IntroML

4 July 2022

IntroML

Welcome

The ML ⇌ Science Colaboratory

Plan for the morning

Agenda

9:00	Opening and introductions
9:20	**Core ML Concepts** **ML & software ⁄ learning paradigms ⁄ supervised learning ⁄ NNs**
~10:30	**short break**
10:40	**Tutorials!**
11:45	**Machine Learning in Practice** **project pipeline: data, models & evaluation ⁄ ML in Science**
~12:30	**end (of our part)**

?! **Questions** welcome at any time

Slides, links, and notebook at
mlcolab.org/**introml-rhetai**

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! Tell us…

… your name

… field of research

… the role ML plays for your research

The ML ⇌ Science Colaboratory

@mlcolab

At the Cluster ML: new perspectives in science

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

via ① cooperations – ② training – ③ scientific ML software

@sethaxen
Seth Axen

@elenasizana
Elena Sizana

@alpiges
Alexandra Gessner

@alvorithm
Álvaro Tejero

@HanqiZhou_Ivy
Hanqi Zhou

- **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!

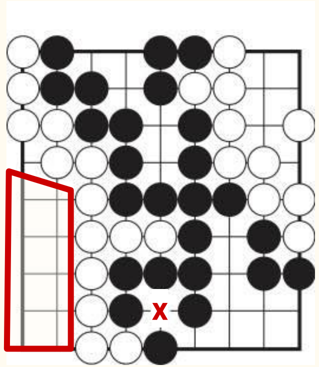
Core Concepts in Machine Learning

Core Concepts in Machine Learning

What is ML?

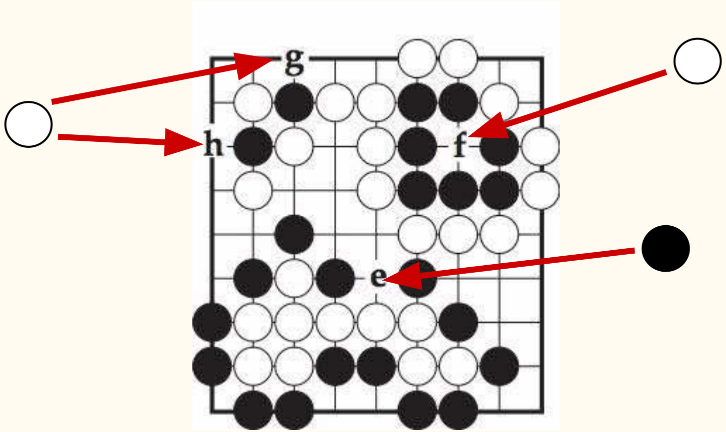
What problems does it solve?

Games and the nature of intelligence

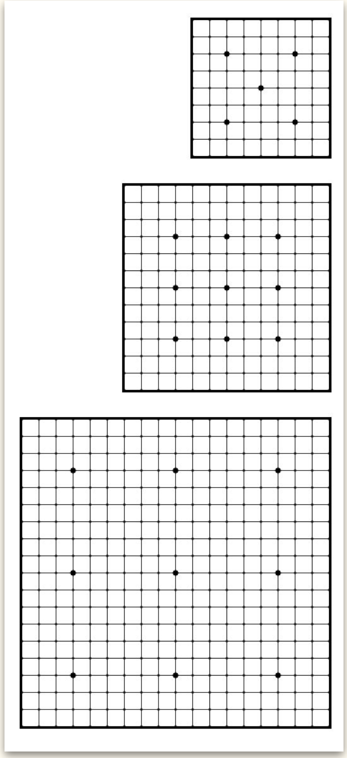


- form territories by surrounding empty areas
- capture by completely surrounding (*prisoners*) x

Points = crossings + prisoners



Real go is 19x19



British Go association

A surprising move

See also [AlphaGo - The Movie]

Definition of Machine Learning

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: points at the end of the game
- D: database of past games

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

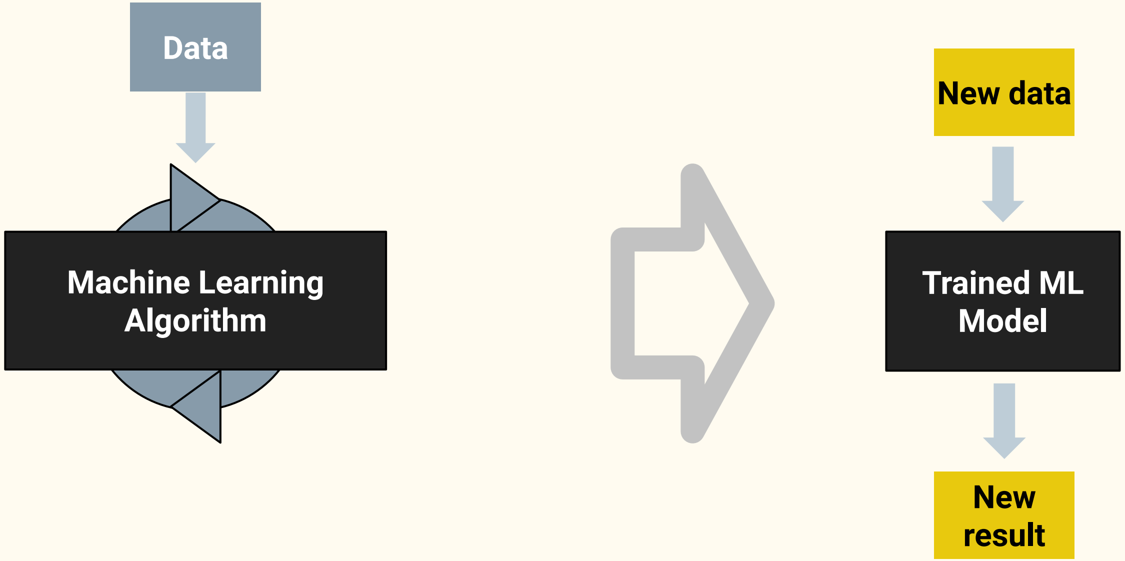
***Machine* learning**: the study of **algorithms** that allow computer programs to automatically improve **on a specific task** through **experience**

