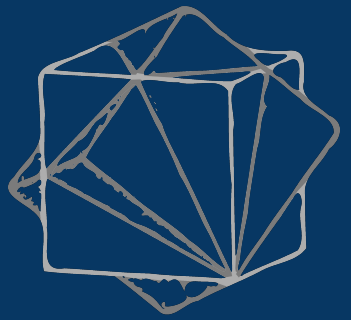
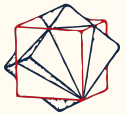


IntroML

14 July 2023

IntroML

Welcome



The ML ⇌Science Colaboratory

Plan for the day

Agenda

9:10	Round of introductions
Morning	**Core ML Concepts** **ML intro ⁄ learning paradigms ⁄ neural networks**
~12:30	**LUNCH BREAK**
Afternoon	**Machine Learning in Practice** **project pipeline: data, models & evaluation ⁄ ML in Science**
	Hands-on: Define your ML project **question ⁄ learning task ⁄ dataset ⁄ evaluation**
16:30	Closing – feedback survey

Practical aspects

- Pauses 2x15’ + 60’ lunch break
- Lunch options include Kasino and Crona cafe
- Internet → eduroam, share laptop if needed
- Toilet → right side of the exit

?! Questions welcome at any time

Slides, links, notebook and feedback at https://mlcolab.org/public-events/introml-workshop-july-2023

Round of introductions

Tübingen is a great place to form a community of practice and mutual support around ML and science.

Let’s get to know each other! We can share

- Name & where you work
- Expectations from ML related to your research (1 sentence)

The ML ⇌ Science Colaboratory

At the Cluster ML: new perspectives in science

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

«Establish machine learning across disciplines at the University of Tübingen»

via ① cooperations — ② training — ③ scientific ML software

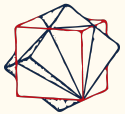


Emily Gabaldon | Hanqi Zhou | Álvaro Tejero | Seth Axen | Vladimir Starostin

- **Part of the** Cluster ML in Science
- Scientists **like you**
  Astro, **atmospheric**, & **quantum physics**, environmental sciences, **neuroscience**, **genomics**, urban systems, int'l policy, **structural bio**…
- Let's work together!

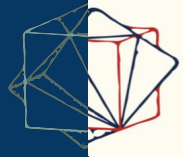
@mlcolab @mlcolab@fediscience.org

Core Concepts in Machine Learning



The ML ⇌ Science Colaboratory

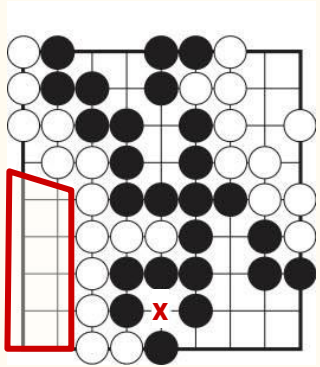
Core Concepts in Machine Learning



What is ML?

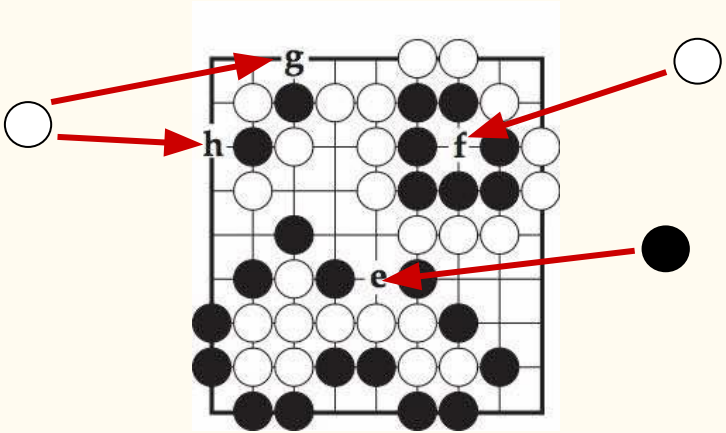
What problems does it solve?

Games and the nature of intelligence

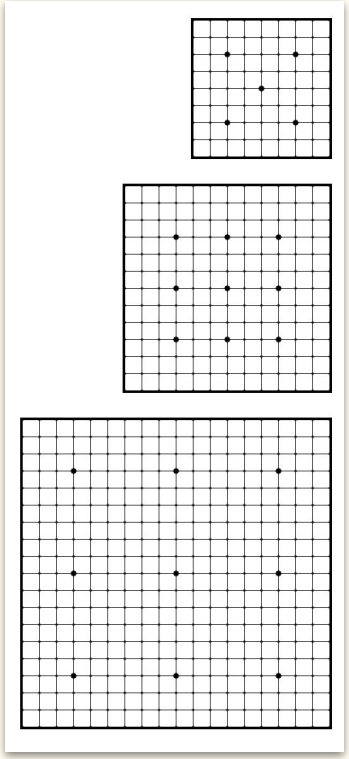


- form territories by surrounding empty areas
- capture by completely surrounding (*prisoners*) **x**

Score grows with crossing and prisoners

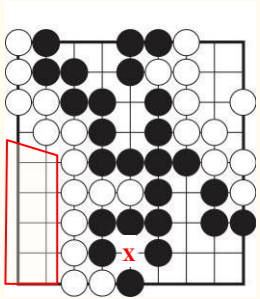


Real go is 19x19

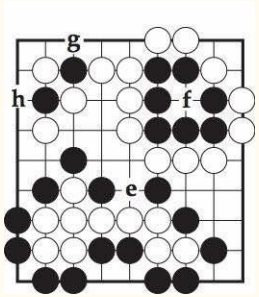


British Go association

Games and the nature of intelligence



- form territories by surrounding empty areas
- capture by completely surrounding (*prisoners*) **x**

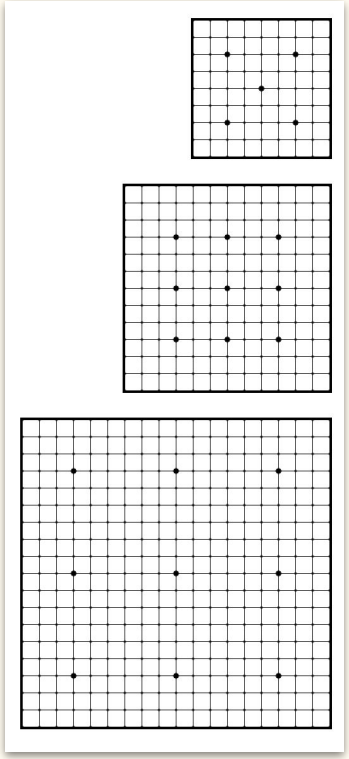


○ at **f** → **7** ● captured

○ at **g** or **h** → 1 ● each

● at **e** → 10 ○ and 10 crossings !

Points = crossings + prisoners



Real go is 19x19

British Go association

A surprising move



See also AlphaGo - The Movie

Definition of Machine Learning

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Learning

Improving with experience at some task

- Improve over **task** T,
- with respect to **performance** measure M (**metric**)
- based on **experience** D (**data**)

e.g., Learn to play Go

- T: playing Go,
- M: final score
- D: many games between humans

AlphaGo: Deep-Learning powered (ML) Monte-Carlo Tree *Search*.

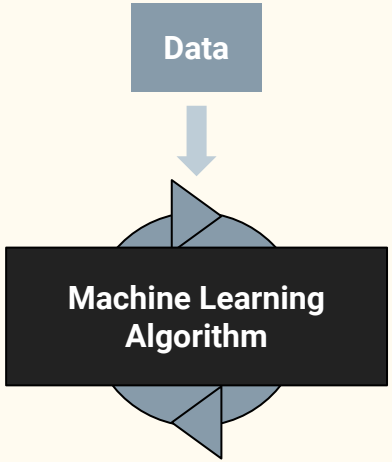
***Machine* learning**: the study of **algorithms** that allow computer programs to automatically improve **on a specific task** through **experience**

Mitchell, 1997

ML: Training and Use phases

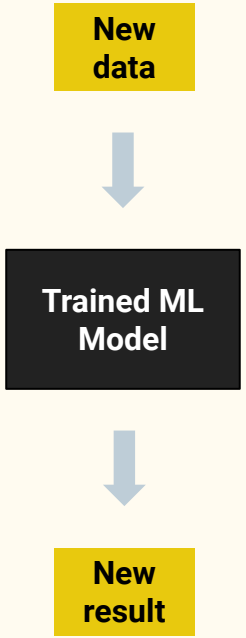
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Preparation: training



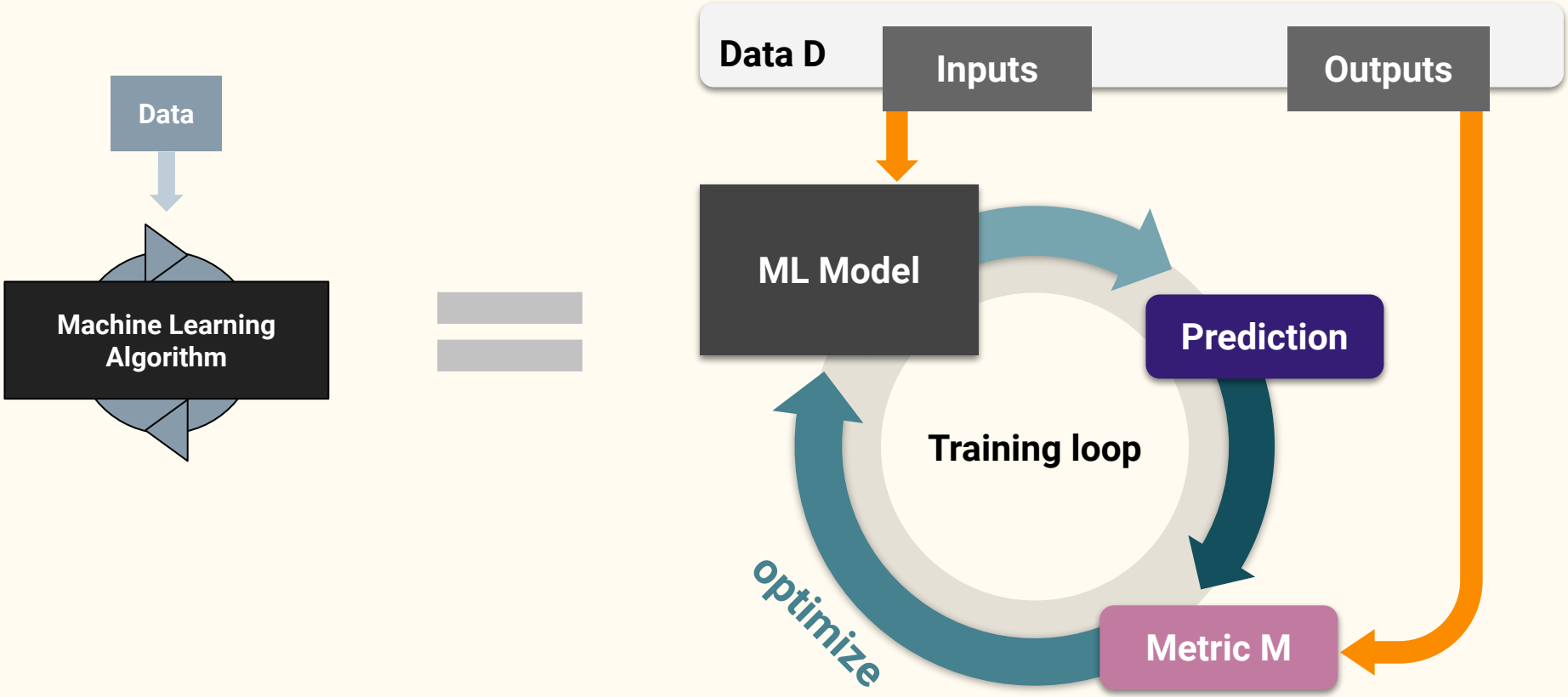


Use*



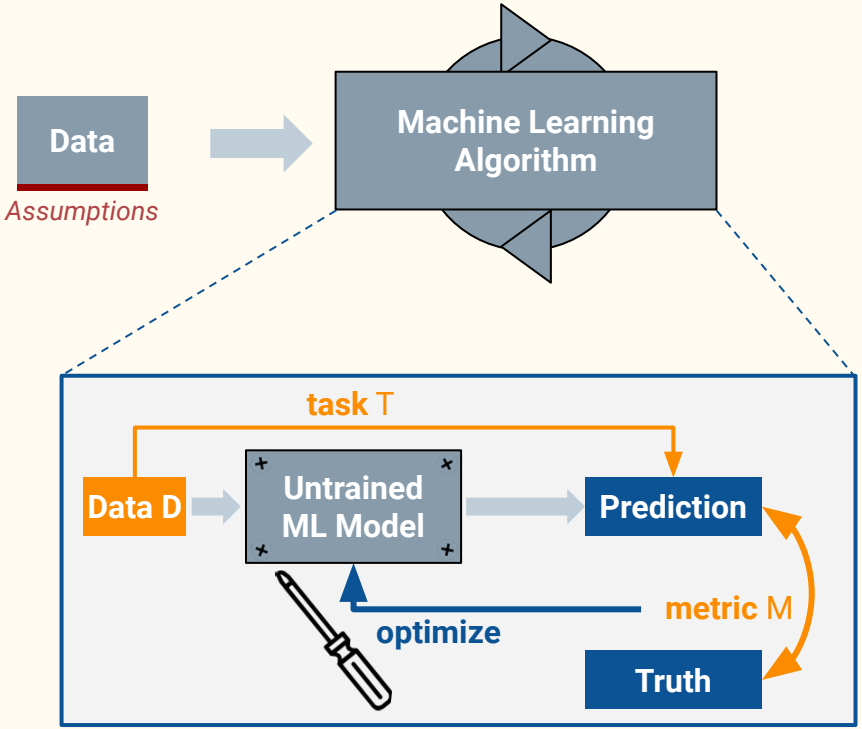
* also called prediction or “inference”

