Part 2

Data Science in Practice

About me

Join us!

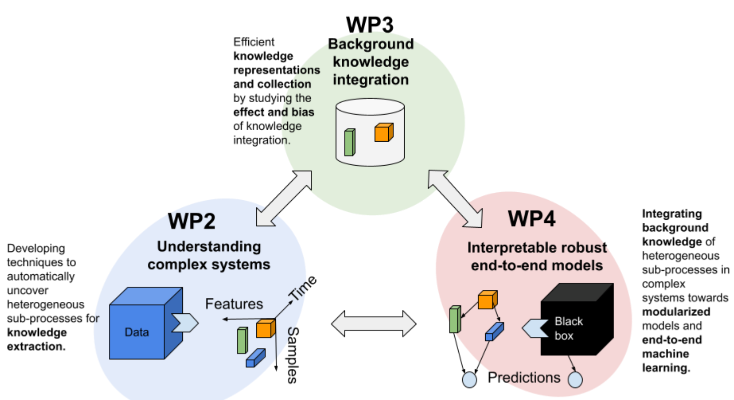
Becker Lab, University of Rostock
Junior Professor
Chair for Intelligent Data Analytics
BMBF Research Group Leader (Themis)

Nima Aghaeepour's Lab, Stanford University
Postdoc
Artificial Intelligence, Machine Learning, and Multiomics Integration for Translational Medicine

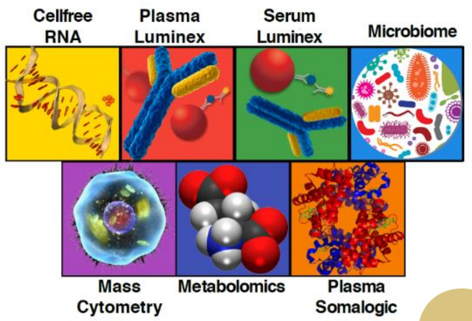
Andreas Hotho's Lab, University of Wuerzburg
PhD
Data Mining and Information Retrieval Group

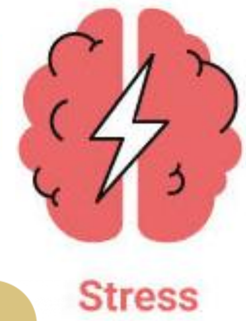
Becker Lab

Knowledge-centric AI
- Knowledge extraction
- Knowledge integration
- Impactful applications

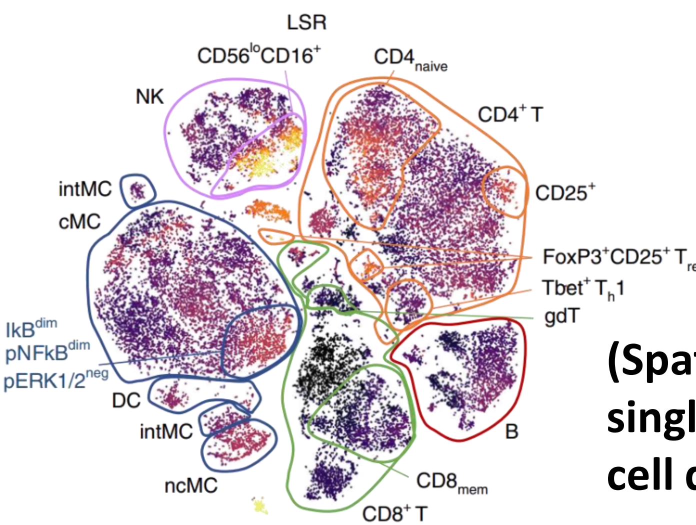


Applications: Biomedical systems, behavioral analysis, intrusion detection, environmental modeling, etc.





Multiomics integration for profiling
- Pregnancy
- Aging
- Rare diseases
- Cardiovascular systems
- Nutrition



(Spatial) single cell data

Part 2: Goal

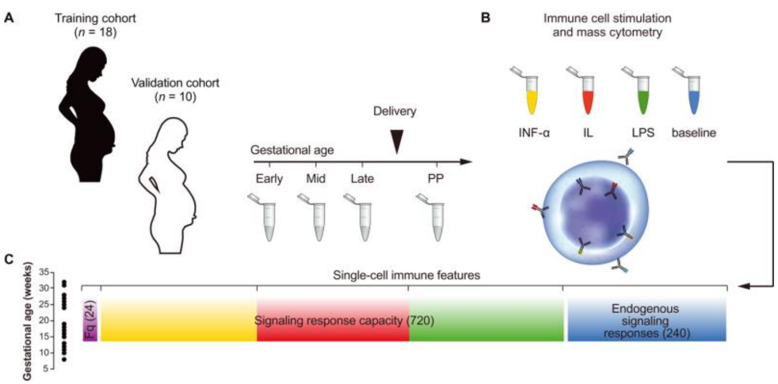
- Learn to complete a “simple” data science project

- (Very) preliminary structure
  - Basics (fitting prediction models)
  - Data preprocessing, exploration, and statistics
  - Practical aspects
  - Interpretability, fairness, etc.
  - Advanced prediction models (e.g., neural networks) and outlook

Part 2: “Simple” Data Science Project

Science Immunology

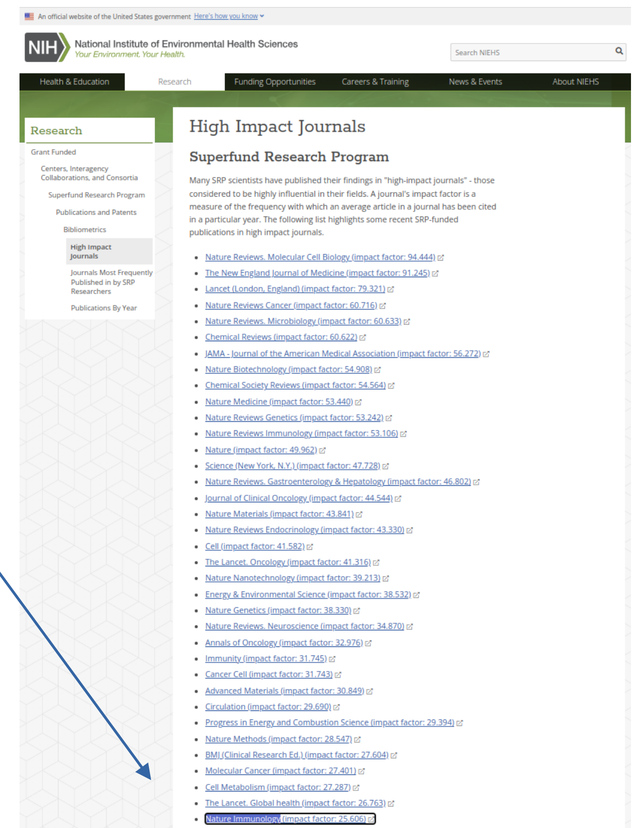
An immune clock of human pregnancy



Immune function is altered during pregnancy to protect the fetus from an immunological attack without disrupting protection against infection.

Mass cytometry to **examine** the precise **timing of these pregnancy-induced changes** in immune function and regulation.

By **defining this immunological chronology** during normal term pregnancy, they can now begin to determine which alterations associate with pregnancy-related pathologies.


An official website of the United States government Here's how you know

NIH National Institute of Environmental Health Sciences
Your Environment. Your Health.

Search NIEHS

Health & Education | Research | Funding Opportunities | Careers & Training | News & Events | About NIEHS

Research

- Grant Funded
  - Centers, Interagency Collaborations, and Consortia
    - Superfund Research Program
      - Publications and Patents
        - Bibliometrics
          - High Impact Journals
          - Journals Most Frequently Published in by SRP Researchers
          - Publications By Year

High Impact Journals

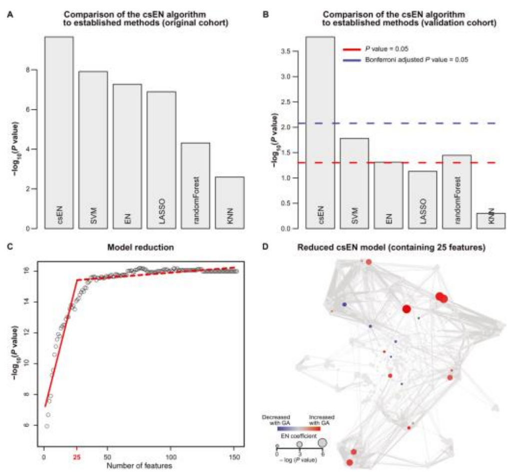
Superfund Research Program

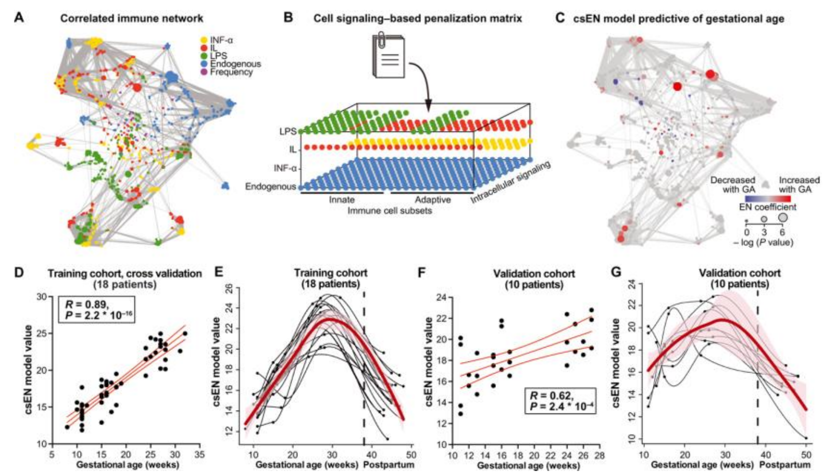
Many SRP scientists have published their findings in "high-impact journals" - those considered to be highly influential in their fields. A journal's impact factor is a measure of the frequency with which an average article in a journal has been cited in a particular year. The following list highlights some recent SRP-funded publications in high impact journals.

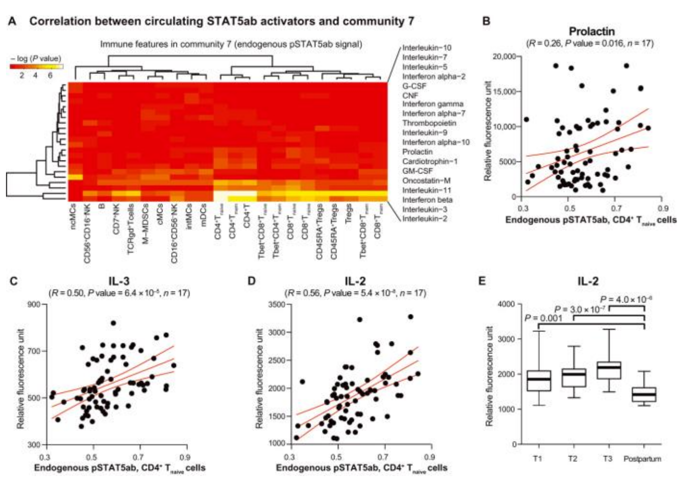
- Nature Reviews. Molecular Cell Biology (impact factor: 94.444)
- The New England Journal of Medicine (impact factor: 91.245)
- Lancet (London, England) (impact factor: 79.321)
- Nature Reviews Cancer (impact factor: 60.716)
- Nature Reviews. Microbiology (impact factor: 60.633)
- Chemical Reviews (impact factor: 60.622)
- JAMA - Journal of the American Medical Association (impact factor: 56.272)
- Nature Biotechnology (impact factor: 54.908)
- Chemical Society Reviews (impact factor: 54.564)
- Nature Medicine (impact factor: 53.440)
- Nature Reviews Genetics (impact factor: 53.242)
- Nature Reviews Immunology (impact factor: 53.106)
- Nature (impact factor: 49.962)
- Science (New York, N.Y.) (impact factor: 47.728)
- Nature Reviews. Gastroenterology & Hepatology (impact factor: 46.802)
- Journal of Clinical Oncology (impact factor: 44.544)
- Nature Materials (impact factor: 43.841)
- Nature Reviews Endocrinology (impact factor: 43.330)
- Cell (impact factor: 41.582)
- The Lancet. Oncology (impact factor: 41.316)
- Nature Nanotechnology (impact factor: 39.213)
- Energy & Environmental Science (impact factor: 38.532)
- Nature Genetics (impact factor: 38.330)
- Nature Reviews. Neuroscience (impact factor: 34.870)
- Annals of Oncology (impact factor: 32.976)
- Immunity (impact factor: 31.745)
- Cancer Cell (impact factor: 31.743)
- Advanced Materials (impact factor: 30.849)
- Circulation (impact factor: 29.690)
- Progress in Energy and Combustion Science (impact factor: 29.394)
- Nature Methods (impact factor: 28.547)
- BMJ (Clinical Research Ed.) (impact factor: 27.604)
- Molecular Cancer (impact factor: 27.401)
- Cell Metabolism (impact factor: 27.287)
- The Lancet. Global health (impact factor: 26.763)
- Nature Immunology (impact factor: 25.606)

