A perspective on MLOps from experimentation: Weights and Bias implementations

Globant
Daniel Jiménez

2024-05-17

Content

- Motivation
  ✅ Machine Learning and Hardware
  ✅ AI Computing
  ✅ Challenges of experimentation
- Weights and Bias Platform
- Weights and Bias Example

Motivation

Thinking about machine learning in terms of three components:

- Algos (LSTM, XGBoost, Random Forest)
- Data (From accessibility, and the cookies that you and I accept)
- Hardware (Gpus, CPus, architecture)

Motivation

Thinking about machine learning in terms of three components:

Algos
Data
Hardware

> Computing > Resource demand

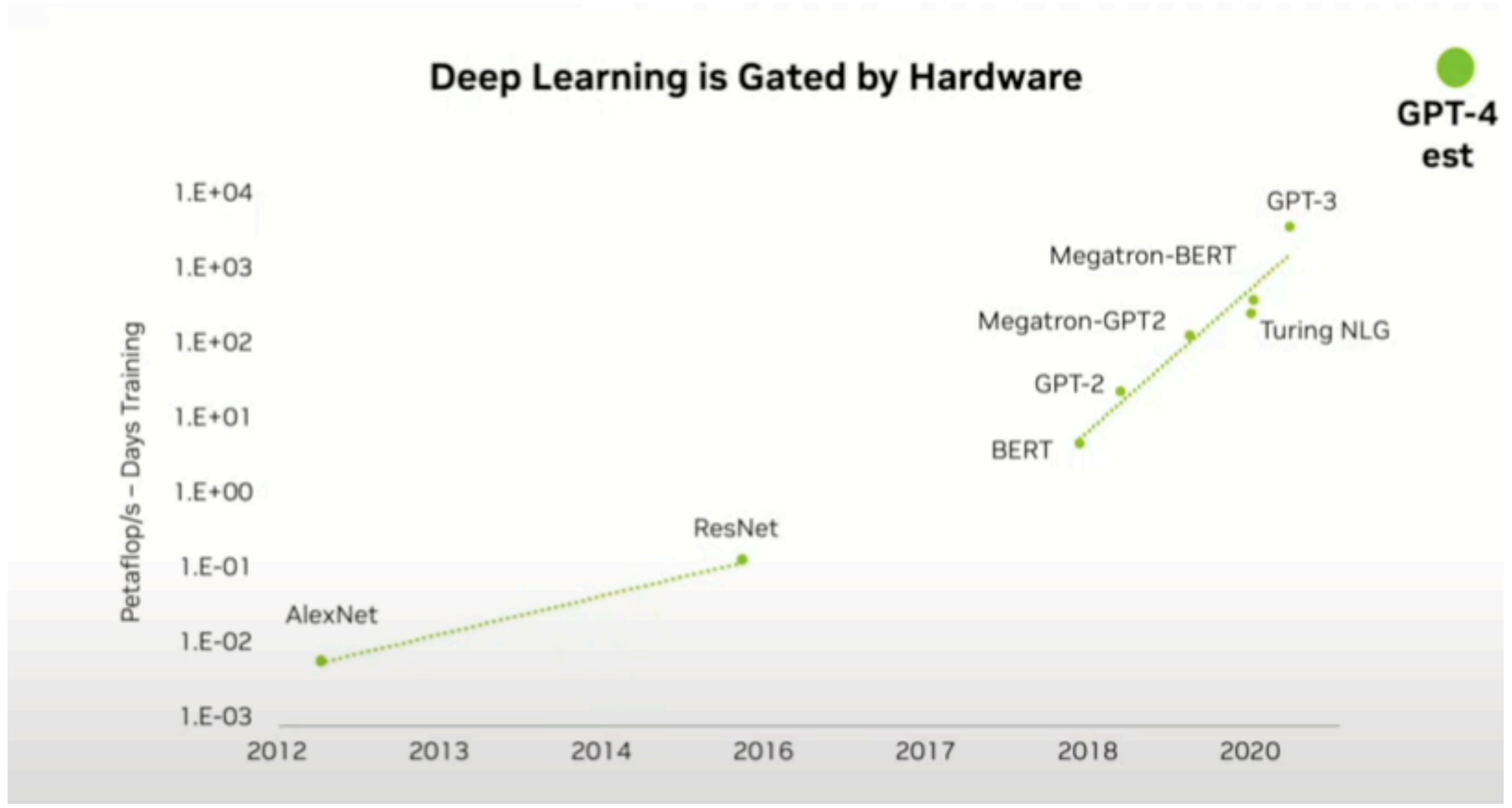
Motivation

Thinking about machine learning in terms of three components:

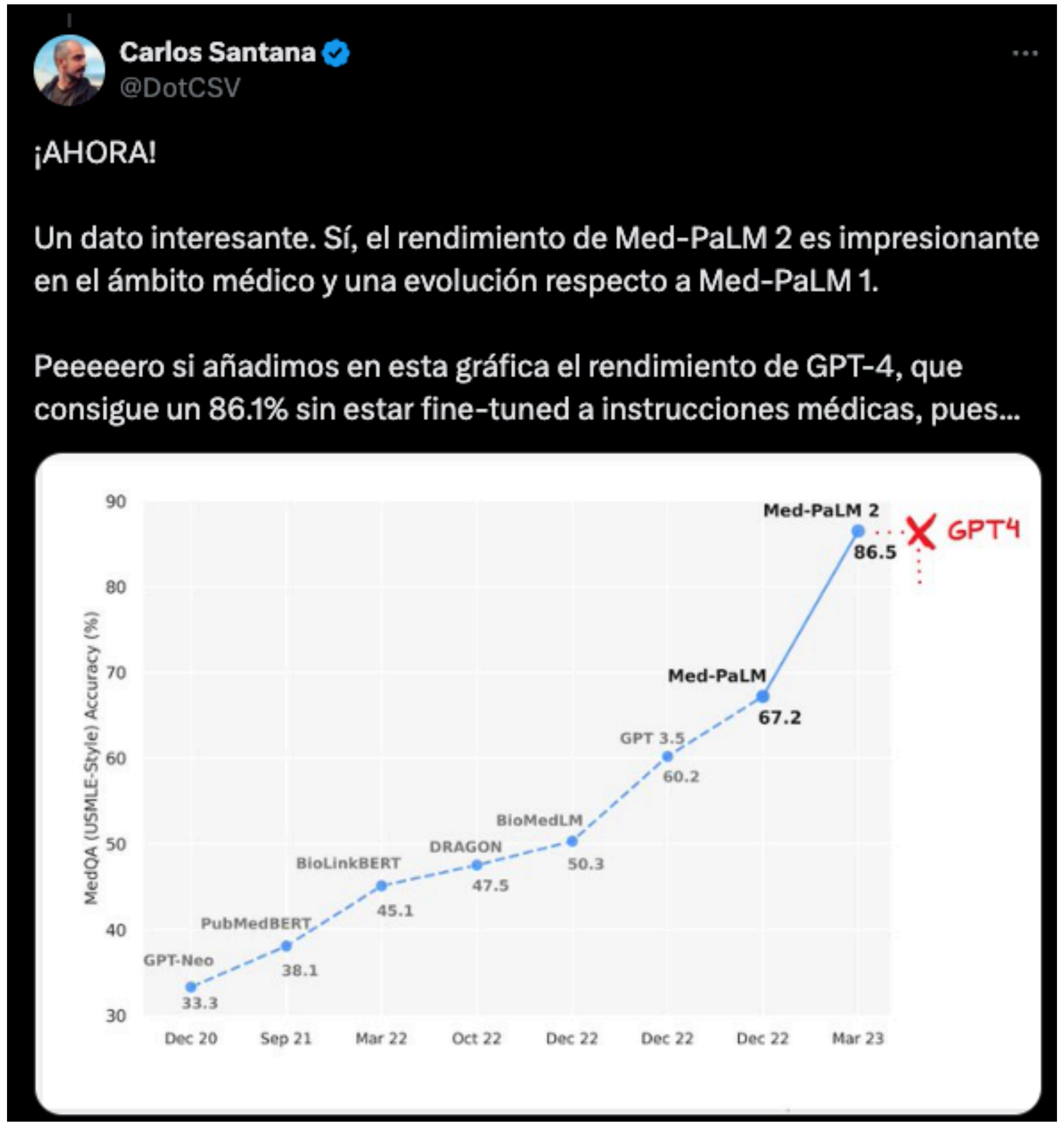
Algos
Data
Hardware

$> \underbrace{\text{Computing}}_{\text{Inference}} >$ Resource demand

Motivation



Nvidia Reference


Carlos Santana
@DotCSV

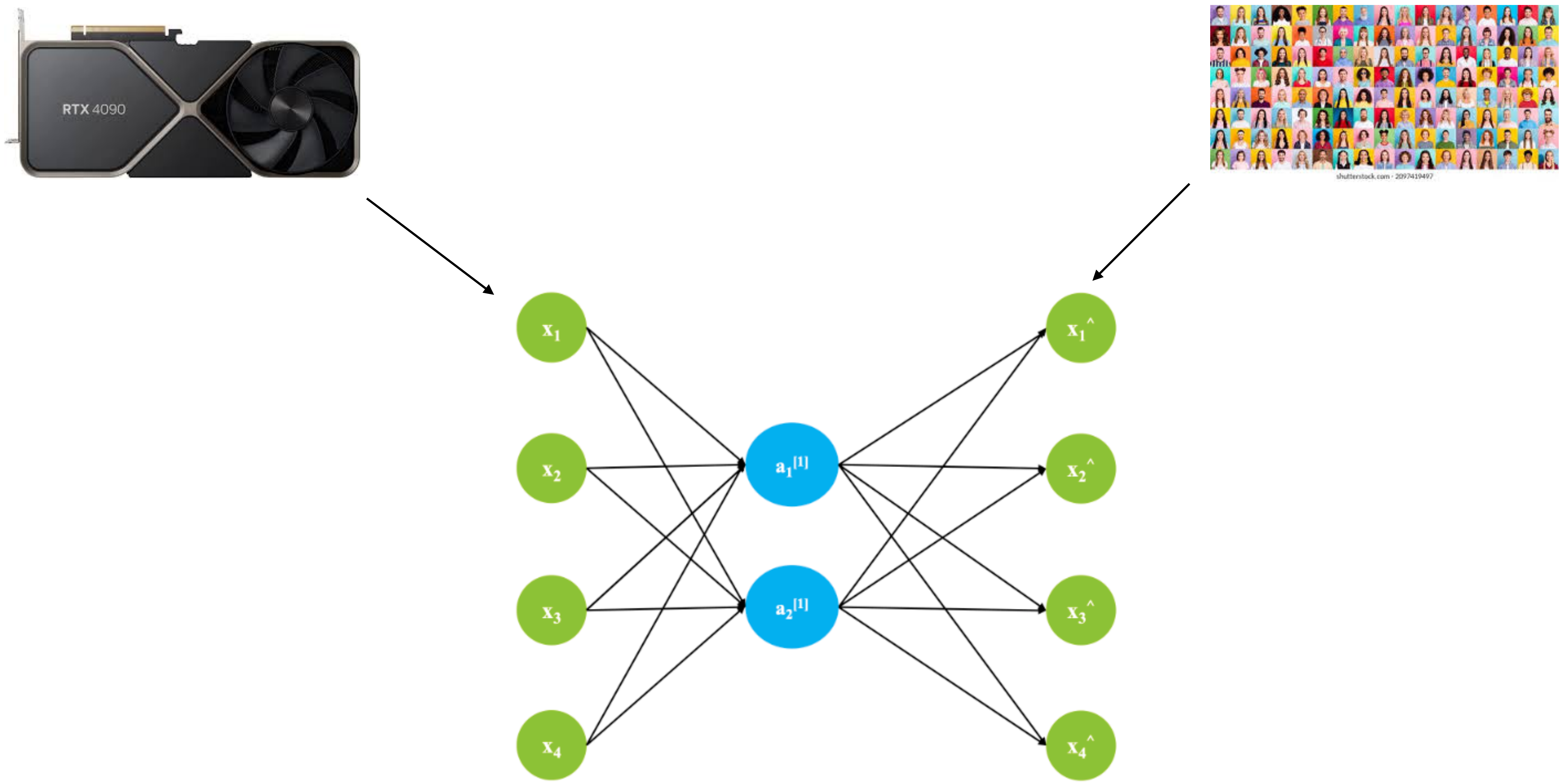
¡AHORA!

