

# Business Intelligence

## 01 (Meta-)Introduction

Prof. Dr. Bastian Amberg  
(summer term 2024)

17.4.2024

### Professur für Wirtschaftsinformatik

#### Prof. Dr. Natalia Kliewer

Operations Research & Analytics  
Planungssysteme in Transport und Verkehr  
Robuste Effizienz  
Revenue Management

#### Juniorprofessur für Advanced Decision Analytics

#### Prof. Dr. Bastian Amberg

Entscheidungsunterstützungssysteme  
Robuste Effizienz  
In Dienstleistungsindustrien

#### Assozierte ECDF-Professur

Demnächst wieder



#### Assozierte Professur

#### Assozierte Professur



#### Juniorprofessur für BWL, insb. Digital Entrepreneurship und Diversity

#### Prof. Dr. Janina Sundermeier

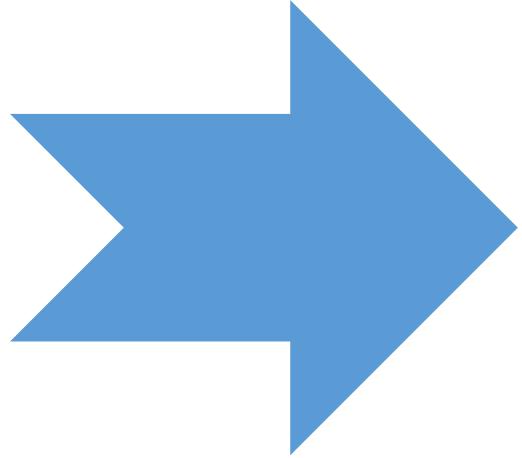
Digital Entrepreneurship  
Unternehmerische Diversität  
Gründungsbezogene Persönlichkeitsmerkmale

### Professur für BWL, Information und Organisation

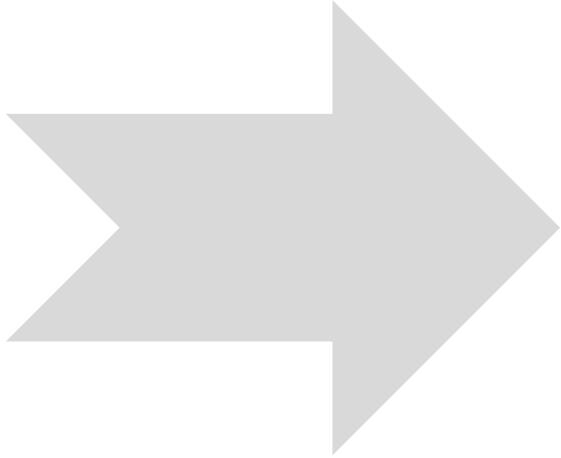
#### Prof. Dr. Martin Gersch

E-Business  
Informationsmanagement  
Service Engineering  
Entrepreneurship Education

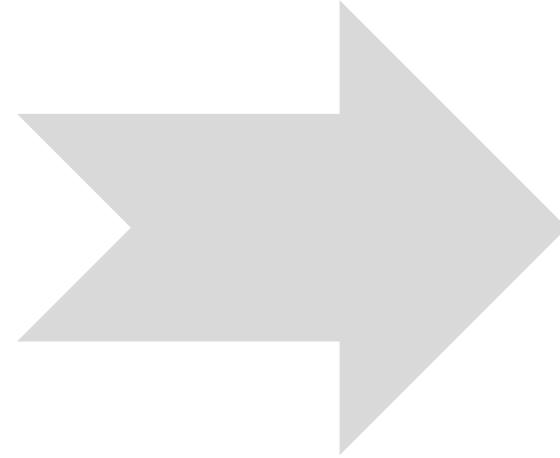
# Agenda



(1)  
Metaintroduction



(2)  
Decision Support and  
Business Intelligence



(3)  
Data-analytic thinking

...zur Pflichtvorlesung im 2. Semester des  
Studiengangs **Master-Wirtschaftsinformatik**

Interaktive Vorlesung/Seminar mit  
Übungseinheiten/Miniprojekten am Rechner

Seminaristischer Unterricht mit **regelmäßiger**  
und **aktiver Teilnahme**: auch Sie produzieren  
Inhalte

Unterrichtssprache: Deutsch;  
Materialien in der Regel auf Englisch

**Modul:** Business Intelligence

**Hochschule/Fachbereich/Institut:** Freie Universität Berlin/FB Wirtschaftswissenschaft/Institut für Wirtschaftsinformatik

**Modulverantwortliche/r:** Dozentinnen und Dozenten des Moduls

**Zugangsvoraussetzungen:** Keine

**Qualifikationsziele:**

Die Studentinnen und Studenten können mithilfe von Methoden der intelligenten Datenanalyse Erkenntnisse aus der Analyse großer und komplexer Datenmengen gewinnen. Sie besitzen die Fähigkeit, Simulationssysteme für die Entscheidungsunterstützung insbesondere unter Unsicherheit im betriebswirtschaftlichen Umfeld zu entwerfen, zu implementieren und einzusetzen. Sie sind in der Lage, die vermittelten Modelle, Methoden und Algorithmen in der den Fragestellungen angemessenen Weise auszuwählen und anzuwenden sowie die Handlungsempfehlungen aus der Methodenanwendung abzuleiten. Das Seminar am PC adressiert insbesondere auch überfachliche Qualifikationsziele, insb. eigenständiges Arbeiten, analytisches Denken, Präsentationsfähigkeiten sowie Fähigkeiten zur technikassisierten Aufgabenlösung in Teams.

**Inhalte:**

Spezielle Modelle und Algorithmen des Datamining, Modelle, Methoden und Grundlagen der Simulation sowie Nutzung einschlägiger Softwarewerkzeuge zur Datenanalyse und Simulation, eine Auswahl aus speziellen Techniken, wie z. B. Clustering, Assoziationsanalyse, Klassifikation, diskrete und ereignisgesteuerte, stochastische, agentenbasierte Simulation etc.

| Lehr- und Lernformen                       | Präsenzstudium (Semesterwochenstunden = SWS) | Formen aktiver Teilnahme  | Arbeitsaufwand (Stunden)   |
|--|--|---|--|
| Seminaristischer Unterricht                | 1  | Unterrichtsgespräch, Beantwortung von Diskussionsfragen, Diskussion von Anwendungsproblemen | Präsenzzeit Seminaristischer Unterricht 15<br>Vor- und Nachbereitung Seminaristischer Unterricht 30            |
| Seminar am PC                              | 2  | Kurvvorträge mit Diskussion, Diskussion von Literatur und Anwendungsbeispielen              | Präsenzzeit Seminar am PC 30<br>Vor- und Nachbereitung Seminar am PC 30<br>Prüfungsvorbereitung und Prüfung 75 |
| <b>Veranstaltungssprache:</b>              |  |   | Deutsch  |
| <b>Pflicht zur regelmäßigen Teilnahme:</b> |  |   | Ja   |
| <b>Arbeitszeitaufwand insgesamt:</b>       |  |   | 180 Stunden   6 LP   |

Umfang der Veranstaltung: 6 LP/ **3 SWS** Dies beinhaltet:

Mittwoch, **10-12 Uhr**

→ interaktive Vorlesungen

Freitag, ~~12-14 Uhr~~ **14-16 Uhr**

→ digitale Übungen, Zeit für Vor- und Nachbereitung der Vorlesungen

Terminplan und Ankündigungen im **Blackboard** beachten!

Kommunikation und Materialienbereitstellung über Blackboard

*Kurs Business Intelligence (SoSe 2024) (WIWISS\_S\_10180206\_24S)*

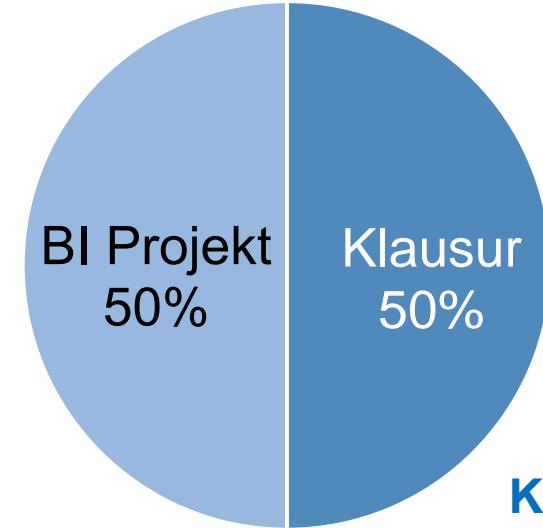
Unklarheiten direkt im Kontext, am Ende der Veranstaltung oder zu Beginn der nächsten Veranstaltung klären,  
ansonsten: [bastian.amberg@fu-berlin.de](mailto:bastian.amberg@fu-berlin.de)

Oder Blackboard-Forum „Organisatorische Fragen“ / „Inhaltliche Fragen“

Prüfungsleistung besteht aus **Klausur** und **Mini-Projekt**

# Prüfungsleistung

## Zusammensetzung



### BI-Projekt

macht 50% der Note aus.

Bearbeitungsdauer ca. 6 Wochen, semesterbegleitend

### Klausur

macht 50% der Note aus.

Dauer 60 Minuten, im Klausurenzeitraum

(1.Termin im Zeitraum 22.7. bis 3.8.)

(2.Termin im Zeitraum 23.9. bis

In der Veranstaltung lernen Sie **Methoden und Werkzeuge** der Business Intelligence kennen.

Sie finden sich in **4er-Gruppen** zusammen und bearbeiten einen Datensatz auf relevante Fragestellungen.

*Sie wählen sich die dafür geeigneten Methoden und Werkzeuge selbstständig aus.*

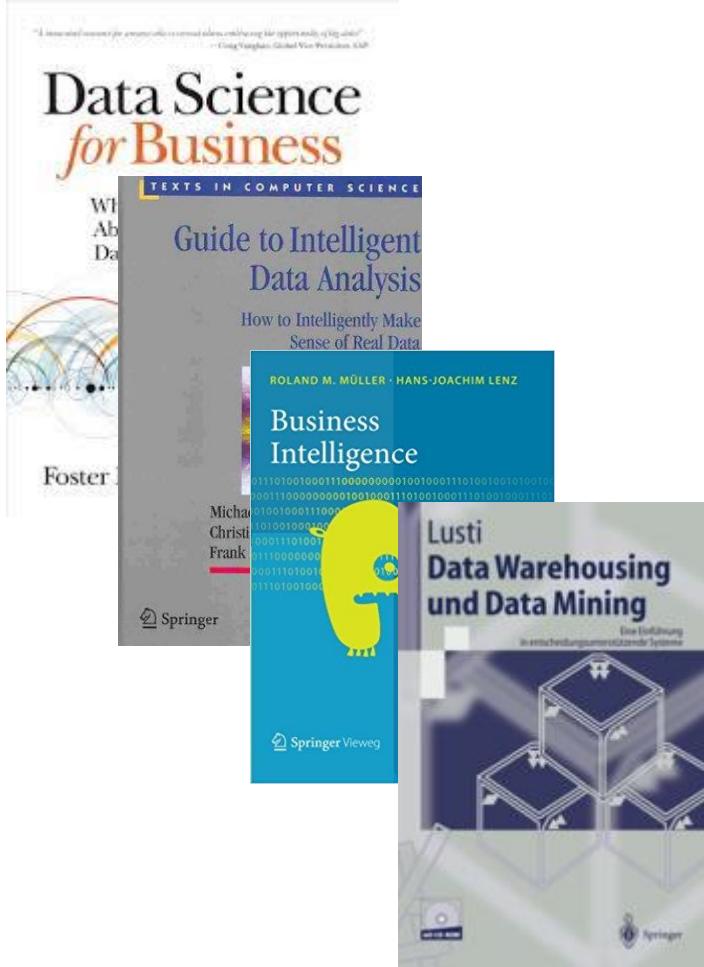
Ca. **20-minütige Präsentation** der Fragestellung, des Vorgehens und Diskussion der Ergebnisse und Erstellen einen kurzen **Projektberichts**. (genaue Vorgaben hierzu später)

Ref.

Einzelleistung

Verständnisfragen und Anwendung

(Beispielaufgaben im späteren Veranstaltungsverlauf)



Provost, F.; Fawcett, T.: Data Science for Business. Fundamental Principles of Data Mining and Data-Analytic Thinking. O'Reilly, 2013

Berthold, M. R.; Borgelt, C.; Höppner, F.; Klawonn, F.: Guide to Intelligent Data Analysis. Springer, 2011

Müller, R. M.; Lenz, H.-J.: Business Intelligence. Springer, 2014

Lusti, M.: Data Warehousing und Data Mining, Springer, 2002

Additional readings: **see Bibliography** at the end of each presentation

# Software

im Verlauf des Kurses

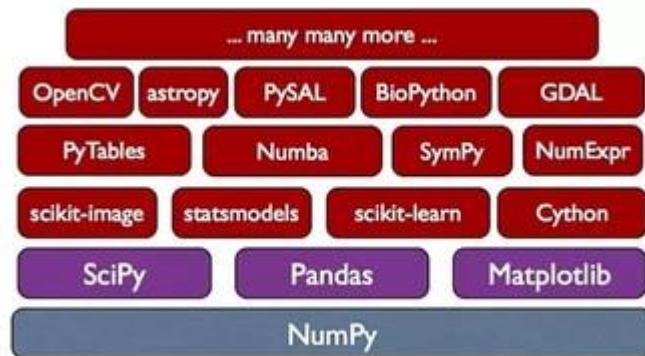


## Python Packages

- Let's agree on commonly using Python version  $\geq 3.7$   
<https://www.python.org/downloads/> (latest version is 3.12.3)
- Install Anaconda (<https://www.anaconda.com/distribution/>) (or the according packages SciPy, Pandas, NumPy, scikit-learn, Matplotlib in your own environment)

Deep Learning:

Tensorflow (<https://www.tensorflow.org/>) with conda ([install](#))



## Konstanz Information Miner (KNIME)\*

- <http://www.knime.org/>



## Weka Data Mining Software\*

- <http://www.cs.waikato.ac.nz/ml/weka/>



\*For individual needs

Ref.

# Overall goals of this class

We have three goals. After this course:

- You know how to solve business problems by **data-analytic thinking**
- You have an overview about principles of how to model and solve upcoming **business problems**.
- You know several **tools** and ways of how to practically **implement** solution methods

## Data Warehousing / Data Engineering

- How to **store and access** huge amounts of data?

## Data Mining / Data Science

- How to **derive knowledge and profitable business action** out of large databases?

# Schedule

|           | Wed., 10:00-12:00 |       | Fr., 14:00-16:00 (Start at 14:30)                   |       | Self-study  |                          |
|-----------|-------------------|-------|---|-------|---|--------------------------|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 | 19.4. |   | Python-Basics Chap. 1    |
|           | W2                | 24.4. | Data Warehouse – Overview & OLAP                    | 26.4. | [Blockveranstaltung SE Prof. Gersch]  | Chap. 2                  |
|           | W3                | 1.5.  |   | 3.5.  | Data Warehouse Modeling I   | Chap. 3                  |
|           | W4                | 8.5.  | Data Warehouse Modeling II                          | 10.5. | Data Mining Introduction  |                          |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     | 17.5. | Python-Basics-Online Exercise   | Python-Analytics Chap. 1 |
|           | W6                | 22.5. | Data Understanding, Data Visualization              | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam | Chap. 2                  |
|           | W7                | 29.5. | Data Preparation                                    | 31.5. |   |                          |
|           | W8                | 5.6.  | Predictive Modeling I                               | 7.6.  | Predictive Modeling II (10:00 -12:00)   | BI-Project Start         |
|           | W9                | 12.6. | Fitting a Model I                                   | 14.6. | Python-Analytics-Online Exercise  |                          |
|           | W10               | 19.6. | Guest Lecture                                       | 21.6. | Fitting a Model II  |                          |
|           | W11               | 26.6. | How to avoid overfitting                            | 28.6. | What is a good Model?   |                          |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I                                  |                          |
|           | W13               | 10.7. |   | 12.7. | From Machine to Deep Learning II  |                          |
|           | W14               | 17.7. | Project presentation                                | 19.7. | Project presentation  | End                      |
| Ref.      |                   |       |   |       | Klausur 1.Termin ~ 22.7. bis 3.8.<br>Klausur 2.Termin ~ 23.9. bis 5.10.                       | Projektbericht           |

# Informations on our digital Python exercises

## A) Exercises for self-study - based on Jupyter notebooks:

Notebooks can be found in Blackboard:

“Kursmaterialien > Readings & Übungen > Python-Übungen > Jupyter Notebooks“

Chapters are unlocked at

### Python-Basics:

|   |                          |
|---|--------------------------|
| 17.04.: Chapter 1 – Erste Schritte              | <i>solutions: 24.04.</i> |
| 24.04.: Chapter 2 – Strings & String-Funktionen | <i>solutions: 02.05.</i> |
| 02.05.: Chapter 3 – Bedingungen                 | <i>solutions: 08.05.</i> |

### Python-Analytics:

|  |                          |
|--|--------------------------|
| 15.05.: Chapter 1 – Spezielle Datentypen | <i>solutions: 22.05.</i> |
| 22.05.: Chapter 2 – Datenanalyse         | <i>solutions: 29.05.</i> |

## B) Exercises via Webex:

### Python-Basics

~ 17.5.

### Python-Analytics

~ 14.6.

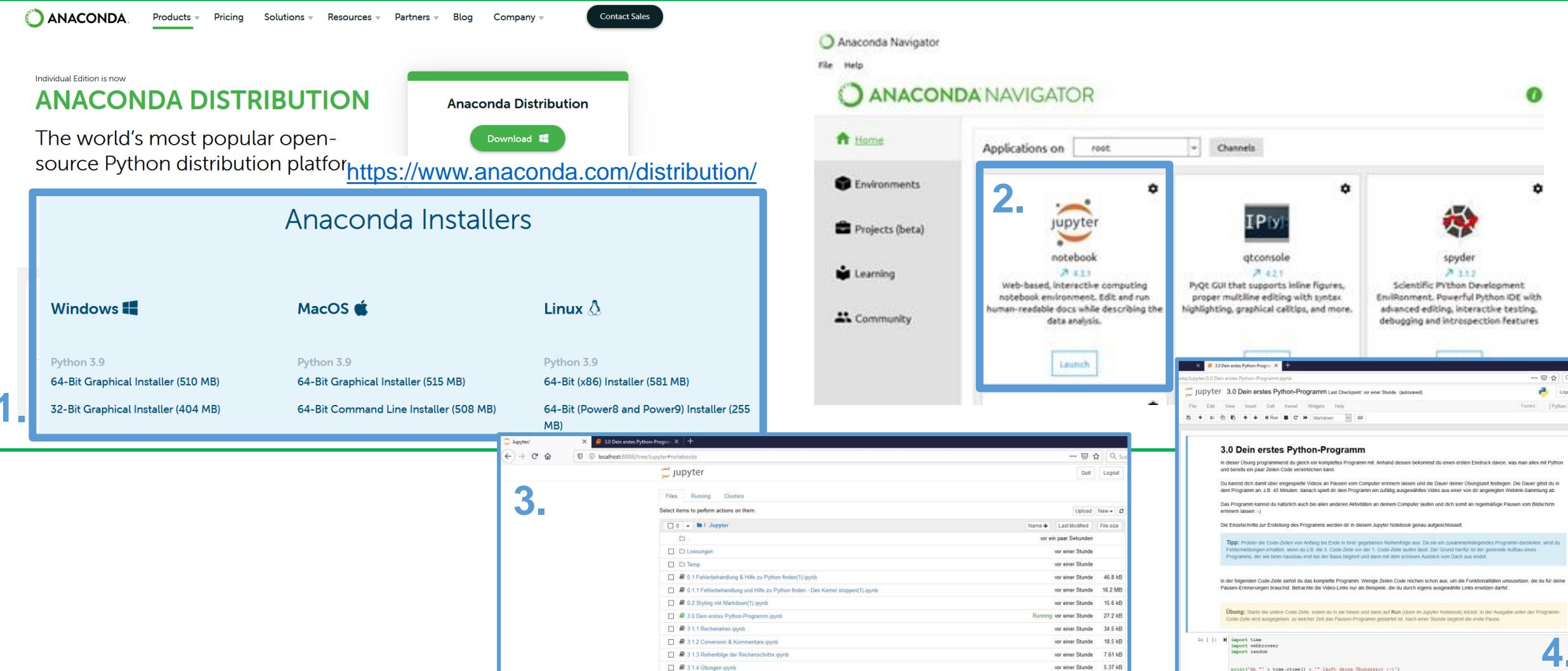
The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.



<https://jupyter.org/>

ReQuestions during/after exercises: [bastian.amberg@fu-berlin.de](mailto:bastian.amberg@fu-berlin.de)

# Getting Started with Jupyter-notebooks



1.

Individual Edition is now  
**ANACONDA DISTRIBUTION**

The world's most popular open-source Python distribution platform <https://www.anaconda.com/distribution/>

2.

Anaconda Installers

Windows

Python 3.9  
64-Bit Graphical Installer (510 MB)  
32-Bit Graphical Installer (404 MB)

MacOS

Python 3.9  
64-Bit Graphical Installer (515 MB)

Linux

Python 3.9  
64-Bit (x86) Installer (581 MB)  
64-Bit (Power8 and Power9) Installer (255 MB)

3.

4.

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Home Applications on Root Channels

Environments Projects (beta) Learning Community

jupyter notebook

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch

IPy qtconsole spyder

Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Jupyter

3.0 Dein erstes Python-Programm

In dieser Übung programmierst du gleich ein komplettes Programm mit. Anhand dessen bekommst du einen ersten Eindruck davon, was man alles mit Python und benutzt ein paar Zeilen Code verkehren kann.

Du kennst dich damit über eingespilte Videos an Pausen vom Computer erinnern lassen und die Dauer deiner Übungssitzung festlegen. Die Dauer gibt ab in Minuten am z.B. 45 Minuten. Danach spielt dir dein Programm ein zufälliges Video aus einer von dir angelegten Webank-Sammlung ab.

Das Programm kann natürlich auch bei allen anderen Aktivitäten an deinem Computer laufen und dich somit an regelmäßige Pausen vom Bildschirm erinnern lassen.:-)

Die Einzelschritte zur Erstellung des Programms werden dir in diesem Jupyter Notebook genau aufgeschlossen.

Tipp: Probier die Code-Zeilen von Anfang bis Ende in ihrer gegebenen Reihenfolge aus. Da sie zusammenhängendes Programm darstellen, wird du Fehlerbehandlungen erhalten, wenn du z.B. die 3. Code-Zeile vor der 1. Code-Zeile laufen lässt. Der Grund hierfür ist der generelle Aufbau eines Programms, der beim Hausaufgabe erst bei der Basis beginnt und dann mit dem schicken Ausblick vom Dach aus endet.

In den folgenden Code-Zeile siehst du das komplette Programm. Wenige Zeilen Code reichen schon aus, um die Funktionalitäten umzusetzen, die du für deine Pausen-Erinnerungen brauchst. Betrachte die Video-Links nur als Beispiele, die du durch eigens ausgewählte Links ersetzen darfst.

Übung: Starte die untere Code-Zelle, indem du in sie hinen und dann auf Run (oben im Jupyter Notebook) klickst. In der Ausgabe unter der Programm-Zelle wird ausgegeben, zu welcher Zeit das Pausen-Programm gestartet ist. Nach einer Stunde beginnt die erste Pause.

```
In [1]: import time
import webbrowser
import random

print("Ab " + time.ctime() + " läuft deine Übungssitzung :-)"
```

**Alternative zum schnellen Testen ohne Installation:** Sind die Notebooks einmal lokal gespeichert, können sie z.B. über CoCalc (*Collaborative Calculation and Data Science Service*) geöffnet, kopiert, umbenannt, ausgeführt und bearbeitet werden. Es ist – Stand April 2024 – allerdings die Erstellung eines (kostenlosen) Accounts notwendig. Link zu CoCalc: <https://cocalc.com/> bzw. <https://cocalc.com/features/jupyter-notebook>. Nach Login kann ein Notebook oder mehrere hochgeladen werden (unter „File > Upload... > Upload“ oder unter „File > Open... > Upload“) und es kann zum Editieren kopiert, umbenannt und bearbeitet werden. Das bearbeitete Notebook kann gespeichert und anschließend heruntergeladen werden, um es lokal zu sichern.

# What are your expectations?



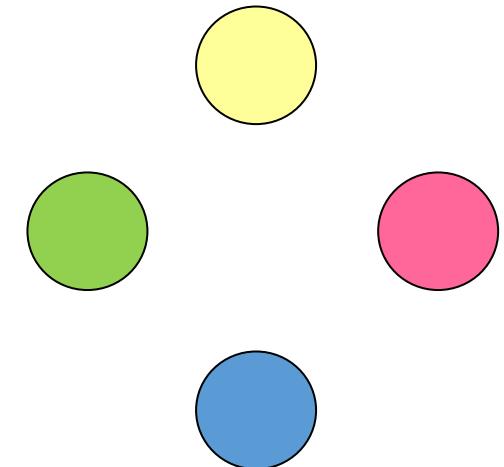
?

?

# What is your background?

!

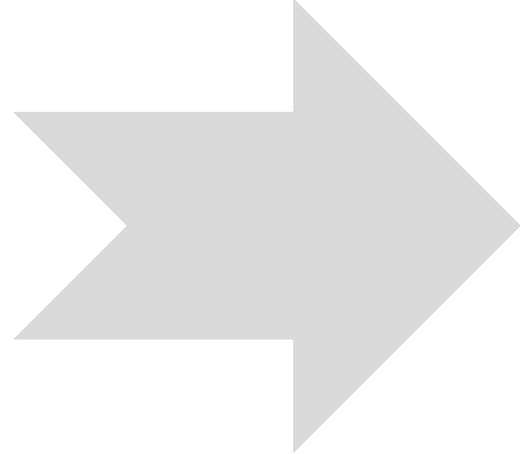
!



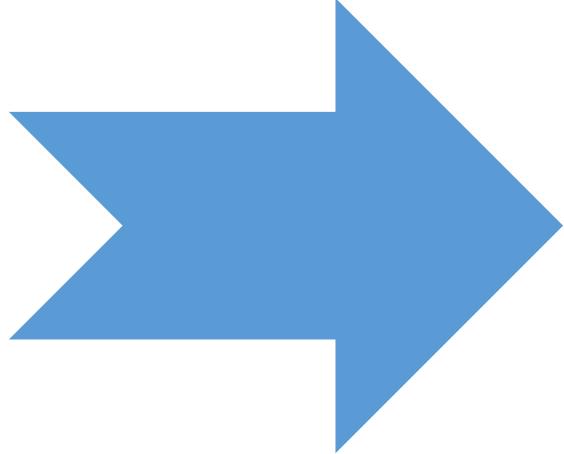
Kahoot  
[www.kahoot.it](http://www.kahoot.it)  
(über Smartphone  
oder Laptop)  
PIN folgt

# Agenda

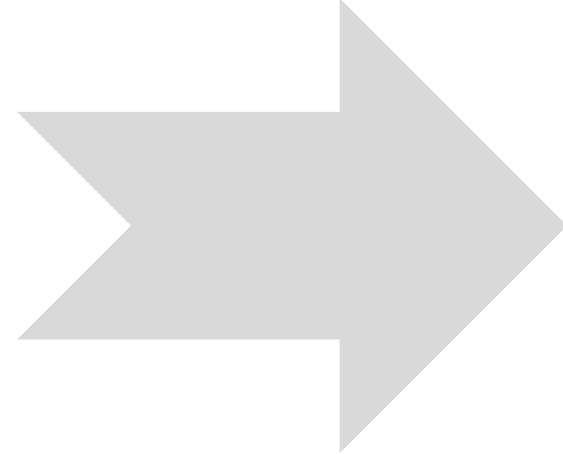
Let's put BI into perspective



(1)  
Metaintroduction



(2)  
Decision Support and  
Business Intelligence



(3)  
Data-analytic thinking

# Decision Support Systems

And variants of problem modelling

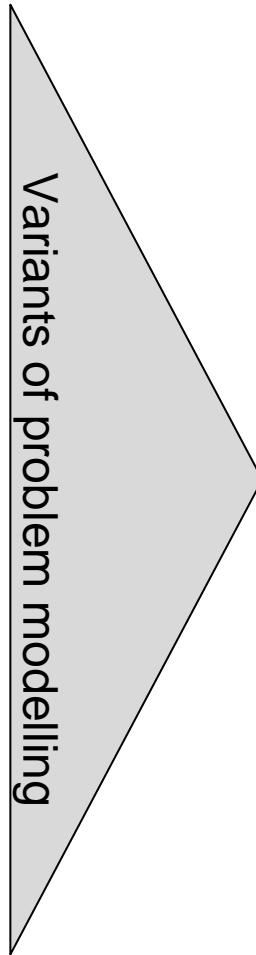
**Decision Support Systems** in the broadest sense can be defined as

*Computer technology solutions that can be used to support complex decision making and problem solving.*

(Shim et al. 2002)

Broad definition that encompasses many areas

- Application systems
- Mathematical modeling
- Data driven modeling



Common **mathematical modelling**

- All the relevant variables and relations can be identified
- The nature of the problem can adequately be caught by mathematical models
- Optimal solution for the underlying decision problem can be derived from the model (e.g., linear programming)

**Data driven (empirical) modelling**

- Problem too complex to identify all the relevant variables and relationships
- Gain insight into problem structure by analysis of historical (transactional) data (e.g., data mining)
- Entails trial-and-error experiments and oftentimes black-boxing

# Business Intelligence: Definition

There is no unique or mathematical definition of Business Intelligence.

The Data Warehousing Institute defines Business Intelligence as...

*... the process, technologies and tools needed to turn data into information, information into knowledge and knowledge into plans that drive profitable business action. Business intelligence encompasses data warehousing, business analytics tools, and content/knowledge management.*

(<http://www.tdwi.org/>)



Ref.

Another (similar) definition:

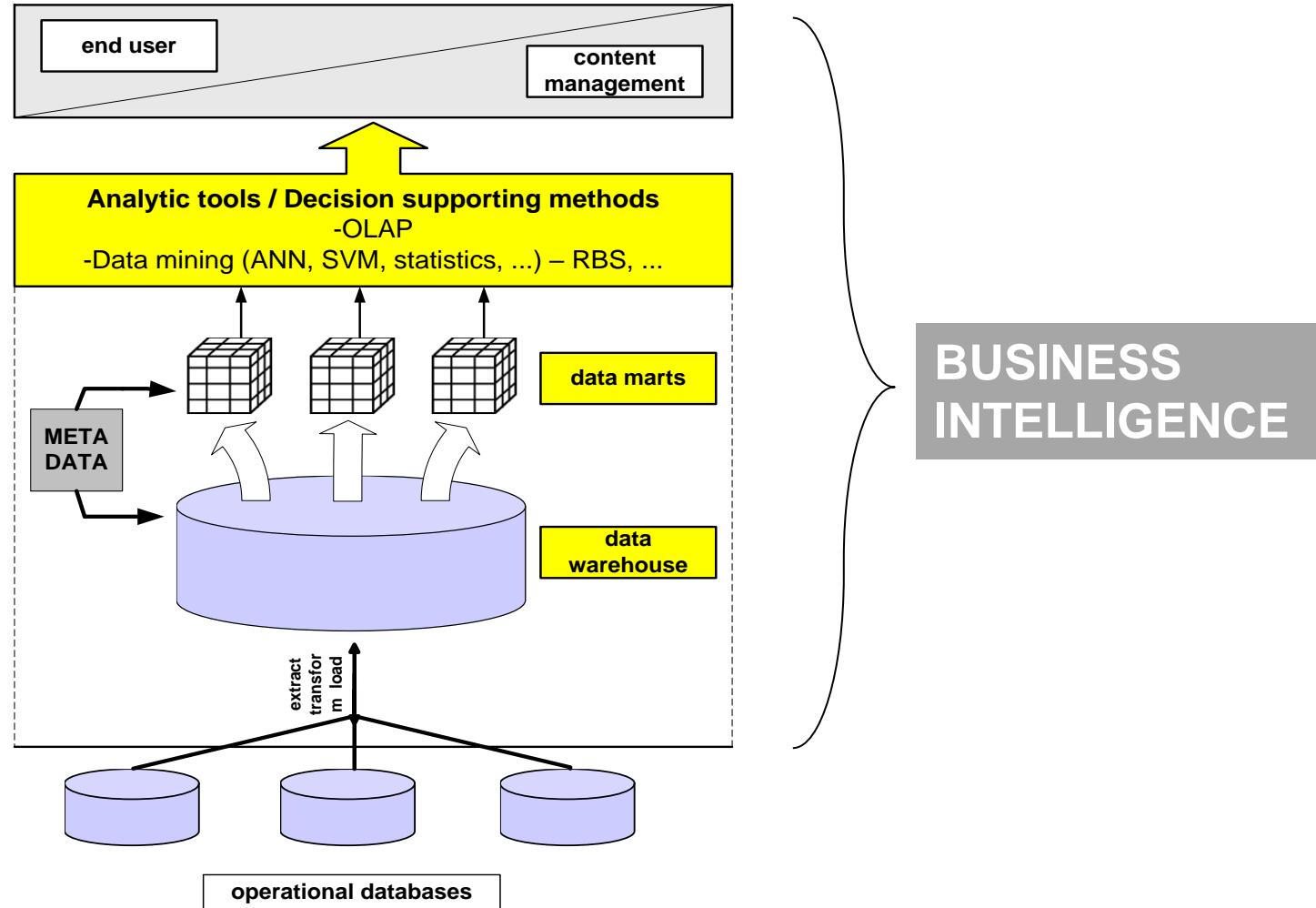
*Business intelligence is the conscious, methodical transformation of data from any and all sources into new forms to provide information that is business driven and results oriented. It will often encompass a mixture of tools, databases and vendors.*

(Mike Biere, specialist for business intelligence analytic tools, IBM)



# Other view on Business Intelligence

A holistic view on BI



## Increased profitability

Distinguish between profitable and non-profitable customers

## Decreased costs

Lower operational costs, improve logistics management

Business Intelligence can help improve businesses in a variety of fields:

Customer analysis  
→ customer profiling

Sales channel analysis

Behavior analysis  
→ fraud detection, shopping trends, web activity, social network analysis

## Improved Customer-Relationship-Management

Analysis of aggregated customer information to provide better customer service, increase customer loyalty



## Decreased risk

Apply Business Intelligence methods to credit data can improve credit risk estimation

Business productivity analysis  
→ defect analysis, capacity planning and optimization, risk management, increase sustainability

Supply chain analysis  
→ supply and vendor management, shipping, distribution analysis, sustainable supplier management



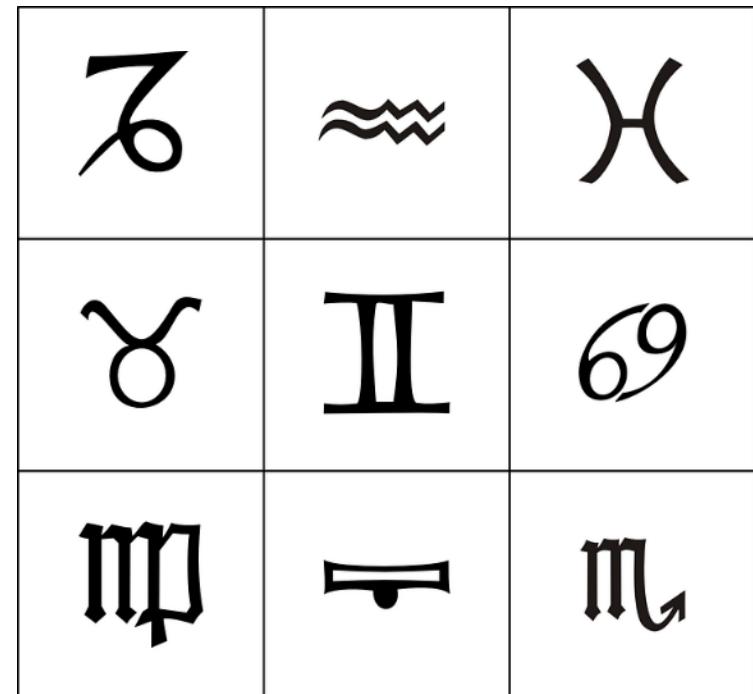
Business Intelligence supports decision makers with the required information at the right time and location, and with sufficient quality (format, visualization, validity etc.)

# Discerning „Data“ from „Information“

## Excursus: Semiotics

Semiotics is a field of research in Epistemology („meaning making“) and studies signs (icons, characters, symbols) and sign systems.

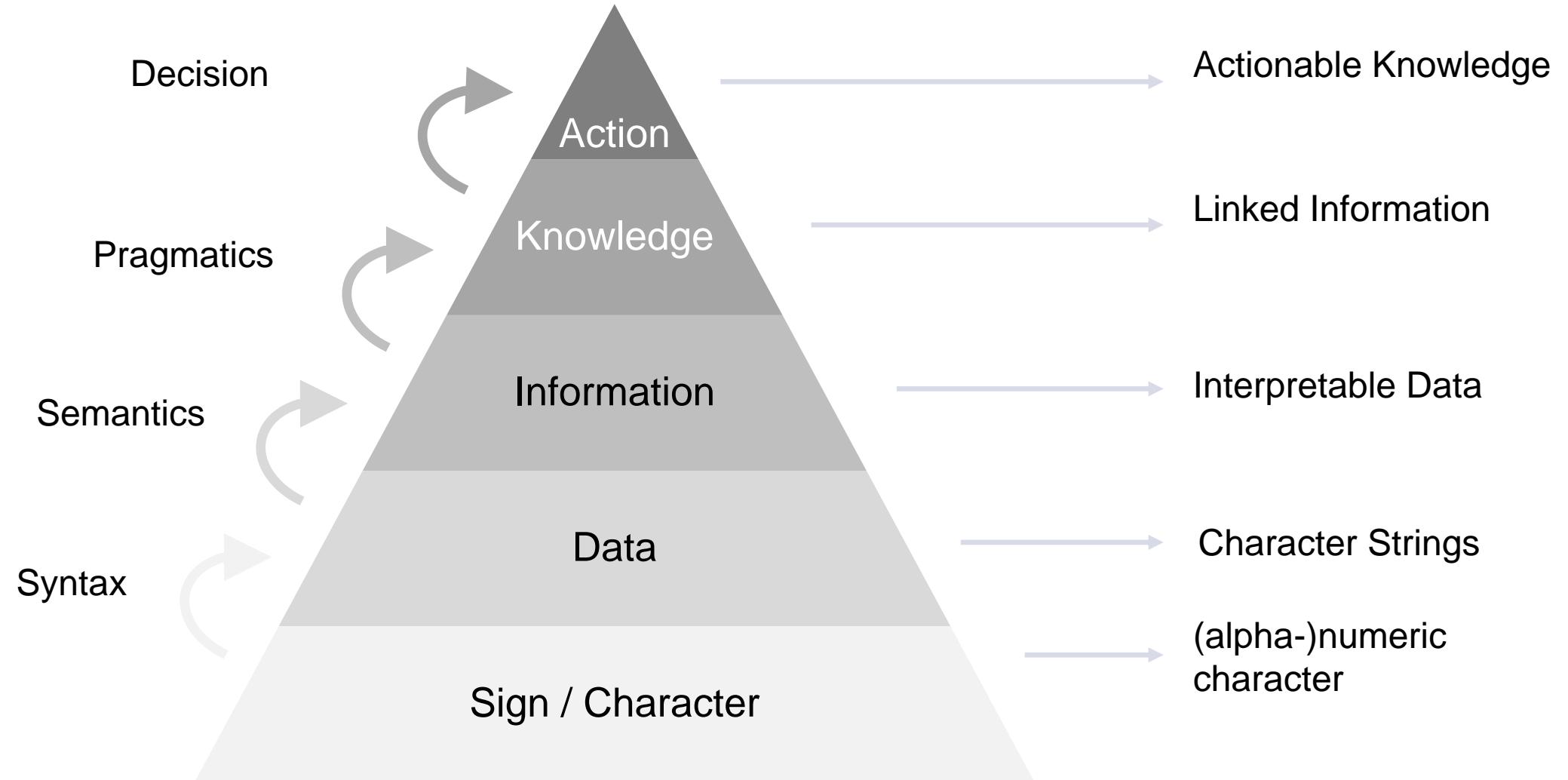
| Layer of observation | Object of observation   | Example  |
|----------------------|---|--|
| Syntax               | <i>“Relations among signs in formal structures”</i>                                     | Is the communicated sign (orthographically/ grammatically) correct?                |
| Semantics            | <i>“Relation between signs and the things to which they refer; their [...] meaning”</i> | What is the meaning of the sign?<br>What is the meaning of „display“ or „windows“? |
| Pragmatics           | <i>“Relation between signs and the effects they have on the people who use them”</i>    | What is the use of this sign?  |



# Data transformation (1/2)



## Knowledge Pyramid



# Data transformation (2/2)

Transforming Data into information; into knowledge; into action

## From data to information

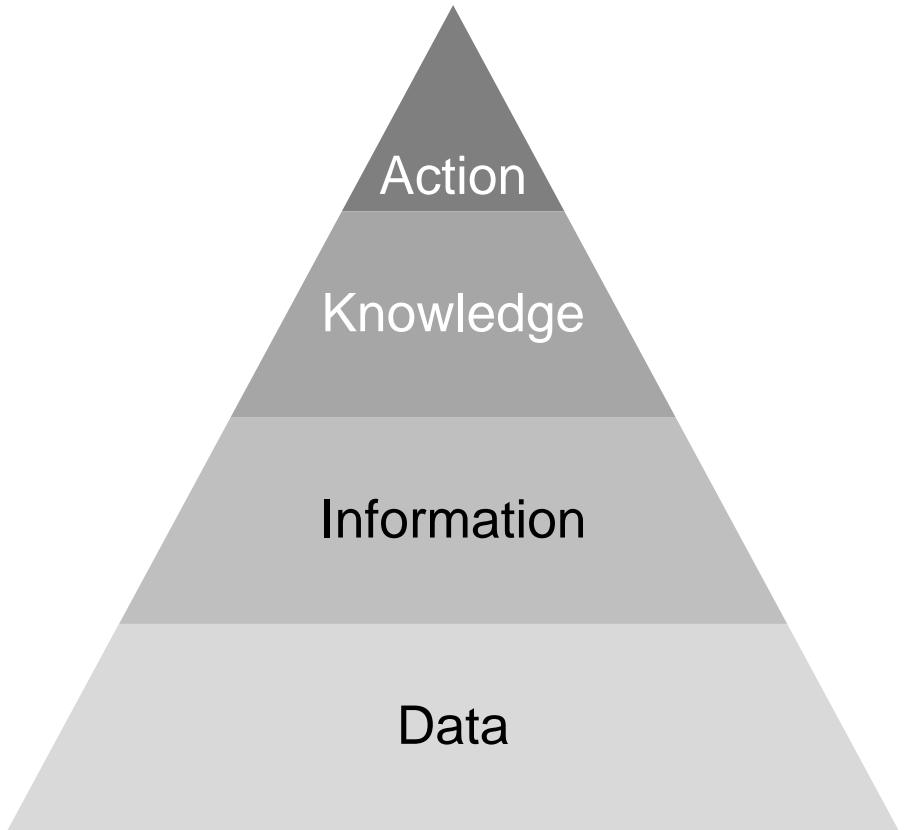
- **Select relevant data** and figure out which configuration makes the data more significant

## From information to knowledge

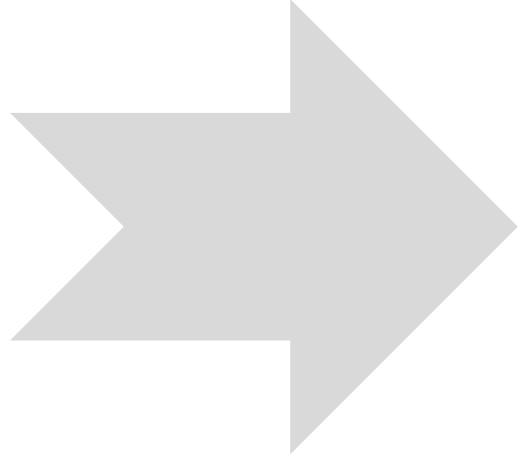
- Piles of information are accumulated and analyzed in different ways
- In this process **analytical tools** are involved

## From knowledge to actionable (critical) knowledge

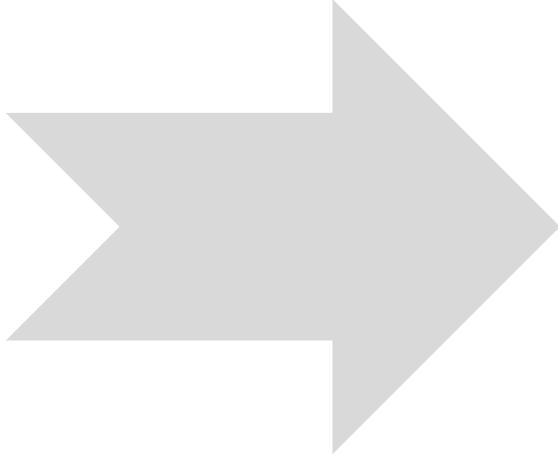
- Knowledge is considered to be critical if it can be used to form a plan of action for **solution of a business problem**



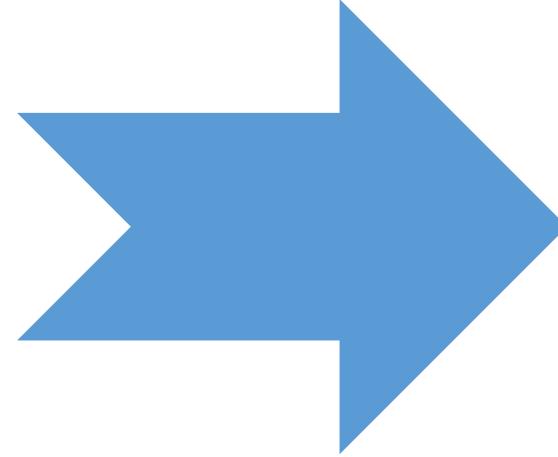
# Agenda



(1)  
Metaintroduction



(2)  
Decision Support and  
Business Intelligence



(3)  
Data-analytic thinking

# Data Driven Decision-making (DDD)

Legitimizing decisions on the basis of data

Data-driven Decision Making (DDD) is a “*practice of basing decisions on the analysis of data rather than purely on intuition.*”

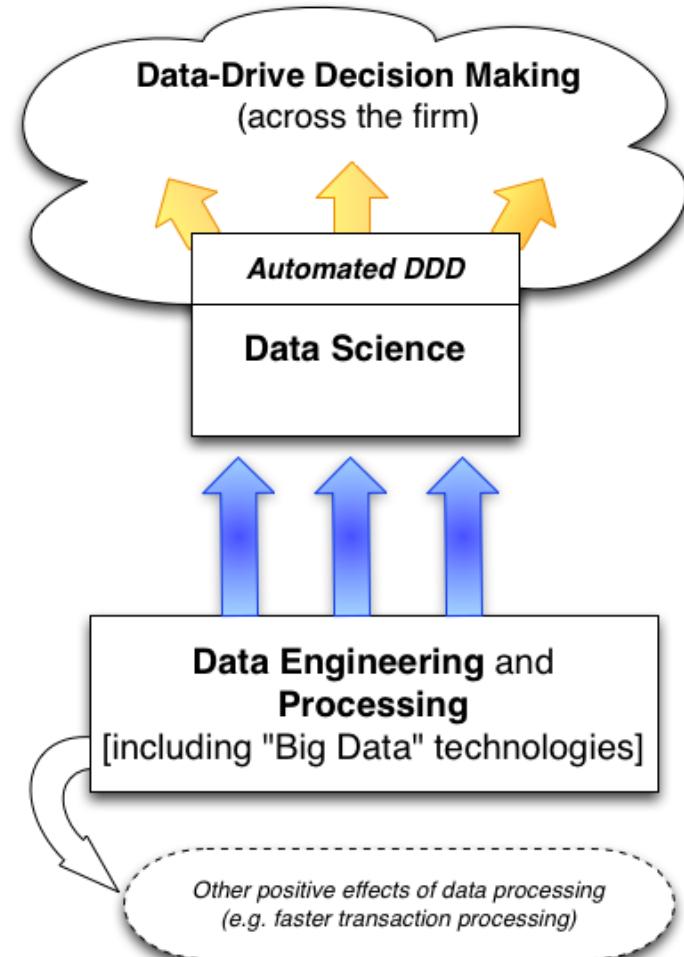
Provost & Fawcett (2013)

## Type-1 decision:

“discover” something new within your data

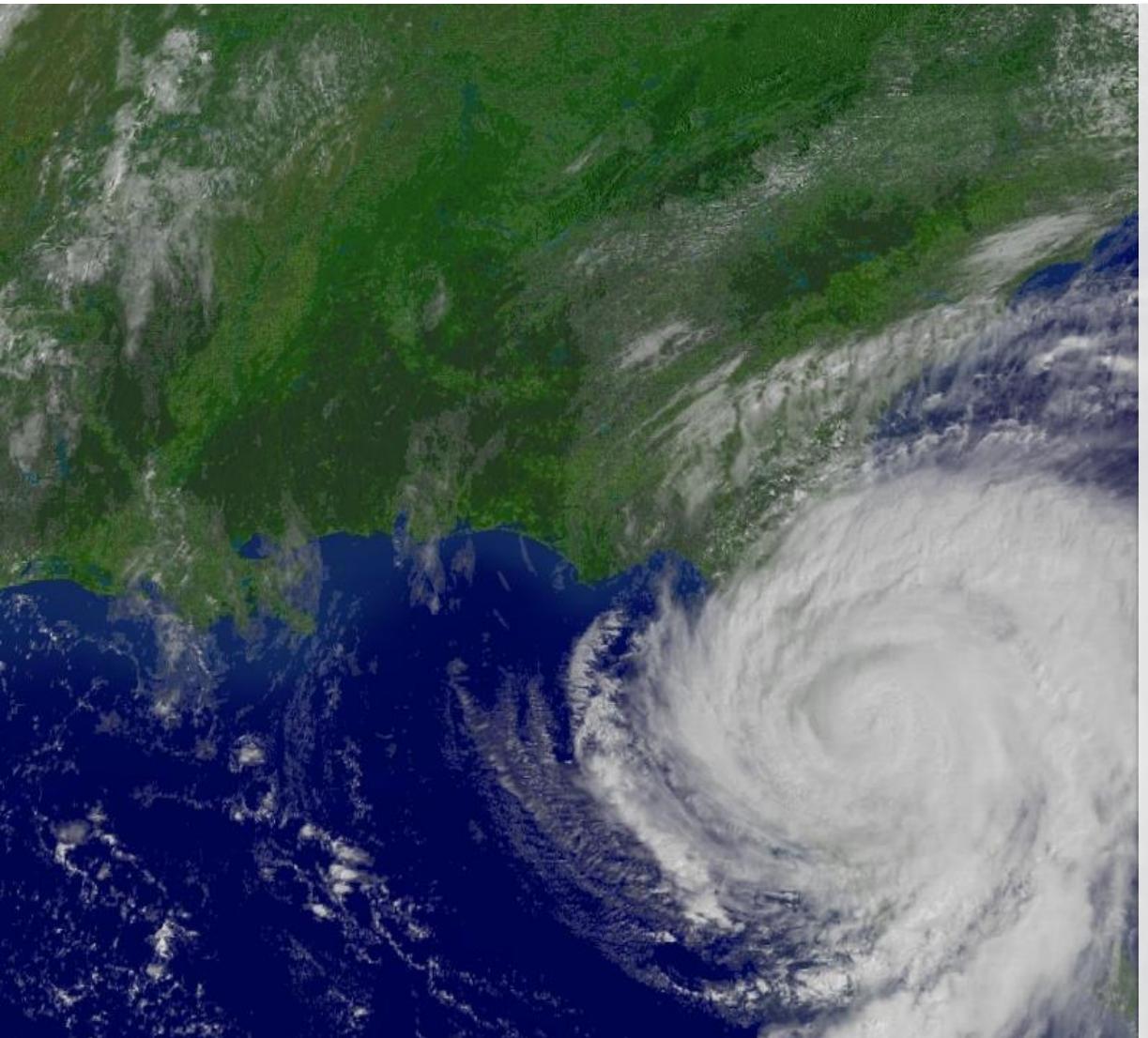
## Type-2 decision:

repeat decisions at massive scale  
(automatic decision making)



# Type 1 decision

Discover something new: Hurricane Frances (1/2)



*Hurricane Frances was on its way, barreling across the Caribbean, threatening a direct hit on Florida's Atlantic coast. Residents made for higher ground, but far away, in Bentonville, Ark., executives at Wal-Mart Stores decided that the situation offered a great opportunity for one of their newest data-driven weapons ... predictive technology.*

*A week ahead of the storm's landfall, Linda M. Dillman, Wal-Mart's chief information officer, pressed her staff to come up with forecasts based on what had happened when Hurricane Charley struck several weeks earlier. Backed by the trillions of bytes' worth of shopper history that is stored in Wal-Mart's data warehouse, she felt that the company could 'start predicting what's going to happen, instead of waiting for it to happen,' as she put it.*  
*(Hays, New York Times, 2004)*

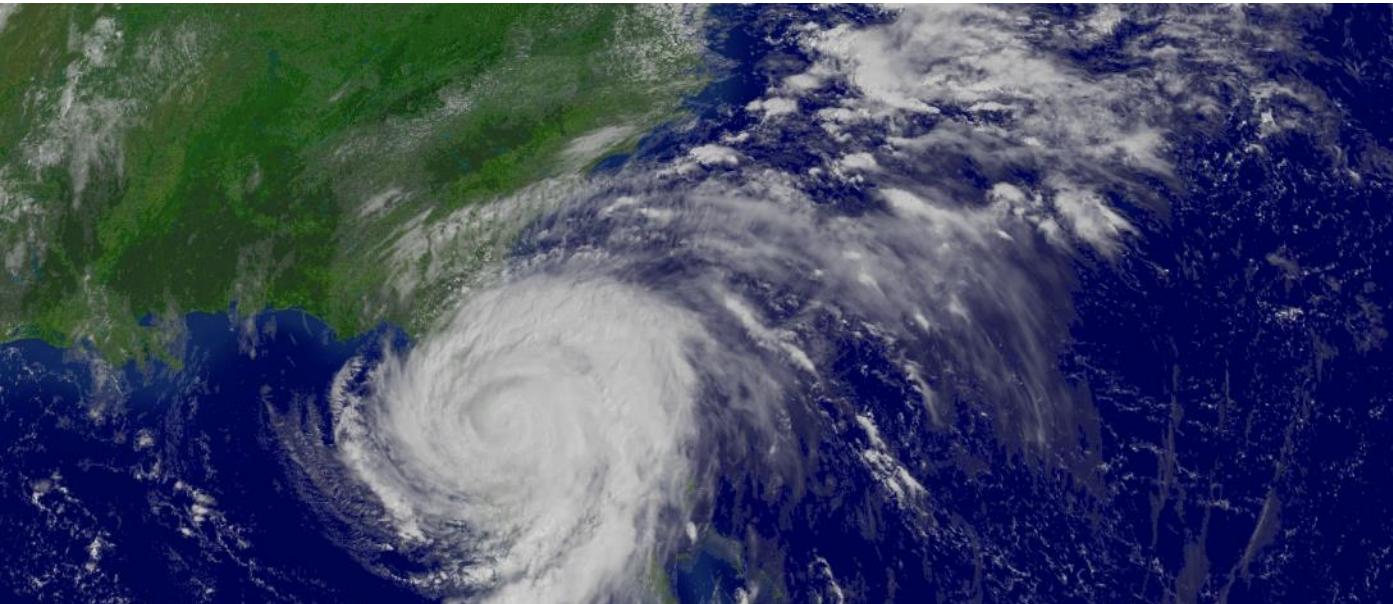
# Type 1 decision

Discover something new: Hurricane Frances (2/2)

Why might data-driven prediction be useful?

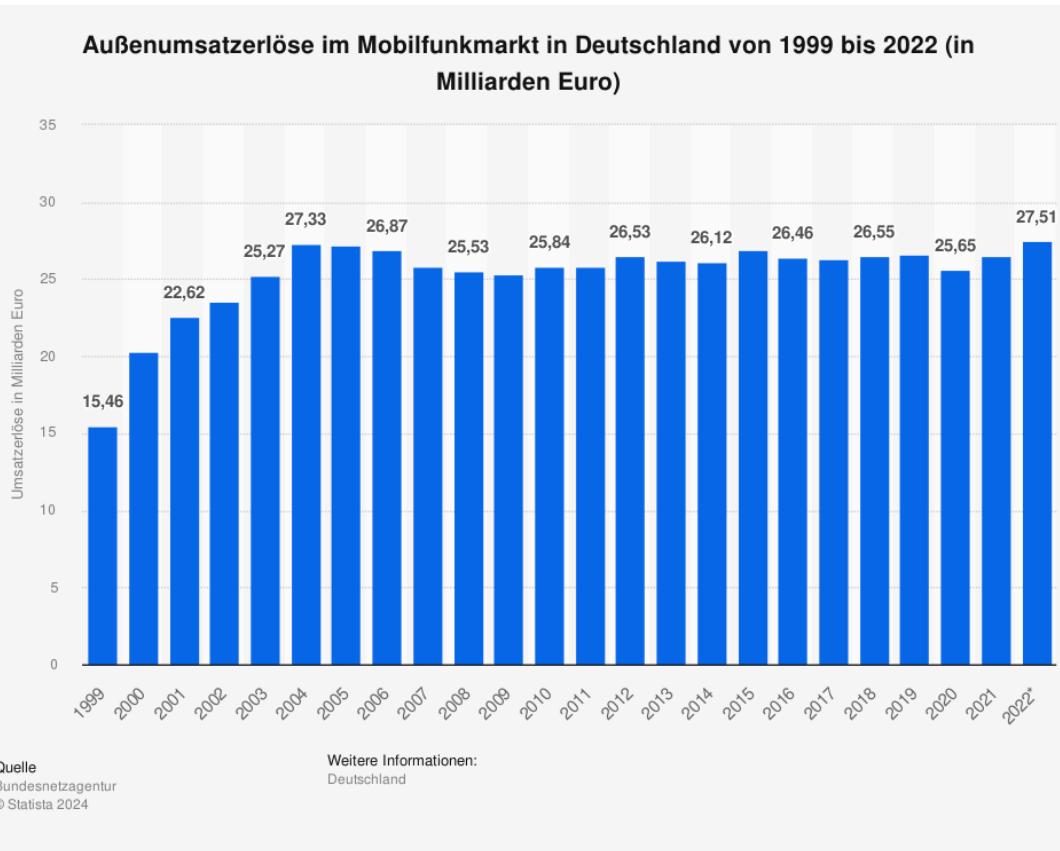
Analytic thinking: discover sales patterns due to the hurricane that are not obvious!

Identify unusual local demand for products (not through hypothesis testing, but data exploration)



# Type 2 decision

Automatic decision making: Predicting Customer Churn



Mobile TeleCom market is highly saturated.



Many TeleCom companies have great issues with customer retention.

(current churn rates between 1 and 3 % for Deutsche Telekom mobile)

Customer churn is expensive.

Who should get a retention offer?

Raising accuracy of a prediction has huge effects on profitability



Ref. [Churn rates from Statista](#)

# Knowing about DDD, how do we proceed?

We follow the path, desribed by semiotics

Data

Information

# Data processing and “Big Data”

Where's the challenge?



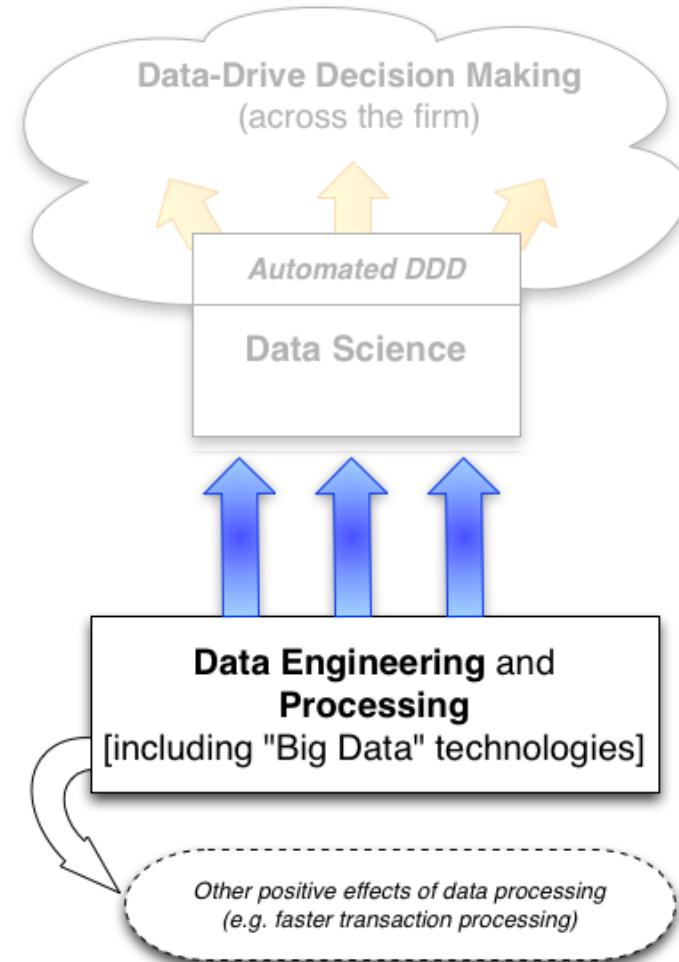
**Data engineering** and data processing are critical to support data science  
But: data engineering is not data science

Data engineering handles for instance Data Warehouses, which

- collect data from operational databases
- accumulate **historical data**
- provide the basis for Business Intelligence applications

*Where is the catch?*

“**Big data**” means data sets that are too large for traditional data processing systems  
Using big data technologies is associated with additional **productivity growth**



## Data science:

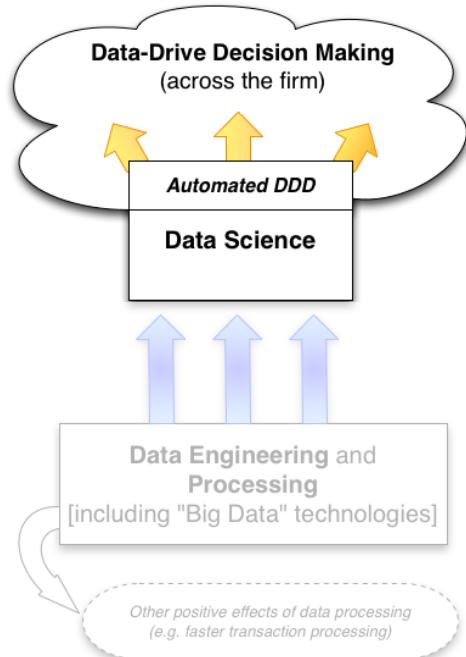
A set of *fundamental principles* that guide the extraction of knowledge from data

*"Data science is an interdisciplinary field aiming to turn data into real value [...] Value may be provided in the form of predictions, automated decisions, models learned from data, or any type of data visualization delivering insights. Data science includes data extraction, data preparation, data exploration, data transformation, storage and retrieval, computing infrastructures, various types of mining and learning, presentation of explanations and predictions, and the exploitation of results taking into account ethical, social, legal, and business aspects"*

(Van der Aalst [2016](#))

## Data mining:

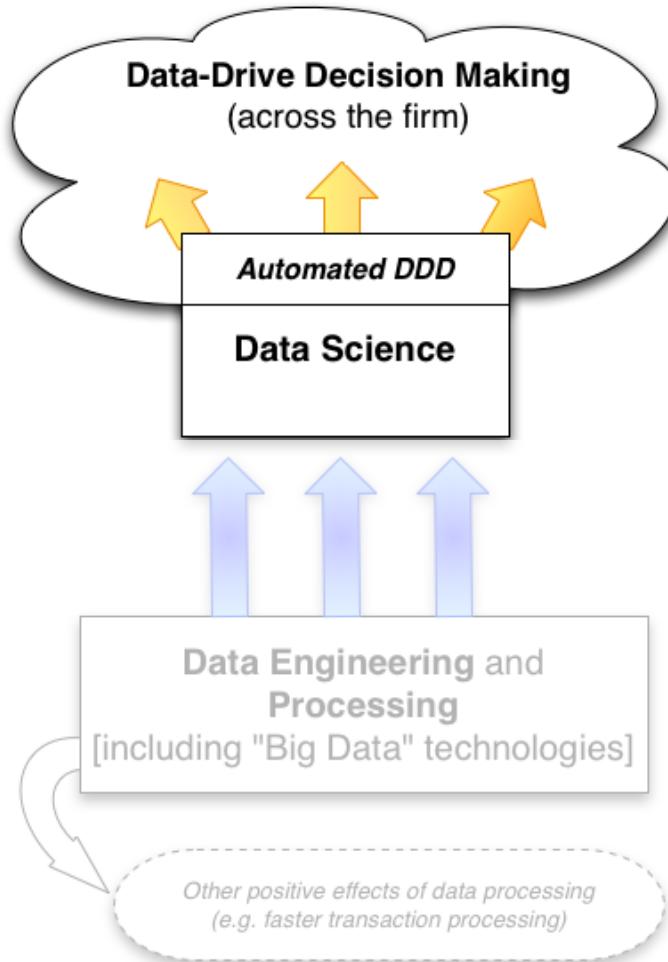
Extraction of knowledge from data via tools/technologies that *incorporate* the principles



In this class, we do both!

Fundamental principles/concepts which we will get to know

- ... use **well-defined** stages for analysis
- ... find informative **descriptive** attributes
- ... be careful with **overfitting**
- ... think about the **context** when evaluating results
- ... data & data science capability as **strategic asset**
- ... *and many more*



# Example: Data & data science capability as strategic asset

- Data and the capability to make decisions from data are complementary assets.

1990s: Signet Bank aims at modeling of profitability of credit card customers.

However, there was only data for the terms they had offered in the past.

They conducted experiments in order to build predictive models from the data (charge-offs!).

Today one of the largest credit-card issuers.



# Some examples

In which fields do we use data for gaining competitive advantages?



## Marketing (Advertising)

- Online advertising
- Recommendations for cross-selling
- Customer relationship management

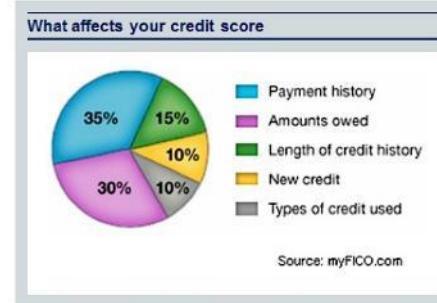


## Production

- Robotics and automation

## Finance

- Credit scoring and (high-frequency) trading
- Fraud detection
- Workforce management



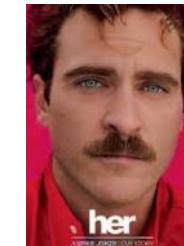
## Retail

- Supply Chain Management (Wal-Mart, Amazon etc.)



## Services

- Voice-activated services
- Human-machine communication (Siri, Alexa, Google Assistant)
- Augmented Reality Services



## Fragen?

- ✓ Metaintroduction
- ✓ Decision Support and Business Intelligence
- ✓ Data-analytic thinking

# Todos for next Week

1. Read short example about data & data science capability as strategic asset (case of Signet Bank)

*Kursmaterial > Readings/Übungen*

2. Read Goes (2014): How does the current acknowledgement of data as a valuable resource influence our research domain?

*Kursmaterial > Readings/Übungen*

3. Start your Python journey

*Kursmaterial > Readings/Übungen > Python Übungen > Jupyter Notebooks*

4. Remember your basics on data base design (including normalization)

|                          |   |   |   |
|--------------------------|---|---|---|
| <b>EDITOR'S COMMENTS</b> |  | Publication history<br>Frequency<br>Ranking (Jourqual)<br>Impact factor (2015)<br>Cited half-life | 1977 - present<br>Quarterly<br>A+<br>5.384<br>>10.0 years |
|--------------------------|---|---|---|

# Bibliography

- Bodendorf, Freimut. *Daten- und Wissensmanagement*. Springer-Verlag, 2006.
- Chen, Hsinchun, Roger HL Chiang, and Veda C. Storey. "Business intelligence and analytics: From big data to big impact." *MIS quarterly* 36.4 (2012): 1165-1188.
- Chandler, Daniel. *Semiotics: the basics*. Routledge, 2007.
- Gluchowski, Peter, and Peter Chamoni, eds.: *Analytische Informationssysteme: Business Intelligence-Technologien und Anwendungen*. Springer-Verlag, 2015.
- George, Gerard, Martine R. Haas, and Alex Pentland. "Big data and management." *Academy of Management Journal* 57.2 (2014): 321-326.
- Krcmar, Helmut. "Informationsmanagement." *Informationsmanagement*. Springer Berlin Heidelberg, 2015. 85-111
- McAfee, Andrew, et al. "Big data." *The management revolution. Harvard Bus Rev* 90.10 (2012): 61-67..
- Provost, F., & Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. *Big data*, 1(1), 51-59.
- Power, Daniel J. "A brief history of decision support systems." *DSSResources. COM, World Wide Web*, <http://DSSResources. COM/history/dsshistory. html>, version 4 (2007).
- Russom, Philip. "Big data analytics." *TDWI best practices report, fourth quarter* (2011): 1-35.
- Schieder, C., Dinter, B., Gluchowski, P.: *Metadatenmanagement in der Business Intelligence – eine empirische Untersuchung unter Berücksichtigung der Stakeholder-Perspektiven*. In: 12th International Conference on Wirtschaftsinformatik, 2015.

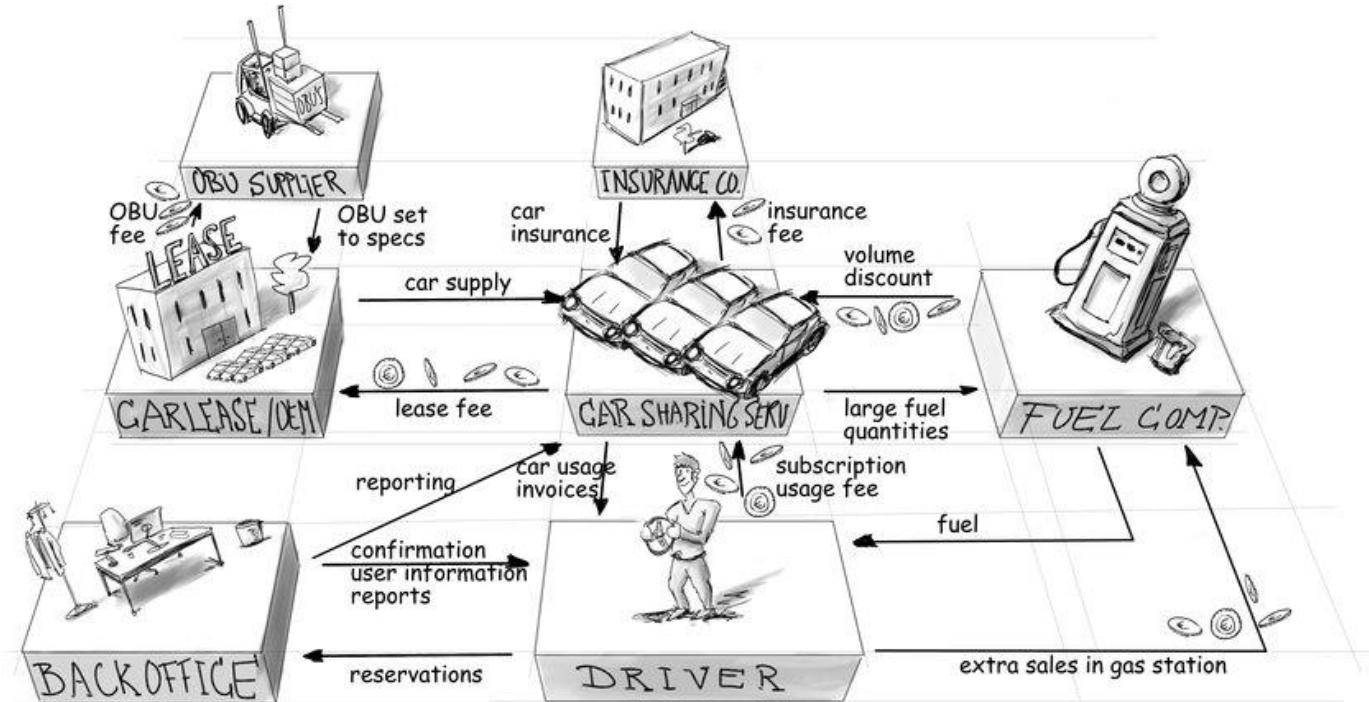
# Data - Why should companies care?

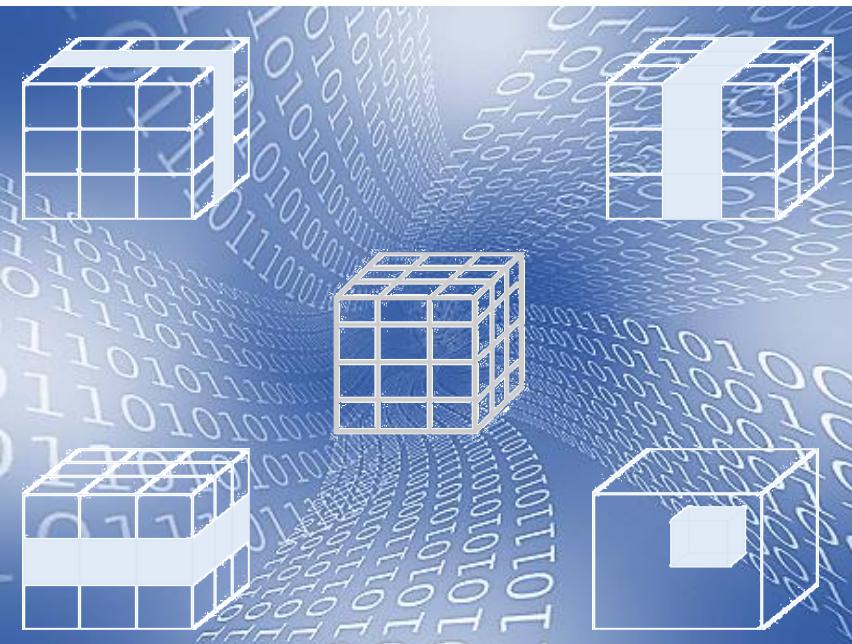
Data changes business models



*“[C]ompanies in the top third of their industry in the use of data-driven decision making were, on average, 5% more productive and 6% more profitable”*

(McAfee und Brynjolfsson, 2012, Result of 330 Interviews with Top-Managers and analysis of annual reports in Northern America)





# Business Intelligence

## 02 Data Warehouse – Overview & OLAP

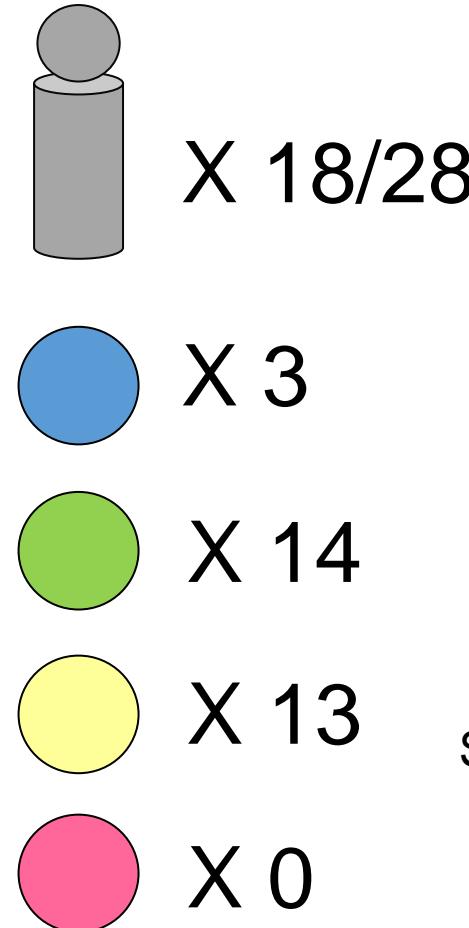
Prof. Dr. Bastian Amberg  
(summer term 2024)  
24.4.'24

# Schedule

|           | Wed., 10:00-12:00 |       | Fr., 14:00-16:00 (Start at 14:30)                   |       | Self-study  |                          |
|-----------|-------------------|-------|---|-------|---|--------------------------|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 | 19.4. |   | Python-Basics Chap. 1    |
|           | W2                | 24.4. | Data Warehouse – Overview & OLAP                    | 26.4. | [Blockveranstaltung SE Prof. Gersch]  | Chap. 2                  |
|           | W3                | 1.5.  |   | 3.5.  | Data Warehouse Modeling I   | Chap. 3                  |
|           | W4                | 8.5.  | Data Warehouse Modeling II                          | 10.5. | Data Mining Introduction  |                          |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     | 17.5. | Python-Basics-Online Exercise   | Python-Analytics Chap. 1 |
|           | W6                | 22.5. | Data Understanding, Data Visualization              | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam | Chap. 2                  |
|           | W7                | 29.5. | Data Preparation                                    | 31.5. |   |                          |
|           | W8                | 5.6.  | Predictive Modeling I                               | 7.6.  | Predictive Modeling II (10:00 -12:00)   | BI-Project Start         |
|           | W9                | 12.6. | Fitting a Model I                                   | 14.6. | Python-Analytics-Online Exercise  |                          |
|           | W10               | 19.6. | Guest Lecture                                       | 21.6. | Fitting a Model II  |                          |
|           | W11               | 26.6. | How to avoid overfitting                            | 28.6. | What is a good Model?   |                          |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I                                  |                          |
|           | W13               | 10.7. |   | 12.7. | From Machine to Deep Learning II  |                          |
|           | W14               | 17.7. | Project presentation                                | 19.7. | Project presentation  | End                      |
| Ref.      |                   |       |   |       | Klausur 1.Termin ~ 22.7. bis 3.8.<br>Klausur 2.Termin ~ 23.9. bis 5.10.                       | Projektbericht           |

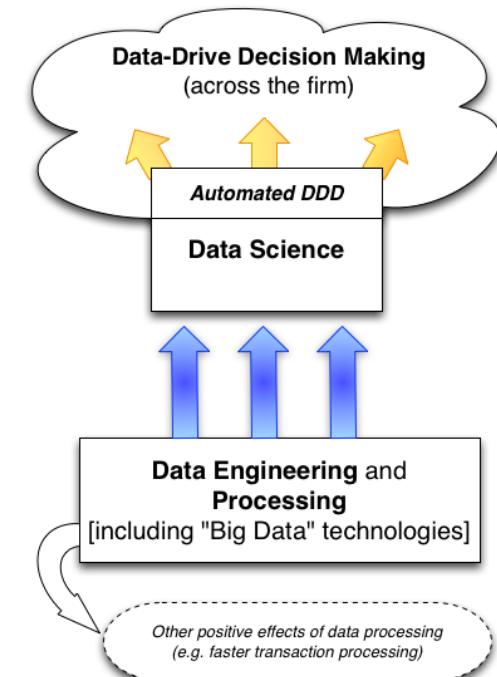
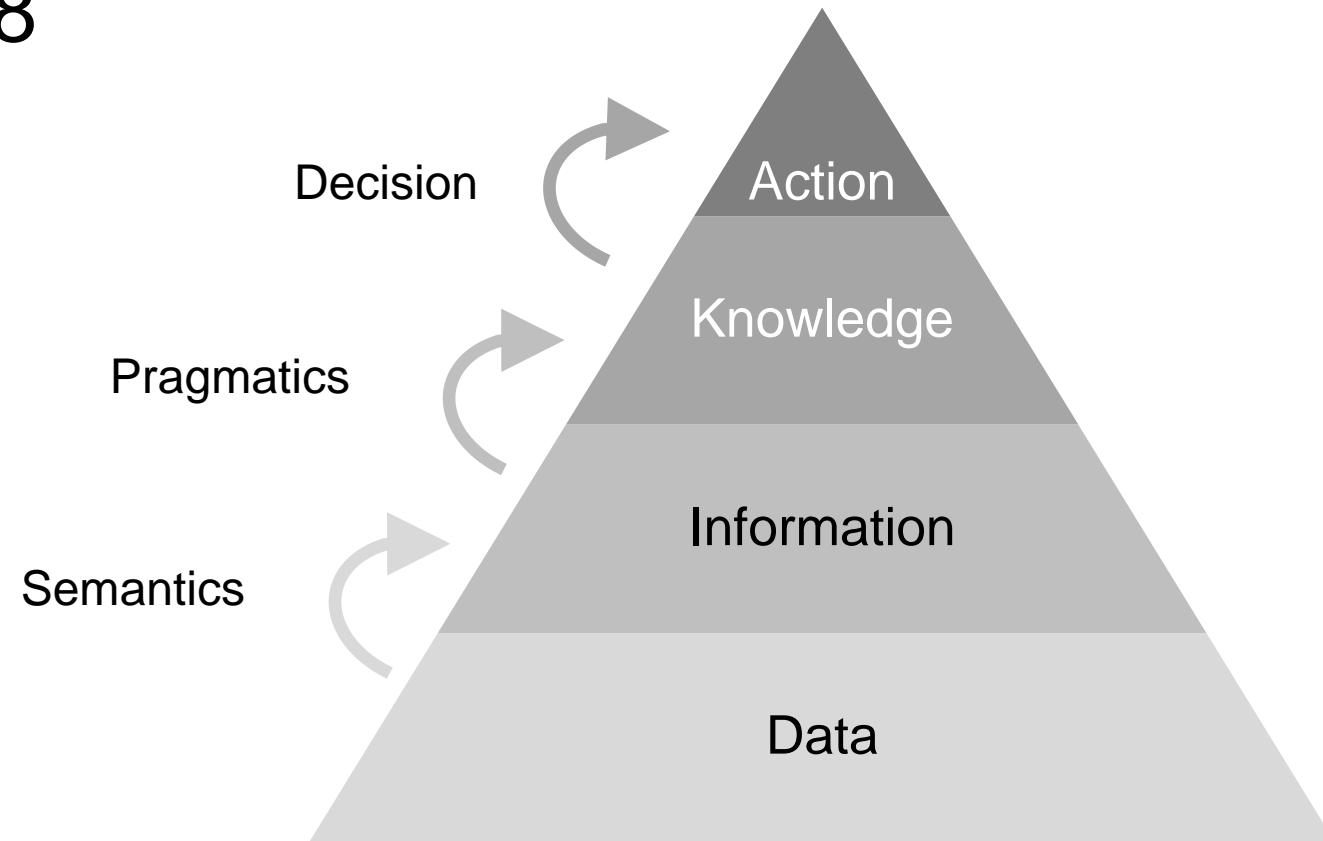
# Last lesson

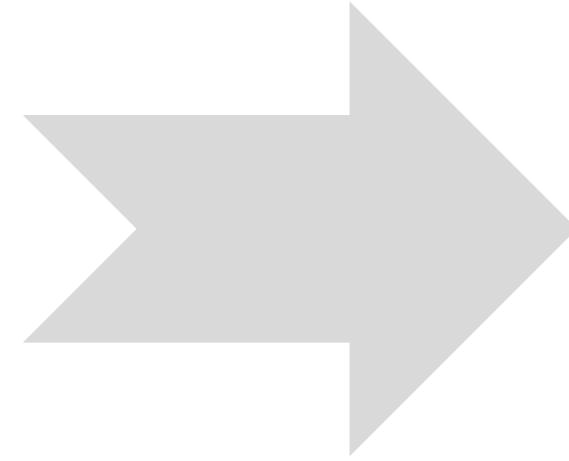
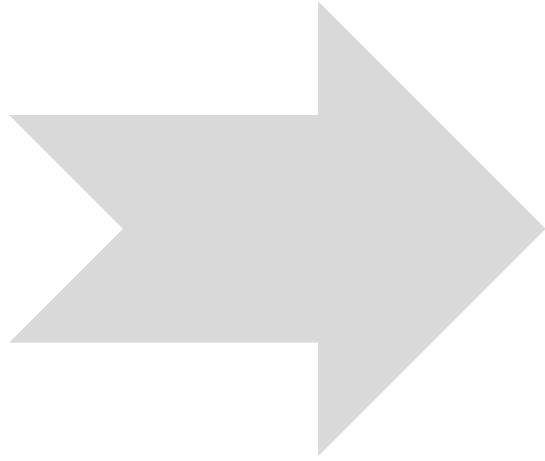
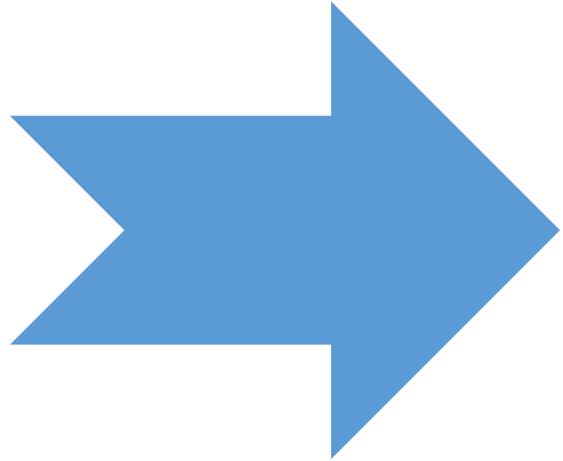
Some data about you...



Data-driven Decision Making (DDD) is a “*practice of basing decisions on the analysis of data rather than purely on intuition.*”

Transforming data to information to knowledge to action





**(1) An illustrative example,  
operational vs.  
analytical issues**

**(2) Basic outline of data warehouses (DWHs)**  
Distinguishing operational databases from DWHs  
Architecture of a DWH system  
Data within the DWH

**(3) Online Analytical Processing (OLAP)**  
Different query methods  
Properties of OLAP  
Common OLAP functionality

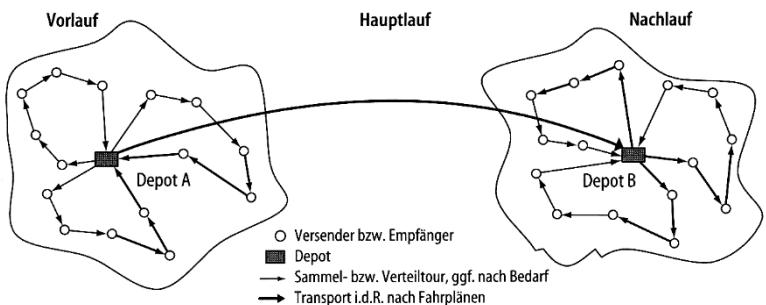
# An illustrative example

## Logistics service providers

Logistics service providers transform freight temporally and spatially (transportation services)

Services offered by logistics service providers differ in

- type and weight of goods,
- time of transportation,
- and price.



Ref.

**The 4 “R” of logistics:**  
the *right product* at the *right time* and  
at the *right place* in the *right quality*

**Courier service:** individually attended transportation of small goods. Transport occurs in the shortest possible time with high reliability

**Express service:** transportation of goods without weight and size limit

**Parcel service:** transportation of goods that are limited in volume

Integration of all processes along the supply chain in order to control transportation

Status of entities (i.e., goods for transportation) is very important

Data collection by Enterprise Resource Planning (ERP) systems



# An express letter from Germany to the US (1/2)



|                                      |  |  |                             |
|--------------------------------------|--|--|-----------------------------|
| <input checked="" type="checkbox"/>  | <b>Luftfrachtbrief: 1421247542 Unterschrieben für von: U RODRIGUEZ</b> | <b>Dienstag, November 06, 2012 am 10:37</b><br><b>Herkunftsgebiet: HANNOVER - braunschweig - GERMANY</b><br><b>Zielgebiet: OMAHA, NE - omaha - USA</b> | <b>JD013056300600107105</b> |
|                                      |  |  |                             |
| 11                                   | Verlässt DHL-Niederlassung in CINCINNATI HUB - USA                     | CINCINNATI HUB, OH - USA   | 05:36                       |
| 10                                   | Verzollung abgeschlossen in CINCINNATI HUB - USA                       | CINCINNATI HUB, OH - USA   | 03:22                       |
| 9                                    | Sendung sortiert in CINCINNATI HUB - USA                               | CINCINNATI HUB, OH - USA   | 03:22                       |
| 8                                    | Ankunft in der DHL-Niederlassung in CINCINNATI HUB - USA               | CINCINNATI HUB, OH - USA   | 01:23                       |
| <b>Donnerstag, November 01, 2012</b> |  | <b>Ort</b>   | <b>Zeit</b>                 |
| 7                                    | Sendung im Transit durch NEW YORK CITY GATEWAY - USA                   | NEW YORK CITY GATEWAY, NY - USA  | 10:43                       |
| 6                                    | Verlässt DHL-Niederlassung in LEIPZIG - GERMANY                        | LEIPZIG - GERMANY  | 03:21                       |
| 5                                    | Sendung sortiert in LEIPZIG - GERMANY                                  | LEIPZIG - GERMANY  | 00:22                       |
| <b>Mittwoch, Oktober 31, 2012</b>    |  | <b>Ort</b>   | <b>Zeit</b>                 |
| 4                                    | Ankunft in der DHL-Niederlassung in LEIPZIG - GERMANY                  | LEIPZIG - GERMANY  | 23:25                       |
| 3                                    | Verlässt DHL-Niederlassung in HANNOVER - GERMANY                       | HANNOVER - GERMANY   | 20:58                       |
| 2                                    | Sendung sortiert in HANNOVER - GERMANY                                 | HANNOVER - GERMANY   | 19:31                       |
| 1                                    | Sendung abgeholt   | HANNOVER - GERMANY   | 15:07                       |

Ref.

# An express letter from Germany to the US (2/2)



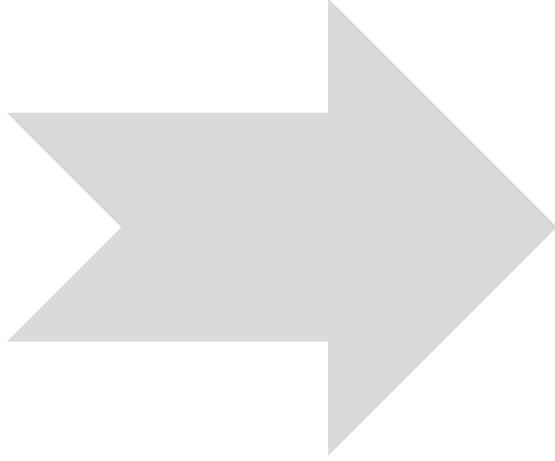
|                                     |  |  |                             |                      |
|-------------------------------------|--|--|-----------------------------|----------------------|
| <input checked="" type="checkbox"/> | <b>Luftfrachtbrief: 1421247542 Unterschrieben für von: U RODRIGUEZ</b> | <b>Dienstag, November 06, 2012 am 10:37</b><br><b>Herkunftsgebiet: HANNOVER - braunschweig - GERMANY</b><br><b>Zielgebiet: OMAHA, NE - omaha - USA</b> | <b>JD013056300600107105</b> |                      |
| <b>Dienstag, November 06, 2012</b>  |  | <b>Ort</b>   | <b>Zeit</b>                 | <b>Stücke</b>        |
| 19                                  | Sendung zugestellt - übernommen von : U RODRIGUEZ                      | omaha  | 10:37                       | JD013056300600107105 |
| 18                                  | Sendung in Zustellung  | OMAHA, NE - USA  | 08:17                       | JD013056300600107105 |
| <b>Montag, November 05, 2012</b>    |  | <b>Ort</b>   | <b>Zeit</b>                 | <b>Stücke</b>        |
| 17                                  | Sendung in Zustellung  | OMAHA, NE - USA  | 17:00                       | JD013056300600107105 |
| 16                                  | Erfolgloser Zustellversuch, Empfänger nicht zu Hause                   | OMAHA, NE - USA  | 08:42                       | JD013056300600107105 |
| <b>Freitag, November 02, 2012</b>   |  | <b>Ort</b>   | <b>Zeit</b>                 | <b>Stücke</b>        |
| 15                                  | Sendung zur Aufbewahrung in DHL-Niederlassung                          | OMAHA, NE - USA  | 20:37                       | JD013056300600107105 |
| 14                                  | Sendung in Zustellung  | OMAHA, NE - USA  | 17:38                       | JD013056300600107105 |
| 13                                  | Erfolgloser Zustellversuch, Empfänger nicht zu Hause                   | OMAHA, NE - USA  | 16:21                       | JD013056300600107105 |
| 12                                  | Ankunft in der DHL-Zustellbasis in OMAHA - USA                         | OMAHA, NE - USA  | 08:32                       | JD013056300600107105 |

DHL: Processing of approx. 70,000 shipments / night per transshipment facility  
(35 transshipment facilities in Germany)

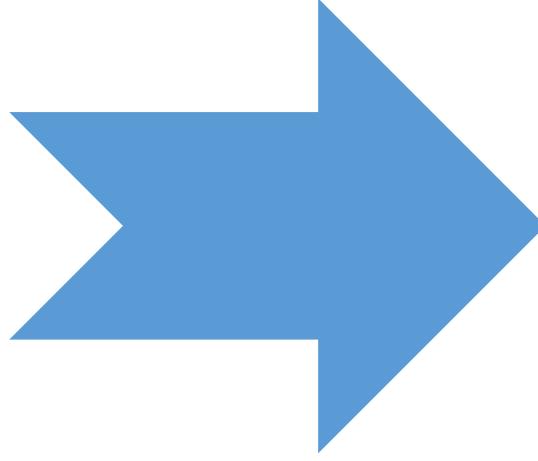
# Operational Issues vs. Analytical Issues

## Differences?

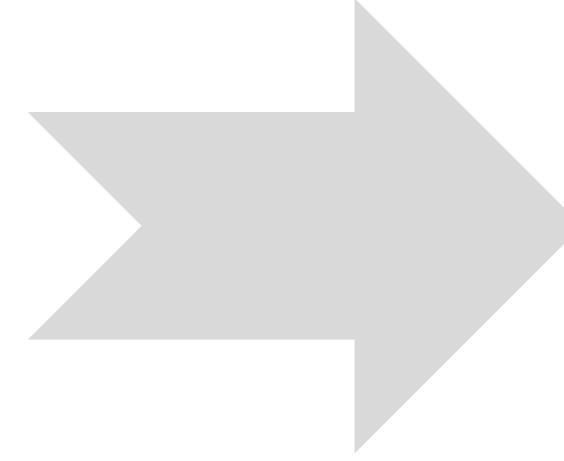
- Wo befindet sich die Sendung XYZ zurzeit?
- An welchen Stationen wurde sie umgeschlagen?
- Ist eine Beschädigung der Sendung eingetreten?
- Wann wird sie voraussichtlich ankommen?
- Wer hat den Erhalt der Sendung quittiert?
- Zu wie viel Prozent ist die Sortieranlage in Cincinnati ausgelastet?
- Welcher Verteilung folgt die Verweildauer für den Hub Leipzig?
- Lohnt ein Direktflug Leipzig – Cincinnati bzw. Hannover – New York?
- Wieviel Prozent der Sendungen werden termingerecht ausgeliefert?
- Welche Kosten verursacht die Zustellung gegenüber dem Transport?



(1) An illustrative example



(2) Basic outline of data warehouses (DWHs)  
**Distinguishing operational databases from DWHs**  
Architecture of a DWH system  
Data within the DWH



(3) Online Analytical Processing (OLAP)  
Different query methods  
Properties of OLAP  
Common OLAP functionality

# Operational databases

A database for the day-to-day business

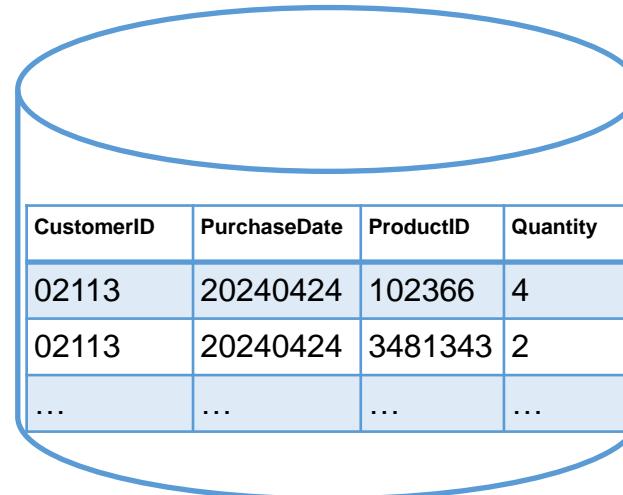
Operational databases...

- primarily *support the day-to-day business* (“real-time databases”)
- record operative business transactions
- aim at storing transactional details with *full integrity* and *without redundancy*

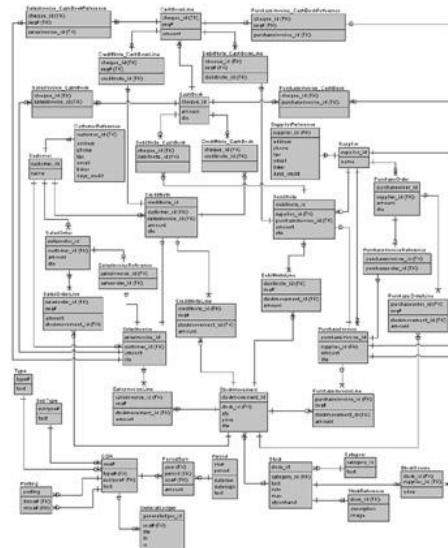
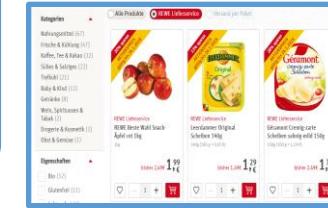
For this reason, data is regularly stored in a **complex** way:

- many details
- many updates
- Highly *normalized* (*1NF, 2NF, 3NF*)

-> Operational databases are seldomly very user-friendly



| CustomerID | PurchaseDate | ProductID | Quantity |
|------------|--------------|-----------|----------|
| 02113      | 20240424     | 102366    | 4        |
| 02113      | 20240424     | 3481343   | 2        |
| ...        | ...          | ...       | ...      |

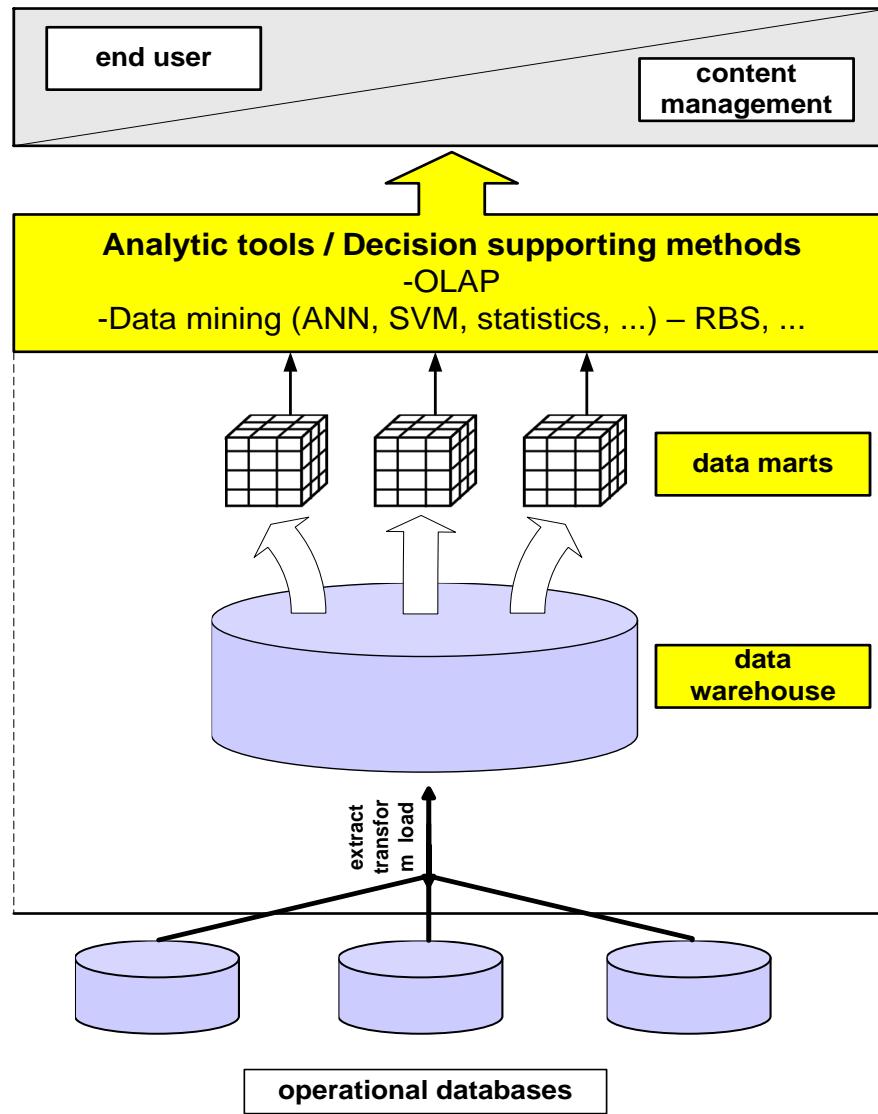


## Excusus: Normalization

- 1. Normalform (1NF)**  
*Herstellung der Grundform einer Relation (atomare Attribute)*
- 2. Normalform (2NF)**  
*Auflösung von Teil-abhängigkeiten (abh. Von Primärschlüssel)*
- 3. Normalform (3NF)**  
*Auflösung transitiver Abhängigkeiten (nicht-Schlüsselattribute sind unabhängig)*

# Operational databases and Data Warehouses

## Reasons for data warehousing



## Data Warehouses

- collect data from operational databases
- accumulate **historical data**
- provide the basis for Business Intelligence applications

Data warehousing has become a **strategic goal** of many companies

1998: 90% of the 2000 biggest companies worldwide were already developing data warehouses

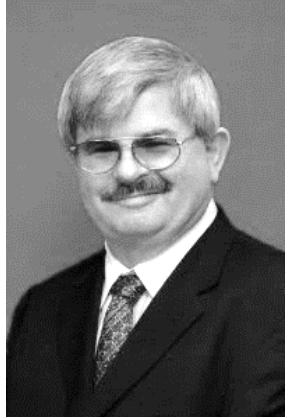
## Reasons for the development of data warehouses:

- Integration of many different data sources into one database
- Creating a better basis for data mining tools
- Controlling the “information flooding” by structuring and aggregating of operative data
- Analyzing tools can be applied to complex questions

For example?



# Definition of Data Warehouses



W.H. Inmon was the first to provide a definition:

*“A Data warehouse is a **subject-oriented**, **integrated**, **time-variant**, and **non-volatile** collection of data in support of management’s decision-making process.”*

(Inmon, 1992)

Detailed look at the keywords of the definition:

## **Subject-oriented**

data gets organized corresponding to the business context of the particular company

## **Integrated**

data from many different internal and external sources is loaded into the DWH

## **Time-variant**

time series analysis is possible by the means of DWHs

## **Non-volatile**

data is stored persistently and read-only access is provided

# Operational databases vs. DWHs

A comparison

Another (more concrete) definition of data warehouses:

*"A data warehouse is a decision supporting database (analytic database), which is separated from the operational databases, and which is primarily used for decision support in a company.*

*A data warehouse is always modeled in a multidimensional way and is used for the long-term storage of historic, cleaned, validated, synthesized, operative data from internal and external sources."*

(A. Kurz, 1998)

| Properties                       | Operational Database | Data Warehouse |
|----------------------------------|----------------------|----------------|
| Operational data                 |                      |                |
| Detailed data                    |                      |                |
| Complex data model               |                      |                |
| Strategic data                   |                      |                |
| Processing requires many queries |                      |                |
| Interface end-user oriented      |                      |                |



Full amount of Data covered

Derived data  
(e.g. aggregates -> redundant data)

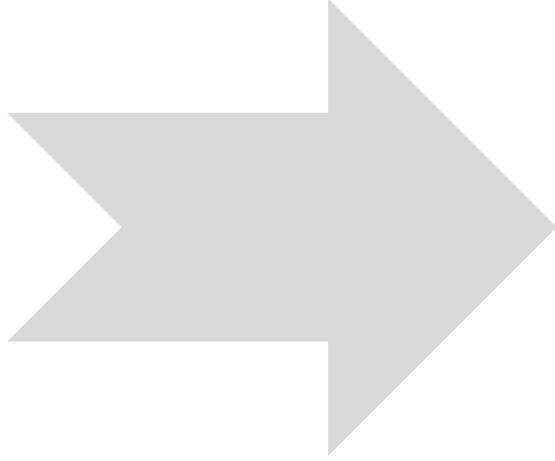
Many Updates

Ad hoc queries

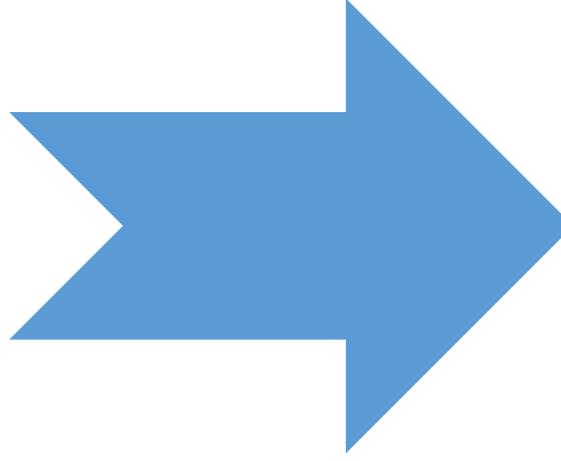
Historical data

Non-redundant data

5 Min.



(1) An illustrative example

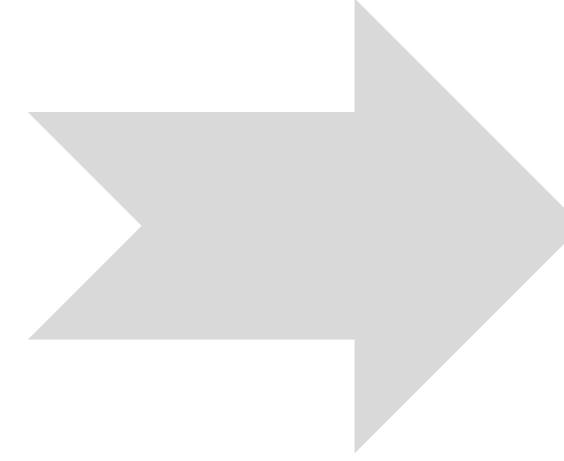


(2) Basic outline of data warehouses (DWHs)

Distinguishing operational databases from DWHs

### **Architecture of a DWH system**

Data within the DWH



(3) Online Analytical Processing (OLAP)

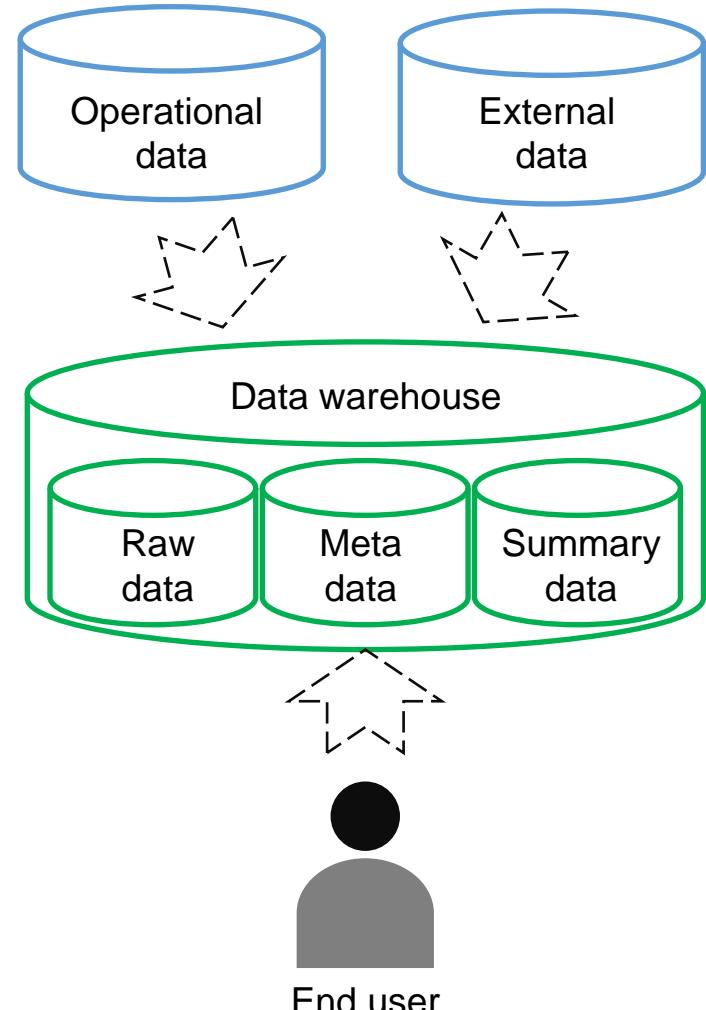
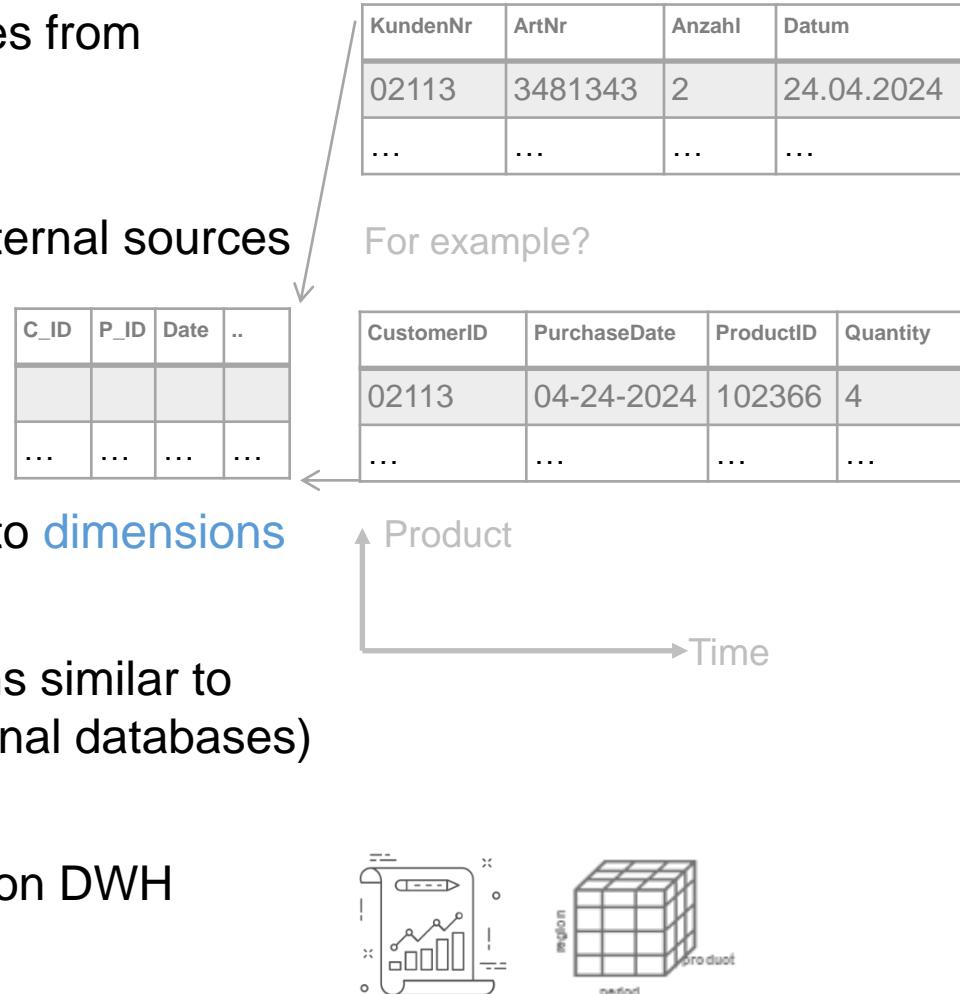
Different query methods

Properties of OLAP

Common OLAP functionality

# Basic steps concerning DWHs

1. Select appropriate attributes from operational databases
2. Add selected data from external sources
3. Transform and load data
4. Store loaded data subject to **dimensions**
5. (Administrational operations similar to those known from operational databases)
6. Query and analyze based on DWH (reports, **OLAP**)



# Key elements of data warehouse systems (1/2)



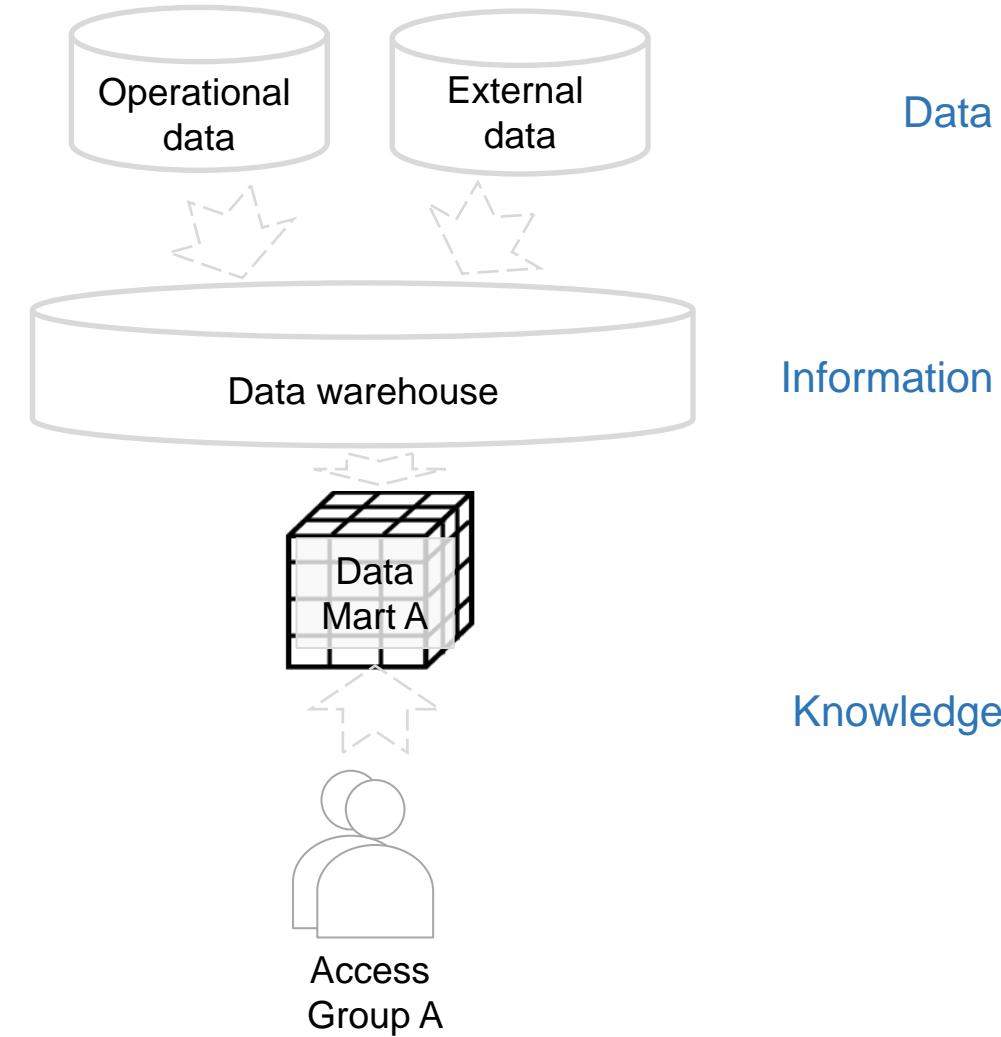
## Data Marts

The core of a data warehouse system can be built out of several components:

### Data marts

(small) analytic databases **for a special group of people** (e.g. department, or workgroup)

- administration by **local** departments instead of Central IT
  - coordination with other analytic databases
  - development less complex than creation of larger DWHs
  - based on **specialized data models**, which are quite easy to understand and which provide efficient access to data
  - end users can be easily involved in the development of the single data marts
- load data from other DWHs or from operational databases
  - Distribution of analytic data among data marts is a difficult task
    - Build homogenous user groups
    - assimilate data model to a functional area



# Key elements of data warehouse systems (2/2)

Data Marts, Central Data Warehouse, Enterprise Data Warehouse

## Central data warehouse

analytical database provides data transformed and coordinated to local data marts not necessarily providing information for the whole company

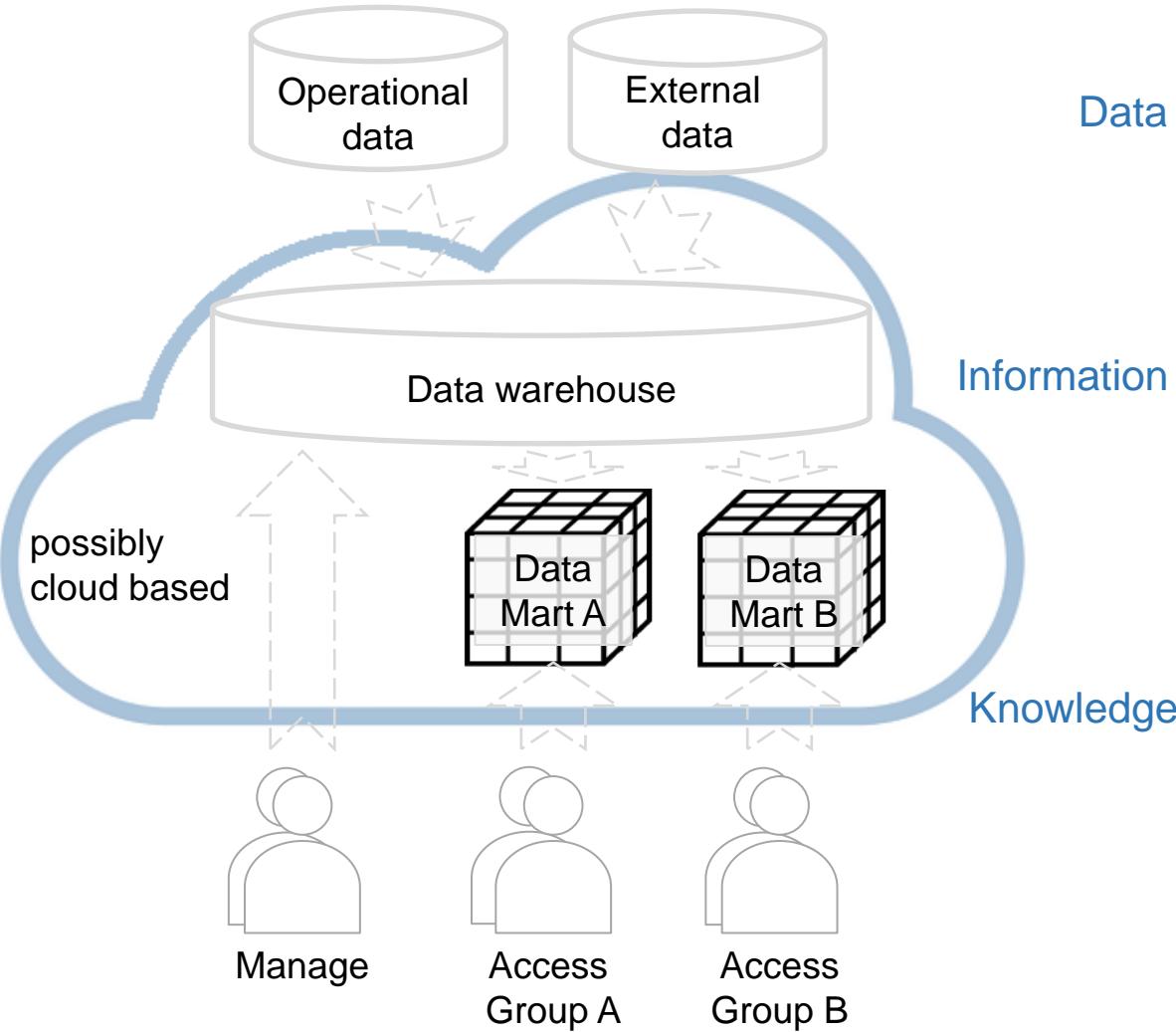
## Enterprise data warehouse (EDWH)

central data warehouse providing data and information for the whole company

Typical Issues?

<https://aws.amazon.com/de/redshift/>  
<https://www.cloudera.com/products/enterprise-data-hub.html>  
<https://cloud.google.com/bigquery/>  
<https://www.vertica.com/overview/>  
<https://www.ibm.com/cloud/db2-warehouse-on-cloud>  
<https://azure.microsoft.com/de-de/services/sql-data-warehouse/>  
[https://cloud.oracle.com/de\\_DE/database](https://cloud.oracle.com/de_DE/database)  
<https://www.sap.com/germany/products/bw4hana-data-warehousing.html>  
<https://www.snowflake.com/product/>  
<https://www.teradata.de/Products/Cloud>

List of April 2024



# Data Warehouse vs. Data Lake

Search by yourself

e.g. <https://blogs.oracle.com/bigdata/data-lake-database-data-warehouse-difference>

*Todo for  
next Friday*

Type of data?

Task?

Users?

Data processing?

Data granularity?

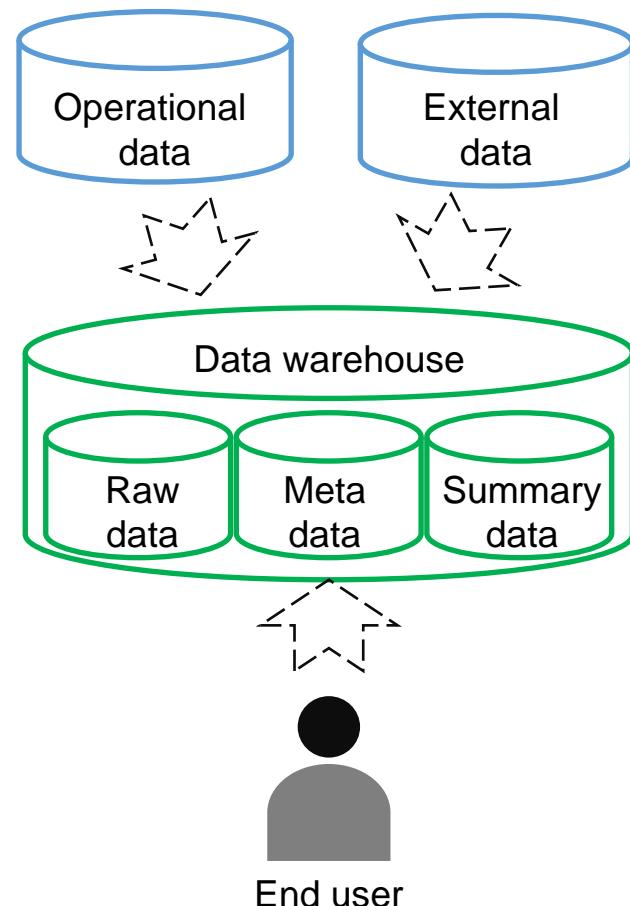
Ref.



# Architectures (1/2)

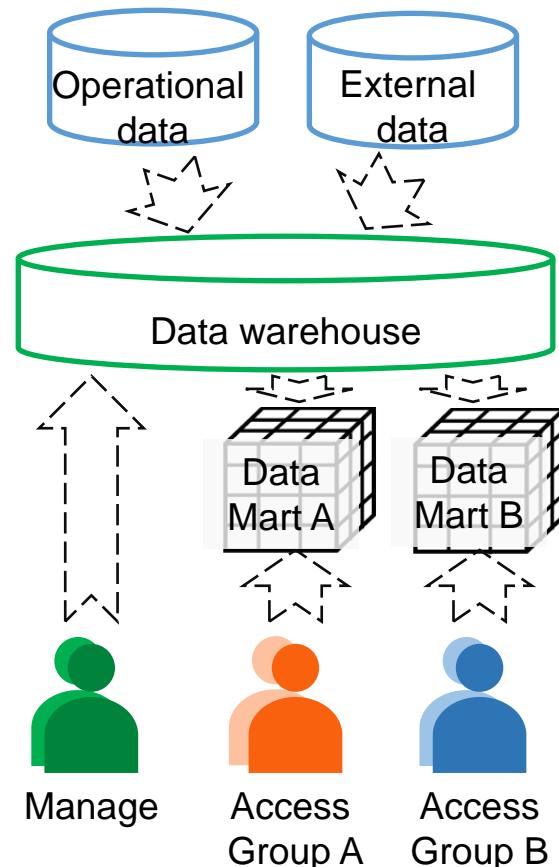
## Centralized architecture

All the analytic data is bundled on **one platform**



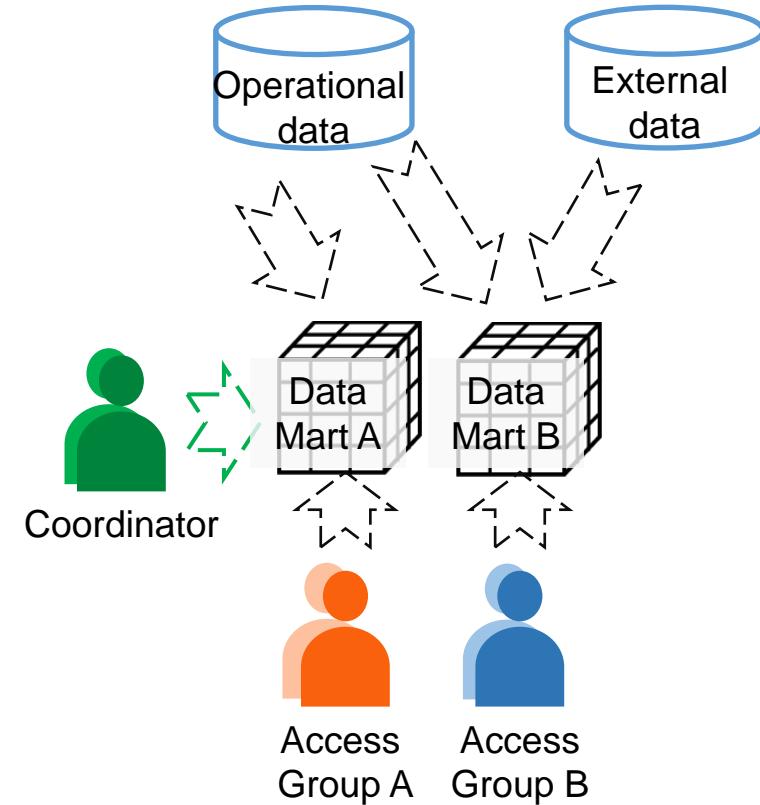
## Hierarchical architecture

Local data marts become **coordinated** by an enterprise data warehouse (EDWH)



## Enterprise data mart architecture

Central DWH functionally replaced by **coordinated data marts**





## Centralized architecture

All the analytic data is bundled on  
**one platform**

Advantages:

- Less redundancy
- Savings concerning hardware  
(For On-Premise-DWH)

Disadvantages:

- Limited possibilities for modularization
- Development too complex for many companies
- Degree of user friendliness and efficiency is only for small companies sufficient



Ref.

## Hierarchical architecture

Local data marts become **coordinated** by an enterprise data warehouse (EDWH)

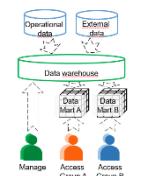
The EDWH extracts, integrates and distributes data

The data marts are...

- ... used for querying and for analyzing data
- ... specialized on a special functional field within the company

Coordination of attributes necessary  
(bijective relationship between attribute and description:  
no homonyms, no synonyms, no aliases)

Example?  
Individual = Customer = Person = Employee?  
Customer = Client = Consumer ?



## Enterprise data mart architecture

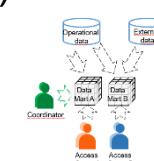
Central DWH functionally replaced by **coordinated data marts**

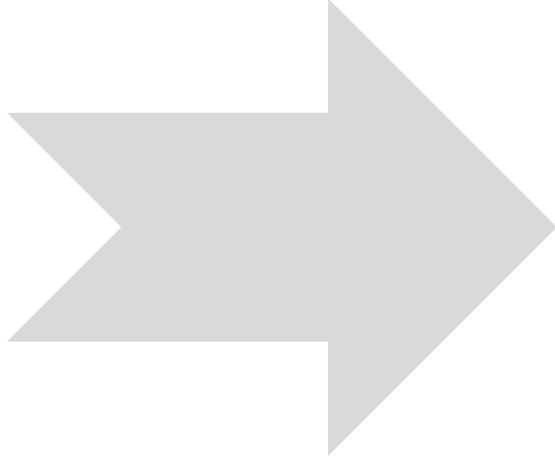
Often based on a distributed database system

Extraordinary focus on the maximization of *intramodularity* & minimization of *intermodularity* of Data Marts

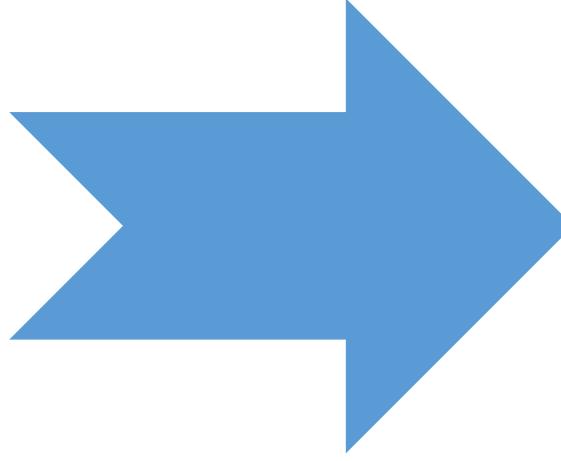
High efforts for coordination required

- Load and access coordination
- Coordination of data model (metadata)





(1) An illustrative example

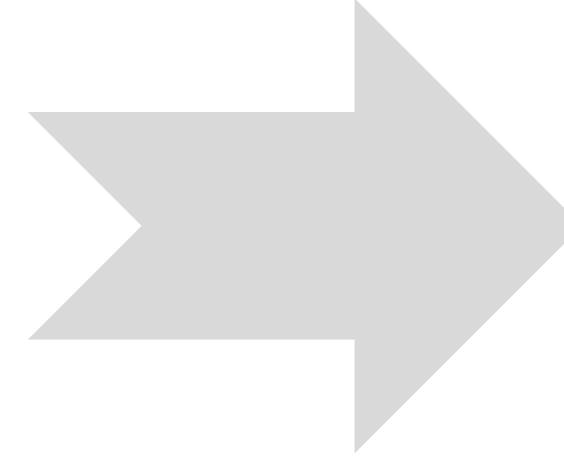


(2) Basic outline of data warehouses (DWHs)

Distinguishing operational databases from DWHs

Architecture of a DWH system

**Data within the DWH**



(3) Online Analytical Processing (OLAP)

Different query methods

Properties of OLAP

Common OLAP functionality

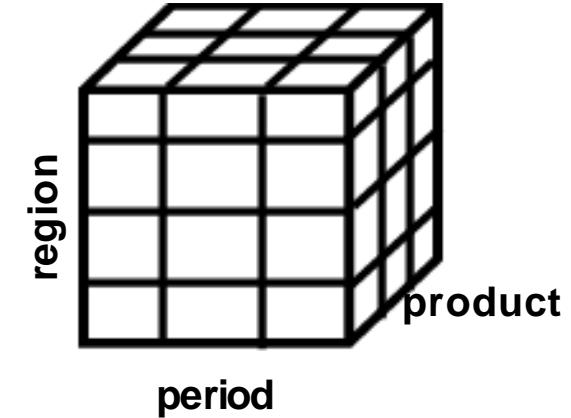
# Multidimensional data

Analytical data are represented by multidimensional data models

Modeling user-friendly and close to business

The hypercube data structure is based on **business measures ("facts")** and **dimensions**

|                    | <b>business measure (or fact)</b>                      | <b>dimension</b>                               |
|--------------------|--|--|
| <b>purpose</b>     | analysis of success subject to several dimensions      | selection, aggregation and navigation of facts |
| <b>examples</b>    | quantity turnover, monetary turnover                   | product, region, period                        |
| <b>synonyms</b>    | fact, performance measure, key business measure        | constraint                                     |
| <b>datatype</b>    | numeric and continuous                                 | symbolic and discrete                          |
| <b>data volume</b> | Large (about 70% of the DWH)                           | small  |
| <b>key</b>         | primary key consists of foreign keys of the dimensions | primary keys                                   |

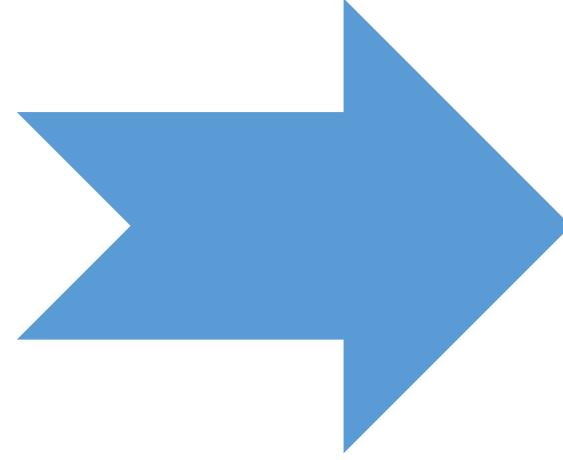
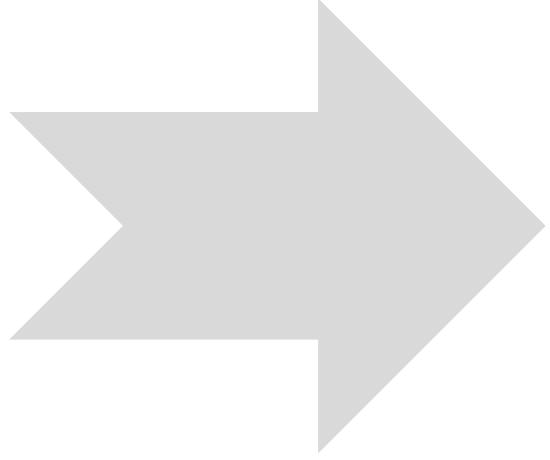
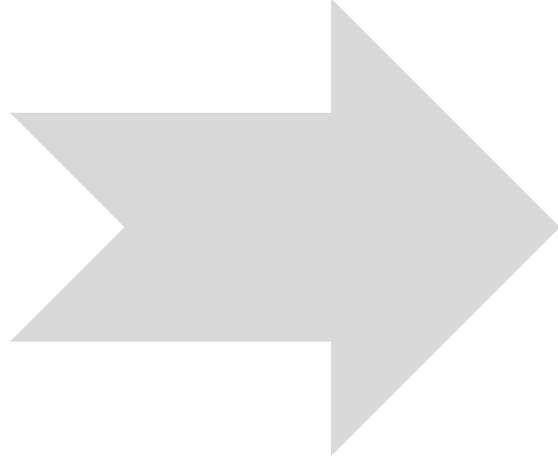


Kahoot-Fragen  
[www.kahoot.it](http://www.kahoot.it)

(über Smartphone oder Laptop)  
PIN folgt

Distinguish facts and dimensions!