Mitchell, 1997

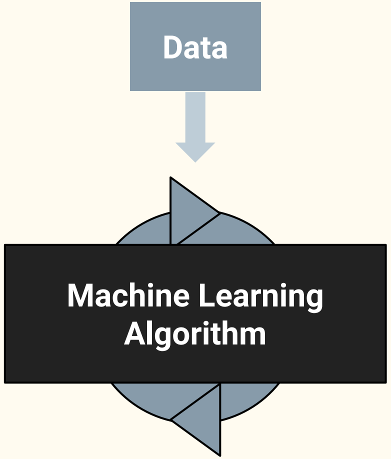
ML: Training and deployment

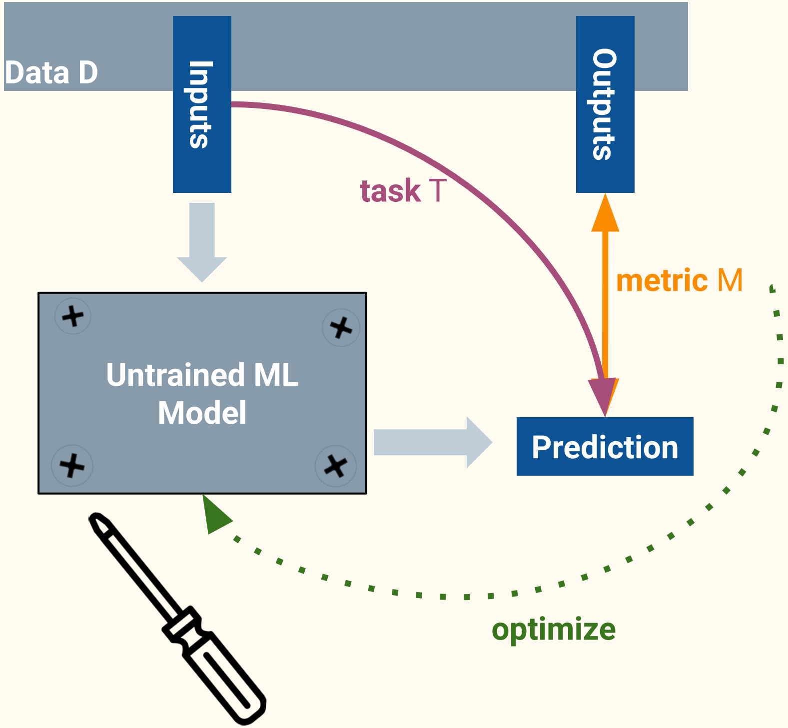
Training phase

Use, or "**inference**" phase

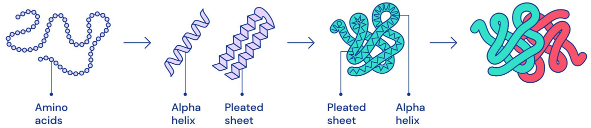


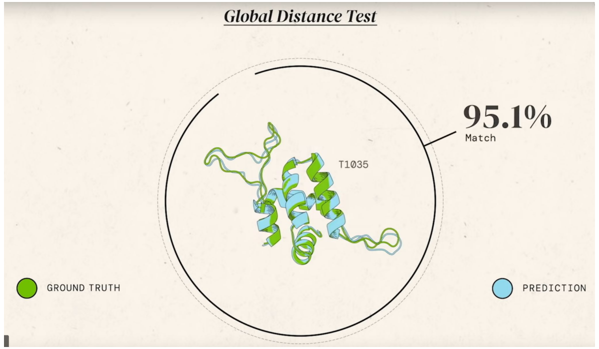
Inside the black box

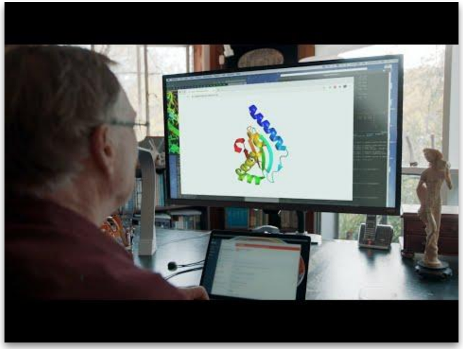




From proteins to planets, ML in science

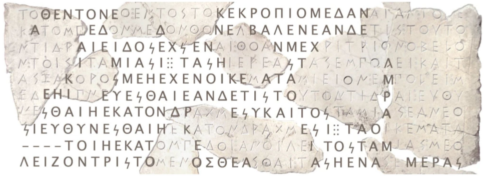






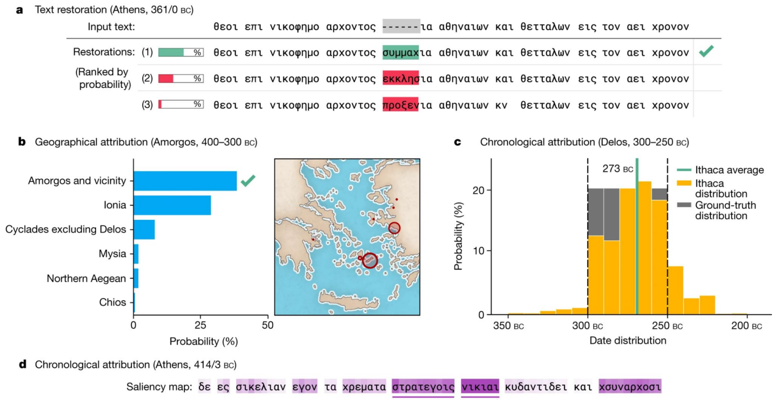
Protein structure prediction with transformers

Senior et al. 2020, Jumper et al. 2021



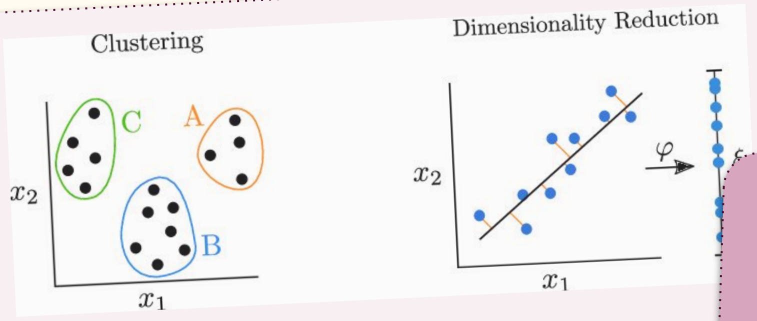
Text restoration and geochronological attribution with data augmentation

