



life.augmented



# Machine Learning Core in ST MEMS sensor

ST Korea

지준영 차장 (Jerry Ji)

# Agenda

- 1 Artificial Intelligence introduction
- 2 AI on the edge
- 3 Machine Learning Core in ST sensor
- 4 [Demo] Application state estimation with MLC

# Artificial Intelligence introduction

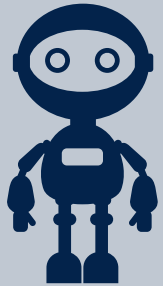


# What is AI?

## The evolution of AI

### Artificial Intelligence

Early Artificial Intelligence stirs excitement



### Machine Learning

Machine Learning begins to flourish



### Deep Learning

Deep Learning breakthroughs drive AI boom



Any technique that enables computer to mimic **human behavior**

**Subset of AI.** Algorithms and methodologies that improve over time through **learning from data**

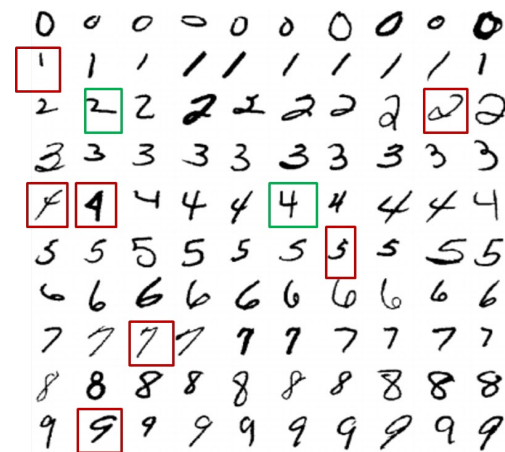
**Subset of ML.** Learning algorithms that derive meaning out of data, by using a hierarchy of multiple layers that **mimic the neural networks of the human brain**

1950' 1960' 1970' 1980' 1990' 2000' 2010' 2020'

# Machine Learning, why do we need it?

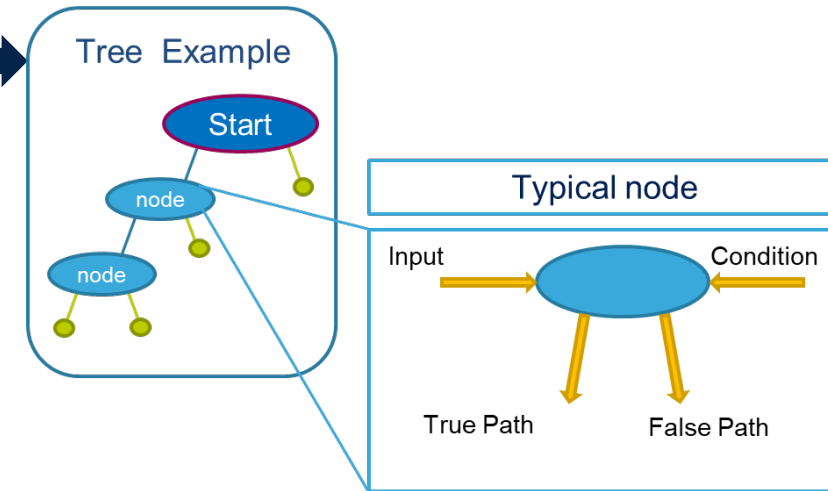
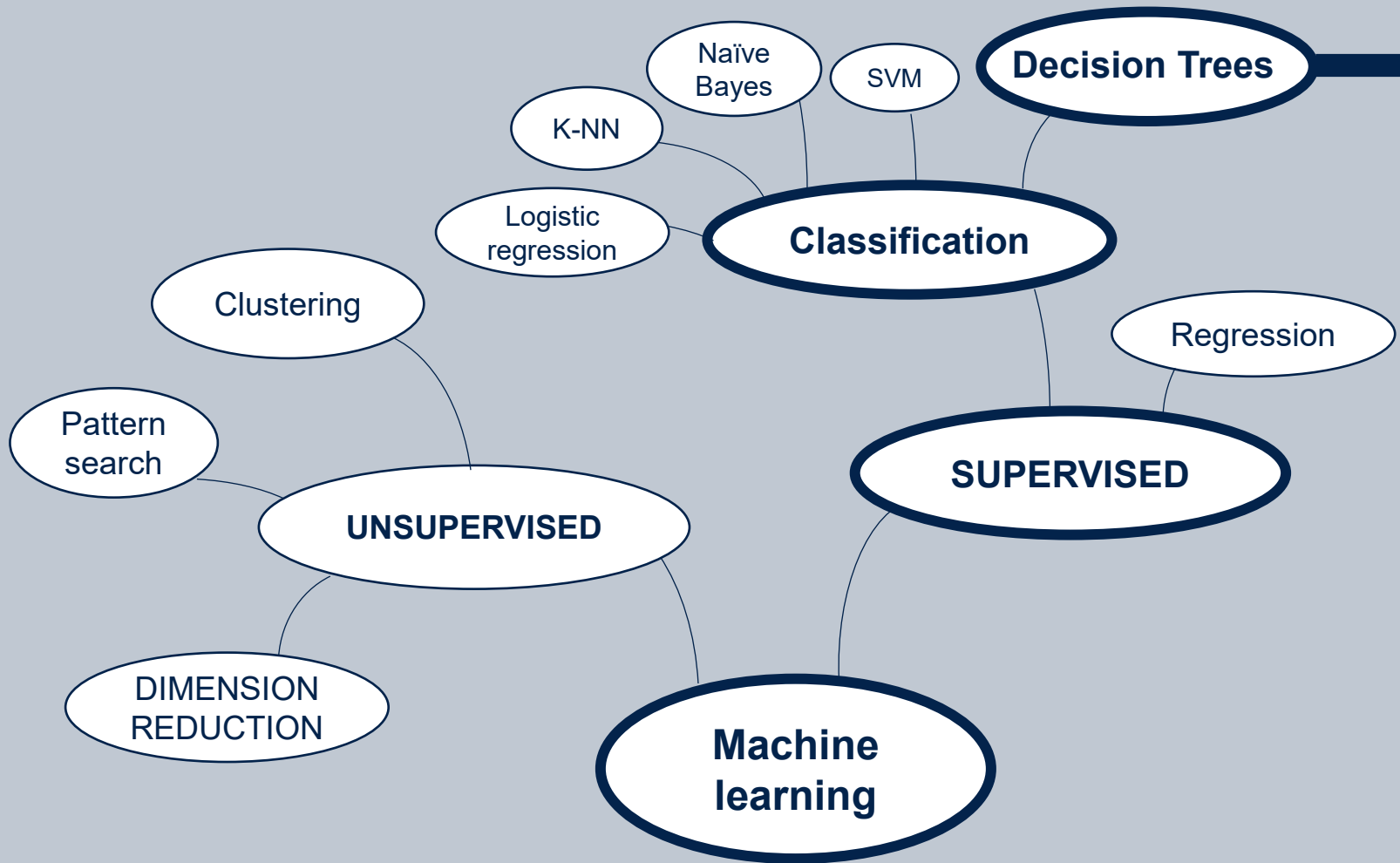
When a complex task or problem involves a large amount of data and lots of variables, **but no existing formula or equation can solve it**