(Diese Folie ist nach der Vorlesung mit Lösungen verfügbar)



(1) An illustrative example

(2) Basic outline of data warehouses (DWHs)

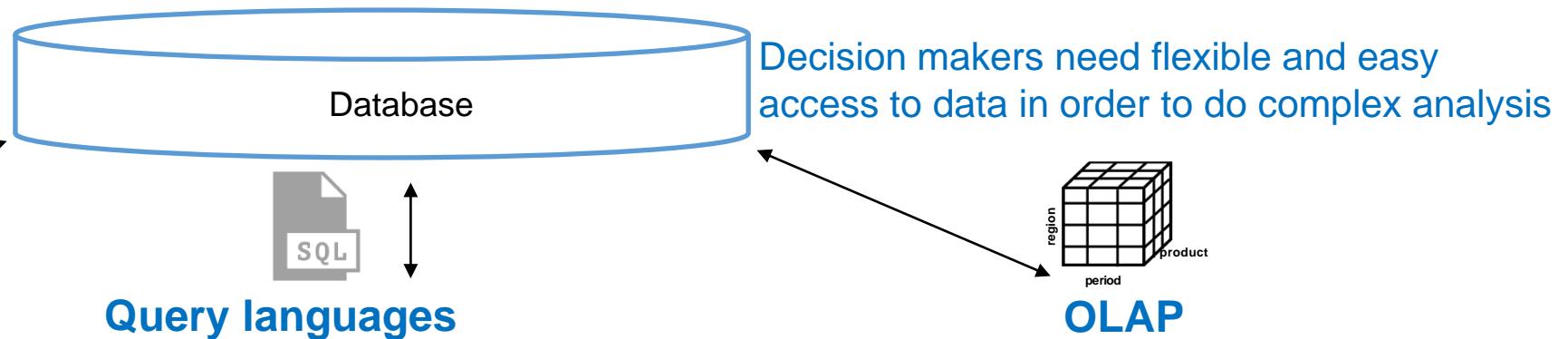
Distinguishing operational databases from DWHs  
Architecture of a DWH system  
Data within the DWH

(3) Online Analytical Processing (OLAP)

**Different query methods**  
Properties of OLAP  
Common OLAP functionality

# Query methods

Three means to query databases



## Programmed reports

- arbitrarily modifiable
- programmer required for changes

dBase code for "Which are the properties of the products of the department „Mobile Computing“?":

```
use PRODUCTS
copy to TMP
use TMP
delete for producttype <> 'MOBILE'
total on PRODUCTS to RESULT
display all
```

## Query languages

- standardized and powerful
- difficult to learn
- e.g. SQL, QBE

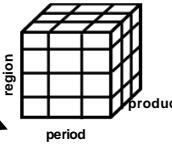
SQL query for "Which are the properties of the products of the department „Mobile Computing“?":

```
SELECT *
FROM Products
WHERE producttype = 'MOBILE'
```

## Using SQL for multidimensional querying is difficult:

- Several **(inner) queries** (and joins) needed in many cases
- Queries often become quite complex
- Difficult to do time series analysis
- Limited ways for doing **statistical calculations**

Decision makers need flexible and easy access to data in order to do complex analysis

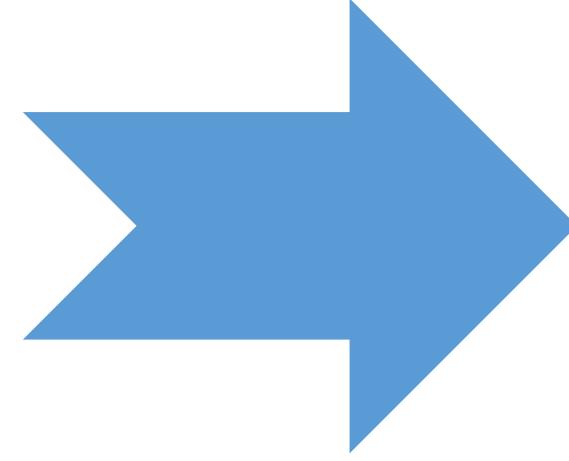
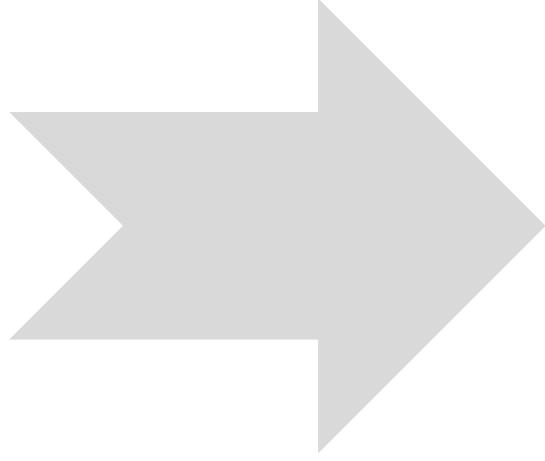
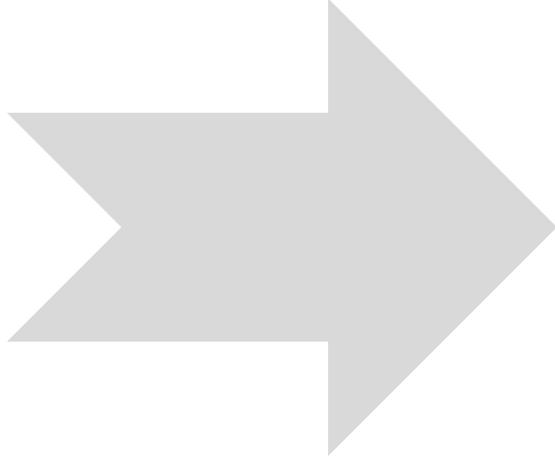


## OLAP

- flexible ad-hoc querying
- possible without expertise

SQL query for "What was the **average sales** of the department **„Mobile Computing“** to **Government customers** for the **third quarter** of calendar year 2001?":

```
SELECT customer, ROUND(AVG(sales),2) as average,
       ROUND(MIN(sales),2) as minimum, ...
  FROM units_cube_cubeview
 WHERE time_calendar_year = 'Q3_2001'
   AND product_ldsc = 'MOBILE'
   AND customer_market_segment_prnt
     = 'MARKET_SEGMENT_GOV'
   AND channel_level = 'TOTAL_CHANNEL'
 GROUP BY customer
 ORDER BY customer;
```



(1) An illustrative example

(2) Basic outline of data warehouses (DWHs)

Distinguishing operational databases from DWHs

Architecture of a DWH system

Data within the DWH

(3) Online Analytical Processing (OLAP)

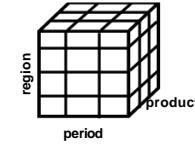
Different query methods

**Properties of OLAP**

Common OLAP functionality

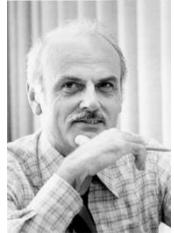
# Online Analytical Processing (OLAP)

Let's focus on end users, and their access to data marts by OLAP systems



## OLAP systems

- combine querying and interactive analysis
- present a multidimensional view on data



OLAP was introduced by E. F. Codd (one of the founding fathers of relational data bases) in 1993, who established 12 rules to define OLAP

## OLAP functionality

- video for illustration

(exemplary <https://www.youtube.com/watch?v=V37vPxIxUwo> )

A more concise definition of OLAP is **FASMI**

**Fast**

**Analysis of**

**Shared**

**Multidimensional** Truly multidimensional conceptual view of the data

**Information**

OLAP systems deliver responses to analyze queries within seconds (ideally maximum 5 – 20 seconds)

Cope with any business logic and statistical analysis that is relevant to the user: Mathematic modeling, time series analysis, goal seeking, what-if, drill-down etc., but no programming

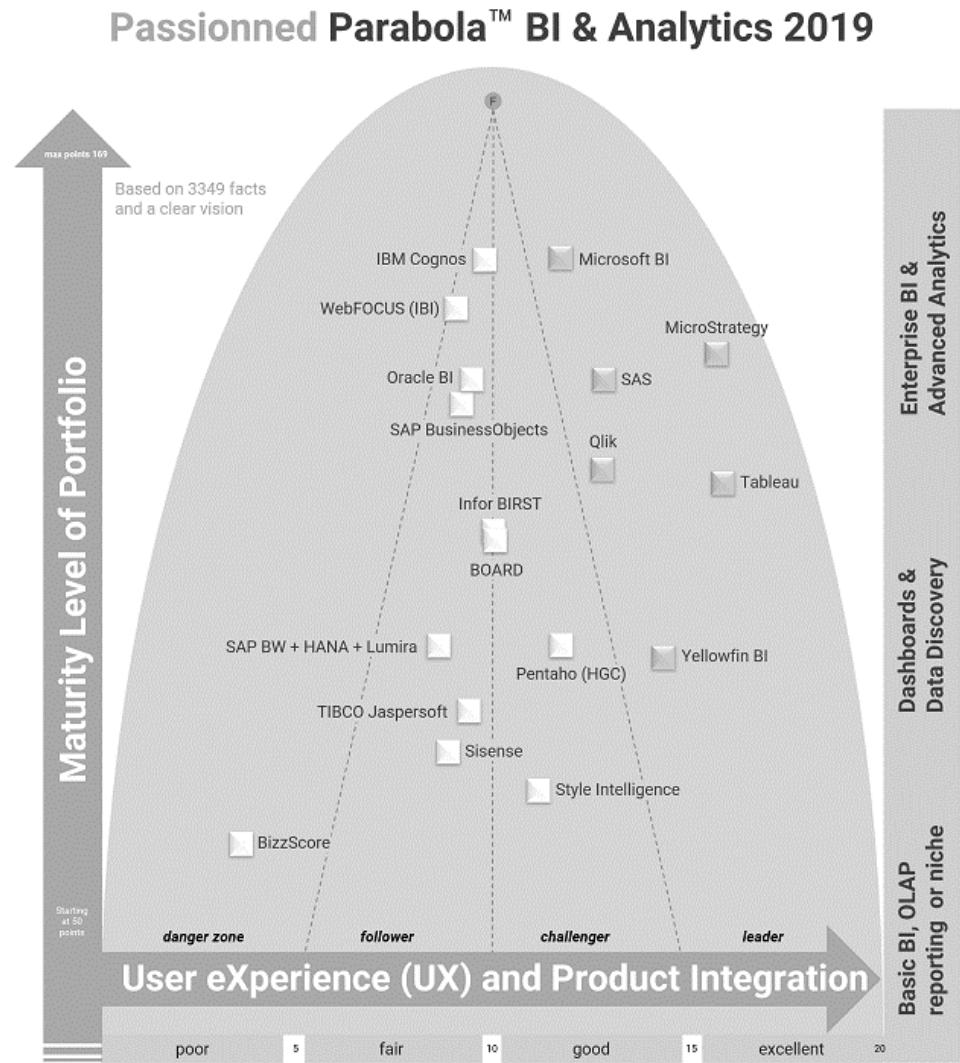
Multiple user access and varying roles with necessary security requirements for confidentiality.

# OLAP functions

OLAP tools provide a number of standard features

- Different representation modes:
  - absolute as well as relative representation of data
  - 3-dimensional analysis using layers
  - various calculation options (internal or plug-ins)
- Special cube operators provide browsing functions:
  - drilling
    - drill up/down ⇒ detailing/aggregating along a dimension
    - drill through ⇒ access to operational databases
    - ...
  - pivoting (rotating) ⇒ switch rows and columns
  - slicing ⇒ reduce number of dimensions
  - dicing ⇒ cutting parts out of the current cube (filtering)
- Various visualization options

OLAP Tools -> part of BI Tools...



<https://www.passionned.com/bi/tools/>

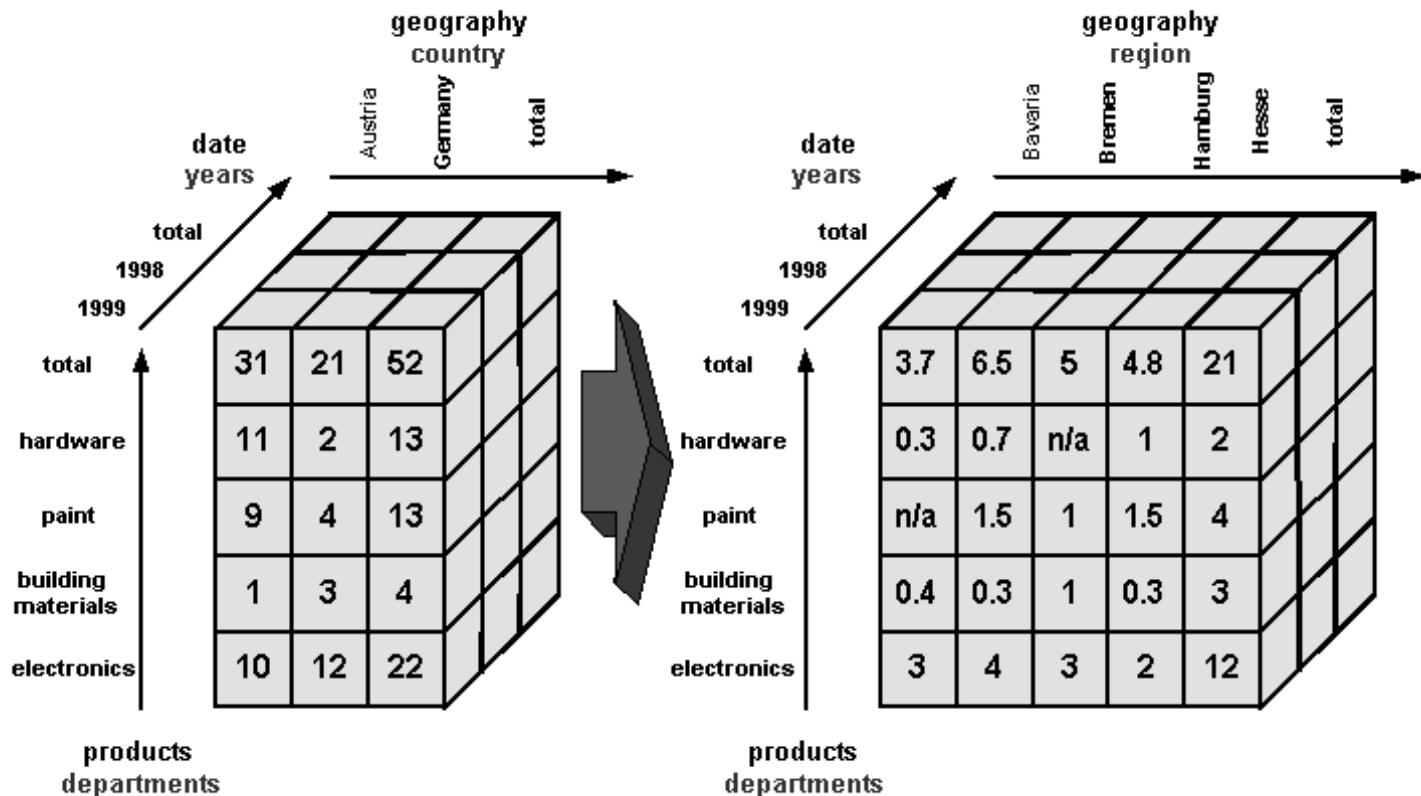
See <https://www.passionned.com/bi/#list-business-intelligence-tools>  
for an up-to-date list with detailed information about BI Tools, April 2024

# Drilling down

More details for specific dimensions



“Show the [regions of Germany in detail.](#)”

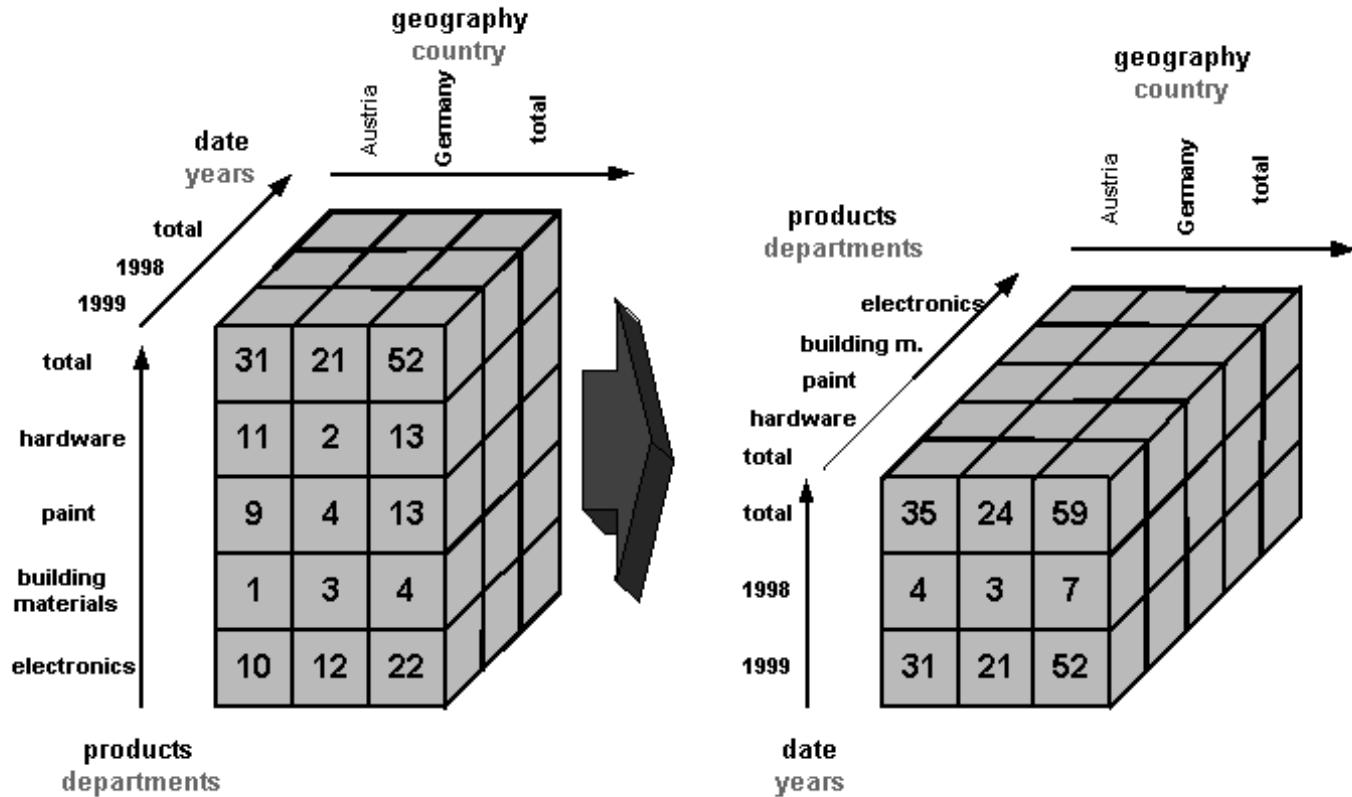


# Pivoting

Rotate the cube

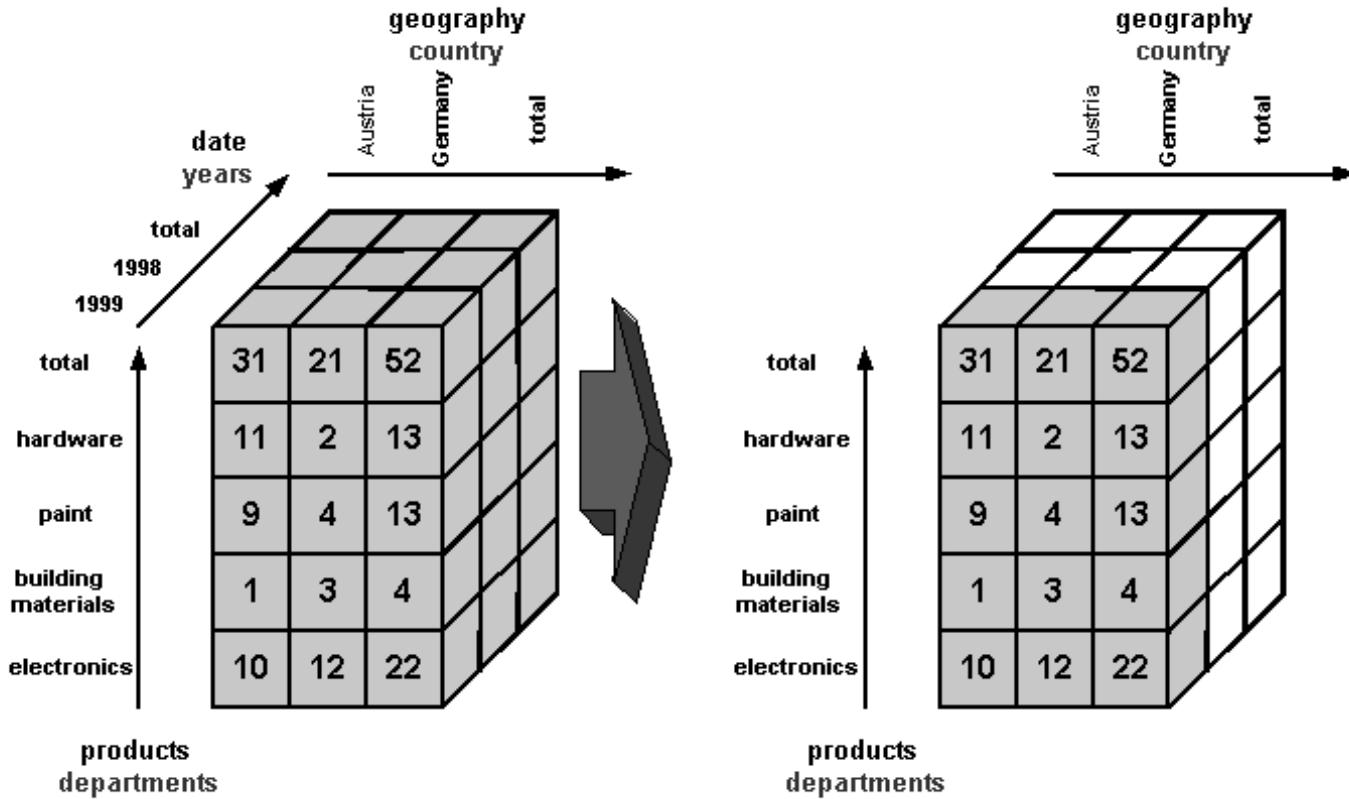


“Show year by country instead of product by country”

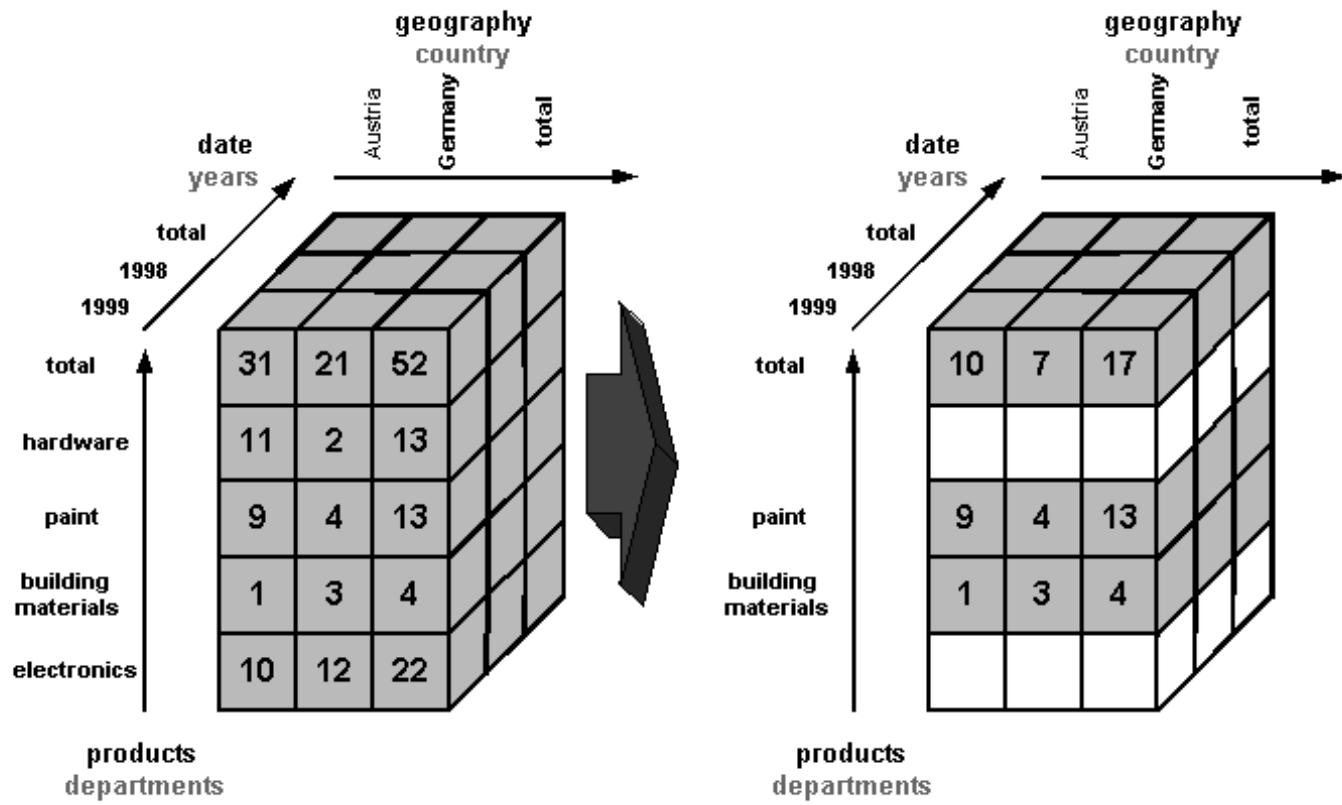


# Slicing

“Show only the values for 1999.”



“Show only the values for the departments ‘paint’ & ‘building materials’ for all the countries and all the years.”



Pro:

- Wide applicability of the method
- OLAP presents quite exact results
- Method is plausible

Con:

- OLAP requires a lot of user interaction
- OLAP regularly requires quite a lot of computing resources
- Difficult to use automated data mining routines in combination with OLAP

## Fragen?

- ✓ An illustrative example
- ✓ Basic outline of data warehouses (DWHs)
  - ✓ Distinguishing operational databases from DWHs
  - ✓ Architecture of a DWH system
  - ✓ Data within the DWH
- ✓ Online Analytical Processing (OLAP)
  - ✓ Different query methods
  - ✓ Properties of OLAP
  - ✓ Common OLAP functionality

# Todos for next Friday

1. Data Warehouse vs. Data Lake: What is the difference?  
([Slide 19](#))
2. Read the short article about recent developments in data warehousing  
“Paradigmenwechsel: Data Warehouses für die Cloud”  
(from iX 5/2020 - Magazin für professionelle Informationstechnik)  
[\*Kursmaterial > Readings/Übungen\*](#)
3. Python-Basics – Chapter 2  
[\*Kursmaterial > Readings/Übungen > Python Übungen - Jupyter Notebooks\*](#)

# Recommended reading

Lusti, M. (2002): Data Warehousing und Data Mining (esp. Chapter 5)

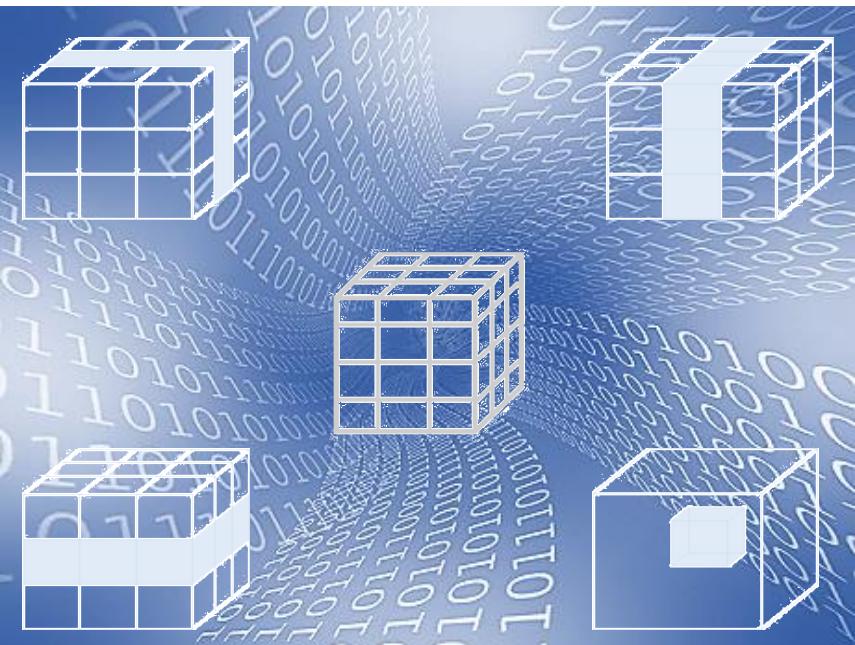
Kurz, A. (1999): Data Warehousing (esp. Chapters 1 and 4)

Inmon, W.H. (1996): Building the Data Warehouse  
(esp. Chapters 1 and 2)

<http://www.tdwi.org>

# Bibliography

- Chamoni, P., & Gluchowski, P. (2000). On-Line Analytical Processing (OLAP). *Das Data-Warehouse-Konzept. Architektur-Datenmodelle–Anwendungen*. Wiesbaden.
- Codd, E. F., Codd, S. B., & Salley, C. T. (1993). *Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate*. Codd and Date, 32.
- Kimball, Ralph, and Margy Ross. *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons, 2011.
- March, S. T., & Hevner, A. R. (2007). Integrated decision support systems: A data warehousing perspective. *Decision Support Systems*, 43(3), 1031-1043.
- Pendse, N., & Creeth, R. (1995). Succeeding with On-Line Analytical Processing. *The OLAP-Report*, 1.
- Powell, Gavin JT. *Oracle high performance tuning for 9i and 10g*. Digital Press, 2003.
- Sen, A., & Sinha, A. P. (2005). A comparison of data warehousing methodologies. *Communications of the ACM*, 48(3), 79-84.
- Stucke, Maurice E., and Allen P. Grunes. "Big Data and Competition Policy." (2016).



# Business Intelligence

## 03 Data Warehouse – OLAP & Modeling I

Prof. Dr. Bastian Amberg  
(summer term 2024)  
~~3.5.2024~~ 8.5.2024

# Schedule

|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study |                          |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|------------|--------------------------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |            |                          |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |            |                          |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  | Data Warehouse Modeling I  |            |                          |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining Introduction  |            |                          |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |            | Python-Analytics Chap. 1 |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization              |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam                 |            | Chap. 2                  |  |  |
|           | W7                | 29.5. | Data Preparation                                    |                                   | 31.5. |   |            |                          |  |  |
|           | W8                | 5.6.  | Predictive Modeling I                               |                                   | 7.6.  | Predictive Modeling II (10:00 -12:00)   |            | BI-Project Start         |  |  |
|           | W9                | 12.6. | Fitting a Model I                                   |                                   | 14.6. | Python-Analytics-Online Exercise  |            |                          |  |  |
|           | W10               | 19.6. | Guest Lecture                                       |                                   | 21.6. | Fitting a Model II  |            |                          |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |            |                          |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I  |            |                          |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |            |                          |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |            | End                      |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin ~ 22.7. bis 3.8.<br>Klausur 2.Termin ~ 23.9. bis 5.10.                                       |            | Projektbericht           |  |  |

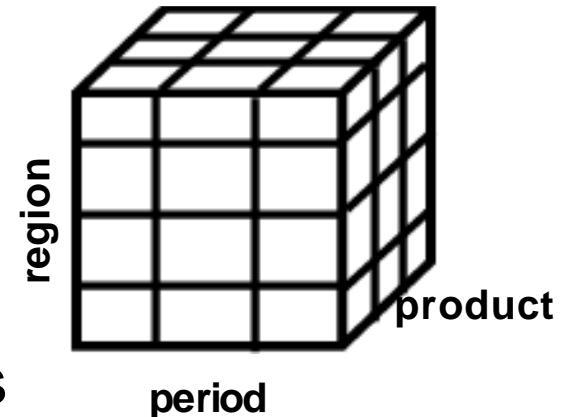
✓ Operational databases vs. Data warehouses (vs. Data lakes)

✓ Basic architecture of a data warehouse system

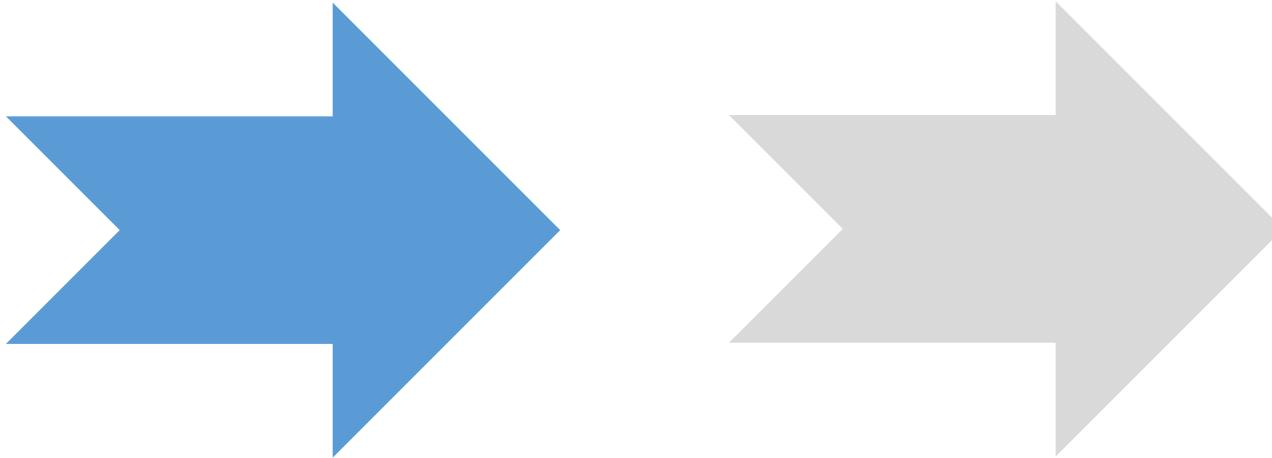
✓ Analytical data are represented by multidimensional data models  
Distinguish facts and dimensions!

○ How to extract information? (→OLAP)

○ How can multidimensional data models be developed and stored?



Kahoot-Fragen zu den Inhalten  
[www.kahoot.it](http://www.kahoot.it)  
(über Smartphone oder Laptop)  
PIN folgt



(1) Online Analytical  
Processing (OLAP)  
**Different query methods**  
Properties of OLAP  
Common OLAP functionality

(2) Modeling layers  
Basic Elements of  
multidimensional modeling  
Conceptual modeling  
Logical modeling  
Physical modeling

# Query methods

Three means to query databases

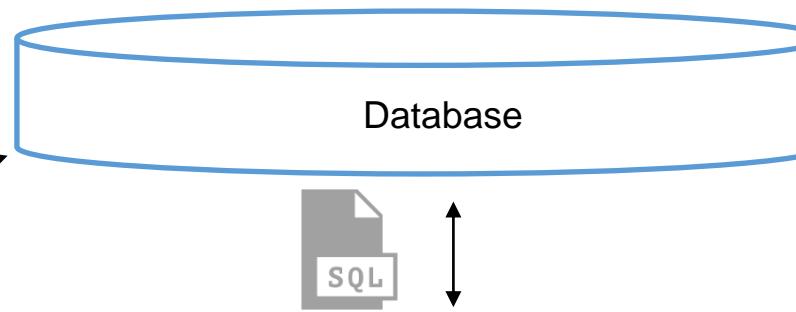


## Programmed reports

- arbitrarily modifiable
- programmer required for changes

dBase code for "Which are the properties of the products of the department ,Mobile Computing?":

```
use PRODUCTS
copy to TMP
use TMP
delete for producttype <> 'MOBILE'
total on PRODUCTS to RESULT
display all
```



Decision makers need flexible and easy access to data in order to do complex analysis

## Query languages

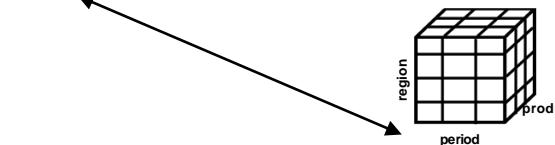
- standardized and powerful
- difficult to learn
- e.g. SQL, QBE

SQL query for "Which are the properties of the products of the department ,Mobile Computing?":

```
SELECT *
FROM Products
WHERE producttype = 'MOBILE'
```

## Using SQL for multidimensional querying is difficult:

- Several (**inner**) queries (and joins) needed in many cases
- Queries often become quite complex
- Difficult to do time series analysis
- Limited ways for doing statistical calculations

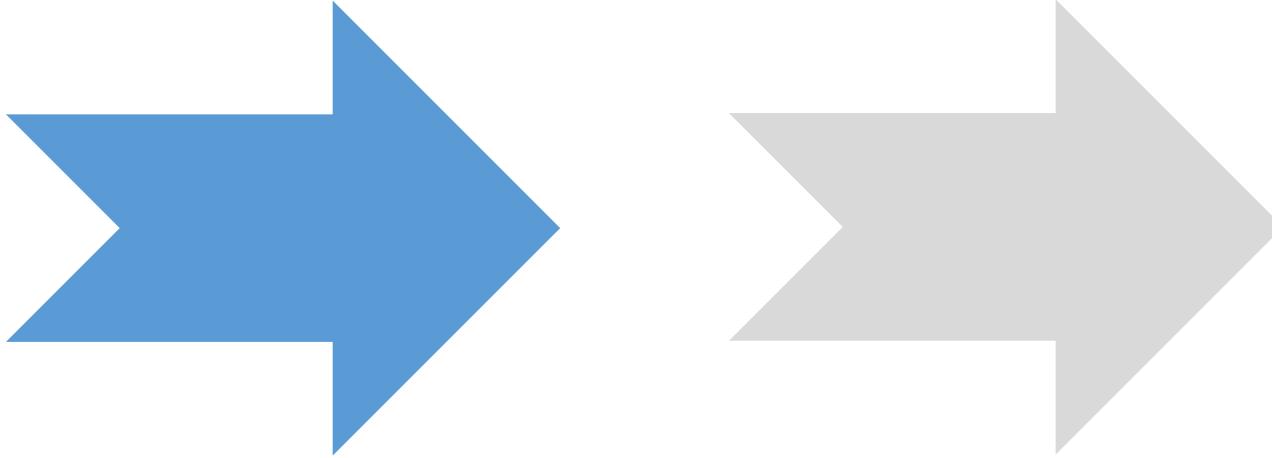


## OLAP

- flexible ad-hoc querying
- possible without expertise

SQL query for "What was the **average sales** of the department **"Mobile Computing"** to **Government customers** for the **third quarter** of calendar year 2001?"

```
SELECT customer, ROUND(AVG(sales),2) as average,
       ROUND(MIN(sales),2) as minimum, ...
  FROM units_cube_cubeview
 WHERE time_calendar_year = 'Q3_2001'
   AND product_ldsc = 'MOBILE'
   AND customer_market_segment_prnt
     = 'MARKET_SEGMENT_GOV'
   AND channel_level = 'TOTAL_CHANNEL'
 GROUP BY customer
 ORDER BY customer;
```

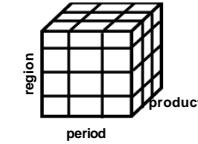


(1) Online Analytical Processing (OLAP)  
Different query methods  
**Properties of OLAP & Common OLAP functionality**

(2) Modeling layers  
Basic Elements of multidimensional modeling  
Conceptual modeling  
Logical modeling  
Physical modeling

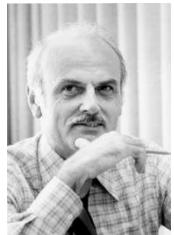
# Online Analytical Processing (OLAP)

Let's focus on end users, and their access to data marts by OLAP systems



## OLAP systems

- combine querying and interactive analysis
- present a multidimensional view on data



OLAP was introduced by E. F. Codd (one of the founding fathers of relational data bases) in 1993, who established 12 rules to define OLAP

## OLAP functionality

- video for illustration

(exemplary <https://www.youtube.com/watch?v=V37vPxIxUwo> )

A more concise definition of OLAP is **FASMI**

Ref. Codd et al. (1993), Chamoni/Gluchowski (2000); Pendse/Creeth (1995)

**Fast**

**Analysis of**

**Shared**

**Multidimensional** Truly multidimensional conceptual view of the data

**Information**

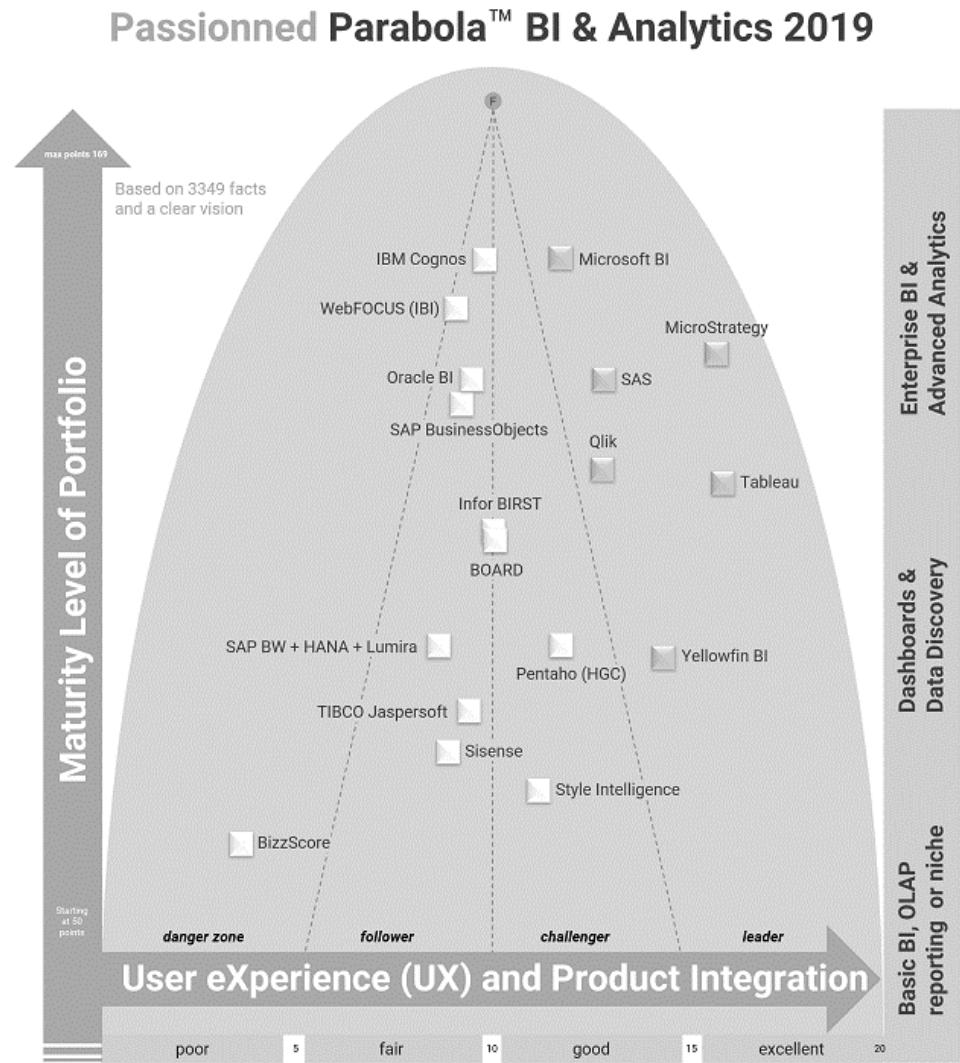


# OLAP functions

OLAP tools provide a number of standard features

- Different representation modes:
  - absolute as well as relative representation of data
  - 3-dimensional analysis using layers
  - various calculation options (internal or plug-ins)
- Special cube operators provide browsing functions:
  - drilling
    - drill up/down ⇒ detailing/aggregating along a dimension
    - drill through ⇒ access to operational databases
    - ...
  - pivoting (rotating) ⇒ switch rows and columns
  - slicing ⇒ reduce number of dimensions
  - dicing ⇒ cutting parts out of the current cube (filtering)
- Various visualization options

OLAP Tools -> part of BI Tools...



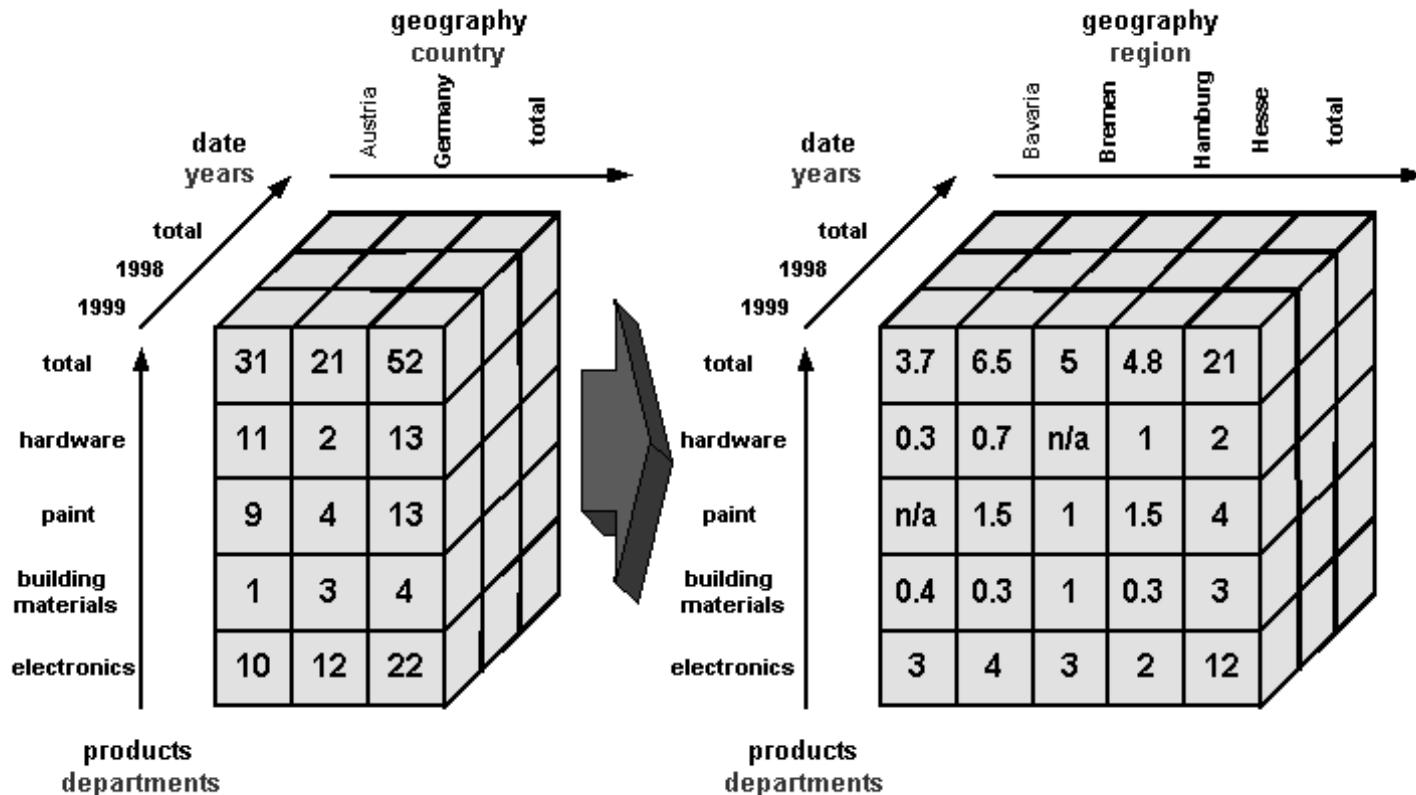
<https://www.passionned.com/bi/tools/>

See <https://www.passionned.com/bi/#list-business-intelligence-tools>  
for an up-to-date list with detailed information about BI Tools, April 2024

# Drilling down

More details for specific dimensions

“Show the [regions of Germany in detail.](#)”

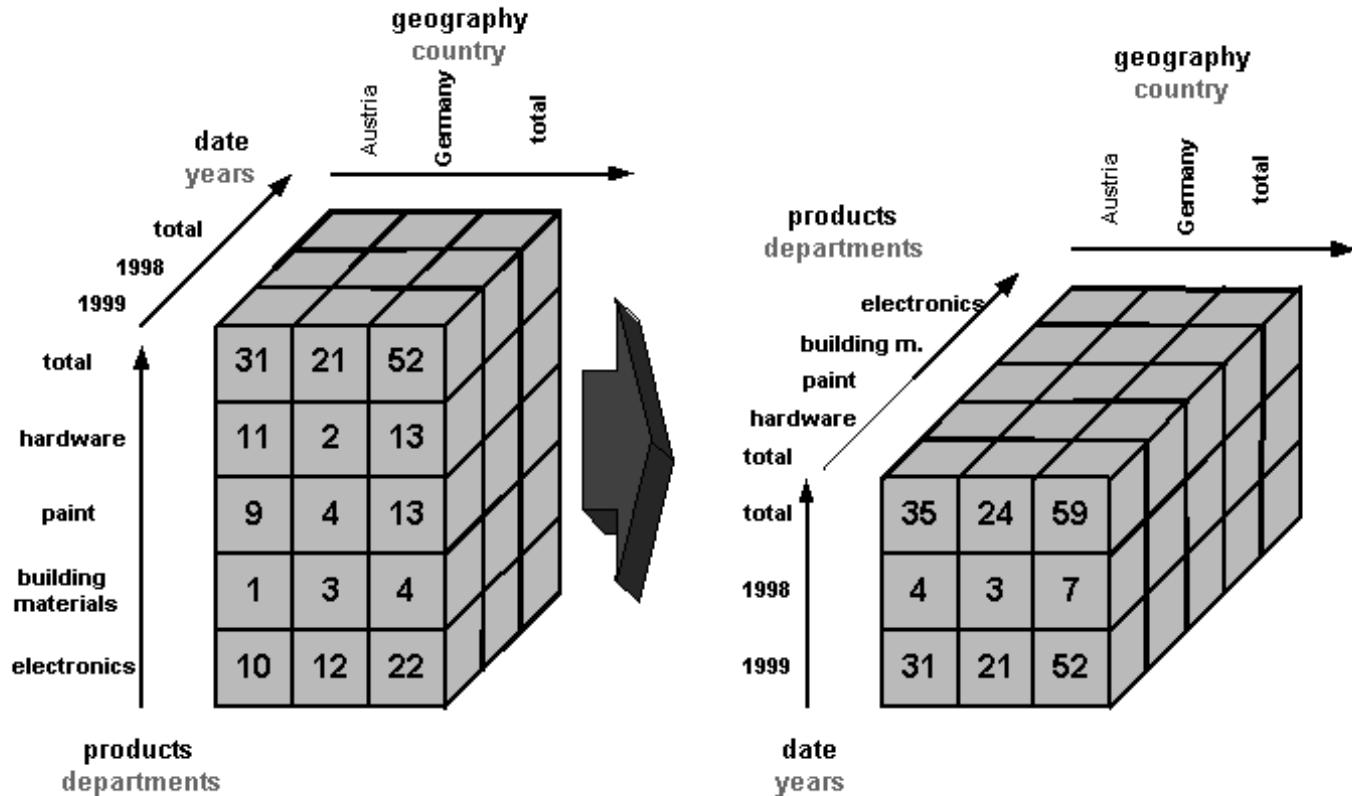


# Pivoting

Rotate the cube

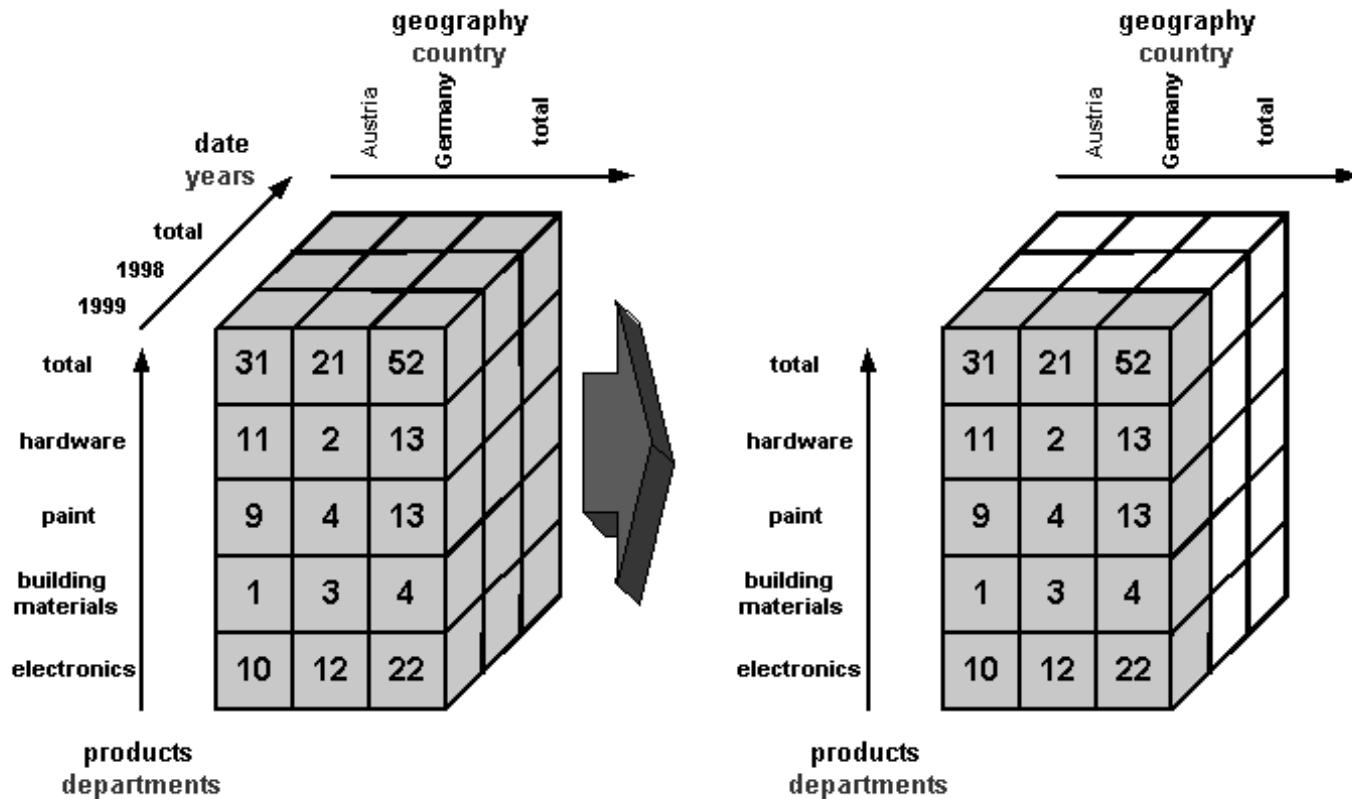


“Show year by country instead of product by country”

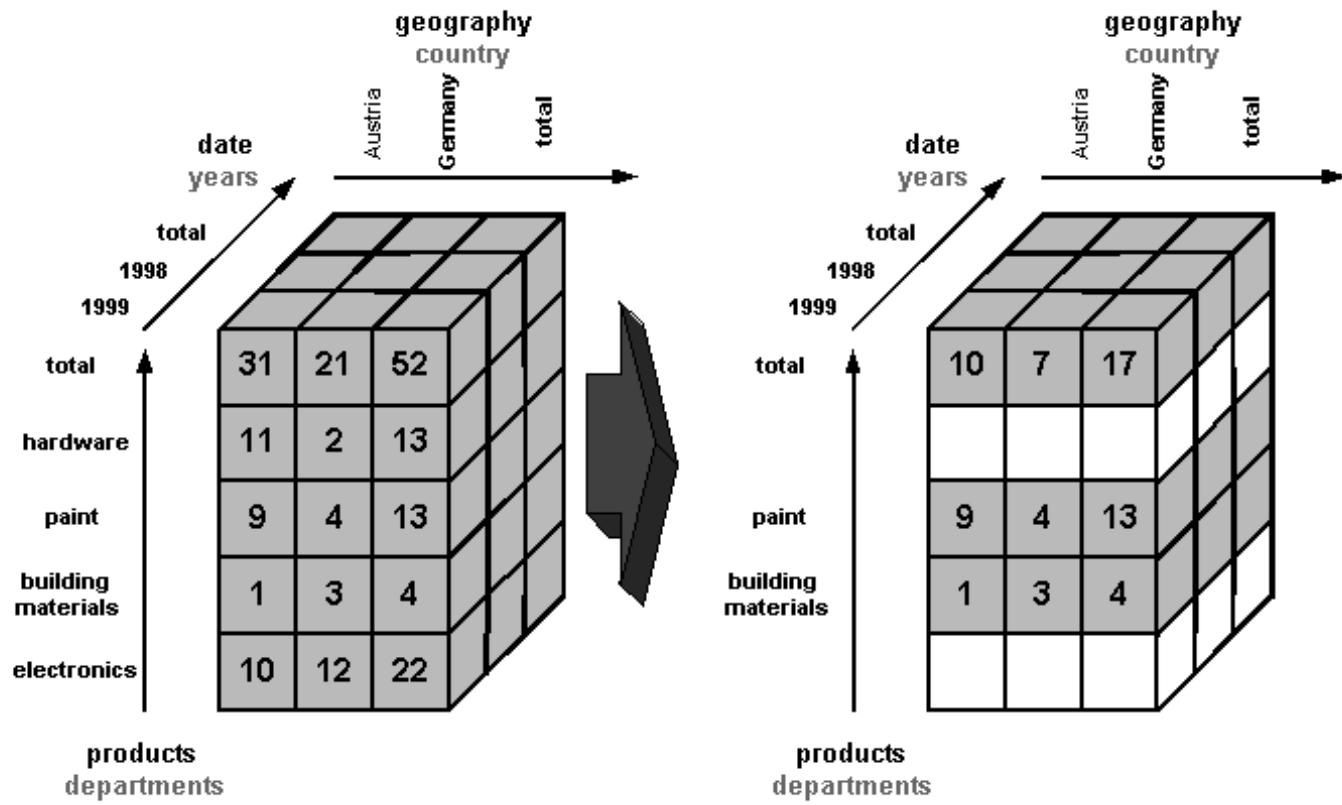


# Slicing

“Show only the values for 1999.”



“Show only the values for the departments ‘paint’ & ‘building materials’ for all the countries and all the years.”



## On-line Transactional Processing (OLTP)

- Common way of transactional processing (INSERT, UPDATE, DELETE)
- Primarily used on operational databases (day-by-day business)
- Treats microscopic transactions (e.g., by processing single accounting transactions or order transactions)
- Does not support strategic decisions, but controls and runs subsequent operations

```
66.249.76.123 - - [14/Oct/2012:03:47:21 +0200] "GET /index.php/de/ HTTP/1.1" 200 23963
178.154.211.123 - - [14/Oct/2012:03:49:28 +0200] "GET /robots.txt HTTP/1.1" 200 370
66.249.76.123 - - [14/Oct/2012:04:00:40 +0200] "GET / HTTP/1.1" 303 -
66.249.76.123 - - [14/Oct/2012:04:00:41 +0200] "GET /index.php/de/ HTTP/1.1" 200 23961
123.125.71.123 - - [14/Oct/2012:04:19:44 +0200] "GET / HTTP/1.1" 303 -
220.181.108.123 - - [14/Oct/2012:04:19:44 +0200] "GET / HTTP/1.1" 303 -
66.249.76.123 - - [14/Oct/2012:04:30:46 +0200] "GET /index.php/de/konferenzen-uebersicht/konferenzuebersicht HTTP/1.1" 200 20598
180.76.5.123 - - [14/Oct/2012:04:35:20 +0200] "GET / HTTP/1.1" 303 -
```

|                           | OLTP                     | OLAP                            |
|---------------------------|--------------------------|---------------------------------|
| <b>data</b>               | operational transactions | management analysis data        |
| <b>user friendliness</b>  | low                      | high                            |
| <b>granularity</b>        | microscopic              | macroscopic                     |
| <b>up-to-dateness</b>     | current status           | historic snapshots              |
| <b>main operations</b>    | update (read/write)      | query and calculate (read only) |
| <b>storage efficiency</b> | high                     | lower                           |
| <b>tools</b>              | e.g. SQL                 | proprietary tools               |

## ➤ OLAP vs. OLTP in a nutshell

<https://www.youtube.com/watch?v=iw-5kFzldgY>  
(IBM Technology Video, last access April 2024)

# Pros and cons of OLAP

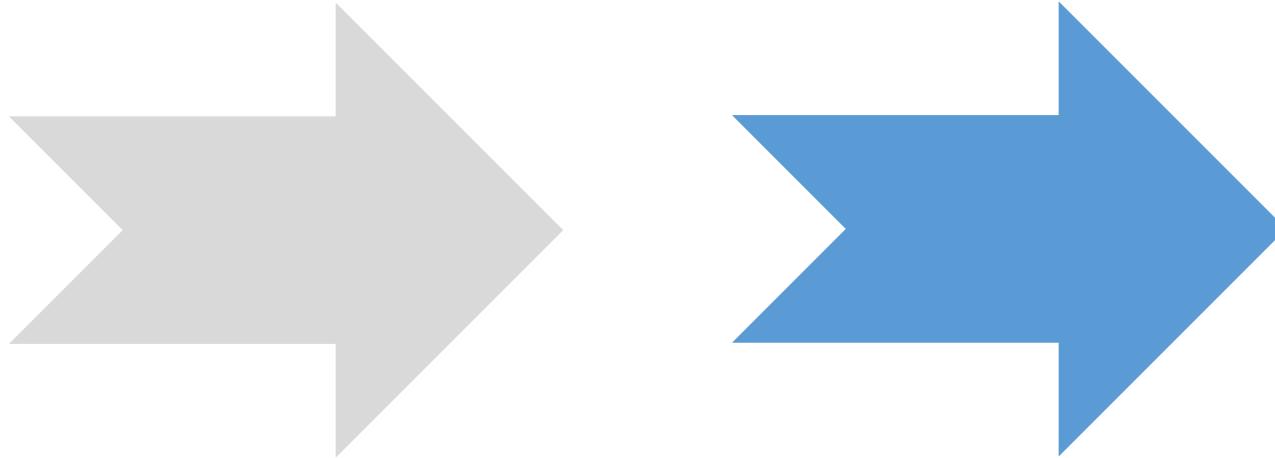
Pro:

- Wide applicability of the method
- OLAP presents quite exact results
- Method is plausible

Con:

- OLAP requires a lot of user interaction
- OLAP regularly requires quite a lot of computing resources
- Difficult to use automated data mining routines in combination with OLAP

Ref.



## (1) Online Analytical Processing (OLAP)

Different query methods

Properties of OLAP

Common OLAP functionality

## (2) Modeling layers

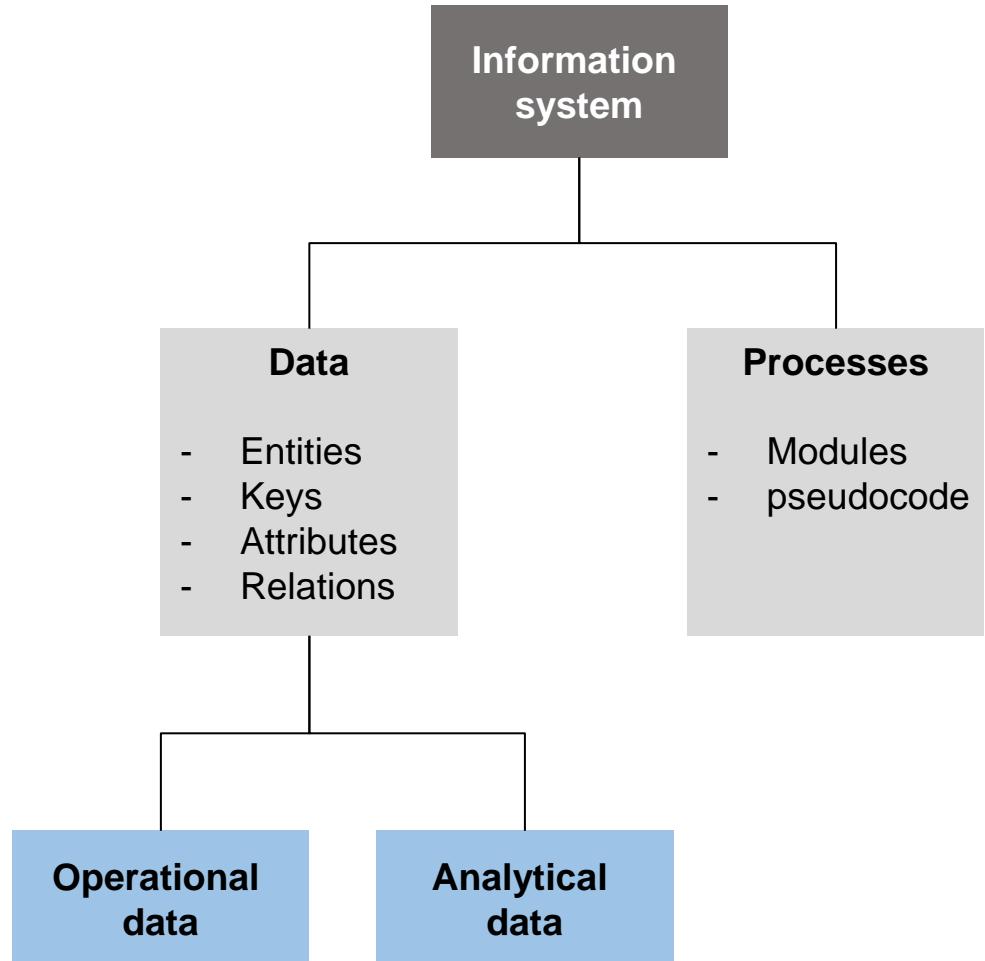
**Basic Elements of multidimensional modeling**

Conceptual modeling

Logical modeling

Physical modeling

# Modeling of information systems



## Operational databases

- Optimize storage efficiency and response time
  - Model data which is
    - fine-grained (many details)
    - dynamic (many updates)
- 
- Normalize data
  - Minimize redundancy
  - Provide data integrity
    - Avoid update anomalies
    - Avoid deletion anomalies
    - Avoid insertion anomalies

## Analytical databases

- Support the decision making process
  - Maximize user-friendliness and querying efficiency
  - Model data which is
    - coarse-grained (less details)
    - static (less updates)
- 
- Data is denormalized
  - Redundancy minimization is secondary

→ *Mirror different views on business measures within the model*



Image: [CTSI-Global](#) | Flickr (cc by-sa 2.0)

# Multidimensional modeling

## Basic Elements

### Common steps compared to operational databases

Leave out operational data  
(not all attributes necessary)

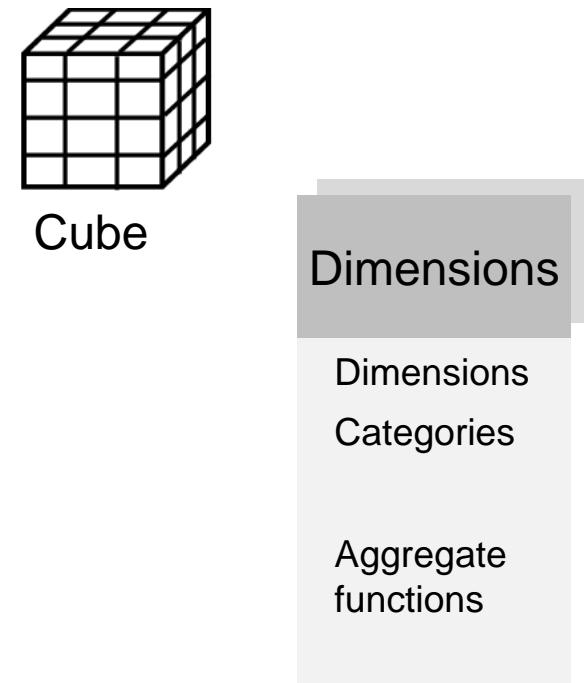
Include time dimension

Integrate pre-calculated attributes

Reduce join operations

### Basic elements of multidimensional models

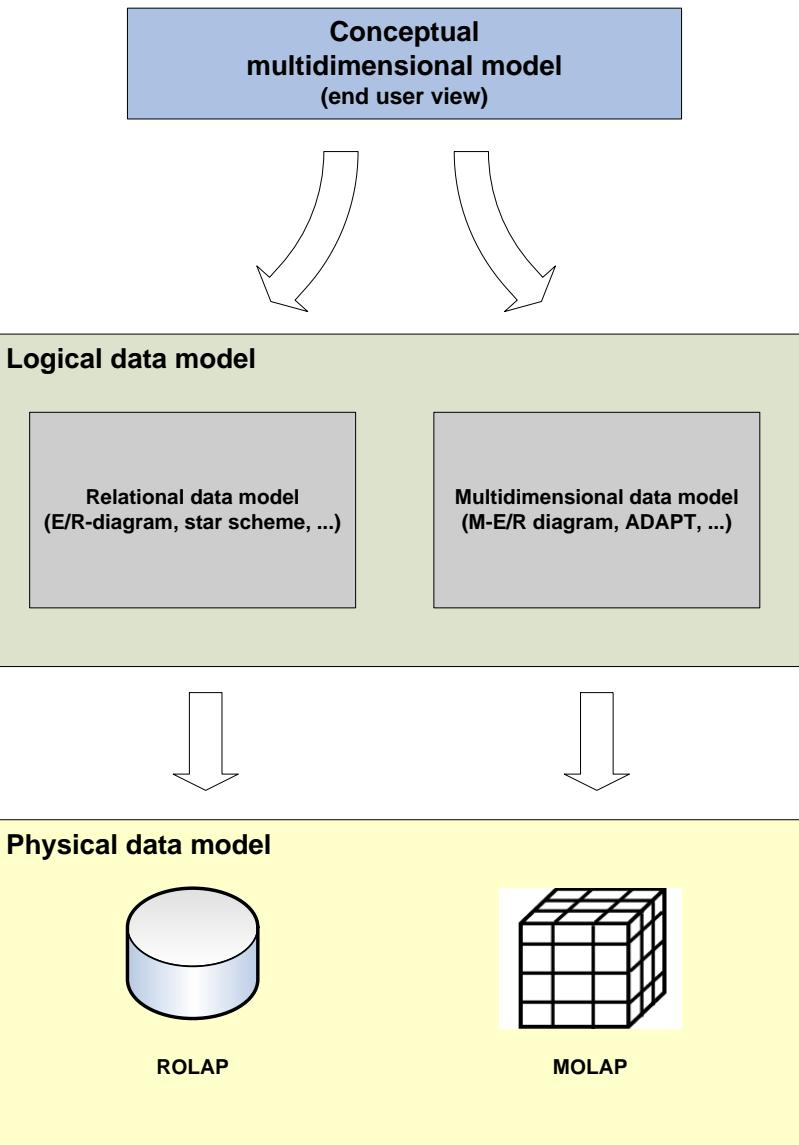
Facts  
Dimensions  
Categories  
Aggregation functions



# Multidimensional modeling

## Major steps

1. Identify **facts and dimensions**
2. Create a **conceptual** data model
3. Derive a **logical** data model from the semantic model
4. Derive a **physical** data model from the logical model



Multidimensional models are designed according to the needs of decision makers

- Business measures are in the center of interest of decision makers

**Definition** of business measure:

*"Business measures are compressed mostly numeric measurements, which refer to **important matters of fact** within the company and which represent them in a **concentrated** manner. They provide **information about business issues** and thereby provide **important support** for the decision processes within the company."*

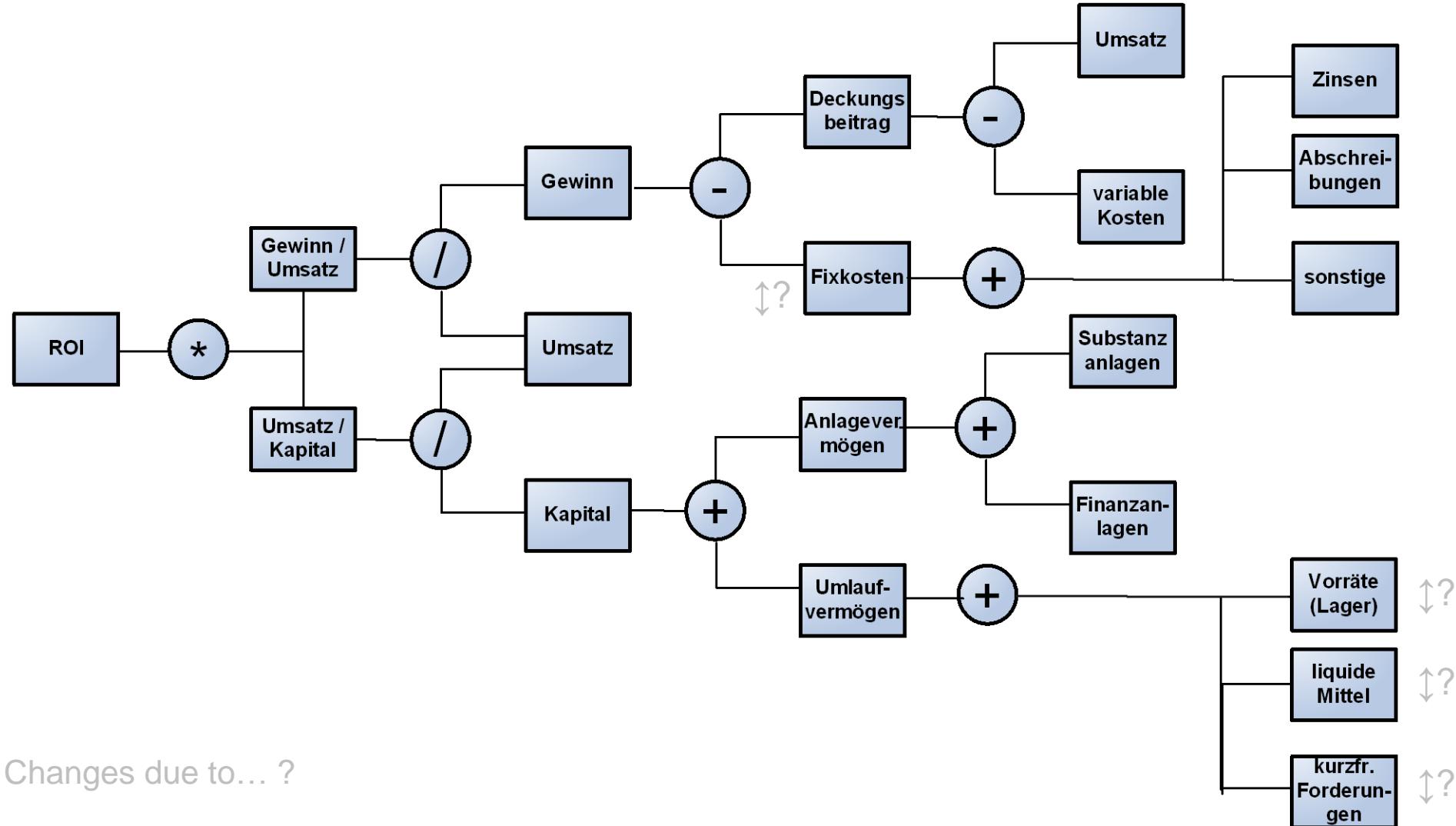
(Langenbeck, 1997, highlights added)

Example business measures: revenues, profits, sales, ROI ...

**Identification of business measures** is one of the basic tasks in multidimensional modeling

# Example business measure system: ROI

~ „Erfolg im Verhältnis zum eingesetzten Kapital“, „Gewinn in Prozent des investierten Kapitals“, ...



Changes due to... ?

Ref.

Decision makers want to **analyze business measures** from **different views** (dimensions)

- Several dimensions are arranged around one fact

*“What amount were the sales revenues for hard disks within the past quarter?”*

*fact: sales revenues*

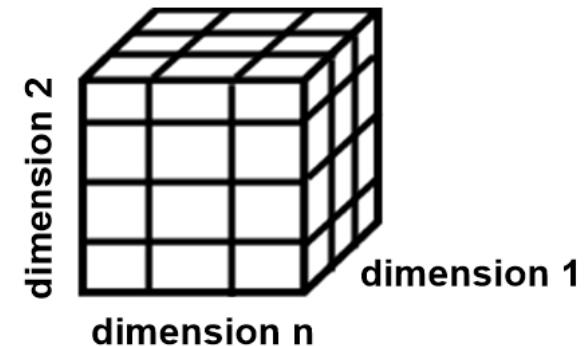
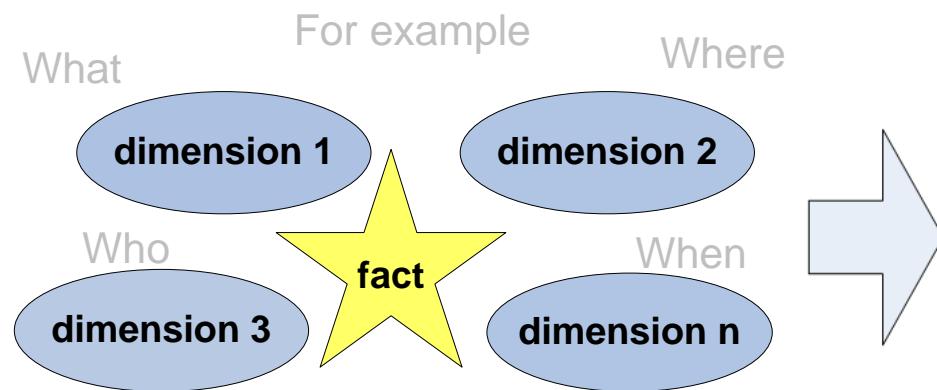
*dimensions: range of products, time*

*“How profitable has our Africa department been on software?”*

*“What is our growth on A-customers throughout the last quarter?”*

During the modeling process, business measures and their set of dimensions are determined

Link to multi-dimensional data structures:



# Dimensions and categories

*Dimension = finite set of categories* which are semantically related to each other with respect to business matters

Categories of one dimension represent a **different levels of aggregation** of the associated business *measures* (facts)  
Categories are also known as aggregation objects

## An example

dimension: “date”

Four categories: day  $\Rightarrow$  month  $\Rightarrow$  quarter  $\Rightarrow$  year

Resp.: “Sales revenues for hard disks within the past day, month, quarter, year, ...?”

# Categories

A category is represented by a varying set of elements

e.g., country = [Germany, Austria, Switzerland], quarter = [q1, q2, q3, q4]

Each dimension consists of **at least one** (real) category

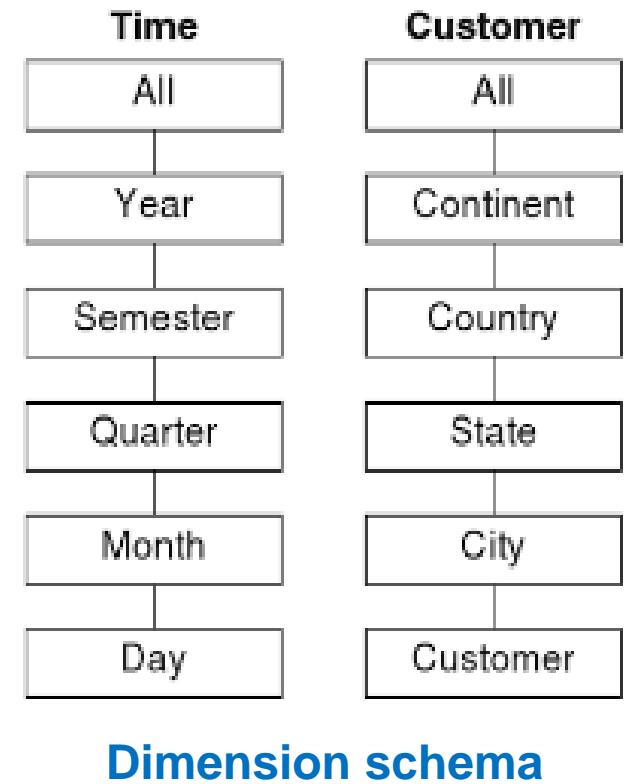
a category represents the level of granularity

**“Log-category”** (e.g., “day” in dimension “date”)

a virtual category, encompassing all the others, like

**“all-category”**: encompasses all elements of the Log-category

Number of categories of a dimension is not limited



# Facts and aggregation functions

**Aggregation function** = formula defining the value of facts with respect to the different categories of a dimension

Facts can be classified with respect to aggregation functions:

## Additive facts

(distributive aggregation function)

Simple addition possible throughout all the categories of all the associated dimensions

e.g., units sold

## Semi-additive facts

(algebraic aggregation function)

Simple addition only possible for a selected number of the categories of the associated dimensions

e.g., not additive over time, but maybe over regions

e.g., current stock/ inventory level, current balance amount

## Non-additive facts

(holistic aggregation function)

Simple addition operations not sufficient

e.g., types of average values or ratio values

e.g., temperature

# Special types of facts

Fact groups, dimensional and virtual facts

## Fact group

**set of facts** featuring a *common* set of dimensions

e.g., *units sold* and *sales in \$ per day*

| Sales        |
|--------------|
| - units sold |
| - in \$      |

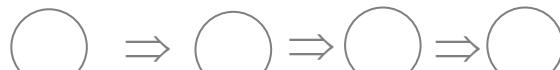
## Dimensional fact:

fact is associated with **only one dimension**

dimensional facts are often numerical, non-dimensional attributes of a dimension's category

e.g., sales area (dimension "geography")

geography country state store



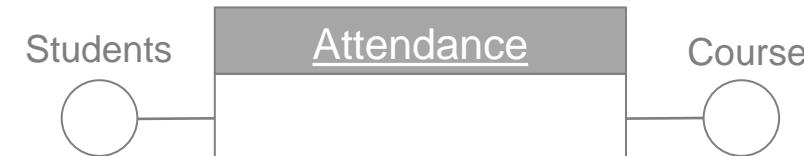
(sales area, address, etc)

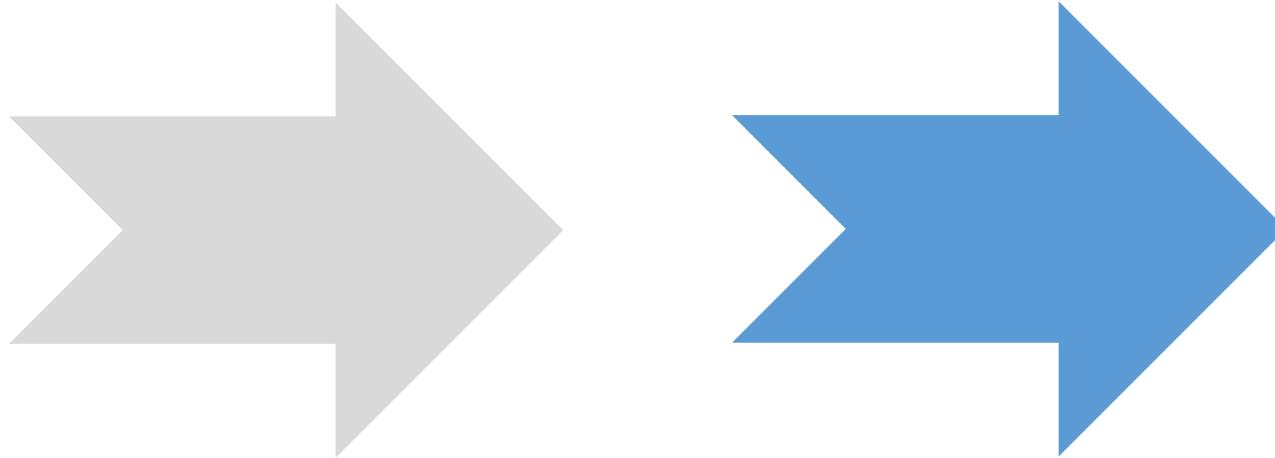
## Virtual fact (a.k.a. "factless fact"):

**Association between dimensions** alone defines the (nominal) fact

e.g., students attendance in class (students, class, time) – virtual fact (0/1) to ask for: *how many students attended class x?*

Relational implementation: only keys in fact table





## (1) Online Analytical Processing (OLAP)

Different query methods

Properties of OLAP

Common OLAP functionality

## (2) Modeling layers

Basic Elements of multidimensional modeling

**Conceptual modeling**

Logical modeling

Physical modeling

# Conceptual Modeling

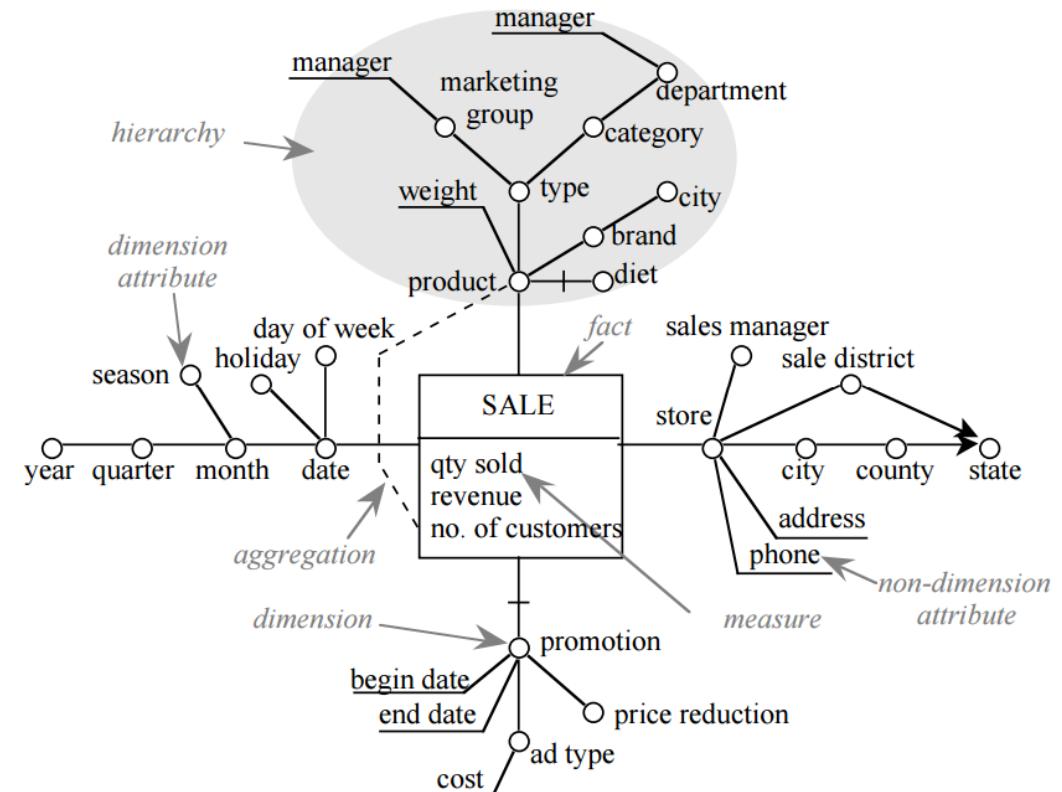
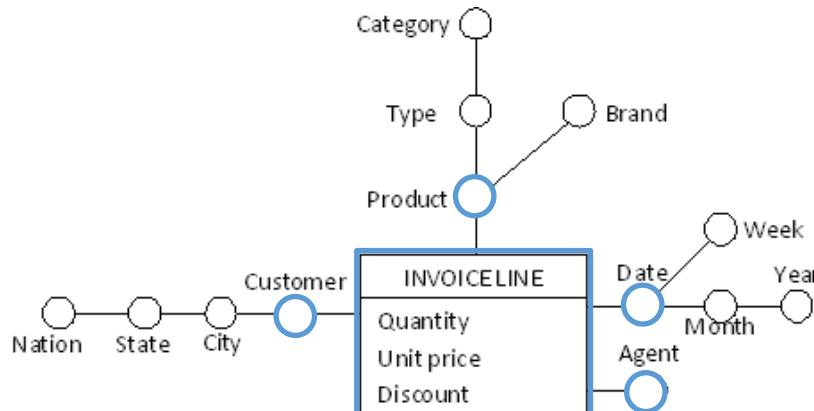
Dimensional fact model (or fact scheme)

Categories of a dimension arranged in a non-cyclic graph, directed between all-category and log-categories

Categories can have an arbitrary number of (non recursive) relations between each other

Several aggregation paths (e.g., sum/count/mean) may be included in the graph

Hierarchies are discrete attributes and define the granularities of facts (i.e., product -> type -> category)



# Exercise

## Conceptual Modeling (Dimensional fact model)



Design a **conceptual model** for the local Food Company:

### Conny's Corner Shop

Your managers need to keep themselves up to date on the number of items in the company's inventory. They especially want to keep an eye on their products with regards to location, and time.

- Conny's Corner Shop sells a range of snacks and beverages. Both categories have different types of products, such as juices and water, as well as pretzels and crackers.
- The products are sold under different brands.
- Products have different package types, sizes, and weights.
- They store products in different stores across Europe

10 Min.

Show your conceptual model as a dimensional fact model. Make reasonable assumptions if necessary.

# Non-& Cross-dimensional attributes

Dimensional fact model

There may be **various types of relations** between individual categories of one (or more) dimension(s)

Categories having 1:1 relations can be summarized into a single category

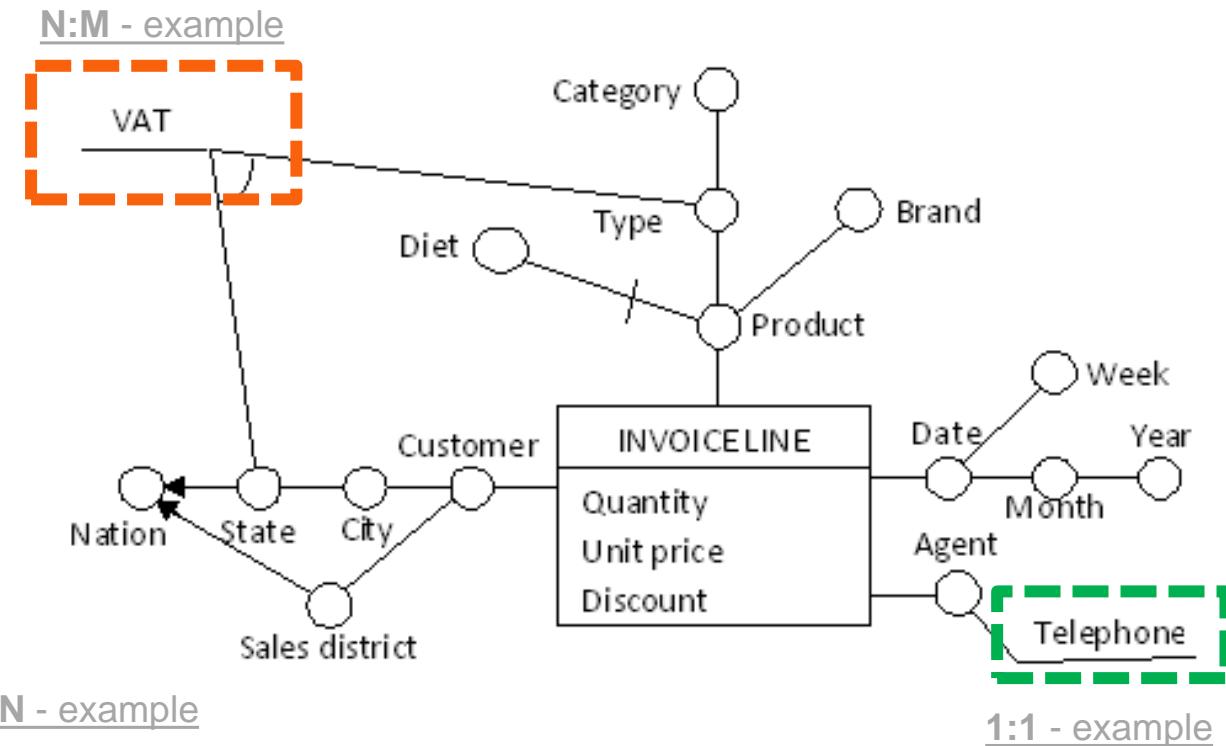
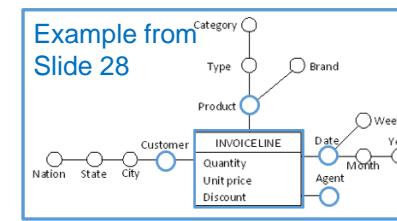
When previous categories become non-dimensional, **descriptive attributes**

Non-dimensional attributes provide additional information and have 1:1 relations with the corresponding categories (phone-No. cannot be aggregated)

OLAP-compliant queries encompassing non dimensional attributes are generally not supported

Categories having **1:N** or **N:M** relations

When value is defined by multiple categories (e.g., product type and state. For instance, VAT for water or books in Germany vs. USA) they become **cross-dimensional attributes**



1:N - example

Nation ← State ← City ← Customer

1:N      1:N      1:N

1:1 - example

# Identifying fact groups

## Single Star Scheme

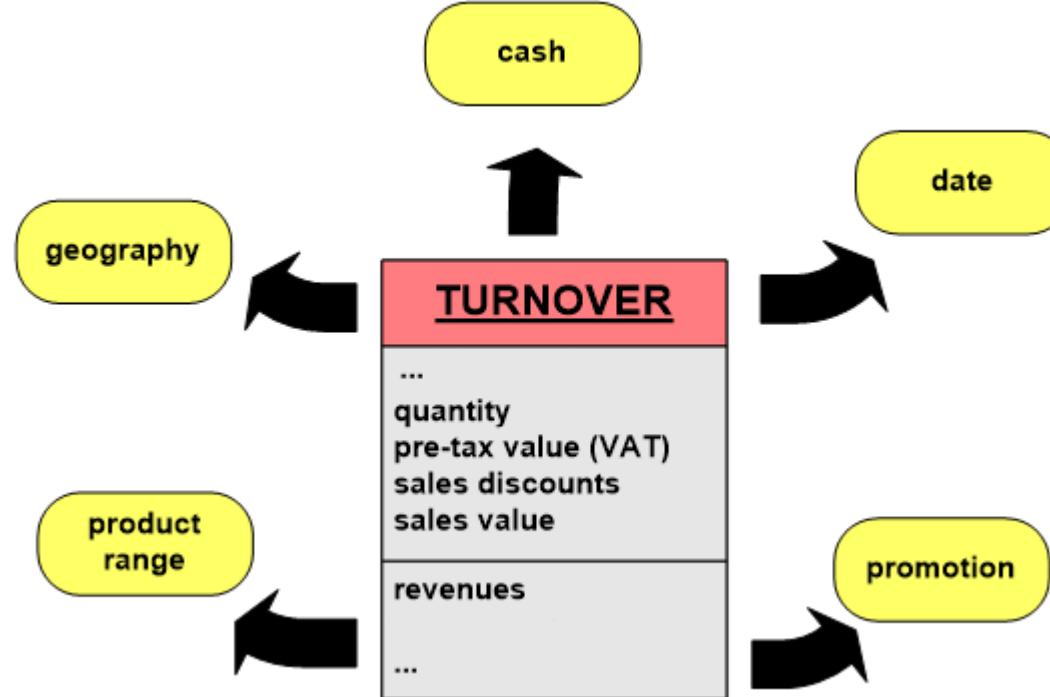


A graphical overview should be created for each fact group

### single star scheme

Distinction between materialized facts and derived facts should be drawn

High level of abstraction:  
aggregation formulas are commonly not modeled



Remember: Fact group = set of facts  
featuring a *common* set of dimensions

# Combining fact groups

## Multiple Star Scheme

A data warehouse data model encompasses a number of fact groups (**multiple star scheme**)

The sets of associated dimensions of different fact groups may overlap.

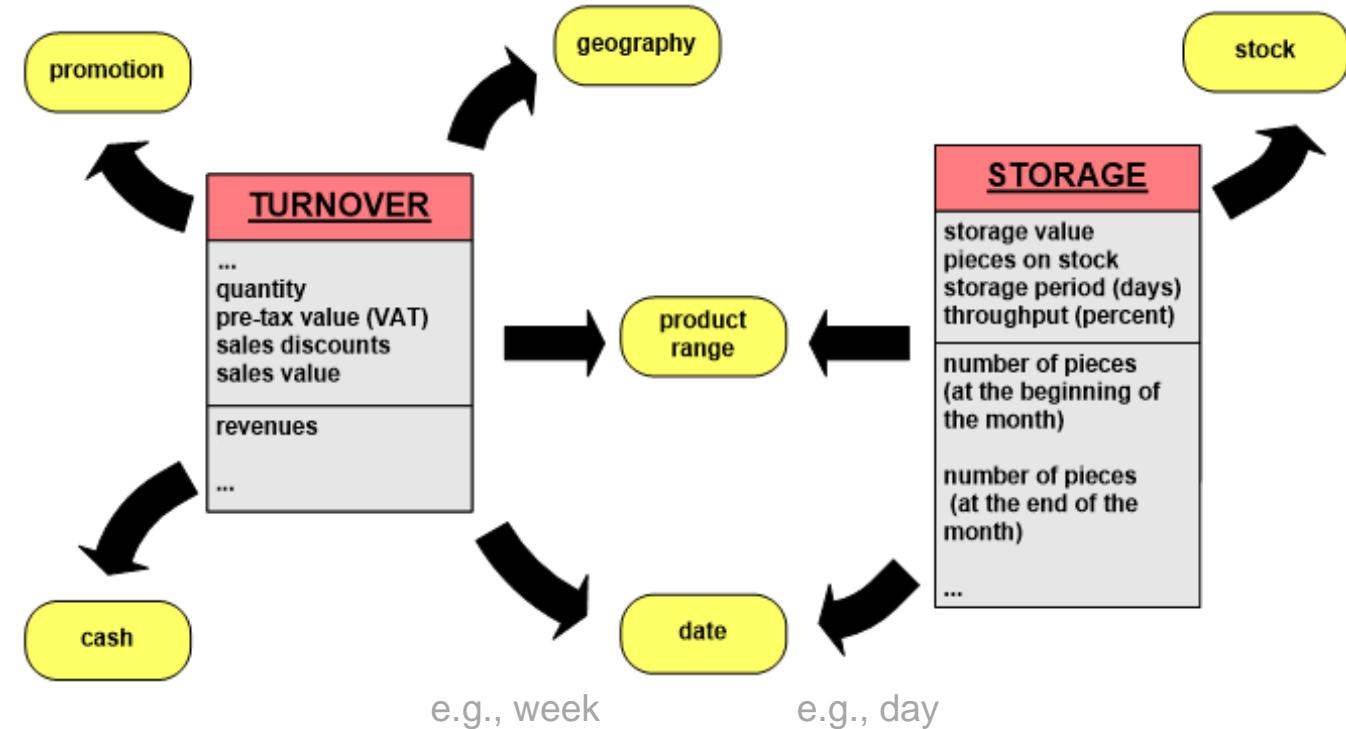
Different fact groups may use different log-categories with respect to one common dimension

e.g., actual / debit values:

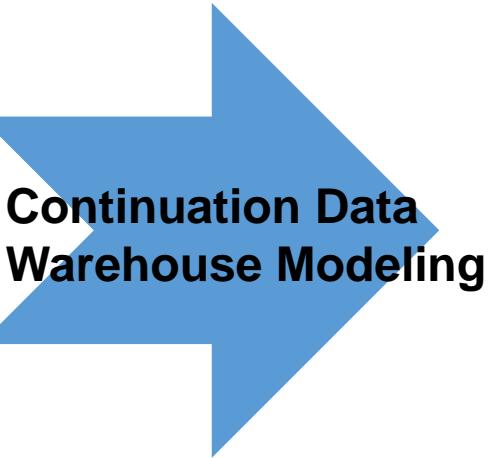
Actual facts: log-category of dimension date: day

Target facts: log-category of dimension date: month

Category used as log-category should be specified in the model (at least if standard log-category is not used)



# Next Lesson

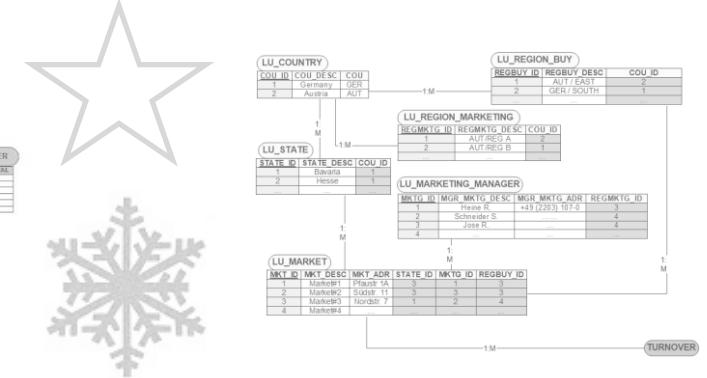
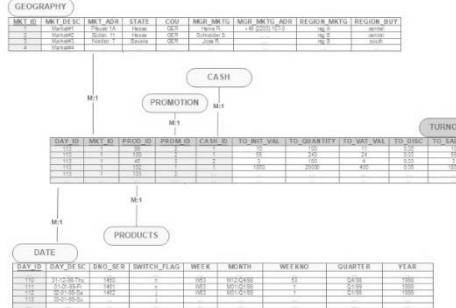
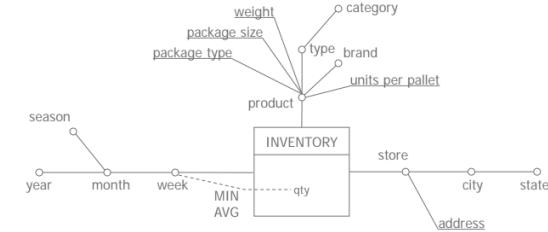


**Continuation Data Warehouse Modeling**

Identify facts and dimensions  
Create a conceptual data model

**Derive a logical data model  
from the semantic model**

**Derive a physical data model  
from the logical model**



## Fragen?

- ✓ Online Analytical Processing (OLAP)
  - ✓ Different query methods
  - ✓ Properties of OLAP
  - ✓ Common OLAP functionality
  
- ✓ Modeling layers
  - ✓ Basic Elements of multidimensional modeling
  - ✓ Conceptual modeling
    - Logical modeling
    - Physical modeling

# Todos for this Week

1. Support Conny's Corner Shop by finishing the conceptual model.

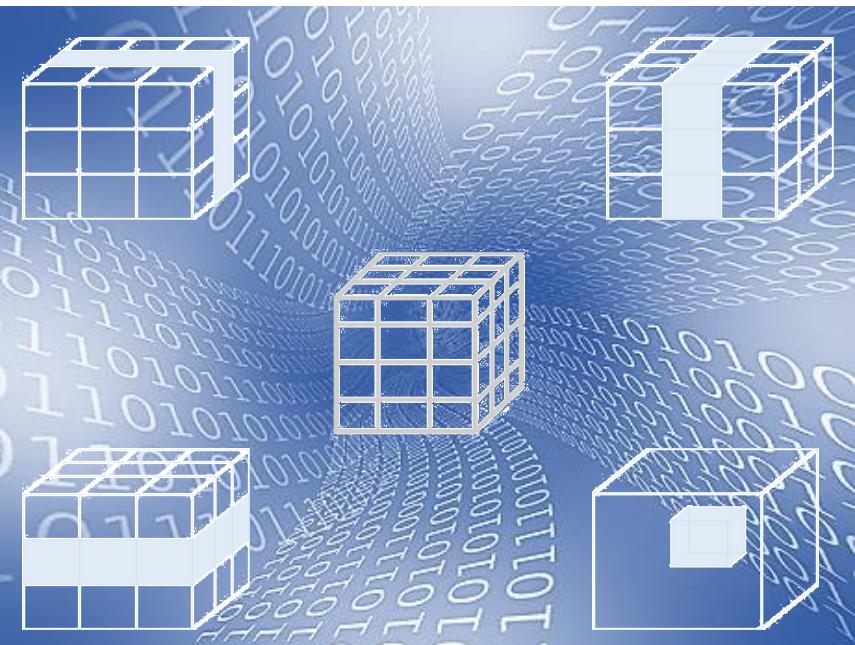
*See exercise on slide 29*

2. Python-Basics – Chapter 3

*Kursmaterial > Readings/Übungen > Python Übungen – Jupyter*

# Bibliography

- Böhnlein, M. (2013). *Konstruktion semantischer Data-Warehouse-Schemata*. Springer-Verlag.
- Bulos, D., & Forsman, S. (2000). *Olap Database Design: Delivering on the Promise of the Data Warehouse*. Morgan Kaufmann Publishers Inc..
- Golfarelli, M., Maio, D., & Rizzi, S. (1998). The dimensional fact model: A conceptual model for data warehouses. *International Journal of Cooperative Information Systems*, 7(02n03), 215-247.
- Hahne M. (2006) Mehrdimensionale Datenmodellierung für analyseorientierte Informationssysteme. In: Chamoni P., Gluchowski P. (eds) Analytische Informationssysteme. Springer, Berlin, Heidelberg
- Jukic, N., Jukic, B., & Malliaris, M. (2008). Online analytical processing (OLAP) for decision support. In Handbook on Decision Support Systems 1 (pp. 259-276). Springer, Berlin, Heidelberg.
- Vaisman, A., & Zimányi, E. (2014). *Data Warehouse Systems*. Springer, Heidelberg



# Business Intelligence

## 03 Data Warehouse – OLAP & Modeling I

Prof. Dr. Bastian Amberg  
(summer term 2024)  
3.5.2024

# Schedule

|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |   | Self-study       |         |
|-----------|-------------------|-------|---|-----------------------------------|---|------------------|---------|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 | 19.4.                             |   | Python-Basics    | Chap. 1 |
|           | W2                | 24.4. | Data Warehouse – Overview & OLAP                    | 26.4.                             | [Blockveranstaltung SE Prof. Gersch]  |                  | Chap. 2 |
|           | W3                | 1.5.  |   | 3.5.                              | Data Warehouse Modeling I   |                  | Chap. 3 |
|           | W4                | 8.5.  | Data Warehouse Modeling II                          | 10.5.                             | Data Mining Introduction  |                  |         |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     | 17.5.                             | Python-Basics-Online Exercise   | Python-Analytics | Chap. 1 |
|           | W6                | 22.5. | Data Understanding, Data Visualization              | 24.5.                             | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam |                  | Chap. 2 |
|           | W7                | 29.5. | Data Preparation                                    | 31.5.                             |   |                  |         |
|           | W8                | 5.6.  | Predictive Modeling I                               | 7.6.                              | Predictive Modeling II (10:00 -12:00)   | BI-Project       | Start   |
|           | W9                | 12.6. | Fitting a Model I                                   | 14.6.                             | Python-Analytics-Online Exercise  |                  |         |
|           | W10               | 19.6. | Guest Lecture                                       | 21.6.                             | Fitting a Model II  |                  |         |
|           | W11               | 26.6. | How to avoid overfitting                            | 28.6.                             | What is a good Model?   |                  |         |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities | 5.7.                              | Similarity (and Clusters)<br>From Machine to Deep Learning I                                  |                  |         |
|           | W13               | 10.7. |   | 12.7.                             | From Machine to Deep Learning II  |                  |         |
|           | W14               | 17.7. | Project presentation                                | 19.7.                             | Project presentation  |                  | End     |
| Ref.      |                   |       |   |                                   | Klausur 1.Termin ~ 22.7. bis 3.8.<br>Klausur 2.Termin ~ 23.9. bis 5.10.                       | Projektbericht   |         |

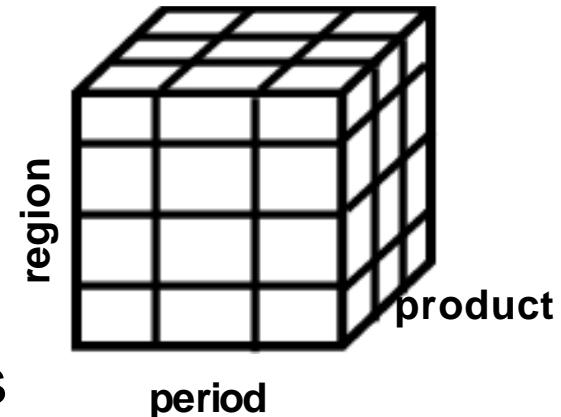
✓ Operational databases vs. Data warehouses (vs. Data lakes)

✓ Basic architecture of a data warehouse system

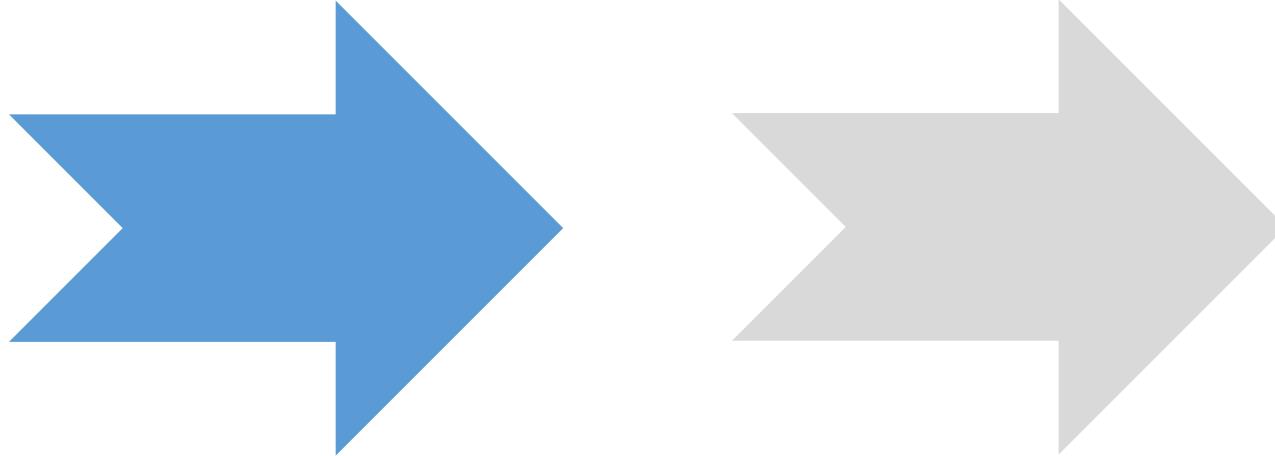
✓ Analytical data are represented by multidimensional data models  
Distinguish facts and dimensions!

○ How to extract information? (→OLAP)

○ How can multidimensional data models be developed and stored?



Kahoot-Fragen zu den Inhalten  
[www.kahoot.it](http://www.kahoot.it)  
(über Smartphone oder Laptop)  
PIN folgt

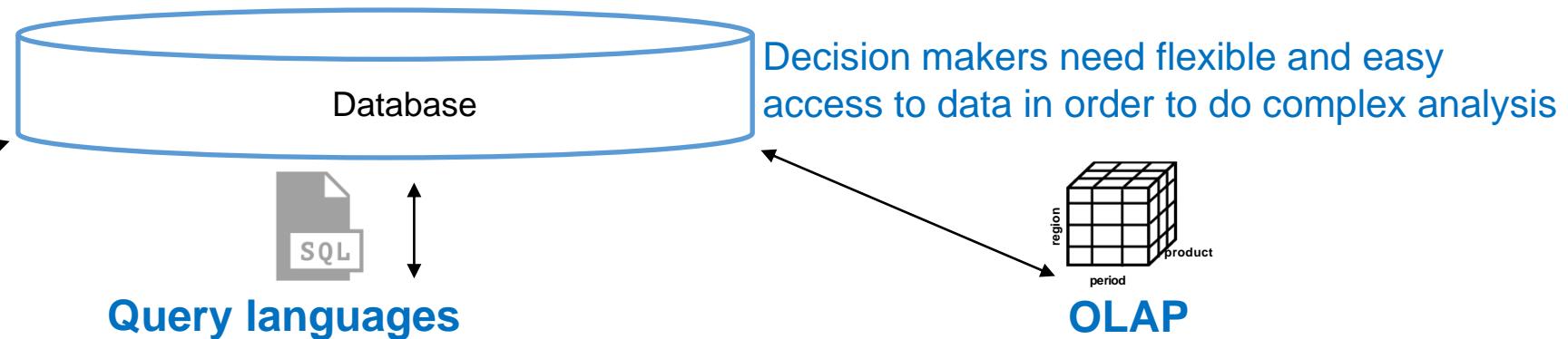


(1) Online Analytical  
Processing (OLAP)  
**Different query methods**  
Properties of OLAP  
Common OLAP functionality

(2) Modeling layers  
Basic Elements of  
multidimensional modeling  
Conceptual modeling  
Logical modeling  
Physical modeling

# Query methods

Three means to query databases



## Programmed reports

- arbitrarily modifiable
- programmer required for changes

dBase code for "Which are the properties of the products of the department ,Mobile Computing?":

```
use PRODUCTS
copy to TMP
use TMP
delete for producttype <> 'MOBILE'
total on PRODUCTS to RESULT
display all
```

## Query languages

- standardized and powerful
- difficult to learn
- e.g. SQL, QBE

SQL query for "Which are the properties of the products of the department ,Mobile Computing?":

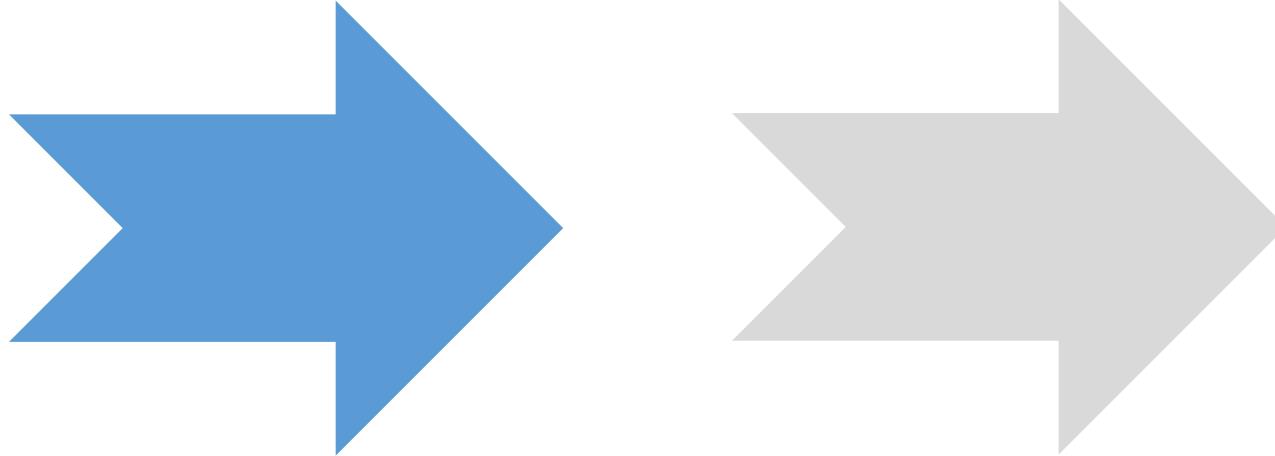
```
SELECT *
FROM Products
WHERE producttype = 'MOBILE'
```

## Using SQL for multidimensional querying is difficult:

- Several **(inner) queries** (and joins) needed in many cases
- Queries often become quite complex
- Difficult to do time series analysis
- Limited ways for doing **statistical calculations**

SQL query for "What was the **average sales** of the department **"Mobile Computing"** to **Government customers** for the **third quarter** of calendar year 2001?"

```
SELECT customer, ROUND(AVG(sales),2) as average,
       ROUND(MIN(sales),2) as minimum, ...
  FROM units_cube_cubeview
 WHERE time_calendar_year = 'Q3_2001'
   AND product_ldsc = 'MOBILE'
   AND customer_market_segment_prnt
     = 'MARKET_SEGMENT_GOV'
   AND channel_level = 'TOTAL_CHANNEL'
 GROUP BY customer
 ORDER BY customer;
```

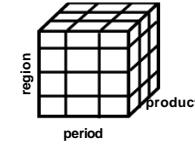


(1) Online Analytical Processing (OLAP)  
Different query methods  
**Properties of OLAP & Common OLAP functionality**

(2) Modeling layers  
Basic Elements of multidimensional modeling  
Conceptual modeling  
Logical modeling  
Physical modeling

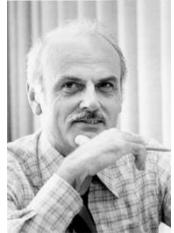
# Online Analytical Processing (OLAP)

Let's focus on end users, and their access to data marts by OLAP systems



## OLAP systems

- combine querying and interactive analysis
- present a multidimensional view on data



OLAP was introduced by E. F. Codd (one of the founding fathers of relational data bases) in 1993, who established 12 rules to define OLAP

## OLAP functionality

- video for illustration

(exemplary <https://www.youtube.com/watch?v=V37vPxIxUwo> )

A more concise definition of OLAP is **FASMI**

**Fast**

**Analysis of**

**Shared**

**Multidimensional** Truly multidimensional conceptual view of the data

**Information**

OLAP systems deliver responses to analyze queries within seconds (ideally maximum 5 – 20 seconds)

Cope with any business logic and statistical analysis that is relevant to the user: Mathematic modeling, time series analysis, goal seeking, what-if, drill-down etc., but no programming

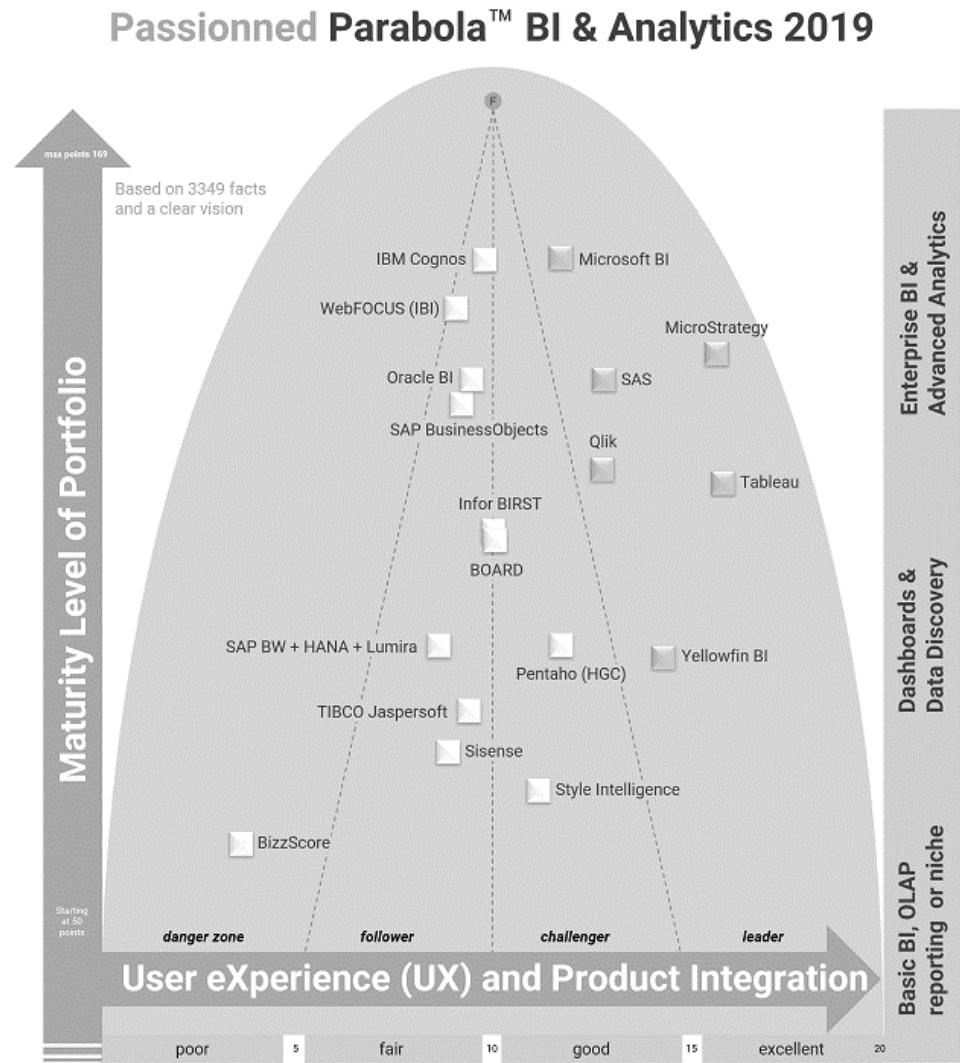
Multiple user access and varying roles with necessary security requirements for confidentiality.

# OLAP functions

OLAP tools provide a number of standard features

- Different representation modes:
  - absolute as well as relative representation of data
  - 3-dimensional analysis using layers
  - various calculation options (internal or plug-ins)
- Special cube operators provide browsing functions:
  - drilling
    - drill up/down ⇒ detailing/aggregating along a dimension
    - drill through ⇒ access to operational databases
    - ...
  - pivoting (rotating) ⇒ switch rows and columns
  - slicing ⇒ reduce number of dimensions
  - dicing ⇒ cutting parts out of the current cube (filtering)
- Various visualization options

OLAP Tools -> part of BI Tools...



<https://www.passionned.com/bi/tools/>

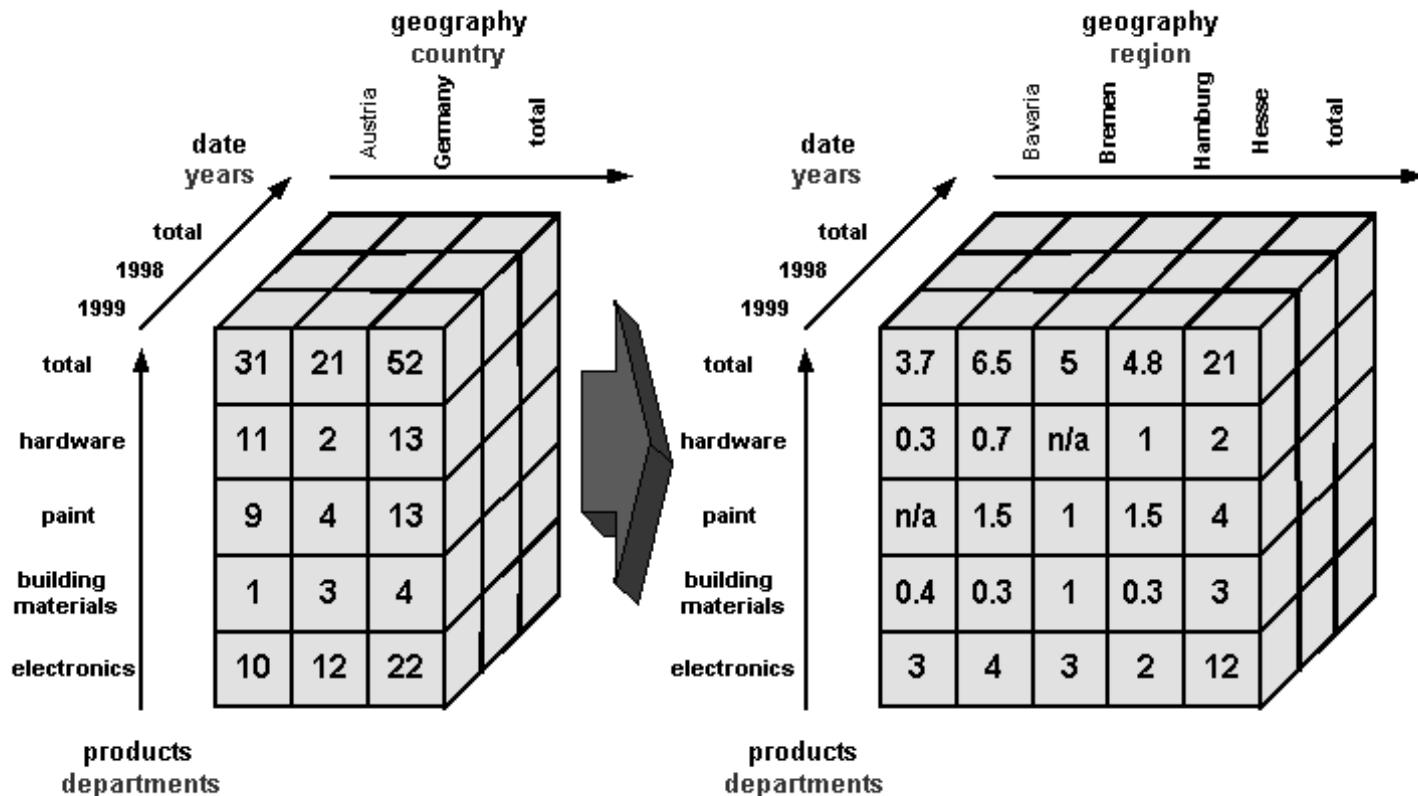
See <https://www.passionned.com/bi/#list-business-intelligence-tools>  
for an up-to-date list with detailed information about BI Tools, April 2024

# Drilling down

More details for specific dimensions



“Show the [regions of Germany in detail.](#)”

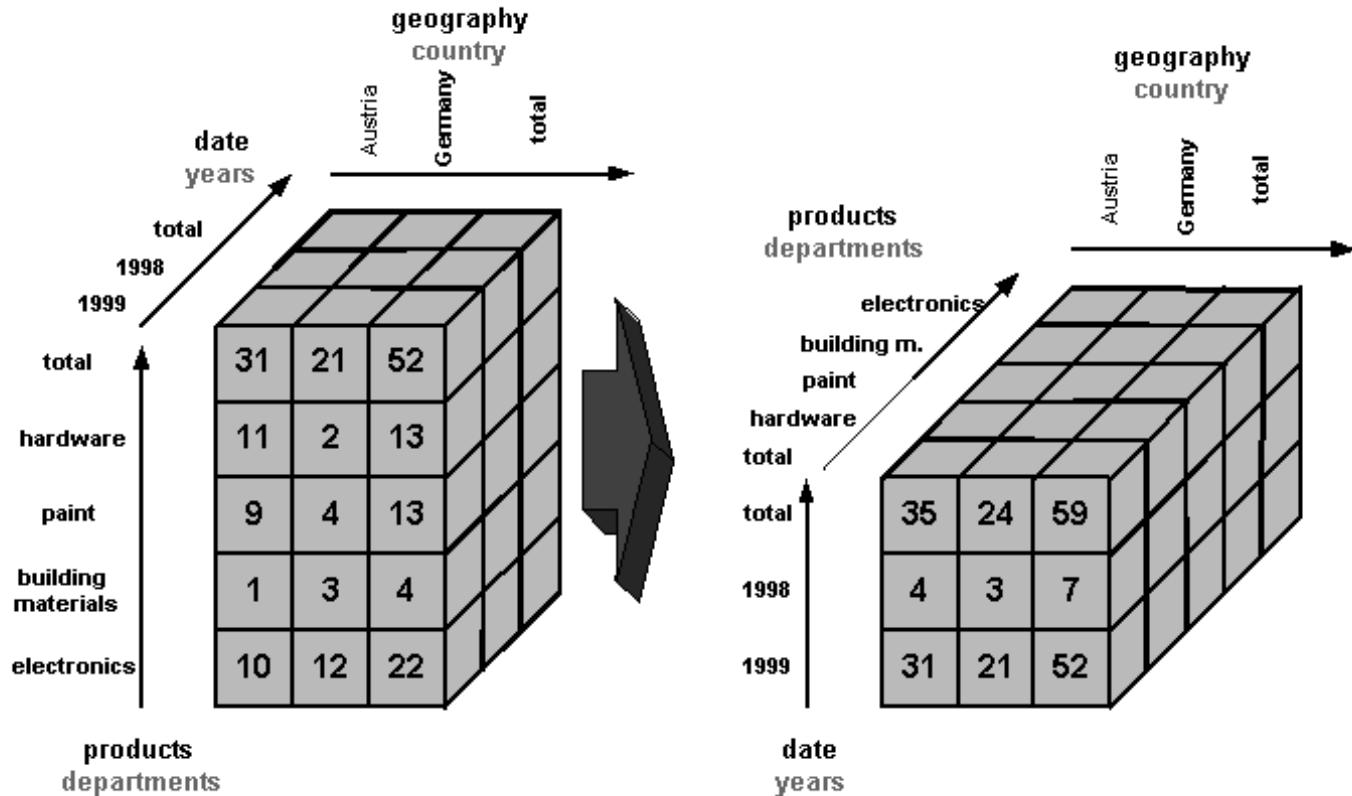


# Pivoting

Rotate the cube

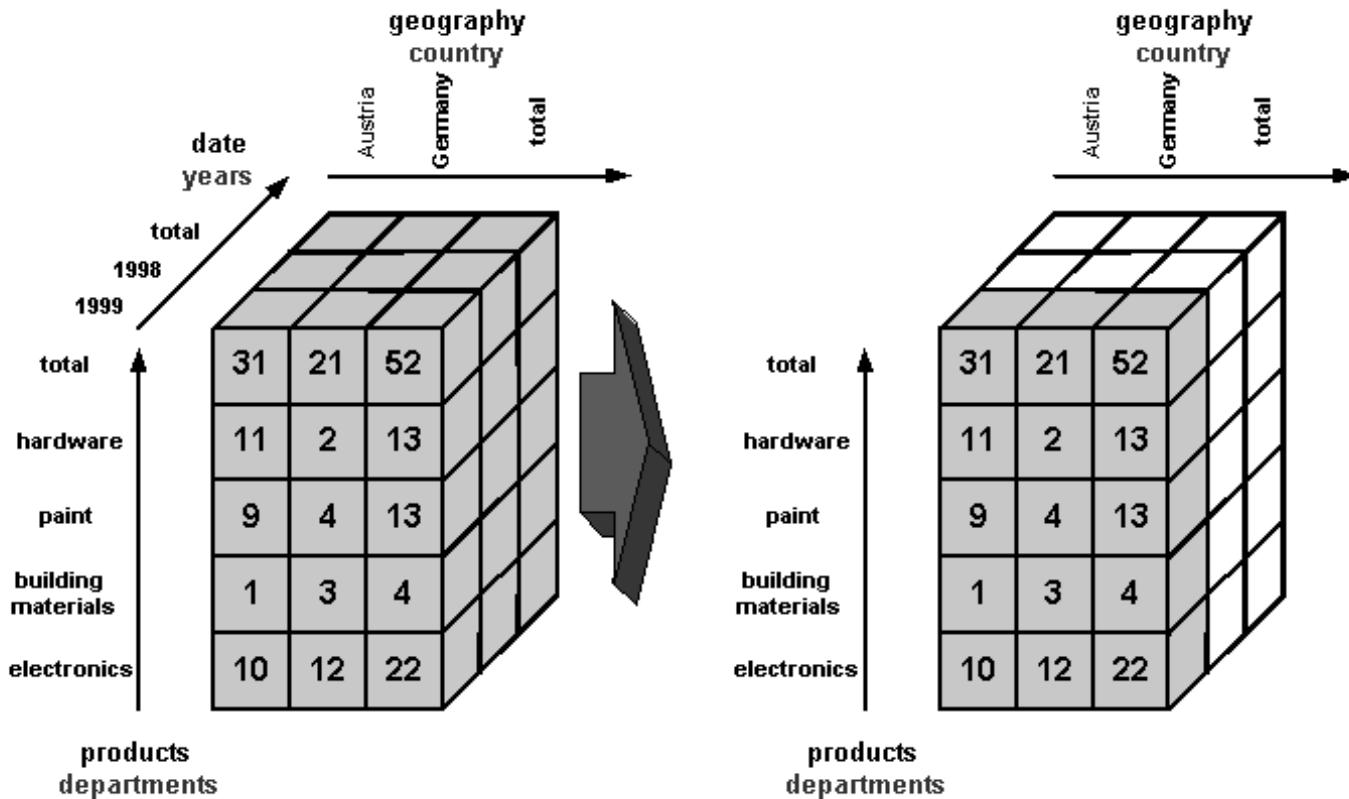


“Show year by country instead of product by country”

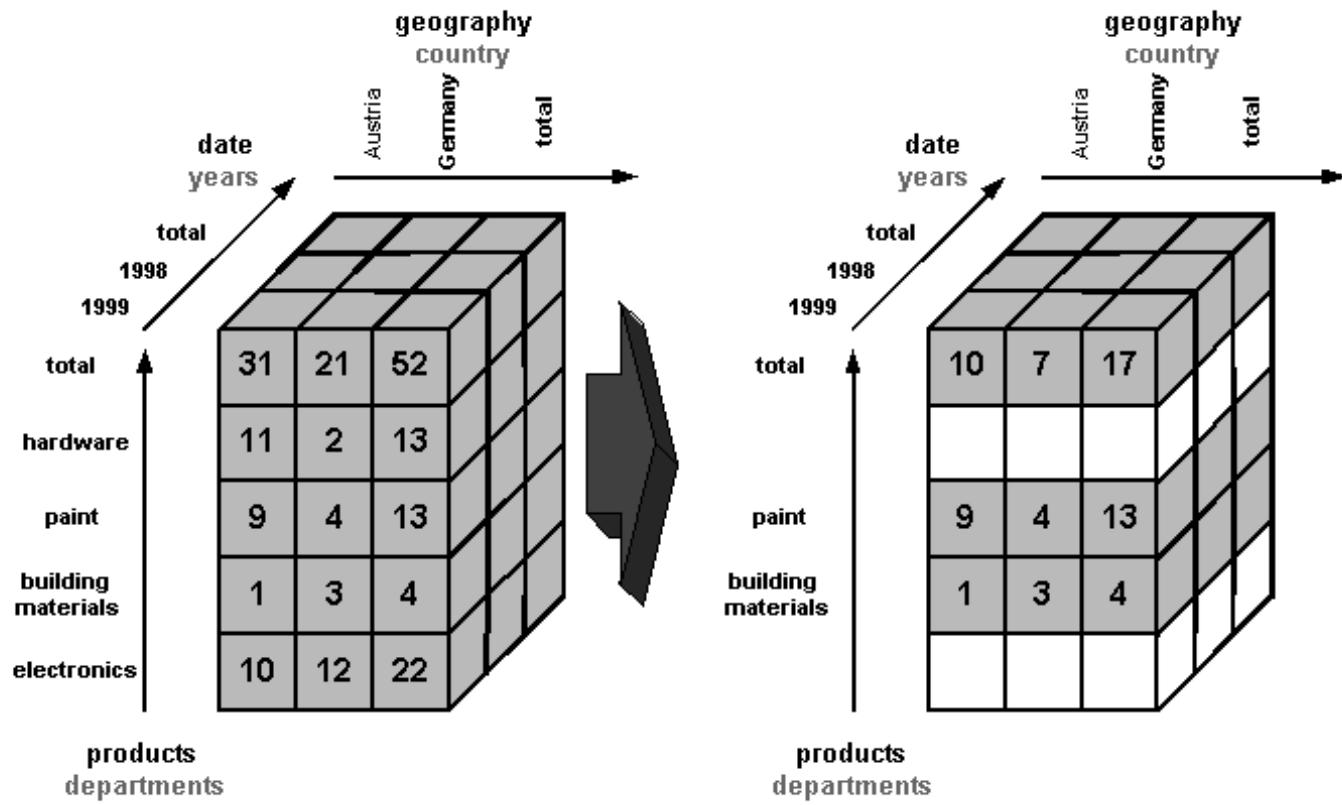


# Slicing

“Show only the values for 1999.”



“Show only the values for the departments ‘paint’ & ‘building materials’ for all the countries and all the years.”



## On-line Transactional Processing (OLTP)

- Common way of transactional processing (INSERT, UPDATE, DELETE)
- Primarily used on operational databases (day-by-day business)
- Treats microscopic transactions (e.g., by processing single accounting transactions or order transactions)
- Does not support strategic decisions, but controls and runs subsequent operations

```
66.249.76.123 - - [14/Oct/2012:03:47:21 +0200] "GET /index.php/de/ HTTP/1.1" 200 23963
178.154.211.123 - - [14/Oct/2012:03:49:28 +0200] "GET /robots.txt HTTP/1.1" 200 370
66.249.76.123 - - [14/Oct/2012:04:00:40 +0200] "GET / HTTP/1.1" 303 -
66.249.76.123 - - [14/Oct/2012:04:00:41 +0200] "GET /index.php/de/ HTTP/1.1" 200 23961
123.125.71.123 - - [14/Oct/2012:04:19:44 +0200] "GET / HTTP/1.1" 303 -
220.181.108.123 - - [14/Oct/2012:04:19:44 +0200] "GET / HTTP/1.1" 303 -
66.249.76.123 - - [14/Oct/2012:04:30:46 +0200] "GET /index.php/de/konferenzen-uebersicht/konferenzuebersicht HTTP/1.1" 200 20598
180.76.5.123 - - [14/Oct/2012:04:35:20 +0200] "GET / HTTP/1.1" 303 -
```

|                           | OLTP                     | OLAP                            |
|---------------------------|--------------------------|---------------------------------|
| <b>data</b>               | operational transactions | management analysis data        |
| <b>user friendliness</b>  | low                      | high                            |
| <b>granularity</b>        | microscopic              | macroscopic                     |
| <b>up-to-dateness</b>     | current status           | historic snapshots              |
| <b>main operations</b>    | update (read/write)      | query and calculate (read only) |
| <b>storage efficiency</b> | high                     | lower                           |
| <b>tools</b>              | e.g. SQL                 | proprietary tools               |

## ➤ OLAP vs. OLTP in a nutshell

<https://www.youtube.com/watch?v=iw-5kFzldgY>  
(IBM Technology Video, last access April 2024)

# Pros and cons of OLAP

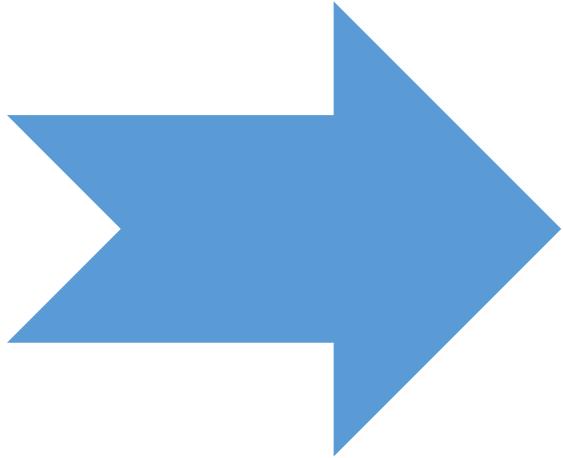
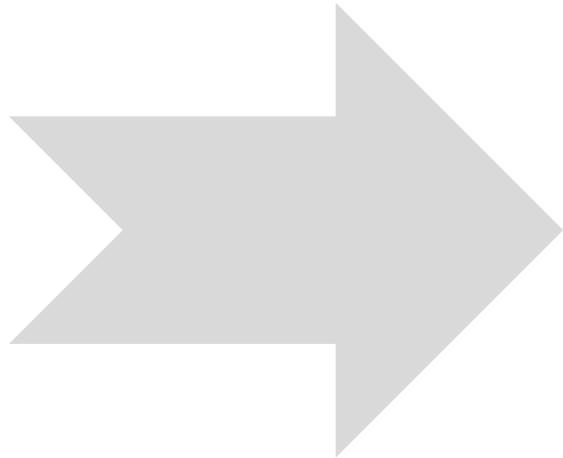
Pro:

- Wide applicability of the method
- OLAP presents quite exact results
- Method is plausible

Con:

- OLAP requires a lot of user interaction
- OLAP regularly requires quite a lot of computing resources
- Difficult to use automated data mining routines in combination with OLAP

Ref.