Un dato interesante. Sí, el rendimiento de Med-PaLM 2 es impresionante en el ámbito médico y una evolución respecto a Med-PaLM 1.

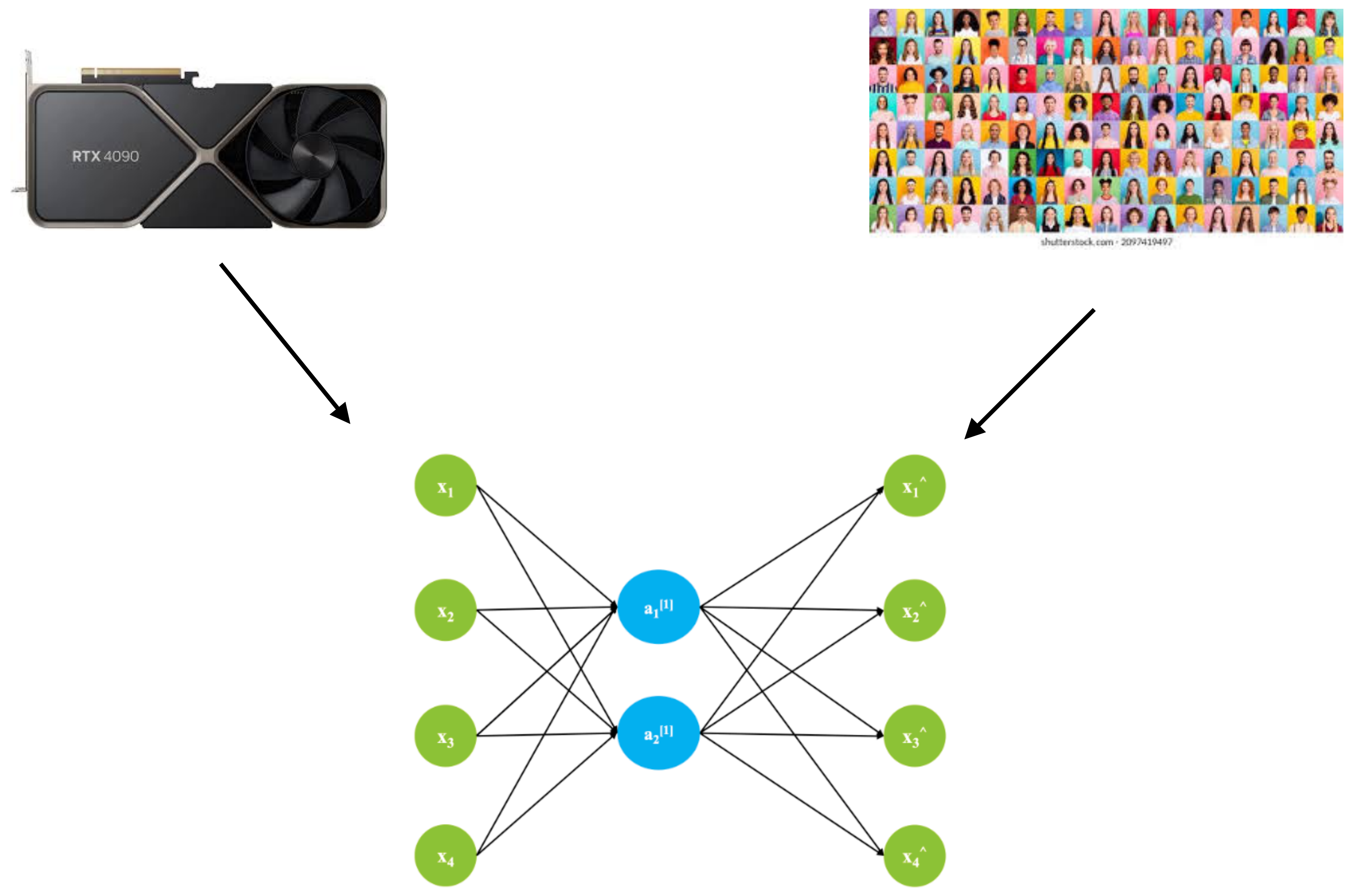
Peeeeero si añadimos en esta gráfica el rendimiento de GPT-4, que consigue un 86.1% sin estar fine-tuned a instrucciones médicas, pues...

Carlos Santana Tweet

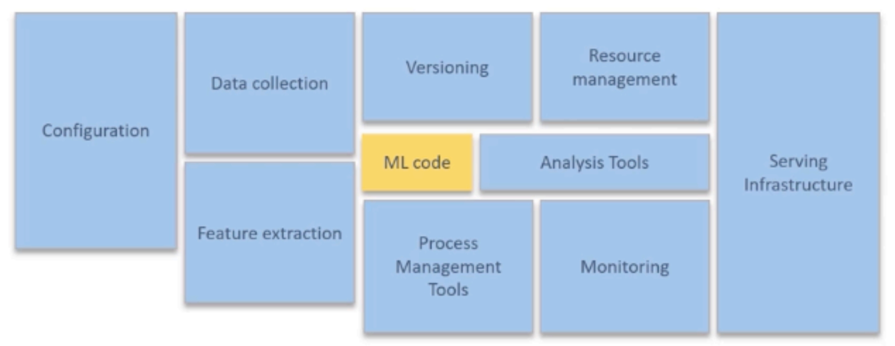
Motivation


RTX 4090
shutterstock.com · 2097419497
x_1
x_2
x_3
x_4
$a_1^{[1]}$
$a_2^{[1]}$
$x_1^\wedge$
$x_2^\wedge$
$x_3^\wedge$
$x_4^\wedge$