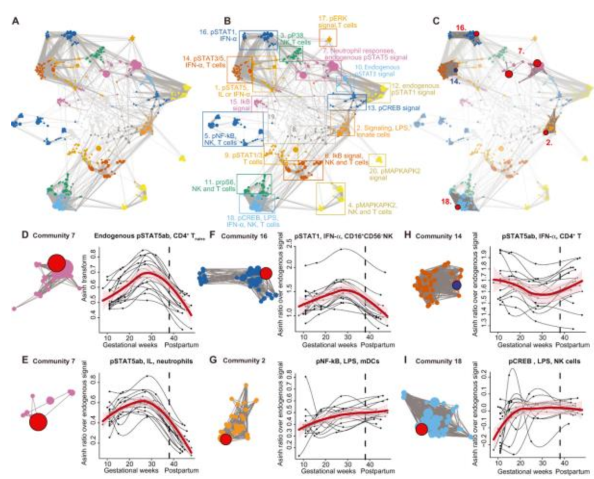
https://tools.niehs.nih.gov/srp/publications/highimpactjournals.cfm

Part 2: “Simple” Data Science Project









Part 2: Time plan

	Content	Project
15.11.2023	**Lecture 1: Terms and Basic Machine Learning Models**	Homework: Setup Jupyter and Tutorial
20.11.2023	Tutorial 1: Basics and fit our first model, **scaling**	Get to know the data and fit a first model
23.11.2023	**Lecture 2: Model evaluation and hyper-parameter tuning**	
27.11.2023	Tutorial 2: Evaluation strategies (including leave one group out), overfitting, hyper-parameter tuning, scaling	**Report first model (1 slide);** Design evaluation strategy, compare models, visualize results
29.11.2023	**Lecture 3: Preprocessing and data analysis**	
04.12.2023	Tutorial 3: T-SNE, Clustering	**Report model comparison and results (1 slide);** T-SNE plots and Clustering
06.12.2023	**Lecture 4: Univariate analysis, feature importance, fairness**	
11.12.2023	Tutorial 4: Univariate analysis	**Report data analysis visualizations (1 slide);** Visualize univariate and feature importance analysis, in-depth analysis, **play**
13.12.2023	**Lecture 5: Advanced models, Outlook**	
18.12.2023	Tutorial 5: Knowledge integration and neural networks	**Show off results and visualizations (1 slide);** Visualize univariate and feature importance analysis

Organizational

- Slides
  - Provided after lecture
- Exercises / Tutorials
  - Small groups working on problem sets
  - Solutions will be provided after tutorial
  - Feedback?
- Project
  - Starts after first tutorial
  - One slide of results every week
  - Optional
- Questions, Feedback
  - Personally
  - E-Mail: martin.becker@uni-rostock.de
  - bjarne.hiller@uni-rostock.de
  - Anonymous: https://moored.co/b/TLyNyULoqB

Outline

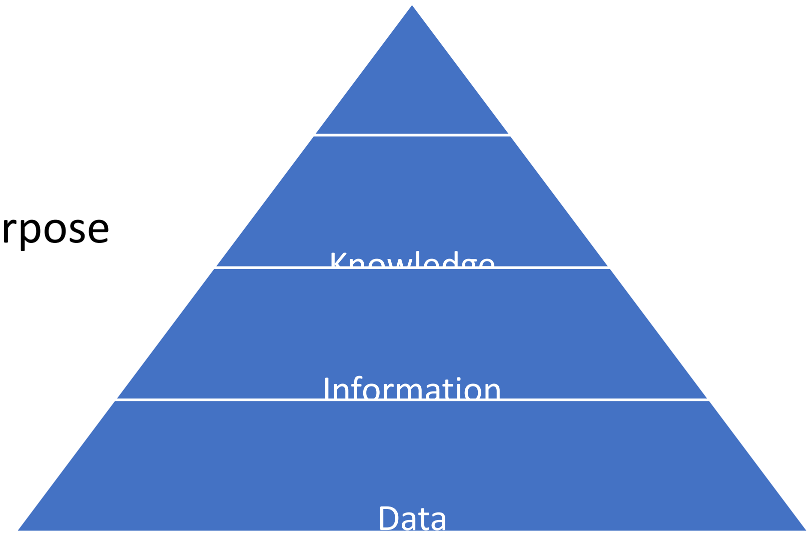
- Basic terms
- Machine Learning

Basic Terms

Data - Information - Knowledge

"Data is not information,
information is not knowledge,
knowledge is not wisdom." [C. Stoll]

- Data
  Raw data (measurements, "facts")
- Information
  Significant, summarized data for a specific purpose
- Knowledge
  Knowledge that people are aware of
- Be aware:
  Many contradictory definitions exist



DIKW Pyramid

History of Data Science

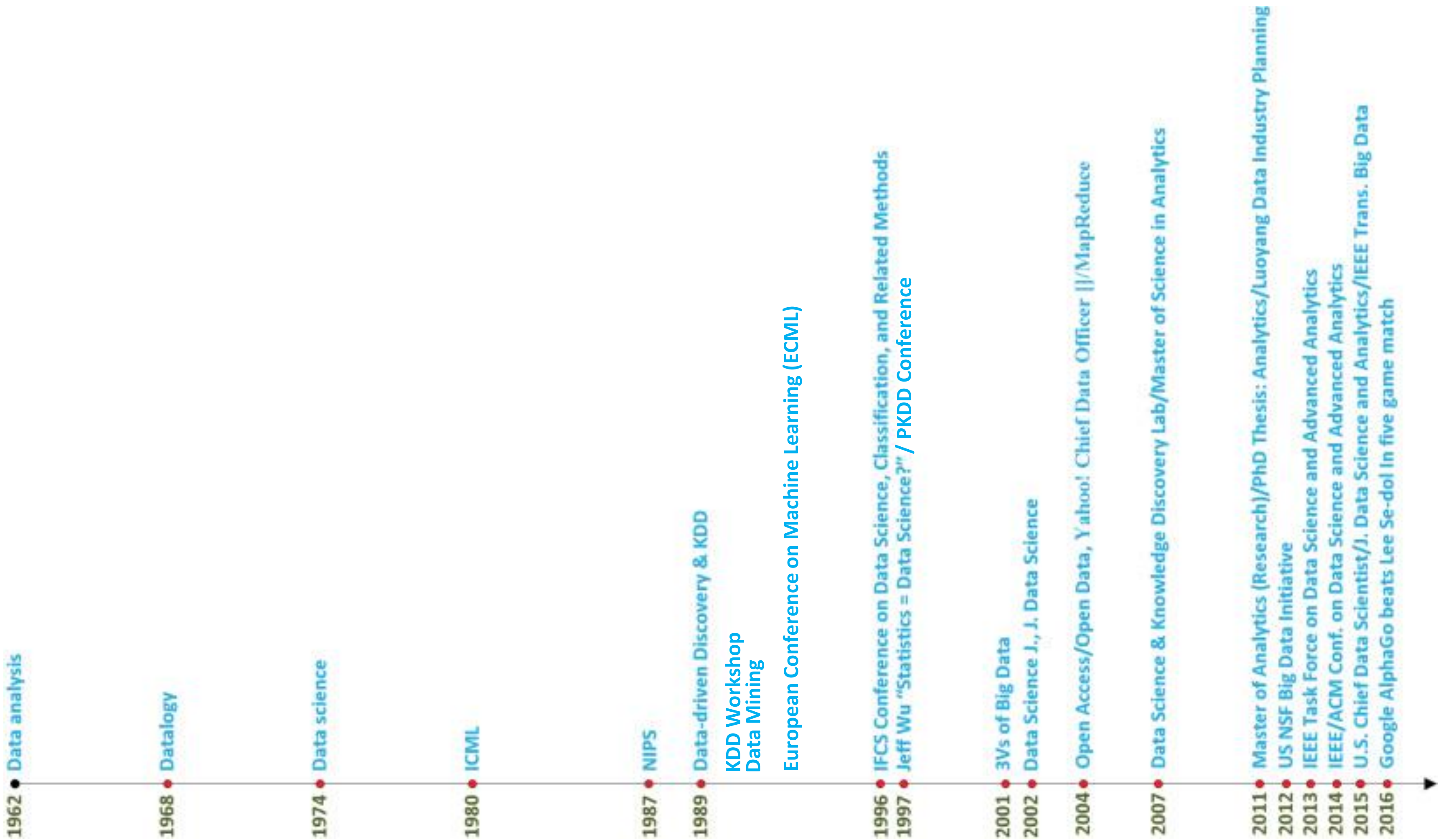
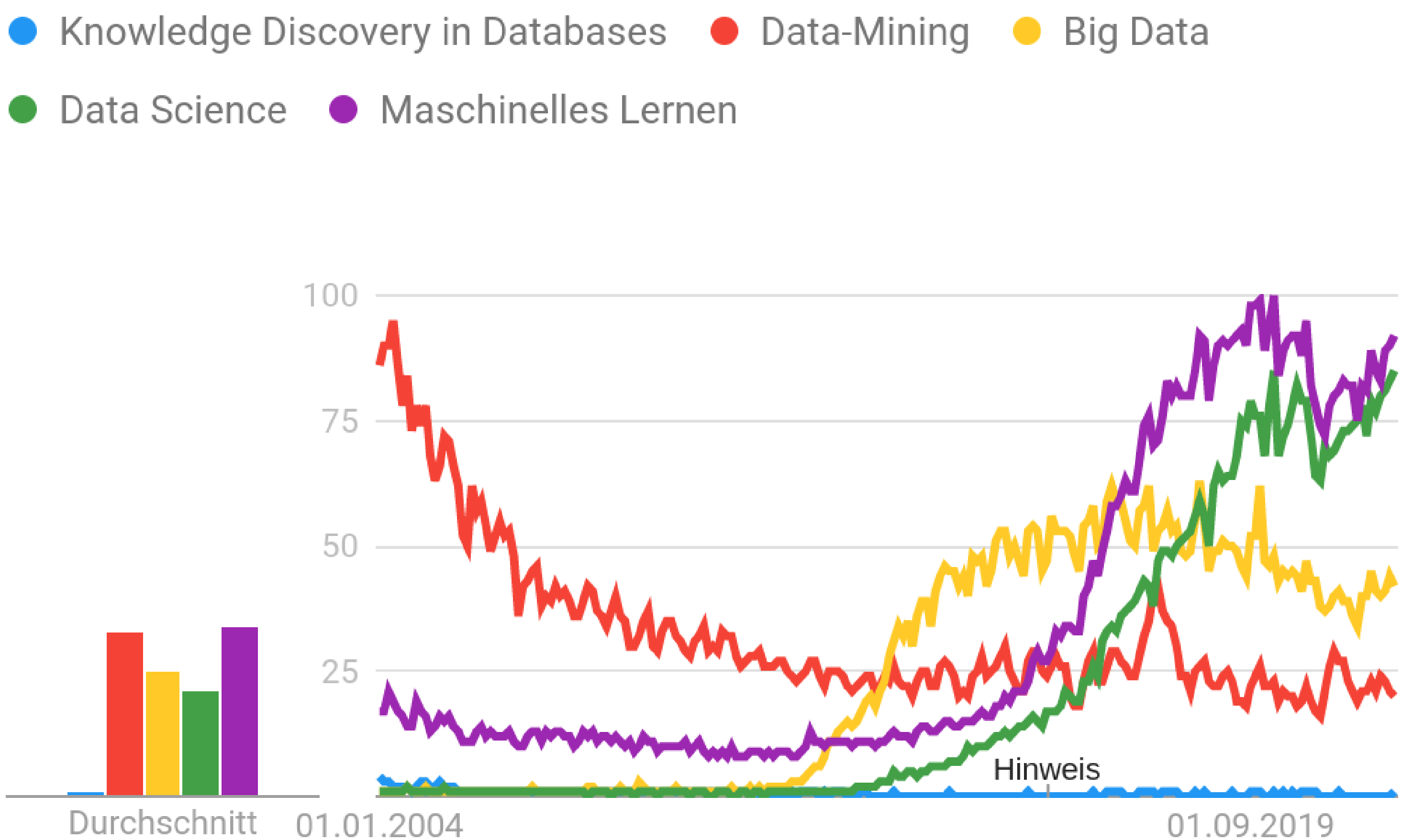