- An example of difficult program
  - How to recognize the hand written digits?
  - Very difficult to define the rules!
  - What makes all these numbers to be identifiable?
  - Is there a pattern?
  - What is it that makes a 2 to be identified as a 2?





# Machine Learning Core(MLC) in ST MEMS sensor



ST sensors embed **Decision trees**

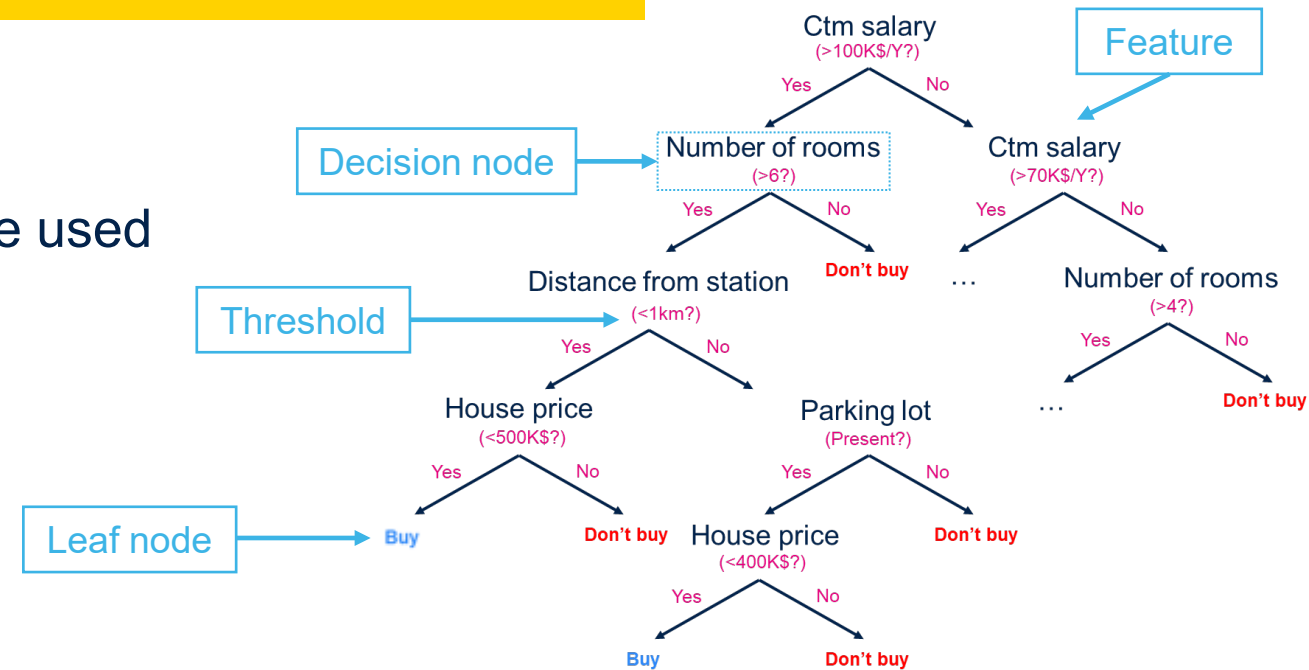
Decision tree runs **Classification**, i.e algorithm that splits objects into classes based on attributes known beforehand