Inside the training black box

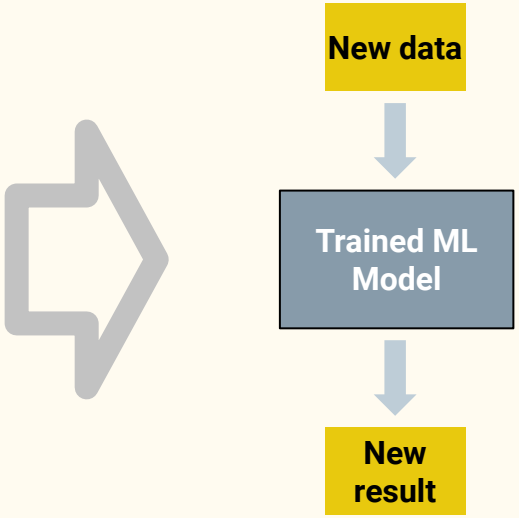


Inside the black box

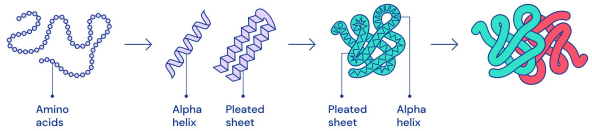
Training phase

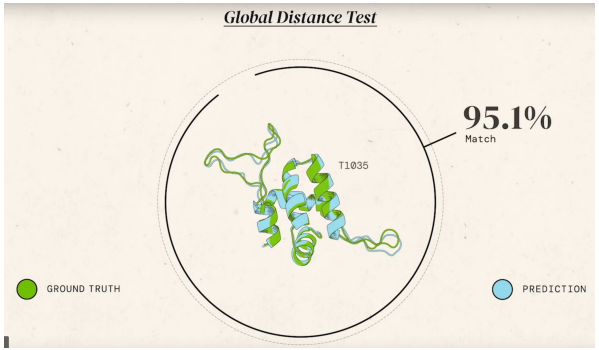


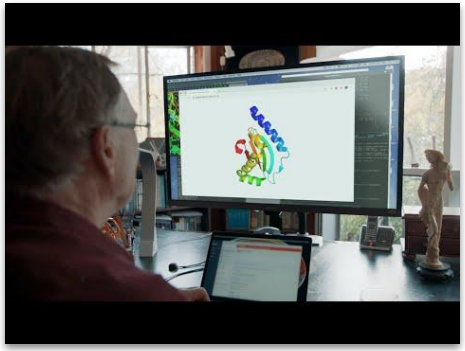
Use, or "**inference**" phase



From proteins to planets, ML in science



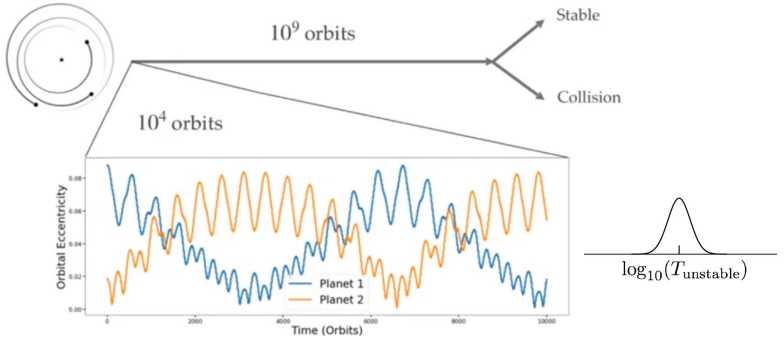




Protein structure prediction with transformers

Senior et al. 2020, Jumper et al. 2021

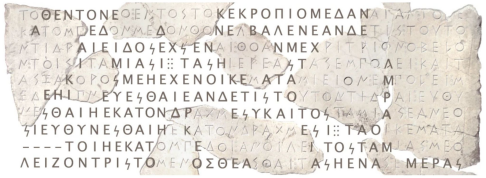




Planetary stability with Bayesian neural networks

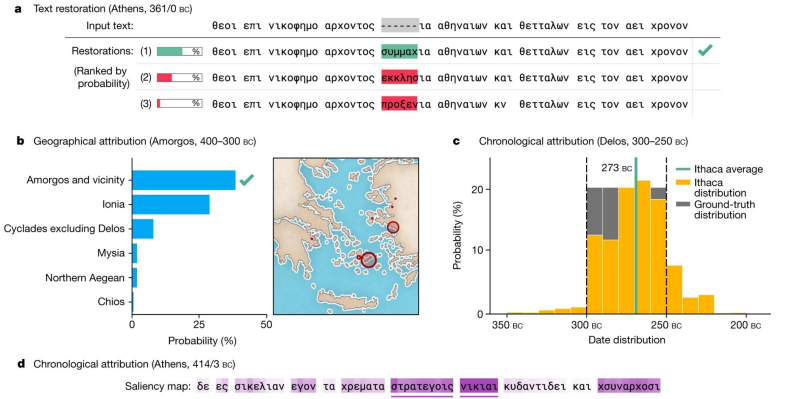
Cranmer, Tamayo et al. 2021

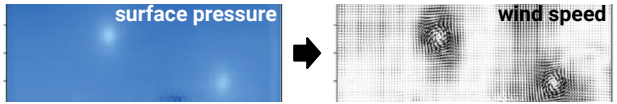
Ancient greeks and seasonal hurricanes



Text restoration and geochronological attribution with data augmentation

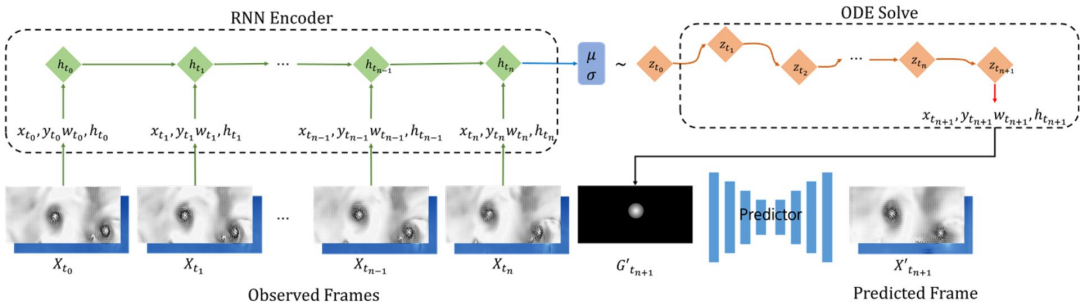
Assael, Sommerschield et al. 2022



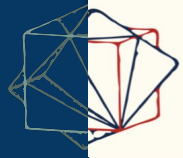


Hurricane path forecast & speed field with nODEs & GANs

Park, Kim et al. 2020



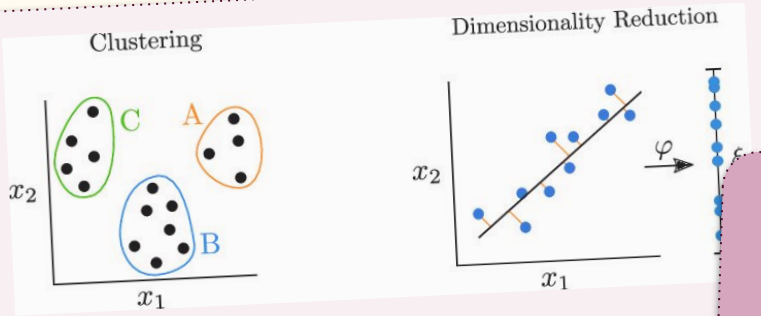
Core Concepts in Machine Learning



Learning paradigms

Main Learning Paradigms

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

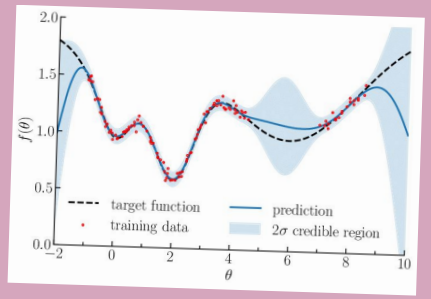


Unsupervised learning

Exploring structure

Supervised learning

Teaching to label



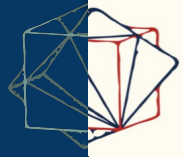
Reinforcement learning

Harnessing interaction



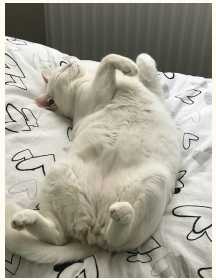
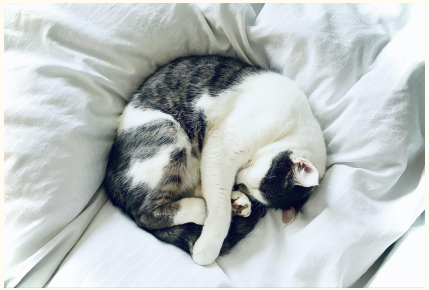
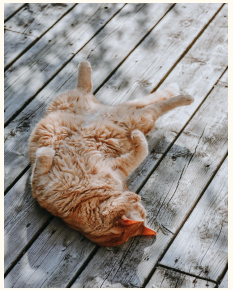
Figures DeepMind 2019, Leclerc 2018, jovian.ai 2021

Core Concepts in Machine Learning



Unsupervised learning

An example dataset

Instance

ID	R01	G01	B01…
1	215	198	180 …
2	61	61	51 …
3	219	227	227 …
4	47	43	37 …

Feature

Photo by Sarah Ball on Unsplash

Photo by Gökhan Konyalı on Unsplash

Photo by Ben Wicks on Unsplash

Photo by Jacalyn Beales on Unsplash

Finding structure in the data



Photo by Sarah Ball on Unsplash

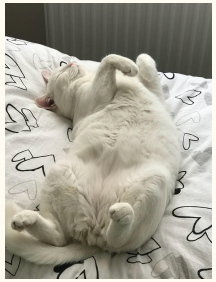


Photo by Gökhan Konyalı on Unsplash

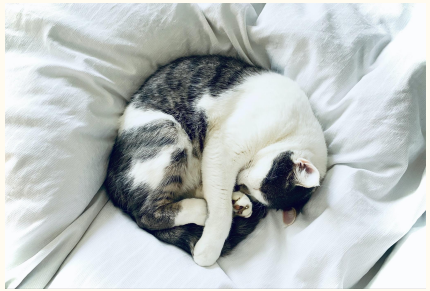


Photo by Ben Wicks on Unsplash

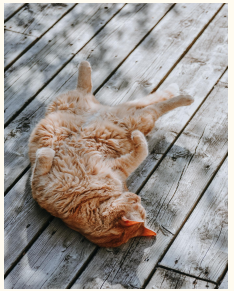
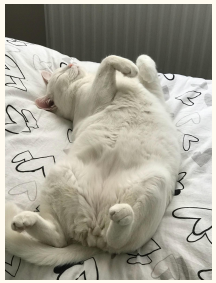


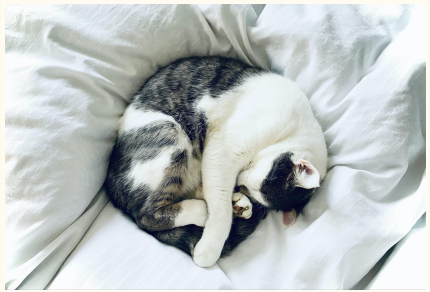
Photo by Jacalyn Beales on Unsplash

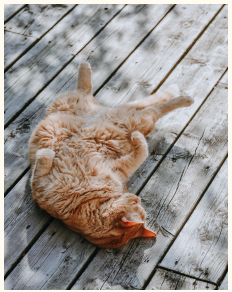
Task: group the images into any numbers of groups that you like

