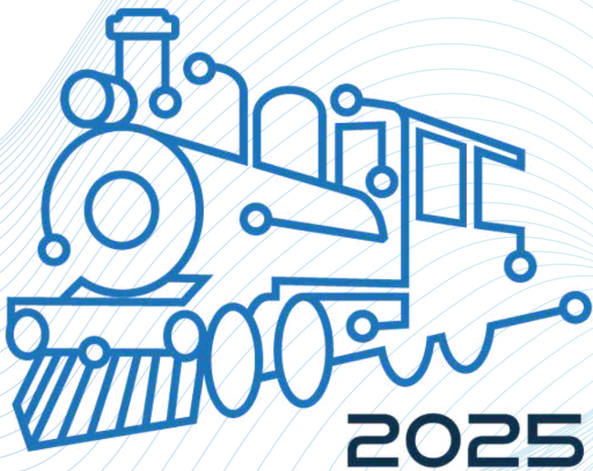
# MÉTODOS AVANÇADOS DE TINYMLOPS: IMPLEMENTAÇÃO DE APRENDIZADO DE MÁQUINA EM SISTEMAS EMBARCADOS

**THOMMAS K. S. FLORES**
IVANNOVITH SILVA
DANIEL G. COSTA



2025
SBAI
XVII Simpósio Brasileiro de Automação Inteligente

**2025**

# HOW I AM

## THOMMAS KEVIN SALES FLORES

### ACADEMIC CAREER

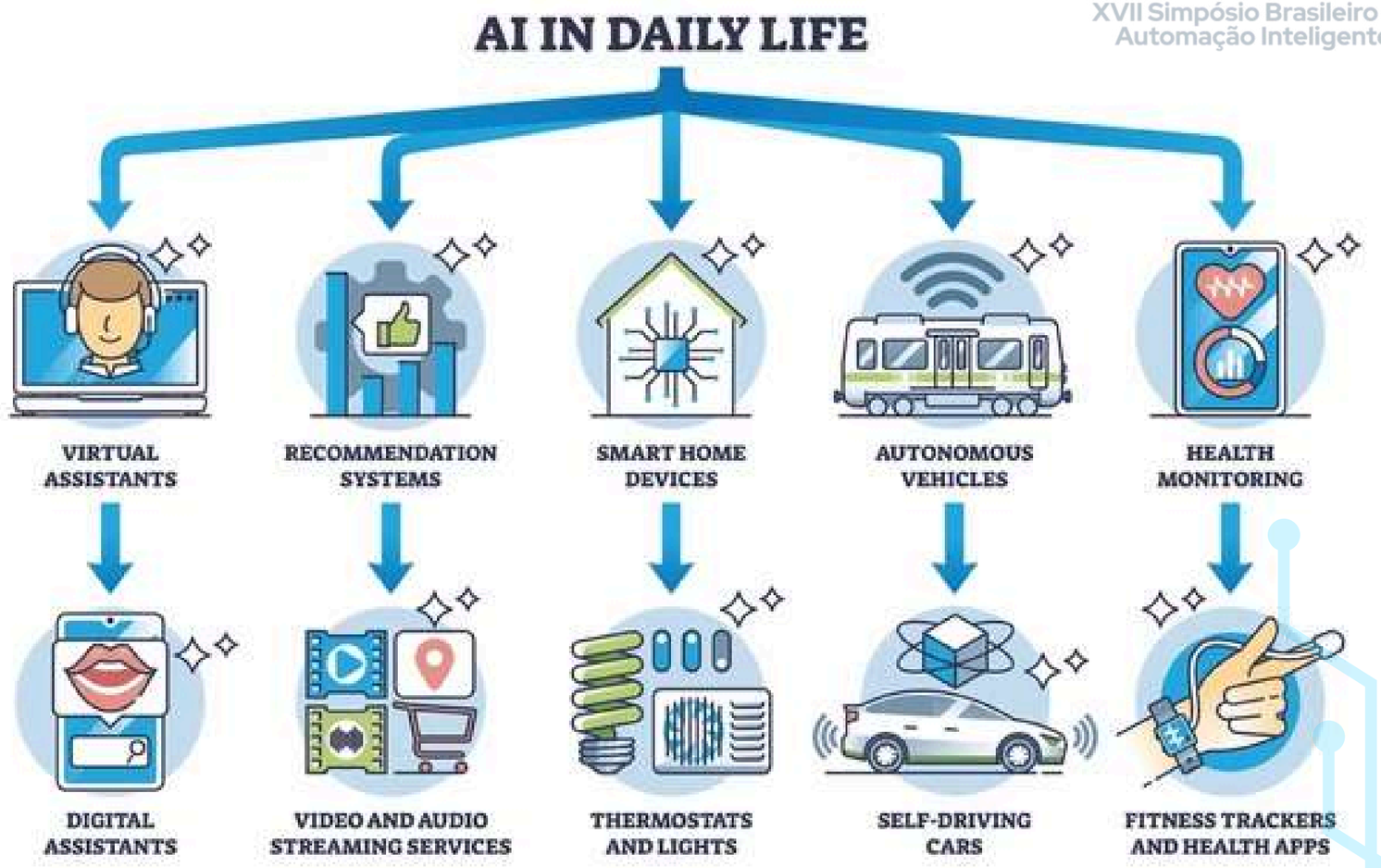
**Doutorando em Engenharia Elétrica e de Computação - UFRN**
Doutorando Sanduiche em Engenharia Elétrica e de Computação - FEUP
Mestre em Engenharia Elétrica - UFPB
Bacharel em Engenharia Elétrica - UFPB

### PROFESSIONAL CAREER

Analista de Tecnologias Operacionais - PROSYS
Trainee em Engenharia Elétrica - NORFIL

# WHAT IS IA?

**Artificial Intelligence (AI)** refers to the ability of machines to simulate human-like cognitive functions such as learning, reasoning, problem-solving, and decision-making



# INTRODUCTION

## Artificial Intelligence

Is the field of study

## Machine Learning

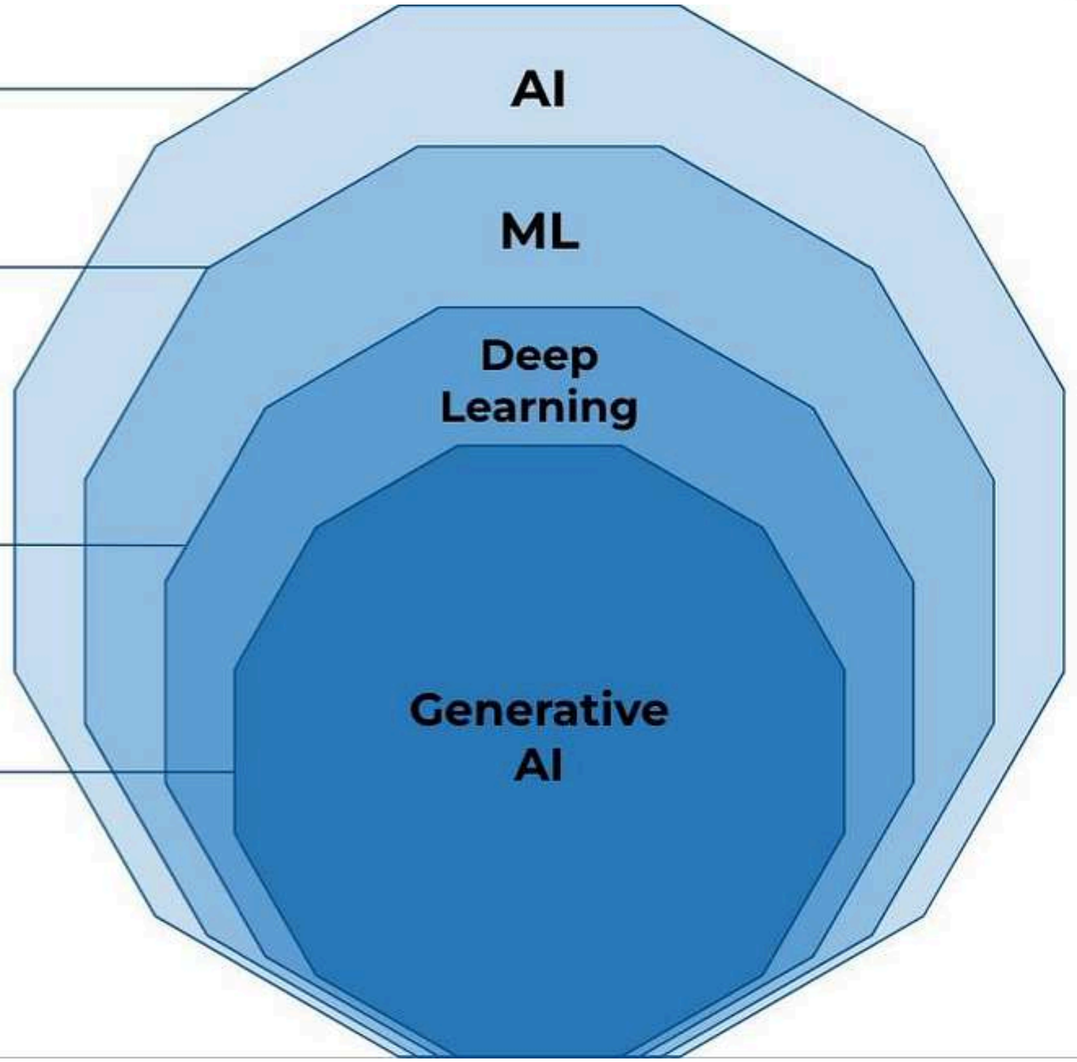
Is a branch of AI that focus on the creation of intelligent machines that learn from data. Another very well know branch inside AI is **Optimization**.

## Deep Learning

Is a subset of Machine Learning methods, based on **Artificial Neural Networks.** Examples: CNNs, RNNs

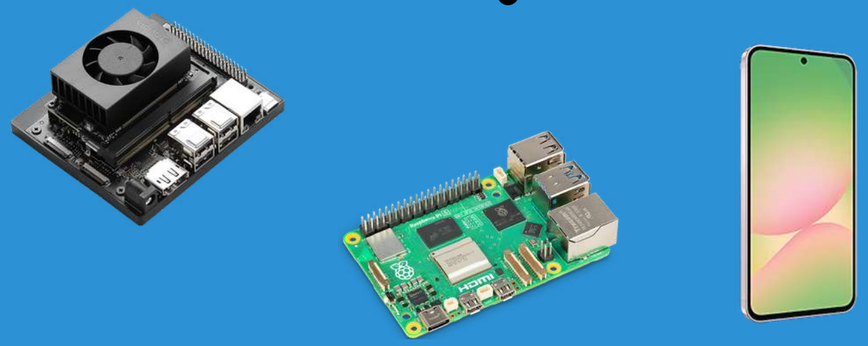
## Generative AI

A type of ANNs that generate data that is similar to the data it was trained on. Examples: GANs, LLMs



# INTRODUCTION

## EdgeAI (or EdgeML)



Refers to the deployment of AI algorithms directly on local edge devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure.

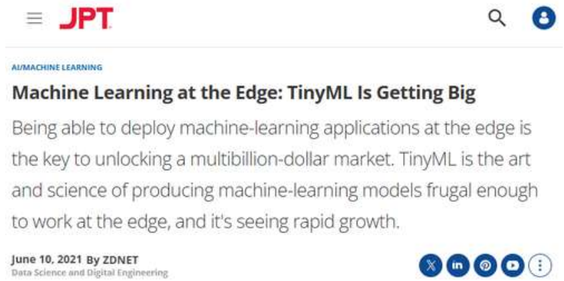
## TinyML

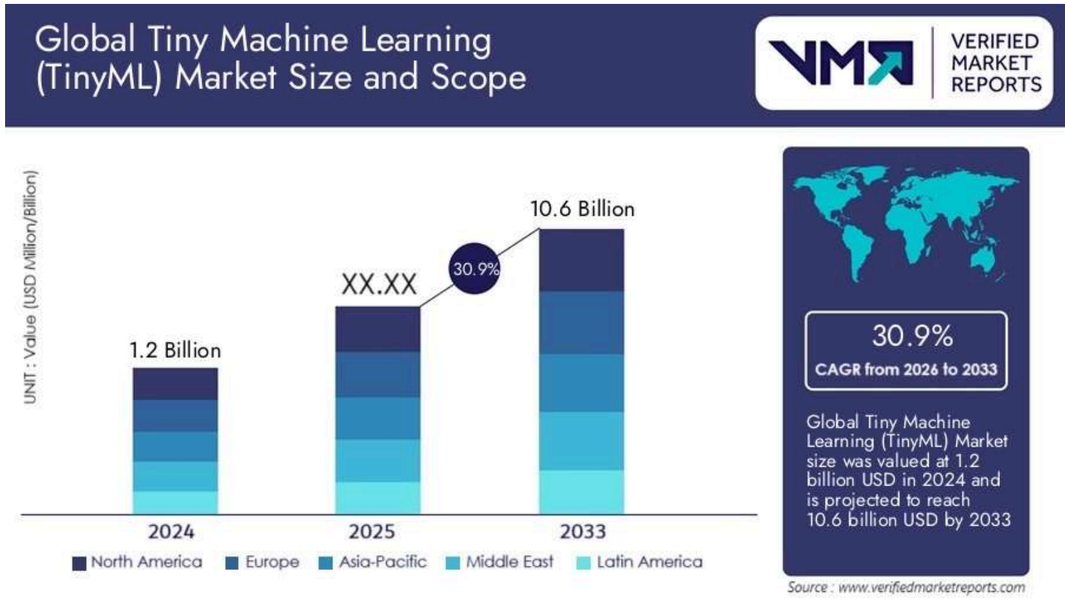
Focused on running machine learning models on resource-constrained devices like microcontrollers and other low-power devices.

# WHY SHOULD I STUDY TINYML?

Fastest-growing

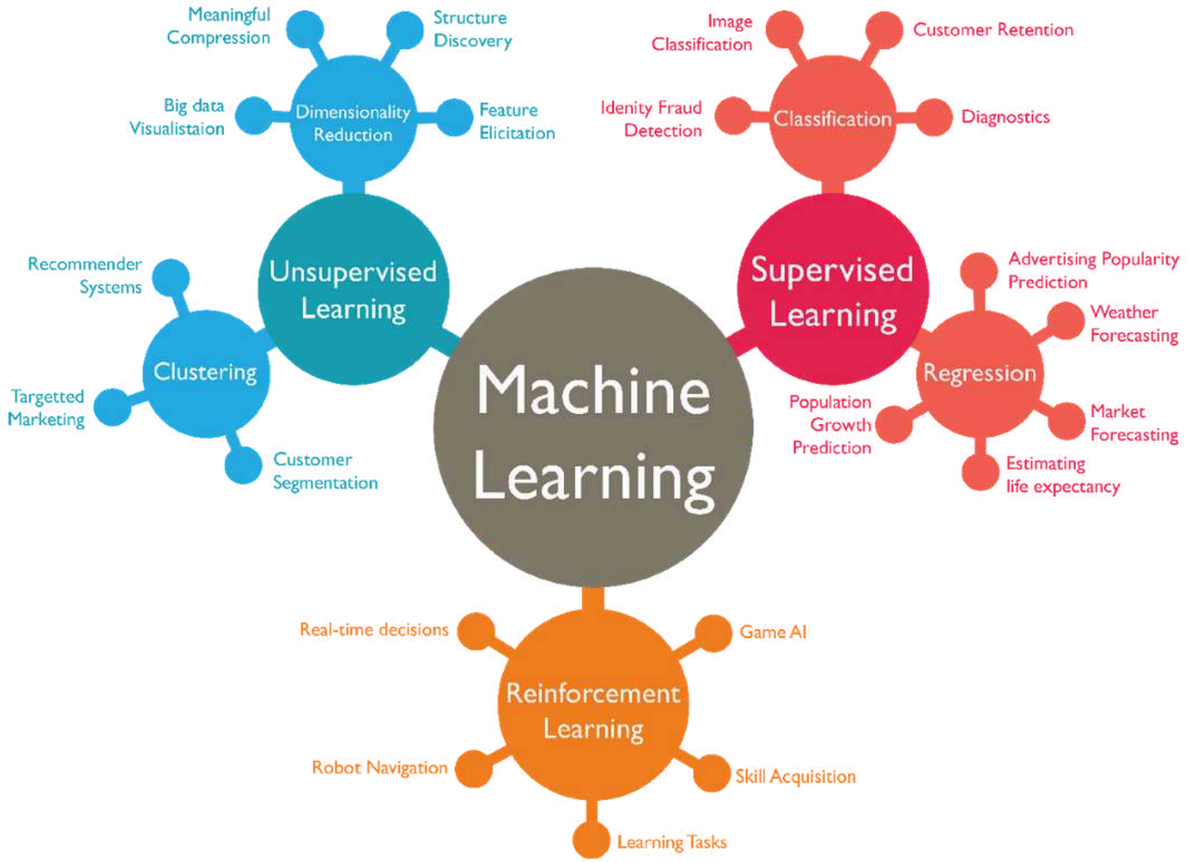




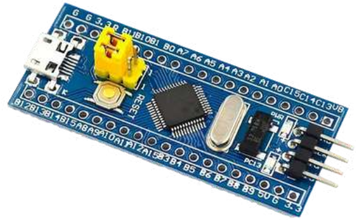
# WHY SHOULD I STUDY TINYML?

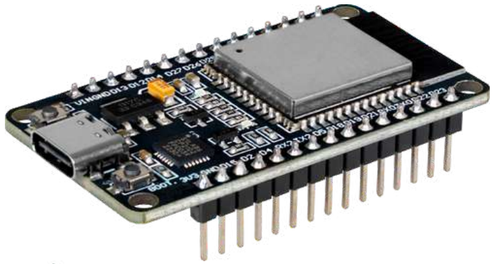
Fastest-growing → Algorithms, hardware, software

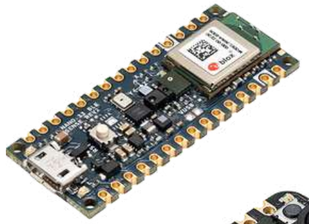


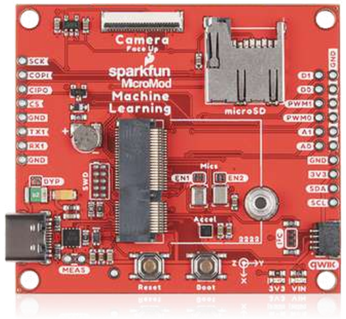
# WHY SHOULD I STUDY TINYML?

