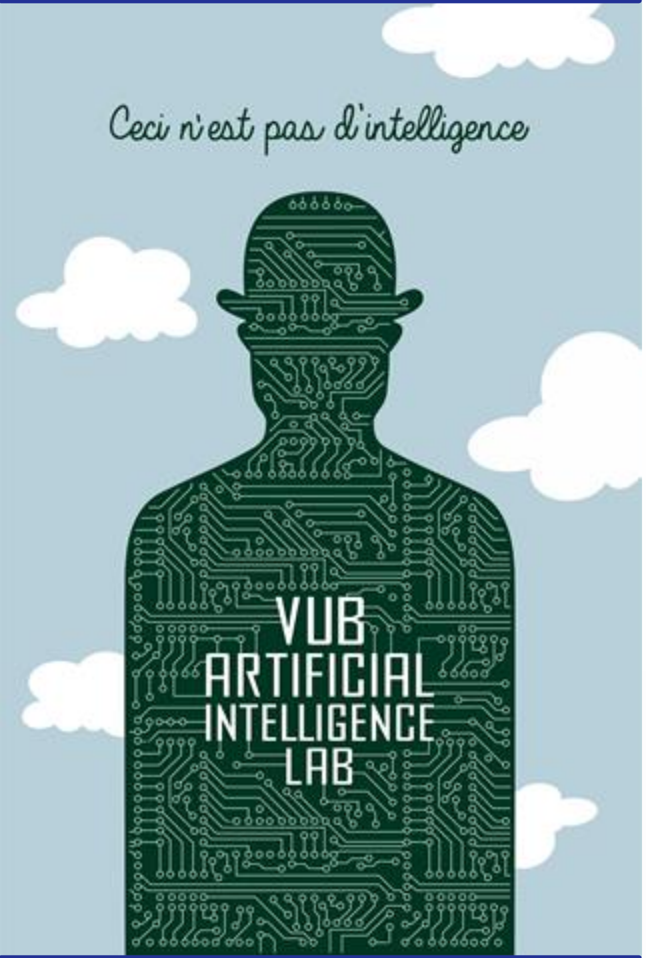


Advice to young AI scientists

Current Trends in AI

13 lessons learnt

Johan Loeckx
jloeckx@ai.vub.ac.be

ai.vub.ac.be
@aibrussels

VUB ARTIFICIAL INTELLIGENCE RESEARCH GROUP

- Add MATHS!

Exercise #1

- Is ChatGPT3 **intelligent**?
- How does ChatGPT3 deal with deciding... **relevance**?
- Can ChatGPT3 perform **common sense** reasoning?
- How does it model domain knowledge? E.g. physical **knowledge**?

#1 – Be critical

We are all part of a culture that boosts learning but also often inhibits fundamental progress.

Exercise

- Everyone tell one thing about/related to chairs

- How would you programme a machine to answer such questions?

#2 - Be humble.

