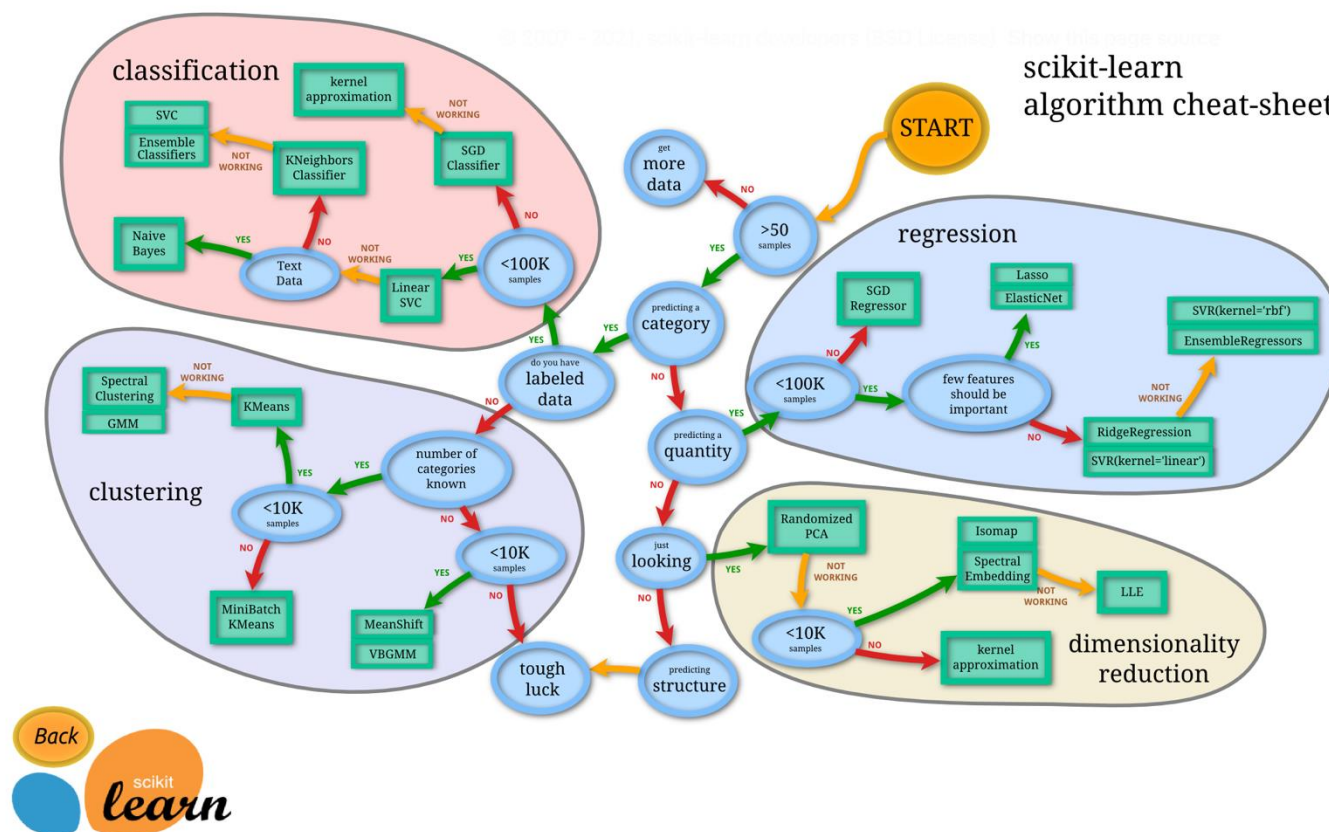


Step 3. Model training

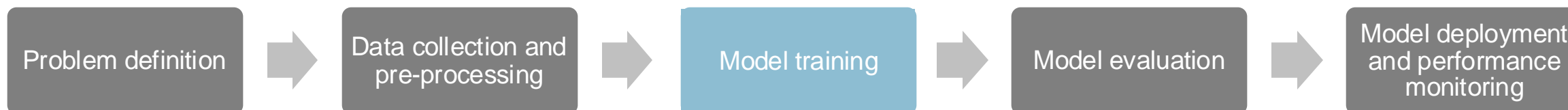
We need to **select an algorithm** to be trained on our data.

- E.g., the prediction of a machine failure can be defined as a classification or a regression problem

To find out which algorithm is the best for our data set we have to test them.



https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

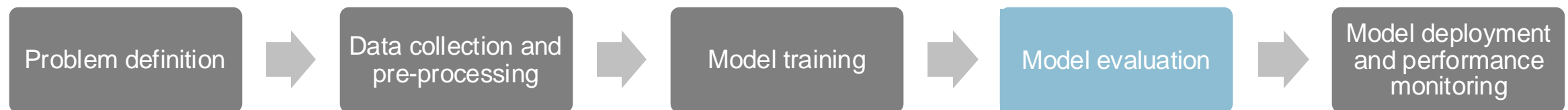


Step 4. Model evaluation

Training and evaluation of the model are iterative processes:

- First, we train our model with the training set
- Then evaluate its performance with validation set with evaluation metrics
- Based on this information, we **tune our algorithm's hyperparameters**

This iterative process continues unless we decide that we can't improve our algorithm anymore.



