KI in der Produktionstechnik

Lingjie Zhang

2024 SS

This course is based on the lecture MW2455 of Technical University of Munich

Contents

1 Introduction

1.1 Definition of the Term Intelligence

Natural intelligence

Intelligence has been defined in many ways: the capacity for logic, understanding, self-awareness, learning, emotional knowledge, reasoning, planning, creativity, critical thinking, and problem-solving. More generally, it can be described as **the ability to perceive or infer information, and to retain it as knowledge to be applied towards adaptive behaviors** within an environment or context. Intelligence is most often studied in humans but has also been observed in both non-human animals and in plants.

The **intelligencer quotient (IQ)** is a **parameter** determined by an intelligence test **to evaluate intellectual performance** in general (general intelligence) or within a certain range (e.g, factors of intelligence) in comparison to a reference group. It always refers to a specific test, since there is no scientifically recognized, unambiguous definition of intelligence.

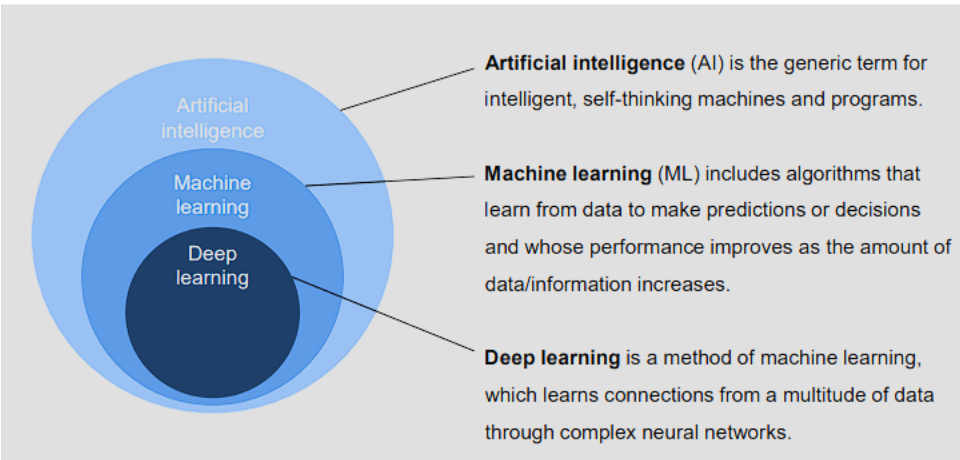
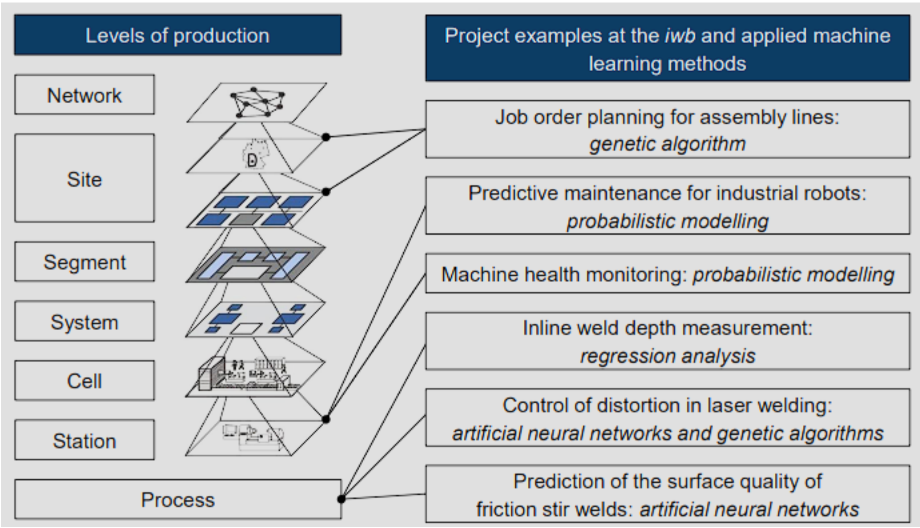


Figure 1.1: Categorization of Artificial Intelligence, Machine Learning and Deep Learning

1.2 Artificial Intelligence in Production Engineering

1.2.1 Machine Learning on the Different Levels of Production

Examples of machine learning at the iwb:



1.2.2 Example 1 - Job Order Planning for Assembly Lines

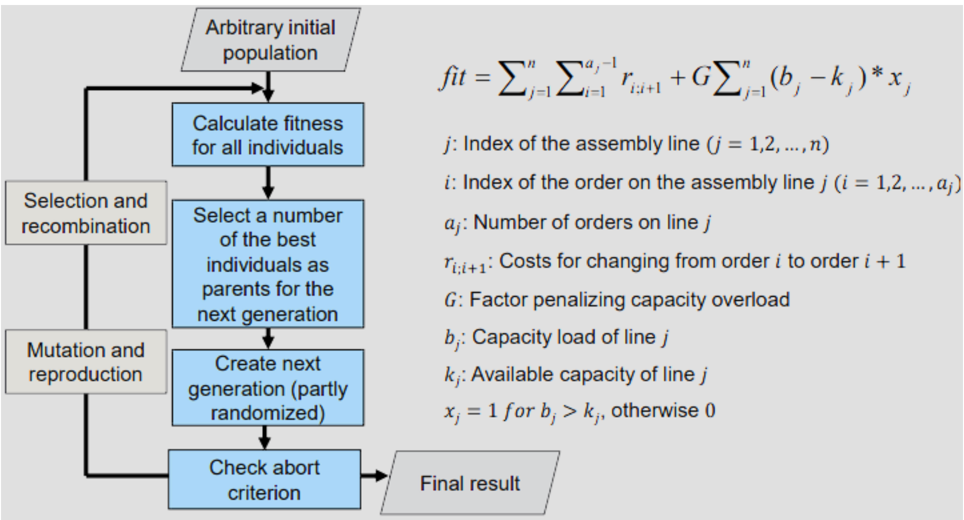
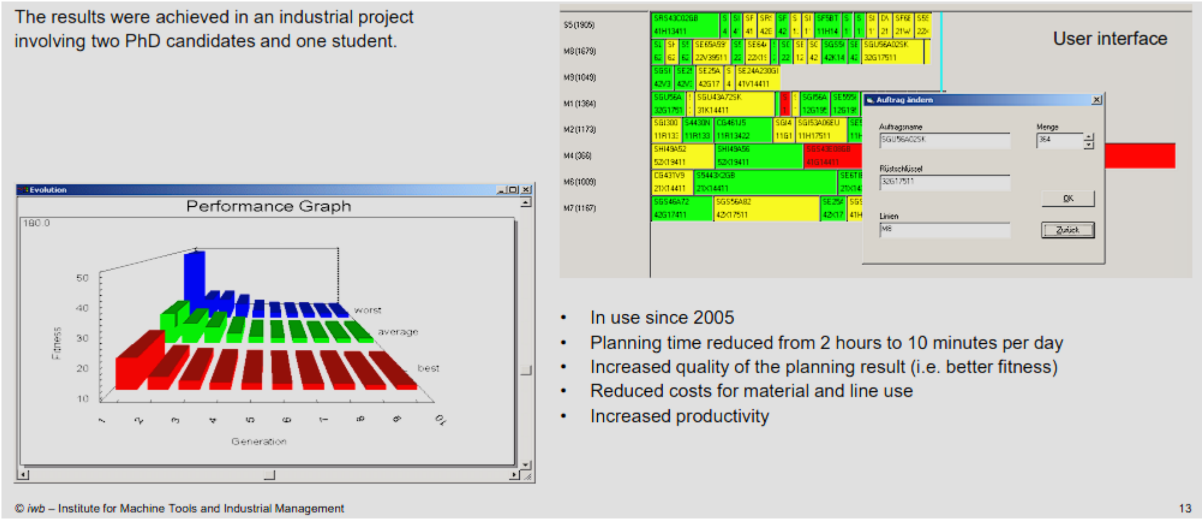


