(my) Principles for Developing Machine Learning Projects

Hugo Sousa

NLP Researcher

Applied Scientist @Amazon

2025

Career context

- Bachelor in Physics
- Master in Applied Mathematics
- (almost) PhD in Computer Science

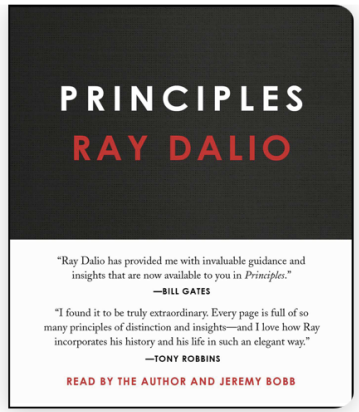
- Visiting Student @CMU
- Internship @Amazon in Palo Alto

- Doing research in NLP and IR

Website: https://hugosousa.net/

(my) Principles for Developing Machine Learning Projects

goodreads Home My Books Browse Community Search books



Want to Read

Buy on Amazon

Rate this book

Principles: Life and Work

Ray Dalio (Author/Narrator), Jeremy Bobb (Reader)

4.10 66,619 ratings · 3,833 reviews

Ray Dalio, one of the world's most successful investors and entrepreneurs, shares the unconventional principles that he's developed, refined, and used over the past forty years to create unique results in both life and business—and which any person or organization can adopt to help achieve their goals.

Show more

Genres Business Nonfiction Self Help Finance Leadership Personal Development Philosophy ...more

16 pages, Audio CD

First published September 19, 2017

Book details & editions

The **Principals**

- Be obsessed with your data

- Be maniacal about your code

- Constrain as little as possible

- Just do it

- Be bold, be simple

Be obsessed with your data

***GIGO**: "Garbage in, garbage out"*

- Coined circa **1957** from US mathematicians
- It **doesn't matter how sophisticate your learning mechanism is** or how much compute you have **if you your data is of poor quality**

The infamous tank story

Neural Network Follies

by Neil Fraser, September 1998

In the 1980s, the Pentagon wanted to harness computer technology to make their tanks harder to attack.

https://neil.fraser.name/writing/tank/

- Neil Fraser later admitted that the story "might be apocryphal."

- But a very similar one the Wolf/Husky has been discussed on the "Why should I trust you paper?"
https://dl.acm.org/doi/10.1145/2939672.2939778

The “it” in AI models is the dataset.

Posted on June 10, 2023 by jbetker

I’ve been at OpenAI for almost a year now. In that time, I’ve trained a **lot** of generative models. More than anyone really has any right to train. As I’ve spent these hours observing the effects of tweaking various model configurations and hyperparameters, one thing that has struck me is the similarities in between all the training runs.

It’s becoming awfully clear to me that these models are truly approximating their datasets to an incredible degree. What that means is not only that they learn what it means to be a dog or a cat, but the interstitial frequencies between distributions that don’t matter, like what photos humans are likely to take or words humans commonly write down.

What this manifests as is – trained on the same dataset for long enough, pretty much every model with enough weights and training time converges to the same point. Sufficiently large diffusion conv-unets produce the same images as ViT generators. AR sampling produces the same images as diffusion.

This is a surprising observation! It implies that model behavior is not determined by architecture, hyperparameters, or optimizer choices. It’s determined by your dataset, nothing else. Everything else is a means to an end in efficiently delivery compute to approximating that dataset.

Then, when you refer to “Lambda”, “ChatGPT”, “Bard”, or “Claude” then, it’s not the model weights that you are referring to. It’s the dataset.

https://nonint.com/2023/06/10/the-it-in-ai-models-is-the-dataset/

Prompt: “Write a metaphor about time”

ChatGPT

Time is a quiet river carving mountains—so gentle you hardly feel it passing, yet so persistent that nothing you know remains unchanged.

Gemini

⏳ **Time as a River**

Time is a **river**, swift and relentless, flowing ever onward to the sea of the future. We can stand by its banks, watch the moments rush past, but we can never step into the same current twice.

Time is a river that carries us forward on its current—we can dip our hands in to feel each moment pass through our fingers, but we can never hold the water still or swim back upstream to where we've already been.