Finding structure in the data



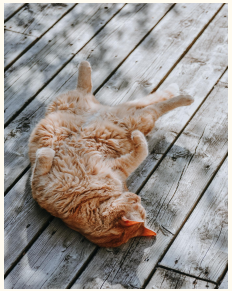
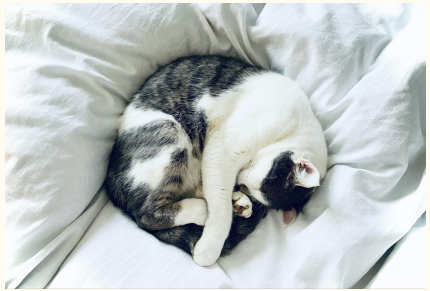
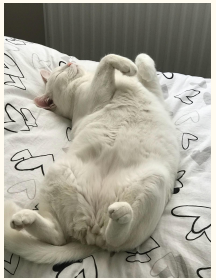






For example: 4 groups based on the breed of cat

Finding structure in the data

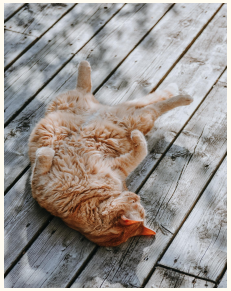





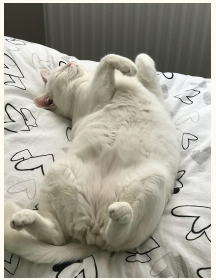
For example: 2 groups based on colors (orange or not orange)

Finding structure in the data









For example: 2 groups based on gesture

Unsupervised Learning

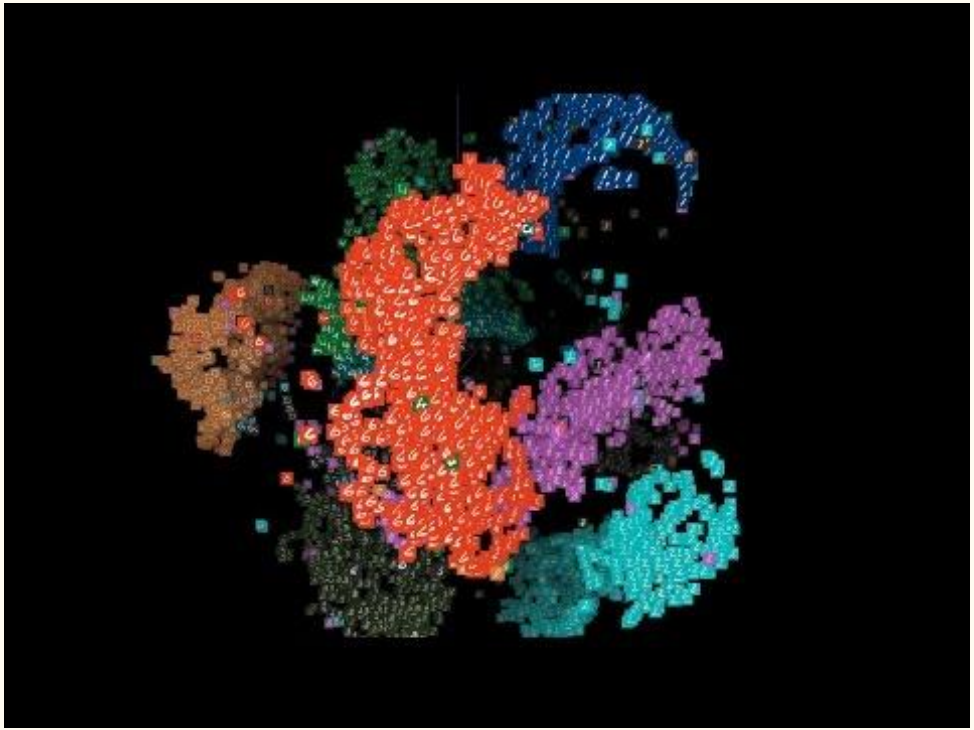
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Goals

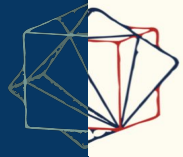
- Structure discovery
- Dimensionality reduction
- Community detection

Requirements

- Data (unlabeled)
- Some notion of similarity of data points



Core Concepts in Machine Learning



Supervised learning

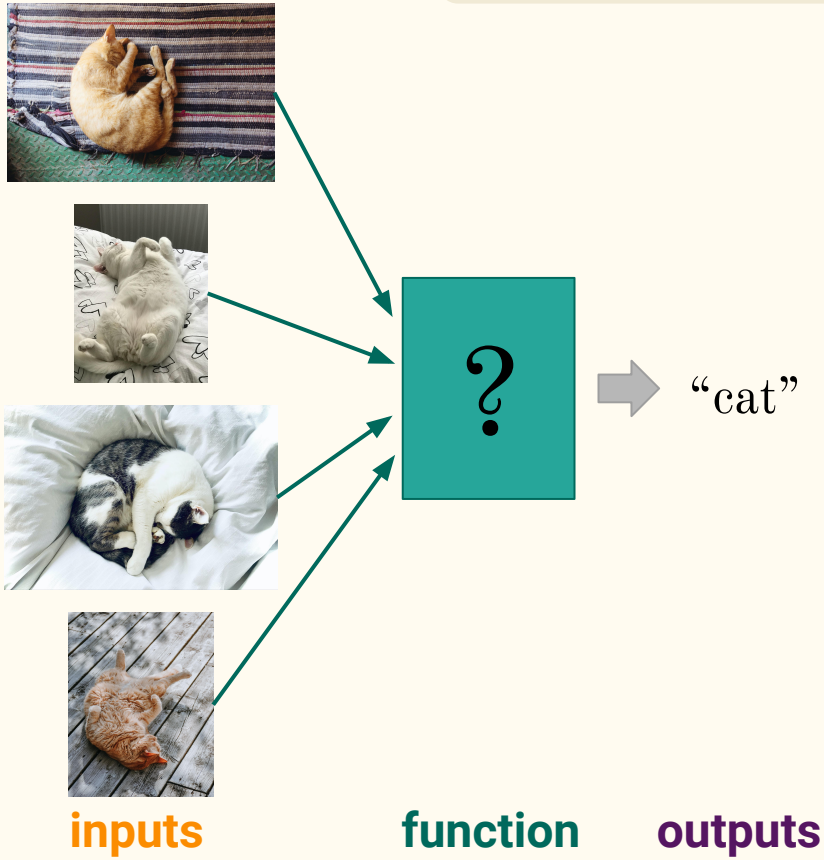
Supervised Learning

Goal
Learn a **function** to map **inputs** to **outputs**

Requirement
Dataset of inputs *and outputs* (“labels”)

Modes

- Classification (category outputs)
- Regression (continuous outputs)



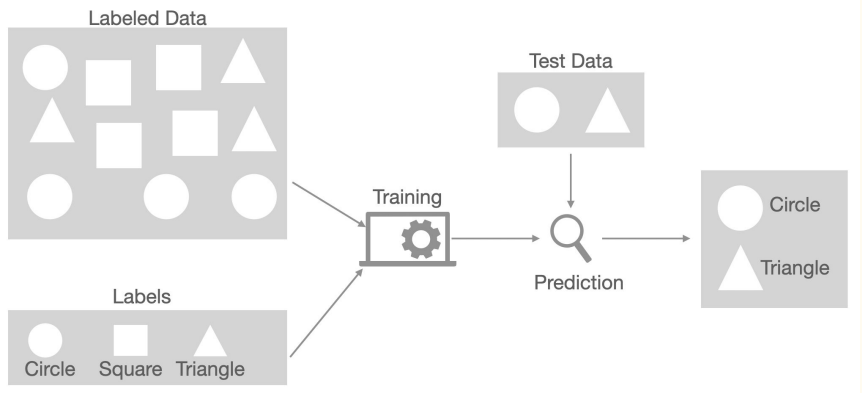
Supervised Learning

Goal
Learn a function to map inputs to outputs

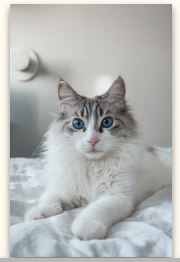
Requirement
Dataset of inputs *and outputs* (“labels”)

Modes

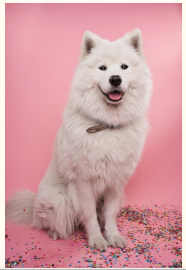
- Classification (category outputs)
- Regression (continuous outputs)



Supervised Learning: Classification



Cat



Dog



Cat



Dog

Category Label

ID	R01	G01	B01…	Type
1	184	187	187 …	Cat
2	218	157	164 …	Dog
3	61	61	51 …	Cat
4	236	236	236 …	Dog

Photo by Esteban Chinchilla on Unsplash Photo by Peri Stojnic on Unsplash Photo by Gökhan Konyalı on Unsplash Photo by FLOUFFY on Unsplash

Supervised Learning: Regression



47



0



2



-33

Continuous Label

ID	R01	G01	B01…	Head pose (degree)
1	116	98	81…	47
2	157	146	164 …	0
3	23	22	21…	2
4	137	130	129 …	-33

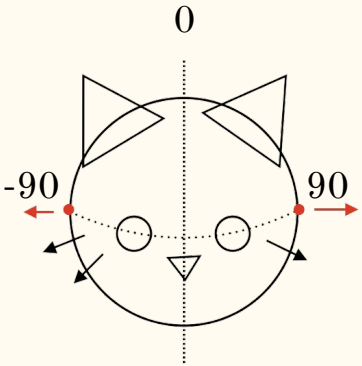
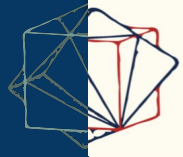


Photo by: cat-breeds-classifier

Core Concepts in Machine Learning



Reinforcement learning

Reinforcement Learning

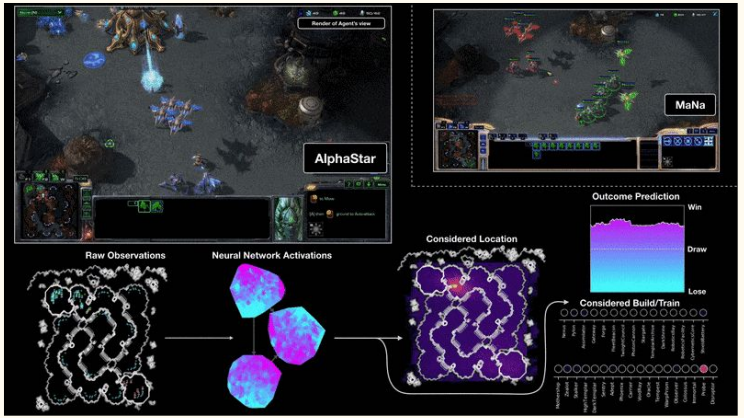
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Goal

Train an agent to interact successfully with some environment

Requirements

- Agent interacting with an environment (real or simulated)
- Reward mechanism for actions taken



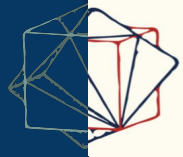
Reinforcement Learning



Reinforcement Learning



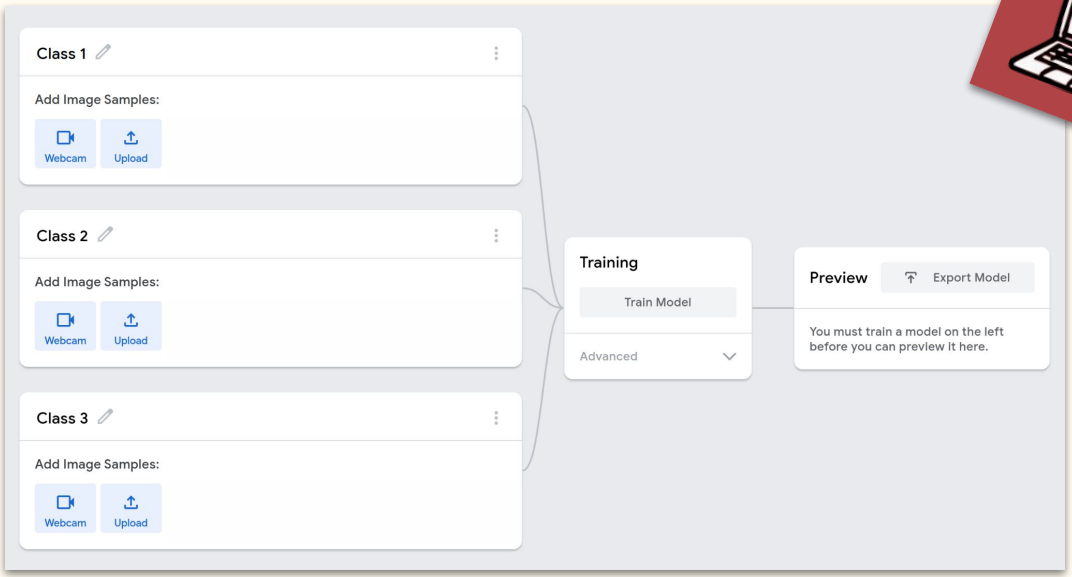
Core Concepts in Machine Learning



Supervised learning

One step at a time

Let's train a supervised classifier





Navigate to:
https://teachablemachine.withgoogle.com/train/image

We've trained a supervised classifier



- Collect data
  image frames + labels
- Train*
  neural network
  * fine-tune
- Infer or predict
  class of new image

ML brings the senses to computers

HOW? Supervised Learning Tutorial

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



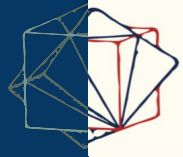
Supervised learning: One step at a time

This notebook was created for the ML ⇌ Science Colaboratory's workshop Introduction to Machine Learning.