Figure 1.2: Solution: genetic algorithm



1.2.3 Example 2 - Minimization of Welding Distortions

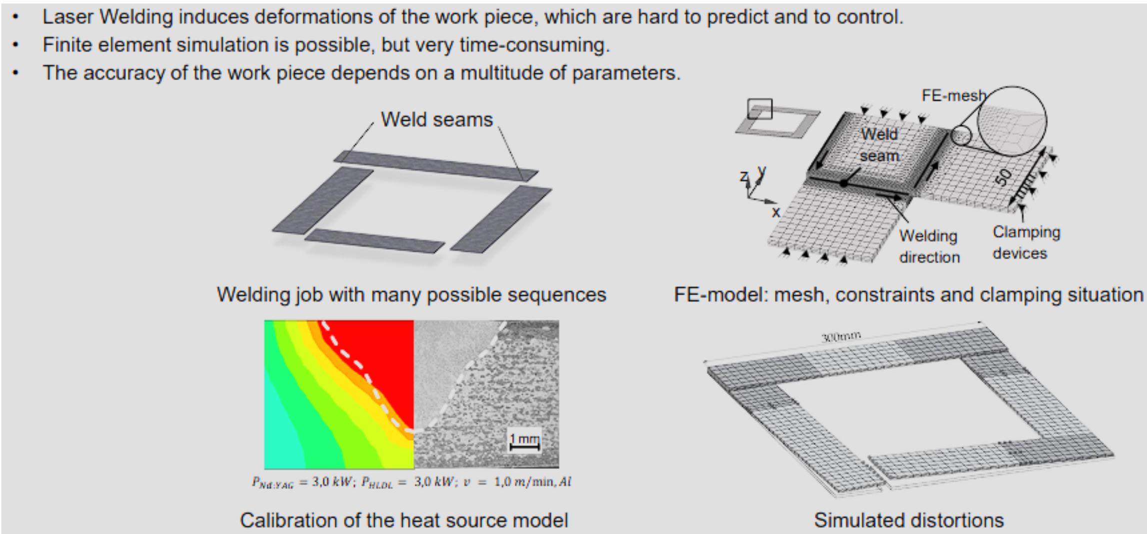
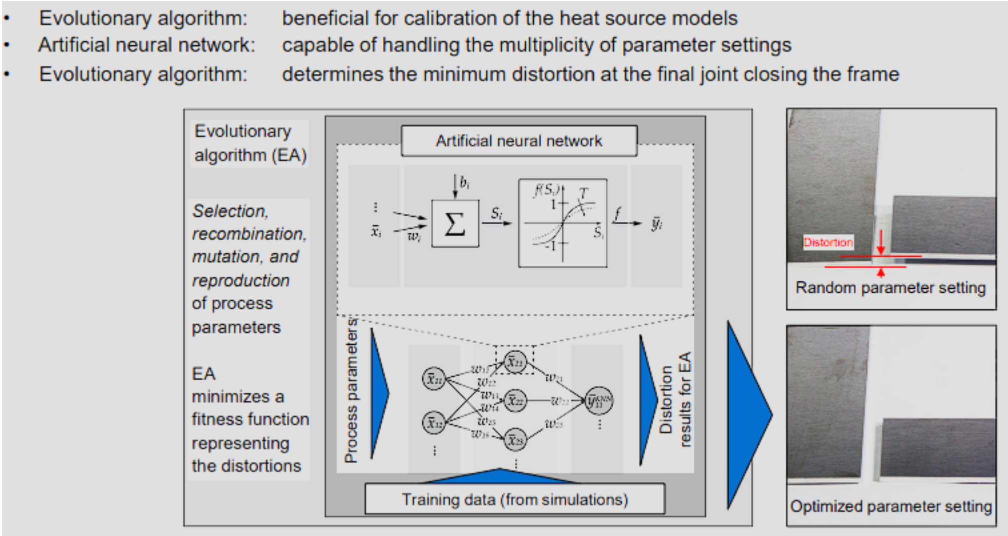


Figure 1.3: Artificial neural networks and evolutionary algorithms



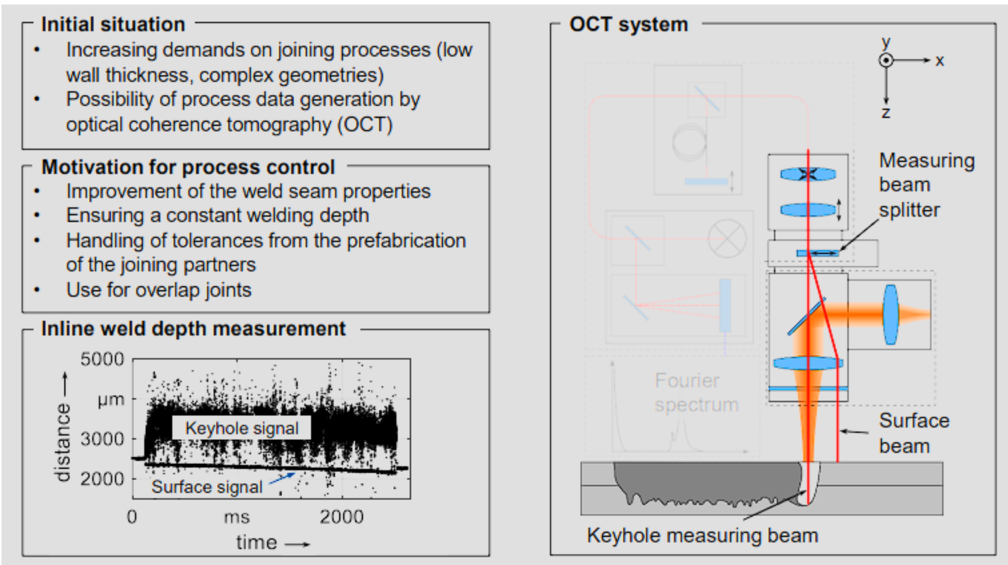


Figure 1.4: Initial situation and motivation

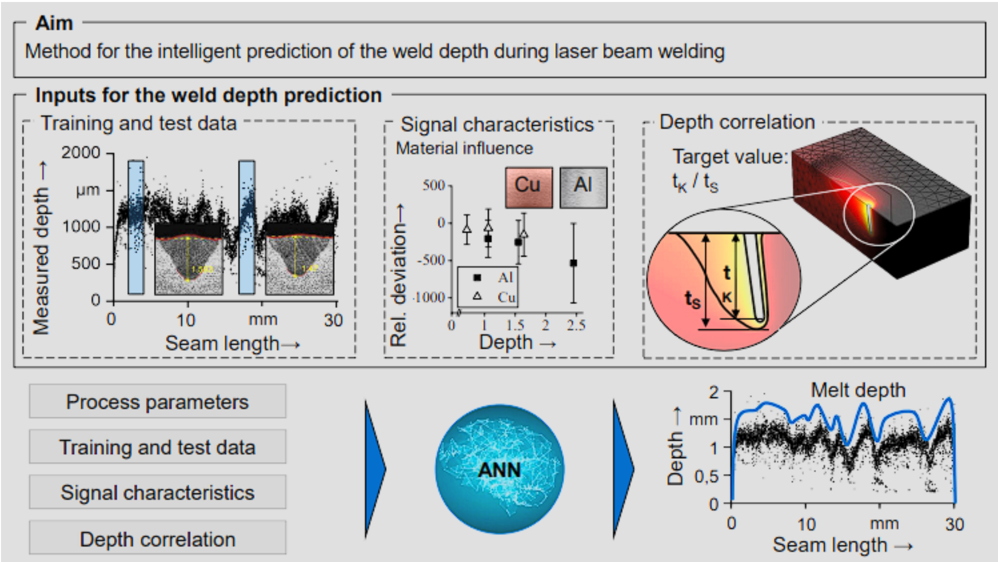


Figure 1.5: AI-based prediction of the weld depth

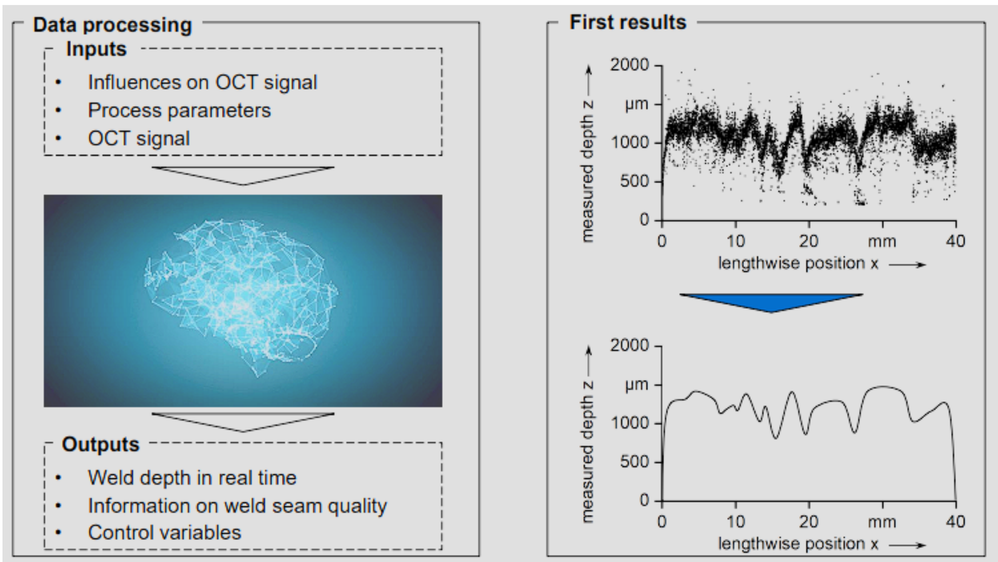


Figure 1.6: Next steps ahead

1.3 Introduction to the course and to Artificial Intelligence (AI)

MATH & STATISTICS (Lecture)
understand and remember the basics of machine learning
understand data structures, storage, preparation, features and models
retrieve, compare and generalize basic methods