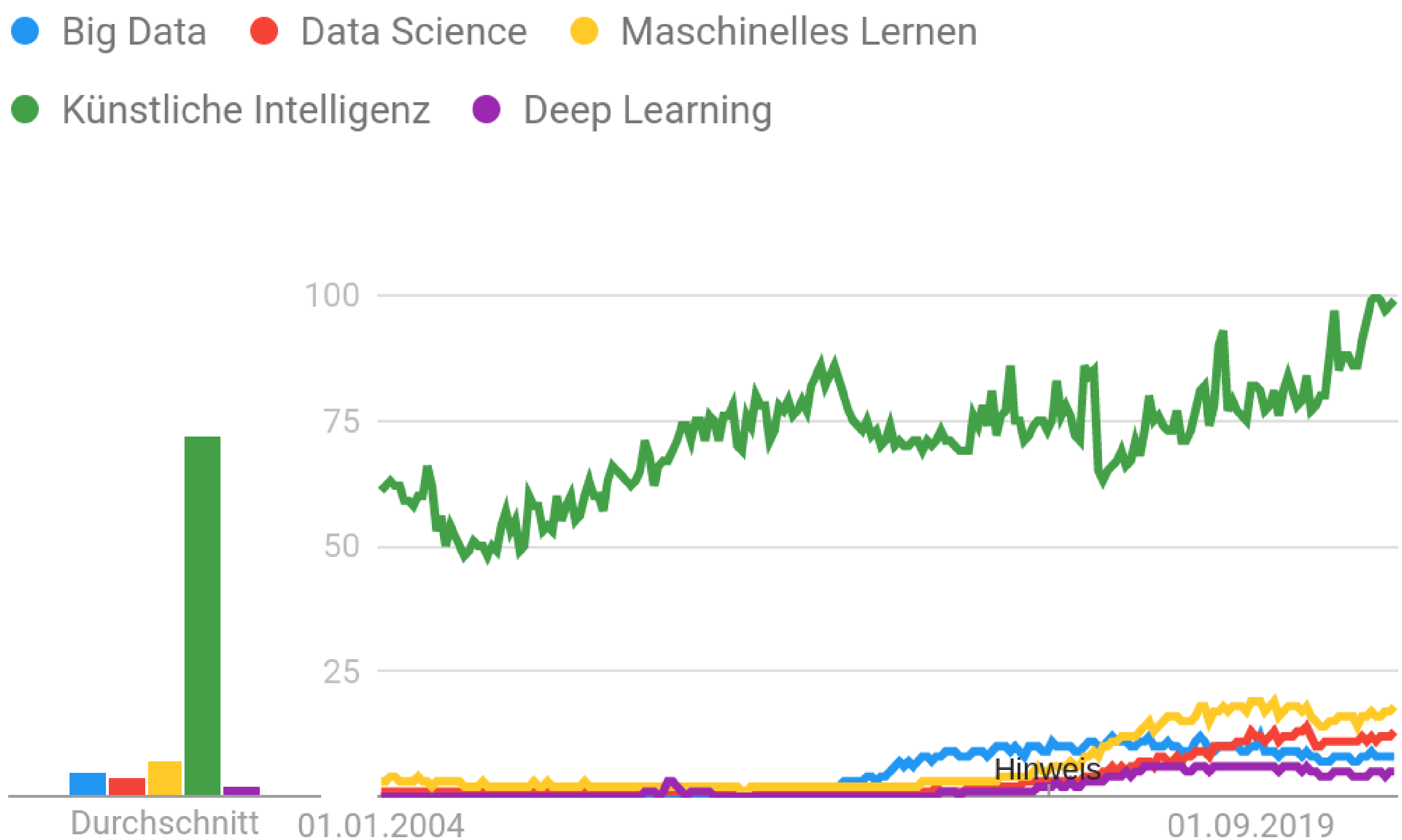
Image Source: Cao, Data Science a Comprehensive Overview, ACM Computing Surveys, Volume 50, Issue 3, 2017

Search History



Google Trends from Jan 1st 2004 to April 28th 2022

Search History



Google Trends from Jan 1st 2004 to April 28th 2022

Knowledge Discovery in Databases (KDD)

- Fayyad et al.* define KDD in 1996 as

  *The nontrivial process of identifying **valid**, **novel**, potentially **useful**, and ultimately **understandable** patterns in data.*

- The four characteristics are explained as follows:

Valid	The found patterns also apply for new data
Novel	The system/user did not know that this pattern existed
Useful	The result can be used to solve a given task
Understandable	The user should know how/why it works (however, this is a subjective measure)

- Fayyad et al. state that

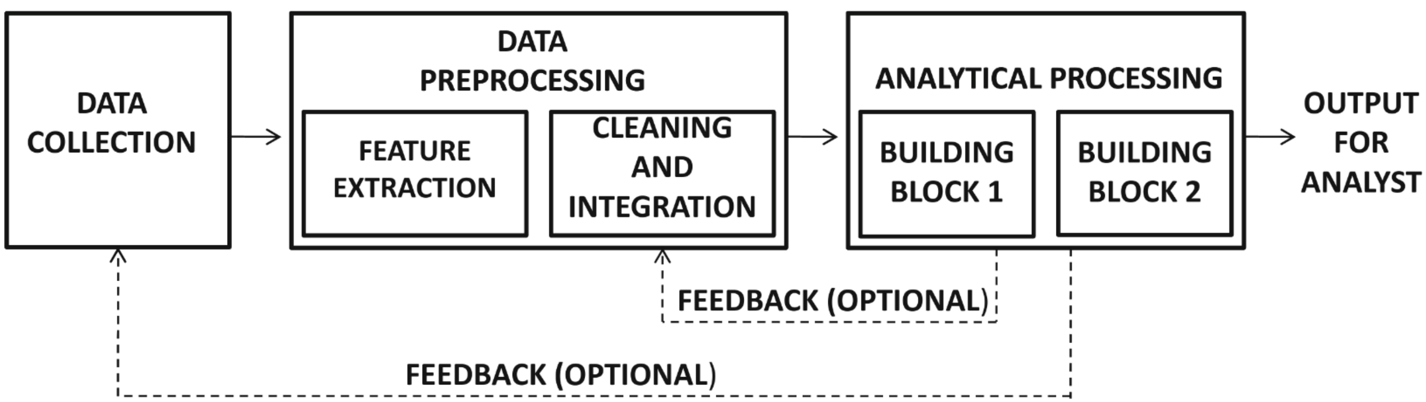
  ***Data mining** is a particular step [in KDD] – application of specific algorithms for extracting patterns (models) from data.*

* The KDD Process for Extracting Useful Knowledge from Volumes of Data, Communications of the ACM, Vol. 39, No. 11, 1996

Data Mining

Aggarwal* defines Data Mining in 2015 as

Data Mining is the study of collecting, cleaning, processing, analyzing, and gaining useful insights from data. […] "Data mining" is a broad umbrella term that is used to describe these different aspects of data processing.



(A standardized Data Mining process will be discussed later)

* Aggarwal, Data Mining: The Textbook, Springer, 2015

Big Data

- De Mauro et al.* define Big Data in 2016 as

  Big Data is the Information asset characterized by such a High Volume, Velocity and Variety to require specific Technology and Analytical Methods for its transformation into Value.

- Big Data Analytics is similar to Data Mining, but especially focuses on large data volumes where "classical methods" can not be used efficiently

* De Mauro et al., A Formal Definition of Big Data Based on its Essential Features, Library Review, Vol. 65, Iss. 3, 2016

Data Science

Cao* defines Data Science in 2017 as

*From the disciplinary perspective, **data science** is the new **interdisciplinary field** that synthesizes and builds on statistics, informatics, computing, communication, management, and sociology **to study data** and its environments (including domains and other contextual aspects, such as organizational and social aspects) in order to **transform data to insights and decisions** by following a data-to-knowledge-to-wisdom thinking and methodology.*

but also gives another (more simple) definition:

Data Science is the science of data.

* Cao, Data Science: A Comprehensive Overview, ACM Computing Surveys, Vol. 50, No. 3, 2017

Relationship of Data Science and Data Mining

Data science, also known as **data-driven science**, is an interdisciplinary field of **scientific** methods, processes, algorithms and systems to extract knowledge or insights from **data** in various forms, either structured or unstructured, similar to **data mining**.

https://en.wikipedia.org/wiki/Data_science

[Dhar; 2013]

"„... At a high level, **data science** is a set of fundamental principles that support and guide the principled **extraction of information and knowledge from data**. Possibly the most closely related concept to **data science is data mining** - the actual extraction of knowledge from data via technologies that incorporate these principles. ...“

[Provost & Fawcett; 2013]

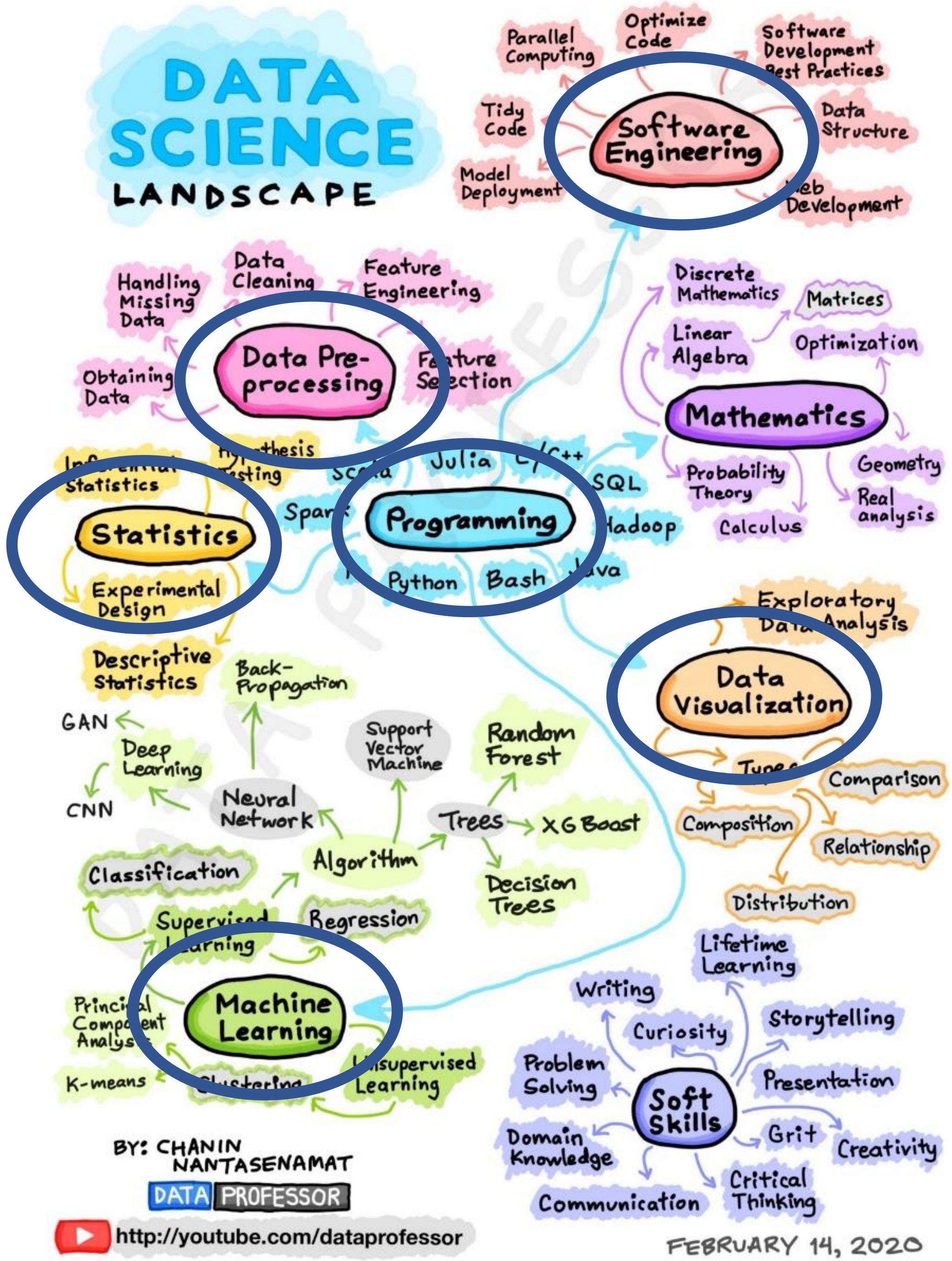
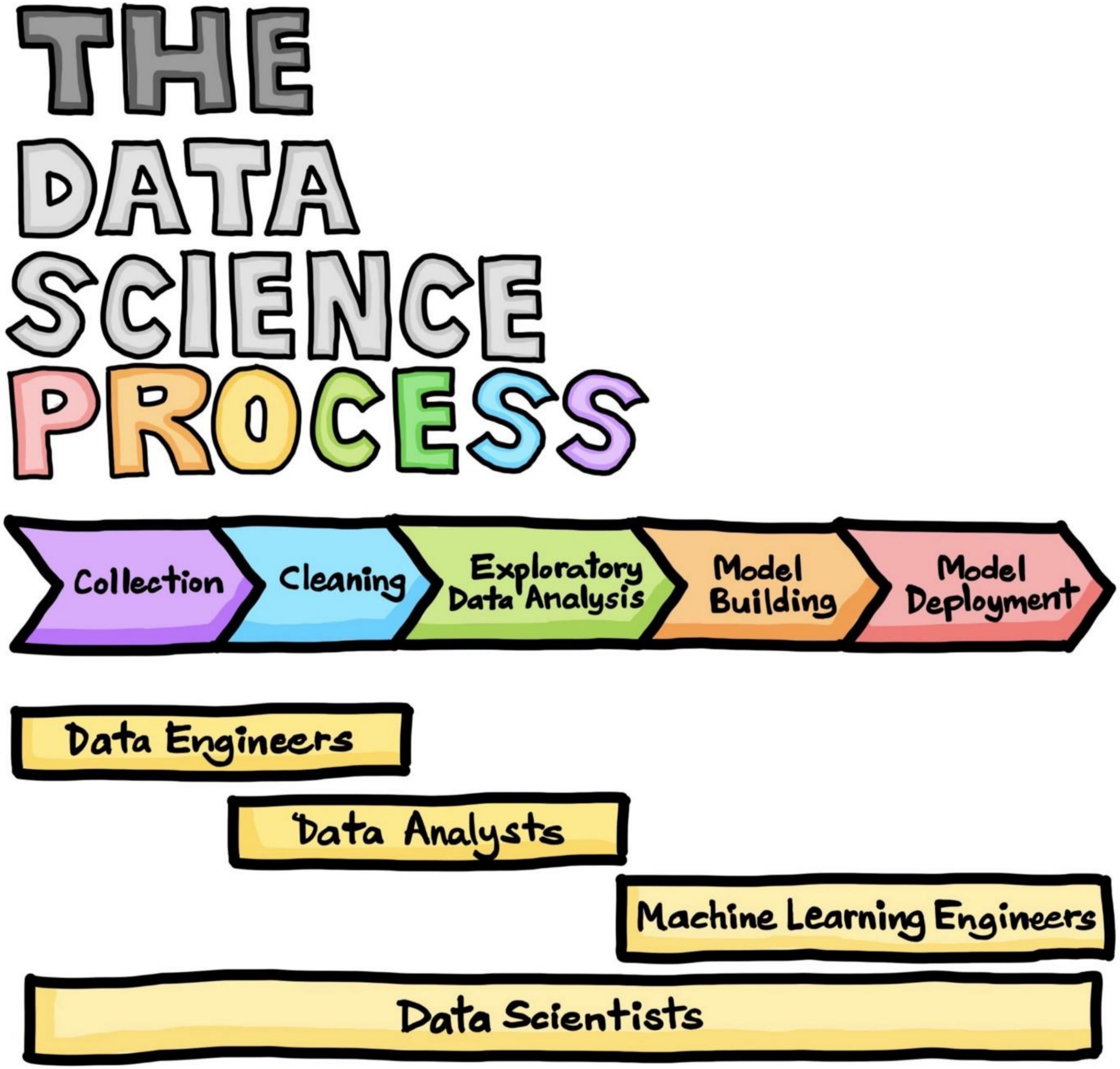