**Supervised machine learning** technique is used to create decision trees

# ML model training Decision tree algorithm

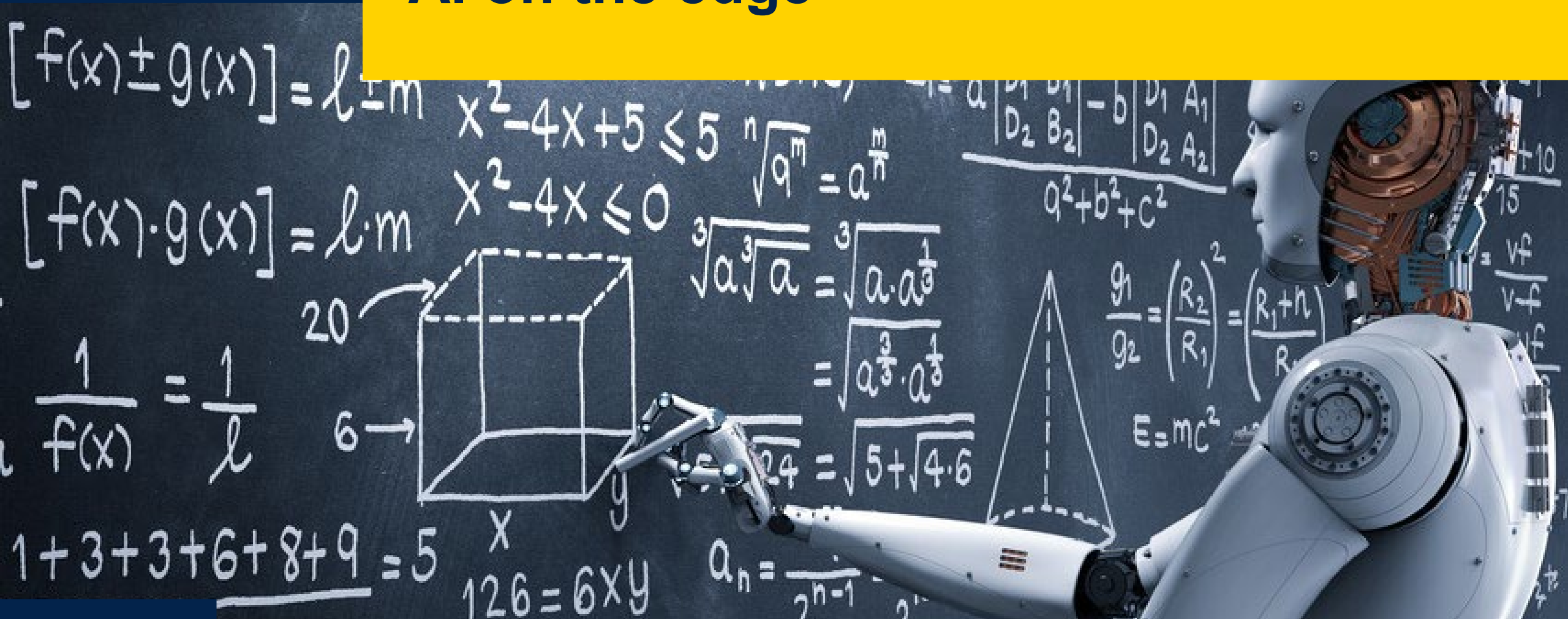
## What the ML model learn from the data?

- Which Features will be used
- In which Decision Nodes the Features will be used
- Threshold values for Decision Nodes
- Leaf Nodes



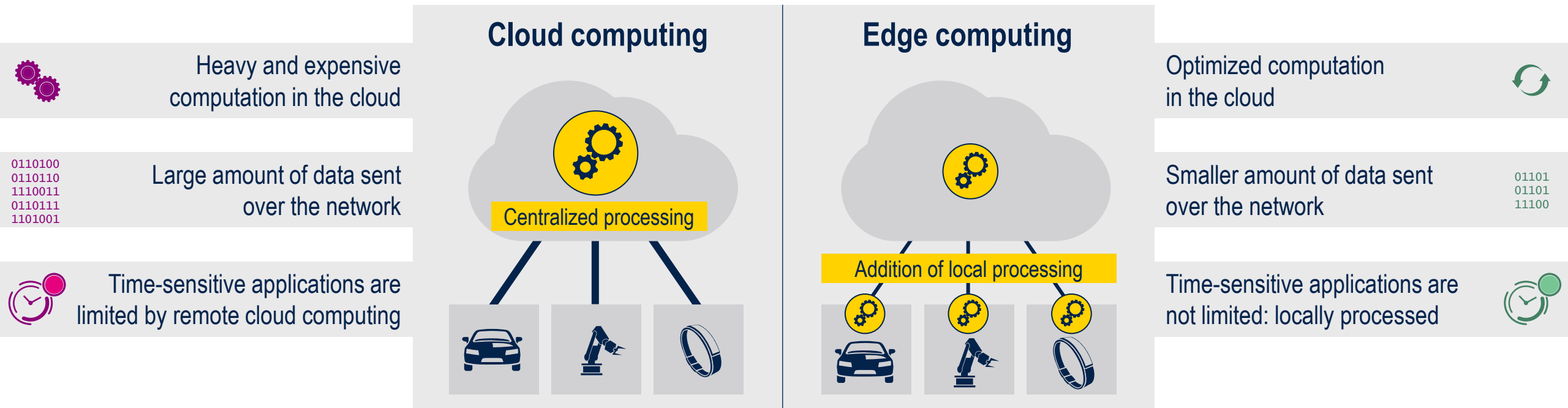
Training the same Machine Learning Model with another dataset (i.e. Houses in Milan), the resulting Decision Tree is different to “best fit” the new data given for training.

# AI on the edge



# Moving AI to edge computing

## Move from Cloud Computing to Edge Computing Now!



**Market needs: Smart sensors with local processing (AI) for real time elaboration and best power efficiency**




**ST MEMS sensor with MLC**



# Artificial Intelligence at STMicroelectronics

- Thanks to STM32Cube.AI, pre-trained **Artificial Neural Networks (ANN)** can be run on STM32 microcontrollers.
- Advanced sensors contain a **Machine Learning Core (MLC)**, a Finite State Machine (FSM), and advanced digital functions. They run custom algorithms on the IMU and share the workload from the main processor enabling system functionality while significantly saving power.