## (1) Online Analytical Processing (OLAP)

Different query methods

Properties of OLAP

Common OLAP functionality

## (2) Modeling layers

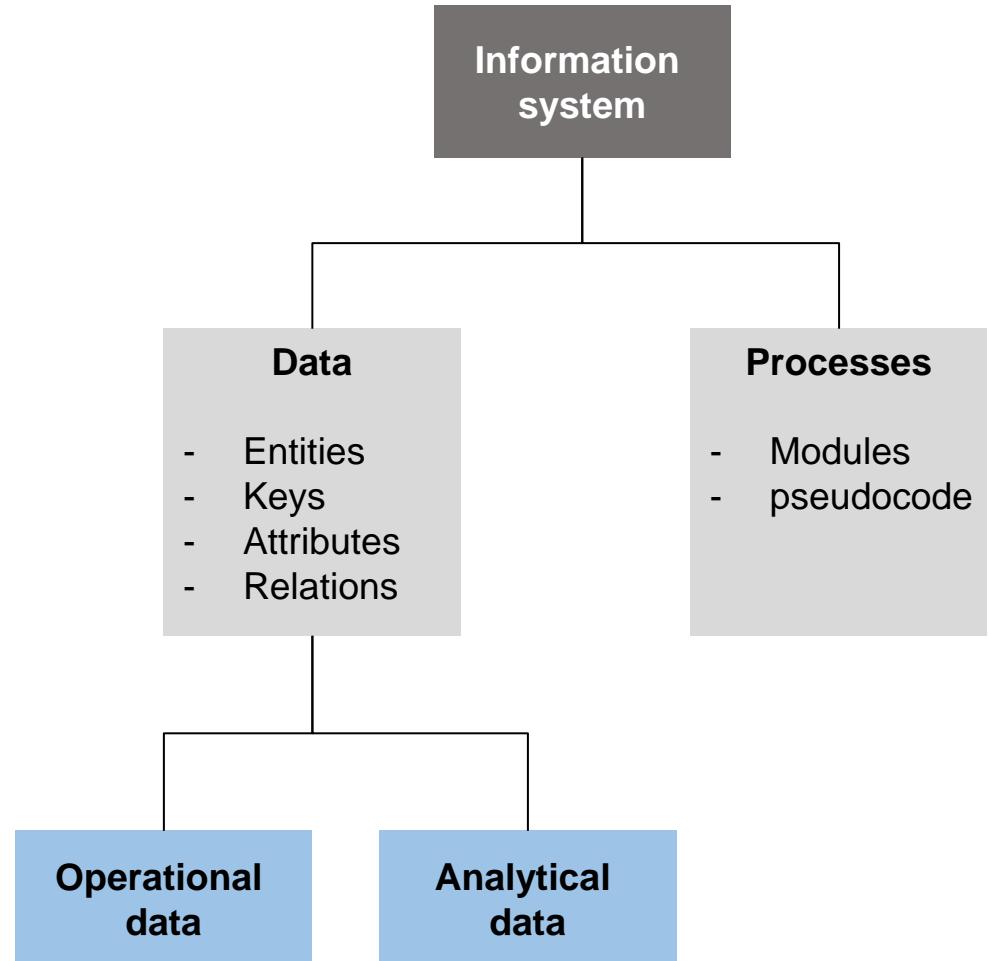
**Basic Elements of multidimensional modeling**

Conceptual modeling

Logical modeling

Physical modeling

# Modeling of information systems



## Operational databases

- Optimize storage efficiency and response time
  - Model data which is
    - fine-grained (many details)
    - dynamic (many updates)
- 
- Normalize data
  - Minimize redundancy
  - Provide data integrity
    - Avoid update anomalies
    - Avoid deletion anomalies
    - Avoid insertion anomalies

## Analytical databases

- Support the decision making process
  - Maximize user-friendliness and querying efficiency
  - Model data which is
    - coarse-grained (less details)
    - static (less updates)
- 
- Data is denormalized
  - Redundancy minimization is secondary

→ *Mirror different views on business measures within the model*



Image: [CTSI-Global](#) | Flickr (cc by-sa 2.0)

# Multidimensional modeling

## Basic Elements

### Common steps compared to operational databases

Leave out operational data  
(not all attributes necessary)

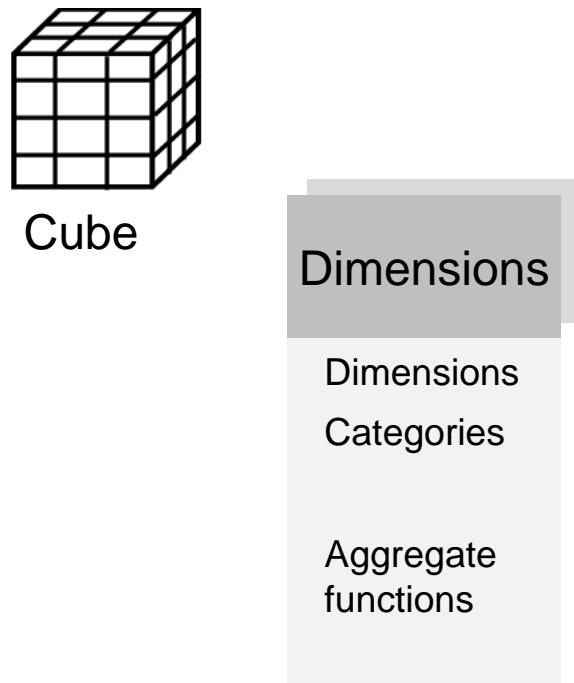
Include time dimension

Integrate pre-calculated attributes

Reduce join operations

### Basic elements of multidimensional models

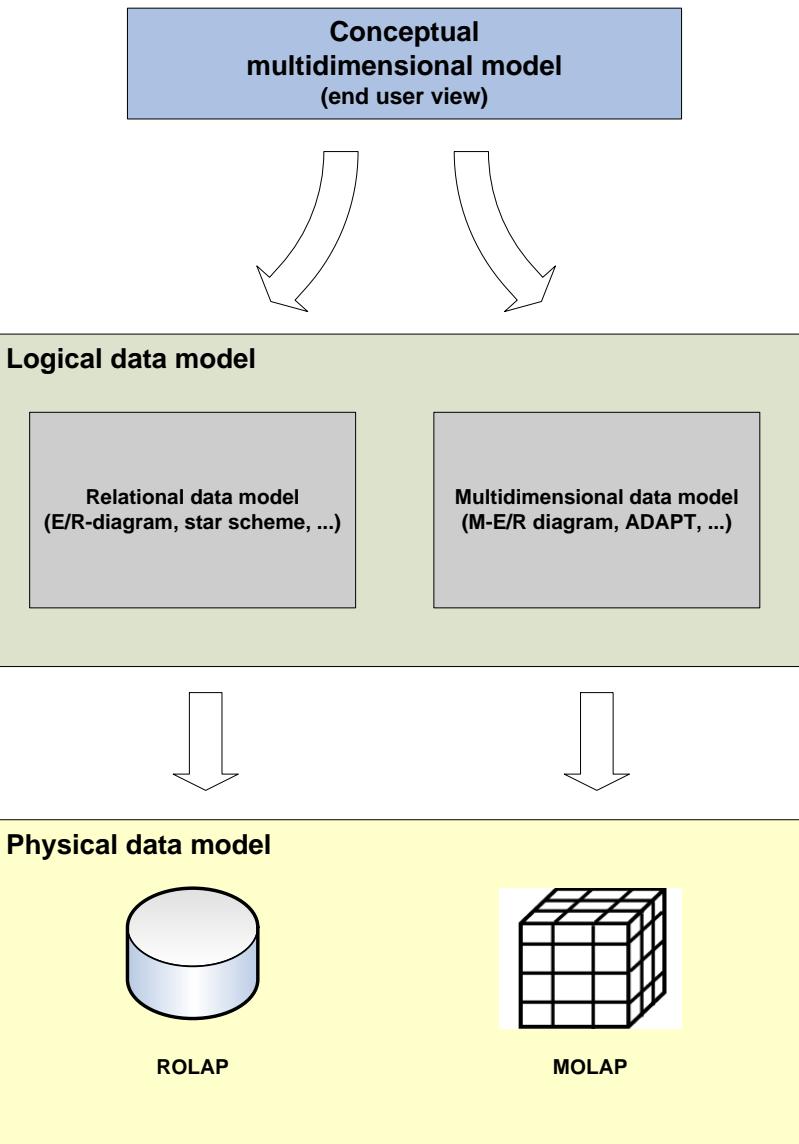
Facts  
Dimensions  
Categories  
Aggregation functions



# Multidimensional modeling

## Major steps

1. Identify **facts and dimensions**
2. Create a **conceptual** data model
3. Derive a **logical** data model from the semantic model
4. Derive a **physical** data model from the logical model



Multidimensional models are designed according to the needs of decision makers

- Business measures are in the center of interest of decision makers

**Definition** of business measure:

*"Business measures are compressed mostly numeric measurements, which refer to **important matters of fact** within the company and which represent them in a **concentrated** manner. They provide **information about business issues** and thereby provide **important support** for the decision processes within the company."*

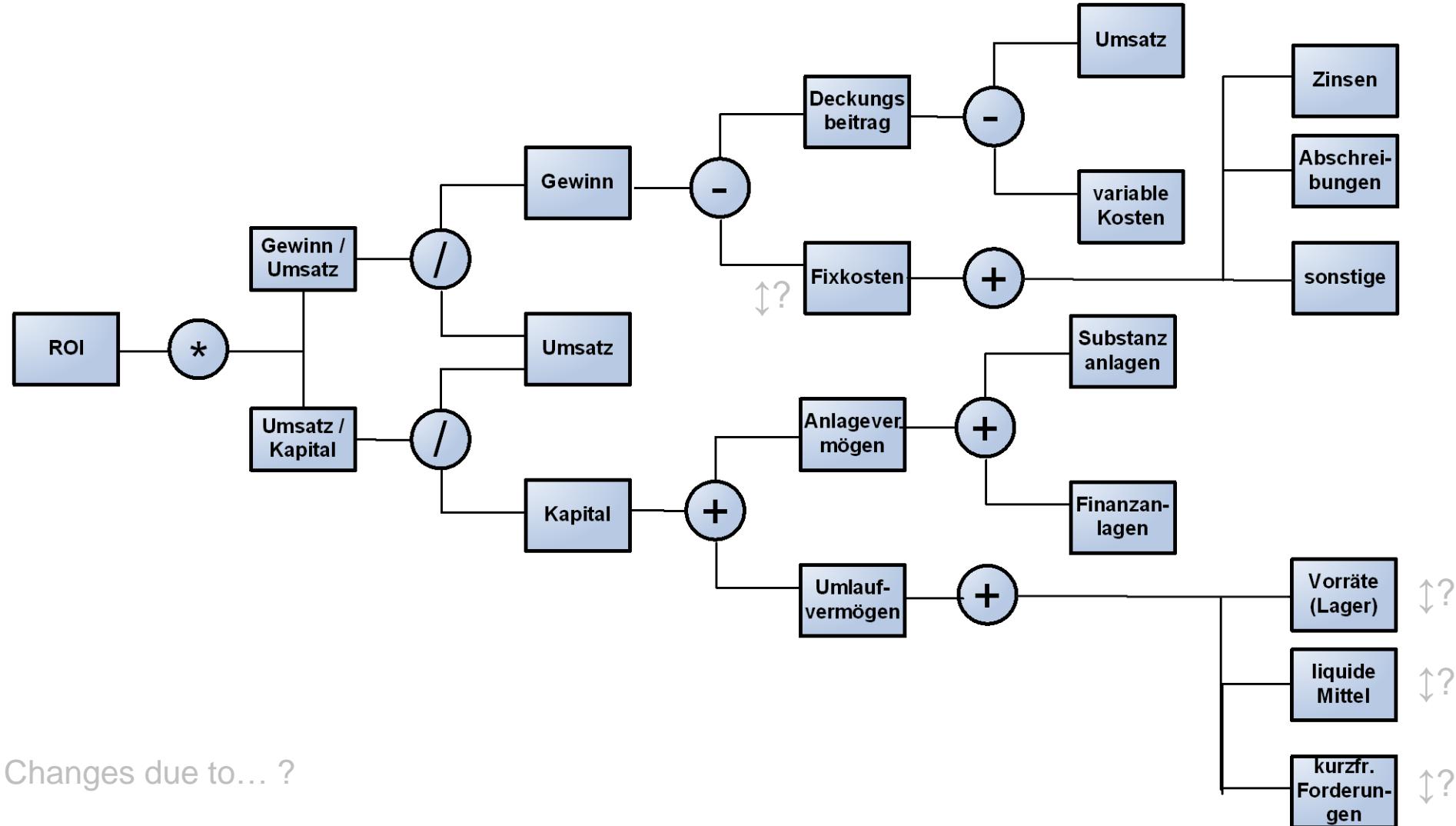
(Langenbeck, 1997, highlights added)

Example business measures: revenues, profits, sales, ROI ...

**Identification of business measures** is one of the basic tasks in multidimensional modeling

# Example business measure system: ROI

~ „Erfolg im Verhältnis zum eingesetzten Kapital“, „Gewinn in Prozent des investierten Kapitals“, ...



Ref.

Decision makers want to **analyze business measures** from **different views** (dimensions)

- Several dimensions are arranged around one fact

*“What amount were the sales revenues for hard disks within the past quarter?”*

*fact: sales revenues*

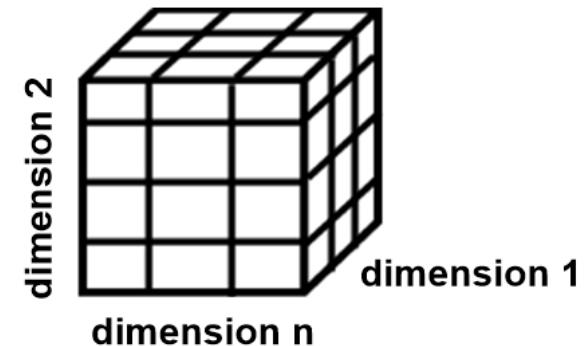
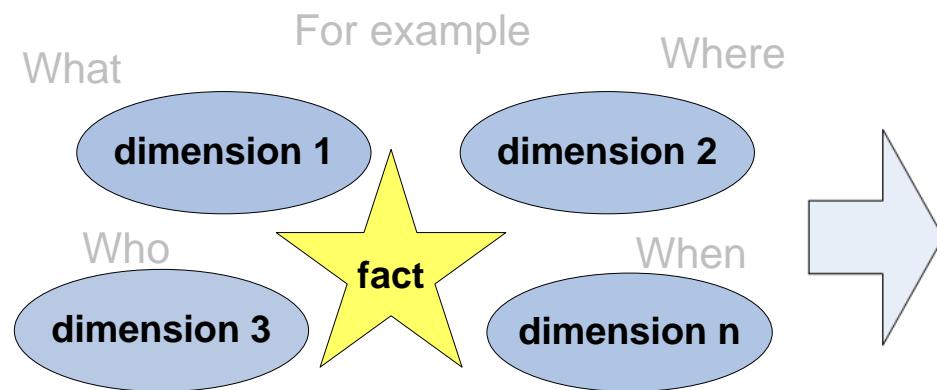
*dimensions: range of products, time*

*“How profitable has our Africa department been on software?”*

*“What is our growth on A-customers throughout the last quarter?”*

During the modeling process, business measures and their set of dimensions are determined

Link to multi-dimensional data structures:



# Dimensions and categories

*Dimension = finite set of categories* which are semantically related to each other with respect to business matters

Categories of one dimension represent a **different levels of aggregation** of the associated business *measures* (facts)  
Categories are also known as aggregation objects

## An example

dimension: “date”

Four categories: day  $\Rightarrow$  month  $\Rightarrow$  quarter  $\Rightarrow$  year

Resp.: “Sales revenues for hard disks within the past day, month, quarter, year, ...?”

# Categories

A category is represented by a varying set of elements

e.g., country = [Germany, Austria, Switzerland], quarter = [q1, q2, q3, q4]

Each dimension consists of **at least one** (real) category

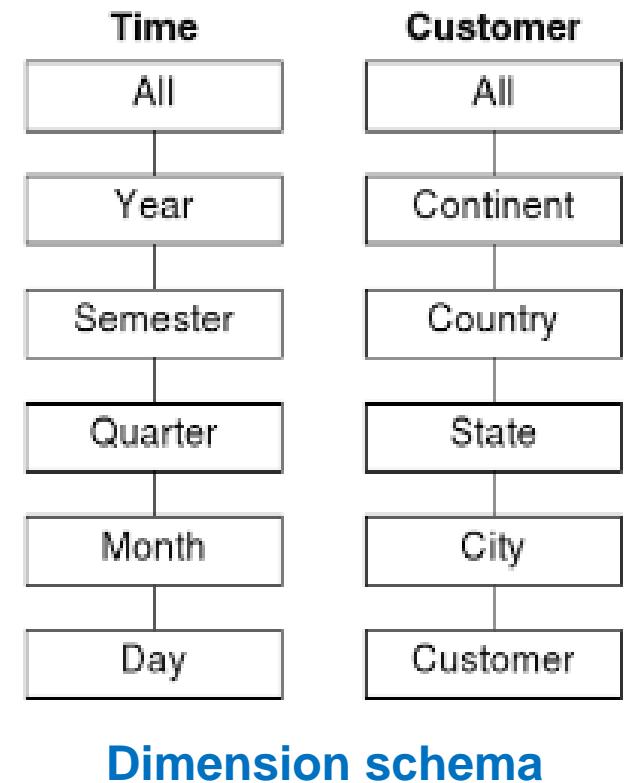
a category represents the level of granularity

**“Log-category”** (e.g., “day” in dimension “date”)

a virtual category, encompassing all the others, like

**“all-category”**: encompasses all elements of the Log-category

Number of categories of a dimension is not limited



# Facts and aggregation functions

**Aggregation function** = formula defining the value of facts with respect to the different categories of a dimension

Facts can be classified with respect to aggregation functions:

## Additive facts

(distributive aggregation function)

Simple addition possible throughout all the categories of all the associated dimensions

e.g., units sold

## Semi-additive facts

(algebraic aggregation function)

Simple addition only possible for a selected number of the categories of the associated dimensions

e.g., not additive over time, but maybe over regions

e.g., current stock/ inventory level, current balance amount

## Non-additive facts

(holistic aggregation function)

Simple addition operations not sufficient

e.g., types of average values or ratio values

e.g., temperature

# Special types of facts

Fact groups, dimensional and virtual facts

## Fact group

**set of facts** featuring a *common* set of dimensions

e.g., *units sold* and *sales in \$ per day*

| Sales        |
|--------------|
| - units sold |
| - in \$      |

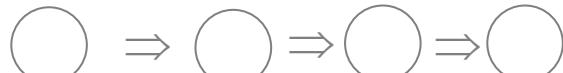
## Dimensional fact:

fact is associated with **only one dimension**

dimensional facts are often numerical, non-dimensional attributes of a dimension's category

e.g., sales area (dimension "geography")

geography country state store



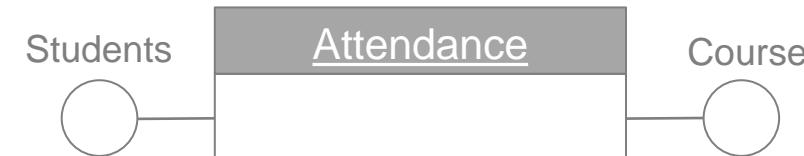
(sales area, address, etc)

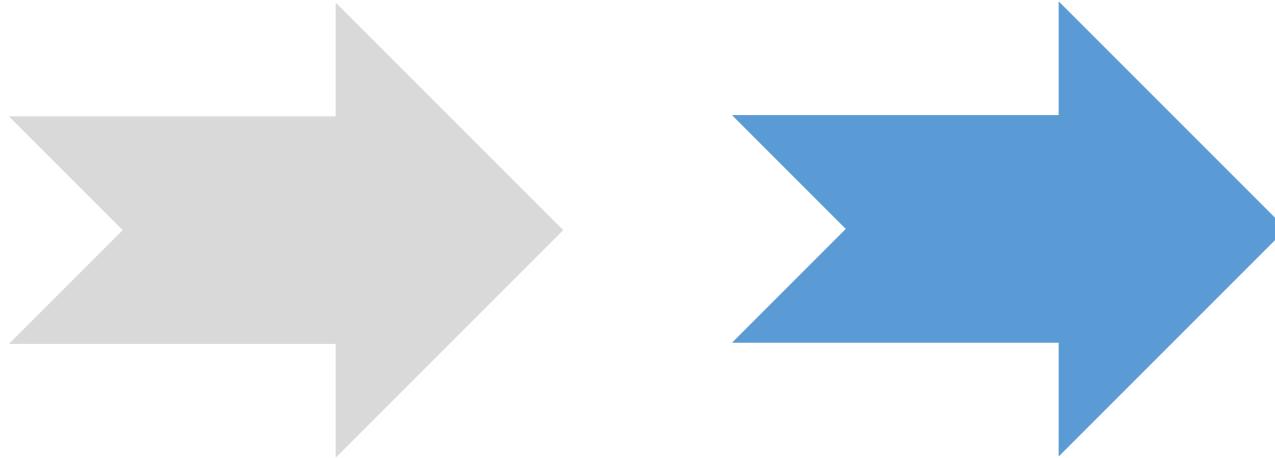
## Virtual fact (a.k.a. "factless fact"):

**Association between dimensions** alone defines the (nominal) fact

e.g., students attendance in class (students, class, time) – virtual fact (0/1) to ask for: *how many students attended class x?*

Relational implementation: only keys in fact table





## (1) Online Analytical Processing (OLAP)

Different query methods

Properties of OLAP

Common OLAP functionality

## (2) Modeling layers

Basic Elements of multidimensional modeling

**Conceptual modeling**

Logical modeling

Physical modeling

# Conceptual Modeling

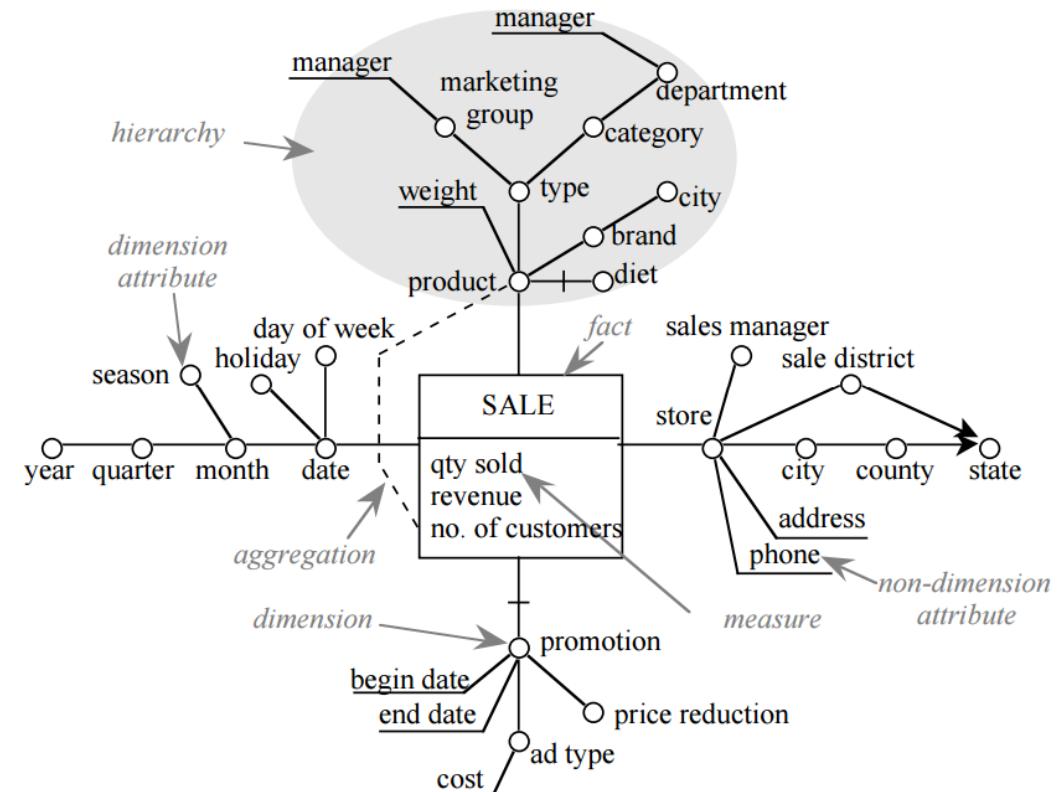
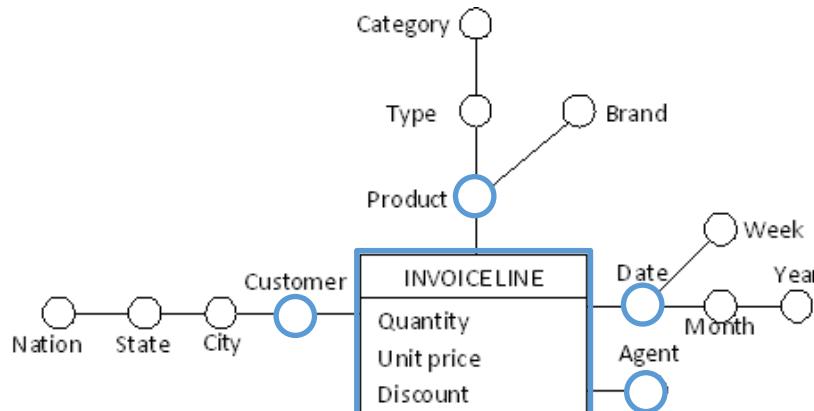
Dimensional fact model (or fact scheme)

Categories of a dimension arranged in a non-cyclic graph, directed between all-category and log-categories

Categories can have an arbitrary number of (non recursive) relations between each other

Several aggregation paths (e.g., sum/count/mean) may be included in the graph

Hierarchies are discrete attributes and define the granularities of facts (i.e., product -> type -> category)



# Exercise

## Conceptual Modeling (Dimensional fact model)



Design a **conceptual model** for the local Food Company:

### Conny's Corner Shop

Your managers need to keep themselves up to date on the number of items in the company's inventory. They especially want to keep an eye on their products with regards to location, and time.

- Conny's Corner Shop sells a range of snacks and beverages. Both categories have different types of products, such as juices and water, as well as pretzels and crackers.
- The products are sold under different brands.
- Products have different package types, sizes, and weights.
- They store products in different stores across Europe

10 Min.

Show your conceptual model as a dimensional fact model. Make reasonable assumptions if necessary.

## Fragen?

- ✓ Online Analytical Processing (OLAP)
  - ✓ Different query methods
  - ✓ Properties of OLAP
  - ✓ Common OLAP functionality
  
- ✓ Modeling layers
  - ✓ Basic Elements of multidimensional modeling
  - ✓ Conceptual modeling
    - Logical modeling
    - Physical modeling

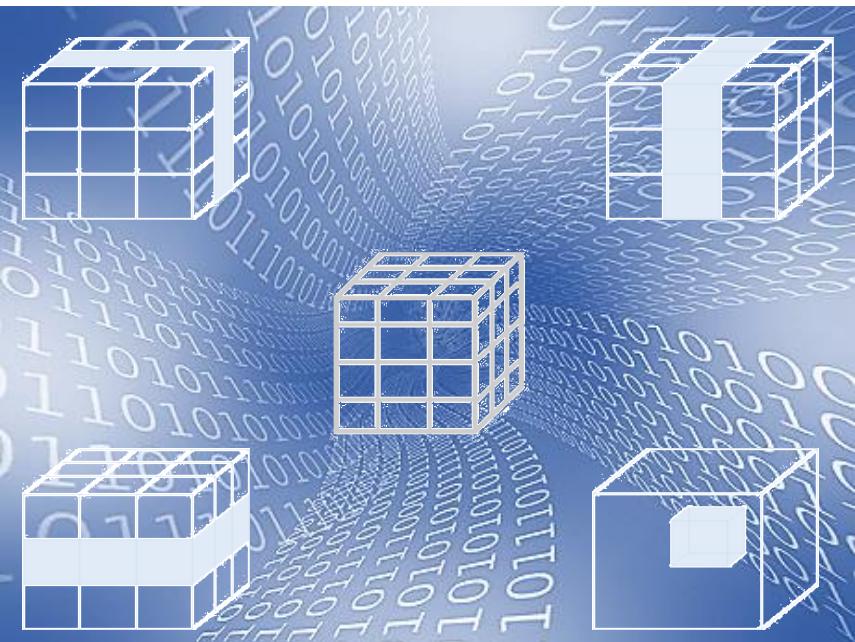
# Todos for next Week

1. Support Conny's Corner Shop by finishing the conceptual model.  
*See exercise on slide 29*

2. Python-Basics – Chapter 3  
*Kursmaterial > Readings/Übungen > Python Übungen – Jupyter*

# Bibliography

- Böhnlein, M. (2013). *Konstruktion semantischer Data-Warehouse-Schemata*. Springer-Verlag.
- Bulos, D., & Forsman, S. (2000). *Olap Database Design: Delivering on the Promise of the Data Warehouse*. Morgan Kaufmann Publishers Inc..
- Golfarelli, M., Maio, D., & Rizzi, S. (1998). The dimensional fact model: A conceptual model for data warehouses. *International Journal of Cooperative Information Systems*, 7(02n03), 215-247.
- Hahne M. (2006) Mehrdimensionale Datenmodellierung für analyseorientierte Informationssysteme. In: Chamoni P., Gluchowski P. (eds) Analytische Informationssysteme. Springer, Berlin, Heidelberg
- Jukic, N., Jukic, B., & Malliaris, M. (2008). Online analytical processing (OLAP) for decision support. In Handbook on Decision Support Systems 1 (pp. 259-276). Springer, Berlin, Heidelberg.
- Vaisman, A., & Zimányi, E. (2014). *Data Warehouse Systems*. Springer, Heidelberg



# Business Intelligence

## 04 Data Warehouse Modeling II

## & First Short Introduction to Data Mining

Prof. Dr. Bastian Amberg  
(summer term 2024)  
10.5.2024

# Schedule

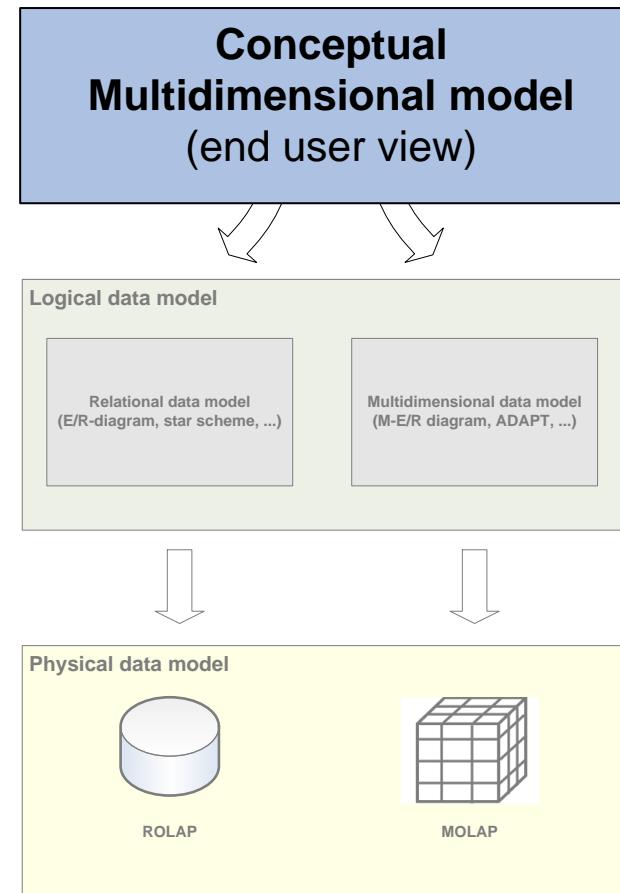
|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study     |                  |               |         |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|----------------|------------------|---------------|---------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |                |                  | Python-Basics | Chap. 1 |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |                |                  |               | Chap. 2 |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  | Data Warehouse Modeling I  |                |                  |               | Chap. 3 |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining   | Introduction   |                  |               |         |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |                | Python-Analytics | Chap. 1       |         |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization              |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam                 |                |                  | Chap. 2       |         |  |  |
|           | W7                | 29.5. | Data Preparation                                    |                                   | 31.5. |   |                |                  |               |         |  |  |
|           | W8                | 5.6.  | Predictive Modeling I                               |                                   | 7.6.  | Predictive Modeling II (10:00 -12:00)   |                | BI-Project       | Start         |         |  |  |
|           | W9                | 12.6. | Fitting a Model I                                   |                                   | 14.6. | Python-Analytics-Online Exercise  |                |                  |               |         |  |  |
|           | W10               | 19.6. | Guest Lecture                                       |                                   | 21.6. | Fitting a Model II  |                |                  |               |         |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |                |                  |               |         |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I  |                |                  |               |         |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |                |                  |               |         |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |                |                  | End           |         |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin ~ 22.7. bis 3.8.<br>Klausur 2.Termin ~ 23.9. bis 5.10.                                       | Projektbericht |                  |               |         |  |  |

- ✓ Online Analytical Processing (OLAP)
- ✓ How can multidimensional data models be **developed** and stored?

1. Identify **facts** and **dimensions**
2. Create a **conceptual** data model

3. Derive a **logical** data model from the semantic model

4. Derive a **physical** data model from the logical model

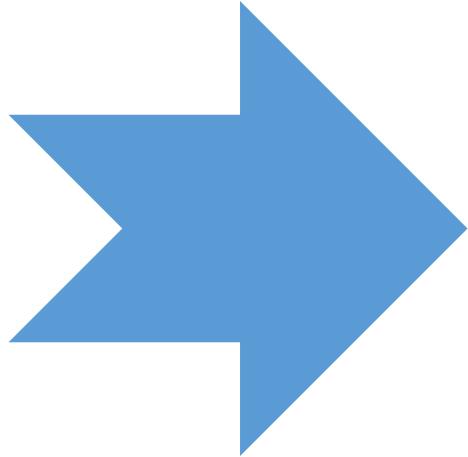


*Dimension = finite set of categories* which are semantically related to each other with respect to business matters

*Categories of one dimension represent a different levels of aggregation of the associated business measures (facts)*

*For example, dimension: "date"*

*Four categories: day  $\Rightarrow$  month  $\Rightarrow$  quarter  $\Rightarrow$  year*



## **Continuation Data Warehouse Modeling**

Basic Elements of  
multidimensional modeling

**Conceptual modeling**

Logical modeling

Physical modeling

# Conceptual Modeling

Dimensional fact model (or fact scheme)

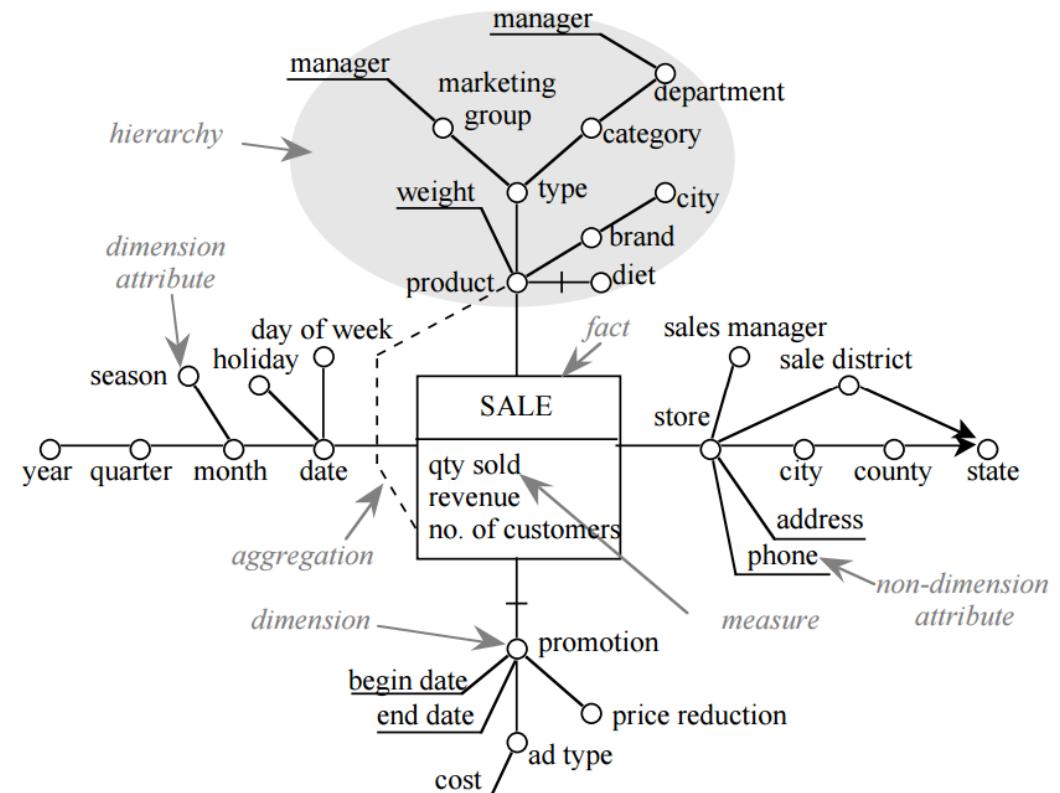
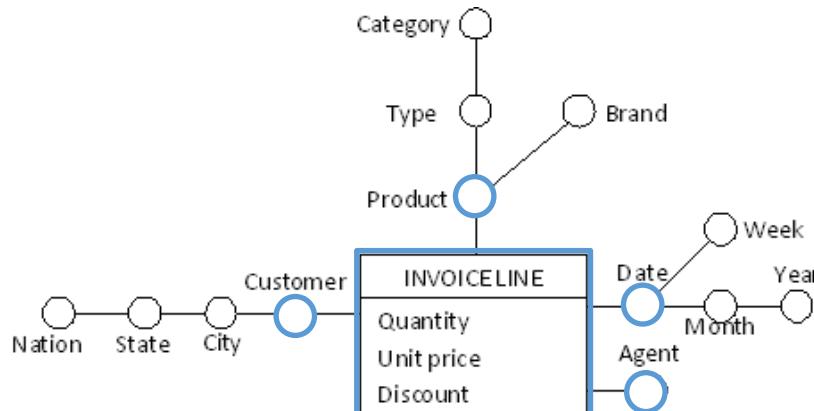
Wdh.

Categories of a dimension arranged in a non-cyclic graph, directed between all-category and log-categories

Categories can have an arbitrary number of (non recursive) **relations** between each other

Several **aggregation paths** (e.g., sum/count/mean) may be included in the graph

**Hierarchies** are discrete attributes and define the granularities of facts (i.e., product -> type -> category)



# Non-& Cross-dimensional attributes

## Dimensional fact model

There may be **various types of relations** between individual categories of one (or more) dimension(s)

Categories having 1:1 relations can be summarized into a single category

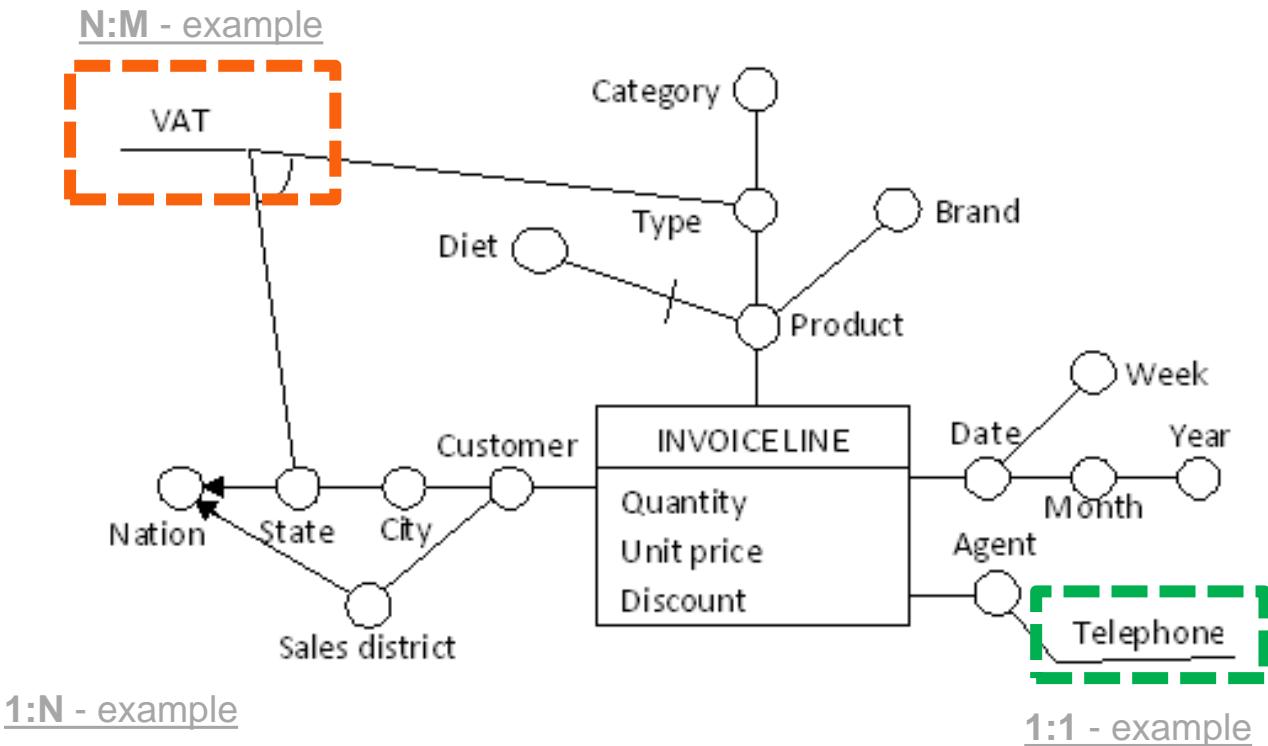
When previous categories become non-dimensional,  
**descriptive attributes**

Non-dimensional attributes provide additional information and have 1:1 relations with the corresponding categories (phone-No. cannot be aggregated)

OLAP-compliant queries encompassing non dimensional attributes are generally not supported

## Categories having 1:N or N:M relations

When value is defined by multiple categories (e.g., product type and state. For instance, VAT for water or books in Germany vs. USA) they become **cross-dimensional attributes**



# Exercise

Conceptual Modeling (Dimensional fact model)

Design a **conceptual model** for the local Food Company:

## Conny's Corner Shop

Your managers need to keep themselves up to date on the number of items in the company's inventory. They especially want to keep an eye on their products with regards to location, and time.

- Conny's Corner Shop sells a range of snacks and beverages. Both categories have different types of products, such as juices and water, as well as pretzels and crackers.
- The products are sold under different brands.
- Products have different package types, sizes, and weights.
- They store products in different stores across Europe

Show your conceptual model as a dimensional fact model. Make reasonable assumptions if necessary.

Wdh.

10 Min.

# Identifying fact groups

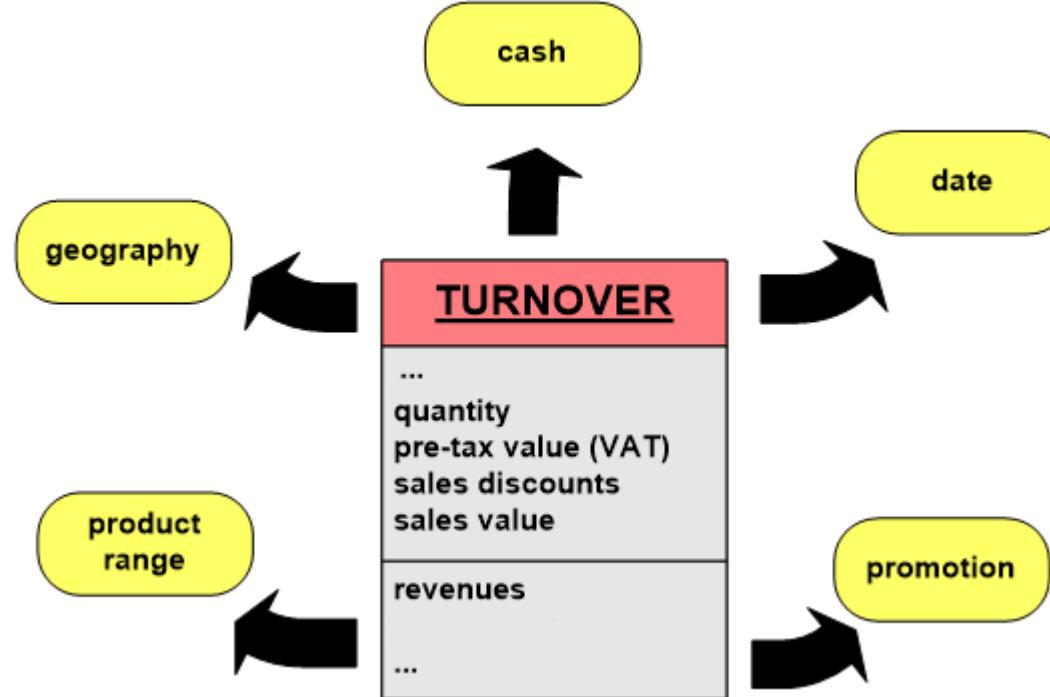
## Single Star Scheme

A graphical overview should be created for each fact group

### single star scheme

Distinction between materialized facts and derived facts should be drawn

High level of abstraction:  
aggregation formulas are commonly not modeled



Remember: Fact group = set of facts  
featuring a *common* set of dimensions

# Combining fact groups

## Multiple Star Scheme

A data warehouse data model encompasses a number of fact groups (**multiple star scheme**)

The sets of associated dimensions of different fact groups may overlap.

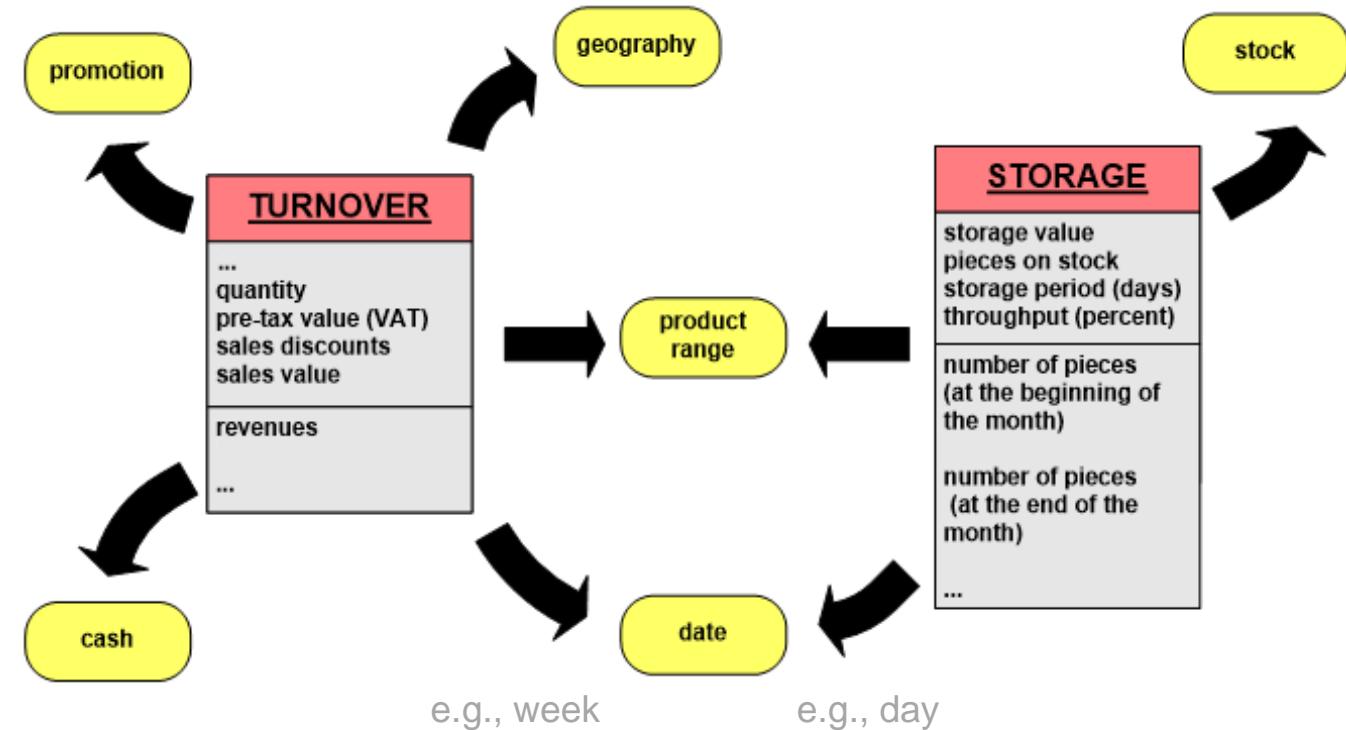
Different fact groups may use different log-categories with respect to one common dimension

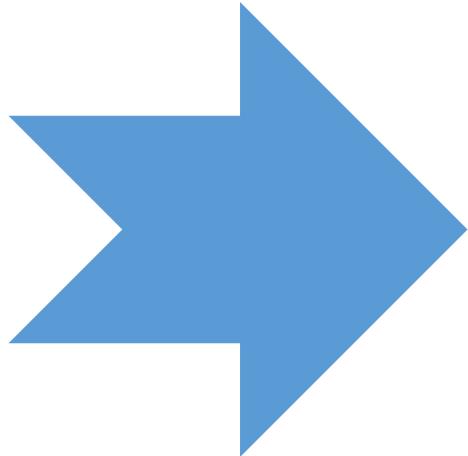
e.g., actual / debit values:

Actual facts: log-category of dimension date: day

Target facts: log-category of dimension date: month

Category used as log-category should be specified in the model (at least if standard log-category is not used)





## **Continuation Data Warehouse Modeling**

Basic Elements of  
multidimensional modeling

Conceptual modeling

**Logical modeling**

Physical modeling

# Basic tasks with logical modeling

**Logical modeling** is about adapting the general conceptual schema to the applied database technology

**Relational database** technology

⇒ ER diagrams, ...

**Multidimensional database** technology

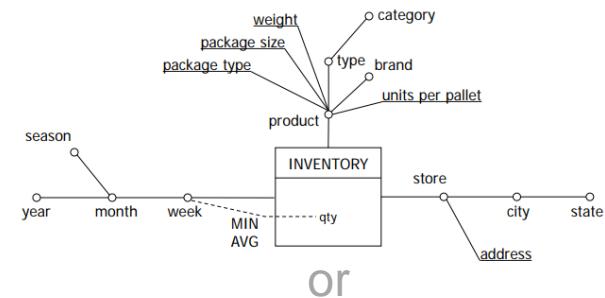
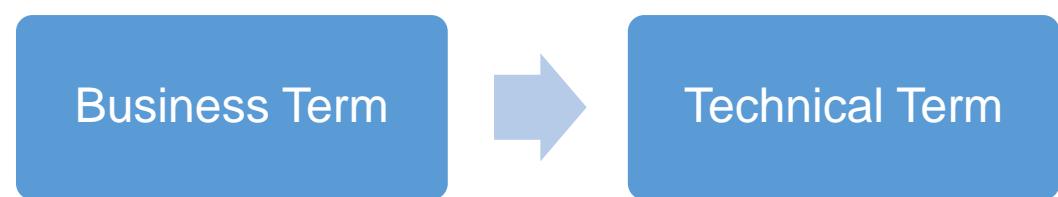
⇒ M-ER diagrams, ADAPT diagrams, ...

M-ER: Multidimensional Entity-Relationship

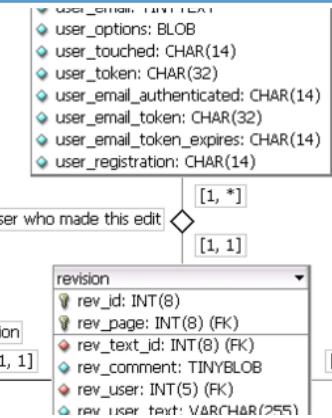
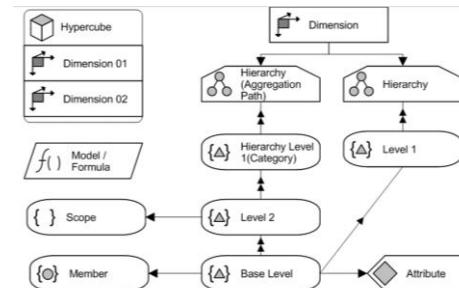
ADAPT: Application Design for Analytical Processing Technologies

(Both also usable as starting point for relational DB design)

**Star Scheme** is the standard way of logical modeling concerning **relational DBMS**



or



**Excusus:** Genealogy of Relational Database Management Systems



Image: AutumnSnow (2008) | Wikimedia (cc by sa 3.0)

# Properties of star scheme models

A number of fact tables are associated with a set of dimension tables each (via **unique keys**)

One dimension is mapped to exactly one relation

Primary key of a fact table consists of the keys of the associated dimensions

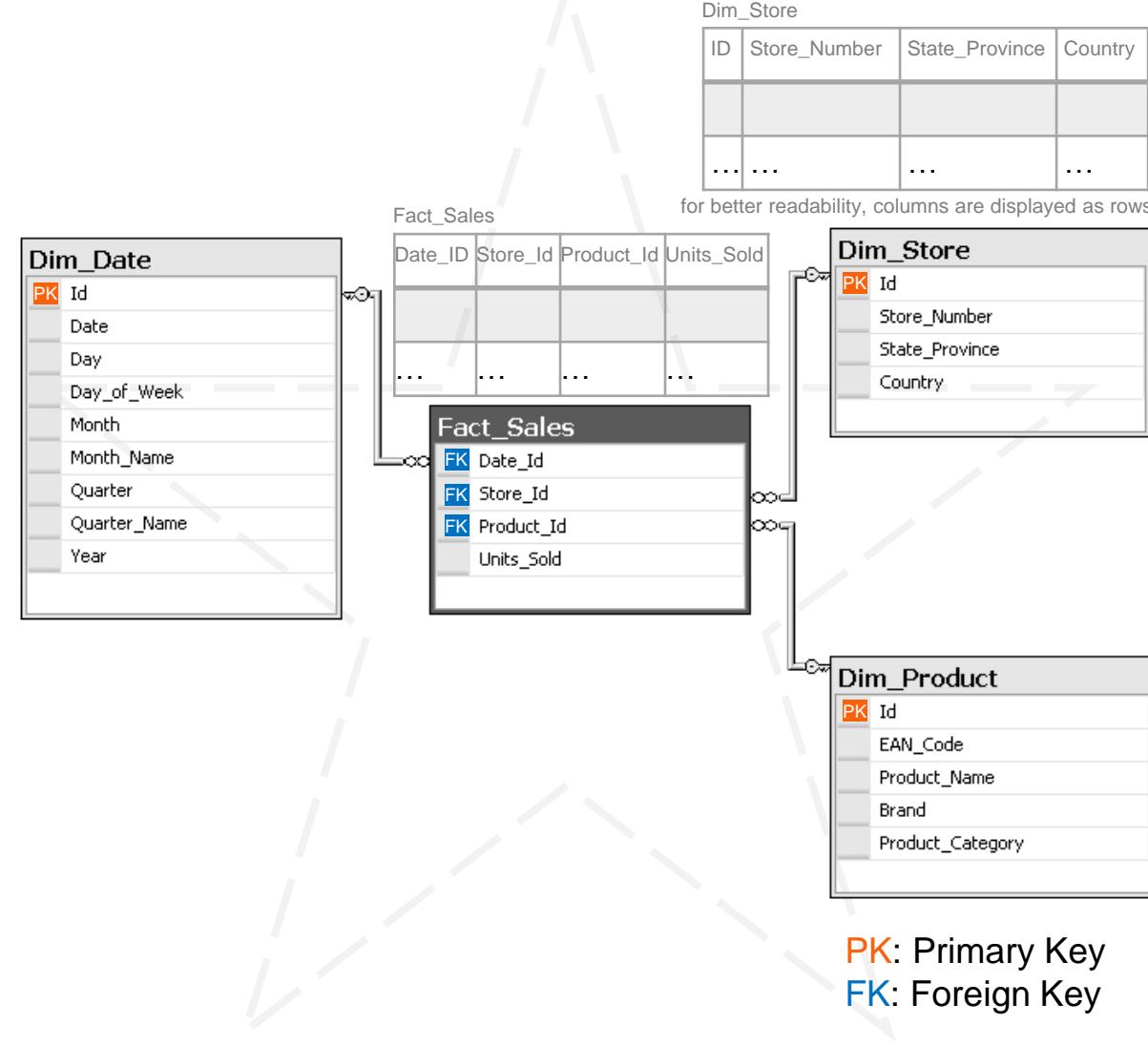
Keys of the dimension tables are usually “artificially” created (e.g. serial numbers)

1:m type relations between dimensions and facts

Typical “multi-join” query for star schemes:

```
SELECT
    D1.State_Province, D2.Month,
    SUM (F1.Units_Sold)
FROM Dim_Store D1, Dim_Date D2, Fact_Sales F1
WHERE
    D1.Id = F1.Store_Id AND
    D2.Id = F1.Date_Id
GROUP BY D1.State_Province, D2.Month
```

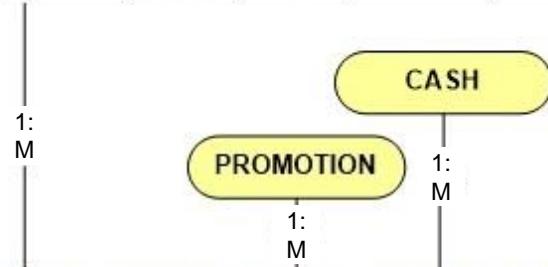
Result?



# Simple star scheme

## GEOGRAPHY

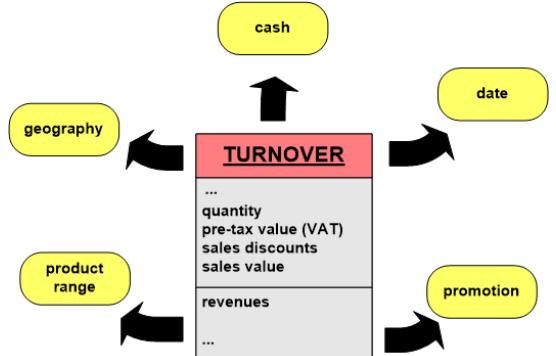
| MKT_ID | MKT_DESC | MKT_ADR    | STATE   | COU | MGR_MKTG     | MGR_MKTG_ADR     | REGION_MKTG | REGION_BUY |
|--------|----------|------------|---------|-----|--------------|------------------|-------------|------------|
| 1      | Market#1 | Pfaustr 1A | Hesse   | GER | Heine R.     | +49 (2203) 107-0 | reg A       | central    |
| 2      | Market#2 | Südstr. 11 | Hesse   | GER | Schneider S. | .....            | reg B       | central    |
| 3      | Market#3 | Nordstr. 7 | Bavaria | GER | Jose R.      | ....             | reg B       | south      |
| 4      | Market#4 | ...        | ...     | ... | ...          | ...              | ...         | ...        |



| DAY_ID | MKT_ID | PROD_ID | PROM_ID | CASH_ID | TO_INIT_VAL | TO_QUANTITY | TO_VAT_VAL | TO_DISC | TO_SAL_VAL |
|--------|--------|---------|---------|---------|-------------|-------------|------------|---------|------------|
| 113    | 1      | 99      | 2       | 1       | 10          | 100         | 11         | 0.05    | 10         |
| 113    | 1      | 100     | 2       | 1       | 55          | 240         | 24         | 0.03    | 55         |
| 113    | 1      | 45      | 3       | 2       | 3           | 100         | 4          | 0.03    | 3          |
| 113    | 1      | 102     | 1       | 1       | 1850        | 20000       | 400        | 0.05    | 1850       |
| 113    | 1      | 103     | 2       | ...     | ...         | ...         | ...        | ...     | ...        |
| ...    | ...    | ...     | ...     | ...     | ...         | ...         | ...        | ...     | ...        |



| DAY_ID | DAY_DESC     | DNO_SER | SWITCH_FLAG | WEEK | MONTH     | WEE_KNO | QUARTER | YEAR |
|--------|--------------|---------|-------------|------|-----------|---------|---------|------|
| ...    | ...          | ...     | ...         | ...  | ...       | ...     | ...     | ...  |
| 110    | 31-12-99-Thu | 1480    | n           | W53  | M12/Q4/99 | 53      | Q4/99   | 1999 |
| 111    | 01-01-99-Fr  | 1481    | n           | W53  | M01/Q1/99 | 1       | Q1/99   | 1999 |
| 112    | 02-01-99-Sa  | 1482    | j           | W53  | M01/Q1/99 | 1       | Q1/99   | 1999 |
| 113    | 03-01-99-Su  | ...     | j           | ...  | ...       | ...     | ...     | ...  |
| ...    | ...          | ...     | ...         | ...  | ...       | ...     | ...     | ...  |



Category or descriptive attribute?  
Hierarchies?  
Performance for aggregation functions?

# Simple star schemes

## Pros and Cons

### PRO

**Simple, intuitive** model

Not many physical join-operations needed

**Not many physical tables**

needed

⇒ maintenance easy

⇒ Extract, Transform, Load (ETL) easy

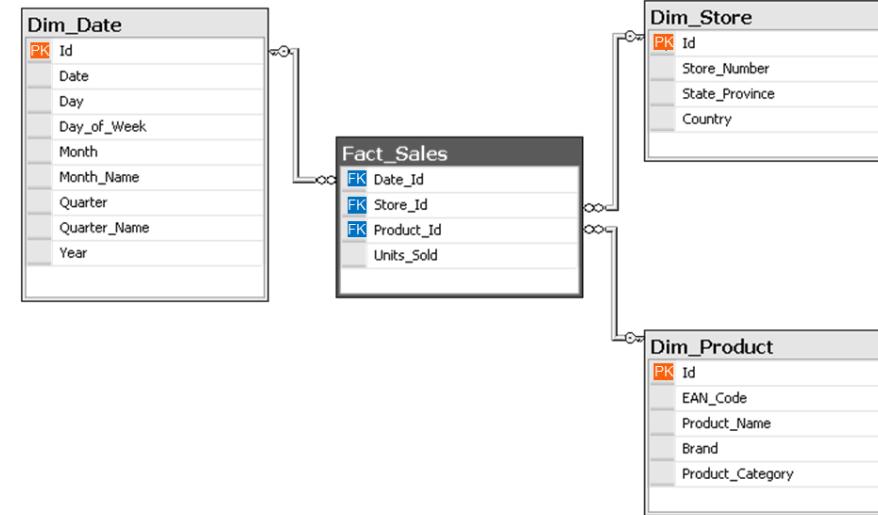
### CON

Large dimension tables may cause **bad response times**

Creation of materialized views (aggregated summary tables) difficult  
⇒ wrong aggregate values may be calculated due to **multiple counting of entries**

Changes of the conceptual model cause extensive reorganization efforts on the tables ⇒ versioning of meta data required

**Redundancy**



**PK:** Primary Key

**FK:** Foreign Key

# Variants of the star scheme

## Snowflake scheme:

**Normalization** of the dimensional tables of a star scheme (see also example on next slide)

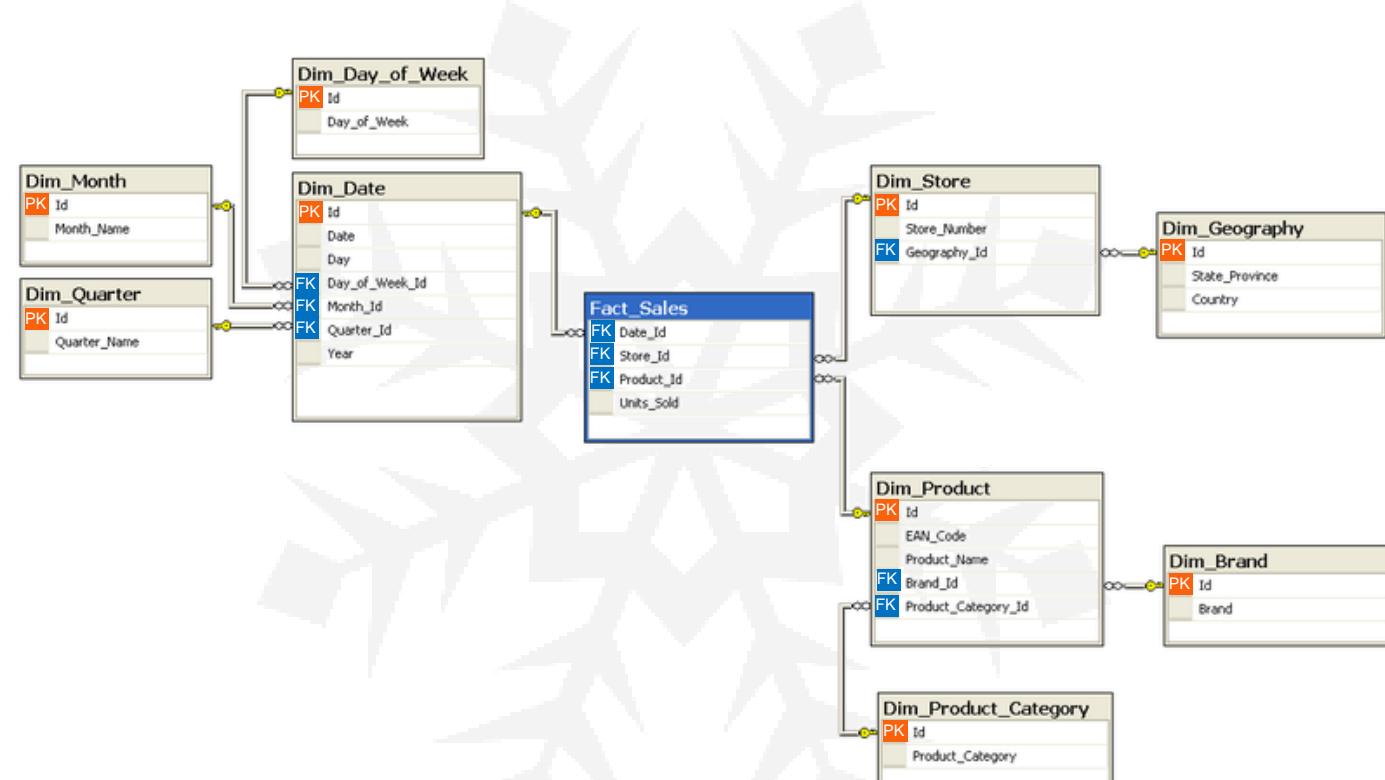
Dimensional tables are broken down into several small lookup-tables  $\Rightarrow$  no double entries

## Consolidated star scheme:

Basic idea: storage of aggregate values **within the fact tables** ("in-table aggregation")

Requires level attributes within the dimensional tables

Provides good response times  $\Rightarrow$  calculations displaced to ETL-procedure (extract, transform, load)

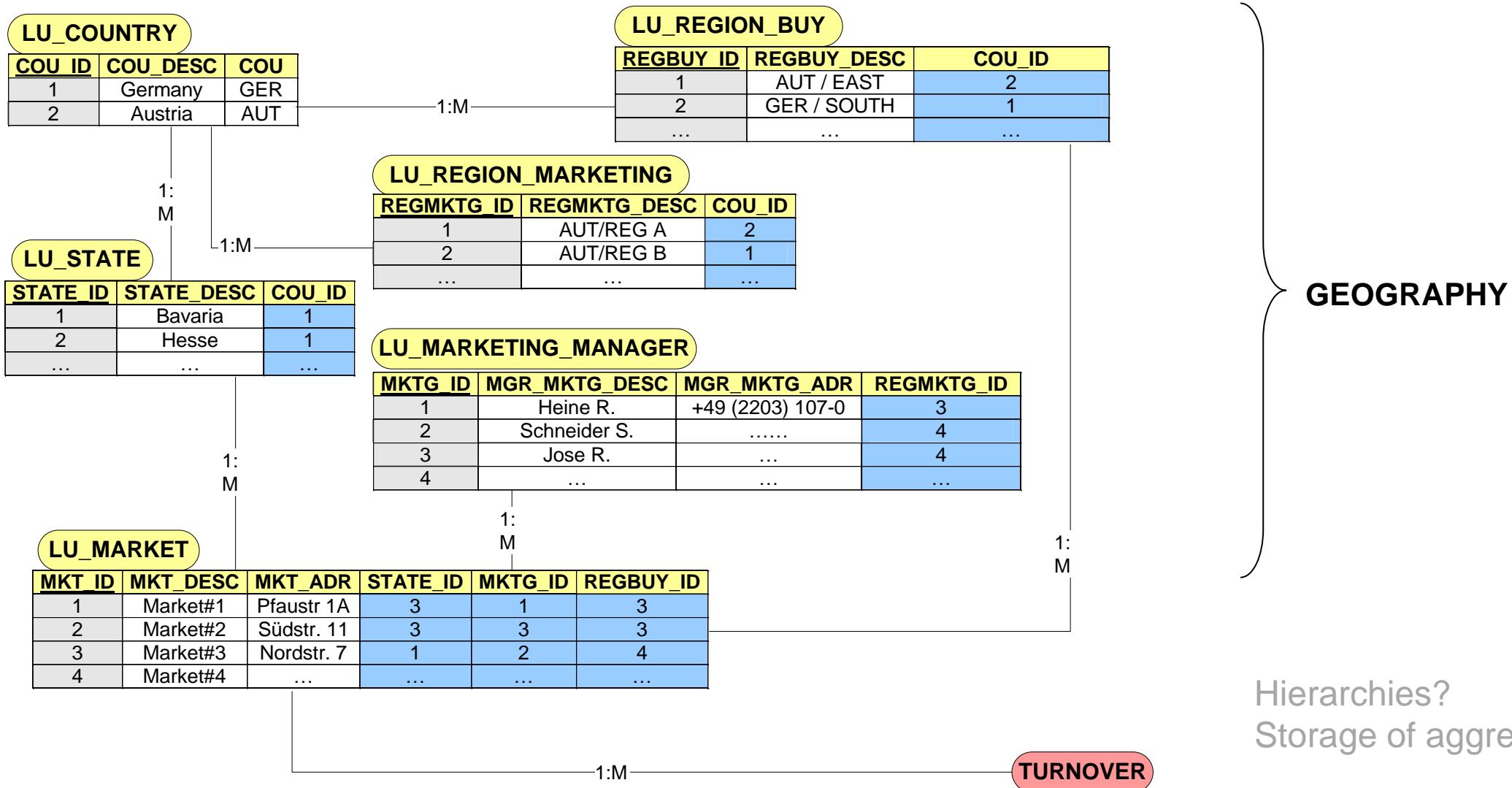


## Fact constellation scheme:

Variant of consolidated star schemes (avoids level-attributes, uses additional fact tables for storage of aggregated values)

PK: Primary Key  
FK: Foreign Key

# Snowflake scheme



# Snowflake scheme

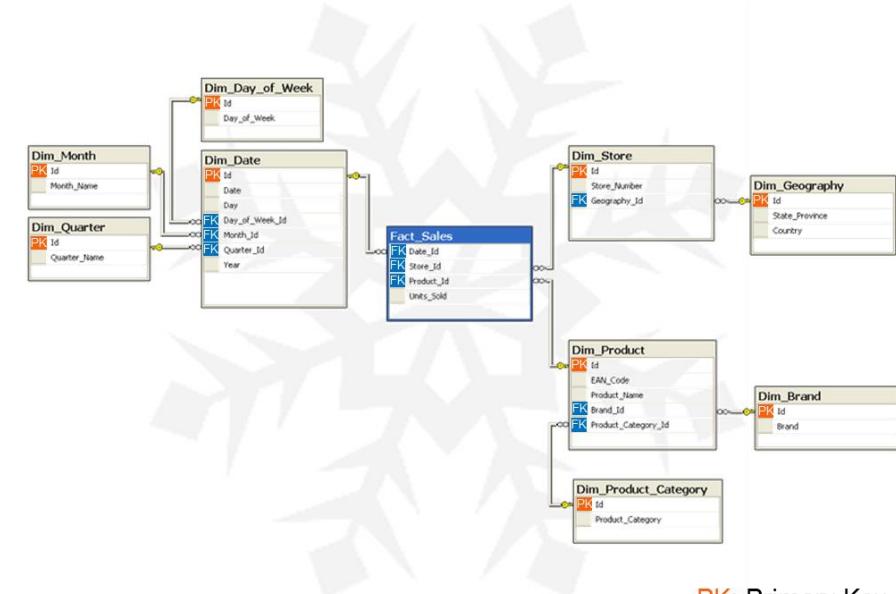
## Pros and Cons

### PRO

- Good support of materialized views
- Browsing can be easily implemented based on snowflake schemes
- No redundancy within the dimension tables

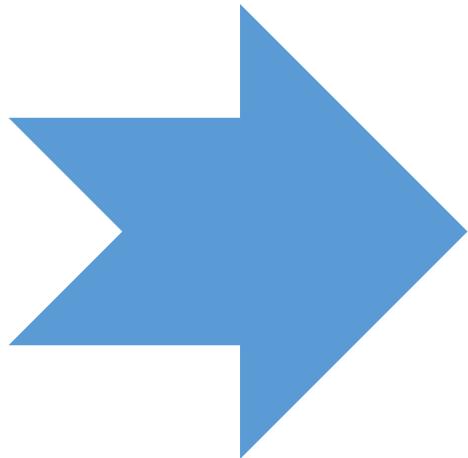
### CON

- More physical join operations required
- More physical tables required
- Higher level of complexity
- ETL-process
- Maintenance
- SQL-queries



PK: Primary Key

FK: Foreign Key



## **Continuation Data Warehouse Modeling**

Basic Elements of  
multidimensional modeling

Conceptual modeling

Logical modeling

**Physical modeling**

# Physical modeling

Database architectures

Physical implementation of logical schemata in a database system.

## Relational database

usually denormalized structure for storage (star scheme)  
„virtual cube“

Examples: MS Access, see  
Genealogy of RDBMS for more

|       |
|-------|
| ..... |
| ..... |
| ..... |
| ..... |

Z.B. Absatz,  
Dimensionen:  
Monat (3),  
Kunde (3),  
Artikel (3)

|           |    |    |           |    |    |           |    |    |           |    |    |           |    |    |           |    |    |           |    |    |           |    |    |           |    |         |  |
|-----------|----|----|-----------|----|----|-----------|----|----|-----------|----|----|-----------|----|----|-----------|----|----|-----------|----|----|-----------|----|----|-----------|----|---------|--|
| 1         | 2  | 3  | 4         | 5  | 6  | 7         | 8  | 9  | 10        | 11 | 12 | 13        | 14 | 15 | 16        | 17 | 18 | 19        | 20 | 21 | 22        | 23 | 24 | 25        | 26 | 27      |  |
|           |    |    |           |    |    |           |    |    |           |    |    |           |    |    |           |    |    |           |    |    |           |    |    |           |    |         |  |
| K1        | K2 | K3      |  |
| Artikel 1 |    |    | Artikel 2 |    |    | Artikel 3 |    |    | Artikel 1 |    |    | Artikel 2 |    |    | Artikel 3 |    |    | Artikel 1 |    |    | Artikel 2 |    |    | Artikel 3 |    |         |  |
|           |    |    |           |    |    |           |    |    | Monat 1   |    |    |           |    |    |           |    |    | Monat 2   |    |    |           |    |    |           |    | Monat 3 |  |

## Select and fine tune the used DBMS technology

Building the database using the specific data definition language (e.g.: SQL-DDL-commands)

Decide on how to index, partition, denormalize and partly pre-aggregate the data.

```
CREATE TABLE employees (
    id INT(6) AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR (50) not null,
    last_name VARCHAR (75) not null,
    age INT(3) not null,
    dateofbirth DATE not null )
```

# Physical modeling

## OLAP architectures

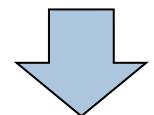
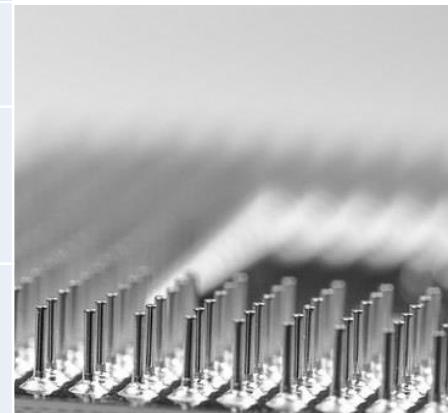
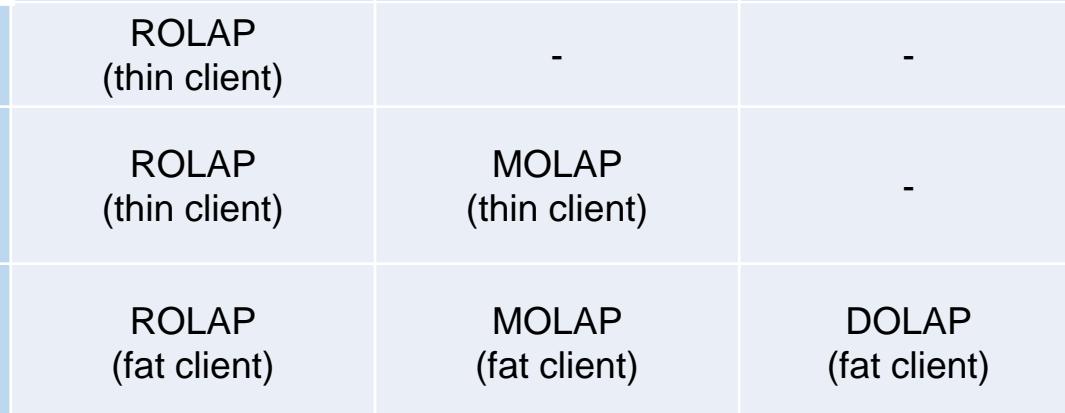
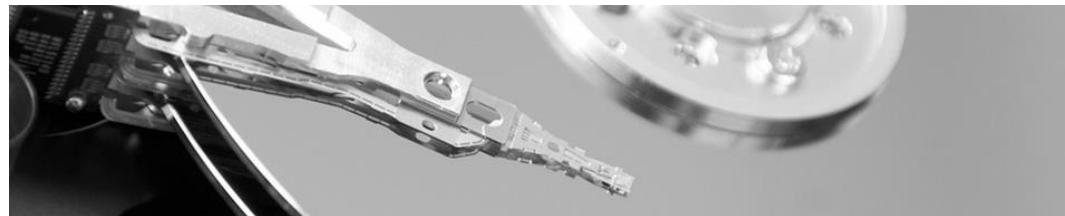
| OLAP - architectures |                                | STORAGE                |                        |                       |
|----------------------|--------------------------------|------------------------|------------------------|-----------------------|
| PROCESSING           | SQL                            | RDBMS                  | MDBMS                  | client-based          |
|                      | multidimensional server engine | ROLAP<br>(thin client) | -                      | -                     |
|                      | client multidimensional engine | ROLAP<br>(fat client)  | MOLAP<br>(thin client) | DOLAP<br>(fat client) |

MOLAP = multidimensional OLAP  
(data in cubes)

ROLAP = relational OLAP

DOLAP = desktop OLAP

Ref. e.g., Jukic et al. (2008)



Recent developments  
in data warehousing,  
see article "Paradigmenwechsel:  
Data Warehouses für die Cloud"  
*Kursmaterial > Readings/Übungen*

# Fragen?

- ✓ Data Warehouse Modeling

# Exercise

## Conceptual Modeling and Logical Model (Star Scheme)

Design a **logical model** for the local Food Company:

### Conny's Corner Shop

Your managers need to keep themselves up to date on the company's situation. They especially want to keep an eye on their products with regards to location, and time.

- Conny's Corner Shop sells beverages and snacks. Both categories have different types of products, such as juices and water, as well as pretzels and crackers. The products are either branded as strictly vegan or organic (bio).
- Products are sold in different cities, regions and EU-countries
- Most managers are interested in quantities, income and discounts of sales.

Create a simplified Star Scheme (logical model) with tables/relations.  
Use your conceptual model as input.

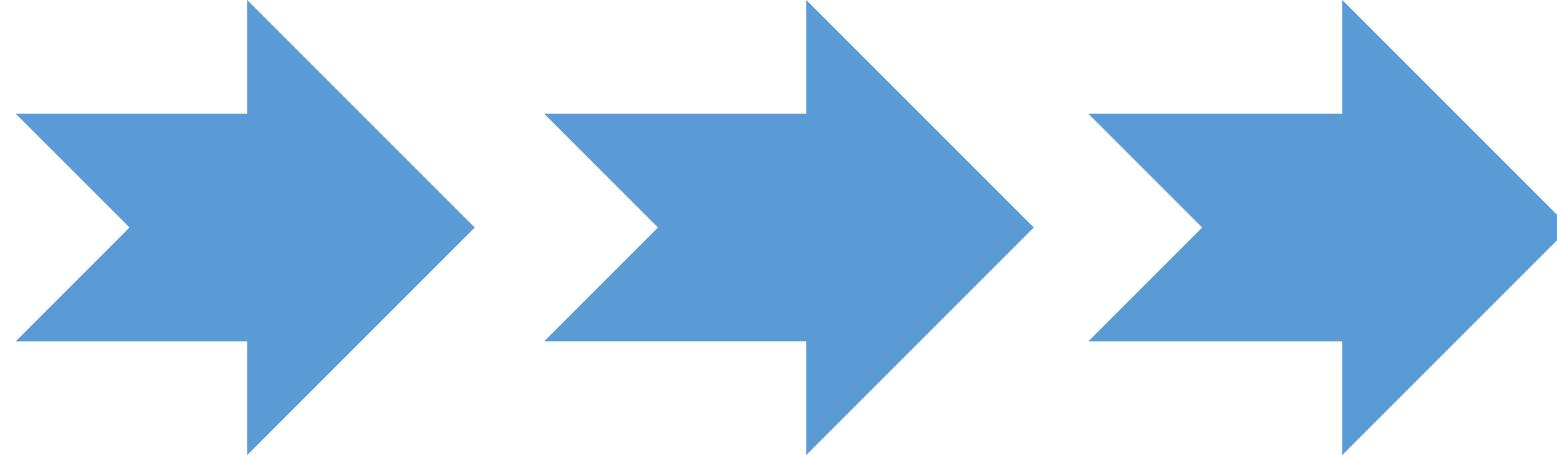
10 Min.

# Outlook next lesson

## Data Mining Introduction



Where are the limits of  
the handling of data  
considered so far?



(1) The need for data mining

(2) From business problems to data mining tasks

(3) Supervised vs. unsupervised methods

*"Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner."*

(Hand, Mannila, Smyth (2001), Principles of Data Mining)



# The need for data mining

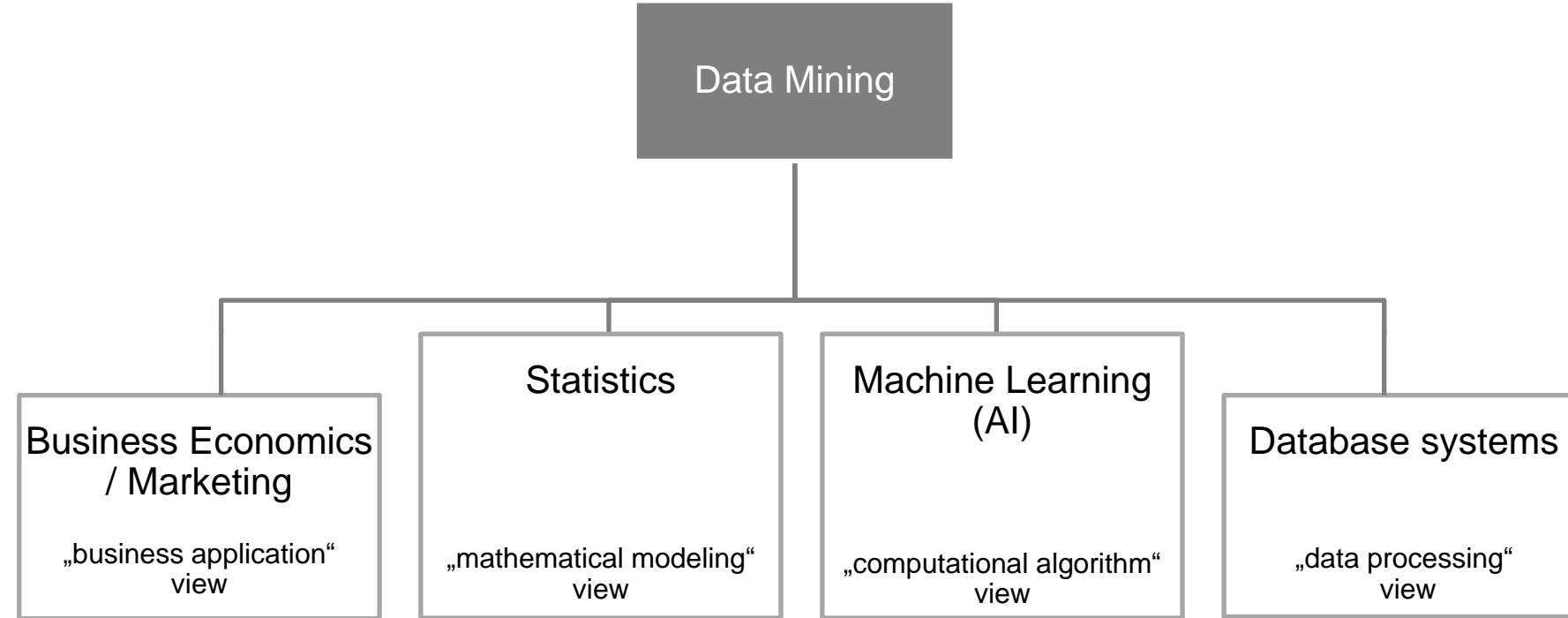
“80% of the knowledge of interest in a business context can be extracted from data using conventional tools.” (Lusti)

- reporting
  - query-languages (SQL, QBE, ...)
  - OLAP and spreadsheets

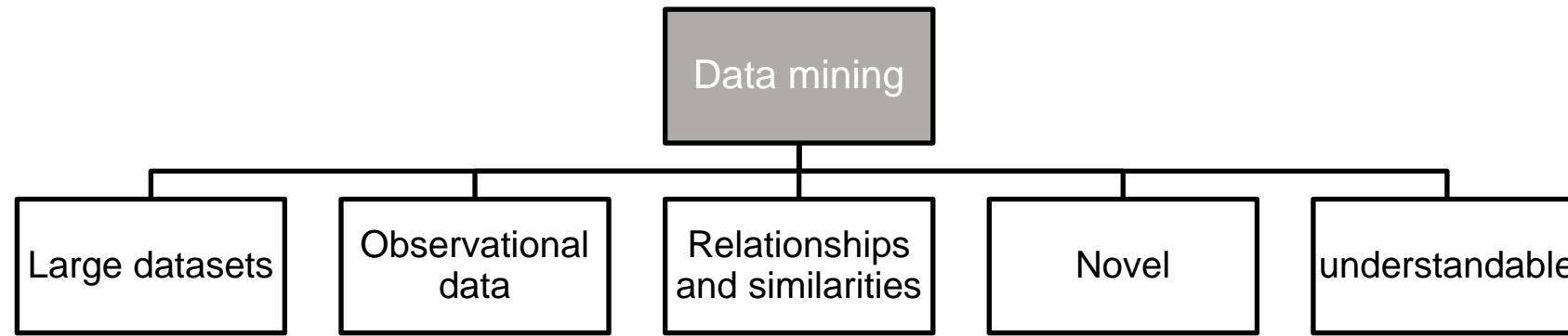
### **Disadvantages of conventional tools:**

- Often, merely simple questions can be answered
  - OLAP: query-focused and low complexity of analysis  
*e.g., performance changes are visible, but what about the overall context/ reasons?*
  - Automation of knowledge discovery is difficult  
*hypothesis needed*
  - Only small amounts of data may be handled  
(esp. spreadsheets)  
*but exploding amount of raw data available*

# Major roots of data mining

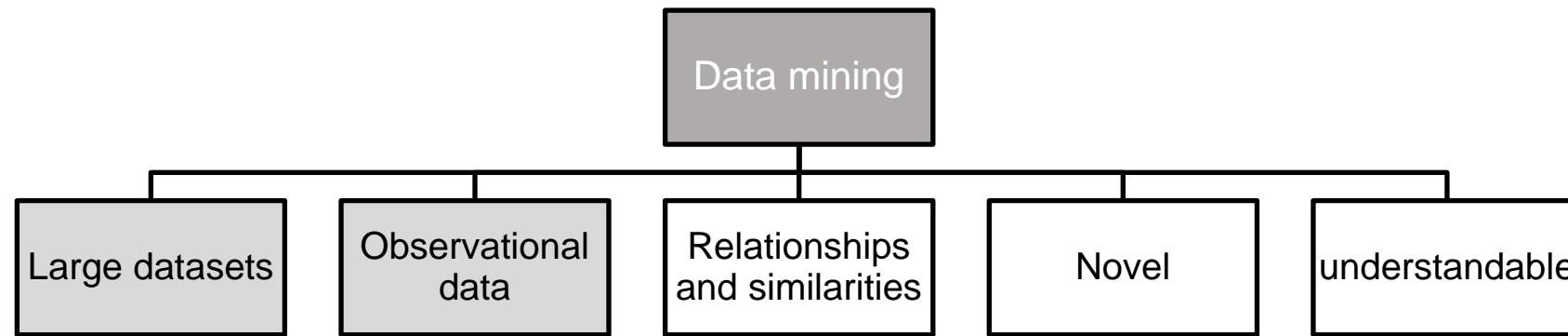


# Data mining: definition (1/3)



*“Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.”*

(Hand, Mannila, Smyth (2001), Principles of Data Mining)

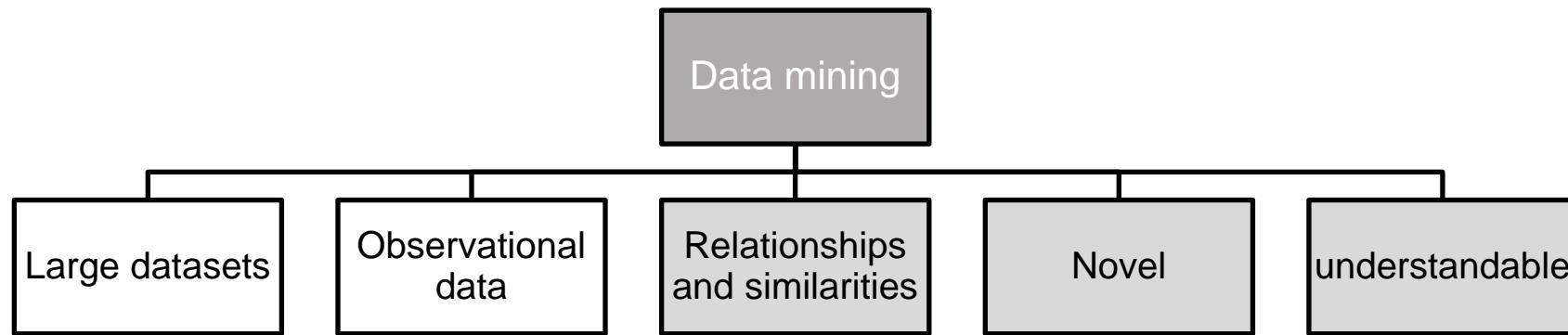


## Often large datasets:

- Small datasets ⇒ exploratory data analysis in statistics
- Large datasets (as they exist in DWHs) provoke new problems
  - Storage and access of data
  - Runtime issues
  - Determination of representativeness of data
  - Difficulty to decide whether an apparent relationship is merely a chance occurrence or not

## Observational data:

- Data often collected for some other purpose than data mining
- Objectives of the data mining exercise play no role in data collection strategy
  - e.g., DWH data relying on an airline reservation system or a bank account administration system
  - opposite: experimental data (as it is used quite often in statistics)



## Relationships and summaries:

often referred to as **models** or **patterns**

e.g., linear equations, tree structures, clusters,  
patterns in time series, ...

## Understandable:

Novelty is not sufficient to qualify  
relationships worth finding

Simple relationships may be preferred to  
complicated ones

## Novel:

Novelty should be measured relative to users prior  
knowledge

# Exercise: Data Mining vs. OLAP

Typical questions

| Fragestellung | Data Mining  | OLAP  |
|---------------|--|---|
| Kundenwert    | Welche Kunden bieten uns das größte Deckungsbeitragspotenzial? | Wer waren letztes Jahr unsere 10 besten Kunden? |

Kahoot-Fragen

[www.kahoot.it](http://www.kahoot.it)

(über Smartphone oder Laptop)

PIN folgt

(Diese Folie ist nach der Vorlesung mit Lösungen verfügbar)

## Fragen?

- Data Mining Introduction  
(will be continued in the next lesson)

# Todo for next Week

Support Conny's Corner Shop by creating a (or finishing the) simplified Star Scheme (logical model) with tables/relations.

*See exercise on slide 23*

# Bibliography



- Hand, David J., Heikki Mannila, and Padhraic Smyth. *Principles of data mining*. MIT press, 2001.
- Vaisman, A., & Zimányi, E. (2014). *Data Warehouse Systems*. Springer, Heidelberg

# Recommended reading (for next lessons)



## Data Mining

Provost, F. Chapter 2

Fawcett, T.

Berthold et al. Chapters 1, B, C

Lusti, M. Data Warehousing und Data Mining (Chapter 6)

Hand, D. et al.: Principles of Data Mining (esp. Chapters 1, 5, 6 and 11)



# Business Intelligence

## 05a Data Mining Introduction (continued)

Prof. Dr. Bastian Amberg  
(summer term 2024)

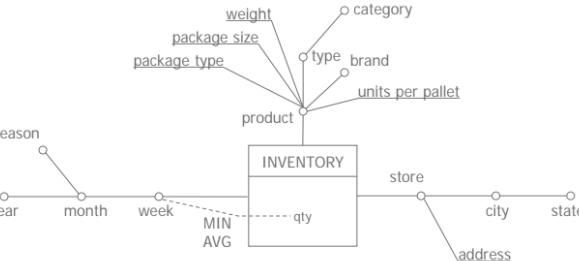
15.5.2024

# Schedule

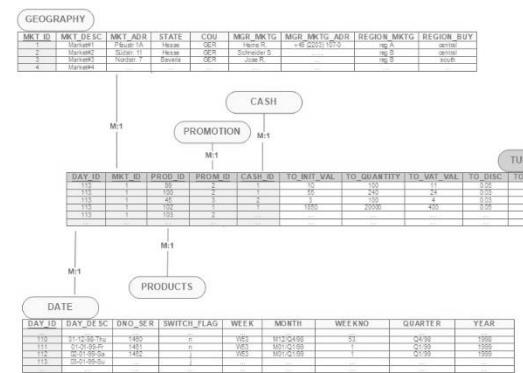
|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study   |  |                  |         |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|--------------|--|------------------|---------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |              |  | Python-Basics    | Chap. 1 |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |              |  |                  | Chap. 2 |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  | Data Warehouse Modeling I  |              |  |                  | Chap. 3 |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining   | Introduction |  |                  |         |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |              |  | Python-Analytics | Chap. 1 |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization              |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam                 |              |  |                  | Chap. 2 |  |  |
|           | W7                | 29.5. | Data Preparation                                    |                                   | 31.5. |   |              |  |                  |         |  |  |
|           | W8                | 5.6.  | Predictive Modeling I                               |                                   | 7.6.  | Predictive Modeling II (10:00 -12:00)   |              |  | BI-Project       | Start   |  |  |
|           | W9                | 12.6. | Fitting a Model I                                   |                                   | 14.6. | Python-Analytics-Online Exercise  |              |  |                  |         |  |  |
|           | W10               | 19.6. | Guest Lecture                                       |                                   | 21.6. | Fitting a Model II  |              |  |                  |         |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |              |  |                  |         |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I  |              |  |                  |         |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |              |  |                  |         |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |              |  |                  | End     |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin ~ 22.7. bis 3.8.<br>Klausur 2.Termin ~ 23.9. bis 5.10.                                       |              |  | Projektbericht   |         |  |  |

- ✓ How can **multidimensional data models** be **developed** and **stored**?

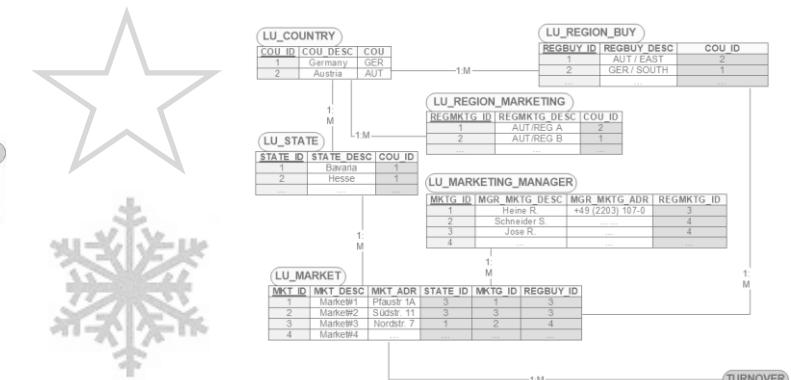
## Conceptual modeling



## Logical modeling



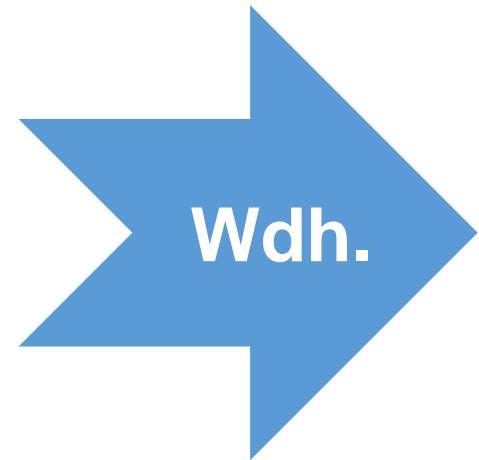
## Physical modeling



- Introduction **Data Mining**

Where are the limits of the handling of data considered so far?

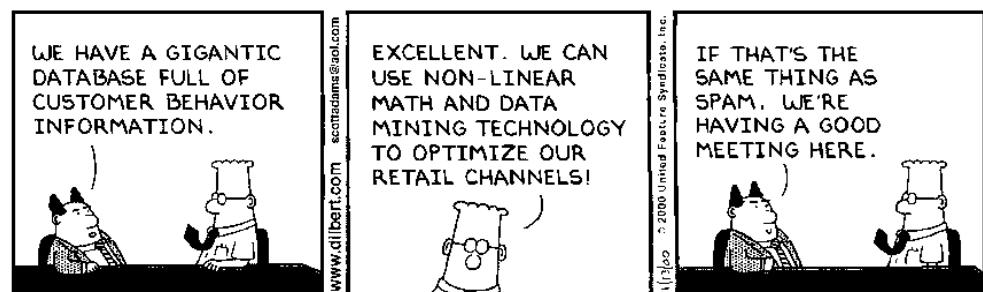
*“Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.”*  
(Hand, Mannila, Smyth (2001), Principles of Data Mining)



## (1) The need for data mining

(2) From business problems to data mining tasks

(3) Supervised vs. unsupervised methods



# The need for data mining

“80% of the knowledge of interest in a business context can be extracted from data using **conventional tools**.” (Lusti)

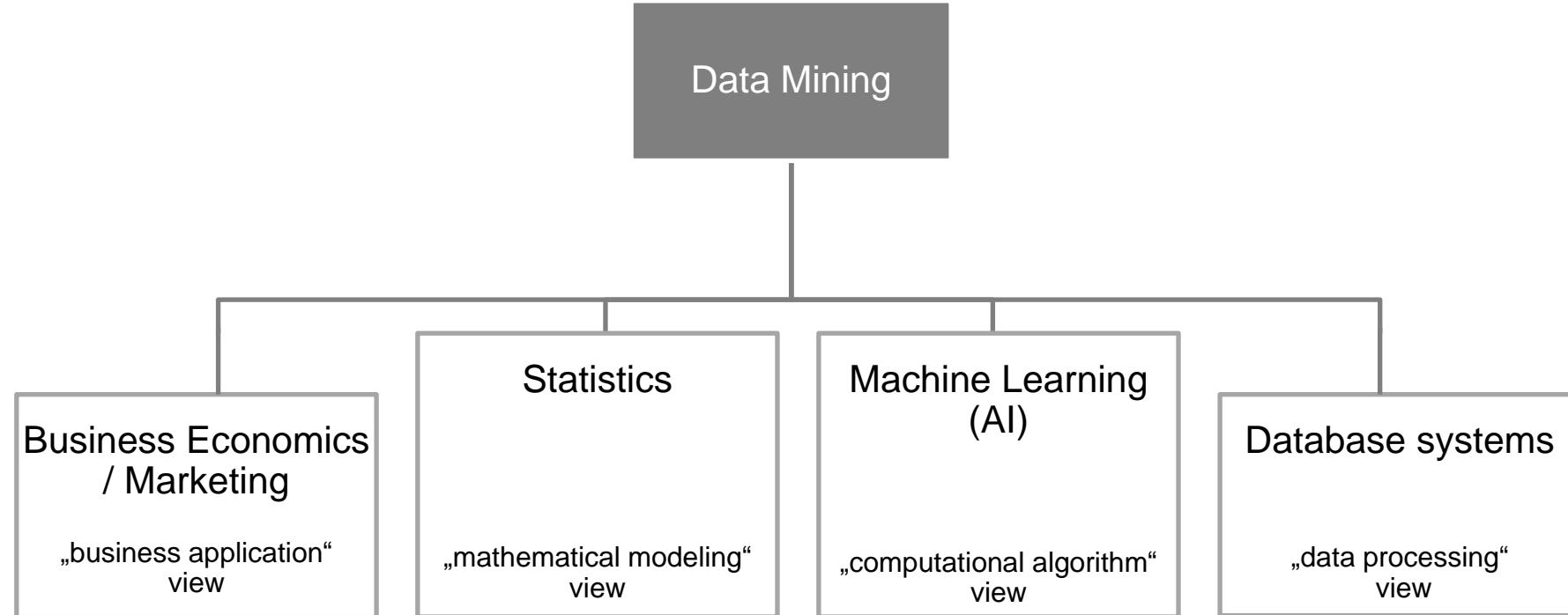
- reporting
  - query-languages (SQL, QBE, ...)
  - OLAP and spreadsheets

| Work Breakdown Structure |  |                            |           |           |           |                  |
|--------------------------|--|----------------------------|-----------|-----------|-----------|------------------|
| Resource Request         |  |                            |           | Resource  |           |                  |
| Phase/Type               | Description  | Resource                   | Allocated | Allocated | Allocated | Allocated        |
| 1.0 Phase                | Define Efficiency requirement on the feature requirement and creating the process. | RPT                        |           |           |           | Resource Manager |
| 1.0 Module               | Create the Resource Requirement  | RPT                        |           |           |           |                  |
| 1.0.1 Phase              | Create the Product Requirement   | Product Manager            |           |           |           |                  |
| 1.0.1.1 Phase            | Proposed and Review the Product Requirement  | Product Manager            |           |           |           |                  |
| 1.0.1 Module             | Elaborate the Product  | RPT                        |           |           |           |                  |
| 1.0.1.1 Phase            | Elaborate the Product Structure  | Product Manager            |           |           |           |                  |
| 1.0.1.1.1 Phase          | Product Manager Approved   | RPT                        |           |           |           |                  |
| 2.0 Phase                | Elaborate the user and product including goals                                     | RPT                        |           |           |           |                  |
| 2.0 Module               | Create the User Work Plan  | RPT                        |           |           |           |                  |
| 2.0.1 Phase              | Create the User Work Plan  | User Manager               |           |           |           |                  |
| 2.0.1.1 Phase            | Create the Work Definition   | Work Definition Manager    |           |           |           |                  |
| 2.0.1.1.1 Phase          | Create the Work Definition Structure   | Work Definition Manager    |           |           |           |                  |
| 2.0.1.1.2 Phase          | Create the Product Definition  | Product and Design Manager |           |           |           |                  |
| 2.0.1.1.3 Phase          | Refine the Product Details and Requirements  | Product and Design Manager |           |           |           |                  |
| 2.0.1.1.4 Phase          | Review Product & Design Requirements   | Review                     |           |           |           |                  |
| 2.0.1.2 Module           | Create the Including Plans for Features  | RPT                        |           |           |           |                  |

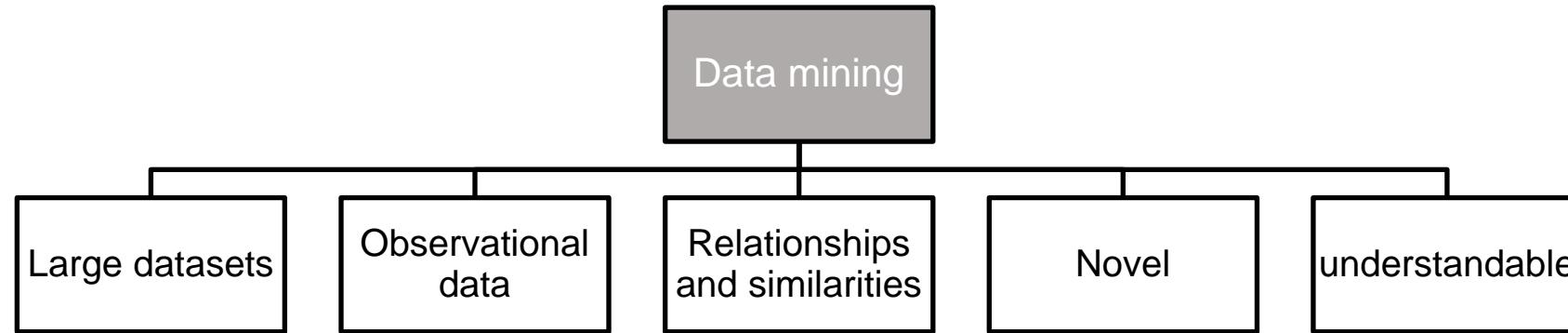
## Disadvantages of conventional tools:

- Often, merely simple questions can be answered
  - OLAP: query-focused and low complexity of analysis  
*e.g., performance changes are visible, but what about the overall context/ reasons?*
  - Automation of knowledge discovery is difficult  
*hypothesis needed*
  - Only small amounts of data may be handled  
(esp. spreadsheets)  
*but exploding amount of raw data available*

# Major roots of data mining



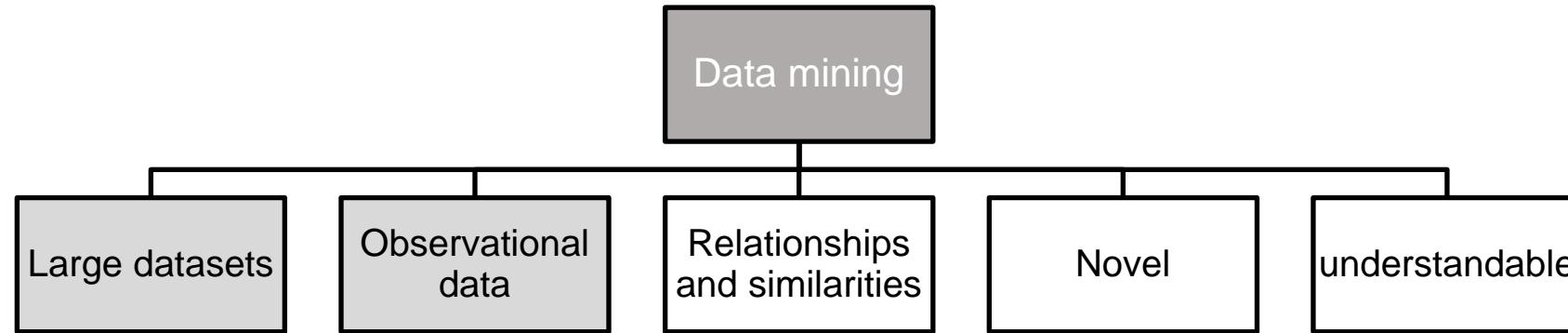
# Data mining: definition (1/3)



*“Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.”*

(Hand, Mannila, Smyth (2001), Principles of Data Mining)

# Data mining: definition (2/3)



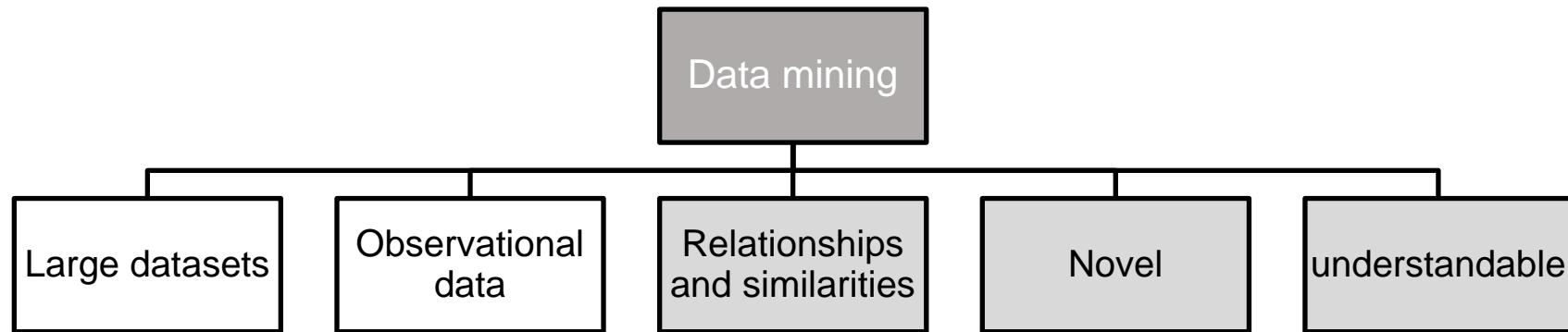
## Often large datasets:

- Small datasets ⇒ exploratory data analysis in statistics
- Large datasets (as they exist in DWHs) provoke new problems
  - Storage and access of data
  - Runtime issues
  - Determination of representativeness of data
  - Difficulty to decide whether an apparent relationship is merely a chance occurrence or not

## Observational data:

- Data often collected for some other purpose than data mining
- Objectives of the data mining exercise play no role in data collection strategy
  - e.g., DWH data relying on an airline reservation system or a bank account administration system
  - opposite: experimental data (as it is used quite often in statistics)

# Data mining: definition (3/3)



## Relationships and summaries:

often referred to as **models** or **patterns**

e.g., linear equations, tree structures, clusters,  
patterns in time series, ...

## Understandable:

Novelty is not sufficient to qualify  
relationships worth finding

Simple relationships may be preferred to  
complicated ones

## Novel:

Novelty should be measured relative to users prior  
knowledge

# Exercise: Data Mining vs. OLAP

Typical questions

| Fragestellung | Data Mining  | OLAP  |
|---------------|--|---|
| Kundenwert    | Welche Kunden bieten uns das größte Deckungsbeitragspotenzial? | Wer waren letztes Jahr unsere 10 besten Kunden? |

Kahoot-Fragen

[www.kahoot.it](http://www.kahoot.it)

(über Smartphone oder Laptop)

PIN folgt

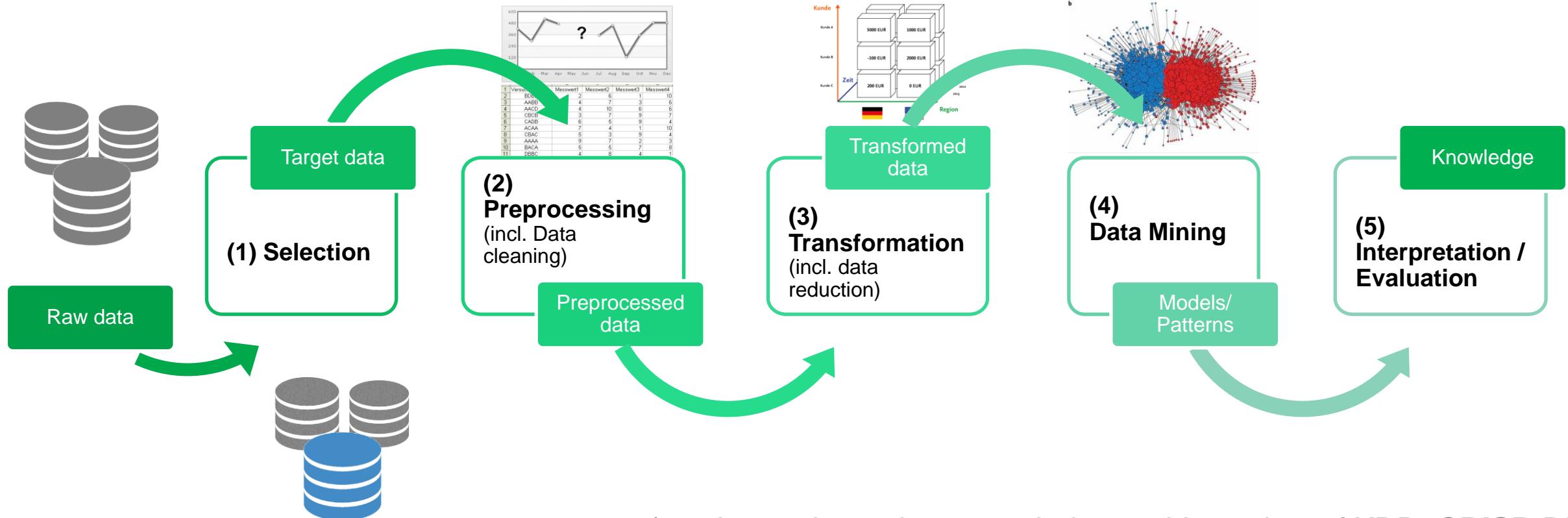
(Diese Folie ist nach der Vorlesung mit Lösungen verfügbar)

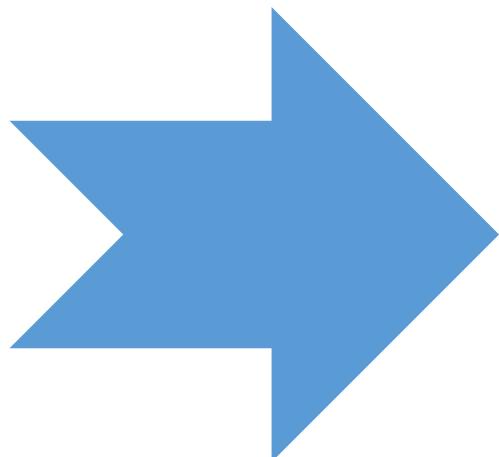
# A typical Data Mining Process

Knowledge discovery in databases (KDD)

Data mining is often set in the broader context of **knowledge discovery in databases** (KDD)

The precise boundaries of the data mining part within the KDD process are not easy to state





(1) The need for data mining

**(2) From business problems to data mining tasks**

(3) Supervised vs. unsupervised methods

# From business problems to data mining

Data mining is a **process** with well-understood stages based on

- application of information technology
- analyst's creativity
- business knowledge
- common sense

Counterexample: Youtube ads, see e.g. [The Guardian, 2017](#)

*"Major brands ... pulled their ads after they were found to be appearing next to videos promoting extremist views or hate speech"*

We look at typical **tasks** and examples, then at the **process**

**Decompose** a data analytics problem into pieces such that you can solve a known task with a tool

There is a large number of data mining algorithms available, but only a limited number of data mining tasks

We will illustrate the **fundamental concepts** based on

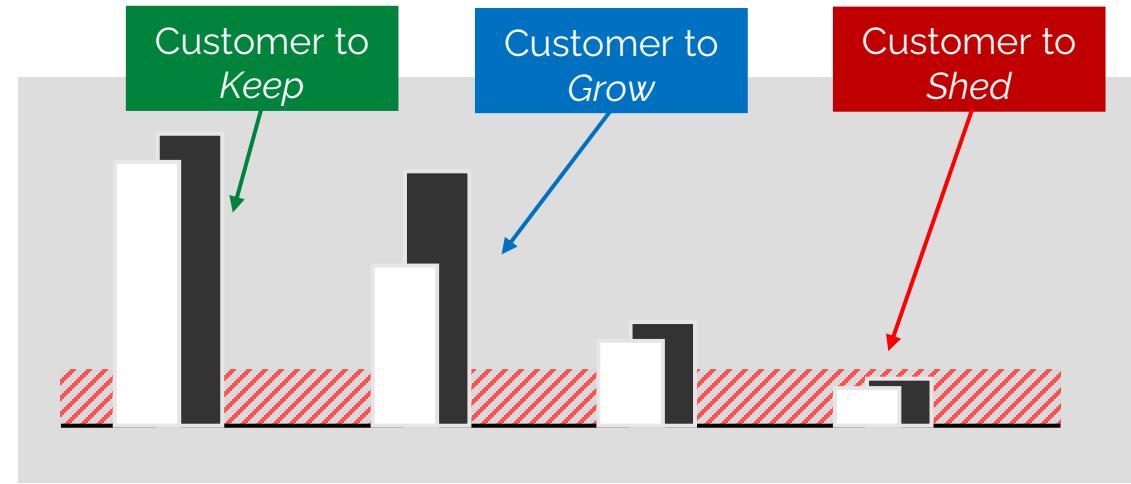
- Classification
- Regression

# Classification

Classification attempts to **predict**, for each individual in a population, which of a (small) set of classes that individual belongs to

*“Among all the customers of a cellphone company, which are likely to correspond to a given offer?”*  
~ predict whether something will happen

Classification algorithms provide models that determine which class a **new** individual belongs to (and its probability).



Classification is related to scoring (for instance in Customer Relationship Management) as the underlying value is categorical.



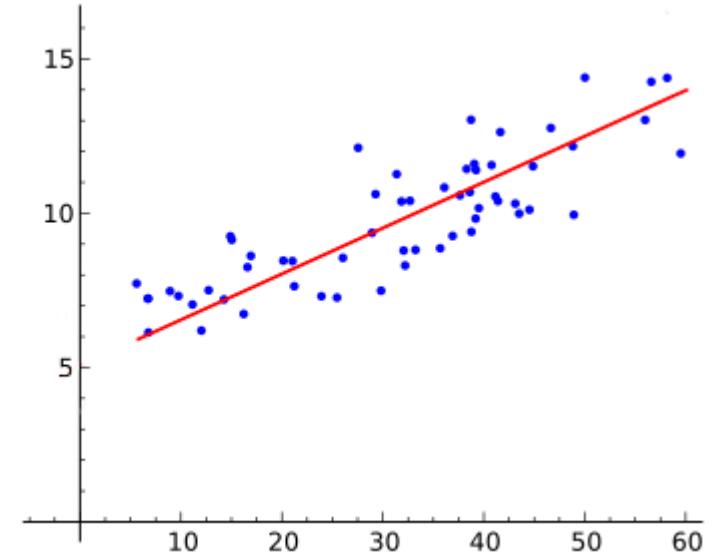
Regression (value estimation) attempts to estimate or **predict**, for each individual, the (continuous) *numerical value* for that individual

*“How much will a given customer use the service?”*

Predicted variable: service usage

~ *predict how much something will happen*

Generate regression model by looking at other, similar individuals in the population



Be careful:  
linear regression vs. logistic regression

# Exercise: Classification or Regression problem?

## Examples

- a) Will this customer purchase service  $S_1$  if given incentive  $I$ ?
- b) Which service package ( $S_1$ ,  $S_2$ , or none) will a customer purchase if given incentive  $I$ ?
- c) How much time will this customer spend on our web service?
- d) How long after buying a product can a repeat purchase by customer X be expected?
- e) Which potentially profitable customers are most likely to move to a competitor?

3 Min.

# Another fundamental data mining task: ...



Quelle: [www.welt.de](http://www.welt.de)

# Clustering

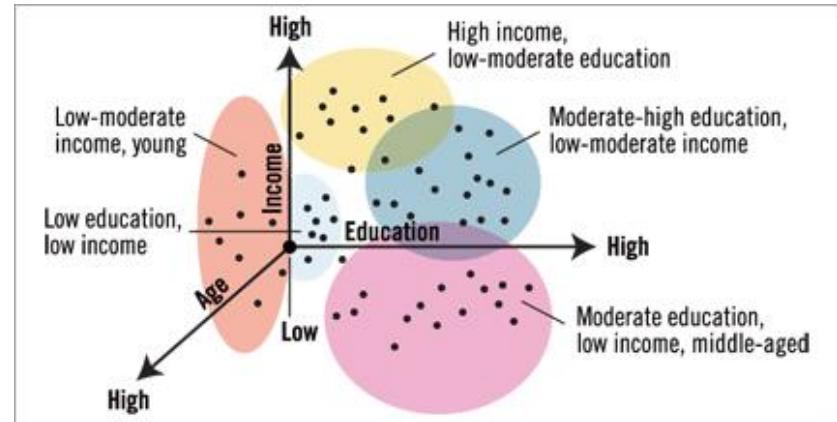
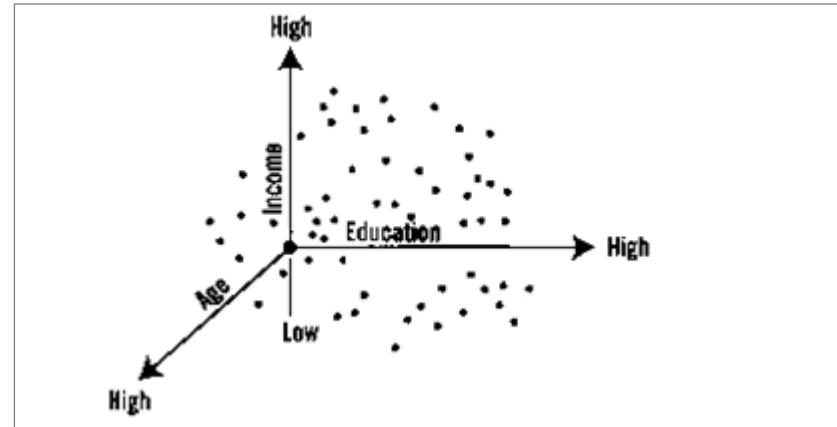


Clustering attempts to **group** individuals in a population together by their similarity, but *without regard to any specific purpose*

*Do customers form natural groups or segments?*

Result: groupings of the individuals of a population

Useful in preliminary domain exploration



# Co-occurrence grouping

Attempts to find associations between entities based on transactions involving them aka **association rules** or **market-basket analysis**

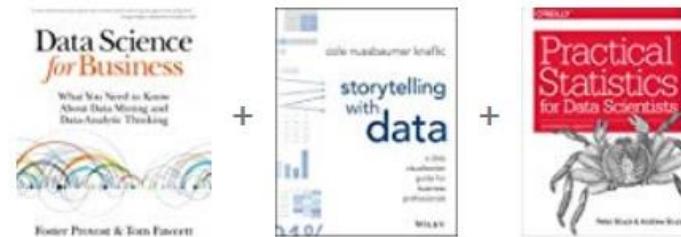
*“What items are commonly purchased together?”*

Considers similarity of objects based on their appearing together in transactions

Included in recommendation systems (people who bought X also bought Y)

Result: a description of items that occur together

Wird oft zusammen gekauft



Gesamtpreis: EUR 95,23

Alle drei in den Einkaufswagen

Customers who bought this item also bought

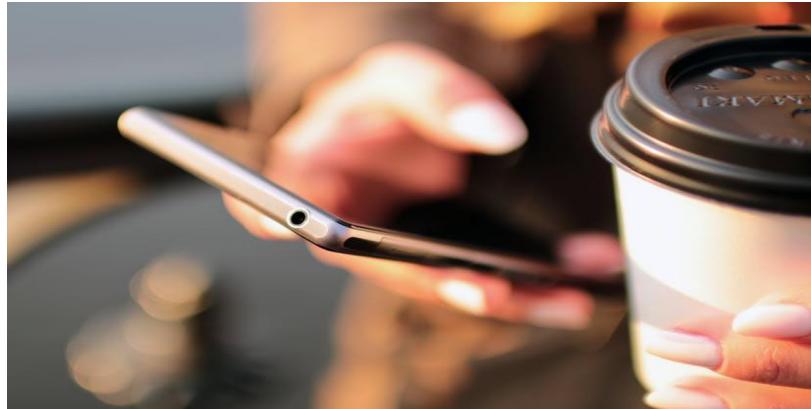


# Some more data mining tasks

- **Profiling** attempts to characterize the typical behavior of a group or population, aka **behavior description**

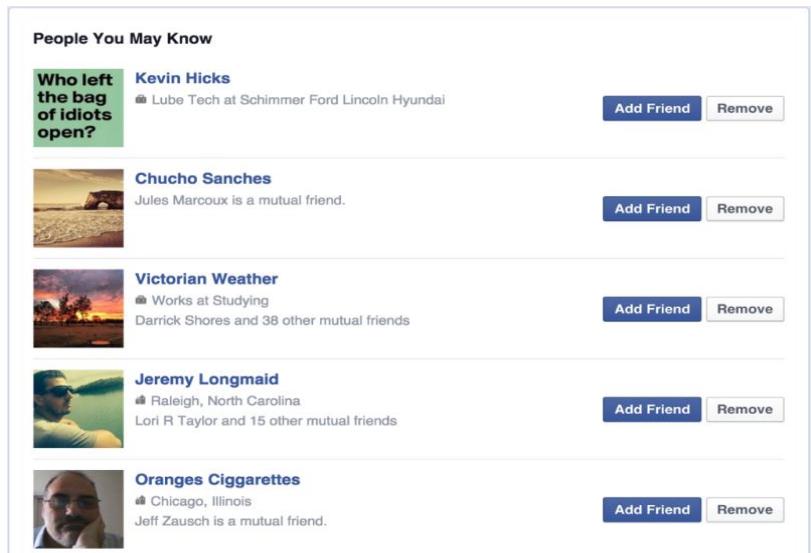
*"What is the typical cellphone usage of this customer segment?"*

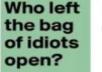
Often used to establish behavioral norms for anomaly detection (fraud detection)



- **Link prediction** attempts to predict connections between data items (→ social network systems)

*"Since you and Karen share ten friends, maybe you'd like to be Karen's friend?"*



| People You May Know  |   |
|--|---|
|  Who left the bag of idiots open? | <b>Kevin Hicks</b><br>Lube Tech at Schimmer Ford Lincoln Hyundai<br><a href="#">Add Friend</a> <a href="#">Remove</a>     |
|  Chucho Sanches                   | Jules Marcoux is a mutual friend.<br><a href="#">Add Friend</a> <a href="#">Remove</a>                                    |
|  Victorian Weather               | Works at Studying<br>Derrick Shores and 38 other mutual friends<br><a href="#">Add Friend</a> <a href="#">Remove</a>      |
|  Jeremy Longmaid                | Raleigh, North Carolina<br>Lori R Taylor and 15 other mutual friends<br><a href="#">Add Friend</a> <a href="#">Remove</a> |
|  Oranges Cigarettes             | Chicago, Illinois<br>Jeff Zausch is a mutual friend.<br><a href="#">Add Friend</a> <a href="#">Remove</a>                 |



(1) The need for data mining

(2) From business problems to data mining tasks

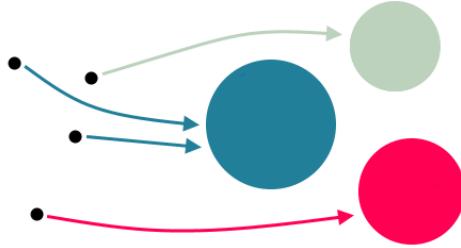
**(3) Supervised vs. unsupervised methods**

# Supervised vs. unsupervised

## Supervised Learning

*“Can we find groups of customers who have particularly high likelihoods of cancelling their service soon after their contracts expire?”*

→ specific target

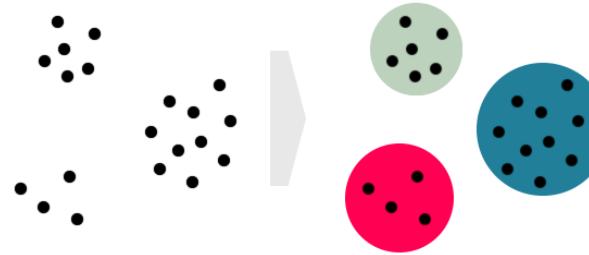


Supervised and unsupervised tasks require different techniques

## Unsupervised Learning

*“Do our customers naturally fall into different groups?”*

→ no specific target



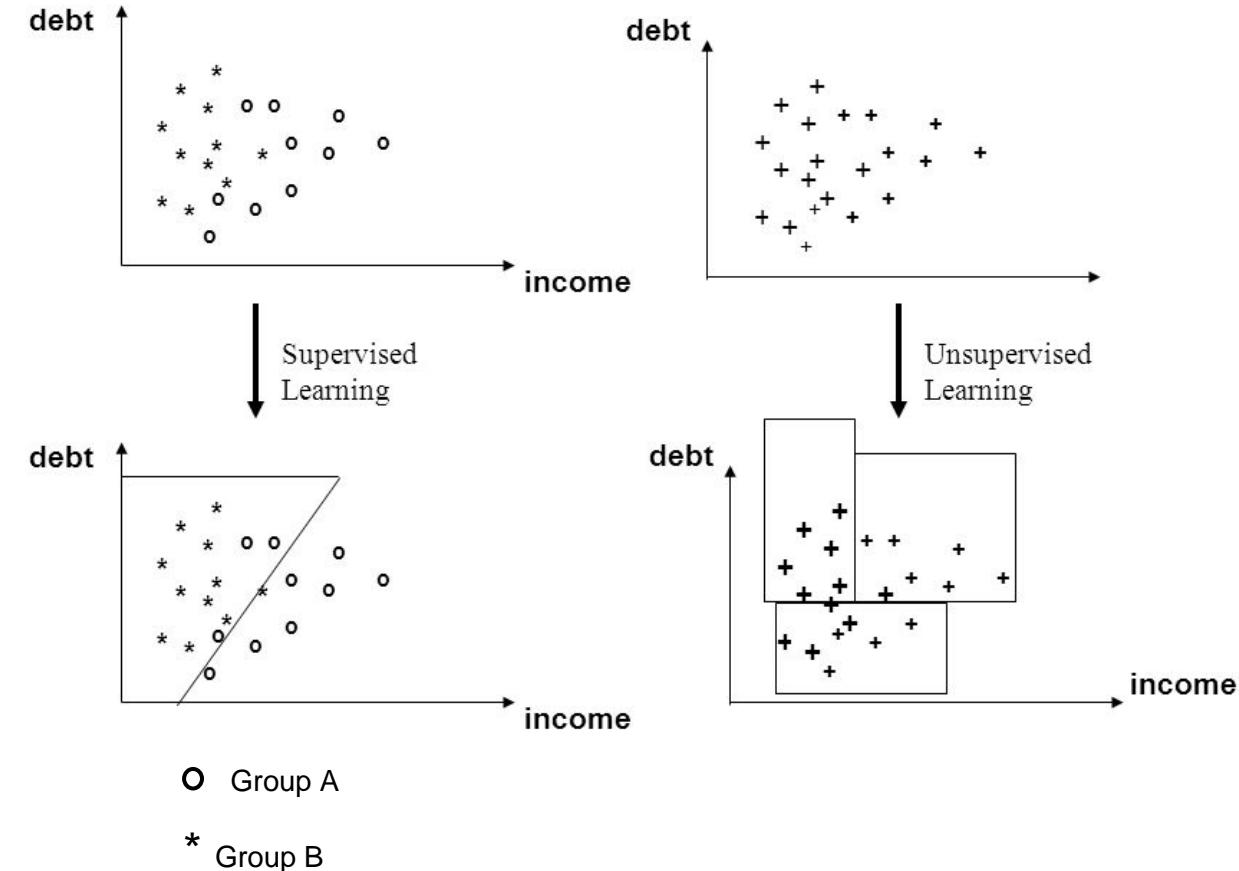
There is no guarantee that unsupervised tasks provide meaningful results

# Supervised and unsupervised techniques

Classification and regression are generally solved with **supervised** techniques

Clustering, co-occurrence grouping, and profiling are generally **unsupervised**

Similarity matching and link prediction could be either



# Supervised vs. Unsupervised vs. Reinforcement Learning

Search by yourself

e.g. <https://towardsdatascience.com/machine-learning-101-supervised-unsupervised-reinforcement-beyond-f18e722069bc>

Learning Approach

Learn from.... (input? - output?)

Learn for.... (target?)

Example?

Supervised

Unsupervised

Reinforcement

e.g., [Spiegel Online, 2021](#)

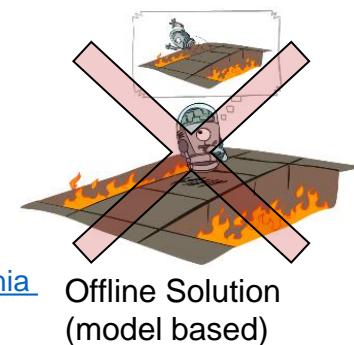
“KI zockt besser als der Mensch“

Corresponding article

Ecoffet, A., Huizinga, J., Lehman, J. et al.

First return, then explore.

[Nature 590, 580–586 \(2021\)](#).



Quelle: Course188 Intro to AI at UC Berkeley

10 Min.

➤ [A.I. Learns to Drive From Scratch in Trackmania](#)  
(very well explained youtube video)

➤ [Demo Reinforcement Learning](#)

Ref.

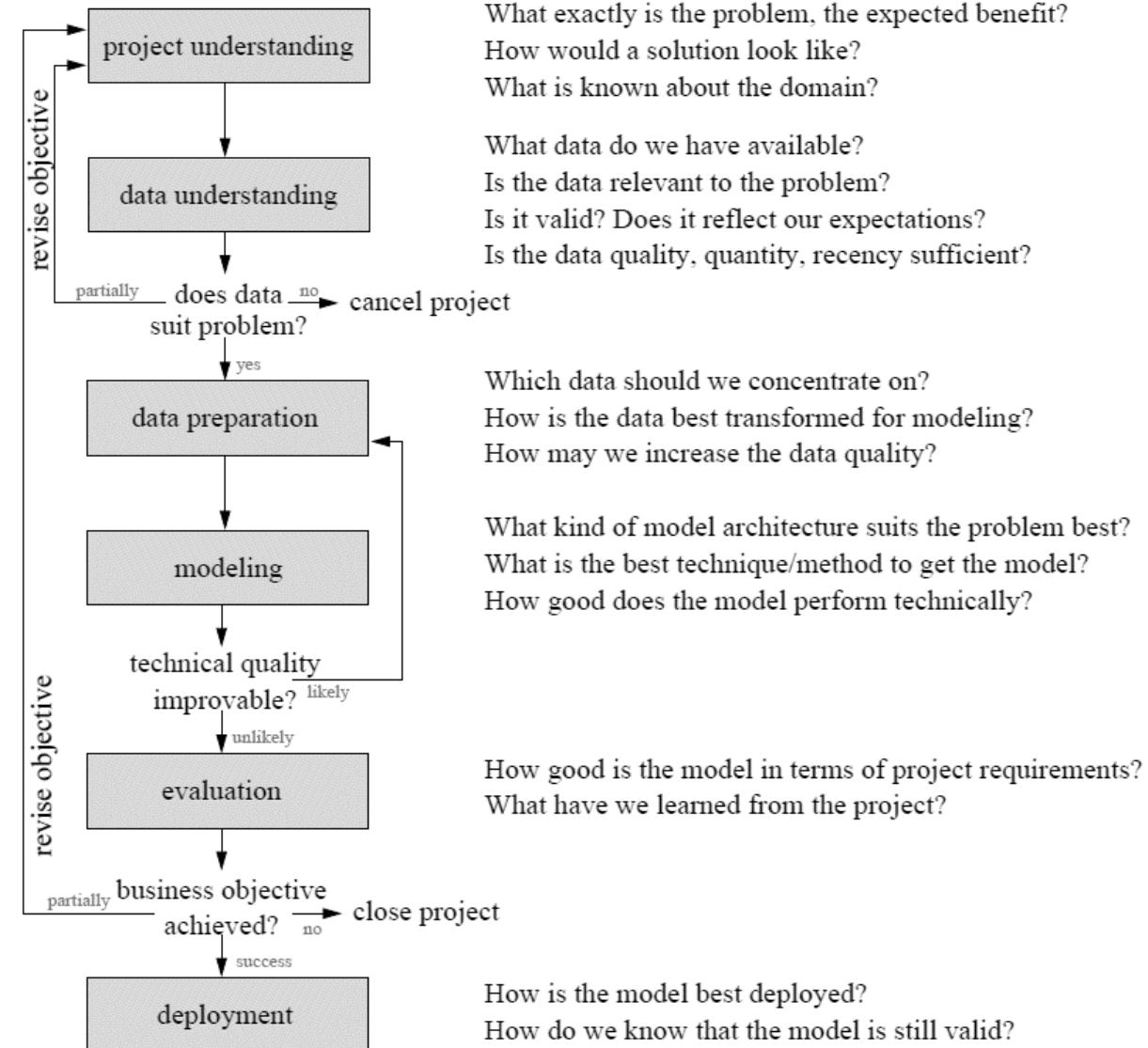
# Outlook: The data mining process - CRISP-DM

Cross  
Industry  
Standard  
Process for  
Data  
Mining

Iteration as  
a rule

Process of data  
exploration

Implementation of the  
KDD Process



What exactly is the problem, the expected benefit?