Image source: https://towardsdatascience.com/the-data-science-process-a19eb7ebc41b, accessed on April 28th, 2022

Data Science Process


THE DATA SCIENCE PROCESS
Collection
Cleaning
Exploratory Data Analysis
Model Building
Model Deployment
Data Engineers
Data Analysts
Machine Learning Engineers
Data Scientists

The Data Science Process

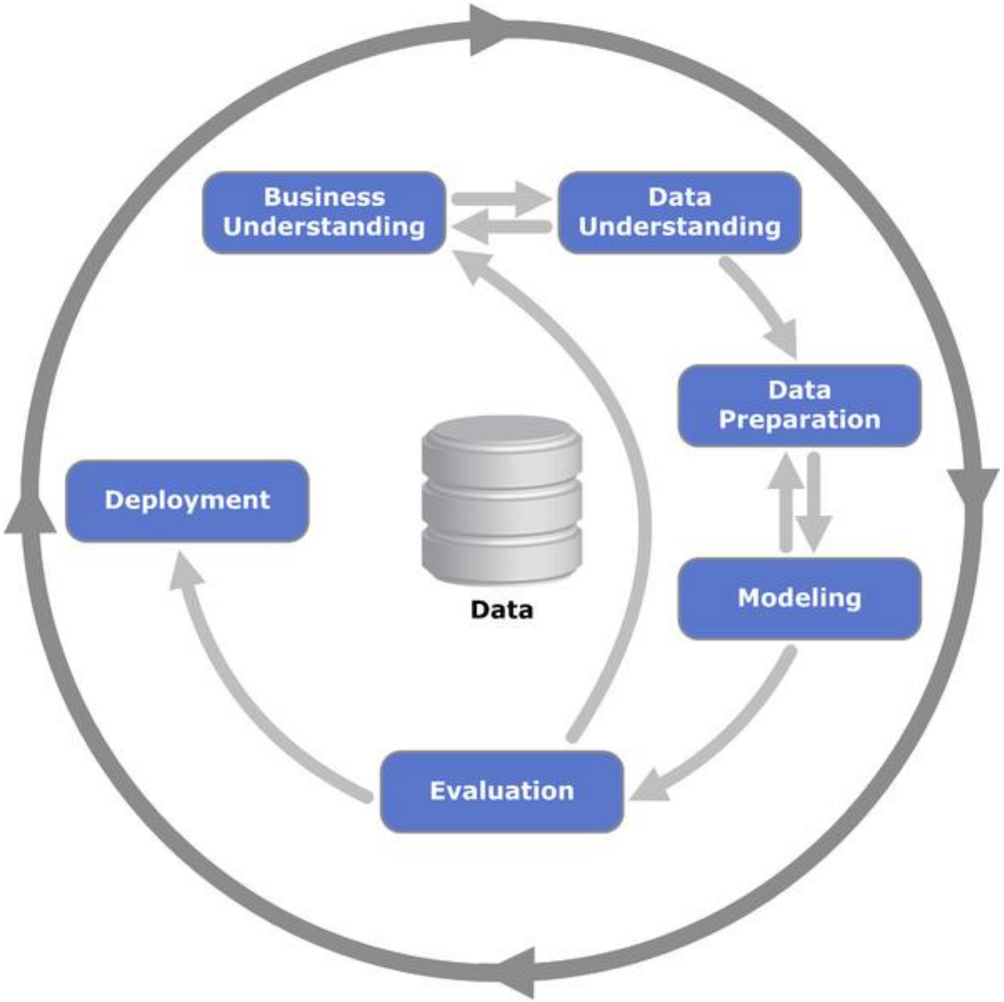
- The process must be related to the task and the user

- The developer needs knowledge about **databases, data analysis** methods and the **application area**

- The process is **interactive** and **iterative**
  - No full automation
  - Results have to be evaluated before making a decision
  - Some steps might be repeated depending on the results

- One well know process definition is the open standard process model CRISP-DM

The CRISP-DM Model



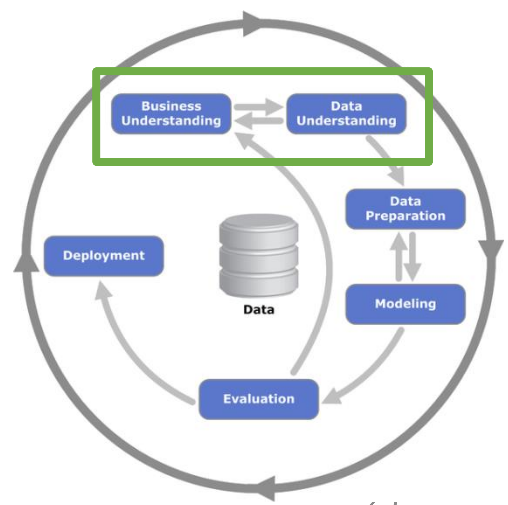
Cross Industry Standard Process for Data Mining

Main phases
(top-level processes)

1. Business Understanding
2. Data Understanding
3. Data Preparation
4. Modeling
5. Evaluation
6. Deployment

Business Understanding, Data Understanding

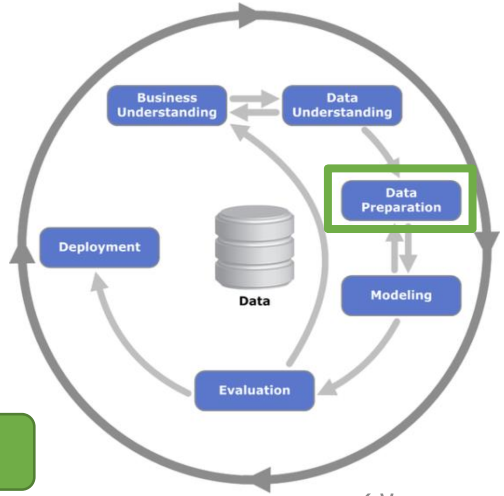
- Understanding the given application
- Defining the goal(s) of the Data Mining project
  - What should be achieved?
- Acquiring data from source(s)
- Clarifying data management
  - File System or DBS?
- Selecting relevant data



Preprocessing

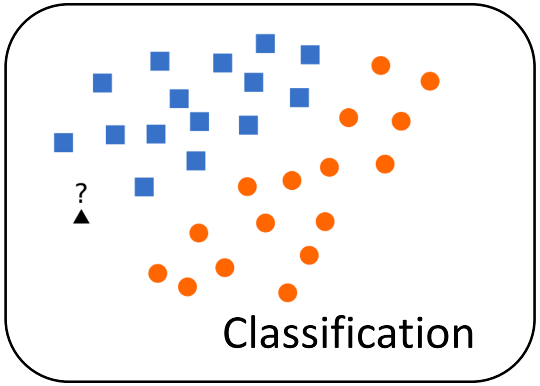
- Integrating data from different sources
- Checking consistency
- Cleaning

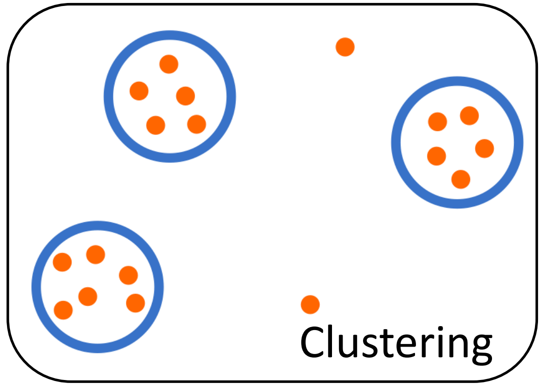
- Discretizing numerical features
- Generating derived features
- …

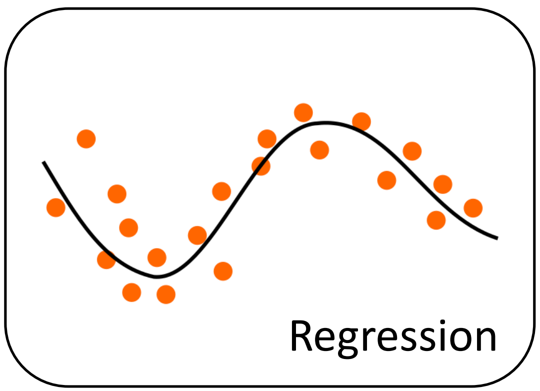


➔ More about this in Chapter 1.5: Preprocessing

Modeling: Methods

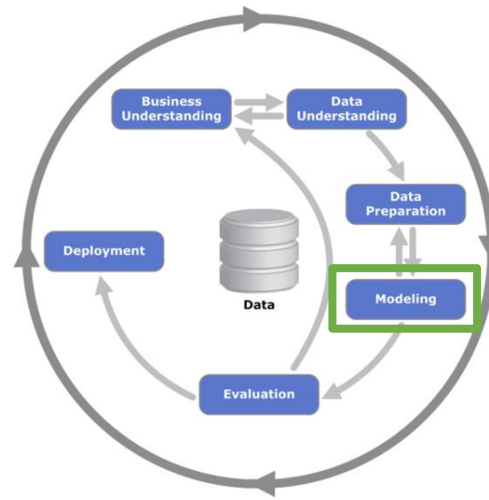






$$A, B \rightarrow C$$

Association Rules

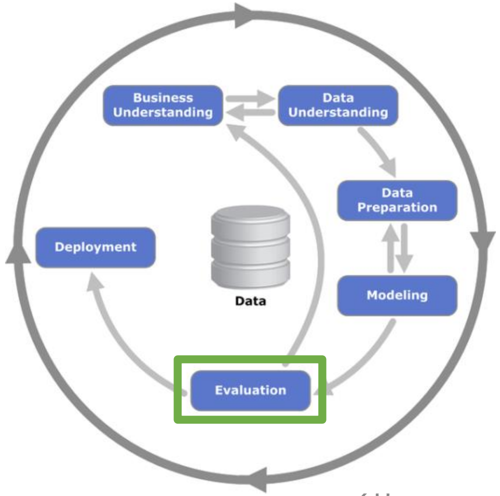


Other tasks:
Subgroup Discovery, Outlier Detection, Segmentation, …

➔ More details in later chapters

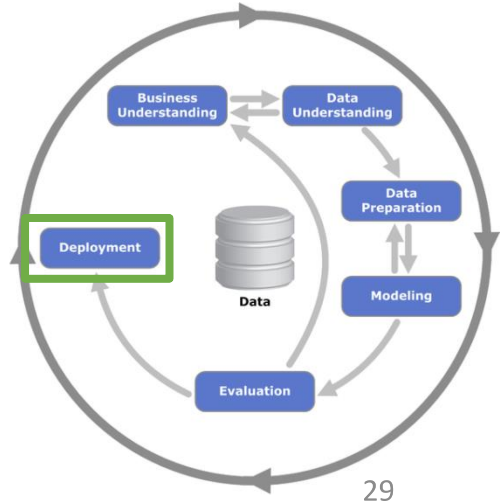
Evaluation

- Presenting the found patterns
  (often through appropriate visualizations)
- Evaluating patterns by the user
  - Predictive power of patterns and/or models
  - Pattern known or surprising?
  - Patterns and/or models applicable to many cases?
- If negative evaluation, then renewed data science with
  - Different parameters, different methods , different data
- If positive evaluation, then
  - Integration of the found knowledge into the knowledge base
  - Use of the new knowledge for future Data Science processes