Grok

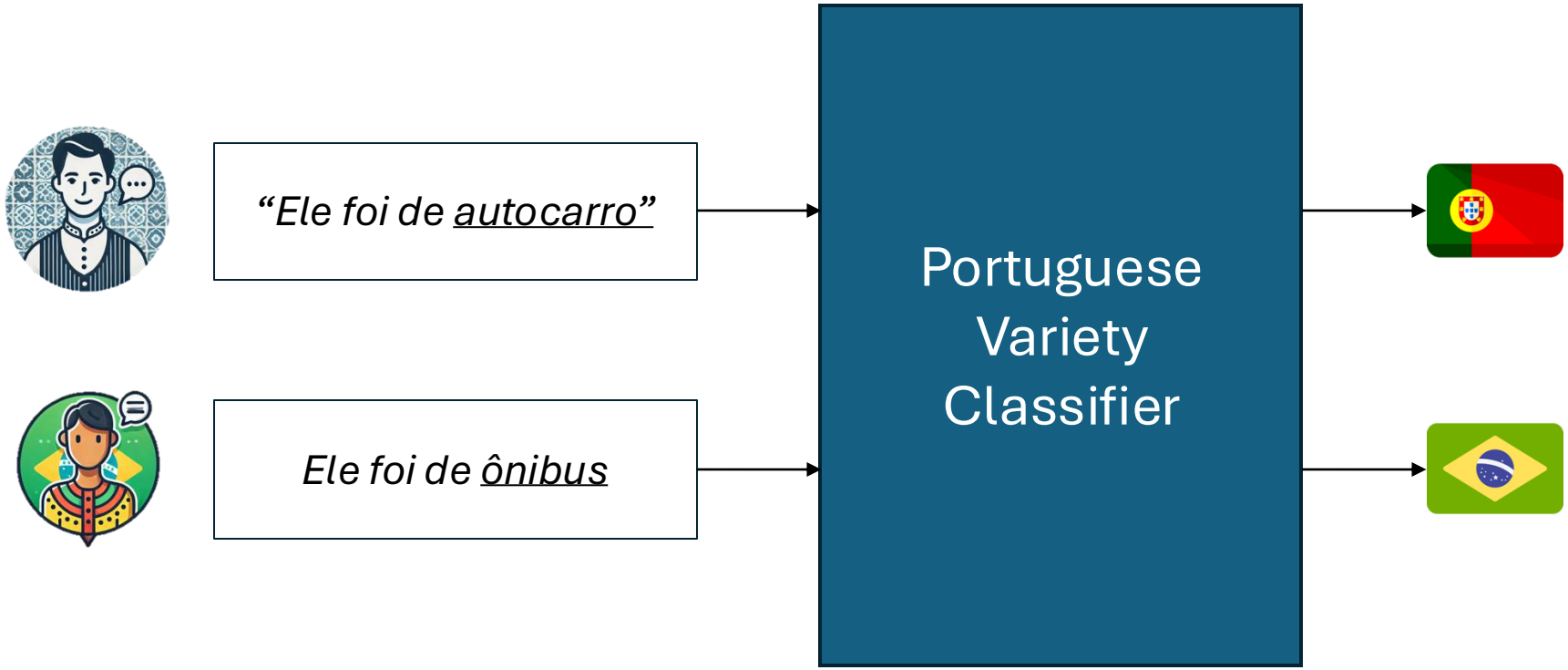
Time is a river that looks calm from the bridge, but if you jump in you realize it’s a whitewater torrent dragging you backward over rocks you never saw coming—every second a hidden rapid, every memory a bruise you only notice once you’re slammed against it, and the shore you keep reaching for is always the one that’s already miles behind you.