Step 4. Model evaluation

Training and evaluation of the model are iterative processes:

- First, we train our model with the training set
- Then evaluate its performance with validation set with evaluation metrics
- Based on this information, we **tune our algorithm's hyperparameters**

This iterative process continues unless we decide that we can't improve our algorithm anymore.

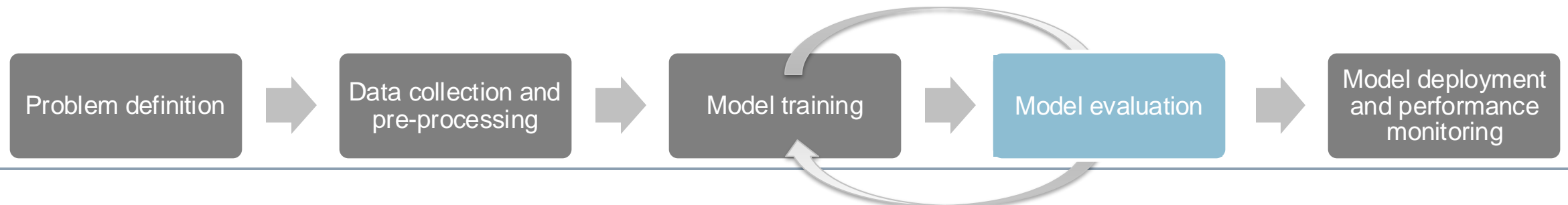


Step 4. Model evaluation

Training and evaluation of the model are iterative processes:

- First, we train our model with the training set
- Then evaluate its performance with validation set with evaluation metrics
- Based on this information, we **tune our algorithm's hyperparameters**

This iterative process continues unless we decide that we can't improve our algorithm anymore.



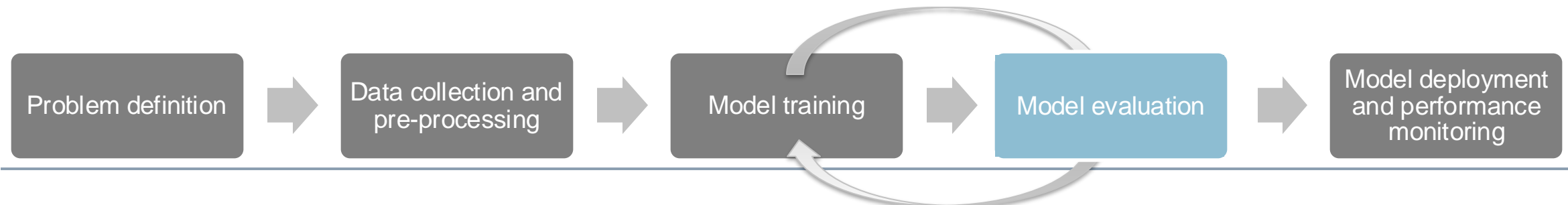
Step 4. Model evaluation

Training and evaluation of the model are iterative processes:

- First, we train our model with the training set
- Then evaluate its performance with validation set with evaluation metrics
- Based on this information, we **tune our algorithm's hyperparameters**

This iterative process continues unless we decide that we can't improve our algorithm anymore.

Then we **use the test set** to see the performance on **unknown data**.



Classification models evaluation – the confusion matrix

Confusion Matrix describes the performance of the model.

There are 4 important terms:

True Positives: The cases in which we predicted YES, and the actual output was also YES

True Negatives: The cases in which we predicted NO, and the actual output was NO

False Positives: The cases in which we predicted YES, and the actual output was NO

False Negatives: The cases in which we predicted NO, and the actual output was YES

Confusion Matrix		
n=165	Predicted:NO	Predicted:YES
	Actual: NO	Actual: YES
n=165	50	10
	5	100

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

$$\text{Accuracy} = (150/165) = 0.909$$

Regression models evaluation – metrics

Mean Absolute Error (MAE):

- Average difference between the original and predicted values
- Measure how far predictions were from the actual output
- Does not give an idea about the direction of error.

$$\text{Mean Absolute Error} = \frac{1}{N} \sum_{j=1}^N |y_j - \hat{y}_j|$$

Mean Squared Error (MSE):