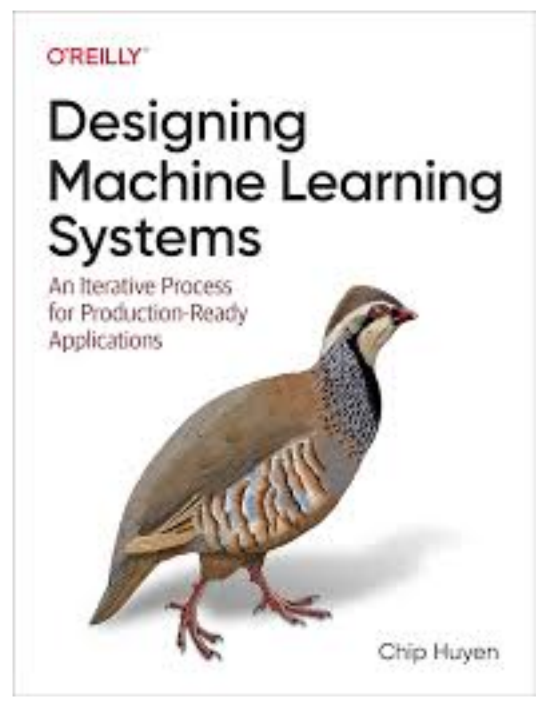
Motivation



Consume

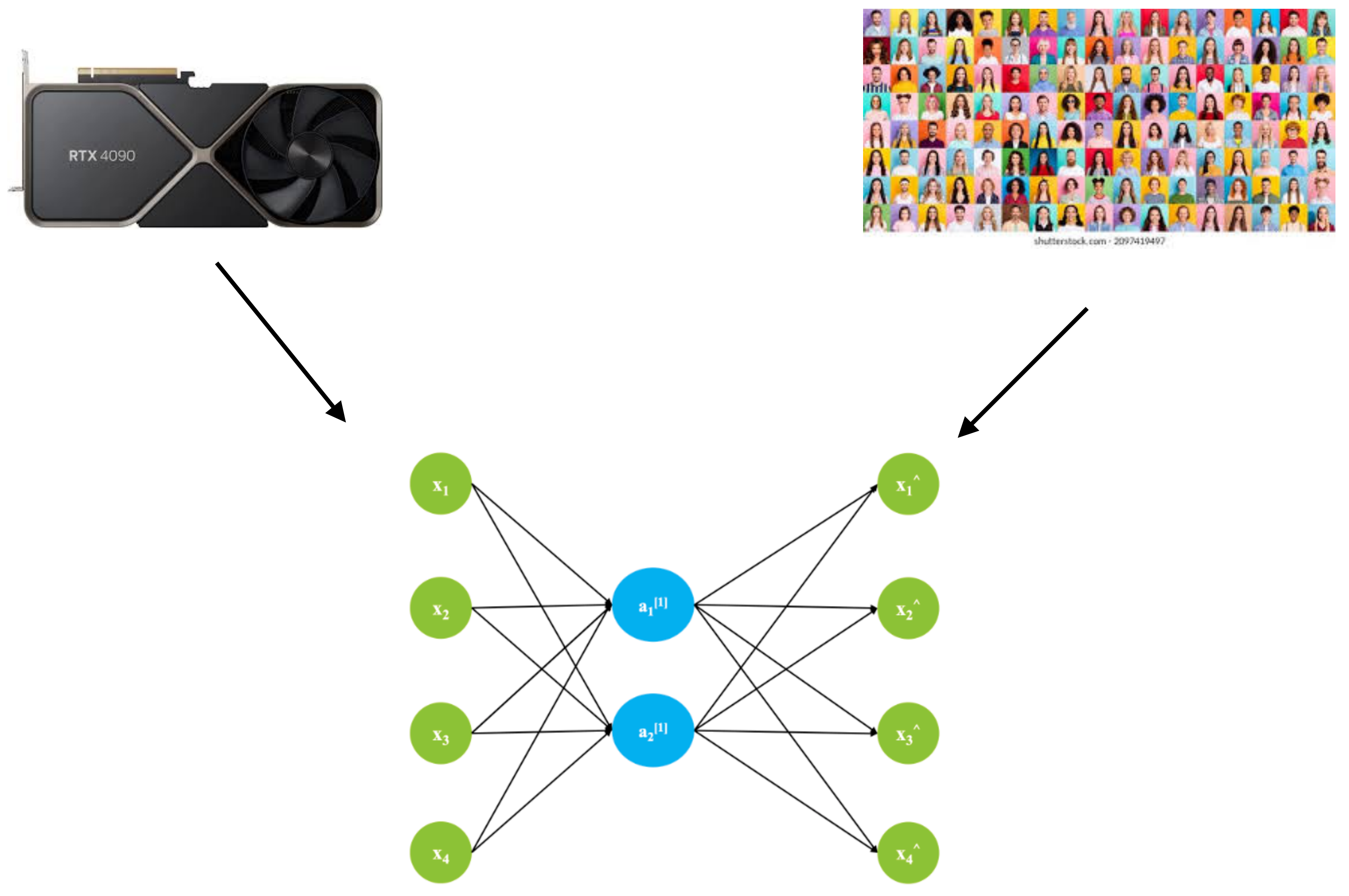


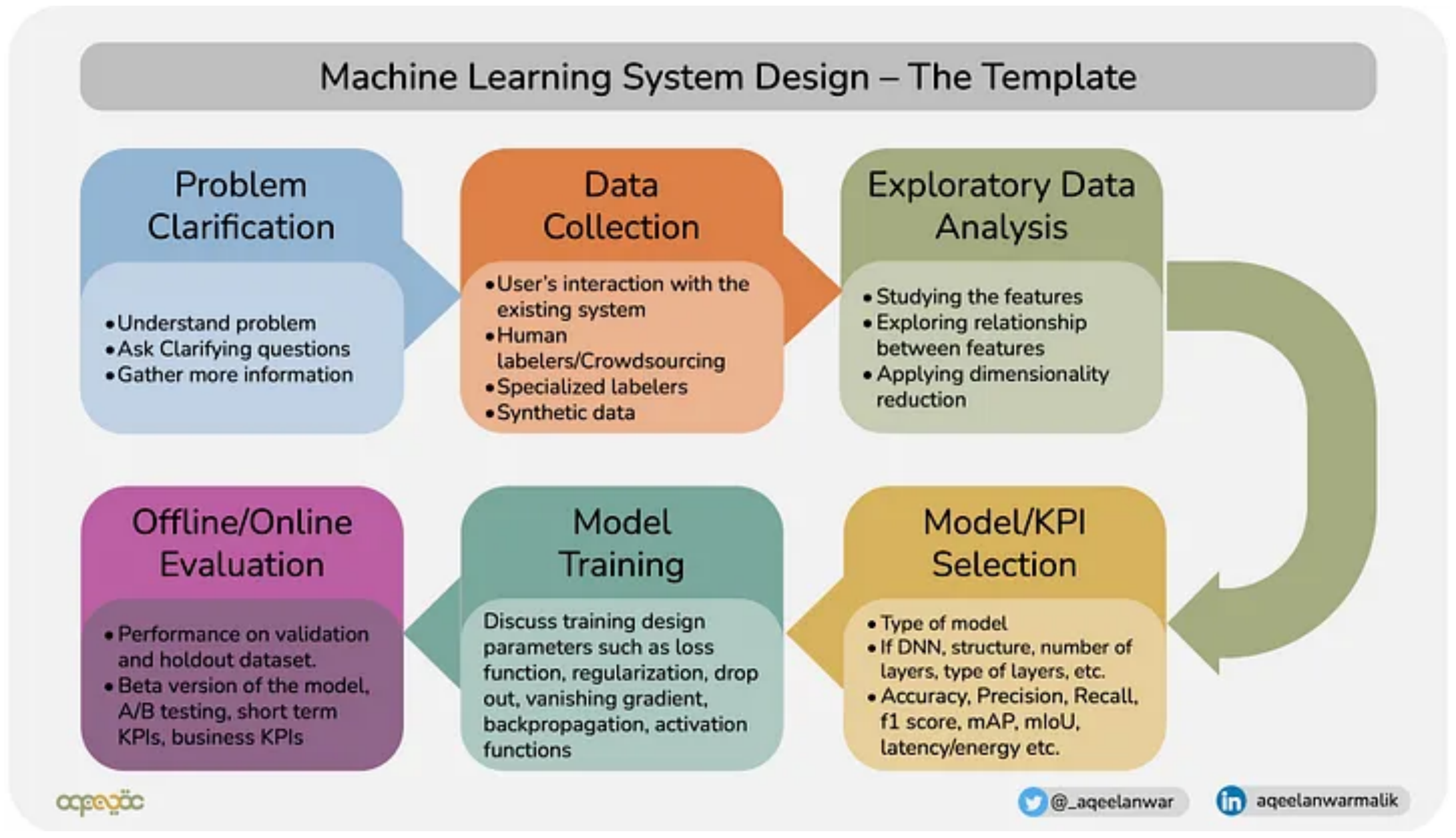
Architecture


O'REILLY
Designing Machine Learning Systems
An Iterative Process for Production-Ready Applications
Chip Huyen

Machine Learning Systems

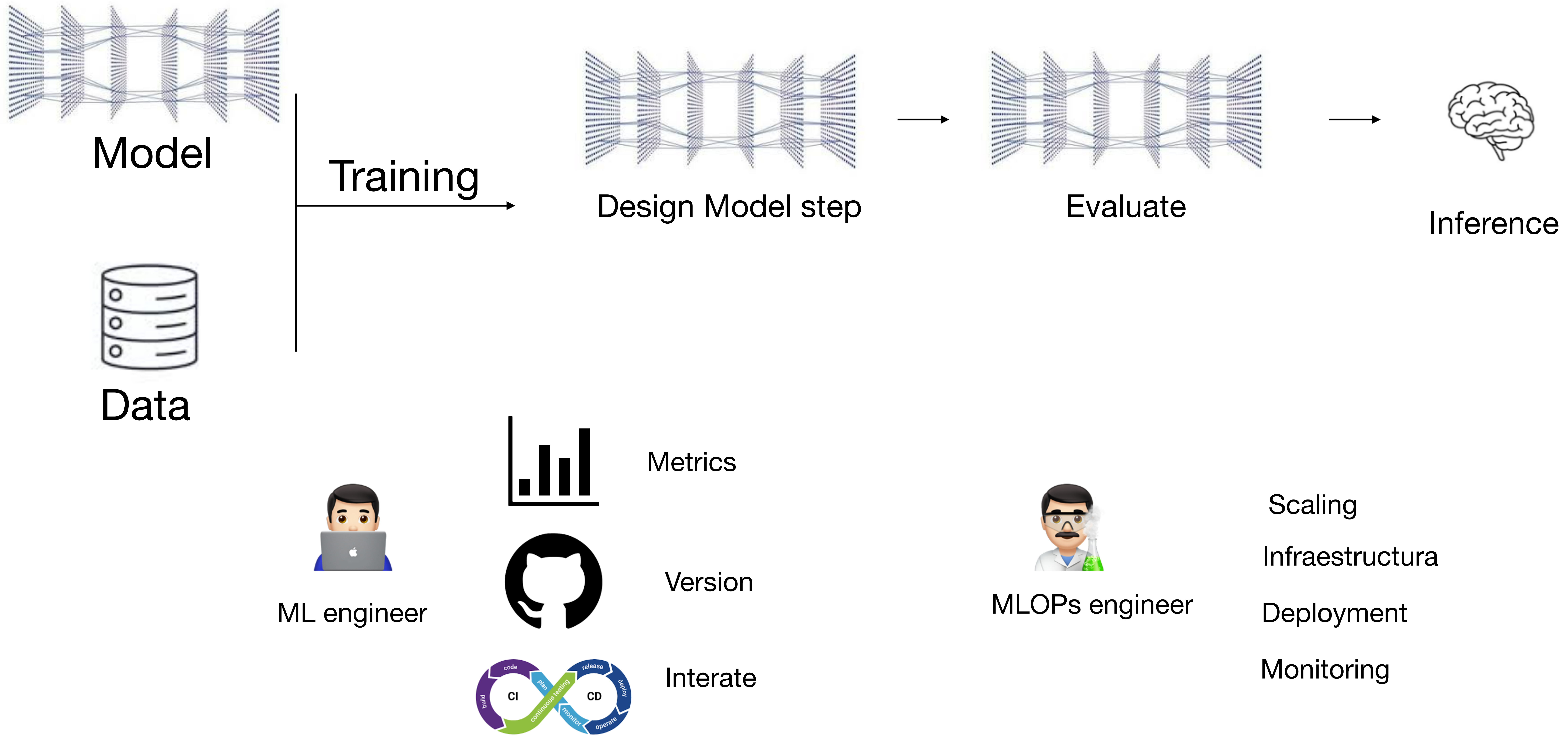
Motivation




Machine Learning System Design – The Template
Problem Clarification
• Understand problem
• Ask Clarifying questions
• Gather more information
Data Collection
• User's interaction with the existing system
• Human labelers/Crowdsourcing
• Specialized labelers
• Synthetic data
Exploratory Data Analysis
• Studying the features
• Exploring relationship between features
• Applying dimensionality reduction
Model/KPI Selection
• Type of model
• If DNN, structure, number of layers, type of layers, etc.
• Accuracy, Precision, Recall, f1 score, mAP, mIoU, latency/energy etc.
Model Training
Discuss training design parameters such as loss function, regularization, drop out, vanishing gradient, backpropagation, activation functions
Offline/Online Evaluation
• Performance on validation and holdout dataset.
• Beta version of the model, A/B testing, short term KPIs, business KPIs
@_aqeelanwar
aqeelanwarmalik