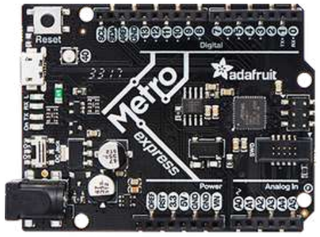
Fastest-growing → Algorithms, hardware, software

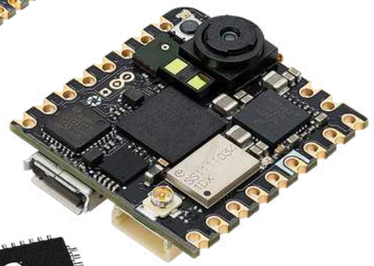













# WHY SHOULD I STUDY TINYML?

Fastest-growing → Algorithms, hardware, software

NANOEDGE AI STUDIO

Micro ML

PyTorch

imagimob

TensorFlow Lite

ELOQUENT ARDUINO

m2cgen

EDGE IMPULSE

Embedded Learning Library

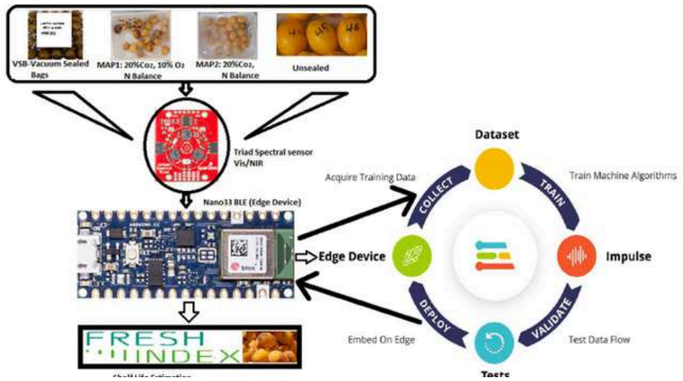
STM32 Cube.AI

MATLAB

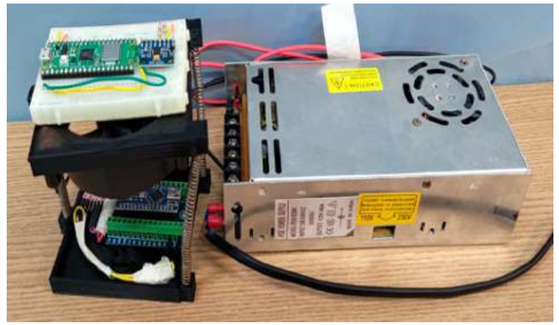
TENSORFLORES

# WHY SHOULD I STUDY TINYML?

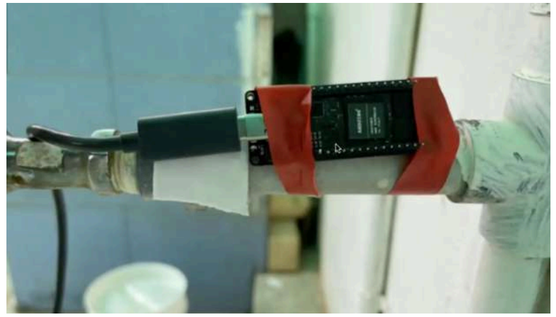
Fastest-growing → Algorithms, hardware, software → On-device sensor analytics




TinyML-Sensor for Shelf Life Estimation of Fresh Date Fruits



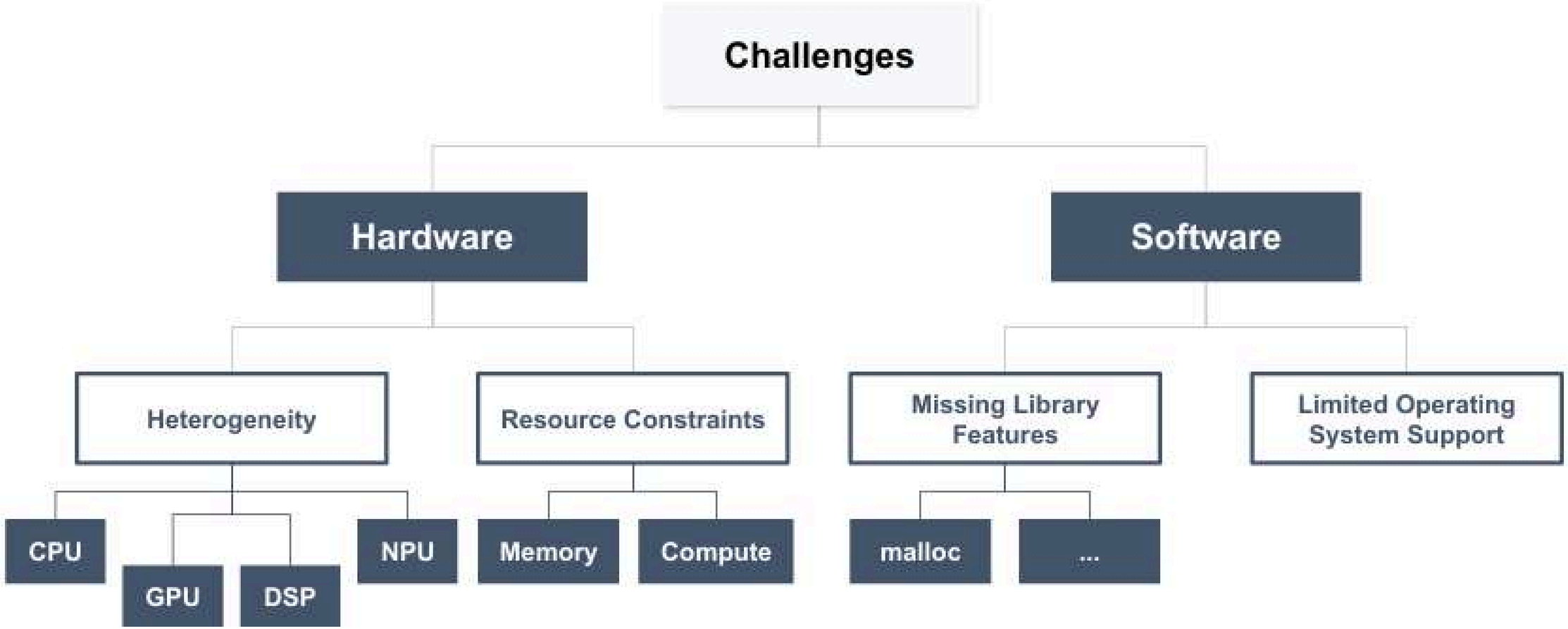
On-Device Tiny Machine Learning for Anomaly Detection Based on the Extreme Values Theory



A TinyML-Based Solution for Acoustic Pipe Leak Detection


A tinyml soft-sensor approach for low-cost detection and monitoring of vehicular emissions

# TINYML CHALLENGES



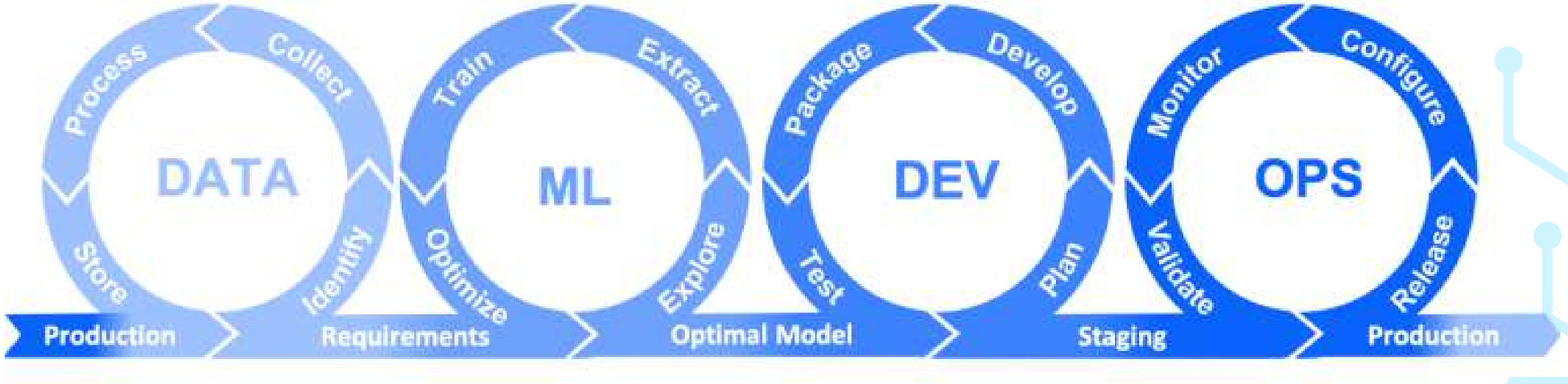
# MLOPS
# DEVOPS
# TINYMLOPS

## SOFTWARE ENGINEERING AND DEVOPS PRACTICES

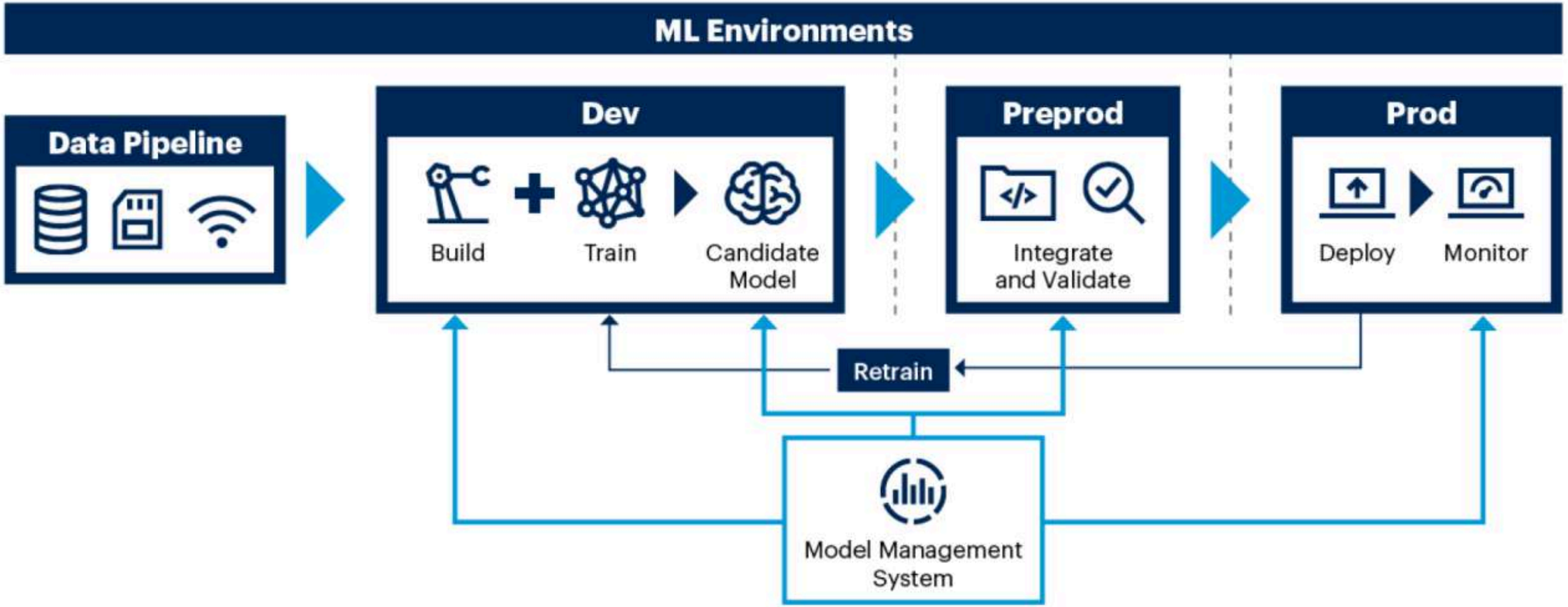
# WHAT IS MLOPS?

*MLOps, or Machine Learning Operations, is defined as "the discipline of applying software engineering principles and DevOps practices to reliably and efficiently manage, scale, and monitor machine learning systems in production."*

(Font: Livro "Practical MLOps" por Noah Gift, Dibya Chakravorty, Kenneth F. Kocher, 2021).

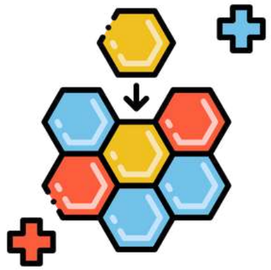


# PIPELINE DE MACHINE LEARNING



# SOFTWARE ENGINEERING PRINCIPLES

Abstraction

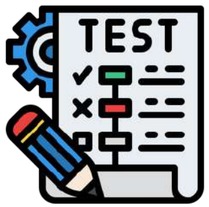


Modularity

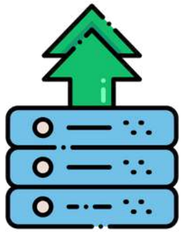
Encapsulation

Maintainability

Reusability



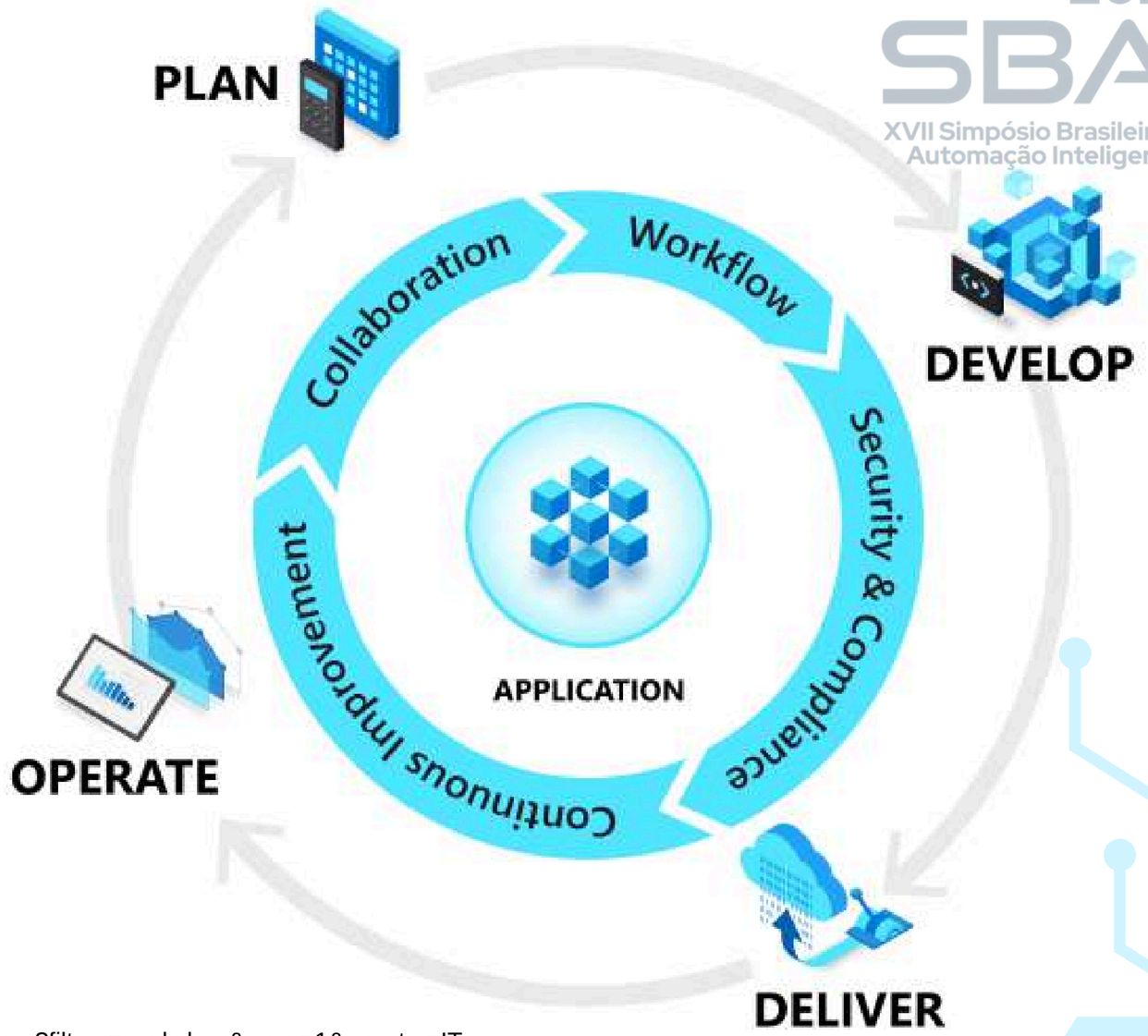

Testability



Scalability

# WHAT IS DEVOPS?

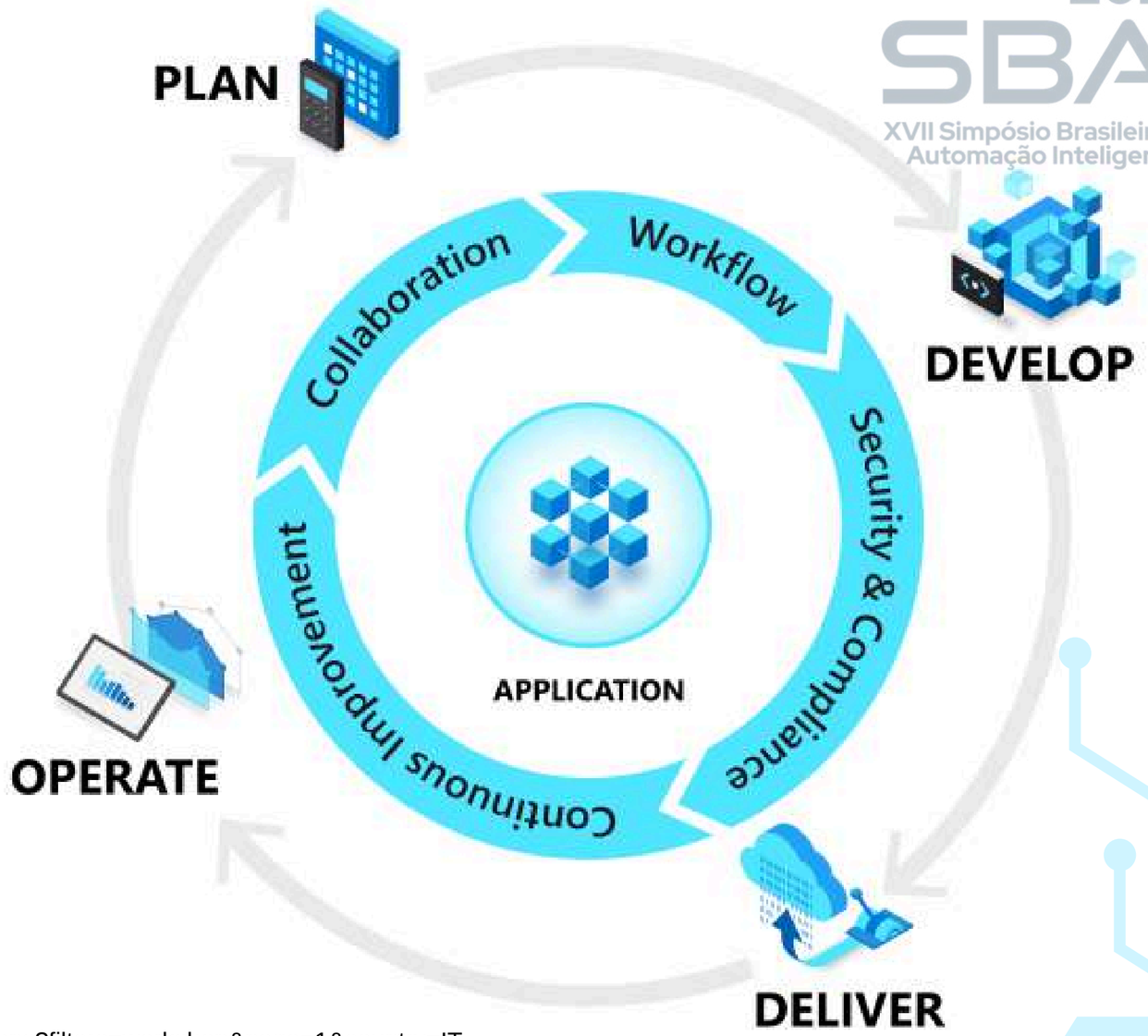
*“DevOps is the union of people, processes and products to enable the continuous delivery of value to your end users.”*