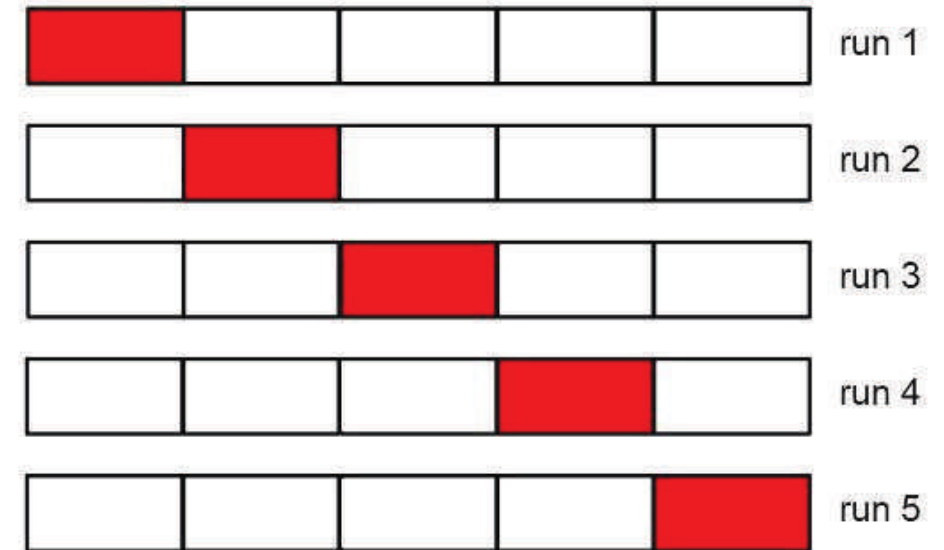
- Similar to MAE
- It takes the average of the square of the difference between original and predicted value
- Larger errors become more pronounced, so that the model can focus on larger errors.

$$\text{Mean Squared Error} = \frac{1}{N} \sum_{j=1}^N (y_j - \hat{y}_j)^2$$

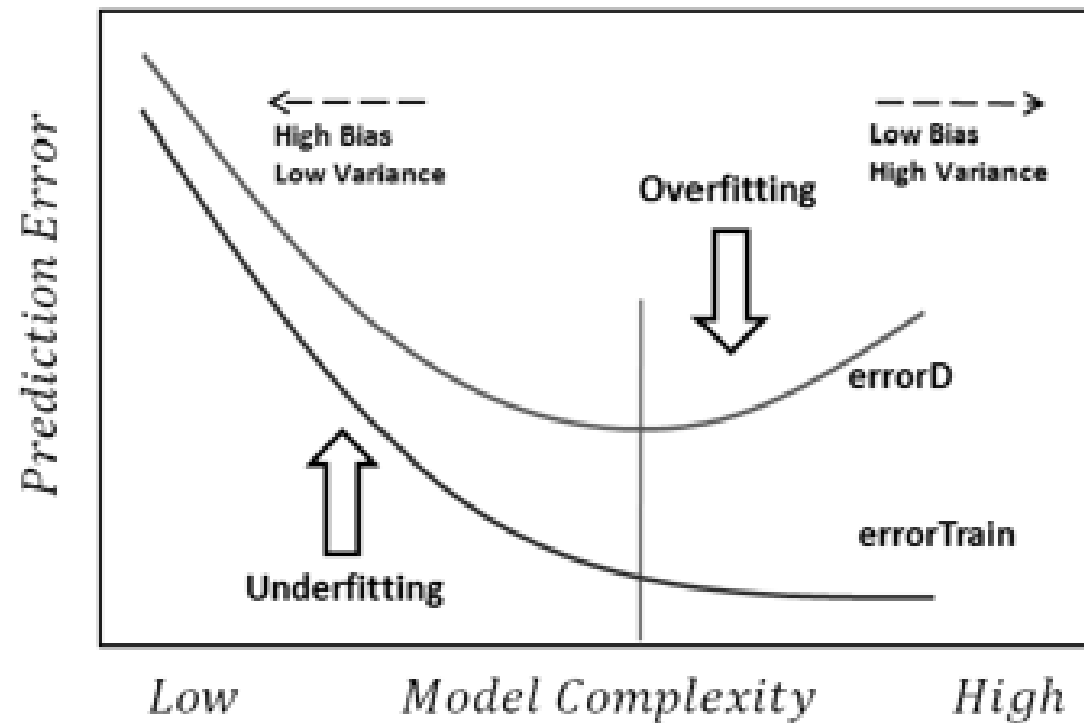
Cross validation

Cross-validation is a resampling method to iteratively use different portions of the dataset to train and validate a model among iterations.