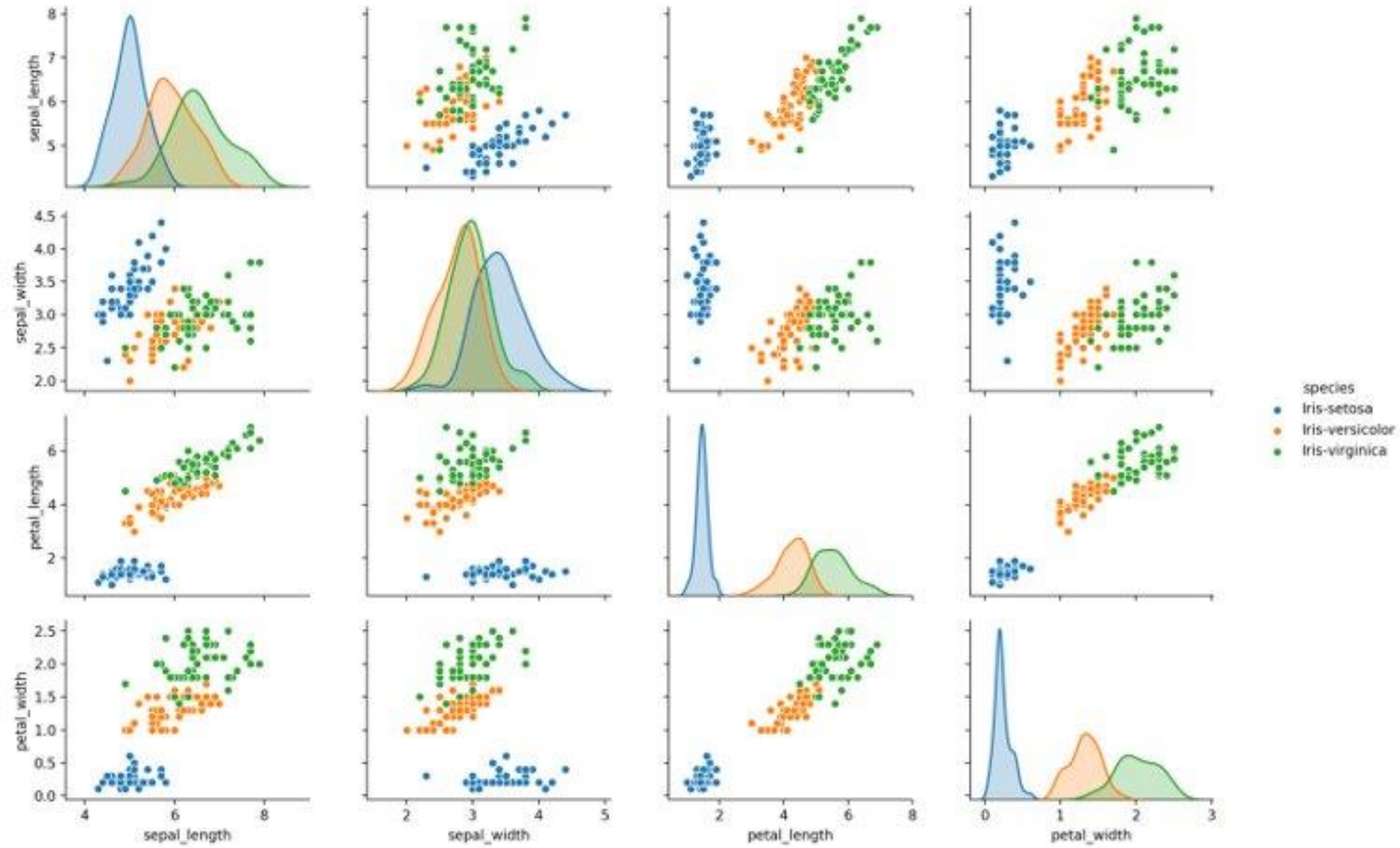
We are *nowhere* yet, there is still a LOT to investigate.

Exercises

- How does Chat GPT3 represent a chair?
- Design a neural network to divide two numbers.
- Design a **minimal** neural network **by hand** that models XOR
  - Hard limiter activation function
  - Two teams: 1) bits encoded as (-1, +1), 2) bits encoded as (0,1)

#3 - Representations are key

The Iris dataset...



ARTIFICIAL INTELLIGENCE RESEARCH GROUP

Exercise

- What do('n t) you know about the Iris dataset?
  - How were the measurements performed?
  - How do you define "length" and "width" of a leaf
  - Where were the samples collected? In one place?
  - By one person? In one year?
  - What is the meaning of the labels?

#4 - Try to bring *real impact*

Reality is complex, problems are ill-defined, everyone talks a different language.