Assael, Sommerschield et al. 2022



Learning paradigms

Main Learning Paradigms

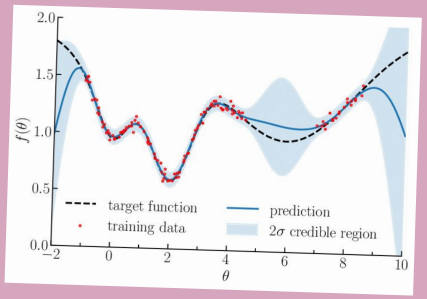


Unsupervised learning

Exploring structure

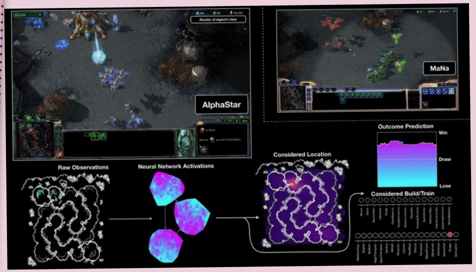
Supervised learning

Teaching to label



Reinforcement learning

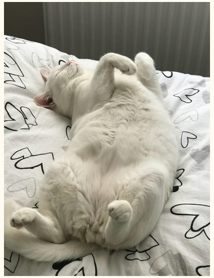
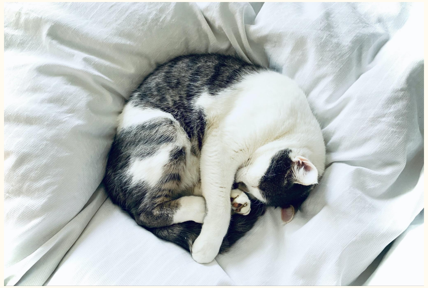
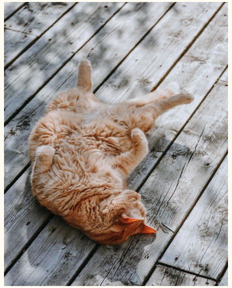
Harnessing interaction



Figures DeepMind 2019, Leclerc 2018, jovian.ai 2021

Unsupervised learning

An example dataset

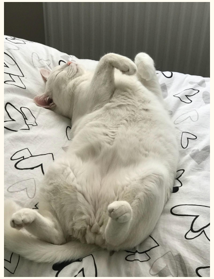
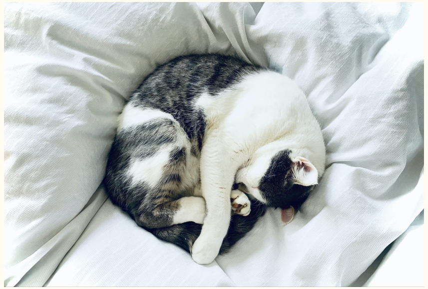
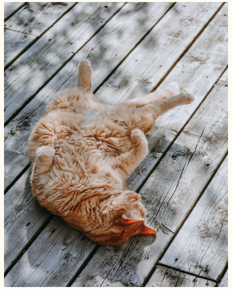
Instance

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

Feature

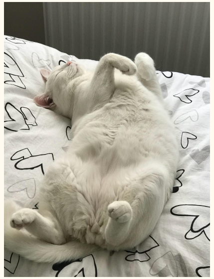
Photo by Sarah Ball on Unsplash Photo by Gökhan Konyalı on Unsplash Photo by Ben Wicks on Unsplash Photo by Jacalyn Beales on Unsplash

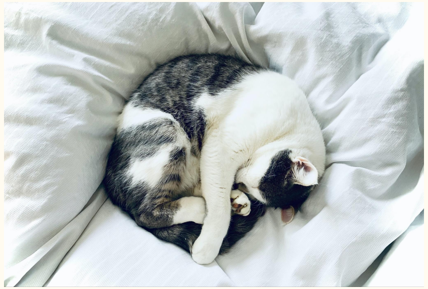
Finding structure in the data

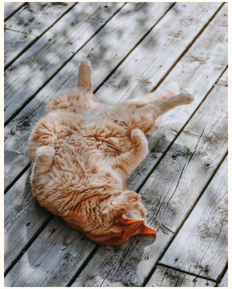





Task: group the images into any numbers of groups that you like

Finding structure in the data

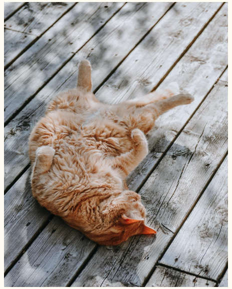
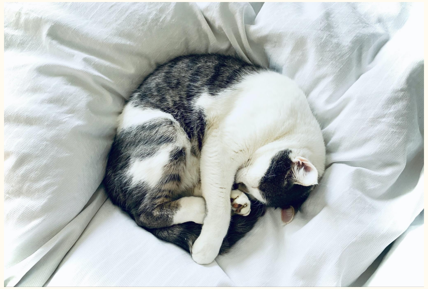
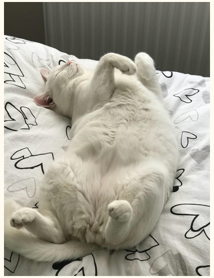






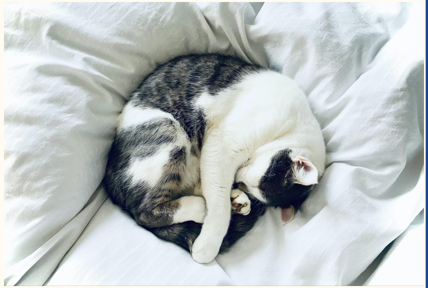
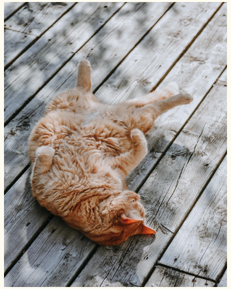
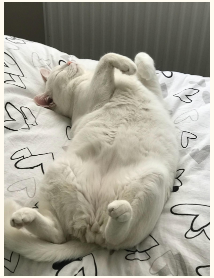
For example: 4 groups based on the breed of cat