DOMAIN KNOWLEDGE (Lecture)
discuss applications of AI methods in production engineering
challenges and approaches for the application of KDD1 in the production
discuss and generalize solution concepts for industrial applications

PROGRAMMING & DATABASE (Practice)
apply tools for data analysis purposes and the implementation of models
apply the basic procedure to an exemplary data set
identify the key challenges and derive appropriate measures to overcome them

COMMUNICATION & VISUALIZATION (Group project)
translate data-driven insights into decisions and actions
analyze practical problems and derive appropriate steps
design concepts for knowledge acquisition from production/process data

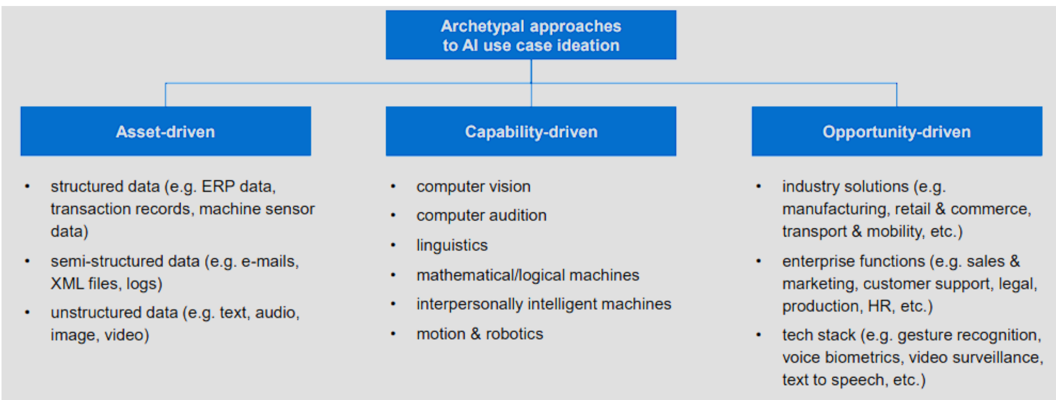


Figure 1.7: Application fields and opportunities of AI

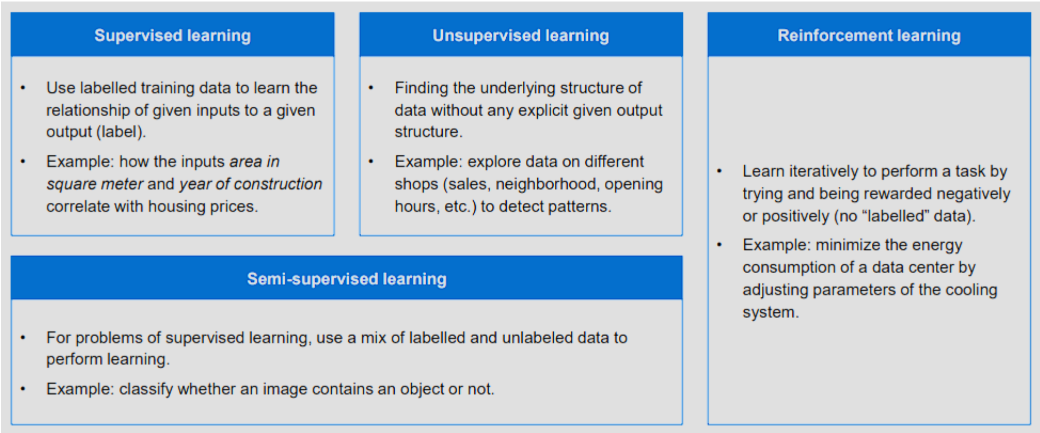


Figure 1.8: What is Artificial Intelligence?

- **Supervised learning** (covered in this course)
  - regression
  - classification
- **Unsupervised learning** (briefly covered in this course)
  - clustering
  - data compression
- **Reinforcement learning** (briefly covered in this course)
  - behavior selection
  - planning
- **Evolutionary learning** (not covered in this course)
  - general purpose learning

1.4 Recommended literature

- C.M. Bishop: Pattern recognition and machine learning. New York: Springer, 2006.
- T.Hastie, R. Tibshirani, und J.Friedman: The Elements of Statistical Learning. New York: Springer, 2009.

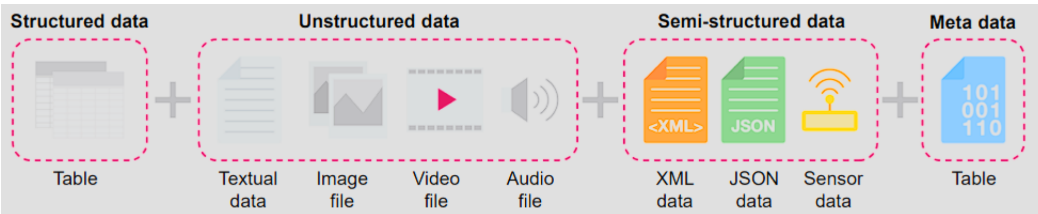
2 Data formats and sources

2.1 Learning objectives

After participating, you will be able to⋯

understand different data structures and formats and remember the respective data sources from the production environment.

2.2 Data formats and structure



Structured data

Adheres to a pre-defined model, that specifies how data can be stored, processed and accessed.
Straightforward to analyze as data can be aggregated quickly from various locations.
Examples: Excel files, SQL databases.

Unstructured data

Information that does not have a pre-defined data model or that is nor organized in a pre-defined manner.
Combination of text with data such as dates, numbers and facts result in irregularities that impede a simple processing.
Examples: Audio files, video files and No-SQL databases.

Semi-structured data

Form of structured data that does not conform with the formal structure of data models associated with relational databases.
Contains tags or other markers to separate semantic elements and enforce hierarchies of records and fields within the data.
Examples: JSON data, XML data

Meta data

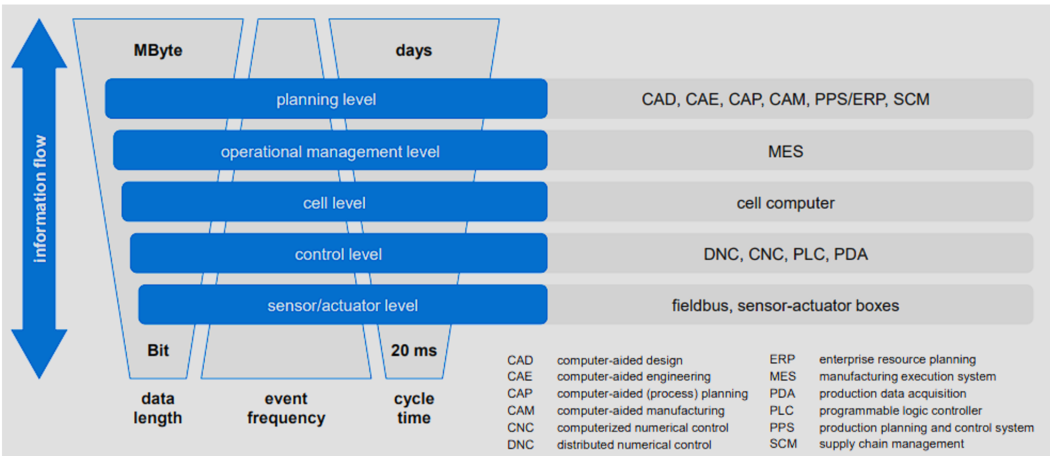
Meta data is technically not a separate form of data structure, but data about data, that provides additional information about a specific set of data.
Frequently used for initial analyses in big data solutions.
Examples: Location and time of a photograph

2.3 Data quality

Data quality			
Intrinsic data quality	Contextual data quality	Representational data quality	Accessibility data quality
believability	value-added	interpretability	accessibility
accuracy	relevancy	ease of understanding	access security
objectivity	timeliness	representational consistency	
reputation	completeness	concise representation	
	appropriate amount of data		