Neural Networks on STM32  
Simple, fast, optimized

▶ Learn more



INEMO

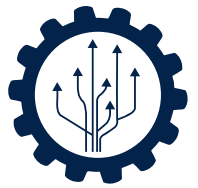
Machine learning core  
6-axis inertial module

▶ Learn more

# Machine Learning Core in ST sensor



life.augmented

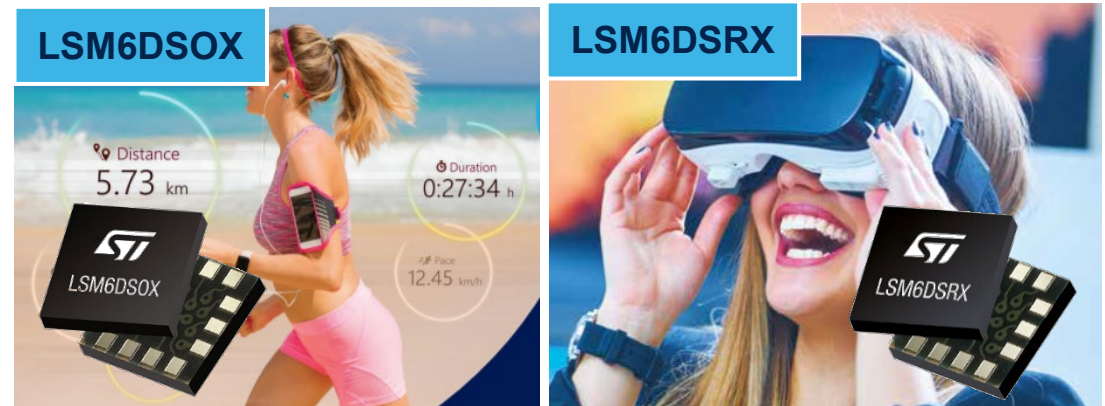


# New X-Factor sensor products with MLC LSM6DSOX, LSM6DSRX, ISM330DHCX , IIS2ICLX

We create the new generation of **sensors** to allow **developers** exploiting their potential by improving the overall system **efficiency**

- ST has **all the building blocks** for the **Machine Learning Core** for tools and software
- Our constantly expanding development ecosystem makes **design fast** and help you **reduce the time-to-market**

## Consumer grade



## Industrial grade



# Basic process of MLC development

How it works in 5 simple steps and with an intuitive use case



User defines **classes** to be recognized and **collects data logs**.



Clean and **label logs**. Define **features** best characterizing the identified classes.



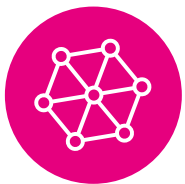
Machine Learning tools **build tree** based on logs and features.



ST tool generates sensor configuration with **embedded decision tree**.



**Configure** the sensor and **run** the application.



Capture data



Label data &  
extract features



Build decision tree



Embed decision tree



Process new data

# Data collection – ST HW and SW tools

## Form Factor

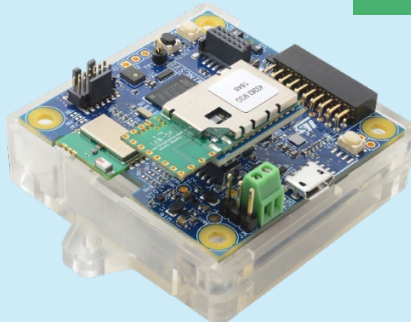
SensorTile.Box  
STEVAL-MKSBOX1V1

CONSUMER



STWIN: Wireless Industrial Node  
STEVAL-STWINKT1

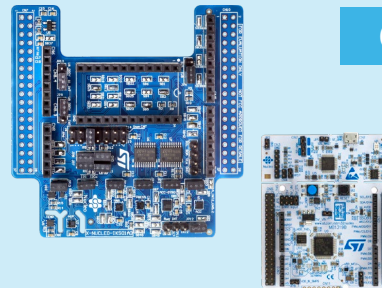
INDUSTRIAL



## Evaluation board

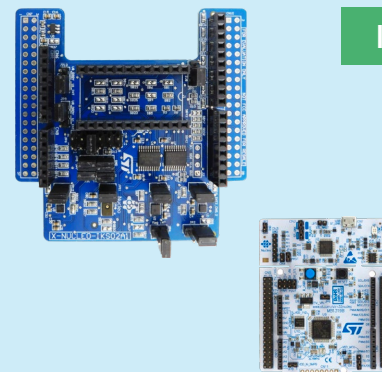
STM32Nucleo expansion  
X-NUCLEO-IKS01A3

CONSUMER



STM32Nucleo expansion  
X-NUCLEO-IKS02A1

INDUSTRIAL



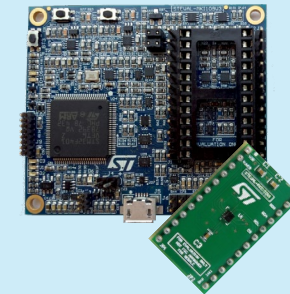
## Professional board

Profi MEMS tool  
STEVAL-MKI109V3

AUTOMOTIVE

INDUSTRIAL

CONSUMER



## Software solutions

AlgoBuilderSuite

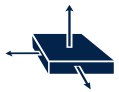
Algo  
Builder



All-in-one software package facilitate the programming of sensors for an easy and intuitive experience without writing any single line of code.

