Introduction to Data Science

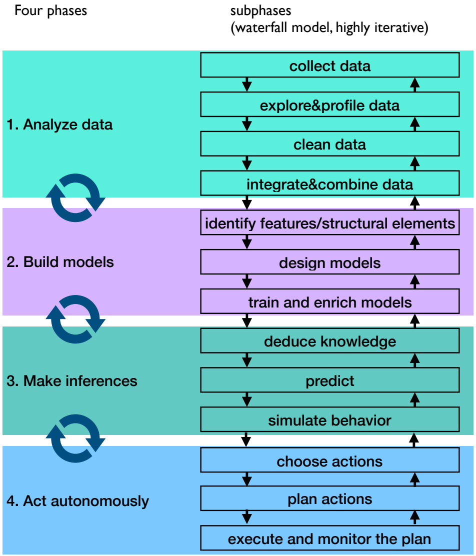
Elements of Data Science and Artificial Intelligence

Prof. Dr. Jens Dittrich

bigdata.uni-saarland.de

October 21, 2019

DSAI Process Model



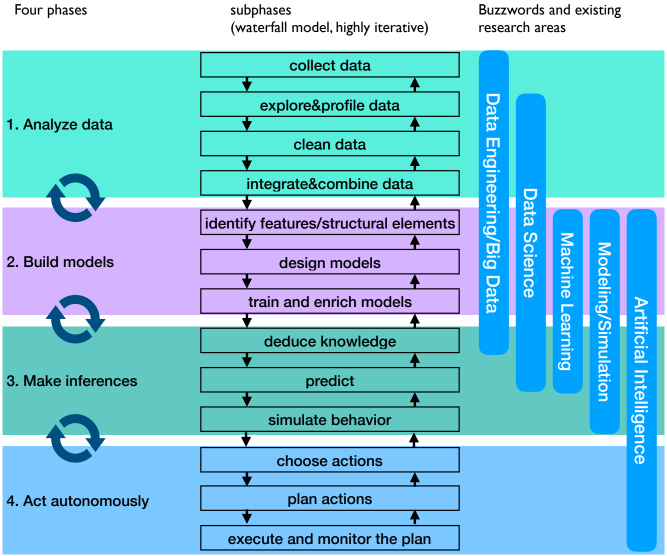
- the DSAI process model describes how we approach any problem in DSAI
- highly iterative, much more iterative than the waterfall model in general computer science

general trade-off

The earlier we identify problems the lower the costs.

Example: Assume you want to model the depth of an earthquake. The data records many quakes happening at 50 meters depth. You proceed and eventually build your model. Later on you realise that '50 meters' is used by seismologists as a default value, i.e. it actually means 'I do not know'. **Impact:** You can throw away your model...

Mapping to Buzzwords/Traditional Research Areas



Four phases

subphases
(waterfall model, highly iterative)

Buzzwords and existing
research areas

1. Analyze data
- collect data
- explore&profile data
- clean data
- integrate&combine data

2. Build models
- identify features/structural elements
- design models
- train and enrich models

3. Make inferences
- deduce knowledge
- predict
- simulate behavior

4. Act autonomously
- choose actions
- plan actions
- execute and monitor the plan

Data Engineering/Big Data

Data Science

Machine Learning

Modeling/Simulation

Artificial Intelligence

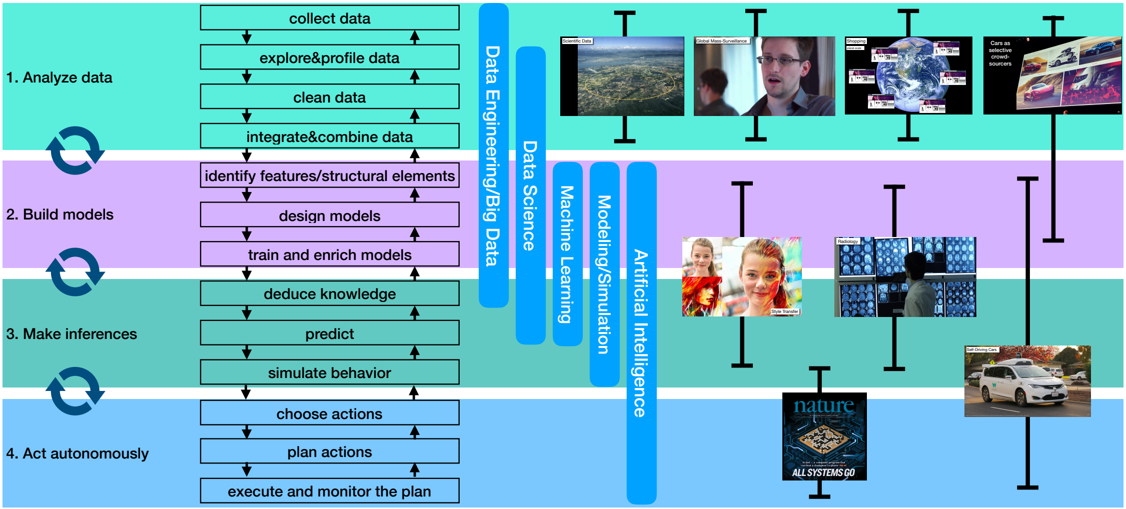
- the buzzwords and traditional research areas can loosely be mapped to certain phases in this model
- this is basically saying: each research area (data engineering, data science, machine learning, etc.) and in particular the methods used in those field have a focus on certain parts of the process model
- again: do not misread this as some sort of mathematical definition
- ML $\approx$ Data Science $\cap$ AI