We here focus on supervised learning. As we progress from a simple linear regression to a first example of a neural network, we introduce the basic concepts needed to understand machine learning.

Interactive notebook downloadable at https://mlcolab.org/resources/introml-july-2023-workshop-materials

Core Concepts in Machine Learning



Neural Networks

Layer upon layer upon layer

Building a neural network - terminology

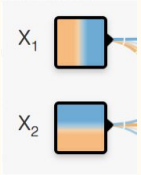
playground.tensorflow.org

DATA

Inputs: coordinates

X_1 horizontal

X_2 vertical



optional: *(hand-crafted)* **features**

X_1^2, X_2^2, X_1X_2, sin X_1, sin X_2 ← augment input

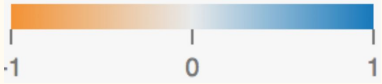
Outputs: Binary classes **orange (-1)** or **blue (+1)** → **TASK: CLASSIFICATION**

MODEL

Hidden layers = number of nestings in the NN; *here: fully connected* } **architecture**

Neurons = units in each layer

Model output: f(X_1, X_2, *features*(X_1, X_2))



Building a neural network - playground

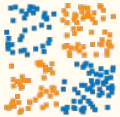
- **Input data**
  - ‘DATA’ *chooser*: blobs, quadrants, concentric & spiral
  - ‘OUTPUT’ *visualizer*
    - Raw features: coordinates of the dots, X_1 and X_2
    - Supervised labels: binary classes
    - Training data; tick checkbox for test data
- **Model**
  - ‘FEATURES’ add *computed features* X_1^2 , X_2^2, X_1X_2, sin X_1, sin X_2 to form an *input layer*
  - ‘HIDDEN LAYERS’ select *architecture* by adding fully connected *hidden layers* of desired size.
- **Output**
  - ‘OUTPUT’: f(X_1, X_2, *features*(X_1, X_2)) as function of X_1, X_2 shown as color level
  - Input dots overlaid for visual comparison.

Neural network playground – Training



blobs

What is the simplest architecture you can find?



xor

What minimal architecture can you find using only X_1 and X_2?

Can you improve using features?





concentric

Does the default setting solve the problem?

Can you simplify the network using features?



spiral

Use 1 layer and features → “feature engineering”

Construct a deep network, but only use X_1 and X_2

Training the neural network

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Training

- → Start: initialize *weights* randomly, (hence *activations* also random)
- □ Learn: blame training loss on specific weights, and adjust them to reduce it
- □ Stop: see when training loss (or test error) plateaus, call it a day

Inference and evaluation

- Check on a separate test set how well we generalize
  - E.g. do all spirals have arms this long? Distance between the arms?

Challenge time!


blobs


concentric


xor


spiral

blobs: find simplest network, using raw features only

concentric: same, then find simplest with engineered features

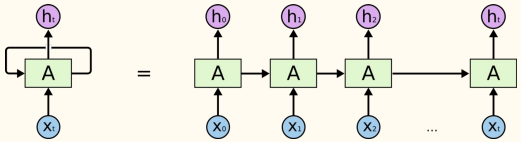
xor: simplest network *with* engineered features, then without

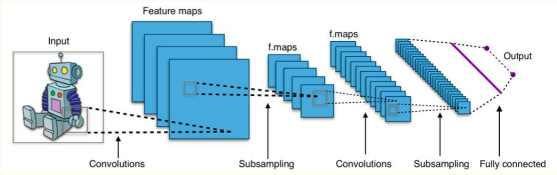
spiral: 1 layer and engineered features vs. deep and raw features only. ReLU

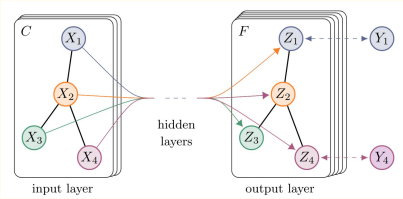
Think about:
How you find/invent features? how/when to trim the network? how good is extrapolation? why oscillations in the loss? why different results every time?

Task-specific architectures

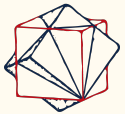
Data	**Time series, sequences (1D)**	**Images (2D)**	**Networks**
e.g.	Text, speech, temperature, …	Microscopy, astronomical shots, density maps, ...	Social networks, molecular graphs, knowledge graphs, …
NN types	RNNs *(recurrent) NNs* Transformers	CNNs – *(convolutional) NNs*	GNNs – *graph NNs*





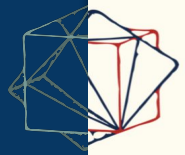


Machine Learning in Practice



The ML ⇌ Science Colaboratory

Machine Learning in Practice



ML development cycle

From data to scientific results

Zooming out: the ML dev cycle

Let's talk about how to…

1. … define the task
2. … obtain data, and prepare it
3. … implement the model
4. … evaluate, interpret and report output

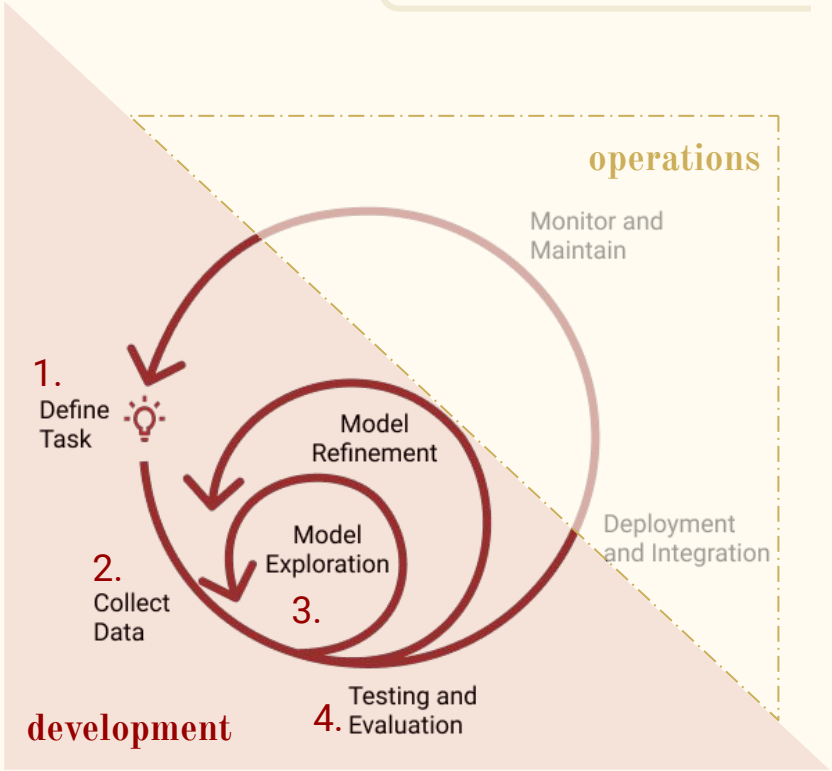


Figure based on Jordan 2018: organizing ML projects

1. Task: Feasibility & Impact of ML Project

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Do you know your data acquisition?

- All relevant conditions covered?
- Cost of additional (labeled) data?

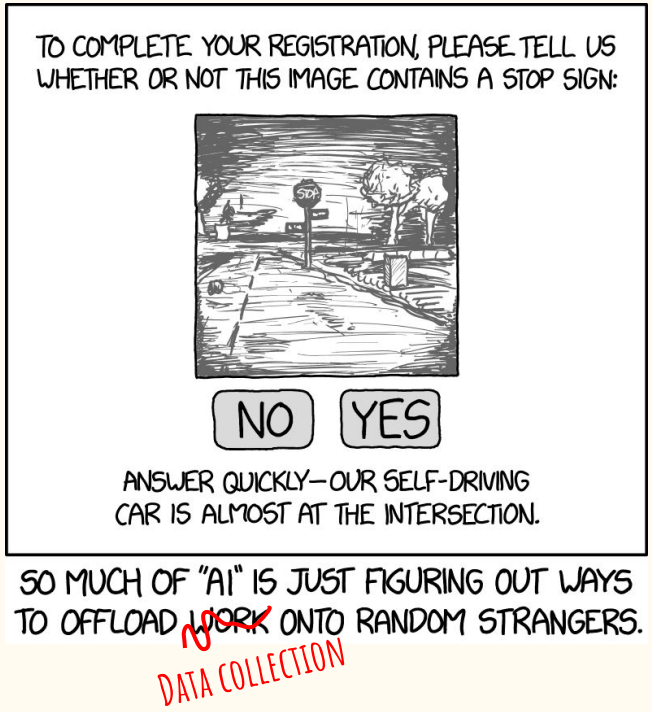
Model requirements, alternatives

- Are there non-ML models known to work well?
- Are there pre-trained models?
- Do you need interpretable results?

How to measure performance?

- Can it be measured? How?
- Can humans do it? How well?
- Are there baselines? (even if non-ML)
- What are the consequence of errors?

2. Data: How to get It



xkcd.com/1897

Strategies for data preparation (YOU!)

- **Complement** with public ancillary data
- **Reduce** via pretrained models
- **Augment** with synthetic or modified data ("Data augmentation")

Data sources

- Your own *sensor* data. Cheap!
- Your own *survey* data
- Public data sources

Data preparation: labeling

- Make it efficient:
  good user interfaces, active learning
- Consider 3rd party services: gamification

2. Data: Common Modalities

- Time series
- Images
- Relational (graph) data
- Tabular
- Video
- Georeferenced measurements
- Spectra
- A mix!

Data modalities provide knowledge that can be exploited, e.g.



The future depends on the past



Close points are usually similar



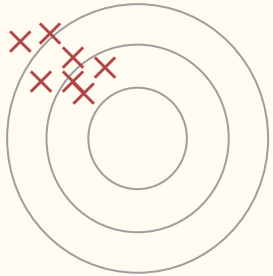
The curvature of the Earth matters

Icons from Icons8

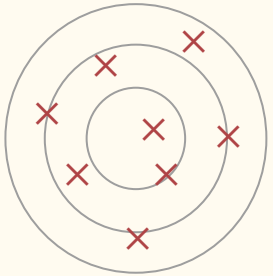
2. Data: Pitfalls

Data might not represent the sampling distribution because of

- Systematic error (bias)
- Noise (i.e. “random” error)
- Sampling bias
- Independence across samples
- Missing data
- Multi-scale phenomena
- Censored data



systematic error



random error

2. Data: Processing

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Validation – verify **data quality**
- Aggregation
- Cleaning
- Transforming
- Reduction
- Summarization, Visualization

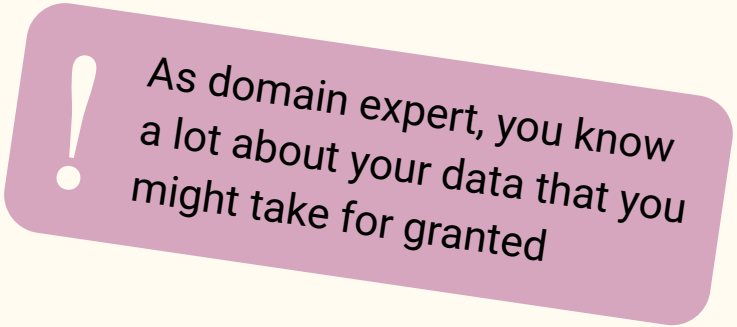


Take home:

- Data processing assumes some model!
- Make data processing programmatic for reproducibility!
- **Never** overwrite **raw** data with **processed** data!