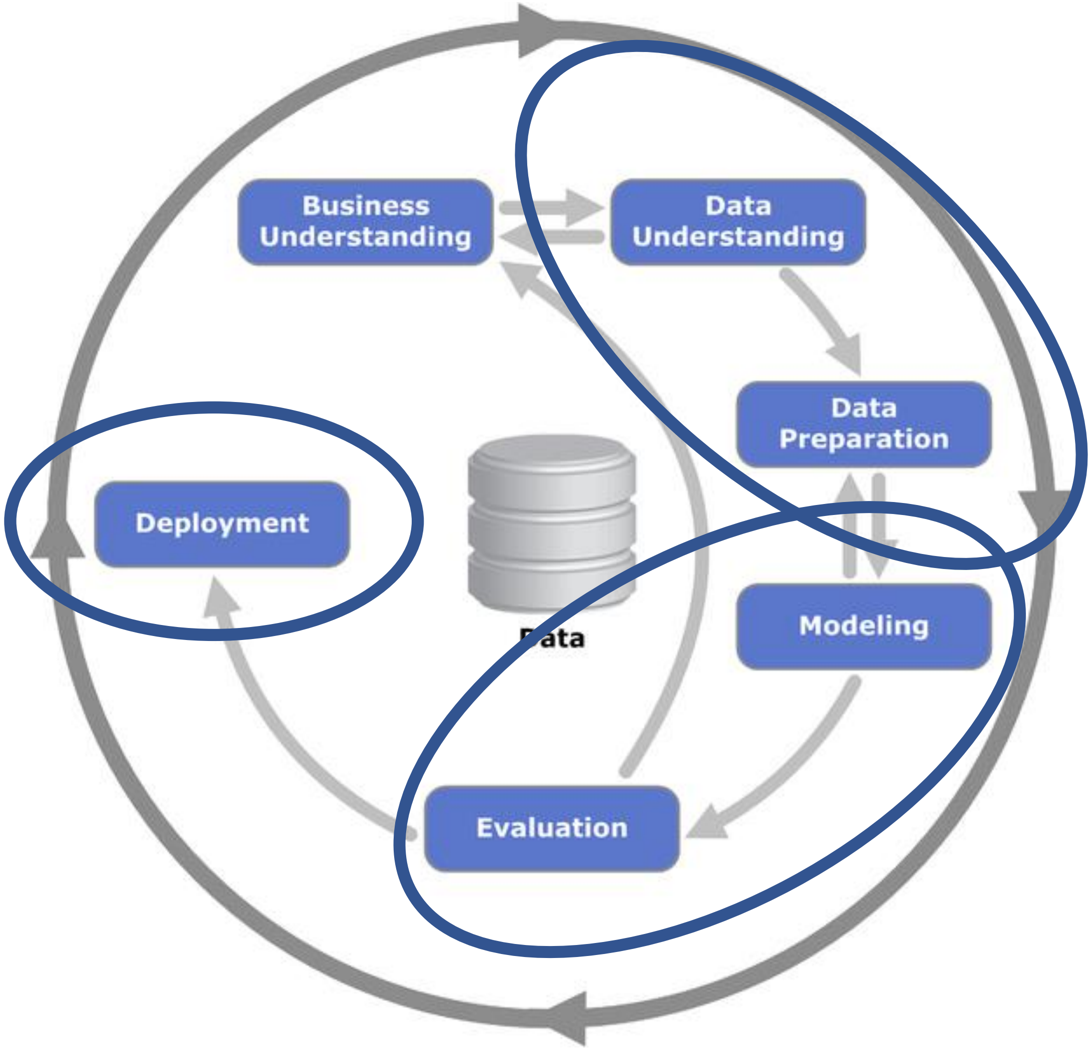


Deployment:
Creation of a Business Application

- Planning the use of the Data Mining application
  - Creation of a plan for the introduction of the application
- Planning of monitoring and maintenance
  - When should models no longer be used?
  - Do business objectives change over time?
- Preparation of the final report
  - Who is the target group for the presentation?
- Review of the project
  - Summary of the most important
    Knowledge and experience
  - Integration of the project results
    into the strategy of the entire company

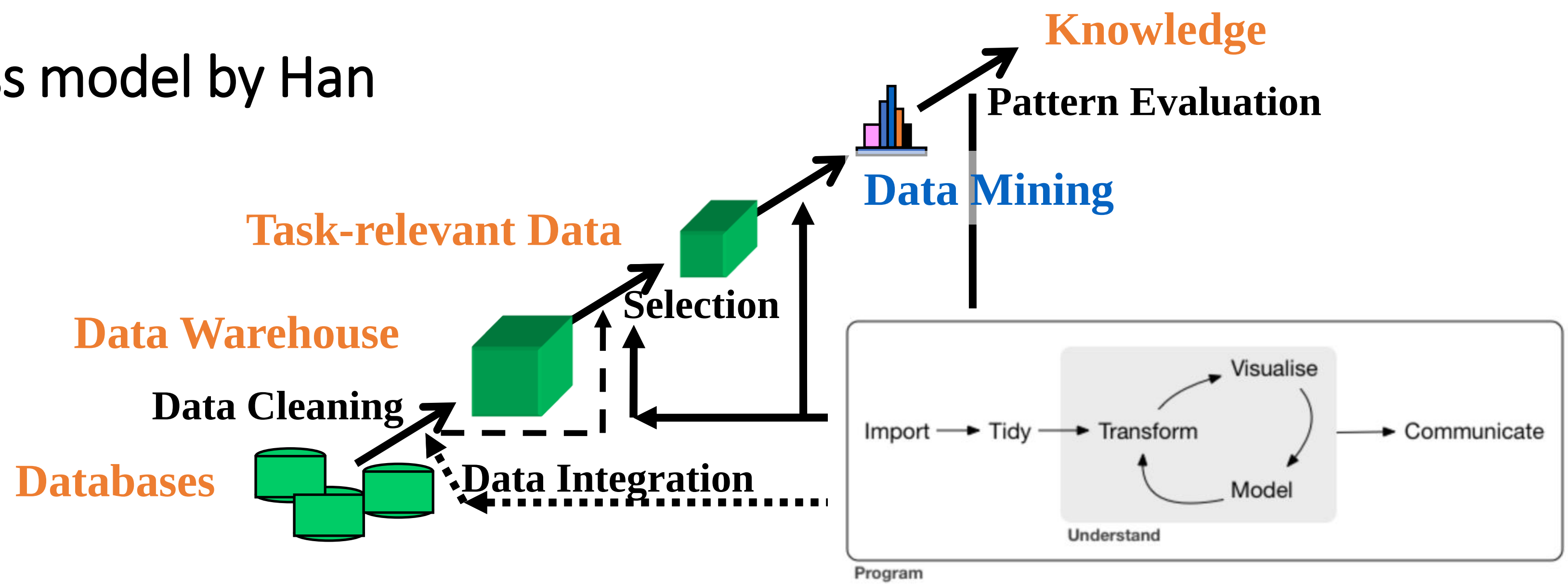


CRISP-DM Model



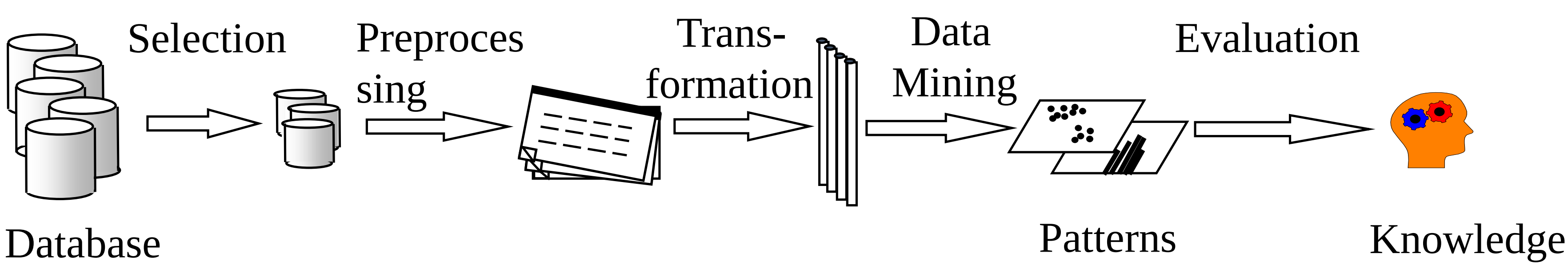
Alternative Process models

Process model by Han



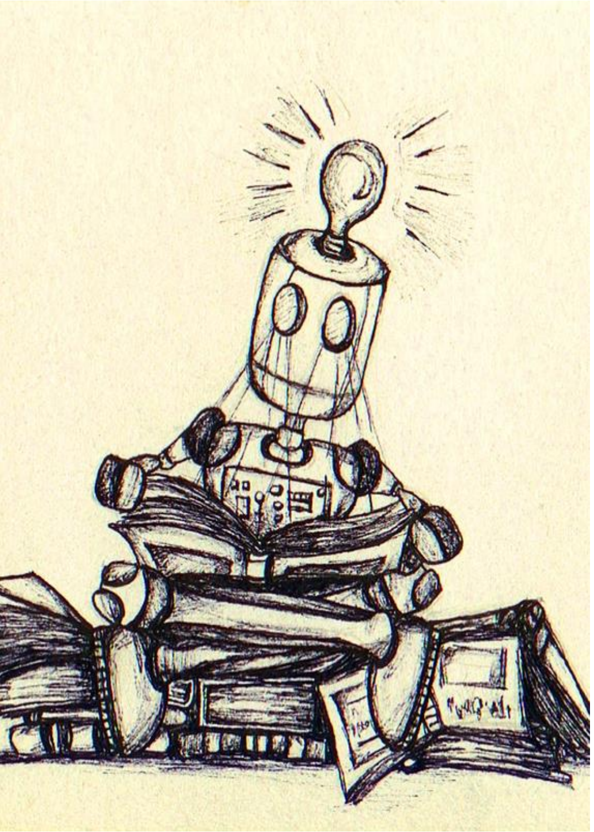
Aus: Wickham & Grolemund, R for Data Science, https://r4ds.had.co.nz/

Process model by Fayyad, Piatetsky-Shapiro & Smyth



Machine Learning

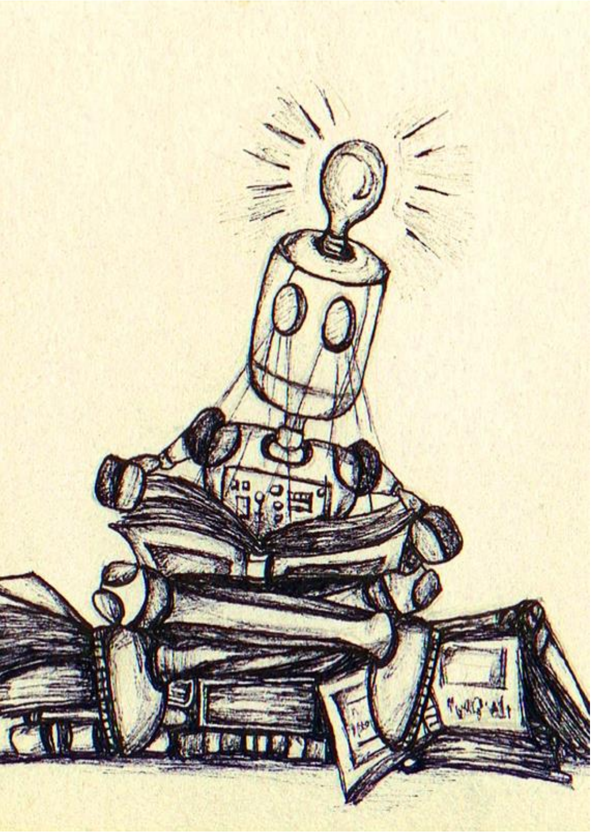
Machine Learning (ML)



- *„**Machine learning** (**ML**) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.**”*** - Wikipedia

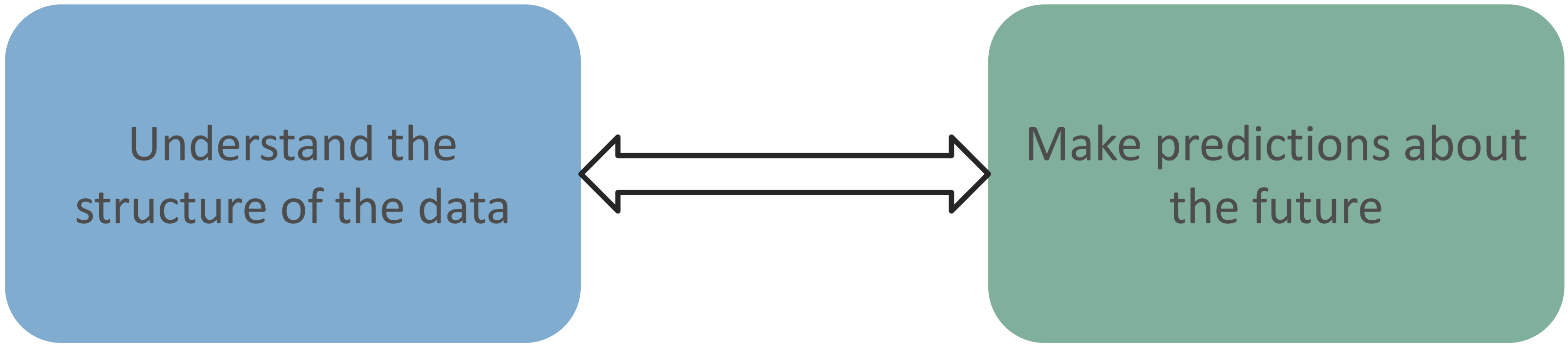
- *Learn a **model** from **training** examples, apply model to make **predictions** about the future*