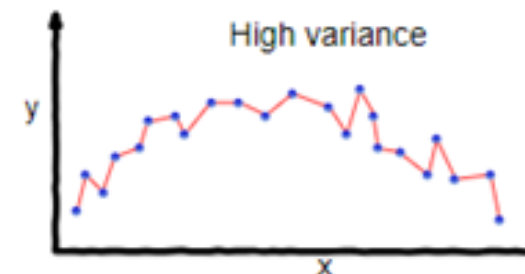
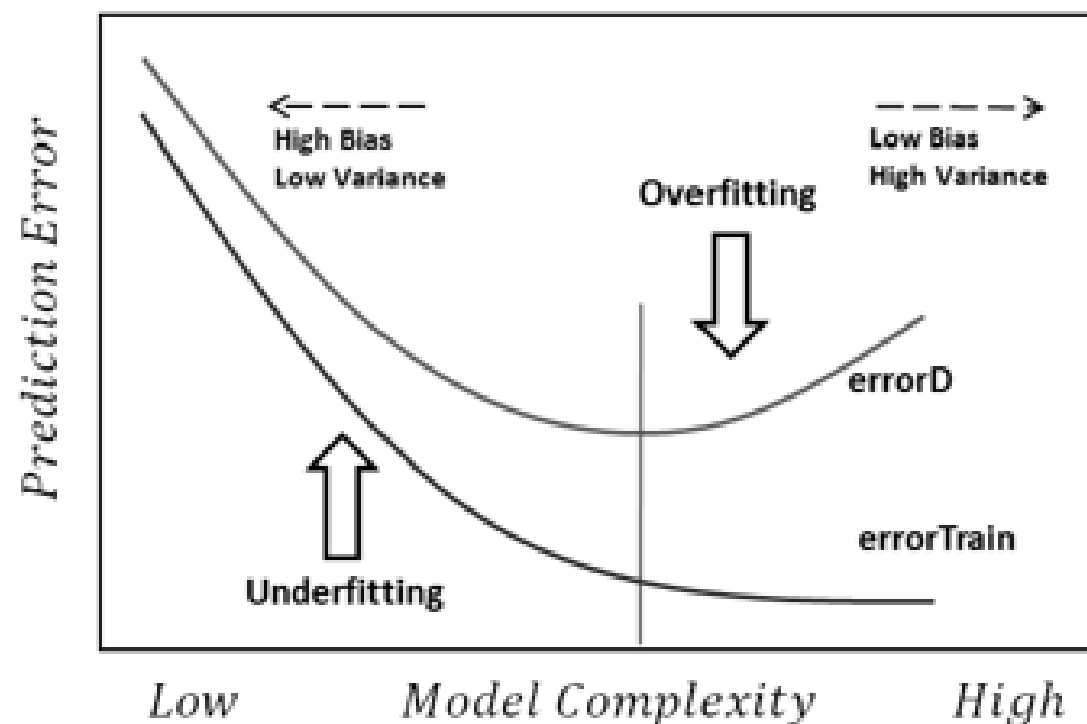
- Particularly useful when the dataset size is small
- It can be also used for hyper-parameters selection



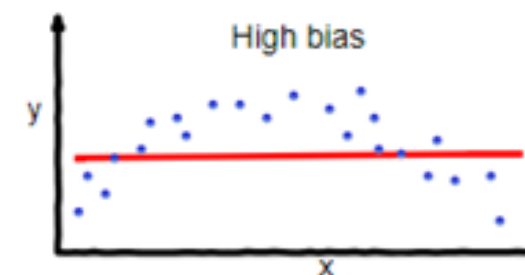
Underfitting and overfitting



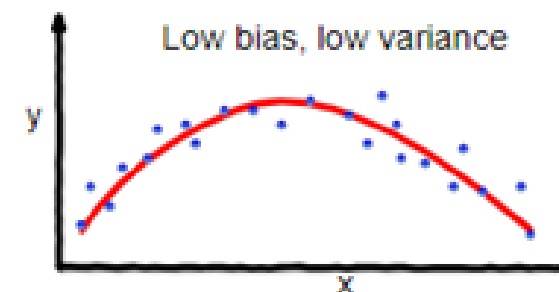
Underfitting and overfitting



overfitting



underfitting

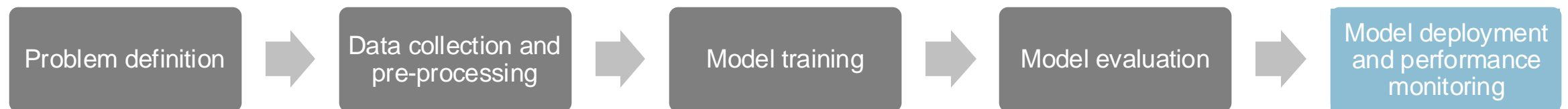


Good balance

Step 5. Model deployment and performance monitoring

After a successful training, we can **deploy the model**. Here we have often to consider the following issues:

- Real time requirements
- Robust hardware (sensors and processor)



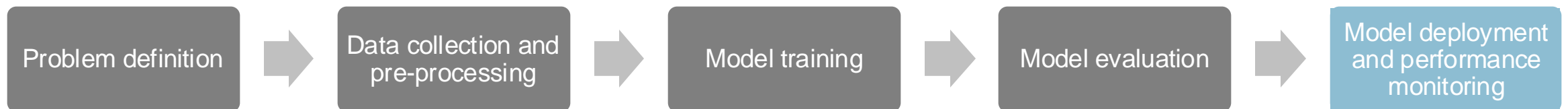
Step 5. Model deployment and performance monitoring

After a successful training, we can **deploy the model**. Here we have often to consider the following issues:

- Real time requirements
- Robust hardware (sensors and processor)

We must **monitor the performance** of our model and make sure it still produces satisfactory results.

- Sensors can malfunction and provide wrong data
- The data can be out of the trained range of the model





ADLTS \ Introduction \ Common ML tasks with time series



The machine learning tasks

Machine learning (ML) can be regarded as a collection of methods that enables us to solve **tasks** which would be too difficult to be solved by a fixed written program designed by human beings.

From a philosophical point of view this is interesting because it can be seen as an attempt to formalize the concept of **intelligence**.

ML usually describes how machines should process **examples**.

- Examples are collections of **features**
- In the case of time series, features are the observations sorted in time.