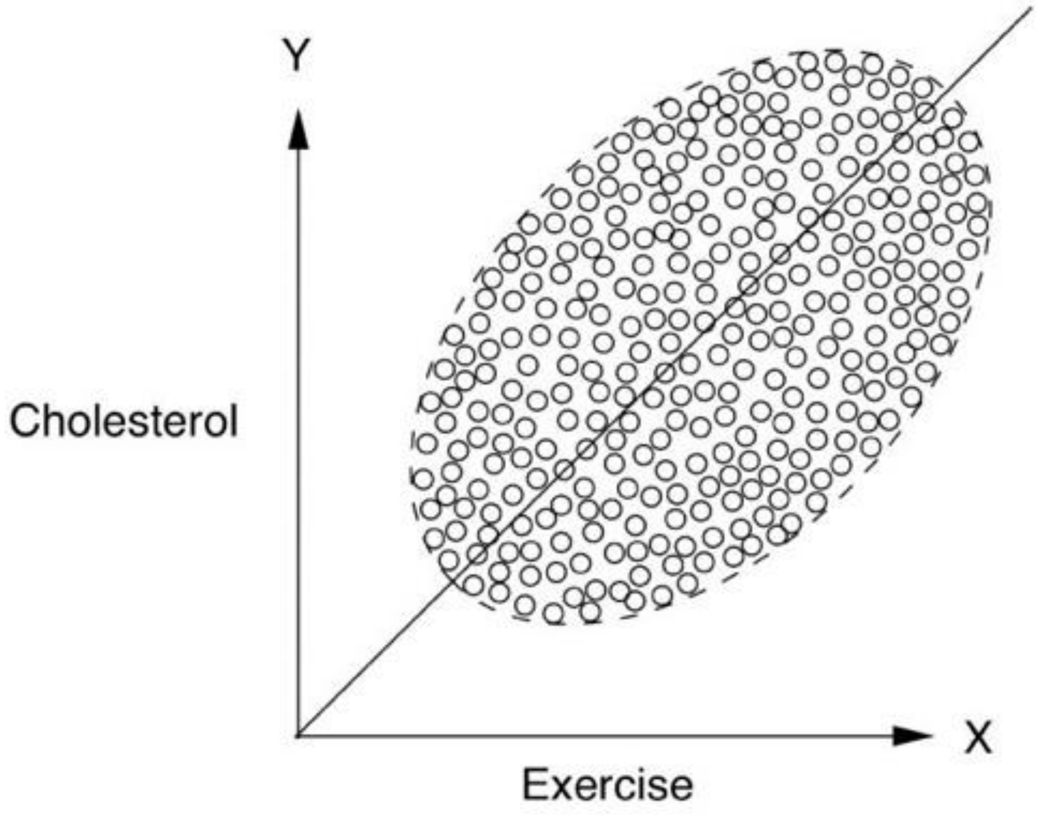
And intriguing!!!

But start with the lowest possible goal.

You can always increase complexity if you move *faster than expected**.