Machine Learning (ML)

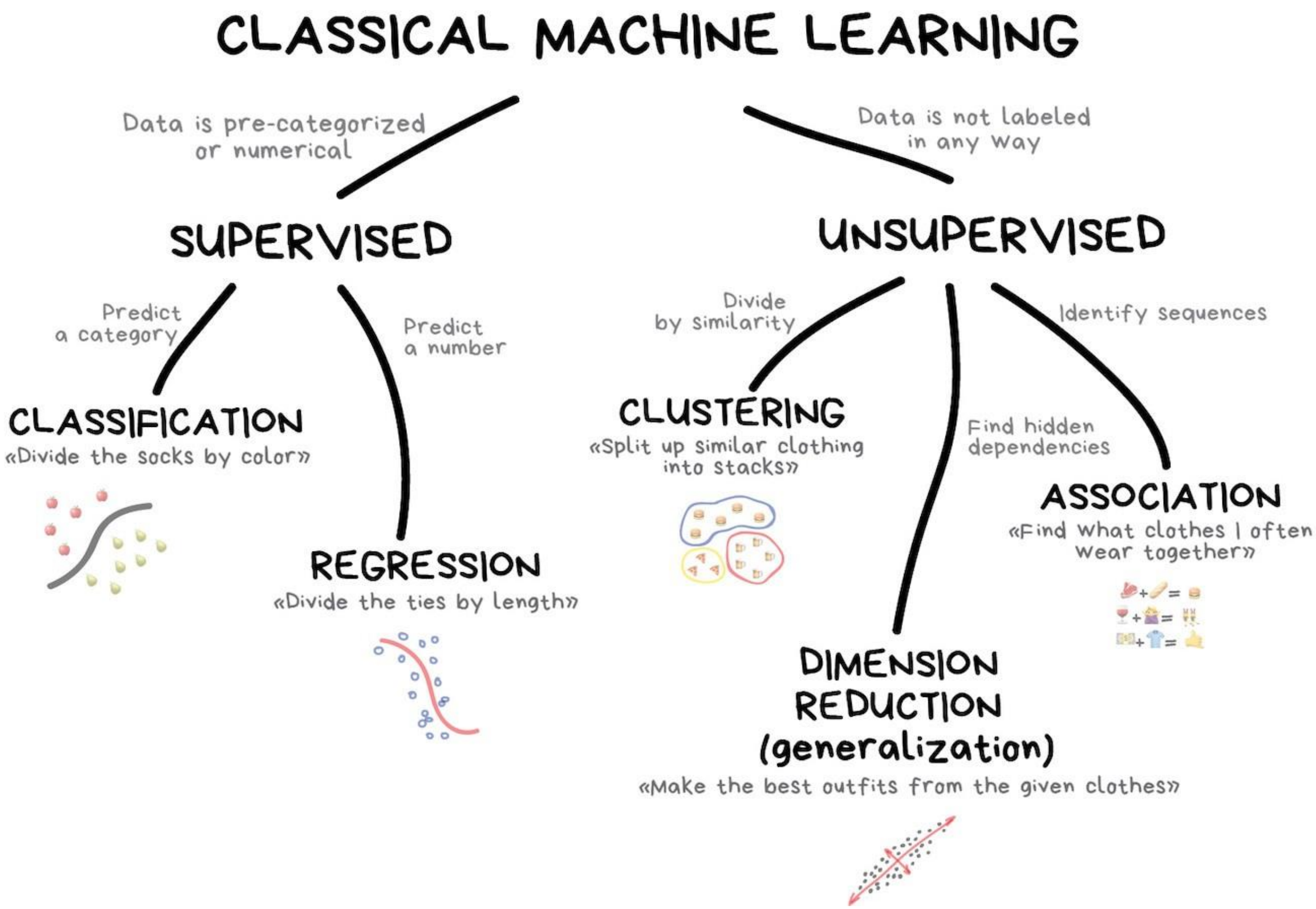


- „*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.*" – T. Mitchell

Main Goals


Understand the structure of the data
Make predictions about the future

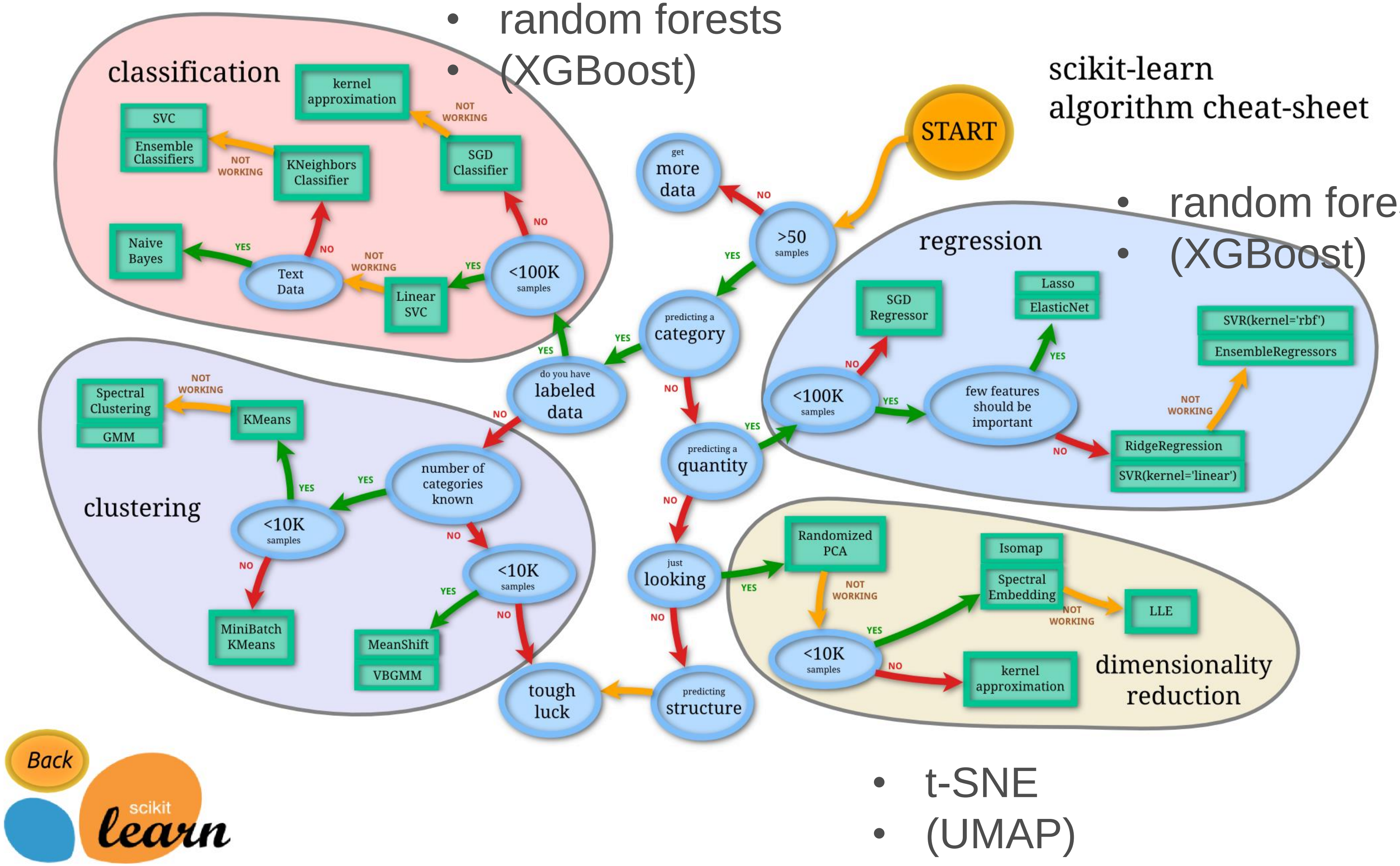
Overview of „Classical“ Machine Learning



https://vas3k.com/blog/machine_learning/

Overview of Machine Learning by Scikit-Learn

- random forests
- (XGBoost)



- random forests
- (XGBoost)

- t-SNE
- (UMAP)

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

Machine Learning Tasks by Training Feedback

- Supervised Machine Learning: $\{(x_i, y_i)\}_{i=1}^N$
  - There is one (or more) specific things to predict
  - Learn model parameters from feature-label pairs
  - Training examples are given that include information on that thing

- Unsupervised Machine Learning $\{(x_i)\}_{i=1}^N$
  - No prediction of a specific thing
  - Learn useful properties about the structure of the features
  - Learn model parameters using dataset without labels

Supervised Machine Learning

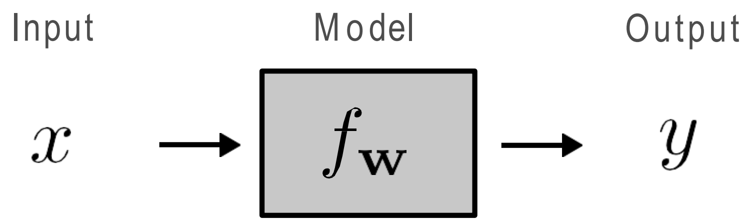
$$f: X \to \mathbb{N}$$

$$f: X \to \mathbb{R}$$

$$f: X \to Y$$

- Inputs $x \in X$ can be any kind of objects
  - Images, text, audio, sequence of amino acids, . . .
  - Often: Just vector of numbers

- Output discrete or a real number or complex structure
  - **Classification**: Output is prediction of a class (class or probability)
  - **Multiclass-Classification:** Choose between more than 2 classes
  - **Regression**: Output is a number
  - **Structured Prediction:** Output is „more complex“



- **Learning:**
  - Estimate parameters $\mathbf{w}$ from training data $\{(x\ ,\ y\ )\}$
  - Hyper-parameters are parameters that are set by the user that determine the learning procedure (not learned)

- **Inference:** Make novel predictions: $y = f_{\mathbf{w}}(x)$

Hyperparameters

- Parameters
  - Are fitted automatically using the training data
- Hyperparameters
  - Define the structure and cost functions of the model
  - Are set (fixed) by the machine learning engineer

Parameters vs. Hyperparameters: Example

Linear Regression: General Solution

Assume we have n instances of p input variables $X_1, \ldots, X_p$ (independent variables, regressors, predictors, features) and one output variable Y (dependent variable, response, target).

The i^{th} instance is given by a row vector $(y_i, x_{i1}, \ldots, x_{ip})$. We assume for all i that y_i linearly depends on the x_{ij} values plus some error e_i:

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + e_i$$

Parameters

This set of equations can be written in matrix / vector form as

$$\underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix}}_{\mathbf{y}} = \underbrace{\begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{i1} & \cdots & x_{ip} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}}_{\mathbf{X}} \underbrace{\begin{pmatrix} \beta_0 \\ \vdots \\ \beta_p \end{pmatrix}}_{\boldsymbol{\beta}} + \underbrace{\begin{pmatrix} e_1 \\ \vdots \\ e_i \\ \vdots \\ e_n \end{pmatrix}}_{\mathbf{e}}$$

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica

Pictures: https://miro.medium.com/max/1000/1*Hh53mOF4Xy4eORjLiIKOwA.png

$$\text{where } \widehat{\text{Petal.Length}} = \underbrace{1.29}_{\hat\beta_1} \text{ Sepal.width} + \underbrace{1.2}_{\hat\beta_0}$$

Regularization: Lasso

Obviously, we can imagine other magnitude penalties besides the squared penalty. A different shrinkage strategy, the least absolute shrinkage and selection operator (Lasso), uses absolute values.