2.4 Data sources

2.4.1 Architectural model of the computer-integrated production



2.4.2 4.2 Information flow

Information flow - KDD process

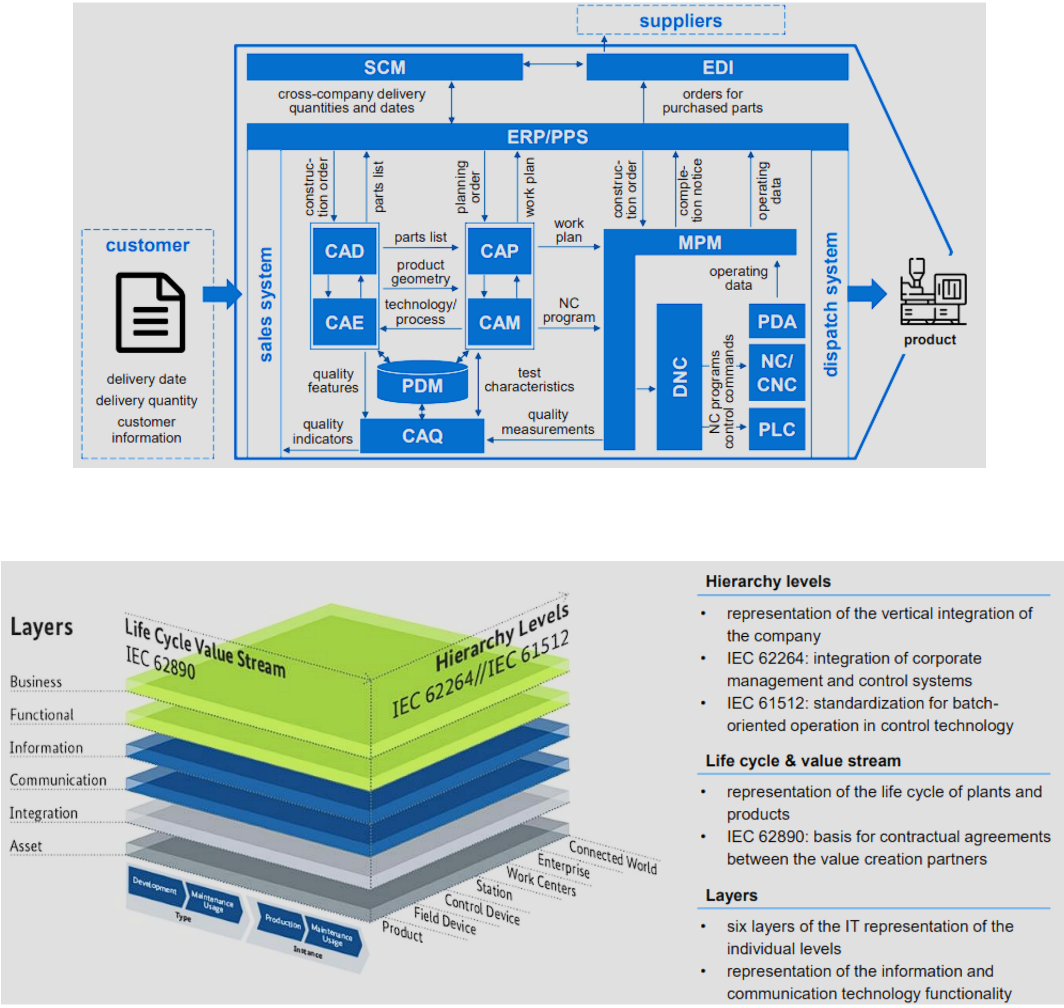


Figure 2.1: Information flow - Reference Architecture Model Industry 4.0 (RAMI 4.0)

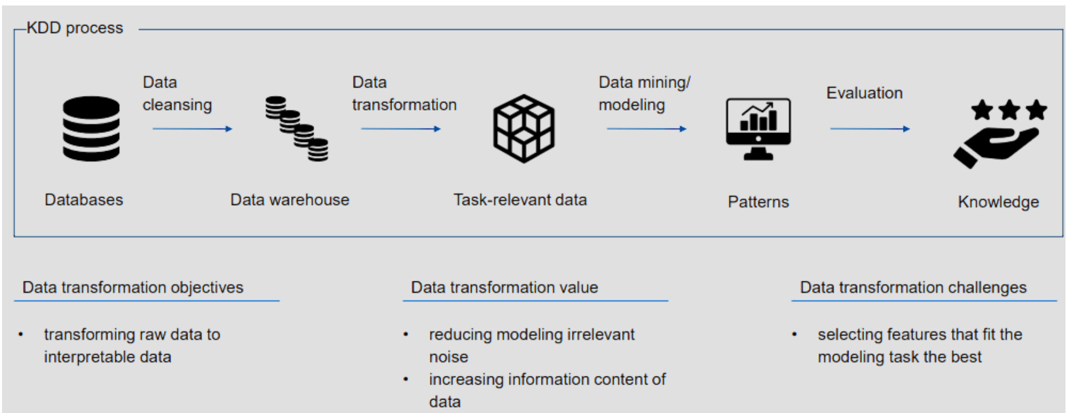


Figure 2.2: Information flow - KDD process

2.5 Summary

What you might have gathered throughout this lecture

- the characteristics of different data types and formats
- the dimensions of data quality
- representative models of production processes and architectures and the allocation of data sources within these models

After a recap, you should be able to⋯

- understand different data structures and formats and remember the respective data sources from the production environment.

3 Databases and Data Cleansing

3.1 Knowledge Discovery in Databases (KDD)

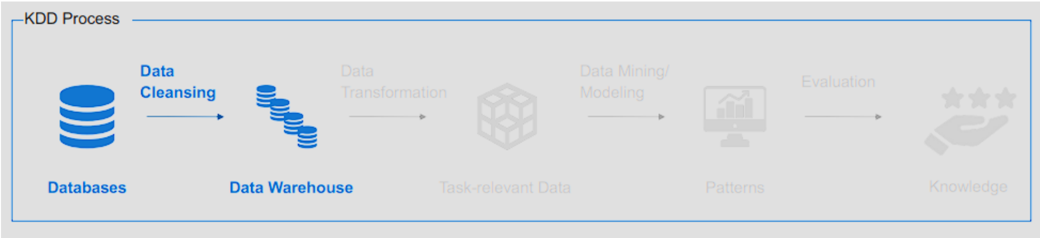


Figure 3.1: Overview

The data passes through an operational data storage and requires cleansing to ensure the data quality before it is used in the data warehouse for reporting and analysis.

3.2 Learning Objectives

Databases

- Understanding how data can be stored in databases and data warehouses
- Understanding the structure of different databases
- Processing of database queries with standardized query language (SQL)

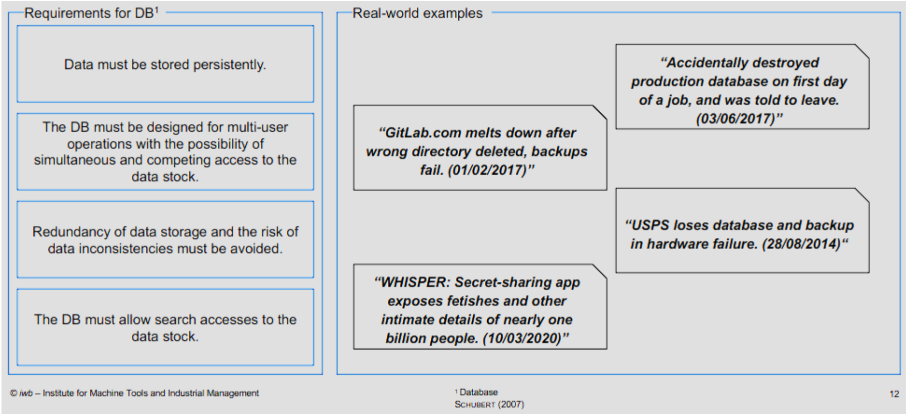
Data Cleansing

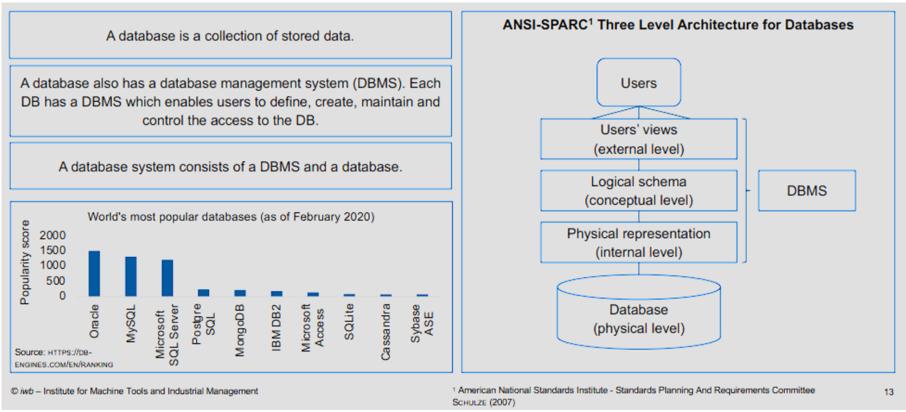
- Understanding the quality of data in the database
- Getting to know the workflow of data cleansing
- Understanding how data quality issues can be identified
- Getting familiar with the methods of data cleansing

3.3 Databases

Databases manage and access data efficiently.