Hive mind paper: https://arxiv.org/pdf/2510.22954

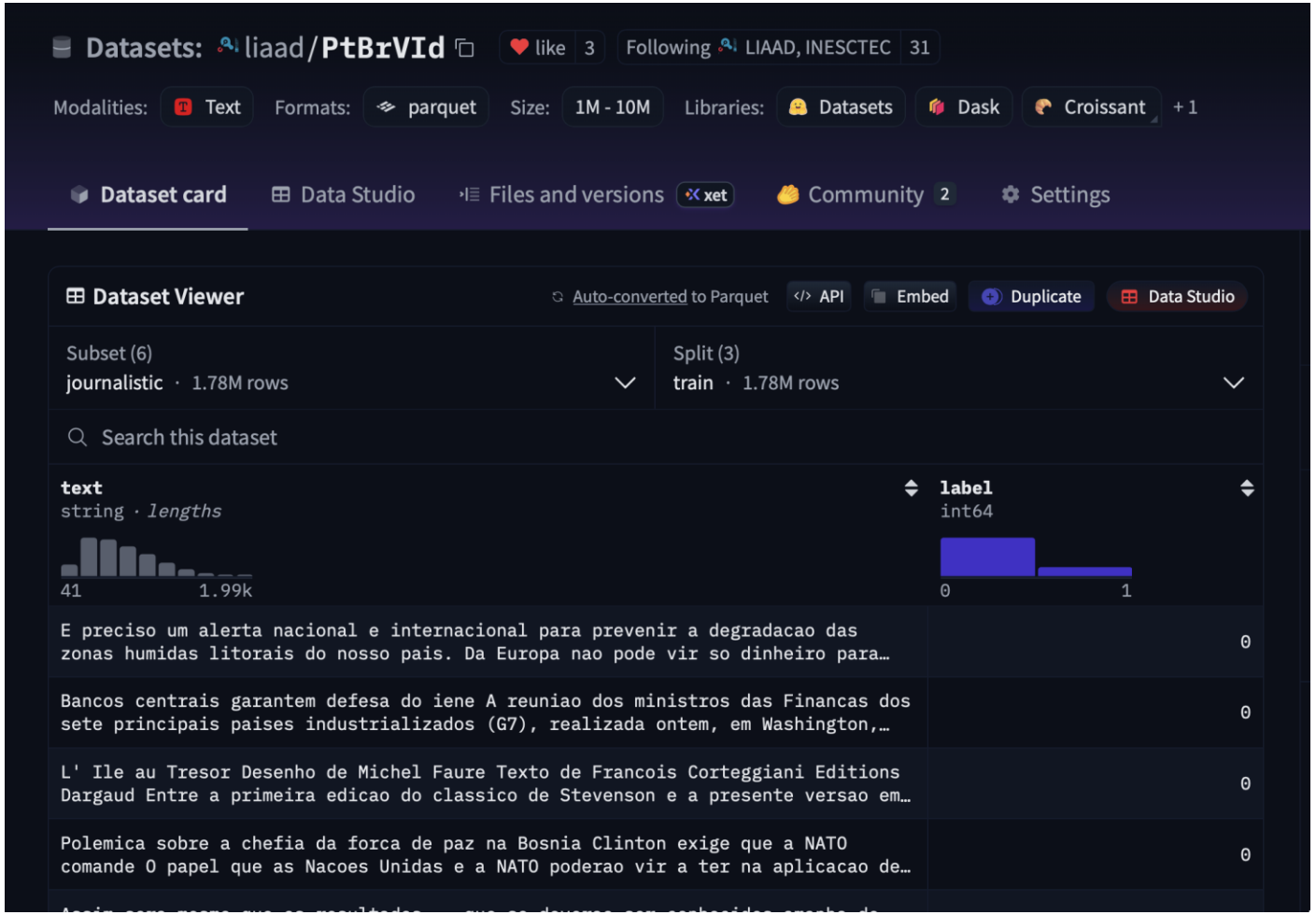
The PtVId story

Country	Population of country[75][76][77]	More information	Native language of the majority	Spoken by
Brazil	203,062,512	Portuguese in Brazil	✓	95% as a native language[78]
Angola	35,981,281	Portuguese in Angola	✗	40% as a native language, 60% total[79]
Mozambique	32,513,805	Portuguese in Mozambique	✗	17% as a native language, 44% total[78]
Portugal	10,467,366	Portuguese in Portugal	✓	95% as a native language[80]
Guinea-Bissau	2,078,820	Portuguese in Guinea-Bissau	✗	0.3% as a native language, 20% total[81]

The PtVId story: the goal



The PtVId story: the dataset



	Label	Tokens Count	Docs Count
Journalistic	🇵🇹	189,506,320	1,443,422
	🇧🇷	27,077,538	333,903
Literature	🇵🇹	1,859,660	24,090
	🇧🇷	3,805,896	52,458
Legal	🇵🇹	152,717,737	2,957,980
	🇧🇷	221,167	4,653
Politics	🇵🇹	7,203,739	27,887
	🇧🇷	1,012,586	3,656
Web	🇵🇹	22,598,587	43,630
	🇧🇷	23,913,771	44,313
Social Media	🇵🇹	44,758,304	2,363,261
	🇧🇷	94,177	5,504