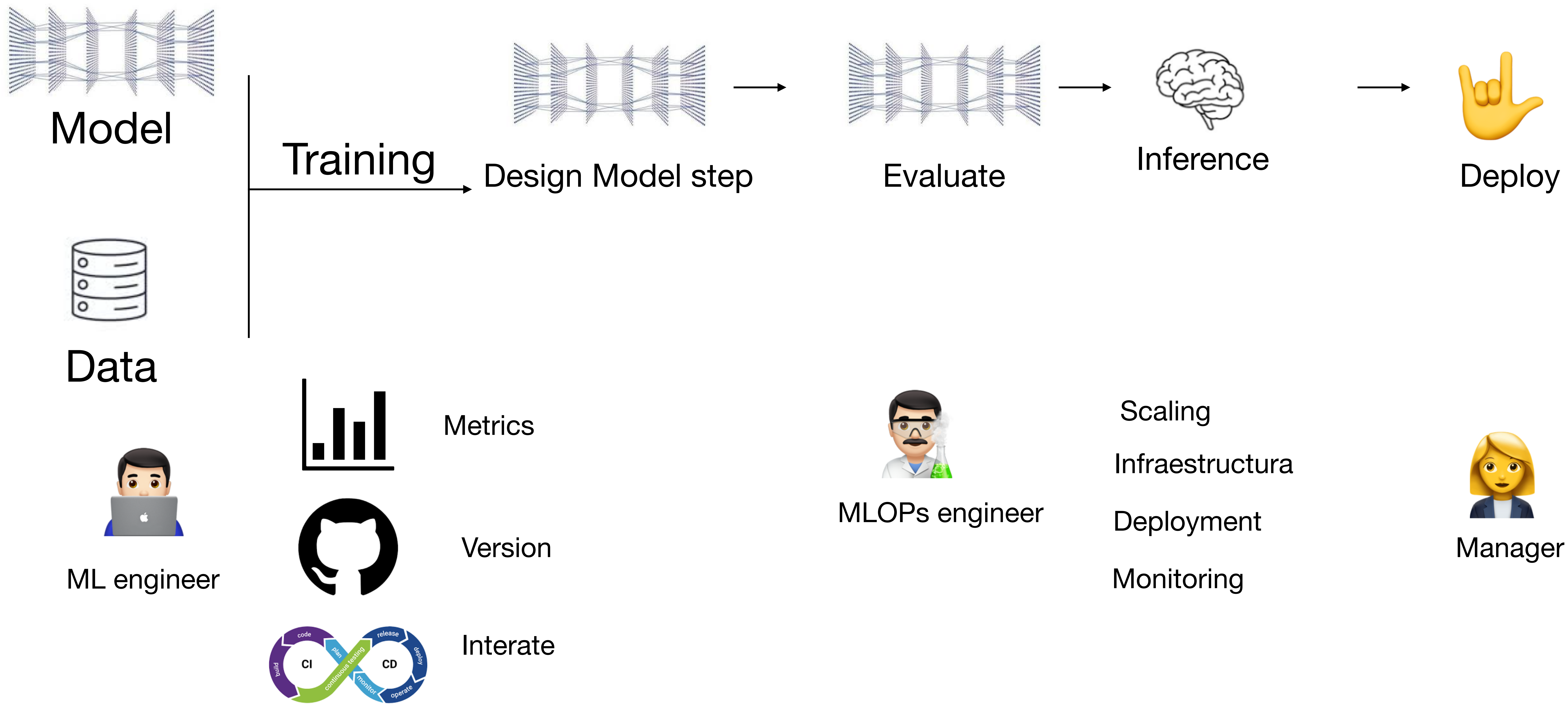
<u>Machine Learning System Design — The template</u>

Motivation

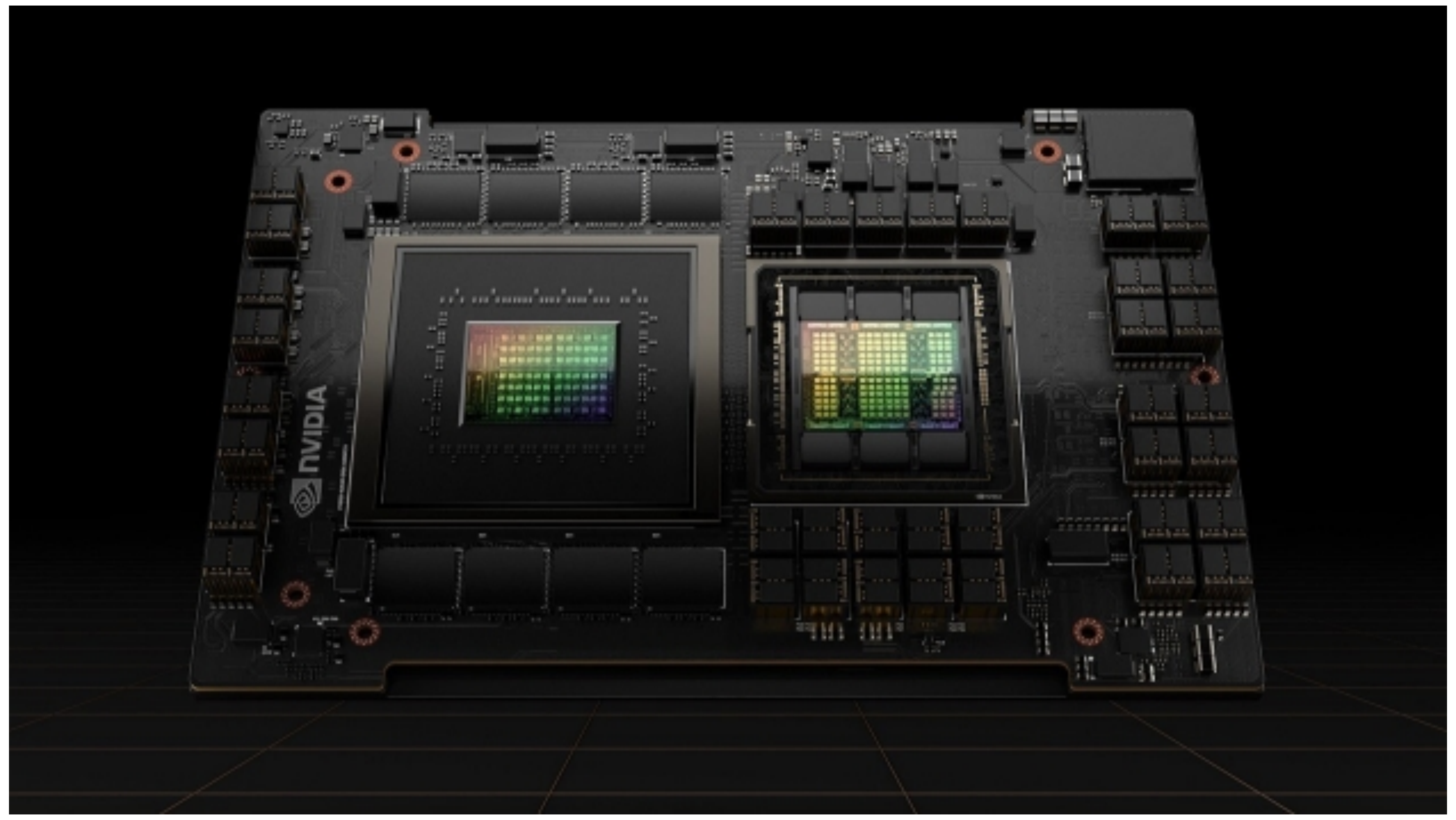


Model
Data
Training
Design Model step
Evaluate
Inference
Metrics
Version
Interate
code
plan
release
deploy
build
continuous testing
monitor
operate
CI
CD
ML engineer
MLOPs engineer
Scaling
Infraestructura
Deployment
Monitoring

Motivation


Model
Data
Training
Design Model step
Evaluate
Inference
Deploy
ML engineer
Metrics
Version
code
plan
build
CI
continuous testing
monitor
release
deploy
CD
operate
Interate
MLOPs engineer
Scaling
Infraestructura
Deployment
Monitoring
Manager