Time series classification

Let $\mathcal{D} = \{(S^{(1)}, c^{(1)}), \dots, (S^{(N)}, c^{(N)})\}$ be a dataset of pairs, where

- $S^{(i)}$ is a time series
- and $c^{(i)} \in \{0,1\}^K$ denotes the one-hot encoded class vector (also said, labels vector).

Then, a time series classification task is about learning a mapping function f , such that:

$$f(S^{(i)}) = c^{(i)}, \forall i \in \{1, \dots, N\}$$

$$f\left(\text{[blue time series plot]}\right) = \boxed{\text{[gray circle] [yellow circle]}}$$

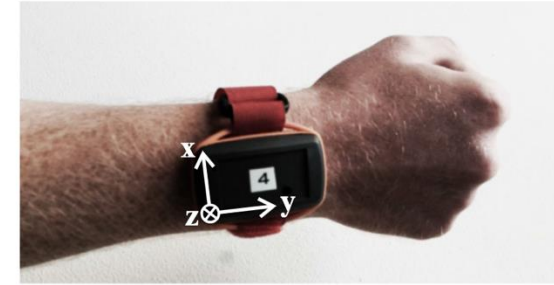
$$f\left(\text{[brown time series plot]}\right) = \boxed{\text{[gray circle] [yellow circle]}}$$

Example of time series classification

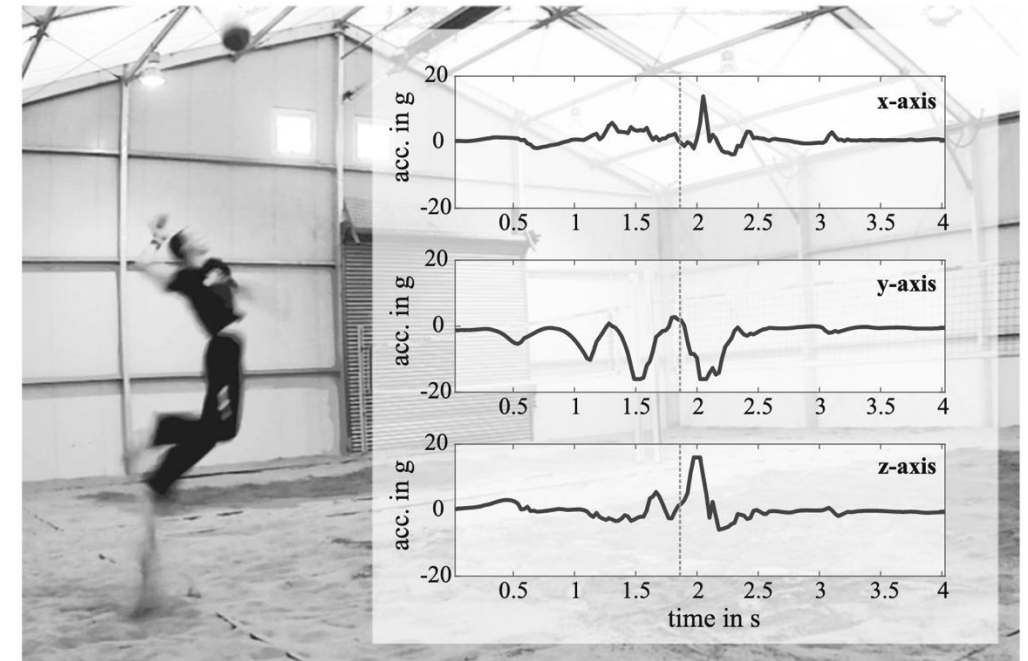
- Monitoring of player actions could help identifying and understanding risk factors and prevent such injuries.

Actions:

- Underhand serve
- Overhand serve
- Jump serve
- Underarm set
- Overhead set
- Shot attack
- Spike
- Block
- Dig
- Null class.



Sensor attachment at the wrist of the dominant hand with a soft, thin wristband

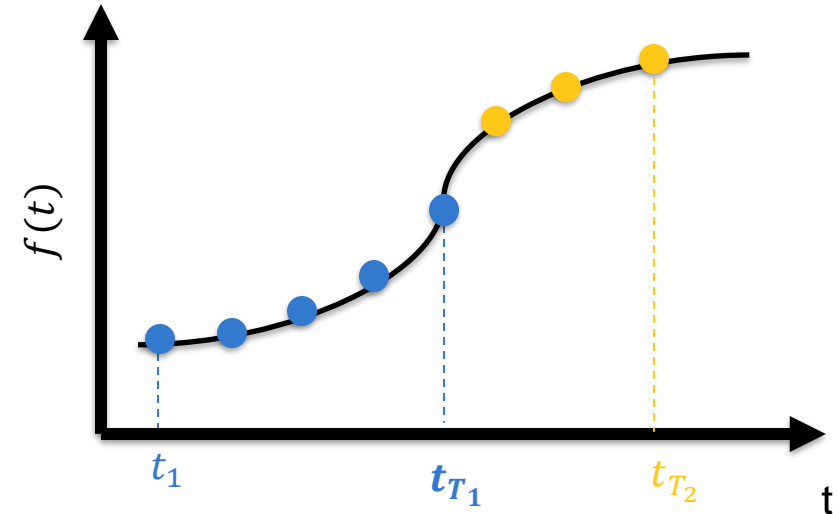


Time series forecasting

Let $S = \{s_1, \dots, s_{T_1}, s_{T_1+1}, \dots, s_{T_2}\}$ be a time series, with s_i being the i -th observation collected at time t_i , and $t_i < t_j, \forall j$.