How would a solution look like?

What is known about the domain?

What data do we have available?

Is the data relevant to the problem?

Is it valid? Does it reflect our expectations?

Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?

How is the data best transformed for modeling?

How may we increase the data quality?

What kind of model architecture suits the problem best?

What is the best technique/method to get the model?

How good does the model perform technically?

How good is the model in terms of project requirements?

What have we learned from the project?

How is the model best deployed?

How do we know that the model is still valid?

## Fragen?

- ✓ The need for data mining
- ✓ From business problems to data mining tasks
- ✓ Supervised vs. unsupervised methods (vs. reinforcement learning)
  - The data mining process – CRISP-DM

Provost, F. Chapter 2

Fawcett, T.

Berthold et al. Chapters 1, B, C

Lusti, M. Data Warehousing und Data Mining (Chapter 6)

Hand, D. et al.: Principles of Data Mining (esp. Chapters 1, 5, 6 and 11)



# Bibliography

- Azevedo, A. I. R. L. (2008). KDD, SEMMA and CRISP-DM: a parallel overview. *IADS-DM*.
- Ecoffet, A., Huizinga, J., Lehman, J. et al. (2021) First return, then explore. *Nature* 590, 580–586.  
<https://doi.org/10.1038/s41586-020-03157-9>
- Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3), 37.
- Hand, David J., Heikki Mannila, and Padhraic Smyth. *Principles of data mining*. MIT press, 2001.
- Wirth, R., & Hipp, J. (2000, April). CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (pp. 29-39).



# Business Intelligence

## 05b CRISP-DM – Project Understanding

Prof. Dr. Bastian Amberg  
(summer term 2024)  
15.5.2024

# Schedule

|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study   |  |                  |         |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|--------------|--|------------------|---------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |              |  | Python-Basics    | Chap. 1 |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |              |  |                  | Chap. 2 |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  | Data Warehouse Modeling I  |              |  |                  | Chap. 3 |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining   | Introduction |  |                  |         |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |              |  | Python-Analytics | Chap. 1 |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization              |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam                 |              |  |                  | Chap. 2 |  |  |
|           | W7                | 29.5. | Data Preparation                                    |                                   | 31.5. |   |              |  |                  |         |  |  |
|           | W8                | 5.6.  | Predictive Modeling I                               |                                   | 7.6.  | Predictive Modeling II (10:00 -12:00)   |              |  | BI-Project       | Start   |  |  |
|           | W9                | 12.6. | Fitting a Model I                                   |                                   | 14.6. | Python-Analytics-Online Exercise  |              |  |                  |         |  |  |
|           | W10               | 19.6. | Guest Lecture                                       |                                   | 21.6. | Fitting a Model II  |              |  |                  |         |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |              |  |                  |         |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I  |              |  |                  |         |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |              |  |                  |         |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |              |  |                  | End     |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin ~ 22.7. bis 3.8.<br>Klausur 2.Termin ~ 23.9. bis 5.10.                                       |              |  | Projektbericht   |         |  |  |

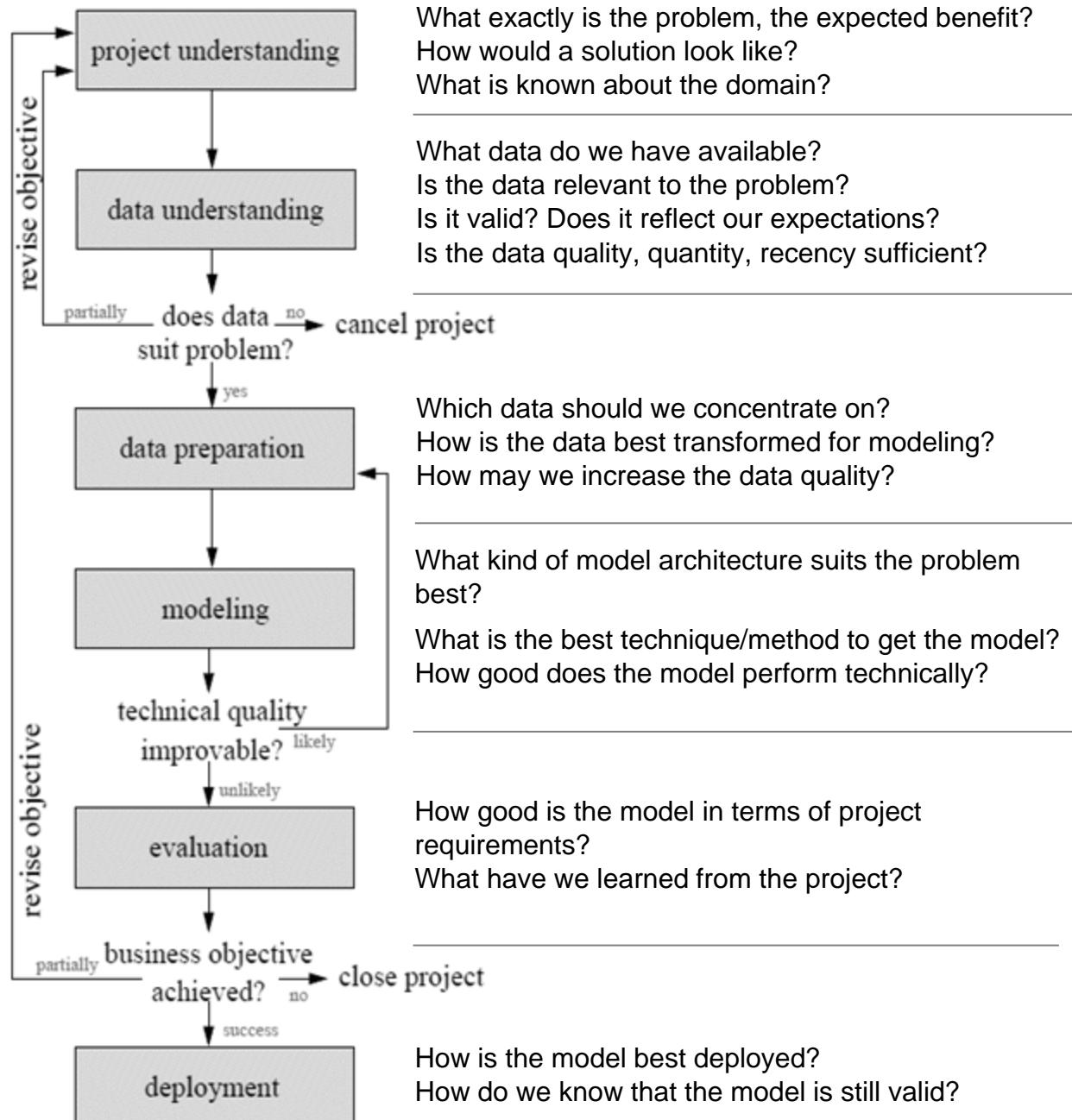
# CRISP-DM

# Cross Industry Standard Process for Data Mining

# Iteration as a rule

# Process of data exploration

## Implementation of the KDD Process



**Understand the problem** to be solved, its context, and the subsequent requirements for a solution.

Data are the available **raw materials** from which the solution will be built.  
Match business problem to one or several data mining tasks

- Data often need to be manipulated and converted into forms that yield better results
- Match **data** and **requirements** of DM techniques
- Select the relevant **variables**

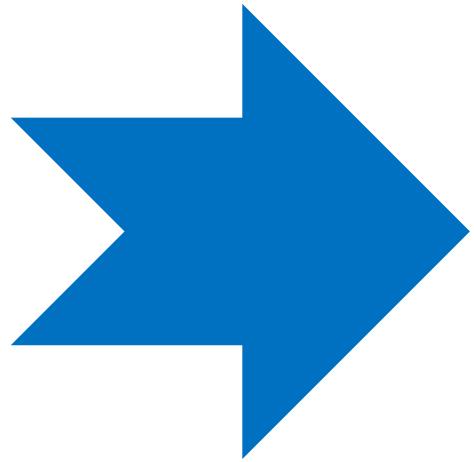
This is the primary place where **DM techniques** are applied to the data  
Select a Model, generate a test design, build the model and assess it.

- Assess the DM results **rigorously** (Gain confidence that results are valid and reliable)
  - Ensure that the model satisfies the original **business goals** (support decision making)

Ensure **comprehensibility** of the model to stakeholders

Evaluation framework needed (tested environments)

Models are put into **real use** in order to realize some return on investment.



## (1) Project Understanding

Assess the situation

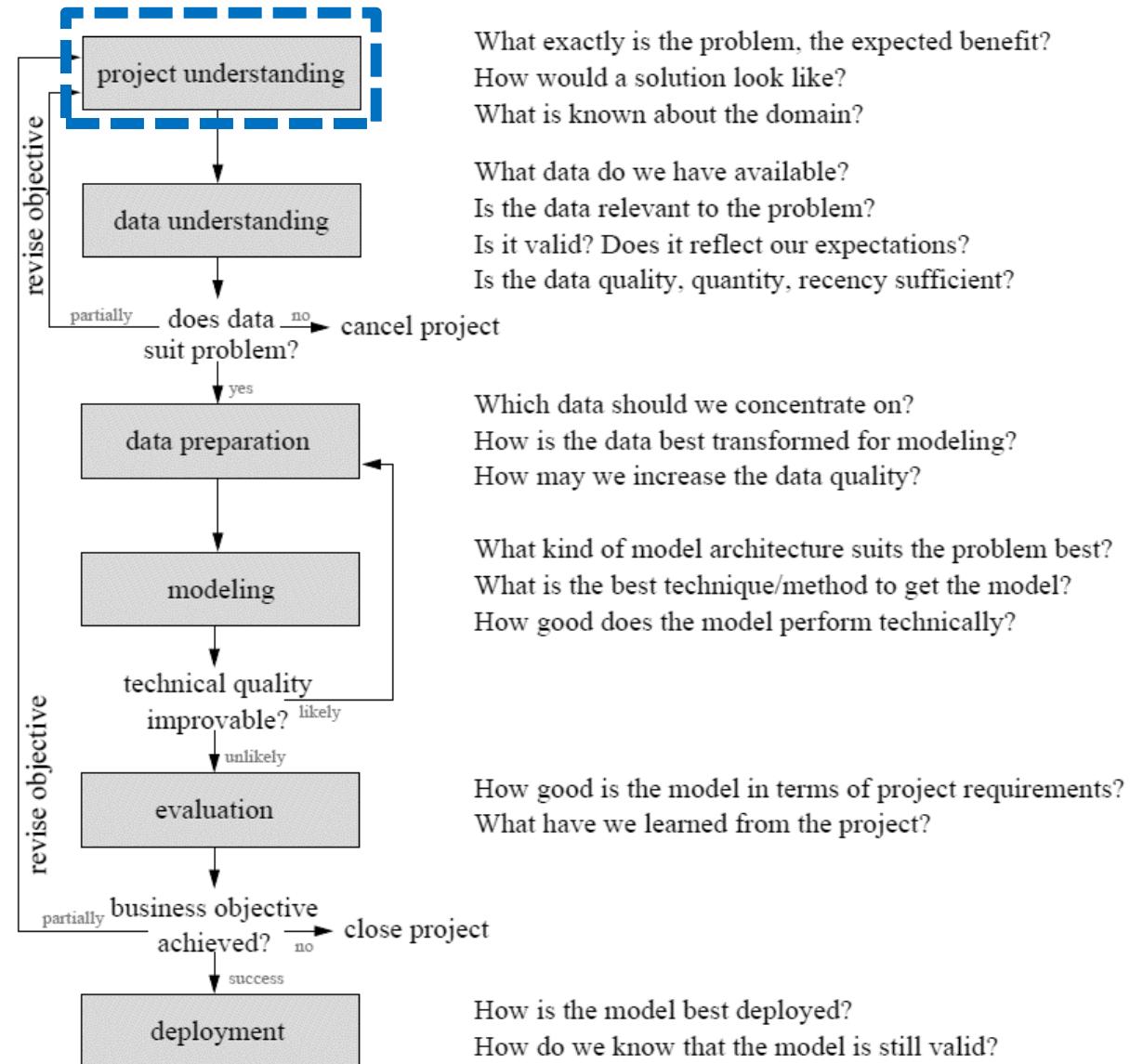
Determine analysis goals

## Cross Industry Standard Process for Data Mining

Iteration as a rule

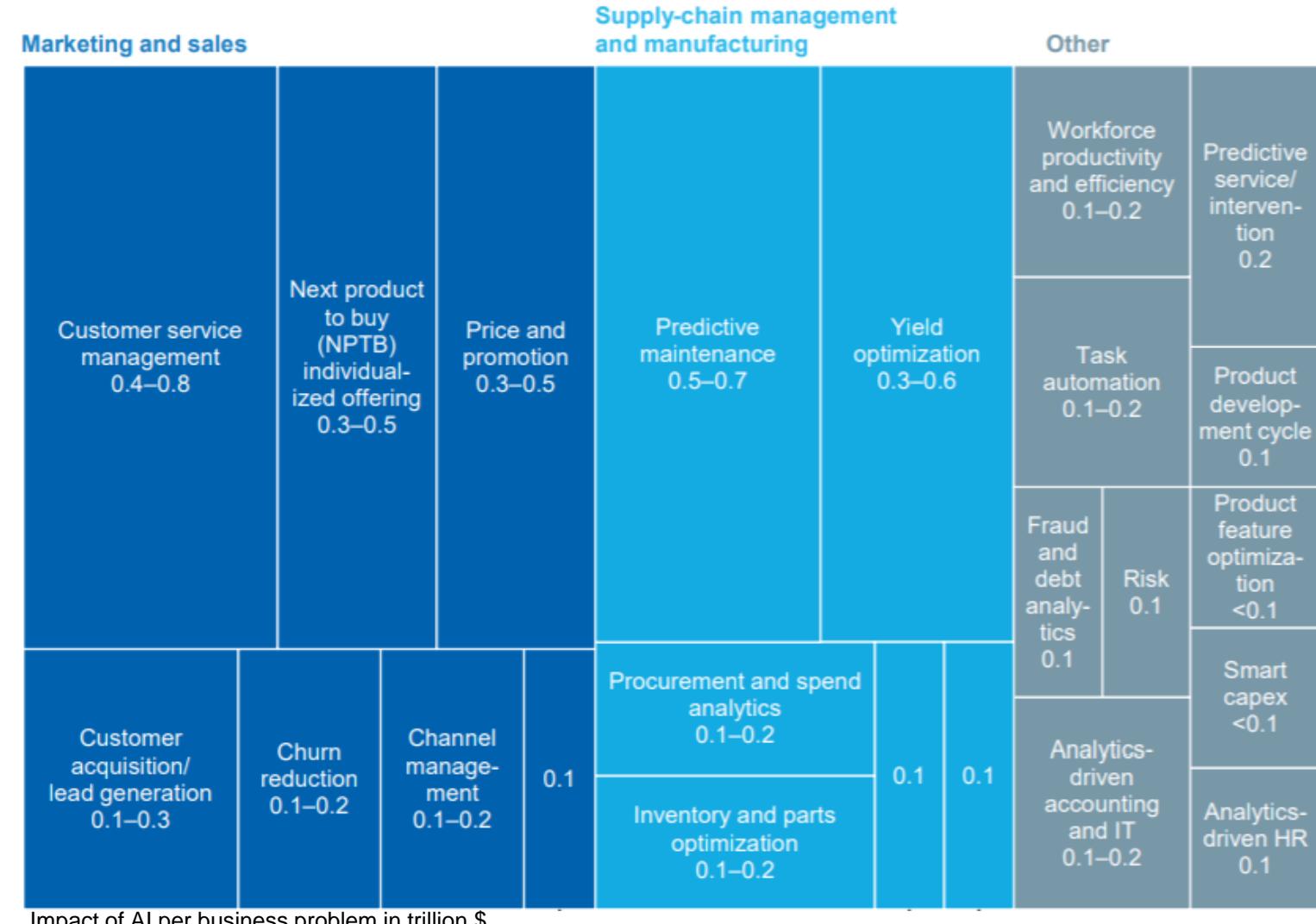
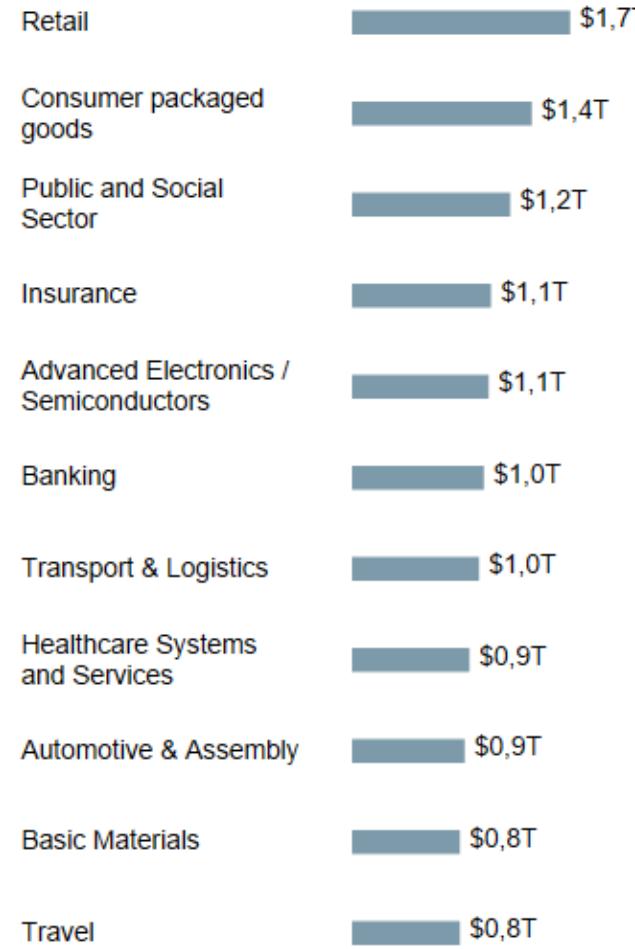
Process of data exploration

Implementation of the KDD Process



# Excursus: Business Problem Domains

AI and other analytics impact by industry, function and business problem.

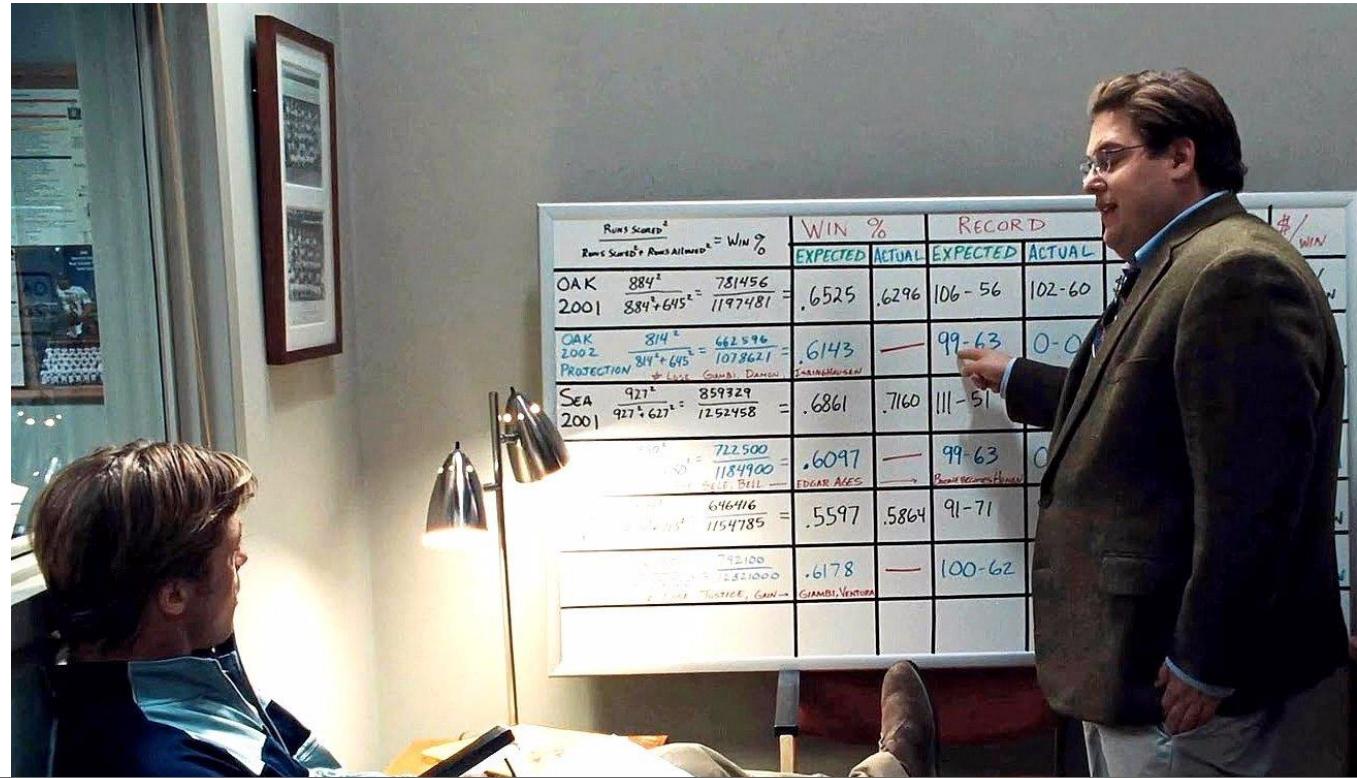


# Project understanding

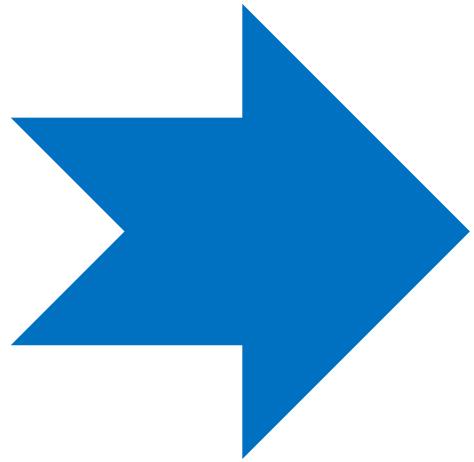
Assess **the main objective**, the potential benefit, as well as the constraints, assumptions and risks.

- Problem formulation
- Map the problem formulation to a data analysis task
- Understand the situation  
(available data, suitability of the data, ...)

- **Assess the situation**
- **Determine analysis goals**



Average time spent for project and data understanding within the CRISP-DM model: 20%  
Importance for success: 80%



## (1) Project Understanding

**Assess the situation**

Determine analysis goals

# Assess the situation

Project's success

Estimate chances of a successful data analysis project

Resources (data!), requirements and risks

*Does the given data satisfy the project's needs?*

**Typical requirements and constraints:**

## Model requirements

e.g., model has to be explanatory, because decisions must be justified clearly

vs. blackbox behavior

## Ethical, political, legal issues

e.g., variables such as gender, ethnicity must not be used

e.g., no racial profiling

(Example from Antidiskriminierungsstelle des Bundes)

## Technical constraints

e.g., applying the technical solution must not take more than  $n$  seconds

e.g., spam detection

# Assess the situation

Determine the project objective

The aim of the project should be clearly defined

Criteria to measure the success of the project  
should be agreed upon

**Aim/ Objective:** increase revenues (per campaign and or/per customer) in direct mailing campaigns by personalized offer and individual customer selection

**Deliverable:** software that automatically selects a specified number of customers from the database to whom the mailing shall be sent, runtime max. half a day

**Success criteria:** improve order rate by 5% or total revenues by 5%, measured within 4 weeks after mailing was sent



# Assess the situation

## Assumptions

### Representativeness:

Sample in the database must be representative for the whole population for which we intend to generalize.\*

### Informativeness:

To cover all aspects by the model, most of the influencing factors (e.g. identified in the *cognitive map*) should be represented by attributes in the database

### Good data quality:

The relevant data must be correct, complete, up-to-date and unambiguous thanks to the available documentation.

### Presence of external factors:

We may assume that the external world does not change constantly

\*Excuse: Not representative “84% want to abolish the time changeover”

[Link to European Commission](#), [Link to newspaper article](#)

# Assess the situation

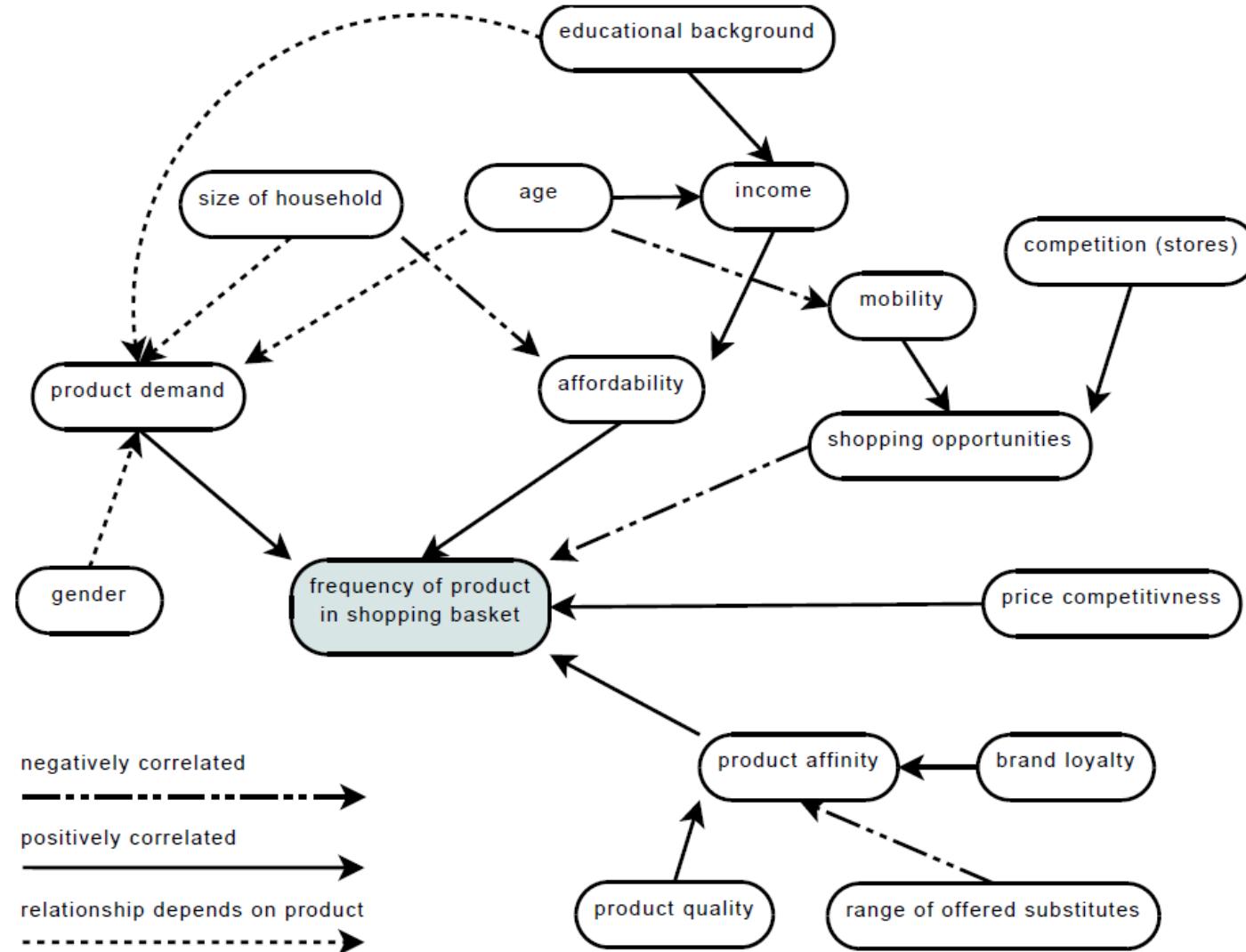
Cognitive map for domain knowledge

Perception of a reality

Directed graph of variables (**causal concept**) and relations (**causal connections**) in the decision problem domain and their strength (**causal value**).

The development of a cognitive map supports **domain understanding** and adjustment of expectations

- Include only direct dependencies to keep the map clear
- Choose labels of nodes carefully so they are easily interpretable
- Stick to the labels in project communication



# Exercise: Cognitive map

Domain knowledge?



Final grade in a  
mandatory course

(Operations Research)  
(Electronic Business)  
(Business Intelligence)  
(Service Engineering)

5 Min.

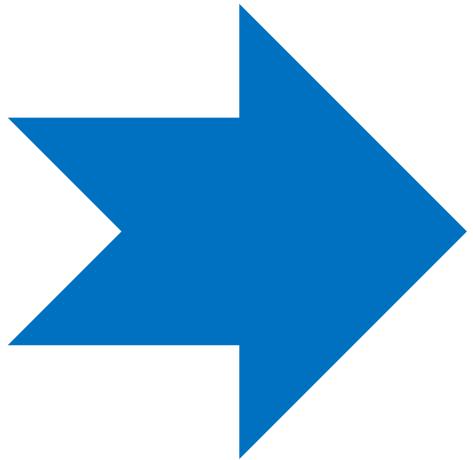
# Assess the situation

Risks: Domain experts and data analysis experts

| Problem source        | Project owner perspective  | Analyst perspective  |
|-----------------------|--|--|
| Communication         | Does not understand the <i>technical terms</i> of an analyst                                       | Does not understand the <i>terms of the domain</i>                                 |
| Lack of understanding | Is not sure <i>what</i> the analyst <i>could do</i> or achieve                                     | Finds it hard to understand <i>how to help</i> the project owner                   |
| Organization          | <i>Requirements</i> have to be adopted <i>in later stages</i> as problems with data become evident | Project owner is an <i>unpredictable group</i> (not so concerned with the project) |



-> Possible solutions?



## **(1) Project Understanding**

Assess the situation

**Determine analysis goals**

# Determine analysis goals

Problem decomposition

Determine DM tasks and decompose problem

- Classification, regression, cluster analysis, ...

Specify the requirements for the models that will be constructed by the DM tasks

There is no unique best method for a task

## Interpretability

If the goal of the analysis is a report that sketches possible explanations for a certain situation, the ultimate goal is to **understand** the delivered model.

For some **black box models** it is hard to comprehend how the final decision is made, and their model lacks interpretability. (i.e., deep learning)



# Determine analysis goals

Stability and Flexibility



## Reproducibility / stability

If the analysis is carried out more than once, we may achieve similar performance – but not necessarily similar models.

This does no harm if the model is used as a black box, but hinders a direct **comparison** of subsequent models to investigate their differences.

## Model flexibility / adequacy

A flexible model can adapt to more (complicated) situations than an inflexible model, which typically makes more assumptions about the real world and requires less parameters.

If the problem domain is complex, the model learned from data must also be complex to be successful. With flexible models the risk of **overfitting** increases.



Ref.

Image: [Creative Tools](#) (2015) | Flickr (cc-by 2.0)

# Determine analysis goals

## Runtime

If restrictive runtime requirements are given (either for building or applying the model), this may exclude some computationally expensive approaches.

## Interestingness and use of expert knowledge

The more an expert already knows, the more challenging it is to **surprise** her with new findings. Some techniques are known for their large number of findings, many of them redundant and thus uninteresting.

So if there is a possibility of including any kind of previous knowledge, this may ease the search for the best model considerably and may **prevent us from re-discovering** too many well-known artefacts.



## Fragen?

- ✓ The data mining process – CRISP-DM
- ✓ Business / Project understanding

*Starting in week W8, you will continue to deepen this content by working on your project*

# Recommended reading (for this week)

Berthold et al.     Guide to Intelligent Data Analysis  
                        Chapter 3, 4

Provost, F.,       Data Science for Business  
Fawcett, T.        Chapter 2

Pyle, D.           Business Modeling and Data Mining. Morgan Kaufmann, San Mateo (2003)



# Business Intelligence

## 06 Data Understanding & Data Visualization

Prof. Dr. Bastian Amberg  
(summer term 2024)  
22.5.2024

# Schedule

|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study |                          |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|------------|--------------------------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |            |                          |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |            |                          |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  | Data Warehouse Modeling I  |            |                          |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining Introduction  |            |                          |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |            | Python-Analytics Chap. 1 |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization              |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam                 |            | Chap. 2                  |  |  |
|           | W7                | 29.5. | Data Preparation                                    |                                   | 31.5. |   |            |                          |  |  |
|           | W8                | 5.6.  | Predictive Modeling I                               |                                   | 7.6.  | Predictive Modeling II (10:00 -12:00)   |            | BI-Project Start         |  |  |
|           | W9                | 12.6. | Fitting a Model I                                   |                                   | 14.6. | Python-Analytics-Online Exercise  |            |                          |  |  |
|           | W10               | 19.6. | Guest Lecture                                       |                                   | 21.6. | Fitting a Model II  |            |                          |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |            |                          |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I  |            |                          |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |            |                          |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |            | End                      |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin, 31.7.'24<br>Klausur 2.Termin, 2.10.'24  |            | Projektbericht           |  |  |

# Note on Bonus tasks

Vom 24.5.'24 bis spätestens 7.6.'24

Bitte jeweils die genaue Aufgabenstellung inklusive Abgabeformalitäten beachten!

Diese ist ab Freitag, 24.5., 10 Uhr in Blackboard verfügbar.

## 1. Co-Create your exam

Einzelleistung (bzw. Leistung des gesamten Kurses)

- Zwei Aussagen im Kontext der Veranstaltungen 01 bis 04 formulieren (eine wahr, eine falsch)
- Freitextaufgabe zu Veranstaltungen 01 bis 04 formulieren, die mit einem Wort (keine Aussage über wahr oder falsch!) beantwortet werden kann
- Falls bis zum 7.6.'24  $\geq 84$  (= 3 Fragen x 28 gemeldete Teilnehmer:innen) unterschiedliche, sinnvolle Fragen/Aussagen zusammen kommen, ist eine Teilmenge davon Bestandteil der Klausur

## 2. Earn bonus points for the exam

Gruppenarbeit (min. 2 bis max. 3 Personen pro Gruppe)

- Rechercheaufgabe zu Supervised, Unsupervised und Reinforcement Learning
- Grafische Darstellung auf einer Folie inklusive Beschreibung und kritischer Würdigung
- Dabei Einsatz von generativer KI möglich (z.B. ChatGPT, Bing AI),  
sofern die verwendeten Anfragen/Befehle nachvollziehbar dokumentiert werden
- Maximal drei Bonuspunkte, die auf die in der Klausur erreichte Punktzahl addiert werden  
(sofern die Klausur bestanden wurde)

- ✓ From business problems to data mining tasks
- ✓ Supervised vs. unsupervised methods vs. reinforcement learning

- [Demo Reinforcement Learning](#) (from the last lesson)
- [A.I. Learns to Drive From Scratch in Trackmania](#) (very well explained youtube video)
- Optimization tasks in stochastic, dynamic environments (project example)



| Learning Approach | Learn from....<br>(input? - output?) | Learn for....<br>(target?) | Example? |
|-------------------|--------------------------------------|----------------------------|----------|
|-------------------|--------------------------------------|----------------------------|----------|

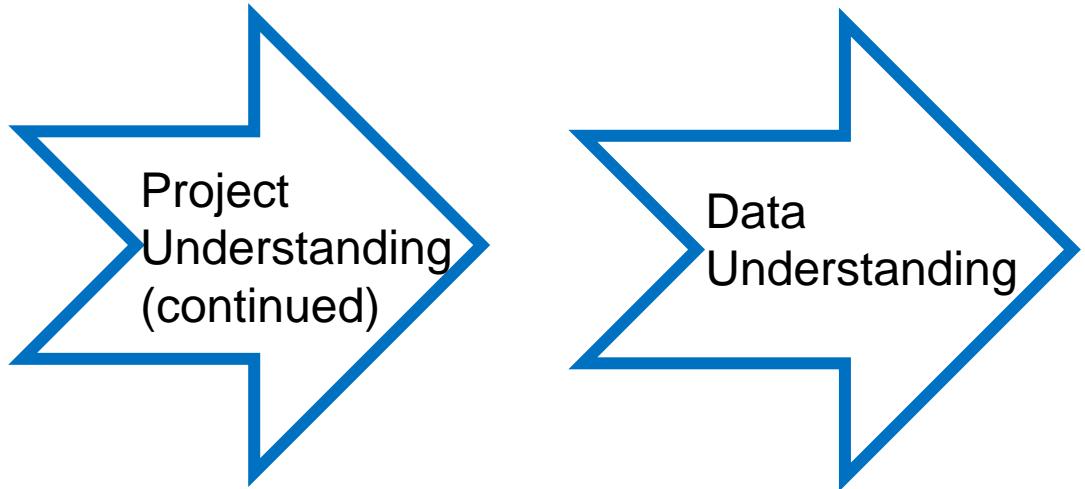
Kahoot-Fragen zu den Inhalten  
[www.kahoot.it](http://www.kahoot.it)  
(über Smartphone oder Laptop)  
PIN folgt

- ✓ The data mining process – CRISP-DM
  - Business / Project understanding

# Today's Agenda



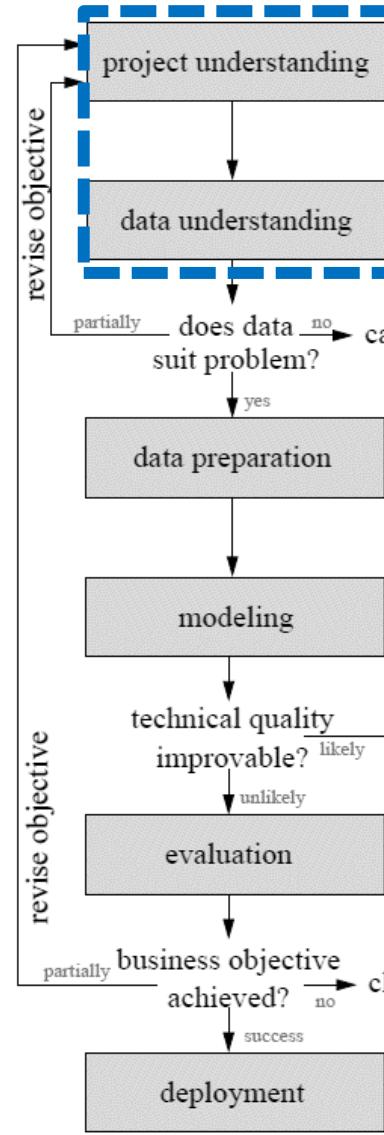
## First Part



See slides 5b  
from last week

## Second Part

*This set of slides*



What exactly is the problem, the expected benefit?

How would a solution look like?

What is known about the domain?

What data do we have available?

Is the data relevant to the problem?

Is it valid? Does it reflect our expectations?

Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?

How is the data best transformed for modeling?

How may we increase the data quality?

What kind of model architecture suits the problem best?

What is the best technique/method to get the model?

How good does the model perform technically?

How good is the model in terms of project requirements?

What have we learned from the project?

How is the model best deployed?

How do we know that the model is still valid?

# First Part

(see slides 10 to 18 from slide set 5b)

## ✓ The data mining process – CRISP-DM

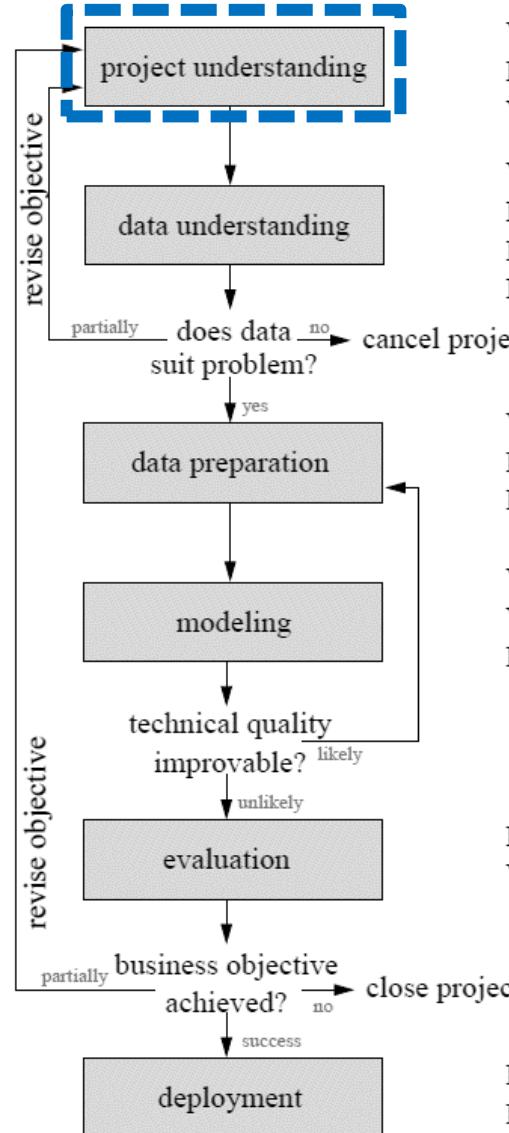
## ✓ Business / Project understanding



Image: "[Moneyball](#)"/Columbia Pictures, Video

- Assess the situation
- Determine analysis goals

Ref.



What exactly is the problem, the expected benefit?

How would a solution look like?

What is known about the domain?

What data do we have available?

Is the data relevant to the problem?

Is it valid? Does it reflect our expectations?

Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?

How is the data best transformed for modeling?

How may we increase the data quality?

What kind of model architecture suits the problem best?

What is the best technique/method to get the model?

How good does the model perform technically?

How good is the model in terms of project requirements?

What have we learned from the project?

How is the model best deployed?

How do we know that the model is still valid?

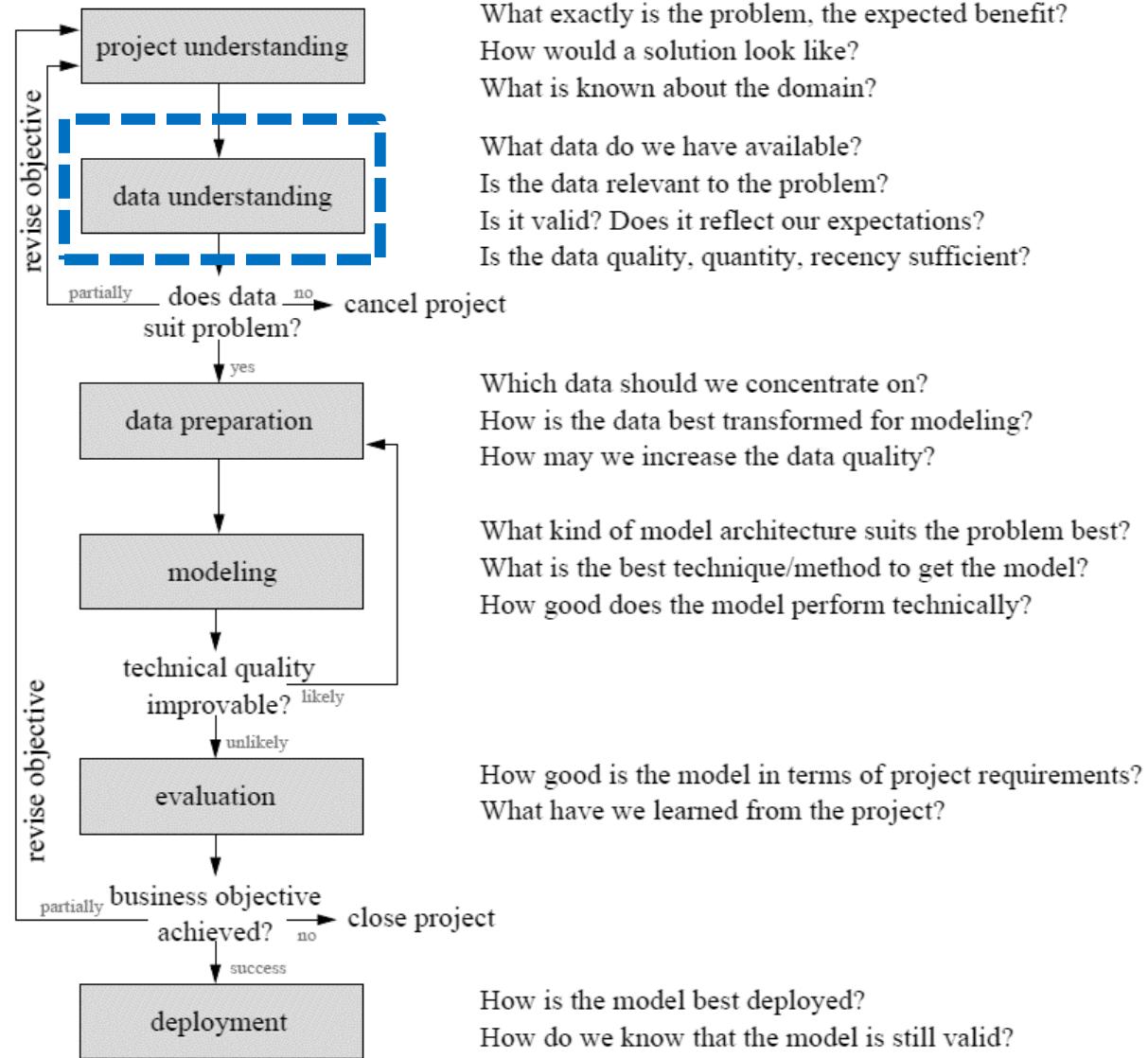
## Second Part

# Cross Industry Standard Process for Data Mining

# Iteration as a rule

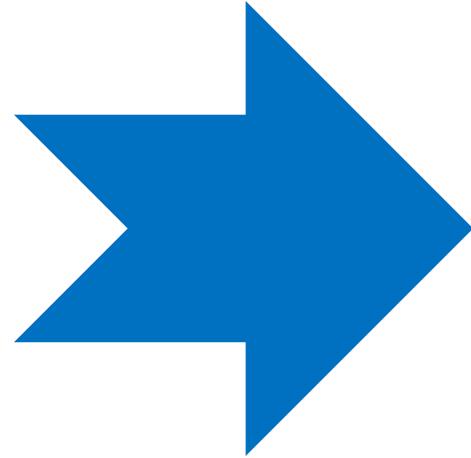
# Process of data exploration

# Implementation of the KDD Process



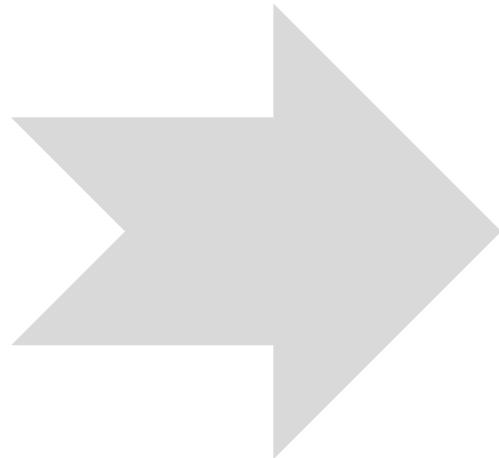
# Agenda for Data Understanding

## (1) Data Understanding

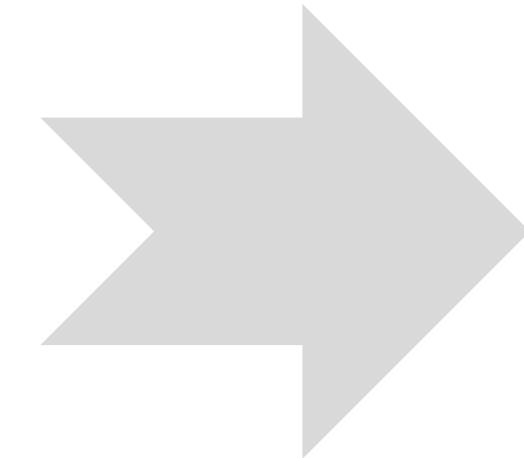


Attribute Understanding  
Data Quality

## (2) Data Visualization ( = Data Understanding – Part 2)



Low-dimensional relationships  
Univariate Analysis  
Bivariate Analysis



Higher-dimensional relationships  
Principal Component Analysis  
Parallel Coordinates

# Goals of data understanding

Gain **general insights** about the data (independent of the project goal)

Check the assumptions made during the project understanding phase (representativeness, informativeness, data quality, presence/absence of external factors, dependencies, ...)

Check the specified **domain knowledge**

Check suitability of the data for the project goals

**Never trust any data** as long as you have not carried out some simple plausibility checks!



# Attribute understanding

## And types of attributes

We often assume that the data set is provided in form of a simple table

The rows of the table are called **instances, records or data objects**

The columns of the table are called **attributes, features or variables**

**Categorical (nominal)**: finite domain. The values of a categorical attribute are often called classes or categories

Examples: [female, male, diverse], [ordered, received]

**Ordinal**: finite domain with a linear ordering on the domain.

Example: [B.Sc., M.Sc., Ph.D.], [Dawn, Noon, Afternoon, Evening, Night]

**Numerical**: values are numbers

**Discrete**: categorical attribute or numerical attribute whose domain is a subset of an integer number

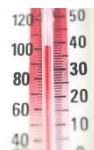
**Continuous**: numerical attribute with values in the real numbers or in an interval (float)

|          | Attribute 1 | ... | Attribute M |
|----------|-------------|-----|-------------|
| Record 1 |             |     |             |
| ...      |             |     |             |
| Record n |             |     |             |

Scales for numerical attributes

**Interval scale**: the definition of the value 0 is arbitrary. Ratios are meaningless.

Examples: date (Unix standard time: time point zero is in the year 1970), temperature (°C or °F)



**Ratio scale**: 0 has a canonical meaning

(none of the measured quality exists)

Ratios make sense.

Examples: distance, duration

**Absolute scale**: domain with a unique measurement unit.

Examples: any kind of counting process (number of children, number of visits to the doctor)

# Specific problems of categorical attributes

Levels of granularity; Dynamic domains

Different **levels of granularity** might be definable.

Examples:

product categories/types:

- └ General category: drinks, food, clothes, ...
  - └ More refined categories for drinks: water, beer, wine, ...
    - └ Further refinement for water based on the producer.
      - └ Further refinement of the of each producer based on the bottle size (0.33 l, 0.5 l, 1 l, 1.5 l)

⇒ The most refined level provides the **most detailed information**, but will not help to discover general associations like “Wine and cheese are often bought together”

**Dynamic domains**: The possible values of the domain might change over time.

Example: certain product categories or products might not be sold anymore. New product categories or products are introduced.

⇒ The analysis of such data will be **biased** to values (example: products) that have been in the domain for a long time.



# Data quality

Syntactic accuracy vs. Semantic accuracy

**Syntactic accuracy** is violated if an entry does not belong to the domain of the attribute

The entry *female* for the categorical attribute *gender*

Text entries for numerical attributes

Values out of range for numerical attributes (negative numbers for weight, distance, counting process, ...)

Syntactic accuracy can be checked quite easily

**Semantic accuracy** is violated if an entry is not correct although it belongs to the domain of the attribute

| Name           | Gender |
|----------------|--------|
| John Smith     | Female |
| Lisa McIntosh  | Female |
| Rick Rickerton | Male   |
| Jane Smith     | Male   |
| John Doe       | Male   |

Semantic accuracy is more difficult to check than syntactic accuracy.

Can only be investigated based on “business rules” and plausibility checks.

# Data quality

Completeness, Unbalanced data and timeliness

complete **attribute values**:

fraction of “null” entries for an attribute.

Note that missing values are not always marked explicitly as missing, for instance in the case of *default entries*!

e.g., Time: 12 o'clock  
Birthday: 01.01.xxxx

complete **records**: complete records might be missing.

*Example 1:* Three years ago, a new system was introduced and not all customer data were transferred to the new system.

*Example 2:* The data set is biased, e.g., a bank might have rejected customers with no income, but they did not protocol it.

**Unbalanced data**:

the data set might be biased extremely to one type of records

Production line for goods including quality control → defective goods will be a very small fraction of all records!

e.g., 99.9% (+)  
0.01% (-)

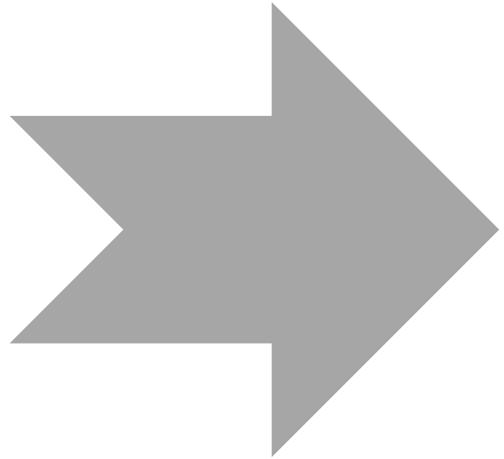
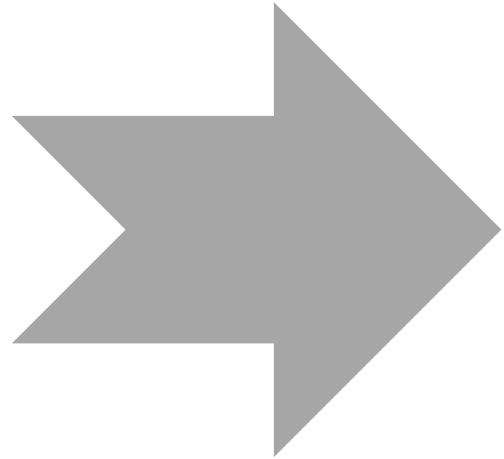
**Timeliness**:

is the available data up to date to be considered to be representative?

| Observations on the weather<br>Philadelphia 1776 |           |             |     |             |             |  |
|--|-----------|-------------|-----|-------------|-------------|--|
| July   | hour.     | Humid.      | day | A. m.       | P. m.       |  |
| 1.   | 9-0 A.M.  | 61 <i>1</i> | 9   | 5-30 A.M.   | 75          |  |
|  | 7- P.M.   | 62          |     | 9           | 77 <i>1</i> |  |
| 2.   | 6- A.M.   | 78          |     | 6-30 P.M.   | 91 <i>2</i> |  |
|  | 9- A.M.   | 78          |     | 9- 45       | 78          |  |
|  | 9- P.M.   | 74          |     | 10- 8. A.M. | 75          |  |
| 3.   | 5-30 A.M. | 71 <i>1</i> |     | 9- 15       | 76 <i>2</i> |  |
|  | 1-30 P.M. | 76          |     | 2- 6. P.M.  | 80          |  |
|  | 8- 10     | 74          |     | 4- 45       | 82          |  |



## Data Visualization (= Data Understanding – Part 2)



### Low-dimensional relationships

Univariate Analysis  
Bivariate Analysis

### Higher-dimensional relationships

Principal Component Analysis  
Parallel Coordinates

# First: Python libraries for data visualization

used libraries...  
(among others)

**pandas**

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$


[home](#) // [about](#) // [get pandas](#) // [documentation](#) // [community](#) // [talks](#) // [donate](#)

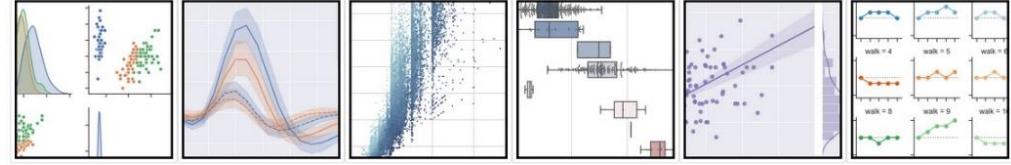
Python Data Analysis Library

*pandas* is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the [Python](#) programming language.

*pandas* is a [NumFOCUS](#) sponsored project. This will help ensure the success of development of *pandas* as a world-class open-source [to donate](#) to the project.

<https://pandas.pydata.org/>

seaborn: statistical data visualization



VERSIONS

Version 1.4.2,  
Release date Apr 02, 2022  
[download](#) // [docs](#) // [pdf](#)

Contents

- Introduction
- Release notes
- Installing
- Example gallery

Features

- Relational: API | Tutorial
- Categorical: API | Tutorial
- Distributions: API | Tutorial
- Regressions: API | Tutorial

<https://seaborn.pydata.org/>

How to use?  
>> See sample code on the slides.

# First: Getting started with a Python-IDE

# Anaconda Spyder

```
 Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
[untitled] inData_data_understanding.py

19# Create DataFrame using Pandas and set column names
20iris = pd.read_csv('~/00 - data/irisData.csv', names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])
21iris = sns.load_dataset("iris")
22iris_list = iris.values.tolist()
23iris_dict = iris.to_dict()
24
25# Show descriptive statistics on dimensional distributions
26print(iris.describe())
27
28# Describe relationships among variables in scatter plot
29# hue variable used for color mapping
30sns.pairplot(iris, hue="species", palette="husl")
31plt.show()
32plt.clf()
33
34
35"""
36Principal Component Analysis
37
38"""
39from sklearn.decomposition import PCA
40from sklearn.preprocessing import scale
41
42#Select only metric data
43raw_iris = iris[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']]
44
45#Center data to mean
46norm_iris = scale(raw_iris)
47
48#create pca with 2-dimensions
49pca = PCA(n_components=2)
50
51# pca data
52pca_iris = pca.fit_transform(norm_iris)
53print("Show PCA results!")
54print(norm_iris.shape)
55print(pca_iris.shape)
56
57
58vis_iris = pd.DataFrame(pca_iris, columns=['pc1', 'pc2'])
59vis_iris['species'] = iris['species']
60g = sns.FacetGrid(vis_iris, hue='species', size=5)
61g.map(plt.scatter, 'pc1', 'pc2')
62g.set_xlabels('principal component 1')
63g.set_ylabels('principal component 2')
64
65plt.show()
66plt.clf()
67
68"""
69Parallel Coordinates
70
71"""
72
73#from pandas.tools.plotting import parallel_coordinates
74#
75#parallel_coordinates(iris, 'species')
76# plt.show()
77# plt.clf()
78#
79#
80"""
81#Correlation Analysis
82#2017-05-28
83"""
84
```

# Code

# Variable Explorer

## iPython Console (input/output)

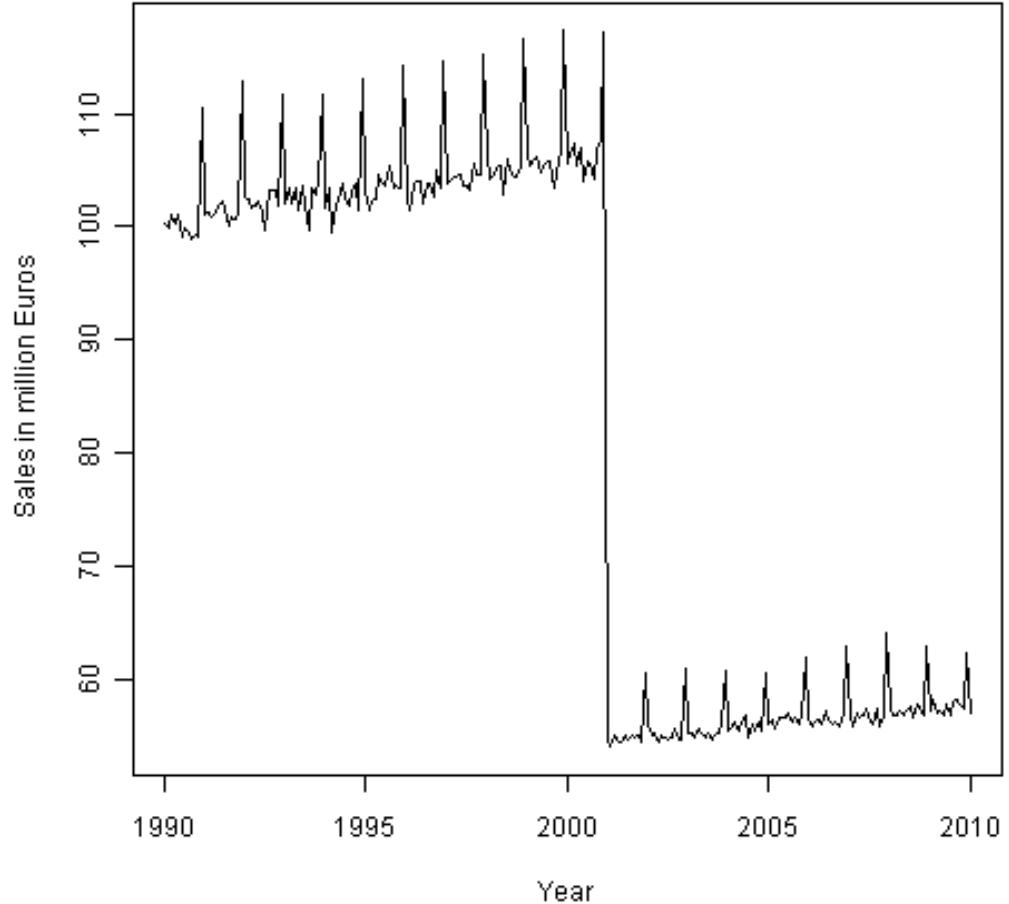
Ref.

# Data visualization



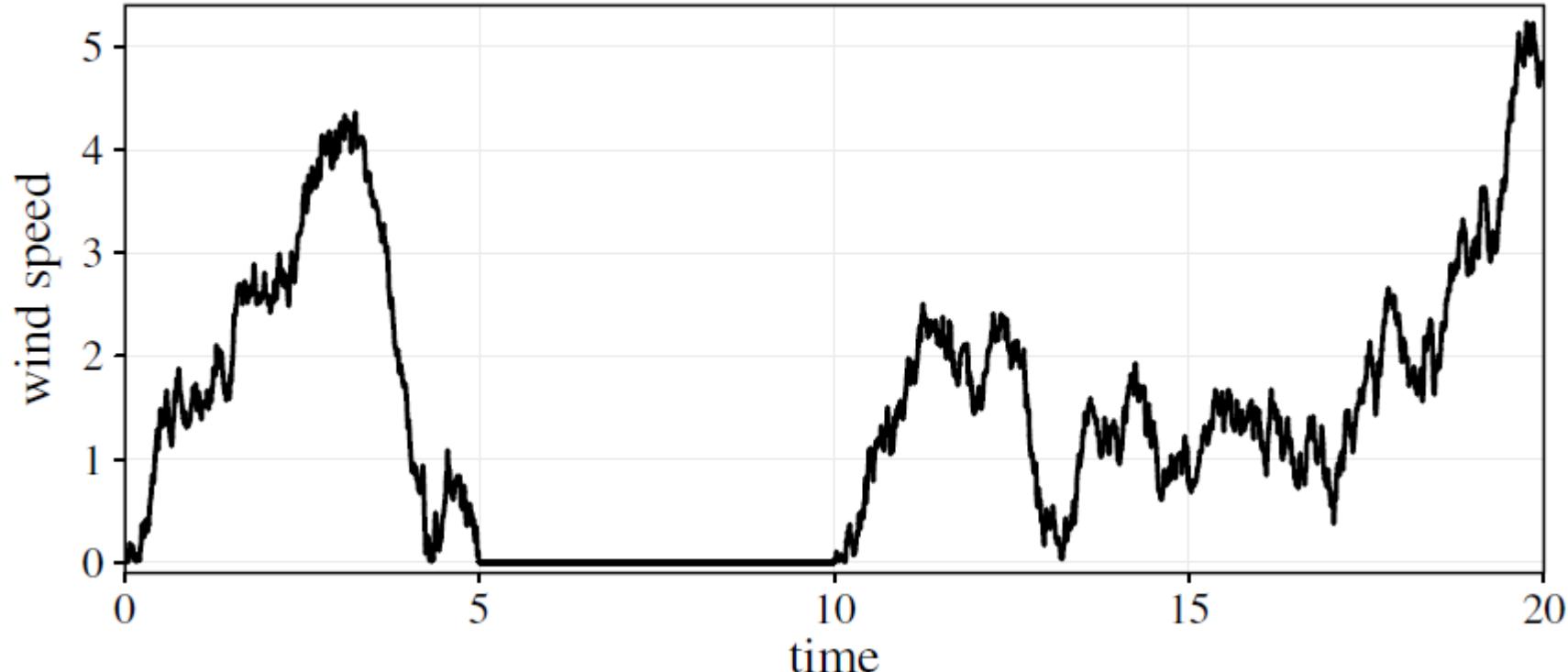
*“There is no excuse for failing to plot and look”.*

Tukey (1977)



Ref. Tukey (1977)

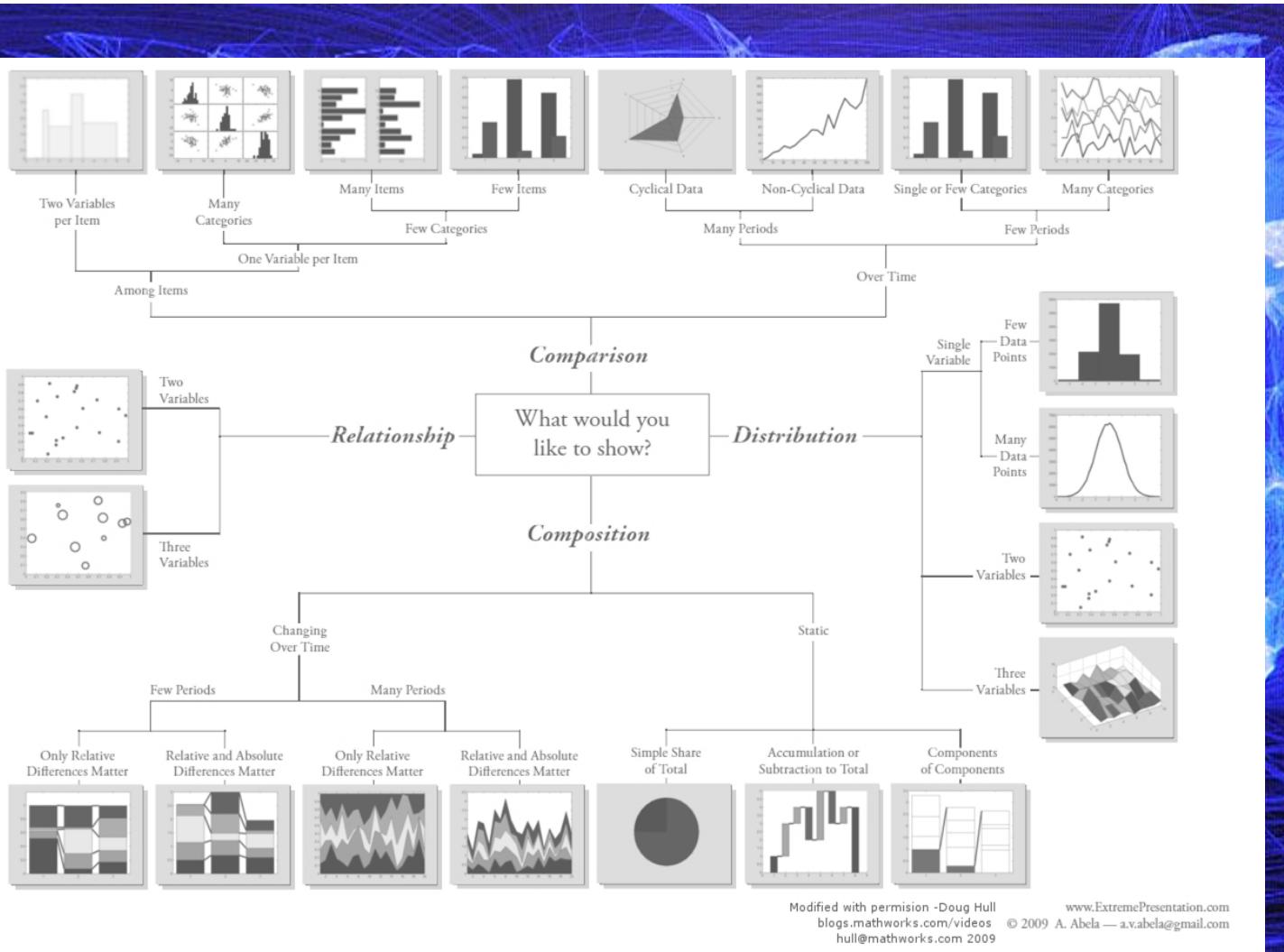
# Hidden missing values



Ref.

# Visualization Types

Selecting the 'right' visualization



infodook

Ref. <http://extremepresentation.typepad.com>, Bertin (1983), Woolman (2002), Cairo (2012),

Graphic representation constitutes one of the **basic sign-systems** conceived by the human mind for the purposes of **storing, understanding, and communicating** essential **information**. As a "language" for the eye, graphics benefits from the ubiquitous properties of visual perception. As a monosemic system, it forms the rational part of the world of images.

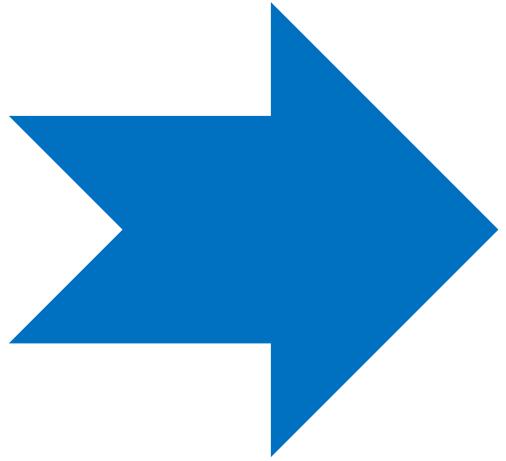
(Bertin, 1983)

"They must **make sense to the user** and require a **visual language system** that uses colour, shape, line, hierarchy and composition to communicate clearly and appropriately, much like the alphabetic and character-based languages used worldwide between humans."

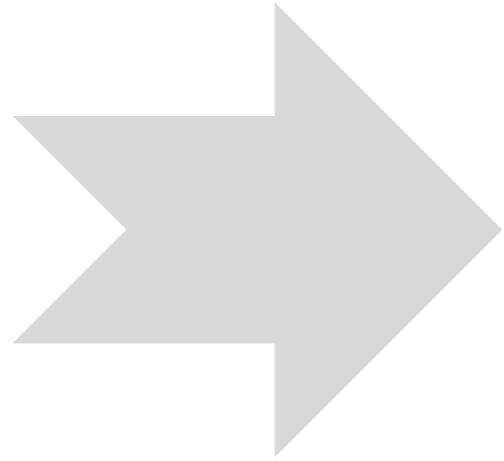
(Woolman, 2002)

A good infographic should be **functional** as a hammer, **multilayered** as an onion, and **beautiful** and **true** as an equation (or as a scientific theory). An information graphic must be precise, accurate, efficient, and deep before the designer can apply his or her own visual style or typographical and color preferences to the display.

(Alberto Cairo, 2012)



**Low-dimensional  
relationships**  
Univariate Analysis  
Bivariate Analysis

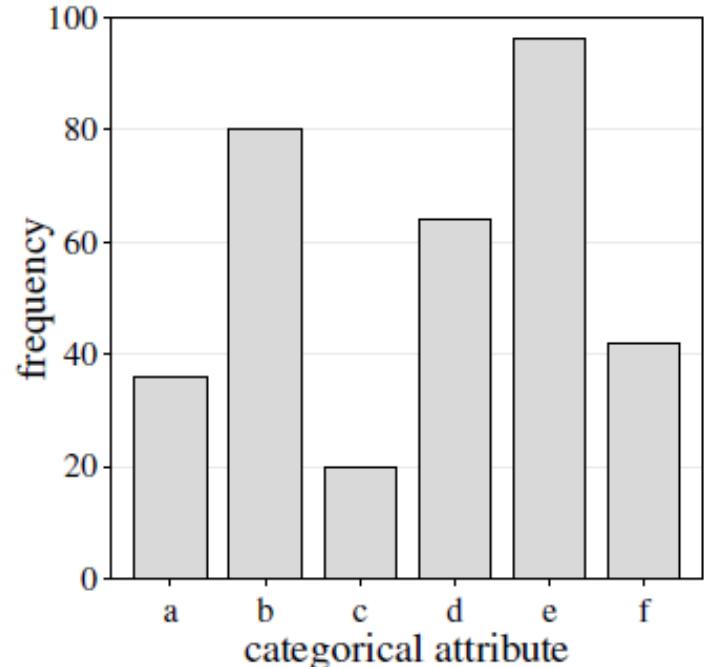


**Higher-dimensional  
relationships**  
Principal Component Analysis  
Parallel Coordinates

# Common visualizations

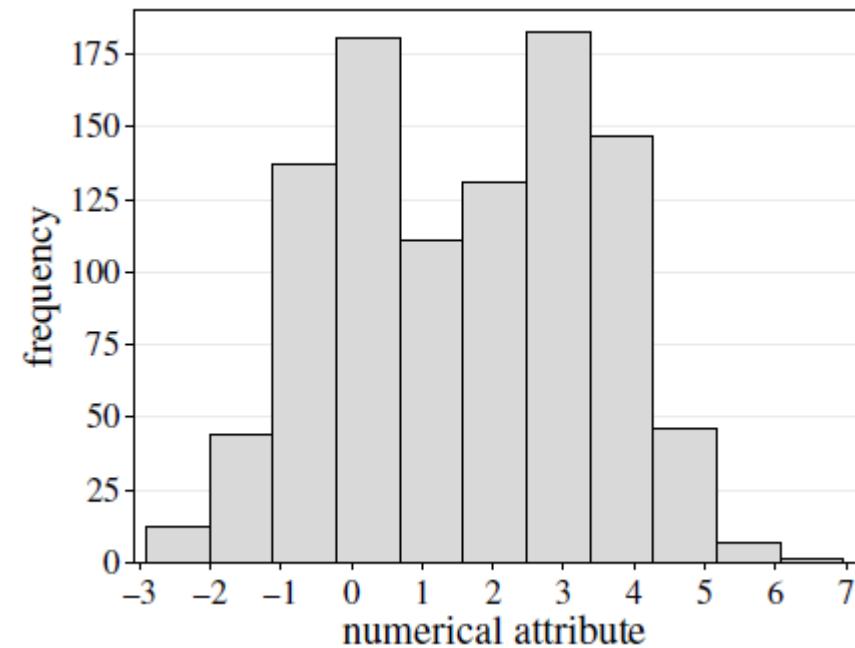
## Bar charts and Histograms

A **bar chart** is a simple way to depict the frequencies of the values of a categorical attribute.



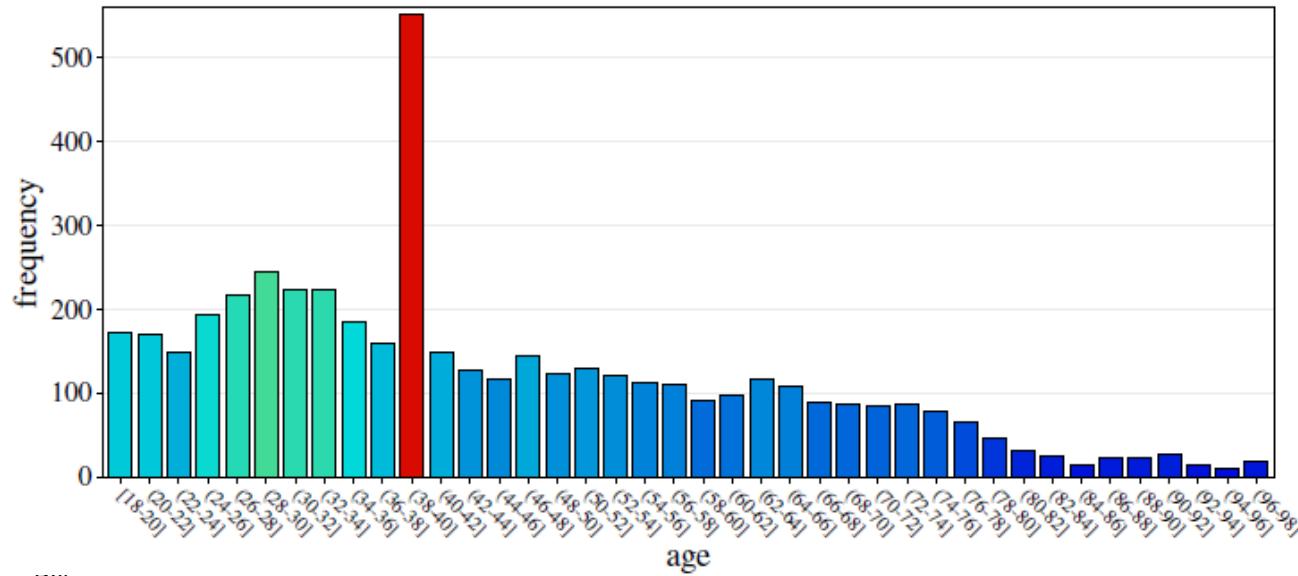
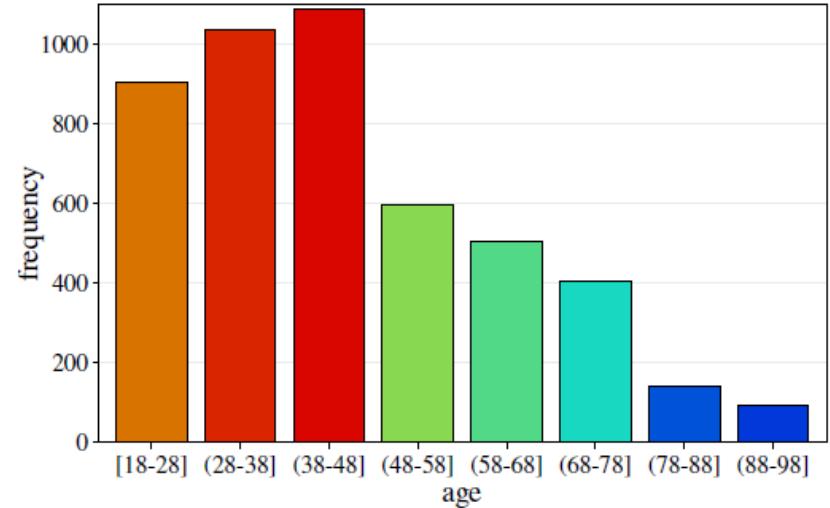
A **histogram** shows the frequency distribution for a numerical attribute.

The range of numerical attribute is discretized into a fixed number of intervals ("bins"), usually of equal length. For each interval, the (absolute) frequency of values falling into it is indicated by the height of a bar.

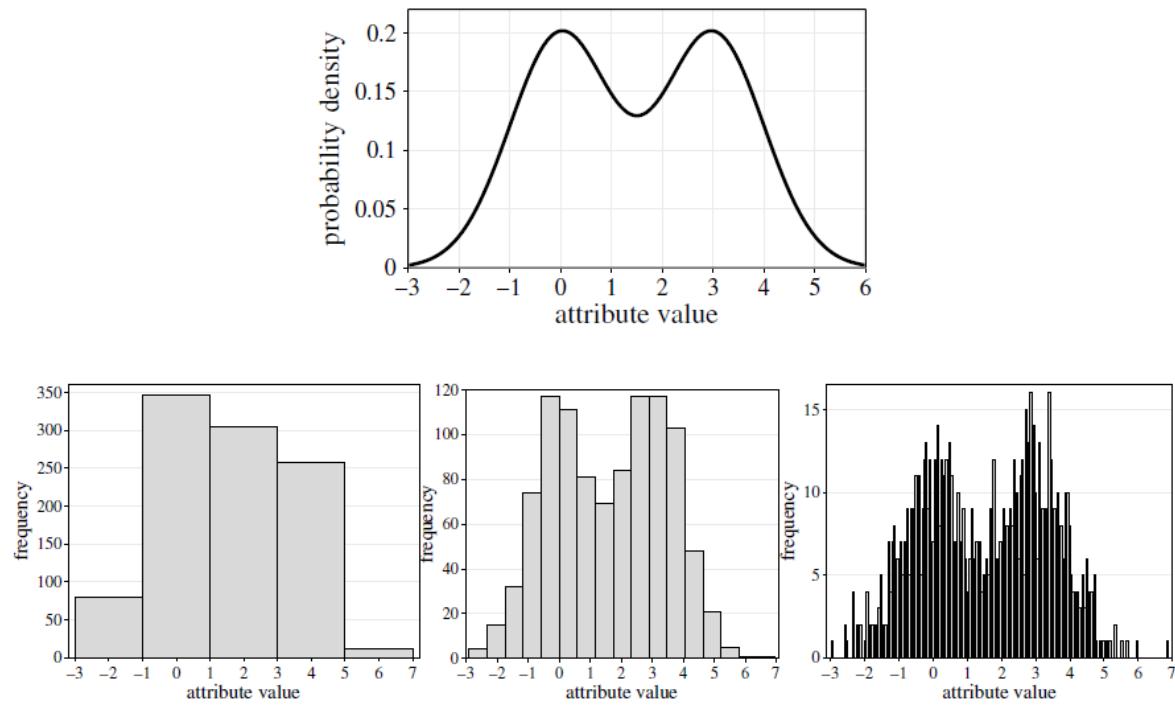


# Common visualizations

Histograms: The number of bins is very important.



Three histograms with 5, 17 and 200 bins for a sample from the same bimodal distribution. Sample size is  $n = 1000$ .



# Example data set

## Iris data

Collected by E. Anderson in 1935

Contains measurements of four real-valued variables of 150  
**iris flowers** of types Iris Setosa, Iris Versicolor, Iris Virginica

- Sepal length [Kelchblatt]
- Sepal widths
- Petal lengths [Blütenblatt]
- Petal widths

The fifth attribute is the name of the flower type

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species         |
|--------------|-------------|--------------|-------------|-----------------|
| 5.1          | 3.5         | 1.4          | 0.2         | Iris-setosa     |
| ...          |             |              |             |                 |
| ...          |             |              |             |                 |
| 5.0          | 3.3         | 1.4          | 0.2         | Iris-setosa     |
| 7.0          | 3.2         | 4.7          | 1.4         | Iris-versicolor |
| ...          |             |              |             |                 |
| ...          |             |              |             |                 |
| 5.1          | 2.5         | 3.0          | 1.1         | Iris-versicolor |
| 5.7          | 2.8         | 4.1          | 1.3         | Iris-virginica  |
| ...          |             |              |             |                 |
| ...          |             |              |             |                 |
| 5.9          | 3.0         | 5.1          | 1.8         | Iris-virginica  |

Ref.



Iris Setosa



Iris Versicolor



Iris Virginica

```
import pandas as pd
# Create DataFrame using Pandas and set Column names
iris = pd.read_csv('irisData.csv', names=['sepal_length','sepal_width','petal_length','petal_width','species'])
# Show descriptive statistics on dimensional distributions
print(iris.describe())
# Show histogram
iris.hist(column='sepal_length', bins = (4.0,4.5,5.0,5.5,6.0,6.5,7.0,7.5,8))
```



# Iris data set: boxplots

**Boxplots** are a very compact way to visualize and summarize main characteristics of a sample from a numerical attribute

Line in the middle = median

Box = interquartile range

Whiskers =  $1.5 \times$  interquartile range

```
import pandas as pd
import seaborn as sns

iris = pd.read_csv('irisData.csv', names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])
sns.boxplot(x="species", y="sepal_length", data=iris, notch=True)
```



## Reminder:

### **Median:**

the value in the middle (for the values given in increasing order)

### **q%-quantile (0<q<100):**

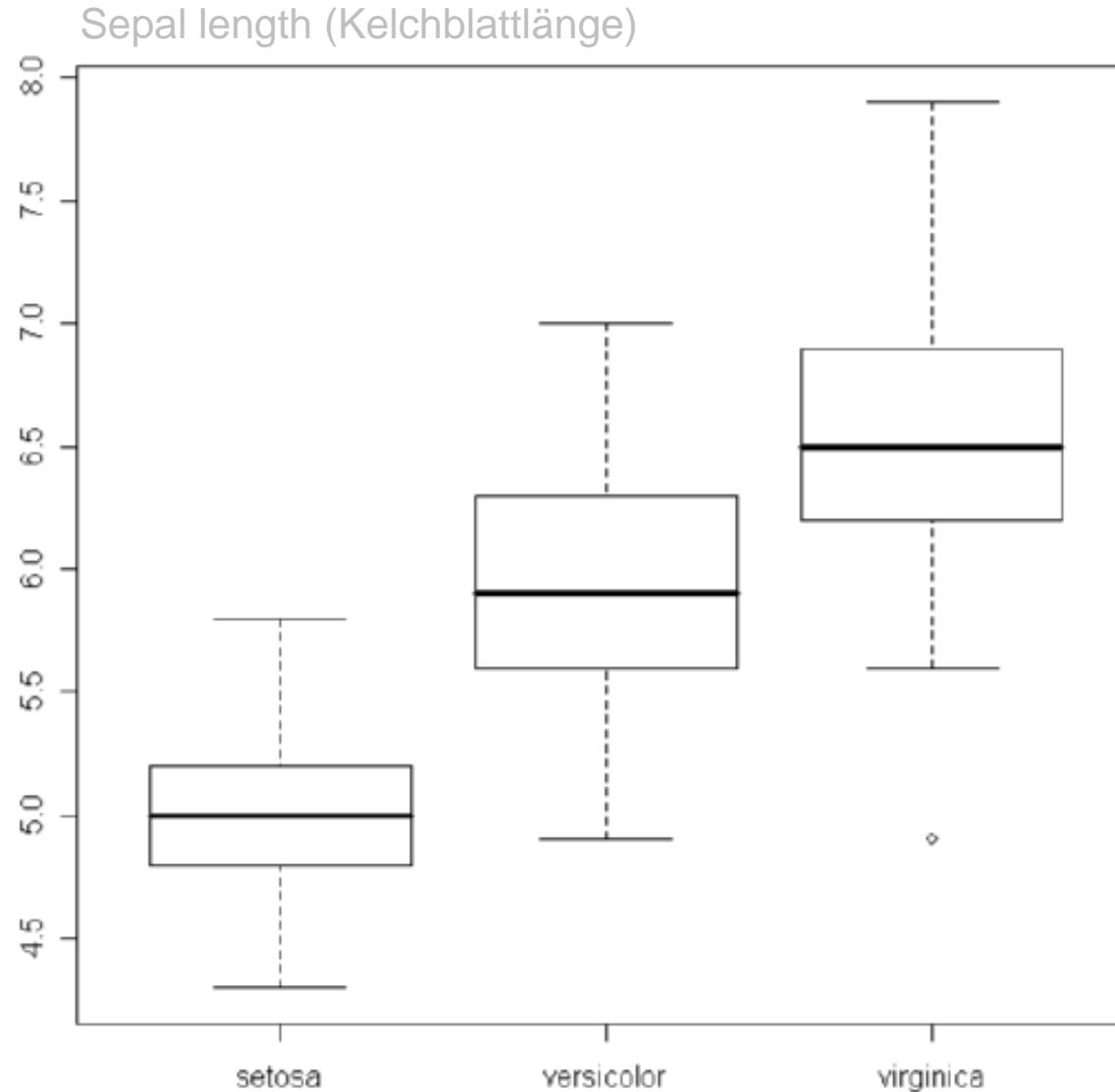
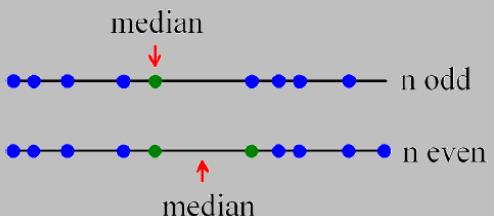
The value for which q% of the values are smaller and 100-q% are larger. The median is the 50%-quantile.

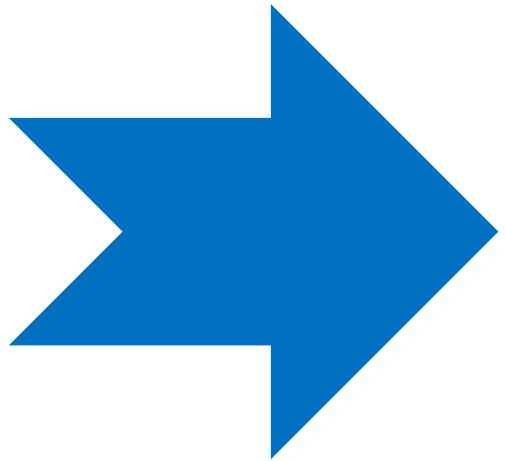
### **Quartiles:**

25%-quantile (1<sup>st</sup>), median (2<sup>nd</sup>), 75%-quantile (3<sup>rd</sup>)

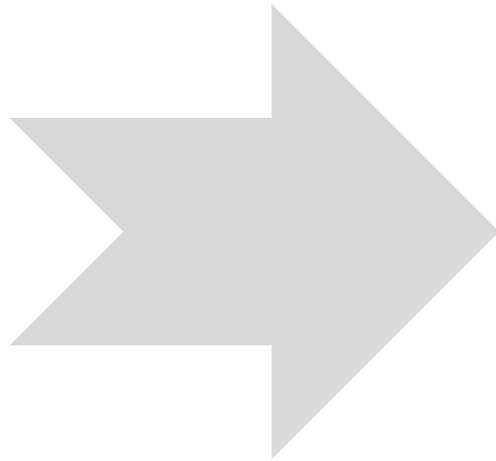
### **Interquartile range:**

$3^{\text{rd}} \text{ quantile} - 1^{\text{st}} \text{ quantile}$





**Low-dimensional  
relationships**  
Univariate Analysis  
[Bivariate Analysis](#)



**Higher-dimensional  
relationships**  
Principal Component Analysis  
Parallel Coordinates

# Common visualizations

## Scatter plots

Scatter plots visualize two variables in a two-dimensional plot

Each axes corresponds to one variable

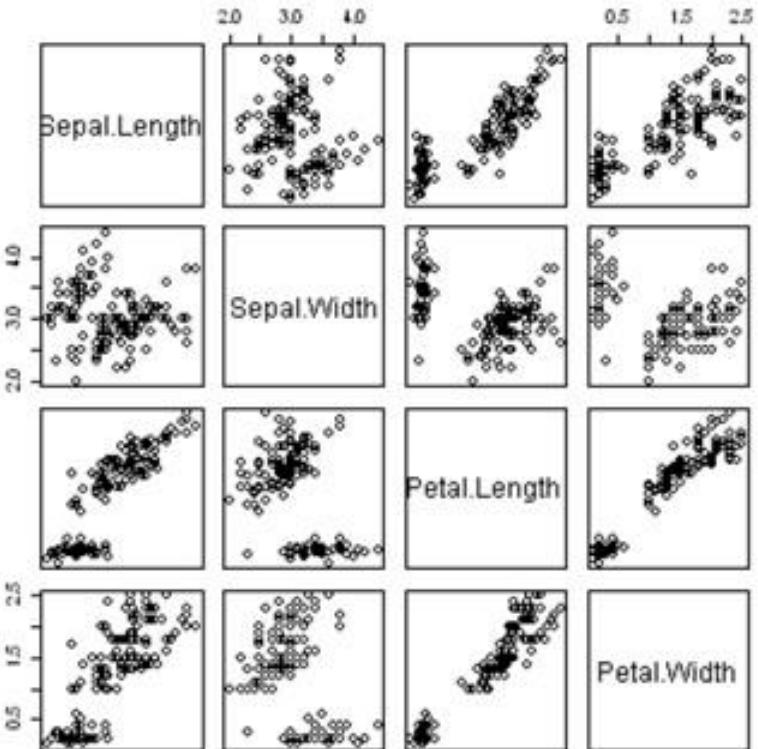
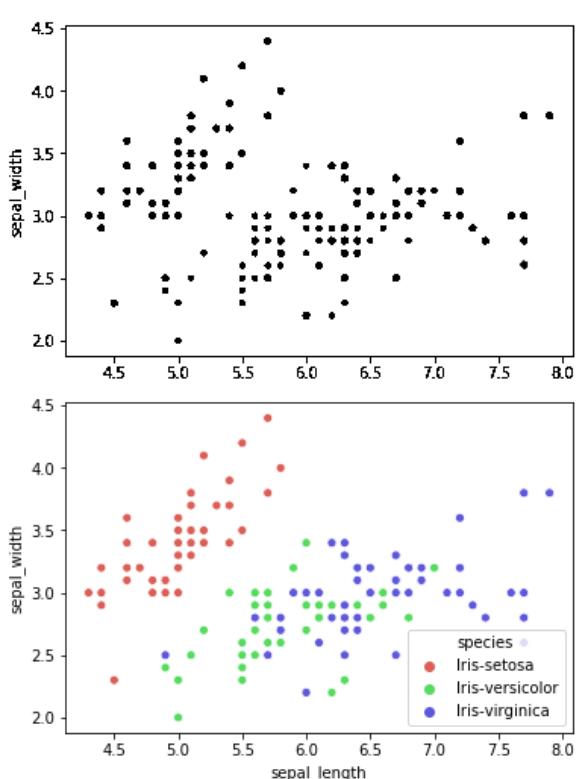
Not suited for larger data sets

```
import pandas as pd
import seaborn as sns

iris = pd.read_csv('irisData.csv', names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])

# Describe relationships among variables in scatter plot
# hue: Variable used for color mapping
sns.scatterplot(data=iris, x="sepal_length", y="sepal_width", hue="species", palette="hls")

# Plot pairwise relationships in a dataset.
sns.pairplot(iris, hue="species", palette="hls")
# see https://seaborn.pydata.org/generated/seaborn.pairplot.html
```



# Common visualizations

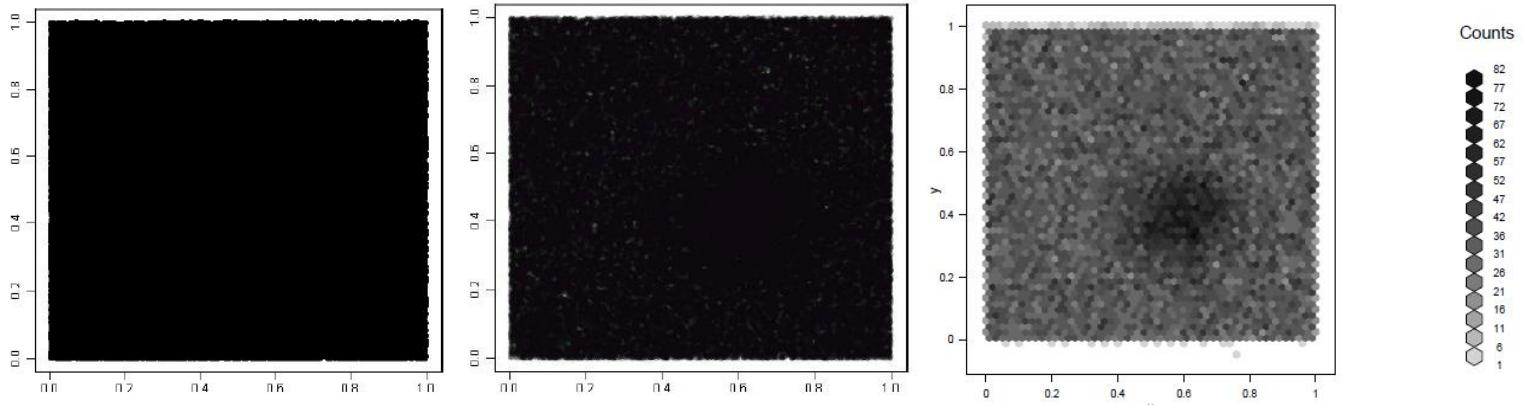
## Scatter plots: density

For large data sets, points are plotted over each other and density information is lost.

Left:  
1000000 objects

Middle:  
Instead of solid points, semitransparent points are plotted

Right:  
hexagonal binning. Grey intensity denotes number of points



### Iris Data Set Example

```
import pandas as pd
import seaborn as sns

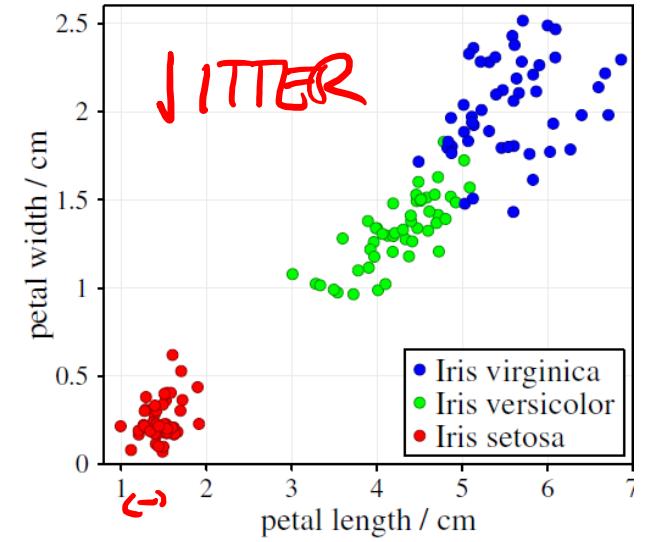
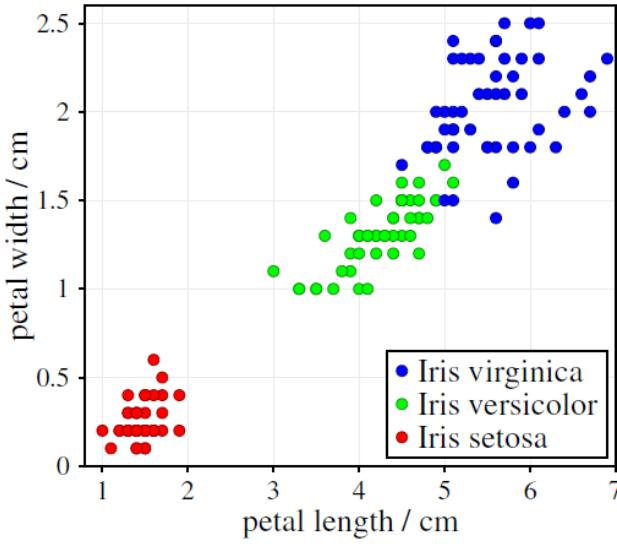
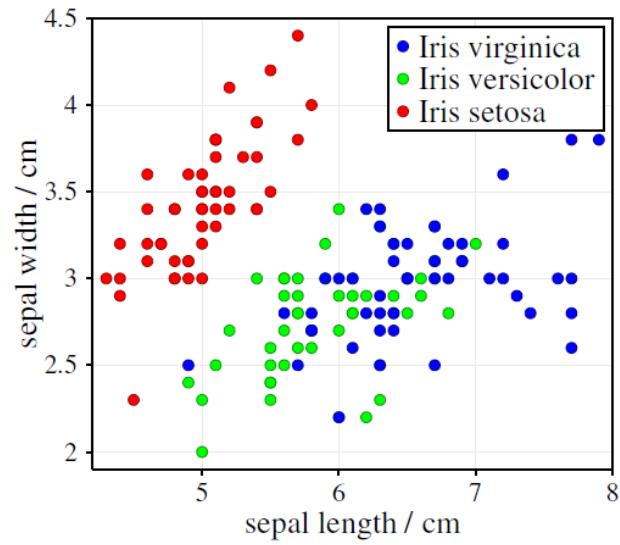
iris = pd.read_csv('irisData.csv', names=['sepal_length','sepal_width',
'petal_length','petal_width','species'])

iris.plot.hexbin(x="sepal_length", y="sepal_width", gridsize=20)
sns.jointplot(data=iris, x="sepal_length", y="sepal_width", kind="hex",
color="k", joint_kws=dict(gridsize=20), marginal_kws=dict(bins=15, rug=True))
```



# Common visualizations

Scatter plots: further elaboration



Scatter plots can be **enriched** with additional information:  
color or different symbols incorporate **a third attribute** in the scatter plot.

What differences does this reveal?

Data objects with the same values cannot be distinguished in a scatter plot → **jitter** (adding random noise)

# Correlation analysis

Scatter plots can “visually” reveal correlations or dependencies between two attributes.

Statistical measures for correlation are a more formal approach to correlation analysis and can be carried out automatically.

We briefly sketch...

Pearson's correlation coefficient

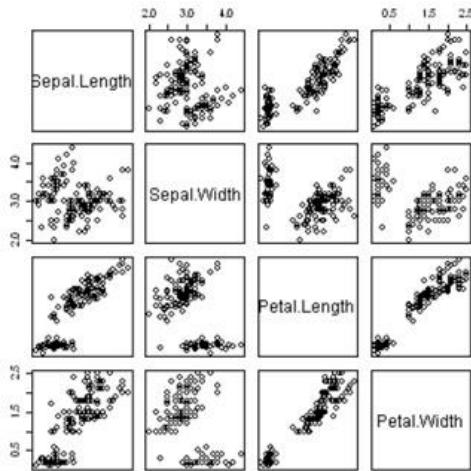
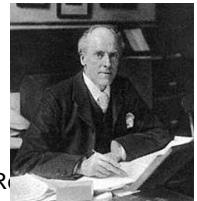
>> [video for explanation](#)

Rank correlation coefficients

>> [video for explanation](#)

Spearman's rho

Kendall's tau



```
import pandas as pd

iris = pd.read_csv('irisData.csv', names=...)

print("Show Pearson's correlation:")
print(iris.corr())
#
print()
print("Show Spearman's rho correlation:")
print(iris.corr('spearman'))
#
print()
print("Show Kendall's tau correlation:")
print(iris.corr('kendall'))
```



Show Pearson's correlation:

|              | sepal_length | sepal_width | petal_length | petal_width |
|--------------|--------------|-------------|--------------|-------------|
| sepal_length | 1.000000     | -0.109369   | 0.871754     | 0.817954    |
| sepal_width  | -0.109369    | 1.000000    | -0.420516    | -0.356544   |
| petal_length | 0.871754     | -0.420516   | 1.000000     | 0.962757    |
| petal_width  | 0.817954     | -0.356544   | 0.962757     | 1.000000    |

Show Spearman's rho correlation:

|              | sepal_length | sepal_width | petal_length | petal_width |
|--------------|--------------|-------------|--------------|-------------|
| sepal_length | 1.000000     | -0.159457   | 0.881386     | 0.834421    |
| sepal_width  | -0.159457    | 1.000000    | -0.303421    | -0.277511   |
| petal_length | 0.881386     | -0.303421   | 1.000000     | 0.936003    |
| petal_width  | 0.834421     | -0.277511   | 0.936003     | 1.000000    |

Show Kendall's tau correlation:

|              | sepal_length | sepal_width | petal_length | petal_width |
|--------------|--------------|-------------|--------------|-------------|
| sepal_length | 1.000000     | -0.072112   | 0.717624     | 0.654960    |
| sepal_width  | -0.072112    | 1.000000    | -0.182391    | -0.146988   |
| petal_length | 0.717624     | -0.182391   | 1.000000     | 0.803014    |
| petal_width  | 0.654960     | -0.146988   | 0.803014     | 1.000000    |

# Pearson's correlation coefficient

The (sample) Pearson's correlation coefficient is a measure for a linear relationship between two numerical attributes  $X$  and  $Y$  and is defined as

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)s_x s_y}$$

where  $\bar{x}$  and  $\bar{y}$  are the mean values of the attributes  $X$  and  $Y$ , respectively.  $s_x$  and  $s_y$  are the corresponding (sample) standard deviations.

The larger the absolute value of the Pearson correlation coefficient, the stronger the **linear relationship** between the two attributes.

$$-1 \leq r_{xy} \leq 1$$

Pearson's correlation assumes normal distribution (vulnerable to skewed data) and linear relationships.

Applicable to **continuous** variables.

# Rank correlation coefficient

Please read  
on your own

Pearson's correlation coefficient measures linear correlation. Even for monotone functional, but non-linear relationship Pearson's correlation coefficient will not be -1 or 1. It can even be close to zero despite a monotone functional relationship.

**Rank correlation coefficients** avoid this by ignoring the exact numerical values of the attributes and *considering only the ordering* of the values.

They intend to measure monotonous correlations between attributes, where the monotonous function does not have to be linear.

Example: Aggregate Single Sales (US)

| Pos | Artist and Title                     | Sales estimate | This year |
|-----|--------------------------------------|----------------|-----------|
| 1   | Mark Ronson - Uptown Funk            | 7,470,000      | 120,000   |
| 2   | Pharrell Williams - Happy            | 7,280,000      | 40,000    |
| 3   | Katy Perry - Dark Horse              | 6,230,000      | 20,000    |
| 4   | Taylor Swift - Shake It Off          | 5,840,000      | 60,000    |
| 5   | Meghan Trainor - All About That Bass | 5,710,000      | 20,000    |

ordinal    continuous

# Rank correlation coefficients

## Spearman's rho

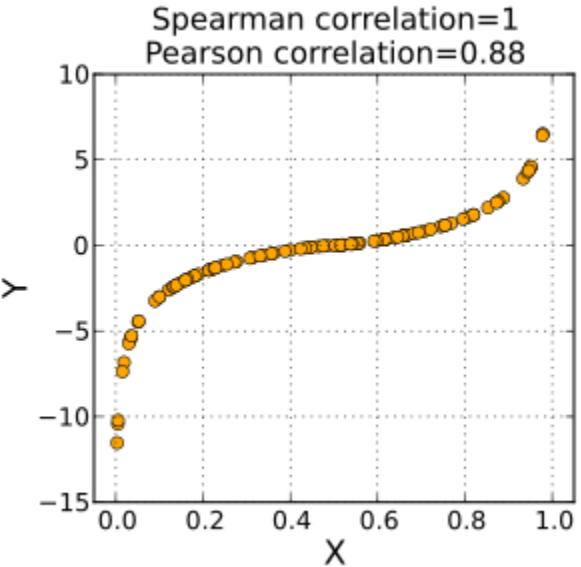
Spearman's rank correlation coefficient (**Spearman's rho**) is defined as

$$\rho = 1 - 6 \frac{\sum_{i=1}^n (r(x_i) - r(y_i))^2}{n(n^2 - 1)},$$

where we sum the deviations between  $r(x_i)$  – the rank of value  $x_i$  when we sort the list  $(x_1, \dots, x_n)$  in increasing order – and  $r(y_i)$ .

When the rankings of the  $x$ - and  $y$ -values are exactly in the same order, Spearman's rho will yield the value 1.

If they are in reverse order, we will obtain the value -1.



Spearman's rho makes no assumption on the distribution and is applicable to **continuous** and **discrete** (ordinal) variables.

It is sensitive to large deviations.

# Rank correlation coefficients

## Kendall's tau

Kendall's tau rank correlation coefficient (Kendall's tau) is defined as

$$\tau_a = \frac{C - D}{\frac{1}{2}n(n-1)}$$

where  $C$  and  $D$  denote the numbers of concordant (similar rank order) and discordant pairs with similar ranks, respectively.

$$C = |\{(i, j) | x_i < x_j \text{ and } y_i < y_j\}|$$

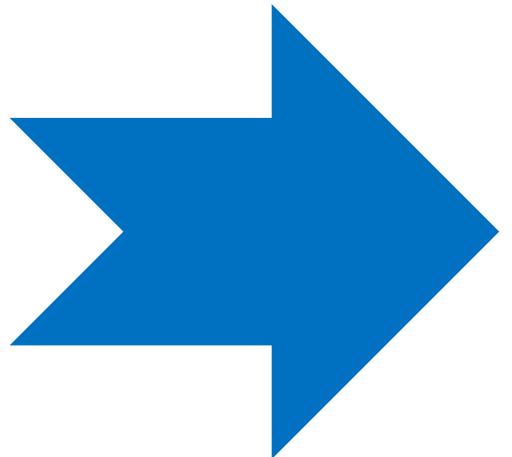
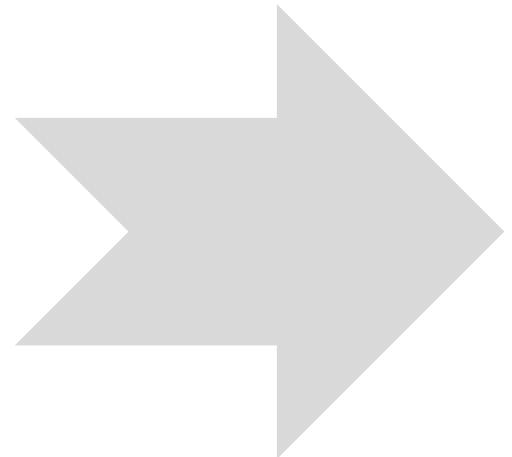
$$D = |\{(i, j) | x_i < x_j \text{ and } y_i > y_j\}|$$

Kendall's tau makes no assumption on the distribution.

Kendall's tau<sub>a</sub> is applicable to **continuous** and **discrete** (incl. ordinal) variables

Less sensitive to errors and discrepancies in the data as Spearman.





## Low-dimensional relationships

Univariate Analysis  
Bivariate Analysis

## Higher-dimensional relationships

Principal Component Analysis  
Parallel Coordinates

# Outlook I: Methods for higher-dimensional data

(and an introductory example about the main idea of **Principal Component Analysis**)

General approach for incorporating all attributes in a plot:

There is no unique measure for structure preservation.

Try to preserve as much of the “structure” of the high-dimensional data set when **representing (plotting) the data in two (or three) dimensions**

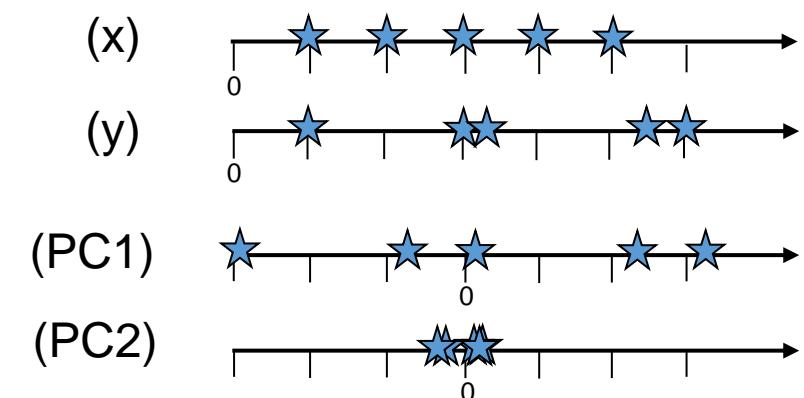
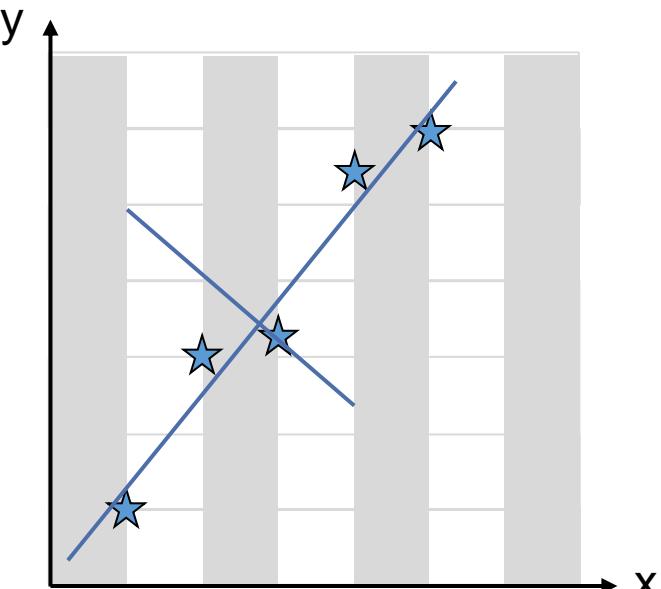
Define a measure that evaluates lower-dimensional representations (plots) of the data in terms of **how well a representation preserves the original “structure”** of the high-dimensional data set.

Find the representation (plot) that gives the best value for the defined measure.

Next Lesson



From  $\mathbb{R}^2$  to  $\mathbb{R}^1$



Next Lesson: More details about PCA

# Outlook II: Data understanding vs. Data preparation

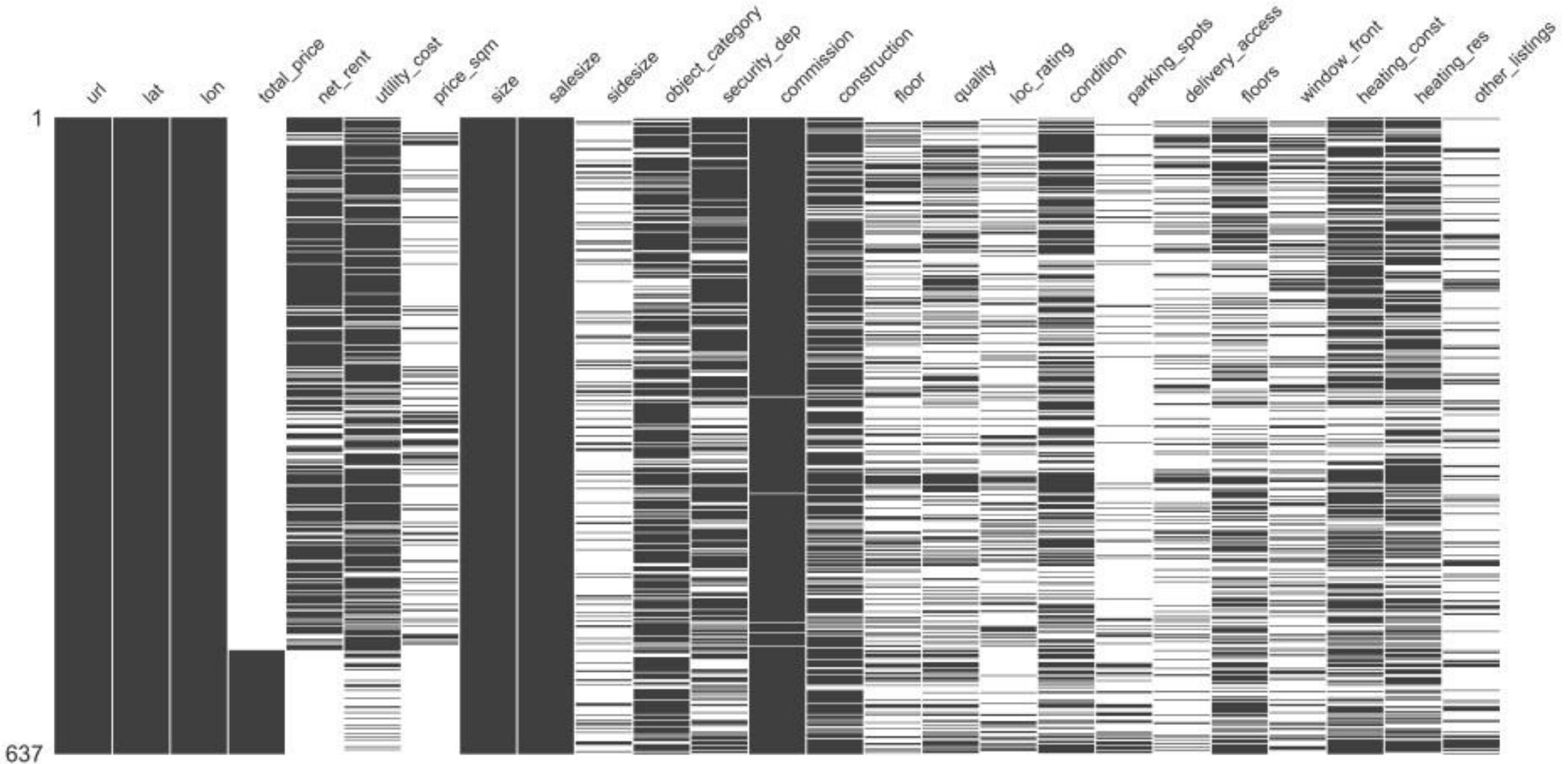
Next Lesson

Freie Universität Berlin



Berlin

Example: Which attributes should be selected?



Ref. Master Thesis Konrad (2019)

## Fragen?

- ✓ Data understanding I
  - ✓ Attribute Understanding
  - ✓ Data Quality
- ✓ Data visualization, correlation analysis  
(Data understanding II)
- ✓ Low-dimensional relationships
  - ✓ Univariate Analysis
  - ✓ Bivariate Analysis
- Higher-dimensional relationships
  - Principal Component Analysis
  - Parallel Coordinates

# Recommended reading

Berthold et al. Chapter 4

Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann, 2011

# Todos for next Week

- Python-Analytics – Chapter 2  
*Kursmaterial > Readings/Übungen > Python Übungen – Jupyter*
- Please try the sample code on the Iris dataset on your own.
- Please get familiar with the PCA (incl. excursus) on the following slides.

# Principal Component Analysis (PCA)

Structure preservation through variance in data set

[Next Lesson](#)

Freie Universität



Berlin

PCA compresses a large data set to capture the essence of the *original data* through linear transformation

PCA constructs **a projection** from the high-dimensional space to a lower-dimensional space (plane or hyperplane) using only the most relevant dimensions

PCA uses the **variance in the data** set as the structure preservation criterion.

Assumption: Large variances describe interesting dynamics, smaller noise.

PCA preserves as much of the original variance of the data when projected to a lower-dimensional space

**(Sample) variance** for a numerical attribute:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{n \sum_{i=1}^n x_i^2 - \left( \sum_{i=1}^n x_i \right)^2}{n(n-1)}$$

# Principal Component Analysis

Next Lesson

Freie Universität



Procedure: Objective

The data points are first **centered around the origin** by subtracting the mean values

Objective:

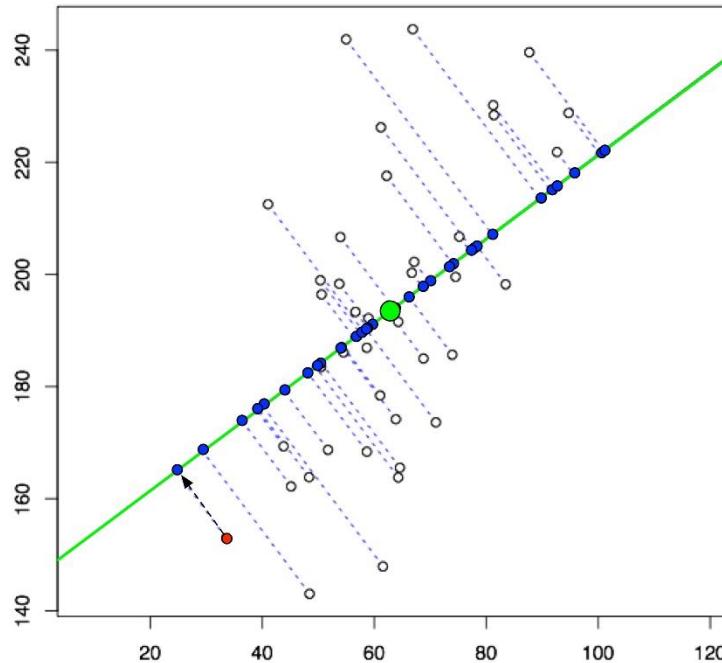
find a projection in the form of a linear mapping given by  $y = M(x - \bar{x})$ , where  $M$  is a  $q \times m$  matrix such that the **variance** of the projected data  $y_i = M(x_i - \bar{x})$  is **maximized**

( $2 \times m$  for projections to a plane)

PCA uses the **covariance matrix** which holds information on spread (variance) and orientation (covariance)

$$\Sigma = \begin{bmatrix} \sigma(x, x) & \sigma(x, y) \\ \sigma(y, x) & \sigma(y, y) \end{bmatrix}$$

*Projecting 2 dimensions on 1*



See excursus for in-depth information

# Principal Component Analysis

Next Lesson

Freie Universität



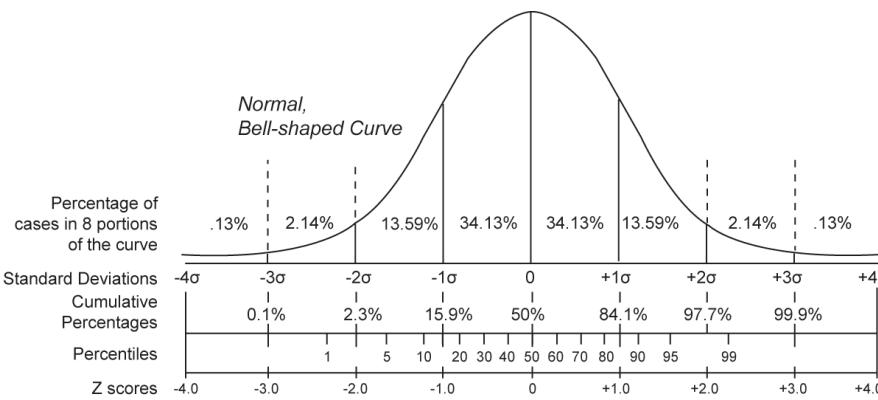
## Procedure: Problem

Problem:

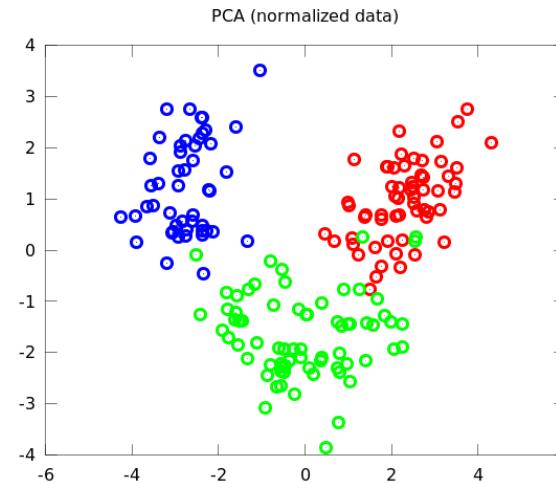
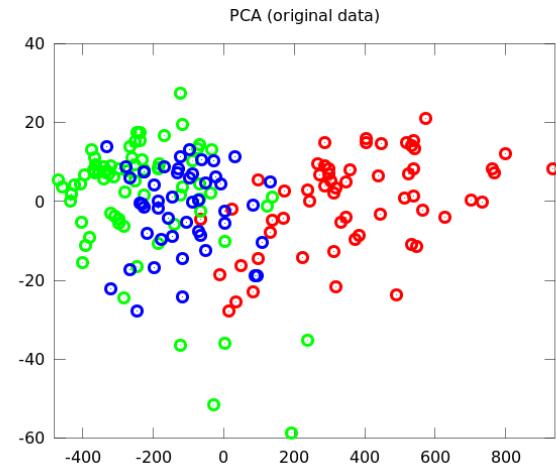
Without restriction for the matrix  $M$ , the entries in  $M$  can be chosen arbitrary large so that the data are not only projected, but also **scaled**, leading to an arbitrary large variance of the projected data.

We introduce **constraints** such that the matrix  $M$  is only a projection:

The row  $v_i$  of the matrix  $M = (v_1, \dots, v_q)$  must be **normalized**, i.e.,  $\|v_i\| = 1$ .



Ref. e.g., Kristensen & Terje (2016, p. 81 ff.)



Usually, the data should be **zero-score standardized** ( $x \rightarrow \frac{x - \hat{\mu}_x}{\hat{\sigma}_x}$ ) to ensure that all attributes contribute equally to the overall variance (with  $\hat{\mu}_x$  being the mean value and  $\hat{\sigma}_x$  the sample standard deviation of attribute  $X$ , z-score: numeric distance of  $x$  in standard deviations from mean)

Images: [Wagner \(2011\)](#)

# Principal Component Analysis

Choosing principal components

Solution of the constraint optimization problem:

The projection matrix  $M$  is given by  $M = (v_1, \dots, v_q)$ ,

where the **principal components**  $v_1, \dots, v_q$

are the *normalized eigenvectors of the covariance matrix* of the attributes in the data set

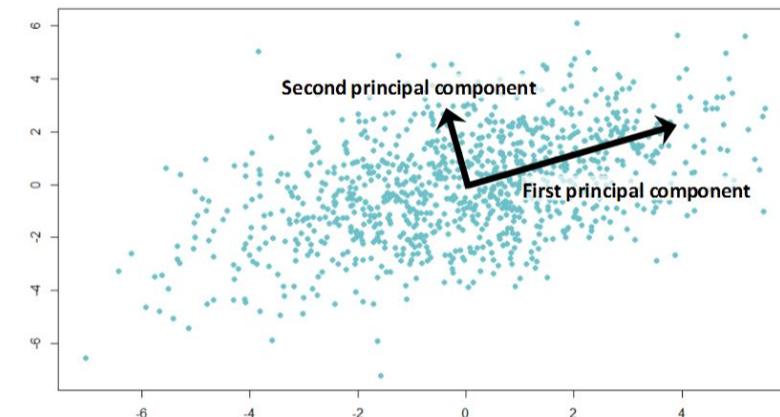
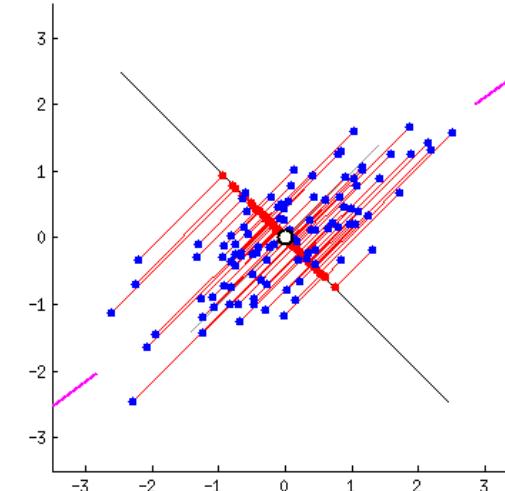
$$\sigma_{jk} = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)^T$$

for the  $q$  **largest eigenvalues**  $\lambda_1 \geq \dots \geq \lambda_q$ .

$\lambda$  is called an eigenvalue of a matrix  $A$ , if there is a non-zero vector  $v$  such that  $A\mathbf{v} = \lambda\mathbf{v}$  holds. The vector  $v$  is called eigenvector (direction of the data) to the eigenvalue  $\lambda$  (magnitude of its spread).

Next Lesson

Freie Universität



# Principal Component Analysis

Dimension reduction

Next Lesson

Freie Universität

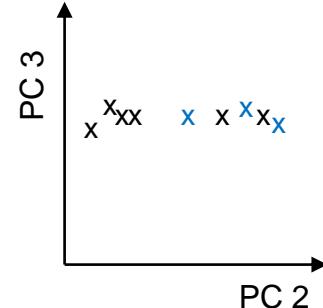


Let  $\lambda_1 \geq \dots \geq \lambda_m$  be the eigenvalues of the covariance matrix.

When we project the data to the first  $q$  principal components  $v_1, \dots, v_q$  corresponding to the eigenvalues  $\lambda_1, \dots, \lambda_q$ , this projection will preserve a fraction of the variance of the original data.

$$\frac{\lambda_1 + \dots + \lambda_q}{\lambda_1 + \dots + \lambda_m}$$

Only principal components which explain little variance in the data, like...



Iris data set:

|                        | PC1  | PC2   | PC3    | PC4     |
|------------------------|------|-------|--------|---------|
| Proportion of variance | 0.73 | 0.229 | 0.0367 | 0.00518 |
| Cum. proportion        | 0.73 | 0.958 | 0.9948 | 1.00000 |

Ref.

# PCA – Iris data set example (1/2)

[Next Lesson](#)

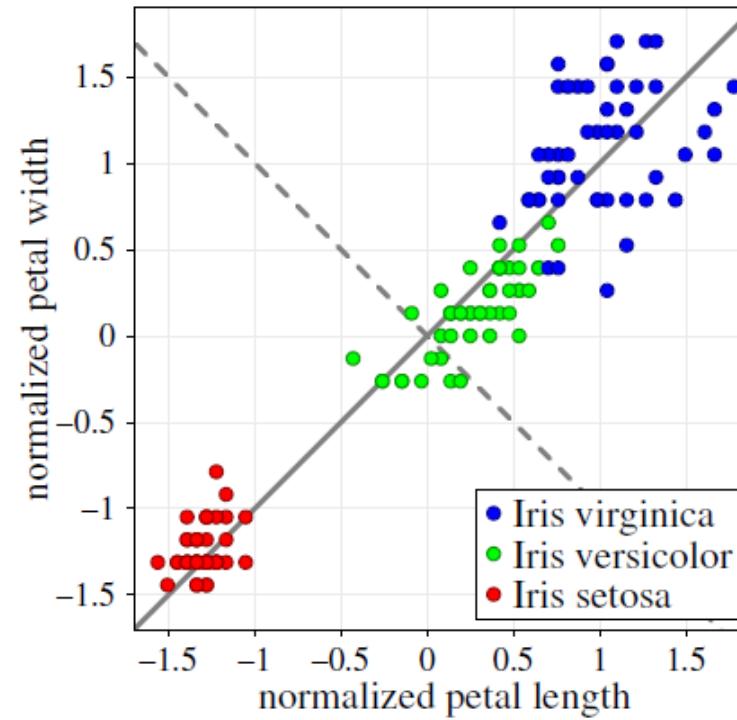
Freie Universität



Berlin

PCA applied to the [Iris data set](#) restricted to the (normalized) petal length and width

The principal components are always *orthogonal*



# PCA – Iris data set example (2/2)

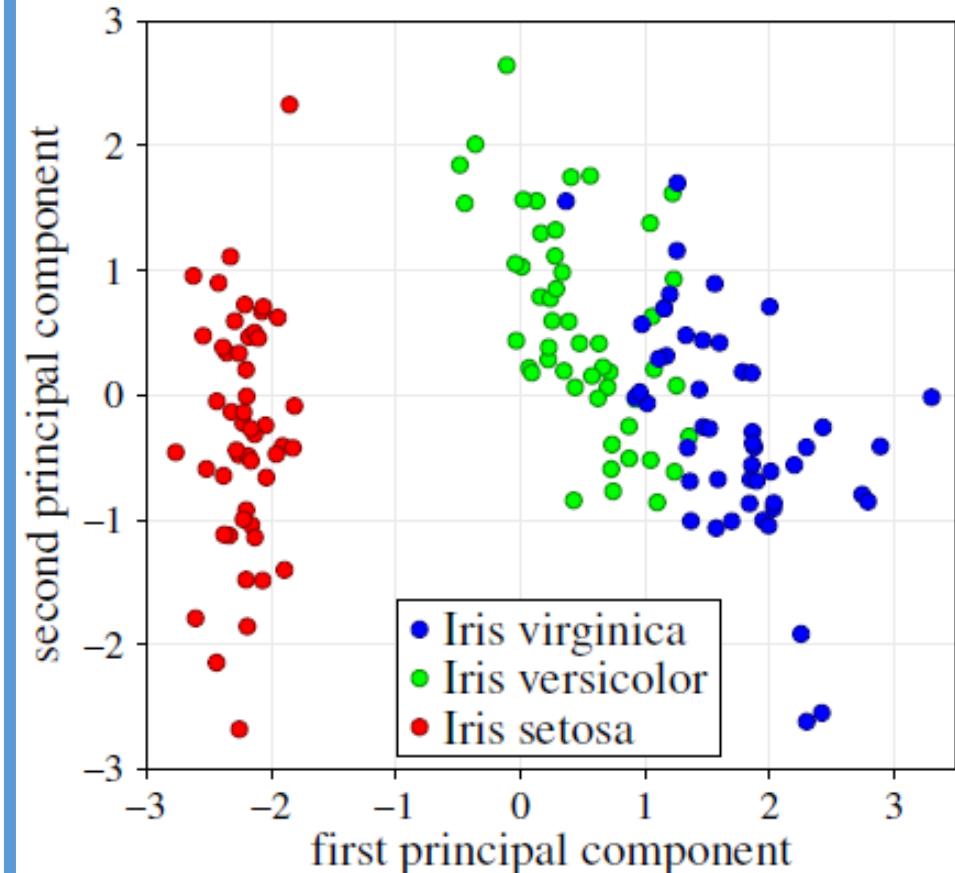
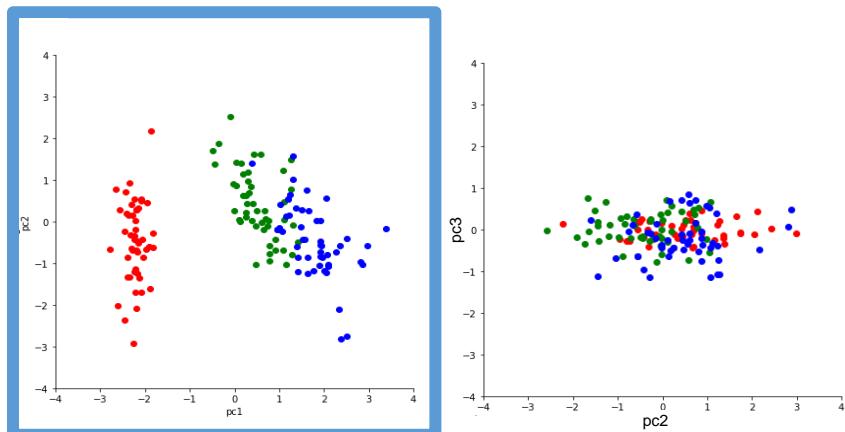
Next Lesson

Freie Universität



Berlin

Projection to the first two principal components of PCA taking all four numerical attributes into account



Original data is **reconstructable** from the principal components

Ref.

# PCA – Chessboard example (1/2)

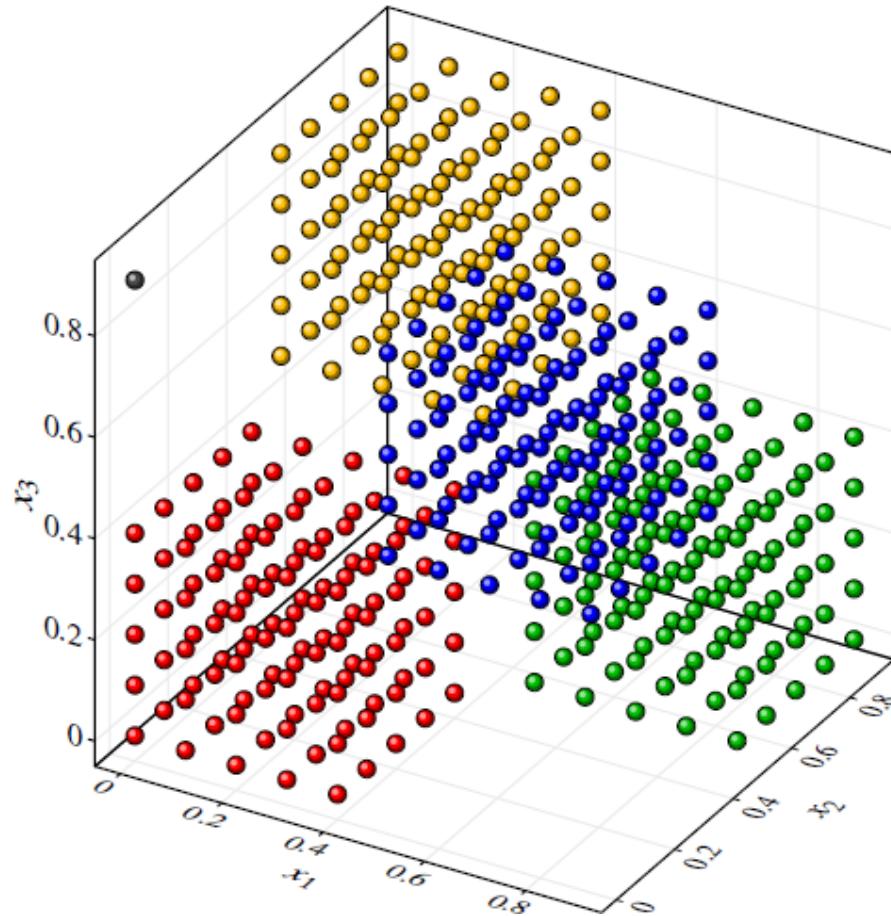
[Next Lesson](#)

Freie Universität



Berlin

An artificial data set filling a cube in a chessboard-like manner



Ref. Berthold et al. (2010)

# PCA – Chessboard example (2/2)

Next Lesson

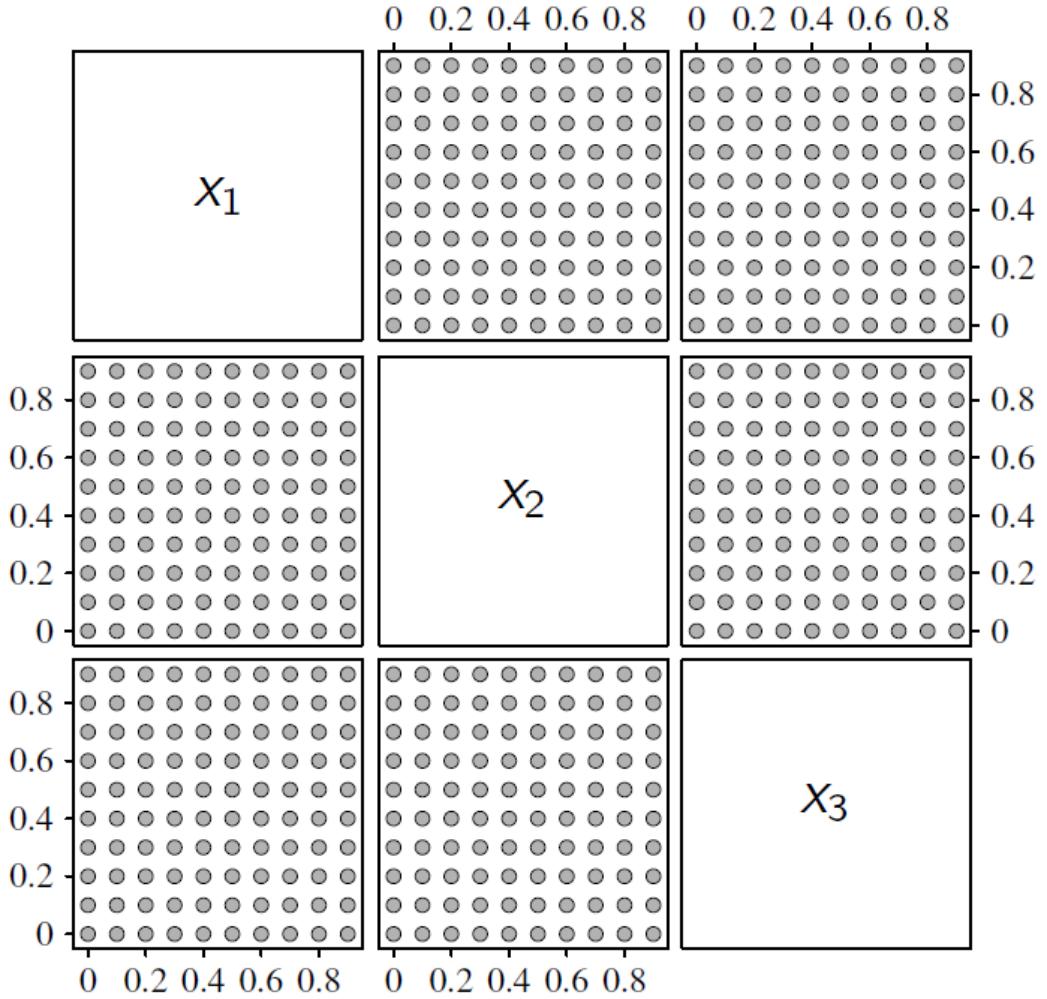
Freie Universität



Berlin

## Scatter plots

Is data uniformly distributed over the grid?