*(*this never happens)*

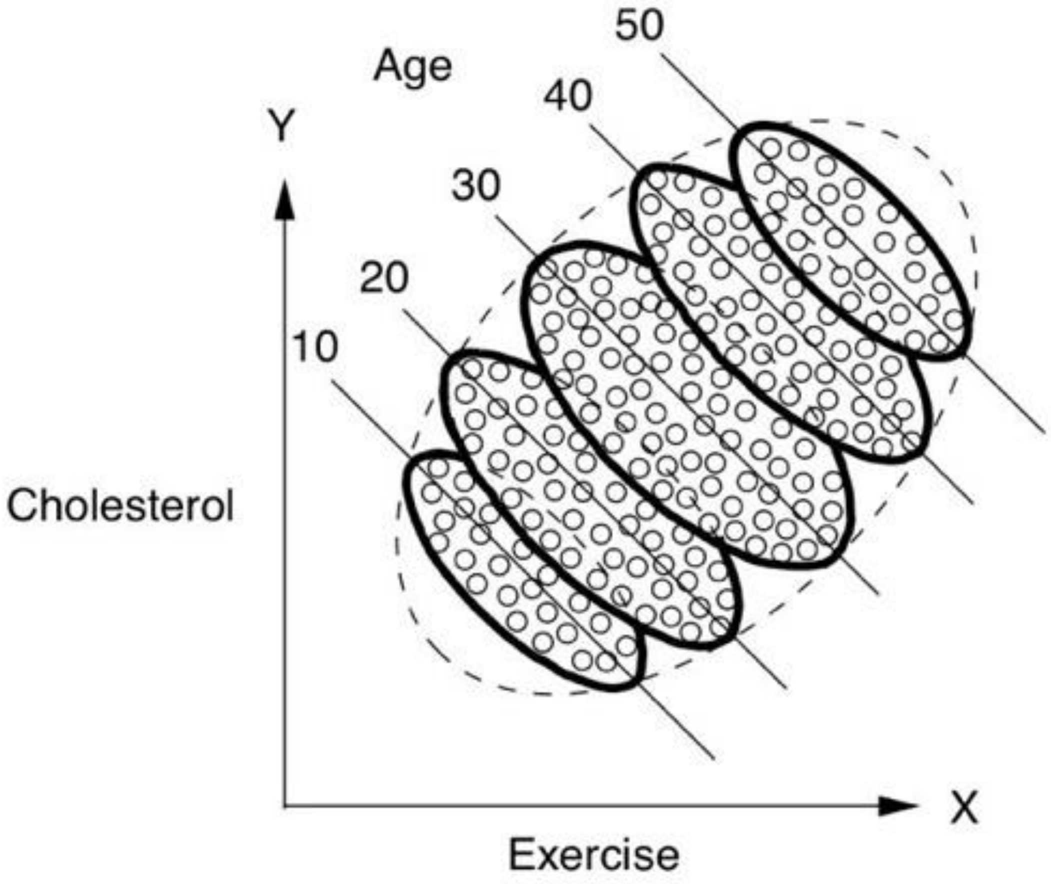
Exercise leads to higher cholesterol levels

