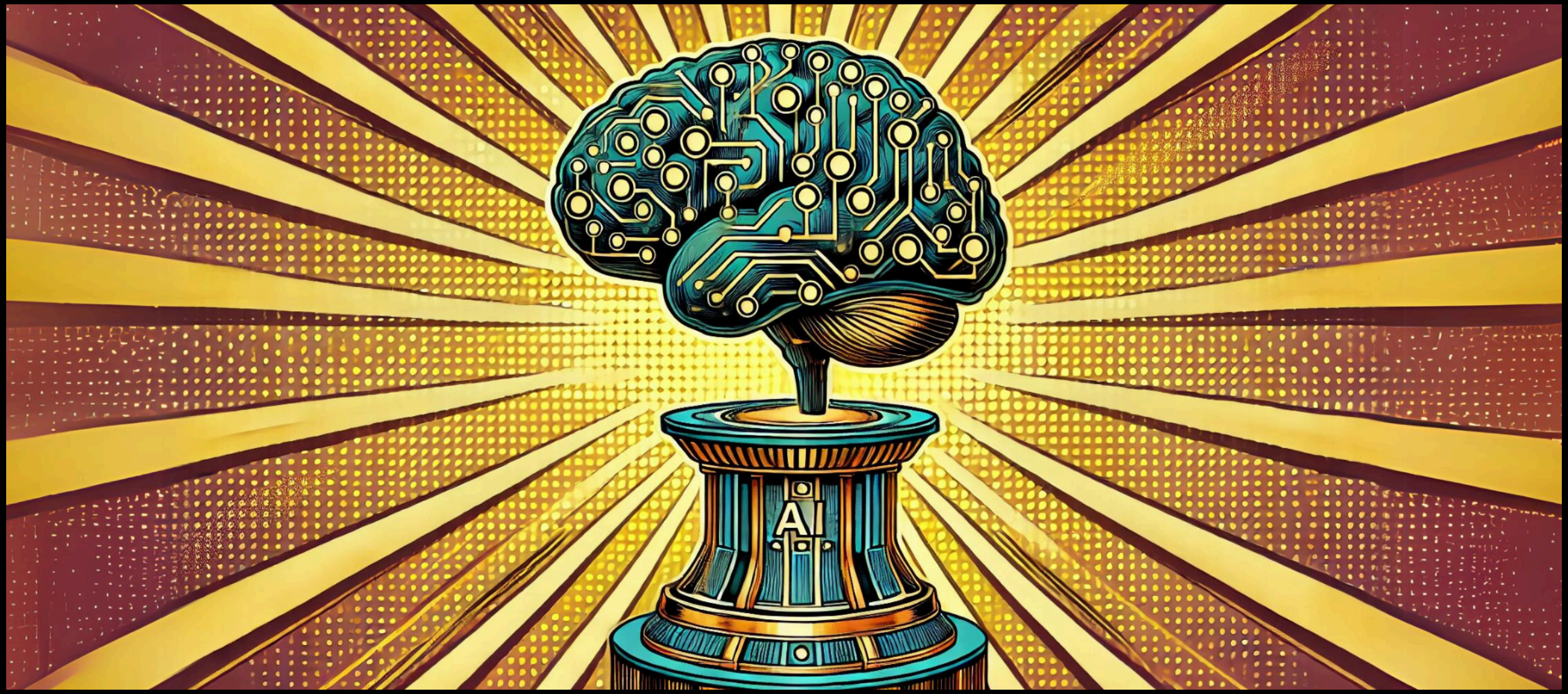
LLM Engineering

MASTER AI & LARGE LANGUAGE MODELS



PROGRESS REPORT

Just before we get into coding, let's take stock of the process

What you can now do

- Generate text and code with Frontier Models including AI Assistants with Tools, and with open-source models with HuggingFace transformers
- Confidently choose the right LLM for your project, backed by metrics
- Create advanced RAG solutions with LangChain
- Select, investigate and curate a Dataset

After today you'll be equipped with new important skills

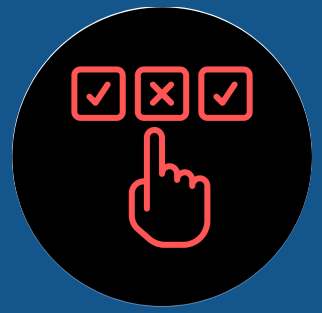
- Lay out a 5 step strategy for selecting, training and applying an LLM
- Contrast the 3 techniques for improving performance
- Give common use cases for each of the techniques

