

# **Deep Learning 1**

**Raoul Grouls, 25 april 2022**

# Hello world

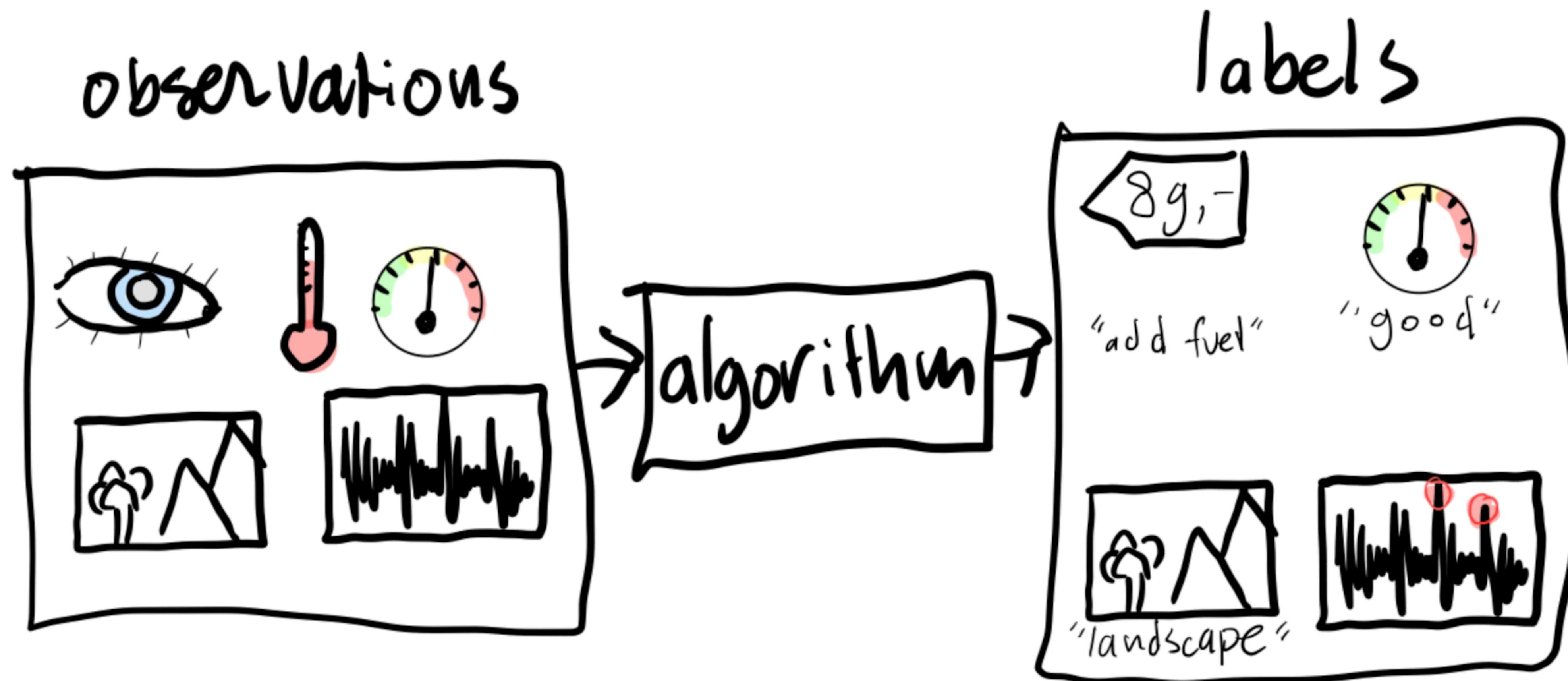
- Raoul Grouls, data alchemist, <https://the-pttrn.nl/>
- Building AI solutions from strategy to deployment
- Specialised in Deep Learning & Responsible AI strategy

# What is intelligence?

Exercise:

- 3 minute break out
- Find a definition of intelligence

# What is supervised AI?



# Exercise

3 minute break out

- Identify observations and labels in your work field
- Identify employees that have the job of connecting observations and labels

# Clarifying some terminology

This is an hierarchical list, where every item is a subset of the preceding item.

1. **Automation:** Autonomous agent executing fixed rules
2. **Artificial Intelligence:** Autonomous agent optimizing an objective (including Expert Systems, Fuzzy Logic, Symbolic AI, Constraint Satisfaction Problems, Cellular Automata, Genetic Algorithms and Swarm Intelligence)
3. **Machine Learning:** Learning from data. This includes both gradient-based and non-gradient-based methods. (linear regression, random forests, Support Vector Machines)
4. **Deep Learning:** Complex learning from data, gradient-based and with backpropagation (CNN, RNN, Transformers)

# What is deep learning?

A machine learning technique that uses **multiple layers of non-linear transformations**

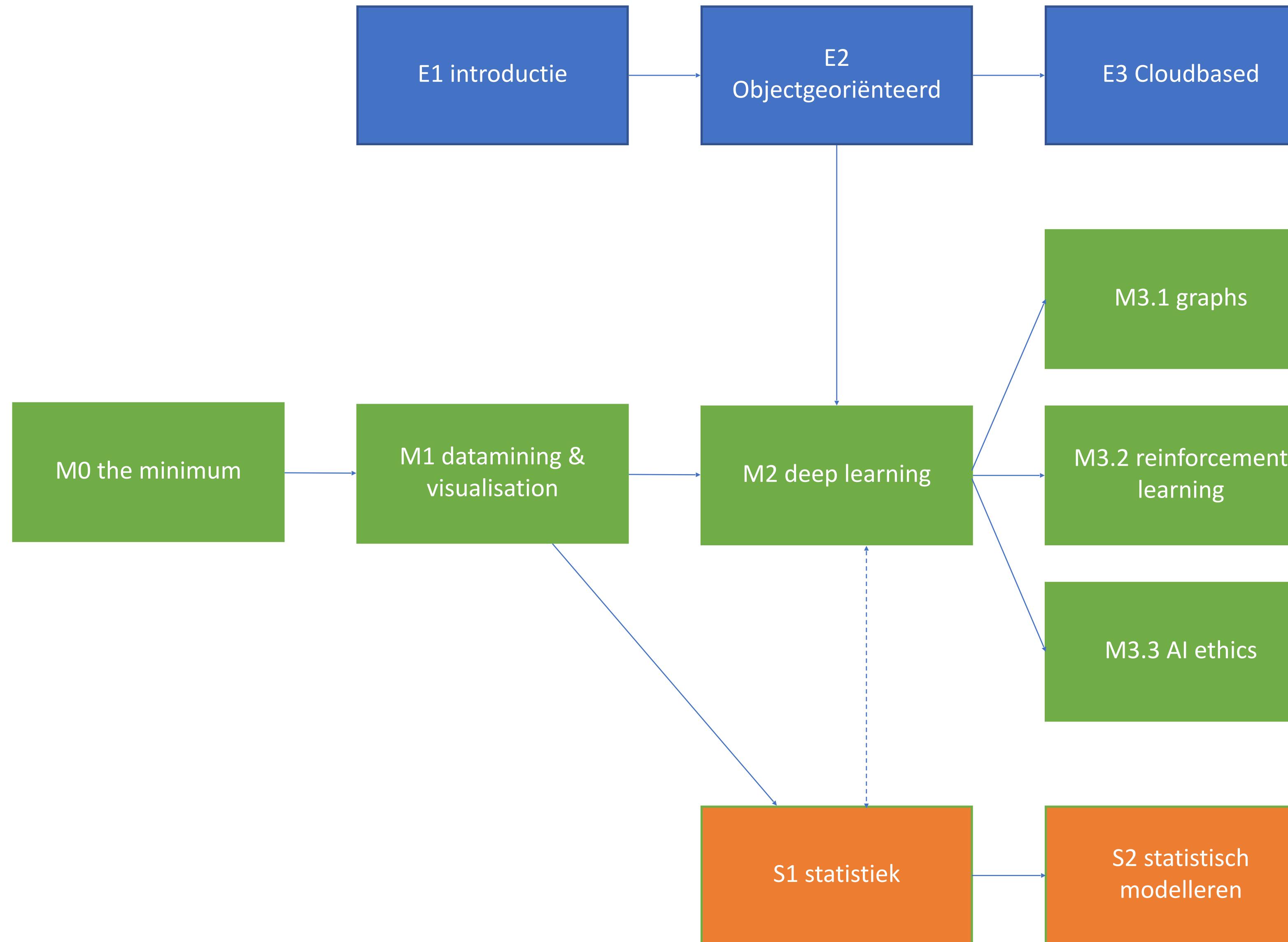
complex learning techniques using gradient-based optimization and backpropagation to process large amounts of data effectively