2. Data: Processing

- Validation – verify **data quality**
  - Are data types correct and consistent?
  - Is data plausible / consistent?
  - Do data satisfy ranges and constraints?
  - Reasons for data imbalance?
- Aggregation
- Cleaning
  - Outlier detection/removal
  - Integrating data from different sources
- Transforming
  - Standardizing (shifting and scaling)
  - Smoothing/denoising
  - Imputation of missing data
  - Augmentation
- Reduction
  - Downscaling, feature selection
- Summarization, Visualization



Take home:

- Data processing assumes some model!
- Make data processing programmatic for reproducibility!
- **Never** overwrite **raw** data with **processed** data!

2. Data: Processing

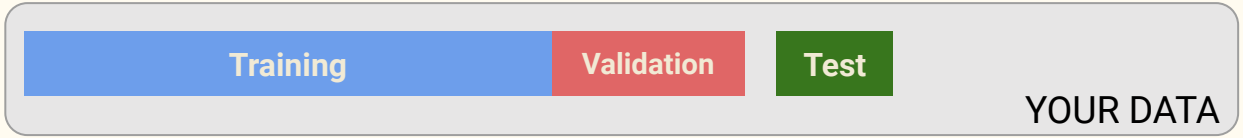
- Every dataset has a story; being able to explain that story is important for ML.

- As a domain expert, you know a lot about the data that you might take for granted

- Examples:
  - pollen counts
  - outlier removal



3. Model: Training, Validating and Testing



- Training data ▮▮▯ to tune **parameters:** minimize loss function

- Validation data* ▯▯▮ to set ***hyper*parameters:** evaluate metrics
  *or *development set*

TRAINING

EVALUATE

- Test data ▯▯▯ ▮🔒 to estimate **generalization performance:** compare metrics <u>after</u> training with **baselines** (random or specific) and toplines (human performance). *Held out* of training.

3. Model: Implementation & Embodiments



- **“Classical ML”**
  - \+ stable, documented, interpretable
  - ○ CPU arrays (numpy, R)

  `R, sklearn, mlj`

- **Probabilistic programming**
  - \+ uncertainty quantification
  - \+ interpretable
  - – CPU mostly, expensive
  - – expert knowledge

  `PyMC, Pyro, Stan, turing`

- **Deep learning**
  - \+ expressive, versatile
  - – hard to interpret
  - ○ Training vs inference
    - ■ Resources & devices
  - ○ GPU

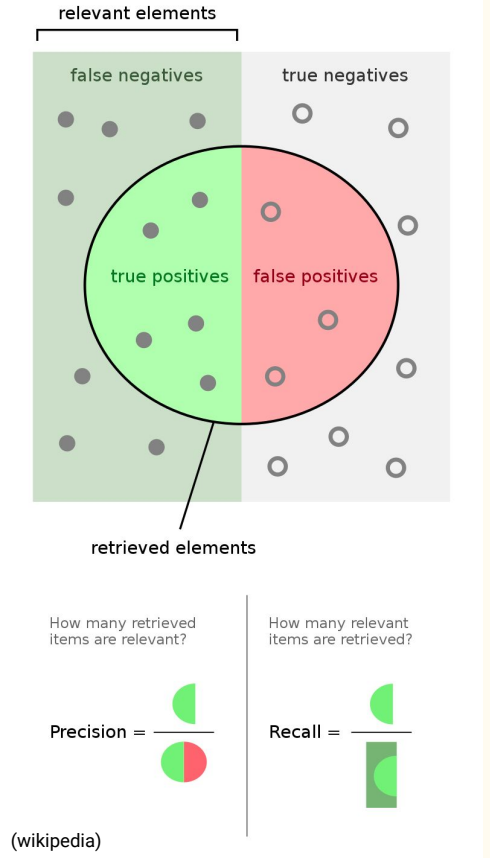
  `PyTorch, TensorFlow, jax, flux`

- **Pretrained**
  - ○ API serving (query only) or
  - ○ download-and-finetune (transfer learning)

  `Huggingface, GPT, ...`

4. Evaluation & Performance Metrics

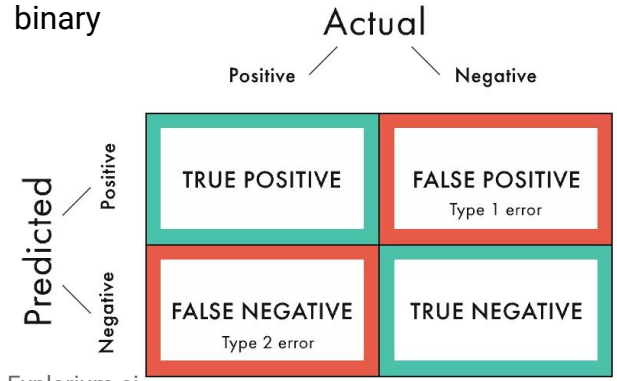
- Use held-out test data 🔒 to report & compare models
  - *Benchmarks* are combinations of task + dataset used to drive development in ML
- You get what you optimize for…
  … but not all you care about we can optimize for
  → need **metrics**, not just loss
- Metrics for:
  - **Regression**: MSE (cf. poly regression), ...
  - **Classification**: precision, recall, F-score, …



4. Evaluation: Classification Metrics

- False alarms (FPs) vs. omissions (FNs)
- ML → decisions, can't ignore **asymmetric losses**, e.g. early diagnostic w. $L_{FP} << L_{FN}$
- Know **your goals (error penalties)** *and* **your data (class imbalance)**
- Derived metrics: precision, recall, …

binary

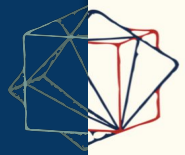


Explorium.ai

5-class

Vowel produced \ Perceived vowel	i	e	a	o	u
i	**15**		1		
e	1		1		
a			**79**	5	
o			4	**15**	3
u				2	**2**

Wikipedia

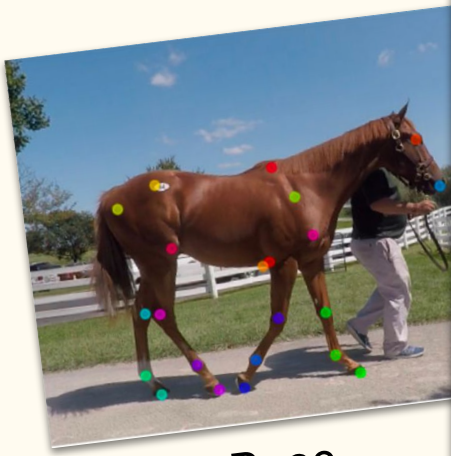


Machine Learning in Science

Learning from language, images, simulations...

Computer Vision (CV)

Data: images or videos



Pose estimation

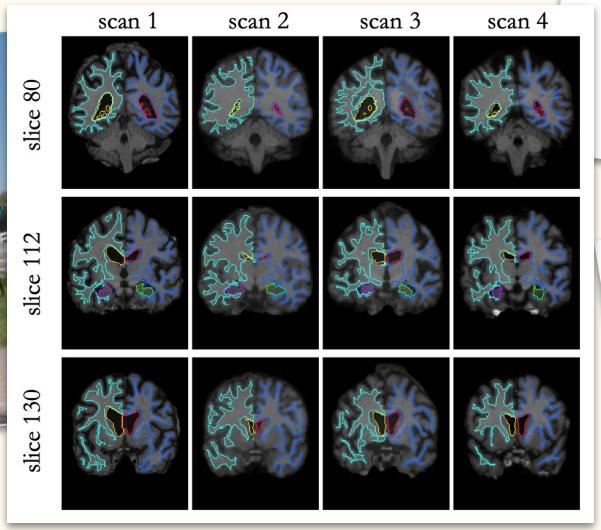


Image segmentation



Object detection

Balakrishnan, Zhao et al. 2019,
Nath, Mathis et al.2018, Resler, Yeshurun et al. 2021

CV: pose estimation without body markers

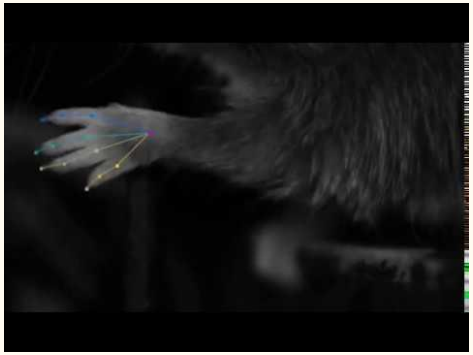
Hypothesis We can detect body parts of multiple animal species from unstaged video without body markers

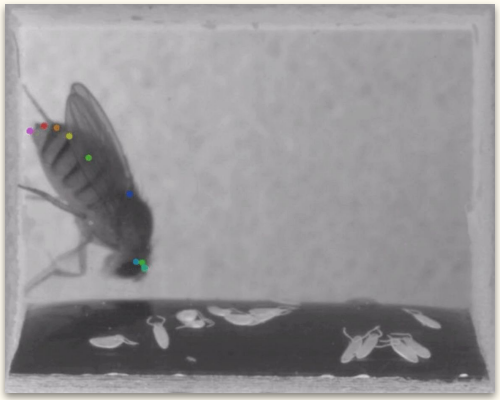
Dataset video frames with *labeled* positions of body parts