5 Step Strategy

To selecting, training and applying an LLM to a commercial problem

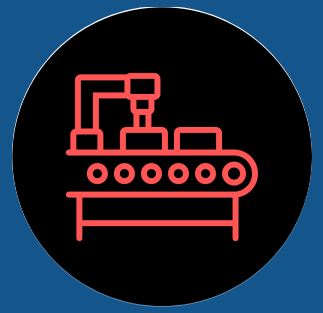
Understand

Prepare



Select

Customize

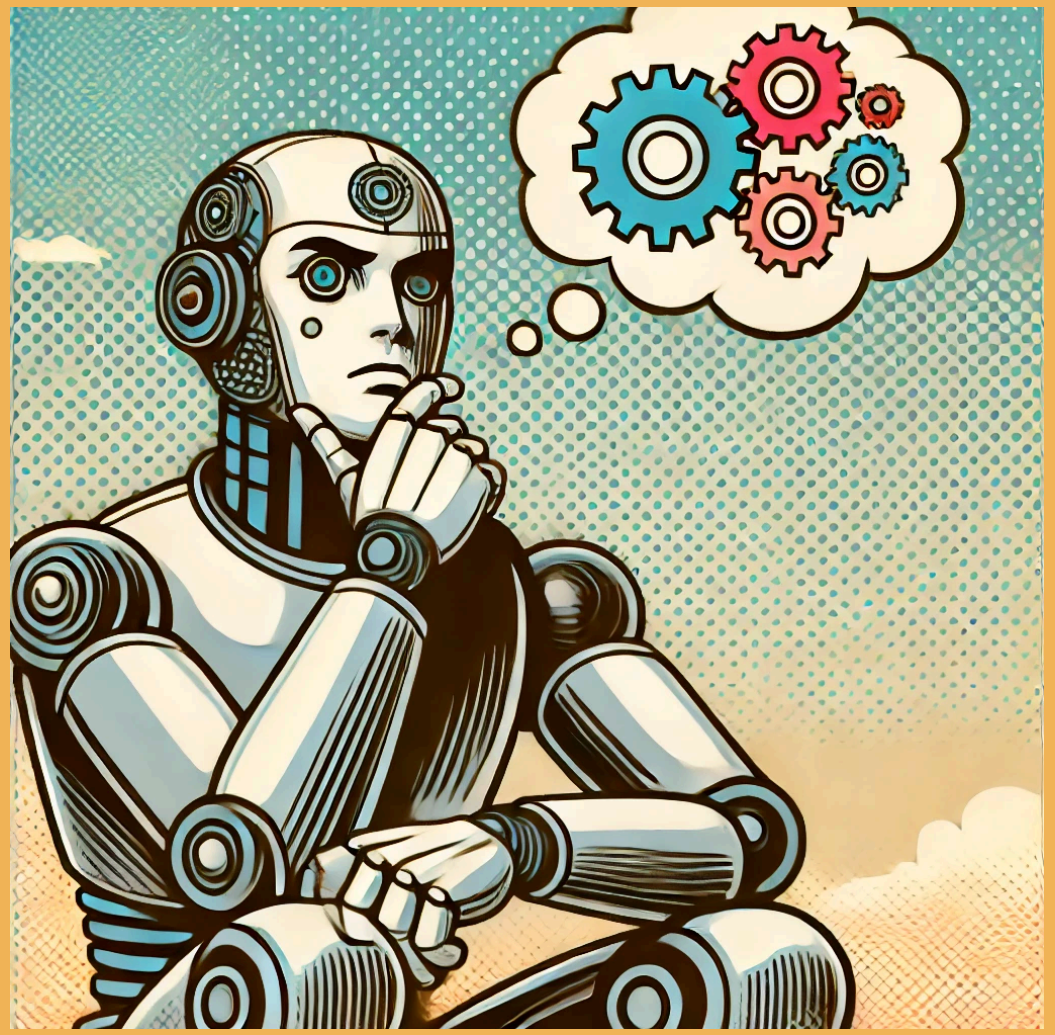


Productionize

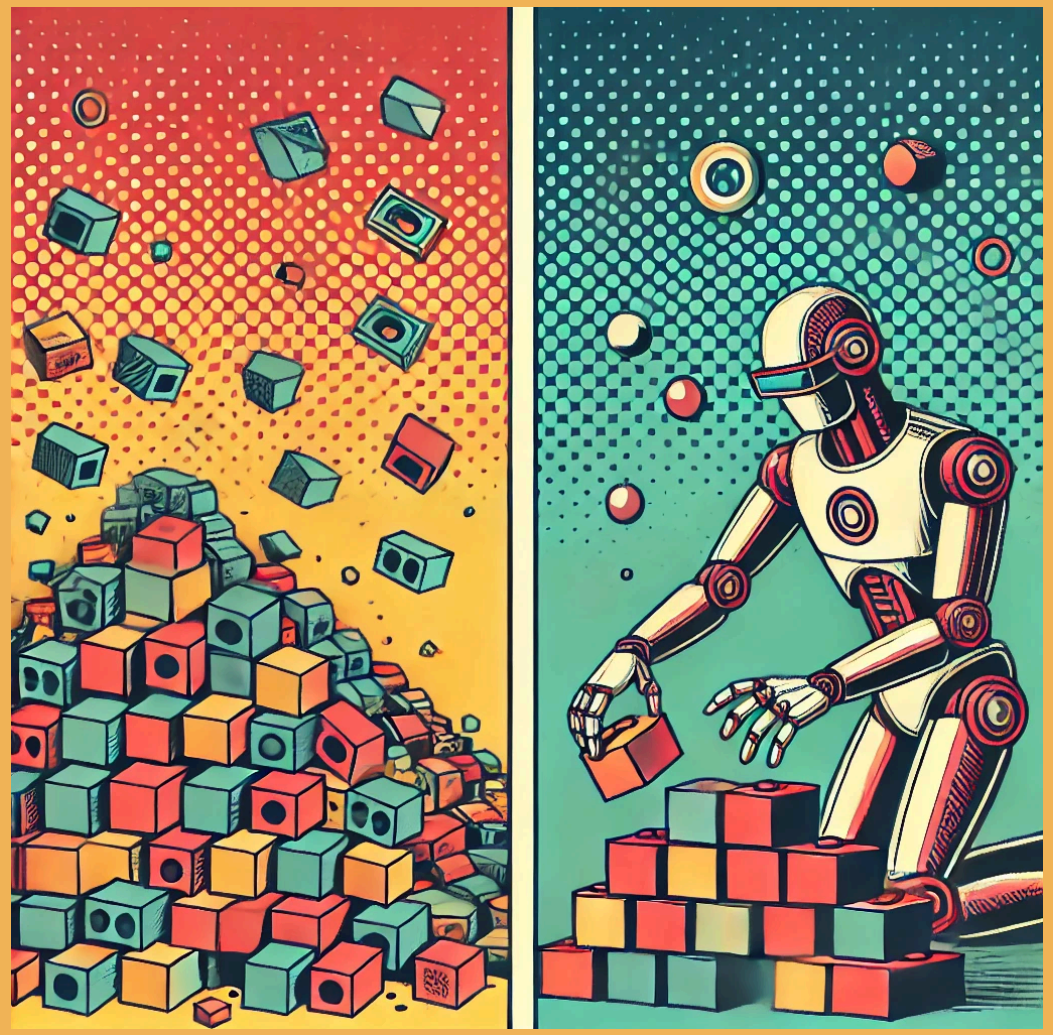
1. Understand

Activities:

- Gather business requirements for the task
- Identify performance criteria
  Particularly the Business Centric metrics
- Understand the data: quantity, quality, format
- Determine non-functionals
  Cost constraints, scalability, latency
  R&D / build budget and implementation timeline