A technique to solve problems by providing data, and objective and often some sort of label, letting the computer find the solution using neural networks.

E: Machine learning / Engineering

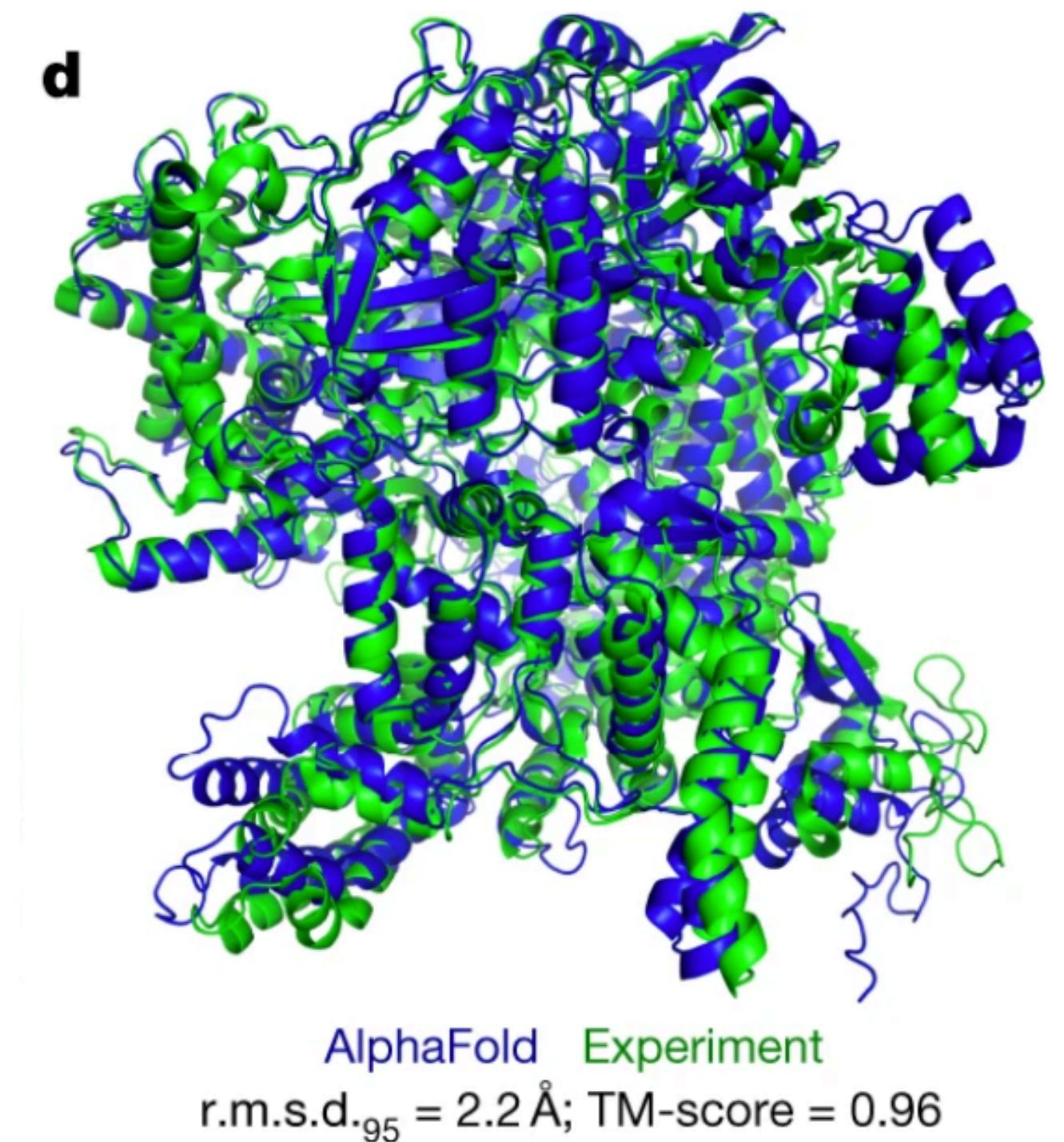
M: Machine learning / Modelleren

S: Statistiek



# A very short history of AI

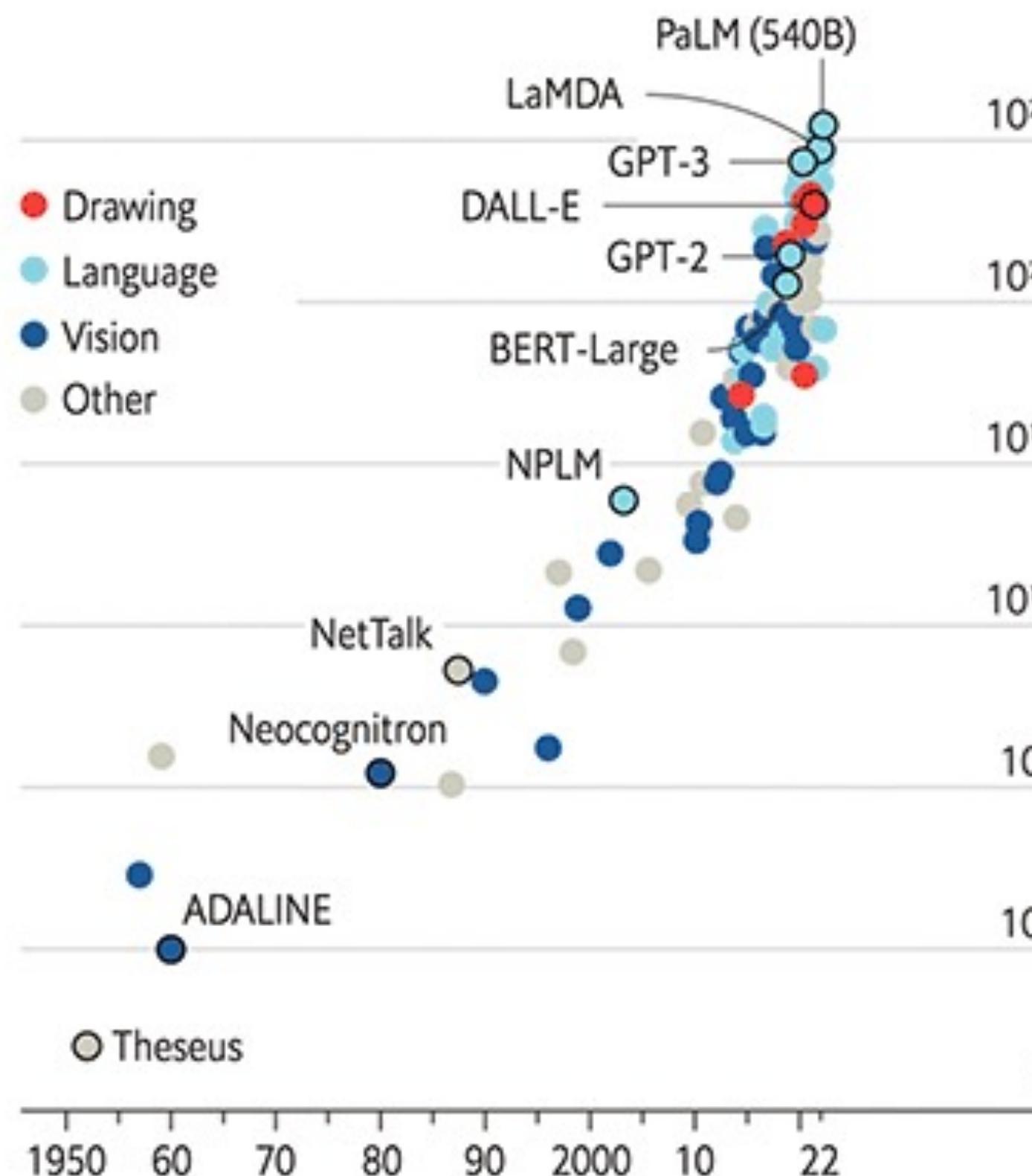
- 1950 Mathematics of Neural Networks invented