Example Applications and their Focus in the DSAI Process Model

Four phases

subphases
(waterfall model, highly iterative)

Buzzwords and existing
research areas



1. Analyze data
   - collect data
   - explore&profile data
   - clean data
   - integrate&combine data
2. Build models
   - identify features/structural elements
   - design models
   - train and enrich models
3. Make inferences
   - deduce knowledge
   - predict
   - simulate behavior
4. Act autonomously
   - choose actions
   - plan actions
   - execute and monitor the plan

Data Engineering/Big Data

Data Science

Machine Learning

Modeling/Simulation

Artificial Intelligence

Example Applications and their Focus in the DSAI Process Model

Note

- again: this does not mean that **in general** that any of these applications (scientific data or Amazon etc.) is mainly about the first phase
- however, for certain applications, what you do in this first phase is important (e.g. scalability, data cleaning) and something that must be solved before being able to proceed to any of the other phases

Example: for organisations like CERN, NSA, Amazon, Tesla, etc. it is not just one application, but possibly hundreds or even thousands, yet the success of those applications depends strongly on getting the first phase of the DSAI process model right

Why Data Science now?

- high synergy effects through combination of techniques from various — historically grown — sub-areas
- strongly growing added value in research and industry
- sensational progress in the area of DSAI
- much more data is collected
- much more data sources available
- rapid advances in hardware (GPUs, TPUs)
- better usability of data science tools
- from closed to open data storage and analysis

Research Field Arithmetics

Research Field Arithmetics

- The phase model shows that ML $\approx$ Data Science $\cap$ AI.
- The first phase can be seen as “1. Analyse data” $\approx$ Data Science $\setminus$ ML.
- We will have a separate introductions to ML in this lecture later on.
- Therefore let’s focus on the first phase in the following.

The Data Analysis Phase: Important (Sub-)Research Areas

Databases:

Key questions:

- How to store and query data?
- How to make query processing efficient and scalable?
- How to recover after a failure?
- How to make this happen for just any kind of data?

Killer contributions: relational model, relational algebra, structured query language (SQL), and all kinds of algorithms and system that make the former efficient and robust

Famous products: Oracle, PostgreSQL, MySQL, SQLite, MS SQL Server, SAP Hana, Tableau, Spark, ...

Biggest Failures: XQuery (XML query processing), NoSQL (mostly reinvents very old relational technology), native, non-relational storage (LOL!)

History: huge, very active research field since the early 60ies, ACM SIGMOD, VLDB

The Data Analysis Phase: Important (Sub-)Research Areas

Data Mining[1]:

Key questions:

- How to gain knowledge from my data ?
- How to learn about to the data generating process?
- How to understand something about the data if I do not really know what to look for, or what question to ask?
- How to extract easily interpretable models and results from data?

Killer contributions: Pattern and Motif Discovery, Clustering, Outlier and Anomaly Detection, Privacy-Preserving and Algorithmic Fairness, Graph Mining and Social Network Analysis, Recommender Systems

Famous products: RapidMiner, KNIME, SQL Server Data Mining: Microsoft Analytics

[1] Thanks to Jilles Vreeken for input to this slide!

The Data Analysis Phase: Important (Sub-)Research Areas

Data Mining (continued):

Biggest Failures:

- lack of monetization of results
  ('interpretable ML', really? That's what we have been doing for two decades!)
- lack of salesmanship
  (privacy, fairness, graphs, all started in DM, but got re-invented a few years later in ML)
- hype-sensitivity
  (graphs, deep learning)

History: big, very active research field since the 90ies, ACM SIGKDD, ...