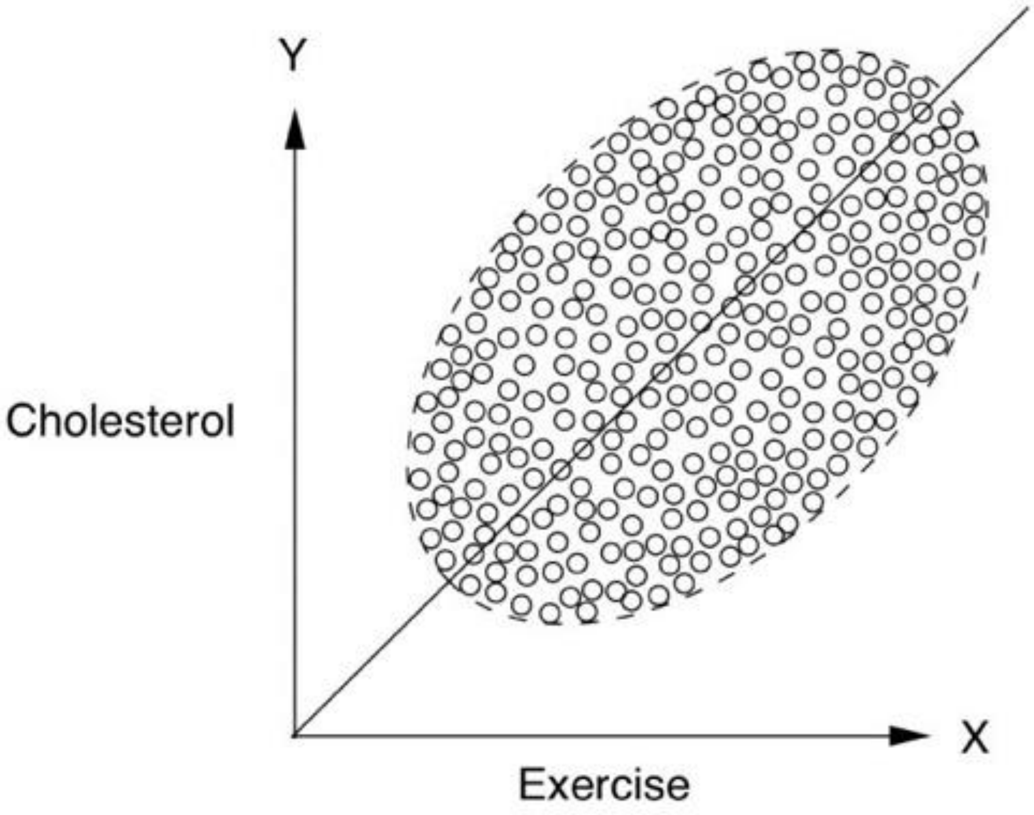


Causality

- Age
  - Has a causal link to exercise
  - And potentially to cholesterol
- Age is thus a confounder of exercise and cholesterol
- So we should control for it!

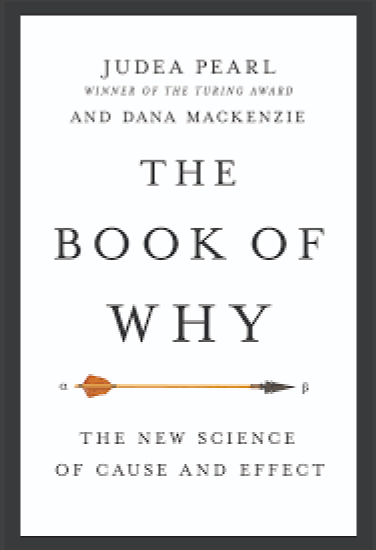
Exercise leads to lower cholesterol levels





#5 – Remain curious

Education starts when you graduate.

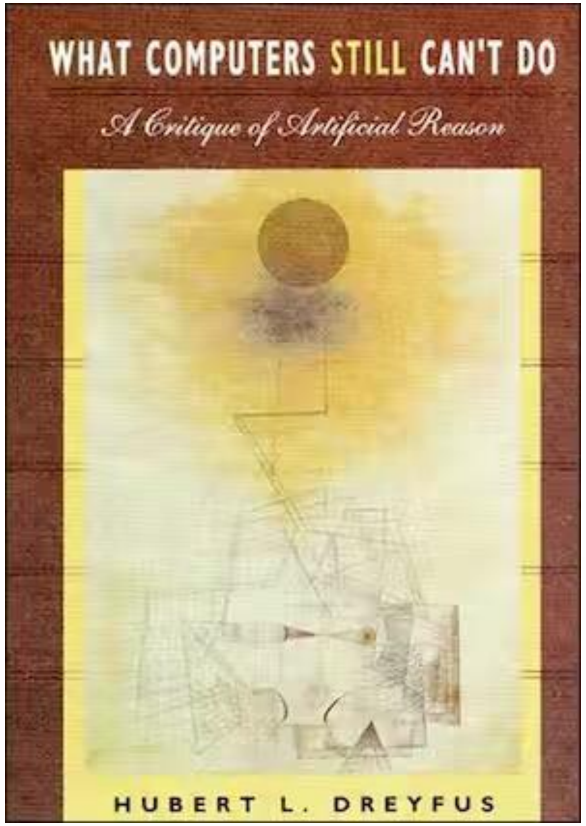


Quiz

- When was the term "Machine Learning" designed?
  - 1959 by Arthur Samuel
- When was Object Oriented Programming (+/-) invented?
  - Late 50ies – in the context of LISP programming
- When were CNNs invented?
  - 50ies, 80, 89, …
- Who can be considered the pioneer of NNs?
  - Frank Rosenblatt, a psychologist. Perceptron: 1957

New is really not always better

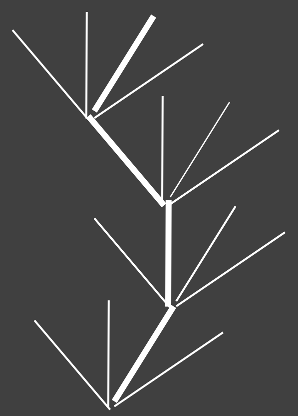
- A lot of garbage
- Big ideas come slowly
- Publication pressure
- Roots are great sources of insipiration
  - How did the original researchers come to this idea?