- 1990-2010 Support Vector Machines, Random Forest
- 2010 Deep learning
- 2015 Graph Convolutional Networks
- 2017 Attention mechanisms
- 2021 Protein folding
- 2023 chatGPT



## The blessings of scale

AI training runs, estimated computing resources used

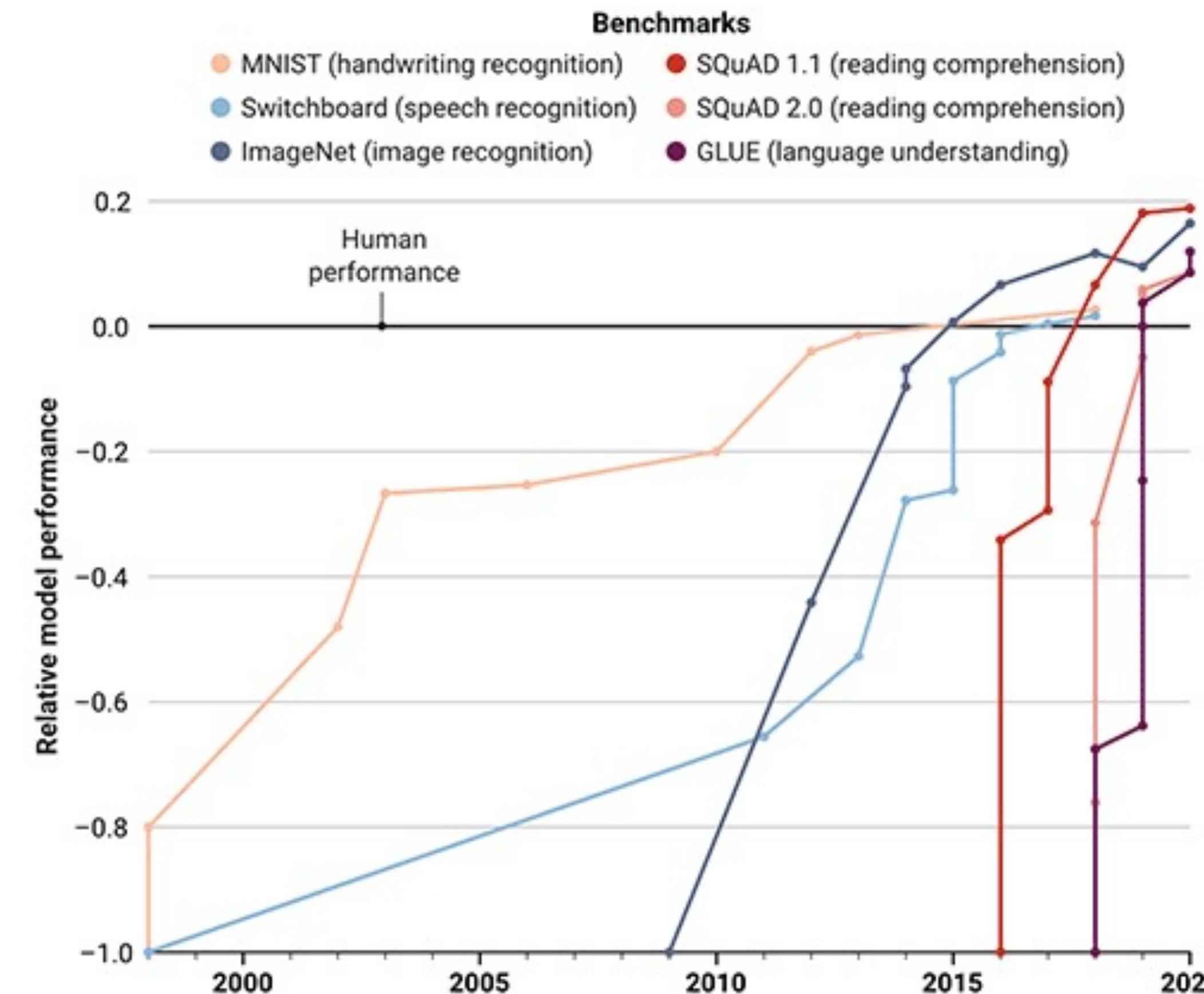
Floating-point operations, selected systems, by type, log scale