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

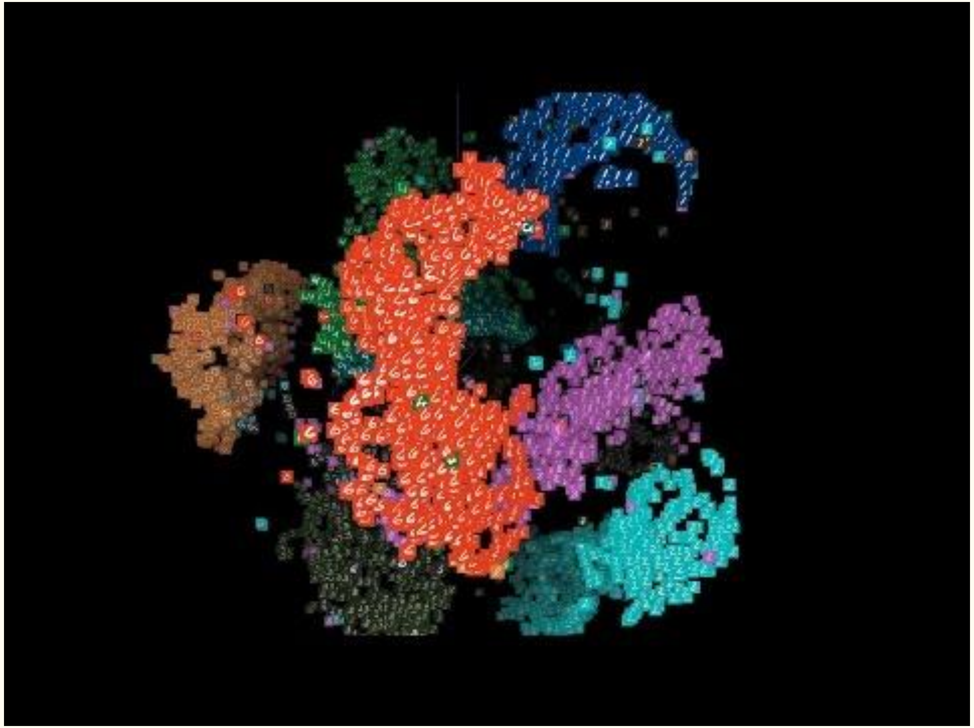
Unsupervised Learning

Goals

- Structure discovery
- Dimensionality reduction
- Community detection

Requirements

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



Supervised learning

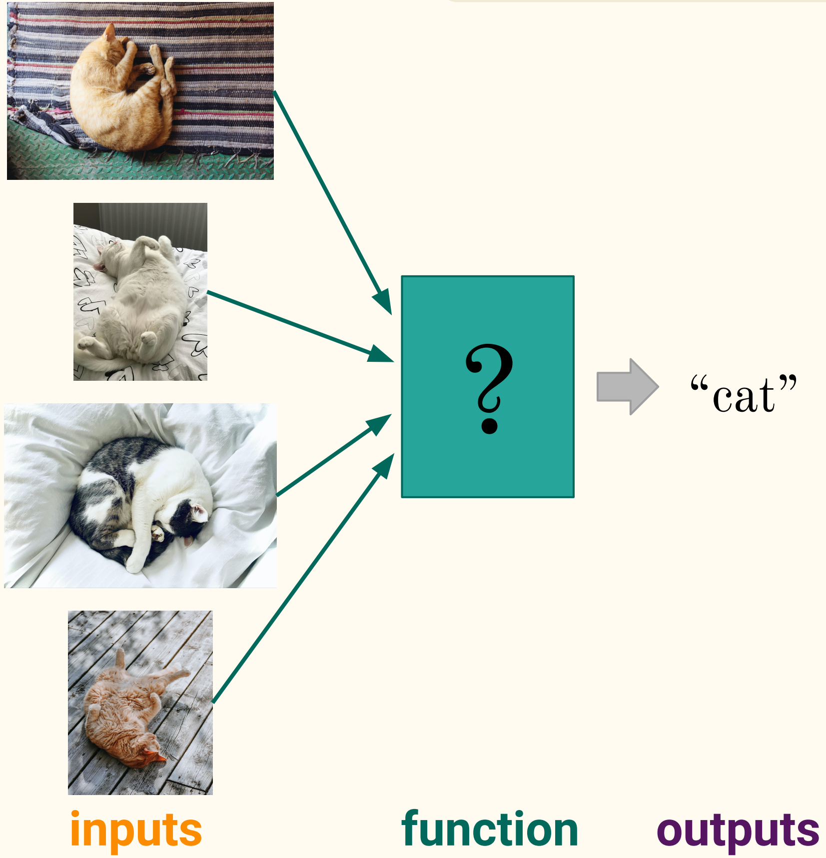
Supervised Learning

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

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

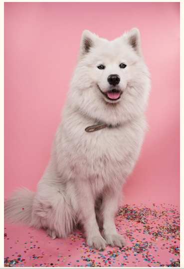
Modes

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

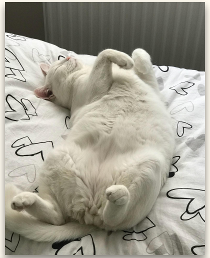


Supervised Learning: Classification

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

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

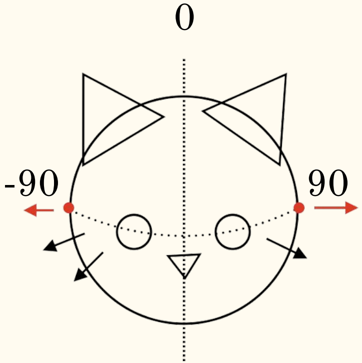


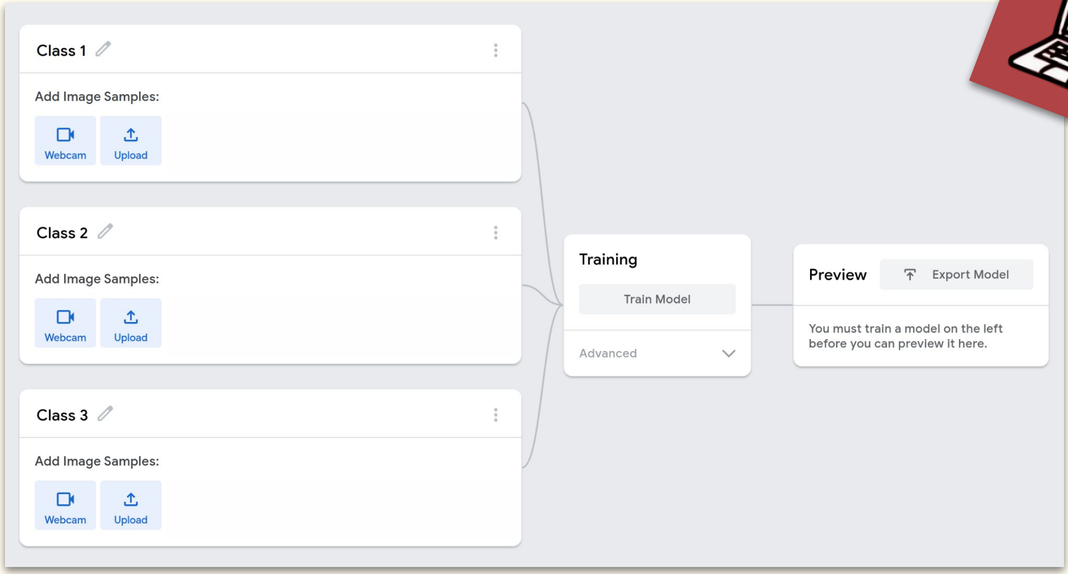
Photo by: cat-breeds-classifier

Core Concepts in Machine Learning

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

Supervised learning: One step at a time

In this notebook, we slowly introduce supervised learning, along with basic machine learning concepts we encounter on the way.