https://ojs.aaai.org/index.php/AAAI/article/view/34705

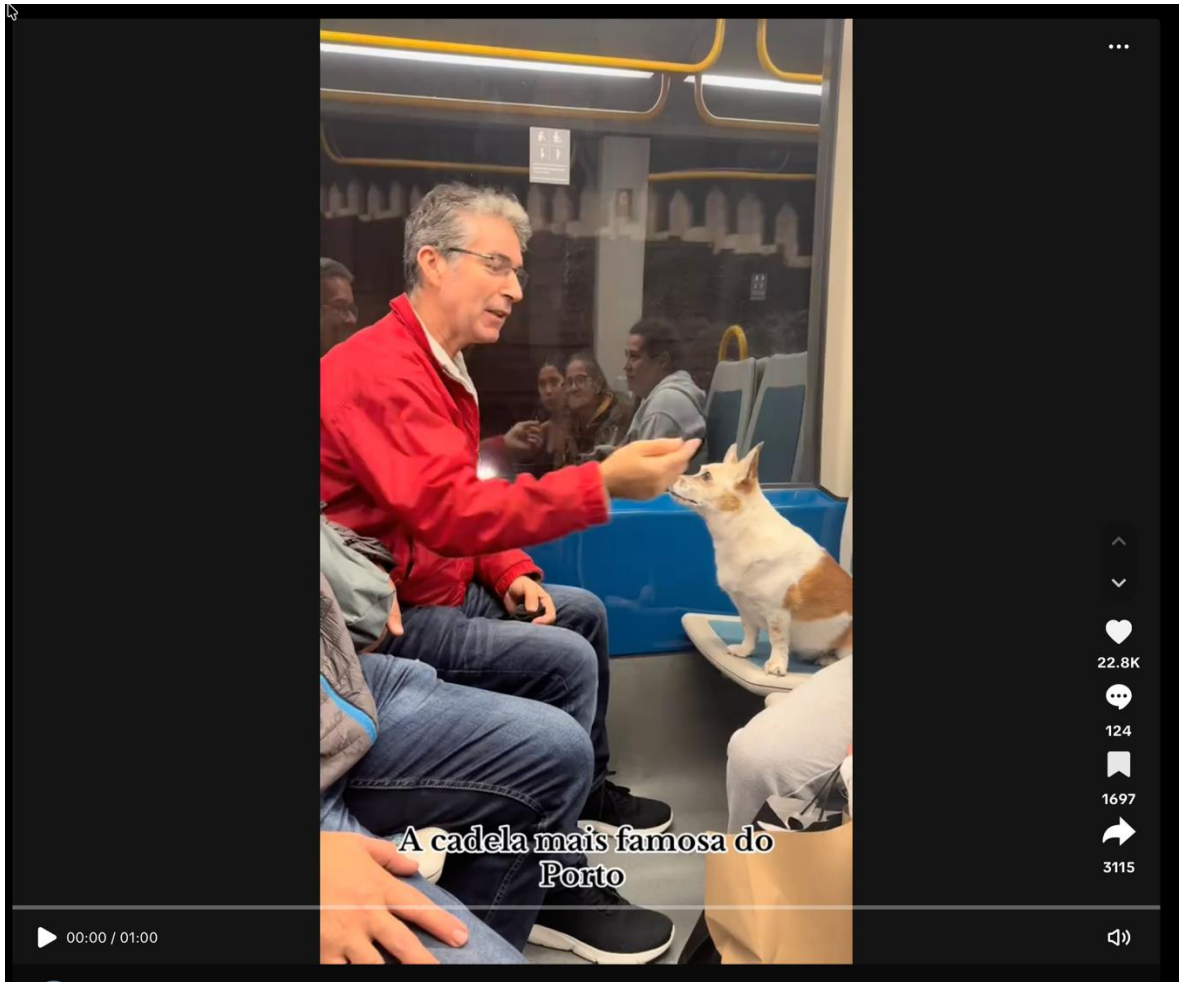
The PtVId story: the filtering

Code Blame 306 lines (225 loc) · 6.87 KB

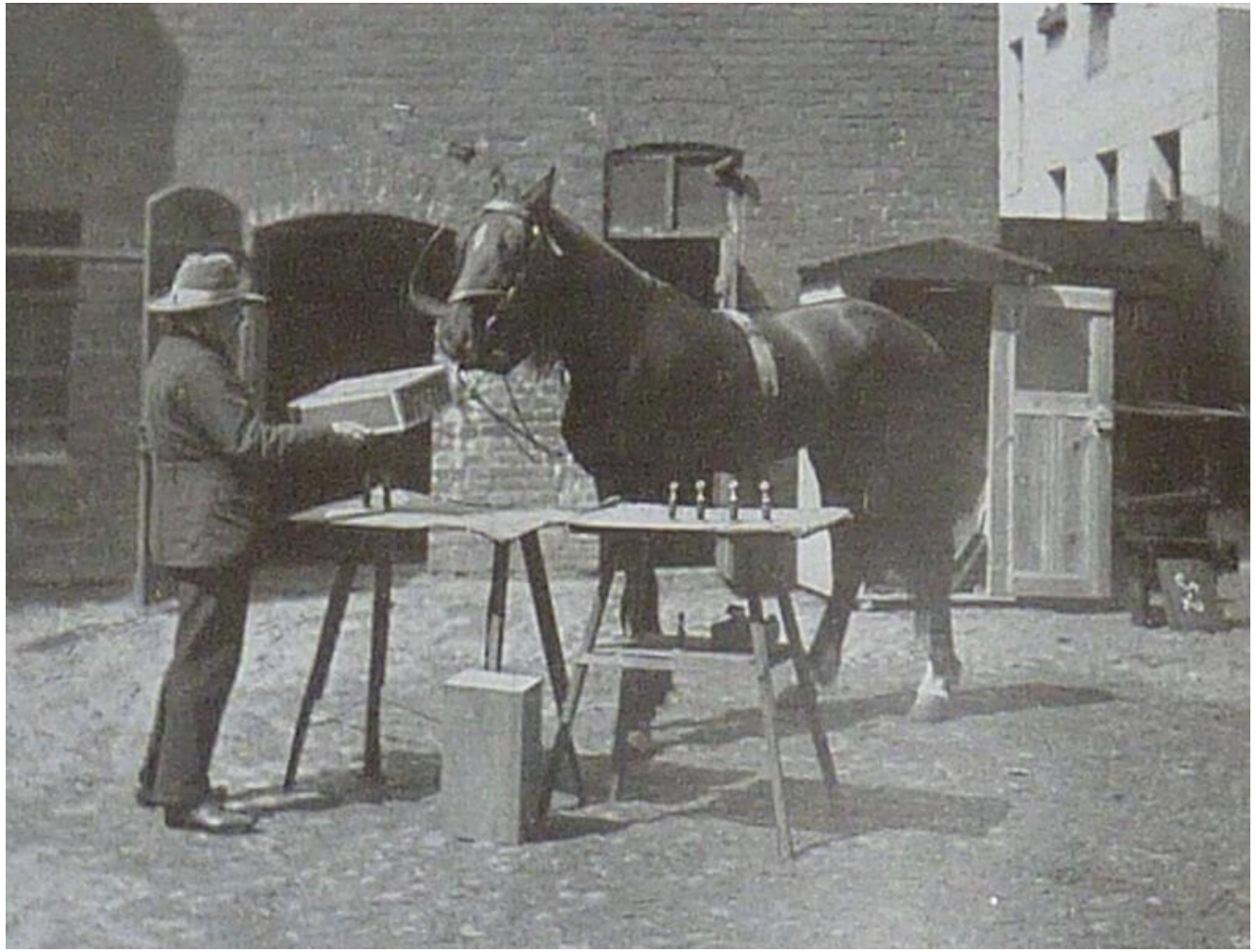
```python
import multiprocessing as mp
import re

import justext
import numpy as np
from bs4 import BeautifulSoup
from transformers import AutoTokenizer

HTML_RE = re.compile(r"<[^>]+>")
URL_RE = re.compile(
    r"((http|https)\:\/\/)?[a-zA-Z0-9\.\/\?\:@\-_=#]+\.([a-zA-Z]){2,6}([a-zA-Z0-9\.\&\/\?\:@\-_=#…])*"
)
HASHTAG_RE = re.compile(r"#(\w+)")
QUOTE_SPACE_START_RE = re.compile(r"^\"\s")
QUOTE_SPACE_END_RE = re.compile(r"\s\"$")
MENTION_RE = re.compile(r"@(\w+)")
RETWEET_RE = re.compile(r"RT @(\w+):")
COD_RE = re.compile(r"COD _ (\w+) ")
BULLET_RE = re.compile(r"^(\d)+.\s")
THREE_DASH_RE = re.compile(r"---.*---")
MORE_THAN_THREE_POINTS_RE = re.compile(r"\.{4,}")


TOKENIZER = AutoTokenizer.from_pretrained("meta-llama/Meta-Llama-3-8B")

VALID_CHARS = "0123456789abcdefghijklmnopqrstuvwxyzàáâãäåèéêëéēîïìıīĵłñńôöòóõōšśûüùúūÿýźçćčñń!\"#$%&'()*
```