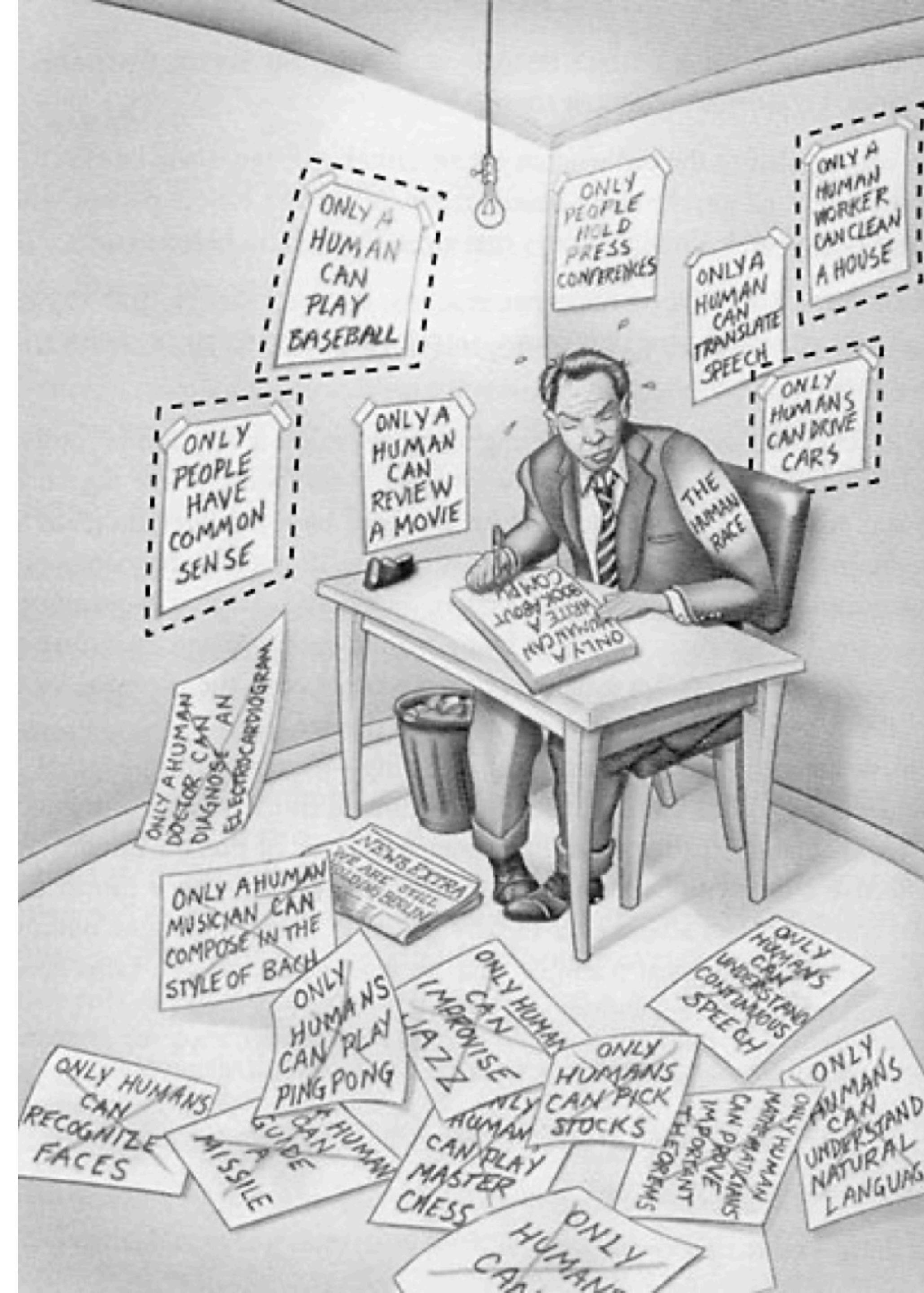
Sources: "Compute trends across three eras of machine learning", by J. Sevilla et al., arXiv, 2022; Our World in Data

## Quick learners

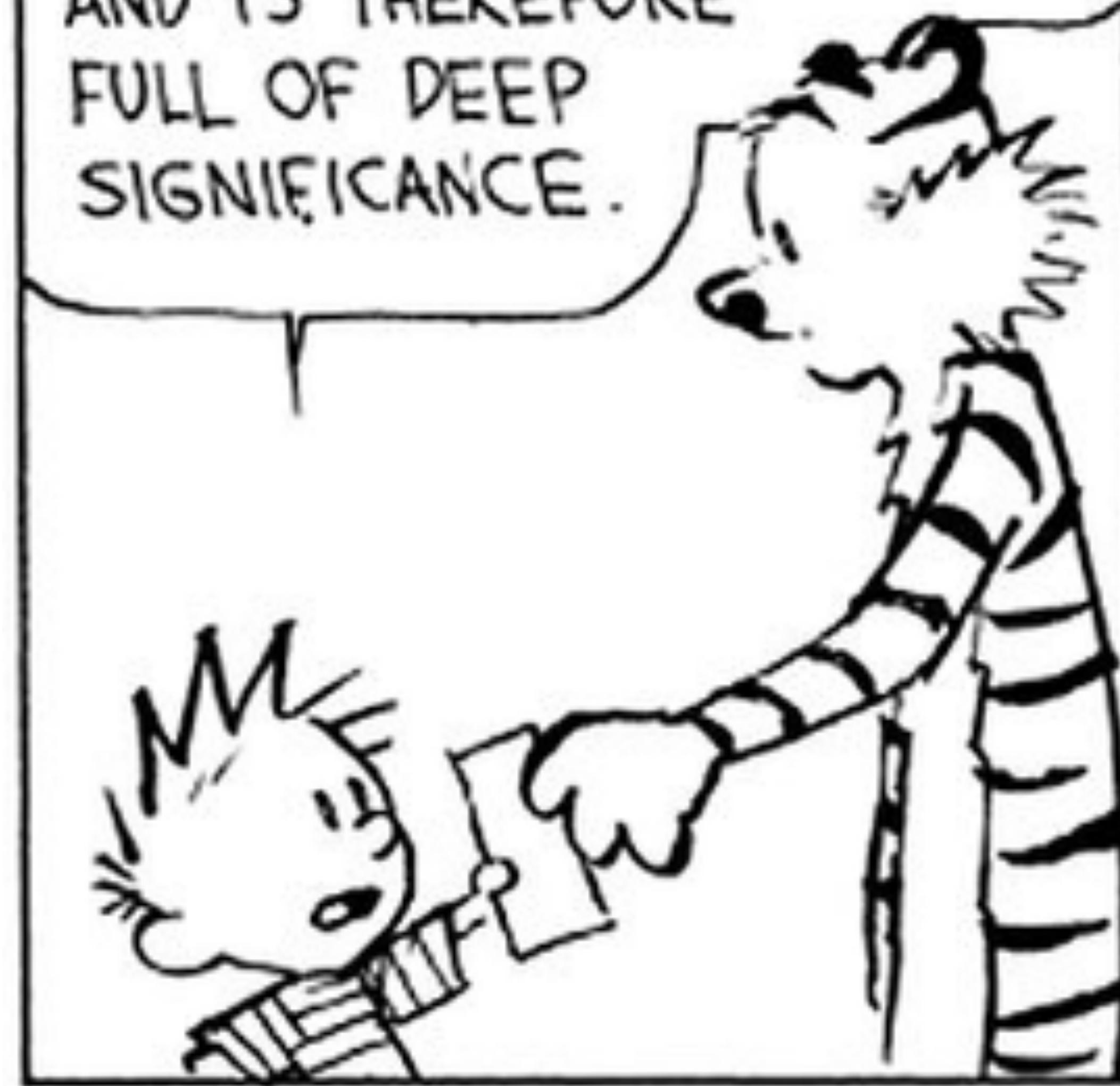
The speed at which artificial intelligence models master benchmarks and surpass human baselines is accelerating. But they often fall short in the real world.



(GRAPHIC) K. FRANKLIN/SCIENCE; (DATA) D. KIELA ET AL., DYNABENCH: RETHINKING BENCHMARKING IN NLP, DOI:10.48550/ARXIV.2104.14337



AS MY ARTIST'S STATEMENT  
EXPLAINS, MY WORK IS  
UTTERLY INCOMPREHENSIBLE  
AND IS THEREFORE  
FULL OF DEEP  
SIGNIFICANCE.