Note: famous data mining techniques like clustering sometimes referred to as "unsupervised machine learning"

Clustering Example

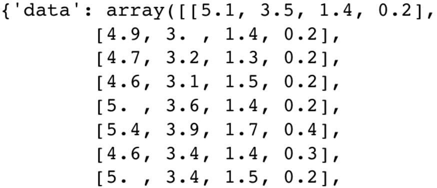
Iris virginica:

Iris versicolor:

Iris setosa:

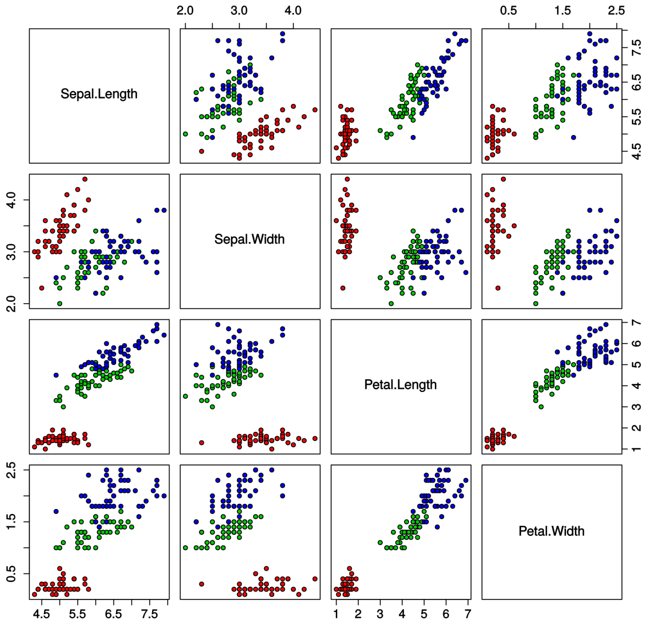
- one of the most famous toy-datasets in data science: iris (Schwertlilien)
- four attributes (so-called *features*) are measured for each tuple in the dataset:

1. the length of the sepal (Kelchblatt)
2. the width of the sepal
3. the length of the petal (Blütenblatt)
4. the width of the petal



```
{'data': array([[5.1, 3.5, 1.4, 0.2],
        [4.9, 3. , 1.4, 0.2],
        [4.7, 3.2, 1.3, 0.2],
        [4.6, 3.1, 1.5, 0.2],
        [5. , 3.6, 1.4, 0.2],
        [5.4, 3.9, 1.7, 0.4],
        [4.6, 3.4, 1.4, 0.3],
        [5. , 3.4, 1.5, 0.2],
```

Iris Data (red=setosa,green=versicolor,blue=virginica)



The Data Analysis Phase: Important (Sub-)Research Areas

Information Retrieval[2]:

Key questions:

- How can I find stuff in unstructured data in particular text or webpages?
- How can I leverage massive data from user queries and clicks?

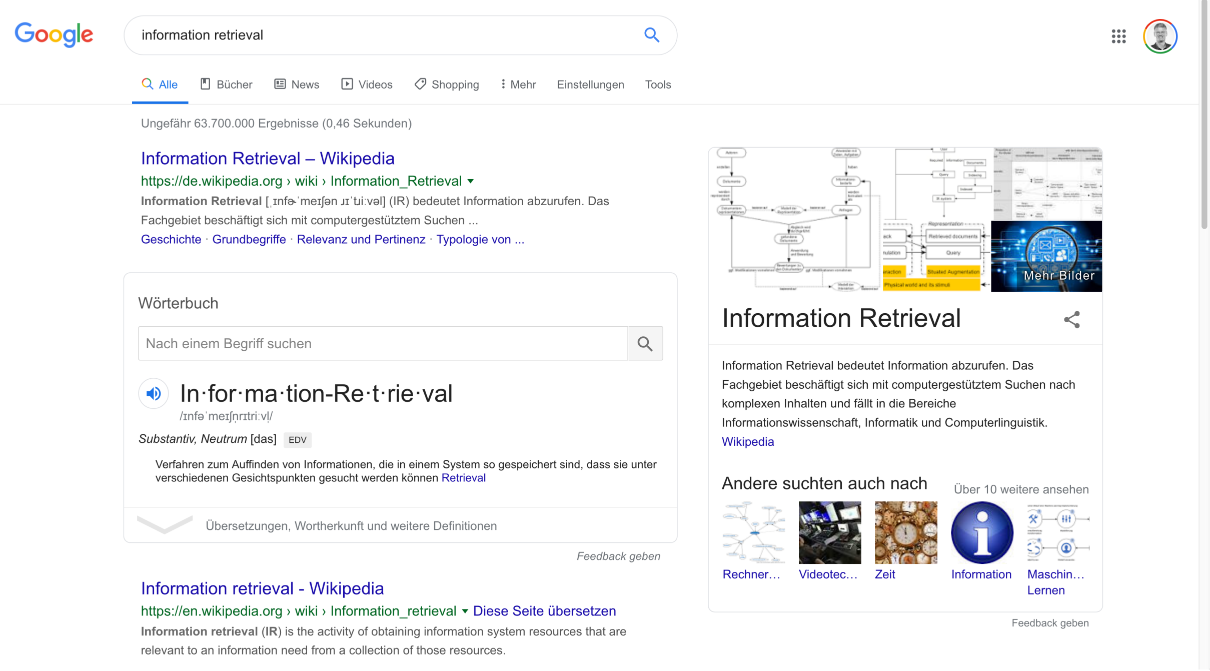
Killer contributions: search engines, inverted files