Be maniacal about your code

The smart dog



Clever Hans



The same can happen to your **model!**

```
Python 3.13.3 (main, Apr  8 2025, 13:54:08) [Clang
Type "help", "copyright", "credits" or "license" f
>>> hash([42])
Traceback (most recent call last):
  File "<python-input-0>", line 1, in <module>
    hash([42])
    ~~~~^^^^^^
TypeError: unhashable type: 'list'
>>>
```

```
>>> train()
100%|                                                                    | 100/100
Training finished
Your model is ready great!
>>>
```

TRUST ME!

Training Neural Models

Start simple
- Small architecture

Build your training loop
- No fancy features
- No dropout, batch norm, LR decay, et cetera

Setup Tensorboard
- Log train and valid loss of every batch
- After every train step run a forward pass on a random sample of a valid set

Overfit to a 1 example
- The network should be able to memorize this example
- Train loss should be zero

Overfit to a 10 example
- Now the input will be taken into consideration
- Train loss should be zero
- Val loss should go up

Use more data
- Now valid should start to go down
- If so, the model is learning

Profile your code
- Make your training runs go faster

optional

Log more metrics
- ROC-curves
- Prediction distributions
- et cetera

Hyperparameter tunning
- ideally, in parallel
- data augmentation: never apply the transformations to the valid set

Try different approaches
- In practice, the baseline all models are trying to beat is the best approach

Constrain as little as possible

The Bitter Lesson

Rich Sutton

March 13, 2019

http://www.incompleteideas.net/IncIdeas/BitterLesson.html

"The biggest lesson that can be read from 70 years of AI research is that **general methods that leverage computation are ultimately the most effective**, and by a large margin."

"Seeking an improvement that makes a difference in the shorter term, researchers **seek to leverage their human knowledge** of the domain, but **the only thing that matters in the long run is the leveraging of computation**."


GARRY
KASPAROV
DEEP
BLUE


ALPHAGO
01:58:37
AlphaGo
Lee Sedol
Google DeepMind
Challenge Match
8 - 15 March 2016

In practice

- **Don't hard-code your own understanding** of the world into the model, because **your understanding is flawed and doesn't scale**.
- **Compute scales, human intuition doesn't.**

- Inductive bias:
  - Try to find ways to reduce the bias of the model so that it generalizes
  - Increasing the amount of data is usually the best way

Just do it

Taste vs. Skill

- There is a gap between what we imagine we can do and what we can do

"Man is the only animal that laughs and weeps; for he is the only animal that is struck with the difference between what things are and what they ought to be." — William Hazlitt