# Complexity 1990 - SVM

$$\max_a \sum_{i=1}^m a_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m a_i a_j y_i y_j K(x_i, x_j)$$

$m$  number of observations

$a$  counter for errors

$$K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$$

$x$  observations

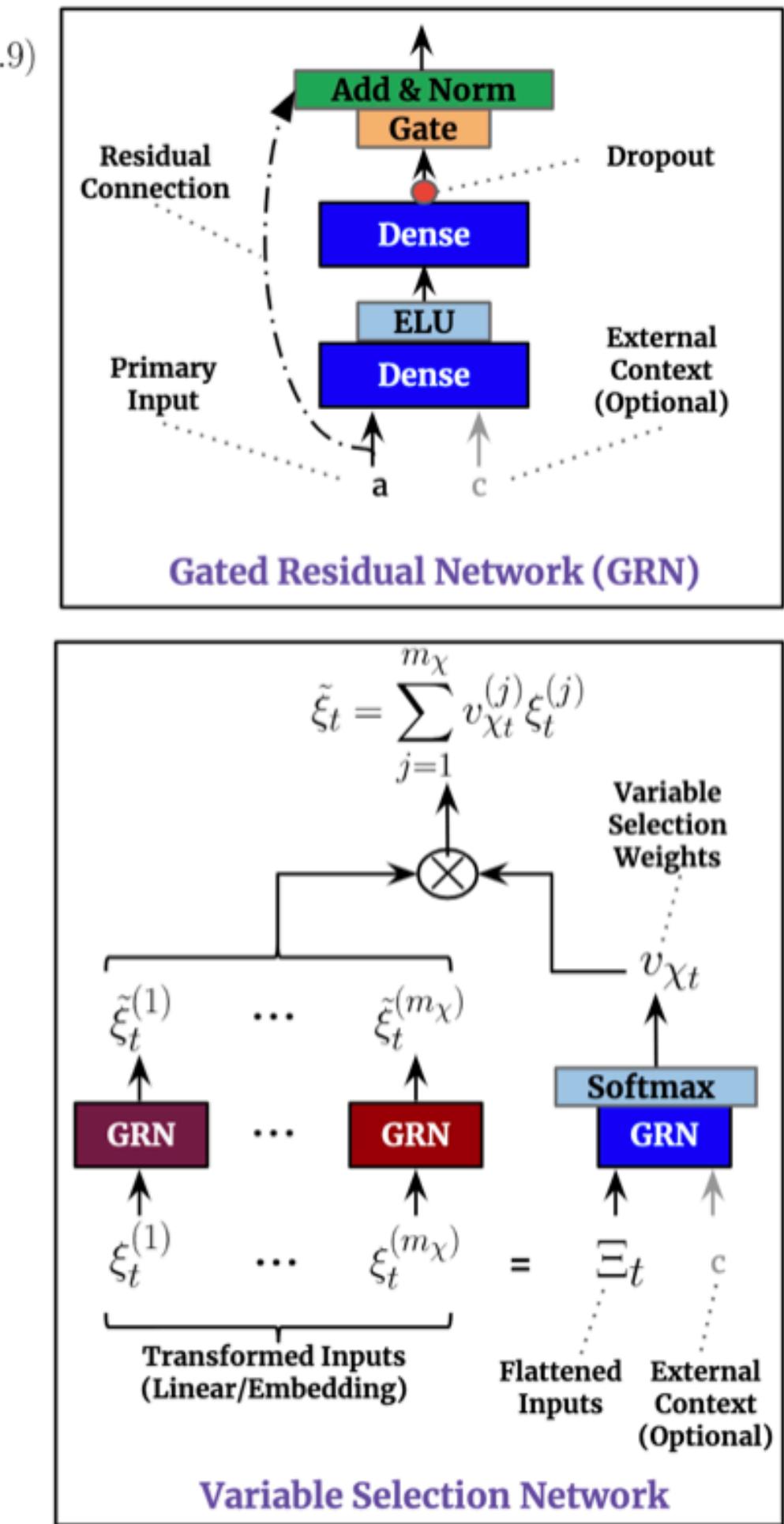
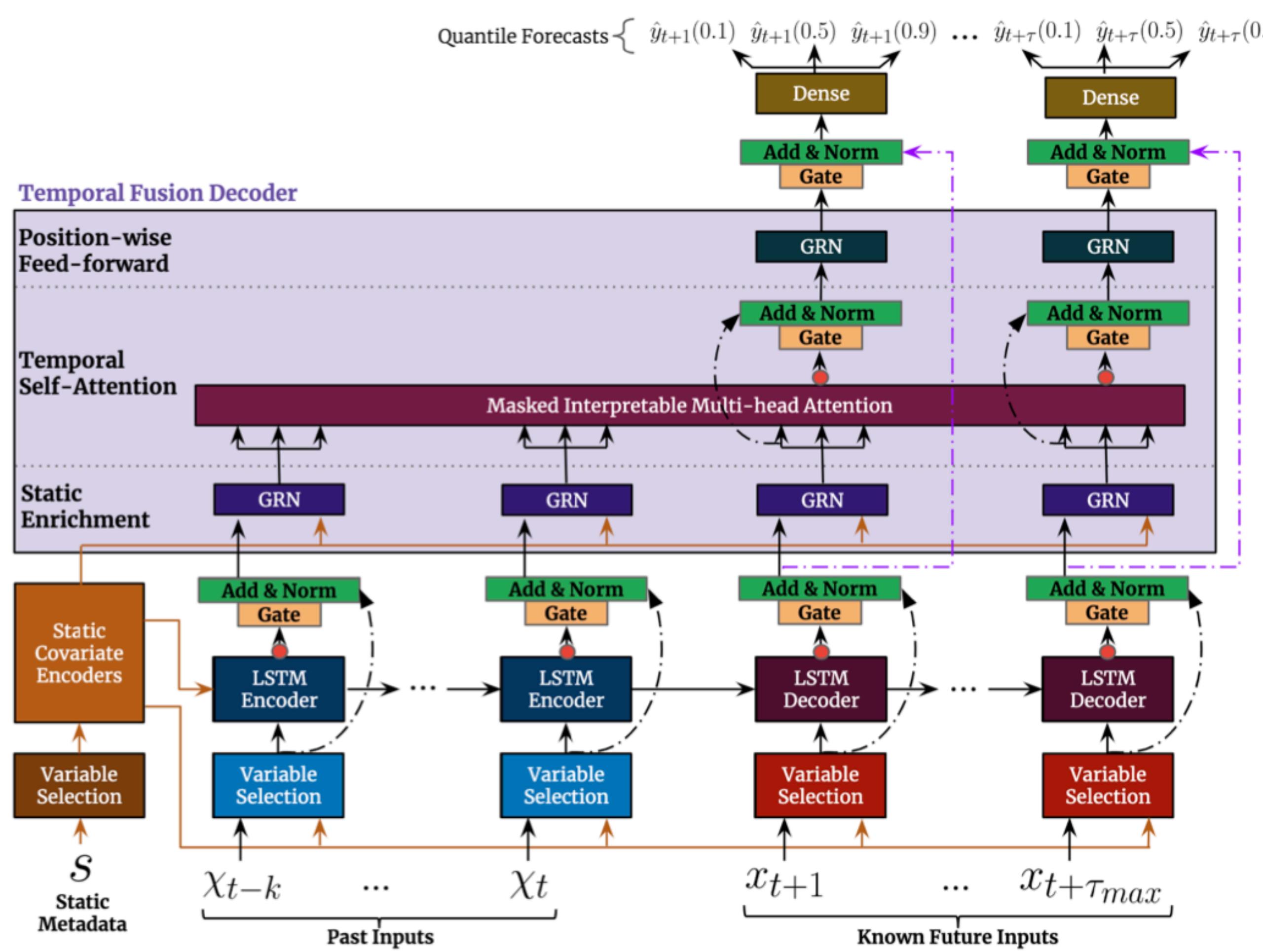
$$J(w, b) = \frac{1}{2} w^T w + C \sum_{i=1}^m \max(0, 1 - t^i(w^T x^i + b))$$