# Features calculation inside MEMS

## Raw data



Acc\_X, Acc\_Y, Acc\_Z  
Acc\_V, Acc\_V2



Gyro\_X, Gyro\_Y, Gyro\_Z  
Gyro\_V, Gyro\_V2



ExtSens\_X, ExtSens\_Y,  
ExtSens\_Z  
ExtSens\_V, Ext\_Sens\_V2



## Filters

Filtered data  
(High-pass, Band-pass,  
IIR1 and IIR2 filters)

## Features

Mean

$$\frac{1}{WL} \sum_{k=0}^{WL-1} I_k$$

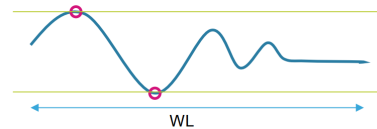
Variance

$$\left( \frac{\sum_{k=0}^{WL-1} I_k^2}{WL} \right) - \left( \frac{\sum_{k=0}^{WL-1} I_k}{WL} \right)^2$$

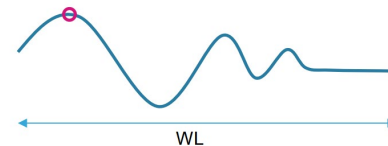
Energy

$$\sum_{k=0}^{WL-1} I_k^2$$

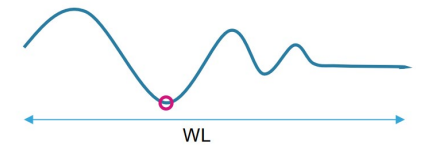
Peak-to-Peak



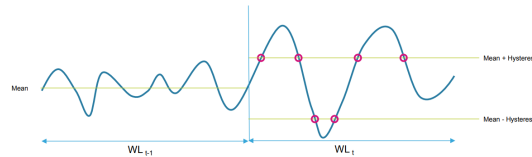
Maximum



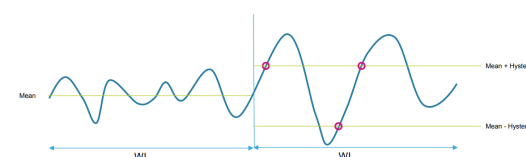
Minimum



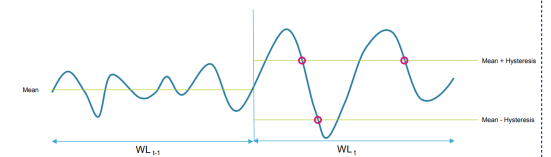
Zero crossing



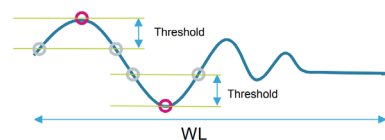
Positive Zero crossing



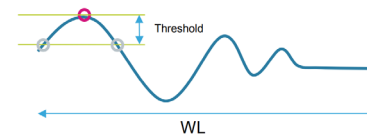
Negative Zero crossing



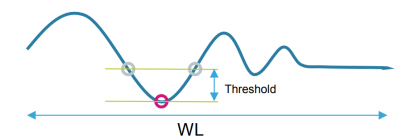
Peak detector



Positive Peak detector



Negative Peak detector



Note: the window for features computation (WL) is configurable (from 1 to 255 samples). It is not a moving window

# Decision tree algorithm

## Failed motor detection example

### PROBLEM:

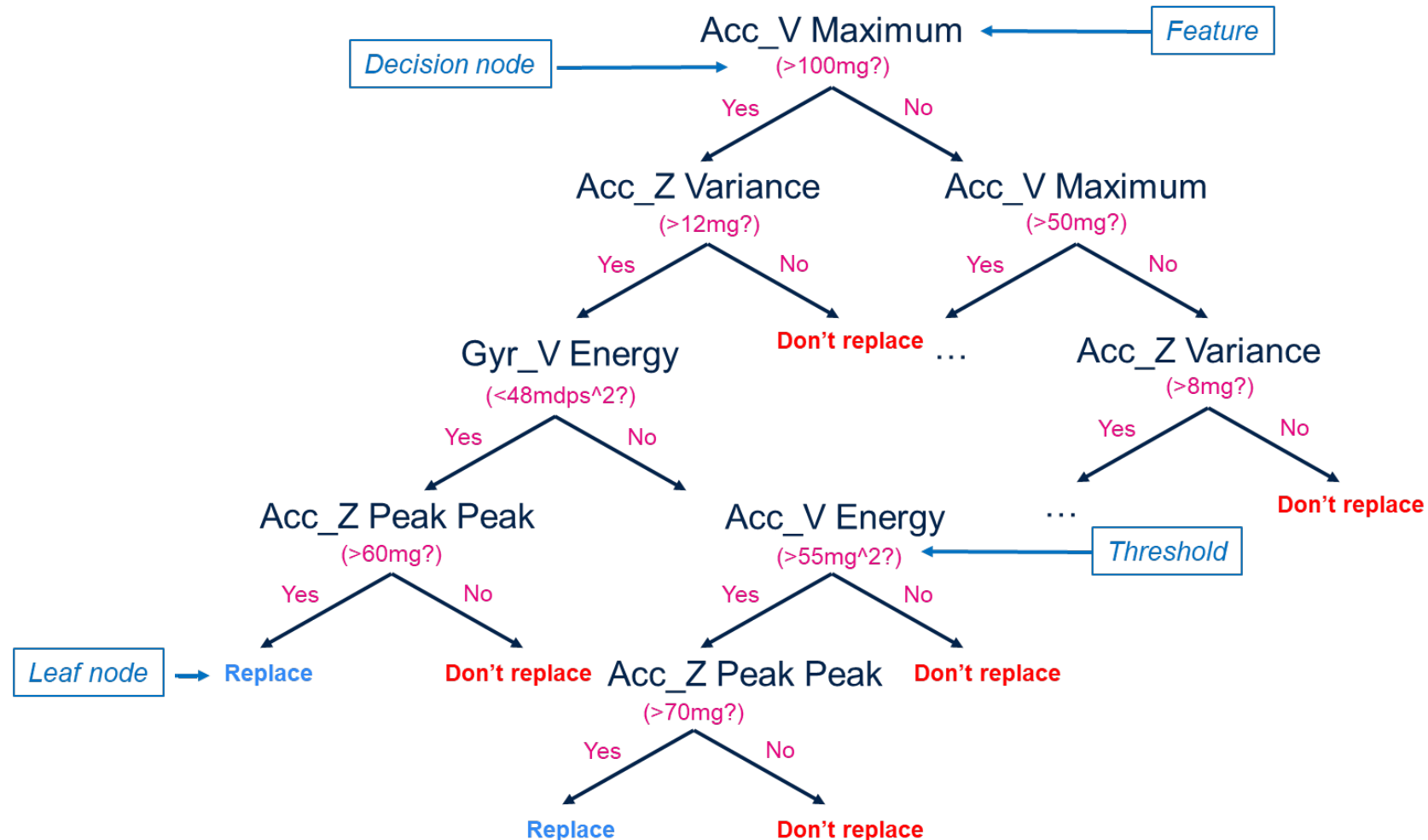
- Does the motor need replacement?