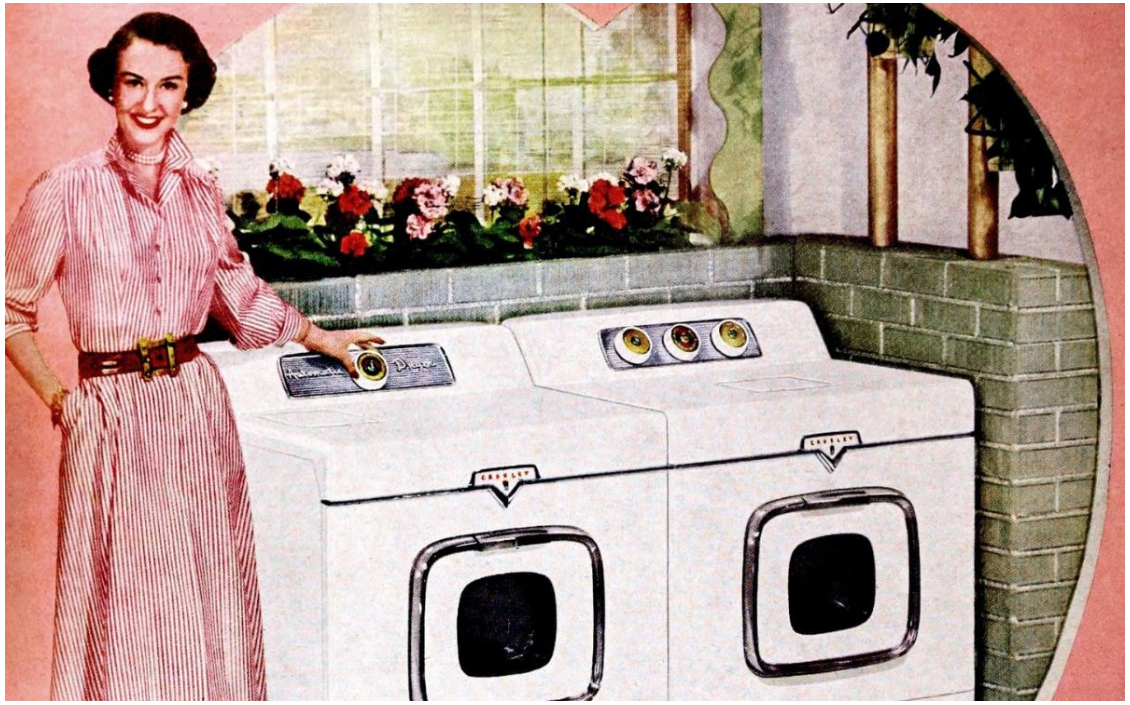
Procrastination

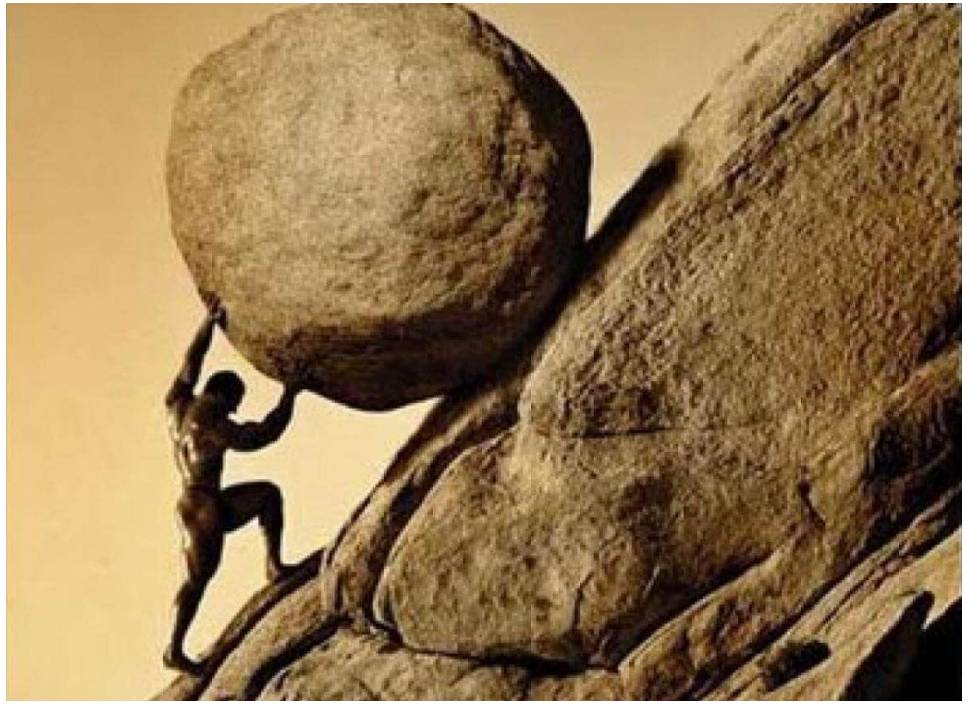
- Procrastination is a defence mechanism to protect the "perfect version" of the project in your head from becoming the "imperfect reality.”
- Solution: Take action

- The photography class test

“The best is the enemy of the good”