$y$  labels

$\gamma, C$  hyperparameters

$w, b$  weights we need to learn

# Complexity 2021 - TFT



# Examples

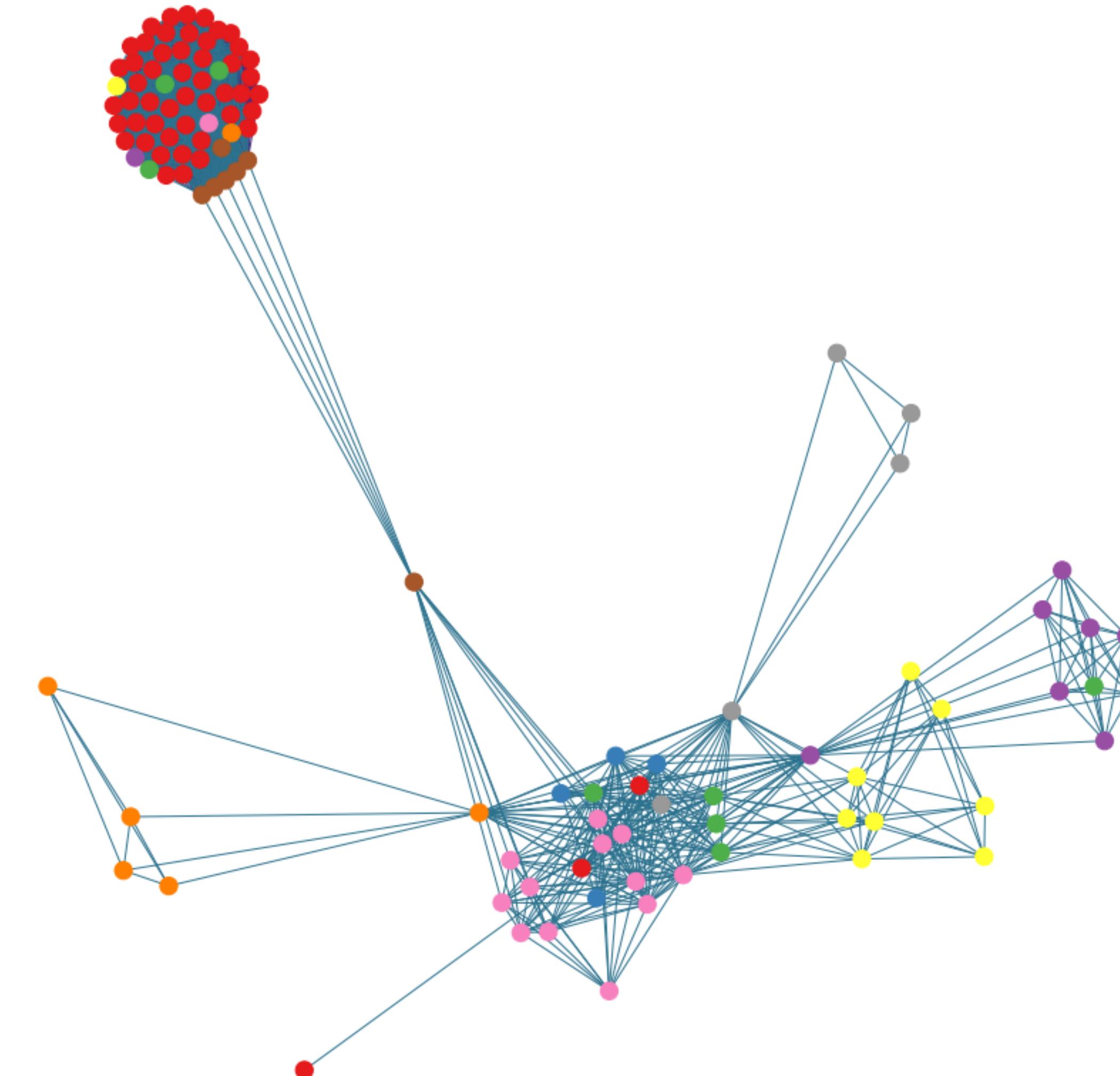
## Graphs in factories

Anillo Lleb KHS 301104.69.104/0 (local\_material\_id)

Problem:

Relocate 30.000 items within 6 weeks to European factories.

Solution: Locate similar machine parts in a database of 300.000 items, with unstructured and noisy descriptions in 10+ languages.

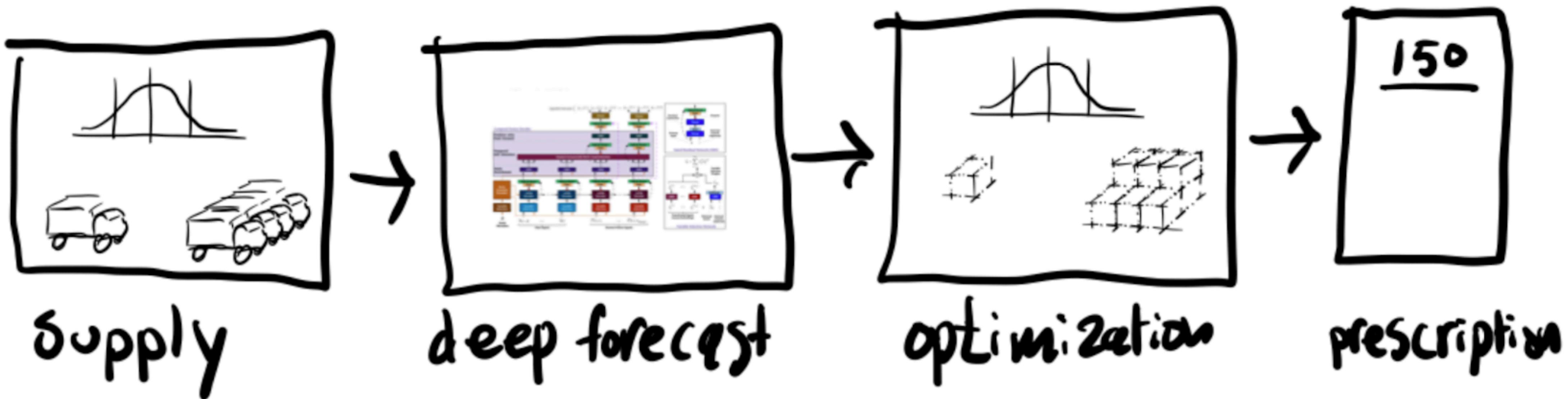


# Examples

## Probabilistic Stock Optimization

Problem: find optimal stock levels, while supply and demand have a lot of uncertainty.