**FONTE:** https://azuremarketplace.microsoft.com/nb-no/marketplace/consulting-services/reply.cluster_devops?filters=workshop&page=1&country=IT

# DEVOPS PRACTICES

- Continuous Integration (CI)
- Continuous Delivery (CD)
- Continuous Deployment
- Infrastructure as Code (IaC)
- Collaboration Culture

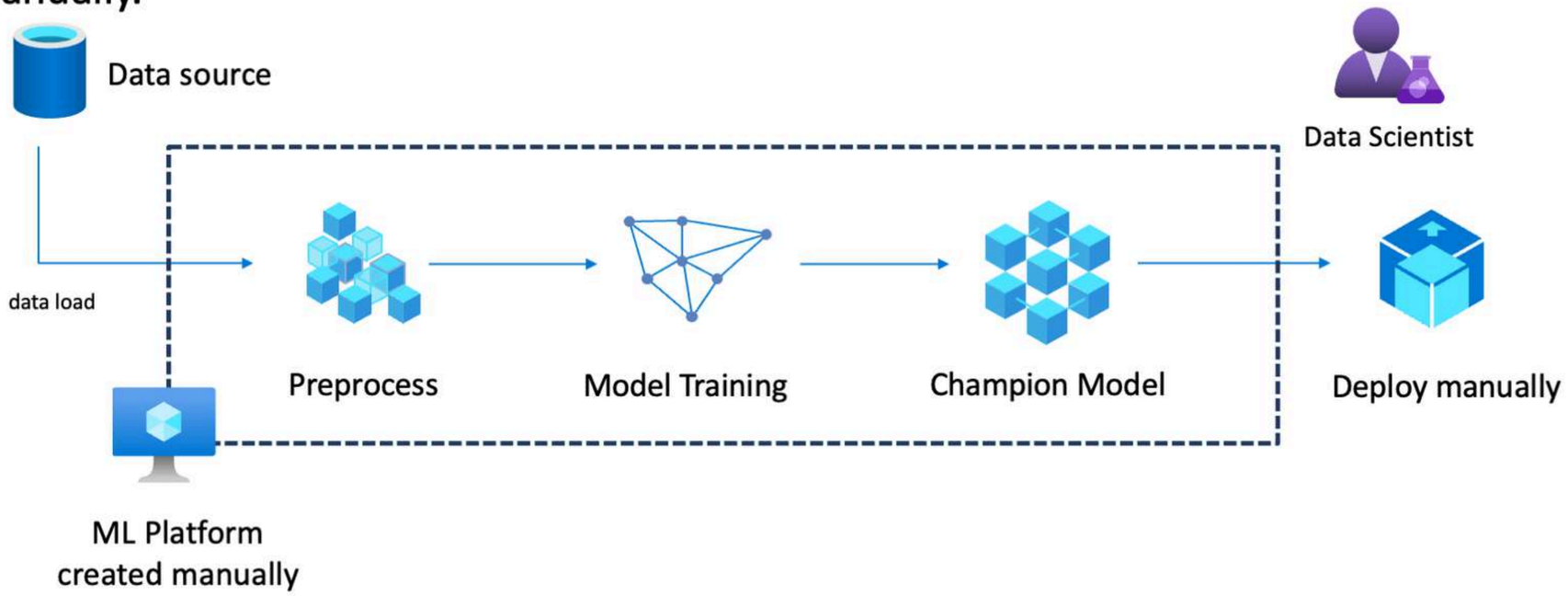



**FONTE:** https://azuremarketplace.microsoft.com/nb-no/marketplace/consulting-services/reply.cluster_devops?filters=workshop&page=1&country=IT

# MLOPS MATURITY LEVELS

## Level 0 – No MLOps

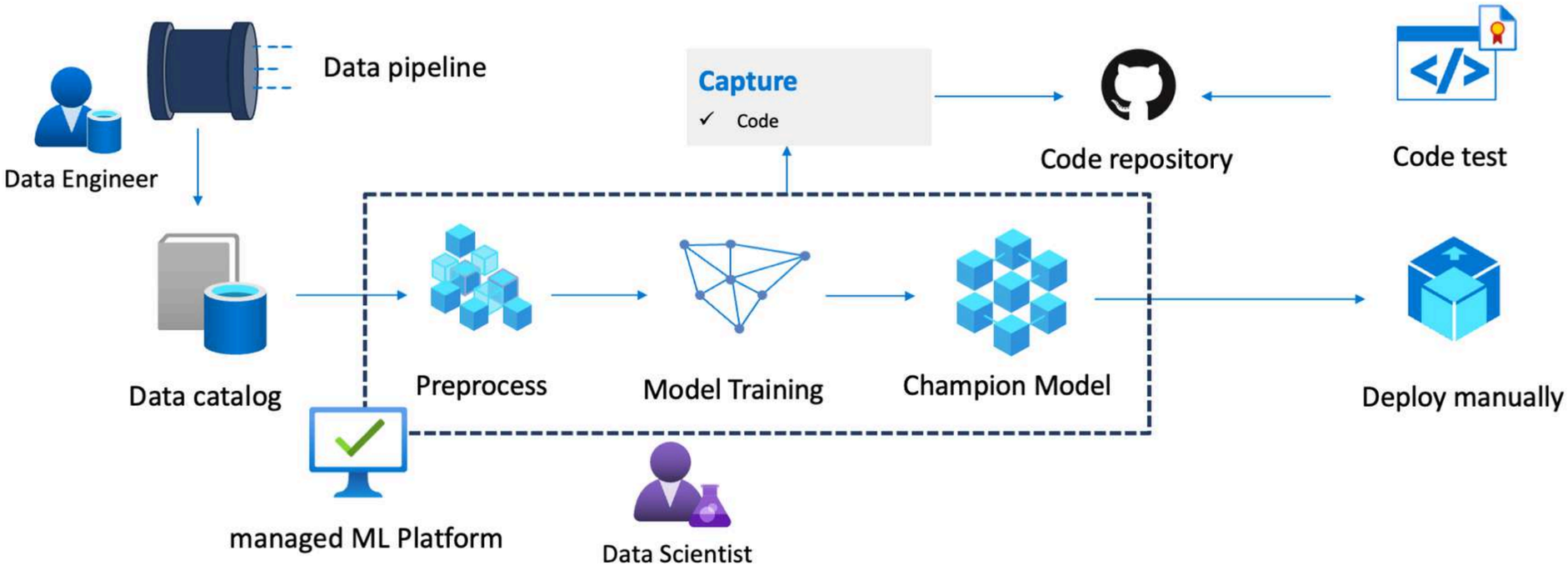
- Find best model interactively and exploratory.
- Create environment, gather and preprocess data, model training, deploy and test manually.



# MLOPS MATURITY LEVELS

## Level 1 – DevOps no MLOps

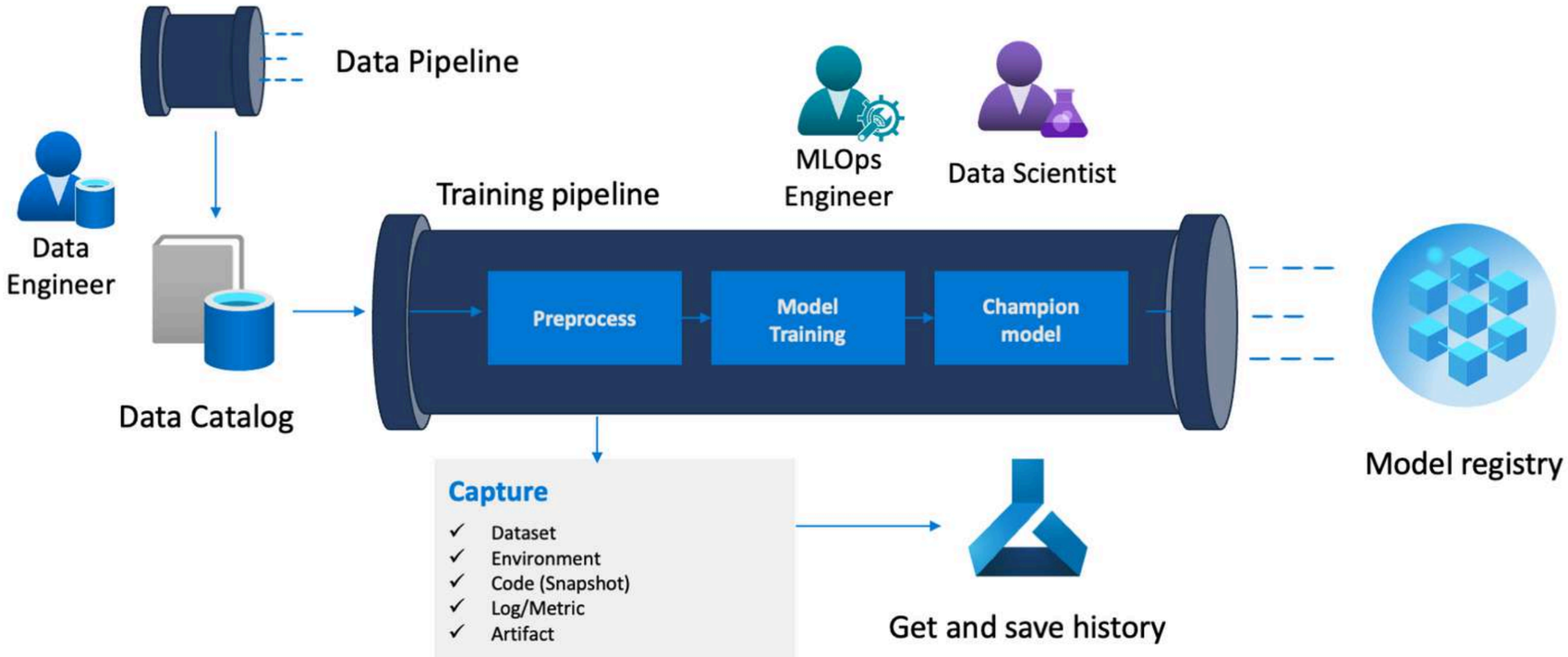
- Create managed ML platform.
- Maintain code test against application and training/inference scripts.



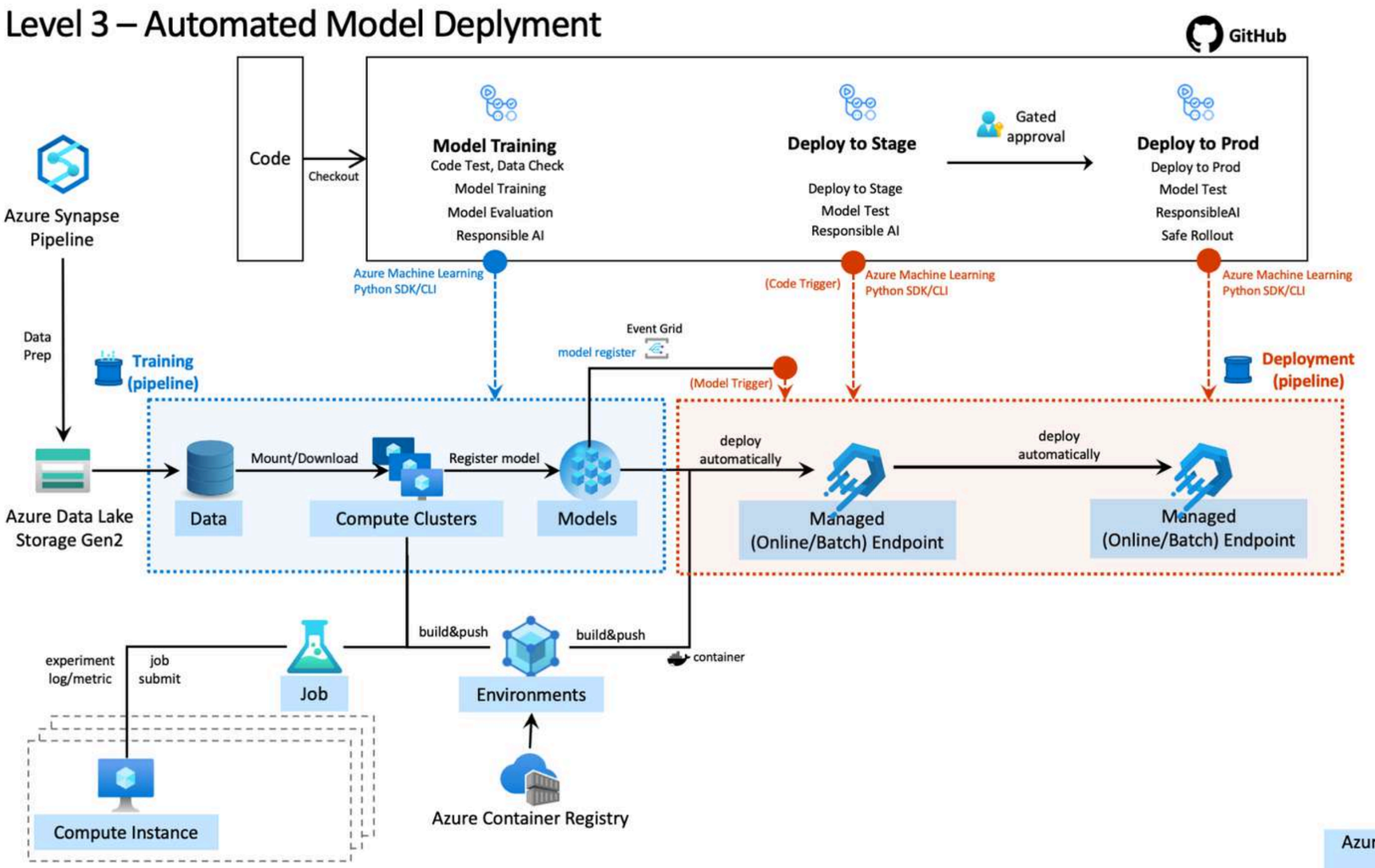
# MLOPS MATURITY LEVELS

## Level 2 – Automated Training

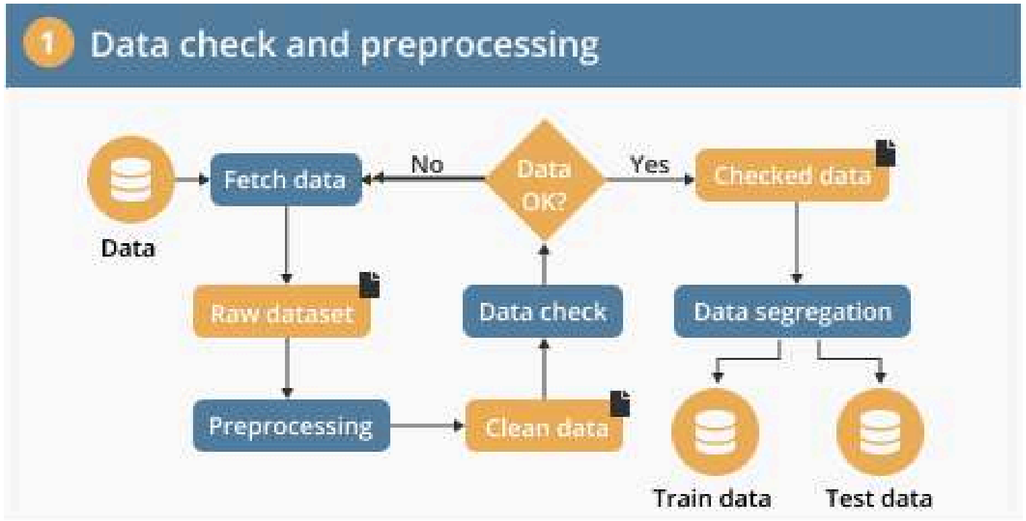
- Make Code, Data, Model tracked, saved and version controlled.
- Automate training process using pipeline.