#6 -Go back to the roots

- Read the classics!
- Find the original ideas.
- Research = a very asymmetric tree
- Unexplored ideas are often valuable
- **Follow people** first, then papers.

The thick line is what they teach you



#7 - Break the cat

Romanian saying. Start with what you fear the most

You think it will be too slow, you don't know how to implement this piece, ...?

Exercise

- The **stable matching problem**, in its most basic form, takes
  - as input equal numbers of two types of participants (n medical students and n internships, for example),
  - an ordering for each participant giving their preference for whom to be matched to among the participants of the other type.
- A matching is *not* stable if:
  - there is an element *A* of the first matched set which prefers some given element *B* of the second matched set over the element to which *A* is already matched, and
  - *B* also prefers *A* over the element to which *B* is already matched.

Gale-Shapley algorithm

```
Initialize all men and women to free
while there exist a free man m who still has a woman w to propose to
{
    w = m's highest ranked such woman to whom he has not yet proposed
    if w is free
       (m, w) become engaged
    else some pair (m', w) already exists
       if w prefers m to m'
          (m, w) become engaged
           m' becomes free
       else
          (m', w) remain engaged
}
```

#8 - Problem complexity ≠ Solution complexity

Problems that can be formulated in simple terms, can have complex solutions. And vice versa.

DO NOT confuse the solution with the problem!

#9 - Move slowly. Slower...

Take more time to understand the problem than the solution.

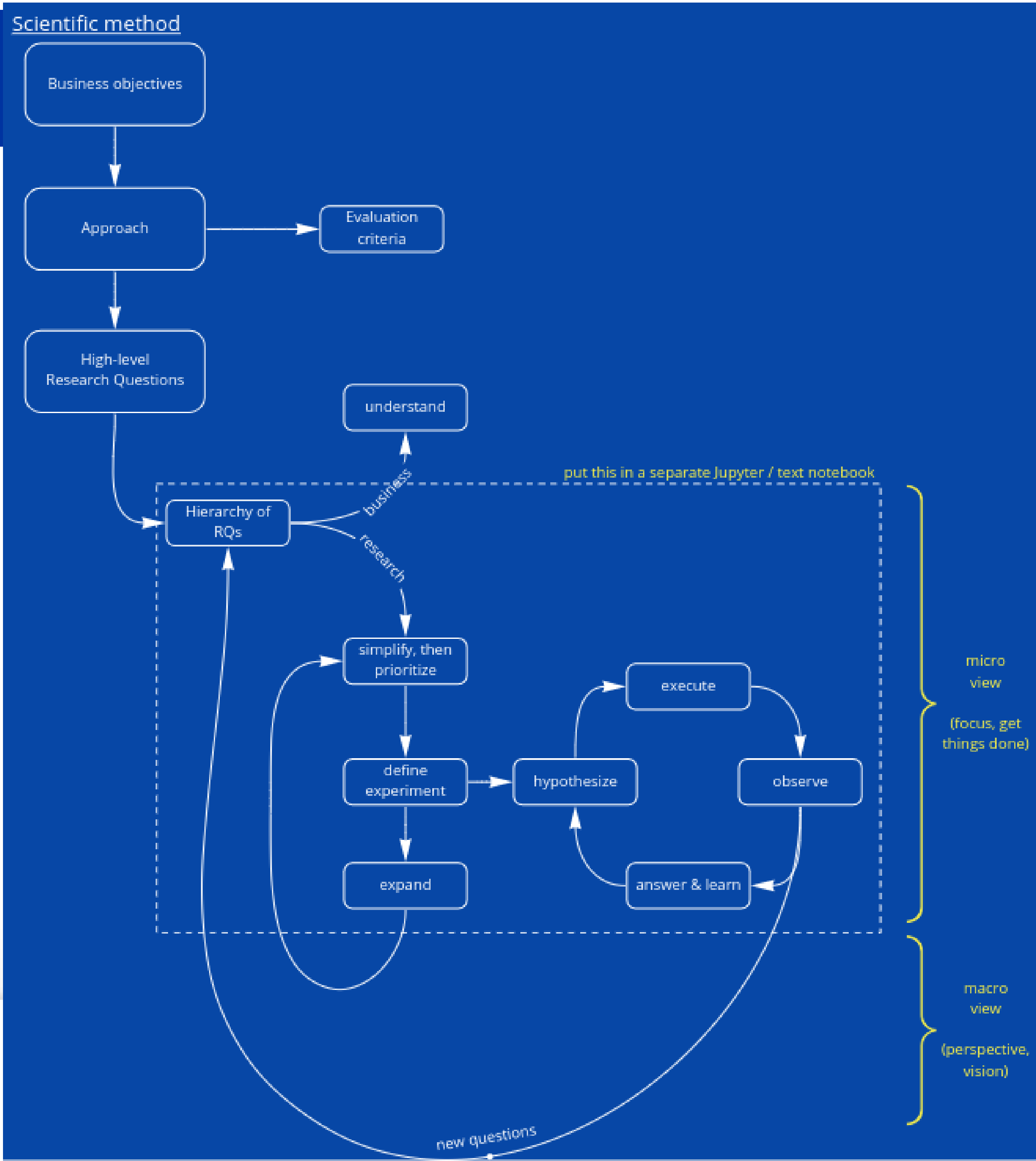
"Once you *really* understand the problem,
you're probably smart enough to find a solution"
(Michiel Steyaert)