Then, a time series forecasting task is about predicting future values of a time series given some past data, i.e.,

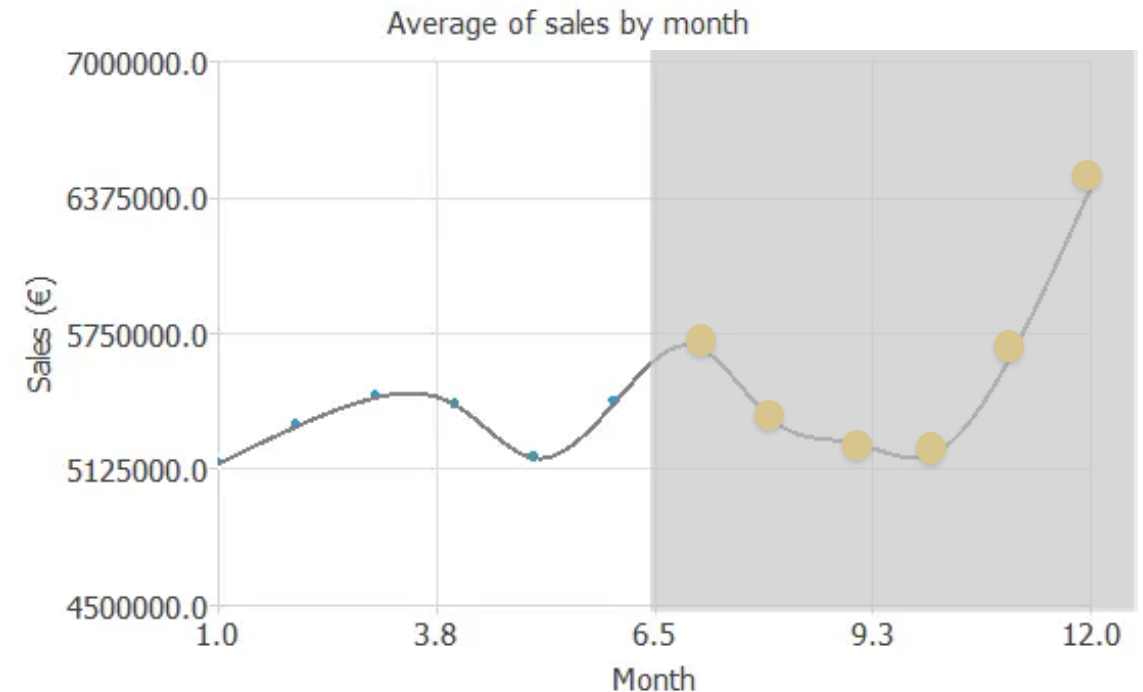
$$f(s_1, \dots, s_{T_1}) = (s_{T_1+1}, \dots, s_{T_2})$$



Example of time series forecasting: Forecasting Monthly Sales for a Retail Store

Scenario: A retail store wants to predict future sales to optimize inventory management, staffing, and promotions. The data available consists of monthly sales figures for the last 3 years.

Objective: Predict sales for the next 6 months.