The data

At regularly spaced 1-dimensional points x, we have generated fake, noisy 1-dimensional observations y. We assume that there is some true but unknown underlying process f, so that

$$y_i = f(x_i) + \text{noise}.$$

x_i is a single **raw feature**. y_i is an **output** or continuous **label**.

Interactive notebook downloadable at mlcolab.org/introml-rhetai

Core Concepts in Machine Learning

Neural Networks

Layer upon layer upon layer

Building a neural network - terminology

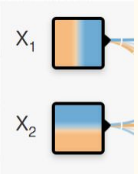
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

DATA

Inputs: coordinates

X_1 horizontal

X_2 vertical



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

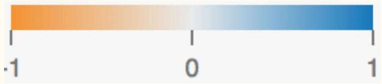
$X_1^{\,2}$, $X_2^{\,2}$, X_1X_2, sin X_1, sin X_2 ← augment input

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

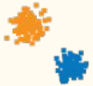
MODEL

$\left.\begin{array}{l}\textbf{Hidden layers} = \text{number of nestings in the NN; } \textit{here: fully connected}\\ \textbf{Neurons} = \text{units in each layer}\end{array}\right\}$ **architecture**

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

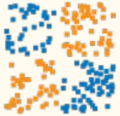


Neural network playground – Training



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

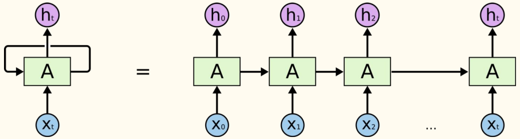
Use 1 layer and features → "feature engineering"

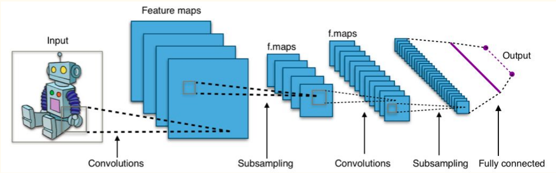
Construct a deep network, but only use X_1 and X_2

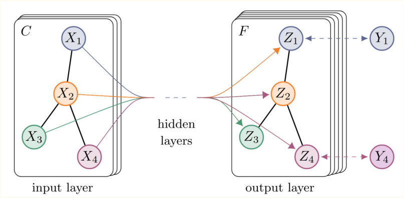
Testing: Use various sizes for the test set. How well do we generalize?

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

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

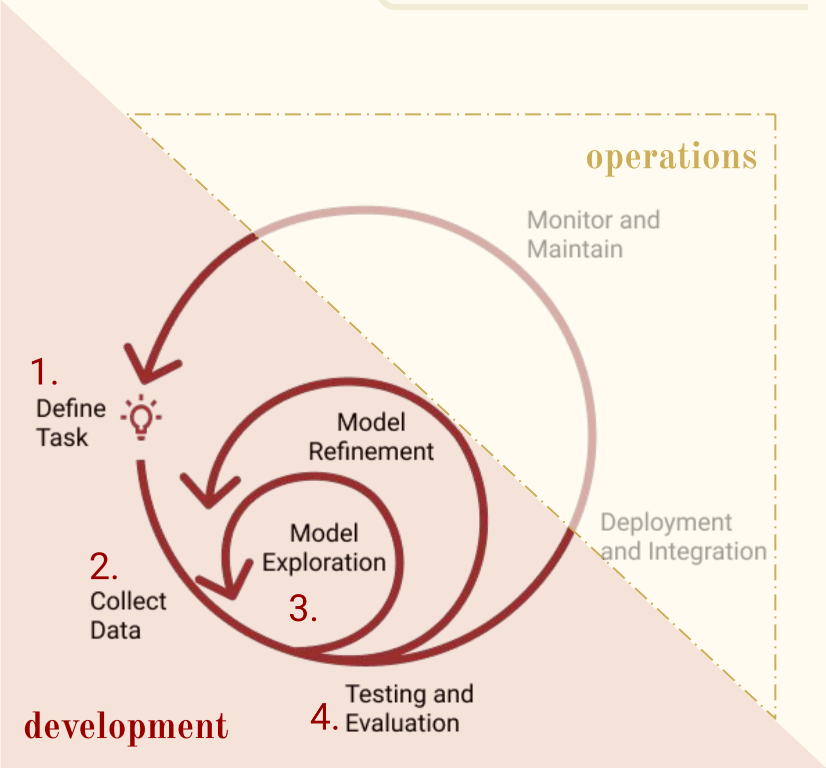


Figure based on [Jordan 2018: organizing ML projects]

1. Task: Feasibility & Impact of ML Project

Do you know your data?

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

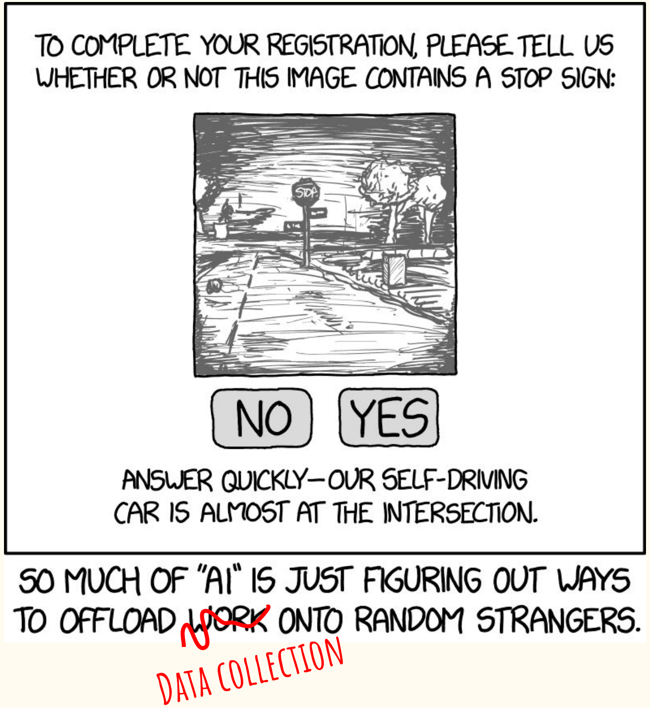
Model requirements, alternatives

- Are there pre-trained models?
- Does it need to be interpretable?
- Are there classical/hand-crafted models known to work well?

How to measure performance?