Solution: build a probabilistic prescriptive model



# Main libraries

- Numpy
- Pydantic
- Pandas / Polars
- PyTorch
- Ray, Tensorboard, MLFlow
- Black

# Code style standards

- Use `pathlib.Path` instead of `os.path`
- Use `loguru` instead of `print`
- Use `pydantic` for dataclasses and settings
- Use typehinting and linting
- Organize your code with classes
- Create custom types with `Protocol` or `pydantic`
- Manage your environment with `poetry`, not `conda`
- Use the data science cookiecutter template

See <https://staticsitecodestyle.z6.web.core.windows.net/tldr.html>

# Linear models

Even simpler models are called “retarded”

Linear models are one of the simplest models available. Their advantages are:

- The simplicity helps us to avoid overfitting
- They are fast and we don't have to worry about getting stuck in a local minimum.
- They are ideal as a baselinemodel

# Linear models

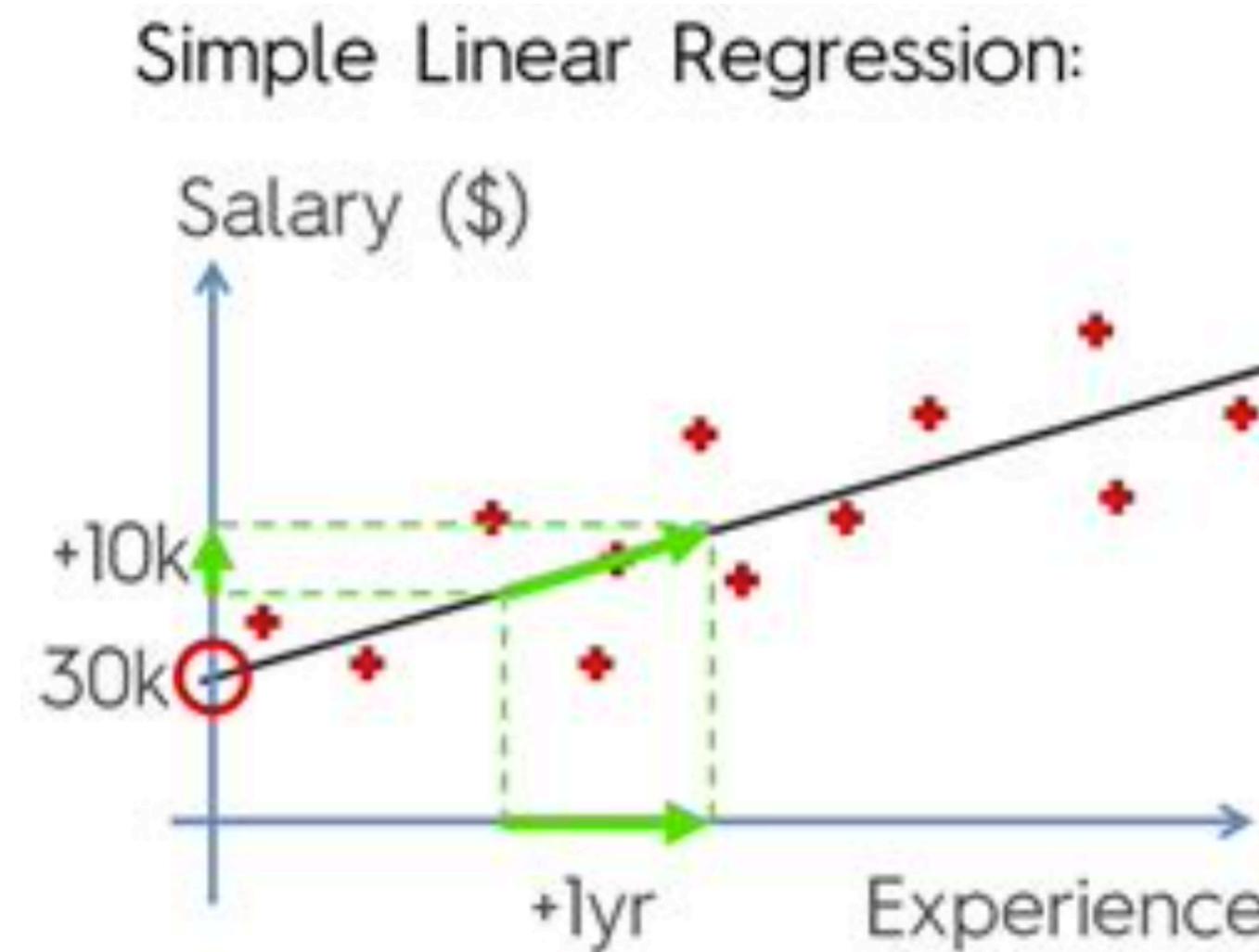
The basic mathematical shape of a linear model is:

$$Y = WX + b$$

- $Y = \{y_1, \dots, y_m\}$  are labels, so we have  $m$  labels.
- Every label  $Y_i$  has corresponding features  $X_i = \{x_{i,1}, \dots, x_{i,n}\}$ , so we have  $n$  features.
- We can store the observations in an  $(m, n)$  matrix  $X$
- We can store  $n$  weights for every feature in a matrix  $W$
- $b$  is an additional bias weight

The matrix notation is concise. We could write this out in full as:

$$y_i = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b$$



$$\begin{array}{c} y = b_0 + b_1 x \\ \downarrow \\ \text{Salary} = b_0 + b_1 \cdot \text{Experience} \end{array}$$

# Hyperplanes

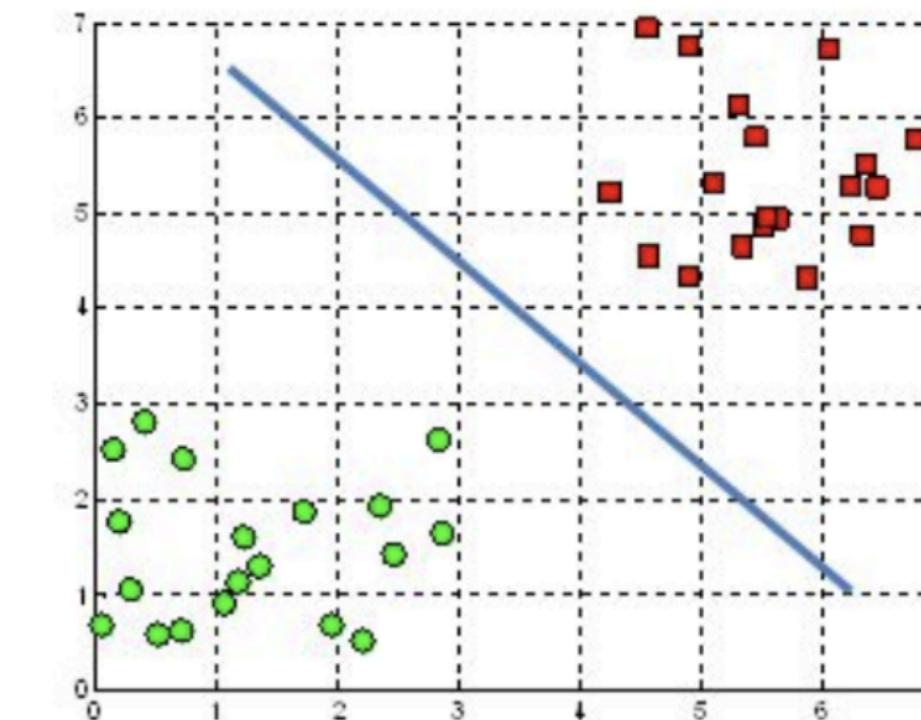
It's a bird... It's a plane...

- For classification, our outcome are discrete classes (e.g. “yes” or “no”)
- For regression, our outcome is a real number (e.g. 1.4 or 26.834)

We can use a linear model for both cases. We call these models “hyperplanes”: they have one dimension less than the ambient space.