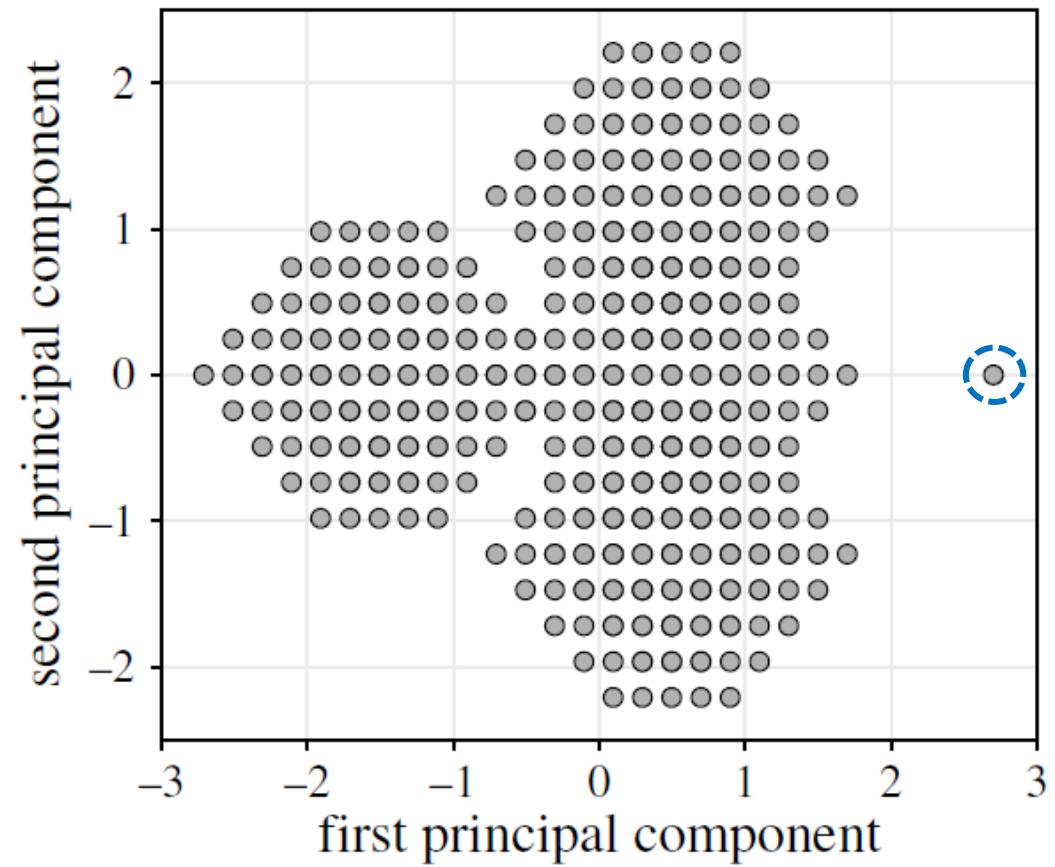
Ref. Berthold et al. (2010)

Projection to the first two principal components

Data is not uniformly distributed.

There is a pattern in the data set.

Data can be recreated from PCA.



# Principal Component Analysis – Iris Data Set

## Python Example

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import scale

iris = pd.read_csv('irisData.csv', names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])
#select only metric data
raw_iris = iris[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']]

#center data to mean
norm_iris = scale(raw_iris)

#create pca with 4-dimensions
pca = PCA(n_components=4)
#pca data
pca_iris = pca.fit_transform(norm_iris)

print(pca.explained_variance_)
print(pca.explained_variance_ratio_)
print(pca.explained_variance_ratio_.cumsum())

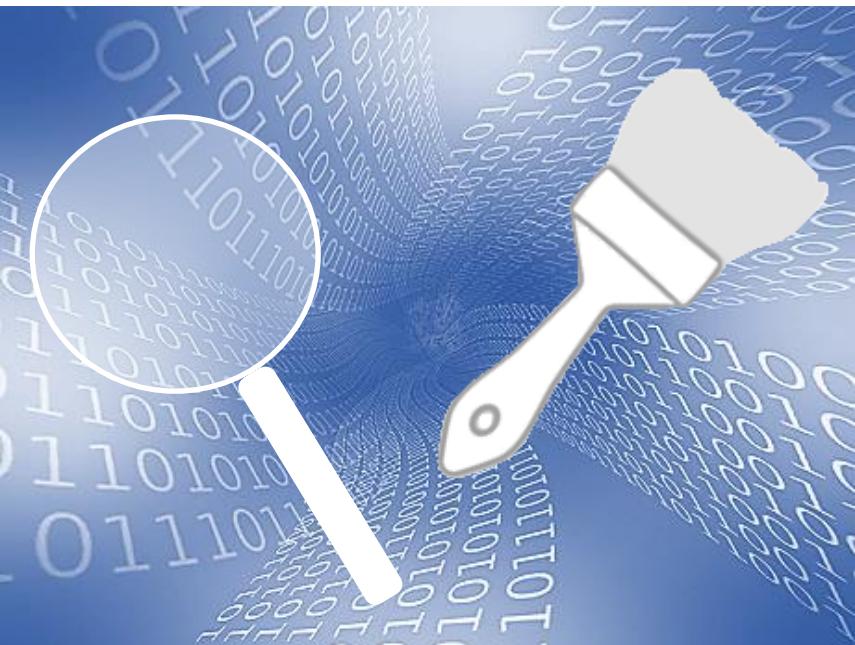
#visualize pcas
vis_iris = pd.DataFrame(pca_iris, columns=['pc1', 'pc2', 'pc3', 'pc4'])
vis_iris['species'] = iris['species']
g = sns.FacetGrid(vis_iris, hue='species', palette=['r', 'g', 'b'], ylim=(-4,4), xlim=(-4,4), height = 6)
g.map(plt.scatter, 'pc1', 'pc2')
plt.show()

g = sns.FacetGrid(vis_iris, hue='species', palette=['r', 'g', 'b'], ylim=(-4,4), xlim=(-4,4), height = 6)
g.map(plt.scatter, 'pc2', 'pc3')
plt.show()

g = sns.FacetGrid(vis_iris, hue='species', palette=['r', 'g', 'b'], ylim=(-4,4), xlim=(-4,4), height = 6)
g.map(plt.scatter, 'pc3', 'pc4')
plt.show()
```

Example in Python





# Business Intelligence

## 07 Data Visualization II

Prof. Dr. Bastian Amberg  
(summer term 2024)

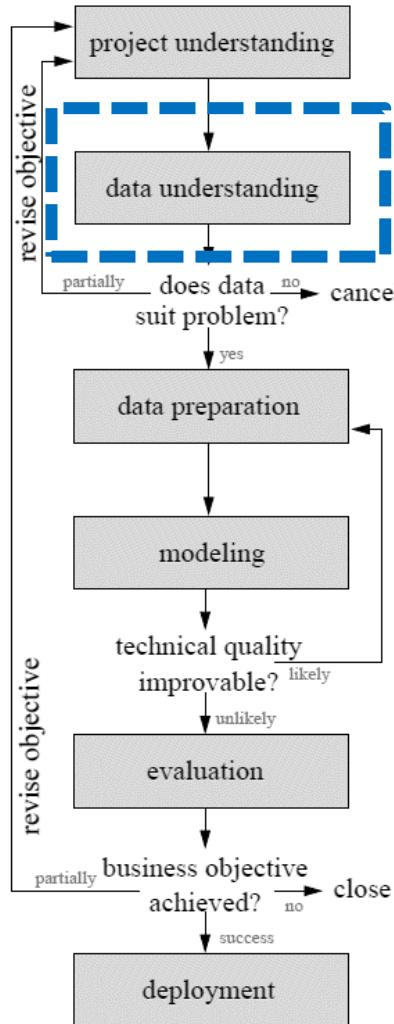
29.5.2024

# Schedule (slightly adjusted)

|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study |                          |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|------------|--------------------------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |            |                          |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |            |                          |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  |            |            |                          |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining Introduction  |            |                          |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |            | Python-Analytics Chap. 1 |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization I            |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam |            | Chap. 2                  |  |  |
|           | W7                | 29.5. | Data Visualization II                               |                                   | 31.5. |   |            |                          |  |  |
|           | W8                | 5.6.  | Data Preparation                                    |                                   | 7.6.  | Predictive Modeling I (10:00 -12:00)  |            | BI-Project Start         |  |  |
|           | W9                | 12.6. | Predictive Modeling II, Fitting a Model I           |                                   | 14.6. | Python-Analytics-Online Exercise  |            |                          |  |  |
|           | W10               | 19.6. | Guest Lecture Dr. Ionescu                           |                                   | 21.6. | Fitting a Model II  |            |                          |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |            |                          |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I                                  |            |                          |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |            |                          |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |            | End                      |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin, 31.7.'24<br>Klausur 2.Termin, 2.10.'24                                      |            | Projektbericht           |  |  |

# Last Lesson

Data understanding I (attribute understanding, data quality)



What exactly is the problem, the expected benefit?

How would a solution look like?

What is known about the domain?

What data do we have available?

Is the data relevant to the problem?

Is it valid? Does it reflect our expectations?

Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?

How is the data best transformed for modeling?

How may we increase the data quality?

What kind of model architecture suits the problem best?

What is the best technique/method to get the model?

How good does the model perform technically?

How good is the model in terms of project requirements?

What have we learned from the project?

How is the model best deployed?

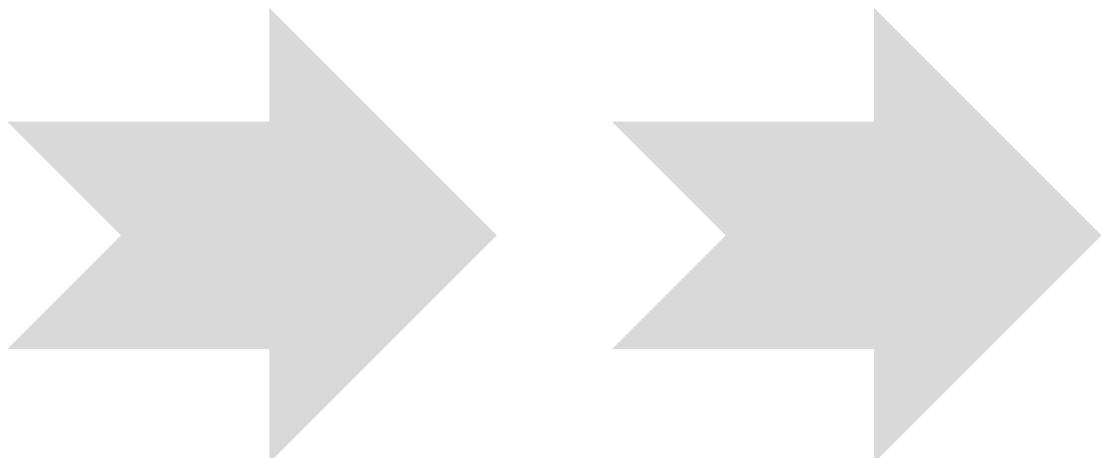
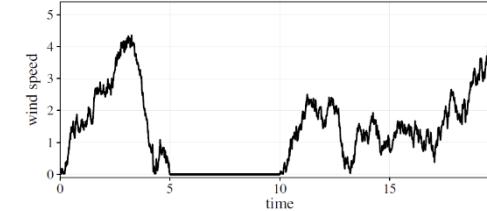
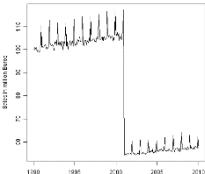
How do we know that the model is still valid?

# Short Introduction

Freie Universität Berlin



Data understanding II (data visualization, correlation analysis)



**Low-dimensional  
relationships**

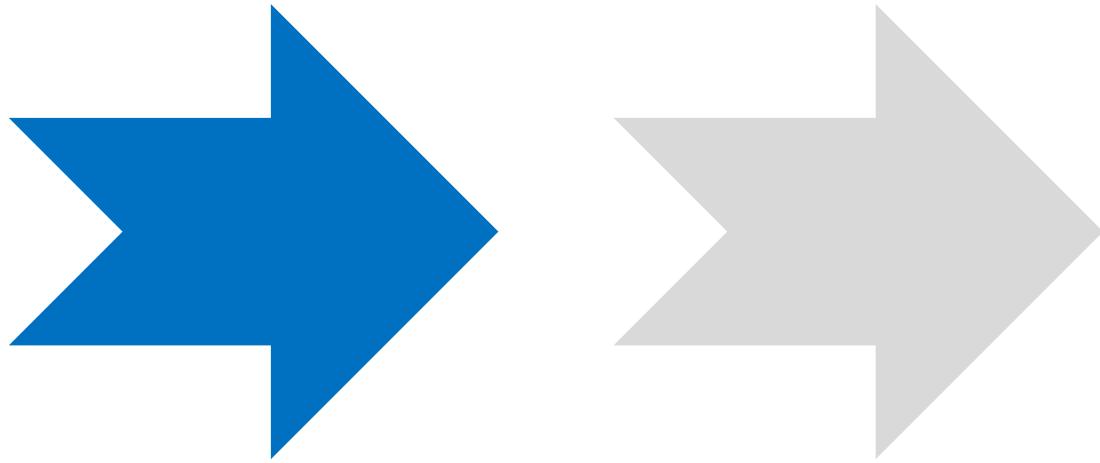
Univariate Analysis

Bivariate Analysis

**Higher-dimensional  
relationships**

Principal Component Analysis

Parallel Coordinates



**Low-dimensional  
relationships**

Univariate Analysis  
Bivariate Analysis

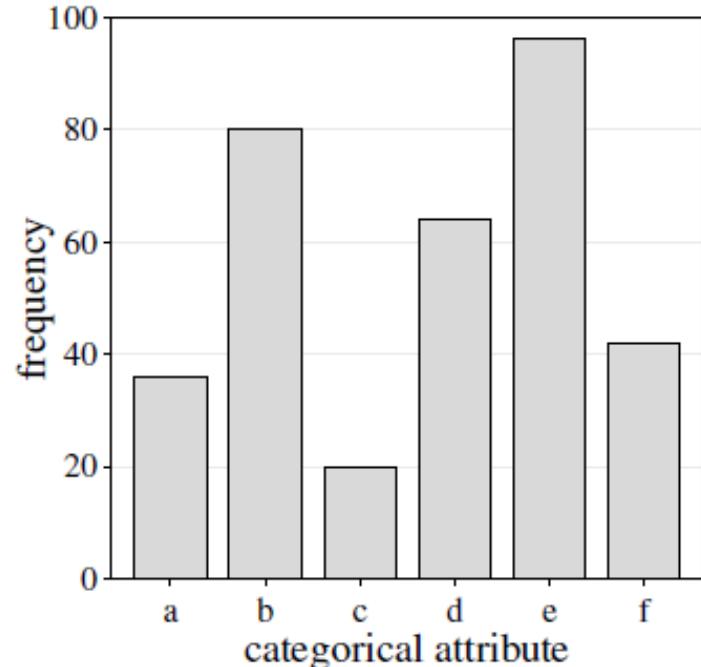
**Higher-dimensional  
relationships**

Principal Component Analysis  
Parallel Coordinates

# Common visualizations

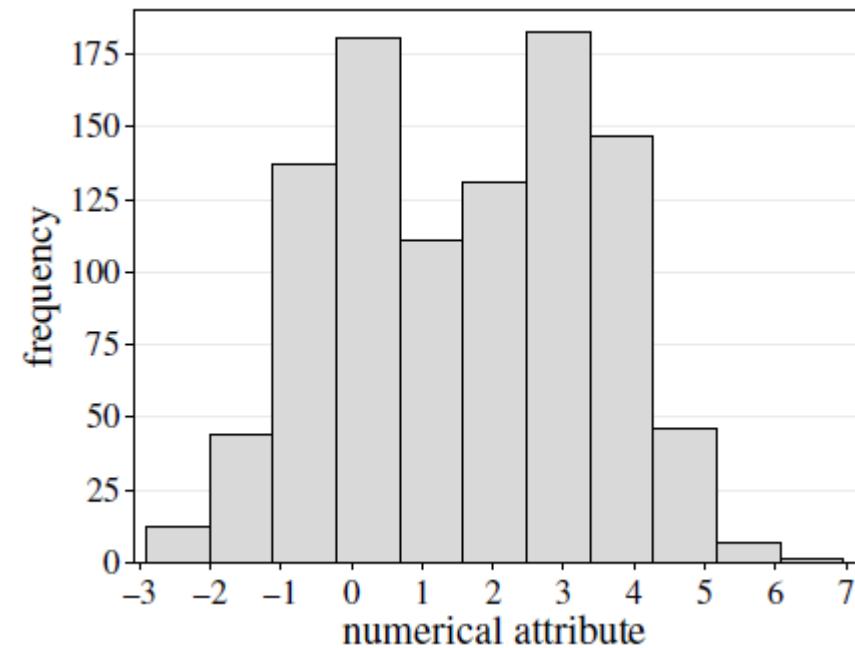
## Bar charts and Histograms

A **bar chart** is a simple way to depict the frequencies of the values of a categorical attribute.



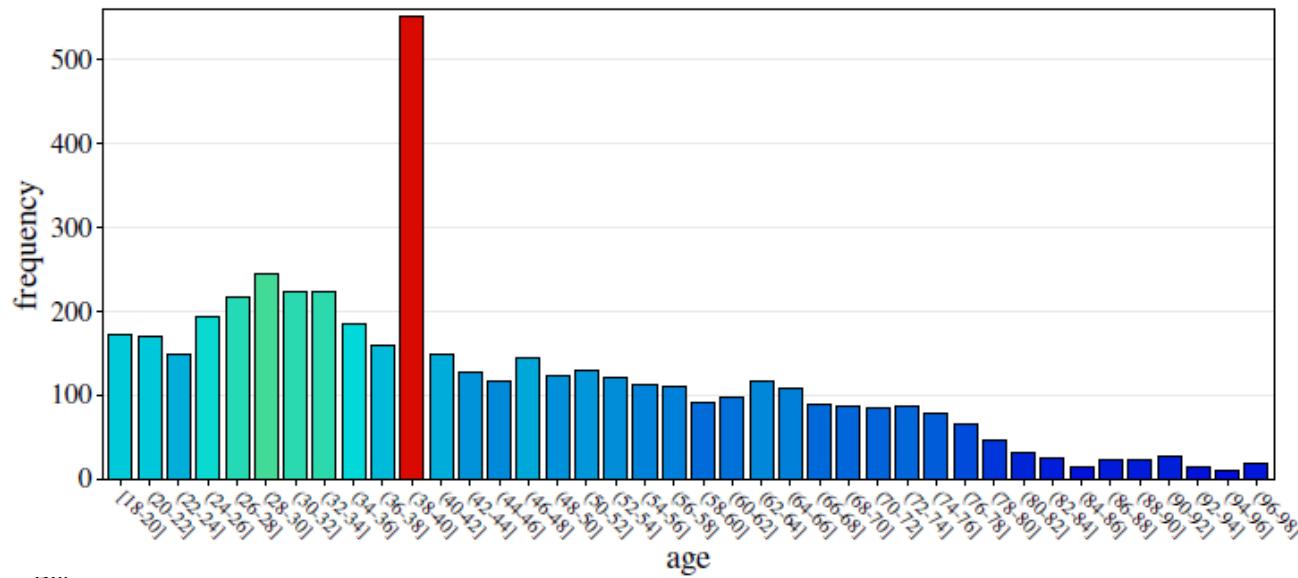
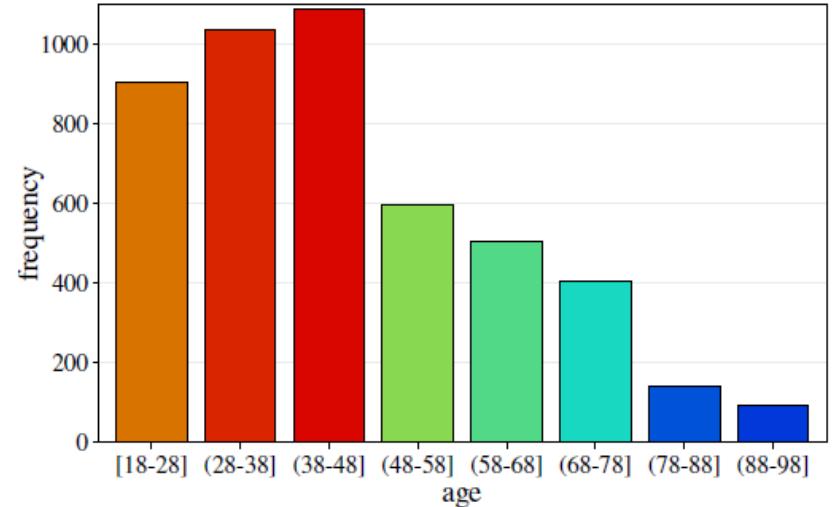
A **histogram** shows the frequency distribution for a numerical attribute.

The range of numerical attribute is discretized into a fixed number of intervals ("bins"), usually of equal length. For each interval, the (absolute) frequency of values falling into it is indicated by the height of a bar.

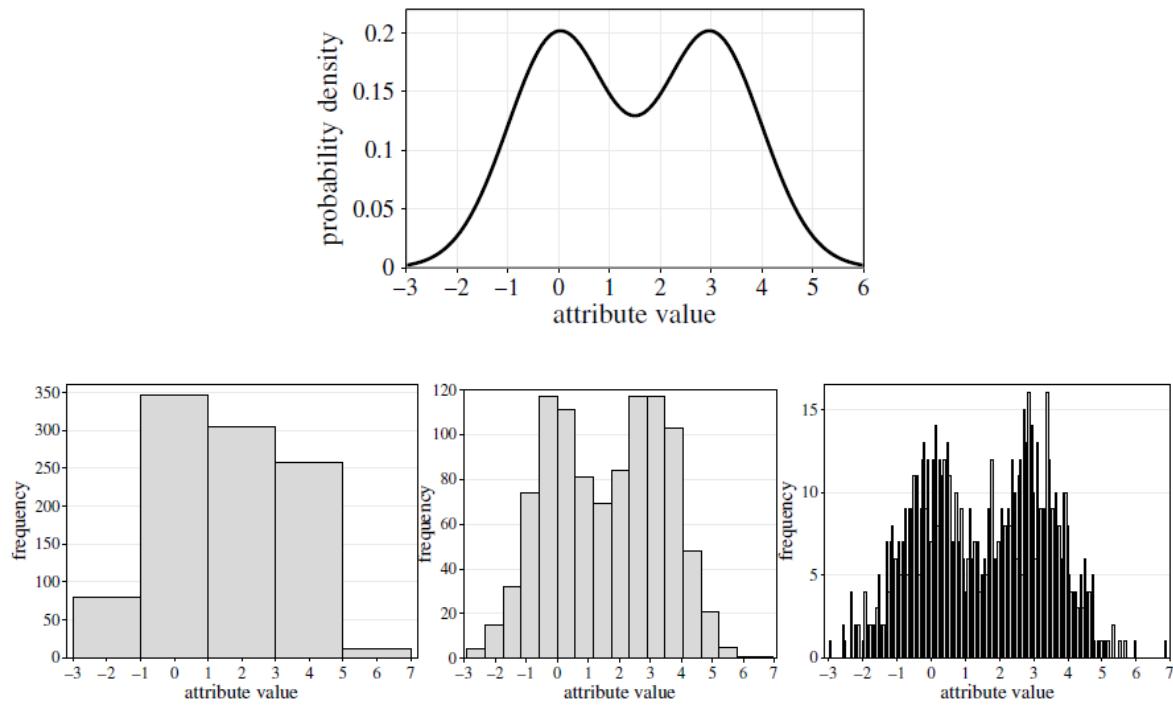


# Common visualizations

Histograms: The number of bins is very important.



Three histograms with 5, 17 and 200 bins for a sample from the same bimodal distribution. Sample size is  $n = 1000$ .



# Example data set

## Iris data

Collected by E. Anderson in 1935

Contains measurements of four real-valued variables of 150  
**iris flowers** of types Iris Setosa, Iris Versicolor, Iris Virginica

- Sepal length [Kelchblatt]
- Sepal widths
- Petal lengths [Blütenblatt]
- Petal widths

The fifth attribute is the name of the flower type

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species         |
|--------------|-------------|--------------|-------------|-----------------|
| 5.1          | 3.5         | 1.4          | 0.2         | Iris-setosa     |
| ...          |             |              |             |                 |
| ...          |             |              |             |                 |
| 5.0          | 3.3         | 1.4          | 0.2         | Iris-setosa     |
| 7.0          | 3.2         | 4.7          | 1.4         | Iris-versicolor |
| ...          |             |              |             |                 |
| ...          |             |              |             |                 |
| 5.1          | 2.5         | 3.0          | 1.1         | Iris-versicolor |
| 5.7          | 2.8         | 4.1          | 1.3         | Iris-virginica  |
| ...          |             |              |             |                 |
| ...          |             |              |             |                 |
| 5.9          | 3.0         | 5.1          | 1.8         | Iris-virginica  |

Ref.



Iris Setosa



Iris Versicolor



Iris Virginica



```
import pandas as pd
# Create DataFrame using Pandas and set Column names
iris = pd.read_csv('irisData.csv', names=['sepal_length','sepal_width','petal_length','petal_width','species'])
# Show descriptive statistics on dimensional distributions
print(iris.describe())
# Show histogram
iris.hist(column='sepal_length', bins = (4.0,4.5,5.0,5.5,6.0,6.5,7.0,7.5,8))
```

# Iris data set: boxplots

**Boxplots** are a very compact way to visualize and summarize main characteristics of a sample from a numerical attribute

Line in the middle = median

Box = interquartile range

Whiskers =  $1.5 \times$  interquartile range

```
import pandas as pd
import seaborn as sns

iris = pd.read_csv('irisData.csv', names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])
sns.boxplot(x="species", y="sepal_length", data=iris, notch=True)
```



## Reminder:

### **Median:**

the value in the middle (for the values given in increasing order)

### **q%-quantile (0<q<100):**

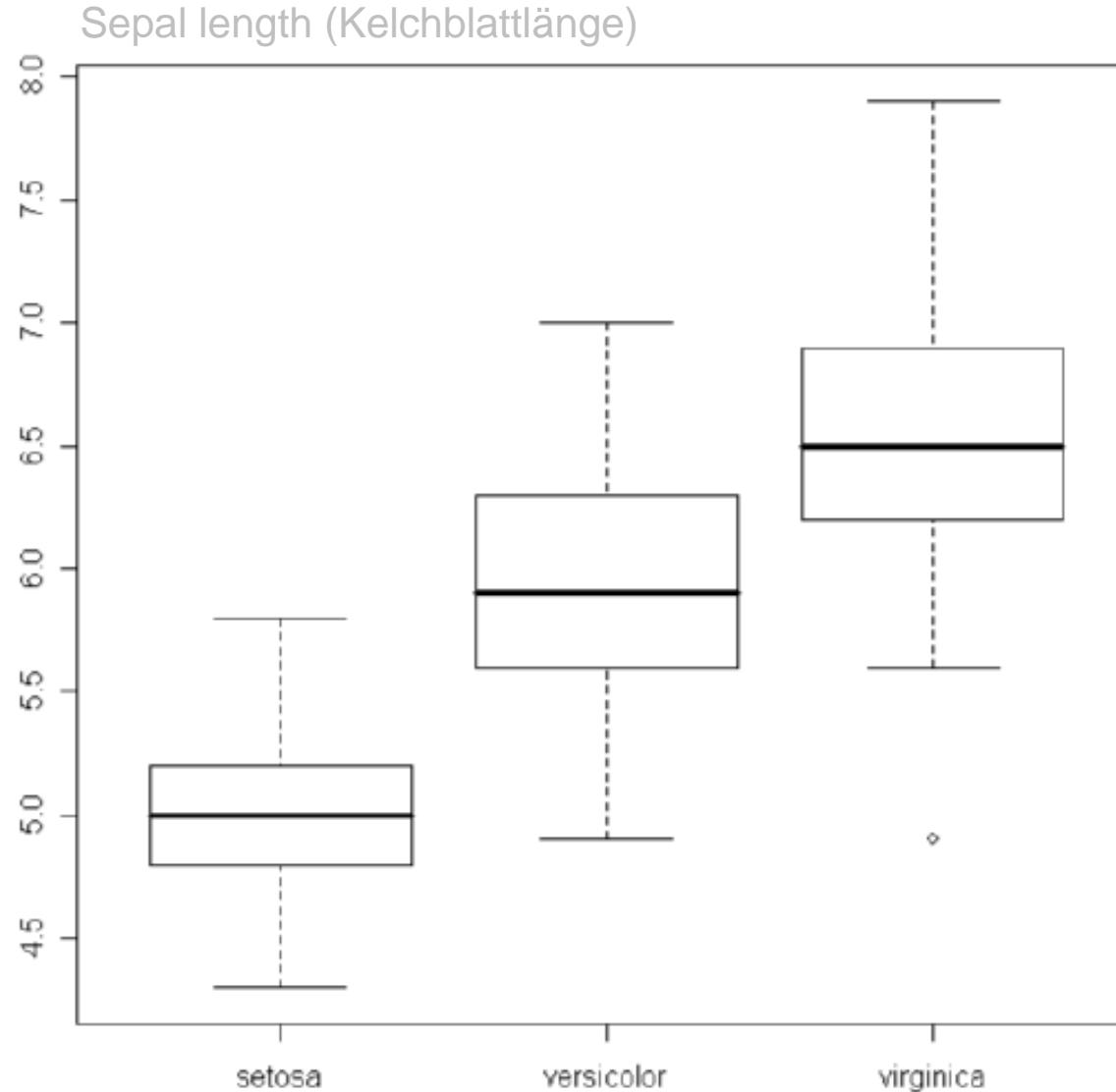
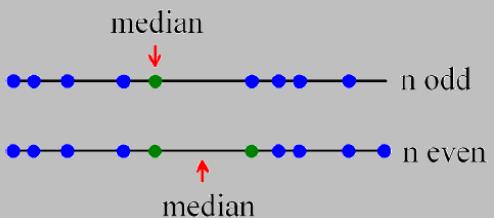
The value for which q% of the values are smaller and 100-q% are larger. The median is the 50%-quantile.

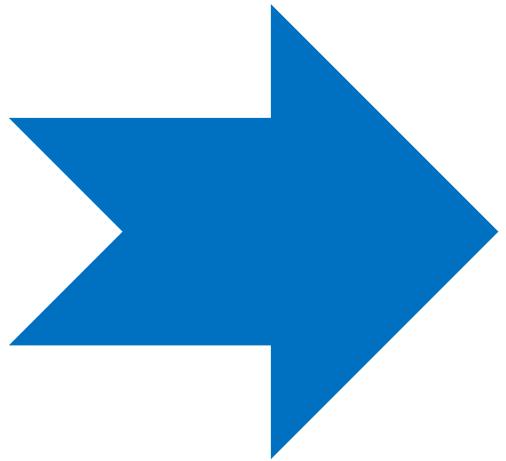
### **Quartiles:**

25%-quantile (1<sup>st</sup>), median (2<sup>nd</sup>), 75%-quantile (3<sup>rd</sup>)

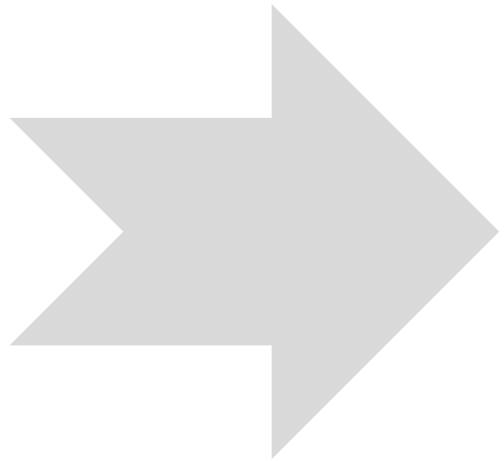
### **Interquartile range:**

$3^{\text{rd}} \text{ quantile} - 1^{\text{st}} \text{ quantile}$





**Low-dimensional  
relationships**  
Univariate Analysis  
[Bivariate Analysis](#)



**Higher-dimensional  
relationships**  
Principal Component Analysis  
Parallel Coordinates

# Common visualizations

## Scatter plots

Scatter plots visualize two variables in a two-dimensional plot

Each axes corresponds to one variable

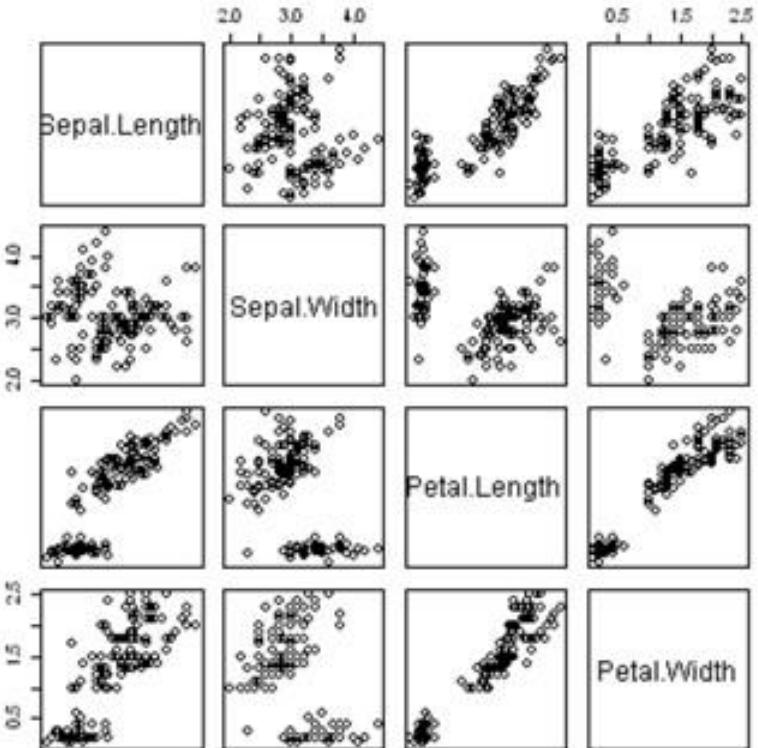
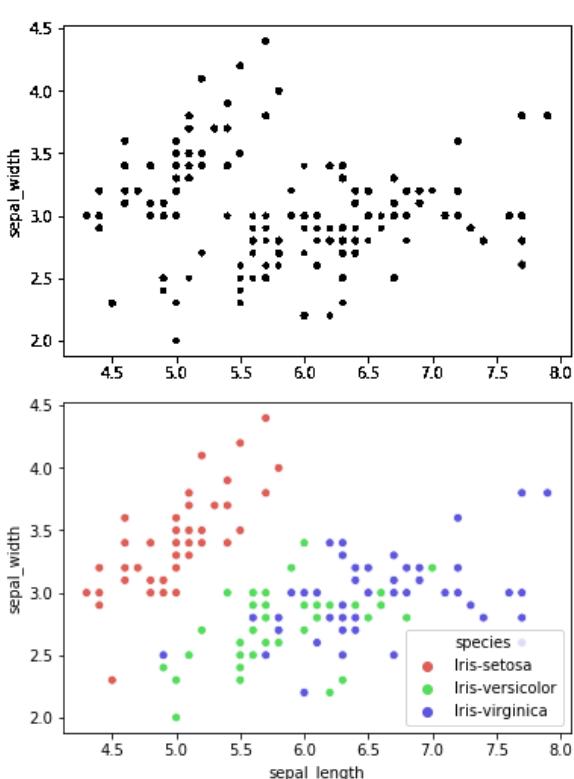
Not suited for larger data sets

```
import pandas as pd
import seaborn as sns

iris = pd.read_csv('irisData.csv', names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])

# Describe relationships among variables in scatter plot
# hue: Variable used for color mapping
sns.scatterplot(data=iris, x="sepal_length", y="sepal_width", hue="species", palette="hls")

# Plot pairwise relationships in a dataset.
sns.pairplot(iris, hue="species", palette="hls")
# see https://seaborn.pydata.org/generated/seaborn.pairplot.html
```



# Common visualizations

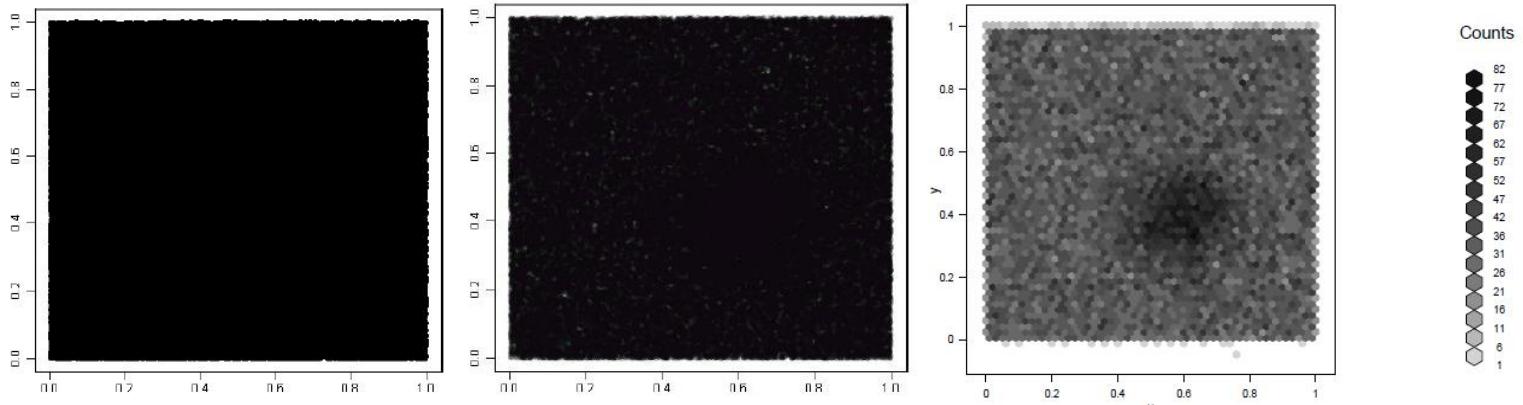
## Scatter plots: density

For large data sets, points are plotted over each other and density information is lost.

Left:  
1000000 objects

Middle:  
Instead of solid points, semitransparent points are plotted

Right:  
hexagonal binning. Grey intensity denotes number of points



### Iris Data Set Example

```
import pandas as pd
import seaborn as sns

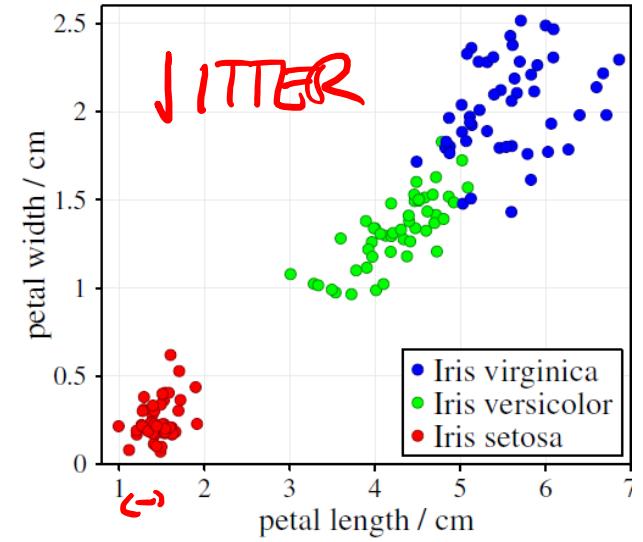
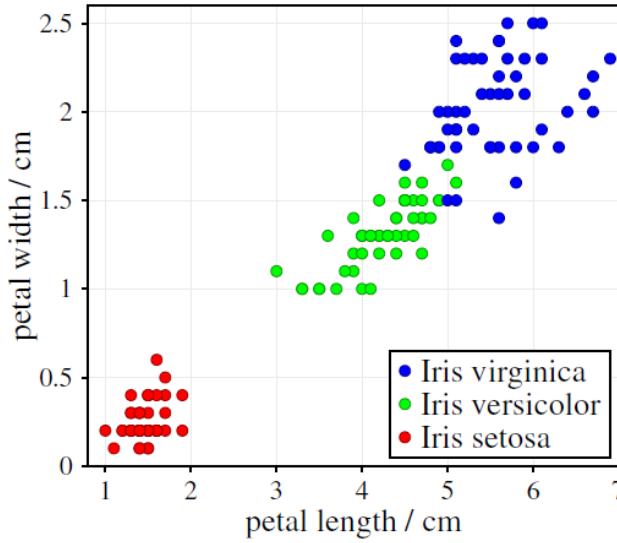
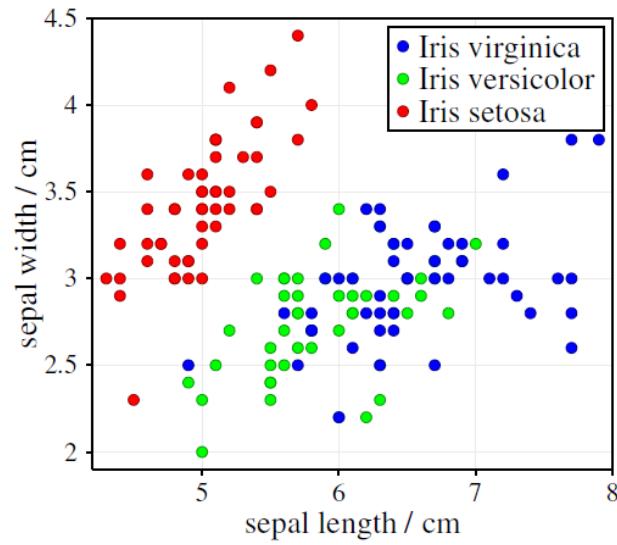
iris = pd.read_csv('irisData.csv', names=['sepal_length', 'sepal_width',
                                           'petal_length', 'petal_width', 'species'])

iris.plot.hexbin(x="sepal_length", y="sepal_width", gridsize=20)
sns.jointplot(data=iris, x="sepal_length", y="sepal_width", kind="hex",
               color="k", joint_kws=dict(gridsize=20), marginal_kws=dict(bins=15, rug=True))
```



# Common visualizations

Scatter plots: further elaboration



Scatter plots can be **enriched** with additional information:  
color or different symbols incorporate **a third attribute** in the scatter plot.

What differences does this reveal?

Data objects with the same values cannot be distinguished in a scatter plot → **jitter** (adding random noise)

# Correlation analysis

Scatter plots can “visually” reveal correlations or dependencies between two attributes.

Statistical measures for correlation are a more formal approach to correlation analysis and can be carried out automatically.

We briefly sketch...

Pearson's correlation coefficient

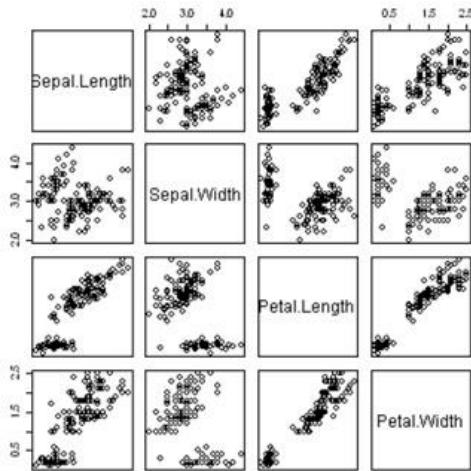
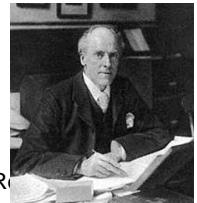
>> [video for explanation](#)

Rank correlation coefficients

>> [video for explanation](#)

Spearman's rho

Kendall's tau



```
import pandas as pd

iris = pd.read_csv('irisData.csv', names=...)

print("Show Pearson's correlation:")
print(iris.corr())
#
print()
print("Show Spearman's rho correlation:")
print(iris.corr('spearman'))
#
print()
print("Show Kendall's tau correlation:")
print(iris.corr('kendall'))
```



Show Pearson's correlation:

|              | sepal_length | sepal_width | petal_length | petal_width |
|--------------|--------------|-------------|--------------|-------------|
| sepal_length | 1.000000     | -0.109369   | 0.871754     | 0.817954    |
| sepal_width  | -0.109369    | 1.000000    | -0.420516    | -0.356544   |
| petal_length | 0.871754     | -0.420516   | 1.000000     | 0.962757    |
| petal_width  | 0.817954     | -0.356544   | 0.962757     | 1.000000    |

Show Spearman's rho correlation:

|              | sepal_length | sepal_width | petal_length | petal_width |
|--------------|--------------|-------------|--------------|-------------|
| sepal_length | 1.000000     | -0.159457   | 0.881386     | 0.834421    |
| sepal_width  | -0.159457    | 1.000000    | -0.303421    | -0.277511   |
| petal_length | 0.881386     | -0.303421   | 1.000000     | 0.936003    |
| petal_width  | 0.834421     | -0.277511   | 0.936003     | 1.000000    |

Show Kendall's tau correlation:

|              | sepal_length | sepal_width | petal_length | petal_width |
|--------------|--------------|-------------|--------------|-------------|
| sepal_length | 1.000000     | -0.072112   | 0.717624     | 0.654960    |
| sepal_width  | -0.072112    | 1.000000    | -0.182391    | -0.146988   |
| petal_length | 0.717624     | -0.182391   | 1.000000     | 0.803014    |
| petal_width  | 0.654960     | -0.146988   | 0.803014     | 1.000000    |

# Pearson's correlation coefficient

The (sample) Pearson's correlation coefficient is a measure for a linear relationship between two numerical attributes  $X$  and  $Y$  and is defined as

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)s_x s_y}$$

where  $\bar{x}$  and  $\bar{y}$  are the mean values of the attributes  $X$  and  $Y$ , respectively.  $s_x$  and  $s_y$  are the corresponding (sample) standard deviations.

The larger the absolute value of the Pearson correlation coefficient, the stronger the **linear relationship** between the two attributes.

$$-1 \leq r_{xy} \leq 1$$

Pearson's correlation assumes normal distribution (vulnerable to skewed data) and linear relationships.

Applicable to **continuous** variables.

# Rank correlation coefficient

Please read  
on your own

Pearson's correlation coefficient measures linear correlation. Even for monotone functional, but non-linear relationship Pearson's correlation coefficient will not be -1 or 1. It can even be close to zero despite a monotone functional relationship.

**Rank correlation coefficients** avoid this by ignoring the exact numerical values of the attributes and *considering only the ordering* of the values.

They intend to measure monotonous correlations between attributes, where the monotonous function does not have to be linear.

Example: Aggregate Single Sales (US)

| Pos | Artist and Title                     | Sales estimate | This year |
|-----|--------------------------------------|----------------|-----------|
| 1   | Mark Ronson - Uptown Funk            | 7,470,000      | 120,000   |
| 2   | Pharrell Williams - Happy            | 7,280,000      | 40,000    |
| 3   | Katy Perry - Dark Horse              | 6,230,000      | 20,000    |
| 4   | Taylor Swift - Shake It Off          | 5,840,000      | 60,000    |
| 5   | Meghan Trainor - All About That Bass | 5,710,000      | 20,000    |

ordinal    continuous

# Rank correlation coefficients

## Spearman's rho

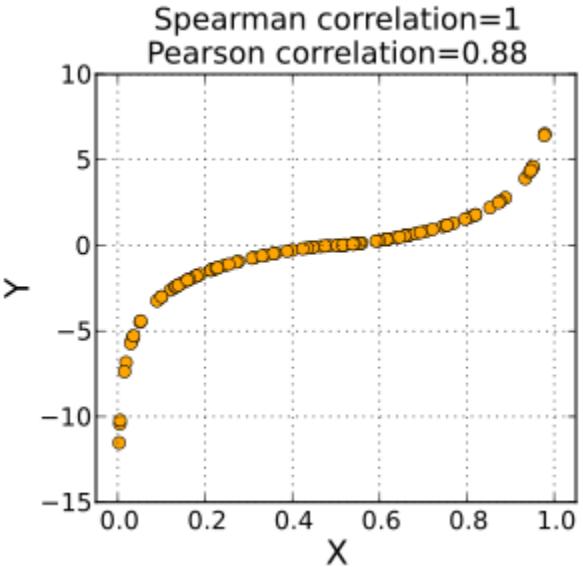
Spearman's rank correlation coefficient (**Spearman's rho**) is defined as

$$\rho = 1 - 6 \frac{\sum_{i=1}^n (r(x_i) - r(y_i))^2}{n(n^2 - 1)},$$

where we sum the deviations between  $r(x_i)$  – the rank of value  $x_i$  when we sort the list  $(x_1, \dots, x_n)$  in increasing order – and  $r(y_i)$ .

When the rankings of the  $x$ - and  $y$ -values are exactly in the same order, Spearman's rho will yield the value 1.

If they are in reverse order, we will obtain the value -1.



Spearman's rho makes no assumption on the distribution and is applicable to **continuous** and **discrete** (ordinal) variables.

It is sensitive to large deviations.

# Rank correlation coefficients

## Kendall's tau

Kendall's tau rank correlation coefficient (Kendall's tau) is defined as

$$\tau_a = \frac{C - D}{\frac{1}{2}n(n-1)}$$

where  $C$  and  $D$  denote the numbers of concordant (similar rank order) and discordant pairs with similar ranks, respectively.

$$C = |\{(i, j) | x_i < x_j \text{ and } y_i < y_j\}|$$

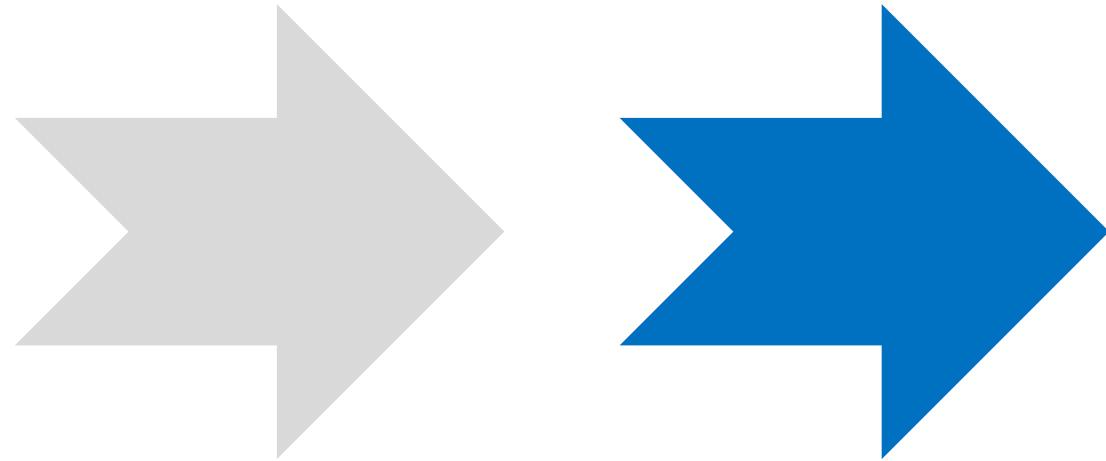
$$D = |\{(i, j) | x_i < x_j \text{ and } y_i > y_j\}|$$

Kendall's tau makes no assumption on the distribution.

Kendall's tau<sub>a</sub> is applicable to **continuous** and **discrete** (incl. ordinal) variables

Less sensitive to errors and discrepancies in the data as Spearman.





## Low-dimensional relationships

Univariate Analysis  
Bivariate Analysis

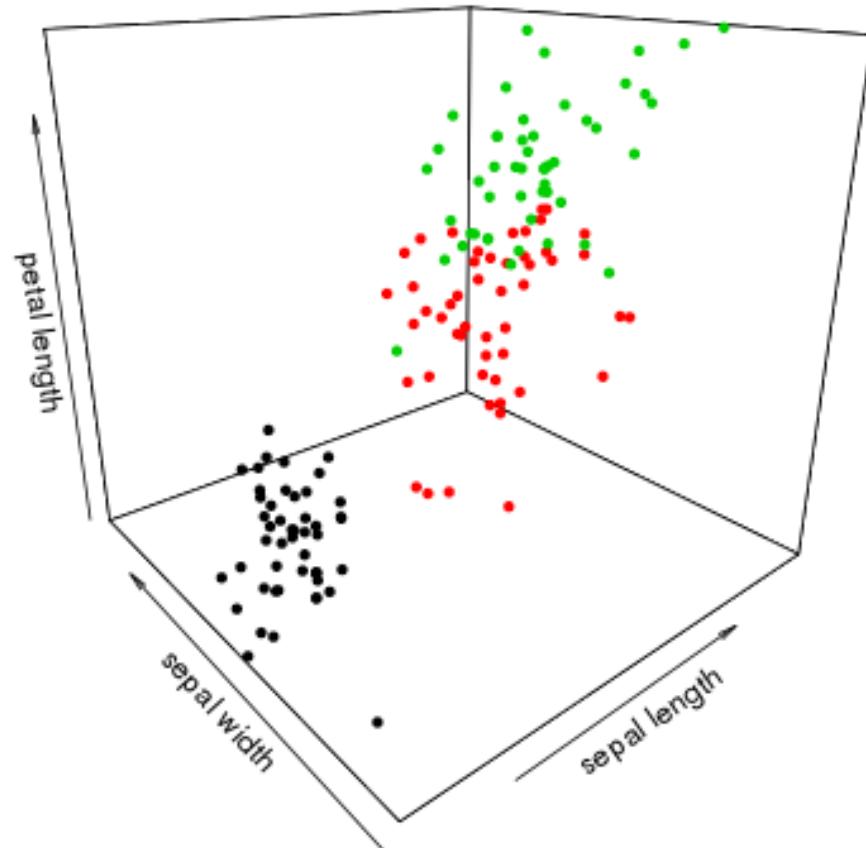
## Higher-dimensional relationships

Principal Component Analysis  
Parallel Coordinates

# 3D scatter plots



For data sets of moderate size, scatter plots can be extended to **three dimensions**.



Ref. <https://www.kaggle.com/andytran/rotating-3d-scatter-plot-for-iris-data>

# Methods for higher-dimensional data

How do we visualize more than 3 dimensions?

A display or **plot** is by definition two-dimensional, so that only two axes (attributes) can be incorporated.

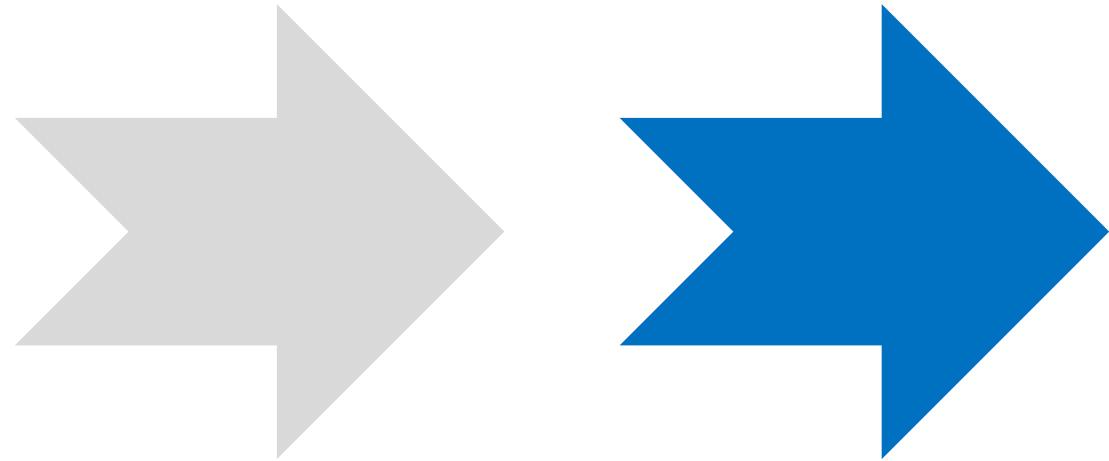


**3D techniques** can be used to incorporate three axes (attributes).

The number of possible scatter plots grows in a quadratic fashion with the number of attributes. For  $m$  attributes, there are  $\binom{m}{2} = \frac{m(m-1)}{2}$  possible scatter plots.

- For instance, 50 attributes → 1225 scatter plots.





## Low-dimensional relationships

Univariate Analysis  
Bivariate Analysis

## Higher-dimensional relationships

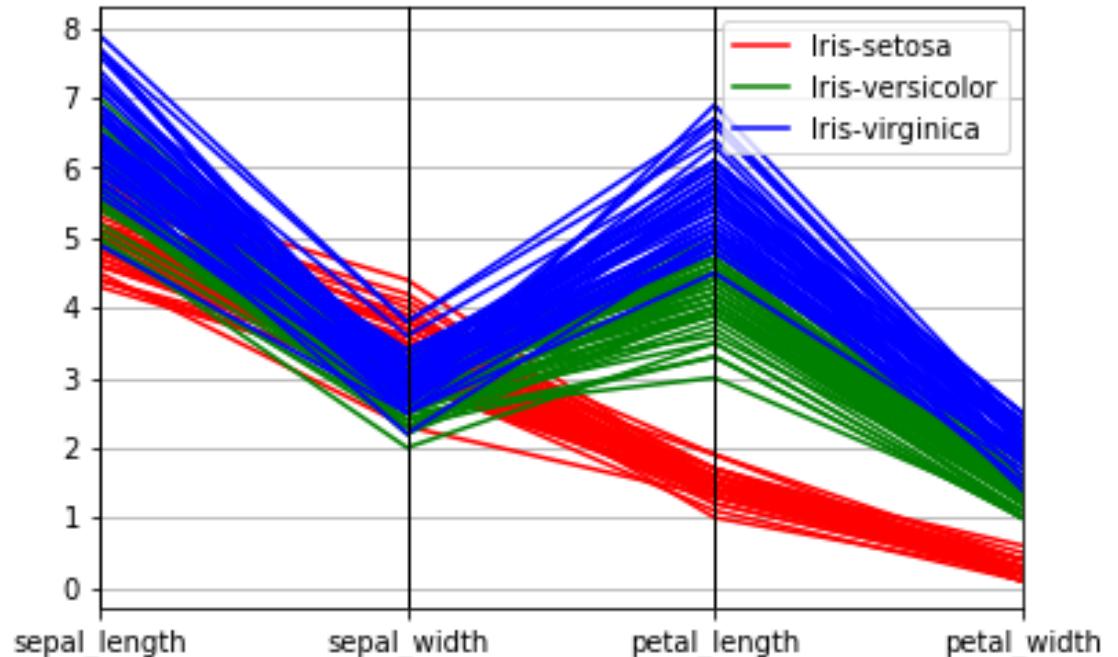
Principal Component Analysis  
[Parallel Coordinates](#)

# Parallel coordinates

Parallel coordinates draw the coordinate axes parallel to each other

There is **no limitation** for the number of axes to be displayed

For a data object, a polyline is drawn connecting the values of the data object for the attributes on the corresponding axes



```
import pandas as pd
from pandas.plotting import parallel_coordinates

iris = pd.read_csv('irisData.csv', names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])

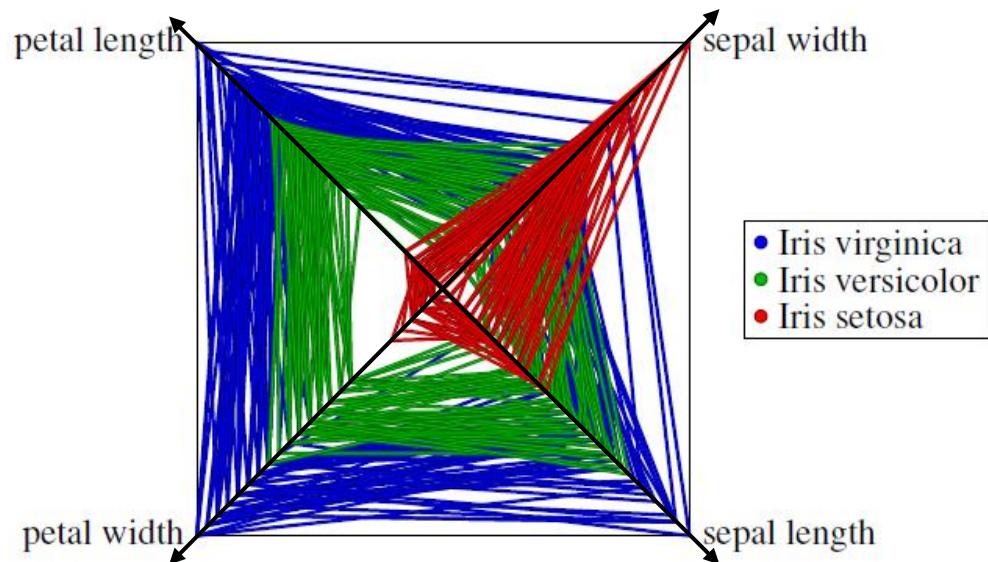
parallel_coordinates(iris, 'species', color = ['r', 'g', 'b'])

# Beispiel um spezifische Datensätze und Attribute auszuwählen
parallel_coordinates(iris[iris.species == "Iris-setosa"], 'species', cols=["sepal_length", "sepal_width", "petal_length", "petal_width"], color = ['r'])
```

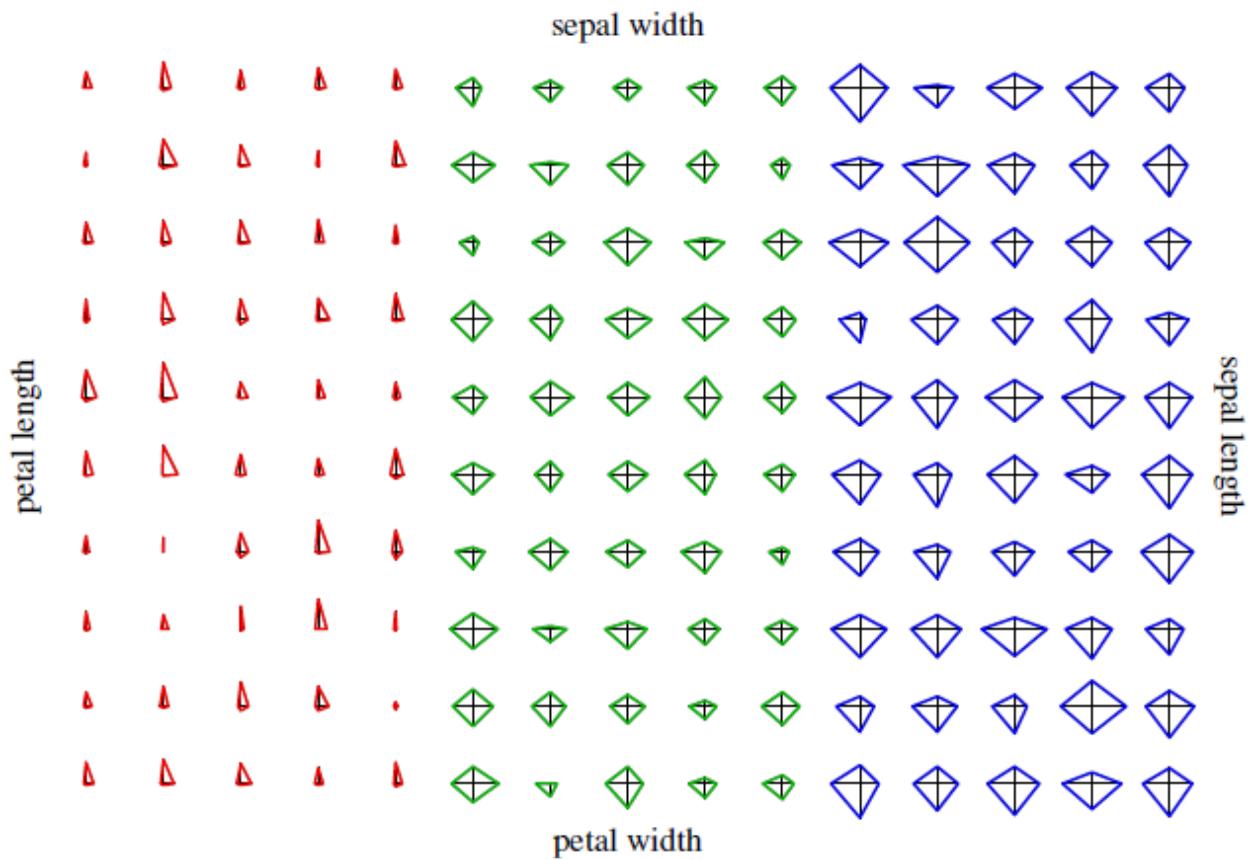


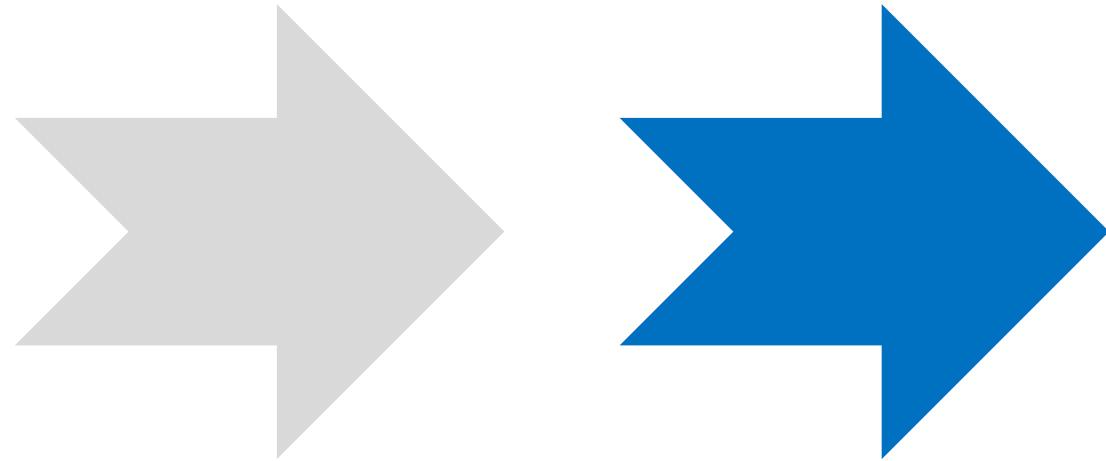
# Radar plots and star plots

**Radar plots** are based on a similar idea as parallel coordinates with the difference that the coordinate axes are drawn as **parallel lines**, but in a star-like fashion intersecting in one point.



**Star plots** are the same as radar plots where each data object is drawn **separately**.





## Low-dimensional relationships

Univariate Analysis  
Bivariate Analysis

## Higher-dimensional relationships

[Principal Component Analysis \(PCA\)](#)  
Parallel Coordinates

# Methods for higher-dimensional data

General approach for incorporating all attributes in a plot:

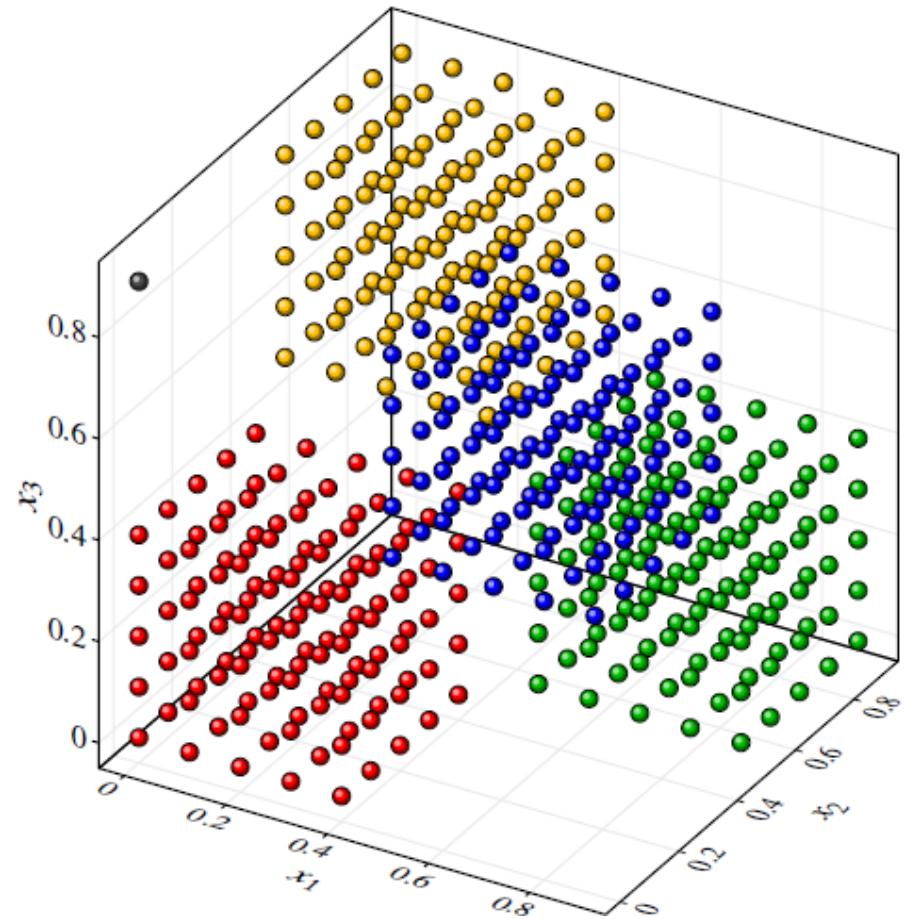
There is no unique measure for structure preservation.

Try to preserve as much of the “structure” of the high-dimensional data set when **representing (plotting) the data in two (or three) dimensions**

Define a measure that evaluates lower-dimensional representations (plots) of the data in terms of **how well a representation preserves the original “structure”** of the high-dimensional data set.

Find the representation (plot) that gives the best value for the defined measure.

PCA – Chessboard example (1/2)



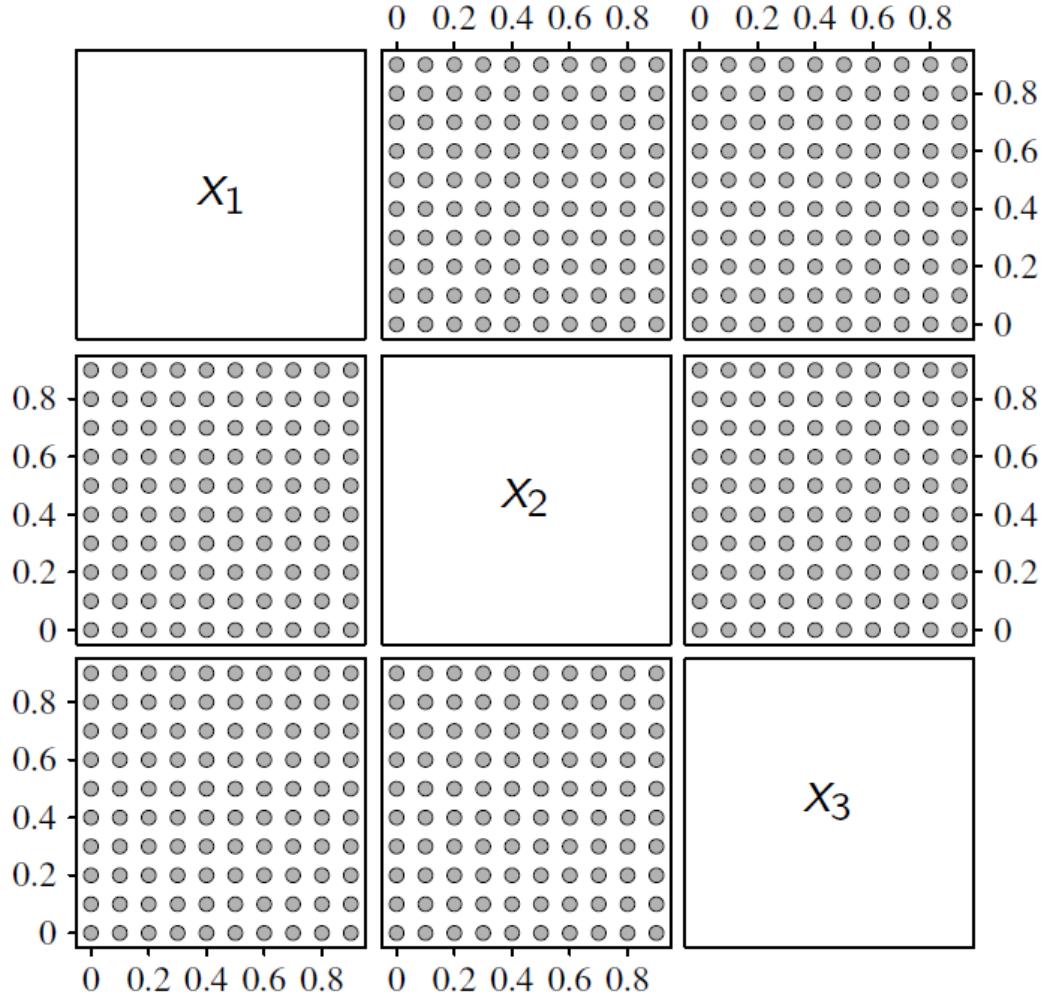
How to preserve “structure” in 2D ?

# PCA – Chessboard example (2/2)



## Scatter plots

Is data uniformly distributed over the grid?



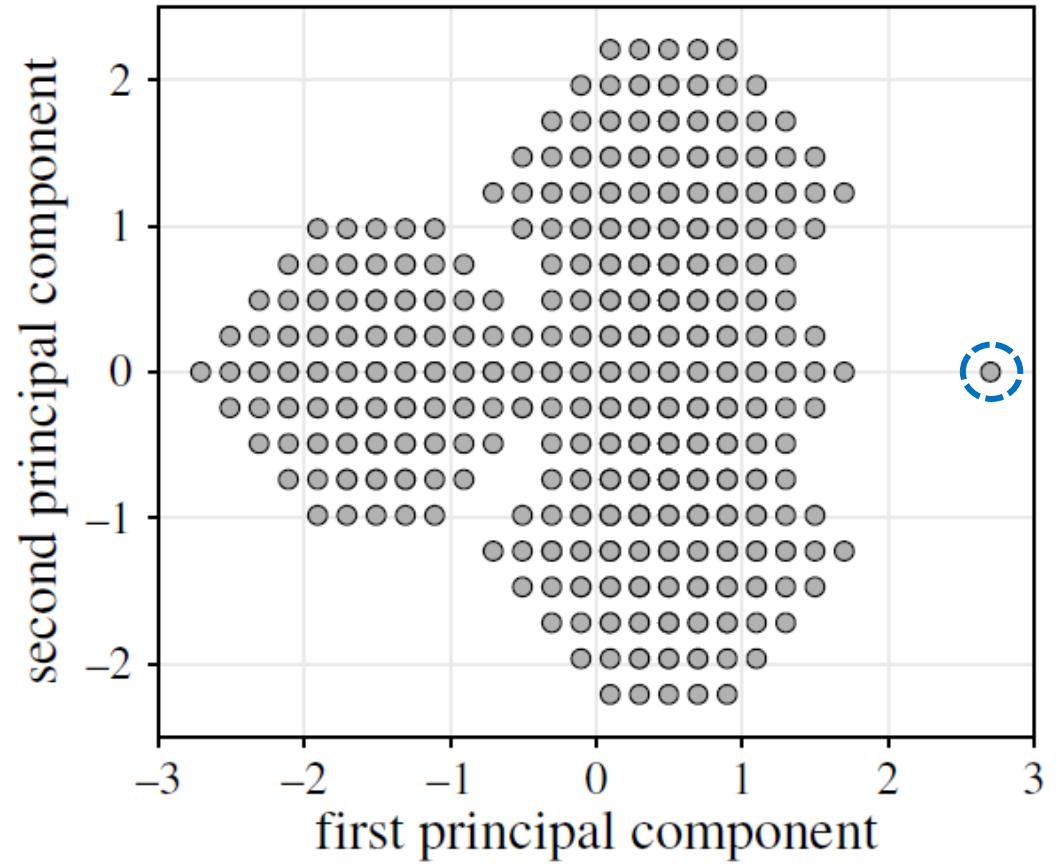
Ref. Berthold et al. (2010)

Projection to the first two principal components

Data is not uniformly distributed.

There is a pattern in the data set.

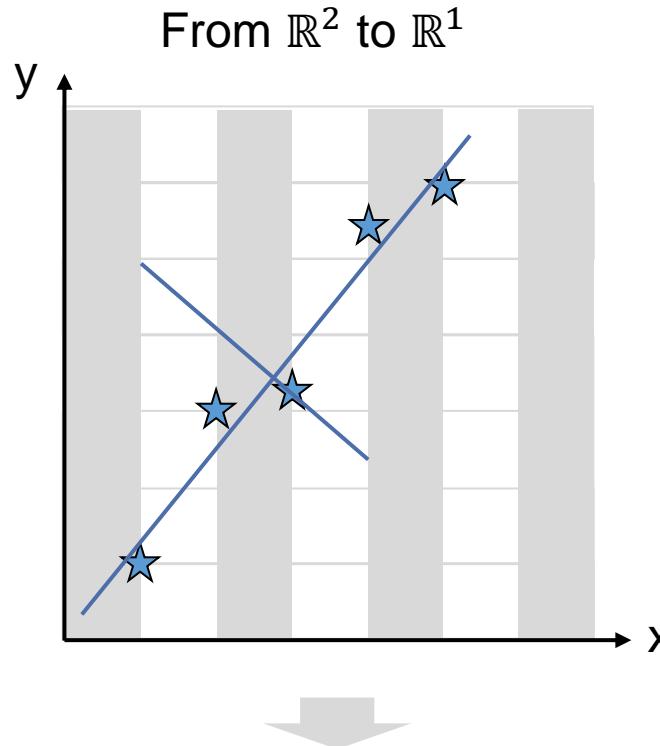
Data can be recreated from PCA.



# Principal Component Analysis (PCA)



Structure preservation through variance in data set

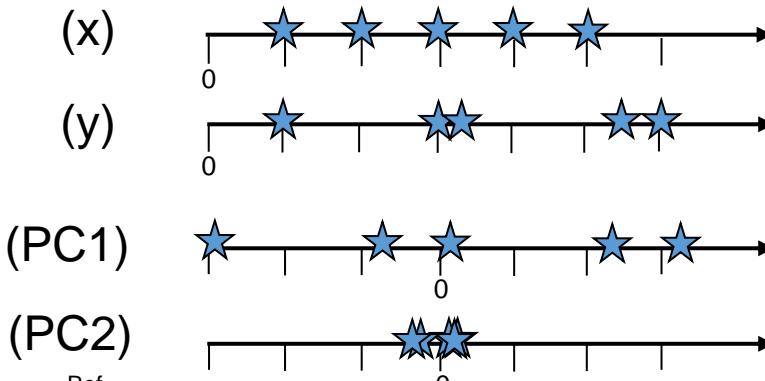


PCA compresses a large data set to capture the essence of the original data through linear transformation

PCA constructs a projection from the high-dimensional space to a lower-dimensional space (plane or hyperplane) using only the most relevant dimensions

PCA uses the variance in the data set as the structure preservation criterion.

PCA preserves as much of the original variance of the data when projected to a lower-dimensional space



Assumption: Large variances describe interesting dynamics, smaller noise.

(Sample) variance for a numerical attribute:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{n \sum_{i=1}^n x_i^2 - \left( \sum_{i=1}^n x_i \right)^2}{n(n-1)}$$

# Principal Component Analysis

Procedure: Objective

The data points are first **centered around the origin** by subtracting the mean values

Objective:

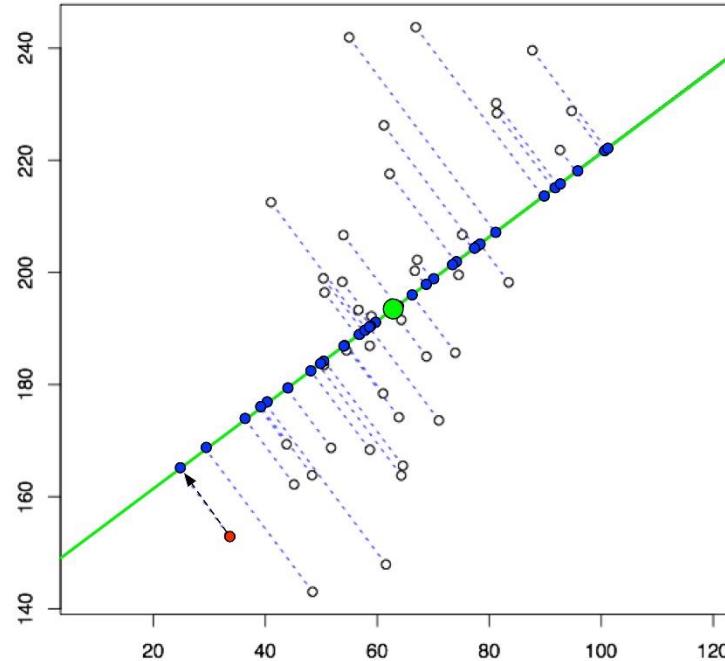
find a projection in the form of a linear mapping given by  $y = M(x - \bar{x})$ , where  $M$  is a  $q \times m$  matrix such that the **variance** of the projected data  $y_i = M(x_i - \bar{x})$  is **maximized**

( $2 \times m$  for projections to a plane)

PCA uses the **covariance matrix** which holds information on spread (variance) and orientation (covariance)

$$\Sigma = \begin{bmatrix} \sigma(x, x) & \sigma(x, y) \\ \sigma(y, x) & \sigma(y, y) \end{bmatrix}$$

*Projecting 2 dimensions on 1*



See excursus for in-depth information

# Principal Component Analysis

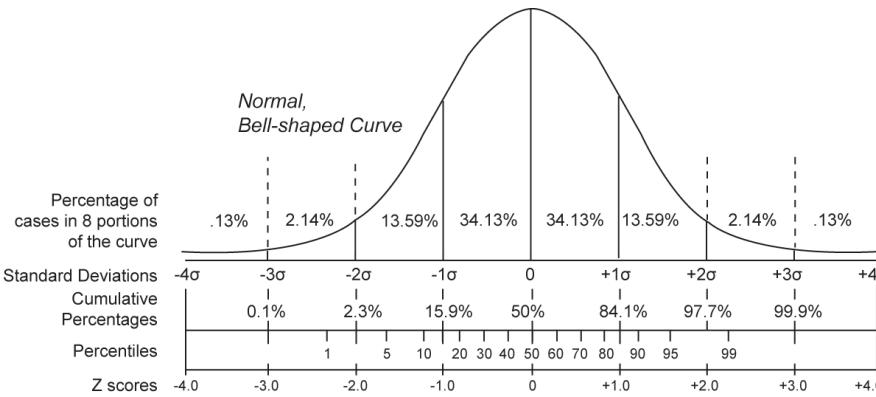
## Procedure: Problem

Problem:

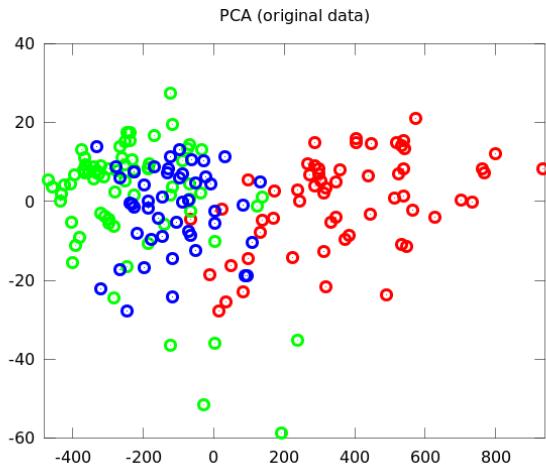
Without restriction for the matrix  $M$ , the entries in  $M$  can be chosen arbitrary large so that the data are not only projected, but also **scaled**, leading to an arbitrary large variance of the projected data.

We introduce **constraints** such that the matrix  $M$  is only a projection:

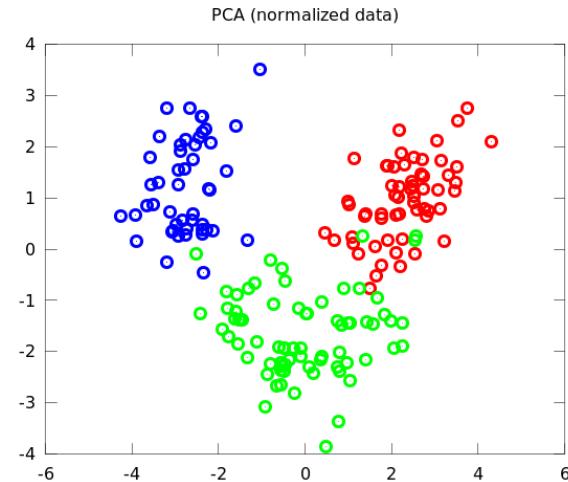
The row  $v_i$  of the matrix  $M = (v_1, \dots, v_q)$  must be **normalized**, i.e.,  $\|v_i\| = 1$ .



Ref. e.g., Kristensen & Terje (2016, p. 81 ff.)



Usually, the data should be **zero-score standardized** ( $x \rightarrow \frac{x - \hat{\mu}_x}{\hat{\sigma}_x}$ ) to ensure that all attributes contribute equally to the overall variance (with  $\hat{\mu}_x$  being the mean value and  $\hat{\sigma}_x$  the sample standard deviation of attribute  $X$ , z-score: numeric distance of  $x$  in standard deviations from mean)



# Principal Component Analysis

Choosing principal components

Solution of the constraint optimization problem:

The projection matrix  $M$  is given by  $M = (v_1, \dots, v_q)$ ,

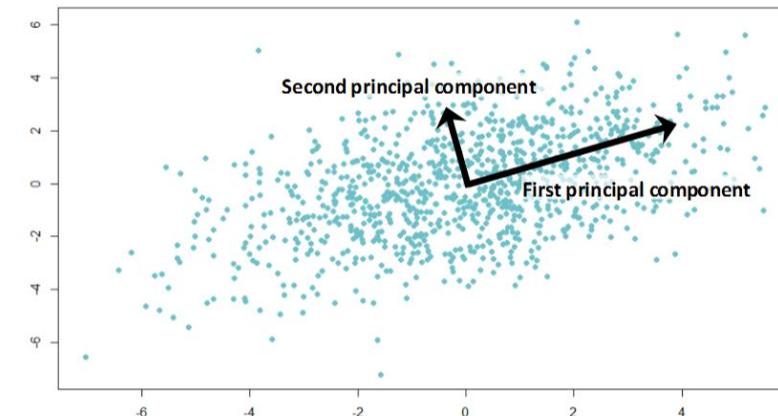
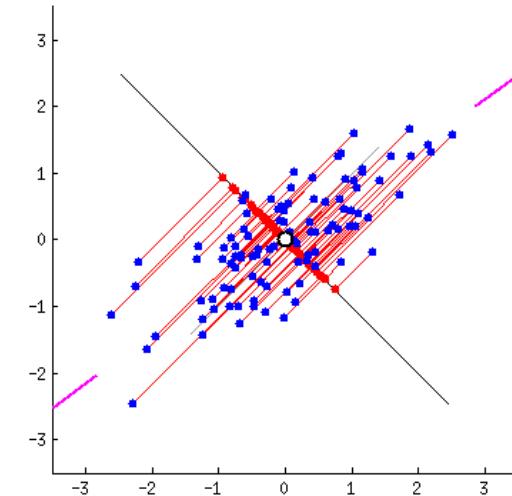
where the **principal components**  $v_1, \dots, v_q$

are the *normalized eigenvectors of the covariance matrix* of the attributes in the data set

$$\sigma_{jk} = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)^T$$

for the  $q$  **largest eigenvalues**  $\lambda_1 \geq \dots \geq \lambda_q$ .

$\lambda$  is called an eigenvalue of a matrix  $A$ , if there is a non-zero vector  $v$  such that  $A\mathbf{v} = \lambda\mathbf{v}$  holds. The vector  $v$  is called eigenvector (direction of the data) to the eigenvalue  $\lambda$  (magnitude of its spread).



# Principal Component Analysis

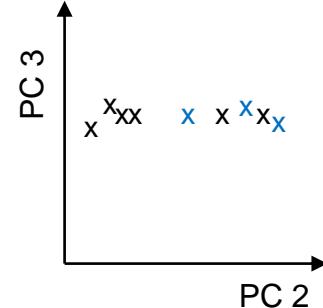
Dimension reduction

Let  $\lambda_1 \geq \dots \geq \lambda_m$  be the eigenvalues of the covariance matrix.

When we project the data to the first  $q$  principal components  $v_1, \dots, v_q$  corresponding to the eigenvalues  $\lambda_1, \dots, \lambda_q$ , this projection will preserve a fraction of the variance of the original data.

$$\frac{\lambda_1 + \dots + \lambda_q}{\lambda_1 + \dots + \lambda_m}$$

Only principal components which explain little variance in the data, like...



Iris data set:

|                        | PC1  | PC2   | PC3    | PC4     |
|------------------------|------|-------|--------|---------|
| Proportion of variance | 0.73 | 0.229 | 0.0367 | 0.00518 |
| Cum. proportion        | 0.73 | 0.958 | 0.9948 | 1.00000 |

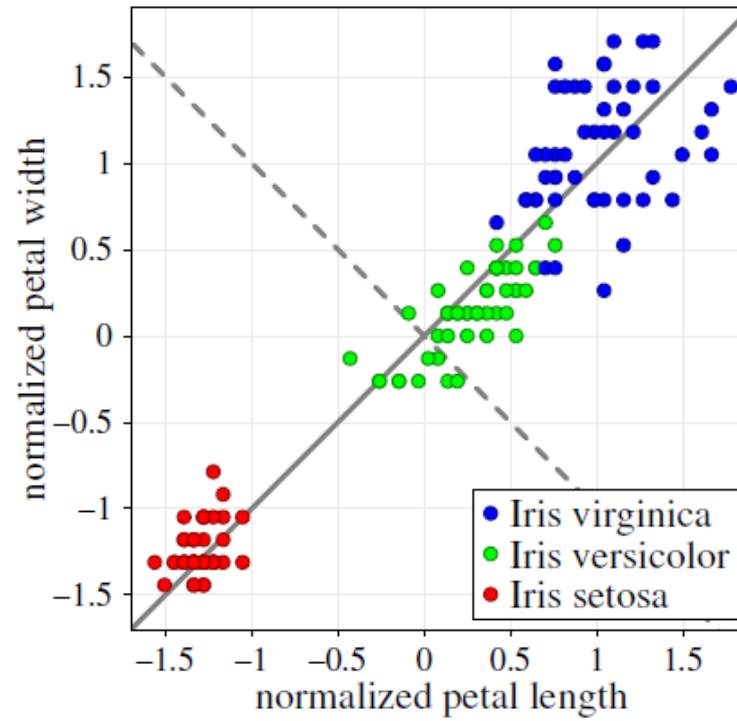
Ref.

# PCA – Iris data set example (1/2)



PCA applied to the **Iris data set** restricted to the (normalized) petal length and width

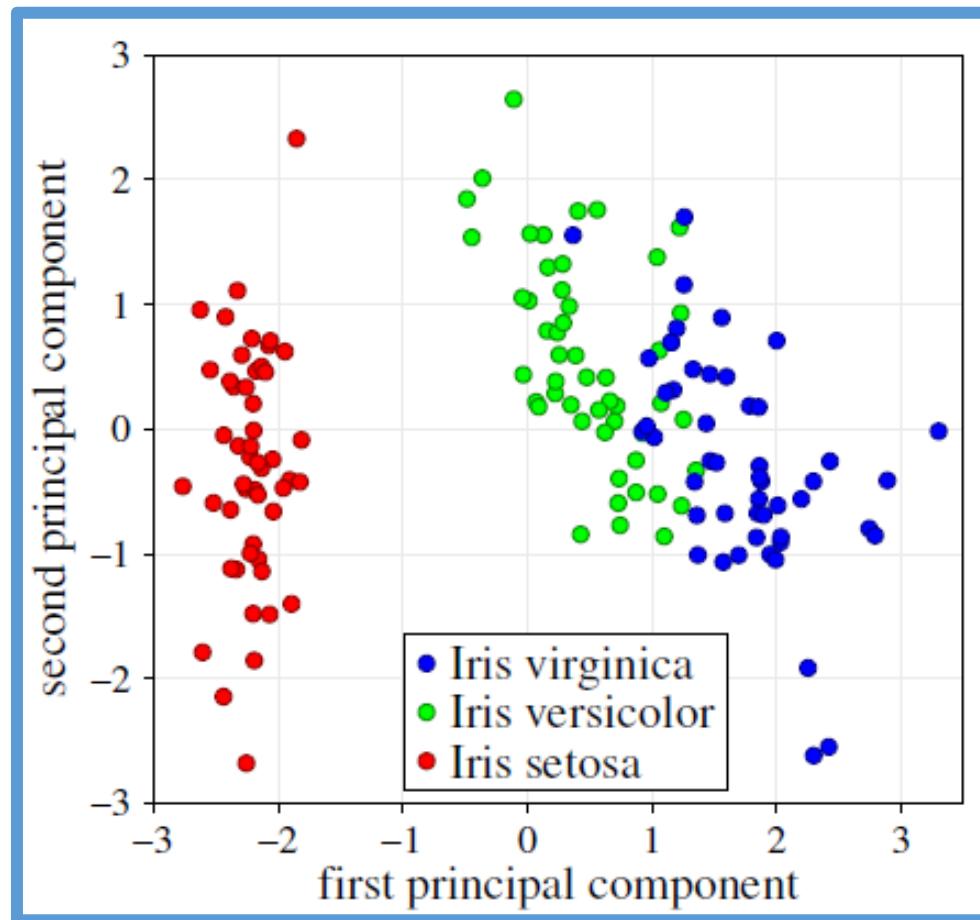
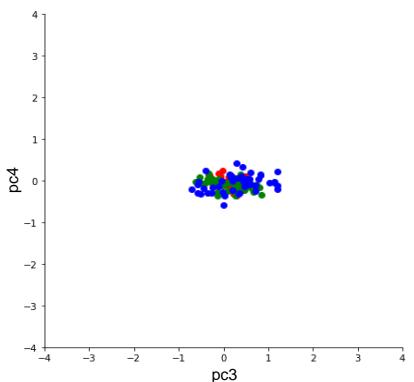
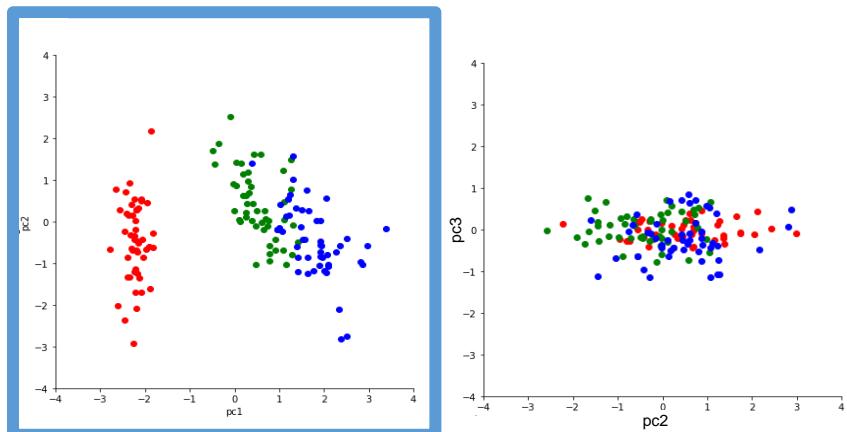
The principal components are always *orthogonal*



# PCA – Iris data set example (2/2)



Projection to the first two principal components of PCA taking all four numerical attributes into account

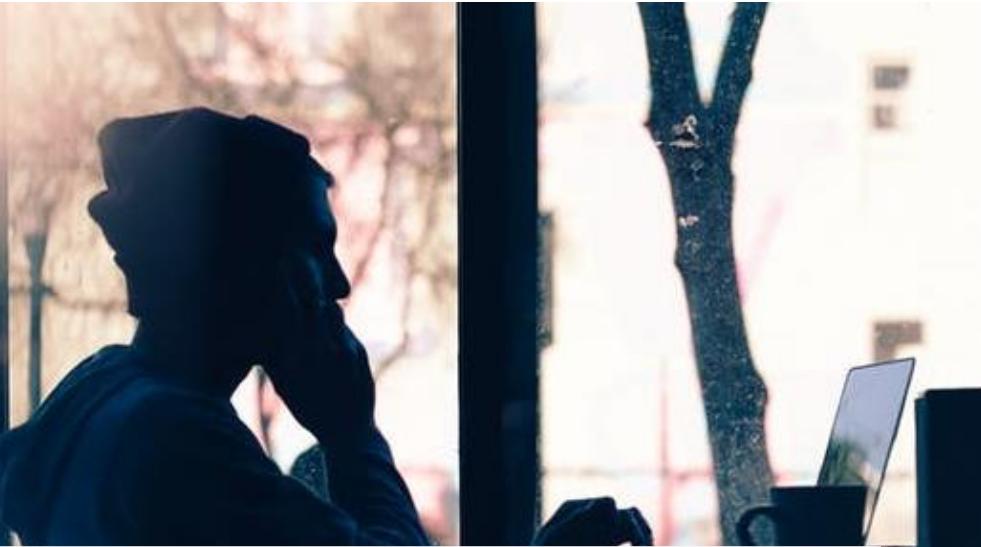


Original data is **reconstructable** from the principal components

Ref.

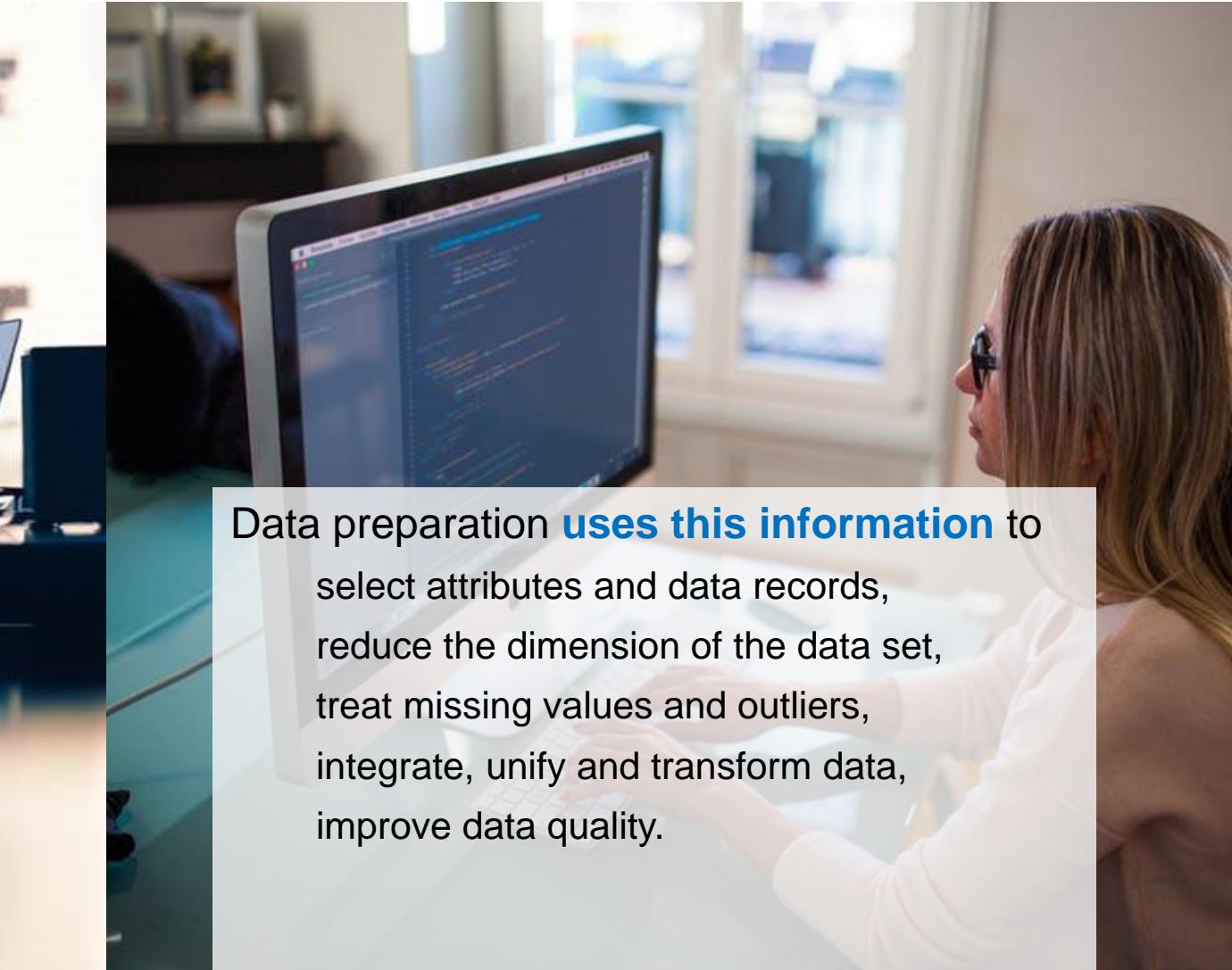
# Next Lesson

## Data understanding vs. Data preparation



Data understanding **provides general information** about the data like

- the existence and the character of missing values,
- outliers,
- the character of attributes,
- dependencies and correlations between attributes.



Data preparation **uses this information** to select attributes and data records, reduce the dimension of the data set, treat missing values and outliers, integrate, unify and transform data, improve data quality.

## Fragen?

- ✓ Data visualization, correlation analysis  
(Data understanding II)
- ✓ Low-dimensional relationships
  - ✓ Univariate Analysis
  - ✓ Bivariate Analysis
- ✓ Higher-dimensional relationships
  - ✓ Principal Component Analysis
  - ✓ Parallel Coordinates

# Todos for next Week



- Think about who you want to form a project group with (4 people per group)



# Recommended reading

Berthold et al. Chapter 4

Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann, 2011



# Business Intelligence

## 08 Data Preparation

Prof. Dr. Bastian Amberg  
(summer term 2024)

5.6.2024

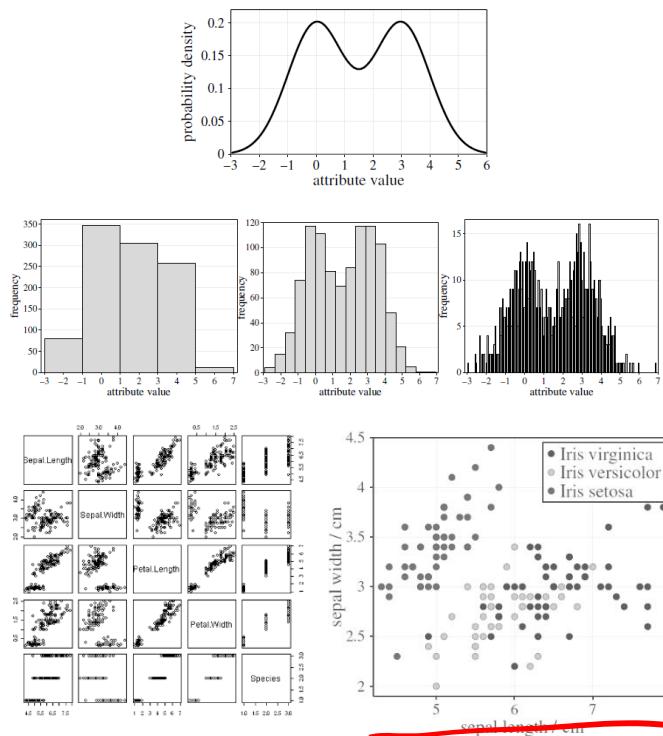
# Schedule

|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study |                          |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|------------|--------------------------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |            |                          |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |            |                          |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  |            |            |                          |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining Introduction  |            |                          |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |            | Python-Analytics Chap. 1 |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization I            |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam |            |                          |  |  |
|           | W7                | 29.5. | Data Visualization II                               |                                   | 31.5. |   |            |                          |  |  |
|           | W8                | 5.6.  | Data Preparation                                    |                                   | 7.6.  | Predictive Modeling I (10:00 -12:00)  |            | BI-Project Start         |  |  |
|           | W9                | 12.6. | Predictive Modeling II, Fitting a Model I           |                                   | 14.6. | Python-Analytics-Online Exercise  |            |                          |  |  |
|           | W10               | 19.6. | Guest Lecture Dr. Ionescu                           |                                   | 21.6. | Fitting a Model II  |            |                          |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |            |                          |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I                                  |            |                          |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |            |                          |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |            |                          |  |  |
|           | Ref.              |       |   |                                   |       | Klausur 1.Termin, 31.7.'24<br>Klausur 2.Termin, 2.10.'24                                      |            | Projektbericht           |  |  |

# Last Lesson

data visualization

- ✓ Low-dimensional relationships
  - ✓ Univariate Analysis
  - ✓ Bivariate Analysis



Free University Berlin

$\text{PCA}_1 = \lambda_1 \cdot A_1 + \lambda_2 \cdot A_2 + \lambda_3 \cdot A_3$

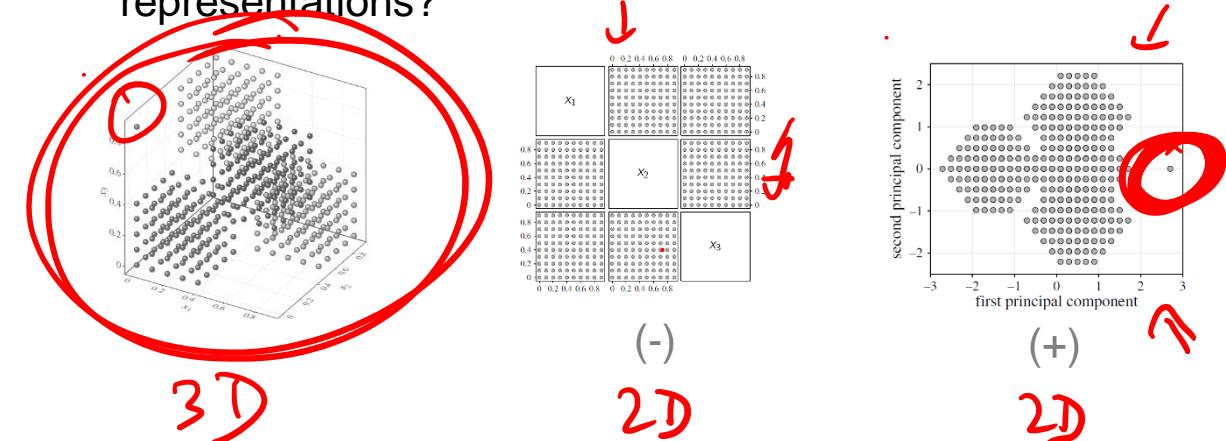
$\text{PCA}_2 = 0,57 A_1 + 0,2 A_2 + 0,1 A_3$

- ✓ Higher-dimensional relationships

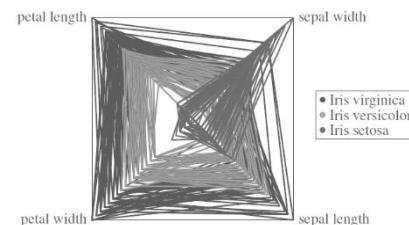
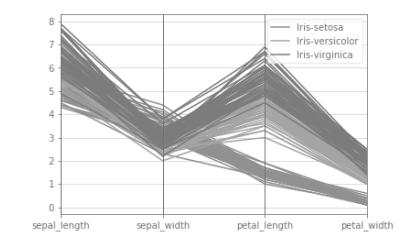
- ✓ Principal Component Analysis

Clusteranalyse

How to preserve original “structure” in lower dimensional representations?



- ✓ Parallel Coordinates



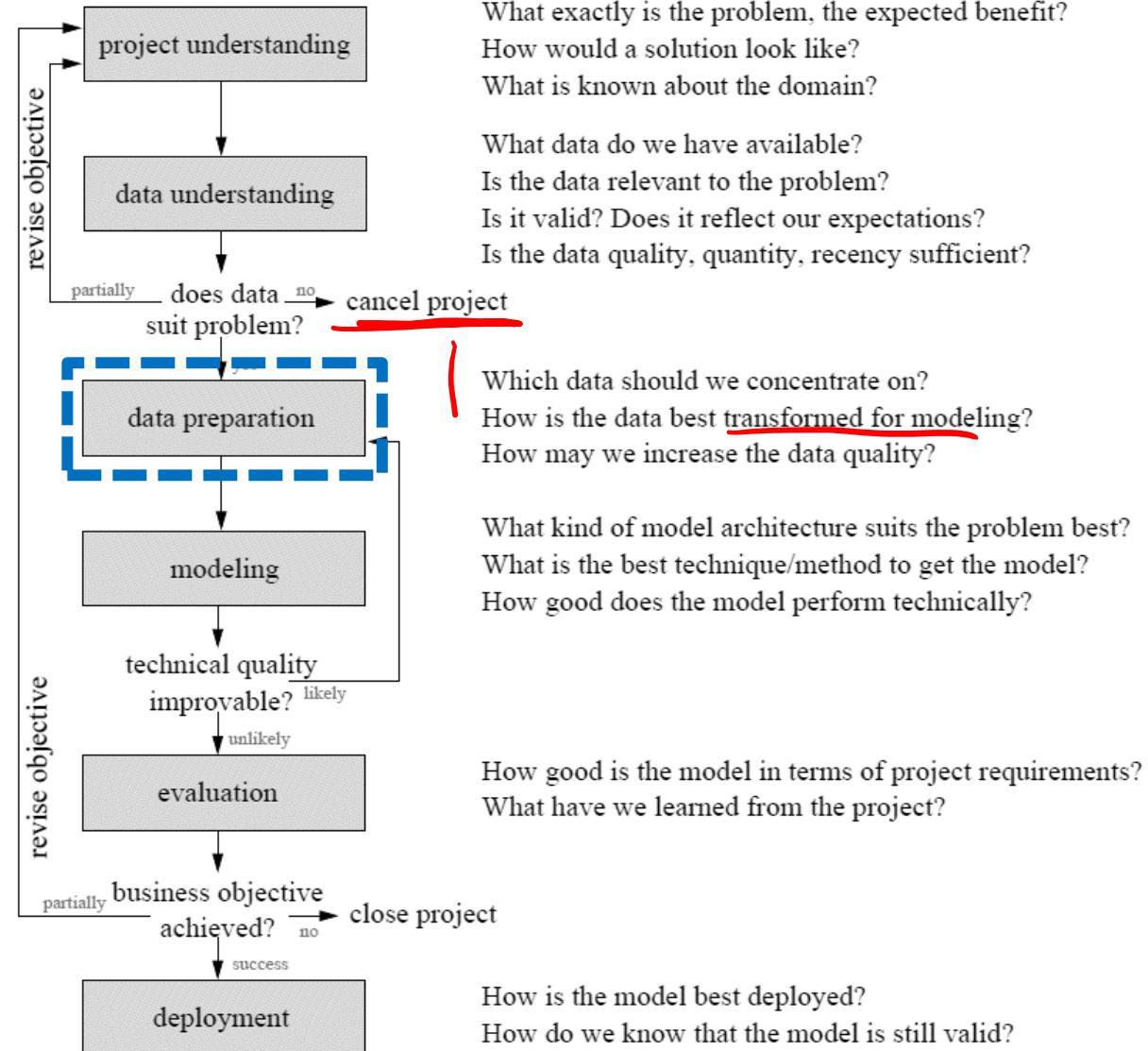
Ref.

## Cross Industry Standard Process for Data Mining

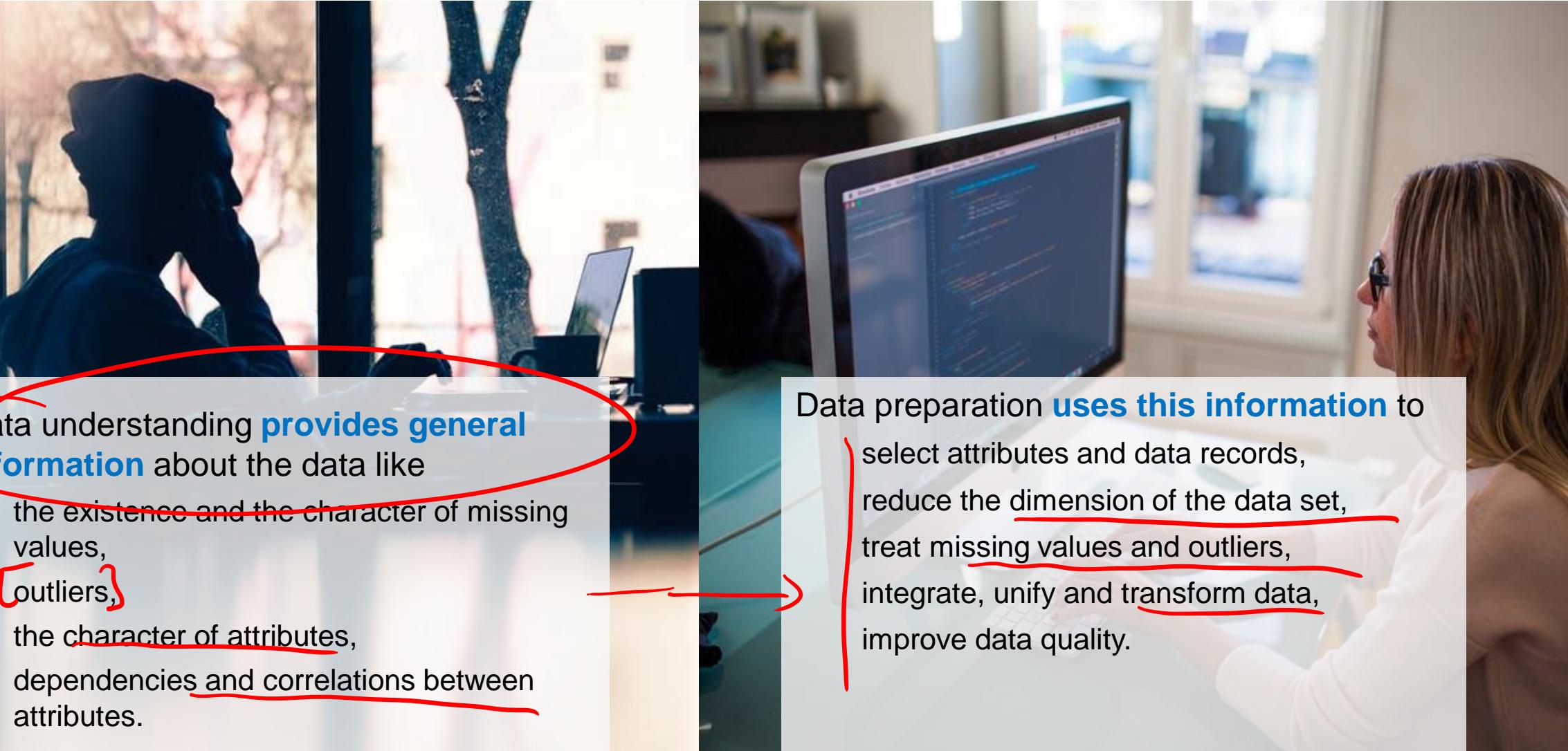
Iteration as a rule

Process of data exploration

Implementation of the KDD Process



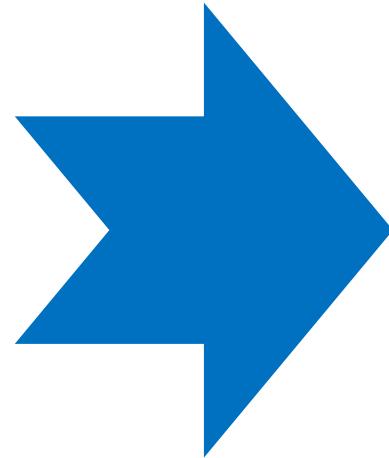
# Data understanding vs. Data preparation



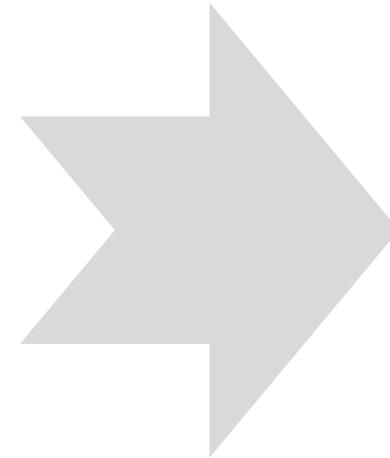
# Agenda



## Data Preparation



(1) Data selection



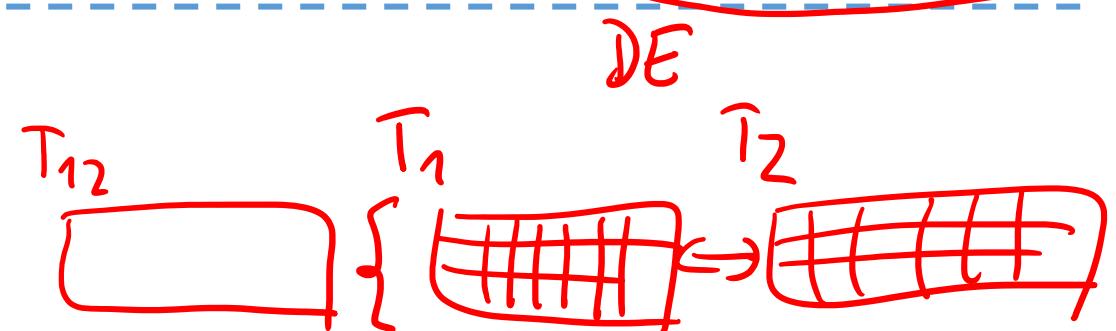
(2) Data cleaning



(3) Data transformation

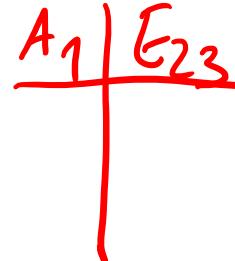
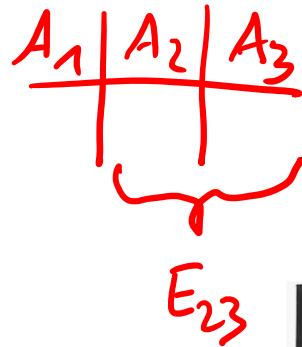


(4) Data integration



# Feature extraction

Constructing (new) features from the given attributes



Freie Universität



Berlin

## Example:

We are interested in finding the best workers in a company.

There are attributes available like

- the tasks a worker has finished within each month, # tasks
- the number of hours he/she has worked each month, # hours
- the number of hours that are normally needed to finish each task.

In principle, these attributes contain information about the efficiency of the worker.

It might be more useful to **define a new attribute** "**efficiency**", which is .... For example?

The proportion of hours spent to finish a task  
to hours normally needed to finish a task.

$$\frac{\text{\# tasks}}{\text{\# hours}}$$



# Dimensionality reduction for feature extraction



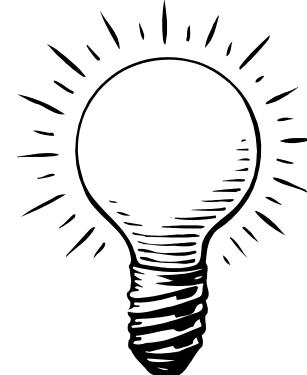
Dimensionality reduction techniques like PCA can also be considered as feature extraction methods.

But such automatic feature extraction methods usually lead to features that **can no longer be interpreted** in a meaningful way.

*How to understand a feature that is a linear combination of 10 attributes?*

Therefore, in most cases, either knowledge-based, problem-dependent feature extraction methods or feature selection techniques are preferred.

$$\lambda_1 A_1 + \lambda_2 A_2 + \lambda_3 A_3$$



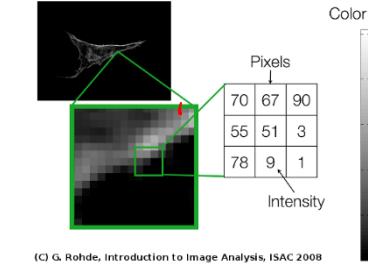
# Feature extraction and selection

NEU

Selecting features and creating subsets

**Feature extraction** is especially relevant for complex data types:

- Text data analysis – frequency of keywords, ...
- Time series and image data analysis – fourier or wavelet coefficients, ...
- Graph data analysis – number of vertices and edges



**Feature selection** refers to techniques that **choose a subset of the features** (attributes) that is as small as possible and sufficient for data analysis.

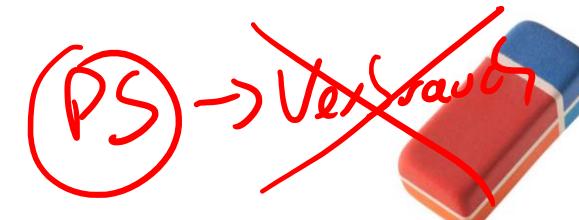
Remove (more or less) irrelevant features

| For removing *irrelevant features*, a performance measure is needed that indicates how well a feature or subset of features performs w.r.t. the considered data analysis task.

Remove redundant features

| For removing *redundant features*, either a performance measure for subsets of features or a correlation measure is needed.

$A_1, A_2, A_3, A_4$



Difference between irrelevant and redundant?

# Feature selection techniques (1/2)

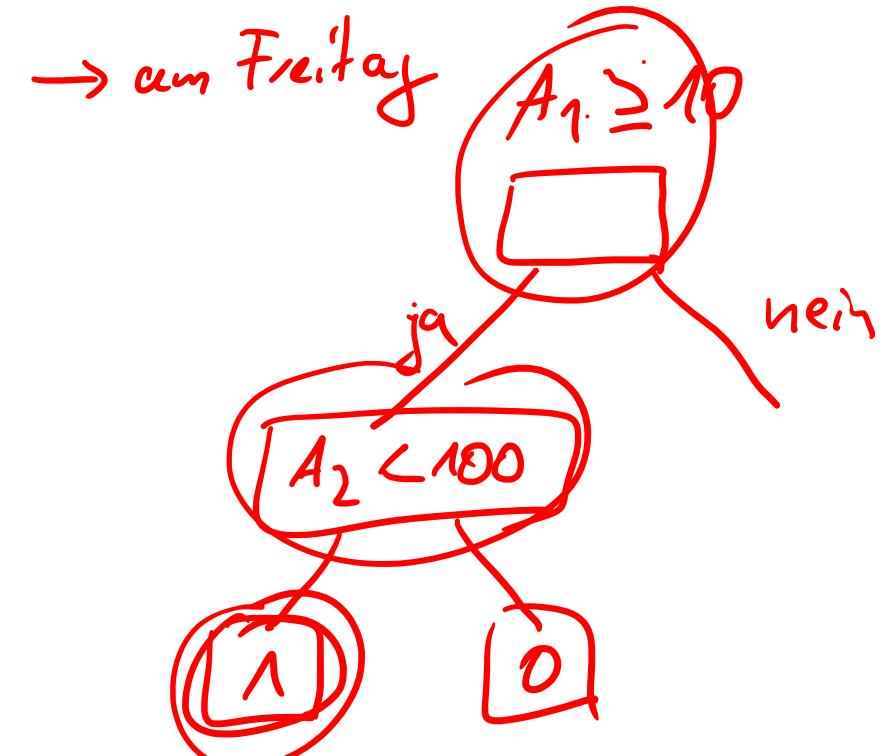
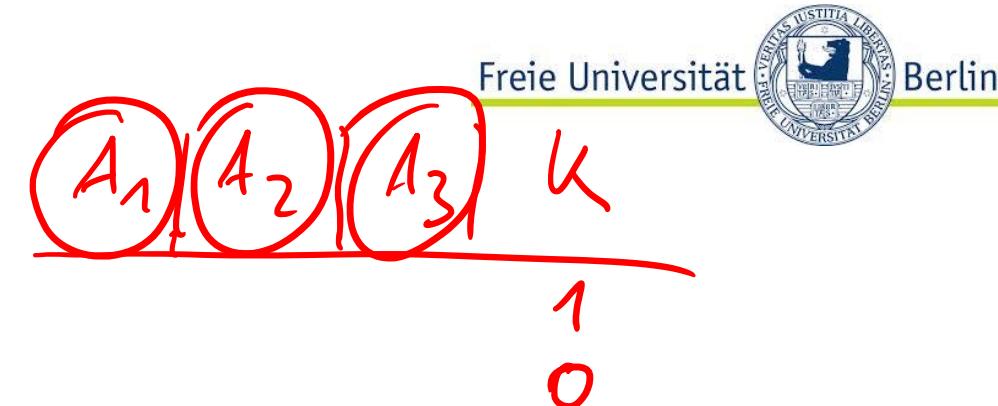
Selecting features with the highest performance

For **classification tasks** with a target attribute, typical performance measures are

- **$\chi^2$  test** for independence. It measures the deviation of the sample marginal distributions from the marginal distribution one would obtain assuming the considered attribute and the target variable are independent.
- **Information gain.** Based on entropy reduction (we'll dive deeper into this later with **Decision Trees**)

**Wrapper methods** can be applied when the model class (e.g. decision trees) is already specified.

- Train the model with different subsets of features and choose the features that lead to the model with the best performance.



# Feature selection techniques (2/2)

Four ways to select features

- Selecting the top-ranked features (single features)

Choose the features with **the best evaluation**.

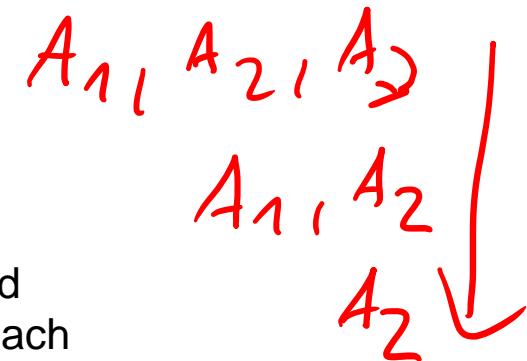


- Forward selection

Start with an empty set of features.  
**Add features one by one.** Consider which one yields the best improvement.

- Selecting the top-ranked subset

Choose **the subset of features** with the best performance. This requires exhaustive search and is impossible for larger numbers of features.



- Backward elimination

Start with the full set of features and **remove features one by one**. In each step, remove the feature that yields to the least decrease in performance.

# Feature selection

Example

Consider the following classification task that consists of 9 repetitions of the four data records in the first table and the four records in the second table.

|     | A | B | C | D | target |
|-----|---|---|---|---|--------|
| 9 × | + | + | + | - | no     |
|     | + | - | + | - | yes    |
|     | - | + | + | - | yes    |
|     | - | - | - | + | no     |
|     | + | + | + | - | no     |
|     | + | - | - | + | yes    |
|     | - | + | - | + | yes    |
|     | - | - | - | - | no     |
|     | + | - | - | - | no     |

|     | A | B | C | D | target |
|-----|---|---|---|---|--------|
| 1 × | + | + | + | - | no     |
|     | + | - | - | + | yes    |
|     | - | + | - | - | yes    |
|     | - | - | + | + | no     |

Performance of the single attributes

| A | target |     | B | target |     | C | target |     | D | target |     |
|---|--------|-----|---|--------|-----|---|--------|-----|---|--------|-----|
|   | no     | yes |
| + | 10     | 10  | + | 10     | 10  | + | 11     | 20  | + | 10     | 0   |
| - | 10     | 10  | - | 10     | 10  | - | 9      | 0   | - | 10     | 20  |

40 Datensätze  
A B C D

A + B

Which set of attributes should be selected for classification?

(Diese Folie ist nach der Vorlesung mit Erläuterungen verfügbar)

A greedy strategy selecting those attributes with **the best performance** would choose attributes *C* and *D* first.

Attributes *C* and *D* together cannot perfectly predict the target value, though.

Attributes *A* and *B* alone provide no information about the target value.

However, attributes *A* and *B* together are sufficient to perfectly predict the target value (if *A=B*: no).

**Evaluation of the performance of isolated attributes** does usually not provide proper information about their performance in combination.

# Record (Instance) selection



## Timeliness

If data have been collected over a long period, some of the **older data might not be useful** or even misleading for the data analysis task. Only the recent data should be selected.

## Representativeness

The sample in the database might not be representative for the whole population. When we have information about the distribution of the population, we can **draw a representative subsample** from our database.

## Rare events

When we are interested in predicting rare events (e.g. stock market crashes, failures of a production line), it can be helpful to incorporate this in the cost function or to **artificially increase the proportion** of these rare events in the data set.

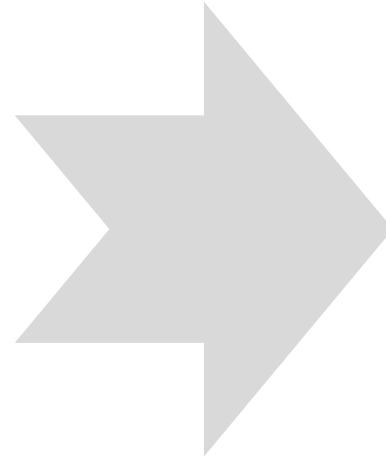
We will address these issues  
in depth when building  
predictive models and  
evaluating their performance.



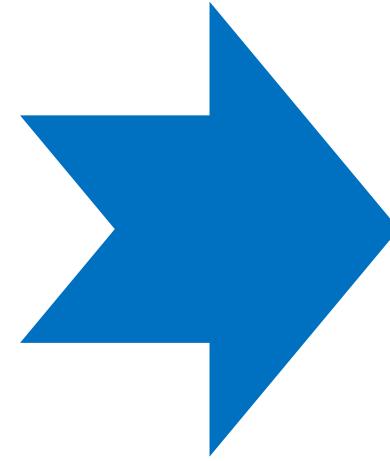
Image: [Elliott Brown \(2016\)](#) | Flickr



## Data Preparation



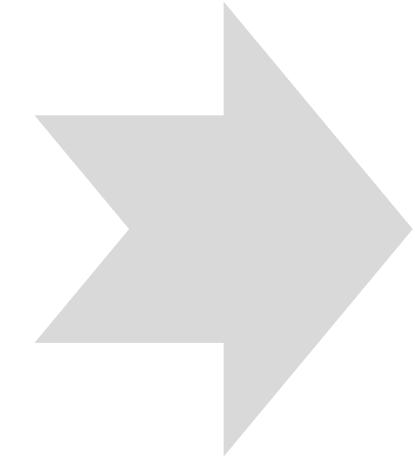
(1) Data selection



(2) Data cleaning



(3) Data transformation



(4) Data integration

# Data clean(s)ing



Data clean(s)ing or data scrubbing refers to **detecting and correcting or removing inaccurate, incorrect or incomplete data records** from a data set.

## Improve data quality

- Turn all characters into capital letters to **level case sensitivity**
- **Remove spaces** and nonprinting characters  
(\n, \t etc.)
- **Fix the format** of numbers, data and time  
(decimal point! Datetime objects or standard date format,  
i.e. YYYY-MM-DD)
- **Split fields** that carry mixed information into separate ones  
("Chocolate, 100g" → "Chocolate" and "100.0")
- Use **spell-checker** or stemming to normalize spelling
- **Replace abbreviations** by their long form (dictionary)
- Normalize the **writing of addresses** and names,  
possibly ignoring the order of title, surname,  
forename, etc. to ease their re-identification
- **Convert numerical values** into standard units,  
especially if data from different sources and  
different countries are used
- **Use dictionaries** containing all possible values of  
an attribute to assure that all values comply with  
the domain knowledge



Ew/  
DM

# Missing values



For some instances, values of single attributes might be missing.

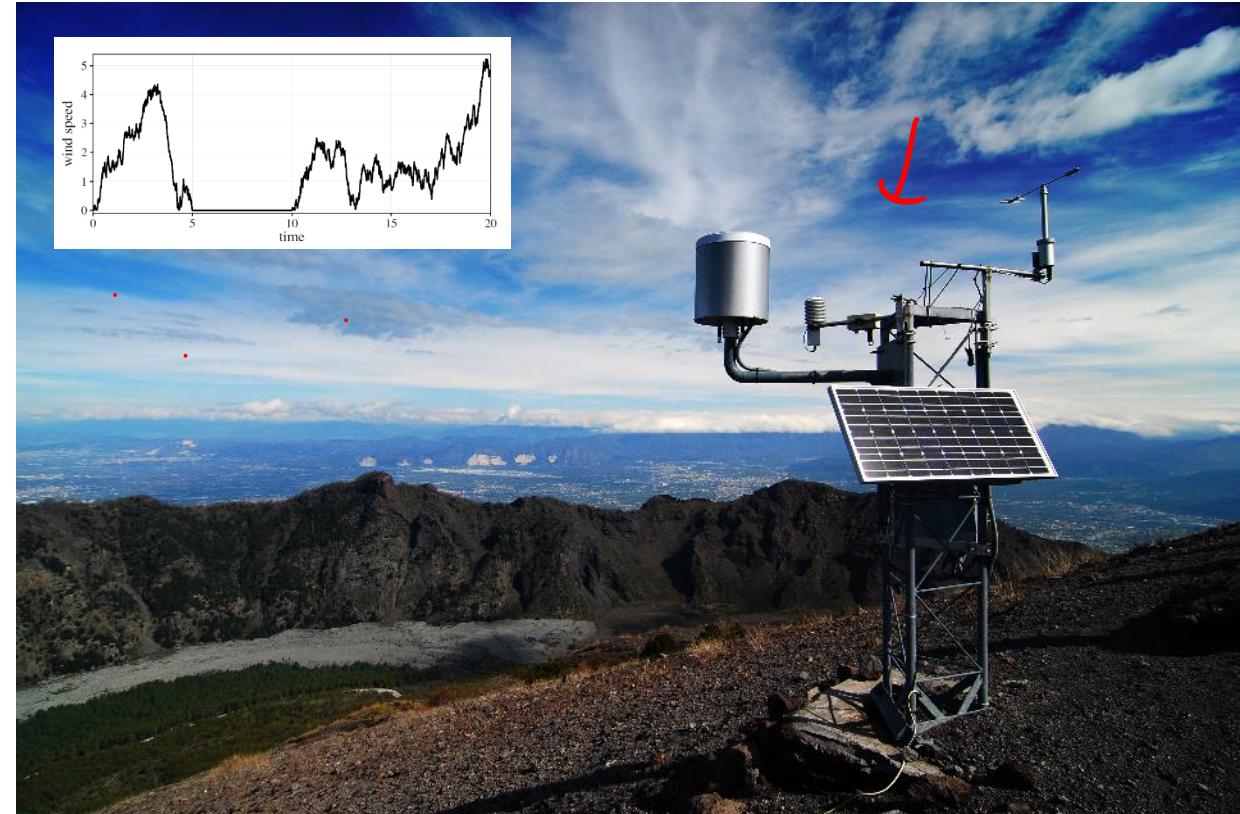
**Causes** for missing values:

Broken sensors

Refusal to answer a question  
(pregnant (yes/no) in a job interview)

Irrelevant attribute for the corresponding object  
(pregnant (yes/no) for men)

Missing value might not necessarily be indicated as missing, instead: it may have a default values (i.e., zero, 99 etc.).



# Types of missing values



Consider the attribute  $X_{obs}$ . A missing value is denoted by “?”.

$X$  is the true value of the considered attribute, i.e., we have  $X_{obs} = X$ , if  $X_{obs} \neq ?$ .

Let  $Y$  be the (multivariate)(random) variable denoting the other attributes apart from  $X$ .

## (1) Missing completely at random (MCAR)

The probability that a value for  $X$  is missing does neither depend on the true value of  $X$  nor on other variables  $\rightarrow P(X_{obs} = ?) = P(X_{obs} = ? | X, Y)$

Example:

*The maintenance staff **sometimes** forgets to change the batteries of a sensor so that the sensor at that times does not provide any measurements*

MCAR is also called Observed At Random (OAR).



Ref. Little, R. J., & Rubin, D. B. (2014). *Statistical analysis with missing data* (Vol. 333). John Wiley & Sons.

## (2) Missing at random (MAR)

The probability that a value for  $X$  is missing does not depend on the true value of  $X$   
 $\rightarrow P(X_{obs} = ? | Y) = P(X_{obs} = ? | X, Y)$

Example:

*The maintenance staff does not change the batteries of a sensor **when it rains**. Thus, the sensor does not always provide measurements when it rains.*



## (3) Nonignorable

The probability that a value for  $X$  is missing depends on the true value of  $X$ .

Example:

*A sensor for the temperature will not work **when there is frost**.*

# Exercise: Type of missing values?

Further examples



Kahoot-Fragen  
[www.kahoot.it](http://www.kahoot.it)  
(über Smartphone oder Laptop)  
PIN folgt

(Diese Folie ist nach der Vorlesung mit Lösungen verfügbar)

# Types of missing values

## Estimation of Missing Values

For MCAR and MAR, the missing values can be **estimated**

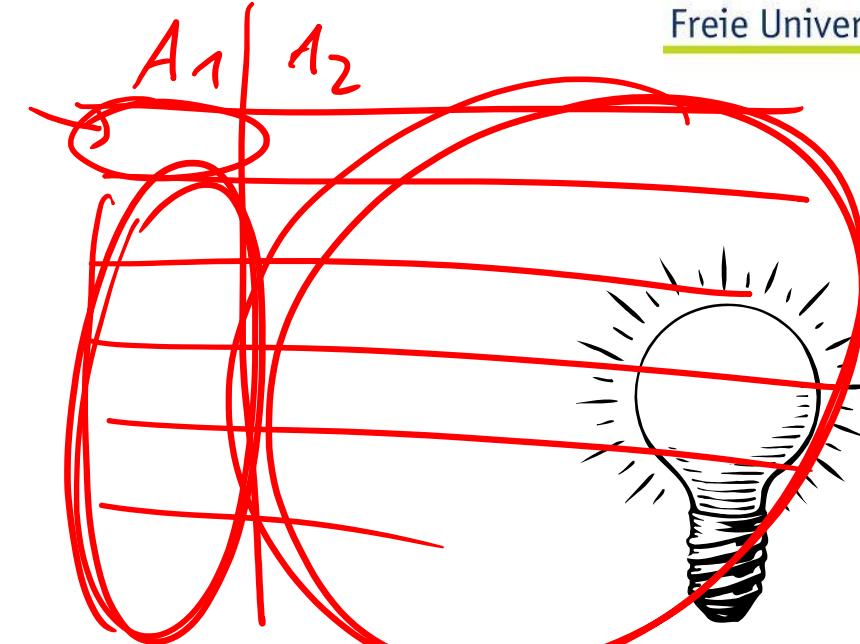
At least in principle, when the data set is large enough – based on the values of the other attributes

The cause for the missing values is ignorable

(1) For MCAR, it can be assumed that the missing values follow **the same distribution** as the observed values of  $X$

(2) For MAR, the missing values might not follow the distribution of  $X$ . But by taking the other attributes into account, it is possible to derive **reasonable imputations** for the missing values.

(3) For *nonignorable* missing values it is **impossible to provide sensible estimations** for the missing values

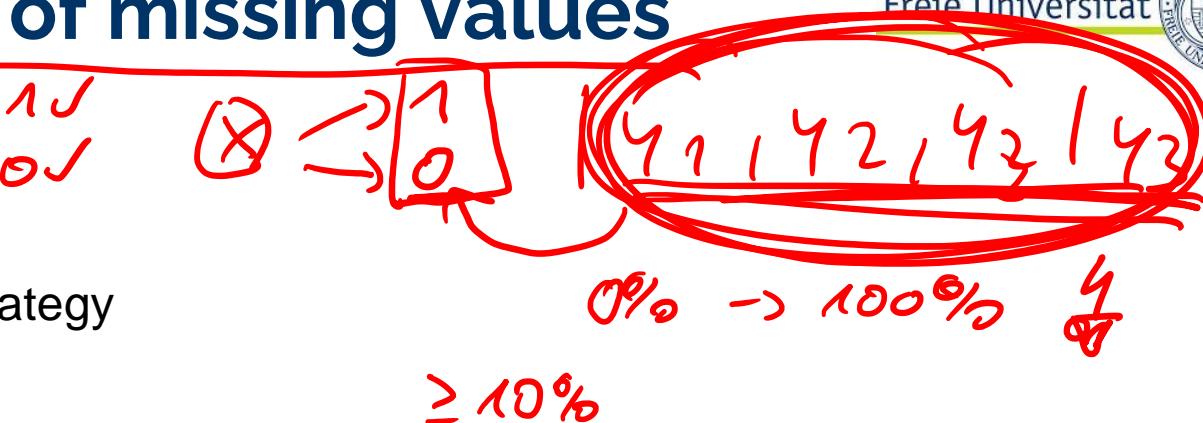


# How to determine the type of missing values



If domain knowledge does not help which kind of missing values can be expected, the following strategy can be applied

- Turn the considered attribute  $X$  into a **binary attribute**, replacing all measured values by “yes” and all missing values by “no”
- **Build a classifier** with the now binary attribute  $X$  as the target attribute, and use all other attributes for the prediction of the class values “yes” and “no”
- **Determine the misclassification rate**, which is the proportion of data objects that are not assigned to the correct class by the classifier.



MAR

- For MCAR, the other attributes should not provide any information, whether  $X$  has a missing value or not. Therefore, the misclassification rate should not differ significantly from **pure guessing**
- If there are 10% missing values for the attribute  $X$ , the misclassification rate of the classifier should not be much smaller than 10%.
- If the misclassification rate is significantly better than pure guessing, this is an indicator that there is a correlation between missing values for  $X$  and the values of the other attributes. The missing values are not MCAR.
- MAR and nonignorable cannot be distinguished in this way.

# How to handle missing values

## Ignorance/Deletion

If only a few records have missing values, and it can be assumed that the values are MCAR, these records can be deleted for the following data analysis step.



## Imputation

The missing values may be replaced by some estimate.

### **Single Imputation**

Mean, median or mode of the attribute (MCAR required)

### **Multiple Imputation**

By an estimation based on the other attribute,  
e.g., max-likelihood estimation, bayesian procedure ...  
(MAR required!)

## **Explicit value**

Missing values are characterized by a specific value („MISSING“). The selected model must be able to hand these specified missing values (most models assume MCAR)!

## **Further reading**

- How to Handle Missing Data,  
<https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4>
- Flexible Imputation of Missing Data  
<https://stefvanbuuren.name/fimd/>

# Outlier detection - Single attributes



An **outlier** is a value or data object that is far away or very different from all or most of the other data



**Categorical attributes:** an outlier is a value that occurs with an extremely lower frequency than the frequency of all other values

In some cases, the outliers can even be the target objects of the analysis

- Example: automatic quality control system
- Goal: train a classifier, classifying the parts as correct or with failures based on measurements of the produced parts
- The frequency of the correct parts will be so high that the parts with failure might be considered as outliers

**Numerical attributes:** outliers can be identified in boxplots or by statistical tests.

Problems: asymmetric distribution, large data sets

**Statistical test** that a sample following a normal distribution does not contain outliers (Grubb's test):

Define the statistic  $G = \frac{\max\{|x_i - \bar{x}| \mid 1 \leq i \leq n\}}{s}$  where  $x_1, \dots, x_n$  is the sample,  $\bar{x}$  its mean value and  $s$  its empirical standard deviation.

For a given significance level  $\alpha$ , the null hypothesis that the sample [coming from a normal distribution](#) does not contain outliers is rejected if

$$G > \frac{n-1}{\sqrt{n}} \sqrt{\frac{t_{1-\frac{\alpha}{2n}, n-2}^2}{n-2 + t_{1-\frac{\alpha}{2n}, n-2}^2}}$$

where  $t_{1-\frac{\alpha}{2n}, n-2}^2$  denotes the  $(1 - \frac{\alpha}{2n})$ -quantile of

the  $t$ -distribution with  $(n - 2)$  degrees of freedom.

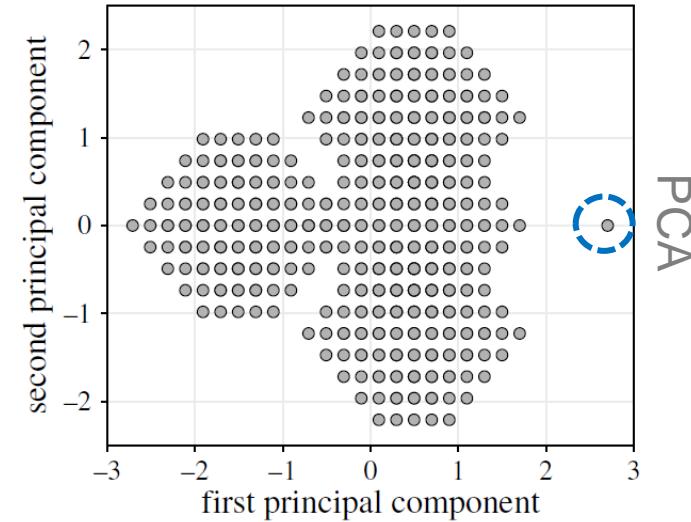


# Outlier detection - Multidimensional data

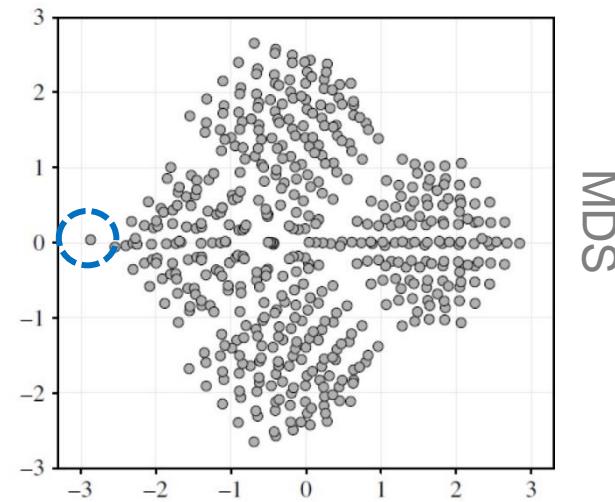
Scatter plots for (visually detecting) outliers w.r.t. two attributes.

PCA (or multi-dimensional scaling) for (visually) detecting outliers.

Cluster analysis techniques:  
outliers are those points which cannot be assigned to any cluster/which are far away from other clusters.



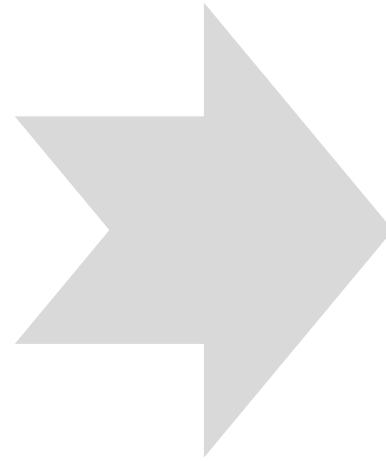
PCA



MDS



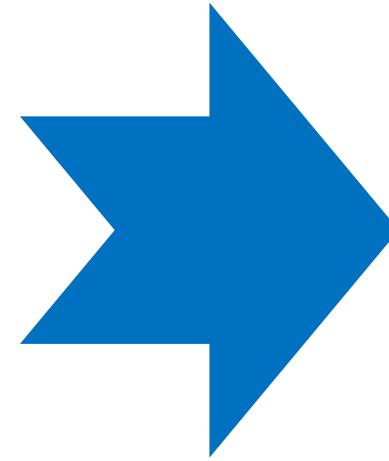
## Data Preparation



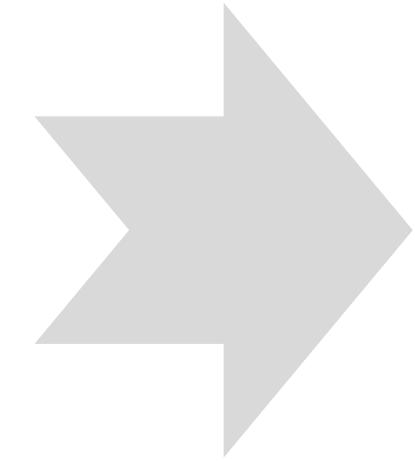
(1) Data selection



(2) Data cleaning



(3) Data transformation



(4) Data integration

# Data transformation

Categorical -> Numerical attributes

Some models can only handle numerical attributes,  
other models only categorical attributes.

In such cases, categorical attributes must be  
transformed into numerical ones or vice versa.

## Categorical attribute -> Numerical attribute:

A binary attribute can be turned into a numerical  
attribute with the values 0 and 1 (aka dummy variable)

A categorical attribute with more than two values, say  
 $a_1, \dots, a_k$ , **should not be turned into a single  
numerical attribute** with the values  $1, \dots, k$ , unless the  
attribute is an ordinal attribute. It should be turned into  $k$   
attributes  $A_1, \dots, A_k$  with values 0 and 1 (dummies).  
 $a_1$  is represented by  $A_i = 1$  and  $A_j = 0$  for  $i \neq j$ .

# Data transformation

Discretization: Numerical -> Categorical attributes

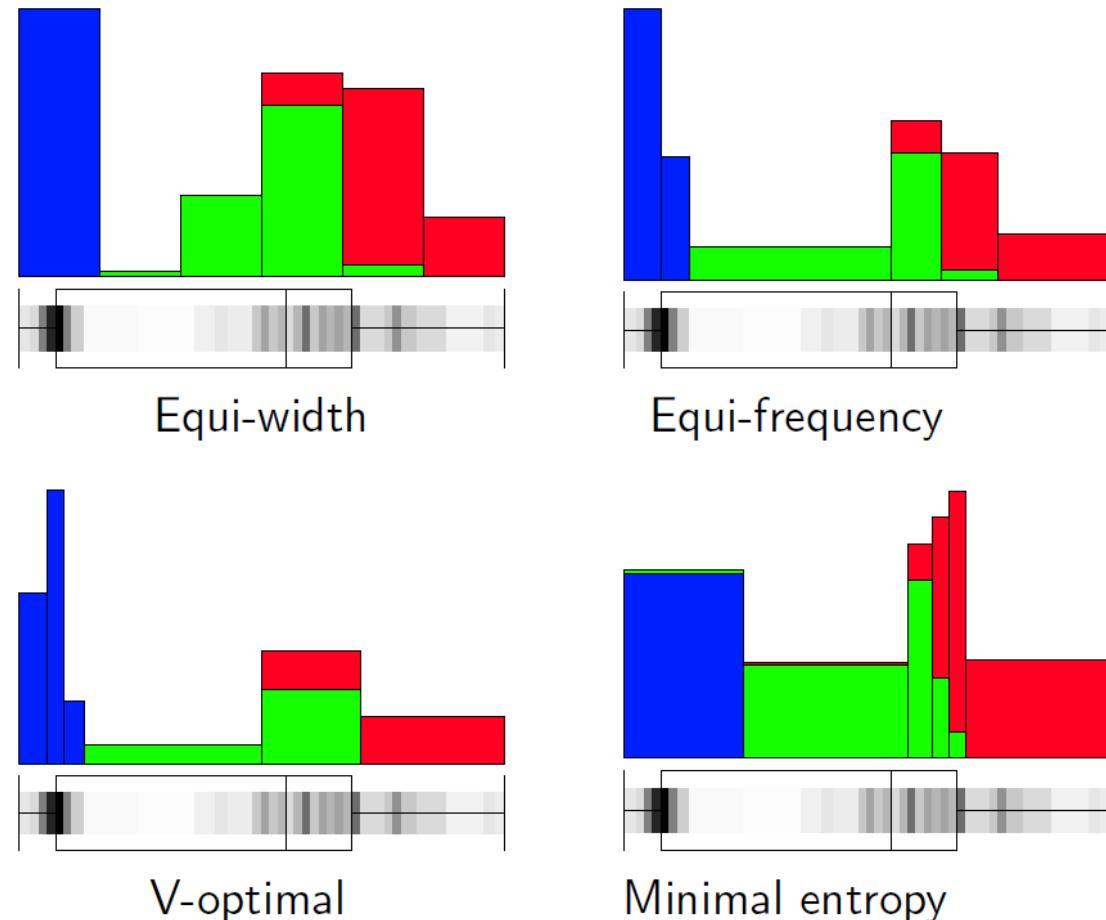
**Discretization techniques** refer to splitting a numerical range into a number of finite bins.

**Equi-width discretization.** Splits the range into intervals (bins) of the *same width*.

**Equi-frequency discretization.** Splits the range into intervals such that each interval (bin) contains (roughly) the *same number of records*.

**V-optimal discretization.** Minimizes  $\sum_i n_i V_i$  where  $n_i$  is the *number of data objects* in the  $i$ th interval and  $V_i$  is the sample *variance* of the data in this interval.

**Minimal entropy discretization.** Minimizes the *entropy*. (Only applicable in the case of classification problems, we'll dive deeper into this with decision trees)

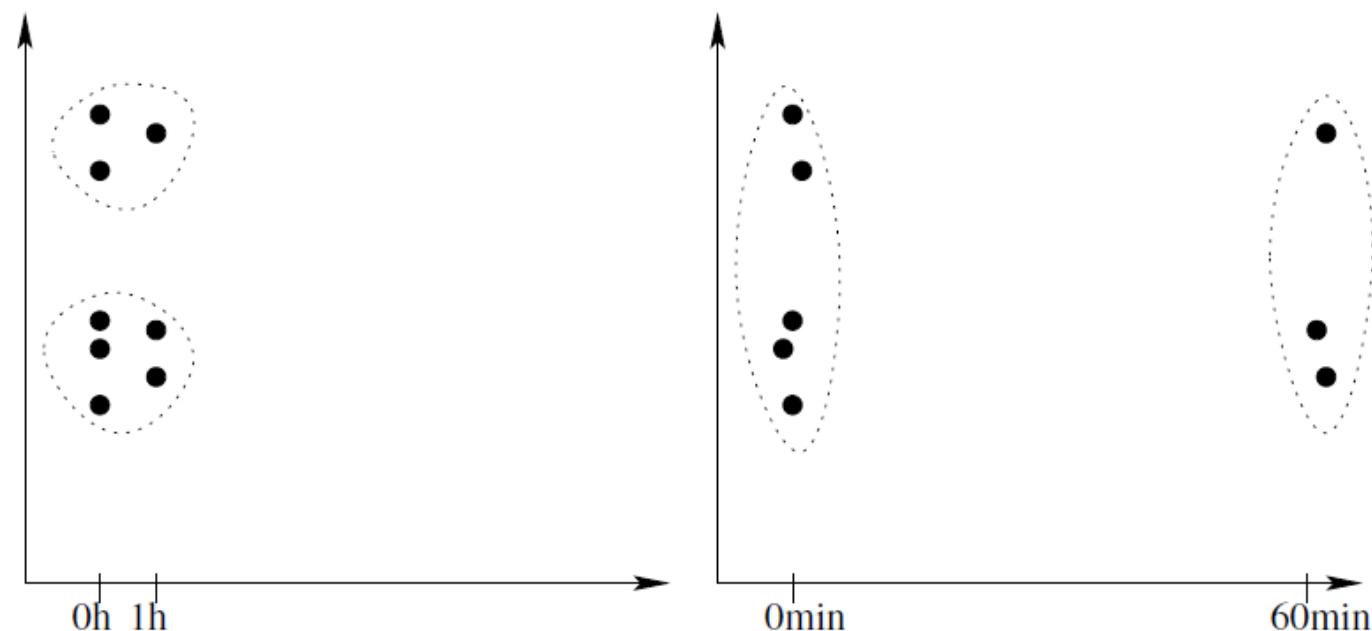


# Normalization | Standardization (1/2)



For some data analysis techniques (PCA, MDS, cluster analysis) the influence of an attribute depends on the **scale** or measurement unit.

To guarantee impartiality, some kind of standardization or normalization should be applied.



Ref.

## Min-max normalization:

For a numerical attribute  $X$  with  $\min_X$  and  $\max_X$  being the minimum and maximum value in the sample, the min-max normalization is defined as

$$n: \text{dom}X \rightarrow [0,1], \quad x \rightarrow \frac{x - \min_X}{\max_X - \min_X}$$

## Z-score standardization:

For a numerical attribute  $X$  with sample mean  $\hat{\mu}_X$  and empirical standard deviation  $\hat{\sigma}_X$ , the z-score standardization is defined as

$$s: \text{dom}X \rightarrow \mathbb{R}, \quad x \rightarrow \frac{x - \hat{\mu}_X}{\hat{\sigma}_X}$$

## Robust z-score standardization:

The sample mean and empirical standard deviation are easily affected by outliers. A more robust alternative is (see also boxplots):

$$s: \text{dom}X \rightarrow \mathbb{R}, \quad x \rightarrow \frac{x - \bar{x}}{IQR_X}$$

## Decimal scaling:

For a numerical attribute  $X$  and the smallest integer value  $s$  that is larger than  $\log_{10}(\max_X)$ , the decimal scaling is defined as

$$d: \text{dom}X \rightarrow [0,1], \quad x \rightarrow \frac{x}{10^s}$$

## Fragen?

- ✓ Data preparation
  - ✓ Data selection
  - ✓ Data cleaning
  - ✓ Data transformation
- Data integration

# Recommended reading

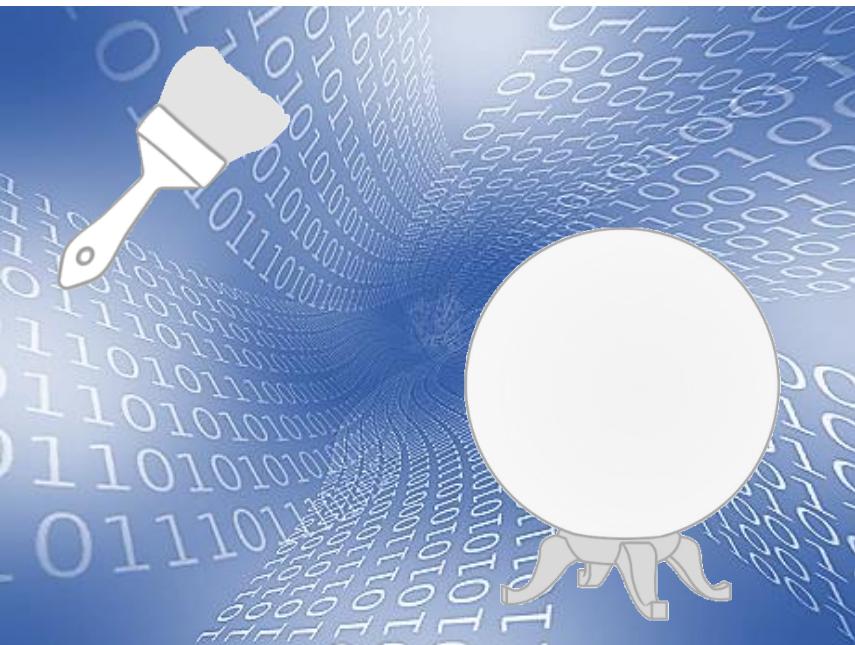
## Data Preparation

Berthold et al. Chapter 4, 6

Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann, 2011

# Bibliography

- J. Bertin (1983) *Semiology of graphics: diagrams, networks, maps*. University of Wisconsin Press. Originally in French: *Semiologie Graphique*, 1967
- Cairo, A. (2012). *The Functional Art: An introduction to information graphics and visualization*. New Riders.
- Mertens, P., & Meier, M. (2009). *Integrierte Informationsverarbeitung*. Wiesbaden: Gabler.
- Woolman, M. (2002). *Digital information graphics*. Watson-Guptill Publications, Inc..



# Business Intelligence

## 09 Predictive Modeling I

Prof. Dr. Bastian Amberg  
(summer term 2024)

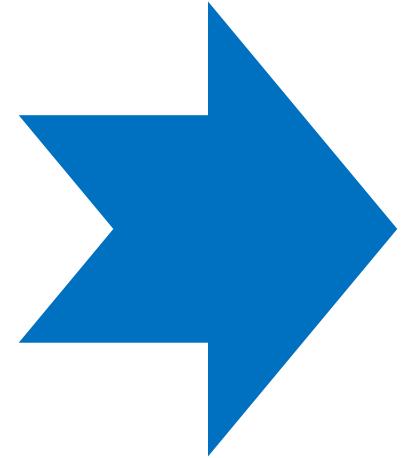
7.6.2024

# Schedule

|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study |                          |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|------------|--------------------------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |            |                          |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |            |                          |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  |            |            |                          |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining Introduction  |            |                          |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |            | Python-Analytics Chap. 1 |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization I            |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam |            | Chap. 2                  |  |  |
|           | W7                | 29.5. | Data Visualization II                               |                                   | 31.5. |   |            |                          |  |  |
|           | W8                | 5.6.  | Data Preparation                                    |                                   | 7.6.  | Predictive Modeling I (10:00 -12:00)  |            | BI-Project Start         |  |  |
|           | W9                | 12.6. | Predictive Modeling II, Fitting a Model I           |                                   | 14.6. | Python-Analytics-Online Exercise  |            |                          |  |  |
|           | W10               | 19.6. | Guest Lecture Dr. Ionescu                           |                                   | 21.6. | Fitting a Model II  |            |                          |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |            |                          |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I                                  |            |                          |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |            |                          |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |            | End                      |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin, 31.7.'24<br>Klausur 2.Termin, 2.10.'24                                      |            | Projektbericht           |  |  |



## Data Preparation

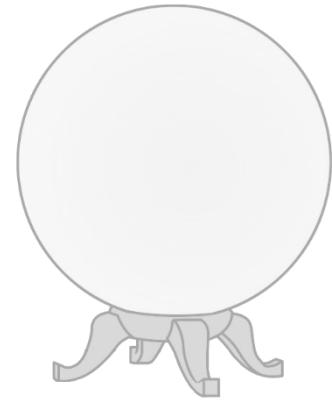


- ✓ Data selection
- ✓ Data cleansing
- **Data transformation**
- ✓ Data integration

## Predictive Modeling I



Introductory example  
Attribute Selection,  
Decision Trees



# Data transformation

Categorical -> Numerical attributes

Some models can only handle numerical attributes,  
other models only categorical attributes.

In such cases, categorical attributes must be  
transformed into numerical ones or vice versa.

## Categorical attribute -> Numerical attribute:

A binary attribute can be turned into a numerical  
attribute with the values 0 and 1 (aka dummy variable)

A categorical attribute with more than two values, say  
 $a_1, \dots, a_k$ , **should not be turned into a single  
numerical attribute** with the values  $1, \dots, k$ , unless the  
attribute is an ordinal attribute. It should be turned into  $k$   
attributes  $A_1, \dots, A_k$  with values 0 and 1 (dummies).  
 $a_1$  is represented by  $A_i = 1$  and  $A_j = 0$  for  $i \neq j$ .

# Data transformation

Discretization: Numerical -> Categorical attributes

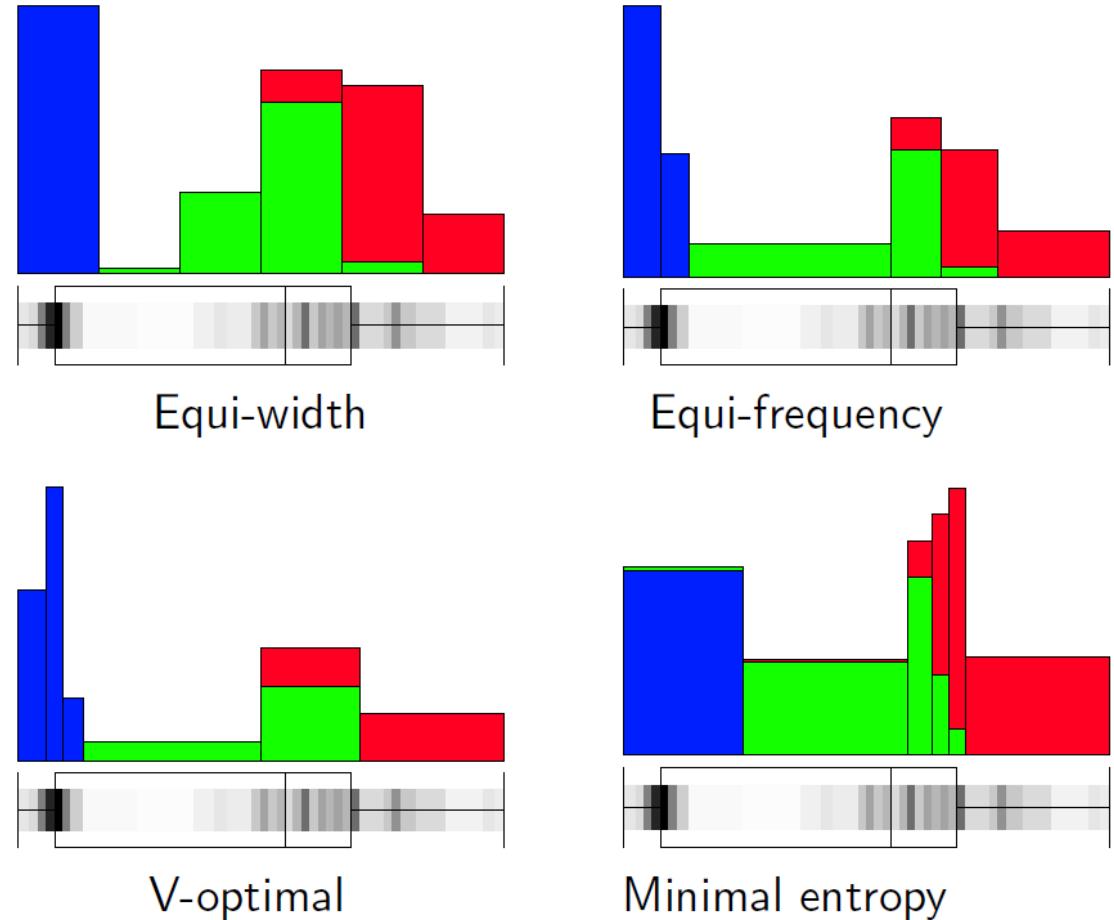
**Discretization techniques** refer to splitting a numerical range into a number of finite bins.

**Equi-width discretization.** Splits the range into intervals (bins) of the *same width*.

**Equi-frequency discretization.** Splits the range into intervals such that each interval (bin) contains (roughly) the *same number of records*.

**V-optimal discretization.** Minimizes  $\sum_i n_i V_i$  where  $n_i$  is the *number of data objects* in the  $i$ th interval and  $V_i$  is the sample *variance* of the data in this interval.

**Minimal entropy discretization.** Minimizes the *entropy*. (Only applicable in the case of classification problems, we'll dive deeper into this with decision trees)

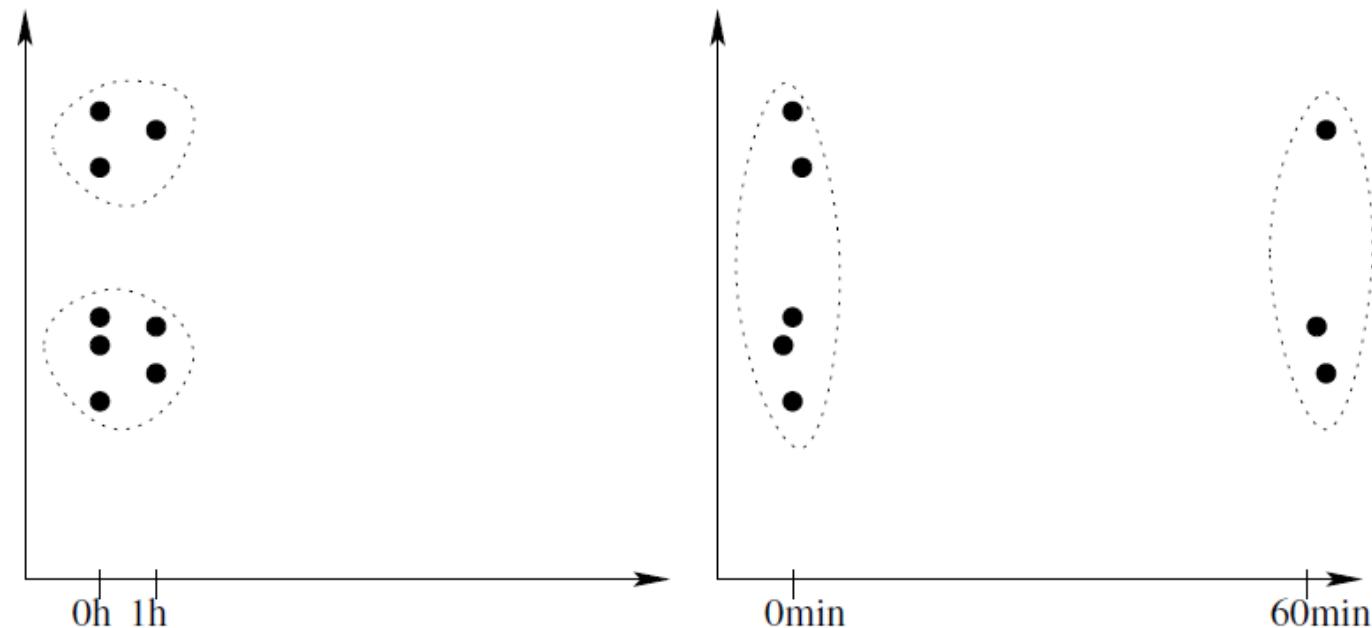


# Normalization | Standardization (1/2)



For some data analysis techniques (PCA, MDS, cluster analysis) the influence of an attribute depends on the **scale** or measurement unit.

To guarantee impartiality, some kind of standardization or normalization should be applied.



Ref.

# Normalization | Standardization (2/2)

## Min-max normalization:

For a numerical attribute  $X$  with  $\min_X$  and  $\max_X$  being the minimum and maximum value in the sample, the min-max normalization is defined as

$$n: \text{dom}X \rightarrow [0,1], \quad x \rightarrow \frac{x - \min_X}{\max_X - \min_X}$$

## Z-score standardization:

For a numerical attribute  $X$  with sample mean  $\hat{\mu}_X$  and empirical standard deviation  $\hat{\sigma}_X$ , the z-score standardization is defined as

$$s: \text{dom}X \rightarrow \mathbb{R}, \quad x \rightarrow \frac{x - \hat{\mu}_X}{\hat{\sigma}_X}$$

## Robust z-score standardization:

The sample mean and empirical standard deviation are easily affected by outliers. A more robust alternative is (see also boxplots):

$$s: \text{dom}X \rightarrow \mathbb{R}, \quad x \rightarrow \frac{x - \bar{x}}{IQR_X}$$

## Decimal scaling:

For a numerical attribute  $X$  and the smallest integer value  $s$  that is larger than  $\log_{10}(\max_X)$ , the decimal scaling is defined as

$$d: \text{dom}X \rightarrow [0,1], \quad x \rightarrow \frac{x}{10^s}$$

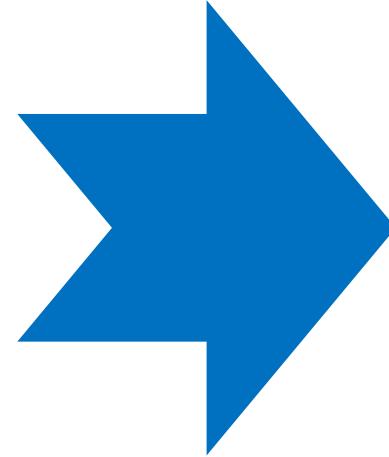


## Data Preparation

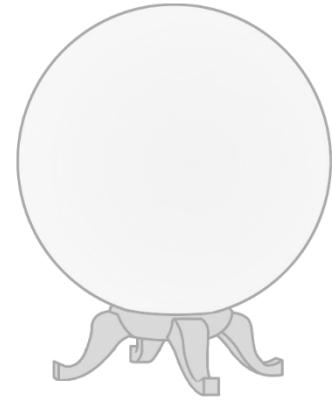


- ✓ Data selection
- ✓ Data cleansing
- ✓ Data transformation
- ✓ Data integration

## Predictive Modeling I



Introductory example  
Attribute Selection,  
Decision Trees

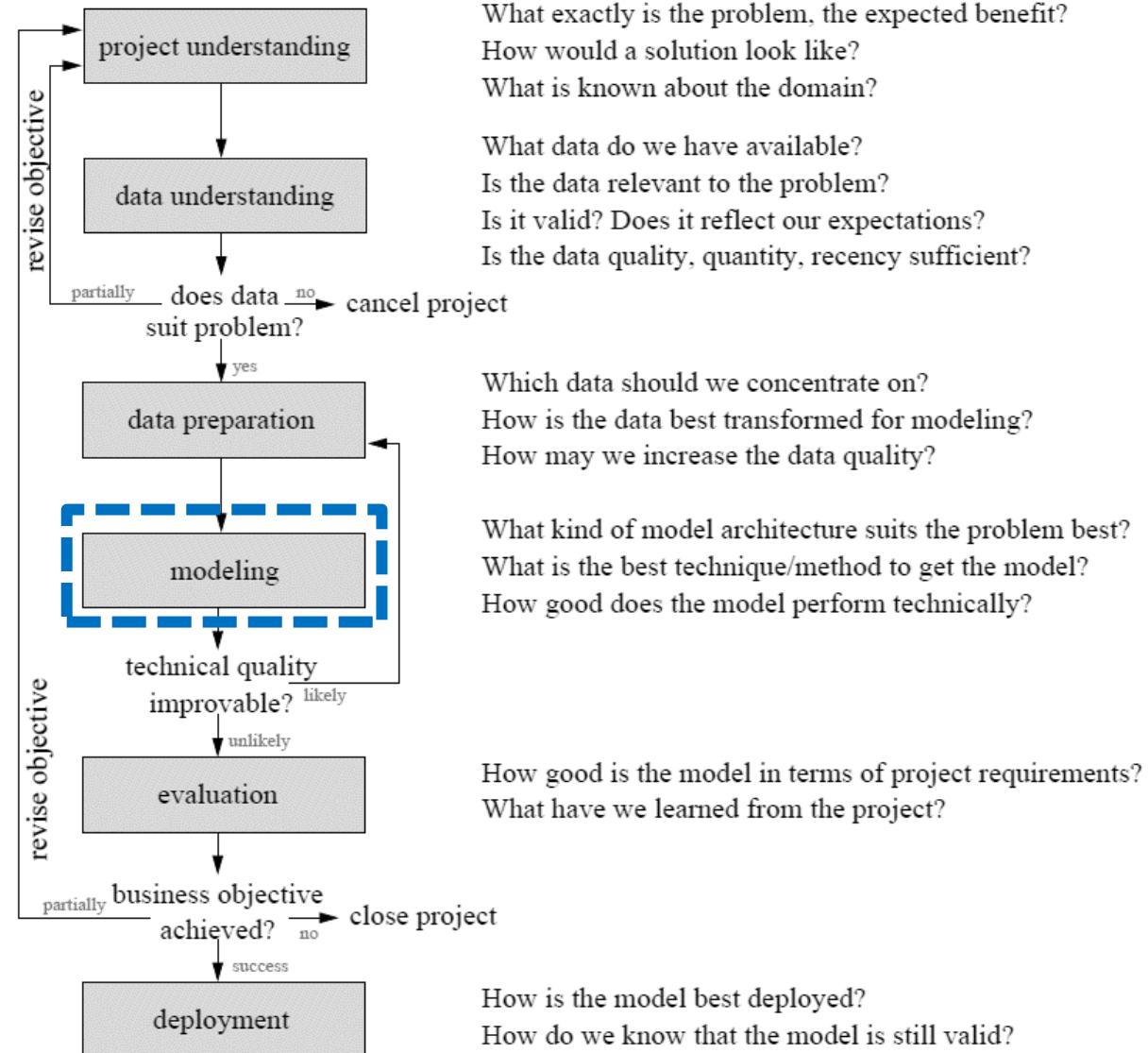


## Cross Industry Standard Process for Data Mining

Iteration as a rule

Process of data exploration

Implementation of the KDD Process



What exactly is the problem, the expected benefit?

How would a solution look like?

What is known about the domain?

What data do we have available?

Is the data relevant to the problem?

Is it valid? Does it reflect our expectations?

Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?

How is the data best transformed for modeling?

How may we increase the data quality?

What kind of model architecture suits the problem best?

What is the best technique/method to get the model?

How good does the model perform technically?

How good is the model in terms of project requirements?

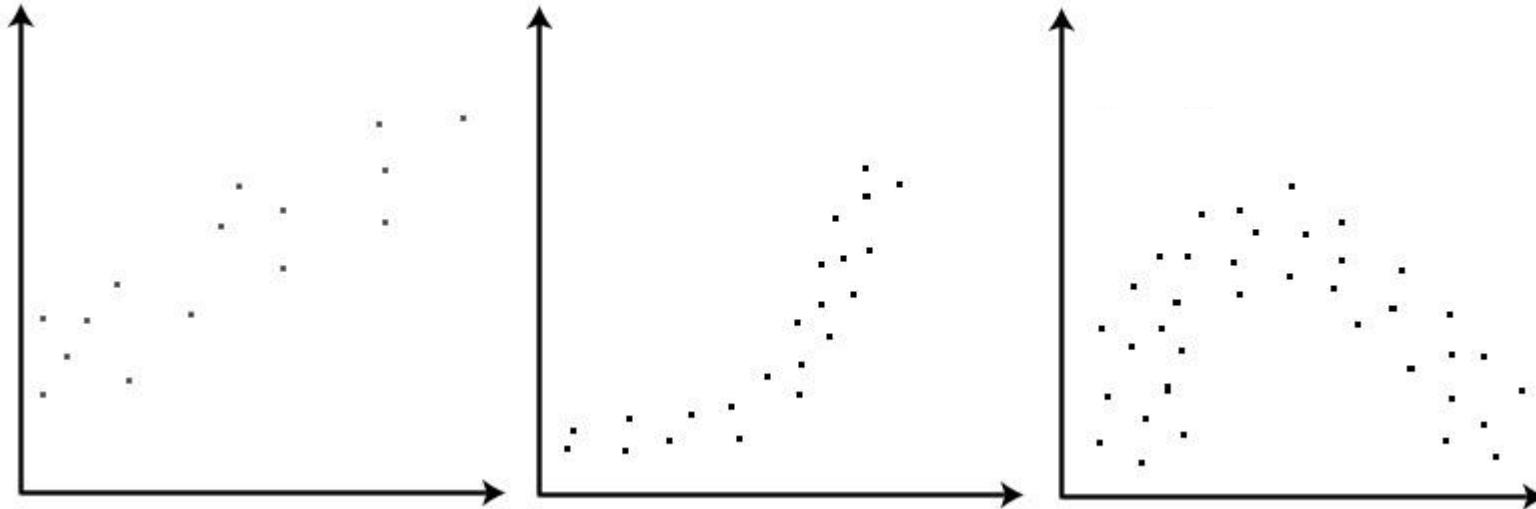
What have we learned from the project?

How is the model best deployed?

How do we know that the model is still valid?

# Let's revisit data understanding

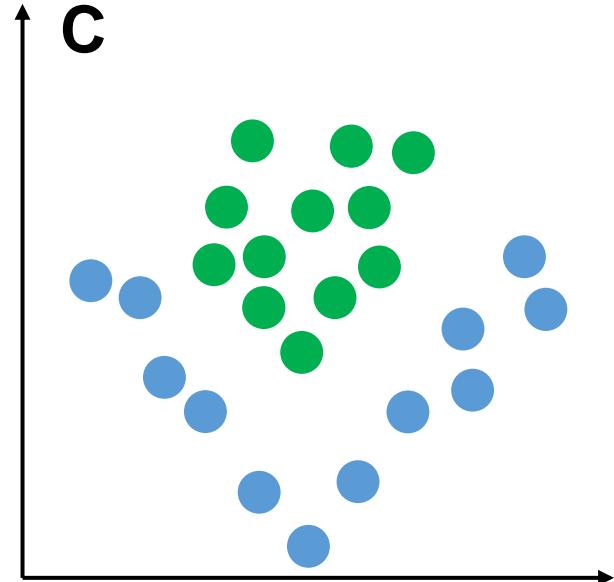
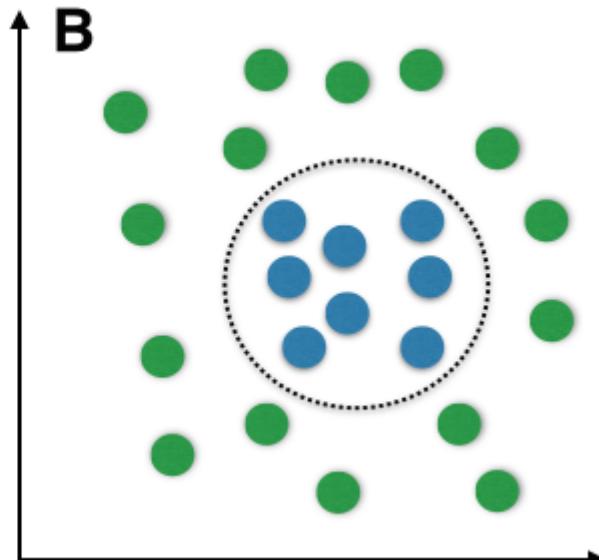
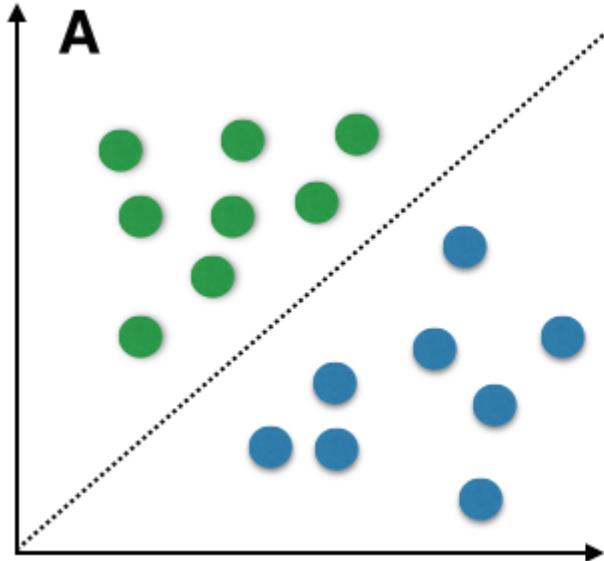
Types of relationships



Ref.

# Let's revisit data understanding

On our way to classification problems



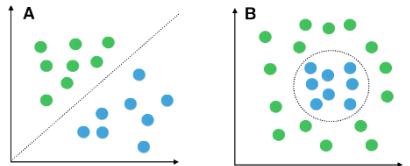
Ref. Data from <https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset> (manipulated)

# Introduction

Fundamental concept of DM: **predictive modeling**

Supervised segmentation: how can we segment the population with respect to something that we would like to predict or estimate

e.g.



„Which customers are likely to leave the company when their contracts expire?“

„Which potential customers are likely not to pay off their account balances?“

Technique: find or **select important, informative variables / attributes** of the entities with respect to a target

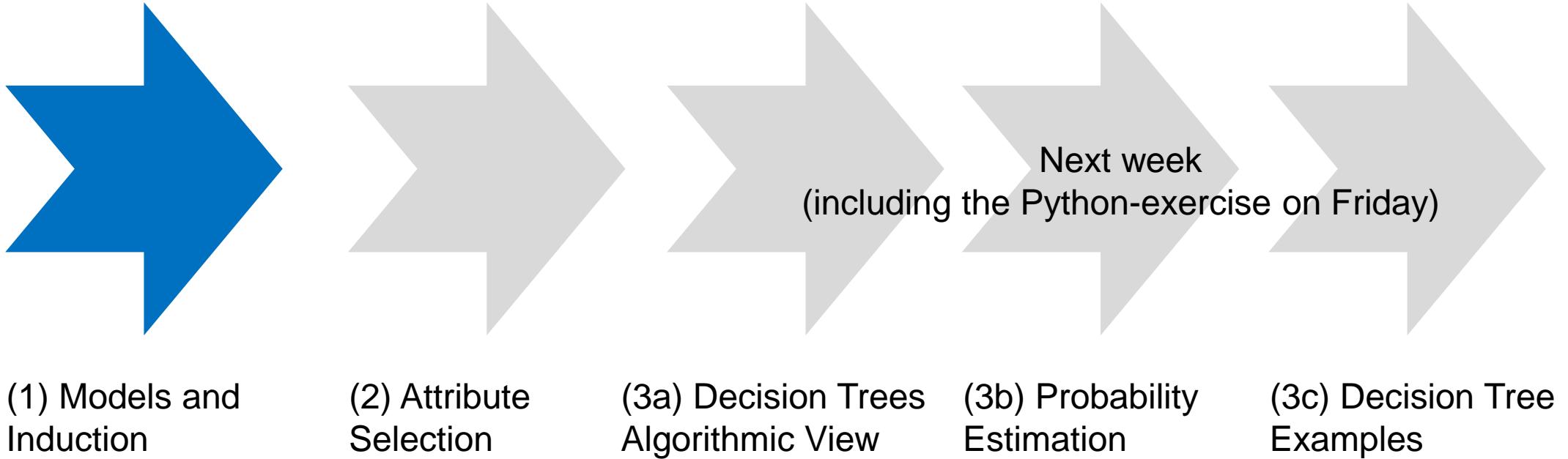
Is there one or more other variables that reduces our uncertainty about the value of the target?

Select **informative subsets** in large databases

Ref.



# Agenda



# Models and Supervised Learning

A model is a simplified representation of reality created to serve a purpose

A predictive model is a formula for **estimating the unknown value of interest**: the target

Classification/class-probability estimation and regression models

**Prediction = estimate an unknown value**

Credit scoring, spam filtering, fraud detection

Descriptive modeling: gain insight into the underlying phenomenon or process



## Supervised learning

Model creation where the model describes a relationship between a set of selected variables (attributes/features) and a **predefined variable** (target)

The model estimates the value of the target variable as a function of the features

## Supervised learning

Model creation where the model describes a relationship between a set of selected variables (attributes/features) and a **predefined variable** (target)

The model estimates the value of the target variable as a function of the features

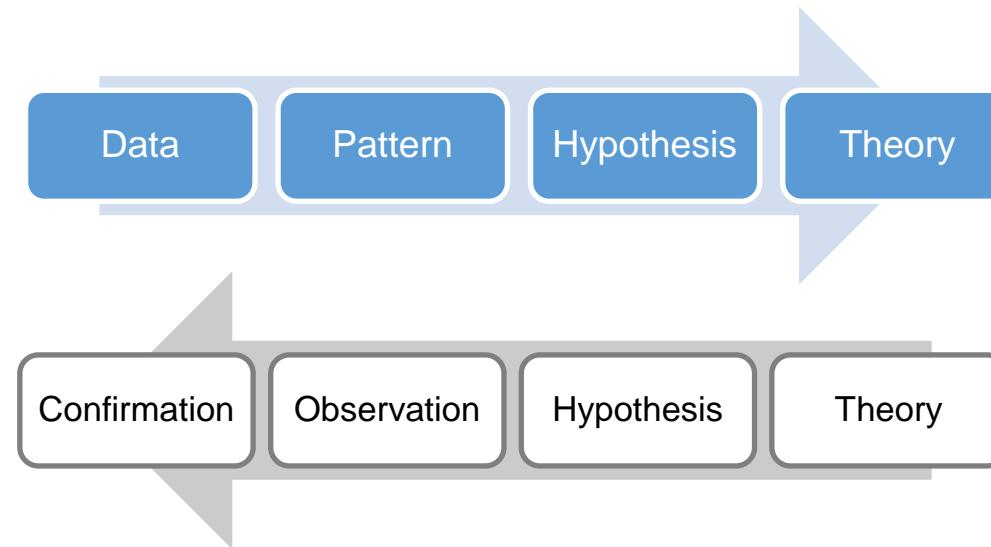
## Induction

Creation of models from data

Refers to generalizing from specific case to general rules

How can we select one or more attributes / features / variables that will best divide the sample w.r.t. our target variable of interest?

Unlike deduction:



# Now: From Data to Decision Trees

| Attributes |           |     |          |           | Target attribute |
|------------|-----------|-----|----------|-----------|------------------|
| Name       | Balance   | Age | Employed | Write-off |                  |
| Mike       | \$200,000 | 42  | no       | yes       |                  |
| Mary       | \$35,000  | 33  | yes      | no        |                  |
| Claudio    | \$115,000 | 40  | no       | no        |                  |
| Robert     | \$29,000  | 23  | yes      | yes       |                  |
| Dora       | \$72,000  | 31  | no       | no        |                  |

This is one row (example).

Feature vector is: <Claudio,115000,40,no>

Class label (value of Target attribute) is no

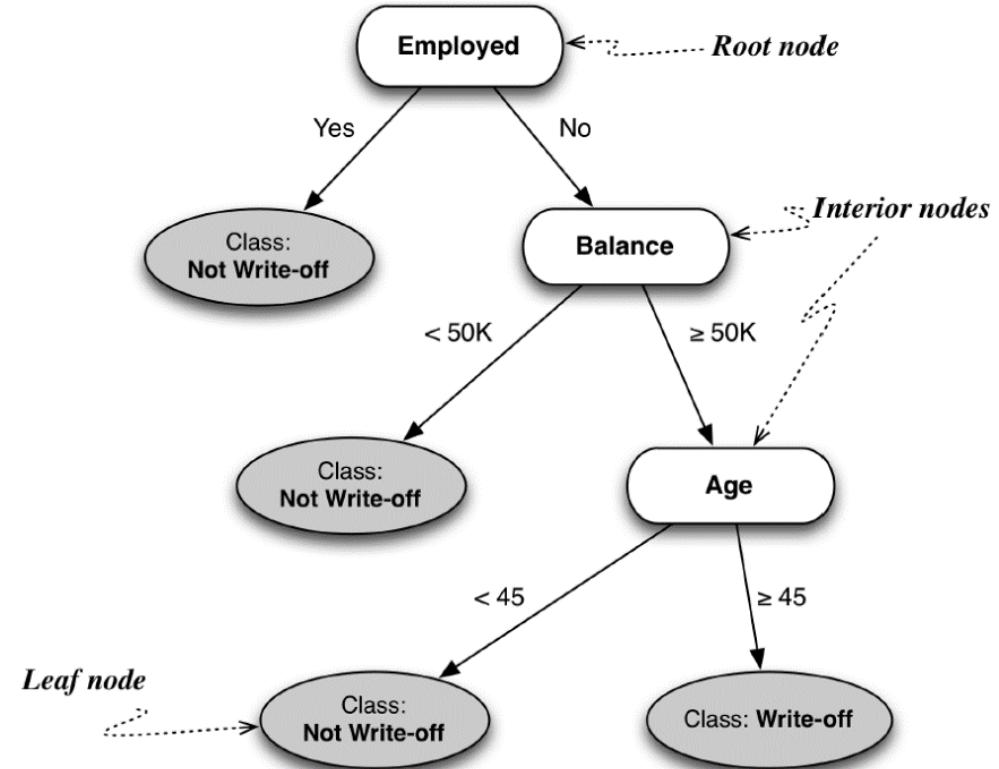
e.g., Michi, 40,000, 24, yes, Write-off?

If we select multiple attributes each giving some information gain, it's not clear how to put them together

→ **decision trees**

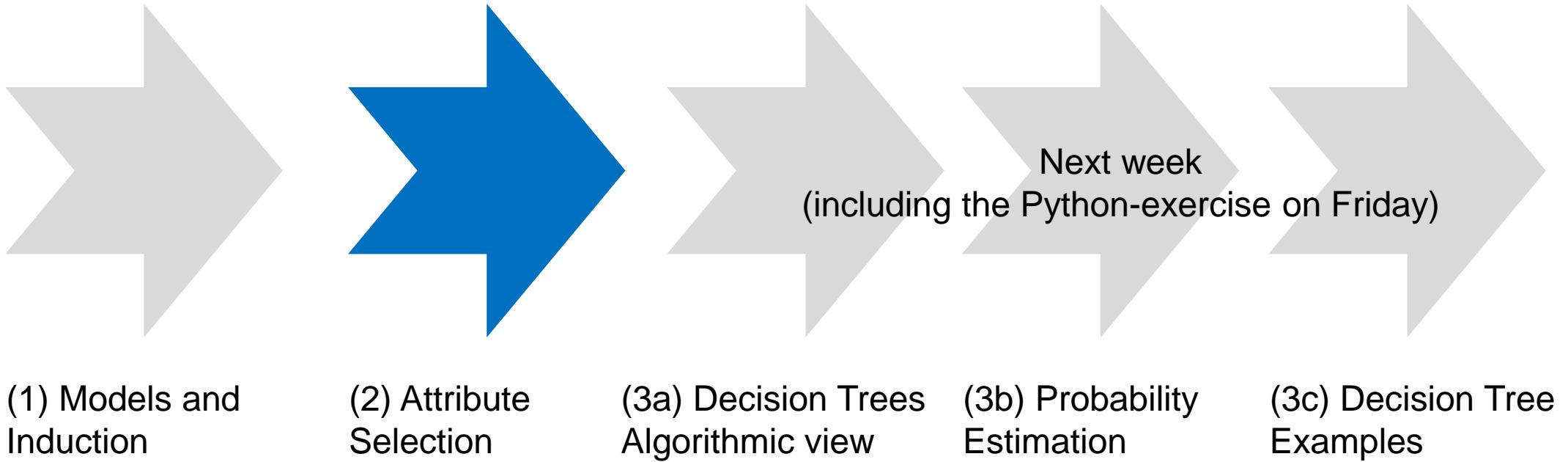
Decision trees are often used as **predictive models**

Ref.



The tree creates a segmentation of the data  
Each *node* in the tree contains a test of an attribute  
Each *path* eventually terminates at a *leaf*  
Each leaf corresponds to a *segment*, and the attributes and values along the path give the characteristics  
Each leaf contains a value for the target variable

# Agenda



# Supervised segmentation

Intuitive approach



Segment the population into **subgroups** which have different values for the target variable (*high inter-group discrimination*) and similar values for the target variable within the subgroup (*low intra-group discrimination*)



Segmentation may provide a **human-understandable set of segmentation patterns** (e.g., „*Middle-aged professionals who reside in New York City on average have a churn rate of 5%*“)

How can we (automatically) judge whether a variable contains important information about the target variable?

*What variable gives us the most information about the future churn rate of the population?*

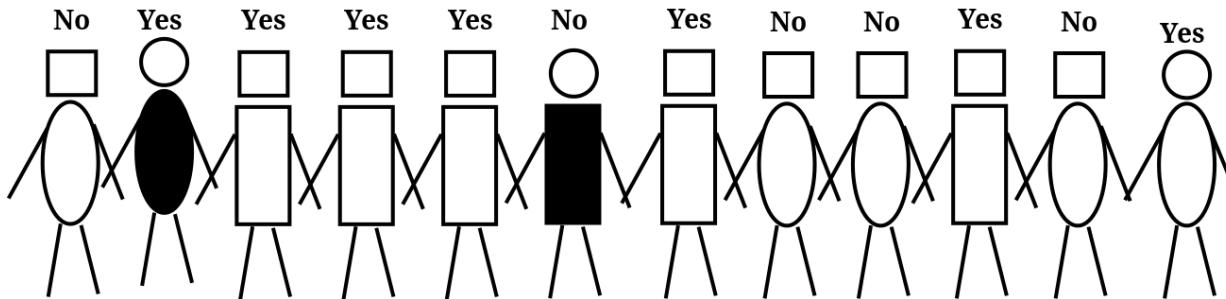
# Supervised Segmentation – Decision Tree Example

How can we (automatically) judge whether a variable contains important information about the target variable?

## Consider a binary (two class) classification problem

Binary target variable: {"Yes", "No"}

Attributes: head-shape, body-shape, shirt-color



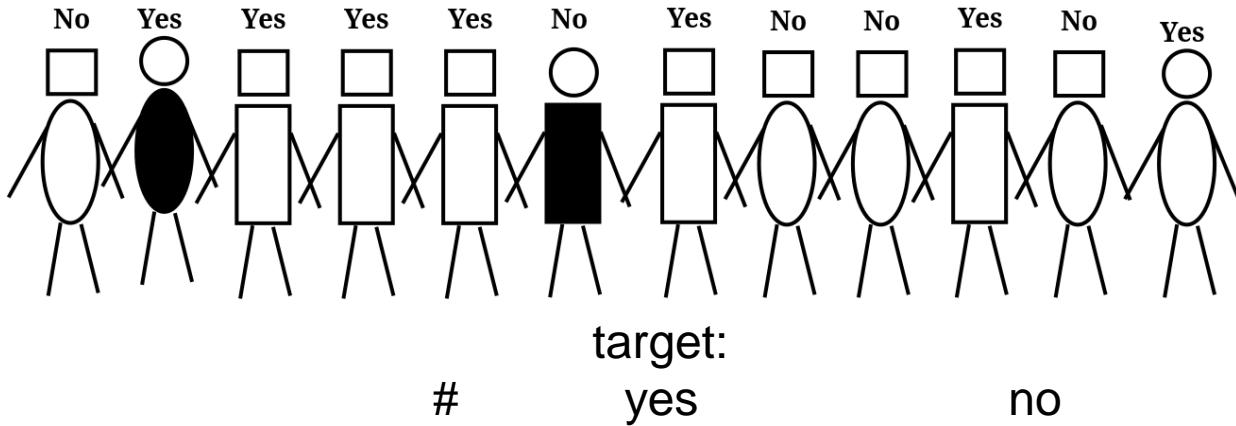
*Which of the attributes would be the best to segment these people in groups such that write-offs will be distinguished from non-write-offs?*

Resulting groups should be as **pure** as possible!

# Exercise – attribute selection

And a first step to build decision trees

Which attribute should be selected first?



head-shape:

- square
- circular

body-shape:

- rectangular
- oval

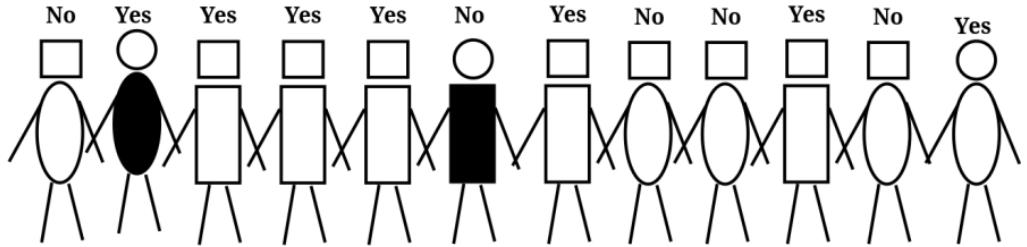
shirt-color:

- white
- black

# Reduce impurity

## Attributes rarely split a group perfectly

- Consider if the second person were not there
  - Then, *shirt-color* would create a pure segment where all individuals have (*write-off*=*no*)
- – Then, the condition *shirt-color=black* would only split off one single data point into the pure subset. Is this better than a split that does not produce any pure subset, but reduces the impurity more broadly?
- Not all attributes are binary. How do we compare the splitting into two groups with **splitting into more groups**?
- Some attributes take on **numeric values**. How should we think about creating supervised segmentations using numeric attributes?



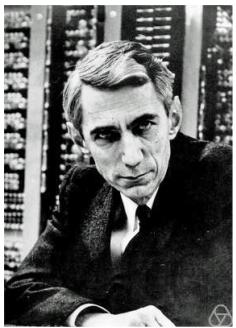
Purity measure → **information gain / entropy**



# Entropy

Entropy is a **measure of disorder** that can be applied to a set

Disorder corresponds to how mixed (impure) a segment is w.r.t. the properties of interest (values of target)



Claude E. Shannon

*entropy*

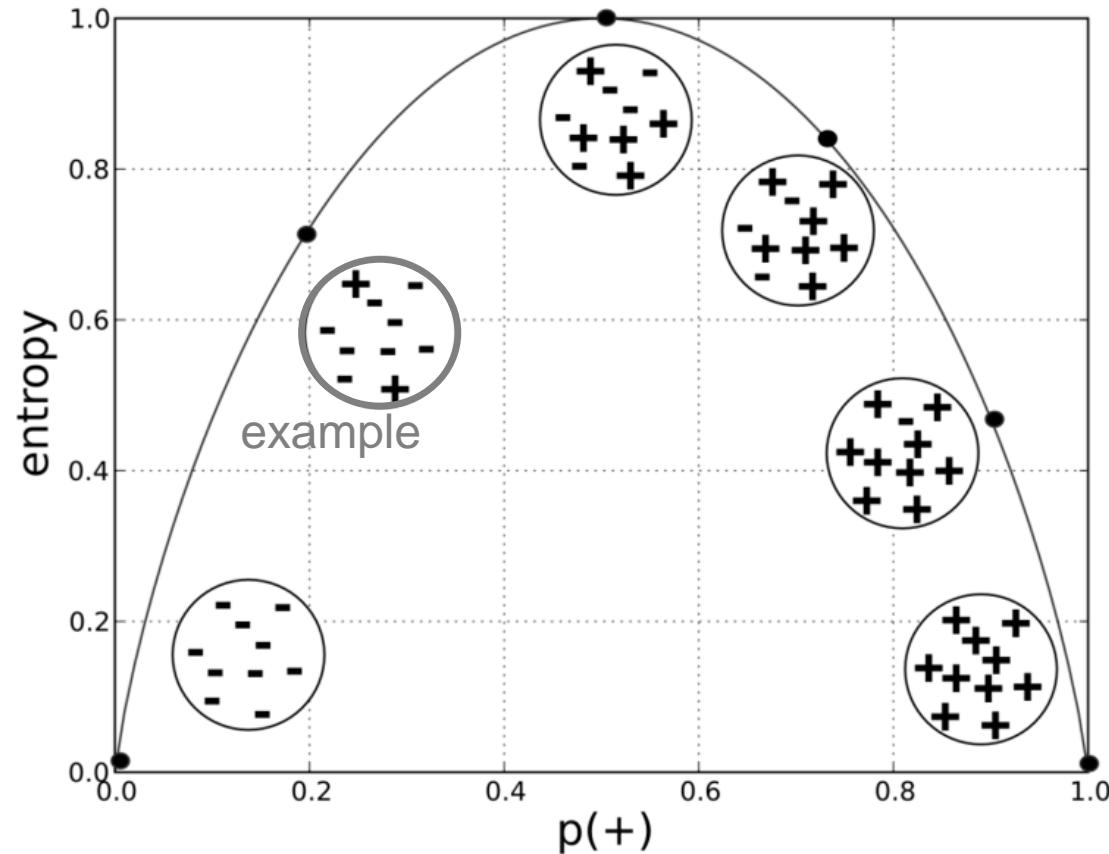
$$= -p_1 \log_2(p_1) - p_2 \log_2(p_2) - \cdots - p_n \log_2(p_n)$$

with  $p_i$  as the relative percentage of property  $i$  within the set, ranging from  $p_i = 0$  to  $p_i = 1$  (all have property  $i$ ).

**Entropy measures the general disorder of a set**, ranging from **zero** at minimum disorder (the set has members all with the same, single property) to **one** at maximal disorder (the properties are equally mixed)

Example