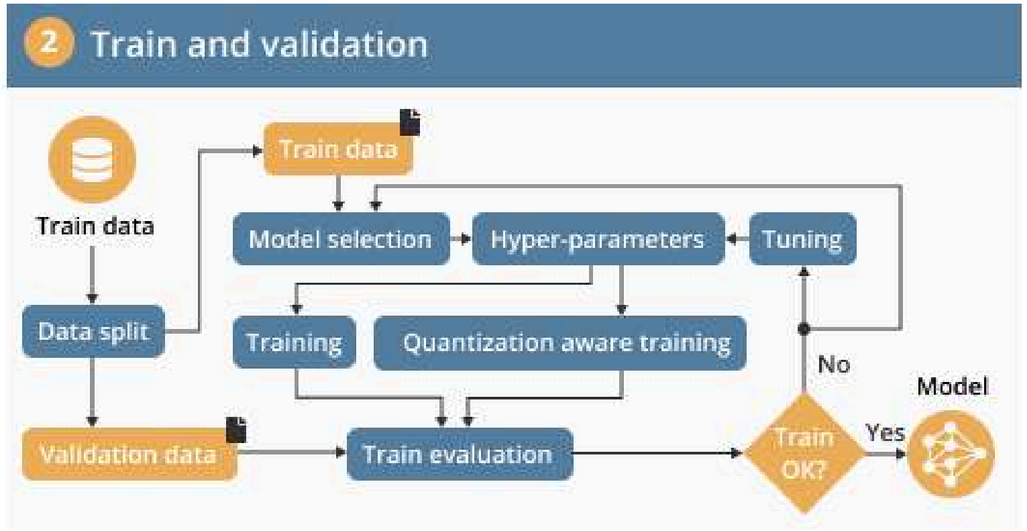


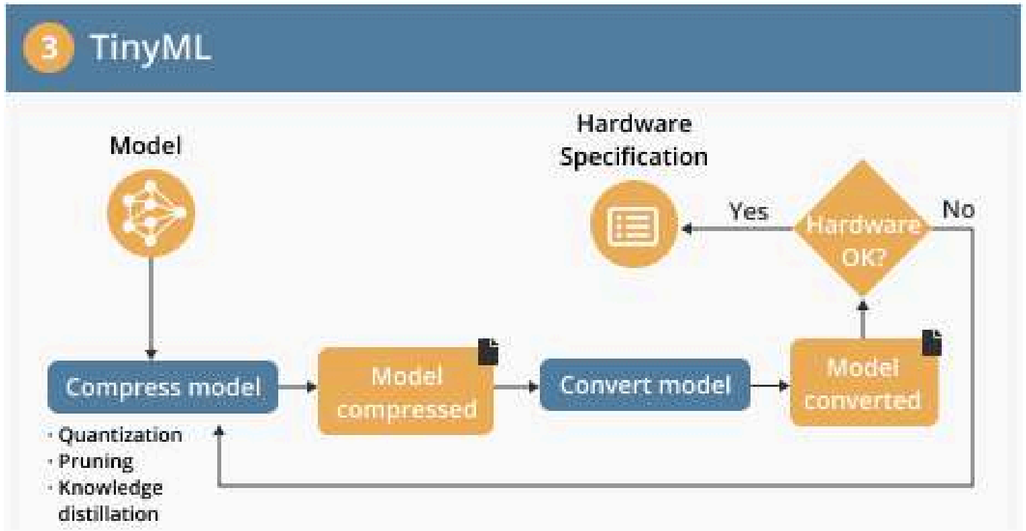
# MLOPS MATURITY LEVELS

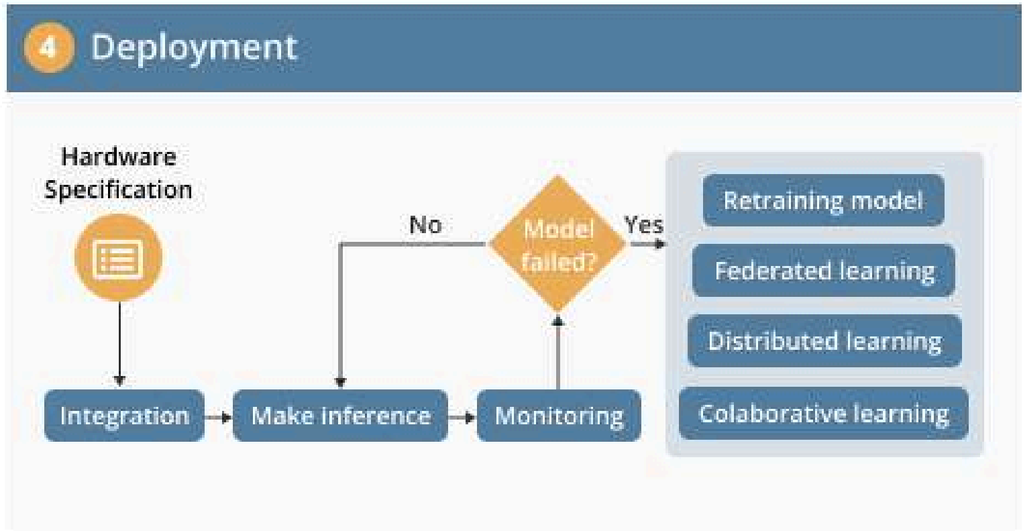


# TINYMLOPS PIPELINE



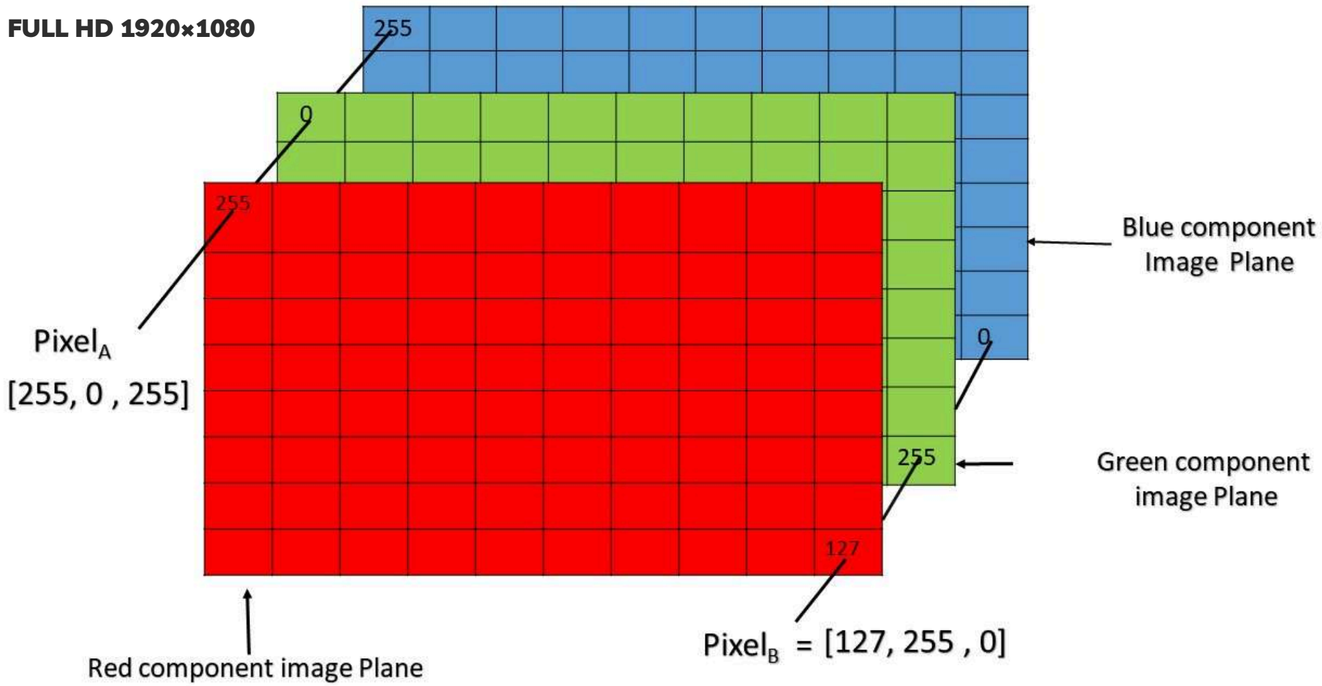



# DATA COLLECTION

## IMAGE, AUDIO AND ANALOG SENSORS

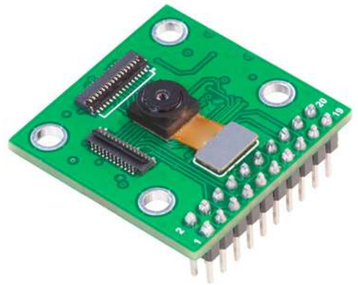
# IMAGE

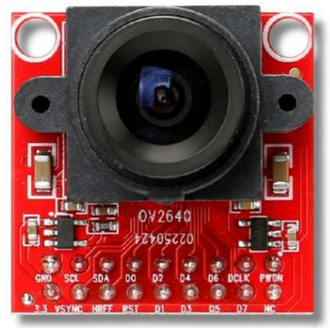


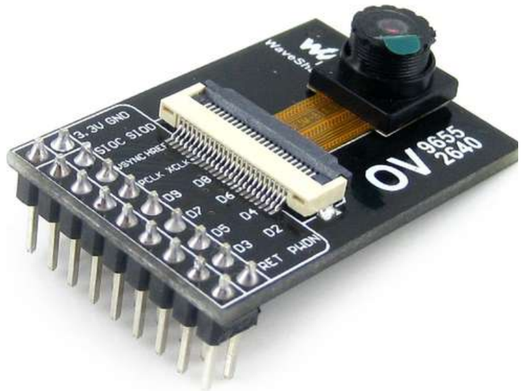

| Tipo de Imagem | Canais | Total de Elementos |
|---|---|---|
| Grayscale | 1 canal | 2.073.600 |
| RGB | 3 canais | 6.220.800 |
| RGBA | 4 canais | 8.294.400 |

# IMAGE

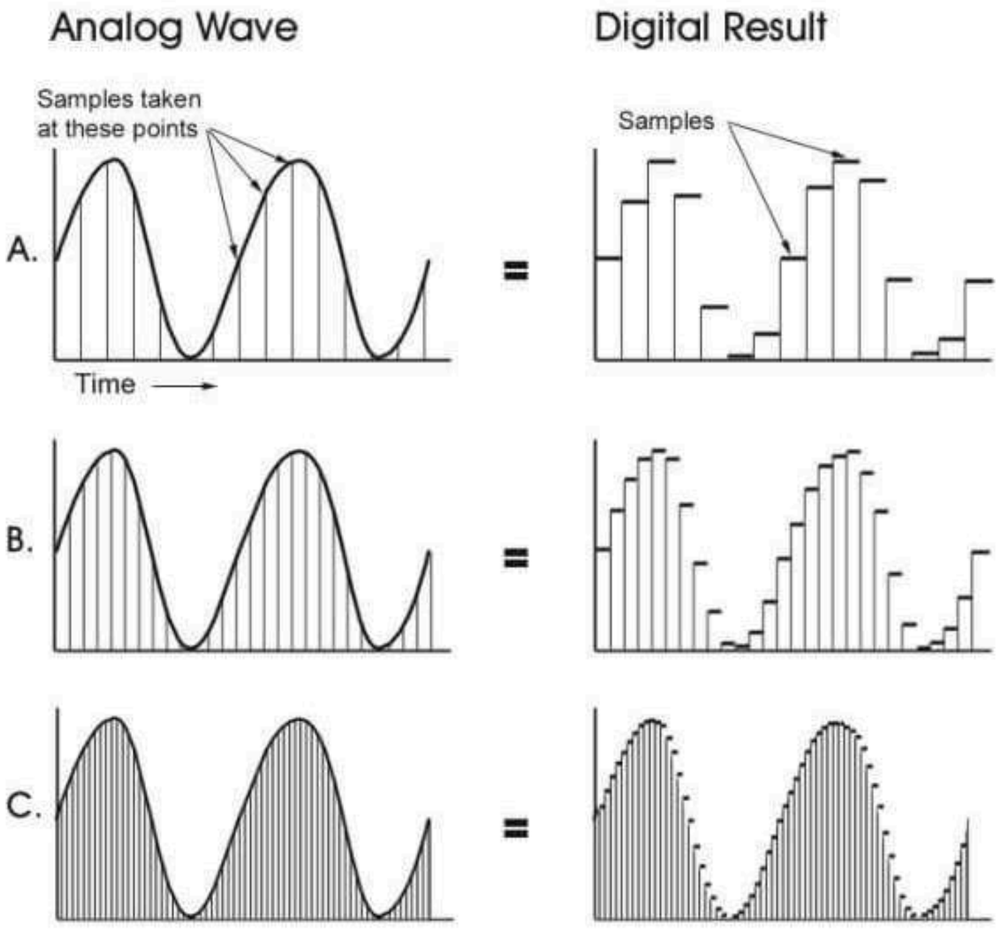
| Camera | Max Resolution | Compatible Microcontrollers | RGB Support |
|---|---|---|---|
| **OV7670** | 640 × 480 (VGA) | AVR (Arduino Uno), STM32, ESP32 | ✅ Yes |
| **OV2640** | 1600 × 1200 (UXGA) | ESP32 (ESP32-CAM, TTGO), STM32 | ✅ Yes* |
| **OV5640** | 2592 × 1944 (5MP) | ESP32-S3, STM32H7, Raspberry Pi (via bridge) | ✅ Yes |
| **OV7725** | 640 × 480 (VGA) | OpenMV, STM32, ESP32 | ✅ Yes |
| **MT9V034** | 752 × 480 (WVGA) | Teensy, OpenMV, STM32, Raspberry Pi Pico | ❌ No (Monochrome) |
| **HM01B0** | 320 × 240 (QVGA) | Arduino Nicla Vision, SparkFun Edge | ❌ No (Monochrome) |
| **HM0360** | 640 × 480 (VGA) | Himax SLAM boards, CMSIS-NN MCUs | ✅ Yes |
| **GC2145** | 1600 × 1200 (UXGA) | ESP32-S3, AI-Thinker boards | ✅ Yes |
| **Arducam SPI** | Up to 5MP (varies) | Any MCU with SPI (ESP32, STM32, RP2040, Arduino, etc.) | ✅ Depends on sensor |



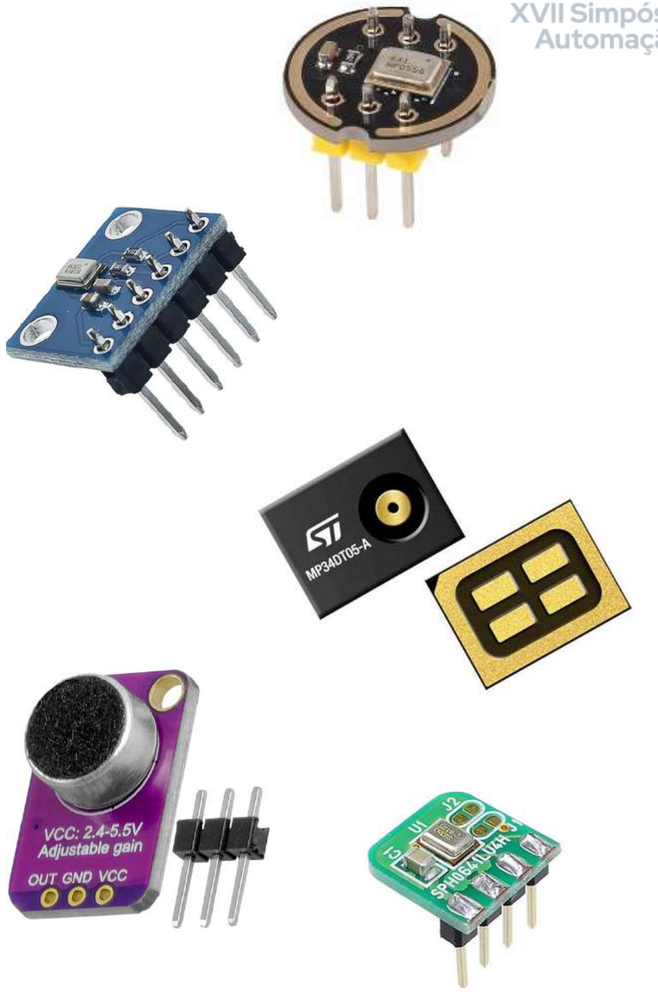
# AUDIO




- ✗ INTERFACE INCOMPATIBILITY
- ✗ RANDOM SAMPLING FREQUENCY
- ✗ EXCESSIVE DATA FOR THE MICROCONTROLLER
- ✗ INCOMPATIBLE OUTPUT TYPE
- ✗ NOISE AND POOR AUDIO QUALITY
- ✗ HIGH POWER CONSUMPTION
- ✗ MICROPHONE DIRECTIONALITY

# AUDIO

| Microphone | Sampling Rate (Typical) | Compatible Microcontrollers | Type |
|---|---|---|---|
| **INMP441** | Up to 48 kHz | ESP32, STM32, RP2040 (I2S support required) | **Digital (I2S)** |
| **SPH0645LM4H** | 16 / 44.1 / 48 kHz | ESP32, STM32, RP2040 (I2S), Raspberry Pi | **Digital (I2S)** |
| **MP34DT05** | Up to 48 kHz | STM32, Raspberry Pi, Arduino Portenta H7 | **Digital (I2S or PDM)** |
| **MP45DT02** | Up to 48 kHz | STM32, ARM Cortex-M MCUs | **Digital (PDM)** |
| **ICS43434** | 16 / 32 / 48 kHz | ESP32, STM32, Arduino Portenta, RP2040 | **Digital (I2S)** |
| **MAX4466** | 8 – 44.1 kHz (depends on ADC) | Any MCU with ADC (Arduino Uno, STM32, ESP32) | **Analog** |
| **MAX9814** | 8 – 44.1 kHz (depends on ADC) | Any MCU with ADC (Arduino, ESP32, RP2040) | **Analog + AGC** |
| **Knowles SiSonic (e.g., SiSonic SPH0641LU4H)** | Up to 48 kHz | STM32, ESP32, advanced ARM MCUs | **Digital (PDM)** |