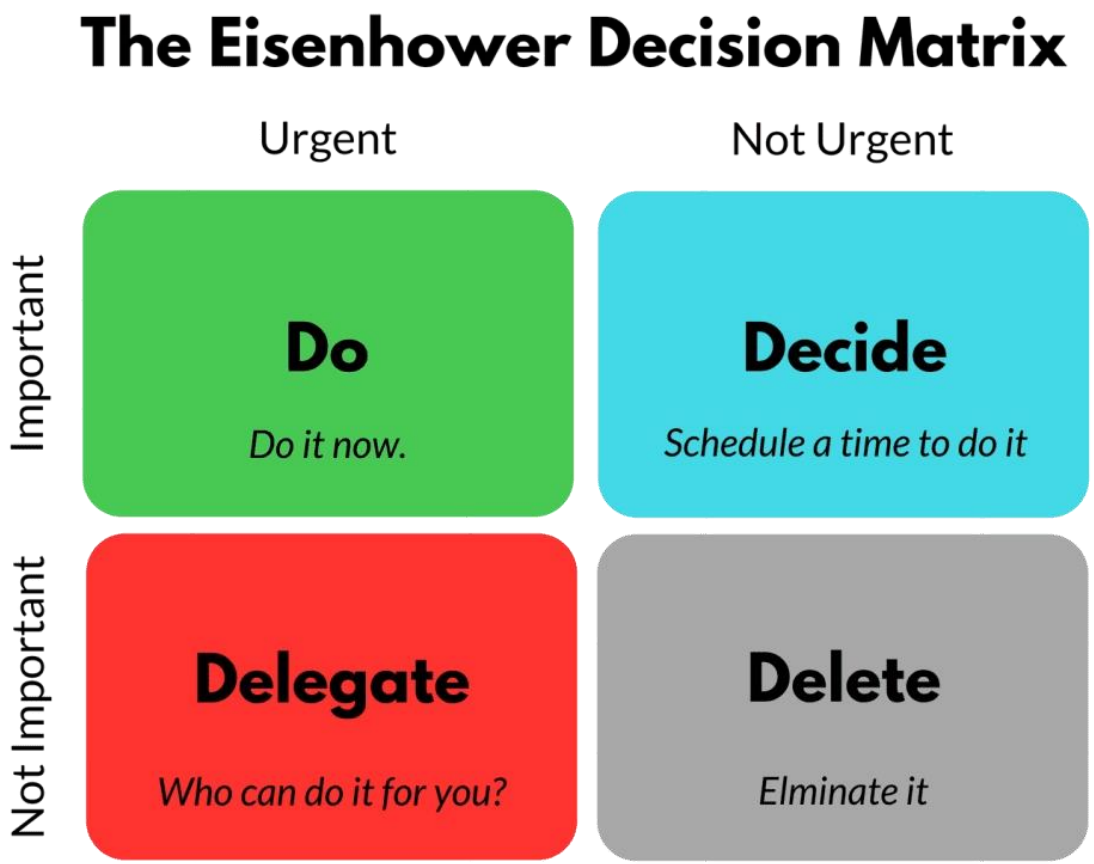
A side note: Productivity in the era of AI





Four Thousand Weeks: Time Management for Mortals

A side note: Productivity in the era of AI



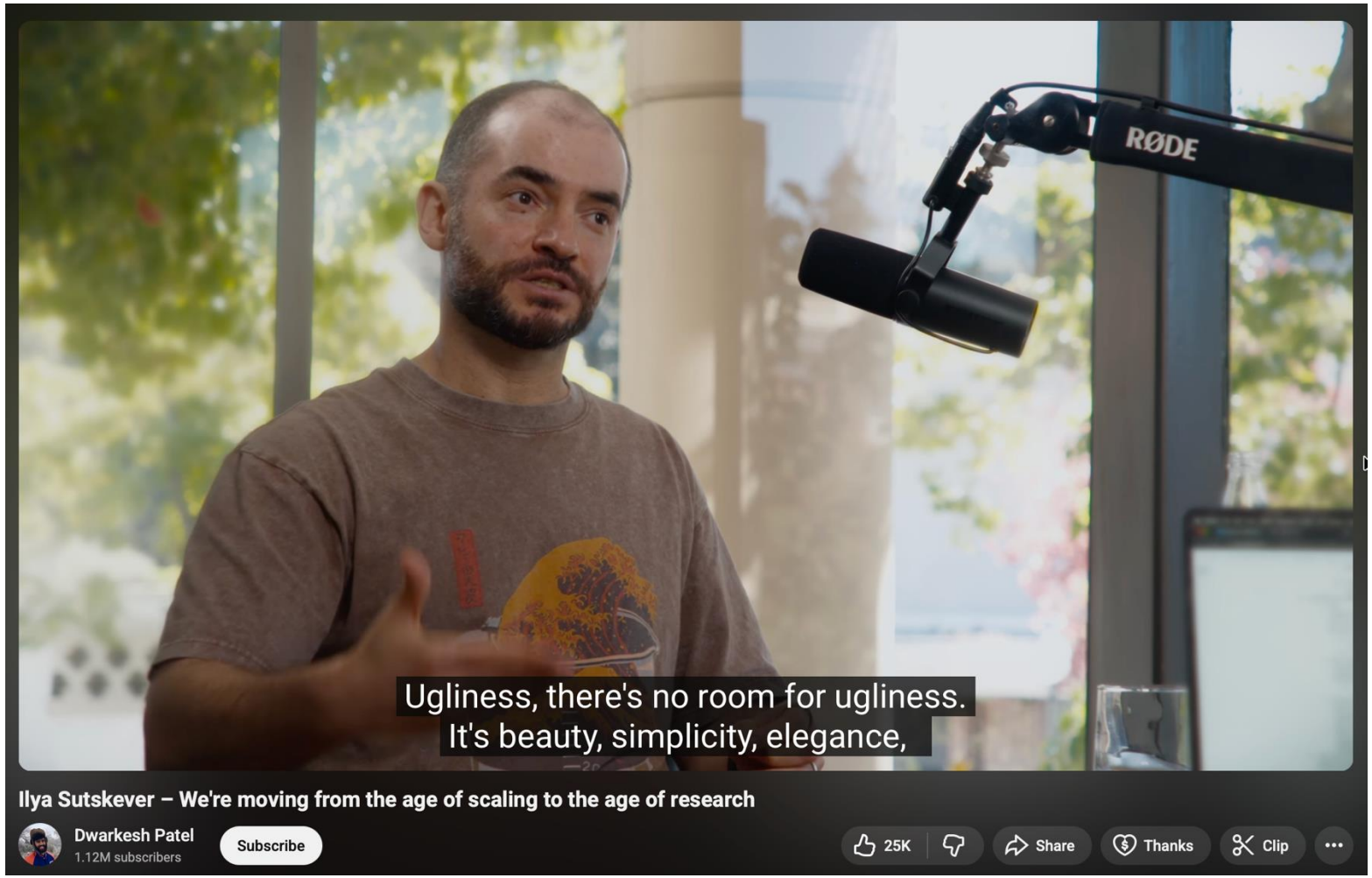
FOMO

↓

JOMO

Be bold, be simple

“What is research taste?”