3.3.1 Functionality and Structure





3.3.2 Database Types

	SQL – Standardized Query Language	NoSQL – Not only SQL
Definition	SQL databases are primarily called RDBMS[1] or relational databases	NoSQL databases are primarily called non-relational or distributed databases
Structure	Table-based databases	Document, wide-column, key-value or graph databases
Main principle	ACID (**A**tomicity, **C**onsistency, **I**solation, **D**urability)	BASE (**B**asically **A**vailable, **S**oft state, **E**ventually consistent)
Scalability	Particularly in the vertical direction combined with an increased administrative overhead	High scalability in vertical and horizontal direction remains constant despite high data volume
Language	Structured query language (SQL)	No declarative query language
Exemplary databases	▪ Oracle ▪ MySQL ▪ MS SQL Server ▪ PostgreSQL ▪ Sybase	**Data model** – **Examples** Wide column – Cassandra, HBase, Microsoft Azure Cosmos DB Document – CouchDB, MongoDB, riak Key-value – Amazon web services – simple DB, Redis Graph-DB – Neo4j, Microsoft Azure Cosmos DB, Arango DB
Exemplary users	Hootsuite, CircleCI, Gauges	Airbnb, Uber, Kickstarter

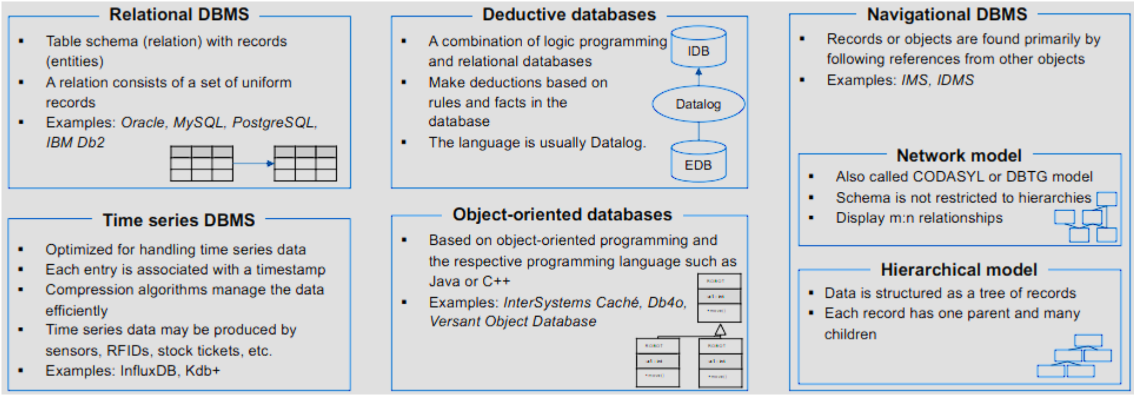


Figure 3.2: RDBMS are constantly being expanded, e.g. with object-oriented features, and are the most important DBMS.

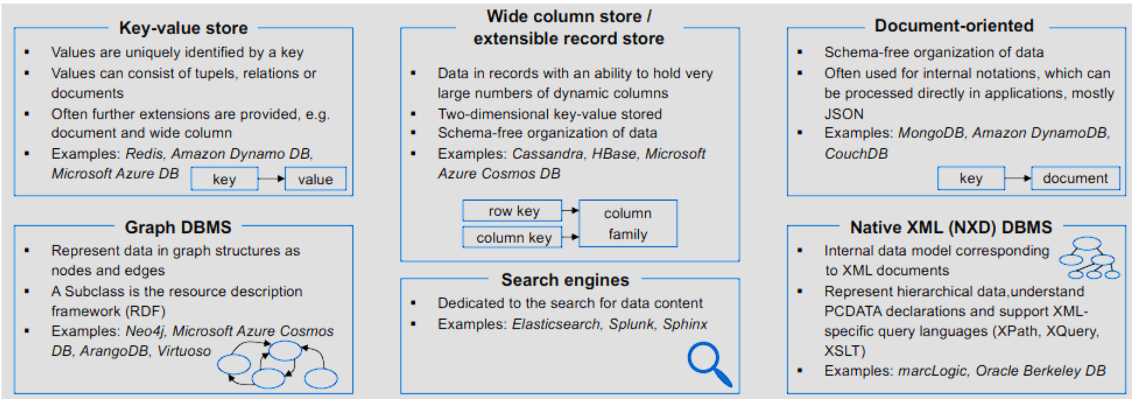


Figure 3.3: NoSQL systems gain popularity especially for Big Data applications.

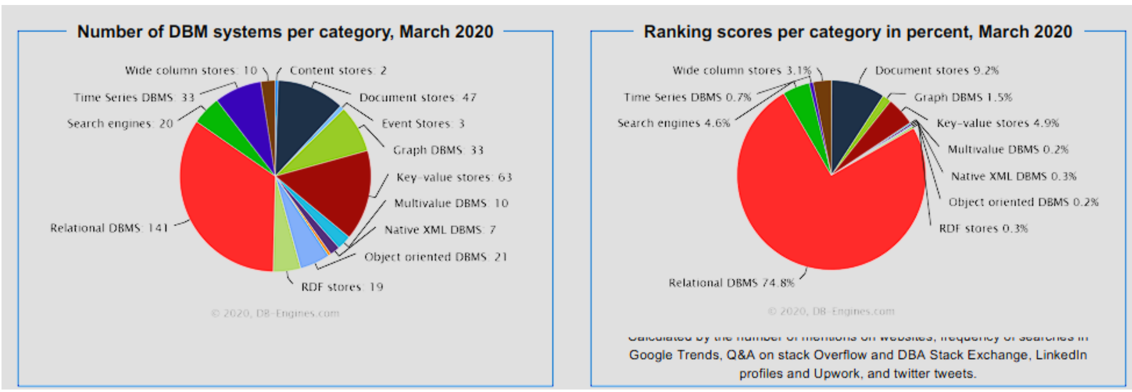
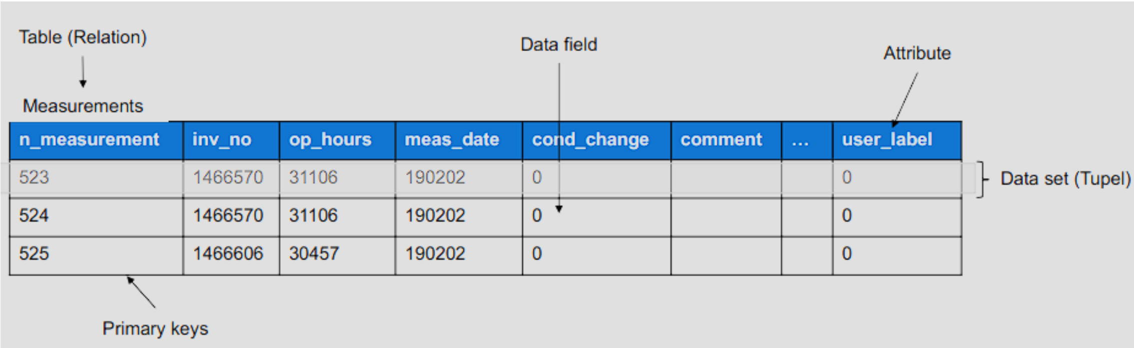


Figure 3.4: Relational DBMS dominate the market.

3.3.3 Relational Databases



Data is stored, changed, inserted and deleted in tables.