### FEATURES (Inputs):

- Acc\_X Peak Peak
- Acc\_Z Peak Peak
- Acc\_Z Variance
- Acc\_V Energy
- Acc\_V Maximum
- Gyr\_Z Variance
- Gyr\_V Energy
- Gyr\_V Maximum
- ...

### OUTPUT:

- Replace or Don't replace



# [Demo] Application state estimation with MLC

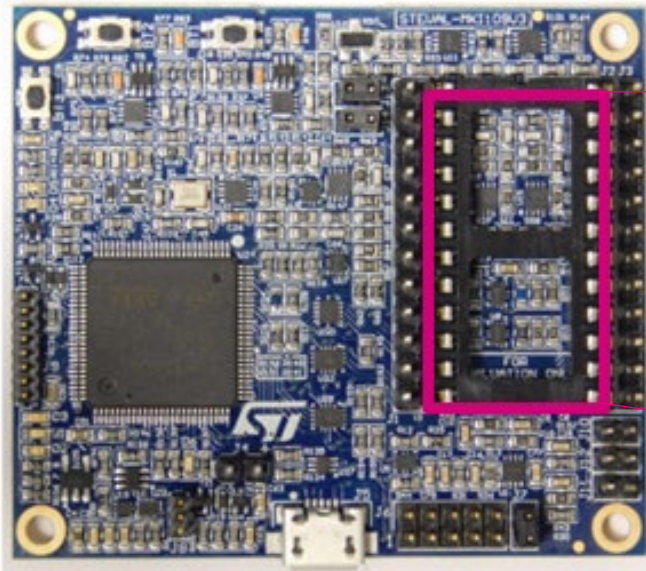


# H/W: ProfiMEMS board with LSM6DSOX

General-purpose MEMS sensor evaluation tool

Professional MEMS mother board

Sensor adapter board



ST MEMS motherboard with STM32F401VE  
and compatible with all ST MEMS adapters

**STEVAl-MKI109V3**



Sensor adapter board  
for a standard DIL24 socket

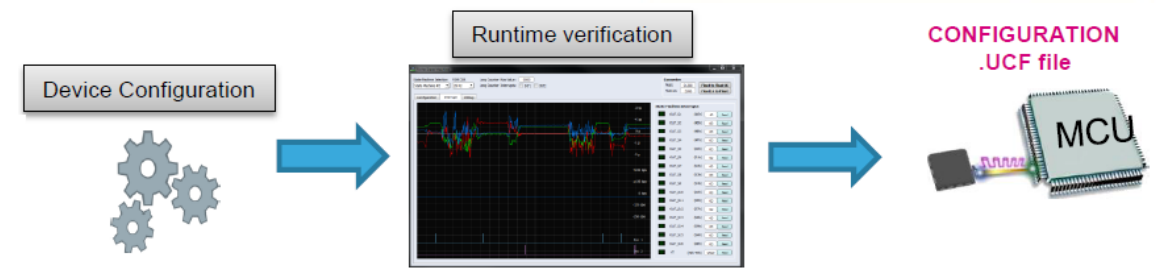
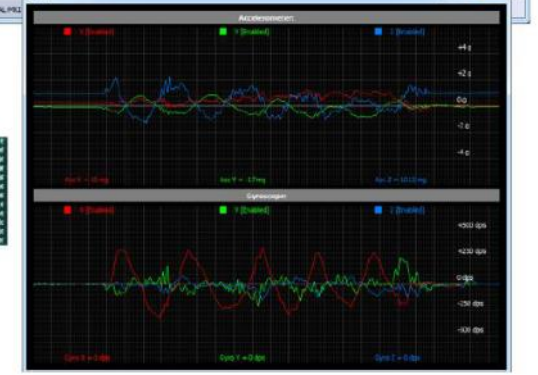
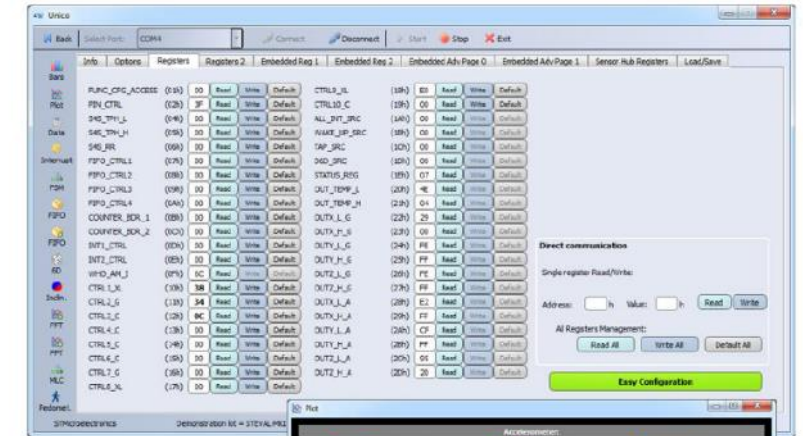
**LSM6DSOX(STEVAl-MKI197V1)**