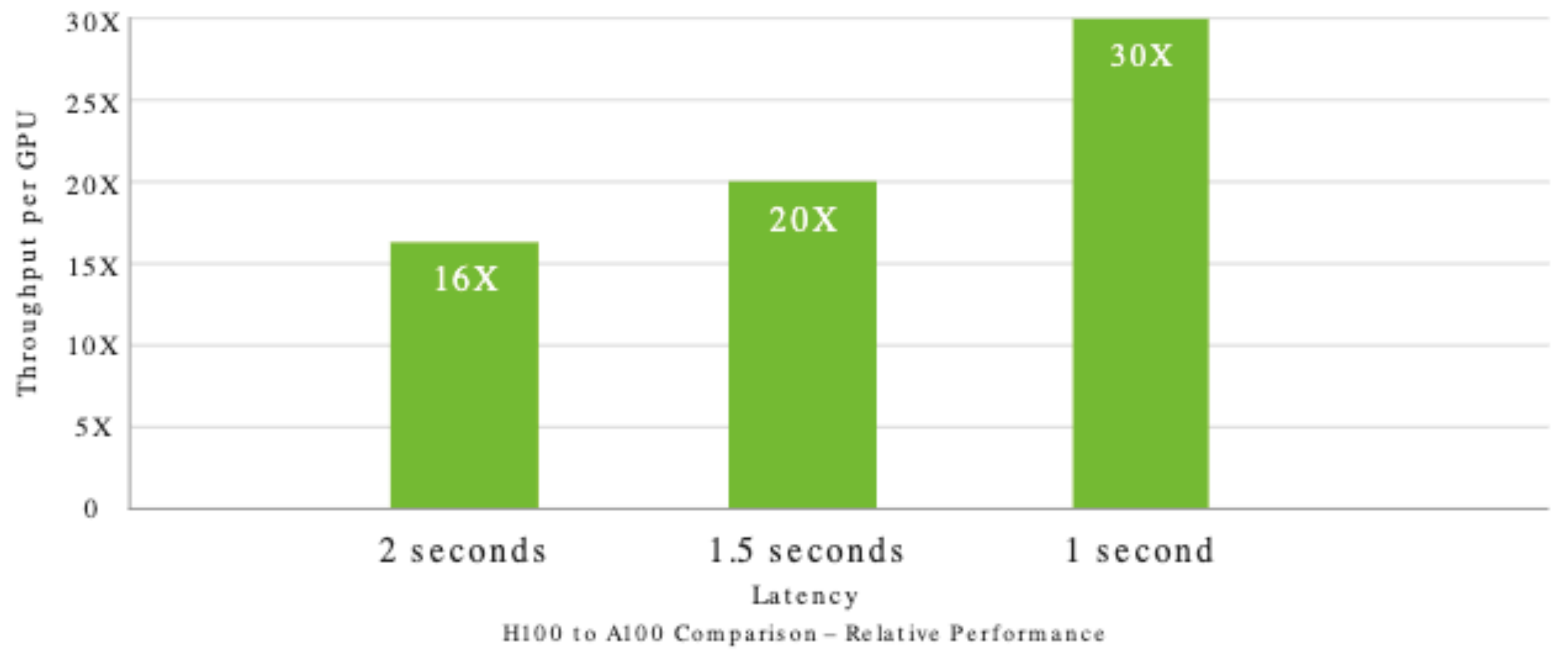
Motivation



Hopper 100

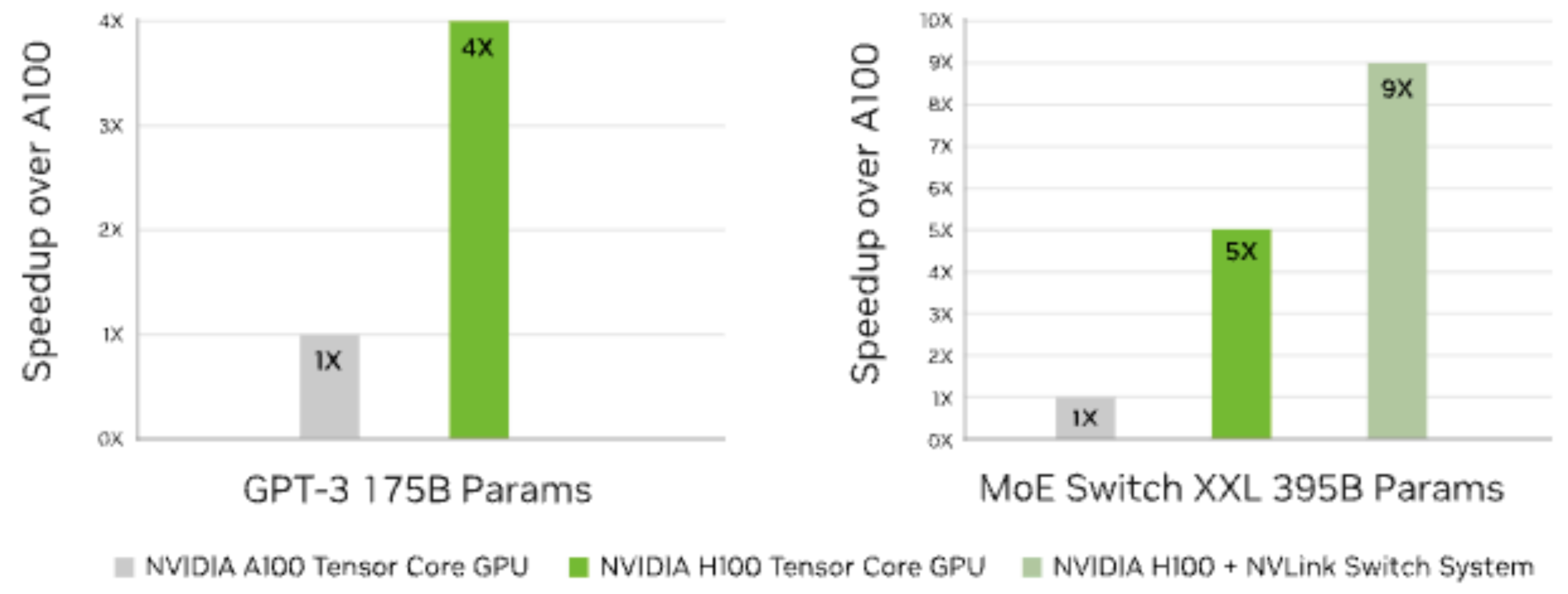
Rendimiento de Inferencia de IA hasta 30 Veces Mayor en los Modelos Más Grandes

Inferencia del chatbot Megatron (530 mil millones de parámetros)



Rendimiento proyectado sujeto a cambios. Inferencia en el chatbot basado en el modelo de parámetros Megatron 530B para longitud de secuencia de entrada = 128, longitud de secuencia de salida = 20 | Clúster A100: red HDR IB | Clúster H100: sistema de conmutador NVLink, NDR IB

Entrenamiento de IA Hasta 4 Veces Superior en GPT-3



Rendimiento proyectado sujeto a cambios. Entrenamiento GPT-3 175B Clúster A100: red HDR IB, clúster H100: red NDR IB | Variante del Switch-XXL del transformador de capacitación de mezcla de expertos (MoE) con parámetros 395B en un conjunto de datos de token de 1T, clúster A100: red HDR IB, clúster H100: red NDR IB con sistema de conmutador NVLink donde se indique.

AI Computing

- Mathematically intensive process to calculate machine learning algorithms using accelerated hardware.
- Three Elements to AI Computing
  - ELT
  - Model Design
  - Inference

Challenges of experimentation

- Making the pipeline of models for inference efficient
- Tracking model experimentation
- GPU-CPU monitoring
- Understanding time consumption per algorithm phase
- Resource configuration for the problem being addressed (Machine Learning system)

Weights and Bias Platform

Weights and Bias

- https://wandb.ai/site
- Build models faster, fine-tune LLMs, develop GenAI applications with confidence, all in one system of record developers are excited to use. (https://wandb.ai/site)

Weights and Bias VS MLflow

Weights and Bias : It is specifically designed for tracking machine learning experiments.

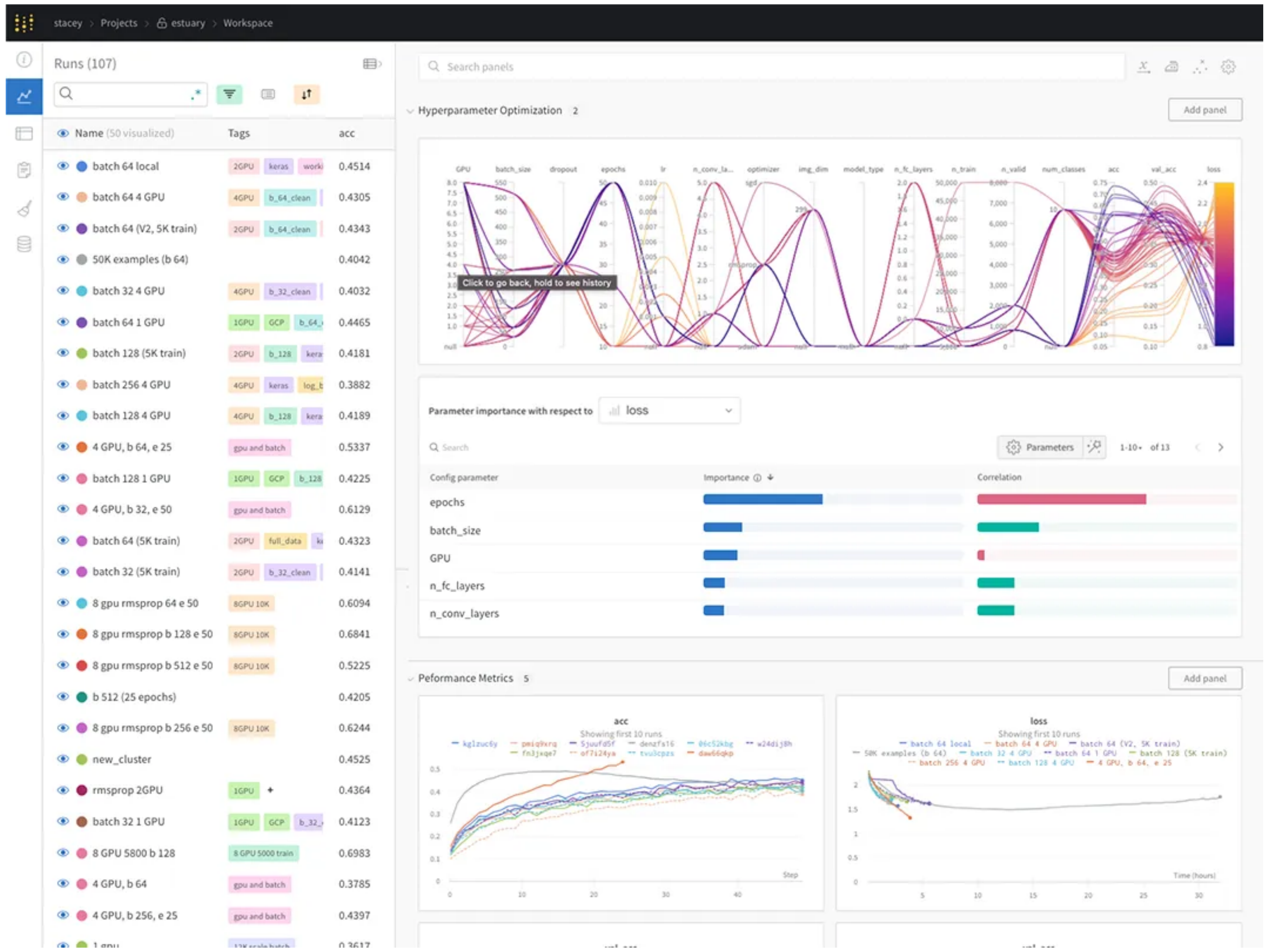
MLflow: It follows the complete lifecycle of machine learning, including experiment tracking, model management, and deployment

Weights and Bias Platform

Function	Tool
Code Tracking	Github, Gitlab, Bitbucket
Experiment tracking	Weights and bias , MLflow
Artifact tracking	Weights and bias
Model repository	Weights and bias
ML pipelines	MLflow
Environment isolation	Conda - Docker
Orchestration	MLflow
Modelling	Tensorflow, Pytorch, scikit-learn