- Can performance be measured? How?
- Can humans do it? With what performance?
- Consequence of wrong predictions?
- Are there performance baseline? (even if non-ML)

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: user interfaces
- Automatic labeling (active learning)
- 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

2. Data: Pitfalls

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

2. Data: Processing

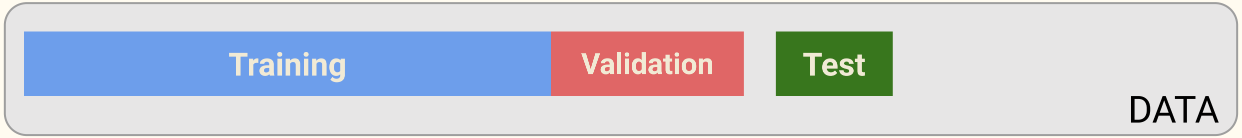
- Validation – verify **data quality:**
  - Are data types correct and consistent?
  - Is data plausible / consistent?
  - Do data satisfy ranges and constraints?
  - Reasons for data imbalance?
- Aggregation
- Cleaning
  - Outlier detection/removal
  - Integrating data from different sources
- Transforming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

! As domain expert, you know a lot about your data that you might take for granted

Take home:

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

3. Model: Training, Validating and Testing



- **Training** for **best parameters:** training data, minimize loss function

- **Validation** for **best *hyper*parameters:** held-out validation data, evaluate metrics <u>during</u> training (sometimes called dev set)

TRAINING

USE

- **Test** for **generalization performance:** test data, report metrics <u>after</u> training
  - Compare with **baselines** (random or specific) and toplines (human performance)

3. Model: Implementation & Embodiments

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

  `R, sklearn, mlj`

- **Probabilistic programming**
  - \+ Uncertainty quantification
  - – CPU mostly, expensive
  - Sampling in probabilistic DSL

  `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)

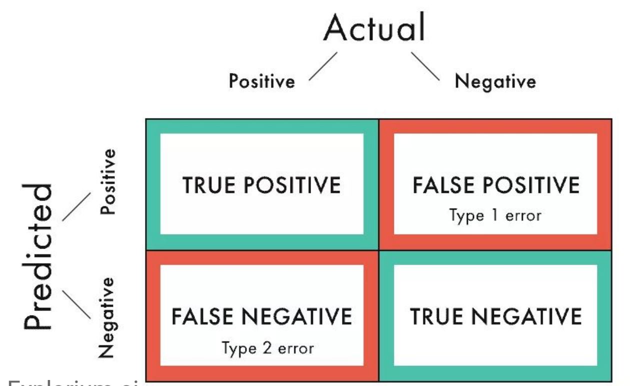
  `Huggingface, GPT, ...`

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, …



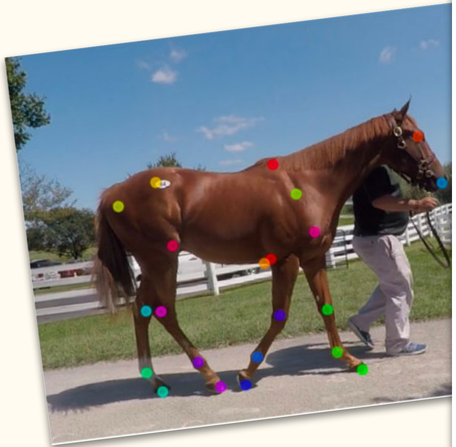
Explorium.ai

Machine Learning in Science

Learning from language, images, simulations…

Computer Vision (CV)

Data: images or videos



Pose estimation

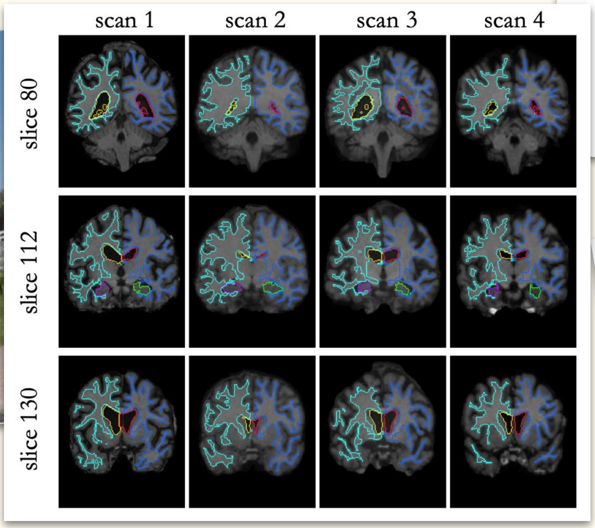
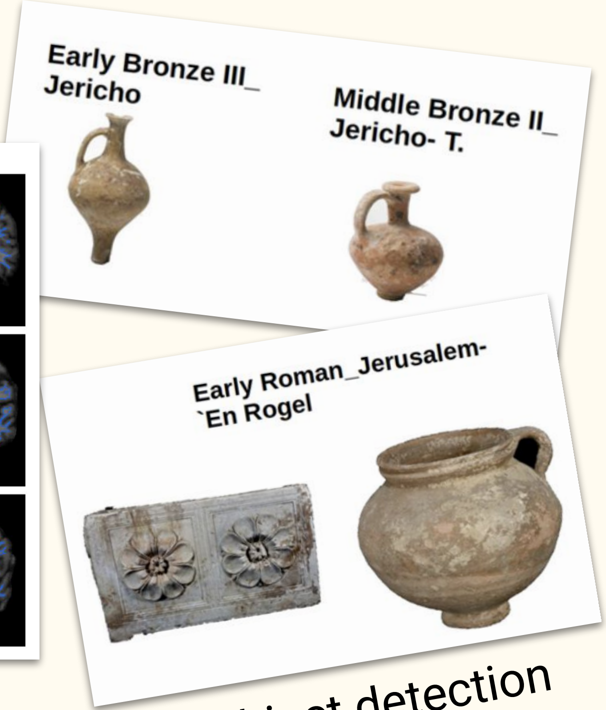


Image segmentation



Object detection

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

CV: pose estimation without body markers

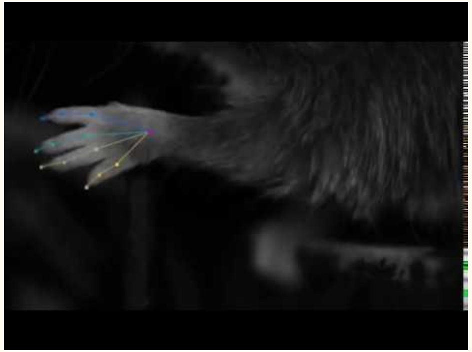
Hypothesis We can detect body parts of multiple animal species from unstaged video without body markers