2. Prepare



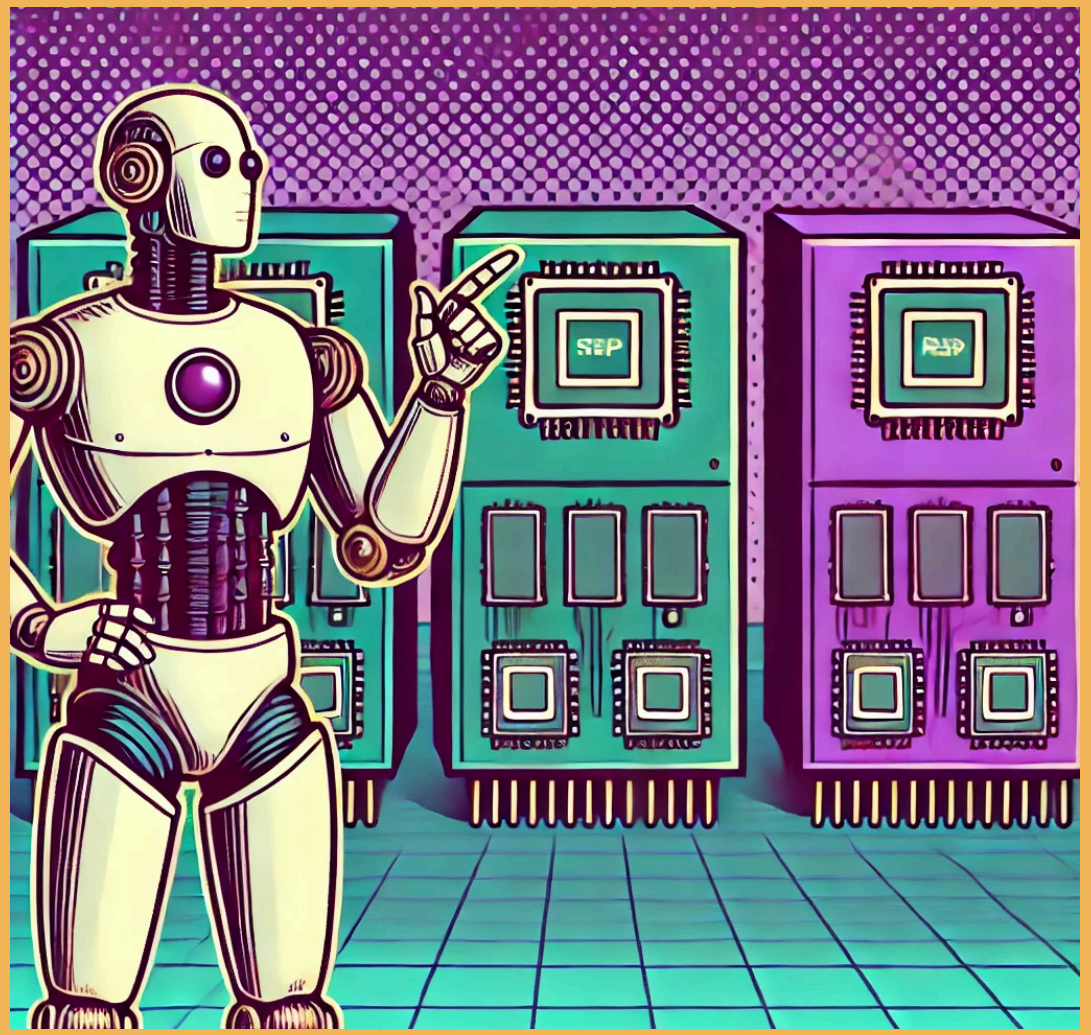
Activities:

- Research existing / non-LLM solutions
  - Potential baseline model
- Compare relevant LLMs
  - The basics, including context length, price and license
  - Benchmarks, Leaderboards and Arenas
  - Specialist scores for the task at hand
- Curate data: clean, preprocess and split

3. Select

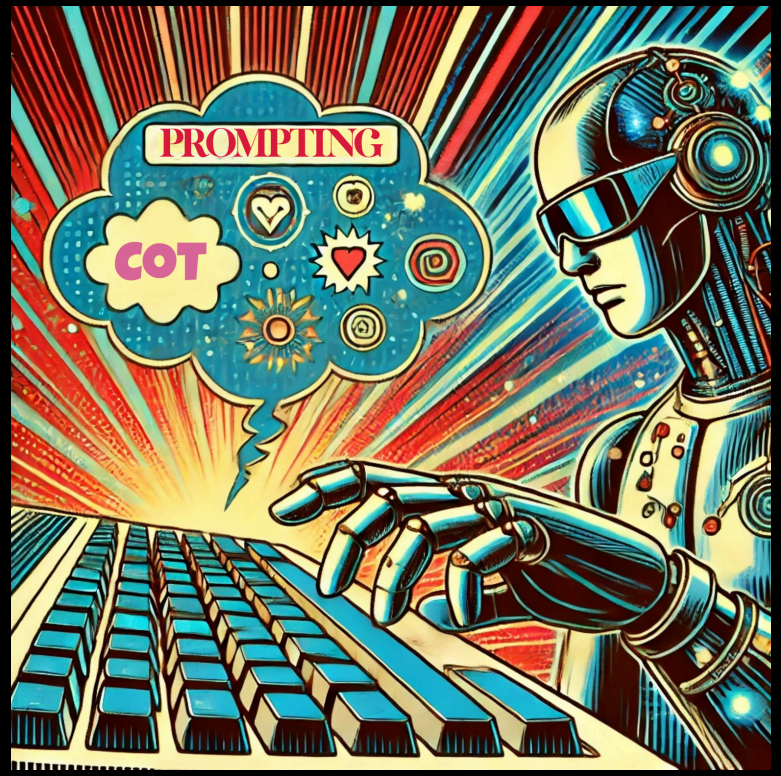
Activities:

- Choose LLM(s)
- Experiment
- Train and validate with curated data



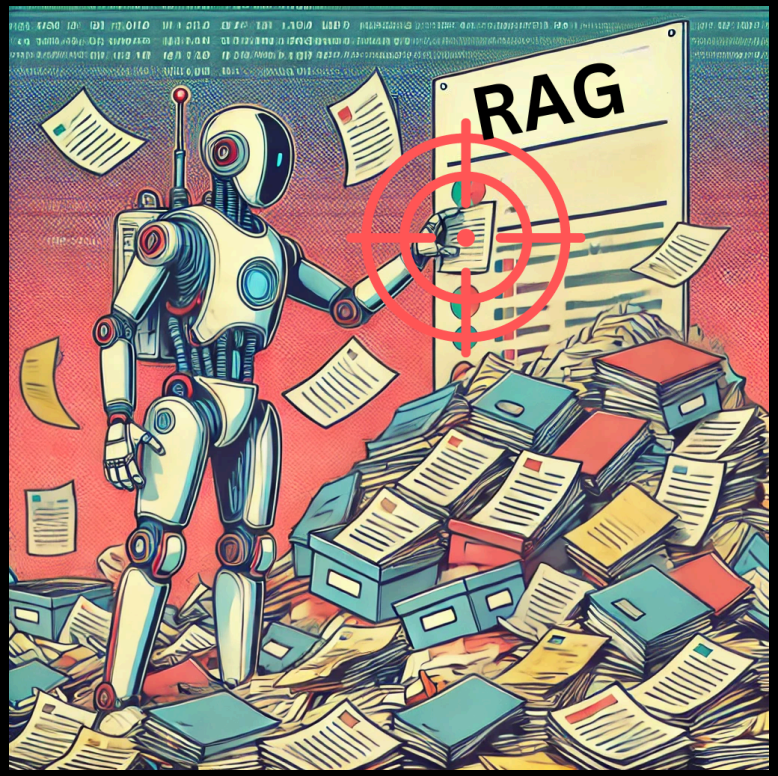
4. Customize

Three techniques to optimize the performance of the model

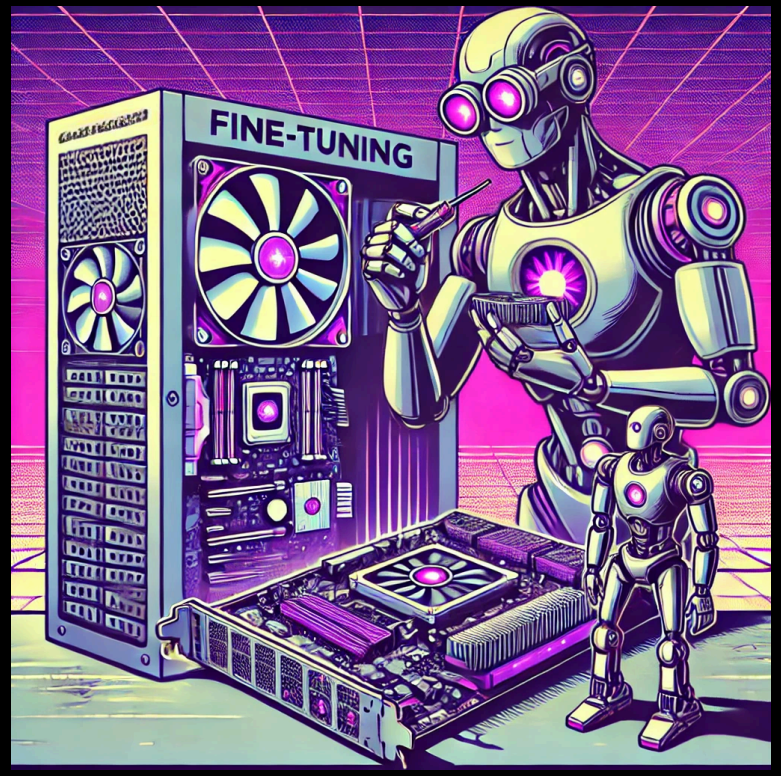


Prompting