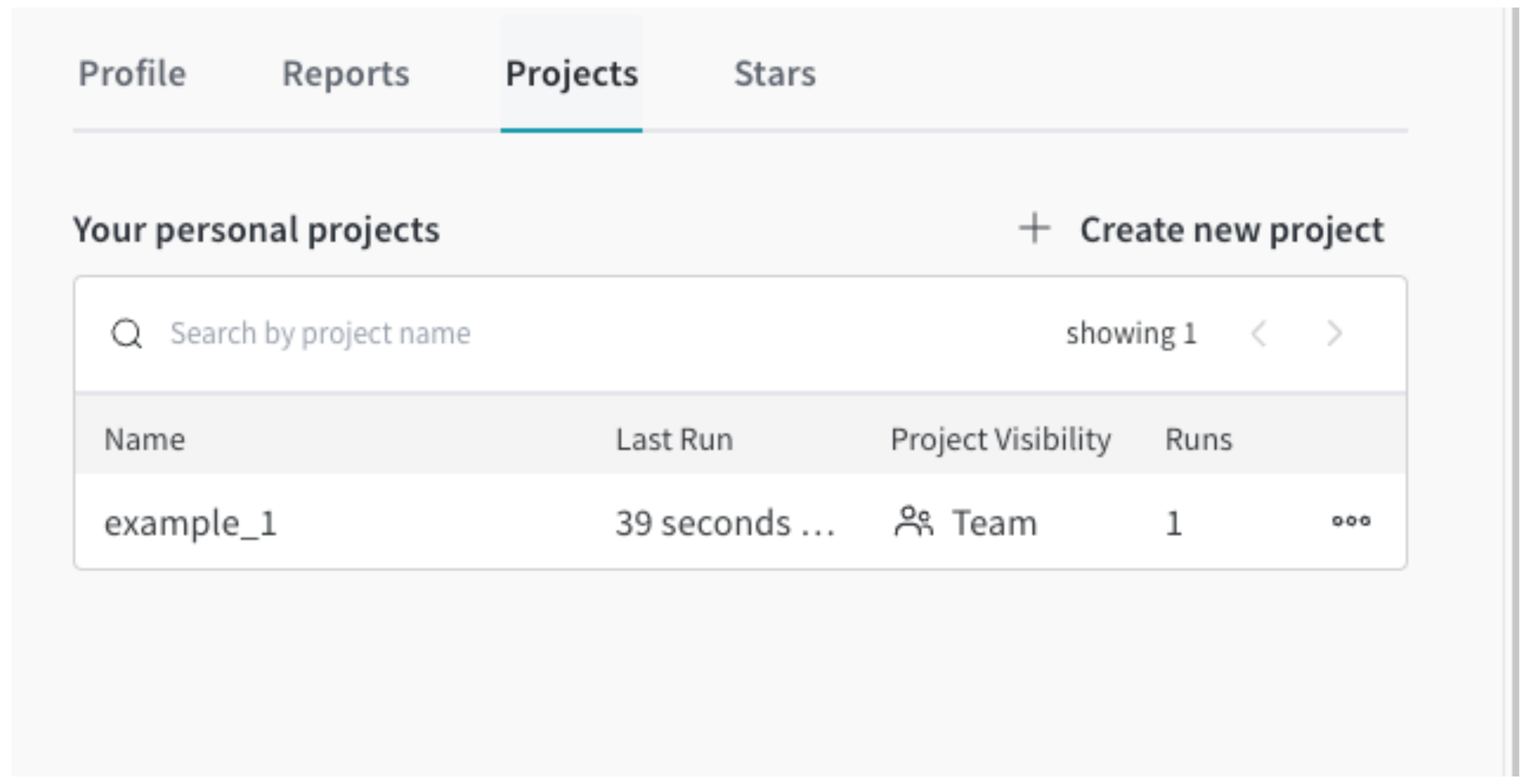
Weights and Bias Platform

- Work is done under artifacts (best practices).
- Pipelines should yield different artifacts as results.
- Artifacts must be versioned, tracked, and saved.


stacey › Projects › estuary › Workspace
Search panels
Hyperparameter Optimization 2
Add panel
Click to go back, hold to see history
Parameter importance with respect to loss
Parameters
Config parameter
Importance
Correlation
epochs
batch_size
GPU
n_fc_layers
n_conv_layers
Peformance Metrics 5
acc
loss

Runs (107)

Name (50 visualized)	Tags	acc
batch 64 local	2GPU keras worki	0.4514
batch 64 4 GPU	4GPU b_64_clean	0.4305
batch 64 (V2, 5K train)	2GPU b_64_clean	0.4343
50K examples (b 64)		0.4042
batch 32 4 GPU	4GPU b_32_clean	0.4032
batch 64 1 GPU	1GPU GCP b_64_	0.4465
batch 128 (5K train)	2GPU b_128 kera	0.4181
batch 256 4 GPU	4GPU keras log_b	0.3882
batch 128 4 GPU	4GPU b_128 kera	0.4189
4 GPU, b 64, e 25	gpu and batch	0.5337
batch 128 1 GPU	1GPU GCP b_128	0.4225
4 GPU, b 32, e 50	gpu and batch	0.6129
batch 64 (5K train)	2GPU full_data k	0.4323
batch 32 (5K train)	2GPU b_32_clean	0.4141
8 gpu rmsprop 64 e 50	8GPU 10K	0.6094
8 gpu rmsprop b 128 e 50	8GPU 10K	0.6841
8 gpu rmsprop b 512 e 50	8GPU 10K	0.5225
b 512 (25 epochs)		0.4205
8 gpu rmsprop b 256 e 50	8GPU 10K	0.6244
new_cluster		0.4525
rmsprop 2GPU	1GPU +	0.4364
batch 32 1 GPU	1GPU GCP b_32_	0.4123
8 GPU 5800 b 128	8 GPU 5000 train	0.6983
4 GPU, b 64	gpu and batch	0.3785
4 GPU, b 256, e 25	gpu and batch	0.4397
1 gpu	12K scale batch	0.3617