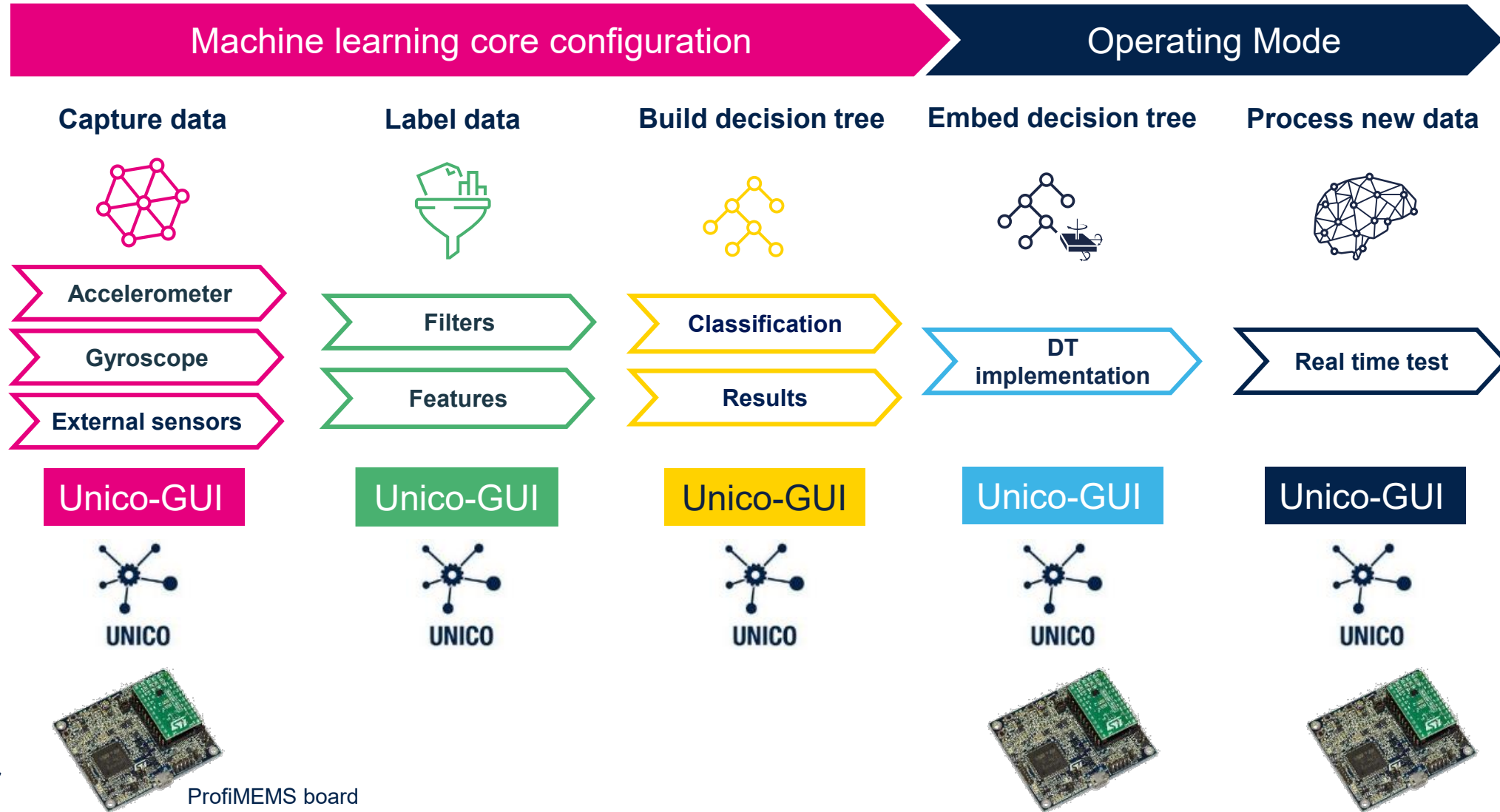
# S/W: Unico-GUI(PC application)

## PC application for sensor evaluation

- All **Motion/ Environmental MEMS** adapter boards (STEVAL-MKIxxx) supported
  - Compatible with Profi MEMS tool (STEVAL-MKI109V3)
- Cross-platform GUI for **Windows, Linux and MacOS X**
  - STSW-MKI109W for Windows OS
  - STSW-MKI109L for Linux OS
  - STSW-MKI109M for Mac OS
- **Features:**
  - Device registers configuration, Data logging
  - Configuration file (.ucf) creation & C code generation
  - **FSM / MLC development support**



# Creating a decision tree with Unico-GUI



# Application state estimation demo with MLC

Detect the following 3 states with the Machine Learning Core!