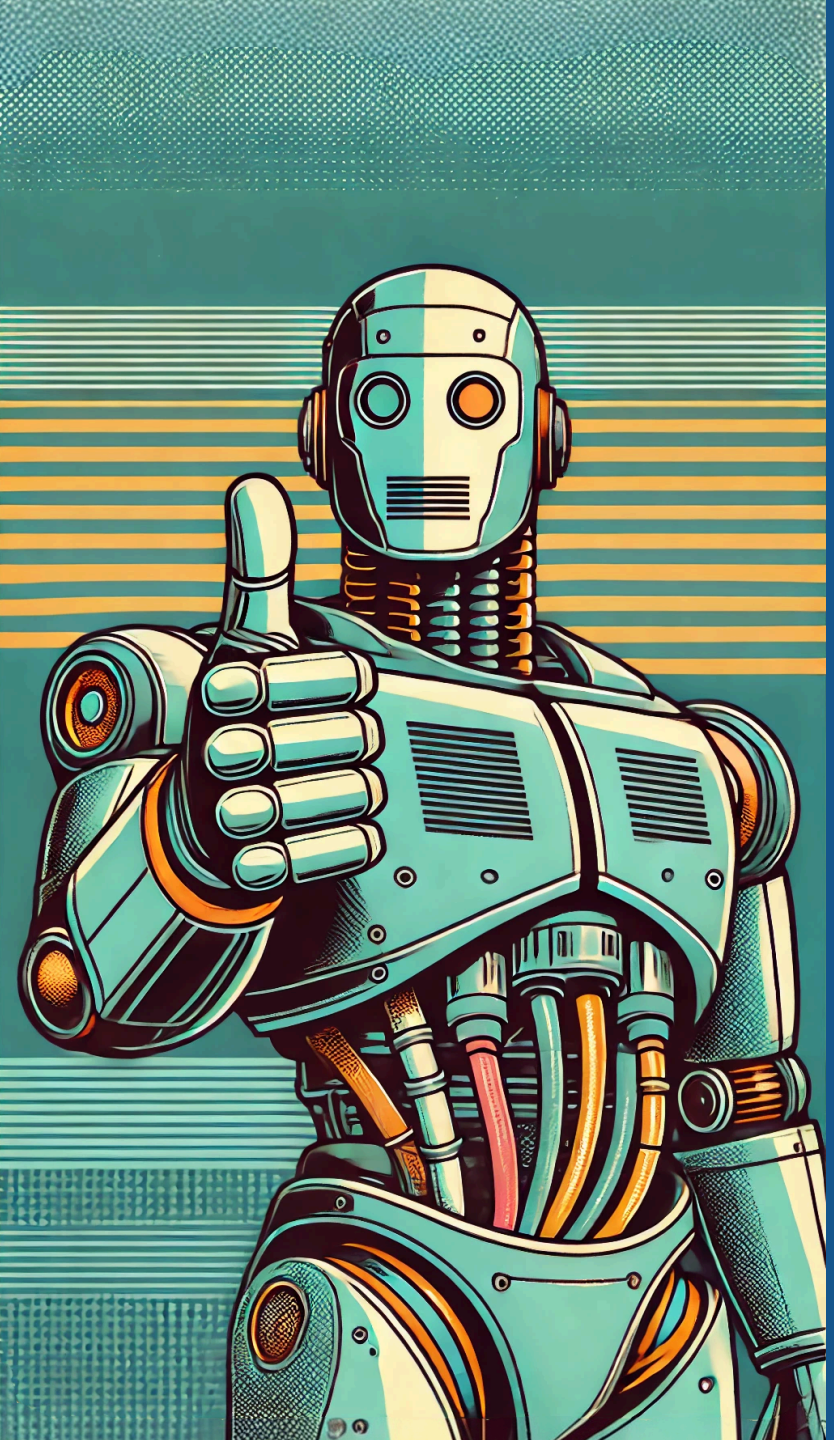
multi-shot, chaining and tools



RAG



Fine-tuning



Three Techniques: **Pros**

Prompting

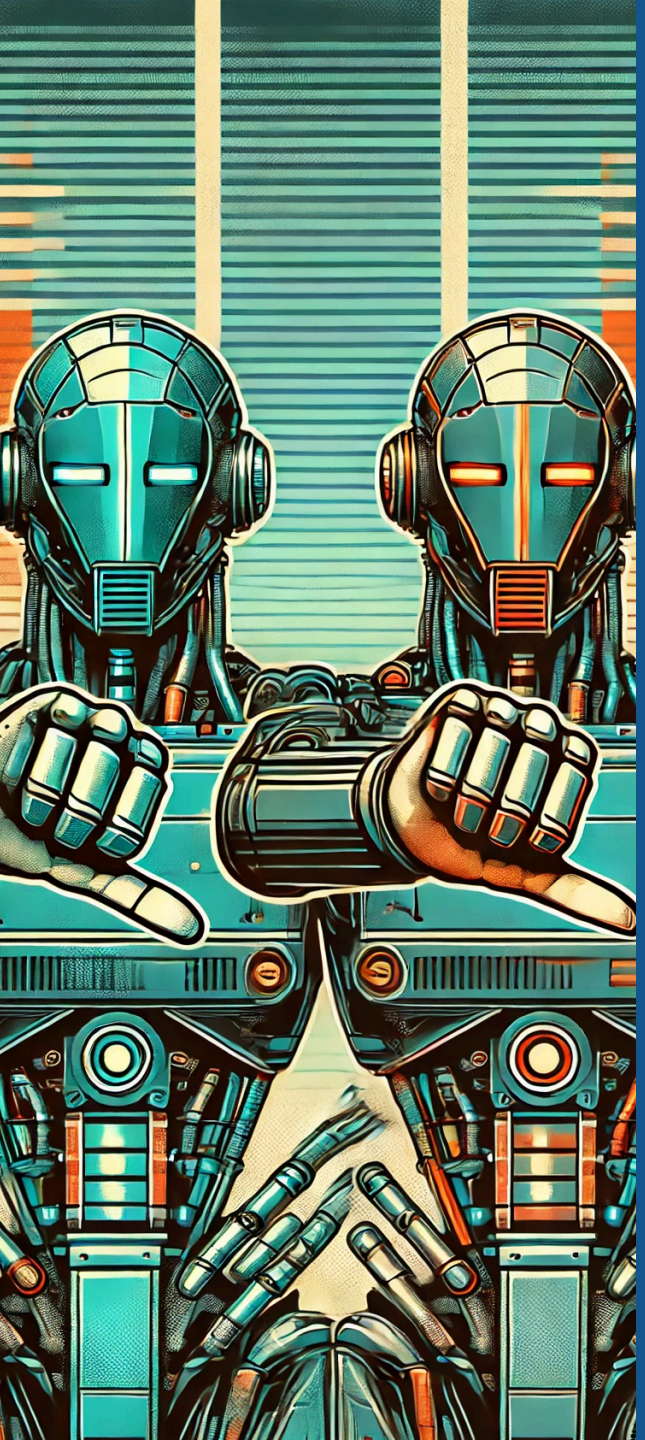
1. Fast to implement
2. Low cost
3. Often immediate improvement