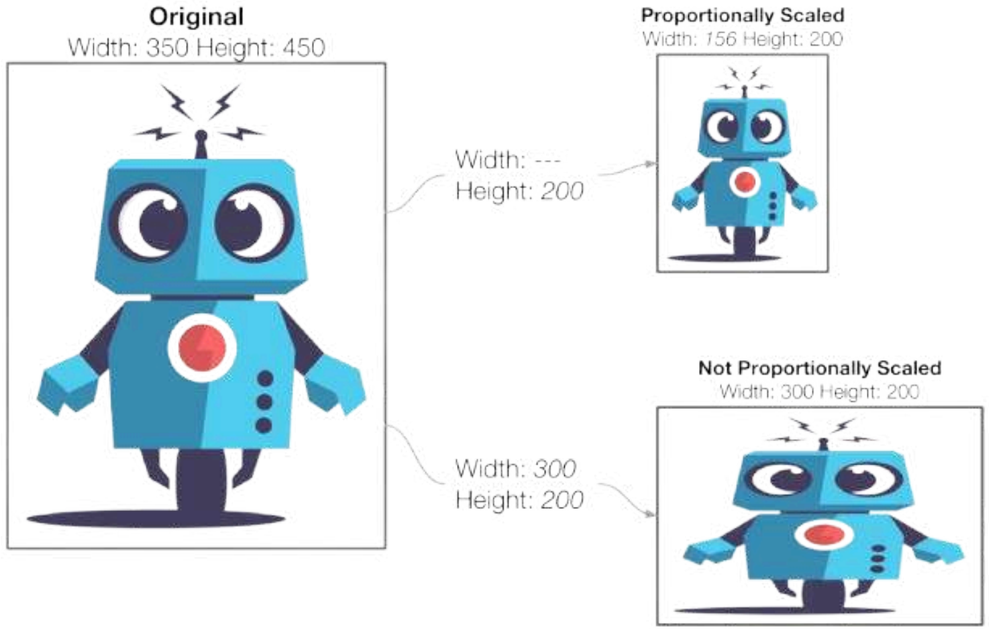
# ANALOG SENSORS

| Problem | Common Cause | Consequence |
|---|---|---|
| ❌ **Incompatible voltage levels** | Sensor outputs **5V**, but the MCU's ADC accepts a maximum of **3.3V** | Risk of **damaging the microcontroller** or **inaccurate readings** |
| ❌ **Narrow output range** | Sensor outputs between **0.5–2.5V**, but the ADC expects **0–3.3V or 0–5V** | **Low effective resolution**, poor precision |
| ❌ **Noisy signal** | No **filtering**, long wires, poor grounding | **Fluctuating readings**, instability in ML or control systems |
| ❌ **High output impedance** | Sensor without buffer or weak analog stage | **Incorrect ADC readings** due to signal distortion |
| ❌ **Slow sensor response** | Sensor has inherent **lag or delay** | Missed fast events, **poor real-time response** |
| ❌ **Lack of calibration** | Sensor requires **offset/gain tuning** and wasn't calibrated | **Inconsistent or inaccurate data** |
| ❌ **Sensor interference** | Multiple analog sensors on shared power or ground lines | **Cross-talk** or voltage instability |
| ❌ **Low ADC resolution** | Using an **8- or 10-bit ADC** with a wide-range sensor | Loss of **fine signal detail**, reduced sensitivity |

# PREPROCESSING DATA

## IMAGE, AUDIO AND ANALOGICT SENSORS

# IMAGE




RESIZE IMAGE

## GRAYSCALE IMAGE

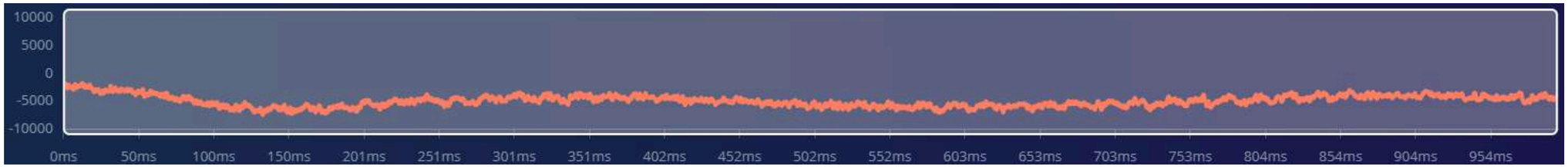
| RED | | |
|---|---|---|
| 56 | 53 | 54 |
| 63 | 62 | 64 |
| 52 | 57 | 61 |

| GREEN | | |
|---|---|---|
| 61 | 58 | 57 |
| 69 | 68 | 67 |
| 59 | 64 | 68 |

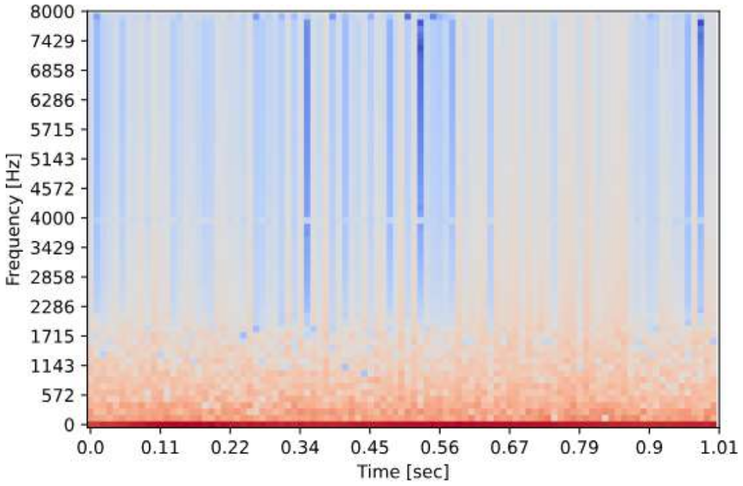
| BLUE | | |
|---|---|---|
| 54 | 51 | 50 |
| 59 | 58 | 58 |
| 43 | 48 | 52 |

| GRAY | | |
|---|---|---|
| 59 | 56 | 55 |
| 66 | 65 | 65 |
| 55 | 60 | 64 |

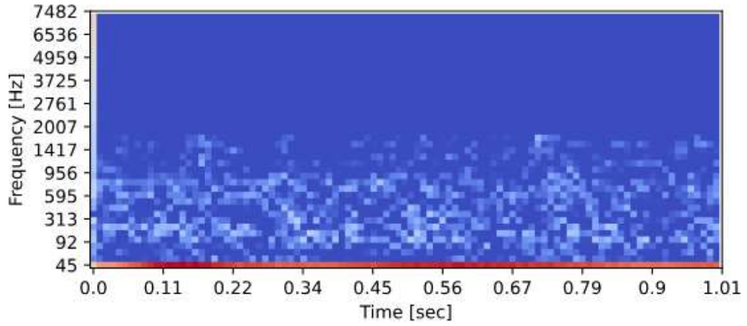
# AUDIO

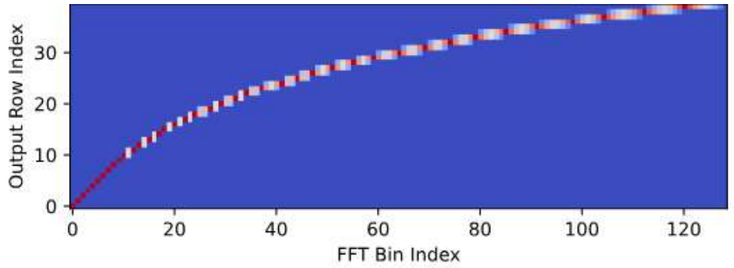



## SPECTROGRAM



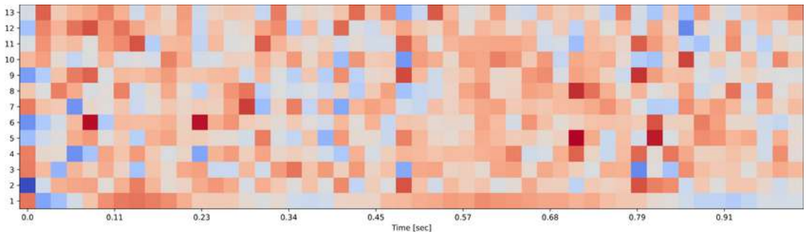

## MEL-FREQUENCY CEPSTRUM



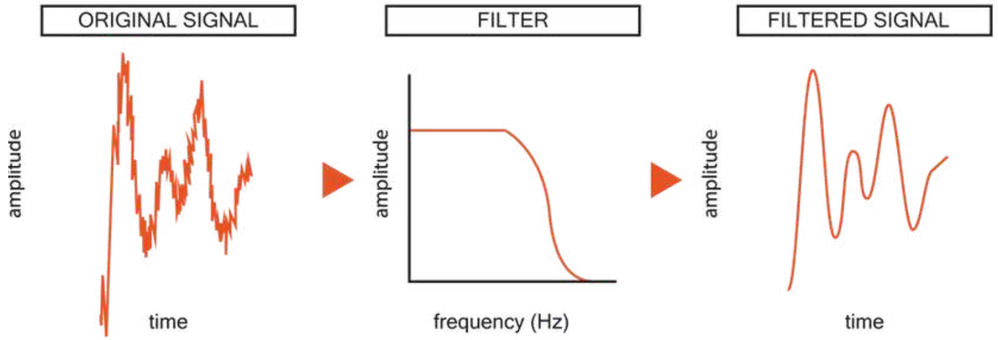


## MEL FREQUENCY CEPSTRUM COEFFICIENT

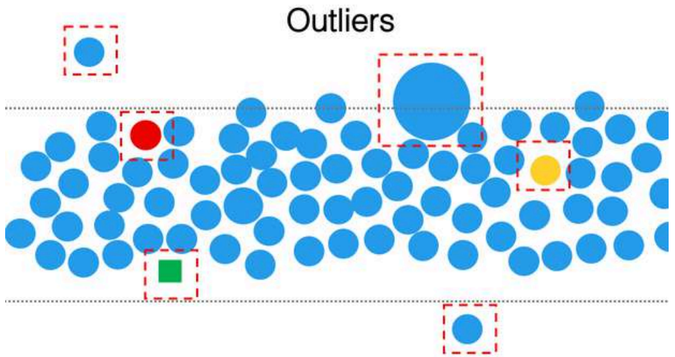


# ANALOG SENSOR

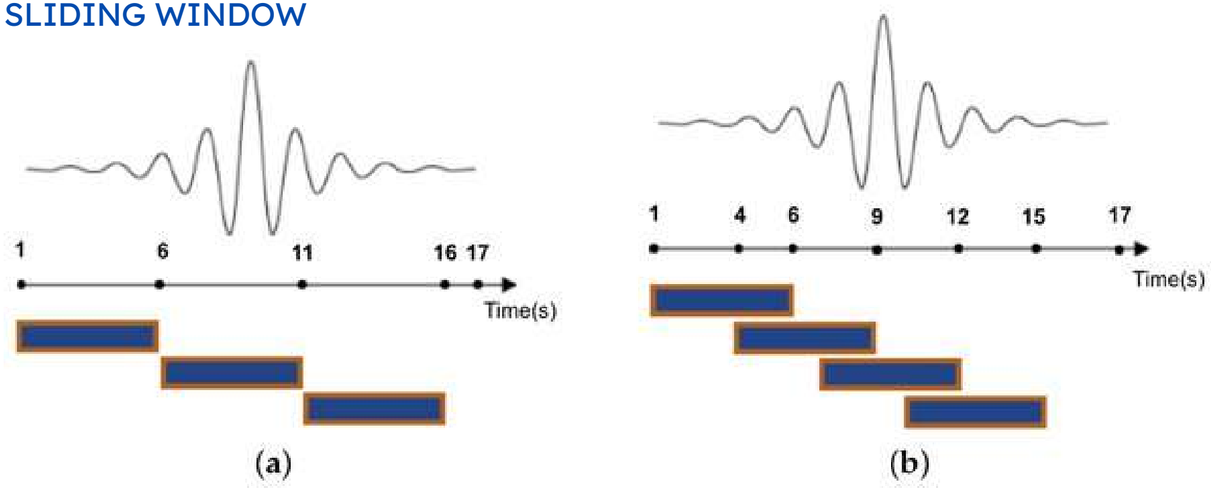
## FILTER




## OUTLIER DETECTION

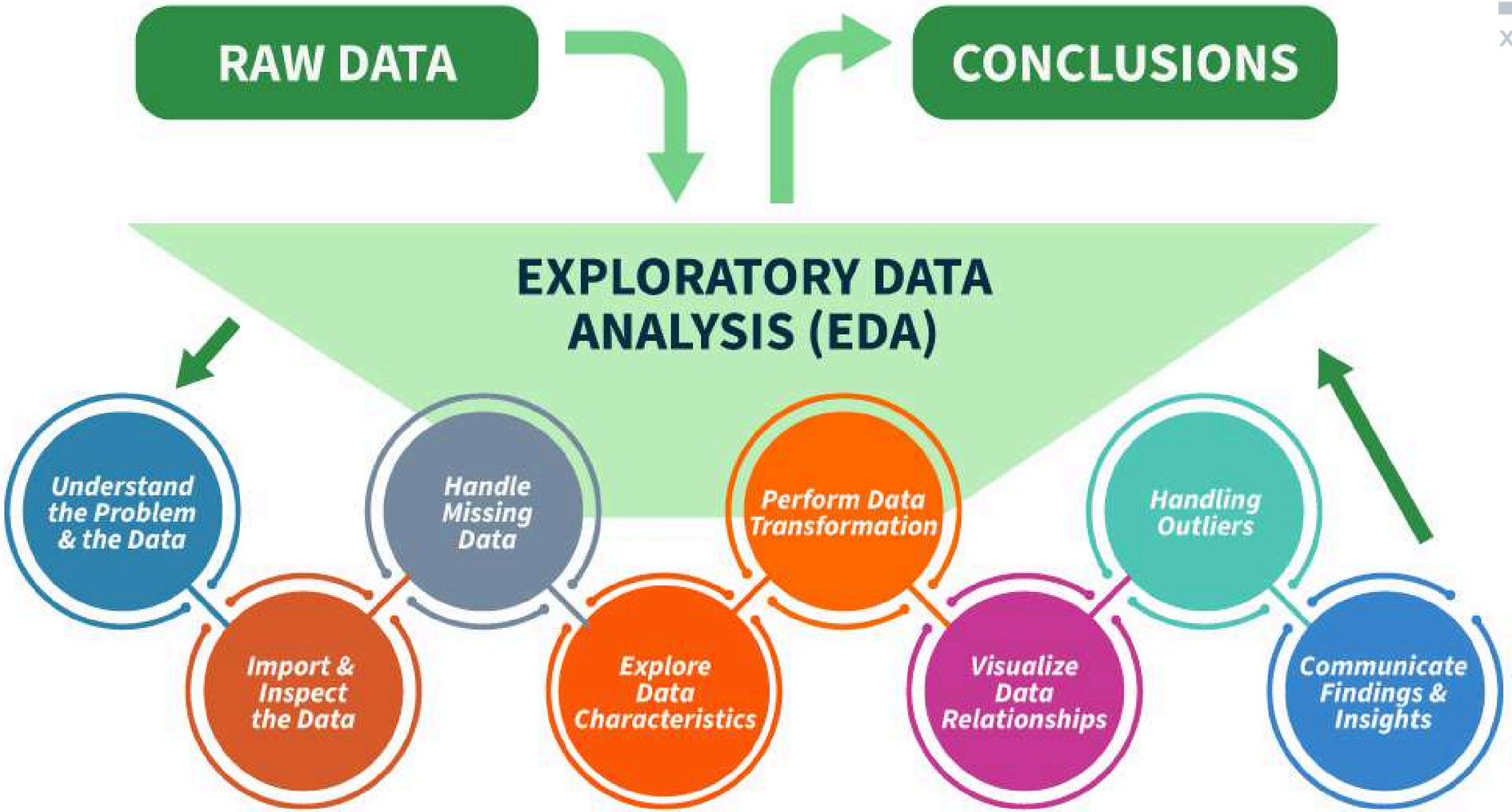



## SLIDING WINDOW

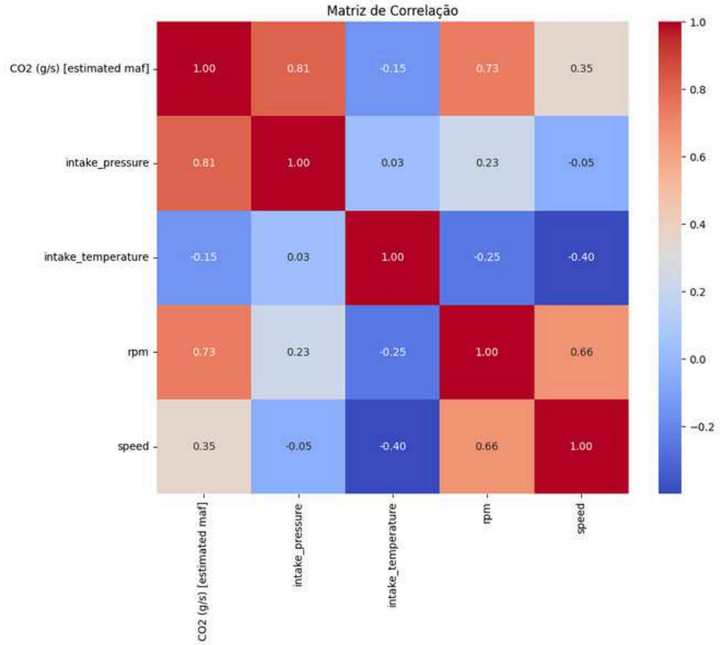


# EXPLORATORY DATA ANALYSIS

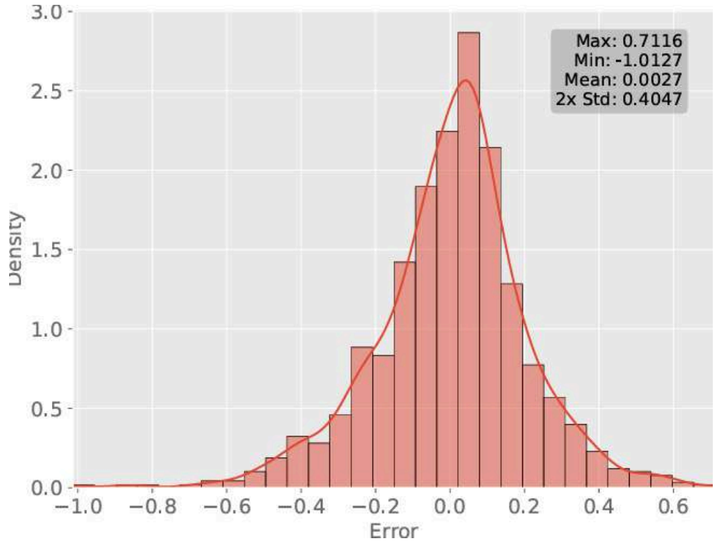
## IMAGE, AUDIO AND ANALOGICT SENSORS


RAW DATA
CONCLUSIONS
EXPLORATORY DATA ANALYSIS (EDA)
Understand the Problem & the Data
Import & Inspect the Data
Handle Missing Data
Explore Data Characteristics
Perform Data Transformation
Visualize Data Relationships
Handling Outliers
Communicate Findings & Insights