```
#Entropy :  
import math as m  
def entropy(p1,p2):  
    return - p1 * m.log2(p1) - p2 * m.log2(p2)  
  
#Excursus (alternatively): gini-coefficient  
def gini(p1,p2):  
    return 1- (p1*p1 + p2*p2)
```



$$p(-) = 8/10 \quad p(+) = 2/10$$

$$\begin{aligned} \text{entropy}(S) &= -[0.8 \times \log_2(0.8) + 0.2 \times \log_2(0.2)] \\ &= -[0.8 \times (-0.32) + 0.2 \times (-2.32)] \approx 0.72 \end{aligned}$$

# Information gain (IG)

Basic idea behind information gain:

Measure how much an attribute improves (**decreases**) entropy over the whole segmentation it creates.

IG measures the change in entropy due to any amount of new information added

How much purer are the **children c** (split set) compared to their **parent** (original set)?

$$\begin{aligned} IG(\text{parent}, \text{children}) \\ = & \text{entropy}(\text{parent}) - [p(c_1) \times \text{entropy}(c_1) + \\ & p(c_2) \times \text{entropy}(c_2) + \dots] \end{aligned}$$

The entropy for each child  $c_i$  is weighted by **the proportion of instances** belonging to that child

# Information gain

## Example 1

Two-class problem ( $\bullet$  and  $\star$ )

Entropy parent:

$$\begin{aligned} &= -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)] \\ &\approx -[0.53 \times -0.9 + 0.47 \times -1.1] \\ &\approx 0.99 \quad (\text{very impure}) \end{aligned}$$

Entropy left child:

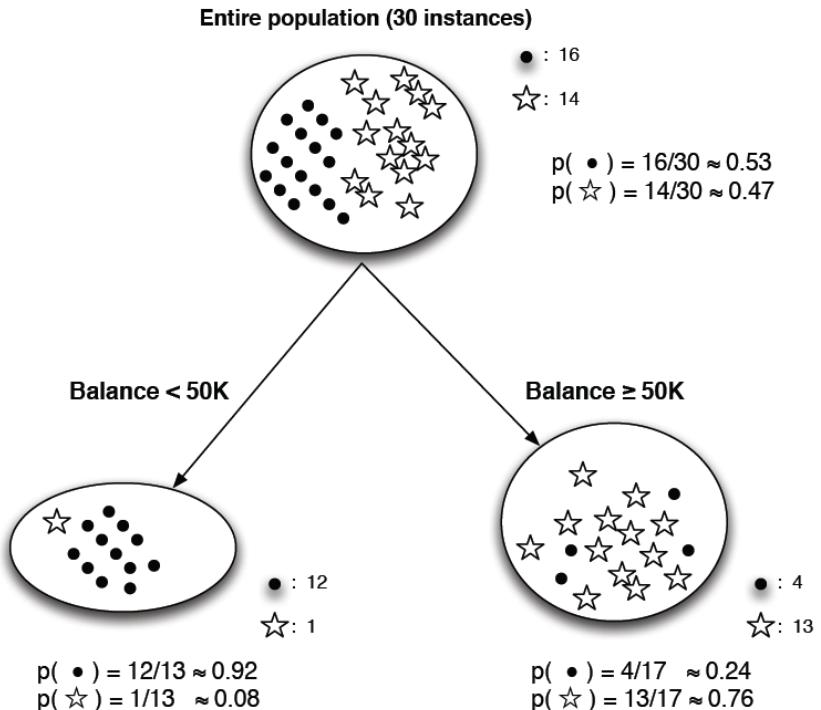
$$\begin{aligned} &= -[p(\bullet) * \log_2 p(\bullet) + p(\star) * \log_2 p(\star)] \\ &\approx -[0.92 \times (-0.12) + 0.08 \times (-3.7)] \\ &\approx 0.39 \end{aligned}$$

Entropy right child:

$$\begin{aligned} &= -[p(\bullet) * \log_2 p(\bullet) + p(\star) * \log_2 p(\star)] \\ &\approx -[0.24 \times (-2.1) + 0.76 \times (-0.39)] \\ &\approx 0.79 \end{aligned}$$

IG:

$$\begin{aligned} &= \text{entropy}(\text{parent}) - [p(\text{Balance} < 50K) \times \text{entropy}(\text{Balance} < 50K) \\ &\quad + p(\text{Balance} \geq 50K) \times \text{entropy}(\text{Balance} \geq 50K)] \\ &\approx 0.99 - [0.43 \times 0.39 + 0.57 \times 0.79] \\ &\approx 0.37 \end{aligned}$$



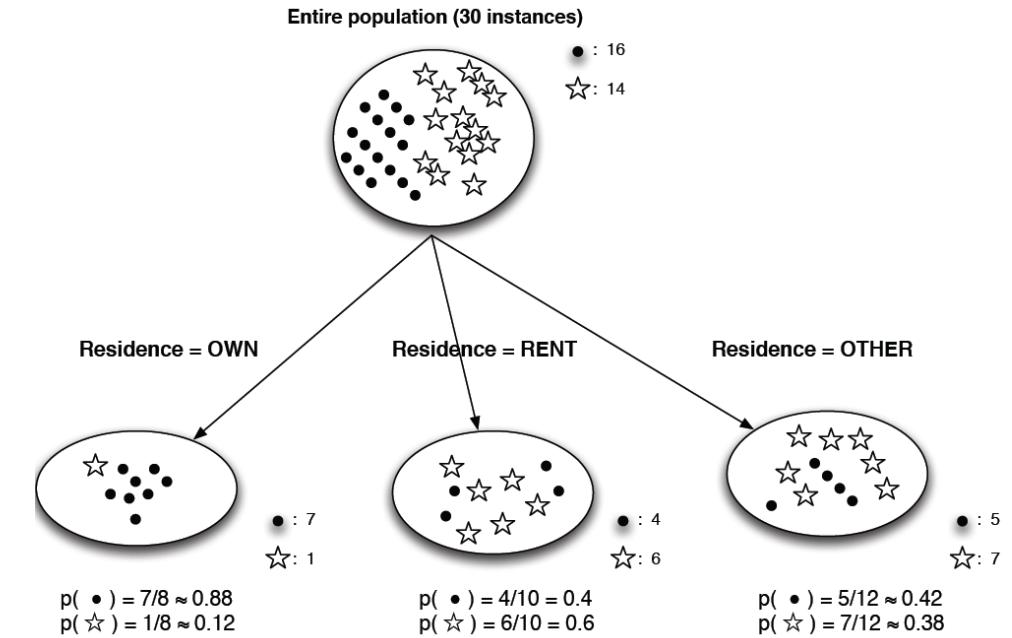
# Information gain

## Example 2

Same example, but different candidate split

attribute here: *residence*

entropy and information gain computations:



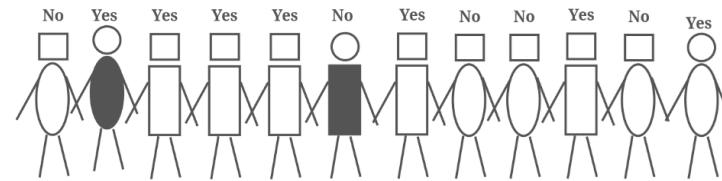
The *residence* variable does have a positive information gain, but it is lower than that of *balance*.

$$\begin{aligned} \text{entropy}(\text{parent}) &\approx 0.99 \\ \text{entropy}(\text{Residence=OWN}) &\approx 0.54 \\ \text{entropy}(\text{Residence=RENT}) &\approx 0.97 \\ \text{entropy}(\text{Residence=OTHER}) &\approx 0.98 \\ \text{IG} &\approx 0.13 \end{aligned}$$

# Exercise – Information gain

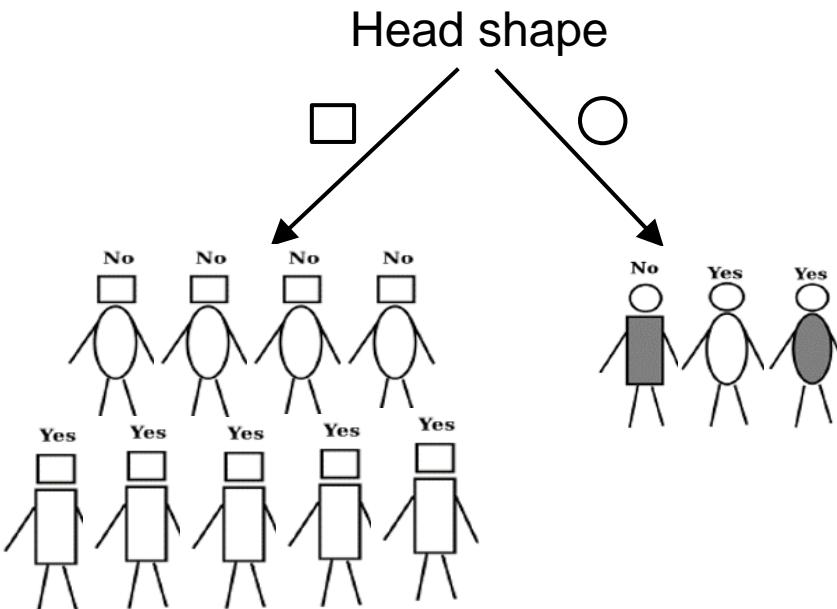
Example 3, 4 and 5

Which attribute to choose?



Yes: 7  
No: 5

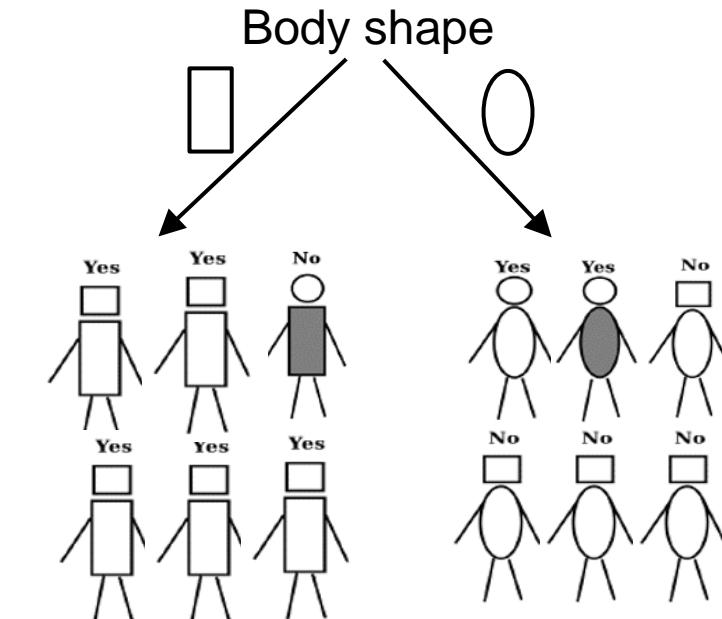
$$\text{entropy}(\text{parent}) \approx 0,98$$



$$\text{entropy}(\square) \approx 0,99 \quad \text{entropy}(\circ) \approx 0,92$$

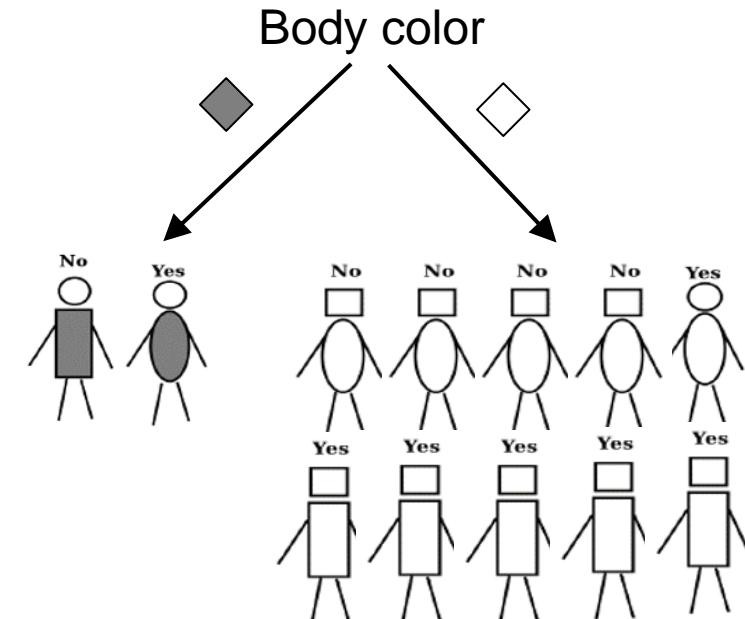
$$\text{IG} \approx$$

Ref.



$$\text{entropy}(\square) \approx$$

$$\text{IG} \approx$$

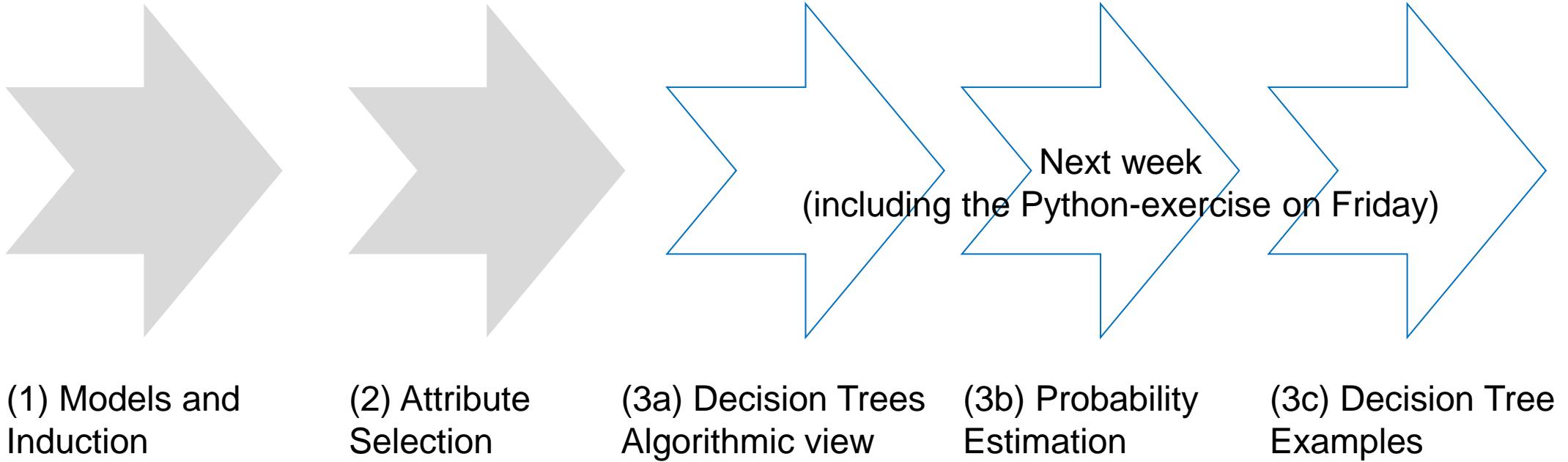


$$\text{entropy}(\circ) \approx$$

$$\text{IG} \approx$$

$$\text{entropy}(\diamond) \approx$$

# Agenda



## Fragen?

- ✓ Predictive modeling I
  - ✓ Models and induction
  - ✓ Attribute selection
  - ✓ Decision trees - introduction
    - Algorithms for tree induction
    - Probability estimation tree

# Todos for this week

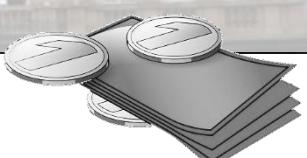
Choose your project and your project group (4 persons per group)

See slides BI-Project ("Folien – Projektaufgabe – ab 7.6.'24" in Blackboard)

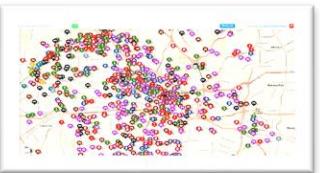


## Costa Rican Household Poverty Prediction

Set of household characteristics from a representative sample of households  
Make sure the right people are given enough aid  
*Goal: Predict the level of need (income level)*

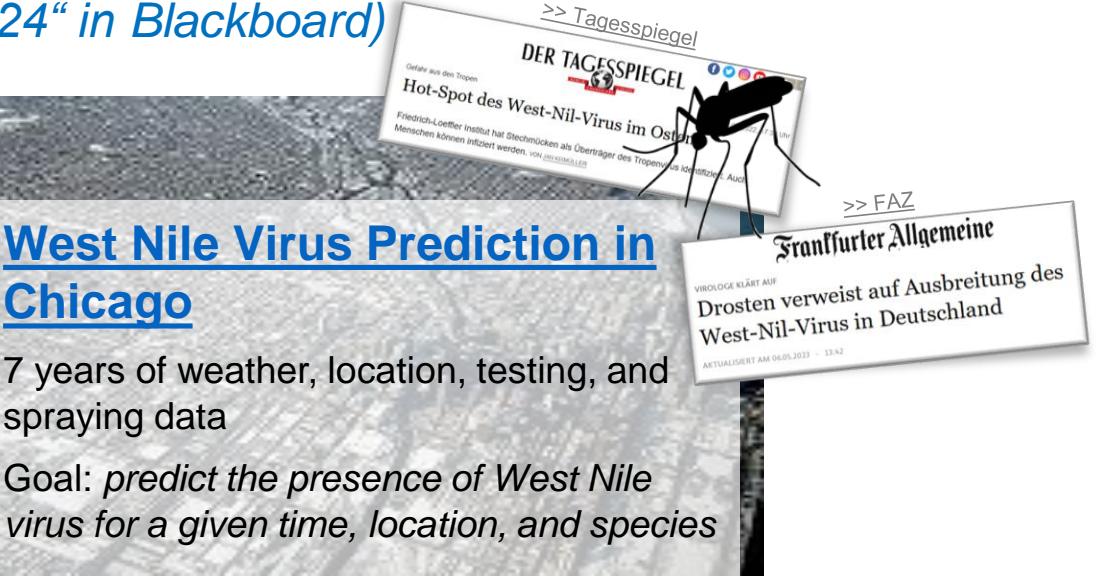


>> Crime Mapping



## West Nile Virus Prediction in Chicago

7 years of weather, location, testing, and spraying data  
*Goal: predict the presence of West Nile virus for a given time, location, and species*



>> Tagesspiegel  
DER TAGESSPIEGEL  
Hot-Spot des West-Nil-Virus im Osten  
Friedrich-Loeffler Institut hat Stechmücken als Überträger des Tropenvirus identifiziert. Auch Menschen können infiziert werden. von JENS HÄCKER

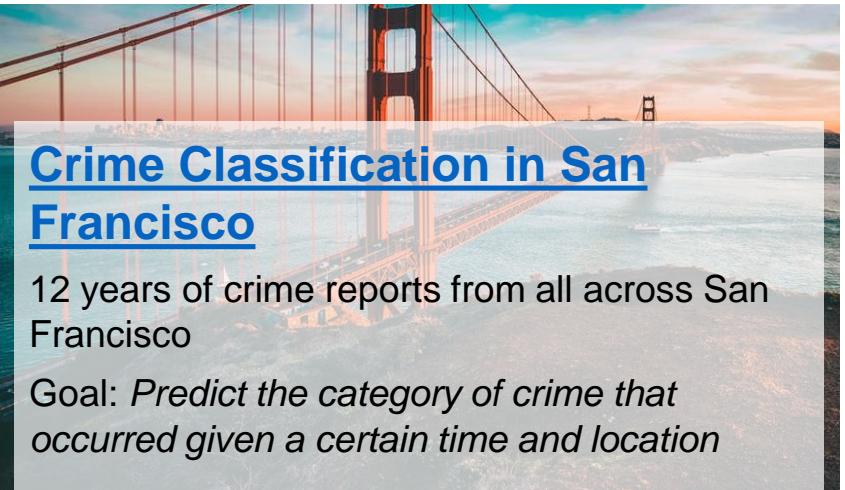
>> FAZ  
Frankfurter Allgemeine  
VIROLOGE KLÄRT AUF  
Drosten verweist auf Ausbreitung des West-Nil-Virus in Deutschland  
AKTUALISIERT AM 06.07.2013 - 13:42

## Crime Classification in Los Angeles

4 years of crime reports from all across Los Angeles  
*Goal: Predict the category of crime that occurred given a certain time and location*

## Crime Classification in San Francisco

12 years of crime reports from all across San Francisco  
*Goal: Predict the category of crime that occurred given a certain time and location*



TEAM



# Recommended reading

## Data Preparation

Berthold et al. Chapter 4, 6

Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann, 2011

## Predictive Modeling

Provost, F., Data Science for Business

Fawcett, T. Chapter 3

Berthold et al. Guide to Intelligent Data Analysis  
Chapter 8.1

Hand, D. Principles of Data Mining  
Chapter 10

Quinlan, J.R. Induction of Decision Trees (in: Machine Learning, 1(1), p. 81-106, 1986)

# Bibliography

- J. Bertin (1983) *Semiology of graphics: diagrams, networks, maps*. University of Wisconsin Press. Originally in French: *Semiologie Graphique*, 1967
- Cairo, A. (2012). *The Functional Art: An introduction to information graphics and visualization*. New Riders.
- Mertens, P., & Meier, M. (2009). *Integrierte Informationsverarbeitung*. Wiesbaden: Gabler.
- Woolman, M. (2002). *Digital information graphics*. Watson-Guptill Publications, Inc..



# Business Intelligence

## 10 Predictive Modeling II

Prof. Dr. Bastian Amberg  
(summer term 2024)

12.6.2024

# Schedule

|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study |                          |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|------------|--------------------------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |            |                          |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |            |                          |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  |            |            |                          |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining Introduction  |            |                          |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |            | Python-Analytics Chap. 1 |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization I            |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam |            | Chap. 2                  |  |  |
|           | W7                | 29.5. | Data Visualization II                               |                                   | 31.5. |   |            |                          |  |  |
|           | W8                | 5.6.  | Data Preparation                                    |                                   | 7.6.  | Predictive Modeling I (10:00 -12:00)  |            | BI-Project Start         |  |  |
|           | W9                | 12.6. | Predictive Modeling II                              |                                   | 14.6. | Python-Analytics-Online Exercise  |            |                          |  |  |
|           | W10               | 19.6. | Guest Lecture Dr. Ionescu                           |                                   | 21.6. | Fitting a Model   |            |                          |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |            |                          |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I                                  |            |                          |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |            |                          |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |            | End                      |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin, 31.7.'24<br>Klausur 2.Termin, 2.10.'24                                      |            | Projektbericht           |  |  |

# Last Lesson

## Predictive Modeling

- (1) Models and induction
- (2) Attribute selection
- (3) Decision Trees

$$G_{ini} = 1 - \sum p_i^2 \text{ Entropy} = \sum p_i \log_2(p_i)$$

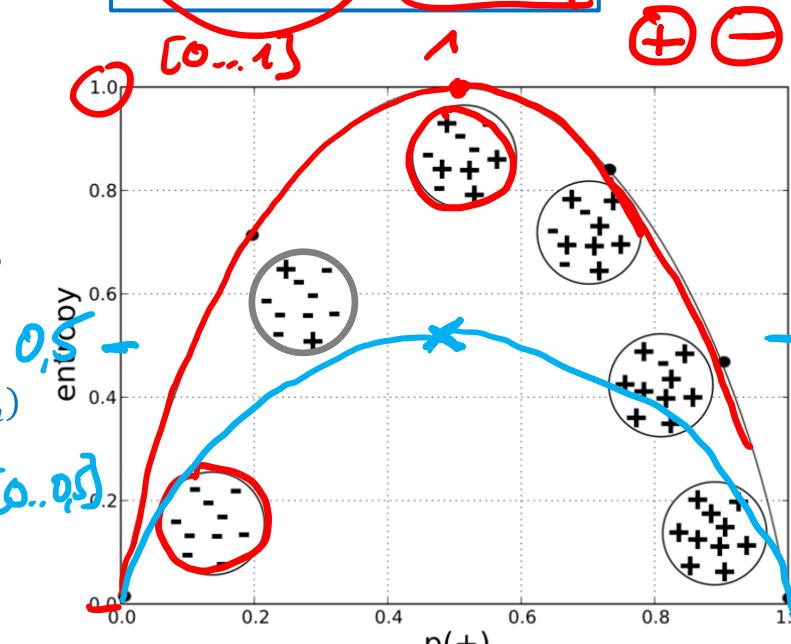
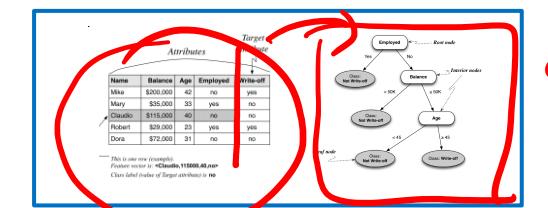
### Entropy

measure of disorder that can be applied to a set

Disorder corresponds to how mixed (impure) a segment is w.r.t. the properties of interest (values of target)

$$\text{entropy} = -p_1 \log_2(p_1) - p_2 \log_2(p_2) - \dots - p_n \log_2(p_n)$$

with  $p_i$  as the relative percentage of property  $i$  within the set, ranging from  $p_i = 0$  to  $p_i = 1$  (all have property  $i$ ).



$$p(-) = 8/10 \quad p(+) = 2/10$$

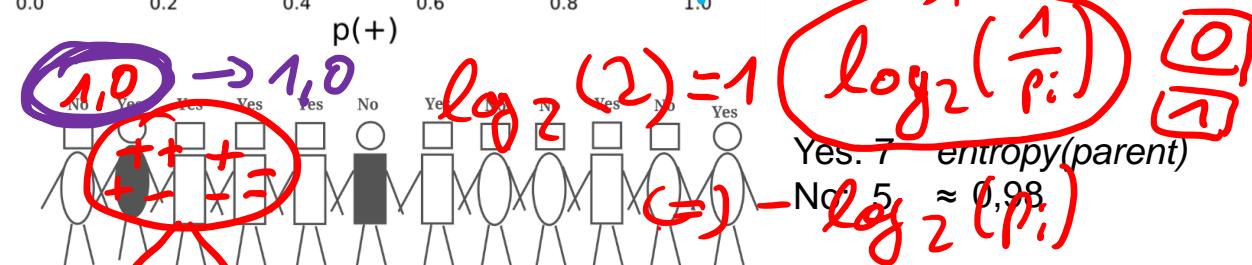
$$\begin{aligned} \text{entropy}(S) &= -[0.8 \times \log_2(0.8) + 0.2 \times \log_2(0.2)] \\ &= -[0.8 \times (-0.32) + 0.2 \times (-2.32)] \\ &\approx 0.72 \end{aligned}$$

$$\begin{aligned} p_i &= 0,5 \\ \frac{1}{p_i} &= 2 \end{aligned}$$

$$\log_2\left(\frac{1}{p_i}\right) = 1$$

$$\text{Yes. } 7 \quad \text{entropy(parent)}$$

$$\text{No. } 5 \approx 0,98$$



### Information Gain:

Measure how much an attribute improves (**decreases**) entropy over the whole segmentation it creates.

IG measures the change in entropy due to any amount of new information added

$$= 1,0 - [0,5 \cdot 0 + 0,5 \cdot 0]$$

$$= 1,0$$

$$IG(\text{parent}, \text{children}) = \text{entropy}(\text{parent})$$

$$(p(c_1) \times \text{entropy}(c_1) + p(c_2) \times \text{entropy}(c_2) + \dots)$$

$$0,5 = 4/8$$



$$4/8 = 0,5$$



Which attribute to choose?

The entropy for each child  $c_i$  is weighted by the proportion of instances belonging to that child

# Exercise – Information gain

$$-\sum p_i \log_2(p_i)$$

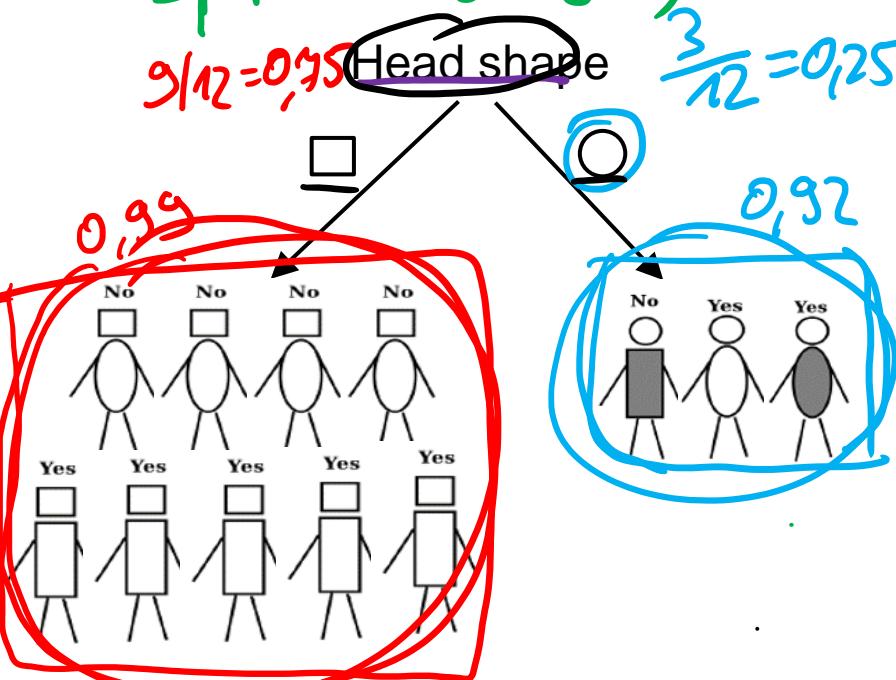
$$\frac{4}{5} \left| \frac{1}{1} \right| \left( -\frac{5}{6} \cdot \log_2(\frac{5}{6}) - \frac{1}{6} \log_2(\frac{1}{6}) \right)$$

Which attribute to choose?

$$\frac{4}{2} \left| \frac{4}{4} \right| \left( -\frac{2}{6} \cdot \log_2(\frac{2}{6}) - \frac{4}{6} \log_2(\frac{4}{6}) \right) = 0,75$$

$$9/12 = 0,75$$

$$\frac{3}{12} = 0,25$$

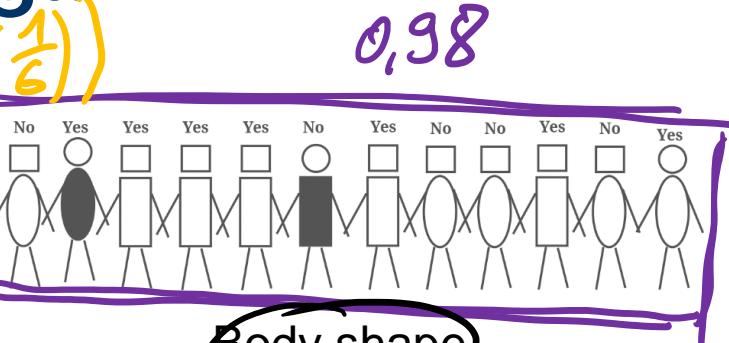


$$\text{entropy}(\square) \approx 0,99 \quad \text{entropy}(\circ) \approx 0,92$$

$$p(\square) \approx 0,75 \quad p(\circ) \approx 0,25$$

$$\text{IG} \approx 0,007$$

$$0,75 \cdot 0,99 - 0,25 \cdot 0,92$$



$$\text{entropy}(\square) \approx 0,65 \quad \text{entropy}(\circ) \approx 0,92$$

$$p(\square) \approx 0,50 \quad p(\circ) \approx 0,50$$

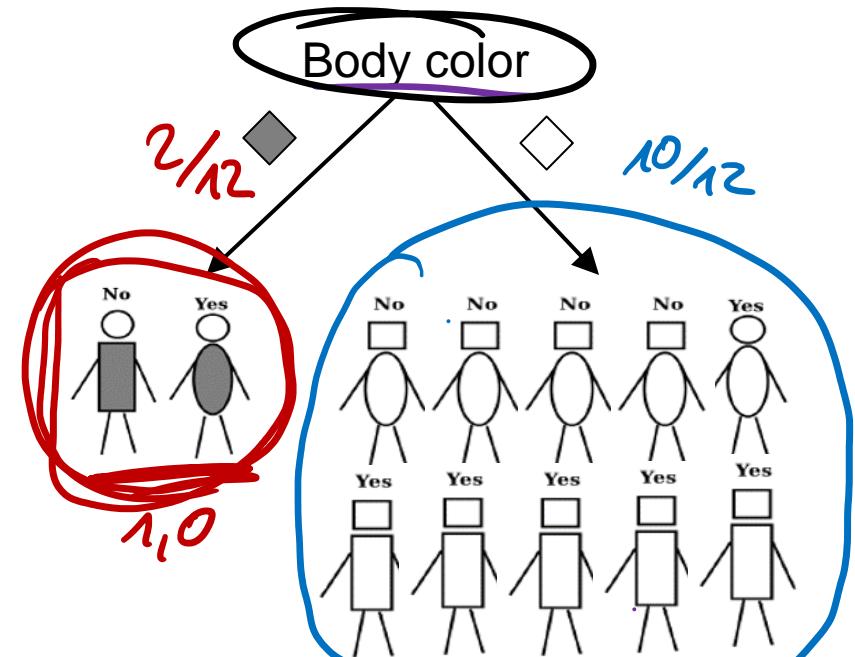
$$\text{IG} \approx 0,196$$

$$0,98 - 0,5 \cdot 0,65 - 0,5 \cdot 0,92$$

$$\frac{4}{6} \left| \frac{1}{1} \right| \left( -\frac{4}{10} \log_2(\frac{4}{10}) - \frac{6}{10} \log_2(\frac{6}{10}) \right)$$

Yes: 7  
No: 5

$$\text{entropy}(\text{parent}) \approx 0,98$$

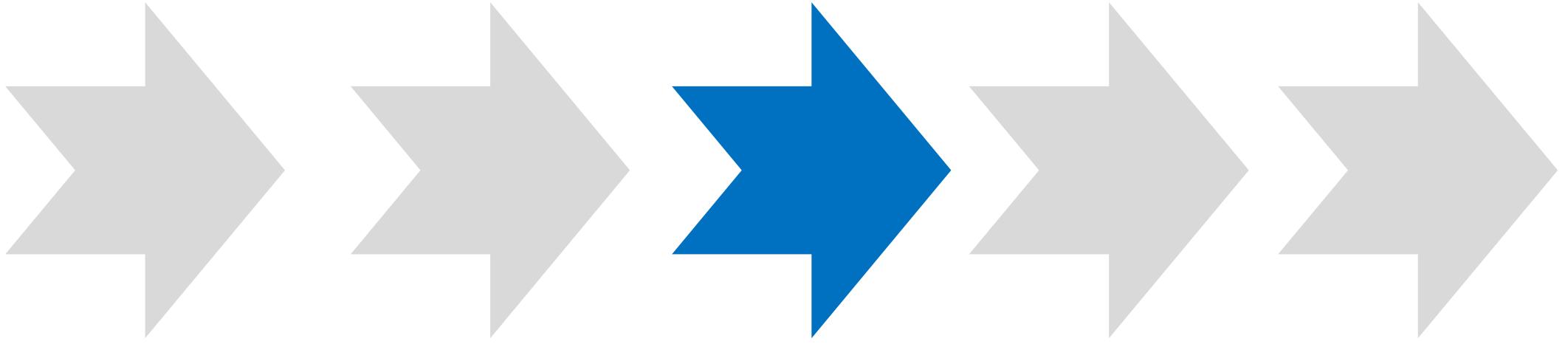


$$\text{entropy}(\diamond) \approx 1,00 \quad \text{entropy}(\square) \approx 0,97$$

$$p(\diamond) \approx 0,117 \quad p(\square) \approx 0,83$$

$$\text{IG} \approx 0,049$$

$$0,98 - 0,117 \cdot 1,0 - 0,83 \cdot 0,97$$



(1) Models and induction

(2) Attribute Selection

(3a) Decision Trees  
Algorithmic View

(3b) Probability Estimation

(3c) Decision Tree Examples

# Decision Trees

If we select multiple attributes each giving some information gain, it's not clear how to put them together → **decision trees**

The tree creates a segmentation of the data

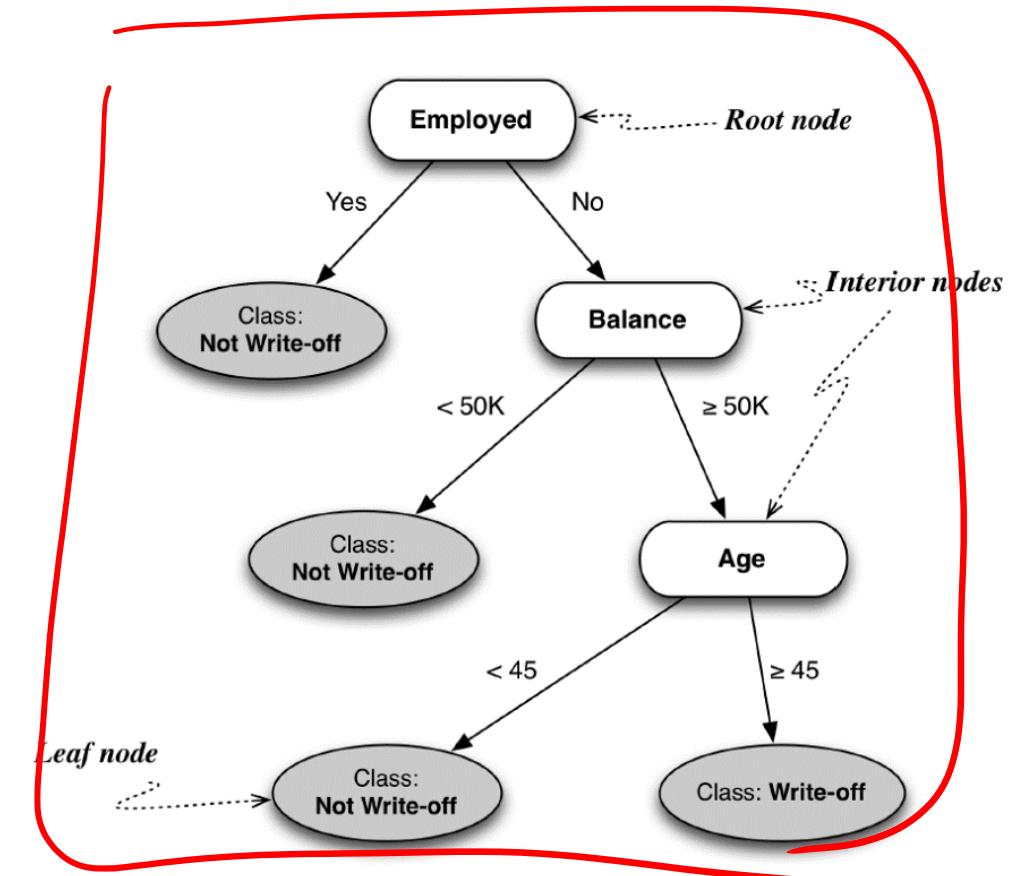
Each *node* in the tree contains a test of an attribute

Each *path* eventually terminates at a *leaf*

Each leaf corresponds to a *segment*,  
and the attributes and values along  
the path give the characteristics

Each leaf contains a value for the  
target variable

Decision trees are often used as **predictive models**



→ Prediction of target attribute, e.g.  
Michi, 40,000, 24, yes, Write-off?  
Micki, 115,000, 40, no, Write-off?

# How to build a decision tree

if `bodyShape = "Rectangular"`  
`!bodyColor = "Black" => NO`

Manually build the tree deductively, based on expert knowledge

- very time-consuming
- trees are sometimes corrupt (redundancy, contradictions, non-completeness, inefficient)

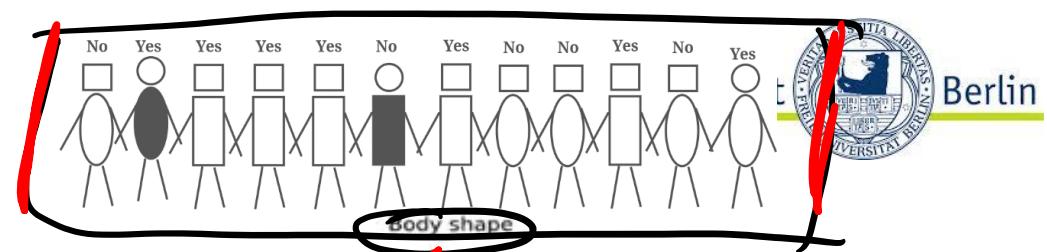
Build the tree **automatically** by induction

- recursively partition the instances based on their attributes (divide-and-conquer)
- easy to understand
- relatively efficient

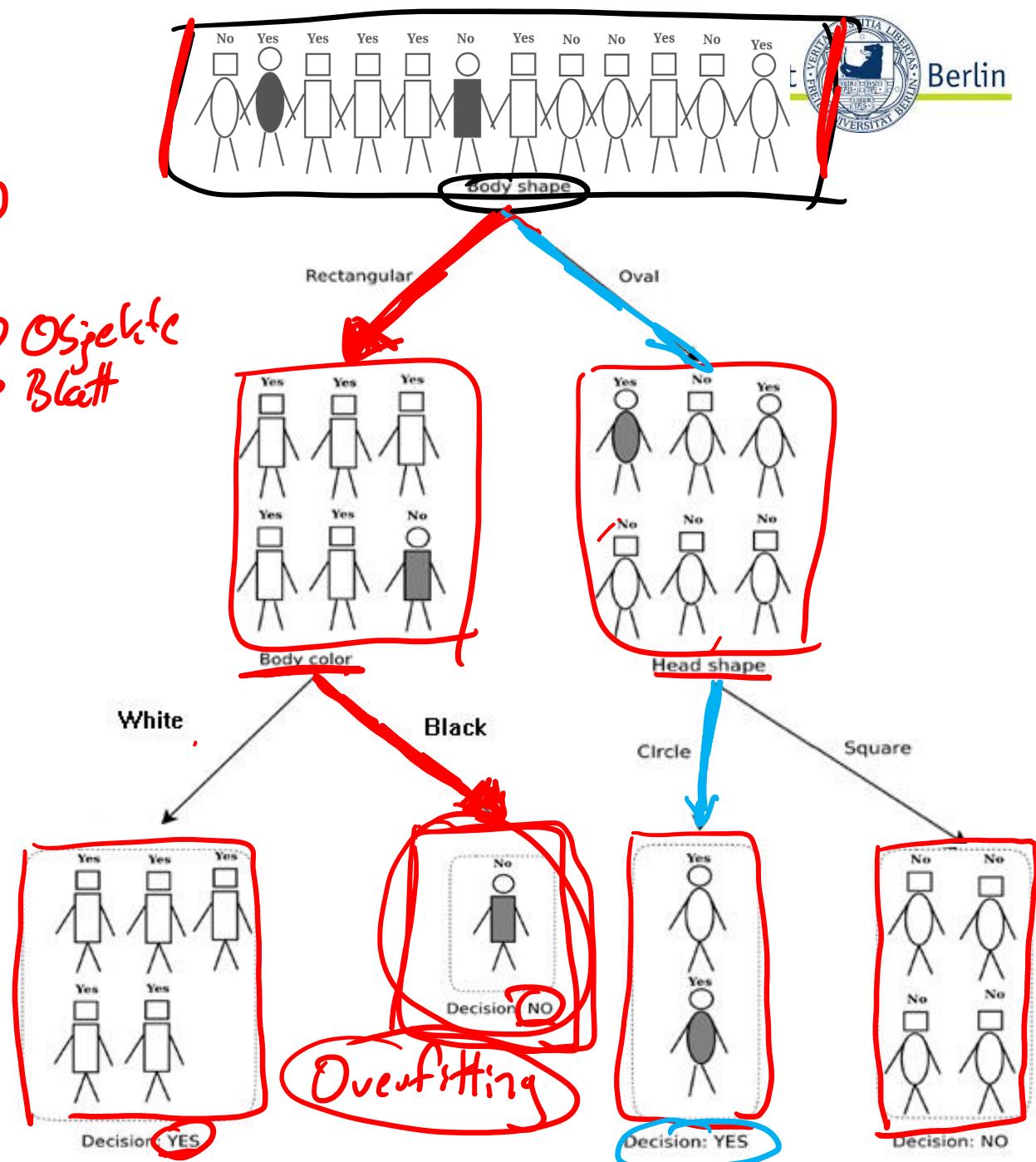
Recursively apply **attribute selection** to find the best attribute to partition the data set

The goal at each step is to select an attribute to partition the current group into subgroups that are as pure as possible w.r.t. the target variable

Ref.



min 10 Objekte pro Blatt

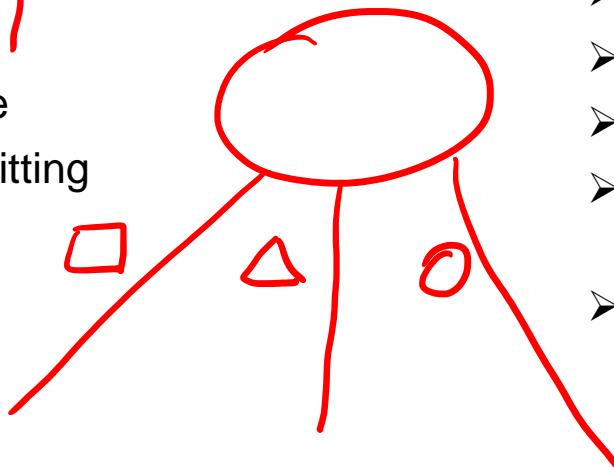


# ID3 – an algorithm for tree induction

0,0 [0,0,Δ] [0...1,0] [1,0..140]  
Körpergröße?  
[0 - 2,5m]

Iterative **Dichotomiser (ID3)** is the most used machine learning algorithm in scientific literature and in commercial systems

- Invented by Ross Quinlan in the early 1980s
- Uses information gain as a measure for the quality of partitioning
- Requires categorical variables
- Has no features of handling noise
- Has no features of avoiding overfitting



ID3 encompasses a **top-down** decision tree building process

Basic step: split sets into disjoint subsets, where all the *individuals within a subset show the same value for a selected attribute*

- One attribute for testing is selected at the **current node**
- Testing separates the set into partitions
- For each partition, a **subtree** is built
- Subtree-building is terminated if all the samples in one partition belong to the same class
- Labels the tree leaves with the corresponding class

Other algorithms, e.g. **CART, CHAID, C4.5**  
We will use CART in the online-exercise.  
(see <https://scikit-learn.org/stable/modules/tree.html#tree-algorithms>)

# ID3: algorithmic structure

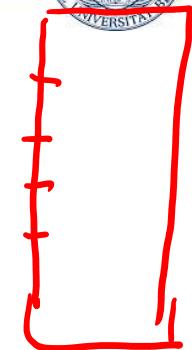
ID3



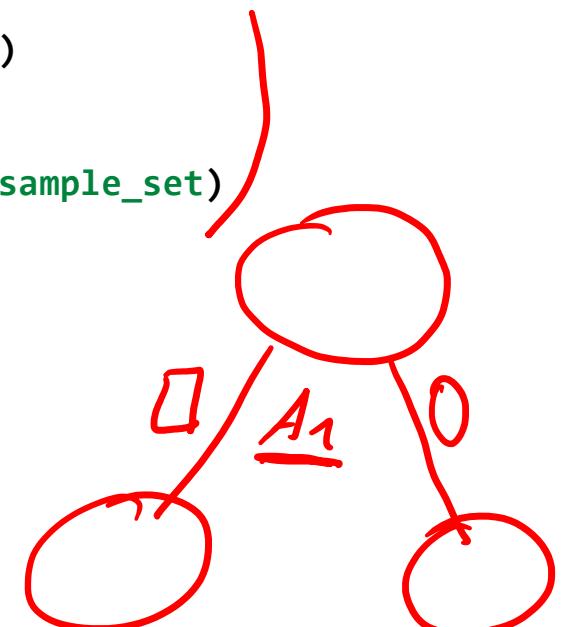
CART

An **intelligent order of the tests** is the key feature of the ID3 algorithm ( $\rightarrow$  information gain)

Recursive algorithmic structure of ID3:

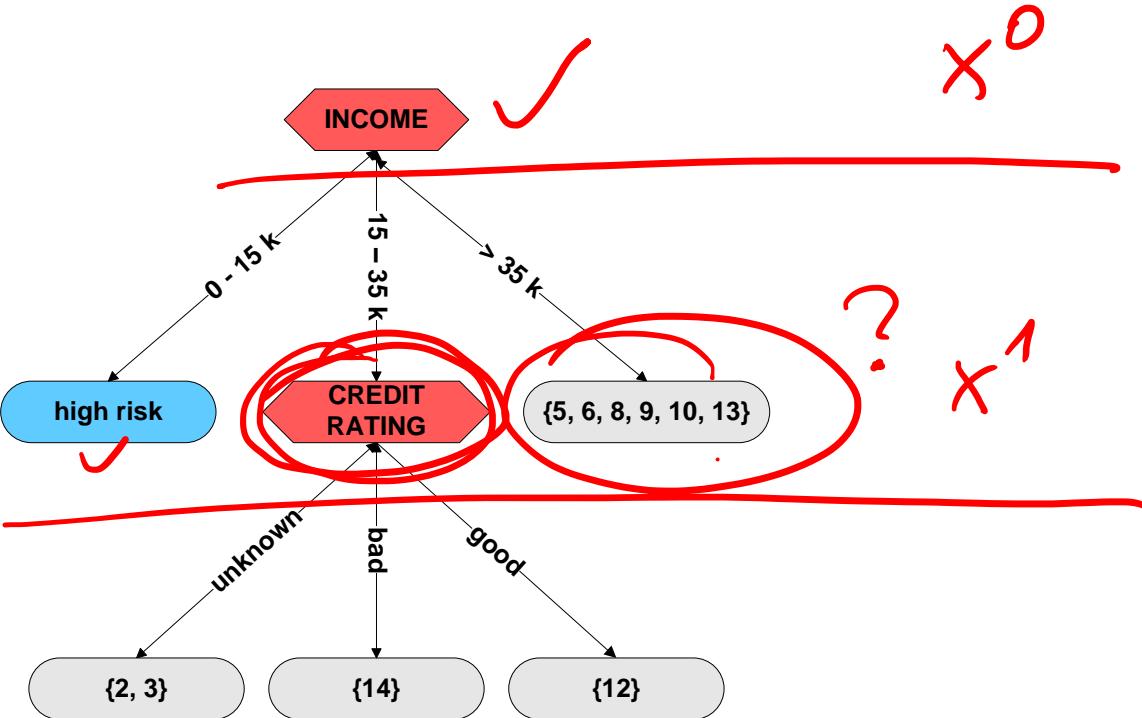
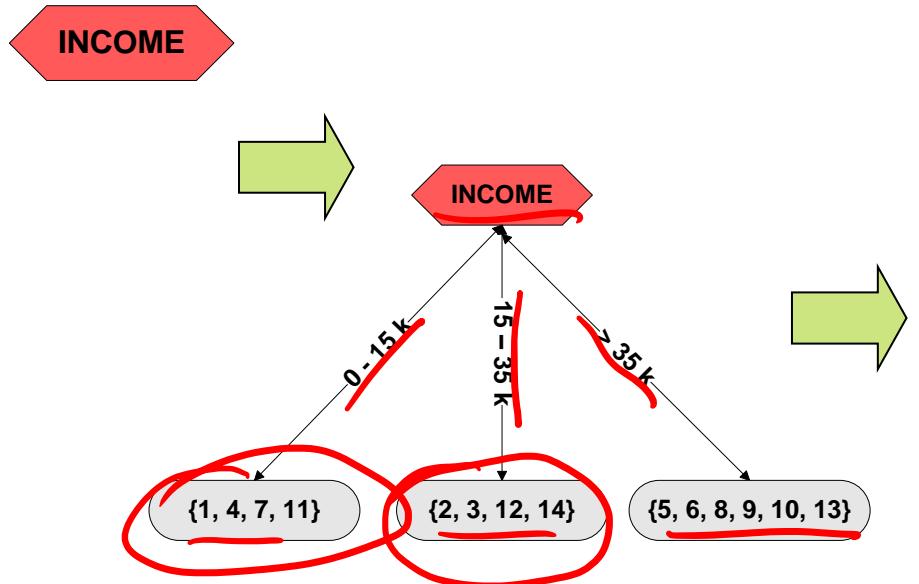