RAG

1. Accuracy improvement with low data needs
2. Scalable
3. Efficient

Fine-tuning

1. Deep expertise & specialist knowledge
2. Nuance
3. Learn a different tone / style
4. Faster and cheaper inference



Three Techniques: **Cons**

Prompting

1. Limited by context length
2. Diminishing returns
3. Slower, more expensive inference

RAG

1. Harder to implement
2. Requires up-to-date, accurate data
3. Lacks nuance

Fine-tuning

1. Significant effort to implement
2. High data needs
3. Training cost
4. Risk of "catastrophic forgetting"

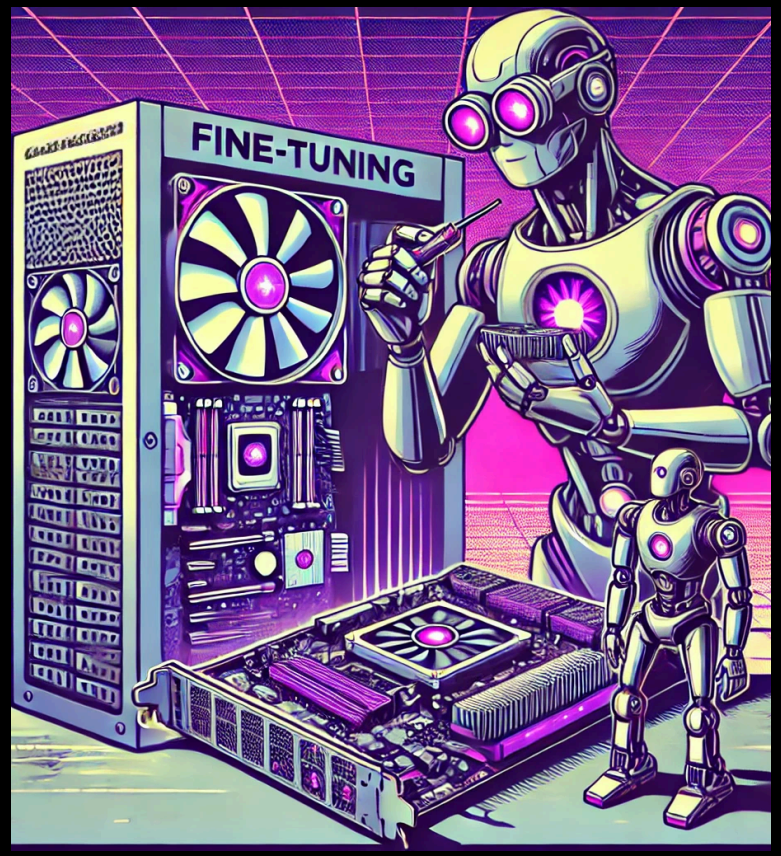
Three Techniques: **Use Cases**

Prompting

Often the starting point for optimizing a project,with a Frontier LLM

RAG

You need high accuracy without the cost of fine-tuning; you have a Knowledge Base