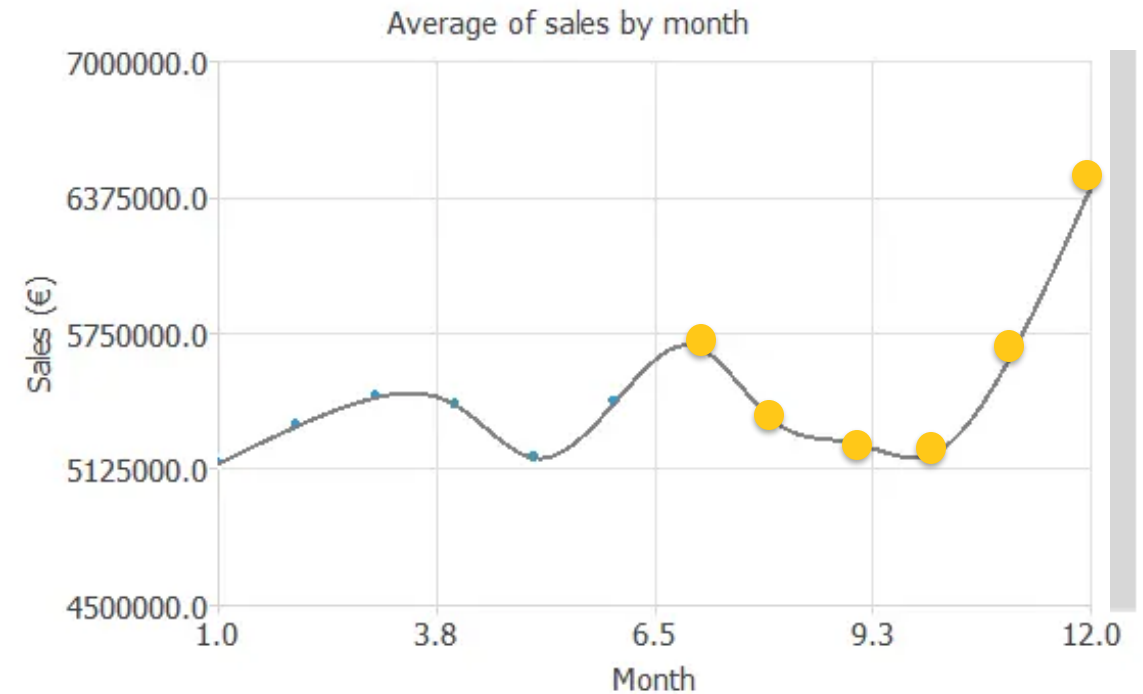
● = Collected data

● = Predicted by a model

Example of time series forecasting: Forecasting Monthly Sales for a Retail Store

Scenario: A retail store wants to predict future sales to optimize inventory management, staffing, and promotions. The data available consists of monthly sales figures for the last 3 years.

Objective: Predict sales for the next 6 months.



● = Collected data

● = Predicted by a model

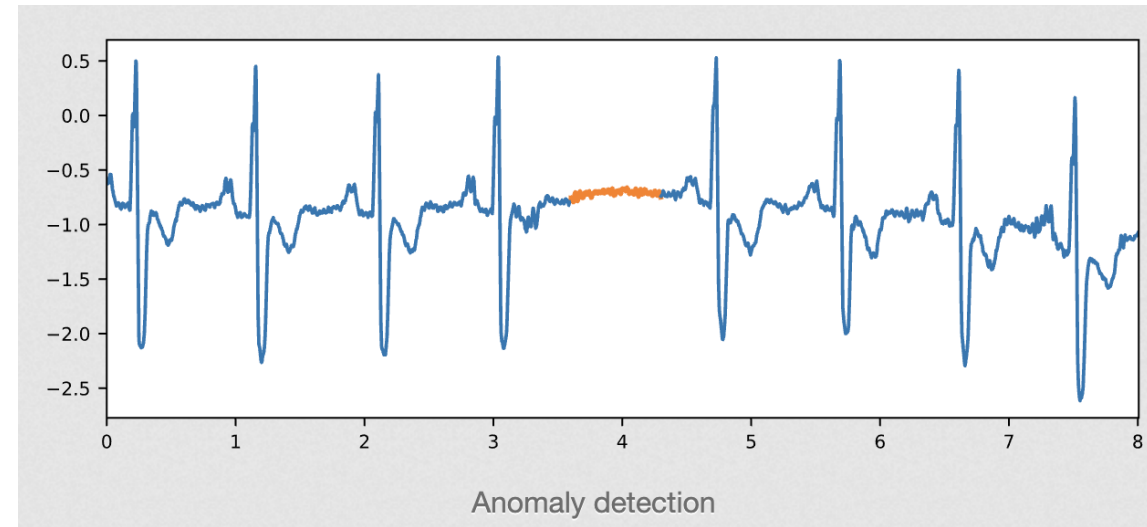
Anomaly detection on time series

Let $S = \{s_1, \dots, s_T\}$ be a time series, with s_i being the i -th observation collected at time t_i .

Then, an anomaly detection task is that of predicting the probability of a certain observation to be anomalous,

$$f(s_i) = p_i, \forall i \in \{1, \dots, T\}$$

with $p_i = 0$ for regular data and $p_i = 1$ for anomalous data.



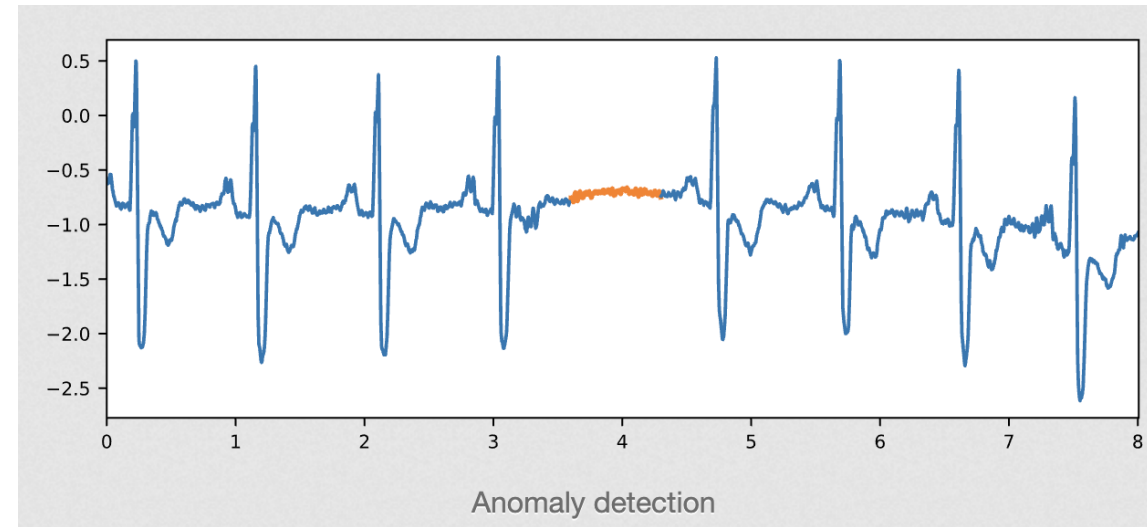
(a) <https://siebert-julien.github.io/time-series-analysis-python/>