While the Ridge objective is

Hyperparameters

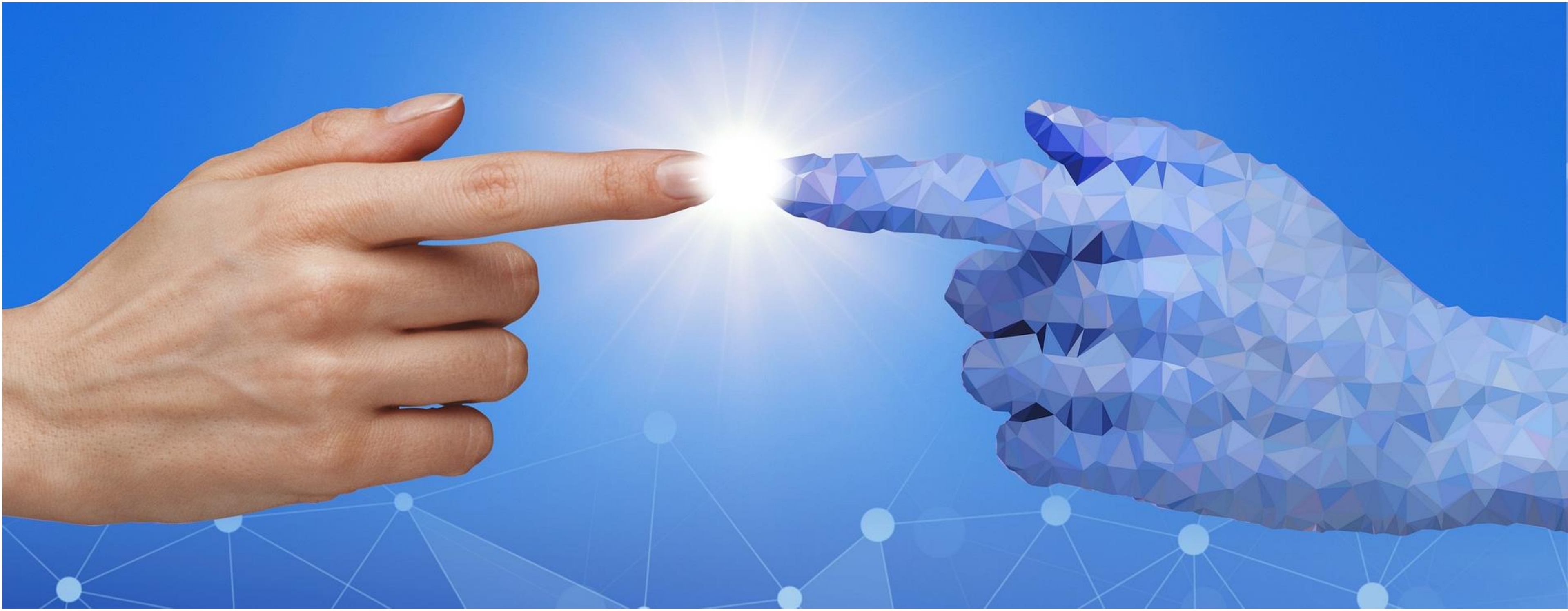
$$J^{\text{Ridge}}(\boldsymbol{\beta}) = J^{\text{OLS}}(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} \beta_j^2,$$

the Lasso objective uses

$$J^{\text{Lasso}}(\boldsymbol{\beta}) = J^{\text{OLS}}(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|.$$

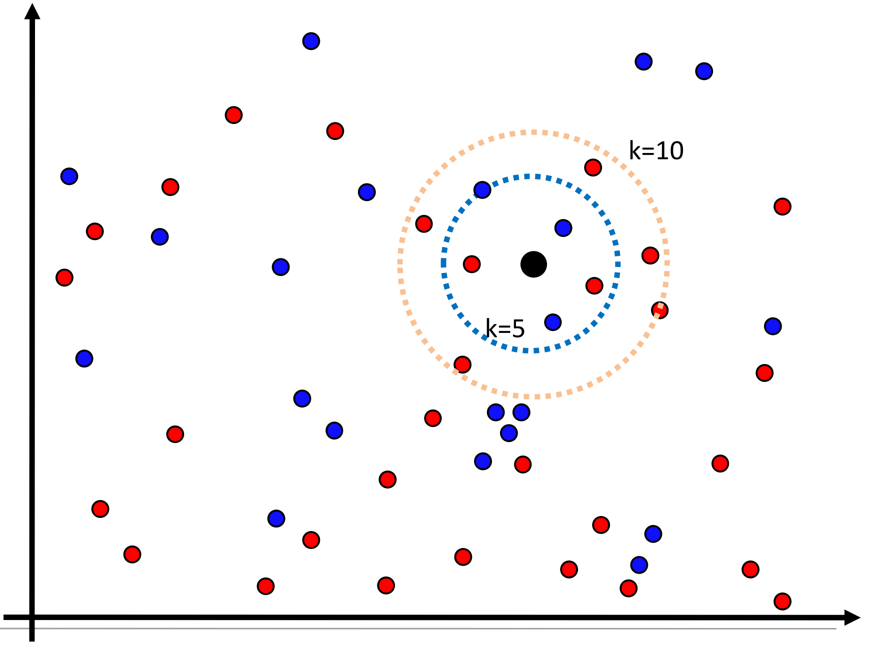
While the Ridge objective puts *equal weight* on highly correlated parameters (thus reducing the degrees of freedom by tying them together), the Lasso objective tries to reduce degrees of freedom by driving correlated parameters towards zero, retaining just one representative.

Prediction Methods



K-Nearest-Neighbor

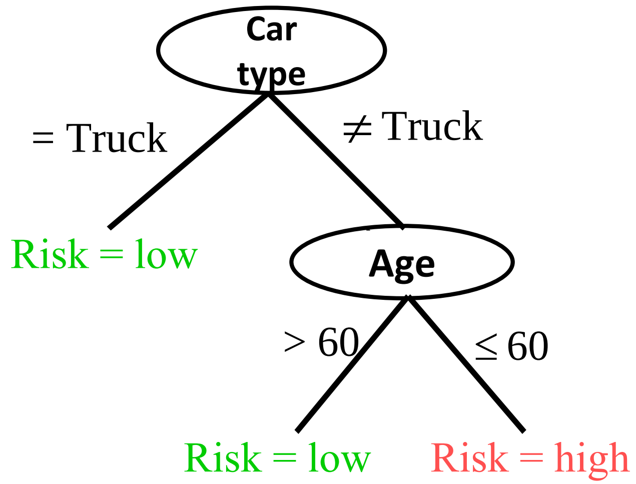
- Define distances between data instances
- Specify a value *k*
- To classify a new data instance:
  - Search the k most similar instances (k-nearest-neighbors)
  - Select the majority class among those neighbors (option: weight by distance



Decision Tree

Toy example: car insurance

ID	Age	car type	Risk
1	23	Family	high
2	17	Sport	high
3	43	Sport	high
4	68	Family	low
5	32	Truck	low



- Decision trees find *explicit* knowledge
- Decision trees are easy to interpret for most users

Construction of Decision Trees

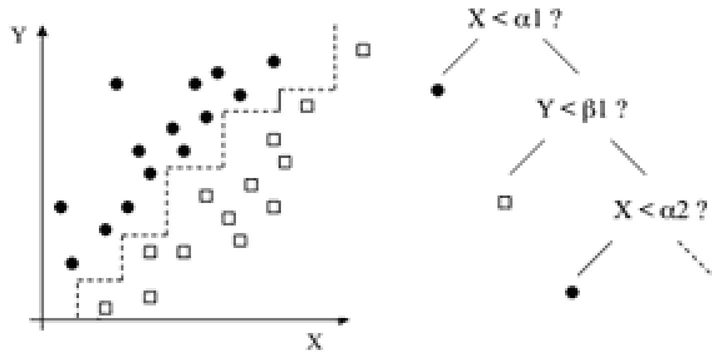
- Base algorithm:
  - At the start: All (training) instances belong to the root
  - Select an attribute *(split strategy)*
  - Partition the training dataset using the split attribute
    I.e., each inner node corresponds to a subset of the data with certain properties
  - Continue recursively for all partitions

$\Rightarrow$Local optimizing algorithms
$\Rightarrow$Finds not always the optimal (=smallest) tree

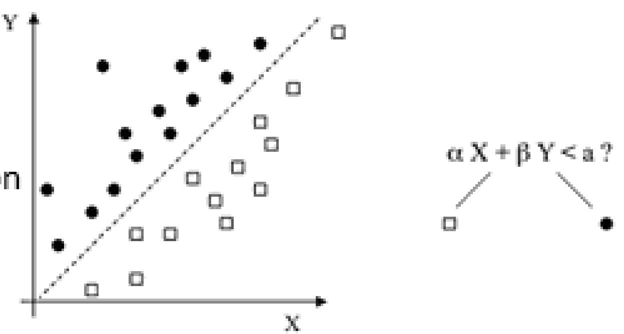
- Conditions for terminations
  - All instances belong to the same class
  - There are no examples in this subset
  - No more split attributes that improve the model

Classification Boundaries

decision tree

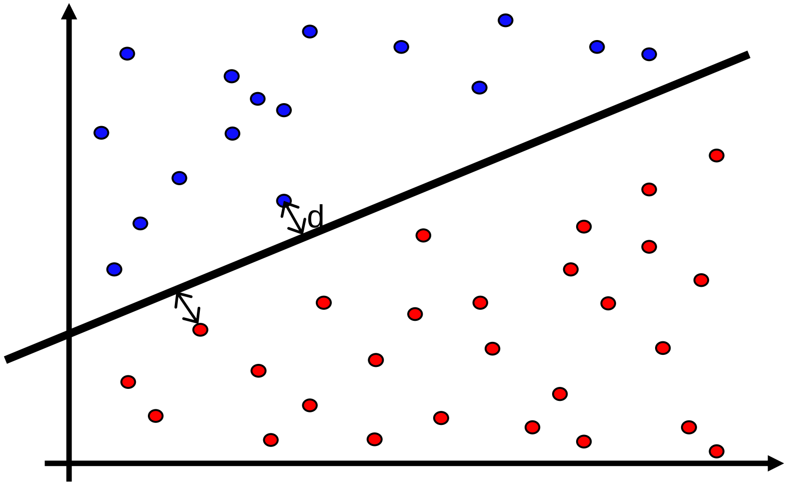


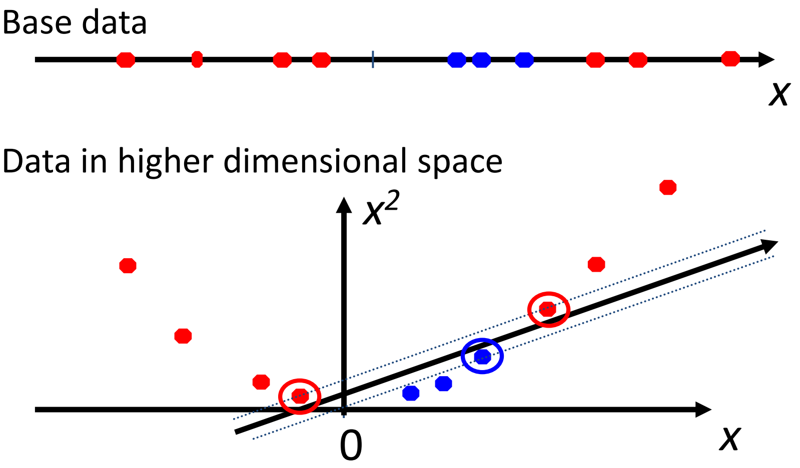
linear regression



SVM

- Project all instance into an n-dimensional space
- Try to find a *hyperplane* that
  - Separates positives and negative instance
  - Maximizes the distance to the closest instances (support vectors)
- Optimization problem
- Extensions:
  - Implicit transformation to higher dimensional space (kernels)
  - Additional error term C for instances “on the wrong side”

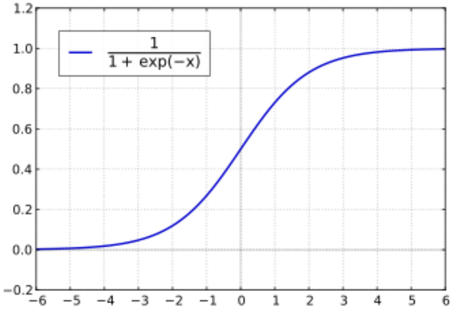




Optimization-based machine learning

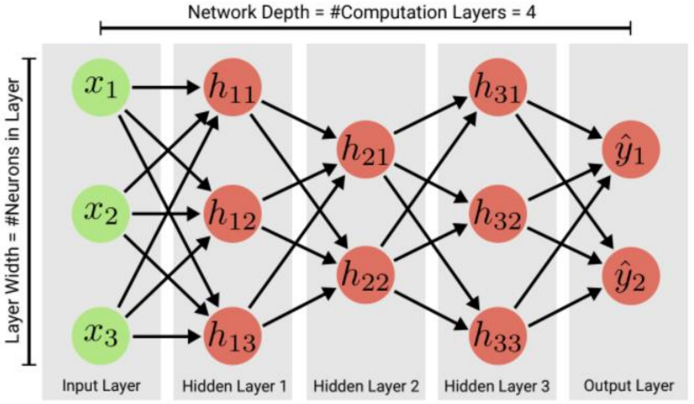
- Many (most?) classification and regression models today are optimization-based
- Our model is trying to optimize a function
- The „skeleton“ of the function is fixed
- The free parameters are **fitted** to the training data to minimize a cost function (= loss)

Linear/Logistic Regression



$$\hat{y}_i = \frac{1}{1+e^{-(w_0+w_1 \cdot x_{i,1}+\ldots+w_k \cdot x_{i,k})}}$$