Famous products: Google, Bing, Baidu

Biggest Risk: Collecting massive data about people

History: big, very active research field since the 70ies, ACM SIGIR, ...

[2] Thanks to Gerhard Weikum for input to this slide!



Ungefähr 63.700.000 Ergebnisse (0,46 Sekunden)

Information Retrieval – Wikipedia
https://de.wikipedia.org › wiki › Information_Retrieval ▾
Information Retrieval [ˌɪnfɚˈmeɪʃən ɹɪˈtiːvəl] (IR) bedeutet Information abzurufen. Das Fachgebiet beschäftigt sich mit computergestütztem Suchen ...
Geschichte · Grundbegriffe · Relevanz und Pertinenz · Typologie von ...

Wörterbuch

Nach einem Begriff suchen

In·for·ma·tion-Re·t·rie·val
/ɪnfəˈmeɪʃn̩rɪtriːvl̩/

Substantiv, Neutrum [das] EDV

Verfahren zum Auffinden von Informationen, die in einem System so gespeichert sind, dass sie unter verschiedenen Gesichtspunkten gesucht werden können Retrieval

Übersetzungen, Wortherkunft und weitere Definitionen

Feedback geben

Information retrieval - Wikipedia
https://en.wikipedia.org › wiki › Information_retrieval ▾ Diese Seite übersetzen
Information retrieval (**IR**) is the activity of obtaining information system resources that are relevant to an information need from a collection of those resources.

Information Retrieval

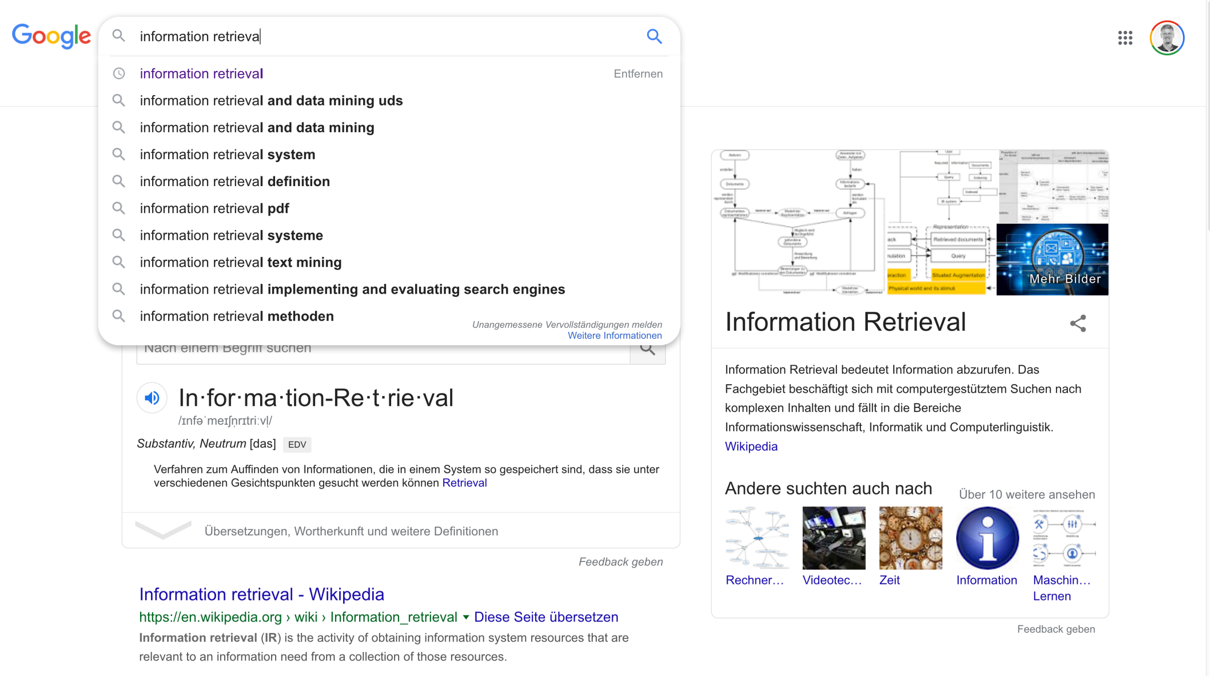
Information Retrieval bedeutet Information abzurufen. Das Fachgebiet beschäftigt sich mit computergestütztem Suchen nach komplexen Inhalten und fällt in die Bereiche Informationswissenschaft, Informatik und Computerlinguistik.
Wikipedia