Some examples of simplicity

$$\vec{F} = m\vec{a}$$

$$i\hbar\frac{\partial}{\partial t}|\Psi\rangle = \hat{H}|\Psi\rangle$$

$$E = mc^2$$

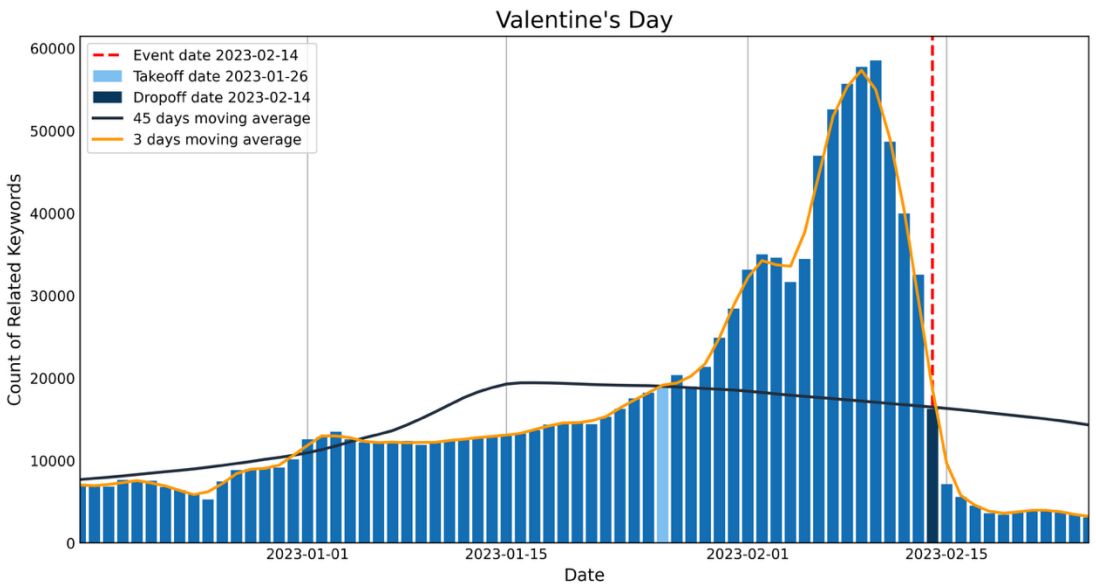
$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

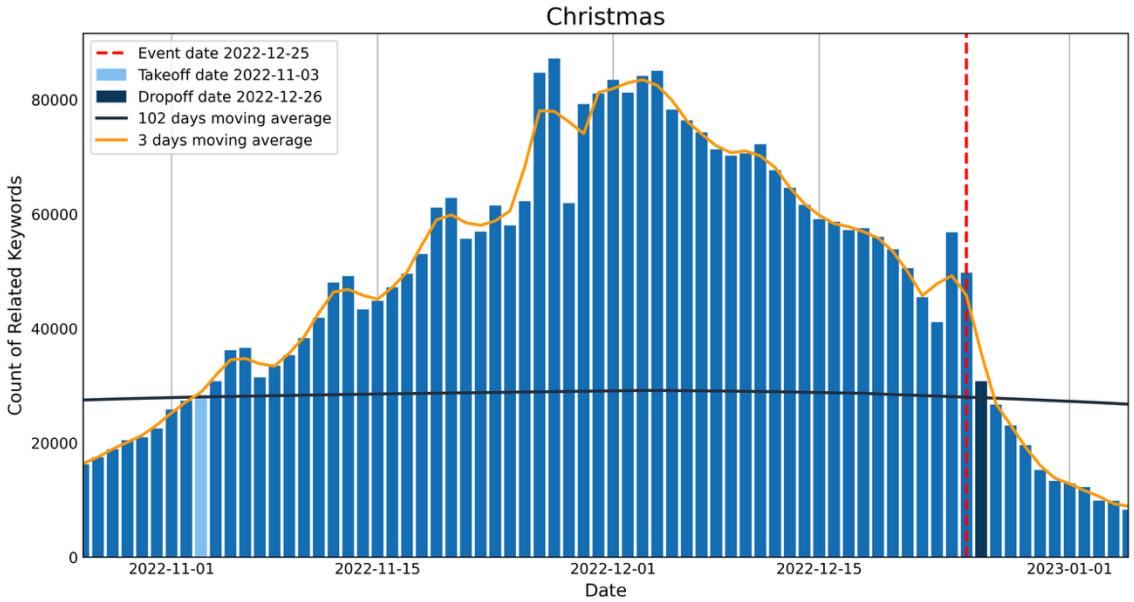
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$

$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Event Temporal Models





[Don't Forget This: Augmenting Results with Event-Aware Search]

Thank you for you attention