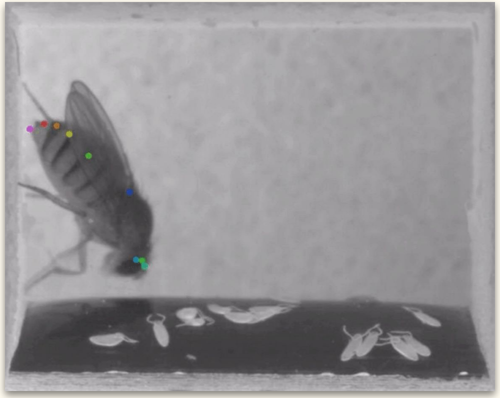
Dataset video frames with *labeled* positions of body parts

Task frame-based pose estimation

Evaluation *RMSE* between predicted and labeled positions

Model *pre-trained* ResNet with fully convolutional upsampling

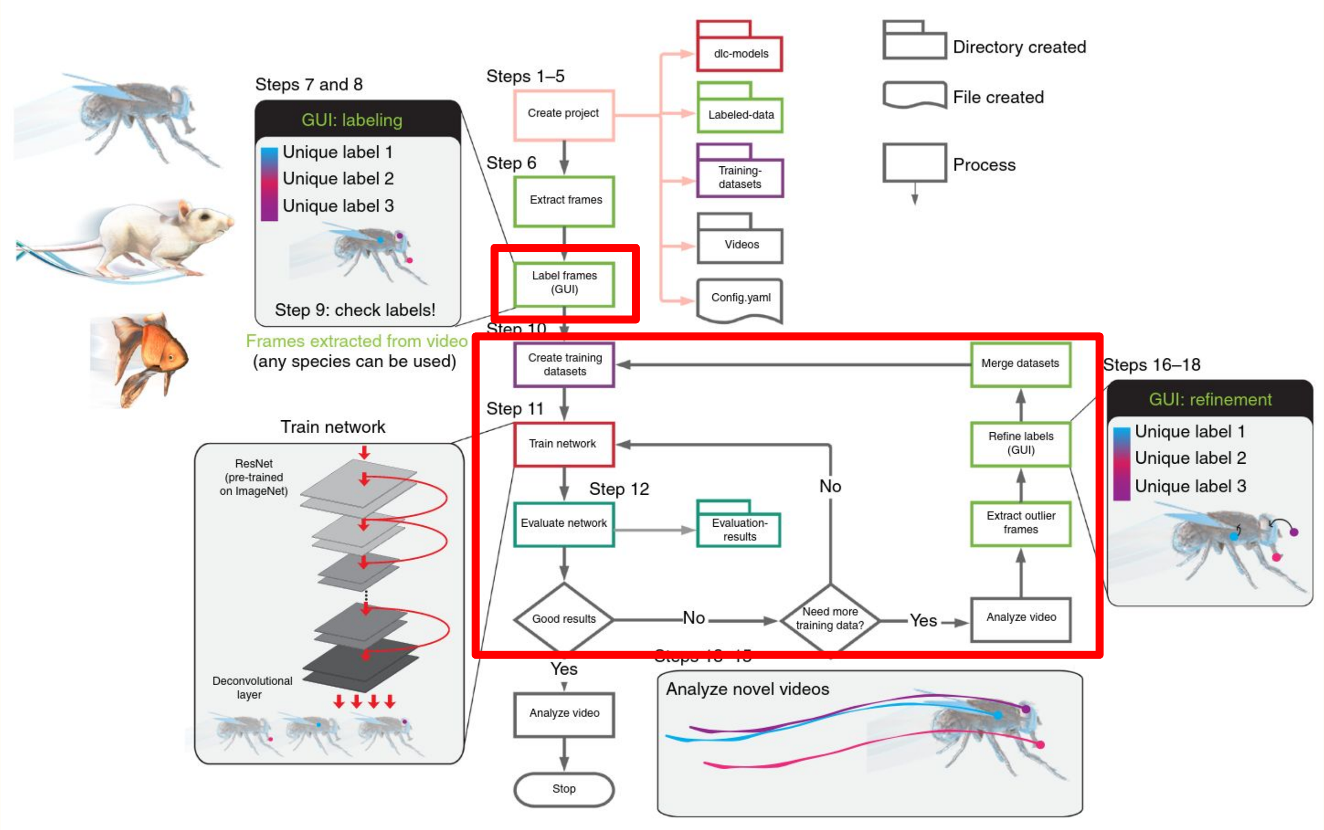




Fruit fly

Nath, Mathis et al.2018

CV: pose estimation without body markers

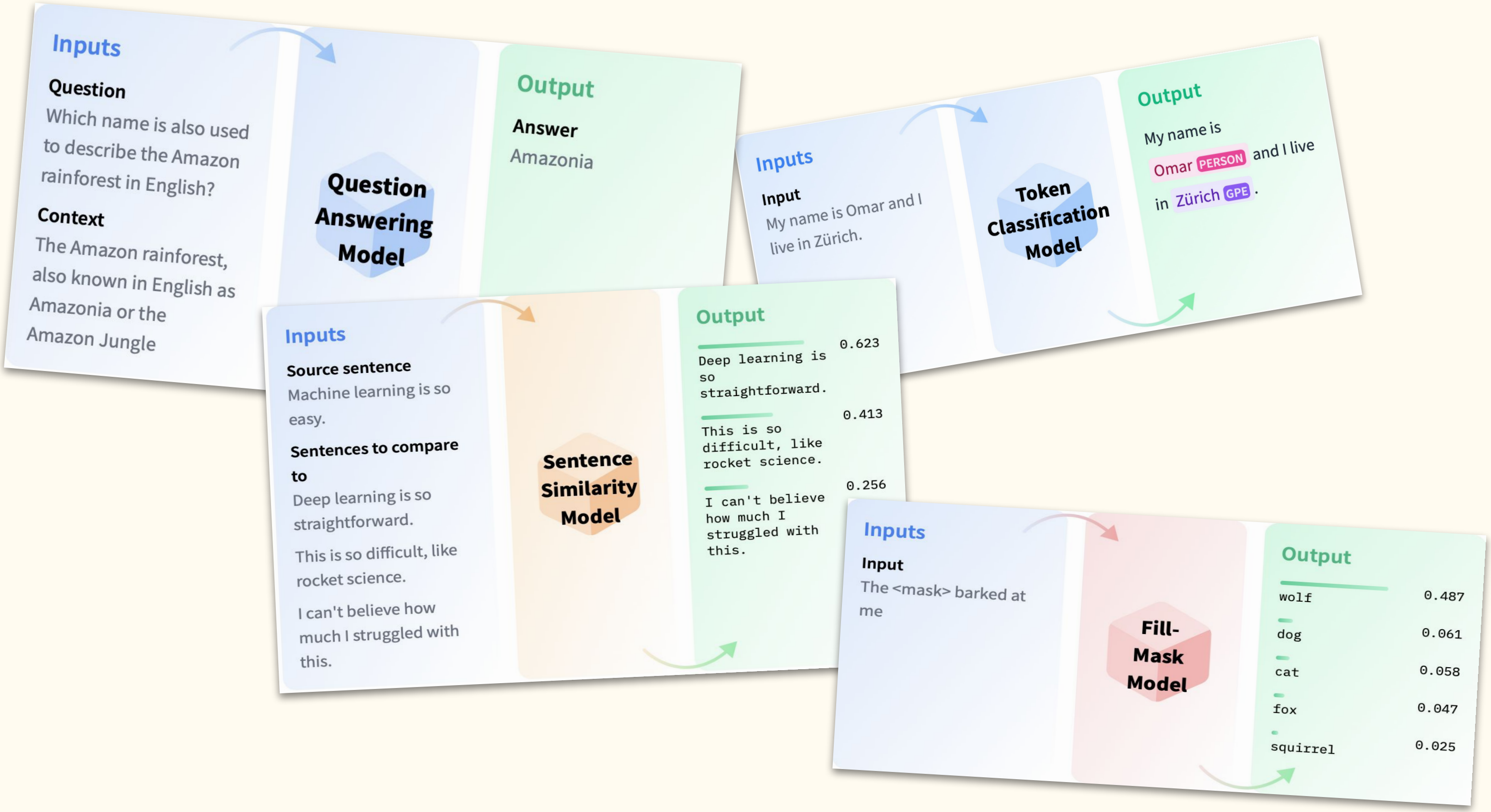


Nath, Mathis et al.2018

Natural Language Processing: tasks

huggingface.co: tasks

NLP: Tasks


Inputs
Question
Which name is also used to describe the Amazon rainforest in English?
Context
The Amazon rainforest, also known in English as Amazonia or the Amazon Jungle
Question Answering Model
Output
Answer
Amazonia
Inputs
Input
My name is Omar and I live in Zürich.
Token Classification Model
Output
My name is Omar PERSON and I live in Zürich GPE .
Inputs
Source sentence
Machine learning is so easy.
Sentences to compare to
Deep learning is so straightforward.
This is so difficult, like rocket science.
I can't believe how much I struggled with this.
Sentence Similarity Model
Output
Deep learning is so straightforward. 0.623
This is so difficult, like rocket science. 0.413
I can't believe how much I struggled with this. 0.256
Inputs
Input
The <mask> barked at me
Fill-Mask Model
Output