Andere suchten auch nach

Über 10 weitere ansehen

Rechner…
Videotec…
Zeit
Information
Maschin… Lernen

Feedback geben



information retrieva

information retrieval — Entfernen

information retrieval **and data mining uds**

information retrieval **and data mining**

information retrieval **system**

information retrieval **definition**

information retrieval **pdf**

information retrieval **systeme**

information retrieval **text mining**

information retrieval **implementing and evaluating search engines**

information retrieval **methoden**

Unangemessene Vervollständigungen melden
Weitere Informationen

Nach einem Begriff suchen

In·for·ma·tion-Re·t·rie·val

/ɪnfəˈmeɪʃn̩rɪtriːvl/

Substantiv, Neutrum [das] EDV

Verfahren zum Auffinden von Informationen, die in einem System so gespeichert sind, dass sie unter verschiedenen Gesichtspunkten gesucht werden können Retrieval

Übersetzungen, Wortherkunft und weitere Definitionen

Feedback geben

Information retrieval - Wikipedia

https://en.wikipedia.org › wiki › Information_retrieval ▾ Diese Seite übersetzen

Information retrieval (**IR**) is the activity of obtaining information system resources that are relevant to an information need from a collection of those resources.

Information Retrieval

Information Retrieval bedeutet Information abzurufen. Das Fachgebiet beschäftigt sich mit computergestütztem Suchen nach komplexen Inhalten und fällt in die Bereiche Informationswissenschaft, Informatik und Computerlinguistik.
Wikipedia

Andere suchten auch nach — Über 10 weitere ansehen

Feedback geben

The Data Analysis Phase: Important (Sub-)Research Areas

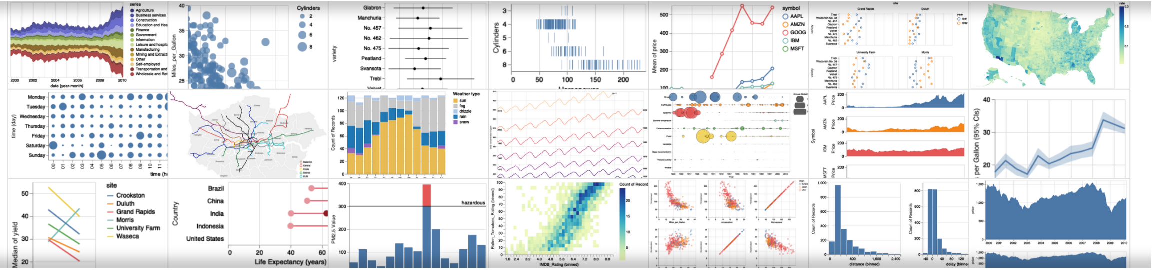
Data Visualization:

Key questions: How can I visualise data to help the user gain insight and/or convey a message?

Killer contributions: zillions

Famous products: Tableau, gnuplot, matplotlib, vega-lite, altair, D3js, etc...

Biggest Failures: ?

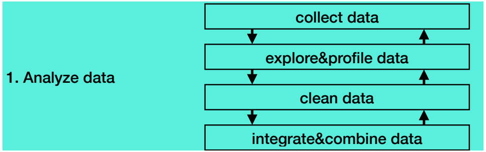


[https://vega.github.io/vega-lite/]

Challenges of the Data Analysis Phase

Let's look at the different subphases of the data analysis phase in more detail.

In an abstract way:



1. Analyze data

collect data

explore&profile data

clean data

integrate&combine data

AND

In the light of a concrete application:

Task; develop a planet-scale bookstore

Collect data: the human in the loop aspect

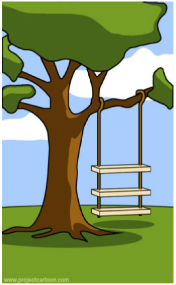
:

- talk to the “user”: i.e. project partner, customer
- understand the problem of the user
- What does the user really want?
- Do I understand what the user understands about DSAI?
- vice versa: Does the user understand what I do and do not understand about the user’s problem?
- What data could help solve the problem?
- Who are the right persons to talk to?