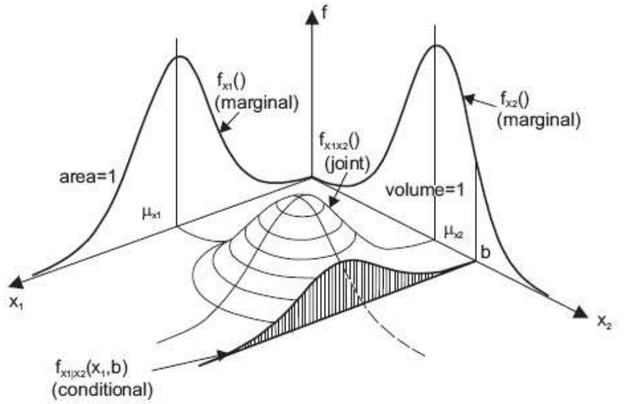
## CORRELATION (PEARSON, KENDALL OR SPEARMAN)




## HISTOGRAM




## JOINT AND MARGINAL PROBABILITY DENSITY FUNCTION

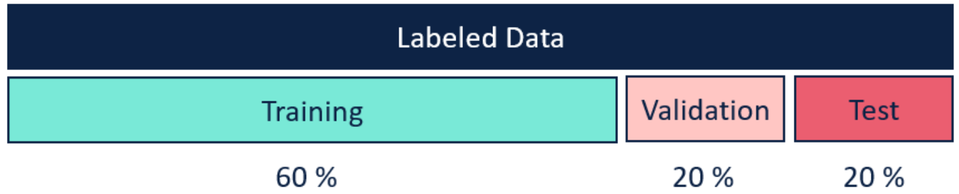


# DATA SEGREGATION

## IMAGE, AUDIO AND ANALOGICT SENSORS

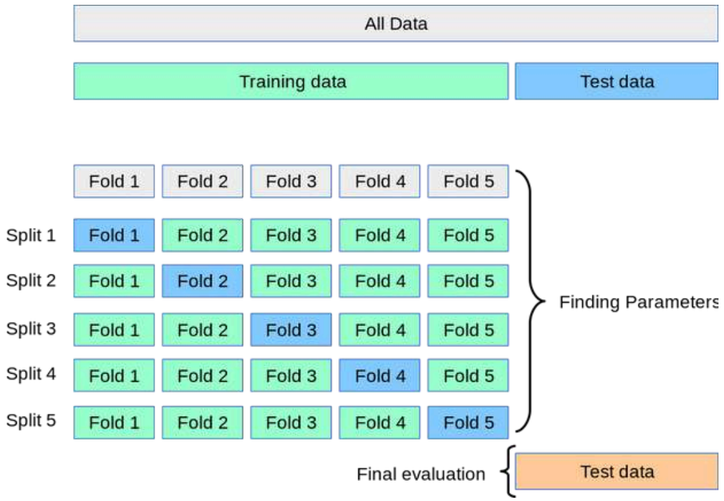
# DATA SEGREGATION

## HOLDOUT METHOD

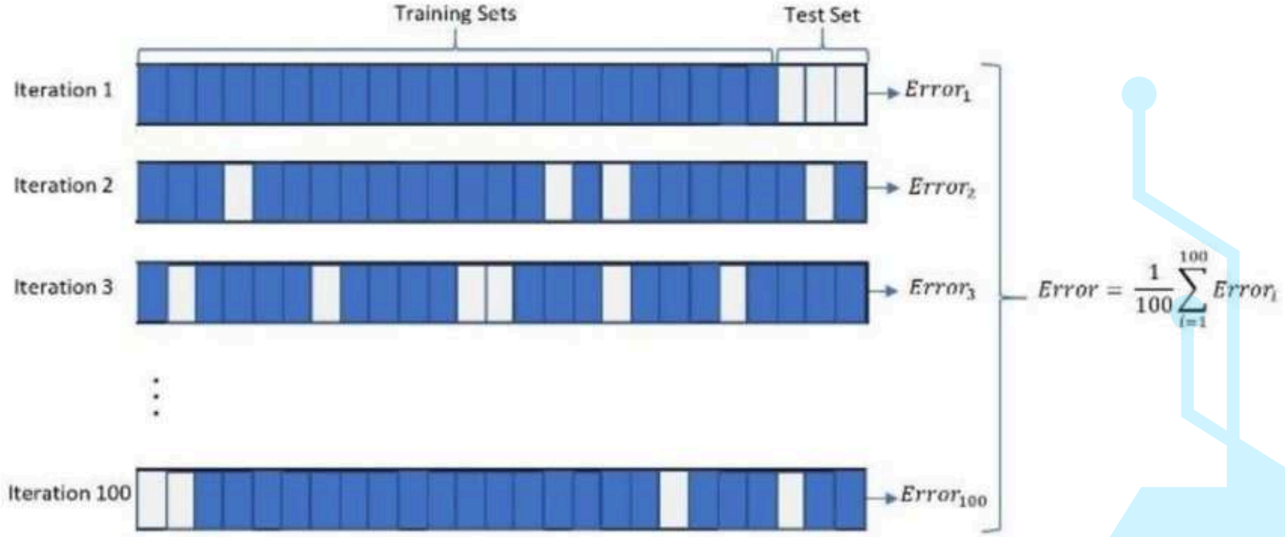



## STRATIFIED SAMPLING


## K-FOLD CROSS-VALIDATION



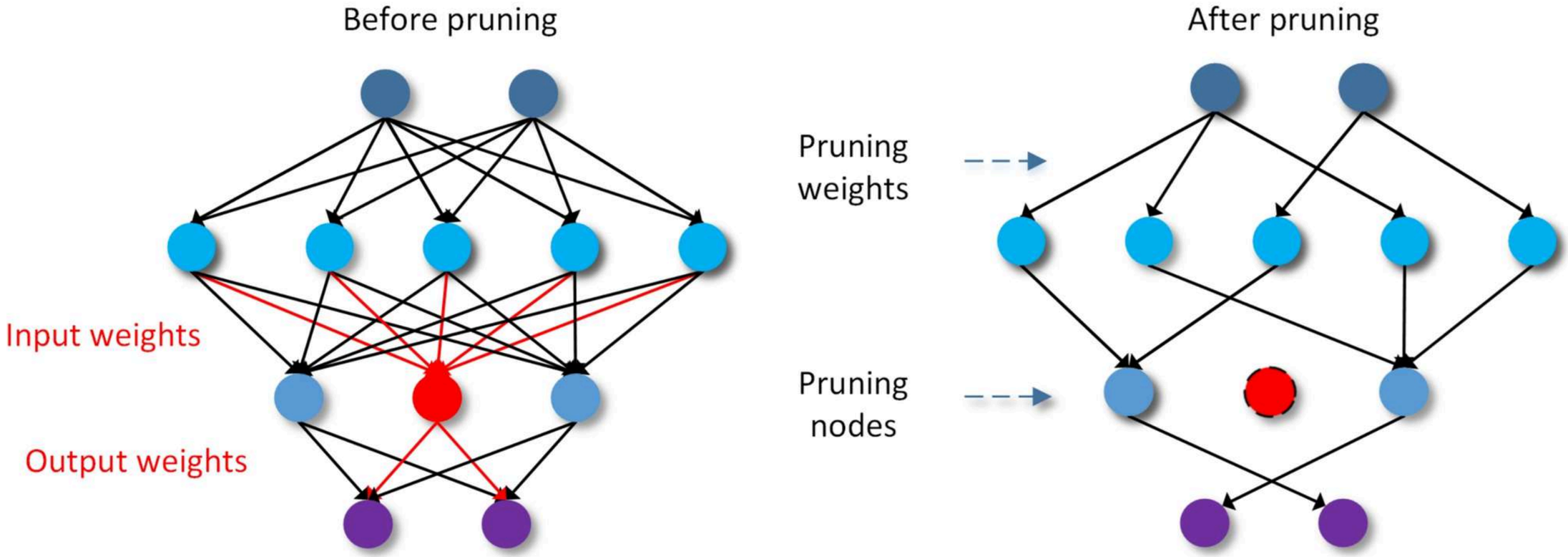

## MONTE CARLO CROSS-VALIDATION



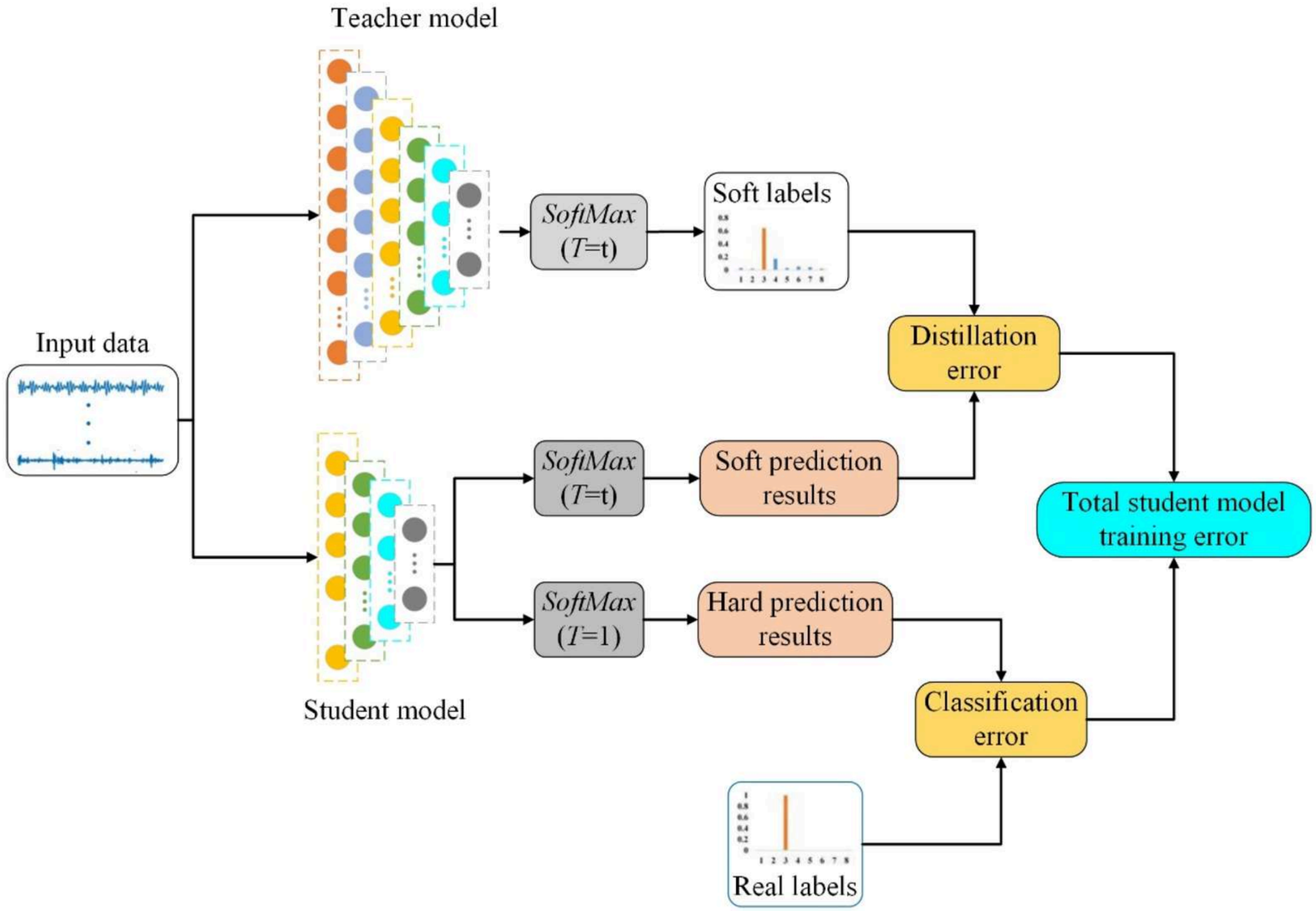
# COMPRESSION METHODS

## PRUNING, KNOWLEDGE DISTILLATION AND QUANTIZATION

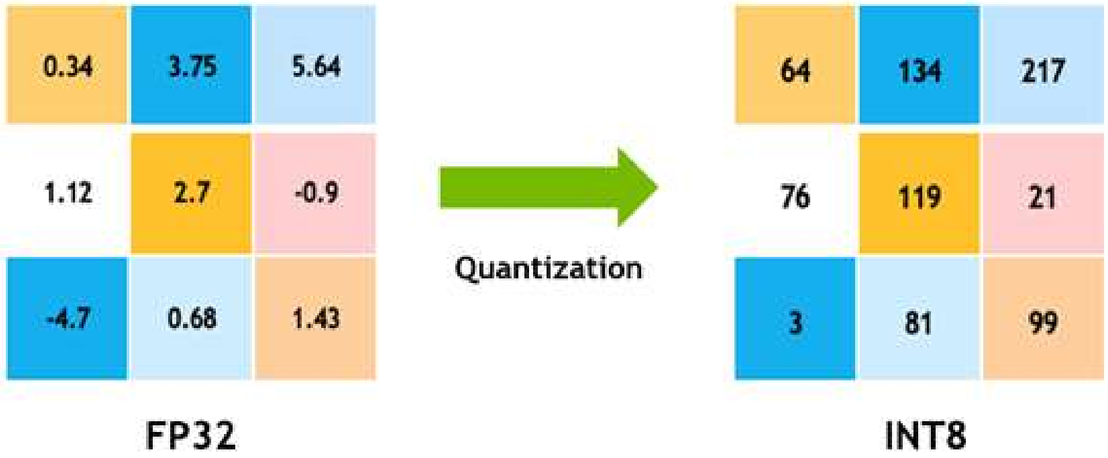
# PRUNING

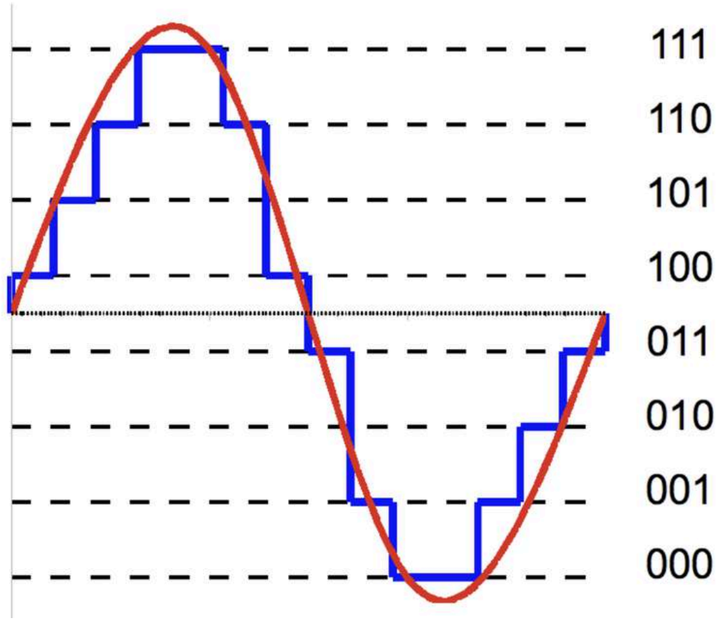

Before pruning
After pruning
Input weights
Output weights
Pruning weights
Pruning nodes