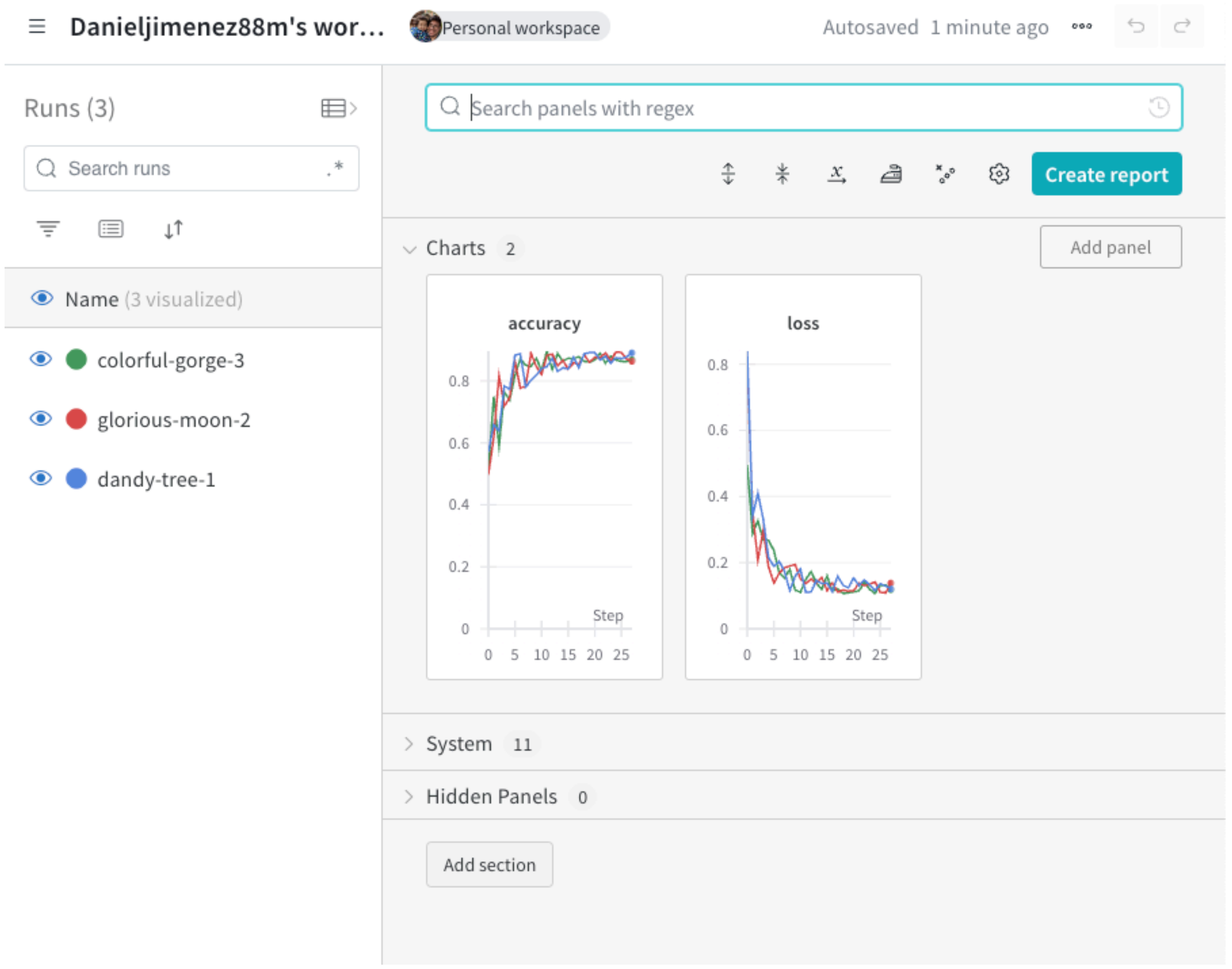


Profile | Reports | Projects | Stars

Your personal projects

Name	Last Run	Project Visibility	Runs
example_1	39 seconds ...	Team	1

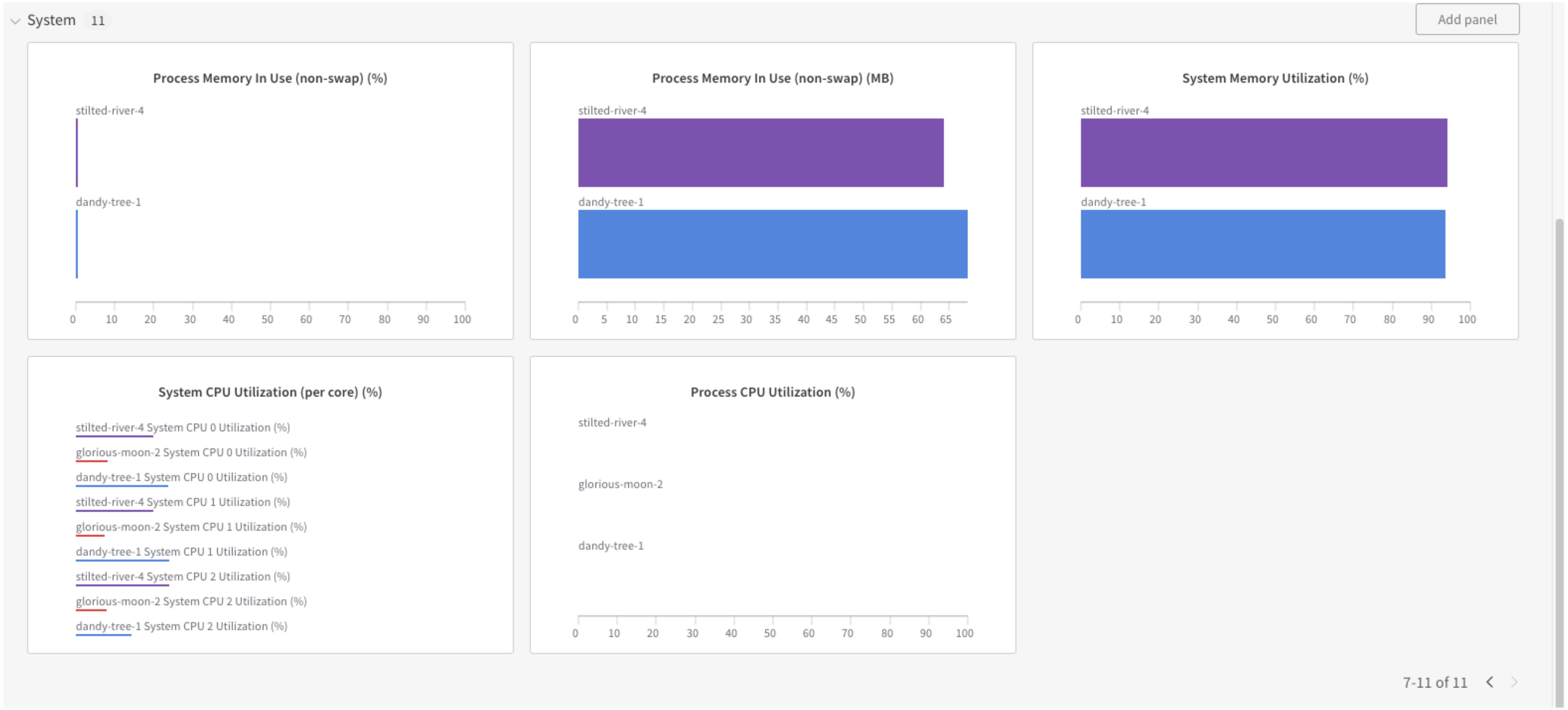
Experiments names



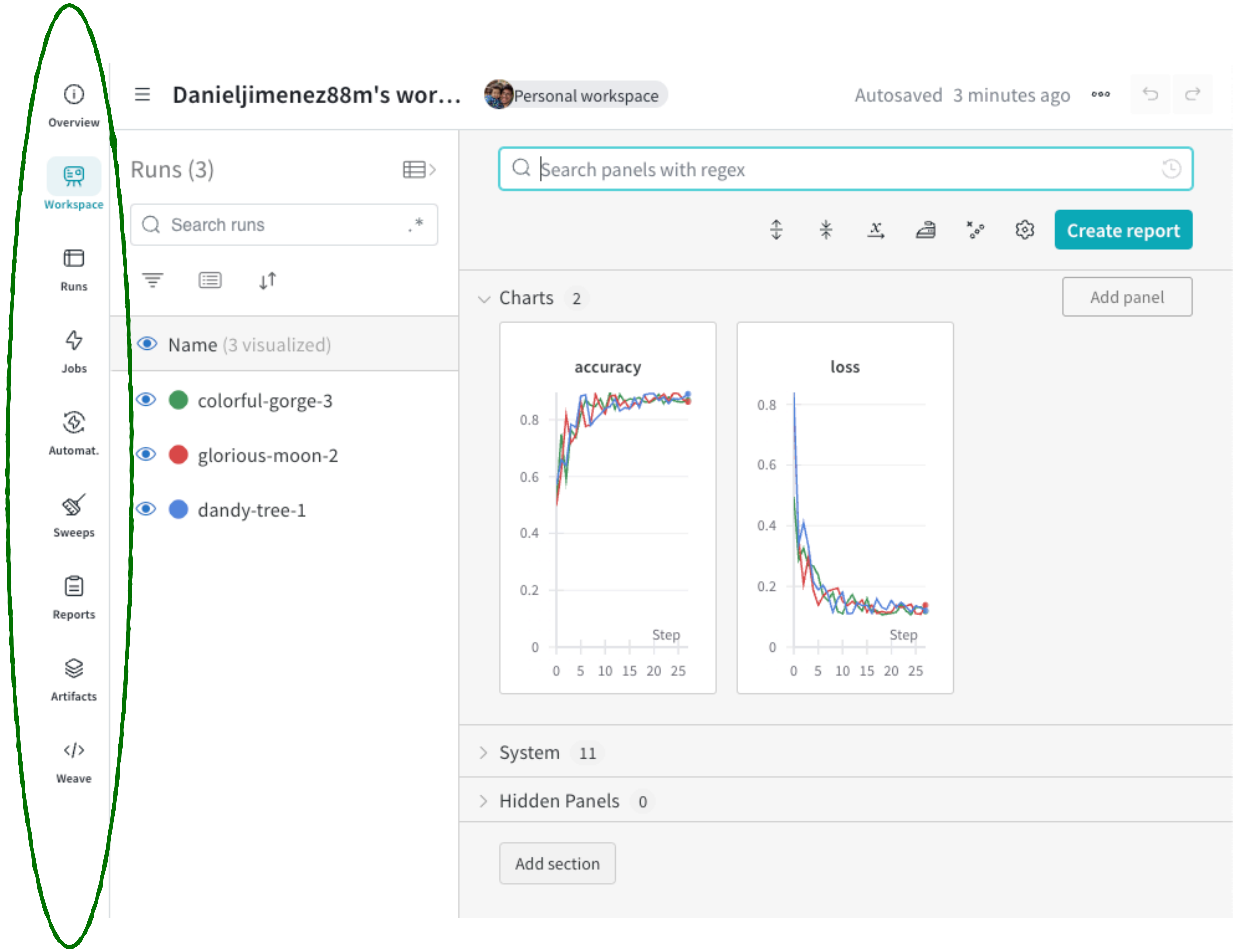
Metrics logs

Runs (4)
Search runs
Name (4 visualized)
stilted-river-4
colorful-gorge-3
glorious-moon-2
dandy-tree-1
Search panels with regex
Create report
System 11
Add panel
Disk I/O Utilization (MB)
stilted-river-4 MB read from disk
glorious-moon-2 MB read from disk
stilted-river-4 MB written to disk
glorious-moon-2 MB written to disk
Disk Utilization (GB)
Disk Utilization (%)
Network Traffic (bytes)
stilted-river-4 Network Traffic Received (bytes)
dandy-tree-1 Network Traffic Received (bytes)
stilted-river-4 Network Traffic Sent (bytes)
dandy-tree-1 Network Traffic Sent (bytes)
Process CPU Threads In Use
Process Memory Available (non-swap) (MB)
1-6 of 11
Hidden Panels 0

System Performs


System 11
Add panel
Process Memory In Use (non-swap) (%)
stilted-river-4
dandy-tree-1
0 10 20 30 40 50 60 70 80 90 100
Process Memory In Use (non-swap) (MB)
stilted-river-4
dandy-tree-1
0 5 10 15 20 25 30 35 40 45 50 55 60 65
System Memory Utilization (%)
stilted-river-4
dandy-tree-1
0 10 20 30 40 50 60 70 80 90 100
System CPU Utilization (per core) (%)
stilted-river-4 System CPU 0 Utilization (%)
glorious-moon-2 System CPU 0 Utilization (%)
dandy-tree-1 System CPU 0 Utilization (%)
stilted-river-4 System CPU 1 Utilization (%)
glorious-moon-2 System CPU 1 Utilization (%)
dandy-tree-1 System CPU 1 Utilization (%)
stilted-river-4 System CPU 2 Utilization (%)
glorious-moon-2 System CPU 2 Utilization (%)
dandy-tree-1 System CPU 2 Utilization (%)
Process CPU Utilization (%)
stilted-river-4
glorious-moon-2
dandy-tree-1
0 10 20 30 40 50 60 70 80 90 100
7-11 of 11

System Performs

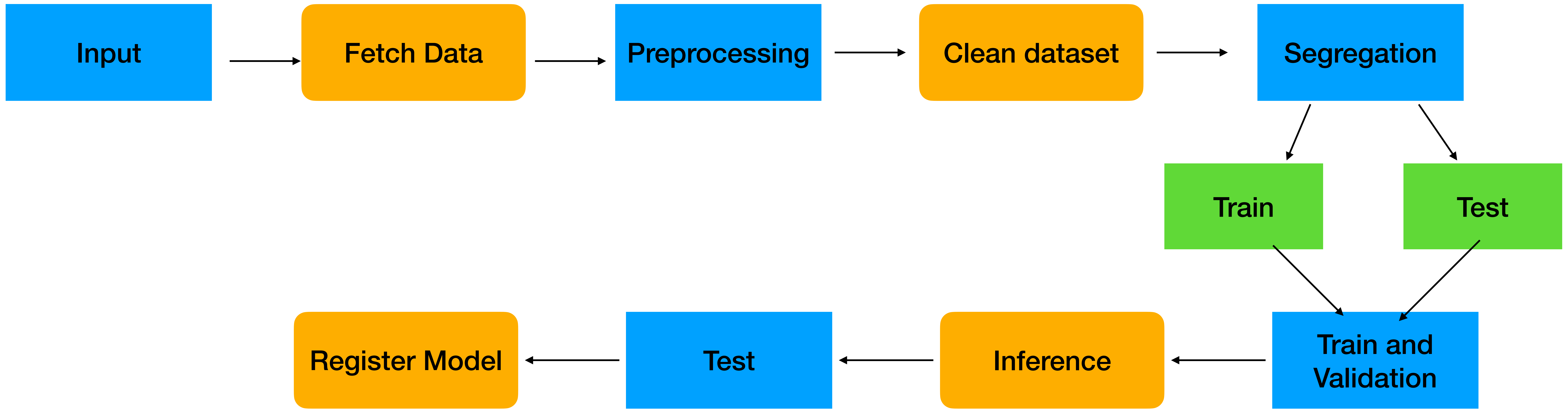

Danieljimenez88m's wor... Personal workspace Autosaved 3 minutes ago

General Options

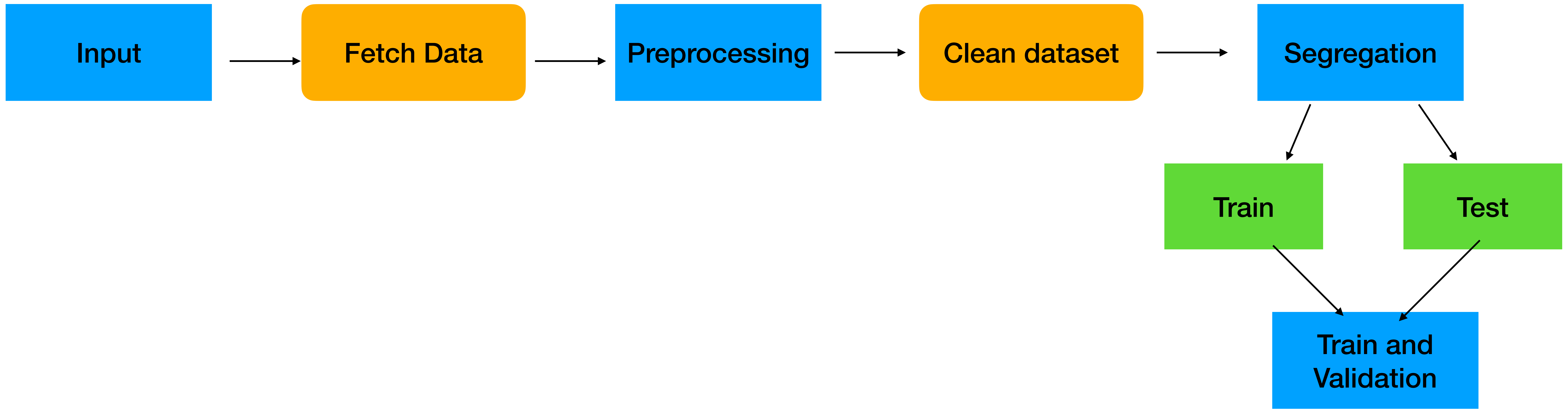
Weights and Bias Propose Roadmap

- Generate a Infrastructure code like a system with Artifacts
- We will follow a design path focused on data science engineering, although we will not build an API.
- It will end with the registration of a model, to be used in inference.