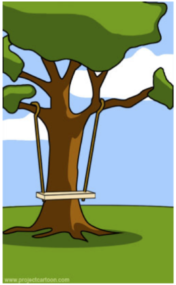
This step is unfortunately often hopelessly underrated.

:

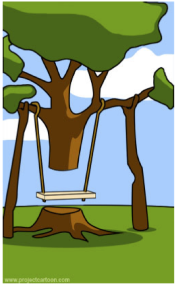
- why just books?
- physical vs digital books?
- what do you mean by “global”?
- Do you mean different languages or scalability?
- What would be a realistic first system for you? One country? One language?
- How should the system interact with the product delivery companies?
- What data is available about the items you want to sell?
- Is Peter the responsible data person at your company?



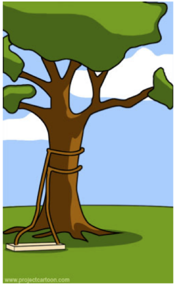
Wie der Kunde es erklärt hat



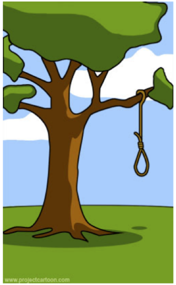
Wie der Projektleiter es verstanden hat



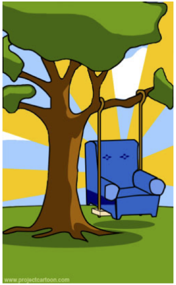
Wie der Analyst es auffasst



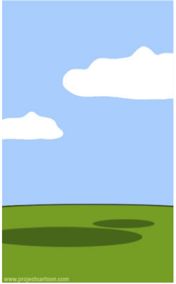
Wie der Programmierer es geschrieben hat



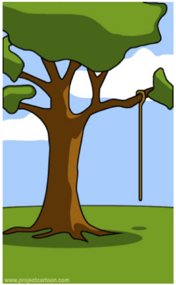
Was die Beta-Tester erhalten



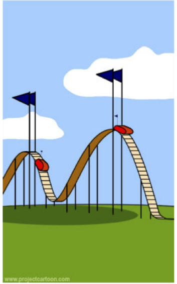
Wie der Wirtschaftsberater es verkauft



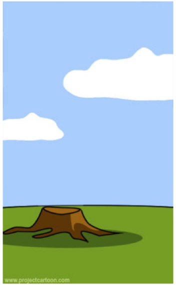
Wie das Projekt dokumentiert wurde



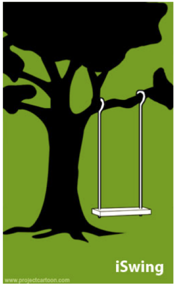
Welche Abläufe installiert wurden



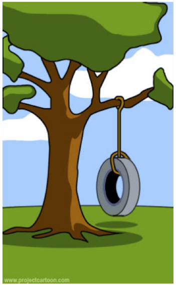
Wie es dem Kunden berechnet wurde



Wie der Support ist



Wie das Marketing damit wirbt



Was der Kunde wirklich gebraucht hätte

General Methodology for any Project in DSAI (and CS)

Step 1: Think

Develop a plan, a design, a concept, an architecture of the system you want to build.

Step 2: Think again!

Revise everything you did in Step 1, get feedback, get peer-review, let people challenge your ideas. Build throw-away prototypes and minimum viable products (MVPs).

If you changed a lot in Step 2, go back to Step 1.

until eventually:

Build!

Build your system/software or whatever you want to build according to your iteratively refined plan. Still allow for some flexibility in your design and future extensions. Make sure that every bit of what you are building is automatically tested from the beginning.

Collect data

:

- clarify privacy policy
- NDAs (Non-Disclosure Agreements)
- preselect data
- transmit and store the data securely
- encryption vs. offline solution
- versioning
- backup and archiving
- possibly certain (but important) data is not yet collected!

→ data collection, i.e. which data has to be collected anyways

:

- How should we handle the private data of your customers?
- Which legal restrictions should we consider?
- Which levels of security and privacy protection should we build in?
- So far you are only running a single (analogue) bookstore in downtown Saarbrücken? Ok.
- You do not have a digital book catalogue yet? Hmm.
- No digital customer catalogue? Hmmmmmm.

Explore&profile data

:

- semantics of the data
  - attributes? data types? schema?
  - data according to user vs. own analysis
- data generation:
  - which devices / input masks are used
  - how was this data generated?
  - can there be measurement errors?
- data quality?
  - what is NULL (i.e. value not assigned)?
  - what is DEFAULT?
  - measurement error, variance?
- if necessary: back to Data collection
- consult with domain experts
- understand data consistency

:

- What does the attribute "customer" mean? The person buying your books? The company shipping your books?
- In your document you wrote that the data you gave us contains a catalogue of all German books having an ISBN. However, we identified several books that are not part of your catalogue. Why is that?
- The data entries for 2015 differ considerably from the ones for 2017. How was the data for 2015 and 2017 generated? What changed? And why?