# KNOWLEDGE DISTILLATION



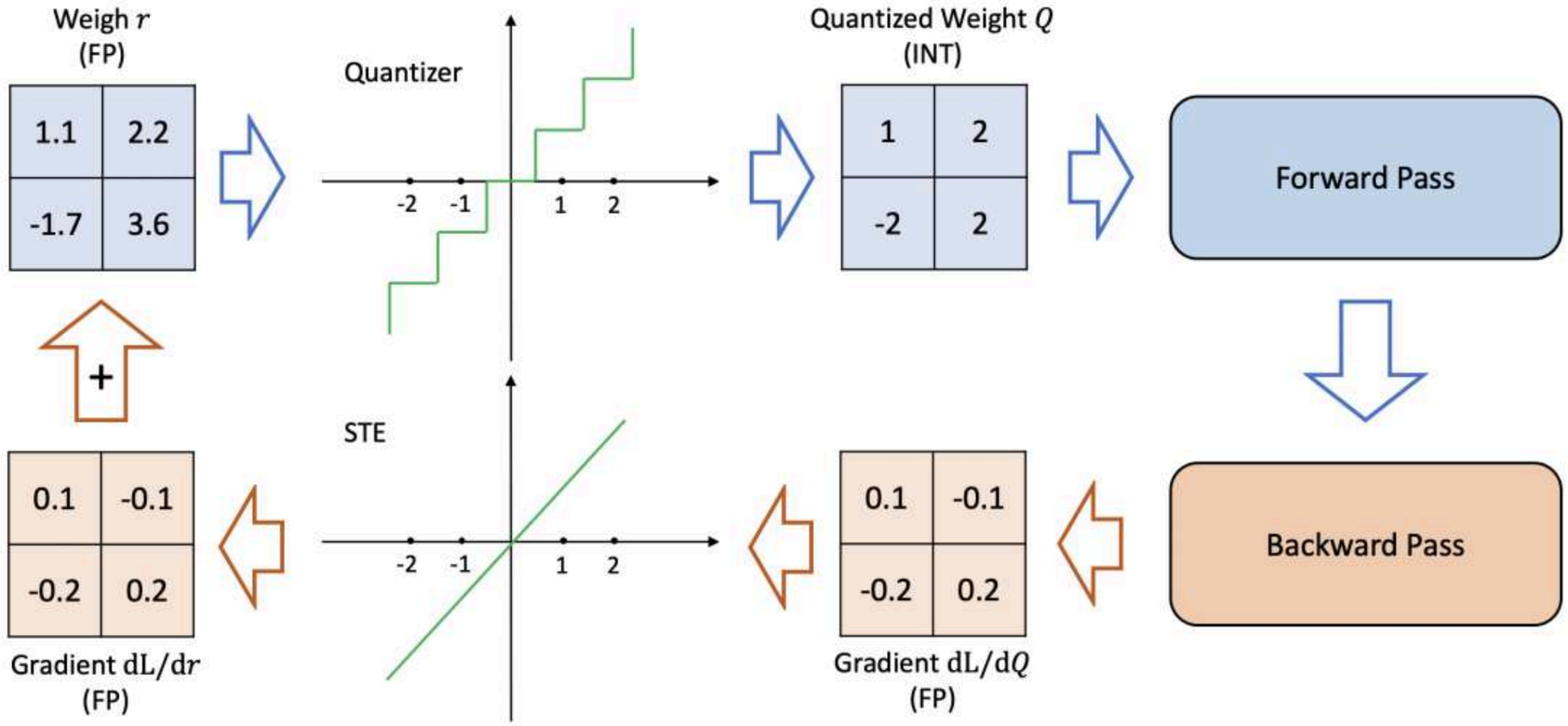
# POST-TRAINING QUANTIZATION



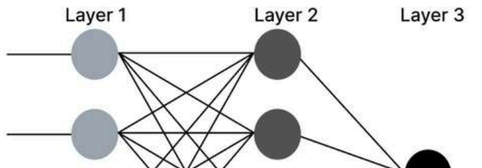



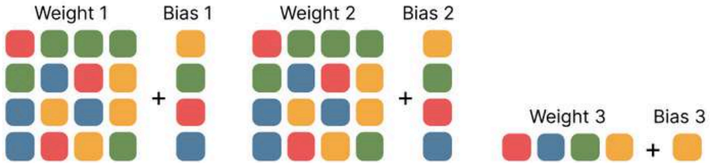

| Técnica | Benefícios | hardware |
|---|---|---|
| Quantização de faixa dinâmica | 4x menor, 2x-3x de aceleração | CPU |
| Quantização inteira completa | 4x menor, 3x+ aceleração | CPU, Edge TPU, Microcontroladores |
| Float16 quantização | 2x menor, aceleração de GPU | CPU, GPU |

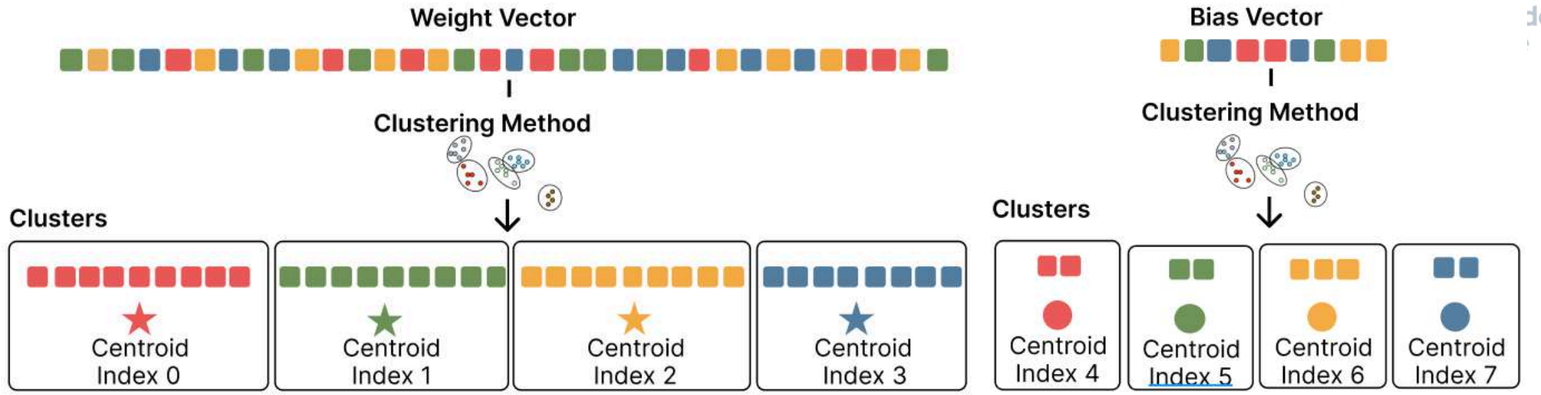
# QUANTIZATION AWARE TRAINING



# VECTOR QUANTIZATION




**Weight and bias With Codebook Indexes (uint8)**

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | + | 4 | 0 | 1 | 1 | 1 | + | 6 | 0 | 3 | 1 | 2 | + | 6 |
| 1 | 3 | 0 | 2 | | 5 | 1 | 3 | 0 | 2 | | 5 | | | | | | |
| 3 | 2 | 3 | 2 | | 7 | 3 | 2 | 3 | 2 | | 4 | | | | | | |
| 3 | 0 | 2 | 1 | | 7 | 3 | 0 | 2 | 1 | | 7 | | | | | | |

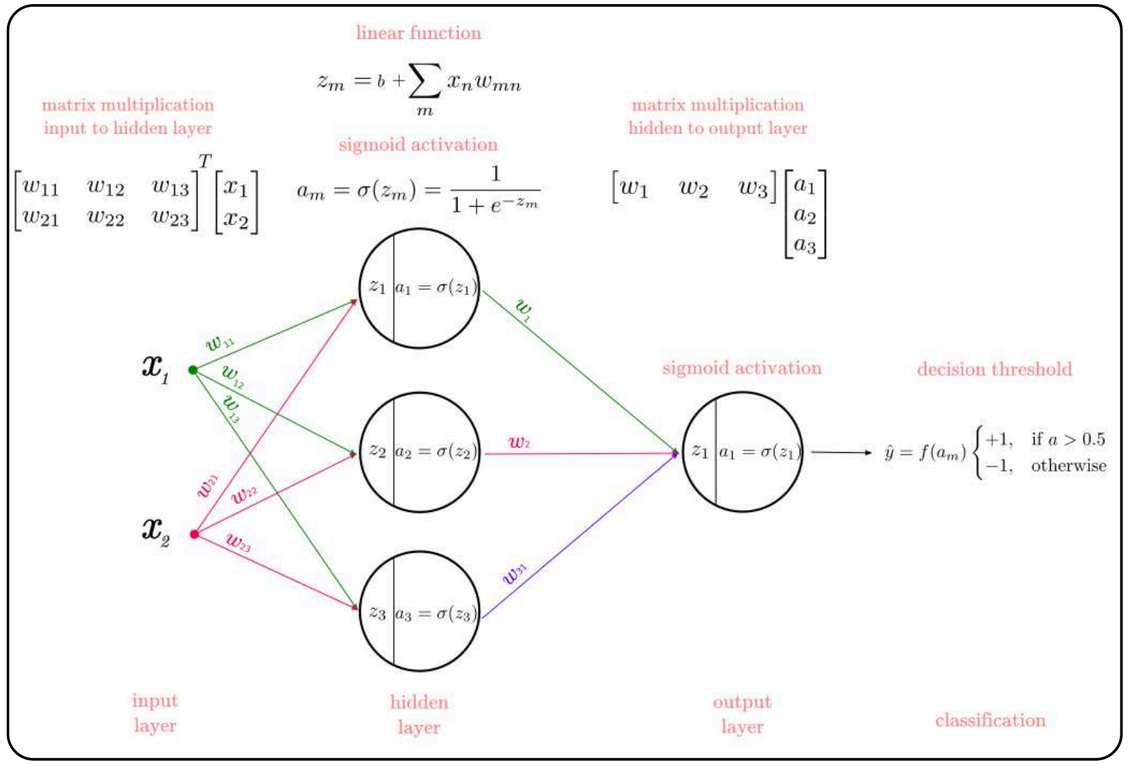
**Codebook (Float32)**

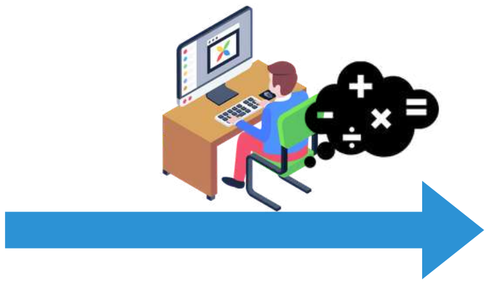
| Centroid | Index | Centroid | Index |
|---|---|---|---|
| ★ | 0 | ● | 4 |
| ★ | 1 | ● | 5 |
| ★ | 2 | ● | 6 |
| ★ | 3 | ● | 7 |

# CONVERSION METHODS

## MANUAL, CODE GENERATOR AND INTERPRETERS

# MANUAL CODING



Untitled-1

```
float predict(float *x) {
float y_pred = 0;
static const float w1[3][16] = {...};

static const float b1[16] = {...};

static const float w2[16][8] = {...};

static const float b2[8] = {...};

static const float w3[8][1] = {...};

static const float b3[1] = {...};

    // Input Layer
    float z1[16];
    for (int i = 0; i < 16; i++)
    {
        z1[i] = b1[i];
        for (int j = 0; j < 3; j++)
        {
            z1[i] += x[j] * w1[j][i];
        }
        z1[i] = relu(z1[i]);
    }

    // Hidden Layer 2
    float z2[8];
    for (int i = 0; i < 8; i++)
    {
        z2[i] = b2[i];
        for (int j = 0; j < 16; j++)
        {
            z2[i] += z1[j] * w2[j][i];
        }
        z2[i] = relu(z2[i]);
    }

    // Output Layer
    float z3 = b3[0];
    for (int i = 0; i < 8; i++)
    {
        z3 += z2[i] * w3[i][0];
        z3 = linear(z3);
    }

y_pred = z3;
return y_pred;
}
protected:
float relu(float x)
{
    return x > 0 ? x : 0;
};

float linear(float x)
{
    return x;
};
```

# INTERPRETERS

TensorFlow

```python
import tensorflow as tf
from tensorflow.keras.layers import Dense

model = tf.keras.Sequential()
model.add(Dense(16, activation = 'relu'))
model.add(Dense(8, activation = 'relu'))
model.add(Dense(1))

# configure the optimizer, loss, and metrics to monitor.
opt = tf.keras.optimizers.Adam(
        learning_rate=0.001,
        beta_1=0.9,
        beta_2=0.999,
        epsilon=1e-07,
        name='Adam')

# Compile the model
model.compile(optimizer=opt, loss='mse', metrics=["mse"])
```

TensorFlow

```
#pragma once

#ifdef __has_attribute
#define HAVE_ATTRIBUTE(x) __has_attribute(x)
#else
#define HAVE_ATTRIBUTE(x) 0
#endif
#if HAVE_ATTRIBUTE(aligned) || (defined(__GNUC__) && !defined(__clang__))
#define DATA_ALIGN_ATTRIBUTE __attribute__((aligned(4)))
#else
#define DATA_ALIGN_ATTRIBUTE
#endif

// automatically configure network
#define TF_NUM_INPUTS 3
#define TF_NUM_OUTPUTS 1
#define TF_NUM_OPS 21
/**
 * Call this function to register the ops
 * that have been detected
 */
template<class TF>
void registerNetworkOps(TF& nn) {
    nn.resolver.AddSlice();
    nn.resolver.AddTranspose();
    nn.resolver.AddFill();
    nn.resolver.AddSoftmax();
    nn.resolver.AddRelu();
    nn.resolver.AddAdd();
    nn.resolver.AddShape();
    nn.resolver.AddLess();
    nn.resolver.AddUnidirectionalSequenceLSTM();
    nn.resolver.AddMul();
    nn.resolver.AddStridedSlice();
    nn.resolver.AddSplit();
    nn.resolver.AddReshape();
    nn.resolver.AddFullyConnected();
    nn.resolver.AddGather();
    nn.resolver.AddMinimum();
    nn.resolver.AddWhile();
    nn.resolver.AddConcatenation();
    nn.resolver.AddTanh();
    nn.resolver.AddMaximum();
    nn.resolver.AddPack();
}
// model data
const unsigned char tfModel[
  14216
] DATA_ALIGN_ATTRIBUTE = {
  0x1c,
  0x00,
  0x00,
  0x00,
  0x54,
  0x46,
  0x4c, ....
        ,
  0x00,
  0x00,
  0x00,
  0x00,
  0x2c
};
```

# CODE GENERATOR

TensorFlow

```python
import tensorflow as tf
from tensorflow.keras.layers import Dense

model = tf.keras.Sequential()
model.add(Dense(16, activation = 'relu'))
model.add(Dense(8, activation = 'relu'))
model.add(Dense(1))

# configure the optimizer, loss, and metrics to monitor.
opt = tf.keras.optimizers.Adam(
        learning_rate=0.001,
        beta_1=0.9,
        beta_2=0.999,
        epsilon=1e-07,
        name='Adam')

# Compile the model
model.compile(optimizer=opt, loss='mse', metrics=["mse"])
```

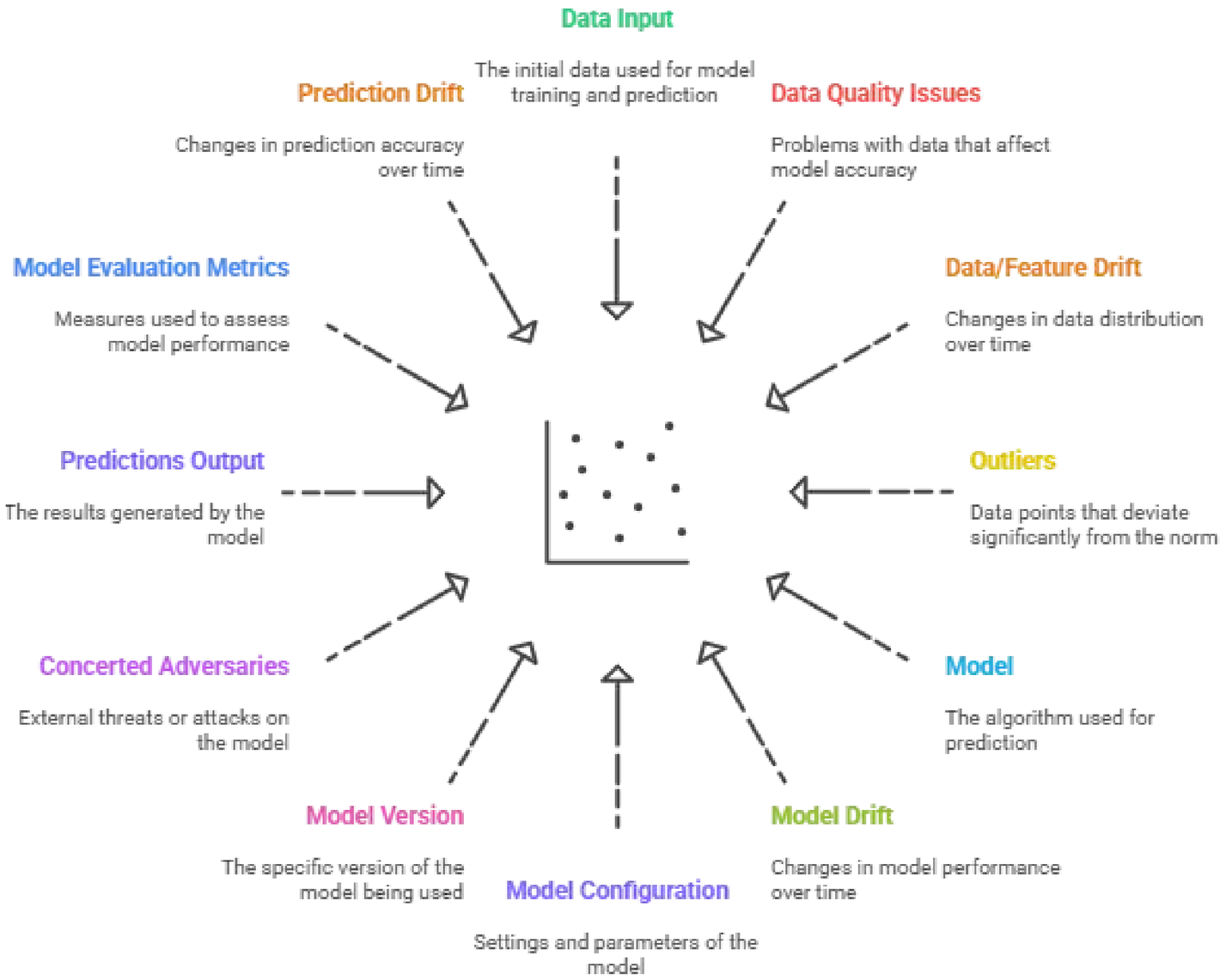
Untitled-1

```
float predict(float *x) {
float y_pred = 0;
static const float w1[3][16] = {...};

static const float b1[16] = {...};

static const float w2[16][8] = {...};

static const float b2[8] = {...};

static const float w3[8][1] = {...};

static const float b3[1] = {...};

    // Input Layer
    float z1[16];
    for (int i = 0; i < 16; i++)
    {
        z1[i] = b1[i];
        for (int j = 0; j < 3; j++)
        {
            z1[i] += x[j] * w1[j][i];
        }
        z1[i] = relu(z1[i]);
    }

    // Hidden Layer 2
    float z2[8];
    for (int i = 0; i < 8; i++)
    {
        z2[i] = b2[i];
        for (int j = 0; j < 16; j++)
        {
            z2[i] += z1[j] * w2[j][i];
        }
        z2[i] = relu(z2[i]);
    }

    // Output Layer
    float z3 = b3[0];
    for (int i = 0; i < 8; i++)
    {
        z3 += z2[i] * w3[i][0];
        z3 = linear(z3);
    }

y_pred = z3;
return y_pred;
}
protected:
float relu(float x)
{
    return x > 0 ? x : 0;
};

float linear(float x)
{
    return x;
};
```