#10 - On methodology

How to cope with too complex problems?

The scientific method

1. Define questions
2. Think how to answer them
3. Code / perform experiment
4. Learn & update



Coming up with questions

- **Don't assume** you know: turn every certainty into a (hierarchy) of research question(s).

- Simplify questions & scope till it seems **trivial** to implement (which it will not be)
- What is the **simplest next** experiment you can think of from which we can learn something?
- To finally arrive to an algorithm/implementation that can be evaluated with a ground truth

Coming up with questions

- An experiment can be any activity that provides you with ANSWERS: building a prototype and measuring outcomes, literature review, strength & weaknesses, brainstorm for new scientific ideas, having a conversation with the customer

- Some thoughts how to generate **hypotheses**: (a) state the obvious (better too many than too few), (b) verbalize you intuition: describe what you think and rephrase in questions, (c) operationalise definitions, verbs

Organising results

- For each question, create a **notebook** / document with a fixed structured and a section for:
  - (a) the question + more specific description,
  - (b) the experiment you construct to answer it (and support it),
  - (c) hypotheses: what do you expect to see in the answers,
  - (d) the implementation itself,
  - (e) observations from the results,
  - (f) link back the answer, and anything else you have learnt

#10 - Build or think?

Decide carefully when you need to think things through, and when you need to build!

Build or think?

- Build
  - When you do not know all factors influencing the problem/solution
  - When there are too many assumptions
  - When moving to the *solution phase*
  - Because it forces you to go all the way
  - Because we always underestimate the complexity of things
- Think
  - When you need to get things **tight**.
  - When you start to get a **full** picture
  - To keep things **clean**
  - After building
  - To get a **fresh** mathematical view

Grasping a tomato

- Would you consider that an AI task?

Social robotics

- Playing cards
- Looking into the eyes

#11 - Learn from building

- Traverse bottom-up and top-down
- End-to-end
- Measure bottlenecks, do not assume them

#12 - Go for *elegance*

The problem with extrinsic evaluation

- If you evaluate the correctness of an answer only
  - System A learns everything by heart
  - System B learns the underlying structure

- Which system is the "smartest"?

- AI is heavily influenced by behaviourism (Turing!). Not necessarily a good thing...

Is a traffic sign AI?

- It's technology...
- And it makes us smarter!

#13 - Don't forget embodiment

Questions?