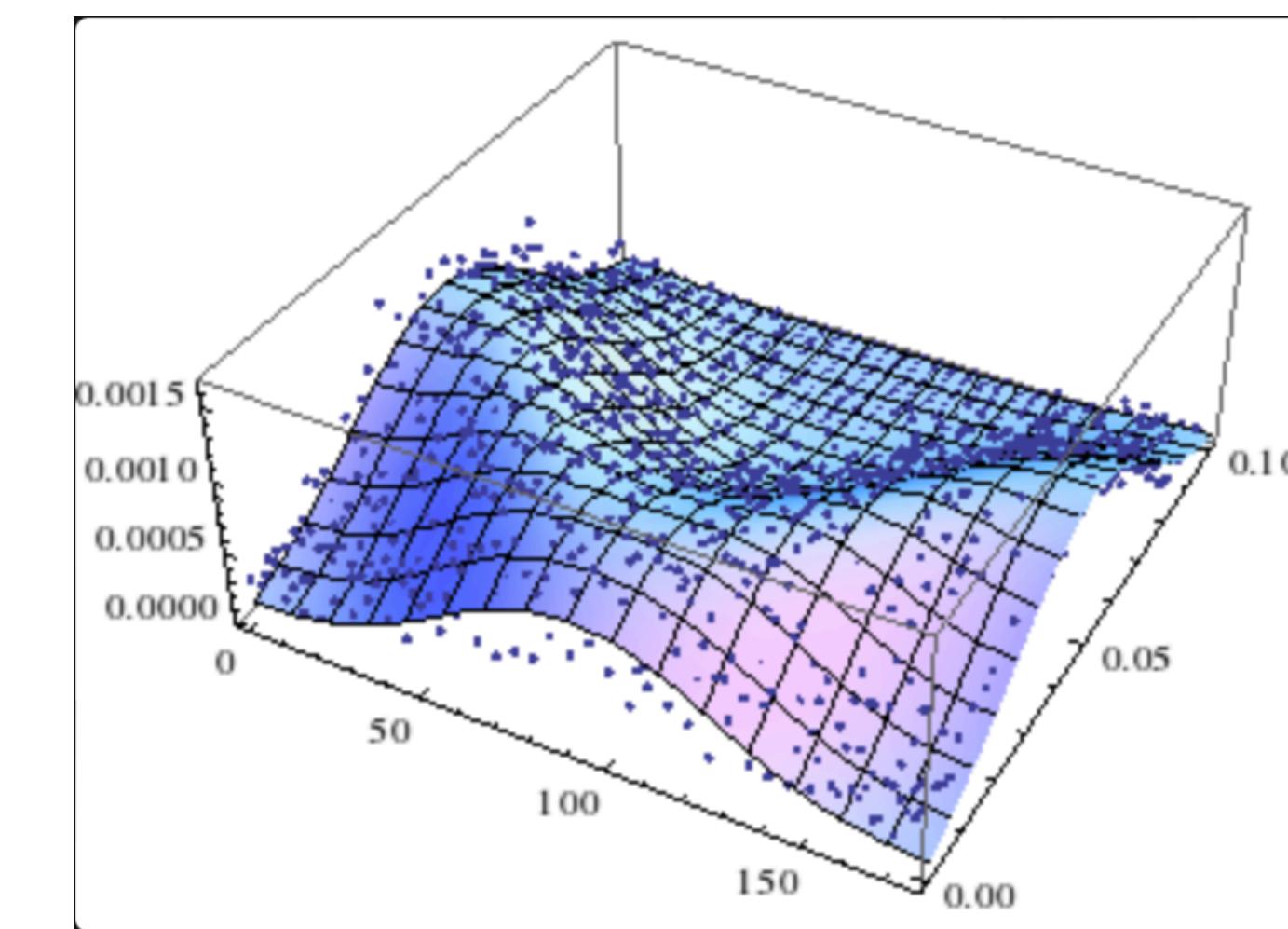
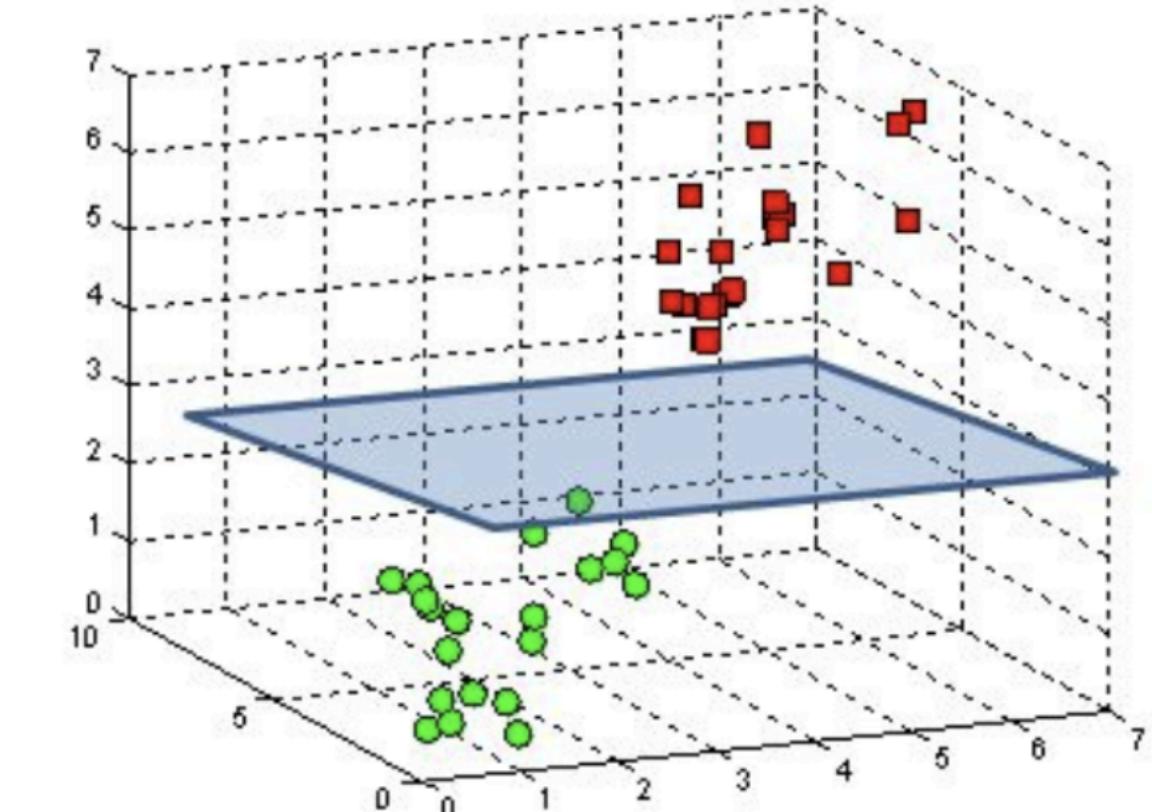
With classification, we want the data to be separated by the hyperplane. With regression, we want points to get as close as possible to the hyperplane

A hyperplane in  $\mathbb{R}^2$  is a line



Hyperplane for classification

A hyperplane in  $\mathbb{R}^3$  is a plane



Hyperplane for regression

# Non-linear models

A lot of data is non-linear. This means we would need a curved hyperplane.

One trick to do this is the kernel-trick, which is commonly used with Support Vector Machines, which we touched upon in the short history.

Deep learning has found another trick, we will look at the labs to see how this works.