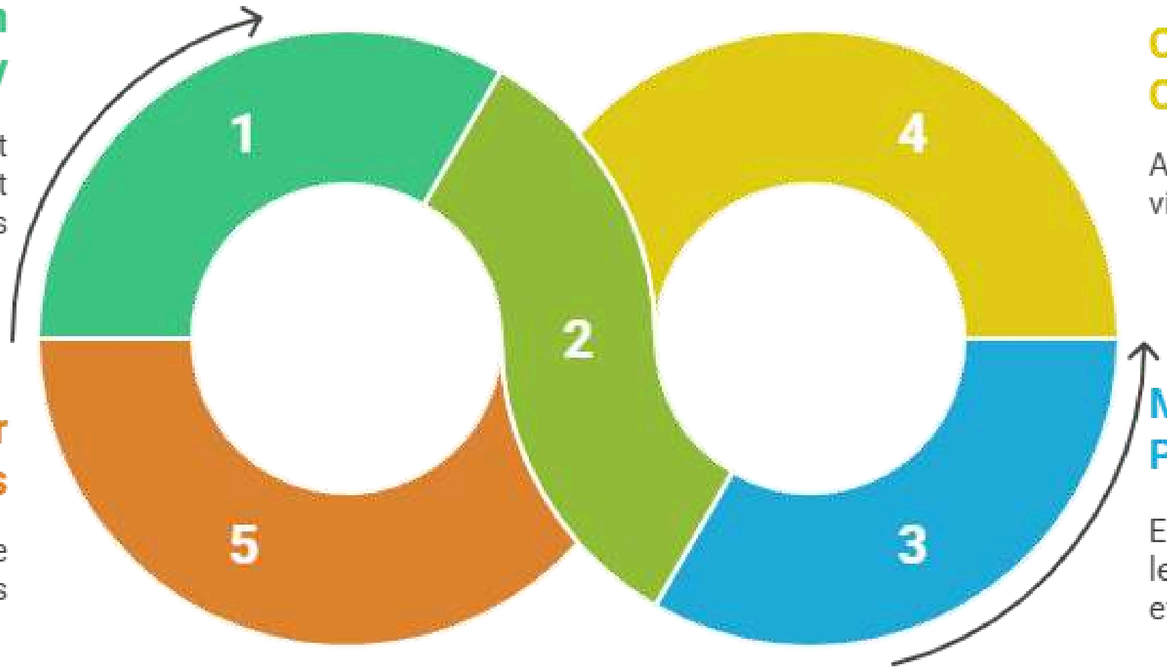
# MODEL MONITORING

## MANUAL, CODE GENERATOR AND INTERPRETERS

# FUNCTIONAL MONITORING



# OPERATIONAL MONITORING




**Evaluate System Reliability**

Ensuring consistent operation without failures

**Consider Hosting Costs**

Assessing financial viability of deployment

**Manage Model Pipelines**

Executing machine learning models effectively

**Manage Data Pipelines**

Facilitating seamless information flow

**Consider Inference Costs**

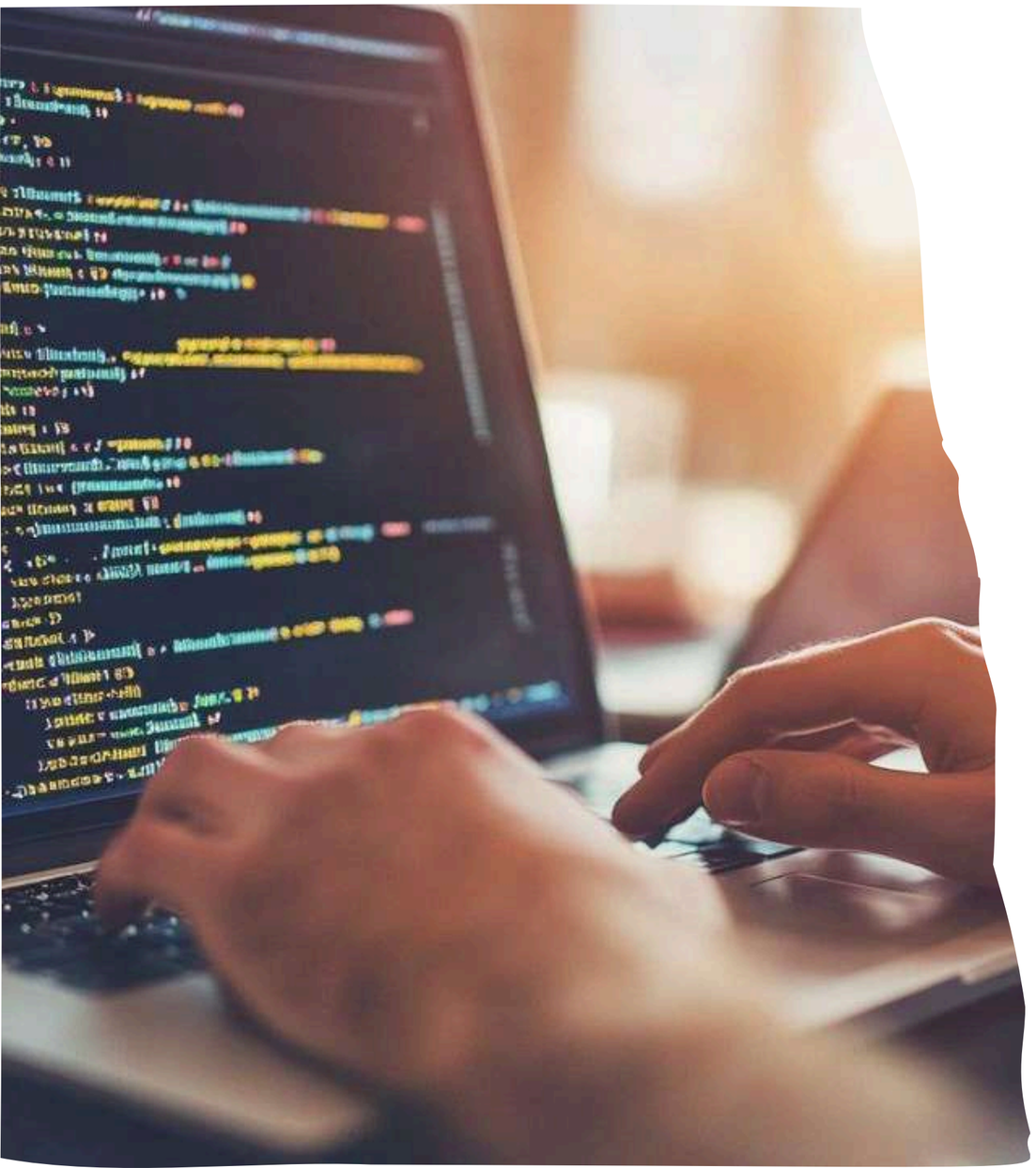
Evaluating maintenance expenses

DESCARBONIZE AI

CONECT2AI

UFRN
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

2025
SBAI
XVII Simpósio Brasileiro de Automação Inteligente



# HANDS ON

# PROJECT REPOSITORY

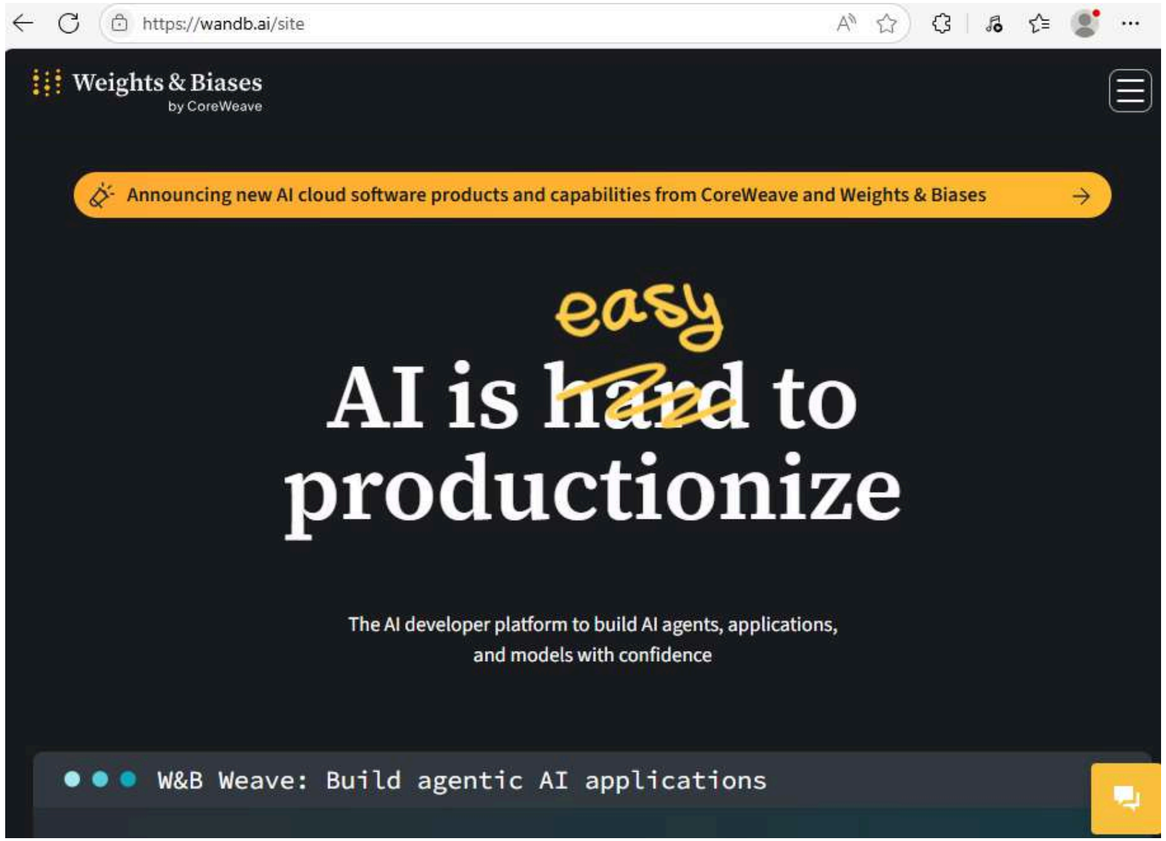
GitHub

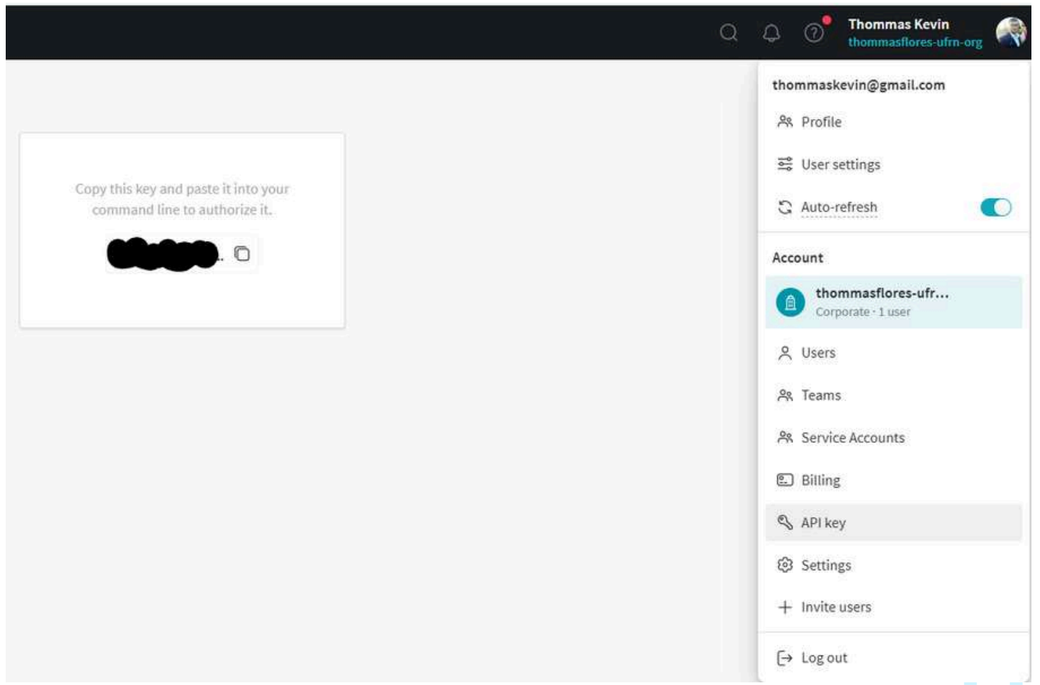
**HTTPS://GITHUB.COM/CONECT2AI/SBAI2025-TENSORFLORES**

**GIVE ME A STAR!**

# CREAT WANDB USER

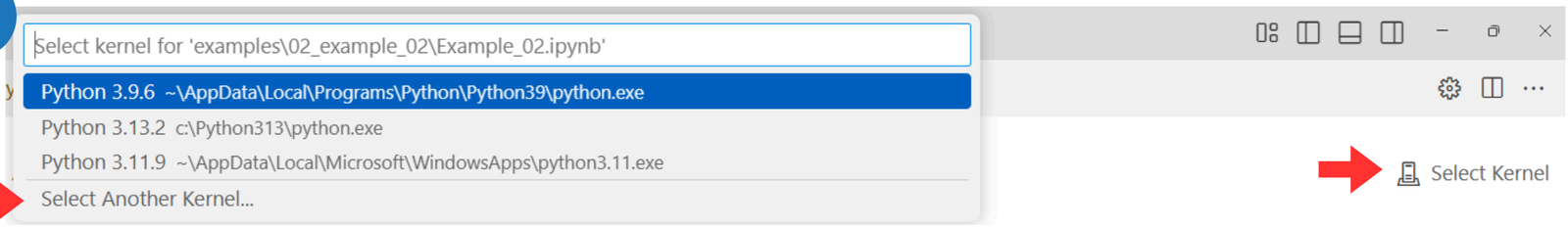
HTTPS://WANDB.AI/SITE



# CREATE VIRTUAL ENVIRONMENT

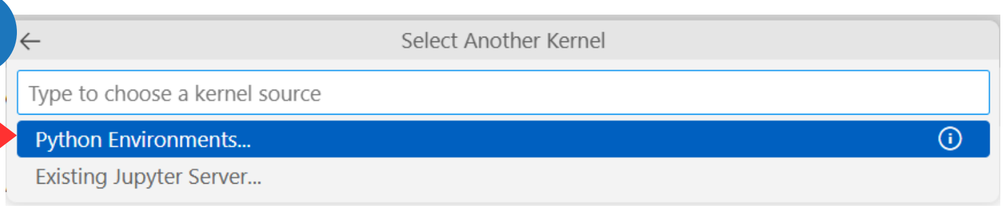
1



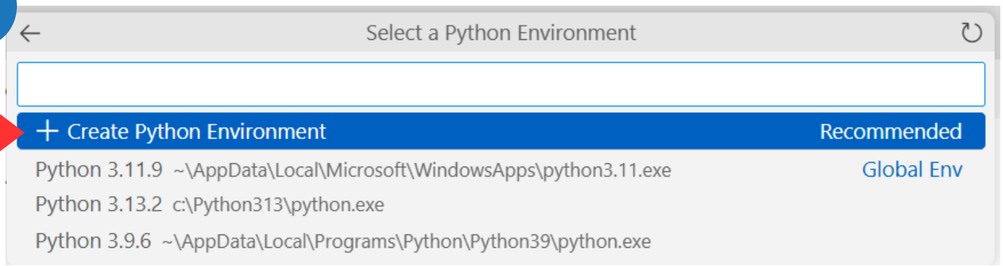




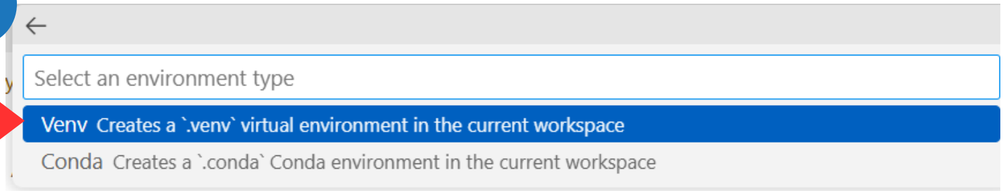
2



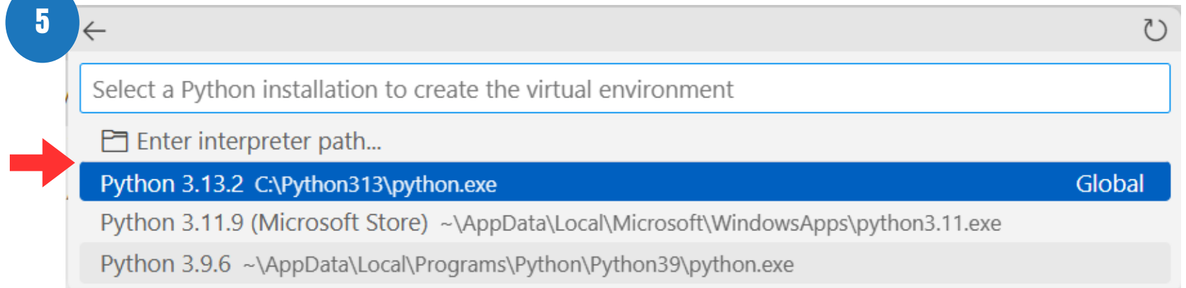

3



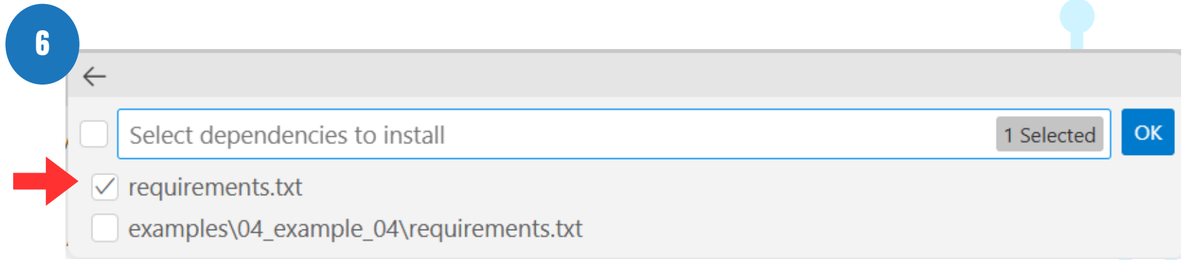

4




5




6



DESCARBONIZEAI

CONECT2AI

UFRN
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

2025
SBAI
XVII Simpósio Brasileiro de Automação Inteligente

# THANK YOU FOR YOUR ATTENTION

CONECT2AI

Medium