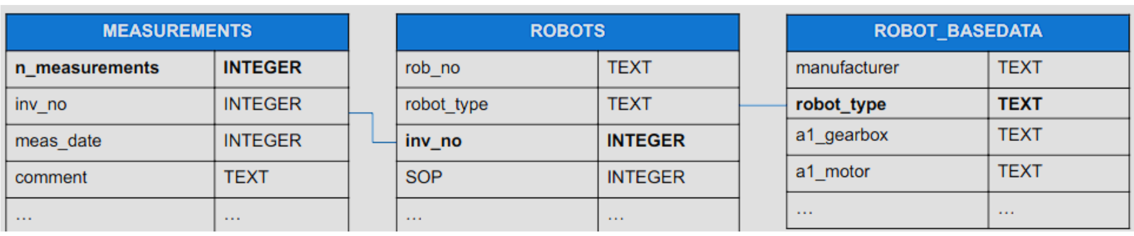
The logical **integrity** of a relational database is defined by the following conditions:

1. Each record in a table has a unique primary key value (entity integrity).
2. For each foreign key in the table T1. there is an identical key value in another table T2, which has been defined when T1 was created (referential integrity).
3. The remaining constraints are fulfilled (domain integrity).

The **primary key** is initially an attribute, or a combination of attributes of a table, that is **unique** for each record of the table.

A **foreign key** is an attribute or an attribute combination of a relation, which refers to a primary key (or key candidate) of another or the same relation.

Data integrity is a term for the assurance of the accuracy and consistency of data over its entire life-cycle.

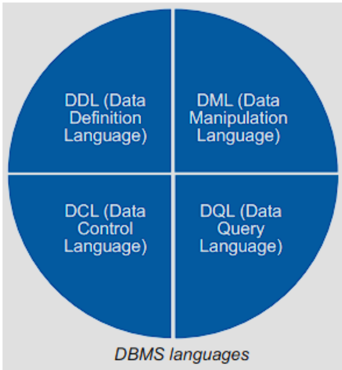


A table can refer to a column of another table by using a foreign key.

Relational representation of entity types

MEASUREMENTS: {[n_measurements: integer, inv_no: integer, meas_date: integer, …]}
ROBOTS : {[rob_no: text, robot_type: text, inv_no: integer, SOP: integer, …]}
ROBOT_BASEDATA : {[manufacturer: text, robot_type: text, a1_gearbox: text, a1_motor: text, …]}

3.3.4 SQL

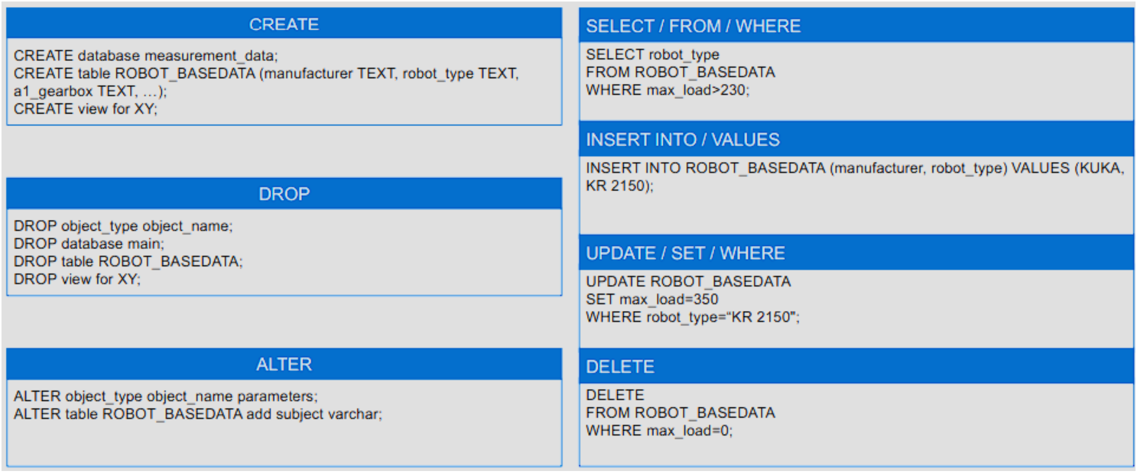


SQL = Structured Query Language

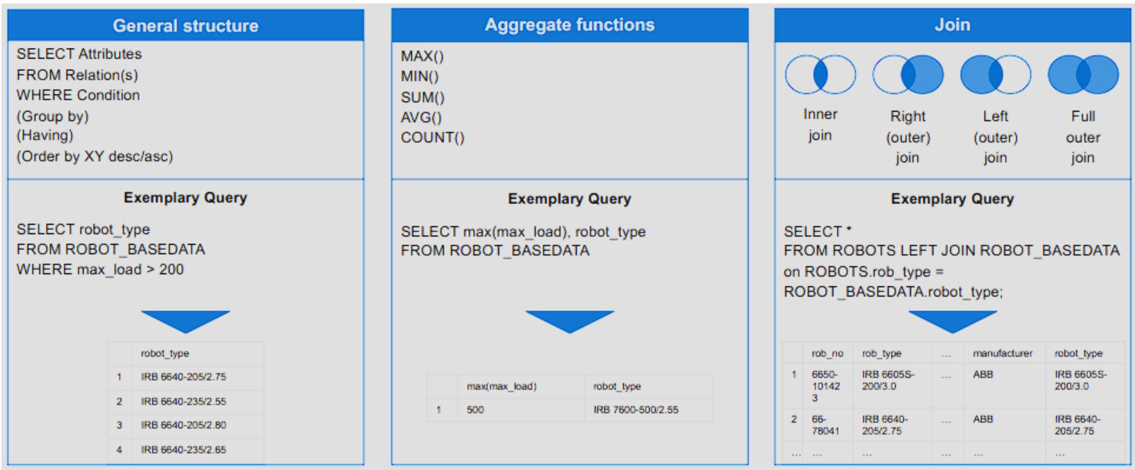
- Based on relational algebra
- Simple syntax
- Requires independence of the queries from the used DBMS.
- Interfaces to programming languages allow SQL commands to be transferred directly to a database system via a function call (e.g. via ODBC or JDBC).
- Even non-relational database systems are often equipped with an SQL-like interface.

DBMS languages are used to read, update and store data in a database and are specific to a particular data model. The dominant language is SQL.

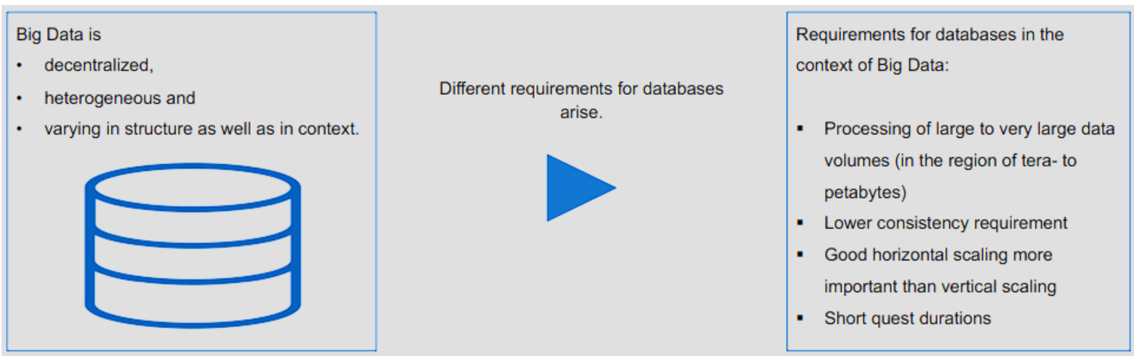
SQL: Data Definition and Manipulation Language



SQL: Data Query Language



Challenges of Big Data



For Big Data approaches NoSQL applications are often superior to RDBMS. However, for many database problems, the RDBMS remains the first choice.

3.4 Data Cleansing

Data cleansing and data integration usually accounts for 60% and more of the total effort.

3.4.1 Data Quality

Data cleansing can help diminish data quality issues concerning incompleteness and incorrectness. These issues are typically caused by human errors, limitations in measurement devices and flaws in the data collection process.