Data Consistency: data types and ranges

- data types:
  - 14. March vs March 14th vs 14. March 2007
  - repair:
    - common data type
    - distinguish internal storage of the data from representation of the data!
    - Model-View Design Pattern
- data type as enumeration:
  - man, woman, indefinite
    possible in passport since 2017, (see e.g. [Der Spiegel])
  - student studies Fach X, which doesn't exist at the university.
  - repair: comparison of data with list of allowed values
- valid data ranges:
  - student took part in lecture in summer semester 2047
  - salary $= -50000$ Euro
  - date of birth of the student is 2017
  - unlimited credit on bank account
  - repair: limit allowed data ranges reasonably

Data Consistency: Mandatory and Default Values

:

- mandatory values:
  - student has no matriculation number
  - employee has no boss
  - repair: force assignment of a value
- defaults values:
  - is 0 used as default?
  - or some other value
  - repair: replace erroneous defaults by NULL (not assigned)

:

- each book must have an ISBN
- books must have an author
- do not use “Goethe”. “NA”, “”, “unbekannt”, “Peter knows the name and will enter it later on” as placeholders for “author unknown”

Data Consistency: Duplicates Keys

:

- two students have the same matriculation number
- ...or the same ID card number
- two men are married to the same woman
- two cars have the same license plate[3]
- repair: define and enforce keys

:

- two books have the same ISBN
- two authors have the same authorID.
- two books have the same title (should be allowed, a book title is not a key)
- two authors have the same name (should be allowed, an author name is not a key)

[3] There is an exception in Germany: The interchangeable number plate, see `https://de.wikipedia.org/wiki/Wechselkennzeichen`. Here, however, the combination of interchangeable element and rigid part is unique!

Data Consistency: Complex Conditions

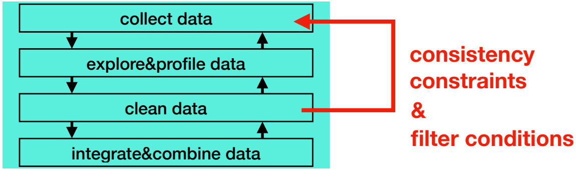
:

- bank transfer between accounts within the bank: sum across both accounts after the transfer differs from sum before the transfer
- boss has more than 30 subordinates, that's not even allowed by company rules.
- exhaust measurements of a car in driving mode "road" are higher than in driving mode "test bench"...

:

- customer should not be allowed to buy items for more than 500€
- customer should not be allowed to buy if his/her unpaid bills are higher than 200€
- we should not order more books than we can store
- we should not offer/sell books that are forbidden to sell (in general, to certain persons, to certain countries)

Data Consistency and Filter Pushdown



collect data

explore&profile data

clean data

integrate&combine data

consistency constraints & filter conditions

Pushdown of consistency constraints

The knowledge gained from data cleaning can be “pushed down” in the DSAI phase model (i.e. brought closer to the data sources) in order to check these consistency conditions in the future *already when the data is collected*.

Pushdown of filter conditions (aka Predicate Pushdown)

At this point already we could change data collection to not collect data we never need.

: do not collect user data on his/her weight, shoe size, age, gender, number of children, employer, income, favourite color, etc.

Consistency Constraints (on creation)

= Clean up data **while** it is being collected

this is the most common case in scenarios where there is a system collecting the data: then we might have control or have an influence on how and what data is collected

- determine in advance which consistency conditions should apply to data that is to be stored in a computer system
- automatically check these consistency conditions every time data is to be stored or modified:
  - either in the application
  - or in the system that stores the data
  - best: both
- if consistency of the data to be inserted/changed is violated: Generate error message and do not perform the change
- DataBase Management Systems (DBMS) offer extensive support for automatic checking of consistency conditions (integrity constraints, domains, foreign key constraints, triggers)

Clean data (afterwards)

= Clean up data **after** it has been collected

this is the most common case in textbook-style machine learning scenarios: the data “falls from the sky” and we have to live with it

What can we do?

- **delete** (omit missing attribute values):
  - throw away item, tuple and/or attribute
  - effect: less data, potentially corrupting the statistical properties of other attributes
- **impute** (replace missing and/or erroneous attribute values):
  - effect: more data, potentially corrupting the statistical properties of this attribute
  - how? median, mean, default values, interpolation
- **upsample/downsample** (enlarge/reduce dataset):
  - for time series adjust sample rate/frequency

Dark Data

Dark Data

with “Dark Data” we mean data which is not visible for analysis purposes, i.e. either:

1. may not yet be used for analytical purposes, or:
2. haven’t been digitised yet, or:
3. we don’t have access to.