wolf 0.487
dog 0.061
cat 0.058
fox 0.047
squirrel 0.025

NLP: Author Attribution in Latin

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

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)

Arma vi|rumque ca|nō, Trō|iae quī| prīmus ab| ōrīs|| → – ∪ ∪| – ∪ ∪| – –| – –| – ∪ ∪| – *X*||

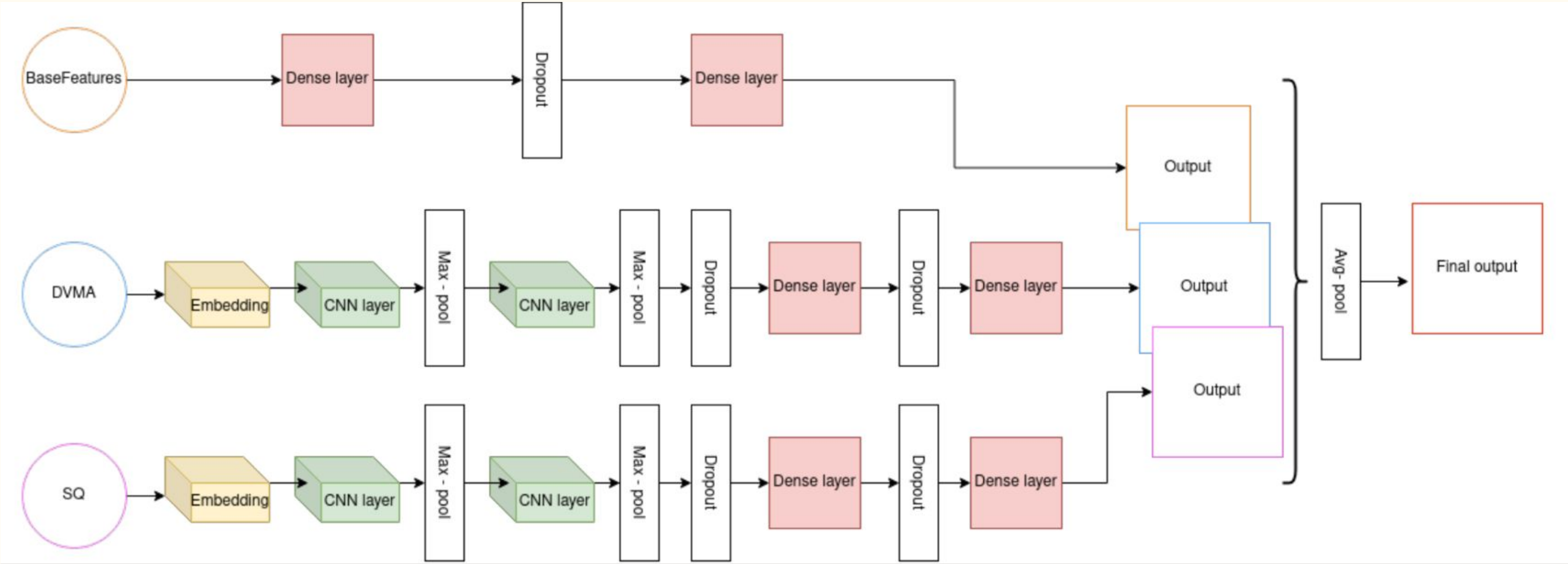
Task author attribution = fragment *classification*

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.

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?

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…

The ML $\rightleftharpoons$ Science Colaboratory

Connecting ML with Science

Engaging with Us

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

WE ARE HERE

IntroML

How to design an ML project

Technical workshops

NLP, probprog, dim red...

Q&A session

one-time advice

Ongoing consultations

we provide regular technical advice

Partner w. ML researchers

we connect you with the right people

Software

ML for science

ML Colab Cooperations

we bring in the ML expertise document the system, and hand it over. Up to 12 person-month

mlcolab@inf.uni-tue.de

In closing...

Thank you very much for attending our IntroML workshop!


Seth


Álvaro


Elena


Alex


Yutong

Please help us with some feedback!

We're looking forward to your workshop :)