Measurement errors	**Data collection errors**
Discrepancy between the recorded value and the true value	Apparent where **data objects or attribute values** are **omitted** or data objects are **inappropriately included**
Systematic or random	Systematic or random

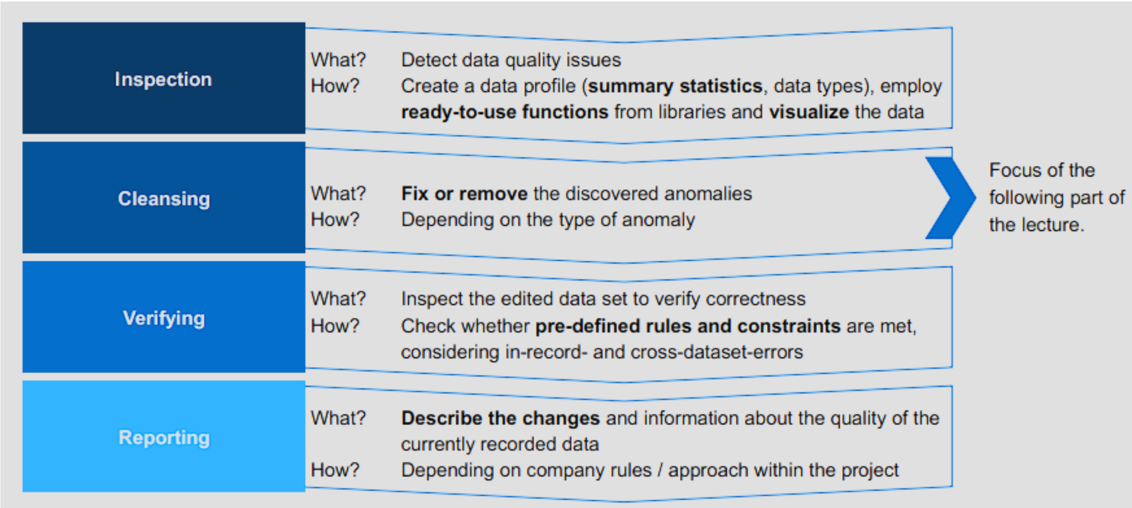
Given the high probability of data quality issues in real-life data, effort should be put towards detecting data quality issues and fixing those.

Data is of high quality, if it is suitable for its intended use!

Timeliness	**Relevance**	**Knowledge about data**
Dataset might only provide a snapshot of an ongoing phenomenon: If data is out of data, so are developed models and identified patterns	Available data must contain the information necessary for the application	Origin of the data must be known
	Objects in available dataset must be relevant	Information on value characteristics, scale of measurements, type of features and precision must be available.
		Strongly related attributes / variables must be identified, since they are likely to provide redundant information

Data quality issues bear even higher risks for data analytics projects, as they might not be discovered until all analysis have been performed. This makes domain knowledge even more valuable for such projects.

3.4.2 Workflow of Data Cleansing



3.4.3 Inconsistencies and Duplicates

Inconsistent values

Description:

Attribute values might be inconsistent, e.g. with regard to the permitted data type, categorical value or range.

Issue:

- Influence the outcome of any analysis and can ultimately lead to incorrect results
- Can only be identified if additional or redundant information is available

Solution:

1. Create a data profile giving insights into the data types, missing values and generate the summary statistics
2. Use libraries to set value constraints and to check for violation of these constraints

Duplicate data

Description:

Duplicates are data objects that are repeated / appear more than once within a data

Issue:

Lead to a discrepancy between the occurrence of data objects with certain characteristics in a dataset compared to the occurrence of such data objects in real life

Sloution:

Remove via numerous functions in different libraries → Attention should be put towards distinguishing real duplicates from **presumed duplicates**

3.4.4 Missing Values

Reasons for missing values

- Information was **not collected**
- Errors in manual data collection
- Equipment errors
- Measurement errors
- Attribute / variable **not applicable** to all objects
- **Non-integrable data sources**

Issues caused

- Loss of efficiency with regard to handling and analysis of the data
- Bias resulting from differences between missing and complete data

Detecting and exploring missing values

- Functions from different programming languages allow to detect and unify missing values.
- Checking the dimensions and verifying the data type
- Visualization of the distribution can be beneficial

It is important to assess the relevance of the missing values with regard to their frequency and their significance for further analysis.

Types of missing values

Missing Completely At Random (MCAR)

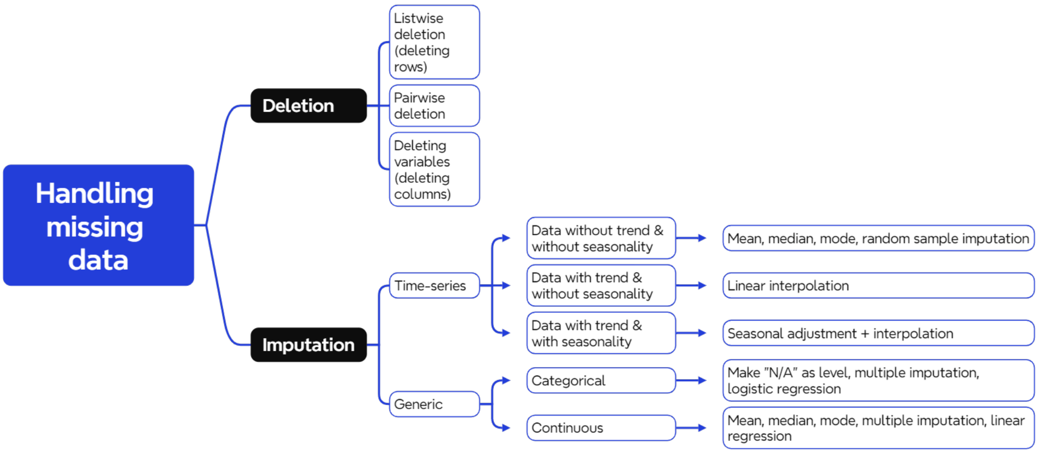
- Definition: Missing of a value is neither related to the variable it describes nor any other variable of the data object.
- Example: The sensor recording the regarded value was unavailable for that measurement.

Missing At Random (MAR)

- Definition: Missing of a value is not dependent on the variable it describes, but dependent on values of one or more other variables of the data object.
- Example: A measurement might not have been taken because another measurement already deemed the product a reject.

Not Missing At Random (NMAR)

- Definition: Missing of a value is dependent on its hypothetical value and/or other variable's values.
- Example: Elderly women are less likely to submit their age in questionnaires. The type of the missing values will influence which approach of handling missing values is feasible. Thus, it is imperative to be familiar with the different types.



3.4.5 Noise

Description

- Random component of a measurement error
- Distortion of a value or addition of spurious objects
- Typical causes:
- Environmental conditions (e.g. vibrations from other machines)
- Deployed sensor systems

Solution

Applying filters to signals decreases the signal-to-noise-ratio, but can also decrease the information content.

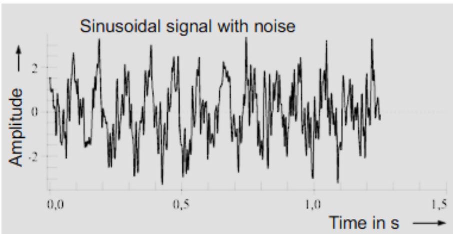
Solutions for noise in time series:

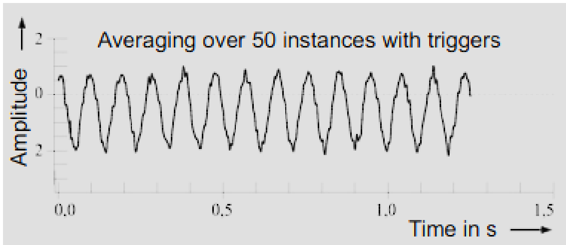
- Filters
- Averaging (only feasible if time-wise synchronized measurement):

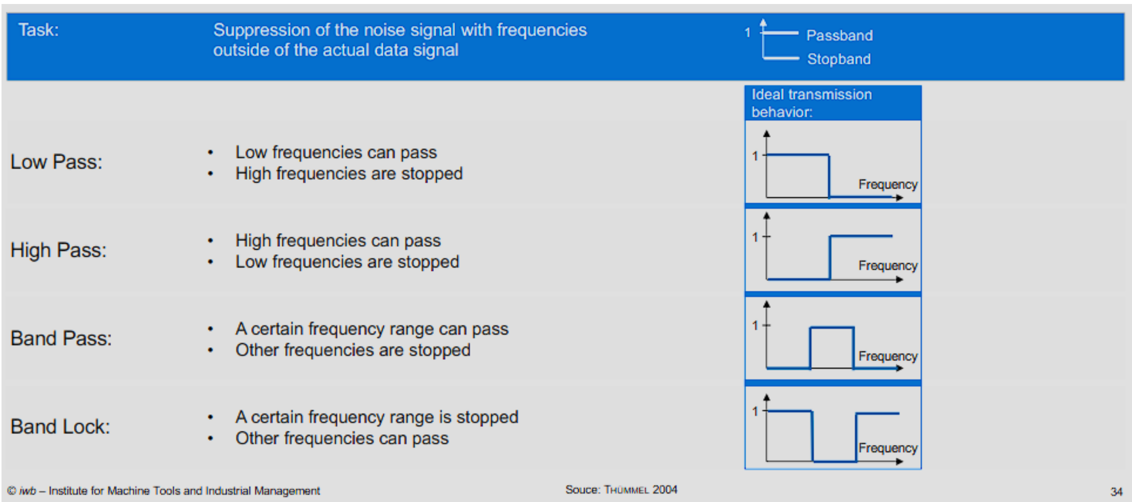
$$\{\bar{x}_m\} = \frac{1}{N}\sum_{n=1}^{N}\{x_m\}_k$$

Solutions for noise in images:

- Convolution with kernels for
- Edge detection
- Sharpening
- Smoothing







3.4.6 Outliers

Distinction from noise

Outliers can be valid and hold important information. → e.g. for condition monitoring

Types of outliers

- Data object outliers differ from the other data objects in the dataset in numerous characteristics.
- Attribute value outliers are identified through a comparison against the distribution of the rest of the values for that attribute.