Examples of anomaly detection on time series

Anomaly detection, sometimes also called outliers detection or novelty detection, is therefore the *task of finding abnormal data points* (equiv., outliers).^(a)

Examples of real world applications of anomaly detection on time series are:

- detecting fraud transactions
- fraudulent insurance claims
- cyber attacks to detecting abnormal equipment behaviors



(a) <https://siebert-julien.github.io/time-series-analysis-python/>

Time series segmentation

Let $S = \{s_1, \dots, s_T\}$ be a time series, with s_i being the i -th observation collected at time t_i .

Time series segmentation is the task of splitting data points into segments, which reveal underlying properties of the generation process, which can be formalized as the process of assigning every sample to its corresponding cluster, i.e., $f(s_i) = c_i$, with $c_i \in \{c_1, \dots, c_M\}$.

Example of time series segmentation

A typical example is that of online handwriting recognition.

A time series describes a list of coordinates, e.g., the point of a pen over a touchscreen surface, collected over time.

The task is to determine segments corresponding to a single letter.

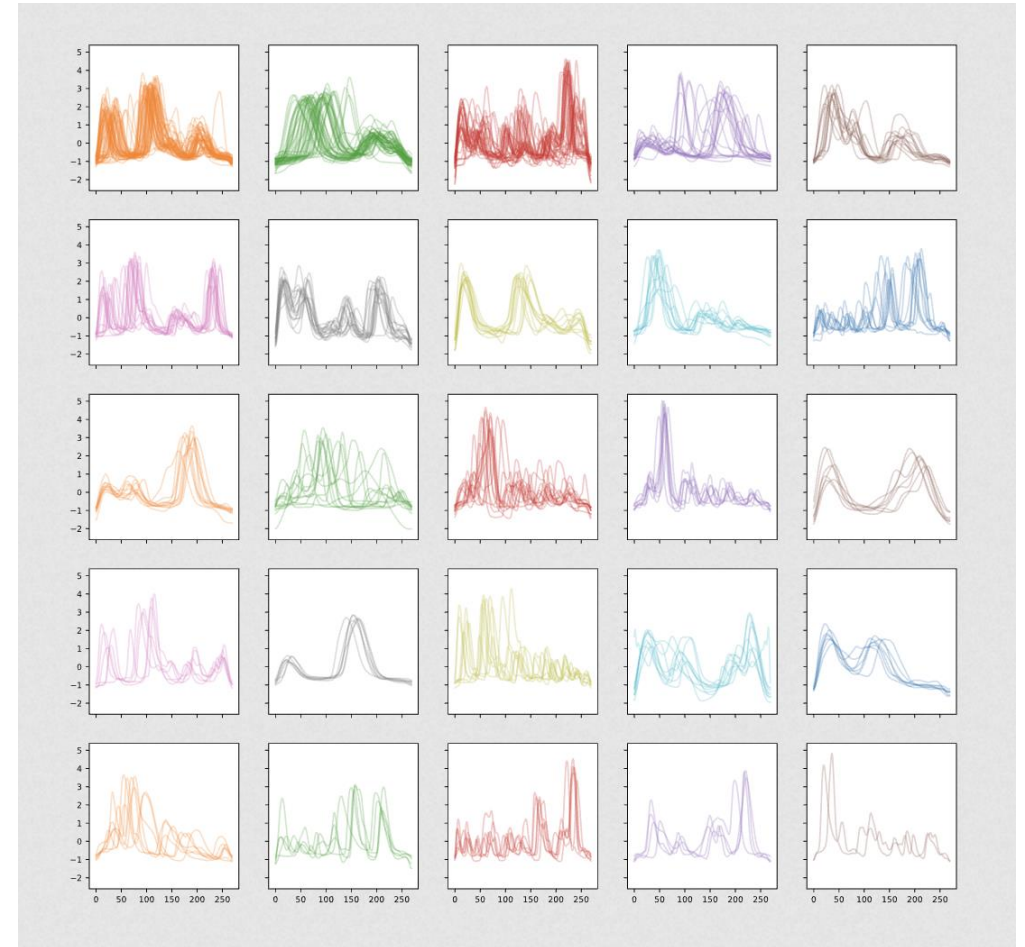


Time series clustering

Clustering can be applied to time series with the goal of grouping together similar sequences.

This finds important application in data analysis and pre-processing

- Find similar customers behaviours and exploit this information in recommender systems



(a) <https://siebert-julien.github.io/time-series-analysis-python/>



ADLTS \ Introduction \ Recap



Lecture outline

1. Motivations and real-world examples
2. Definitions and basic properties
3. Types of ML
4. ML Pipeline
5. ML Tasks for time series