Task frame-based pose estimation

Evaluation *RMSE* between predicted and labeled positions

Model *pre-trained* ResNet with fully convolutional upsampling

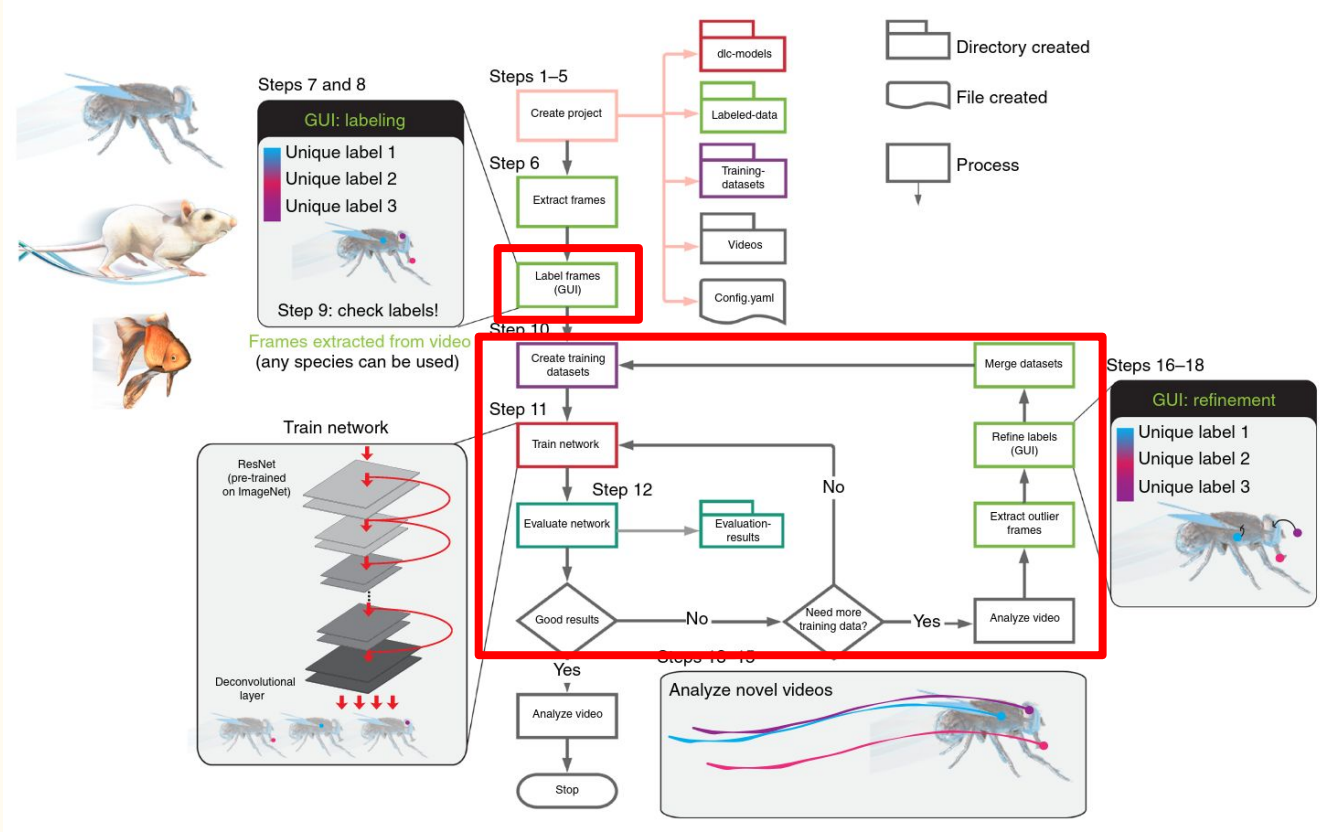




Fruit fly

Nath, Mathis et al.2018

CV: pose estimation without body markers



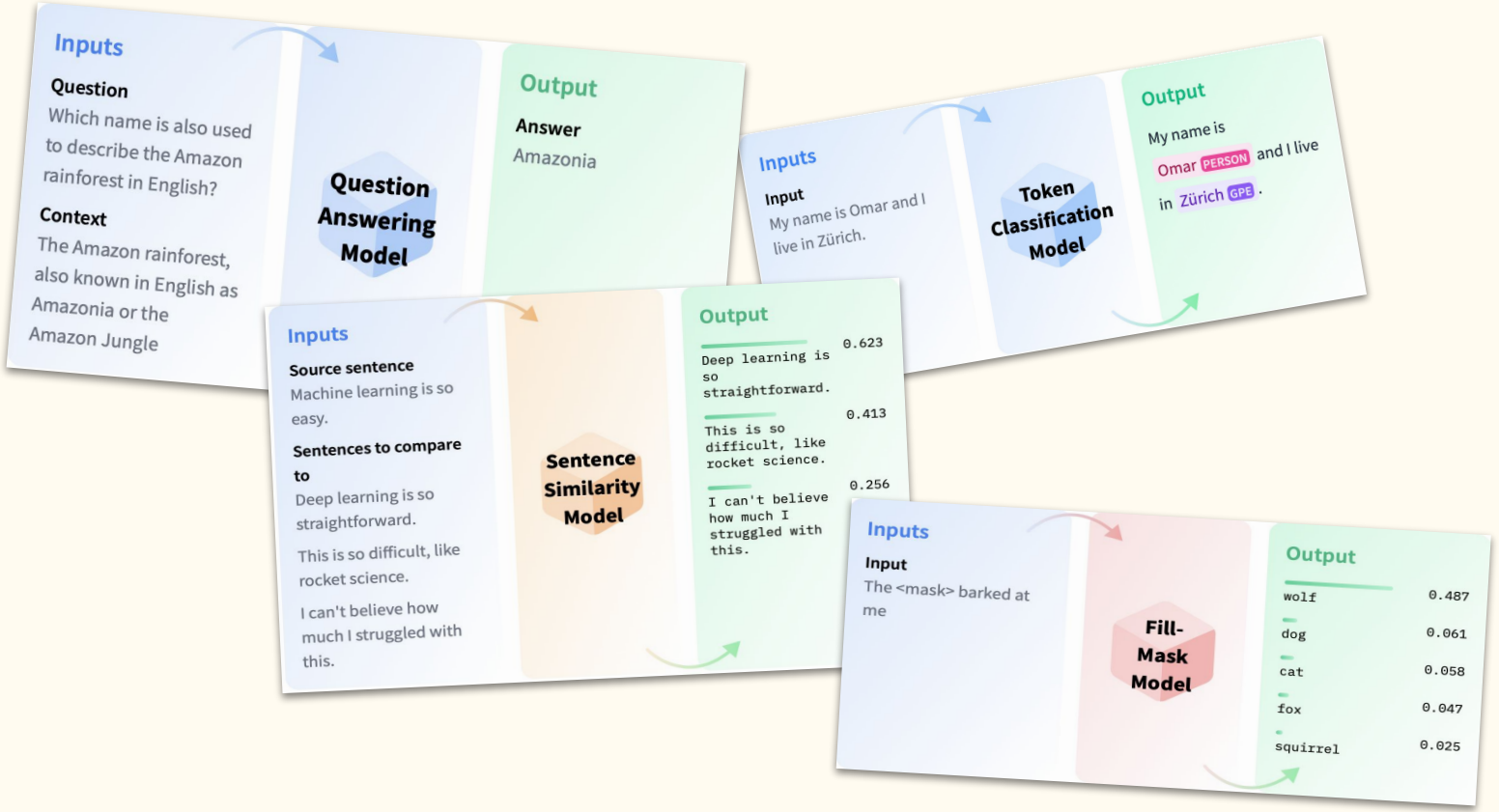
Nath, Mathis et al.2018

Natural Language Processing: tasks



huggingface.co: tasks

NLP: Tasks



huggingface

NLP: Author Attribution in Latin

Hypothesis rhythmic constructions in Latin help reveal author identity

Data *Samples*: ~37k prose fragments of 10 consecutive > 4-word sentences. *Labels*: Author class
Features (computed) Base (BF): histograms of function words, word & sentence lengths
Distorted views (DV): hide topic, keep style (4 strategies)
Syllabic length (SL): short ∪, long —, anceps *X* (ngrams)



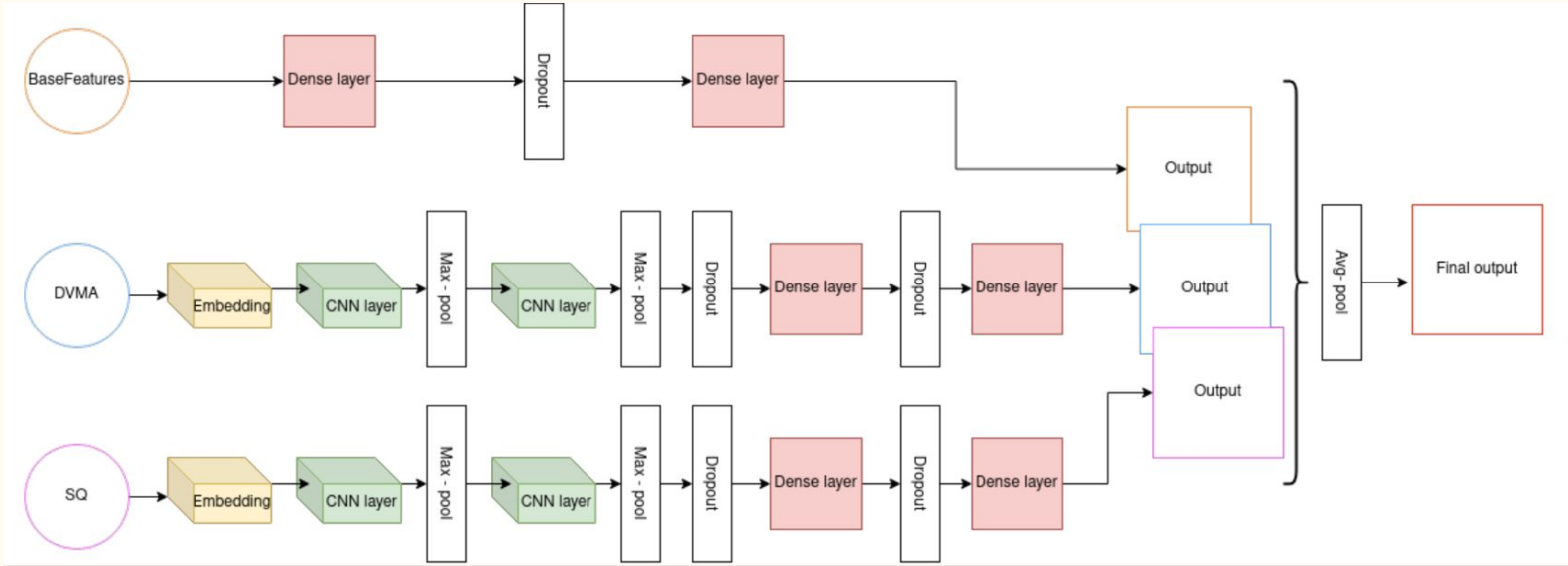
Task author attribution = fragment *classification*

Evaluation cross-entropy *loss* for training, *F*-score M/m *metric* for early stopping & feature importance

Model multi-channel NN (BF, SL, DVs), CharCNN 5 layers, avg proba decision

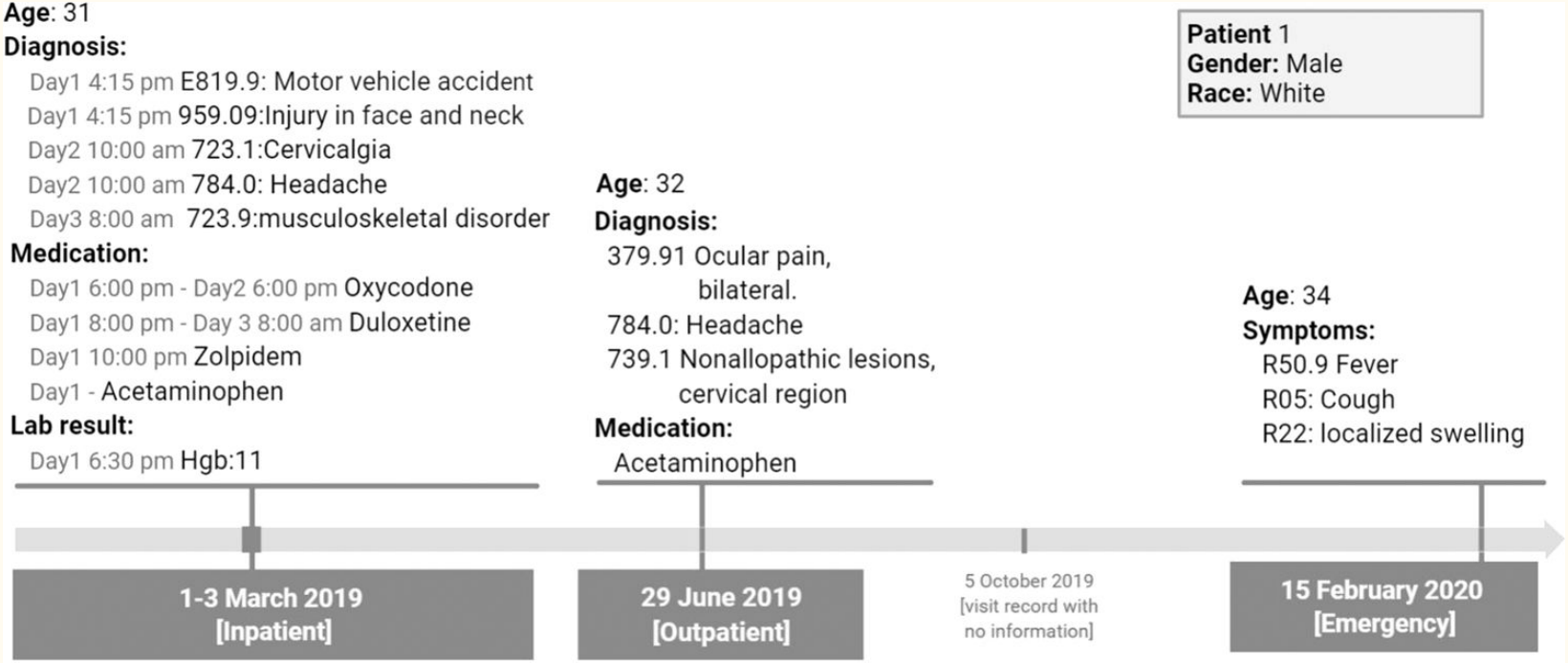
Corbara 2021 (preprint)

NLP: Author Attribution in Latin



NN ensembling approach for decisions from a variety of (computed) features.