Issues caused

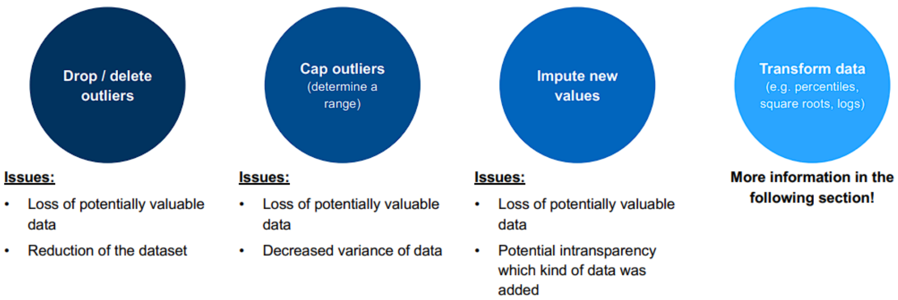
- Outliers can influence the data transformation outcome and thus lead to wrong conclusions in the evaluation step.

There is no precise way to define and identify outliers in general. Instead the raw observations must be interpreted as to whether a value is and outlier or not. Statistical methods can be employed to identify observations.

<u>How to find outliers?</u>

Visualization	• Data can be plotted to gain a quick overview of the distribution of data objects.
Statistical approaches	• Outliers are identified by their probability with regard to a probability distribution model. • Methods: standard deviation method or the interquartile range method (excursus: check Chauvenet's criterion)
Proximity-based detection	• Outliers are identified by their distance from other points. • The k-nearest neighbors are commonly used to measure the distance of objects to each other.
Density-based detection	• Outliers are identified by the density prevailing in the region of an object, the outlier score is the inverse. • Density can be measured by the distance to the k-nearest neighbors or the number of objects within a predefined radius.
Clustering-based detection	• Outliers are identified by assessing their degree of belonging by measuring the distance to cluster centers. • Alternatively, small clusters can be discarded as outliers.

<u>And how to solve outliers? And issues?</u>



Instead of dealing with outliers explicitly, robust algorithms should be chosen wherever feasible. Even more so when working with classification and regression algorithms.

3.4.7 Normalization

Normalization

Description:

- Changes numeric values of different attributes / variables to a common scale, typically setting the range between 0-1

Benefits:

- Helps preventing variables with larger ranges to influence the model more heavily than those with smaller ranges.

Methods:

- Min-Max-Normalization → $x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$
- Unit Vector Normalization
- Z-Normalization

Standardization

Description:

- Assumes that a **Gaussian distribution** is present
- Sets the **mean of the data to 0** and the standard deviation to 1

Benefits:

- Improves the numerical stability of the model and often reduces training time

Methods:

- Z-Normalization (Standardization) $z_i = \frac{x_i - \bar{x}}{s}$

Normalization is used when trying to model relations between attributes / variables. It reduces the bias that might originate from different scales.

3.4.8 Transformation

Methods:

Simple Functions

Description:

- Some functions can be used to reduce skewness and variance.

Methods:

- Logarithm, Square Root, Box Cox

Discretization

Description

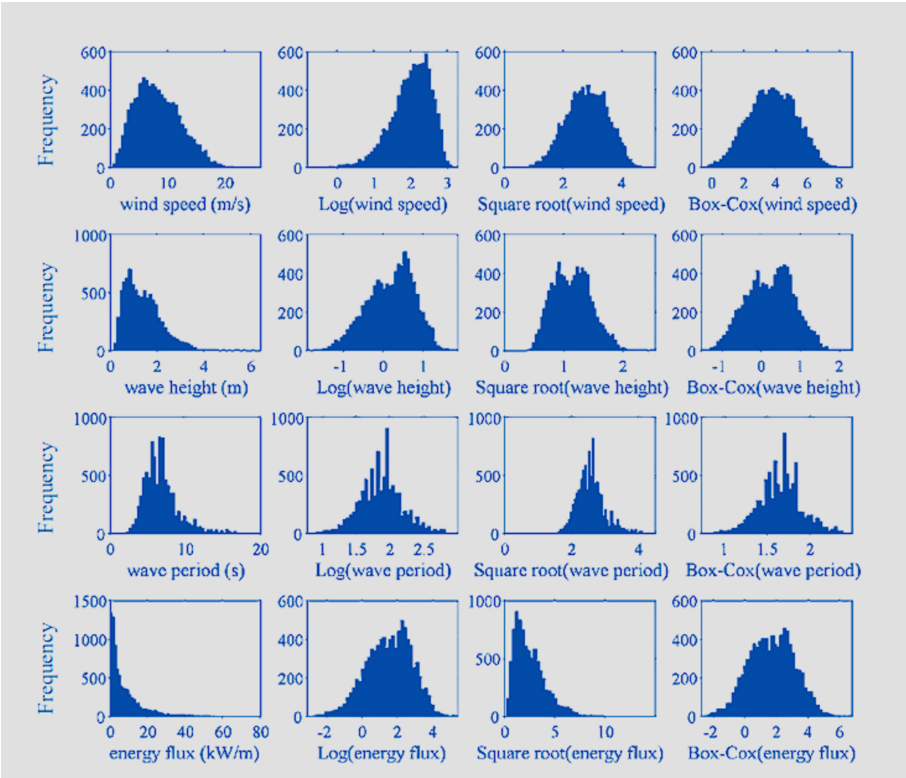
- Process of mapping continuous values to discrete values
- Commonly used for classification

Binarization

Description:

- Maps a continuous or categorial attribute onto one or more binary variables (might increase dimensionality, e.g. one hot encoding)
- Commonly used for association analysis

Attribute / variable transformation maps an entire set of values of a given attribute / variable to a new set of replacement values. It allows to deal with skewness and allows for more performant computing. All methods lead to a loss of information.



3.4.9 Highly Correlated Data

Highly correlated data are unlikely to contribute any further information and very likely to cause overfitting. Thus, a correlation coefficient matrix should be calculated to check for possible correlations between attributes / variables. Overall, domain knowledge helps identifying cases, in which calculation the coefficient is necessary.

Linear relationships:

Person's Correlation Coefficient:

$$r = \frac{cov(x, y)}{\sigma_x \cdot \sigma_y}$$

- Coefficient returns a value between -1 and 1, i.e. a full negative correlation to a full positive correlation

- Value of 0 means no correlation, whereas an absolute value over 0.5 indicates a notable correlation

Non-linear relationships:

Spearman's Correlation Coefficient:

$$r = \frac{cov(rank(x), rank(y))}{\sigma_{rank(x)} \cdot \sigma_{rank(y)}}$$

- Non-Gaussian distribution is no issue (non-parametric correlation)
- Assumes a monotonic relationship
- Rank-based approach quantifies the association between variables using the ordinal relationship between the values rather than the specific values

3.4.10 Dimensionality

Inconsistent dimensionality

<u>Problem</u>

- Machine learning **algorithms require** training and test **data of consistent dimensions**.
- **Data** from time series and real production applications might not fulfill this requirement.

<u>Solution approach</u>

- Methods presented to handle missing values / outliers
- Methods of the data transformation section to unify data dimensionality

High dimensionality

<u>The "Curse of Dimensionality":</u>

- Some data analysis becomes significantly harder
- Data becomes increasingly sparse in the space it occupies
- For classification there are not enough objects to create a model that can predict all classes
- For clustering the density and distance are less meaningful

3.5 Data Warehouse

Definition

A data warehouse (DW) **is a central repository of integrated data optimized for analysis purposes** and combines data from several, usually heterogeneous sources. **Extract-transform-load (ETL) is the main process** to build a data warehouse system.

The decoupling of a central data warehouse from the databases supplying the data leads on the one hand to a relief of the operative systems and on the other hand opens the option to optimize the analysis-oriented system for the needs of evaluations and reports.