Neural Networks

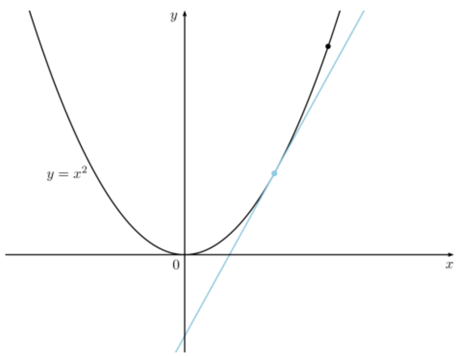


Gradient Descent

Gradient Descent

Gradient

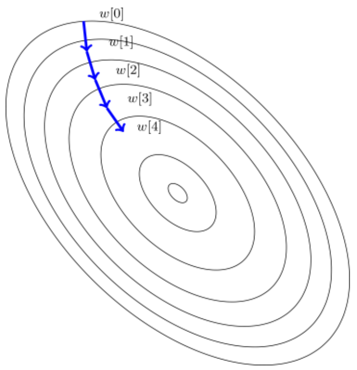
= The vector of all partial derivatives of a function



$$\nabla_{\boldsymbol{x}} f = \text{grad} f = \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} & \frac{\partial f(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f(\boldsymbol{x})}{\partial x_n} \end{bmatrix}$$

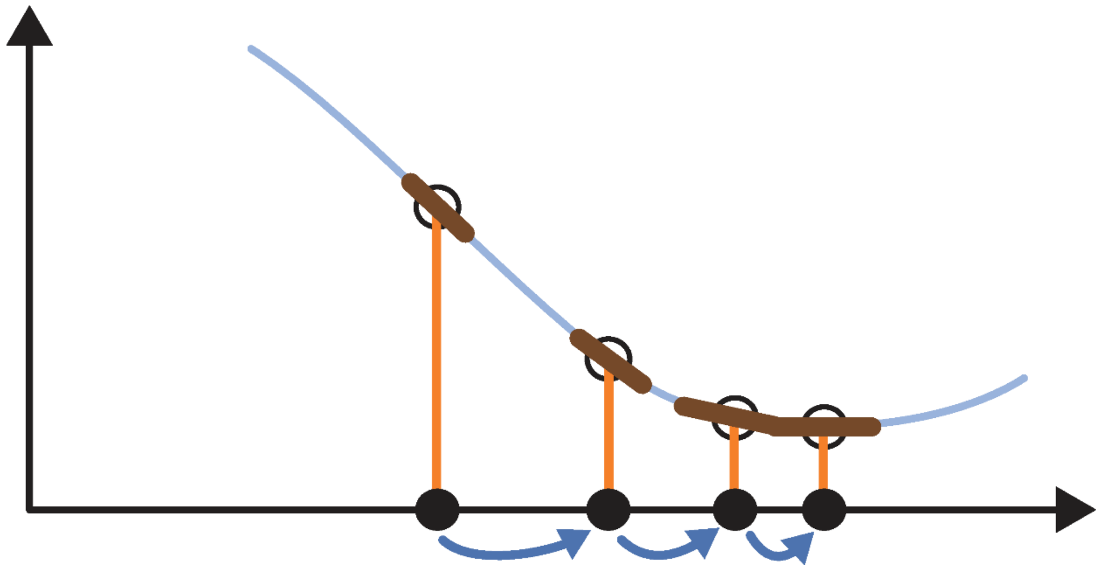
Descent

= Finding a way towards the minimum of the function

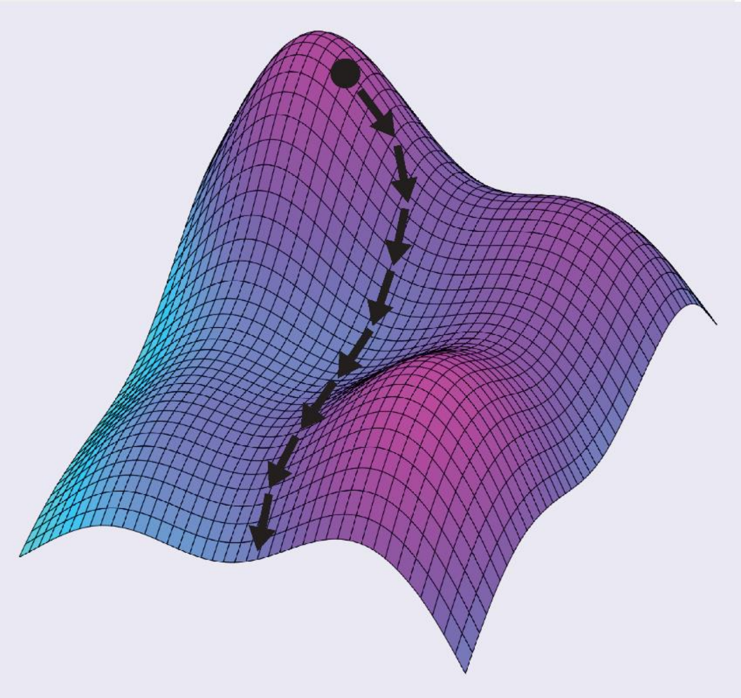


Gradient Descent

- Gradient descent is a standard solution for (m)any optimization problem
- Very often used for fitting parameters to data
- Improve solution step-by-step, reducing the error in each step:
- Gradient gives direction of steepest *ascent*
- A simple way to minimize a (differentiable) function $f(x)$:
  1. Compute derivative function ∇f
  2. Start at some (random) point y and evaluate $\nabla f(y)$
  3. Make a step in the reverse direction of the gradient: $y = y - \alpha \nabla f(y)$
  4. Repeat 2-3 until converged



A. Glassner: Deep Learning – A visual approach, Fig 5-14

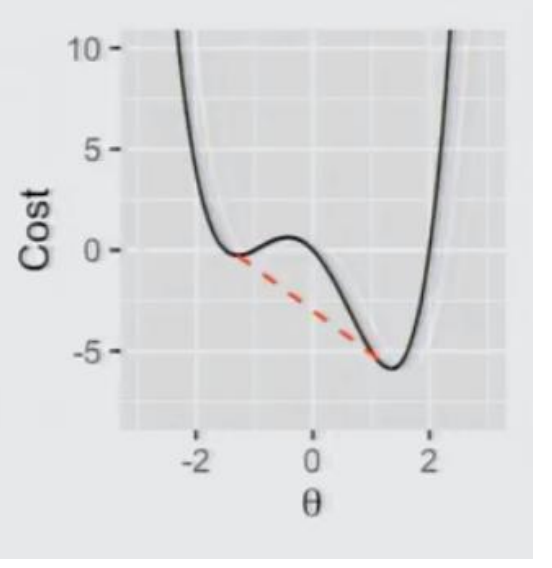


- Here with the error depending on 2 parameter.
- In practice the error depends on k parameters, thus would have to be visualized in k+1 dimensional space!

Convergence

- If α is small enough, we will reduce the error
- But do we end up in a global minimum?

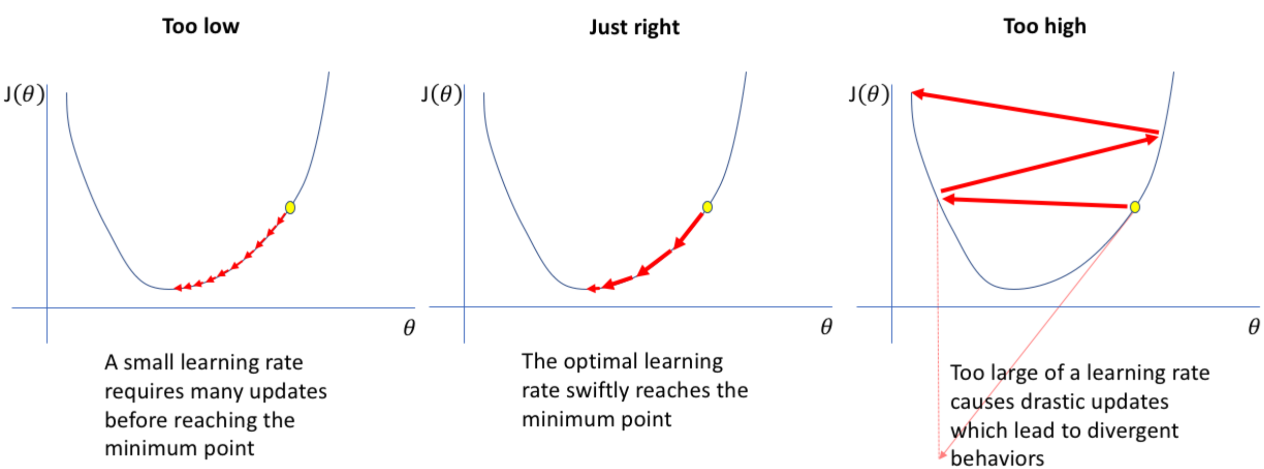
- In general: No
  - Gradient decent finds any minimum
  - Not necessarily the global on



- For Convex Functions,
  low „enough“ learning rate: yes
- Convex function: you can draw a line between any points and the line will pass above the graph

alpha?

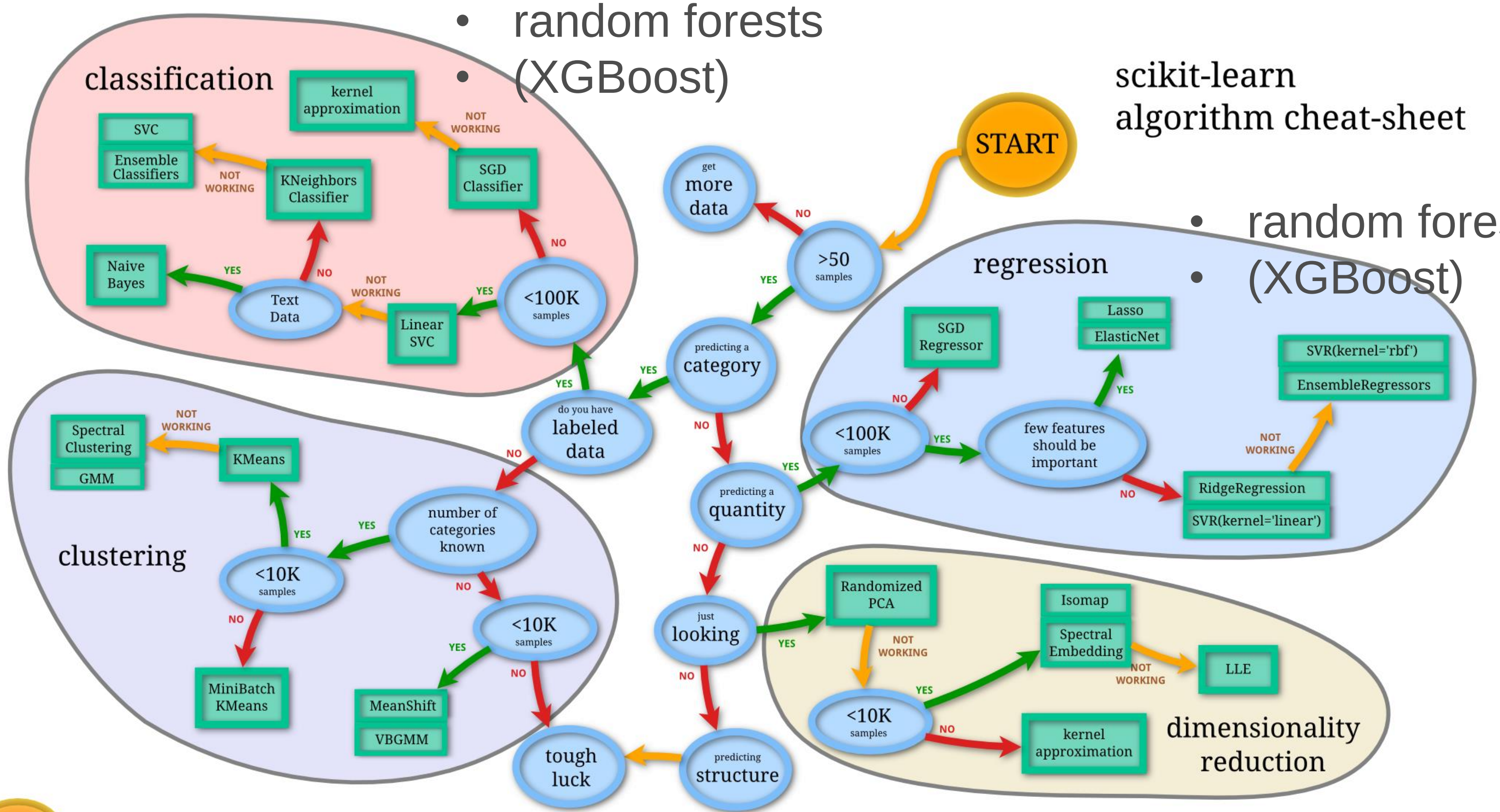
- α is also called the **learning rate**
- α too small:
  - very slow convergence
  - Will get stuck in the tiniest local minimum
- α too large:
  - Will “overshoot“
  - Might not converge at all



Practical example

- Fit a model

Overview of Machine Learning by Scikit-Learn



- random forests
- (XGBoost)

- random forests
- (XGBoost)

- t-SNE
- (UMAP)

Back

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