NLP: MedBert



Rasmy et al., 2021

NLP: MedBert

Hypothesis Structured electronic health records (EHRs) offers rich and well-structured information that reflects the disease progression of each patient

Data

1. Cerner Health Facts: 68 million patients from 2000 to 2017
2. MarketScan Research Database: 170 million patients from 2011 to 2015

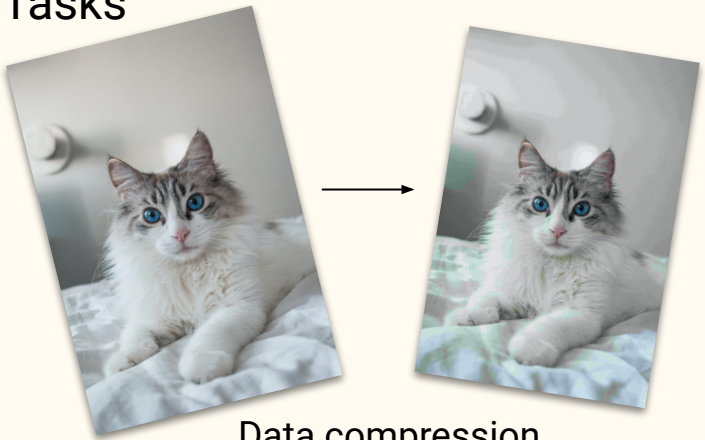
Task prediction of disease & prolonged length of stay (Prolonged LOS) in hospital

Model bidirectional encoder representations from transformers (BERT)

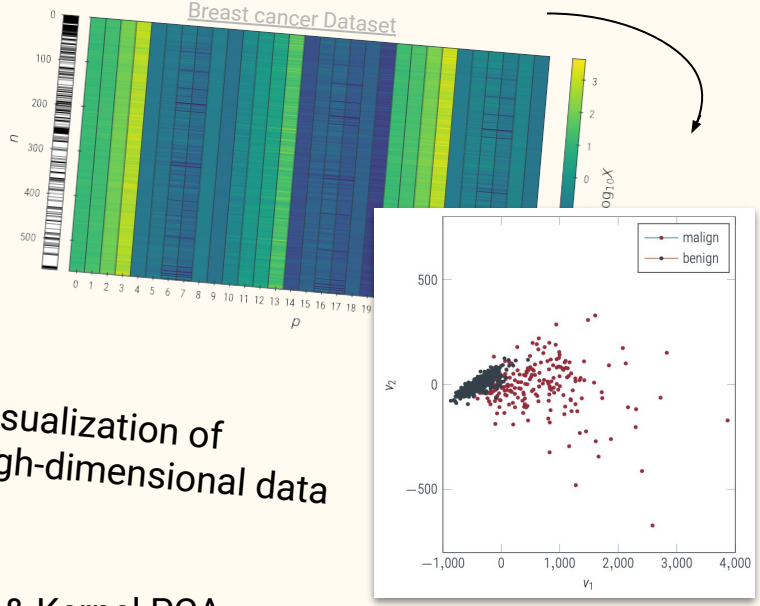
Rasmy et al., 2021

Dimensionality reduction

- Data
  - High-dimensional data (e.g. k space data)
- Tasks



Data compression



Visualization of high-dimensional data

- Models
  - Unsupervised models
    - Principal component analysis (PCA) & Kernel PCA
    - Autoencoders
    - t-SNE, UMAP (for visualization)
- NOTE: Often for pre-processing of the data

Cell-type prediction for single-cell transcriptomics

Hypothesis can identify similar cells via gene expression profiles

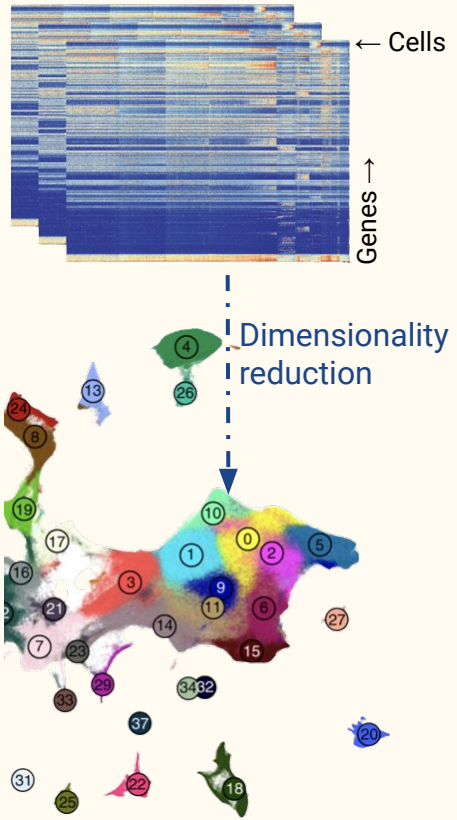
Dataset gene expression matrices (read counts per million)

Task *dimensionality reduction*

Evaluation % neighbor cells preserved in lower-dim space

Model *t-SNE* with PCA initialisation

Luecken,Theis et al.2019

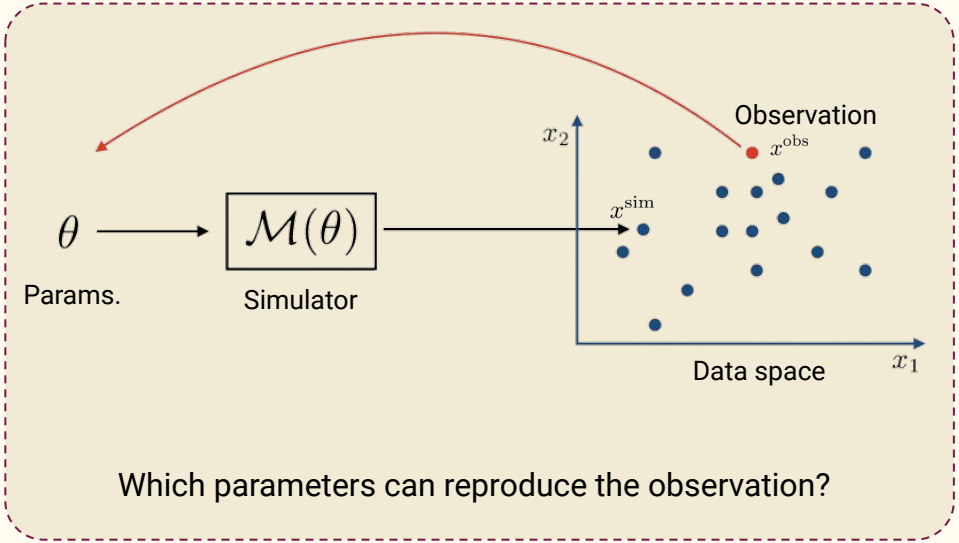


Kobak, Berens et al.2019

Simulation-based inference (SBI)

Tasks

- General inference in absence of likelihoods (losses)
- Particularly suited for black-box simulators $x^{\text{sim}} = \mathbf{M}(\theta)$.
- How can we learn a distribution over the parameters θ such that $\mathbf{M}(\theta) \approx x^{\text{obs}}$



Note

We have a full 3-day workshop on SBI

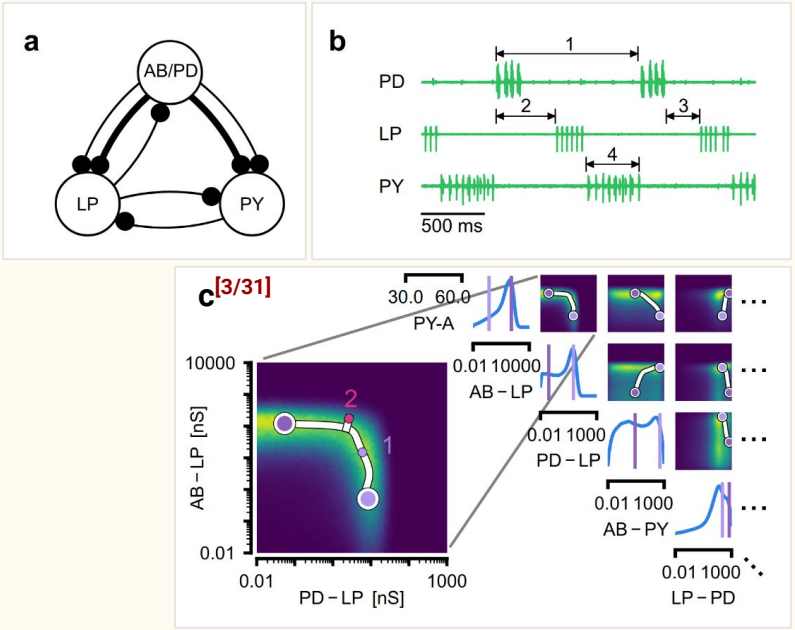
SBI: Stable Firing in the Pyloric Ganglia

Hypothesis the pyloric rhythm is robust to changes in conductances θ.

Dataset summary features of cell-electrophysiological (x^{obs}), computer simulations of the same $x^{sim} = \mathbf{M}(\theta)$.

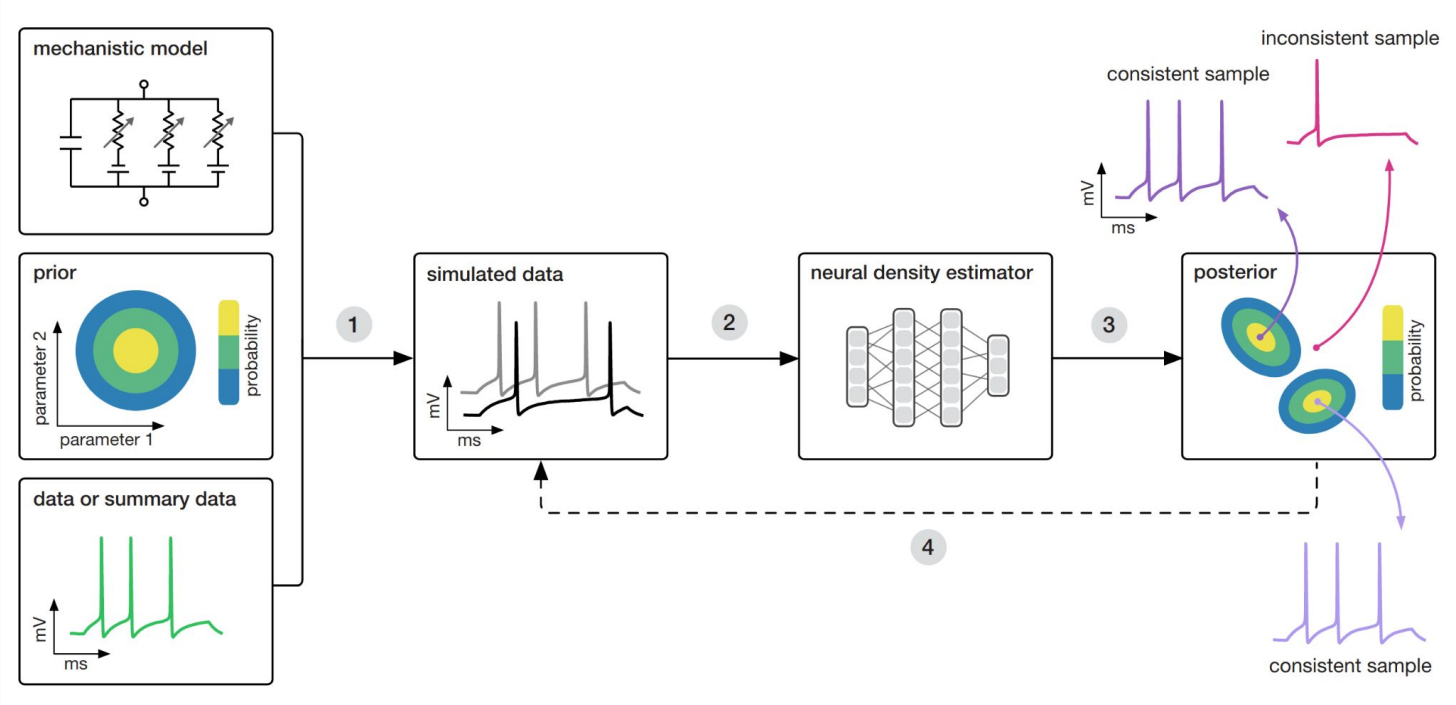
Task Bayesian solution for the inverse problem "$\theta = \theta(x^{obs})$"

Model normalizing flows (invertible NNs) as conditional density estimators



Gonçalves et al. 2020

SBI for mechanistic models



Gonçalves et al. 2020

Limitations of Today's ML for Science

- Inductive paradigm, limited by data
  - Bad at extrapolation
  - Building in inductive biases still a craft
- Compute intensive
- Correlational, mostly not causal
  - But can help you fit mechanistic models that are causal
- Interpretability (mostly post-hoc)
- Uncertainty quantification

Should I use ML for my problem?

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN





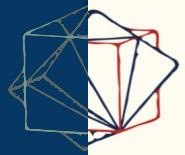
Use ML when the problem…

- … has a simple objective,
- is too complex for explicit rules,
- is constantly changing,
- is perceptive, or
- is observable, but unstudied

Don't use if:

- there exist performant classical models
- every decision must be explainable
- errors are high-consequence,
- getting dense data is infeasible or costly…

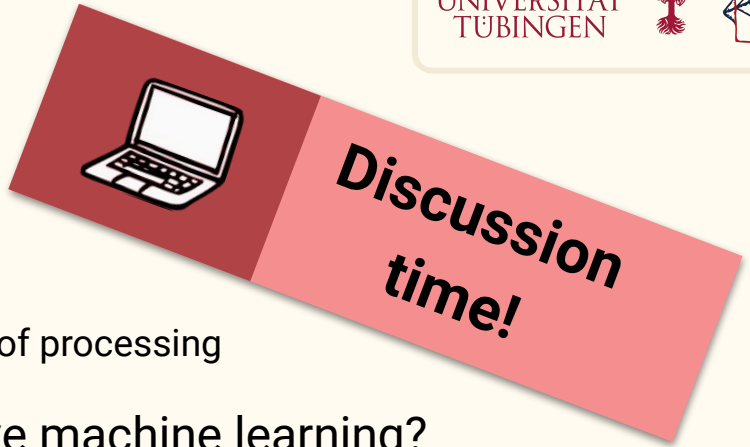
Machine Learning in Science



Bring Your Science Problem

Task, data, metrics...