```
FUNCTION induce_tree(sample_set, attributes)
BEGIN
    IF (all the individuals from sample_set belong to the same class)
        RETURN (leave bearing the corresponding class label)
    ELSEIF (attributes is empty)
        RETURN (leave bearing a label of all the classes present in sample_set)
    ELSE
        1. select an attribute X_j SCORE FUNCTION Entropy bzw IG
        make X_j the root of the current tree
        delete X_j from attributes
        FOR EACH (value V of X_j)
            construct a branch, labeled V
            V_partition = sample individuals for which X_j == V
            induce_tree(V_partition, attributes)
END
```



# ID3: example steps

$$\begin{aligned}
 X^0 &= \{\text{INCOME}, \text{CREDIT RATING}, \text{SECURITIES}, \text{LEVEL OF DEBT}\} \\
 \rightarrow X^1 &= \{\text{CREDIT RATING}, \text{SECURITIES}, \text{LEVEL OF DEBT}\} \\
 X^2 &= \{\text{SECURITIES}, \text{LEVEL OF DEBT}\}
 \end{aligned}$$



# ID3: simplicity



ID3 tries to build a tree, that

- classifies all the training samples correctly and
- provides a high probability for implying only a small number of tests when classifying an individual

Criterion for attribute selection:

- prefer attributes which contribute a **bigger amount of information** to the classification of an individual
- Information content is measured according to entropy



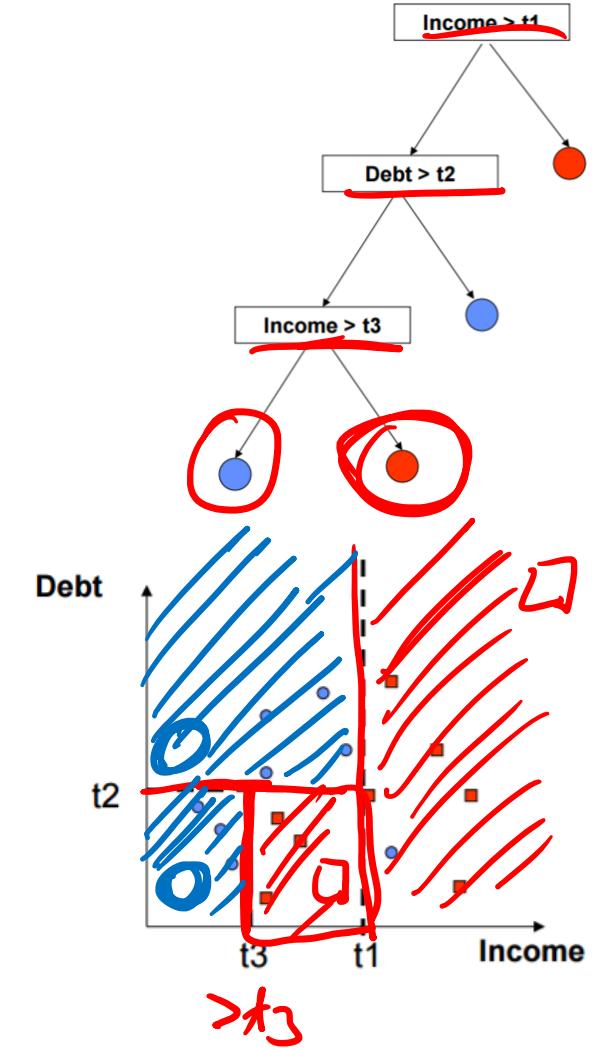
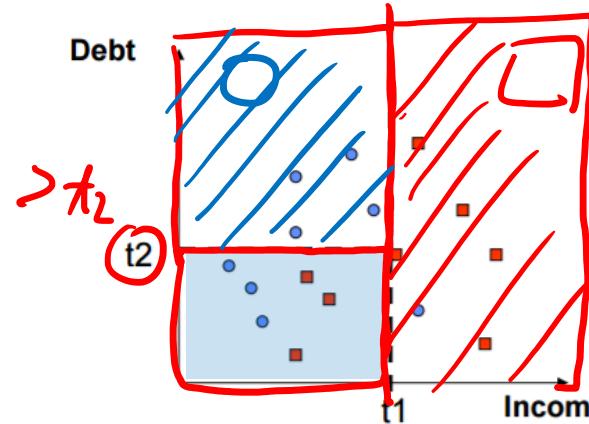
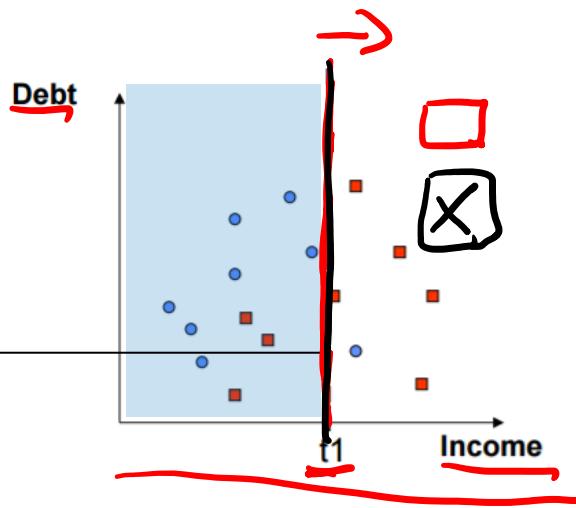
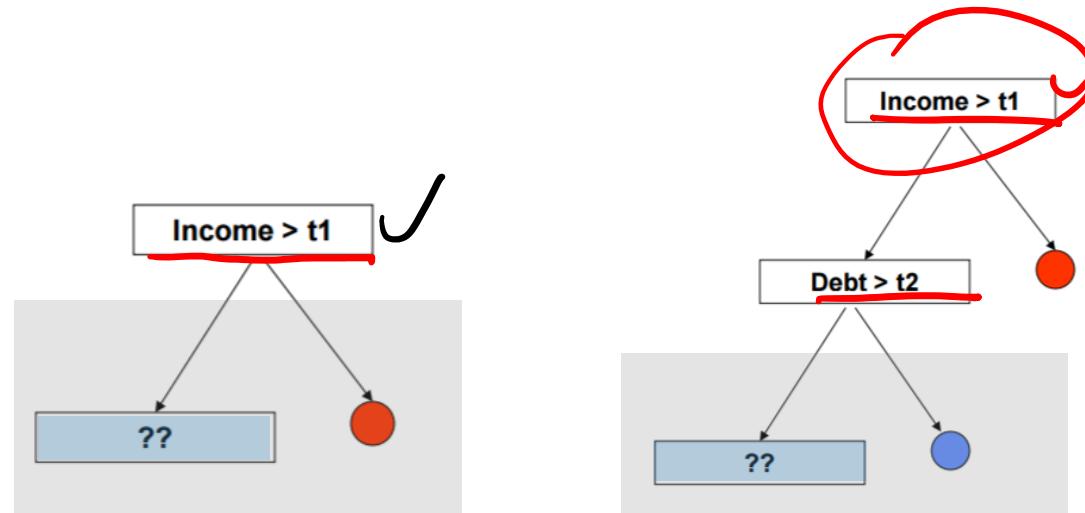
“When you hear hoofbeats, think of horses not zebras” (Sotos, 1991)

## Occam's razor:

the simplest tree produces the smallest classification error rates, when applying the tree to new individuals

# Nonlinearity and Decision Surface

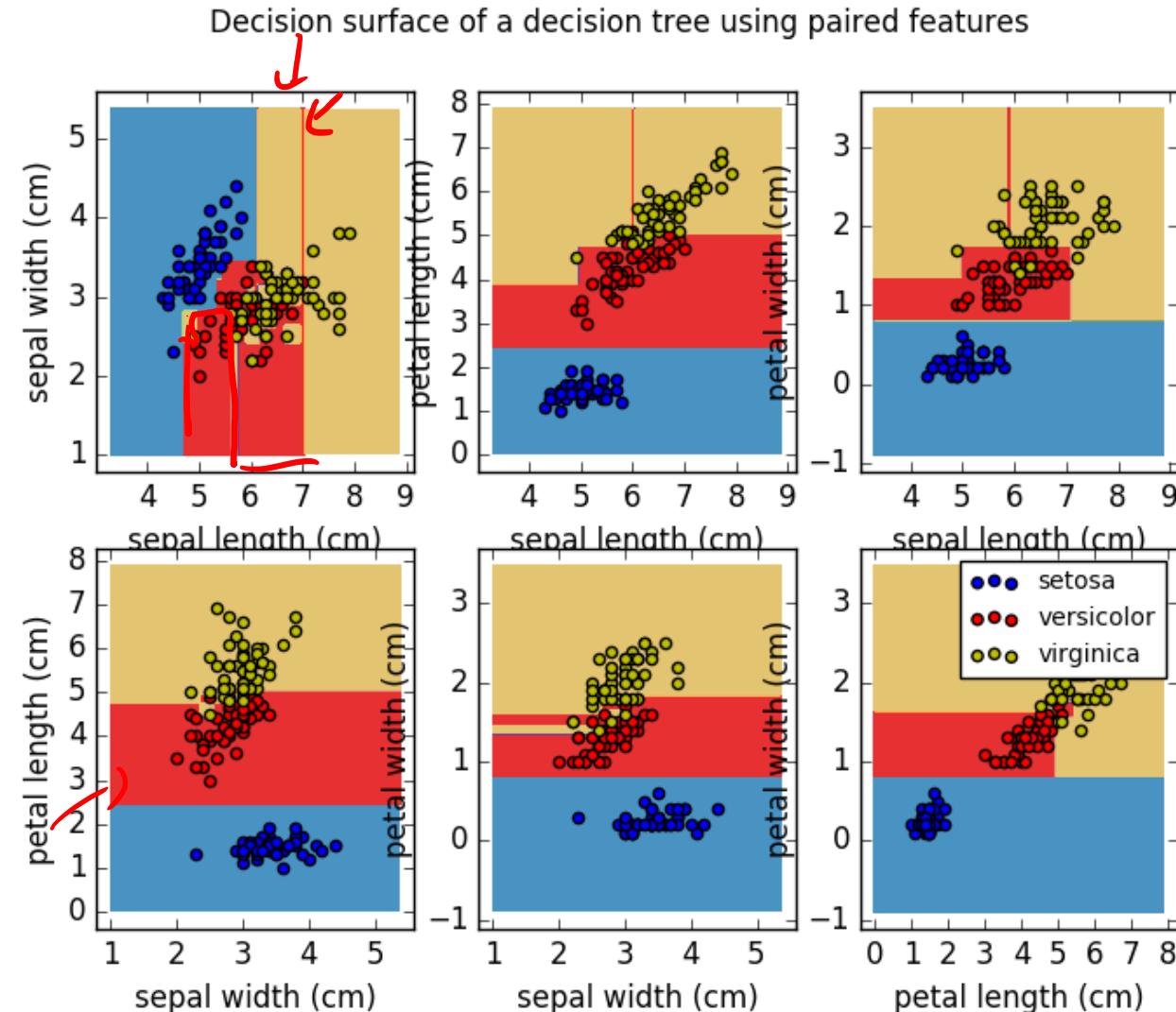
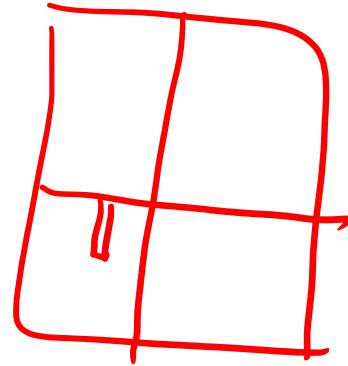
Example – Using two attributes to visualize segmentations

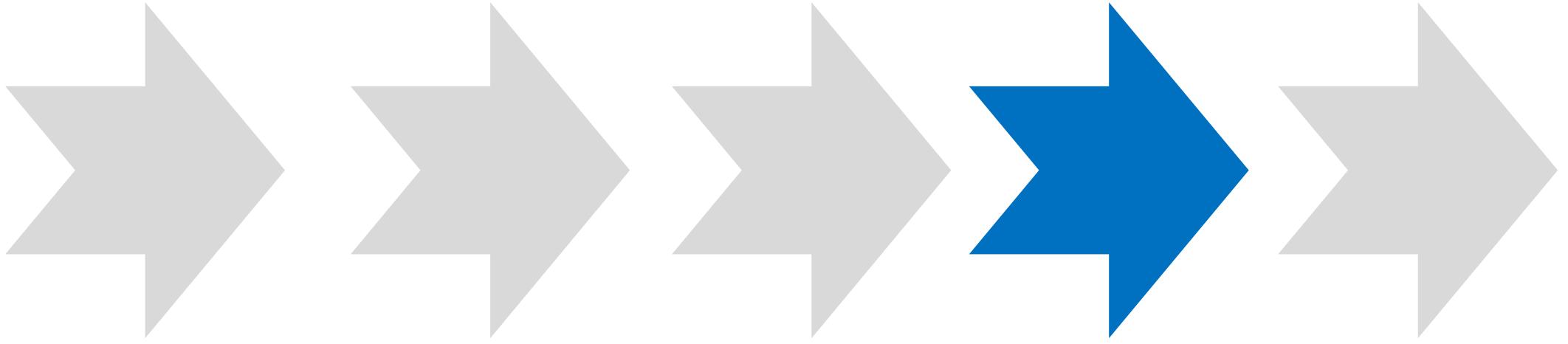


# Nonlinearity and Decision Surface

Example - Iris data

Decision Trees model  
non-linear relationships  
between attributes





(1) Models and induction

(2) Attribute selection

(3a) Decision Trees  
Algorithmic View

(3b) Probability Estimation

(3c) Decision Tree Examples

# Probability estimation (1/3)



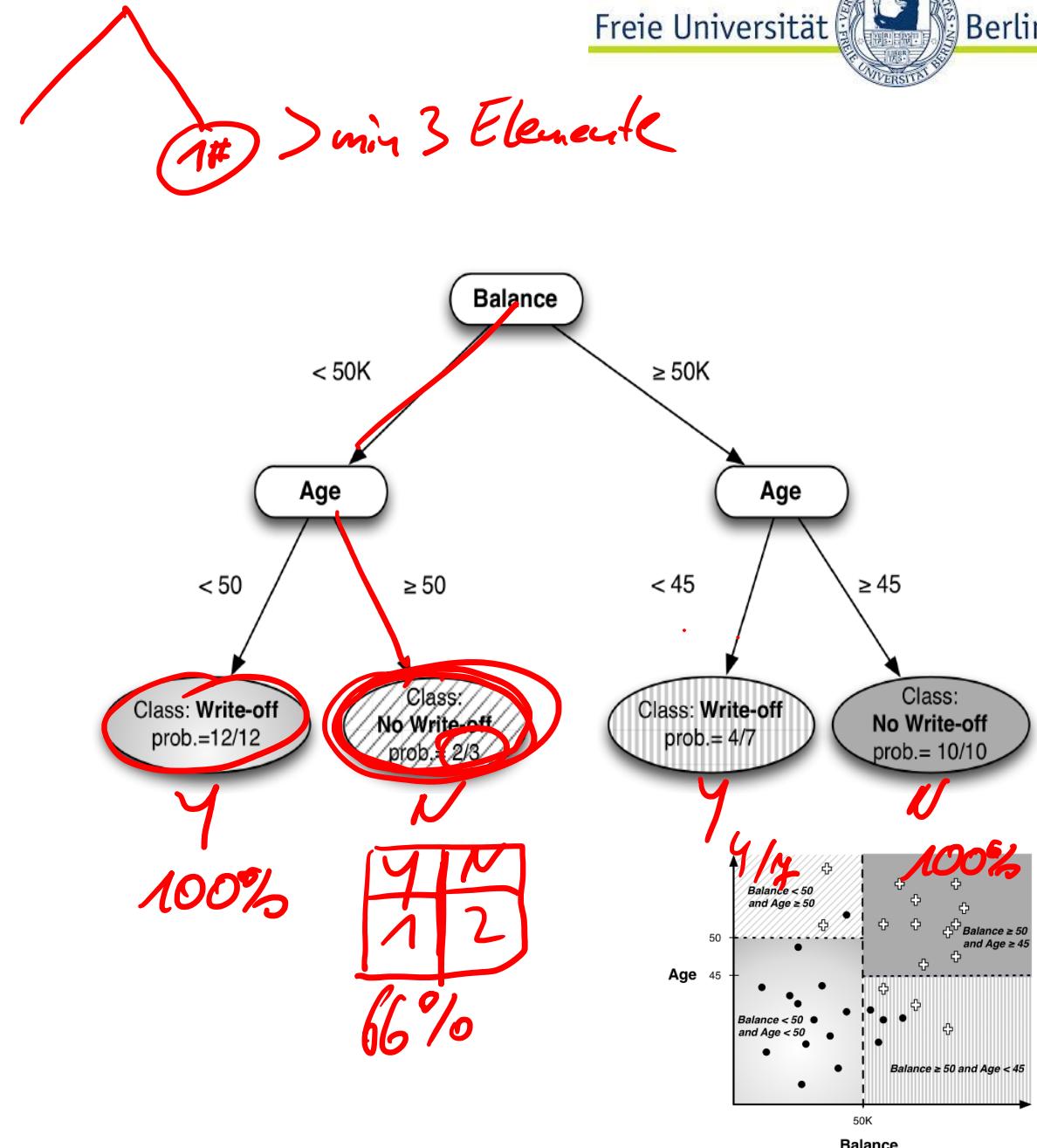
We often need a **more informative prediction** than just a classification

- E.g. allocate your budget to the instances with the highest expected loss
- More sophisticated decision-making process

Classification may oversimplify the problem

- E.g. if all segments have a probability of  $<0.5$  for write-off, every leaf will be labeled „not write-off“
- We would like each segment (leaf) to be assigned an **estimate of the probability of membership** in the different classes

## Probability estimation tree



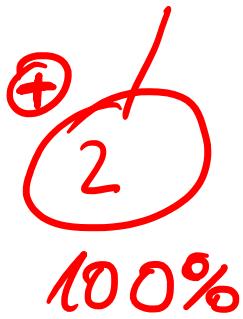
# Probability estimation (2/3)

Tree induction can easily produce probability estimation trees instead of simple classification trees

- **Instance counts** at each leaf provide class probability estimates
- **Frequency-based estimate** of class membership:  
if a leaf contains  $n$  positive and  $m$  negative instances, the probability of any new instance being positive may be estimated as  $n/(n + m)$ .

Approach may be too optimistic for segments with a very small number of instances ( $\rightarrow$  overfitting)

- Smoothed version of frequency-based estimate by **Laplace correction**, which moderates the influence of leaves with only a few instances:  $p(c) = \frac{n+1}{n+m+2}$  with  $n$  as number of instances that belong to class  $c$  and  $m$  instances not belonging to class  $c$



Example 1

$$\begin{aligned} & \text{+ } n \quad \text{- } m \\ & n=2 \quad m=0 \\ & p(c) = \frac{n}{n+m} = \frac{2}{2+0} = 1 \quad 100\% \end{aligned}$$

$$\begin{aligned} & n=2 \quad m=0 \\ & p_{Laplace}(c) = \frac{n+1}{n+m+2} = \frac{2+1}{2+0+2} = 0,75 \quad 75\% \end{aligned}$$

Example 2

$$\begin{aligned} & n=20 \quad m=0 \\ & p(c) = \frac{n+1}{n+m+2} = \frac{20+1}{20+0+2} = 1 \quad 100\% \\ & p_{Laplace}(c) = \frac{n+1}{n+m+2} = \frac{20+1}{20+0+2} = 0,95 \quad 95\% \end{aligned}$$

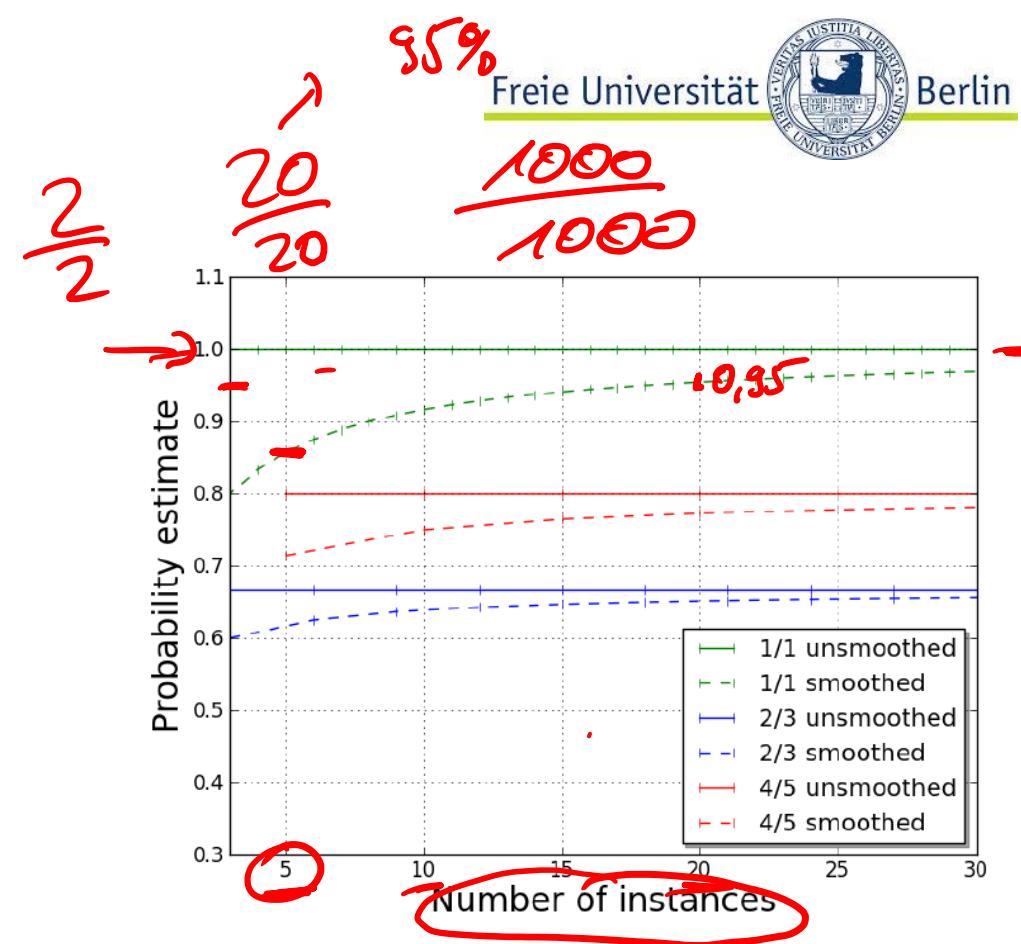
# Probability estimation (3/3)

Effect of Laplace correction on several class ratios as the number of instances increases (2/3, 4/5, 1/1)

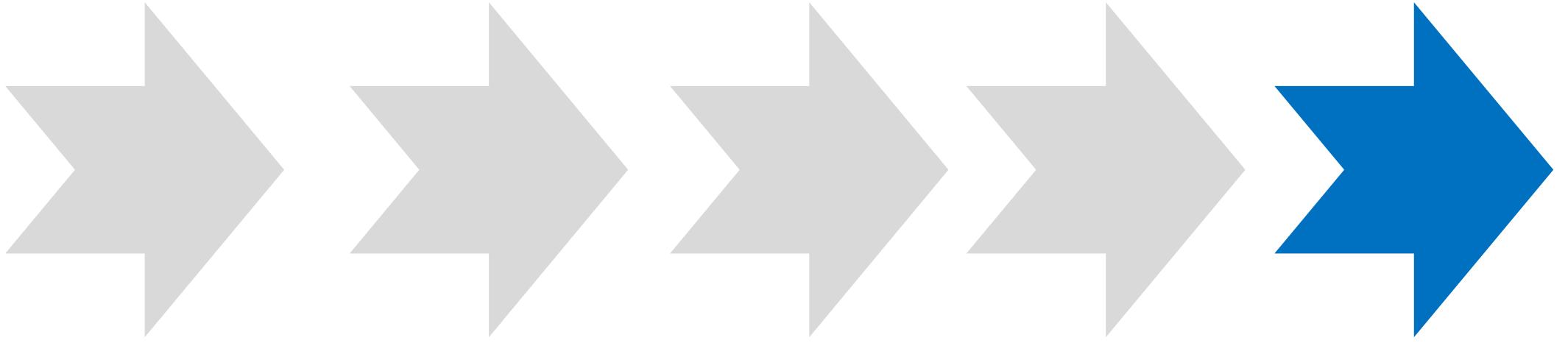
Example:

A leaf of the classification tree that has 2 pos. instances and no negative instances would produce the same f-b estimate ( $p = 1$ ) as a leaf node with 20 pos. and no negatives.

The Laplace correction smooths the estimate of the first leaf down to  $p = 0.75$  to reflect this uncertainty, but it has much less effect on the leaf with 20 instances ( $p \approx 0.95$ )



$$\frac{5+1}{5+2} = \frac{6}{7} \times 0,95$$



(1) Models and induction

(2) Attribute selection

(3a) Decision Trees  
Algorithmic View

(3b) Probability Estimation

(3c) Decision Tree Examples

# Example - The Churn Problem

Solve the churn problem by tree induction

Historical data set of 20,000 customers

Each customer either had stayed with the company or left

Customers are described by the following variables:



O<sub>2</sub>  
e-plus<sup>+</sup>

T-Mobile

BASE

| Variable                | Explanation  |
|-------------------------|--|
| COLLEGE                 | Is the customer college educated?                        |
| INCOME                  | Annual <u>income</u>                                     |
| OVERAGE                 | Average overcharges per month                            |
| LEFTOVER                | Average number of leftover <u>minutes per month</u>      |
| HOUSE                   | Estimated value of dwelling ( <u>from census tract</u> ) |
| HANDSET_PRICE           | Cost of phone  |
| LONG_CALLS_PER_MONTH    | Average number of long calls (15 mins or over) per month |
| AVERAGE_CALL_DURATION   | Average duration of a call                               |
| REPORTED_SATISFACTION   | Reported level of satisfaction                           |
| REPORTED_USAGE_LEVEL    | Self-reported usage level                                |
| LEAVE (Target variable) | <i>Did the customer stay or leave (churn)?</i>           |

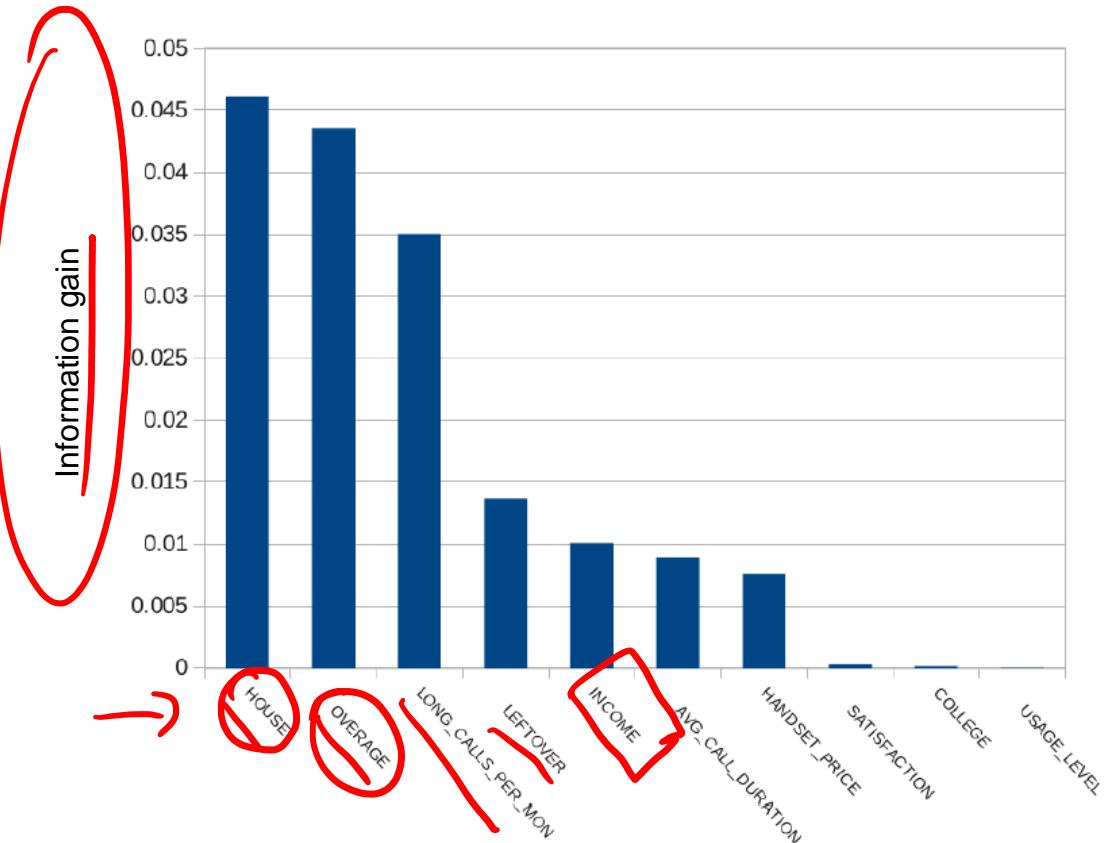
We want to use this data to **predict which new customers are going to churn.**

# Example - The Churn Problem

**How good are each of these variables individually?**

Measure the information gain of each variable

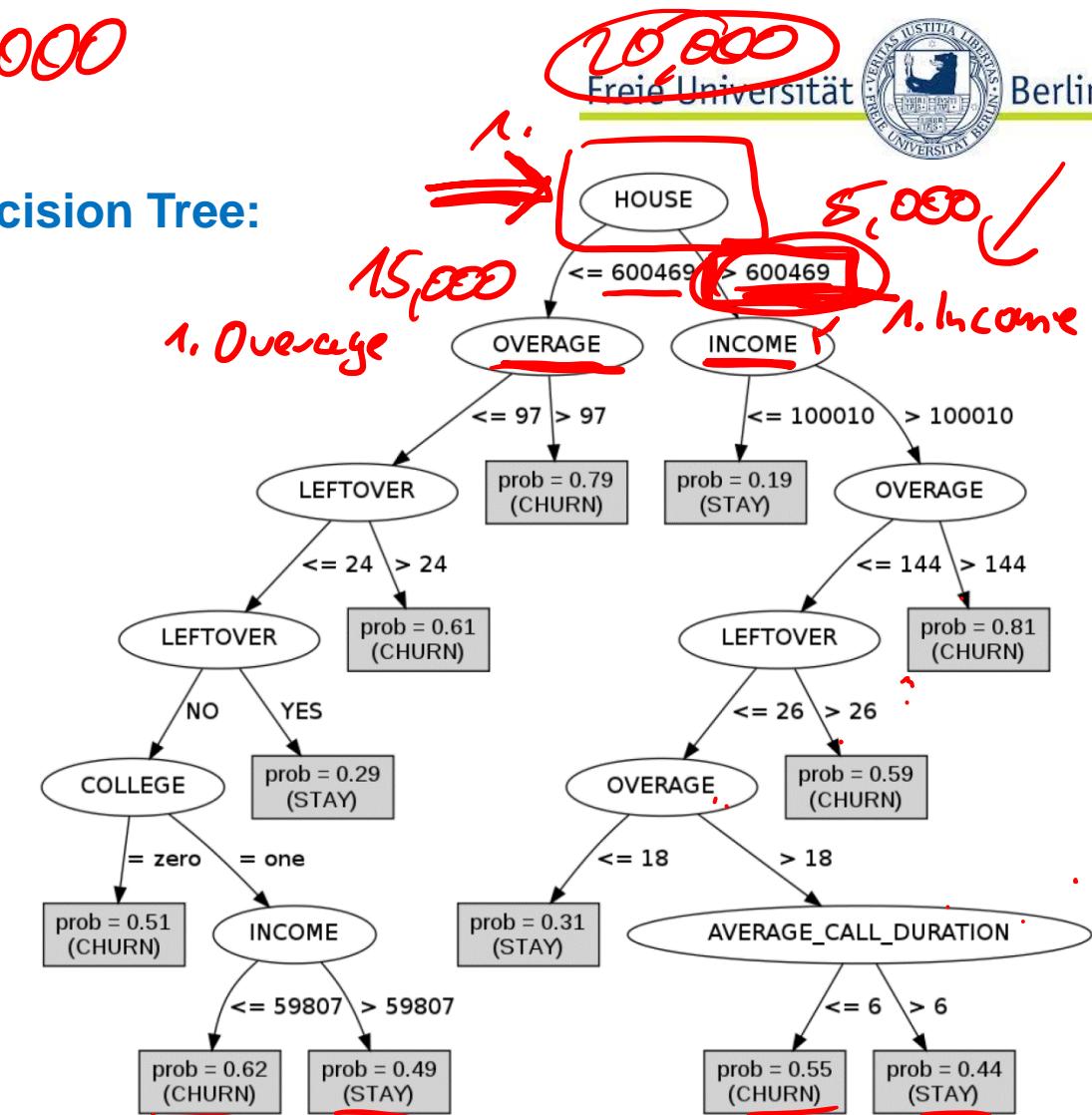
Compute information gain for each variable independently



The highest information gain feature (HOUSE) is at the root of the tree.

20,000

# Decision Tree:



1. Why is the order of features chosen for the tree different from the ranking?
  2. When to stop building the tree?
  3. How do we know that this is a good model?

# Example - Decision Trees with Python

See [Python-Analytics online exercise](#) on Friday, 14.6.

If possible, prepare your system for the exercise:

**Achtung:** Die Pakete pandas und sklearn sind in der Regel schon vorinstalliert. Die Pakete graphviz und pydotplus müssen hingegen zunächst außerhalb von Jupyter Notebooks installiert werden, bevor man sie importieren kann. Öffne hierzu die Kommandozeile (Windows) oder das Terminal (Mac) und führe folgendes nacheinander aus:

1. conda install python-graphviz
2. conda install -c conda-forge pydotplus

Installationen direkt aus Jupyter Notebooks heraus oder mit pip führen in der Regel zu Fehlern!

These components only need to be used for a small part of the exercise.

Example



# Outlook: Tree Induction vs. Fitting a Model

Next Lesson

Freie Universität Berlin

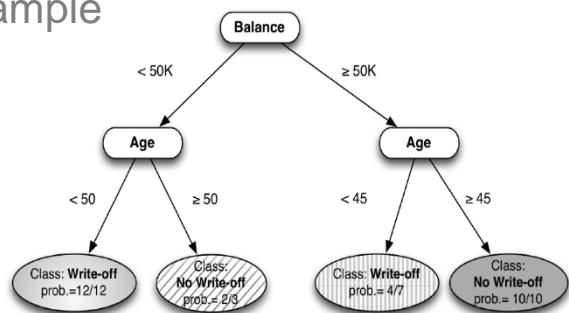


## Classification via Tree Induction

So far:

- ✓ we produced both the **structure of the model** (the particular tree model) and the **numeric parameters** of the model from the data

For example



Questions answered:

- ✓ How do we decide to classify data?
- ✓ Why do we not build „complete“ trees?
- ✓ If we have incomplete trees, we want to assess probabilities. What do we take into consideration?

Ref.

## Classification via Mathematical Functions

Now:

- We **specify the structure of the model**, but leave certain numeric parameters unspecified
- Data Mining calculates the best parameter values given a particular set of training data
- The form of the model and the attributes is specified
- The goal of DM is to tune the parameters so that the model fits the data as good as possible (**parameter learning**)

Simplifying assumptions:

- For classification and class probability estimation, we will consider **only binary classes**.
- We assume that **all attributes are numeric**.  
    (→ see data preparation)
- We **ignore the need to normalize numeric measurements to a common scale** (→ see data preparation)

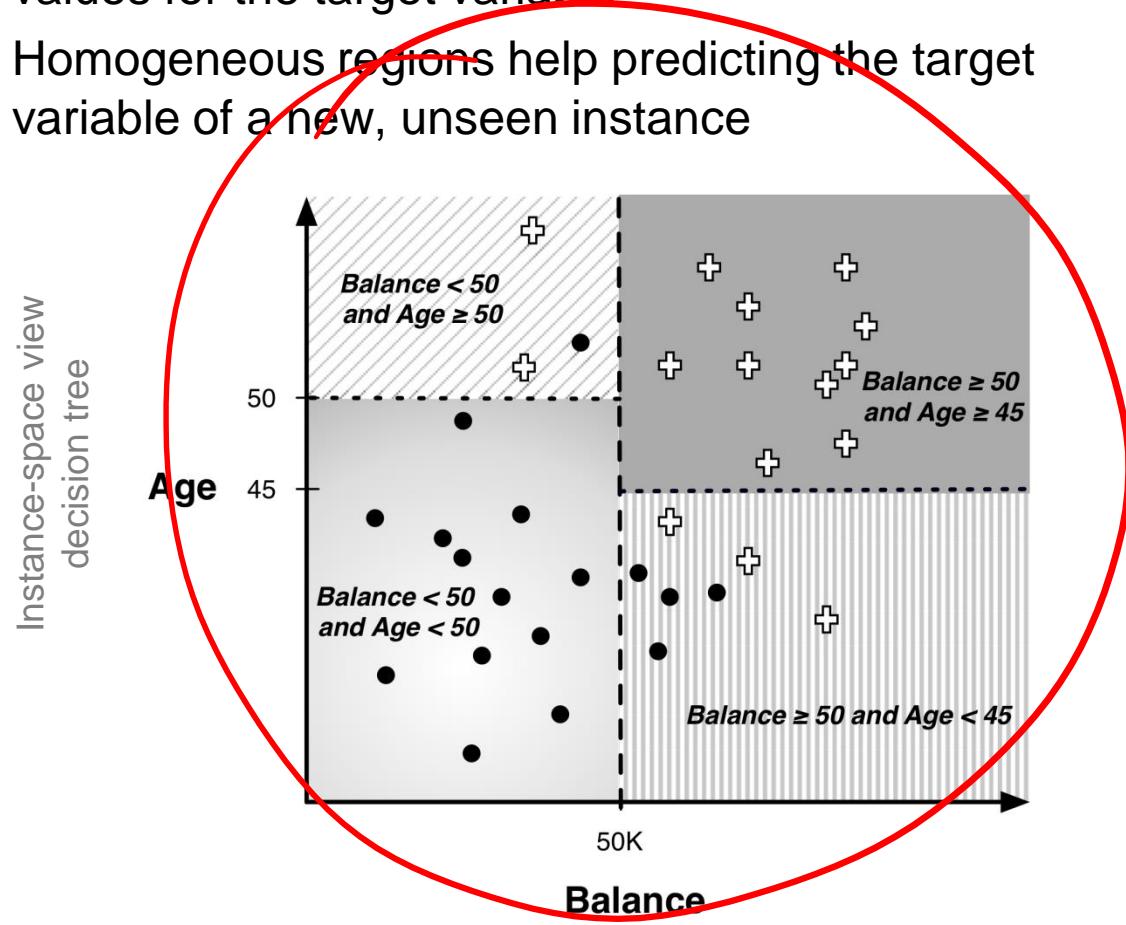
# Linear classifiers

[Next Lesson](#)

## Instance-space view:

shows the space broken up into regions by decision boundaries

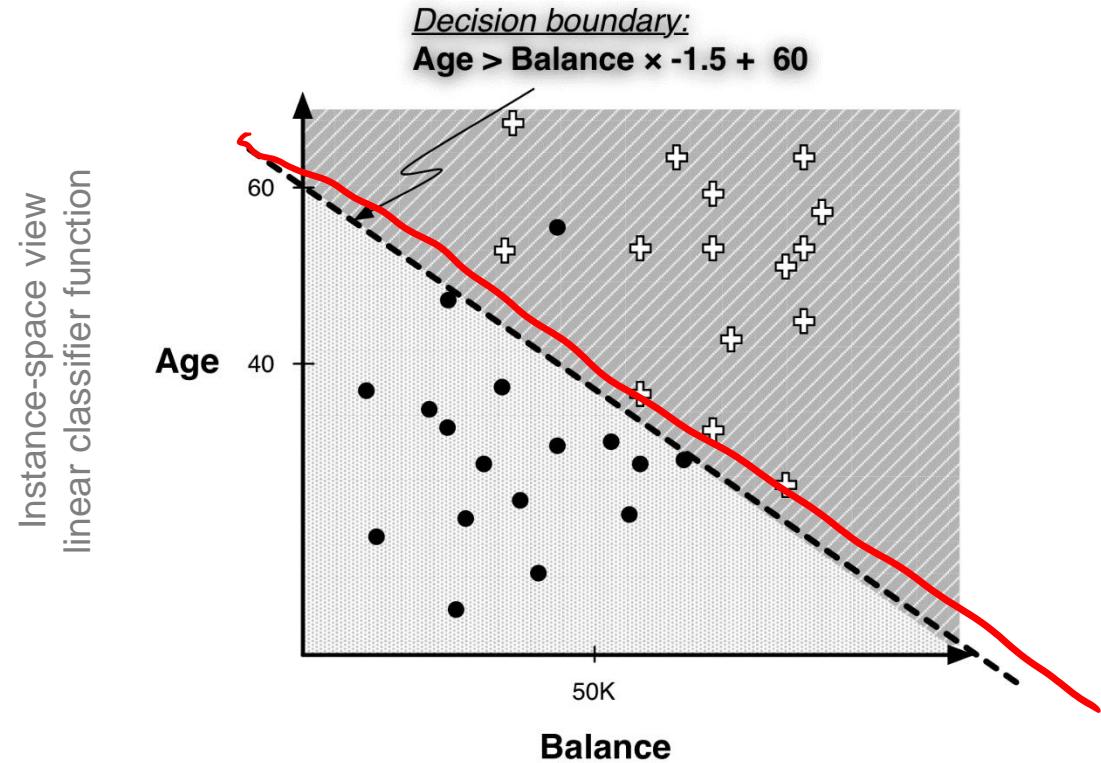
- Examples in each space should have similar values for the target variable
- Homogeneous regions help predicting the target variable of a new, unseen instance



Ref.

We can separate the instances almost perfectly (by class) if we are allowed to introduce a boundary that is still a straight line, but is not perpendicular to the axes

- Linear classifier



## Fragen?

- ✓ Predictive modeling
  - ✓ Decision trees – Introduction
  - ✓ Decision trees – algorithmic view
  - ✓ Probability estimation tree
  - ✓ Decision trees – examples

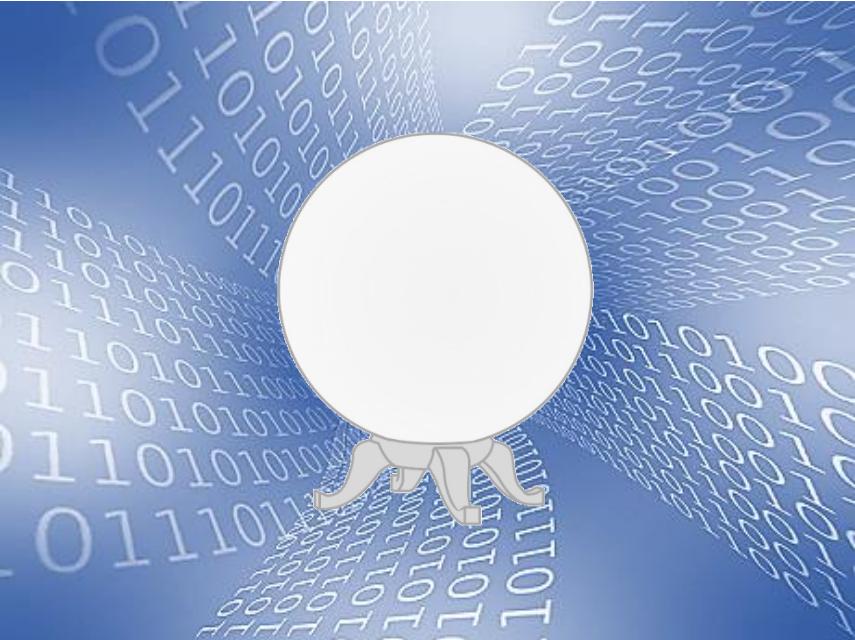
# Recommended reading



- Provost, F., Data Science for Business  
Fawcett, T. Chapter 3
- Berthold et al. Guide to Intelligent Data Analysis  
Chapter 8.1
- Hand, D. Principles of Data Mining  
Chapter 10
- Quinlan, J.R. Induction of Decision Trees (in: Machine Learning, 1(1), p. 81-106, 1986)

# Bibliography

- J. Bertin (1983) *Semiology of graphics: diagrams, networks, maps*. University of Wisconsin Press. Originally in French: *Semiologie Graphique*, 1967
- Cairo, A. (2012). *The Functional Art: An introduction to information graphics and visualization*. New Riders.
- Mertens, P., & Meier, M. (2009). *Integrierte Informationsverarbeitung*. Wiesbaden: Gabler.
- Woolman, M. (2002). *Digital information graphics*. Watson-Guptill Publications, Inc..



# Business Intelligence

## 10 Predictive Modeling II

Prof. Dr. Bastian Amberg  
(summer term 2024)

12.6.2024

# Schedule

|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study |                  |               |         |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|------------|------------------|---------------|---------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |            |                  | Python-Basics | Chap. 1 |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |            |                  |               | Chap. 2 |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  |            |            |                  |               | Chap. 3 |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining Introduction  |            |                  |               |         |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |            | Python-Analytics | Chap. 1       |         |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization I            |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam |            |                  | Chap. 2       |         |  |  |
|           | W7                | 29.5. | Data Visualization II                               |                                   | 31.5. |   |            |                  |               |         |  |  |
|           | W8                | 5.6.  | Data Preparation                                    |                                   | 7.6.  | Predictive Modeling I (10:00 -12:00)  |            | BI-Project       | Start         |         |  |  |
|           | W9                | 12.6. | Predictive Modeling II                              |                                   | 14.6. | Python-Analytics-Online Exercise  |            |                  |               |         |  |  |
|           | W10               | 19.6. | Guest Lecture Dr. Ionescu                           |                                   | 21.6. | Fitting a Model   |            |                  |               |         |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |            |                  |               |         |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I                                  |            |                  |               |         |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |            |                  |               |         |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |            |                  | End           |         |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin, 31.7.'24<br>Klausur 2.Termin, 2.10.'24                                      |            | Projektbericht   |               |         |  |  |

# Last Lesson

## Predictive Modeling

- (1) Models and induction
- (2) Attribute selection
- (3) Decision Trees

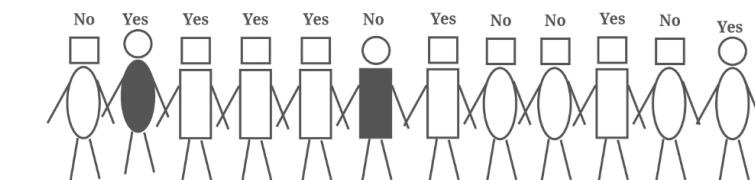
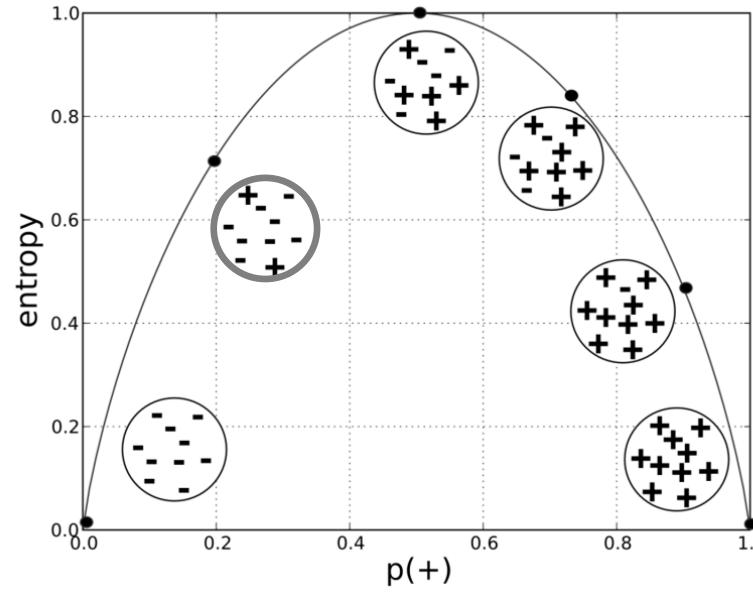
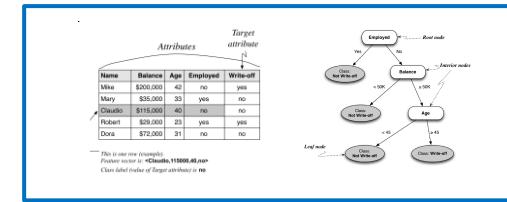
### Entropy:

**measure of disorder** that can be applied to a set

Disorder corresponds to how mixed (impure) a segment is w.r.t. the properties of interest (values of target)

$$\text{entropy} = -p_1 \log_2(p_1) - p_2 \log_2(p_2) - \cdots - p_n \log_2(p_n)$$

with  $p_i$  as the relative percentage of property  $i$  within the set, ranging from  $p_i = 0$  to  $p_i = 1$  (all have property  $i$ ).



$$p(-) = 8/10 \quad p(+) = 2/10$$

$$\begin{aligned} \text{entropy}(S) &= -[0.8 \times \log_2(0.8) + 0.2 \times \log_2(0.2)] \\ &= -[0.8 \times (-0.32) + 0.2 \times (-2.32)] \\ &\approx 0.72 \end{aligned}$$

$$\begin{aligned} \text{Yes: 7} \quad \text{entropy(parent)} \\ \text{No: 5} \quad \approx 0.98 \end{aligned}$$

### Information Gain:

Measure how much an attribute improves (**decreases**) entropy over the whole segmentation it creates.

IG measures the change in entropy due to any amount of new information added

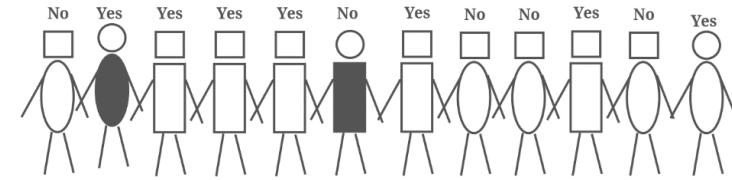
$$\begin{aligned} \text{IG}(\text{parent}, \text{children}) &= \text{entropy}(\text{parent}) \\ &- [p(c_1) \times \text{entropy}(c_1) + p(c_2) \times \text{entropy}(c_2) + \cdots] \end{aligned}$$

The entropy for each child  $c_i$  is weighted by **the proportion of instances** belonging to that child

Which attribute to choose?

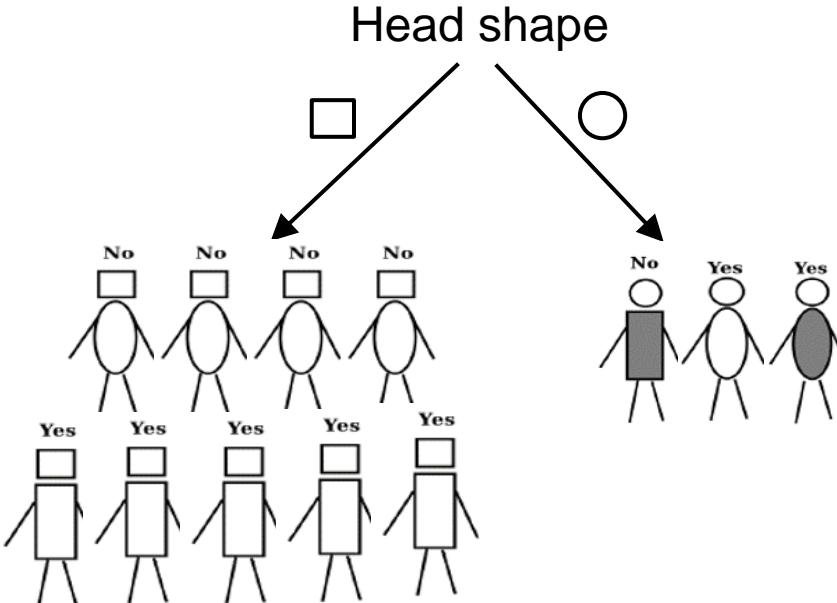
# Exercise – Information gain

Which attribute to choose?

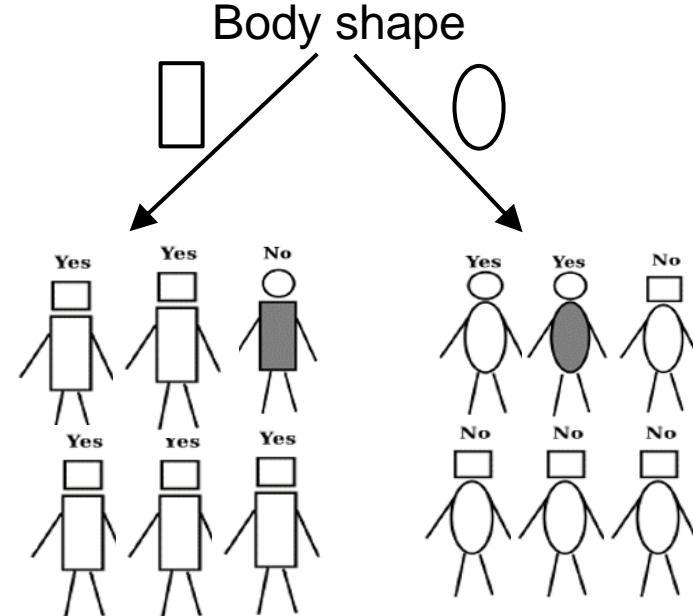


Yes: 7  
No: 5

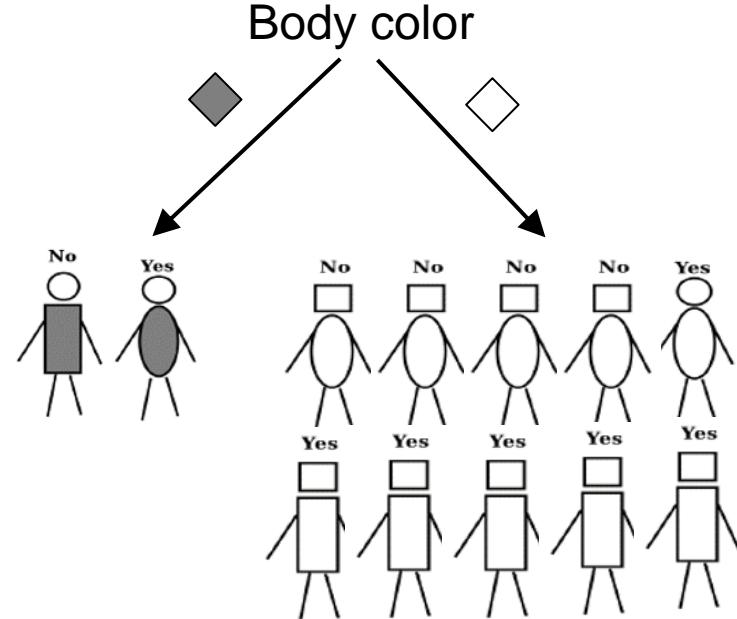
$$\text{entropy}(\text{parent}) \approx 0,98$$



$$\text{entropy}(\square) \approx 0,99 \quad \text{entropy}(\circ) \approx 0,92$$
$$p(\square) \approx 0,75 \quad p(\circ) \approx 0,25$$



$$\text{entropy}(\square) \approx$$
$$p(\square) \approx$$
$$\text{entropy}(\circ) \approx$$
$$p(\circ) \approx$$



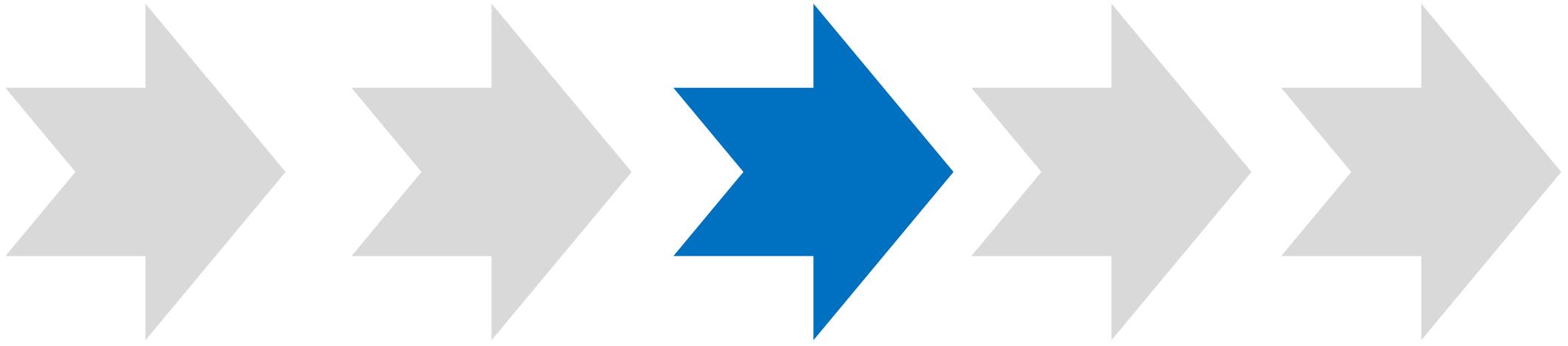
$$\text{entropy}(\diamond) \approx$$
$$p(\diamond) \approx$$
$$\text{entropy}(\square) \approx$$
$$p(\square) \approx$$

IG ≈

IG ≈

IG ≈

# Agenda



(1) Models and induction

(2) Attribute Selection

(3a) Decision Trees  
Algorithmic View

(3b) Probability Estimation

(3c) Decision Tree Examples

# Decision Trees

If we select multiple attributes each giving some information gain, it's not clear how to put them together → **decision trees**

The tree creates a segmentation of the data

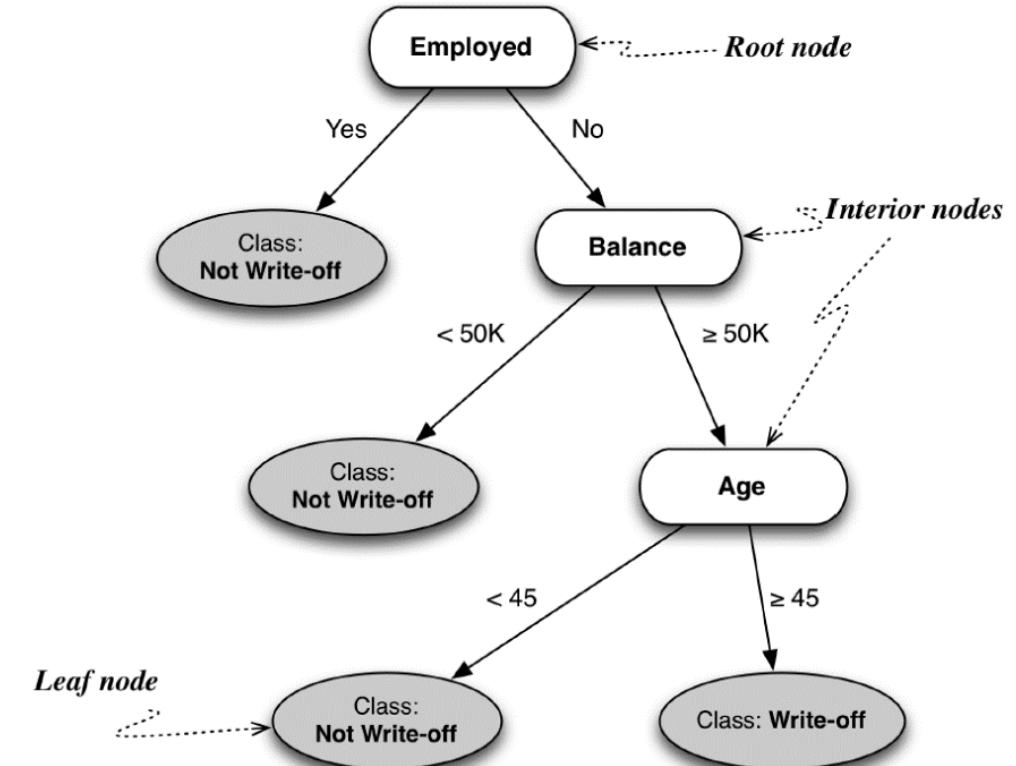
Each *node* in the tree contains a test of an attribute

Each *path* eventually terminates at a *leaf*

Each leaf corresponds to a *segment*,  
and the attributes and values along  
the path give the characteristics

Each leaf contains a value for the  
target variable

Decision trees are often used as **predictive models**



→ Prediction of target attribute, e.g.  
Michi, 40,000, 24, yes, Write-off?  
Micki, 115,000, 40, no, Write-off?

# How to build a decision tree

Manually build the tree deductively, based on expert knowledge

- very time-consuming
- trees are sometimes corrupt (redundancy, contradictions, non-completeness, inefficient)

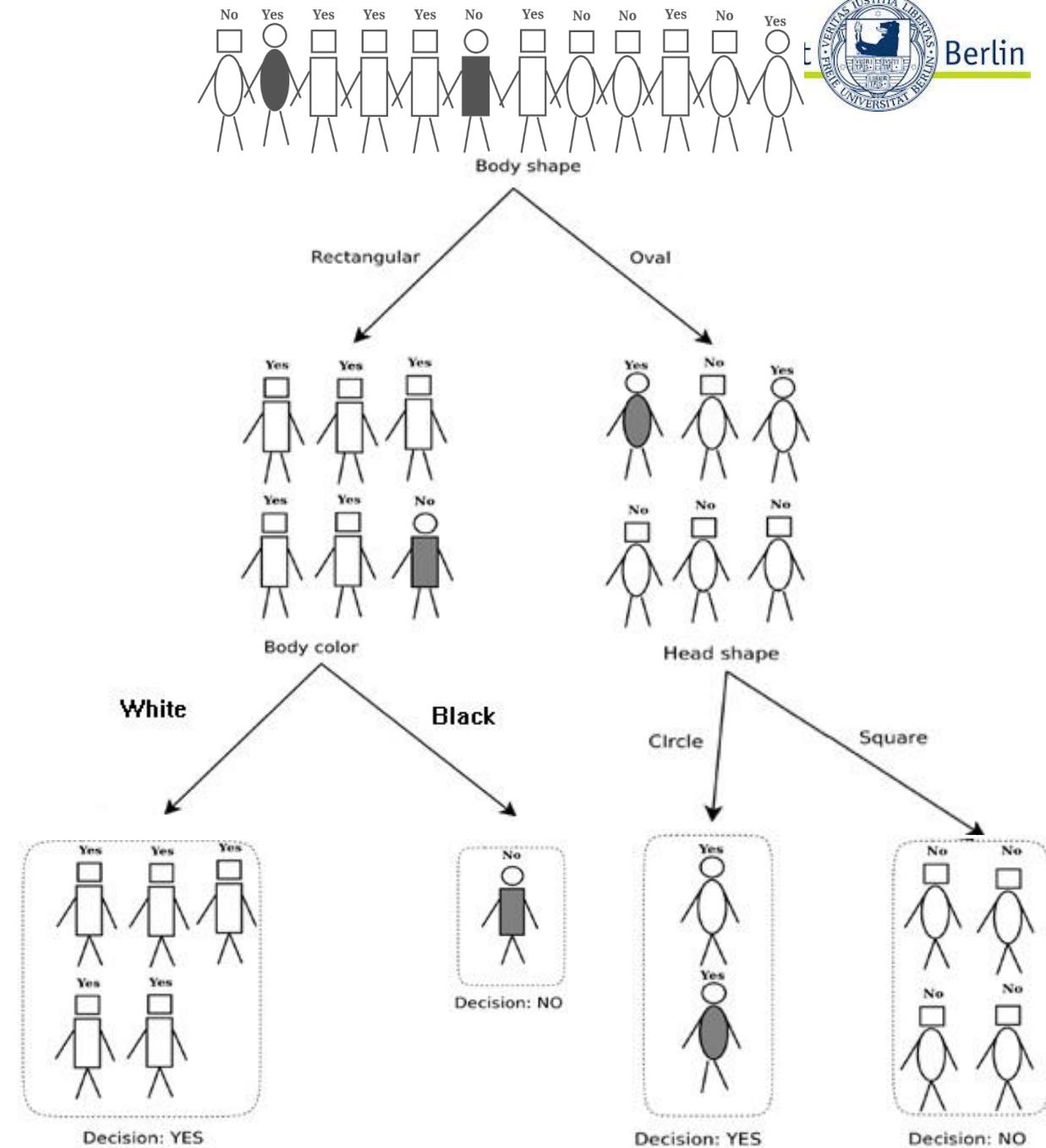
Build the tree **automatically** by induction

- recursively partition the instances based on their attributes (divide-and-conquer)
- easy to understand
- relatively efficient

Recursively apply **attribute selection** to find the best attribute to partition the data set

The goal at each step is to select an attribute to partition the current group into subgroups that are as pure as possible w.r.t. the target variable

Ref.



# ID3 – an algorithm for tree induction

**Iterative Dichotomiser (ID3)** is the most used machine learning algorithm in scientific literature and in commercial systems

- Invented by Ross Quinlan in the early 1980s
- Uses information gain as a measure for the quality of partitioning
- *Requires categorical variables*
- Has no features of handling noise
- Has no features of avoiding overfitting



Ref.

ID3 encompasses a **top-down** decision tree building process

Basic step: split sets into disjoint subsets, where all the *individuals within a subset show the same value for a selected attribute*

- One attribute for testing is selected at the current node
- Testing separates the set into partitions
- For each partition, a subtree is built
- Subtree-building is terminated if all the samples in one partition belong to the same class
- Labels the tree leaves with the corresponding class

Other algorithms, e.g., CART, CHAID, C4.5  
We will use CART in the online-exercise.  
(see <https://scikit-learn.org/stable/modules/tree.html#tree-algorithms>)

# ID3: algorithmic structure

An **intelligent order of the tests** is the key feature of the ID3 algorithm (→ information gain)

*Recursive algorithmic structure of ID3:*

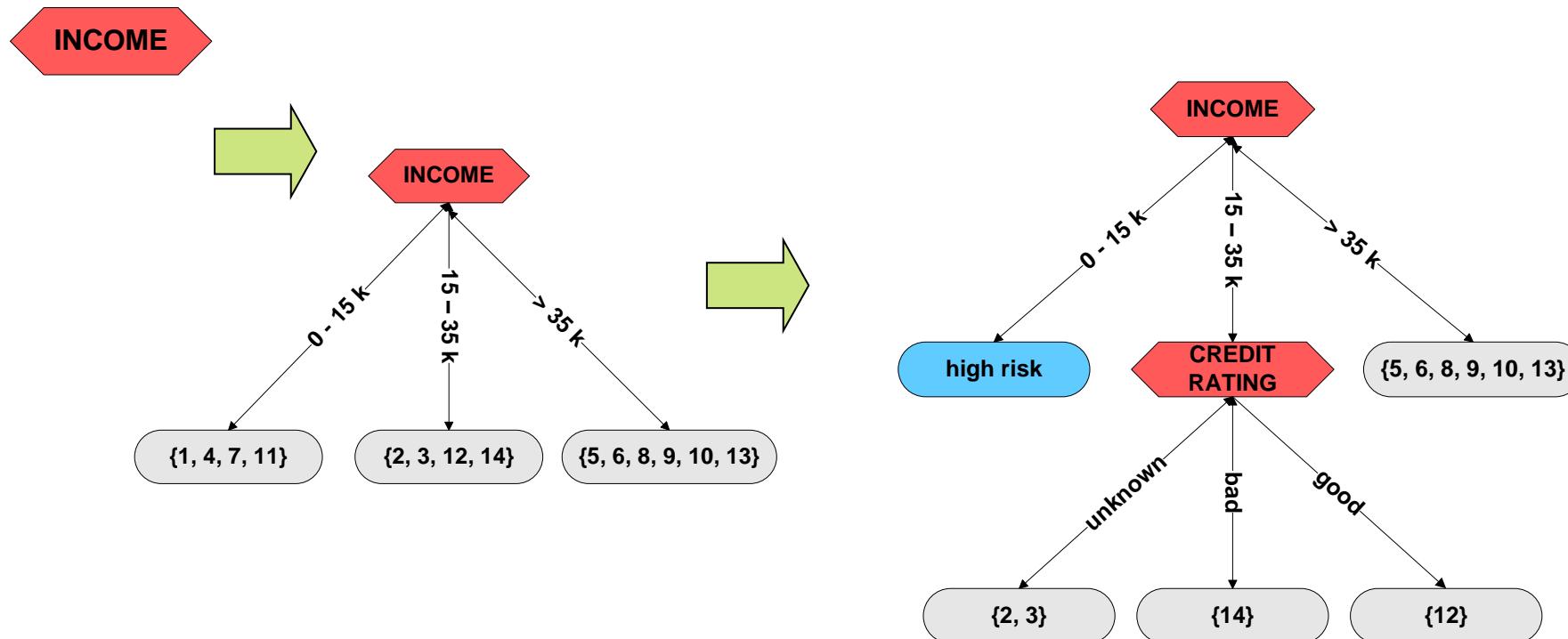
```
FUNCTION induce_tree(sample_set, attributes)
BEGIN
    IF (all the individuals from sample_set belong to the same class)
        RETURN (leave bearing the corresponding class label)
    ELSEIF (attributes is empty)
        RETURN (leave bearing a label of all the classes present in sample_set)
    ELSE
        select an attribute X_j (SCORE FUNCTION)
        make X_j the root of the current tree
        delete X_j from attributes
        FOR EACH (value V of X_j)
            construct a branch, labeled V
            V_partition = sample individuals for which X_j == V
            induce_tree(V_partition, attributes)
    END
```

# ID3: example steps

$$X^0 = \{\text{INCOME}, \text{CREDIT RATING}, \text{SECURITIES}, \text{LEVEL OF DEBT}\}$$

$$X^1 = \{\text{CREDIT RATING}, \text{SECURITIES}, \text{LEVEL OF DEBT}\}$$

$$X^2 = \{\text{SECURITIES}, \text{LEVEL OF DEBT}\}$$



Ref.

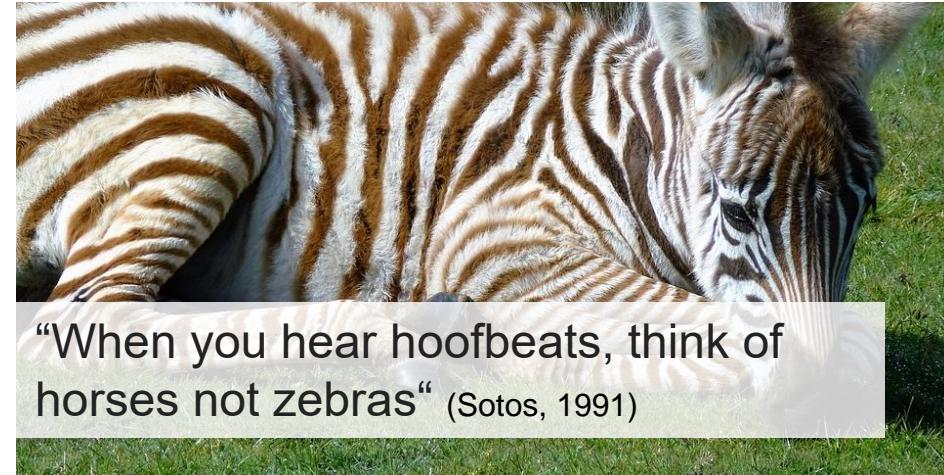
# ID3: simplicity

ID3 tries to build a tree, that

- classifies all the training samples correctly and
- provides a high probability for implying only a small number of tests when classifying an individual

Criterion for attribute selection:

- prefer attributes which contribute a **bigger amount of information** to the classification of an individual
- Information content is measured according to entropy

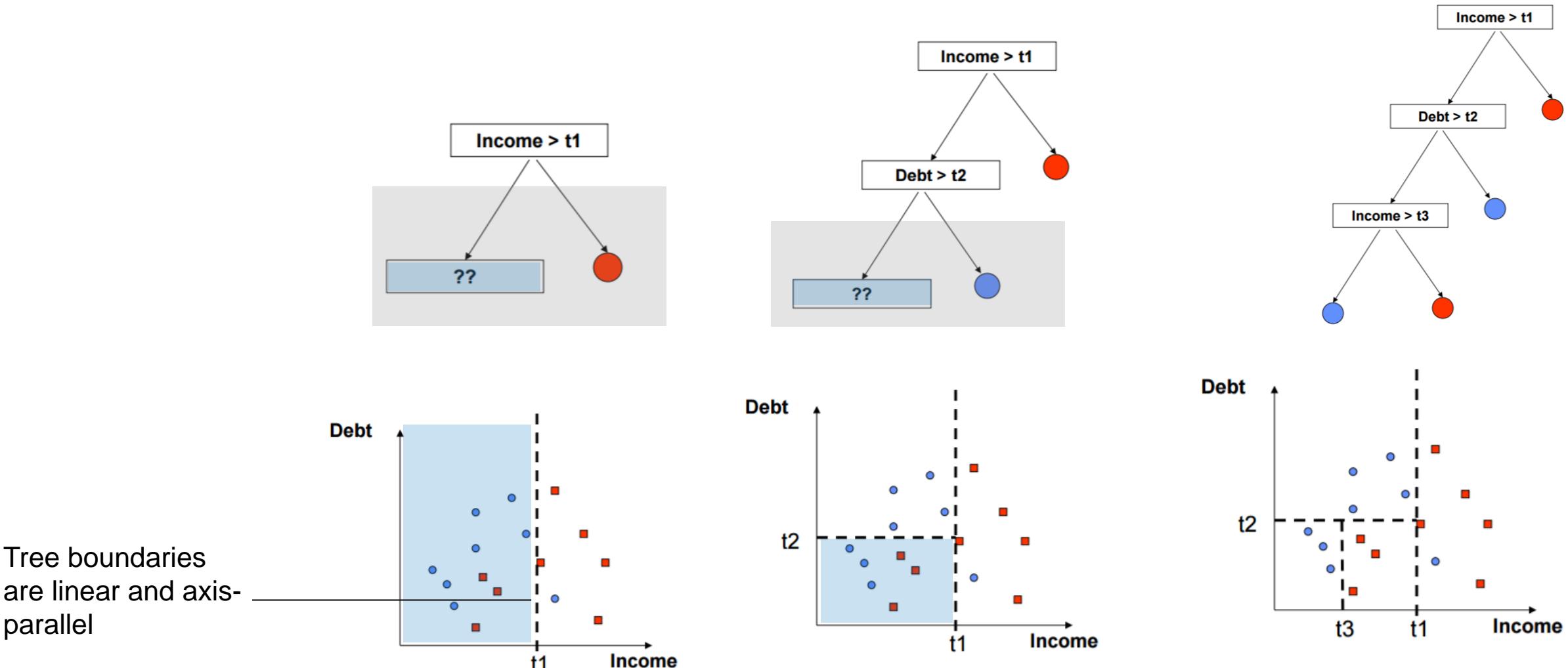


## Occam’s razor:

the simplest tree produces the smallest classification error rates, when applying the tree to new individuals

# Nonlinearity and Decision Surface

Example – Using two attributes to visualize segmentations



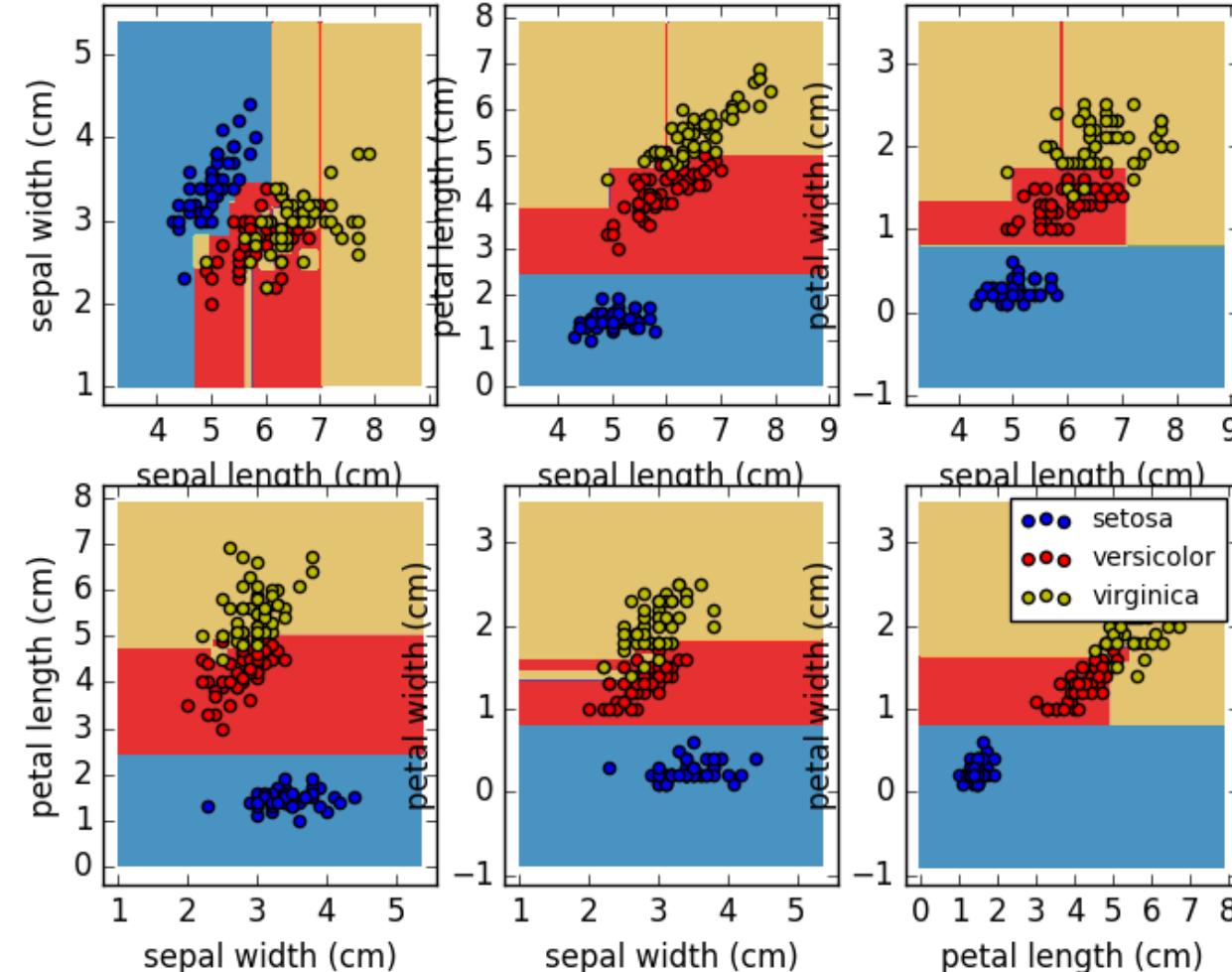
Ref.

# Nonlinearity and Decision Surface

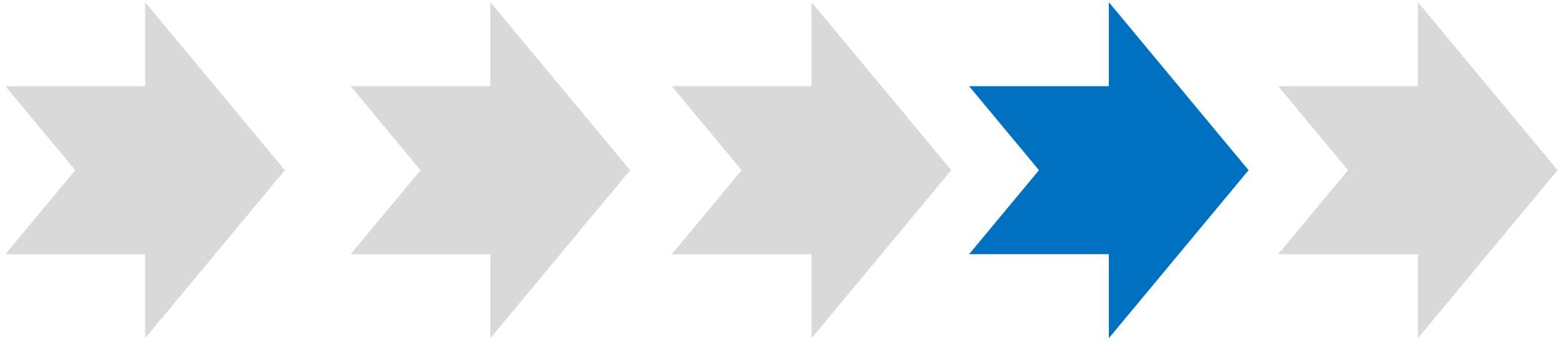
Example - Iris data

Decision Trees model  
non-linear relationships  
between attributes

Decision surface of a decision tree using paired features



# Agenda



(1) Models and induction

(2) Attribute selection

(3a) Decision Trees  
Algorithmic View

(3b) Probability Estimation

(3c) Decision Tree Examples

# Probability estimation (1/3)



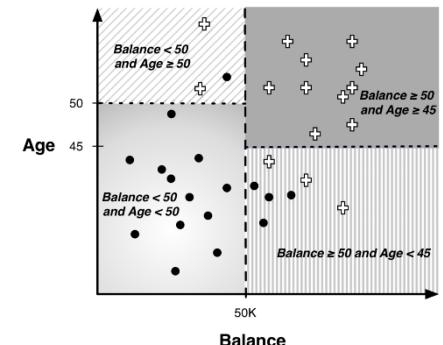
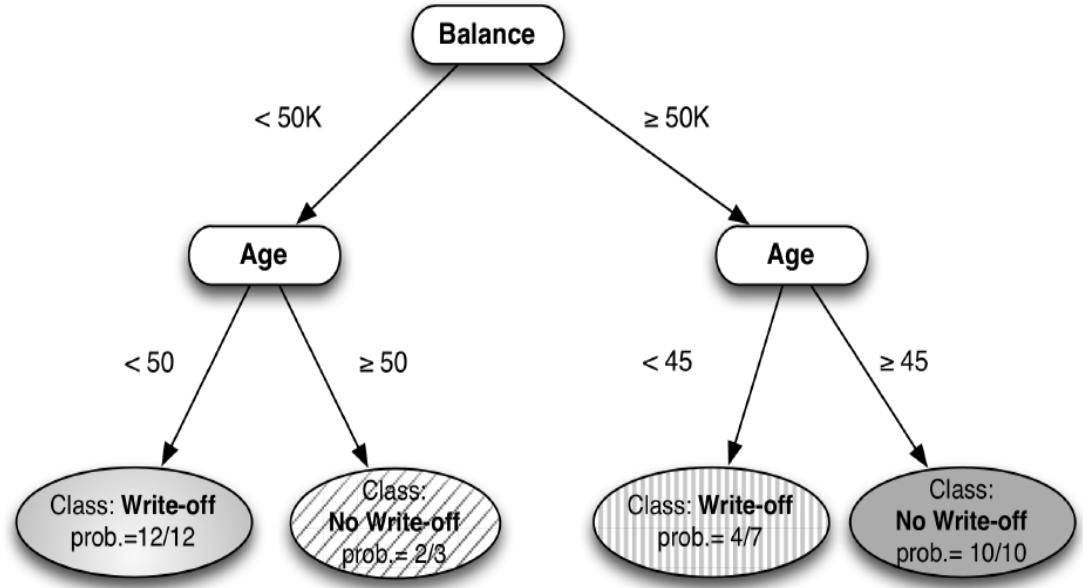
We often need a **more informative prediction** than just a classification

- E.g. allocate your budget to the instances with the highest expected loss
- More sophisticated decision-making process

Classification may oversimplify the problem

- E.g. if all segments have a probability of  $<0.5$  for write-off, every leaf will be labeled „not write-off“
- We would like each segment (leaf) to be assigned an **estimate of the probability of membership** in the different classes

## Probability estimation tree



# Probability estimation (2/3)

Tree induction can easily produce probability estimation trees instead of simple classification trees

- **Instance counts** at each leaf provide class probability estimates
- **Frequency-based estimate** of class membership:  
if a leaf contains  $n$  positive and  $m$  negative instances, the probability of any new instance being positive may be estimated as  $n/(n + m)$ .

Approach may be too optimistic for segments with a very small number of instances ( $\rightarrow$  overfitting)

- Smoothed version of frequency-based estimate by **Laplace correction**, which moderates the influence of leaves with only a few instances:  $p(c) = \frac{n+1}{n+m+2}$  with  $n$  as number of instances that belong to class  $c$  and  $m$  instances not belonging to class  $c$

## Example 1

$$n = 2$$

$$m = 0$$

$$p(c) = \frac{2}{2} = 1$$

$$p_{Laplace}(c) = \frac{2+1}{2+0+2} = 0,75$$

## Example 2

$$n = 20$$

$$m = 0$$

$$p(c) = \frac{20}{20+0} = 1$$

$$p_{Laplace}(c) = \frac{20+1}{20+0+2} = 0,95$$

# Probability estimation (3/3)

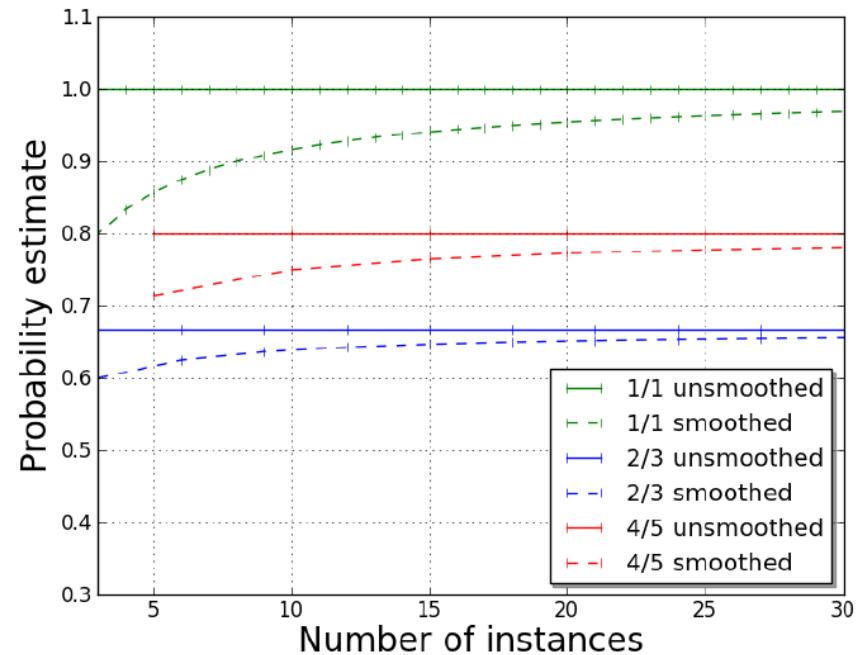


Effect of Laplace correction on several class ratios as the number of instances increases (2/3, 4/5, 1/1)

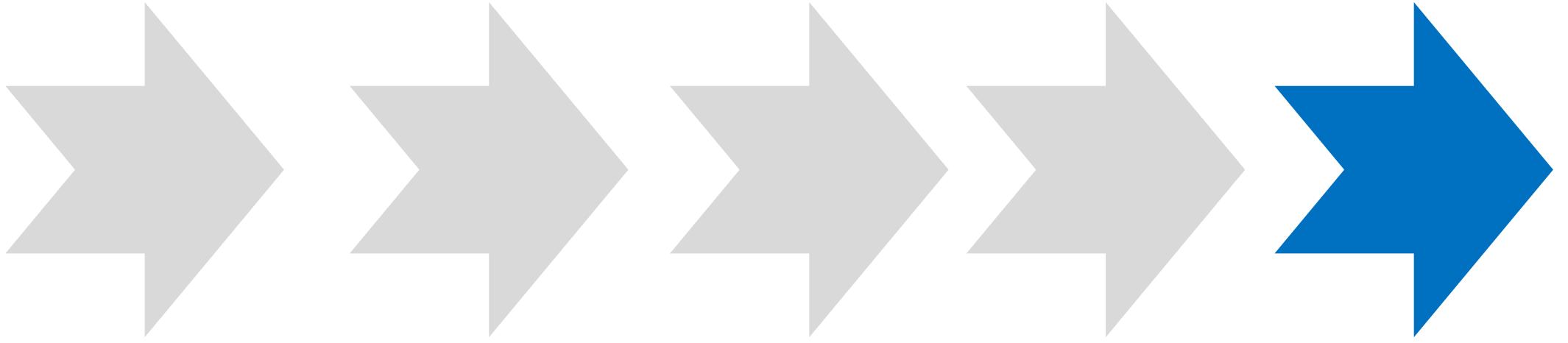
Example:

A leaf of the classification tree that has 2 pos. instances and no negative instances would produce the same f-b estimate ( $p = 1$ ) as a leaf node with 20 pos. and no negatives.

The Laplace correction smooths the estimate of the first leaf down to  $p = 0.75$  to reflect this uncertainty, but it has much less effect on the leaf with 20 instances ( $p \approx 0.95$ )



# Agenda



(1) Models and induction

(2) Attribute selection

(3a) Decision Trees  
Algorithmic View

(3b) Probability Estimation

(3c) Decision Tree Examples

# Example - The Churn Problem

Solve the churn problem by tree induction

Historical data set of 20,000 customers

Each customer either had stayed with the company or left

Customers are described by the following variables:



| Variable                         | Explanation  |
|----------------------------------|--|
| COLLEGE                          | Is the customer college educated?                        |
| INCOME                           | Annual income  |
| OVERAGE                          | Average overcharges per month                            |
| LEFTOVER                         | Average number of leftover minutes per month             |
| HOUSE                            | Estimated value of dwelling (from census tract)          |
| HANDSET_PRICE                    | Cost of phone  |
| LONG_CALLS_PER_MONTH             | Average number of long calls (15 mins or over) per month |
| AVERAGE_CALL_DURATION            | Average duration of a call                               |
| REPORTED_SATISFACTION            | Reported level of satisfaction                           |
| REPORTED_USAGE_LEVEL             | Self-reported usage level                                |
| LEAVE ( <i>Target variable</i> ) | <i>Did the customer stay or leave (churn)?</i>           |

We want to use this data to **predict which new customers are going to churn.**

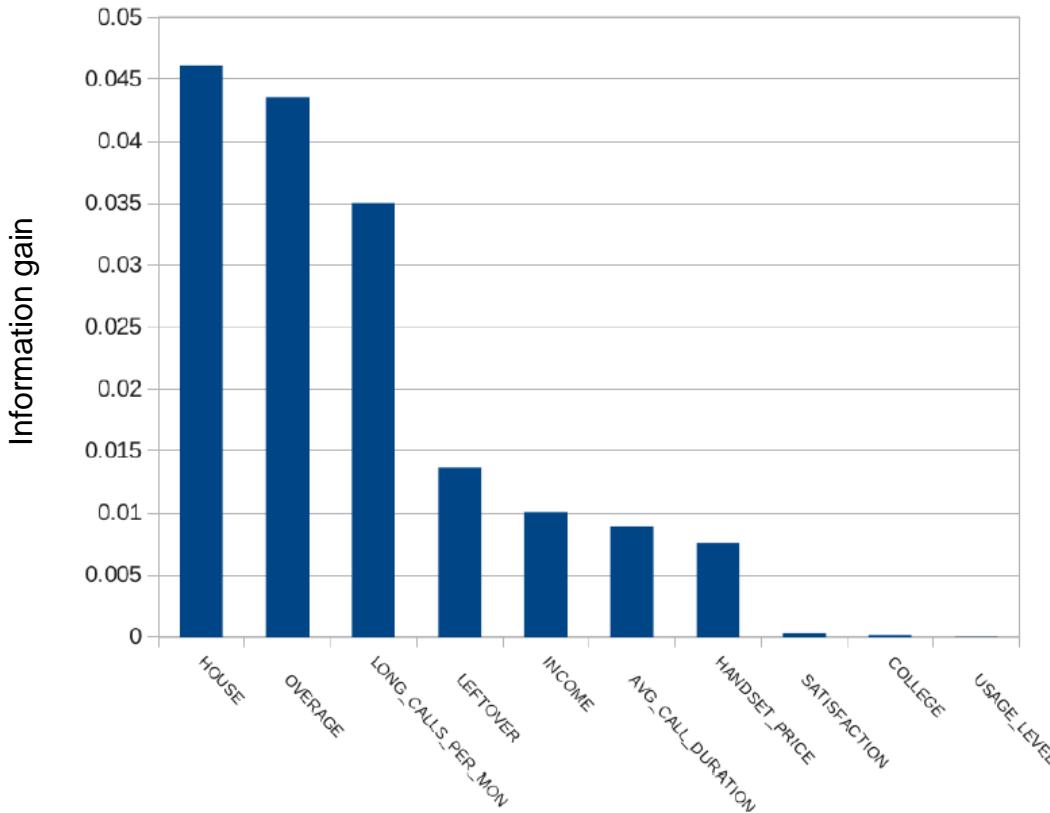
# Example - The Churn Problem



How good are each of these variables individually?

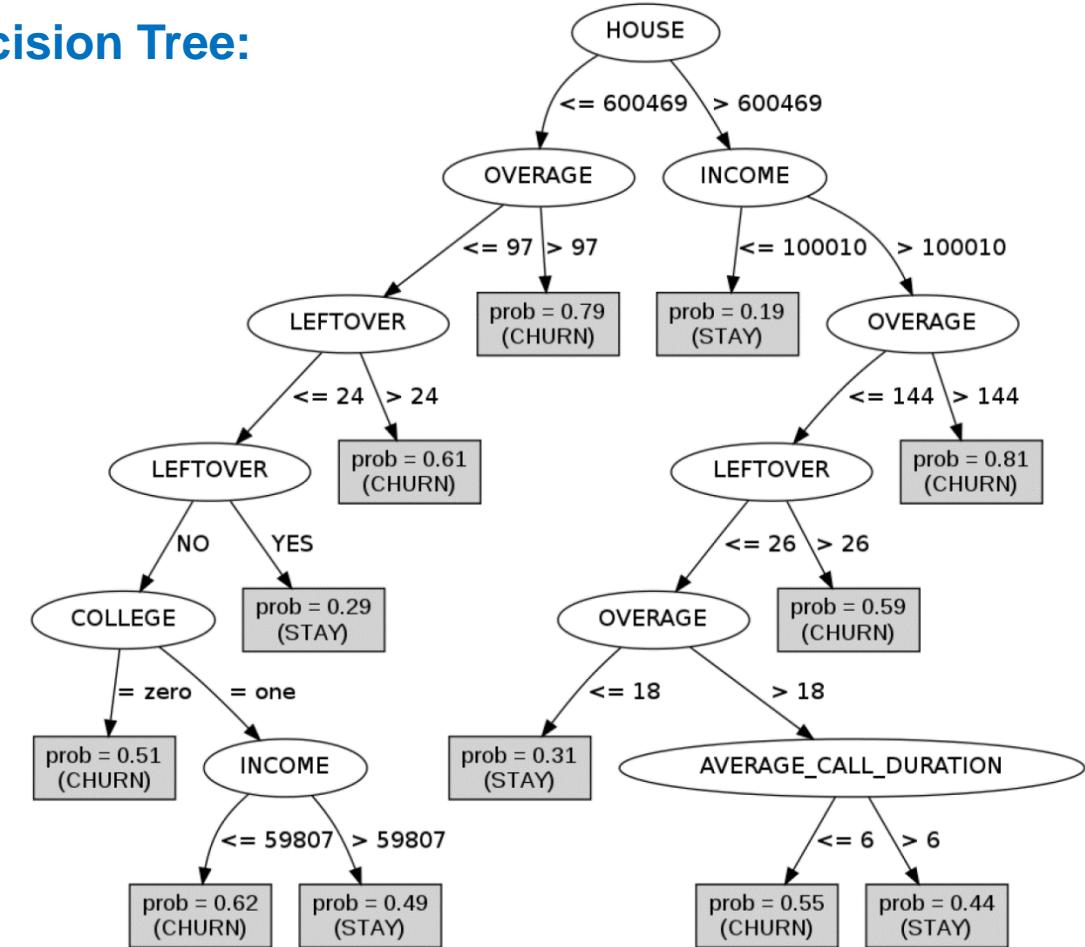
Measure the information gain of each variable

Compute information gain for each variable independently



The highest information gain feature (HOUSE) is at the root of the tree.

Decision Tree:



1. Why is the order of features chosen for the tree different from the ranking?
2. When to stop building the tree?
3. How do we know that this is a good model?

# Example - Decision Trees with Python

See [Python-Analytics online exercise](#) on Friday, 14.6.

If possible, prepare your system for the exercise:

**Achtung:** Die Pakete pandas und sklearn sind in der Regel schon vorinstalliert. Die Pakete graphviz und pydotplus müssen hingegen zunächst außerhalb von Jupyter Notebooks installiert werden, bevor man sie importieren kann. Öffne hierzu die Kommandozeile (Windows) oder das Terminal (Mac) und führe folgendes nacheinander aus:

1. conda install python-graphviz
2. conda install -c conda-forge pydotplus

Installationen direkt aus Jupyter Notebooks heraus oder mit pip führen in der Regel zu Fehlern!

These components only need to be used for a small part of the exercise.

# Outlook: Tree Induction vs. Fitting a Model

Next Lesson

Freie Universität Berlin

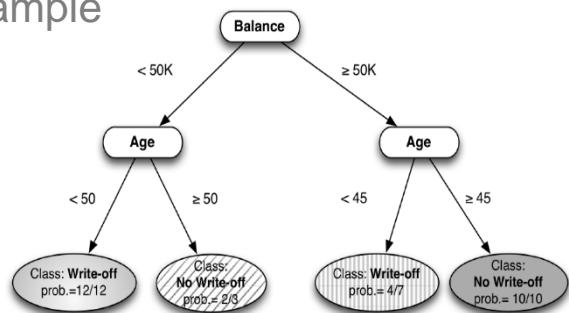


## Classification via Tree Induction

So far:

- ✓ we produced both the **structure of the model** (the particular tree model) and the **numeric parameters** of the model from the data

For example



Questions answered:

- ✓ How do we decide to classify data?
- ✓ Why do we not build „complete“ trees?
- ✓ If we have incomplete trees, we want to assess probabilities. What do we take into consideration?

Ref.

## Classification via Mathematical Functions

Now:

- We **specify the structure of the model**, but leave certain numeric parameters unspecified
- Data Mining calculates the best parameter values given a particular set of training data
- The form of the model and the attributes is specified
- The goal of DM is to tune the parameters so that the model fits the data as good as possible (**parameter learning**)

Simplifying assumptions:

- For classification and class probability estimation, we will consider **only binary classes**.
- We assume that **all attributes are numeric**.  
    (→ see data preparation)
- We **ignore the need to normalize numeric** measurements to a common scale (→ see data preparation)

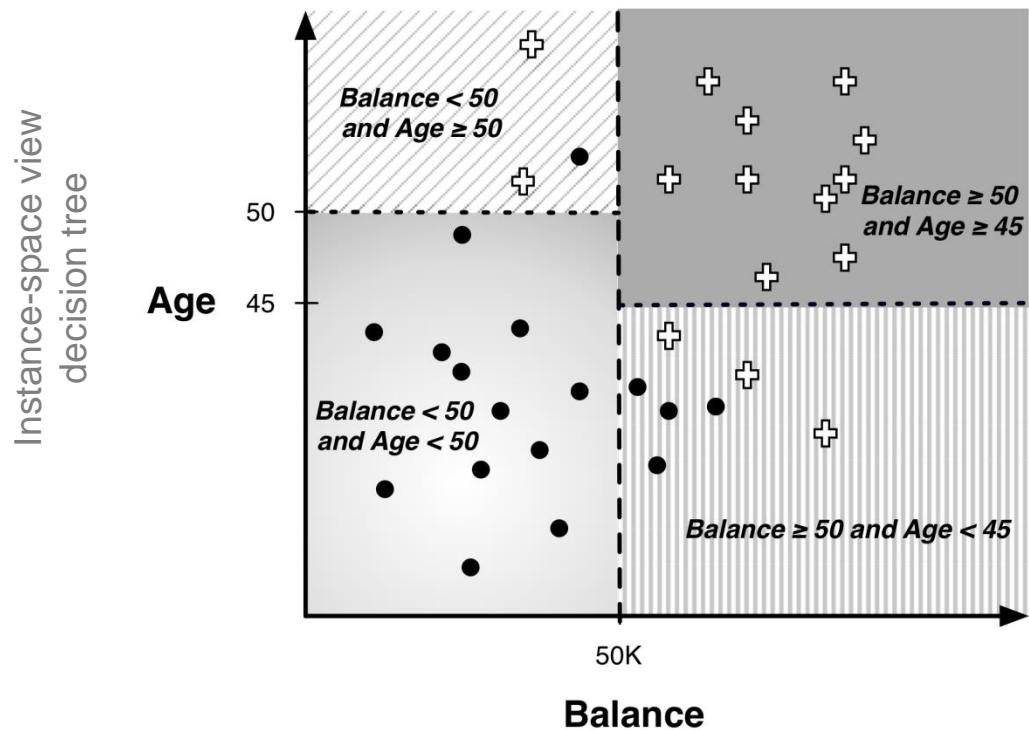
# Linear classifiers

[Next Lesson](#)

## Instance-space view:

shows the space broken up into regions by decision boundaries

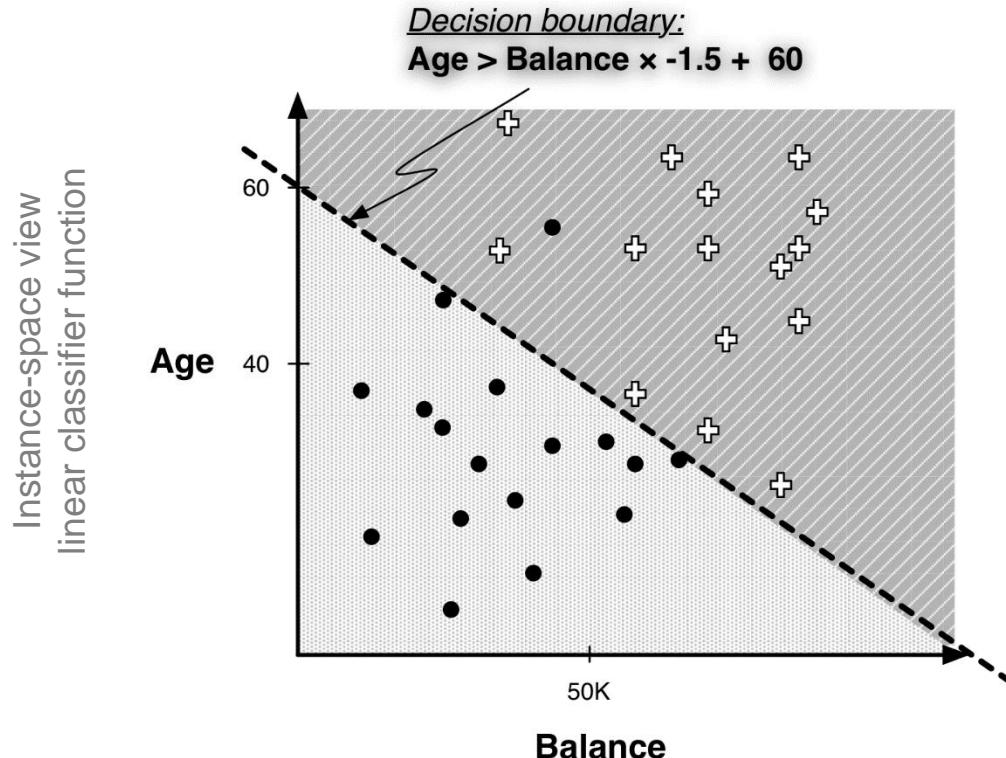
- Examples in each space should have similar values for the target variable
- Homogeneous regions help predicting the target variable of a new, unseen instance



Ref.

We can separate the instance almost perfectly (by class) if we are allowed to introduce a boundary that is still a straight line, but is not perpendicular to the axes

- Linear classifier



## Fragen?

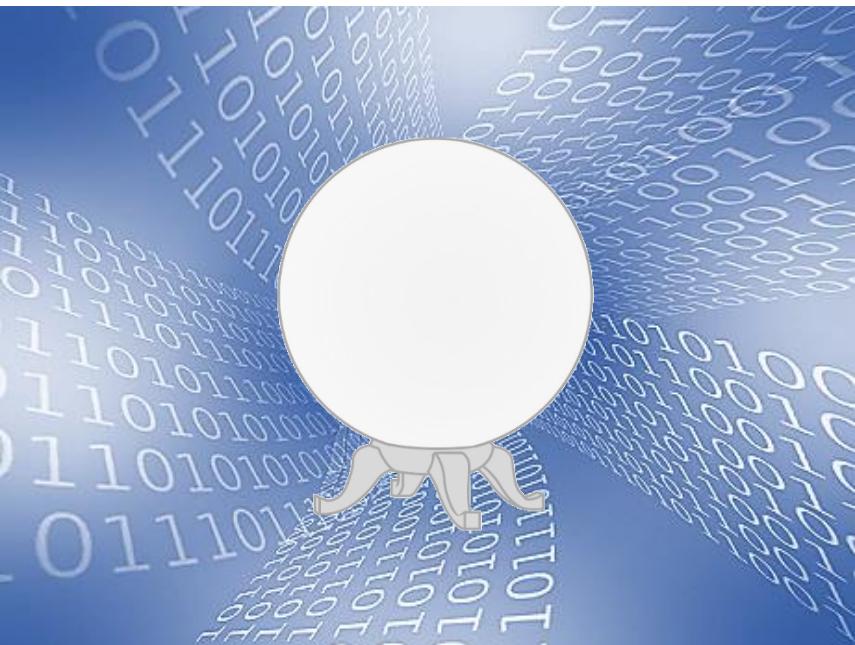
- ✓ Predictive modeling
  - ✓ Decision trees – Introduction
  - ✓ Decision trees – algorithmic view
  - ✓ Probability estimation tree
  - ✓ Decision trees – examples

# Recommended reading

- Provost, F., Data Science for Business  
Fawcett, T. Chapter 3
- Berthold et al. Guide to Intelligent Data Analysis  
Chapter 8.1
- Hand, D. Principles of Data Mining  
Chapter 10
- Quinlan, J.R. Induction of Decision Trees (in: Machine Learning, 1(1), p. 81-106, 1986)

# Bibliography

- J. Bertin (1983) *Semiology of graphics: diagrams, networks, maps*. University of Wisconsin Press. Originally in French: *Semiologie Graphique*, 1967
- Cairo, A. (2012). *The Functional Art: An introduction to information graphics and visualization*. New Riders.
- Mertens, P., & Meier, M. (2009). *Integrierte Informationsverarbeitung*. Wiesbaden: Gabler.
- Woolman, M. (2002). *Digital information graphics*. Watson-Guptill Publications, Inc..



# Business Intelligence

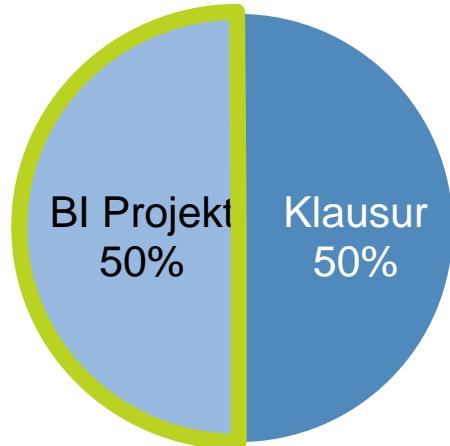
## 11 Fitting a Model

Prof. Dr. Bastian Amberg  
(summer term 2024)

21.6.2024

# Organisatorisches - Prüfungstermine und Prüfungsleistung

Seit Ende Mai sind die Prüfungspläne vom Prüfungsbüro veröffentlicht und seit dem 7.6. können Sie die BI-Projekte bearbeiten.



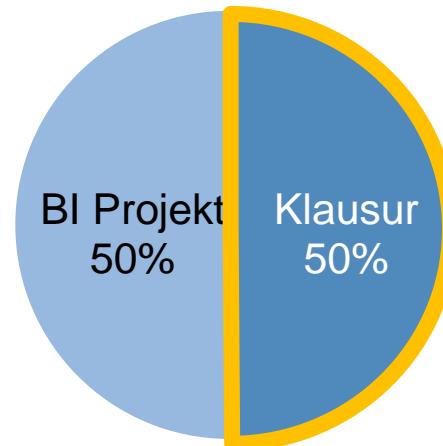
## 1/3: Präsentation

vorherige Abgabe via Blackboard (**16.7.'24 bis 23.59 Uhr**)  
am **17.7.'24 und 19.7.'24**, 20 Minuten Präsentation je Gruppe

## 2/3: Kurzdokumentation

Gliederung gemäß CRISP-DM,  
max. 12 Seiten (3 pro Person)  
Abgabe via Blackboard (spätestens am **14.8.'24 (bis 23.59 Uhr)**).

*Am 3.7. Hinweise zu den Abschlusspräsentationen inkl. Reihenfolge der Präsentationen (gemeinsame Auslosung via Python)*



## 1x Klausur

**1. Termin 31.7.'24, 10.15 Uhr, HS 108, 60 Minuten**  
oder **2. Termin 2.10.'24, 10.15 Uhr, HS 108, 60 Minuten**

*Ab Mittwoch, 26.6., Wie könnten mögliche Klausuraufgaben aussehen? (Beispielaufgaben in Blackboard)*

Prüfungszeitraum Termin 1: 22.07. - 03.08.2024

Prüfungszeitraum Termin 2: 23.09. - 05.10.2024

(<https://www.wiwiiss.fu-berlin.de/studium-lehre/pruefungsbuero/news/Pruefungsplaene-SoSe-2024.html>)

# Hinweise

Wie könnten mögliche Klausuraufgaben aussehen?

40 days left  
until the exam  
(Termin 1)

103 days left  
until the exam  
(Termin 2)



## Zu den Inhalten:

- Die Inhalte aus den Python-Selbstlerneinheiten sowie aus den beiden Python-Übungen fließen nicht in die Klausur ein.  
Kein Python.
- Inhaltliche Schwerpunktsetzung beachten. **Aufteilung: DW/DE** ca. 10-15 Punkte, **DM/DS** ca. 45-50 Punkte.
- Für den Teil **DW/DE** haben Sie im Rahmen der Bonusaufgaben mit *Co-Create your exam* einen Fragenpool formuliert.  
(*BI\_2024\_DW-DE\_Aufgabenpool.pdf* ab dem 26.6. unter Kursmaterial/Übungen -- deckt vollständig diesen Teilbereich der Klausur ab)
- Beispiele für Fragestellungen zum Teil **DM/DS** ergeben sich aus den Übungsaufgaben während der Vorlesung („Exercises“ beachten), bzw. aus dem in Blackboard bereit gestellten Zusatzmaterial ausgewählter Aufgaben vergangener Jahre  
(*BI\_2024\_DM-DS\_Probeaufgaben.pdf* ab dem 26.6. unter Kursmaterial/Übungen)
- Insbesondere sollen die Probeaufgaben ein Gefühl dafür vermitteln, wie die Klausurfragen vermutlich formuliert sind.  
(Die Vorlesungsinhalte zum Teil **DM/DS** bereiten Sie bereits aktuell durch die Bearbeitung der Projektaufgabe nach)

## Erwartungen:

- Die gemeinsam bzw. im Selbststudium „geübten“ Methoden und Berechnungen (z.B. zu Decision Trees, in Kürze auch zu Expected Value Framework, Naive Bayes, k-Nearest Neighbors, ...) können selbstständig durchgeführt werden.
- Sollten darüber hinaus bei der kritischen Beurteilung bzw. Entscheidungsfindung Formeln zwingend notwendig sein, die wir nur überblicksartig behandelt haben (z.B. Korrelationsmaße), werden diese in der Aufgabenstellung mit angegeben.
- In jedem Fall sollten Sie in der Lage sein, Maße, Modelle und Methoden gegenüberzustellen, um sie in konkreten Fällen begründbar auszuwählen.

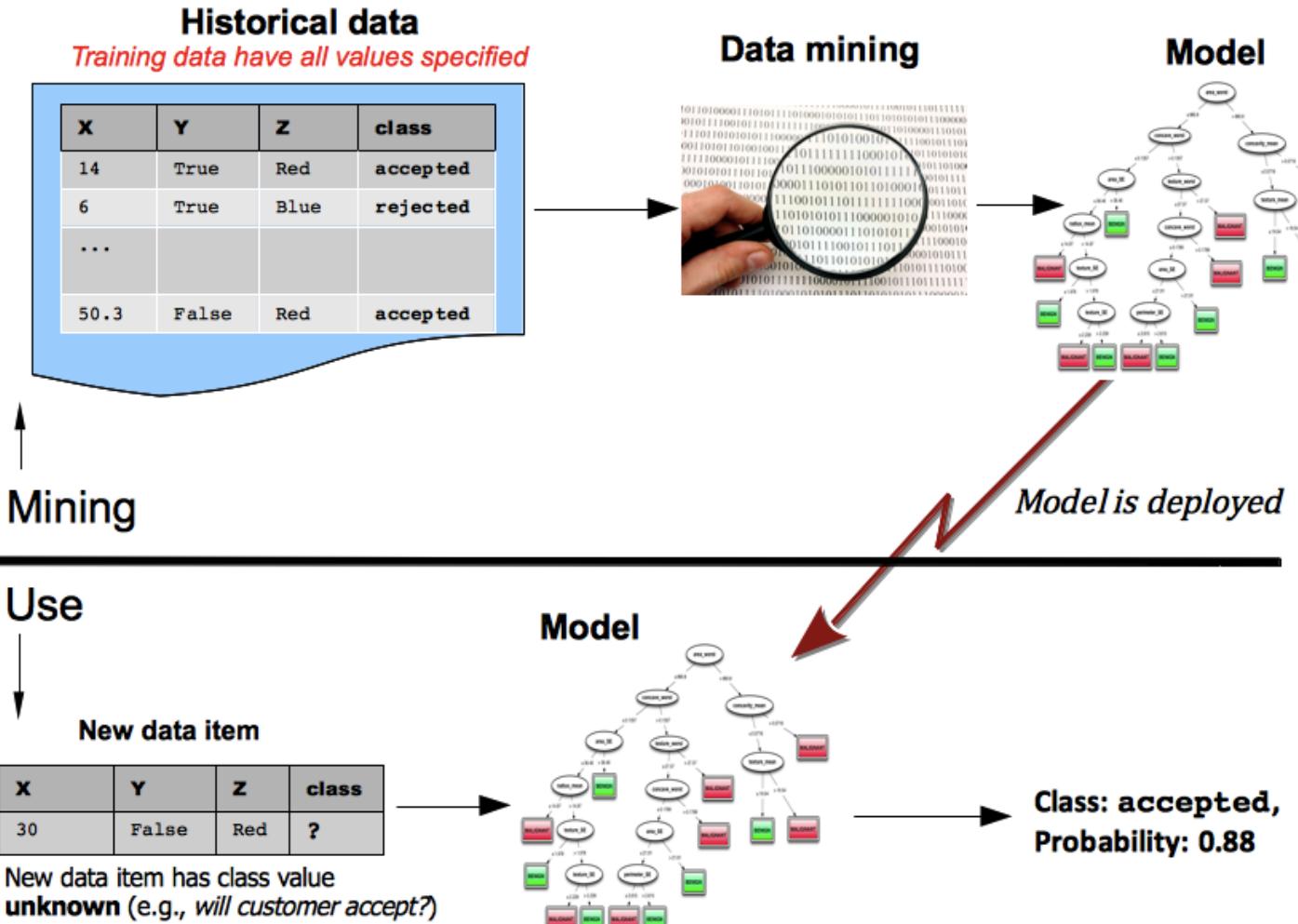
*(In der letzten Veranstaltung am 17.7. folgen noch ergänzende Informationen)*

# Schedule

|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study |                  |               |         |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|------------|------------------|---------------|---------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |            |                  | Python-Basics | Chap. 1 |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |            |                  |               | Chap. 2 |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  |            |            |                  |               | Chap. 3 |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining Introduction  |            |                  |               |         |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |            | Python-Analytics | Chap. 1       |         |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization I            |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam |            |                  | Chap. 2       |         |  |  |
|           | W7                | 29.5. | Data Visualization II                               |                                   | 31.5. |   |            |                  |               |         |  |  |
|           | W8                | 5.6.  | Data Preparation                                    |                                   | 7.6.  | Predictive Modeling I (10:00 -12:00)  |            | BI-Project       | Start         |         |  |  |
|           | W9                | 12.6. | Predictive Modeling II                              |                                   | 14.6. | Python-Analytics-Online Exercise  |            |                  |               |         |  |  |
|           | W10               | 19.6. | Guest Lecture Dr. Ionescu                           |                                   | 21.6. | Fitting a Model   |            |                  |               |         |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |            |                  |               |         |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I                                  |            |                  |               |         |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |            |                  |               |         |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |            |                  | End           |         |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin, 31.7.'24<br>Klausur 2.Termin, 2.10.'24                                      |            | Projektbericht   |               |         |  |  |

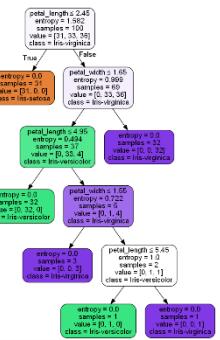
# Recap Python Exercise: Data mining and its use

## Discerning model building and model deployment



See Python-Analytics Exercise  
from Friday 14.6.

Build model using  
100 samples  
(training set)



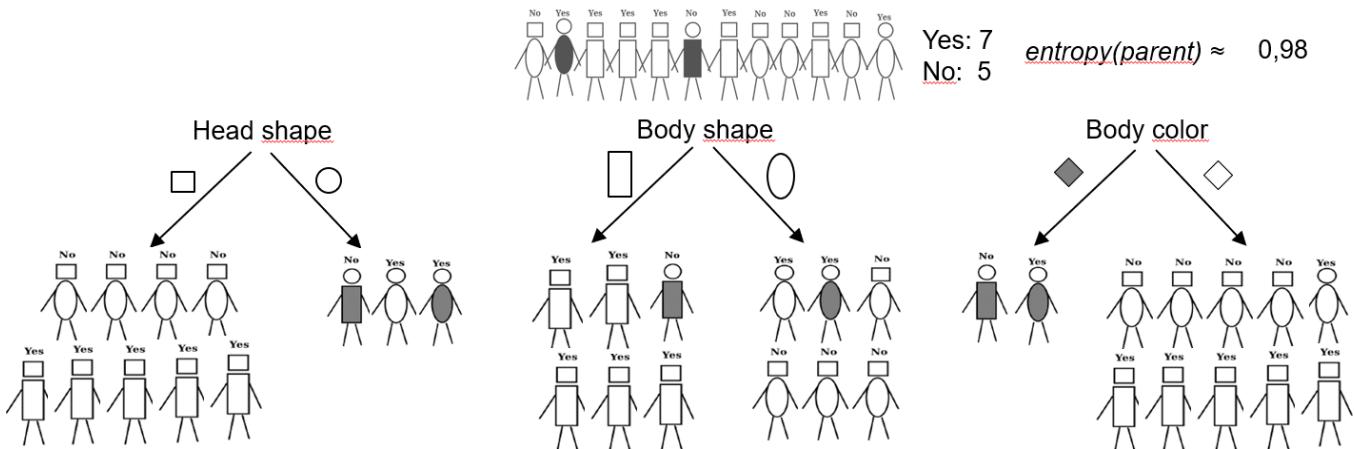
Test model using 50 samples  
(test set)

# Last Lesson

## Predictive Modeling

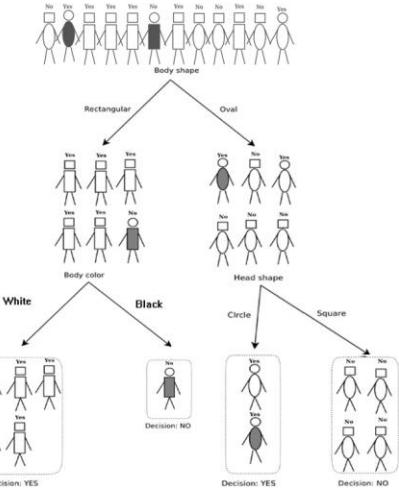
### Decision Trees

Which attribute to choose?



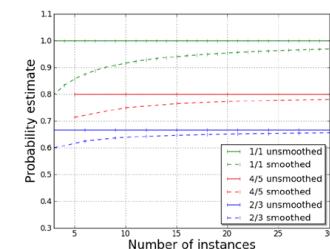
Recursively apply **attribute selection** to find the best attribute to partition the data set

The goal at each step is to select an attribute to partition the current group into subgroups that are as pure as possible w.r.t. the target variable



- Decision Trees model non-linear relationships between attributes
- Different decision tree algorithms generate decision trees with different structure
- Tree induction can easily produce **probability estimation trees** instead of simple classification trees

Smoothed version of frequency-based estimate by **Laplace correction**, which moderates the influence of leaves with only a few instances



Ref.



## Classification via Mathematical Functions

### Fitting a Model to Data



(1) Linear  
Classifiers

(2) Linear  
Regression

(3) Logistic  
Regression

(4) Tree-induction  
vs. logistic  
regression

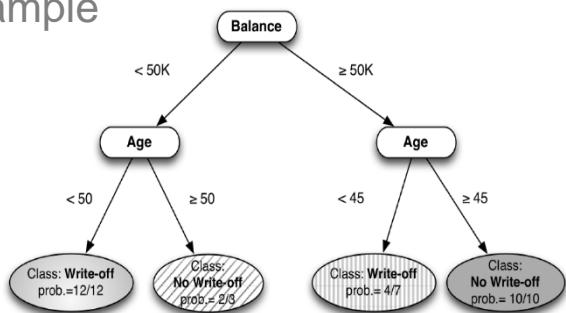
# Introduction

## Classification via Tree Induction

So far:

- ✓ we produced both the **structure of the model** (the particular tree model) and the **numeric parameters** of the model from the data

For example



Questions answered:

- ✓ How do we decide to classify data?
- ✓ Why do we not build „complete“ trees?
- ✓ If we have incomplete trees, we want to assess probabilities. What do we take into consideration?

Ref.

## Classification via Mathematical Functions

Now:

- We **specify the structure of the model**, but leave certain numeric parameters unspecified
- Data Mining calculates the best parameter values given a particular set of training data
- The form of the model and the attributes is specified
- The goal of DM is to tune the parameters so that the model fits the data as good as possible (**parameter learning**)

Simplifying assumptions:

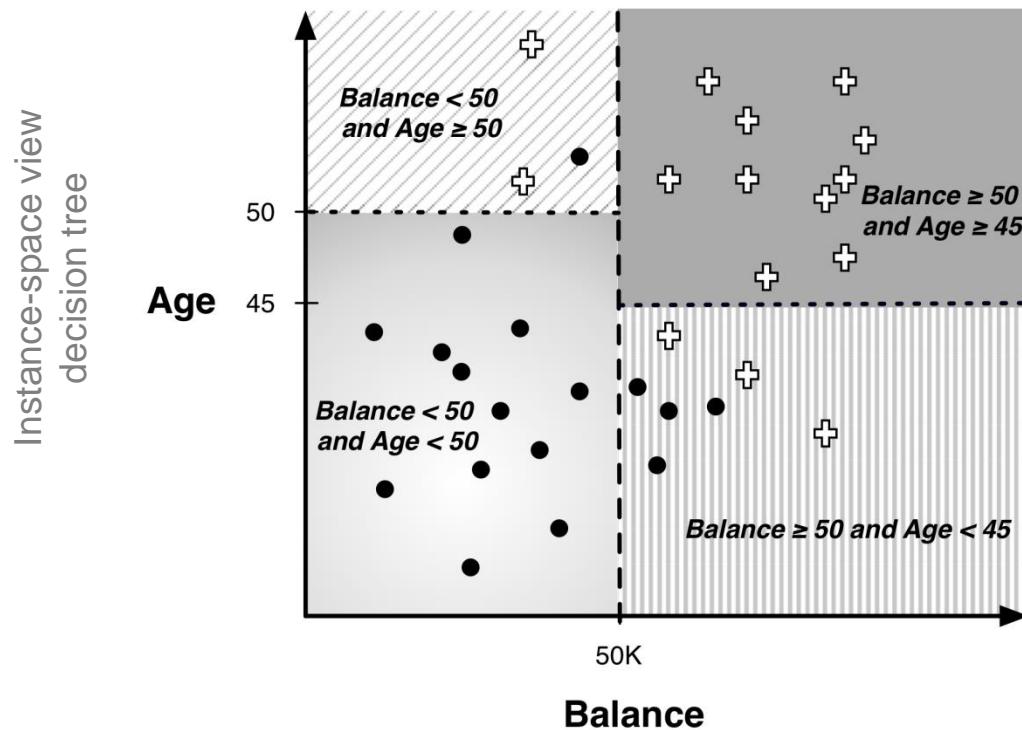
- For classification and class probability estimation, we will consider **only binary classes**.
- We assume that **all attributes are numeric**.  
(→ see data preparation)
- We **ignore the need to normalize numeric** measurements to a common scale (→ see data preparation)

# Linear classifiers

## Instance-space view:

shows the space broken up into regions by decision boundaries

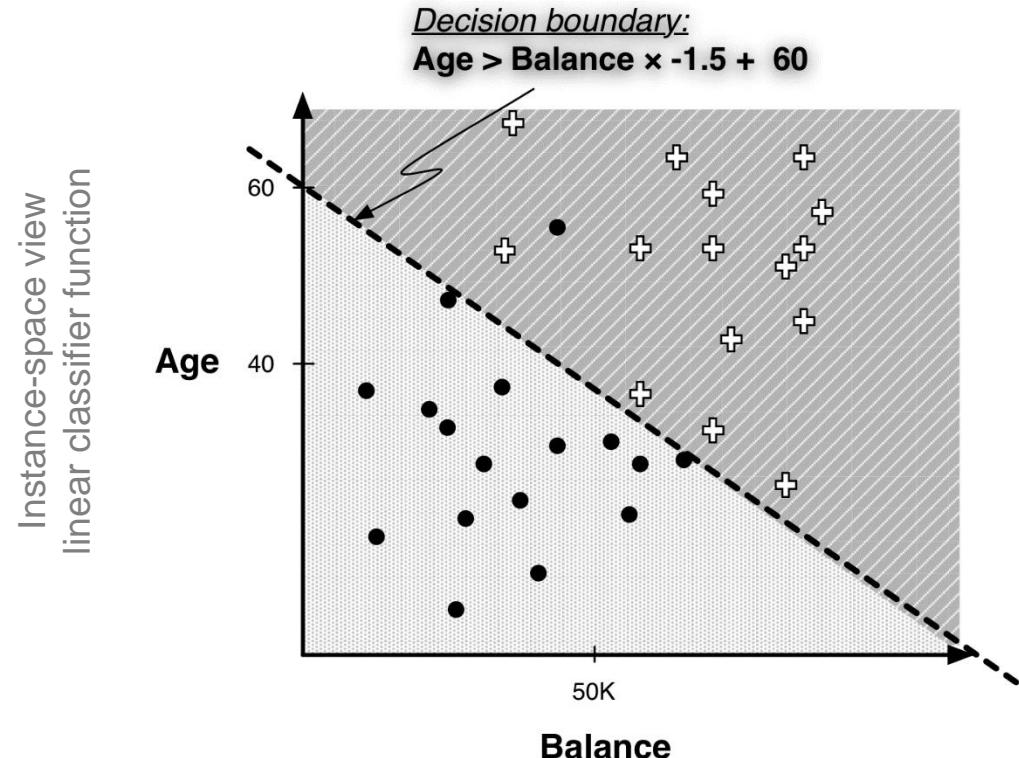
- Examples in each space should have similar values for the target variable
- Homogeneous regions help predicting the target variable of a new, unseen instance



Ref.

We can separate the instance almost perfectly (by class) if we are allowed to introduce a boundary that is still a straight line, but is not perpendicular to the axes

- Linear classifier



# Linear discriminant functions (1/2)

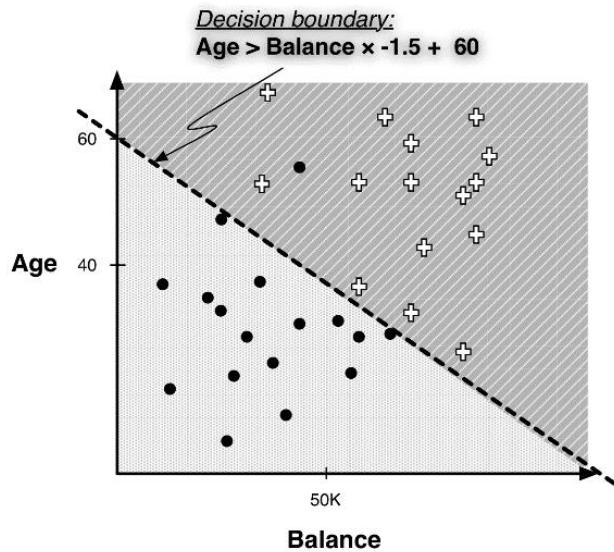
Classifying with a linear function (*parametric learning*)

**Equation of a line:**  $y = mx + b$

with  $m$  being the slope and  $b$  the  $y$  intercept

Line in figure:

$$Age = (-1.5) * Balance + 60$$



We would classify an instance  $x$  as a “+” if it is above the line, and as a “•” if it is below the line.

Ref.

Mathematically:

$$\text{class} = \begin{cases} + & \text{if } -1.0 * \text{Age} - 1.5 * \text{Balance} + 60 > 0 \\ \bullet & \text{if } -1.0 * \text{Age} - 1.5 * \text{Balance} + 60 \leq 0 \end{cases}$$

Linear discriminant discriminates between the Classes

Supervised segmentation by creating a mathematical function of multiple attributes

A **linear discriminant function** is a numeric classification model, which can be written as

$$f(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \dots$$

$w_0, w_1 \dots w_n$  are parameters to be estimated

# Linear discriminant functions (2/2)



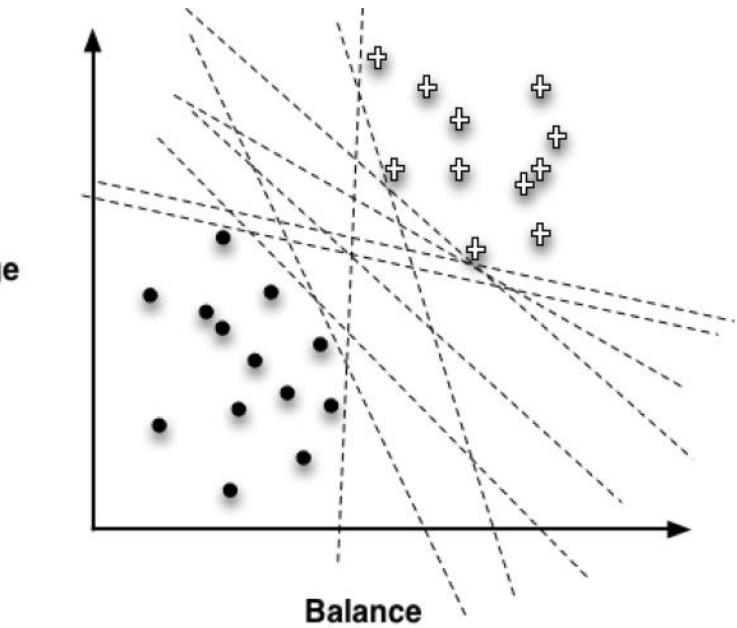
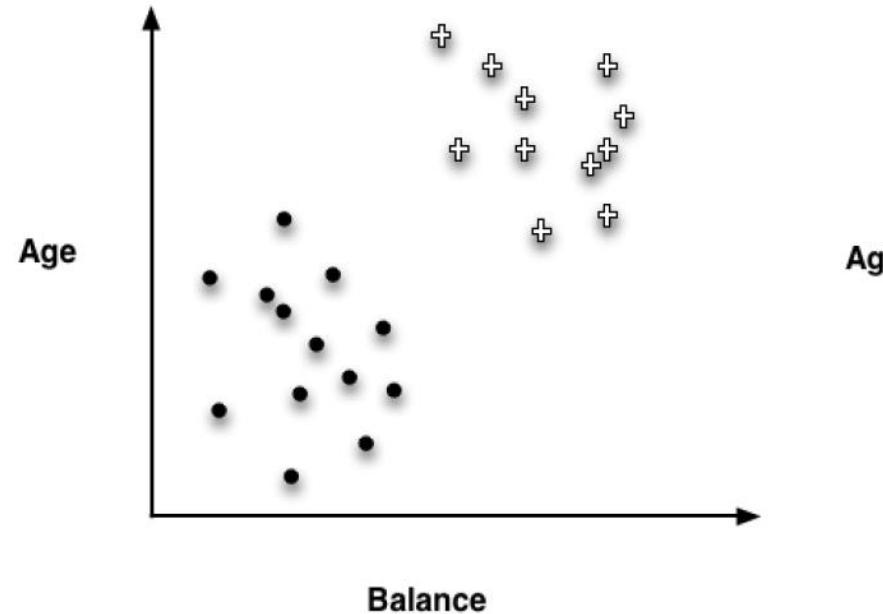
What is the „best“ line to separate the classes?

Fit parameters  $w_i$  to a particular data set

Find a good set of weights (using a given set of features)

Weights may be interpreted as importance indicators

The larger the magnitude of a weight, the more important



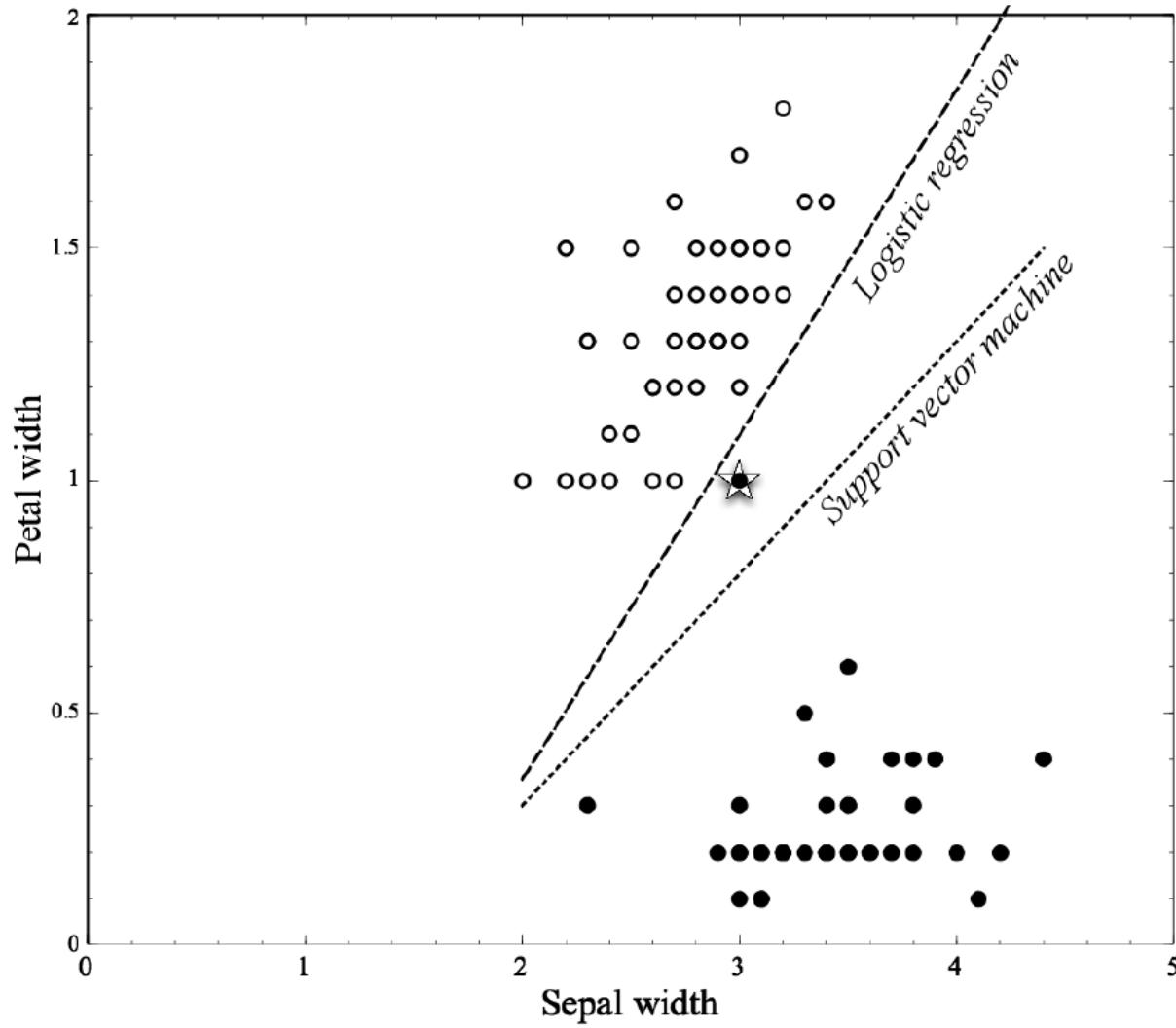
## Optimizing an objective function

What should be our objective in choosing the parameters?  
(Which weights should we choose?)

We need to define an **objective function** that represents our goal sufficiently  
(Optimal solution is found by minimizing or maximizing)

We will consider  
Support Vector Machines  
Linear regression  
Logistic regression

# Mining a linear discriminant for the Iris data set



Ref.

# Linear discriminant functions

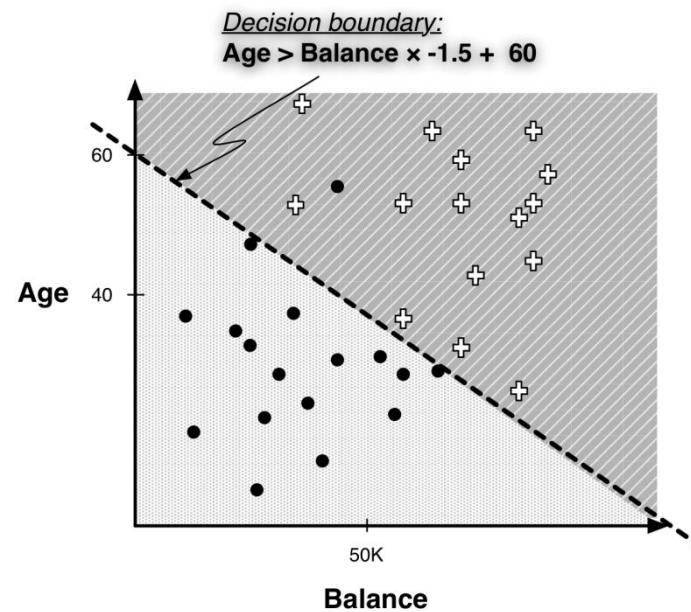
## Scoring and ranking instances

Sometimes, we want some notion of which examples **more or less likely** to belong to a class

- Which customers are most likely to respond to this offer?
- Remember class membership probability

Sometimes, we don't need a precise probability estimate – a **ranking** is sufficient

- Linear discriminant functions provide rankings
- $f(x)$  will be small when  $x$  is near the boundary
- $f(x)$  gives an intuitively satisfying ranking of the instances by their (estimated) likelihood of belonging to the class of interest

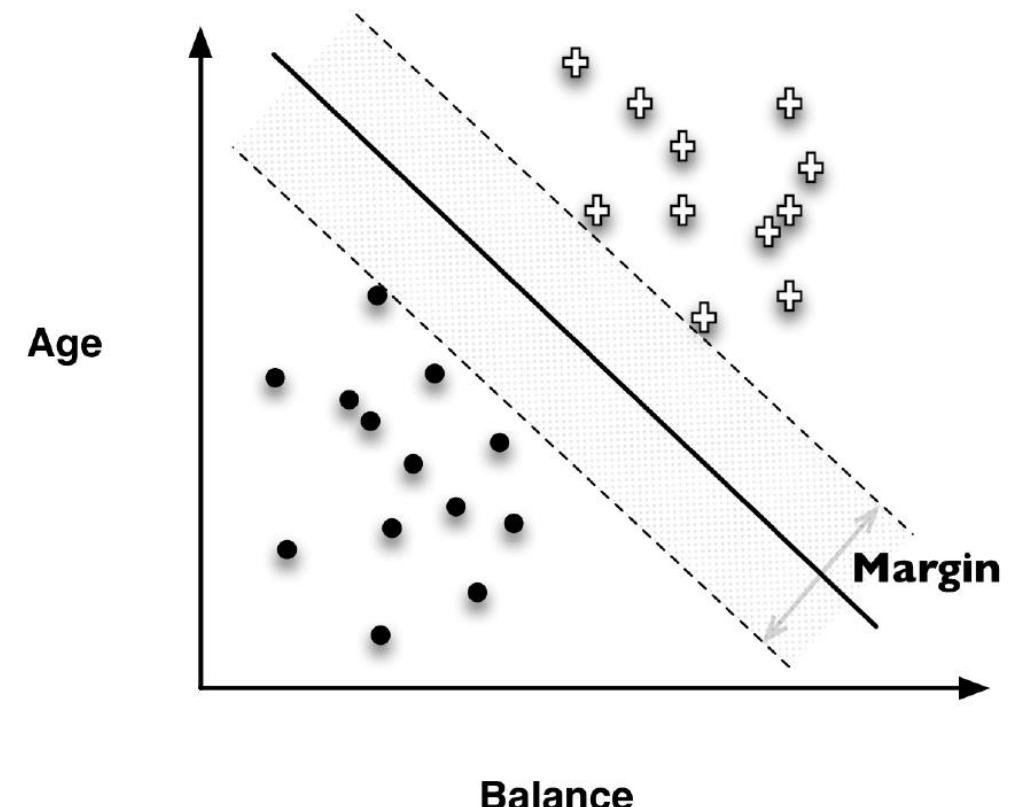


# Support Vector Machines

An intuitive approach

**Support Vector Machines (SVM)** are linear discriminants

- Classify instances based on a linear function of the features
- Objective function based on a simple idea:  
**maximize the margin**
  - Fit the broadest bar between the classes
  - Once the widest bar is found, the linear discriminant will be the center line through the bar
- The margin-maximizing boundary gives **the maximal leeway** for classifying new points



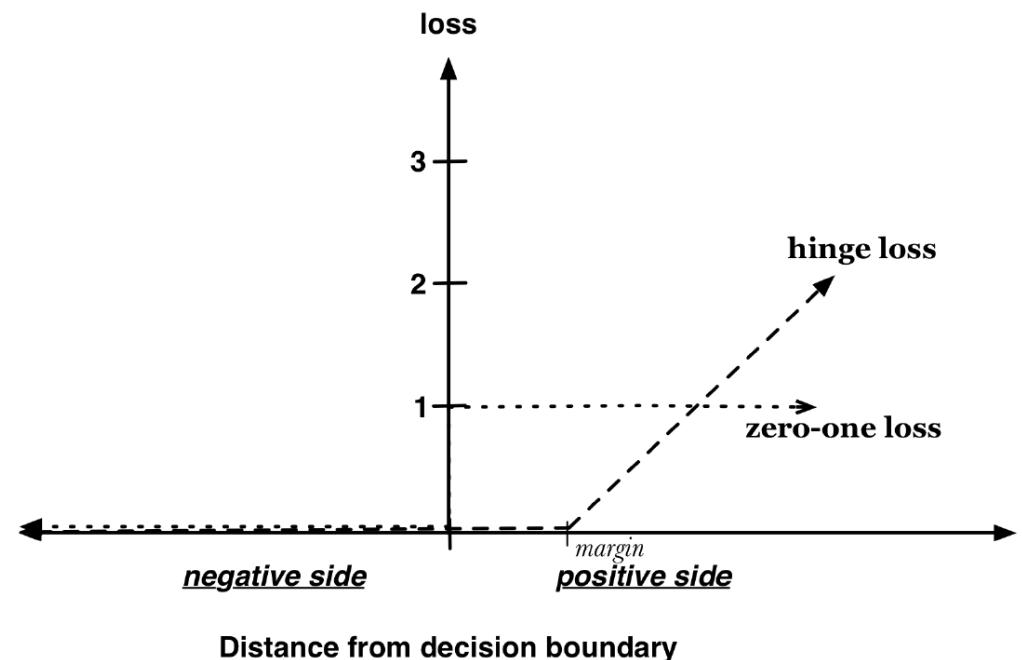
# Support Vector Machines

An intuitive approach in finding a perfect separating function

How to handle data points that are misclassified by the model, i.e., if there is no perfect separating line?

In the objective function, a training point is **penalized** for being on the wrong side of the decision boundary

- If the data are linearly separable, no penalty is incurred and the margin is simply maximized
- If the data are not linearly separable, the best fit is **some balance between a broad margin and a low total error penalty**
- The penalty is proportional to the distance from the decision boundary (hinge loss)

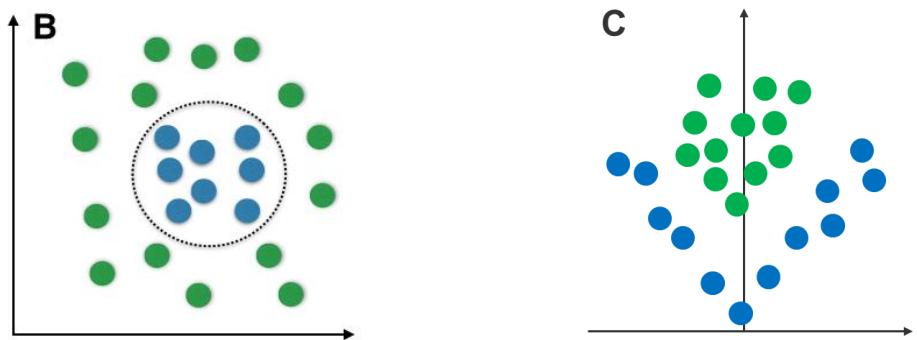


# Support Vector Machines

Modeling non-linear relationships with the kernel trick

We can model non-linear relationships in SVM using the **kernel trick**

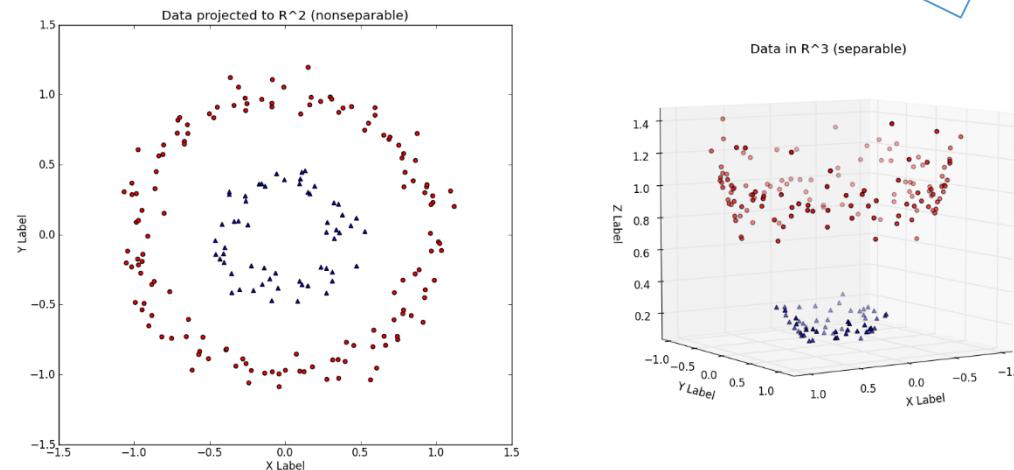
We increase the number of dimensions, to discriminate the classes.



We separate the classes by adding a feature using existing information:

$$z = x^2 + y^2$$

$$z = |x|$$



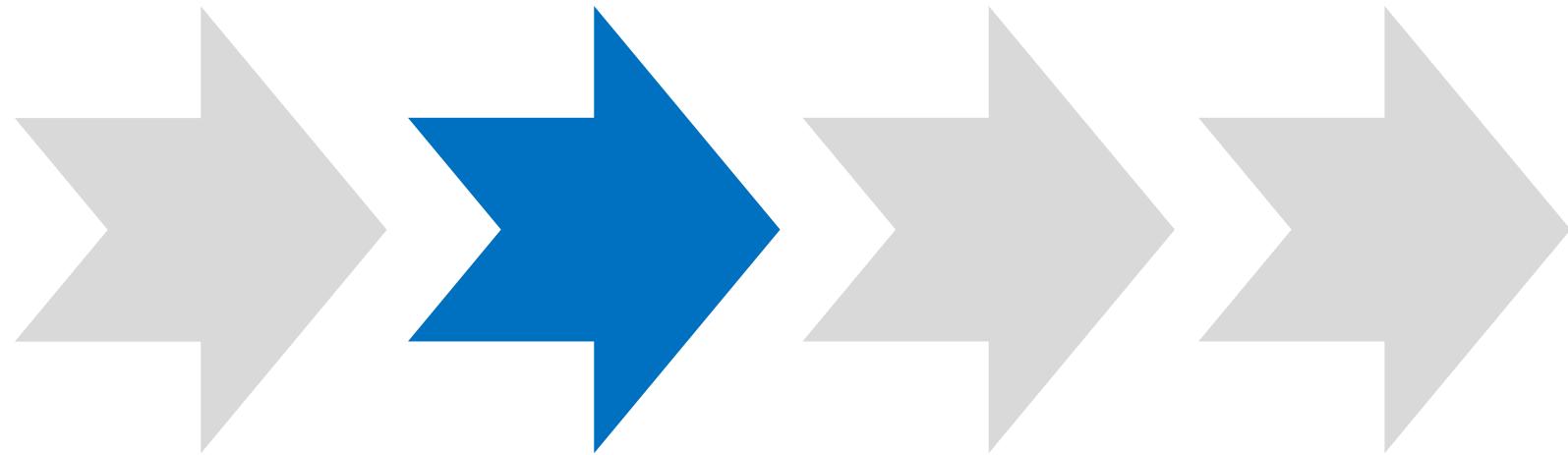
You can fine-tune SVM by choosing

kernels (automatically optimized), like radial basis functional kernel (RBF) or polynomial kernels

and regularization parameter c, which tells the optimizer how broad the margin should be (smoothness vs. correct classification)



## Classification via Mathematical Functions Fitting a Model to Data



(1) Linear  
Classifiers

(2) Linear  
Regression

(3) Logistic  
Regression

(4) Tree-induction  
vs. logistic  
regression

# Linear regression

$$f(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \dots$$

Remember

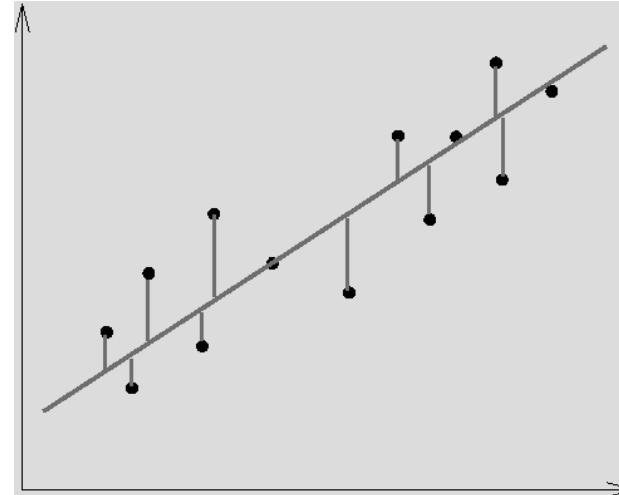
- Which **objective function** should we use to optimize a model's fit to the data?
- Most common choice: how far away are the estimated values from the true values of the training data?
- **Minimize the error of the fitted model**, i.e., minimize the distance between estimated values and true values!

➤ **Regression procedures** choose the model that fits the data best w.r.t. the sum of errors

Sum of **absolute** errors

Sum of **squared** errors

➤ Standard linear regression is convenient (mathematically)!



e.g., sum of errors

$$\sum_{i=1}^n (y_i - \hat{y}_i)^2$$

**A linear regression minimizes the squared error**

Squared error strongly penalizes large errors

Squared error is very sensitive to the data

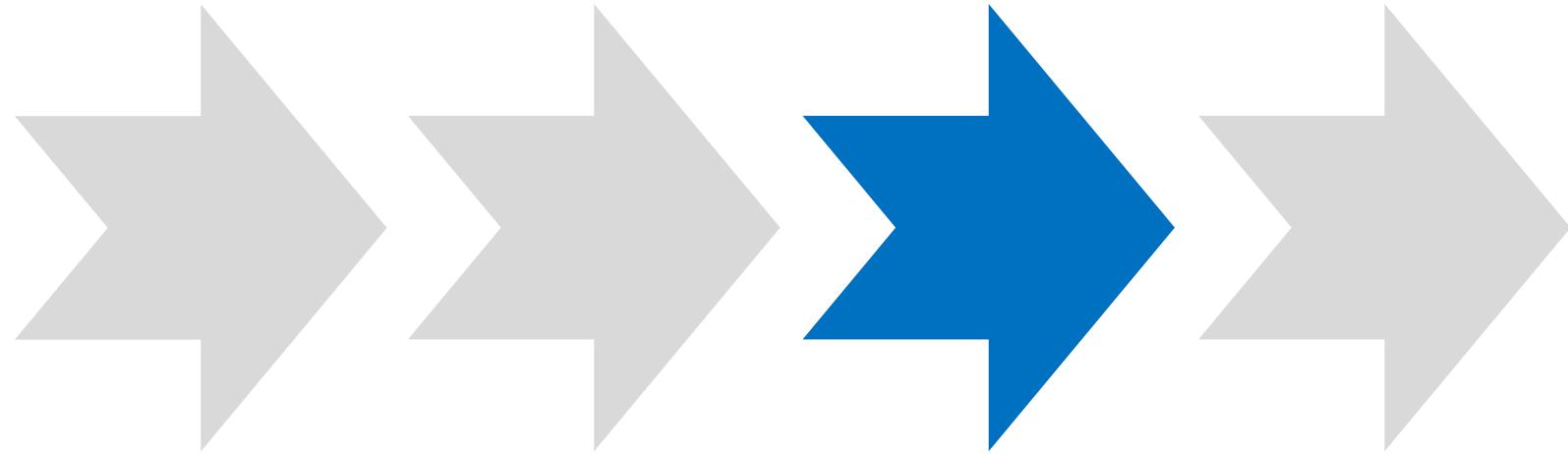
erroneous or outlying data points can severely skew the resulting linear function

For systems that build and apply models automatically, modeling needs to be much more robust

Choose the objective function to optimize with the ultimate business application in mind



## Classification via Mathematical Functions Fitting a Model to Data



(1) Linear  
Classifiers

(2) Linear  
Regression

(3) Logistic  
Regression

(4) Tree-induction  
vs. logistic  
regression

# Logistic regression (1/4)

## Estimation of the Probability

For many applications, we would like to **estimate the probability** that a new instance belongs to the class of interest

*Fraud detection: where is the company's monetary loss expected to be the highest?*

*Spam: What is the probability of the new mail being spam?*

Select different objective function to give accurate estimates of class probability

Well calibrated and discriminative

Recall:

An instance being further from the separating boundary leads to a higher probability of being in one class or the other, and  $f(x)$  gives the distance from the separating boundary

But a probability ranges from zero to one.

Be careful: Distinguish between **target variable** and **probability of class membership!**

# Logistic regression (2/4)

## Likelihood and odds

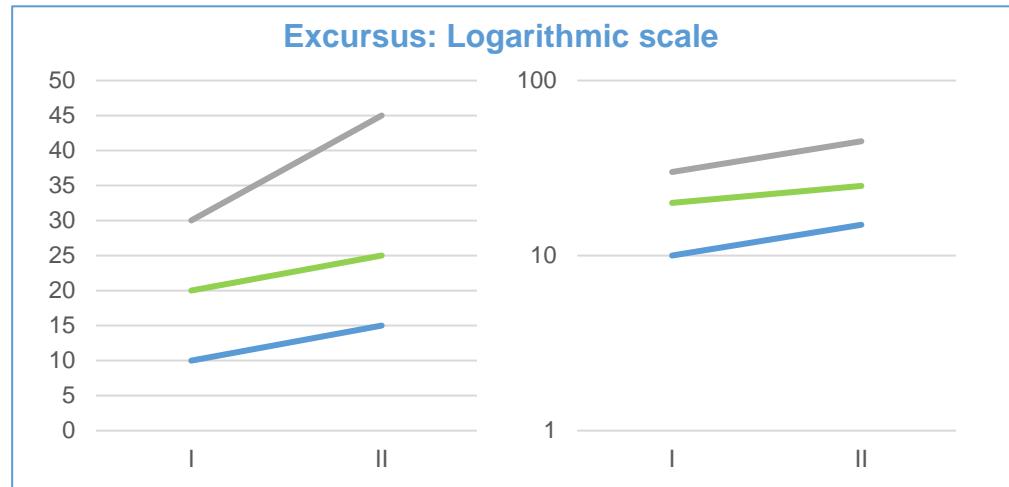
The likelihood of an event can be expressed by **odds**

The **odds of an event** is the ratio of the probability of the event occurring to the probability of the event not occurring

Log-odds convert the scale to  $-\infty$  to  $+\infty$

| Probability | Odds              | Corresponding log-odds* |       |
|-------------|-------------------|-------------------------|-------|
| 0.5         | 50:50 or 1        | 0                       | 0     |
| 0.9         | 90:10 or 9        | 2.19                    | 0,95  |
| 0.999       | 999:1 or 999      | 6.9                     | 3,00  |
| 0.01        | 1:99 or 0.0101    | -4.6                    | -2,00 |
| 0.001       | 1:999 or 0.001001 | -6.9                    | -3,00 |

\* $\ln(x)$   
is used with  
 $\log_{10}(x)$



## Logistic regression model:

$f(x)$  is used as a measure of the log-odds of the “event” of interest

$f(x)$  is an estimation of the log-odds that  $x$  belongs to the positive class

# Logistic regression (3/4)



How to translate log-odds into the probability of class membership?

$\hat{p}_+(x)$  represents the model's estimate of the probability of class membership of a data item by feature vector  $x$

$+$  is the class for the (binary) event we are modeling

$1 - \hat{p}_+(x)$  is the estimated probability of the event *not* occurring

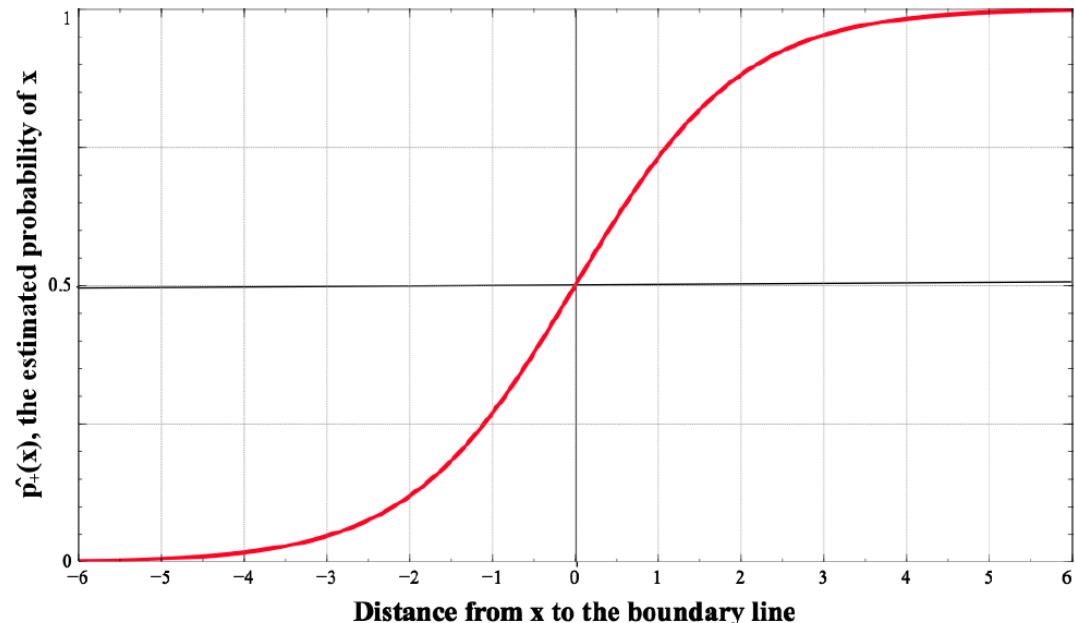
$$\ln \left( \frac{\hat{p}_+(x)}{1 - \hat{p}_+(x)} \right) = f(x) = w_0 + w_1 x_1 + w_2 x_2 + \dots$$

Solve for  $\hat{p}_+(x)$  yields  $\hat{p}_+(x) = \frac{1}{1 + e^{-f(x)}}$

Probability at distance  $x = 0$  is 0.5

Probability varies approximately linearly near to the decision boundary, but then approaches certainty farther away

Determine the slope of the almost linear part within fitting



# Logistic regression (4/4)

What does the objective function look like?

Ideally, any positive example  $x_+$  would have  $\hat{p}_+(x_+) = 1$  and any negative example  $x_-$  would have  $\hat{p}_-(x_-) = 0$

Probabilities are never pure when real-world data is considered

**Compute the likelihood** of a particular labeled example given a set of parameters  $w$  that produces class probability estimates  $\hat{p}_+(x)$

The  $g$  function gives the model's estimated probability of seeing  $x$ 's actual class given  $x$ 's features

For different parameterized models, sum the  $g$  values across all instances in a labeled (training) dataset to get "the maximum likelihood" model

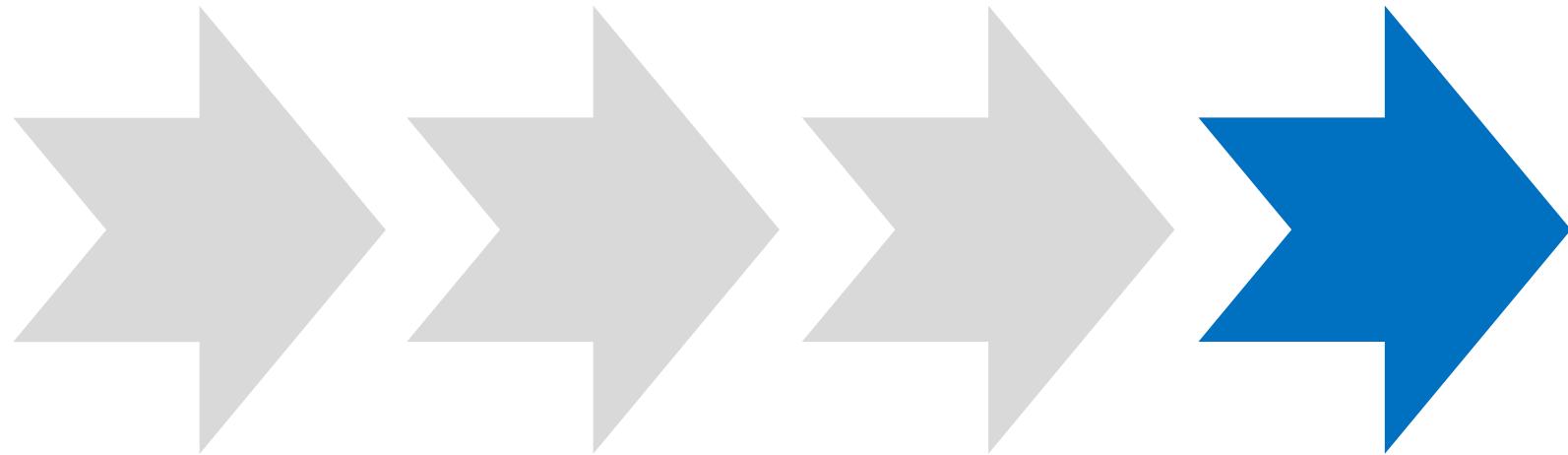
The identified maximum likelihood model "on average" gives the highest probabilities to the positive examples and the lowest probabilities to the negative examples.

$$g(\mathbf{x}, \mathbf{w}) = \begin{cases} \hat{p}_+(\mathbf{x}) & \text{if } \mathbf{x} \text{ is a +} \\ 1 - \hat{p}_-(\mathbf{x}) & \text{if } \mathbf{x} \text{ is a -} \end{cases}$$

$$\arg \max_w g(x, w)$$



## Classification via Mathematical Functions Fitting a Model to Data



(1) Linear  
Classifiers

(2) Linear  
Regression

(3) Logistic  
Regression

(4) Tree-induction  
vs. logistic  
regression

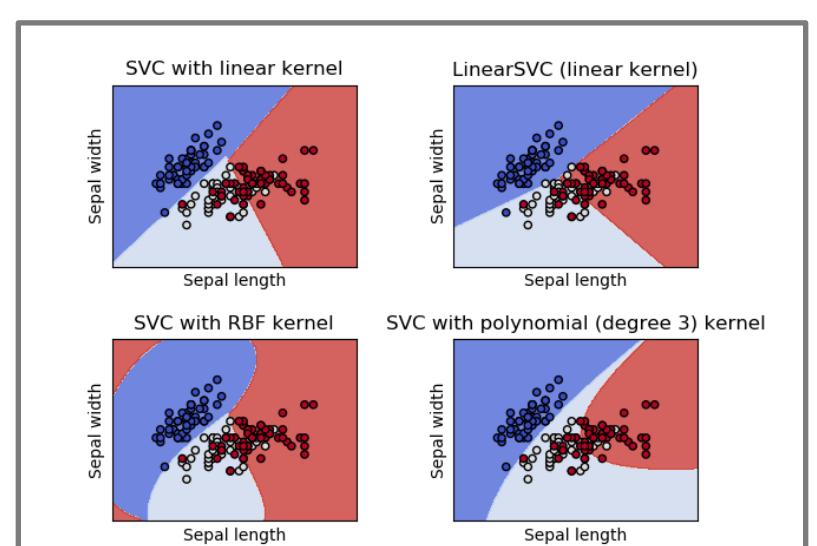
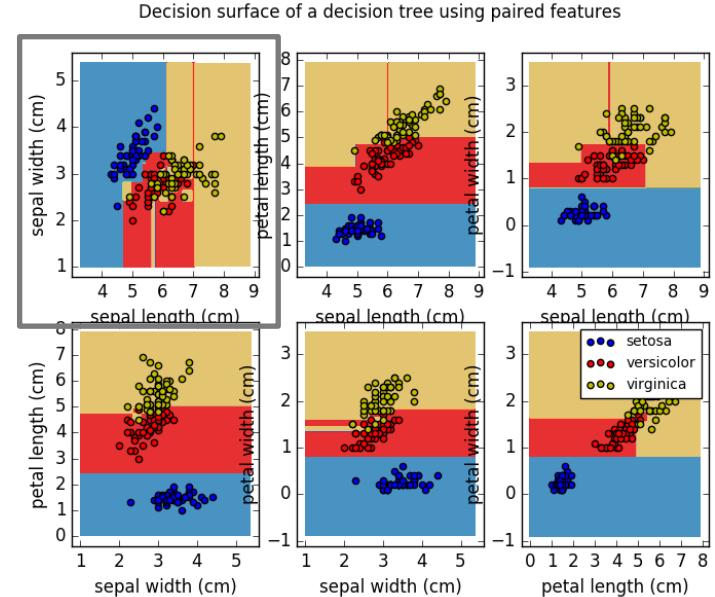
# Tree induction vs. linear classifier (in general)



Important differences between trees and linear classifiers

- A classification tree uses decision boundaries that are **perpendicular** to the instance-space axes.
- The linear classifier can use **decision boundaries of any direction** or orientation
- A classification tree is a “**piecewise** classifier” that segments the instance space recursively → cut in arbitrarily small regions possible.
- The linear classifier places **a single decision** surface through the entire space.

Which of these characteristics are a better match to a given data set?



# Tree induction vs. logistic regression

Example: Breast Cancer Dataset

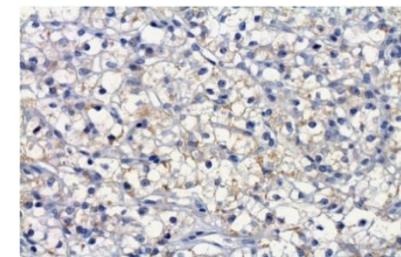
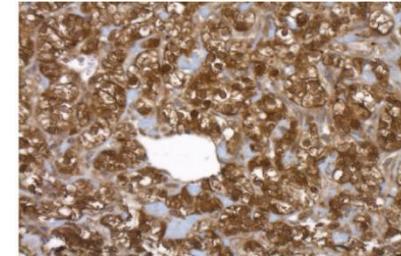
Consider the background of the stakeholders

- A decision tree may be considerably **more understandable** to someone without a strong background in statistics
- Data Mining team does not have the ultimate say how models are used or implemented!

## Example: Wisconsin Breast Cancer Dataset

[http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

- Each record describes characteristics of a cell nuclei image, which has been labeled as either “benign” or “malignant” (**cancerous**)
- Ten fundamental characteristics were extracted and summarized in a mean (\_mean), standard error (\_SE) and mean of the three largest values (\_worst)  
→ 30 measured attributes
- 357 benign images and 212 malignant images



From Mu et al. (2011) doi:10.1038/ncomms1332

# Tree induction vs. logistic regression

Example: Breast Cancer Dataset

## Results of logistic regression

Weights of linear model

Ordered from highest to lowest

Performance: only six mistakes on the entire dataset, accuracy 98.9%

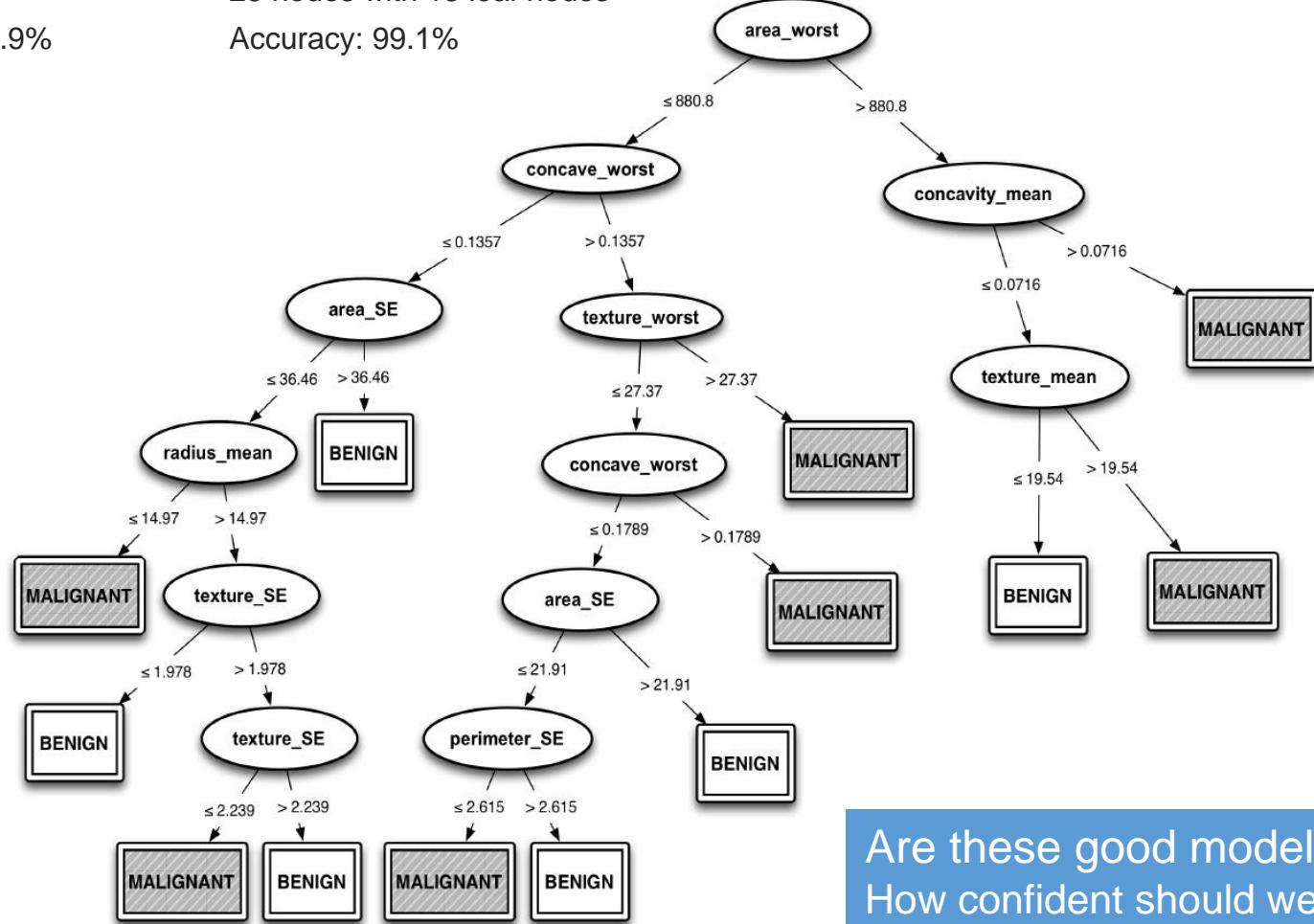
| Attribute         | Weight |
|-------------------|--------|
| Smoothness_worst  | 22.30  |
| Concave_mean      | 19.47  |
| Concave_worst     | 11.68  |
| Symmetry_worst    | 4.99   |
| Concavity_worst   | 2.86   |
| Concavity_mean    | 2.34   |
| Radius_worst      | 0.25   |
| Texture_worst     | 0.13   |
| Area_SE           | 0.06   |
| Texture_mean      | 0.03   |
| Texture_SE        | -0.29  |
| Compactness_mean  | -7.10  |
| Compactness_SE    | -27.87 |
| $w_0$ (intercept) | -17.70 |

## Comparison with classification tree from the same dataset

Weka's J48 implementation

25 nodes with 13 leaf nodes

Accuracy: 99.1%



Are these good models?  
How confident should we be  
in this evaluation?

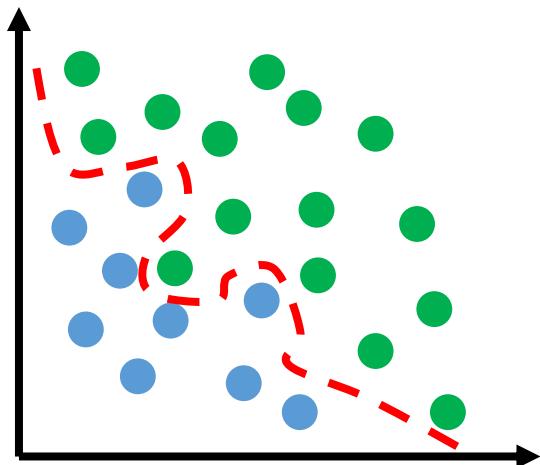
# How to avoid overfitting? – Next Lesson

Introduction

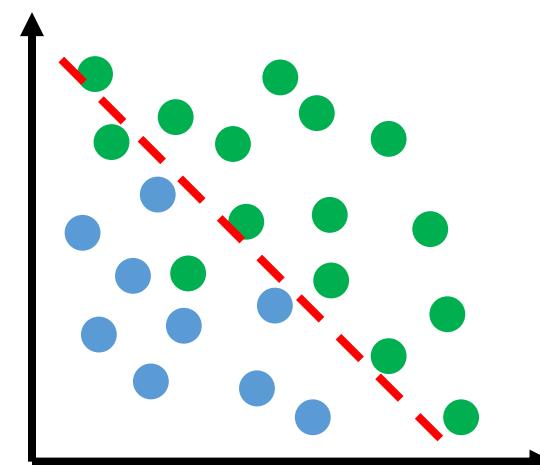
Fundamental trade-off in DM between **overfitting** and **generalization**

If we allow ourselves enough flexibility in searching, we *will* find patterns

Unfortunately, these patterns may be just occurrences by chance



**Overfitting:** finding chance occurrences in data that *look like* interesting patterns, but which do *not generalize*



We are interested in patterns that **generalize**, i.e., that predict well for instances that we have not yet observed

## Fragen?

- ✓ Tree induction vs. linear classifier (in general)
  
- ✓ Linear regression
- ✓ Logistic regression
- ✓ Tree-induction vs. logistic regression

# Recommended reading

## Fitting a Model:

Provost, F., Data Science for Business  
Fawcett, T. Chapter 4

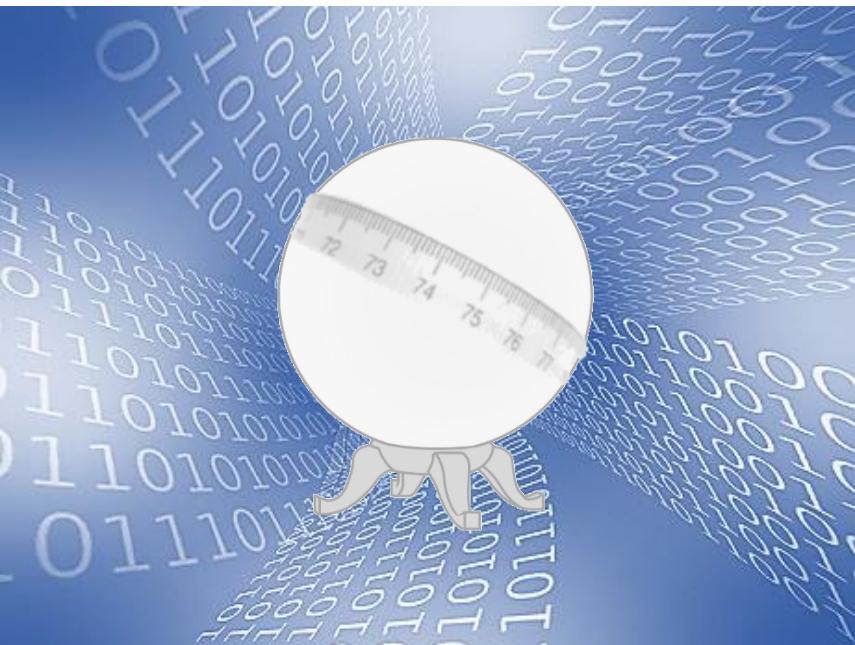
Berthold et al. Guide to Intelligent Data Analysis  
Chapter 8.3

Almost all introductory books on statistics include regression

## How to avoid Overfitting (next lesson):

Provost, F., Data Science for Business  
Fawcett, T. Chapter 5

Berthold et al. Several subchapters



# Business Intelligence

## 12 How to avoid overfitting?

Prof. Dr. Bastian Amberg  
(summer term 2024)

26.6.2024

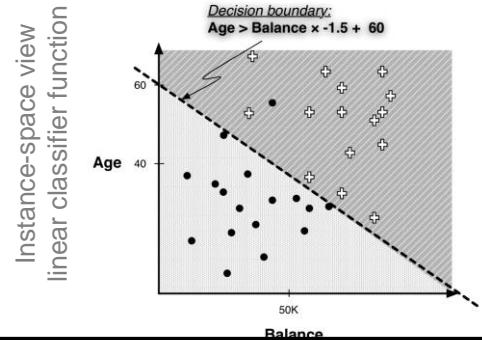
# Schedule

|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study |                  |               |         |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|------------|------------------|---------------|---------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |            |                  | Python-Basics | Chap. 1 |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |            |                  |               | Chap. 2 |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  |            |            |                  |               | Chap. 3 |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining Introduction  |            |                  |               |         |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |            | Python-Analytics | Chap. 1       |         |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization I            |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam |            |                  | Chap. 2       |         |  |  |
|           | W7                | 29.5. | Data Visualization II                               |                                   | 31.5. |   |            |                  |               |         |  |  |
|           | W8                | 5.6.  | Data Preparation                                    |                                   | 7.6.  | Predictive Modeling I (10:00 -12:00)  |            | BI-Project       | Start         |         |  |  |
|           | W9                | 12.6. | Predictive Modeling II                              |                                   | 14.6. | Python-Analytics-Online Exercise  |            |                  |               |         |  |  |
|           | W10               | 19.6. | Guest Lecture Dr. Ionescu                           |                                   | 21.6. | Fitting a Model   |            |                  |               |         |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |            |                  |               |         |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I                                  |            |                  |               |         |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |            |                  |               |         |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |            |                  | End           |         |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin, 31.7.'24<br>Klausur 2.Termin, 2.10.'24                                      |            | Projektbericht   |               |         |  |  |

# Last Lesson

## Predictive Modeling Classification via Mathematical Functions – Fitting a Model to Data

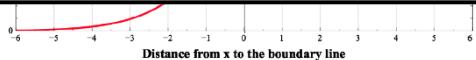
- We **specify the structure of the model**, but leave certain numeric parameters unspecified
- Data Mining calculates the best parameter values given a particular set of training data
- The form of the model and the attributes is specified
- The goal of DM is to tune the parameters so that the model fits the data as good as possible (**parameter learning**)



Kahoot-Fragen  
[www.kahoot.it](http://www.kahoot.it)

(über Smartphone oder Laptop)  
PIN folgt

Diese Folie ist nach der Vorlesung vollständig sichtbar.



The identified maximum likelihood model “on average” gives the highest probabilities to the positive examples and the lowest probabilities to the negative examples.

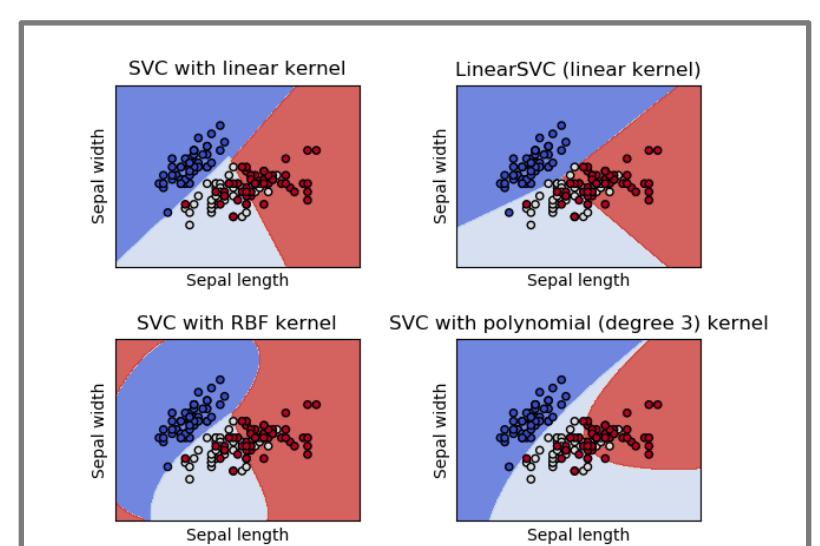
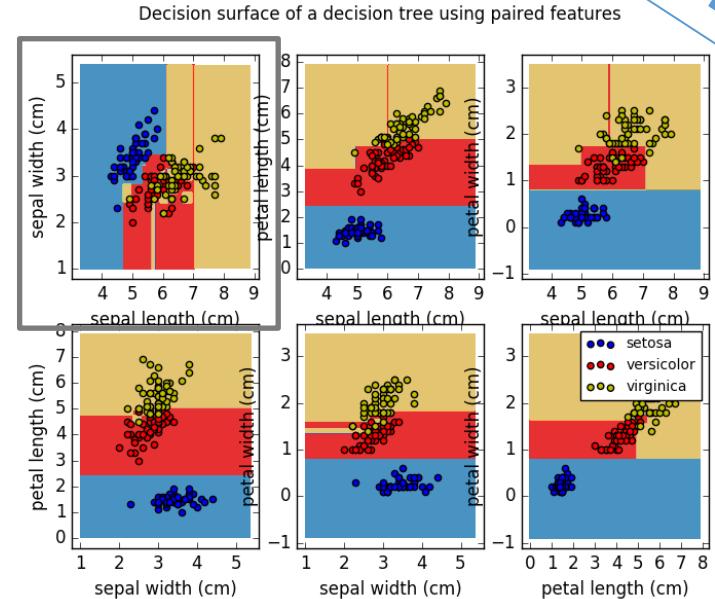
Ref.

# Tree induction vs. linear classifier (in general)

Important differences between trees and linear classifiers

- A classification tree uses decision boundaries that are **perpendicular** to the instance-space axes.
- The linear classifier can use **decision boundaries of any direction** or orientation
- A classification tree is a “**piecewise** classifier” that segments the instance space recursively → cut in arbitrarily small regions possible.
- The linear classifier places **a single decision** surface through the entire space.

Which of these characteristics are a better match to a given data set?



# Tree induction vs. logistic regression

Example: Breast Cancer Dataset

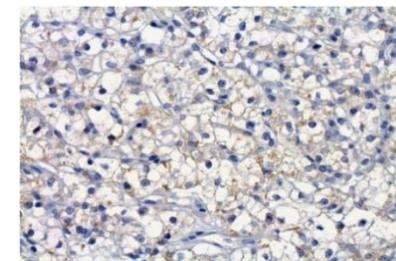
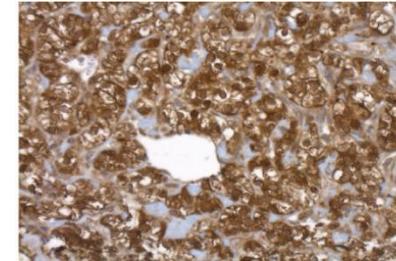
Consider the background of the stakeholders

- A decision tree may be considerably **more understandable** to someone without a strong background in statistics
- Data Mining team does not have the ultimate say how models are used or implemented!

## Example: Wisconsin Breast Cancer Dataset

[http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

- Each record describes characteristics of a cell nuclei image, which has been labeled as either “benign” or “malignant” (**cancerous**)
- Ten fundamental characteristics were extracted and summarized in a mean (\_mean), standard error (\_SE) and mean of the three largest values (\_worst)  
→ 30 measured attributes
- 357 benign images and 212 malignant images



From Mu et al. (2011) doi:10.1038/ncomms1332

# Tree induction vs. logistic regression

Example: Breast Cancer Dataset

## Results of logistic regression

Weights of linear model

Ordered from highest to lowest

Performance: only six mistakes on the entire dataset, accuracy 98.9%

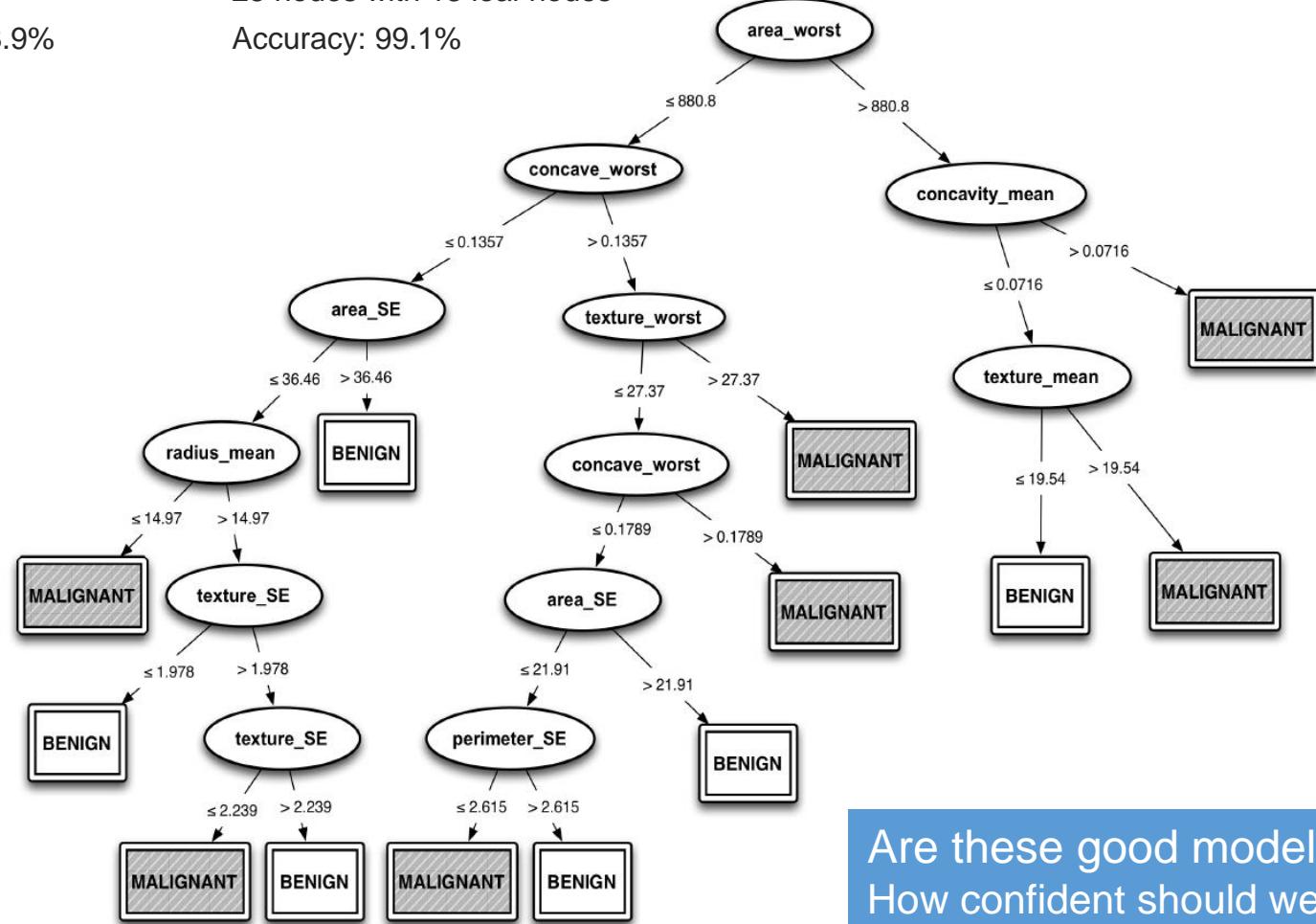
| Attribute         | Weight |
|-------------------|--------|
| Smoothness_worst  | 22.30  |
| Concave_mean      | 19.47  |
| Concave_worst     | 11.68  |
| Symmetry_worst    | 4.99   |
| Concavity_worst   | 2.86   |
| Concavity_mean    | 2.34   |
| Radius_worst      | 0.25   |
| Texture_worst     | 0.13   |
| Area_SE           | 0.06   |
| Texture_mean      | 0.03   |
| Texture_SE        | -0.29  |
| Compactness_mean  | -7.10  |
| Compactness_SE    | -27.87 |
| $w_0$ (intercept) | -17.70 |

## Comparison with classification tree from the same dataset

Weka's J48 implementation

25 nodes with 13 leaf nodes

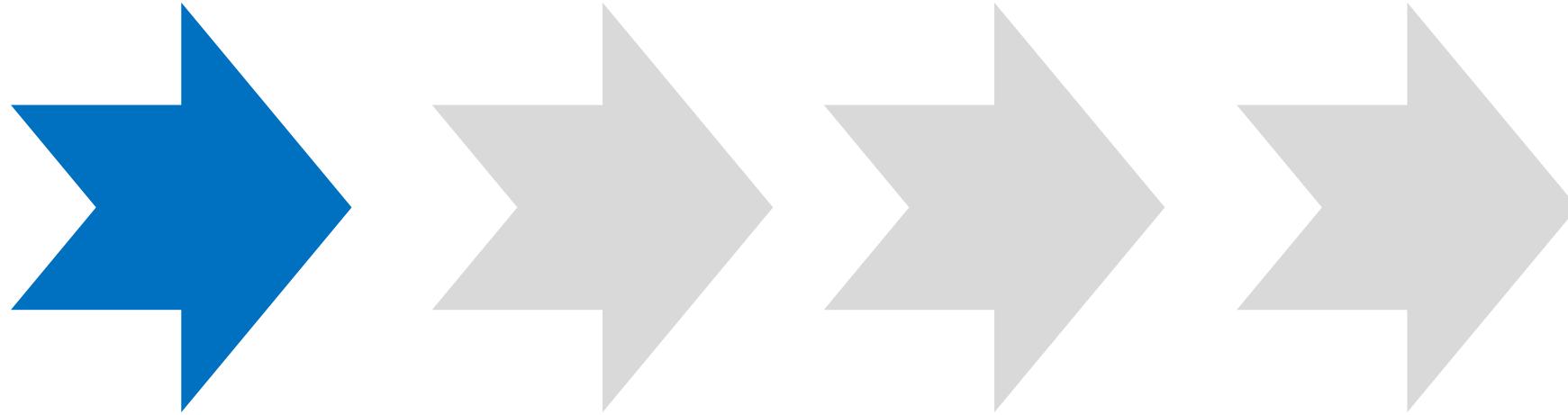
Accuracy: 99.1%



Are these good models?  
How confident should we be  
in this evaluation?



## How to avoid overfitting?



(1) Generalization  
and Overfitting

(2) From  
holdout  
evaluation to  
cross-  
validation

(3) Learning  
curves

(4) Overfitting  
avoidance and  
complexity control

# How to avoid overfitting?

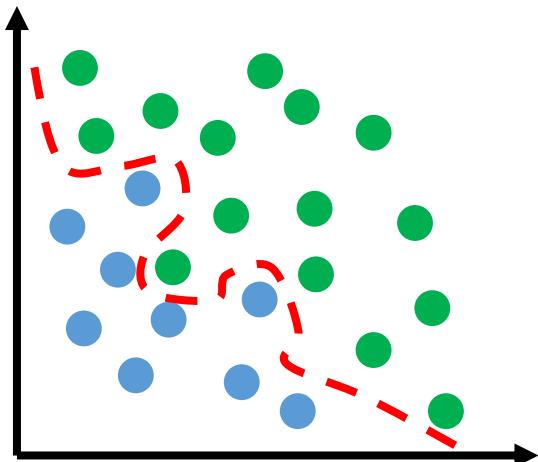


Introduction

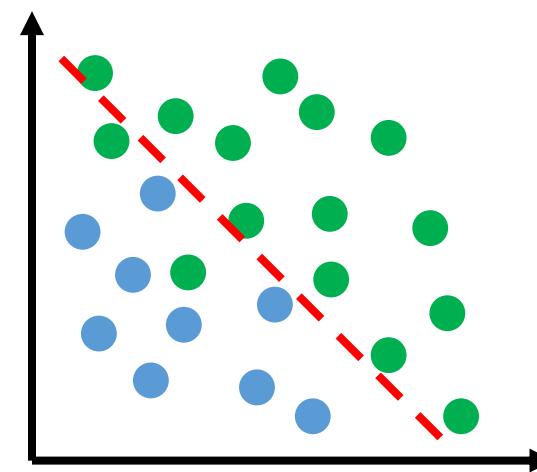
Fundamental trade-off in DM between **overfitting** and **generalization**

If we allow ourselves enough flexibility in searching, we *will* find patterns

Unfortunately, these patterns may be just occurrences by chance



**Overfitting:** finding chance occurrences in data that *look like* interesting patterns, but which do *not generalize*



We are interested in patterns that **generalize**, i.e., that predict well for instances that we have not yet observed

# Generalization

Counterexample: The table model

We can **always** build a perfect model!

Store the feature vector for each customer who churned (*table model*)

Look the customer up when determining the likelihood of churning

| College | Income | Long Calls per Month | Average Call Duration | ... | Leave? |
|---------|--------|----------------------|-----------------------|-----|--------|
| Yes     | 3,000  | 23                   | 28                    | ... | Yes    |
| Yes     | 1,424  | 2                    | 120                   | ... | No     |
| No      | 6,201  | 42                   | 70                    | ... | No     |
| ...     | ...    | ...                  | ...                   | ... | ...    |
| No      | 543.12 | 20                   | 140                   |     | Yes    |



## Example: Churn data set

Historical data on customers who have stayed with the company, and customers who have departed within six months of contract expiration

Task:  
build a model to distinguish customers who are likely to churn based on some features

We test the model based on historical data, and the model is **100% accurate**, identifying correctly all the churners as well as the non-churners

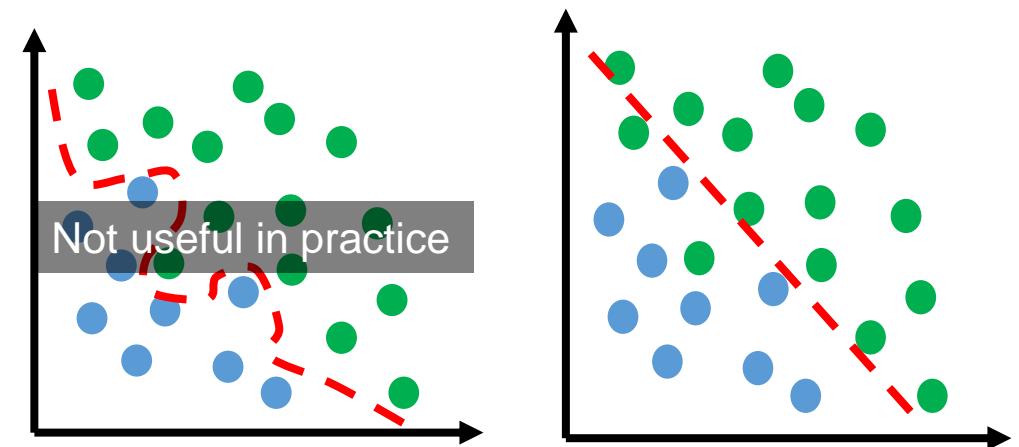
# Generalization

A table model memorizes the training data and performs  
**no generalization**

Useless in practice! Previously unseen customer's  
would all end up with "0% likelihood of churning"

**Generalization** is the property of a model or modeling  
process whereby the model applies to data **that were not  
used to build the model**

If models do not generalize at all, they fit perfectly to  
the training data → they overfit.



Our imperative:  
DM needs to create models that **generalize** beyond  
training data

# Overfitting

„If you torture the data long enough, it will confess.“ (Ronald Coase)



**Overfitting** is the tendency of DM procedures to tailor models to the training data, at the expense of generalization to previously unseen data points.

All DM procedures tend to overfitting

Trade-off between **model complexity** and the possibility of overfitting

Recognize overfitting and manage complexity in a principled way

## Complexity:

Number of decisions to be made by the model, i.e., amount of parameters to be estimated;

Examples?

# Holdout data

Data for testing; data for training

Evaluation on **training data** provides no assessment of how well the model generalizes to unseen cases

Rational:

„Hold out“ some data for which we know the value of the target variable, but which will not be used to build the model → „lab test“

Predict the values of the „**holdout data**“ (aka „**test set**“) with the model and **compare** them with the hidden true values → generalization performance

There is likely to be a difference between the model's accuracy („in-sample“ accuracy) and the model's generalization accuracy

Real world data  
A B A A A B A A B B

Training Data | Hold out data  
A B A A A B | A A B B

Predicted Value | A **B** B B  
Hold out value | A A B B

Where have we already done this?

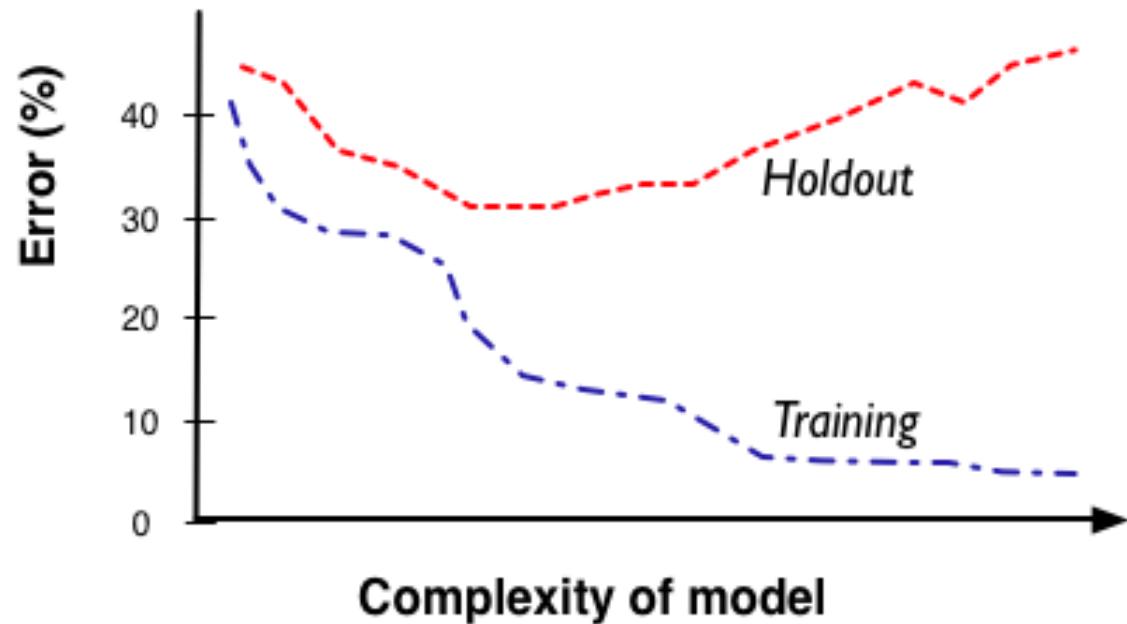
# Fitting graph

Error / Complexity



A fitting graph shows the **accuracy of a model** as a function of complexity  
( accuracy = 1 – error )

Generally, there will be more overfitting as one allows the model to be more complex



# Overfitting in tree induction (1/2)

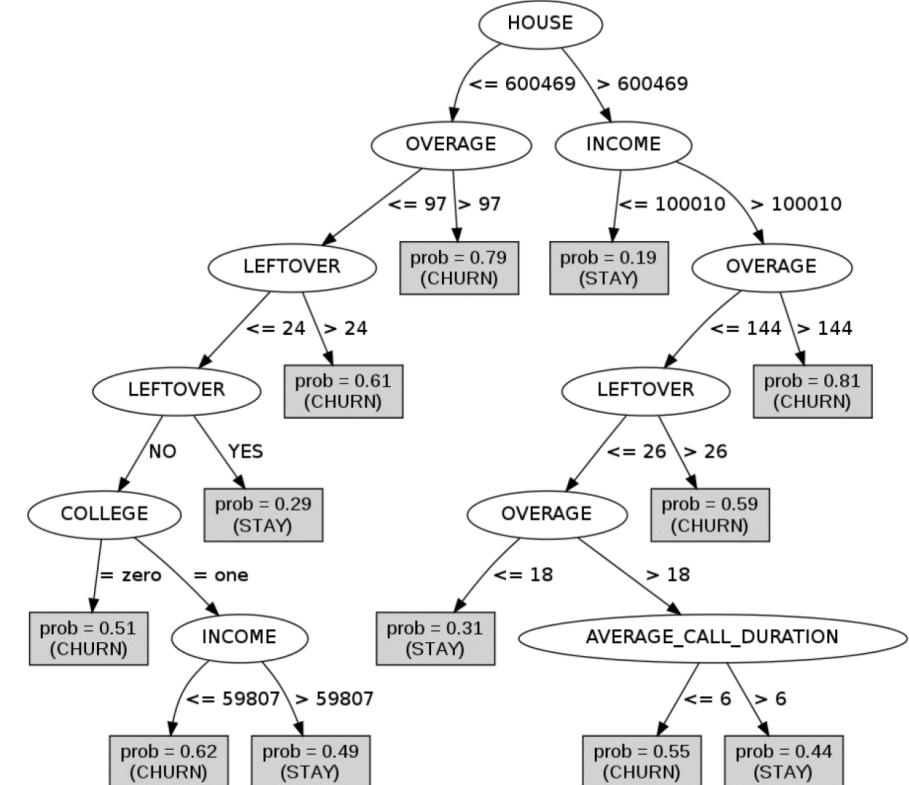


Recall **tree induction**:

find important, predictive individual attributes recursively to smaller and smaller data subsets

- Eventually, the subsets will be pure – we have found the leaves of our decision tree
- The accuracy of this tree will be perfect!
- This is the same as the table model, i.e., an **extreme example of overfitting**
- This tree should be slightly better than the lookup table, because every previously unseen instance also will arrive at some classification rather than just failing to match

Useful for comparison of how well the accuracy on the training data tends to correspond to the accuracy on test data



prob = estimated probabilities of churning

# Overfitting in tree induction (2/2)

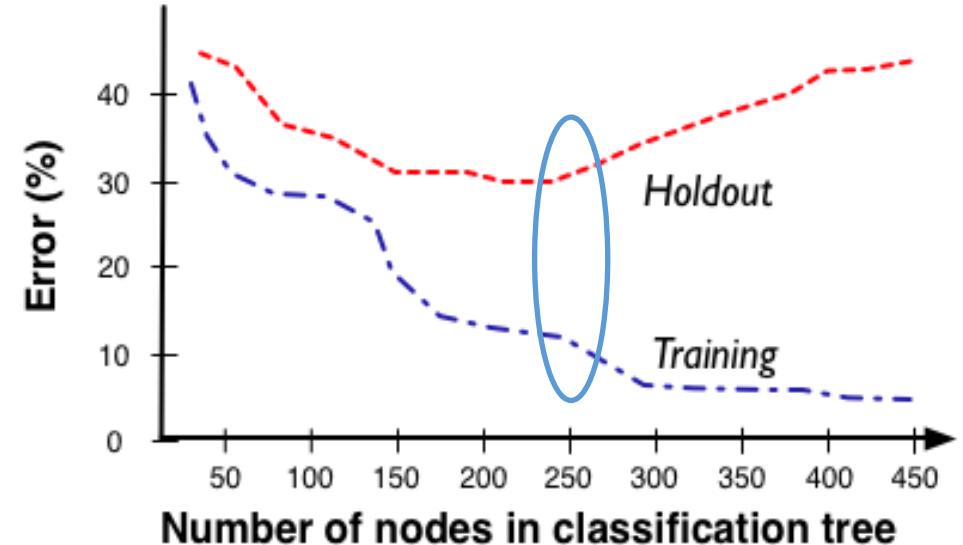


Generally:

A procedure that grows trees until the leaves are pure tends to overfit

If allowed to grow without bound, decision trees can fit any data to arbitrary precision

The **complexity of a tree** lies in the number of nodes



# Overfitting in parametric learning (1/2)

[Next Lesson](#)

Freie Universität



Berlin

There are different ways to allow more or less complexity in mathematical functions

Add more variables (more attributes):

$$f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + w_3x_3$$

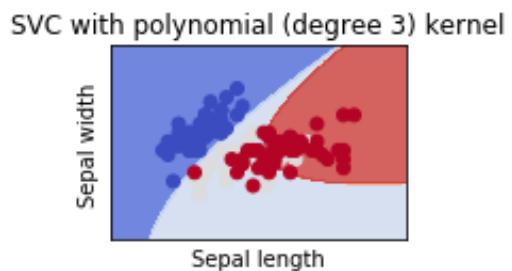
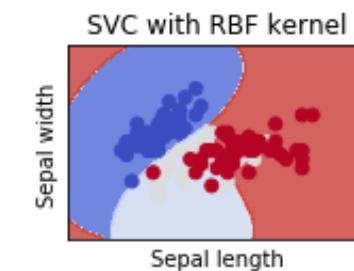
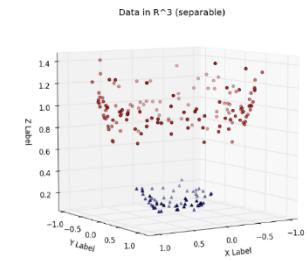
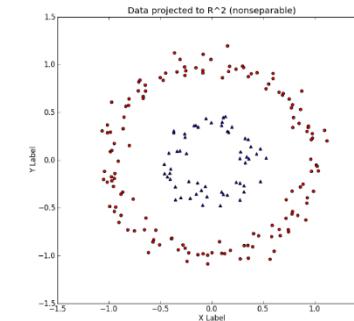
$$f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5$$

Add attributes that are non-linear, i.e.,  $x_5 = x_2/x_3$  or  $x_4 = x_1^2$   
(see kernel trick from SVM)

As you **increase the dimensionality**, you can perfectly fit larger and larger sets of arbitrary points

Often, modelers carefully prune the attributes in order to avoid overfitting → manual selection

Automatic feature selection



# Overfitting in parametric learning (2/2)

Next Lesson

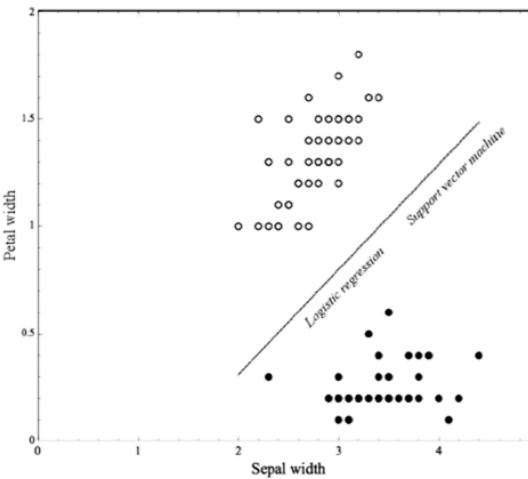
Freie Universität



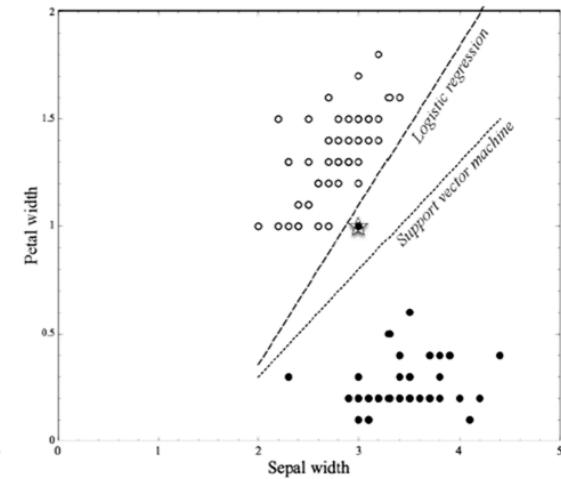
Berlin

## Iris Example

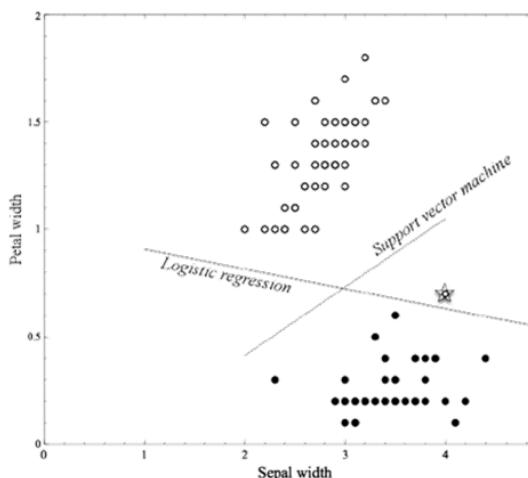
- a) Original Iris data set  
both logistic regression and support vector machines place separating boundaries in the middle
- b) A single new example has been added (3,1) ●  
logistics regression still separates the groups perfectly, while the SVM line barely moves at all
- c) A different outlier has been added (4,0.7) ○  
again, SVM only moves very little – logistic regression appears to be overfitting considerably
- d) Add a squared term of the sepal width  
More flexibility in fitting the data – separating line becomes a parabola



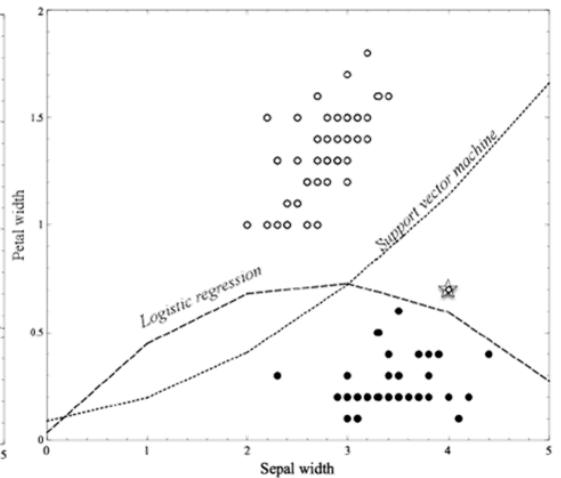
(a) Original dataset. The data are cleanly separable. Both LR and SVM impose the same boundary.



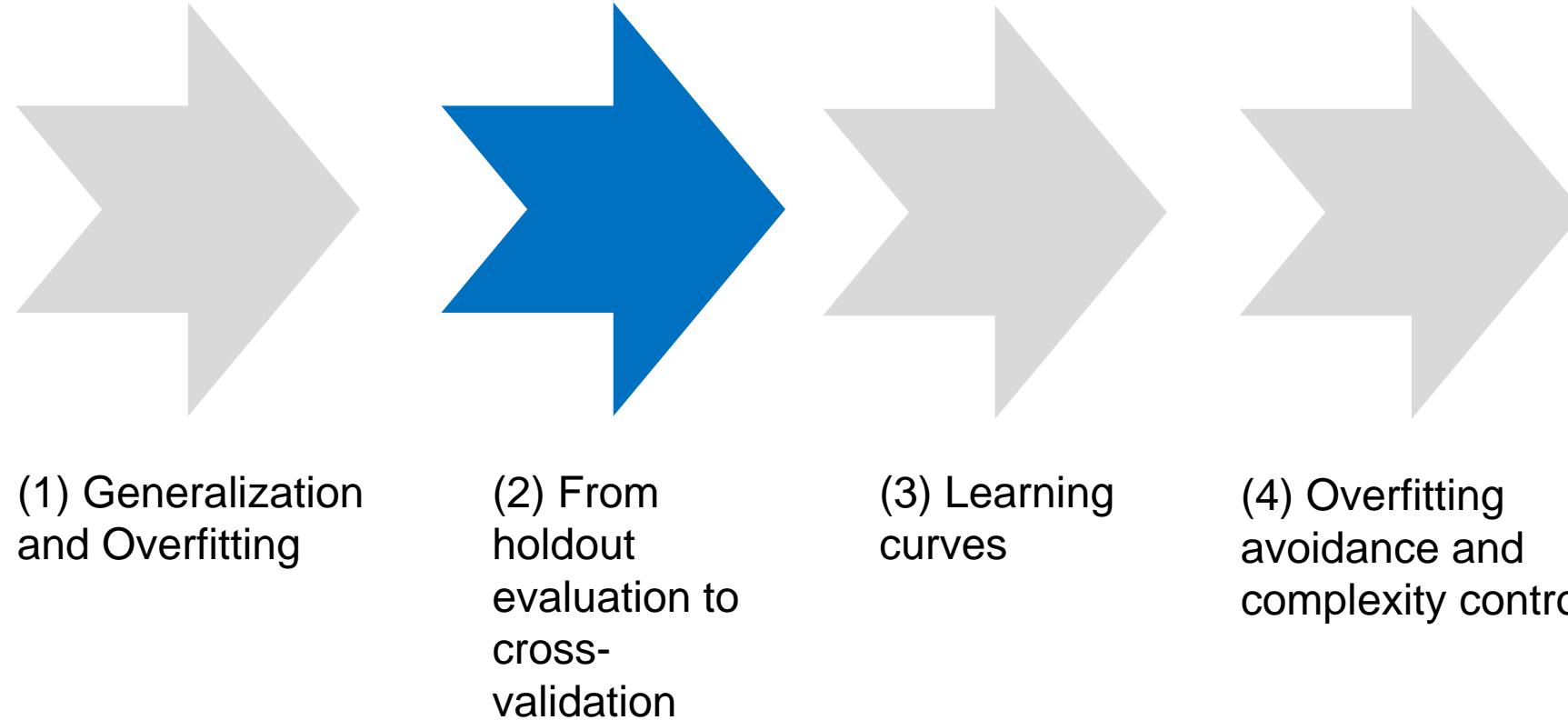
(b) Outlier Setosa (filled dot shown by star) added at (3,1) and resulting separators.



(c) Outlier Versicolor (circle with star) added at (4,0.7) and resulting separators.



(d) Same data as in (c) but with squared Sepal width term added.



# Holdout training and testing

**Cross-validation** is a more sophisticated training and testing procedure

Not only a simple estimate of the generalization performance, but also some statistics on the estimated performance (mean, variance, ...)

How does the performance vary across data sets?

→ assessing confidence in the performance estimate

Cross-validation computes its estimates over all the data by **performing multiple splits** and systematically swapping out samples for testing

Split a data set into  $k$  (equal sized) partitions called **folds** ( $k = 5$  or  $10$ )

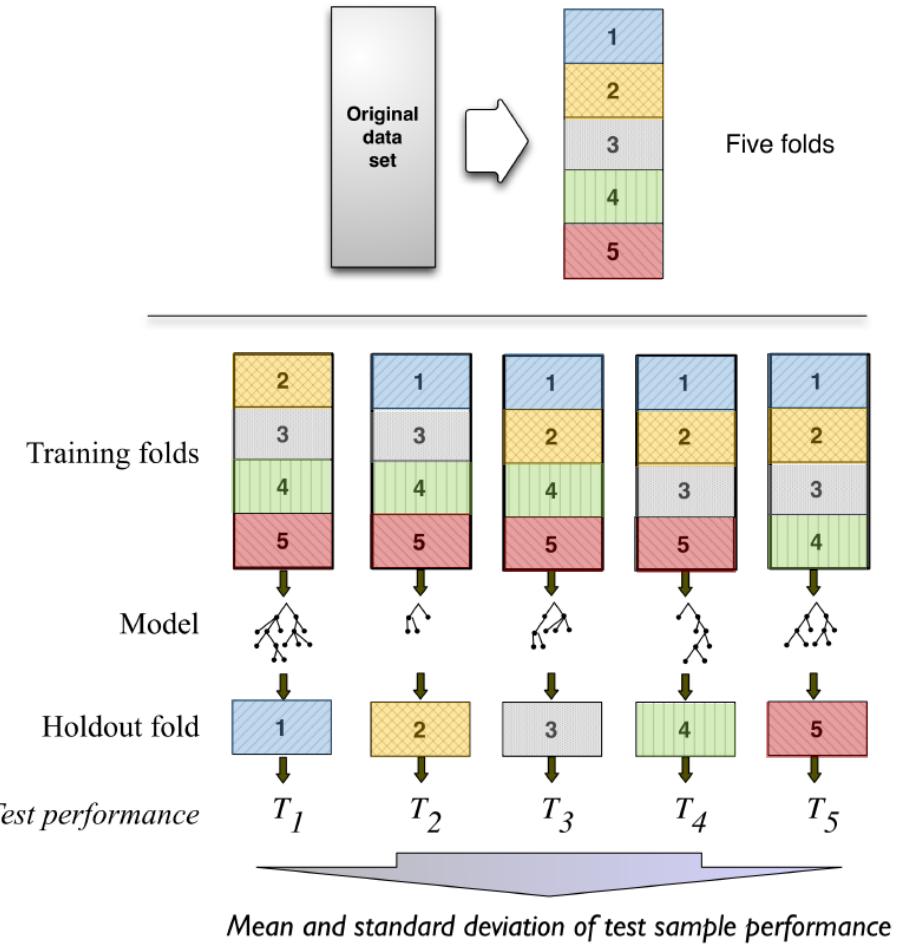
Iterate training and testing  $k$  times

In each iteration, a different fold is chosen as the test data. The other  $k - 1$  folds are combined to form the training data.

## Illustration of cross-validation

Every example will have been used only once for testing but  $k - 1$  times for training

Compute average and standard deviation from  $k$  folds



# Cross-validation for the churn data set

Recall churn dataset with an accuracy of 73%

Cross-validation:

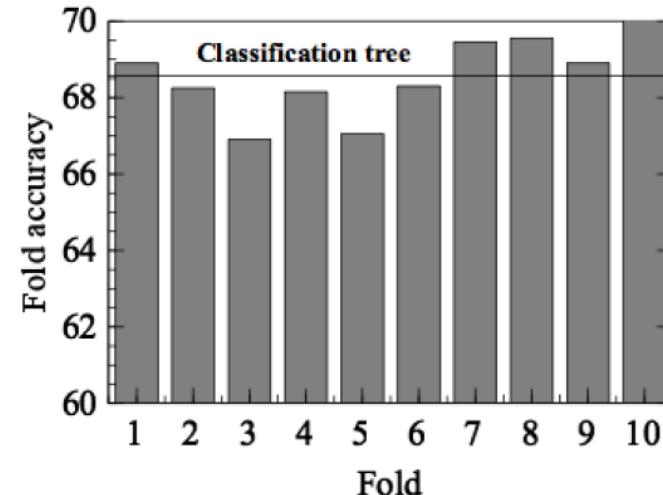
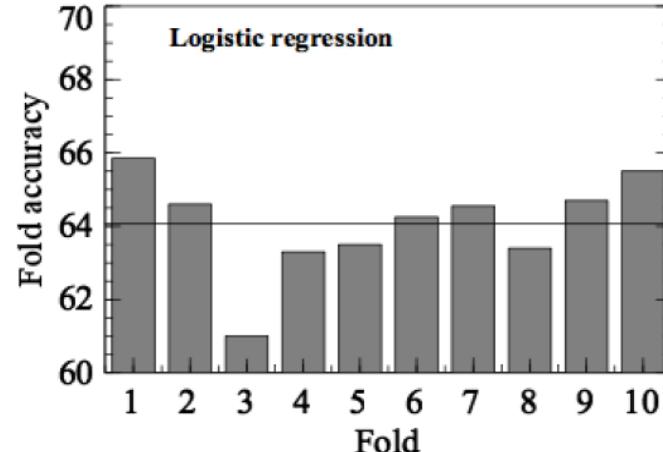
the dataset was first shuffled,  
then divided into ten partitions

Classification trees:

avg accuracy is 68.6% (std 1.1)

Logistic regression models:

avg accuracy is 64.1% (std 1.3)



# Cross-validation in Python



Python sklearn offers multiple ways for validation, but also for cross-validation.

In its easiest form, you train a model:

```
clf = svm.SVC(kernel='linear', C=1)
```

Then, you test it.

Either with a *simple scoring method*

```
scores = clf.score(iris.data,  
iris.target)
```

```
print("Accuracy: %0.2f"  
% (scores))
```

e.g. Accuracy: 0.96

Or with *cross-validation* (X times, with varying splits)

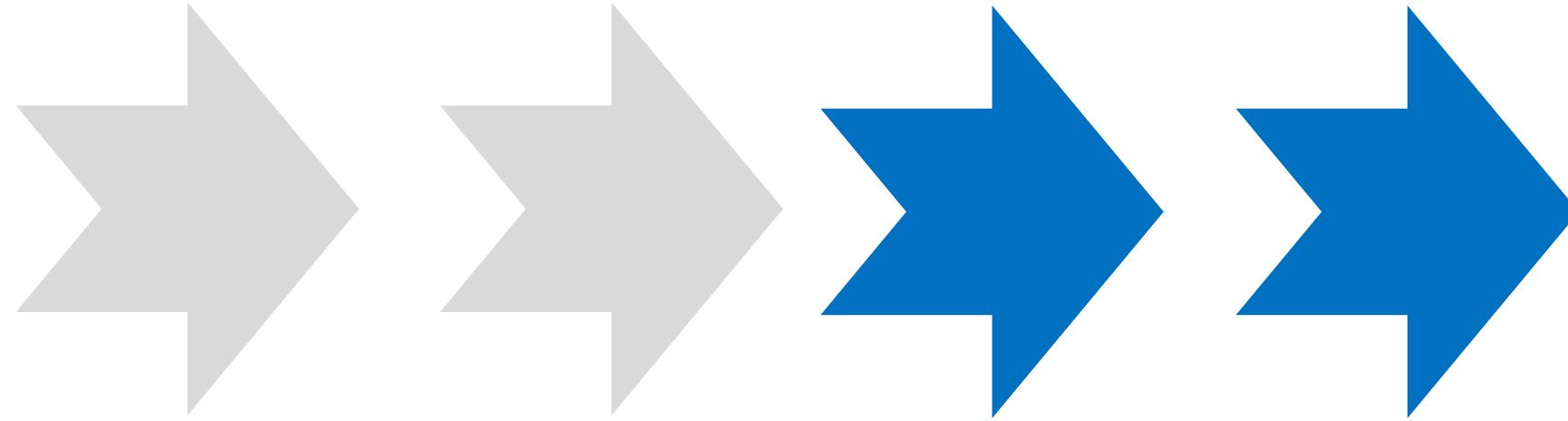
```
from sklearn.model_selection import cross_val_score  
scores = cross_val_score(clf, iris.data, iris.target, cv=5)
```

```
print("Accuracy: %0.2f (+/- %0.2f)"  
% (scores.mean(), scores.std() ))
```

e.g. Accuracy: 0.96 (+/- 0.03)

Example





(1) Generalization  
and Overfitting

(2) From  
holdout  
evaluation to  
cross-  
validation

(3) Learning  
curves

(4) Overfitting  
avoidance and  
complexity control

# Learning curves (1/2)

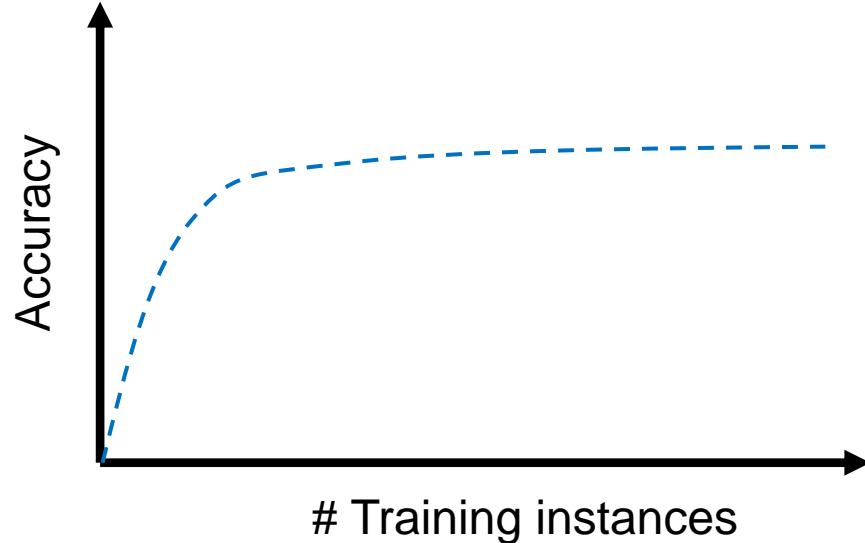


**Learning curve** (accuracy to Instances)

= a plot of the **generalization performance** (testing data) against the amount of training data

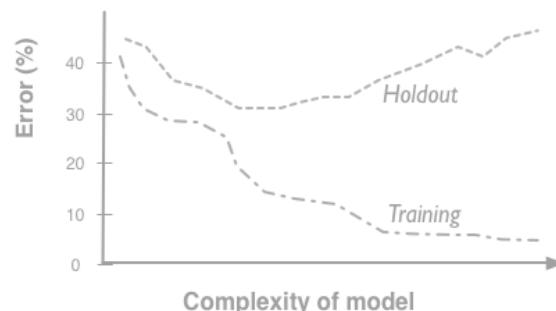
Reflects that generalization performance improves as more training data are available

Characteristic **shape**: steep initially, but then marginal advantage of more data decreases



**Fitting curve** (accuracy to features)

= shows the performance on the training and the testing data against model complexity (for a fixed amount of training data)



# Learning curves (2/2)

Example for the churn data set

Same data – different generalization performance:

For **smaller** training-set sizes, logistic regression yields better generalization accuracy

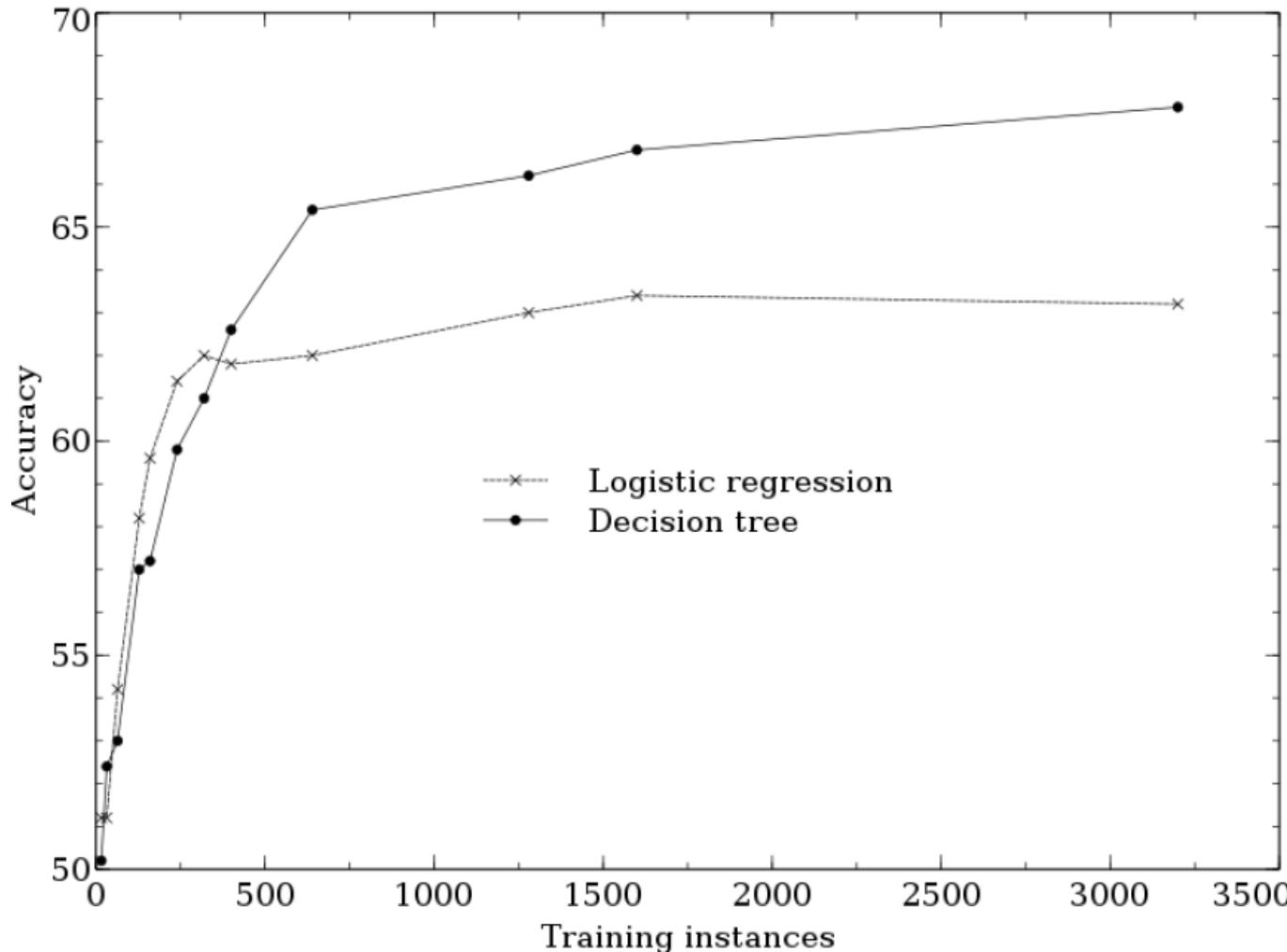
For **larger** training-set sizes, tree induction soon is more accurate

Classification trees are a **more flexible model** representation than linear logistic regression

Smaller data: tree induction will tend to overfit more

Flexibility of tree induction for larger training sets

Learning curve may give recommendations on **how much to** invest in training data



# Avoiding overfitting for tree induction

Tree induction will likely result in large, overly complex trees that overfit the data

- **Stop growing** the tree before it gets too complex



Simplest method to limit tree size: specify a **minimum number of instances** that must be present in a leaf

Automatically grow the tree branches that have a lot of data and cut short branches that have less data

What **threshold** should we use?

1. Experience
2. Conduct a hypothesis test at every leaf to determine whether the observed difference in information gain could have been due to chance (e.g., p-value below 5%)  
Accept split, if it was likely not due to chance

- **Prune back** a tree that is too large (reduce its size)



**Prune** an overly large tree = cut off leaves and branches and replace them with leaves

Estimate whether replacing a set of leaves or a branch with a leaf would reduce accuracy

If not, then prune

Continue process iteratively, until any removal or replacement would reduce accuracy

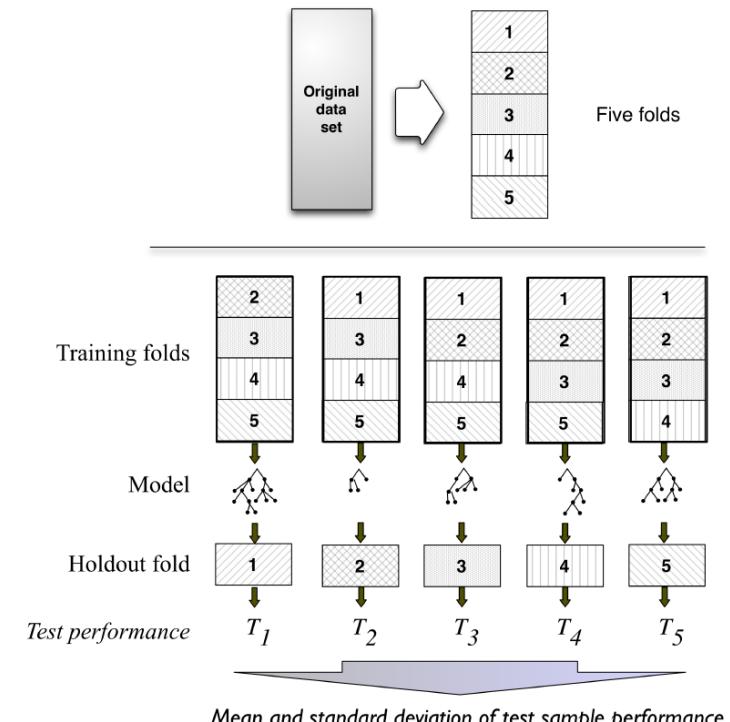
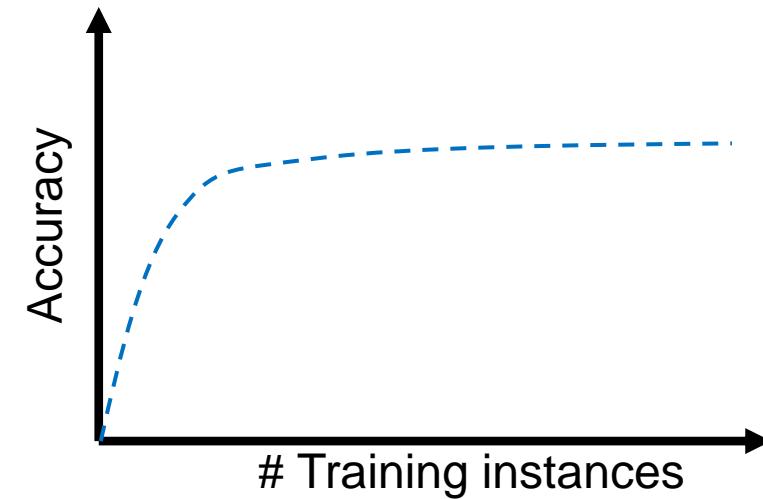
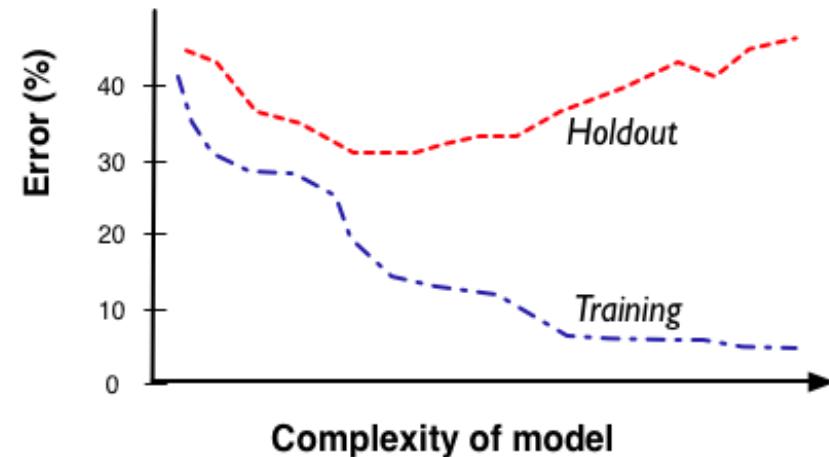
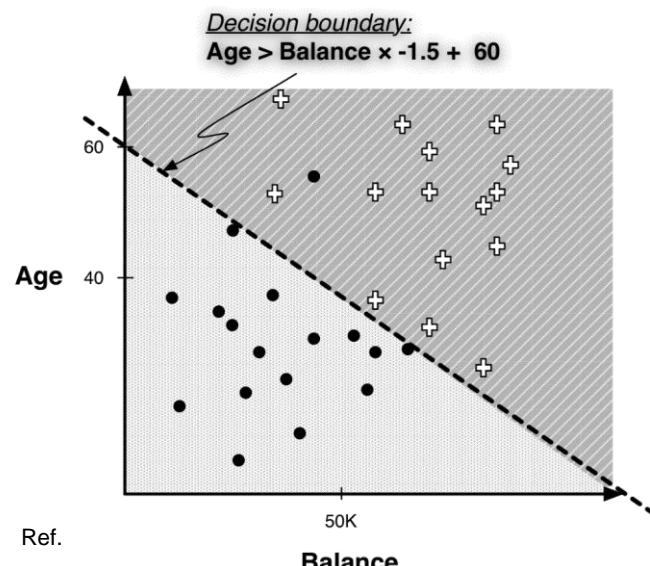
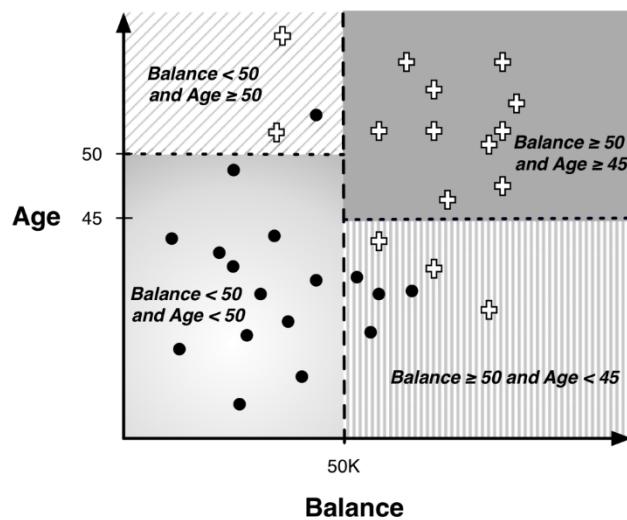
- **Build trees with all sorts of different complexities** and estimate their generalization performance

Pick the one that is estimated to be the best

Excursus: Random Forest, e.g., [kaggle tutorial](#) 

# Summary – Exercise

Explain what you see.



10 Min.

## Fragen?

- ✓ Tree induction vs. linear classifier
- ✓ Generalization and Overfitting
- ✓ From holdout evaluation to cross-validation
- ✓ Fitting graphs
- ✓ Learning curves
- ✓ Overfitting avoidance and complexity control

# Todos for this week

Further information about overfitting and avoiding overfitting

- Please read: Why is overfitting bad? - Example  
*Slides 30-31*
- Please read: A general method for avoiding overfitting  
*Kursmaterial > Readings & Übungen*  
*(short version on slides 32-33)*

# Recommended reading

## How to avoid Overfitting

Provost, F., Data Science for Business  
Fawcett, T. Chapter 5

Berthold et al. Several subchapters

# Why is overfitting bad?



Why is overfitting causing **a model to become worse?**

As a model gets more complex, it is allowed to pick up harmful „spurious“ correlations

These correlations do not represent characteristics of the population in general

They may become harmful when they produce **incorrect** generalizations in the model

- Example: a simple two-class problem

# Why is overfitting bad? - Example

| Instance | x | y | Class |
|----------|---|---|-------|
| 1        | p | r | $c_1$ |
| 2        | p | r | $c_1$ |
| 3        | p | r | $c_1$ |
| 4        | q | s | $c_1$ |
| 5        | p | s | $c_2$ |
| 6        | q | r | $c_2$ |
| 7        | q | s | $c_2$ |
| 8        | q | r | $c_2$ |

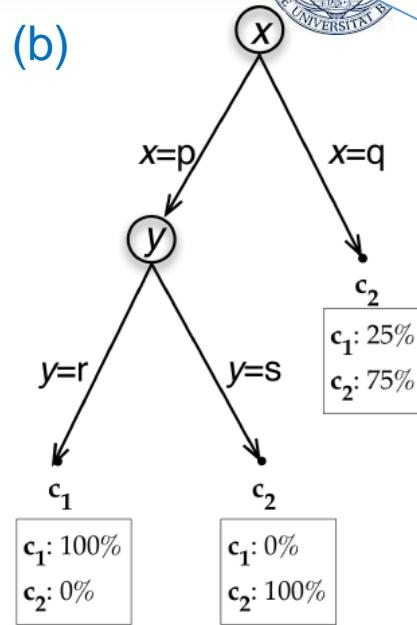
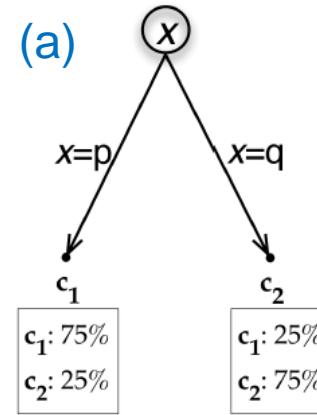
Classes  $c_1$  and  $c_2$ ,  
attributes  $x$  and  $y$   
An evenly balanced  
population of examples  
 $x$  has two values,  $p$  and  $q$ , and  
 $y$  has two values,  $r$  and  $s$

- $x = p$  occurs 75% of the time in class  $c_1$  examples  
and in 25% of  $c_2$  examples  
→  $x$  provides some prediction of that class
- Both of  $y$ 's values occur in both classes nearly equally  
→  $y$  has little predictive value
- The instances in the domain are difficult to separate, with  
only  $x$  providing some predictive leverage (75% accuracy)
- Small training set of examples

A tree learner would split on  $x$  and produce a tree (a) with error 25%

In this particular dataset,  $y$ 's values of  $r$  and  $s$  are not evenly split  
between the classes, so  $y$  seems to provide some predictness

Tree induction would achieve information gain by splitting on  $y$ 's  
values and create tree (b)



- Tree (b) performs better than (a)  
Because  $y = r$  purely by chance correlates with class  $c_1$  in this data sample  
The extra branch in (b) is not extraneous, it is **harmful!**  
The spurious  $y = s$  branch predicts  $c_2$ , which is wrong.

This phenomenon is not particular to decision trees  
It is also not because of atypical training data  
There is **no general analytic** way to avoid overfitting

# A general method for avoiding overfitting

## Nested holdout testing

How to estimate the generalization performance of models with different complexities?

Test data should be strictly independent of model building.

### Nested holdout testing

1. Split the training data set into a training subset and a testing subset
2. Build models on the training subset (**sub-training**) and pick the best model based on the testing subset (**validation**)
3. Validation set is separate from final test set



Real world data

A B A A A B A A B B A A

(Training | Testing) | Validation

(A B A A | A B A A) | B B A A

# A general method for avoiding overfitting

1. Use the **sub-training/validation split** to pick the best complexity without tainting the set
2. Build a model of this best complexity on the **entire training set**

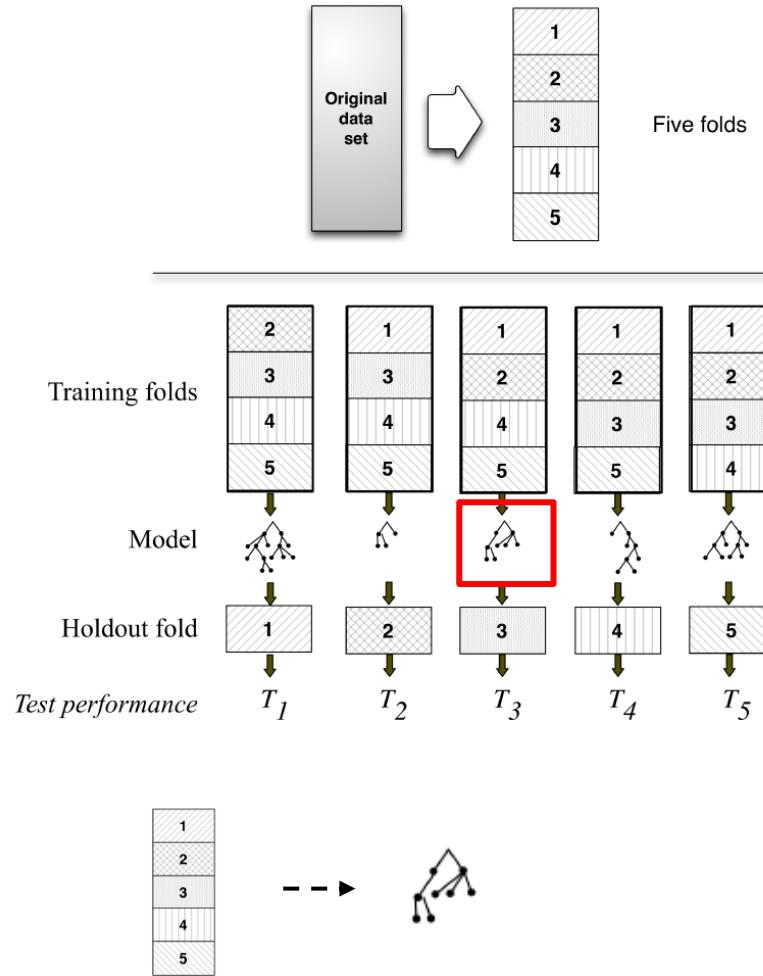
## Example for classification trees

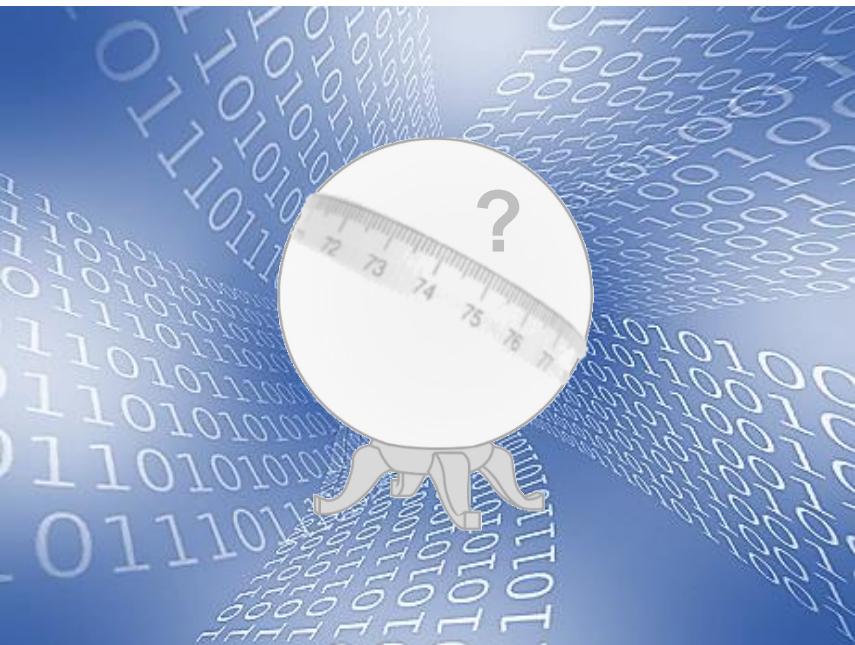
Induce trees of many complexities from sub-training set

Estimate generalization performance from validation set (e.g., the best model has a complexity of 122 nodes)

Estimate the actual generalization performance on final holdout set

For the given complexity, induce a new tree with 122 nodes from the original training set





# Business Intelligence

## 13 What is a good model?

Prof. Dr. Bastian Amberg  
(summer term 2024)

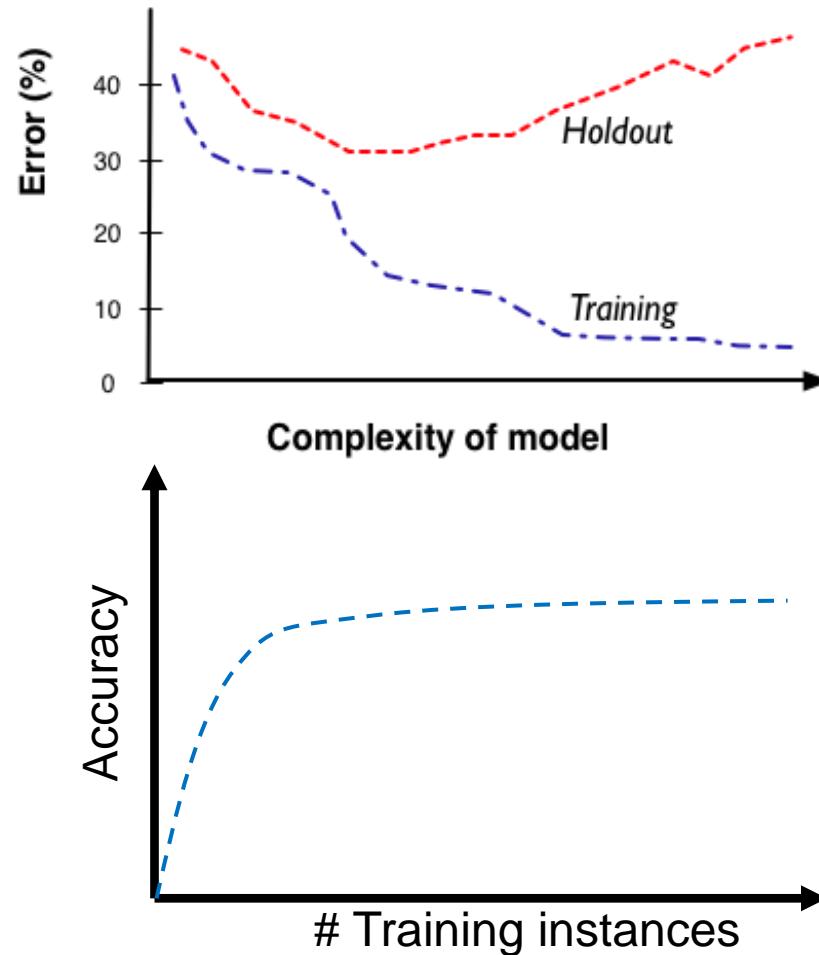
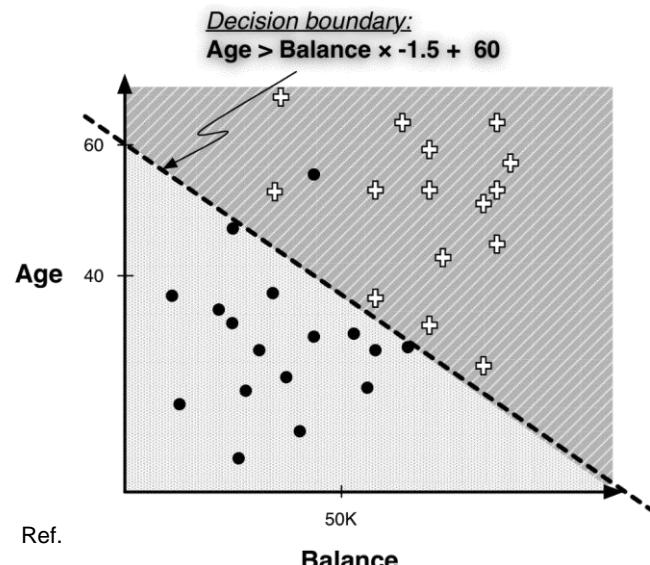
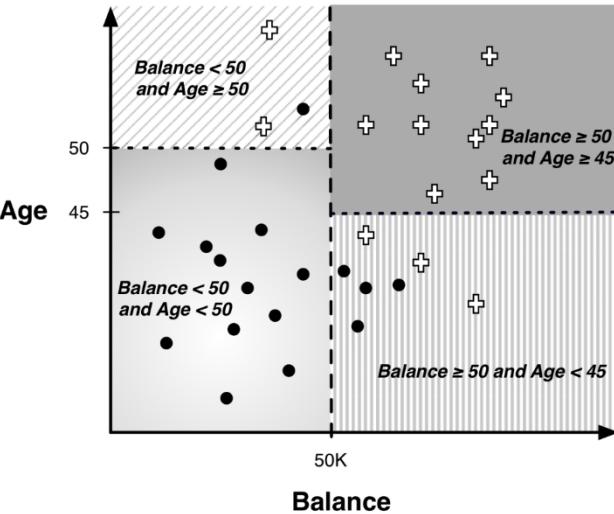
28.6.2024

# Schedule

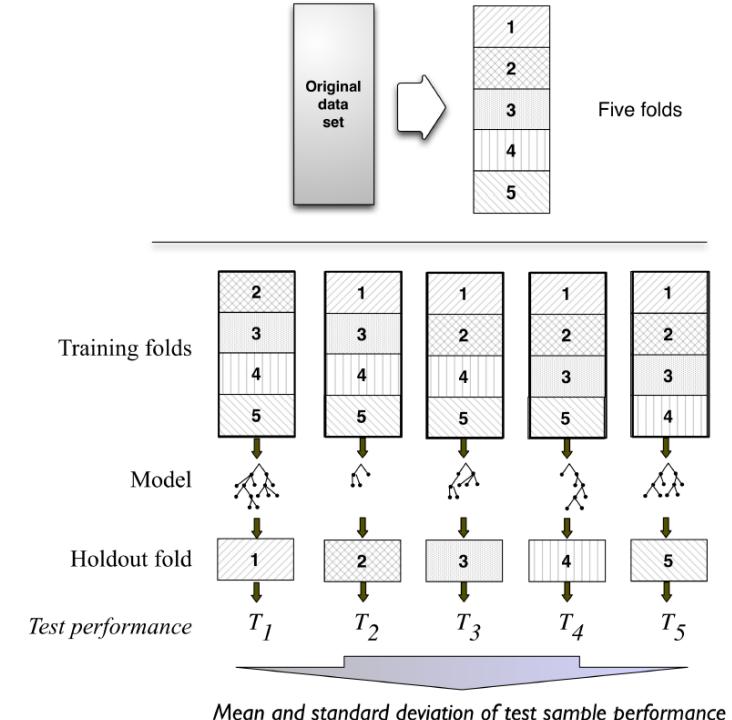
|           | Wed., 10:00-12:00 |       |   | Fr., 14:00-16:00 (Start at 14:30) |       |   | Self-study |                  |               |         |  |  |
|-----------|-------------------|-------|---|-----------------------------------|-------|---|------------|------------------|---------------|---------|--|--|
| Basics    | W1                | 17.4. | (Meta-)Introduction                                 |                                   | 19.4. |   |            |                  | Python-Basics | Chap. 1 |  |  |
|           | W2                | 24.4. | Data Warehouse – Overview                           | & OLAP                            | 26.4. | [Blockveranstaltung SE Prof. Gersch]  |            |                  |               | Chap. 2 |  |  |
|           | W3                | 1.5.  |   |                                   | 3.5.  |            |            |                  |               | Chap. 3 |  |  |
|           | W4                | 8.5.  | Data Warehouse Modeling I                           | & II                              | 10.5. | Data Mining Introduction  |            |                  |               |         |  |  |
| Main Part | W5                | 15.5. | CRISP-DM, Project understanding                     |                                   | 17.5. | Python-Basics-Online Exercise   |            | Python-Analytics | Chap. 1       |         |  |  |
|           | W6                | 22.5. | Data Understanding, Data Visualization I            |                                   | 24.5. | No lectures, but bonus tasks<br>1.) Co-Create your exam<br>2.) Earn bonus points for the exam |            |                  | Chap. 2       |         |  |  |
|           | W7                | 29.5. | Data Visualization II                               |                                   | 31.5. |   |            |                  |               |         |  |  |
|           | W8                | 5.6.  | Data Preparation                                    |                                   | 7.6.  | Predictive Modeling I (10:00 -12:00)  |            | BI-Project       | Start         |         |  |  |
|           | W9                | 12.6. | Predictive Modeling II                              |                                   | 14.6. | Python-Analytics-Online Exercise  |            |                  |               |         |  |  |
|           | W10               | 19.6. | Guest Lecture Dr. Ionescu                           |                                   | 21.6. | Fitting a Model   |            |                  |               |         |  |  |
|           | W11               | 26.6. | How to avoid overfitting                            |                                   | 28.6. | What is a good Model?   |            |                  |               |         |  |  |
| Deepening | W12               | 3.7.  | Project status update<br>Evidence and Probabilities |                                   | 5.7.  | Similarity (and Clusters)<br>From Machine to Deep Learning I                                  |            |                  |               |         |  |  |
|           | W13               | 10.7. |   |                                   | 12.7. | From Machine to Deep Learning II  |            |                  |               |         |  |  |
|           | W14               | 17.7. | Project presentation                                |                                   | 19.7. | Project presentation  |            |                  | End           |         |  |  |
| Ref.      |                   |       |   |                                   |       | Klausur 1.Termin, 31.7.'24<br>Klausur 2.Termin, 2.10.'24                                      |            | Projektbericht   |               |         |  |  |

# Last Lesson(s) – Exercise

Explain what you see.



Kahoot-Fragen  
[www.kahoot.it](http://www.kahoot.it)  
(über Smartphone oder Laptop)  
PIN folgt



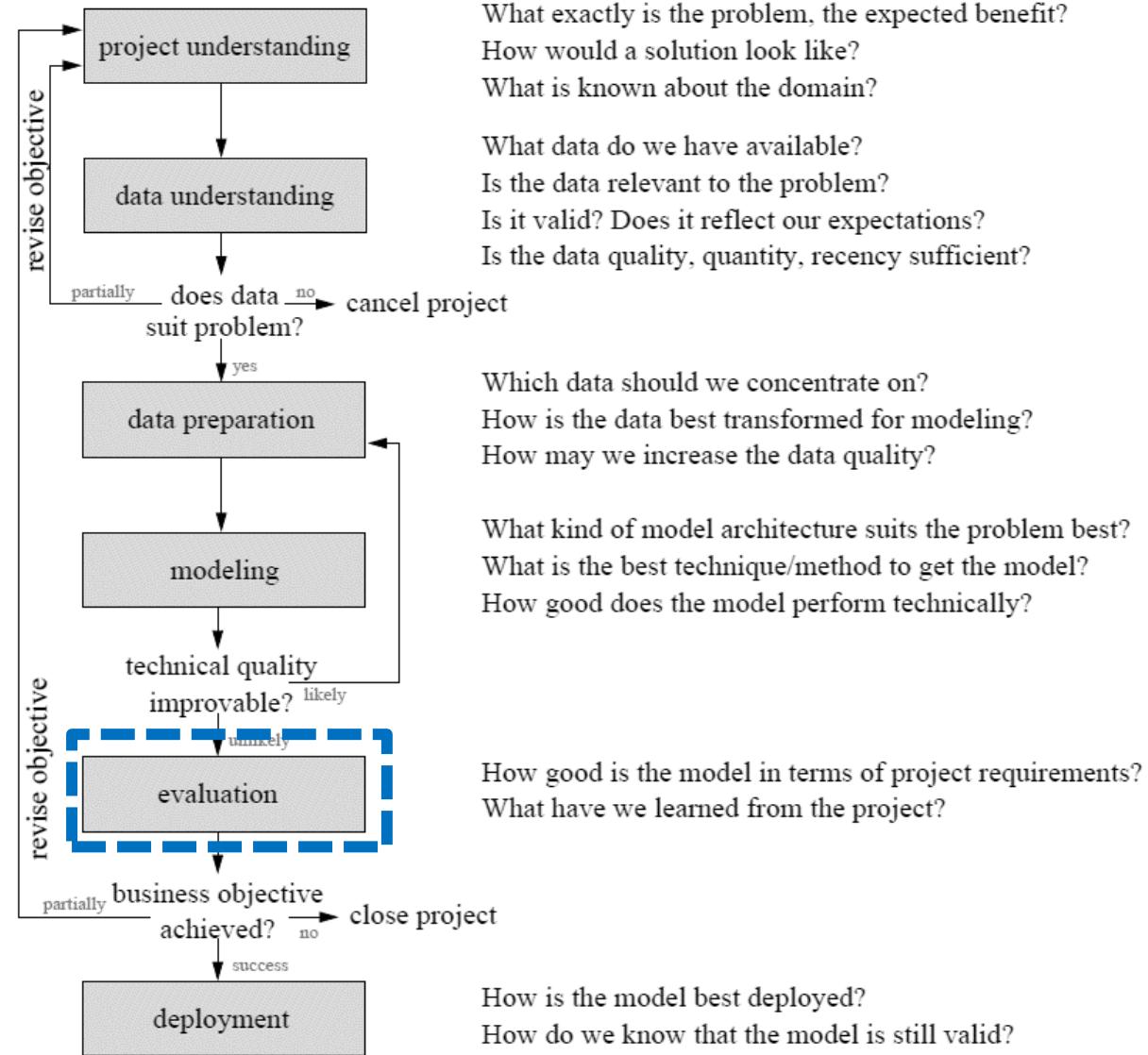
10 Min.

## Cross Industry Standard Process for Data Mining

Iteration as a rule

Process of data exploration

Implementation of the KDD Process



What exactly is the problem, the expected benefit?

How would a solution look like?

What is known about the domain?

What data do we have available?

Is the data relevant to the problem?

Is it valid? Does it reflect our expectations?

Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?

How is the data best transformed for modeling?

How may we increase the data quality?

What kind of model architecture suits the problem best?

What is the best technique/method to get the model?

How good does the model perform technically?

How good is the model in terms of project requirements?

What have we learned from the project?

How is the model best deployed?

How do we know that the model is still valid?

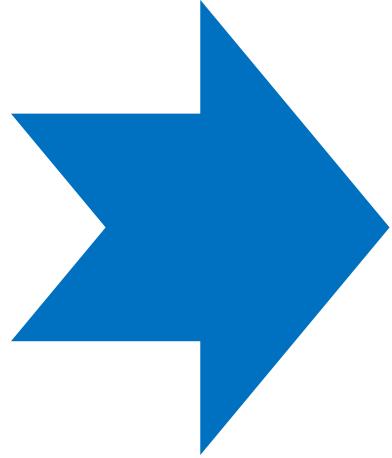
**Modeling:** Open issue from last lesson:

Avoiding Overfitting and Complexity Control in Parametric Learning/Optimization

$$\arg \max_w \text{fit}(x, w) \rightarrow \text{penalize complexity}$$

$$\arg \max_w [\text{fit}(x, w) - \lambda \cdot \text{penalty}(w)]$$

*we will come back to this topic later*



**(1) Measuring accuracy**

- Confusion matrix
- Unbalanced classes

**(2) Expected Value**

- Evaluate classifier use
- Frame classifier evaluation

**(3) Evaluation and baseline performance**

What is desired from data mining results?

How would you **measure** that your model is any good?

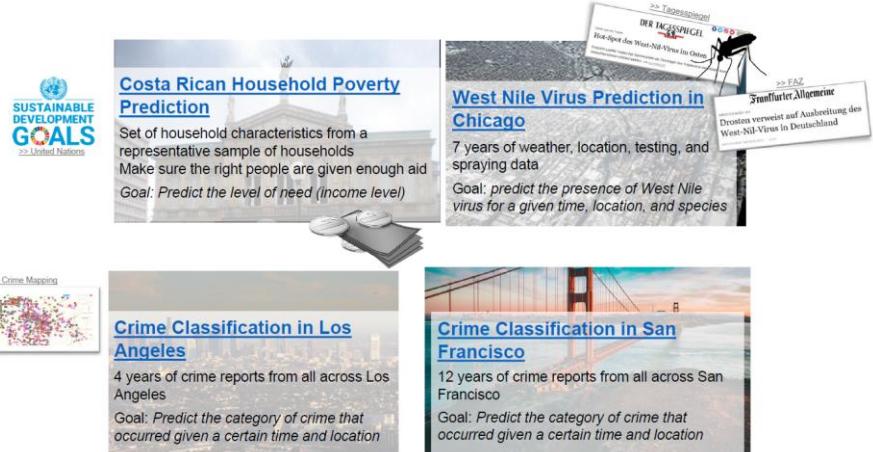
How to measure performance in a meaningful way?

Model evaluation is **application-specific**

We look at common issues and themes in evaluation

Frameworks and metrics for classification and instance scoring

Think about the specific BI project you are working on....



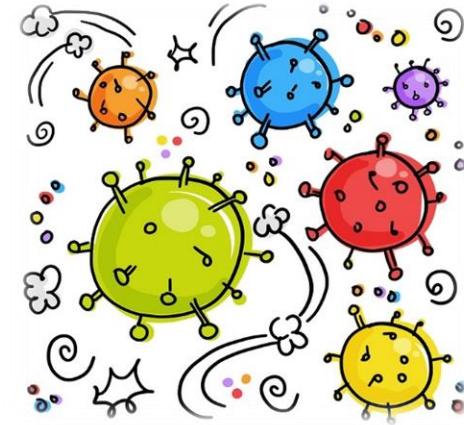
# Bad positives and harmless negatives

## Classification terminology

|                       |                        |                 |
|-----------------------|------------------------|-----------------|
| a <b>bad</b> outcome  | → a “positive” example | [alarm!]        |
| a <b>good</b> outcome | → a “negative” example | [uninteresting] |

## Further examples

|                 |   |
|-----------------|---|
| medical test:   | positive test → disease is present          |
| fraud detector: | positive test → unusual activity on account |



A classifier tries to distinguish the majority of cases (**negatives**, the uninteresting) from the small number of alarming cases (**positives**, alarming)

**number of mistakes** made on **negative** examples (false positive errors) will be relatively high / may dominate

**cost of each mistake** made on a **positive** example (false negative error) will be relatively high / will be higher

# Measuring accuracy and its problems

Up to now: measure a model's performance by some simple metric  
classifier error rate, accuracy  
Simple example: accuracy

$$accuracy = \frac{\text{Number of correct decisions made}}{\text{Total number of decisions made}}$$

Classification accuracy is popular, but usually **too simplistic** for applications of data mining to real business problems

**Decompose** and count the different types of correct and incorrect decisions made by a classifier

# The confusion matrix

A **confusion matrix** for a problem involving  $n$  classes is an  $n \times n$  matrix with the columns labeled with actual classes and the rows labeled with predicted classes.

For a binary variable it is as follows:

|           |   | p               | n               |
|-----------|---|-----------------|-----------------|
| Predicted | Y | True positives  | False positives |
|           | N | False negatives | True negatives  |

Each example in a test set has an **actual class label** and the **class predicted** by the classifier

## Interpretation:

The confusion matrix separates the decisions made by the classifier

- actual/true classes: p(ositive), n(egative)
- predicted classes: Y(es), N(o)
- The main diagonal contains the count of correct decisions

# Mini-Exercise - The confusion matrix

|           | p               | n               |
|-----------|-----------------|-----------------|
| Predicted |                 |                 |
| Y         | True positives  | False positives |
| N         | False negatives | True negatives  |

Diese Folie ist nach der Vorlesung mit Lösungen verfügbar

Kahoot-Fragen

[www.kahoot.it](http://www.kahoot.it)

(über Smartphone oder Laptop)

PIN folgt

Ref.

# The confusion matrix

## Unbalanced classes

In most real world classification problems, one class is often **rare**

Classification is used to find a relatively small number of **unusual ones** (defrauded customers, defective parts, targeting consumers who actually would respond, ...)

The class distribution is unbalanced (“skewed”)

Evaluation based on **accuracy** does not work

Example: 999:1 ratio  
always choose the most prevalent class – 99.9% accuracy!

Fraud detection: skews of 1:99  
Is a model with 80% accuracy always better than a model with 37% accuracy?

We need to have more information about the population



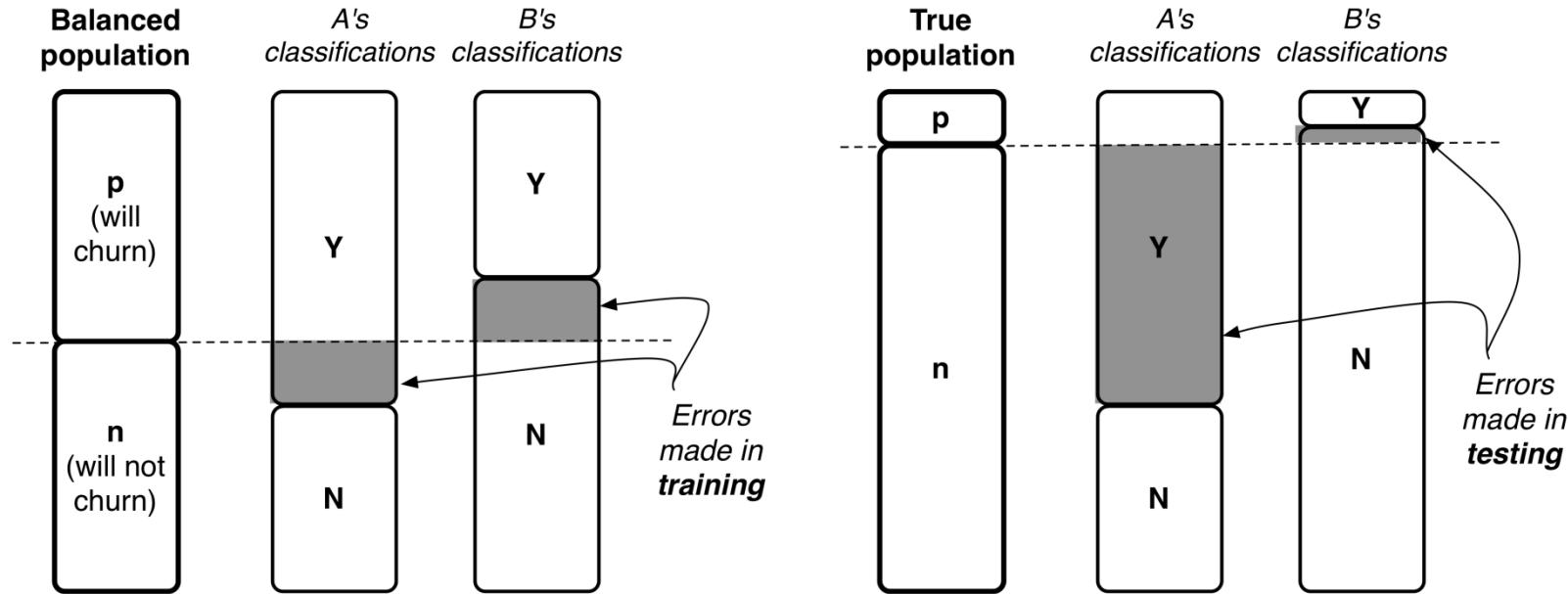
# The confusion matrix

## Unbalanced classes

Consider two models A and B for the churn example (~every tenth customer churns)

We train with a balanced population (1000 customer)

1. Both models correctly classify 80% of the balanced pop.
2. Classifier A often falsely predicts that customers will churn
3. Classifier B makes many opposite errors



Unbalanced population:

4. A's accuracy is 40%,  
B's accuracy is 97%

Note the **different performances** of the models in form of a confusion matrix:

$$CM_A = \frac{Y}{N} \begin{pmatrix} 500 & 200 \\ 0 & 300 \end{pmatrix}$$

$$CM_A = \frac{Y}{N} \begin{pmatrix} 100 & 600 \\ 0 & 300 \end{pmatrix}$$

$$CM_B = \frac{Y}{N} \begin{pmatrix} 300 & 0 \\ 200 & 500 \end{pmatrix}$$

$$CM_B = \frac{Y}{N} \begin{pmatrix} 70 & 0 \\ 30 & 900 \end{pmatrix}$$

Ref.

# Unequal costs and benefits

How much do we care about the different **errors** and correct decisions?

Classification accuracy makes no distinction between **false positive** and **false negative** errors

In real-world applications, different kinds of errors lead to different consequences.

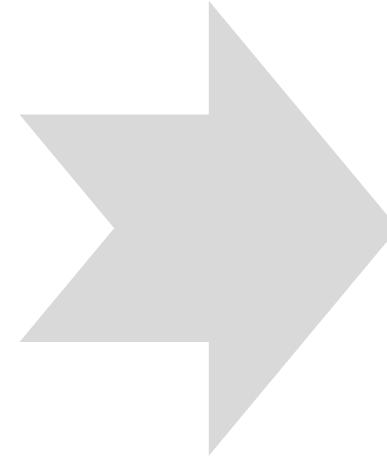
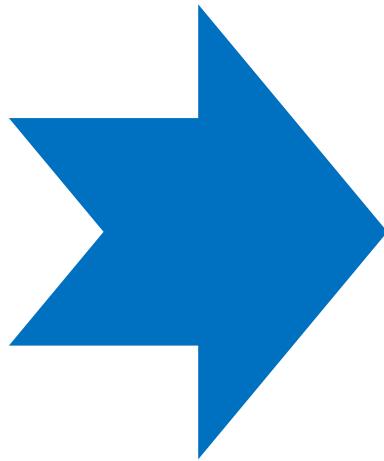
*Examples for medical diagnosis:*

*a patient has cancer (although she/he does not)*  
→ **false positive error**, expensive, but not life threatening

*a patient has cancer, but she/he is told that she/he has not*  
→ **false negative error**, more serious

Errors should be counted separately:

Estimate cost or benefit of each decision



## (1) Measuring accuracy

- Confusion matrix
- Unbalanced classes

## (2) Expected Value

- Evaluate classifier use
- Frame classifier evaluation

## (3) Evaluation and baseline performance

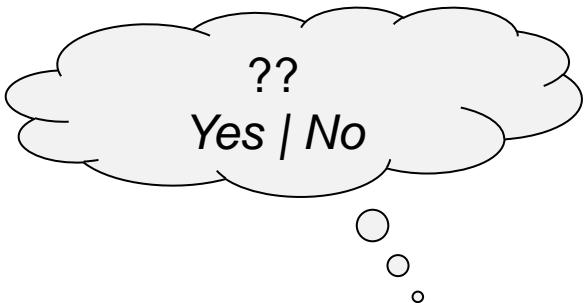
# The expected value framework

Expected value calculation includes **enumeration of the possible outcomes** of a situation

Expected value = weighted average of the values of different possible outcomes, where the weight given to each value is the probability of its occurrence

Example: different levels of profit

We focus on the maximization of expected profit



**General form** of expected value computation:

$$EV = p(o_1) \cdot v(o_1) + p(o_2) \cdot v(o_2) + \dots +$$

with  $o_i$  as possible decision outcome,  
 $p(o_i)$  as its probability, and  $v(o_i)$  as its value.

Probabilities can be **estimated** from available data

We consider two cases

- I. Evaluate classifier use
- II. Frame classifier evaluation

# (I) Expected value for use of a classifier

**Use of a classifier:** predict a class and take some action

*Example target marketing: assign each consumer to either a class „likely responder“ or „not likely responder“*

*Response is usually relatively low – so no consumer may seem like a likely responder*

Computation of the expected value

A model gives an estimated probability of response  $\hat{p}_R(x)$  for any consumer with a feature vector  $x$

**Calculate expected benefit (or costs) of targeting consumer  $x$ :**  $\hat{p}_R(x) \cdot v_R + (1 - \hat{p}_R(x)) \cdot v_{NR}$  with  $v_R$  being the value of a response and  $v_{NR}$  the value from no response

Example:

Price of product: \$200, costs of product: \$100

Targeting a consumer: \$1, profit  $v_R = \$99$ ,  $v_{NR} = -\$1$

Do we make a profit? Is the expected value (profit) of targeting greater than zero?

$$\hat{p}_R(x) \cdot \$99 + (1 - \hat{p}_R(x)) \cdot (-\$1) > 0$$

$$\leftrightarrow \hat{p}_R(x) \cdot \$99 > (1 - \hat{p}_R(x)) \cdot \$1$$

$$\leftrightarrow \hat{p}_R(x) > 0.01$$

We should target the consumer as long as the estimated probability of responding is greater than 1%

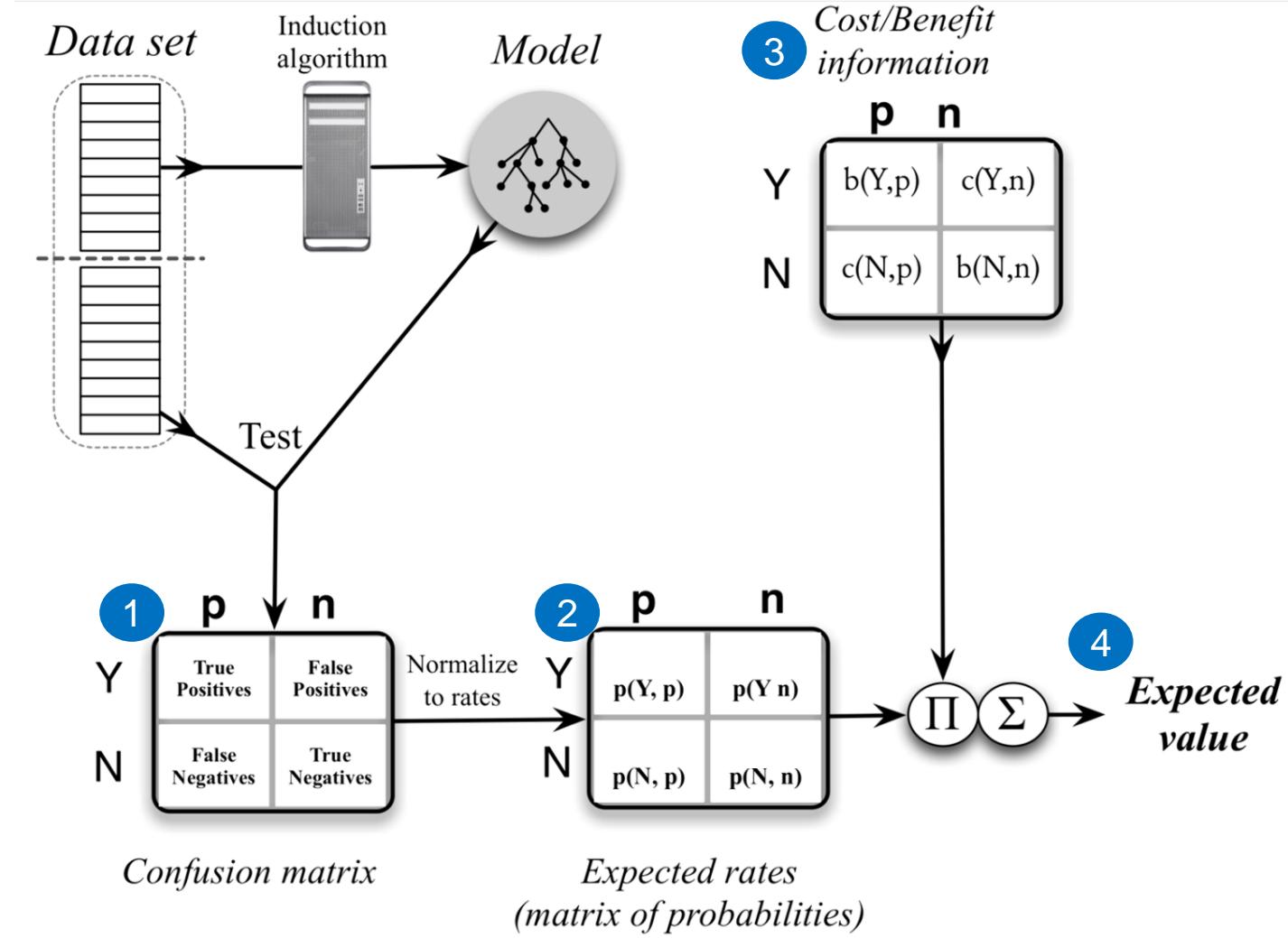
# (II) Expected value classification

Goal: **compare the quality of different models** with each other

- Does the data-driven model perform better than a hand-crafted model?
- Does a classification tree work better than a linear discriminant model?
- Do any of the models perform substantially better than a baseline model?

In short:

How well does each model perform with regards to its expected value?



# Expected rates for evaluation of a classifier

2



**Aggregate** together all the different cases:

When we target consumers, what is the probability that they (do not) respond?

What about when we do not target consumers, would they have responded?

This information is available in the **confusion matrix**

Each  $o_i$  corresponds to one of the possible combinations of the class we predict/the actual class

Where do the probabilities of errors and correct decisions actually come from?

Each cell of the confusion matrix contains a count of the number of decisions corresponding to the combination of (predicted, actual)

$count(h, a)$

Compute estimated probabilities as

$$p(h, a) = count(h, a) / Total$$

Example confusion matrix/estimates of probability

| Actual    |   |    |    |
|-----------|---|----|----|
|           |   | p  | n  |
| Predicted | Y | 56 | 7  |
|           | N | 5  | 42 |

$$T = 110, P = 61, N = 49 \text{ (Positive, Negative)}$$

$$p(Y, p) = \frac{56}{110} = 0.51, \quad p(Y, n) = \frac{7}{110} = 0.06$$

$$p(N, p) = \frac{5}{110} = 0.05, \quad p(N, n) = \frac{42}{110} = 0.38$$

# Costs and benefits

3

Compute **cost-benefit values** for each decision pair

A cost-benefit matrix specifies for each (predicted,actual) pair the cost or benefit for making such a decision

|           |   | Actual |        |
|-----------|---|--------|--------|
|           |   | p      | n      |
| Predicted | Y | b(Y,p) | c(Y,n) |
|           | N | c(N,p) | b(N,n) |

Correct classifications (true positives and negatives) correspond to  $b(Y, p)$  and  $b(N, n)$ , respectively

Incorrect classifications (false positives and negatives) correspond to  $c(Y, n)$  and  $c(N, p)$ , respectively

[often negative benefits or costs]

Costs and benefits cannot be estimated from data

How much is the true value for retaining a customer? (i.e., CLV)

Ref. Often use of average estimated costs and benefits

Targeted marketing example

- **False positive** occurs when we classify a consumer as a likely responder and therefore target her, but she does not respond  
→ cost  $c(Y, n) = -1$  //or negative benefit  $b(Y, n)$
- **False negative** is a consumer who was predicted not to be a likely responder, but would have bought if offered. No money spent, nothing gained  
→ cost  $c(N, p) = 0$  //or negative benefit  $b(N, p)$
- **True positive** is a consumer who is offered the product and buys it  
→ benefit  $b(Y, p) = 200 - 100 - 1 = 99$
- **True negative** is a consumer who was not offered a deal but who would not have bought it  
→ benefit  $b(N, n) = 0$

Sum up in cost-benefit matrix

|           |   | Actual |    |
|-----------|---|--------|----|
|           |   | p      | n  |
| Predicted | Y | 99     | -1 |
|           | N | 0      | 0  |

# Expected profit computation

4

Sufficient for comparison of various models

Compute **expected profit** by cell-wise multiplication of the matrix of costs and benefits against the matrix of probabilities:

$$EP = p(Y, p) \cdot b(Y, p) + p(N, p) \cdot c(N, p) + \\ p(N, n) \cdot b(N, n) + p(Y, n) \cdot c(Y, n)$$

Alternative calculation: factor out the probabilities of seeing each class (class priors) [Use  $p(x, y) = p(y) \cdot p(x | y)$ ]

Class priors  $p(p)$  and  $p(n)$  specify the likelihood of seeing positive versus negative instances

Factoring out allows us to separate the influence of class imbalance from the predictive power of the model

Factoring out priors yields the following

**alternative expression** for expected profit:

$$EP = p(Y|p) \cdot p(p) \cdot b(Y, p) + p(N|p) \cdot p(p) \cdot c(N, p) + \\ p(N|n) \cdot p(n) \cdot b(N, n) + p(Y|n) \cdot p(n) \cdot c(Y, n)$$

$$EP = p(p) \cdot [p(Y|p) \cdot b(Y, p) + p(N|p) \cdot c(N, p)] + \\ p(n) \cdot [p(N|n) \cdot b(N, n) + p(Y|n) \cdot c(Y, n)]$$

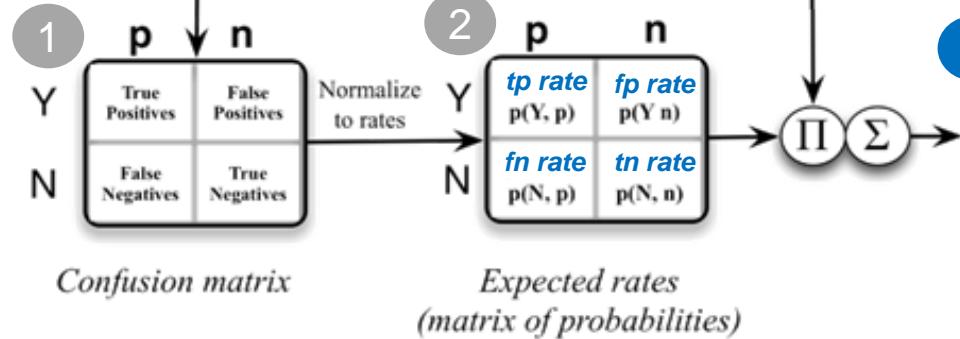
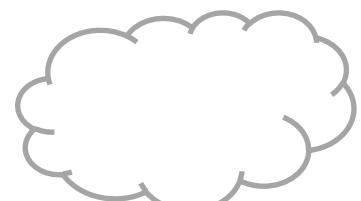
The first component corresponds to the expected profit from the **positive examples**, whereas the second corresponds to the expected profit from the **negative examples**

|   | p         | n         | p         | n         |
|---|-----------|-----------|-----------|-----------|
| Y | $p(Y, p)$ | $p(Y, n)$ | $b(Y, p)$ | $c(Y, n)$ |
| N | $p(N, p)$ | $p(N, n)$ | $c(N, p)$ | $b(N, n)$ |

We call:  
 $p(Y|p)$  : true positive rate  
 $p(N|p)$  : false negative rate  
 $p(Y|n)$  : false positive rate  
 $p(N|n)$  : true negative rate

# Exercise – Expected value computation

Example alternative expression



This expected value means that ...

|           |   | Actual |    | 1 |
|-----------|---|--------|----|---|
|           |   | p      | n  |   |
| Predicted | Y | 56     | 7  | 2 |
|           | N | 5      | 42 |   |

1  
2

$$P = 61, \\ p(p) = 0.55, \\ tp\ rate = 56/61 = 0.92, \\ fn\ rate = 5/61 = 0.08$$

3

| Actual    |   | 3  |    |   |
|-----------|---|----|----|---|
|           |   | p  | n  |   |
| Predicted | Y | 99 | -1 | 3 |
|           | N | 0  | 0  |   |

$N = 49, \\ p(n) = 0.45, \\ fp\ rate = 7/49 = 0.14, \\ tn\ rate = 42/49 = 0.86$

$$EP = p(p) \cdot [p(Y|p) \cdot b(Y,p) + p(N|p) \cdot c(N,p)] + p(n) \cdot [p(N|n) \cdot b(N,n) + p(Y|n) \cdot c(Y,n)]$$

Todo for Wednesday

5 Min.

# Other evaluation metrics

Based on the entries of the confusion matrix, we can describe various evaluation metrics

Accuracy (count of correct decisions):

$$\frac{TP+TN}{P+N}$$

|           |   | p               | n               |
|-----------|---|-----------------|-----------------|
| Predicted | Y | True positives  | False positives |
|           | N | False negatives | True negatives  |

True positive rate / Recall / Specificity :

$$\frac{TP}{TP+FN}$$

|           |   | p               | n               |
|-----------|---|-----------------|-----------------|
| Predicted | Y | True positives  | False positives |
|           | N | False negatives | True negatives  |

False negative rate:

$$\frac{FN}{TP+FN}$$

|           |   | p               | n               |
|-----------|---|-----------------|-----------------|
| Predicted | Y | True positives  | False positives |
|           | N | False negatives | True negatives  |

Sensitivity:

$$\frac{TN}{TN+FP}$$

|           |   | p               | n               |
|-----------|---|-----------------|-----------------|
| Predicted | Y | True positives  | False positives |
|           | N | False negatives | True negatives  |

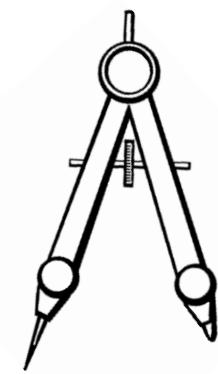
Precision (accuracy over the cases predicted to be positive):

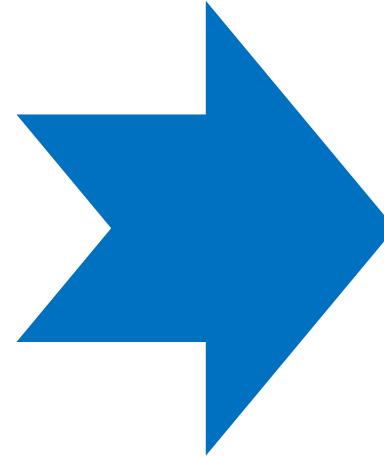
$$\frac{TP}{TP+FP}$$

|           |   | p               | n               |
|-----------|---|-----------------|-----------------|
| Predicted | Y | True positives  | False positives |
|           | N | False negatives | True negatives  |

F-measure (harmonic mean):

$$2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$





## (1) Measuring accuracy

- Confusion matrix
- Unbalanced classes

## (2) Expected Value

- Evaluate classifier use
- Frame classifier evaluation

## (3) Evaluation and baseline performance

# Baseline performance (1/3)



Consider what would be a **reasonable baseline** against which to compare model performance

Demonstrate stakeholder that data mining has added value (or not)

What is the appropriate baseline for comparison?

Depends on the actual application

*There are two basic tests that any weather forecast must pass to demonstrate its merit:*

*(1) It must do **better than** what meteorologists call **persistence**: the assumption that the weather will be the same tomorrow (and the next day) as it was today.*

*(2) It must also **beat climatology**, the **long-term historical average** of conditions on a particular date in a particular area (not only dependent to time/seasonal effects).*



# Baseline performance (2/3)



Baseline performance for classification

Compare to a completely random model (very easy)

Implement a simple (but not simplistic) alternative model

**Majority classifier** = a naive classifier that always chooses the majority class of the training data set

May be challenging to outperform: classification accuracy of 94%, but only 6% of the instances are positive

→ majority classifier also would have an accuracy of 94%!

Pitfall: don't be surprised that many models simply predict everything to be of the **majority class**

Maximizing simple prediction accuracy is usually not an appropriate goal

Hint:



DummyClassifier is a classifier that makes predictions using simple rules. This classifier is useful as a simple baseline to compare with other (real) classifiers.

Strategies, e.g.,  
“most\_frequent”:

always predicts the most frequent label in the training set.

“uniform”:

generates predictions uniformly at random.

(<https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html>)



# Baseline performance (3/3)

Further alternative:

how well does a simple “conditional” model perform?

Conditional → prediction different based on the value of the features

Just **use the most informative variable** for prediction

Decision tree: build a tree with only one internal node (decision stump) → tree induction selects the single most informative feature to make a decision

Compare quality of models based on data sources

Quantify the value of each source

Implement models that are based on **domain knowledge**

## Fragen?

- ✓ Measuring accuracy
  - ✓ Confusion matrix
  - ✓ Unbalanced classes
- ✓ A key analytical framework: Expected value
  - ✓ Evaluate classifier use
  - ✓ Frame classifier evaluation
- ✓ Evaluation and baseline performance

# Todos for next Week

- Please remember:

*Gemäß Vorlesungsplanung ist für nächsten Mittwoch Project status update vorgesehen.*

*Was heißt das nun eigentlich?*

- Bitte jede Projektgruppe **bis Mittwoch 3.7.** einen kurzen „Zwischenbericht“ liefern
- Formlos per E-Mail an [bastian.amberg@fu-berlin.de](mailto:bastian.amberg@fu-berlin.de), Betreff „BI Projektgruppe – Zwischenstand“
  - Aktueller Stand (z.B. durchgeführte Bearbeitungsschritte nach CRISP-DM)
  - Offene Punkte bzw. grobes weiteres Vorgehen
  - **kurz (d.h. ca. 4-5 Sätze)**
  - Optionaler weiterer Punkt: Besteht Sprechstundenbedarf?  
Sprechstunde dann in der Woche vom 8.7. bis 12.7.

- Please do the “Exercise – Expected value computation”

[Slide 23](#)

# Recommended reading

## What is a good model:

Provost, F., Data Science for Business

Fawcett, T. Chapter 7

Berthold et al. Guide to Intelligent Data Analysis, Chapter 5

## Further reading (beyond this class):

Provost, F., Data Science for Business

Fawcett, T. Chapter 8, Visualizing Model Performance (discusses graphical views of model behaviour)



# Business Intelligence – Interim summary

We have three goals. After this course:

- You know how to solve business problems by **data-analytic thinking**
- You know **tools** and ways of how to practically **implement** solution methods
- You have an overview about principles of **how to model and solve** upcoming **business problems**.

Main focus:

**Data Warehousing / Data Engineering** ► How to **store and access** huge amounts of data?

- DW Overview (incl. OLAP) L02
- DW Modelling L03, L04

**Data Mining / Data Science** ► How to **derive knowledge and profitable business action** out of (large) databases?

- DM Overview L04, L05
- CRISP-DM

- project understanding L05
- data understanding L06, L07
- data preparation L08

- modeling L09, L10, L11, L12, ...
- evaluation L13 ...

Exkurs: Weiterführend siehe Schulz et al. 2020 DASC-PM