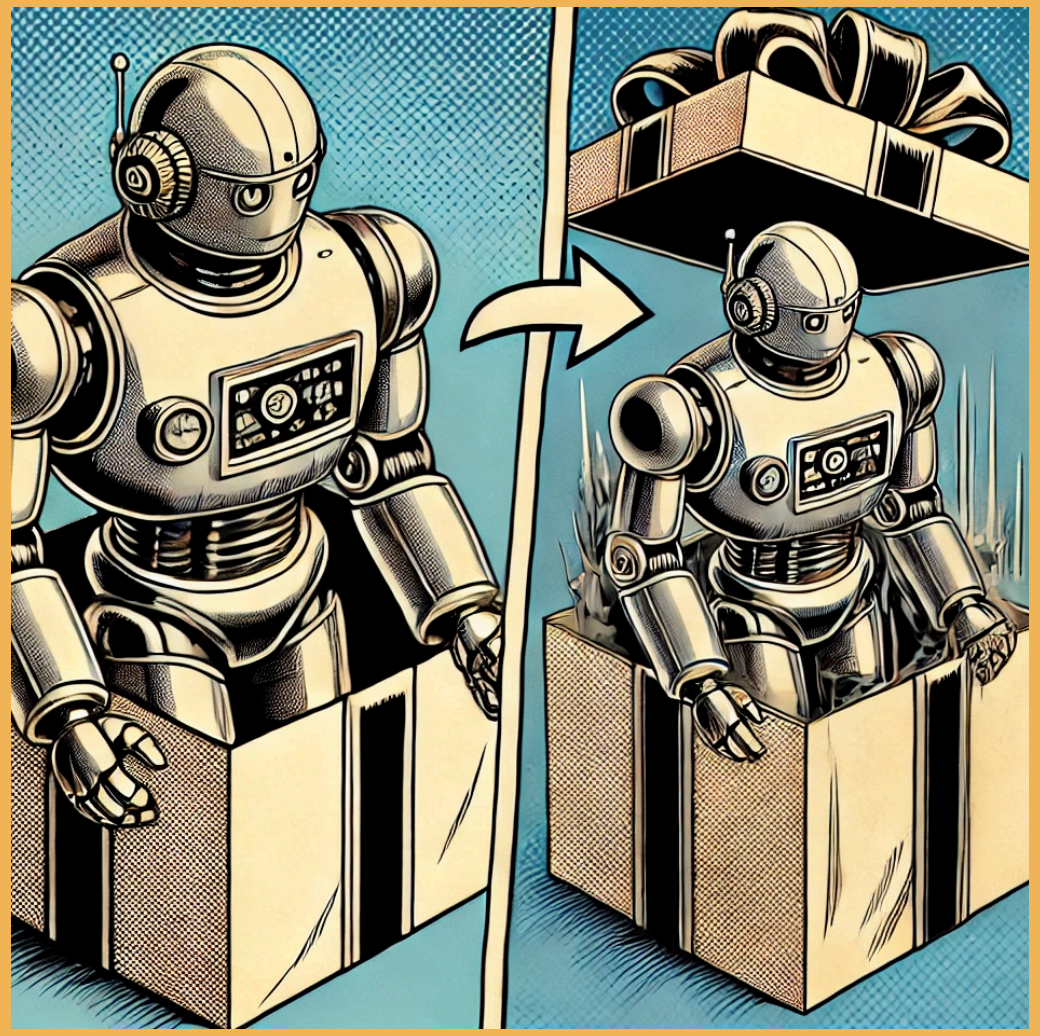
Fine-tuning

You have a specialized task with a high volume of data, and you need top performance

5. Productionize

Activities:

- Determine API between model and platform(s)
- Identify model hosting and deployment architecture
- Address scaling, monitoring, security and compliance
- Measure the Business-Focused Metrics identified in step 1
- Continuously retrain and measure performance



Revisiting the 5 Step Strategy

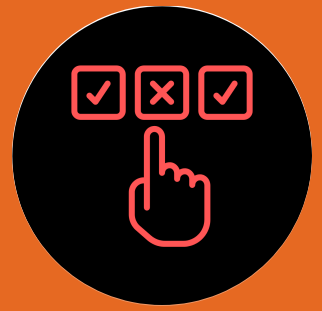
In the context of the Business Problem

Understand

DONE

Prepare

(IN PROGRESS)



Select

TODO

Customize

TODO

Productionize

TODO

PROGRESS REPORT

PHEW - A lot of talking

What you can now do

- Generate text and code with Frontier Models including AI Assistants with Tools, and with open-source models with HuggingFace transformers
- Create advanced RAG solutions with LangChain
- Follow a 5 step strategy to solve problems, including dataset curation

---

Next time, we get to action - you will be able to:

- Explain the role of a baseline model
- Create a traditional ML solution with features and linear regression
- Apply more advanced NLP techniques including SVR