Checklist for group discussion



1. What is the scientific question?
2. **Data:** Describe your dataset:
   - data modalities, labels (if there are), size, state of processing
3. **Task:** Which part of the work might involve machine learning?
   - Identify the learning paradigm that applies to your problem (supervised, unsupervised, ...)
4. **Performance:** How could you measure ‘success’ in your problem?
5. Are there relevant baselines/toplines for performance?

Use Cases!

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

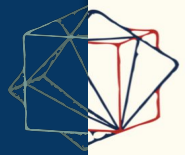




More if time permits.

The ML ⇌ Science Colaboratory

Connecting ML with Science

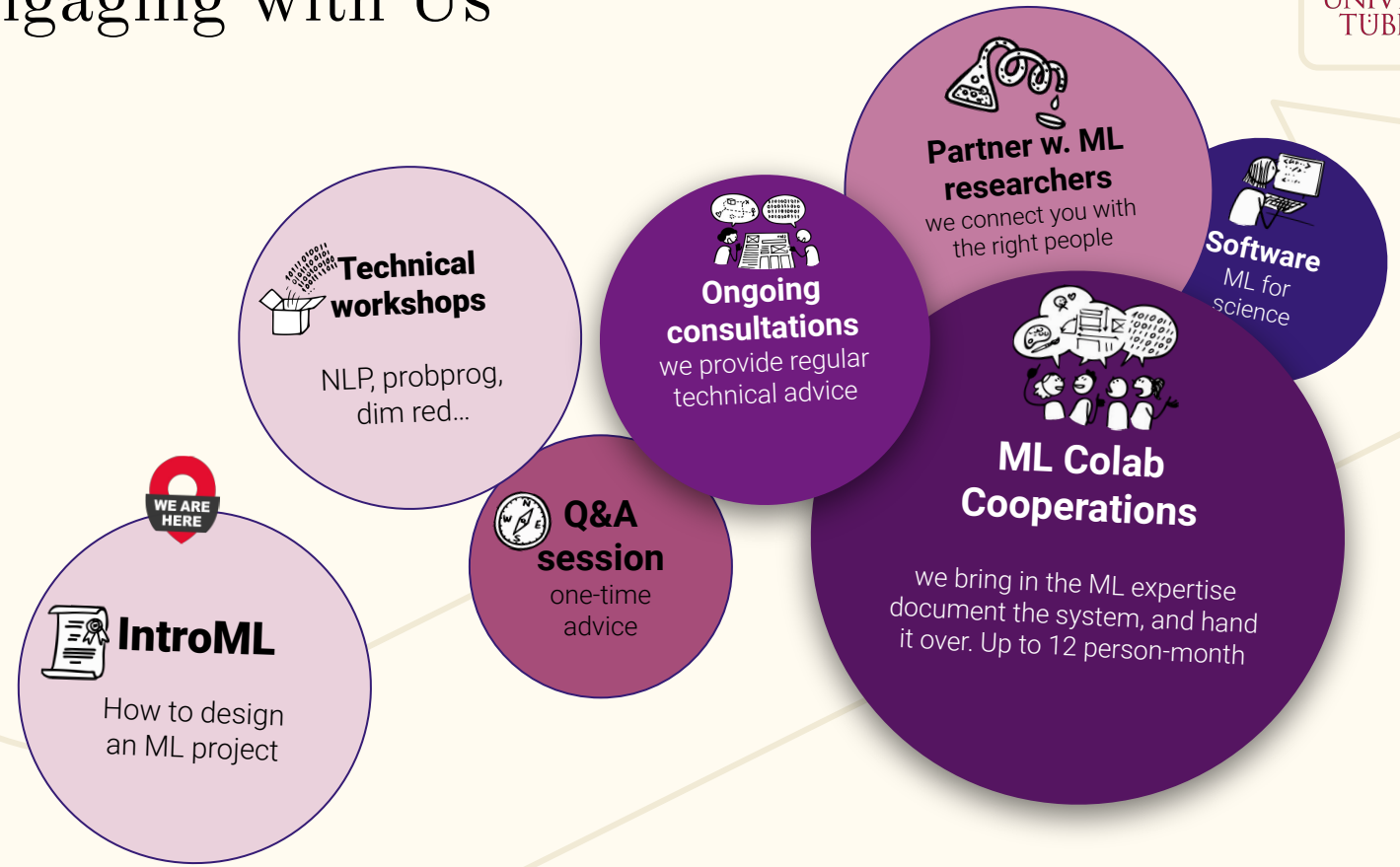


Our Activities

- For Tübingen *domain scientists*
  - Advice on the use of ML
  - Cooperations spanning a few months (you already filled the form!)
  - Trainings
- For Cluster *ML researchers*
  - Matching with domain researchers
    - Interesting problems and new datasets,
  - Increase applied impact of research projects (e.g. PhD innovation fund: structured learning)
- For the *community*
  - Sharing best practices in applied ML
  - Developing software to democratize access to ML
  - Nurturing a welcoming culture for women in applied ML

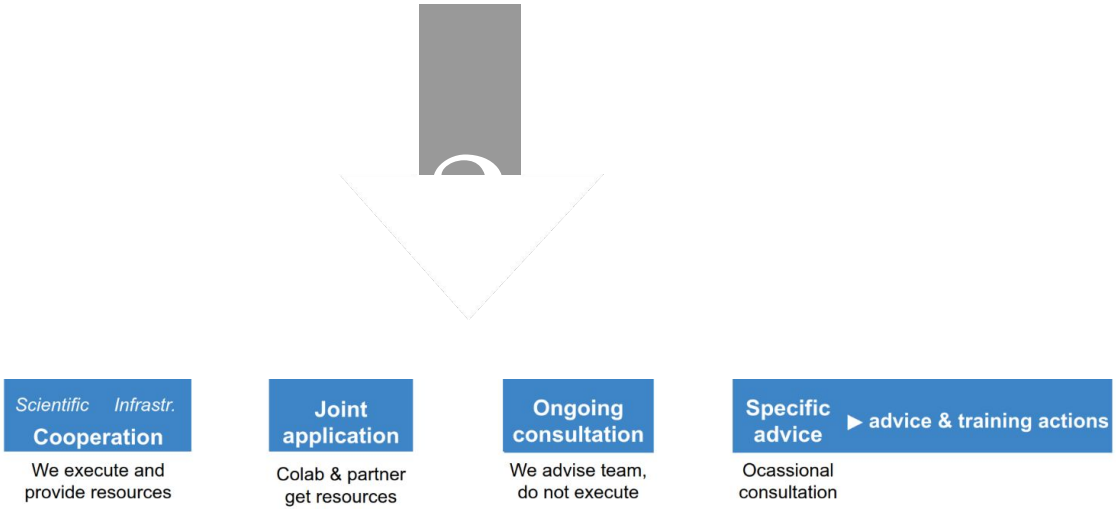
Engaging with Us

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



@mlcolab
https://mlcolab.org
mlcolab@inf.uni-tue.de

Mapping requests to actions



Mapping requests to actions

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Data

Research question

Skills & resources (partner)

MLColab ⇌ partner fit



Scientific *Infrastr.*
Cooperation
We execute and provide resources

Ongoing consultation
We advise team, do not execute

Specific advice ▶ advice & training actions
Ocassional consultation

Mapping requests to actions

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Data	Immediately accessible Sufficient
Research question	Closes domain gap Can tackle with applied ML
Skills & resources (partner)	Understands ML Can define project Can use results Team commitment Has ML-competent member Has access to compute
MLColab ⇌ partner fit	Scheduling fits Colab domain understanding Colab method capability Effort < 12 M FTE



Scientific *Infrastr.*
Cooperation
We execute and provide resources

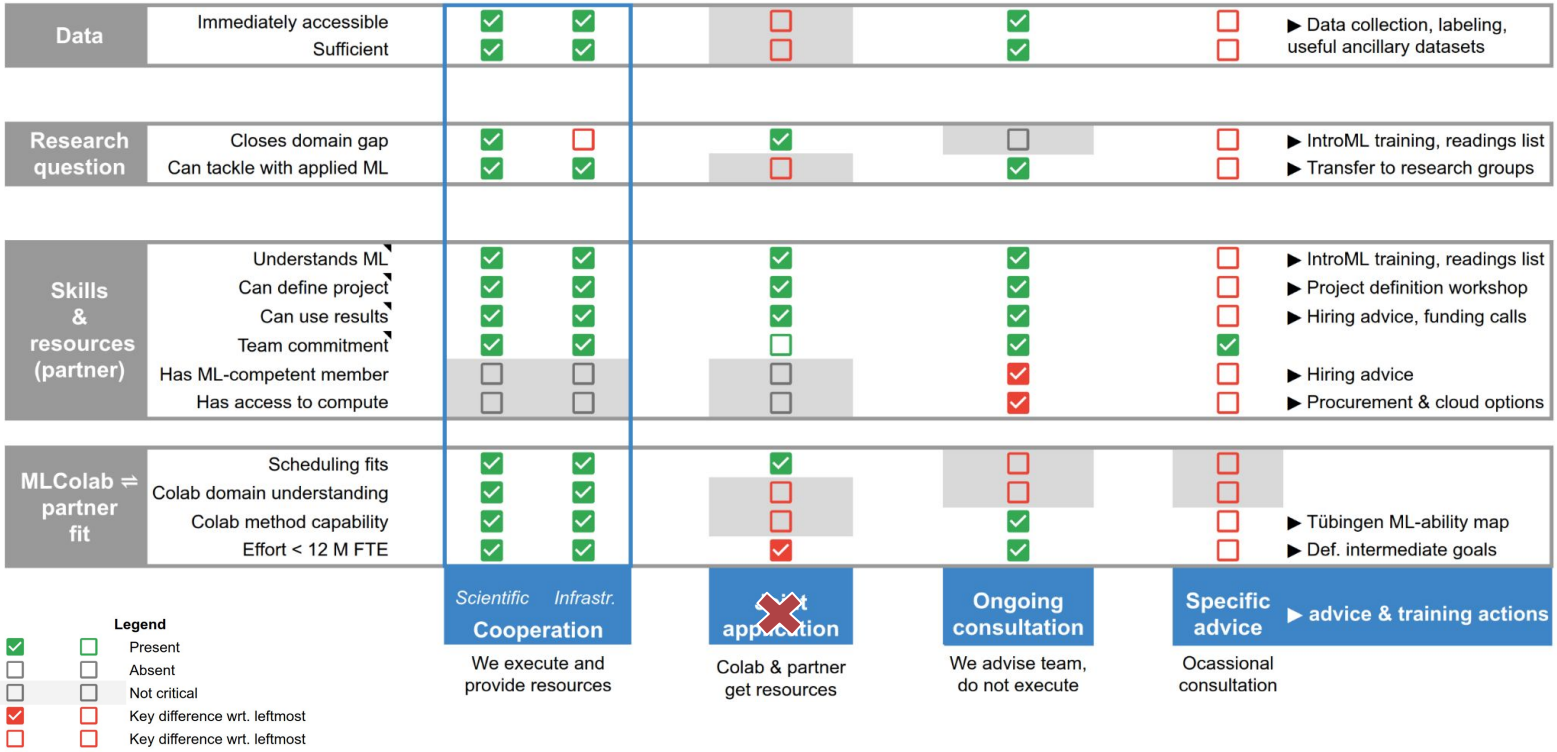
application
Colab & partner get resources

Ongoing consultation
We advise team, do not execute

Specific advice ▶ **advice & training actions**
Ocassional consultation

Mapping requests to actions

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



The Methods Center – *Methodenzentrum*

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

An institute for quantitative analysis **in the social sciences**.

Expertise

- multivariate & multilevel methods,
- (nonlinear) latent variable models,
- item-response theory,
- structural equation models,
- longitudinal studies, and
- causal mediator models.

Services

- consulting
- cooperations
  - publications
  - joint applications.

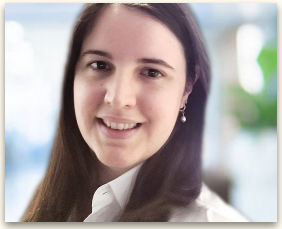
✉ office@mz.uni-tuebingen.de

In closing...

Thank you very much for attending our IntroML workshop!


Seth


Álvaro


Emily


Vladimir


Hanqi

One last thing,

… we thrive on critical, constructive feedback

please visit https://mlcolab.org/public-events/introml-workshop-july-2023 and give us some!