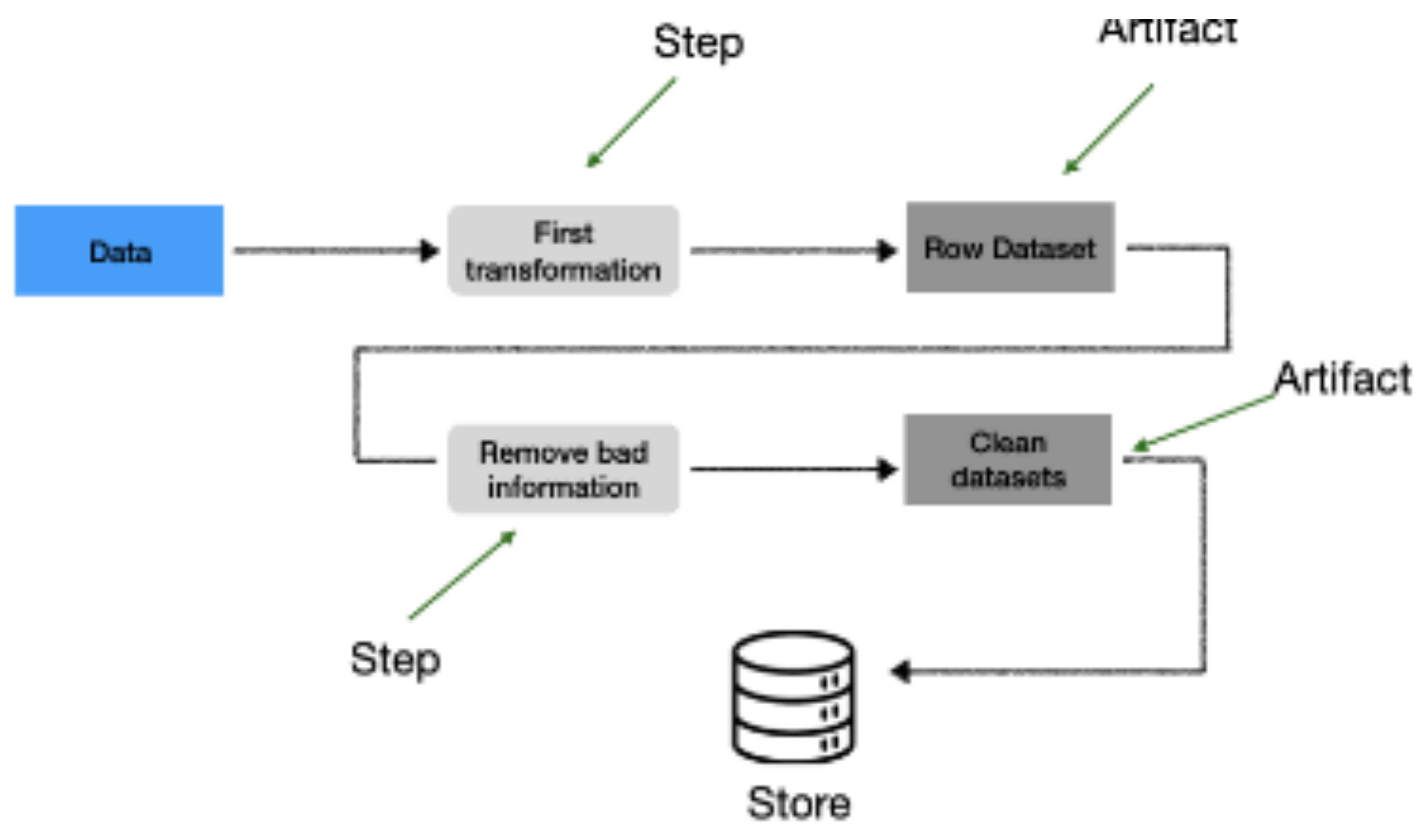
Weights and Bias Propose Roadmap



Weights and Bias Propose Roadmap segmentation



Weights and Bias Platform



Artifacts: These are the products resulting from the components. Artifacts can be used as inputs in subsequent components, thus creating connections between the different steps of the pipeline. Artifacts should be carefully tracked and versioned to ensure traceability and efficient management of the different model development versions.

Weights and Bias Example

Roadmap

$$\begin{tikzcd}
\text{https://wandb.ai/site} \arrow[r] & \text{Register} \arrow[r] & \text{Platform} \\
\text{wandb login} \arrow[r] & \text{Login} \arrow[r] & \text{Terminal or cmd} \arrow[dl] \\
 & \text{Generally, they send a link where there's a code that you must paste into the terminal in order to access} &
\end{tikzcd}$$

Roadmap

Roadmap

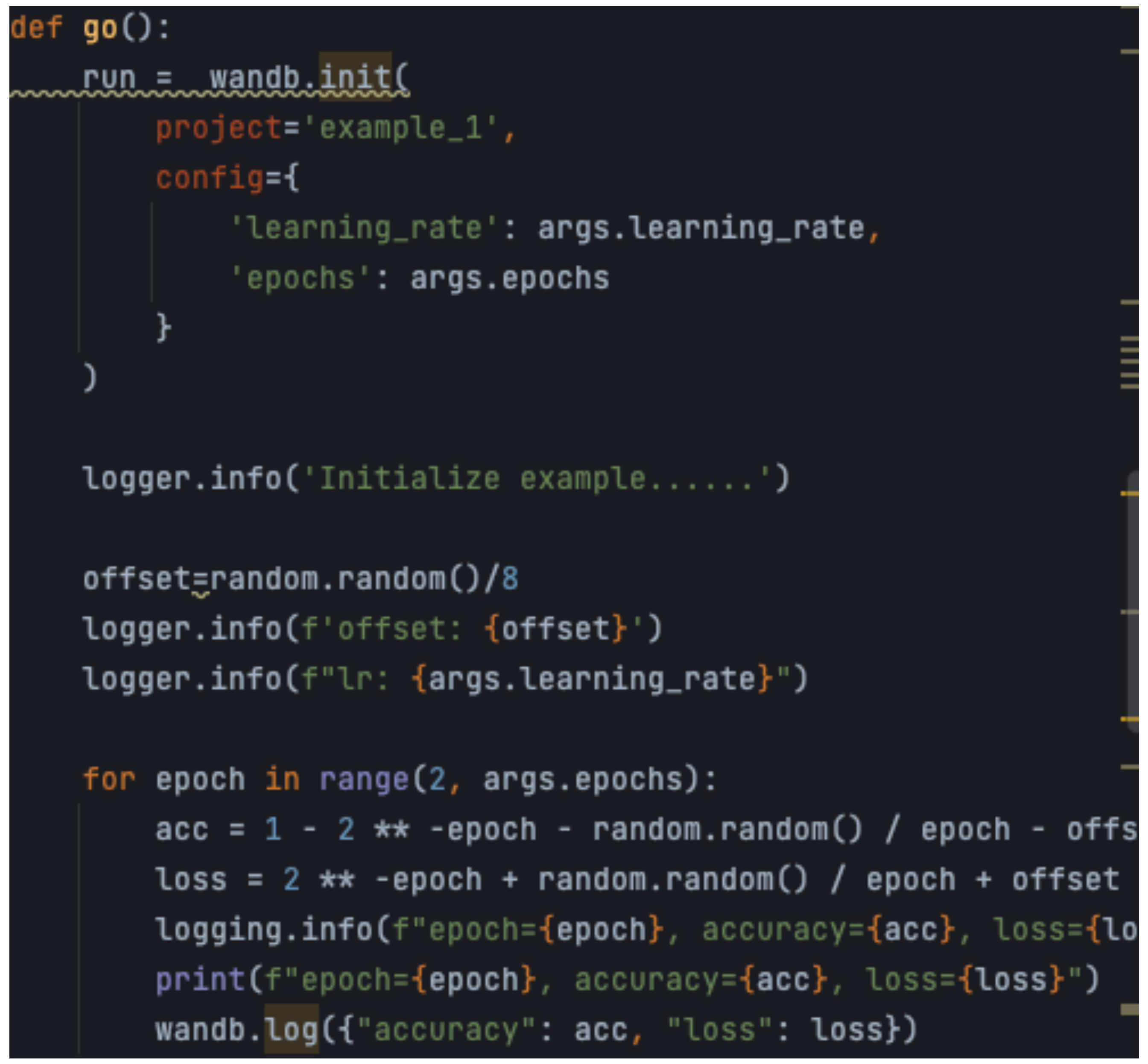
Python Scripting with Argparse

Python Scripting with Argparse

- Automating tasks
- Using the command line for customization of situations in front of model design
- Flexible code configuration
- By having a script with Argparse that contains all the necessary steps to train and evaluate your model, other users can easily reproduce your experiments and results

Python Scripting with Argparse

Colocar el link del código



```
def go():
    run =  wandb.init(
        project='example_1',
        config={
            'learning_rate': args.learning_rate,
            'epochs': args.epochs
        }
    )

    logger.info('Initialize example......')

    offset=random.random()/8
    logger.info(f'offset: {offset}')
    logger.info(f"lr: {args.learning_rate}")

    for epoch in range(2, args.epochs):
        acc = 1 - 2 ** -epoch - random.random() / epoch - offs
        loss = 2 ** -epoch + random.random() / epoch + offset
        logging.info(f"epoch={epoch}, accuracy={acc}, loss={lo
        print(f"epoch={epoch}, accuracy={acc}, loss={loss}")
        wandb.log({"accuracy": acc, "loss": loss})
```

Roadmap into WandB

wand.init()

wand.log()

wand.log_artifact()

Roadmap into WandB

```
wand.init(
project=‘name’,
job_type=‘some amazing process’,
group=‘experiment_final_final_ya_no_va_más’
)
```

```
wandb.Artifact(
name=args.artifact_name,
type=args.artifact_type,
description=args.artifact_description,
metadata={'original_url': args.file_url})
```

Now let's write code!!! That's what we came here for."

https://github.com/Carlos-Jimenez-mlops/wandb-mlops