This term is inspired by the term “dark matter” from physics.

Examples:

- Excel files in a company (1.)
- analog data collection (2.)
- internal data collection by Facebook, Google, Schufa, NSA, etc. (3.)

Survey

Why is it important to understand the origin of the data?

(A): Data is always only an image of “reality”. Confusing or even equating data with reality is a mistake.

(B): How data is generated may have an impact on the quality of the models and analytical results.

(C): The results of our analysis may have an impact on the generation of the data.

(D): Data generation may affect the choice of analytical methods.

Solution (A–D)

all correct!

Survey

How do I know that an“error in the data” is really an error and not an interesting (but e.g. rare or subtle) phenomenon?

(A): I don’t know that.

(B): I can compare my model of data generation (reality) with the data to find out.

Example of alleged measurement errors

Why do physicists claim that every odd number is a prime number?

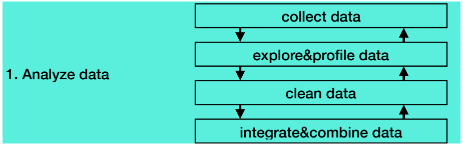
They test this as follows:
1 - prime.
3 - prime.
5 - prime.
7 - prime.
9 - measurement error.
11 - prime.
13 - prime.
... then the measurement series is aborted, because every further number must also be a prime number!

Solution (A&B)

both correct: some of the most important scientific discoveries are based on the fact that the division into “correct” and “wrong” measured values is questioned and newly adjusted, example: relativity theory

Back to the Analysis Phase: Integrate and combine data

We are here →



So we collected all data, know what data we need, and we even cleaned all data. Good!

And now what?

Database Management Systems to the rescue!

At this point you want to use a DBMS. There is a longer story behind this. You will learn about this in the undergrad lecture “Big Data Engineering” (aka Informationssysteme) in the fourth semester. And in the core lecture “Database systems” and or seminars if you want.

Database Management Systems (DBMS)

Relational database systems (RDBMS, mostly only DBMS) have been developed since the 70s. Modern DBMS typically have the following features:

1. advanced relational model: JSON, arrays, text, spatial data, etc.
2. very extensive SQL dialect (depending on system)
3. query optimiser (rule- and cost-based)
4. very high performance (mostly, depending on the system)
5. massive support for physical design (index structures, partitioning, caching, materialisation)
6. support for “modern” hardware: DRAM, PRAM, FPGAs, GPUs, ...

Database Management Systems (DBMS)

Features that deal with robustness and failover/error situations:

7. extensive support for transactions and ACID
8. concurrency control
9. crash recovery, replication, backup
10. automatic consistency control (unfortunately often not used)
11. dynamic views, access rights (logical data independence)

Important DBMS

commercial:

- Oracle
- MS SQL Server
- DB2
- Actian Vector
- SAP Hana
- Exasol
- ...

Open source:

- PostgreSQL
- MySQL
- SQLite
- ...

Good directory of (almost) all databases:
Database of Databases, `https://dbdb.io/`

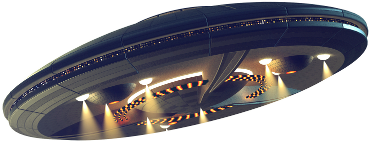
(on Oct 20th, 2019: 657 database management systems)

Different DBMS have different Performance Characteristics

Sqlite

MySQL

PostgreSQL

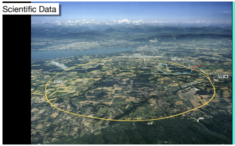


Modern DBMS

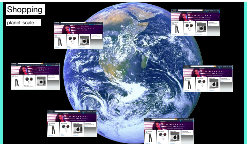
Take-away Message at this Point

99.999% of all data integration/combination/scalability/management/processing problems can be solved using relational DBMS technology.

The difficulty is often more about being able to know about, use, and combine the existing methods and tools in the right way.
In particular for our sample applications:


Scientific Data

Global Mass-Surveillance


Shopping

Cars as selective crowd-sourcers

From a database **research** perspective **none** of these applications is a real challenge anymore!

From a database **practitioner's** perspective **all** of these applications may be a **huge** challenge and require in-depth knowledge about and skills in database technology.

Summary

- The data analysis phase is (often) extremely important for the other phases.
- Do not underestimate the impact of this phase on the other phases! (Recall the Tesla example).
- the effort required in this phase may vary a lot depending on your application.

next Thursday: Introduction to AI