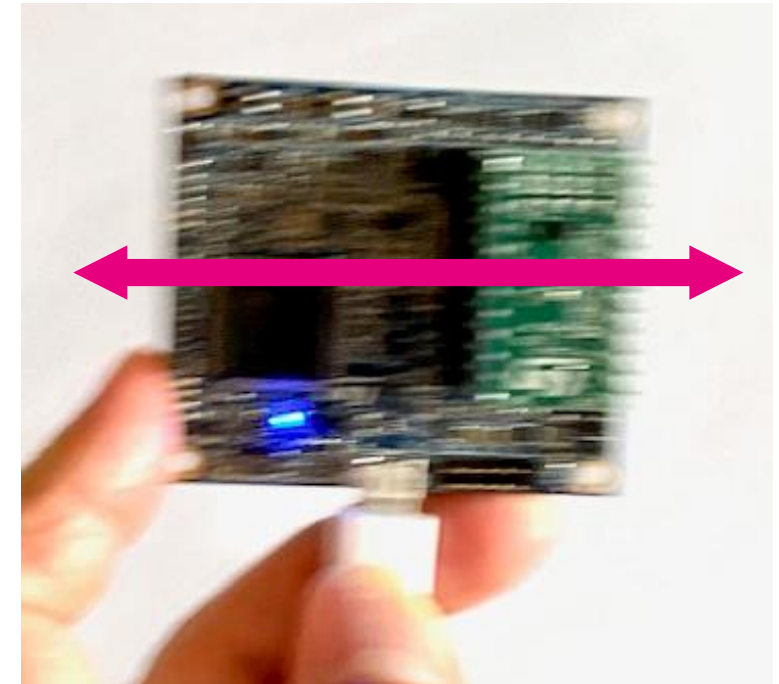
State #1: Face-up



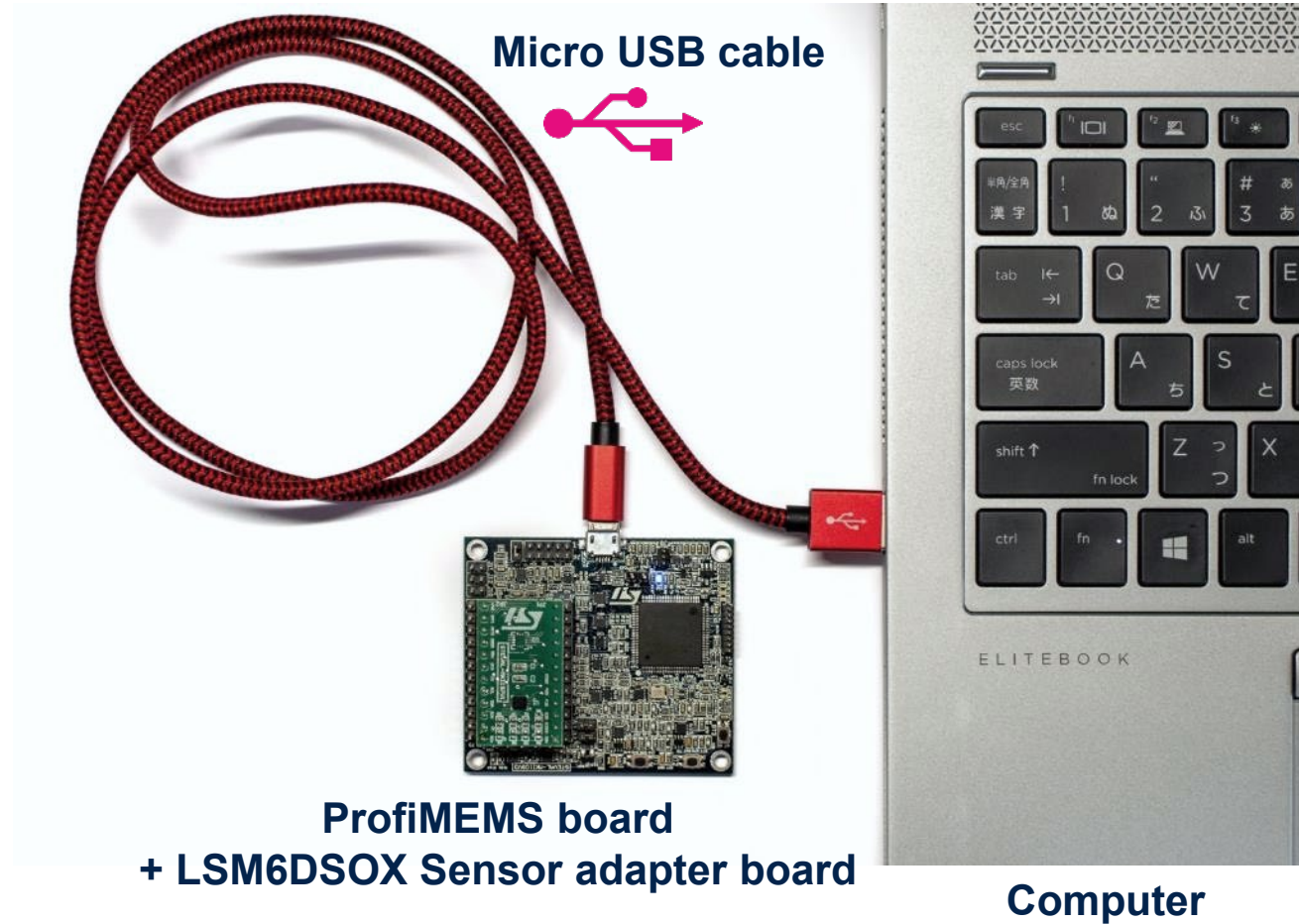
State #2: Face-down



State #3: Shaking

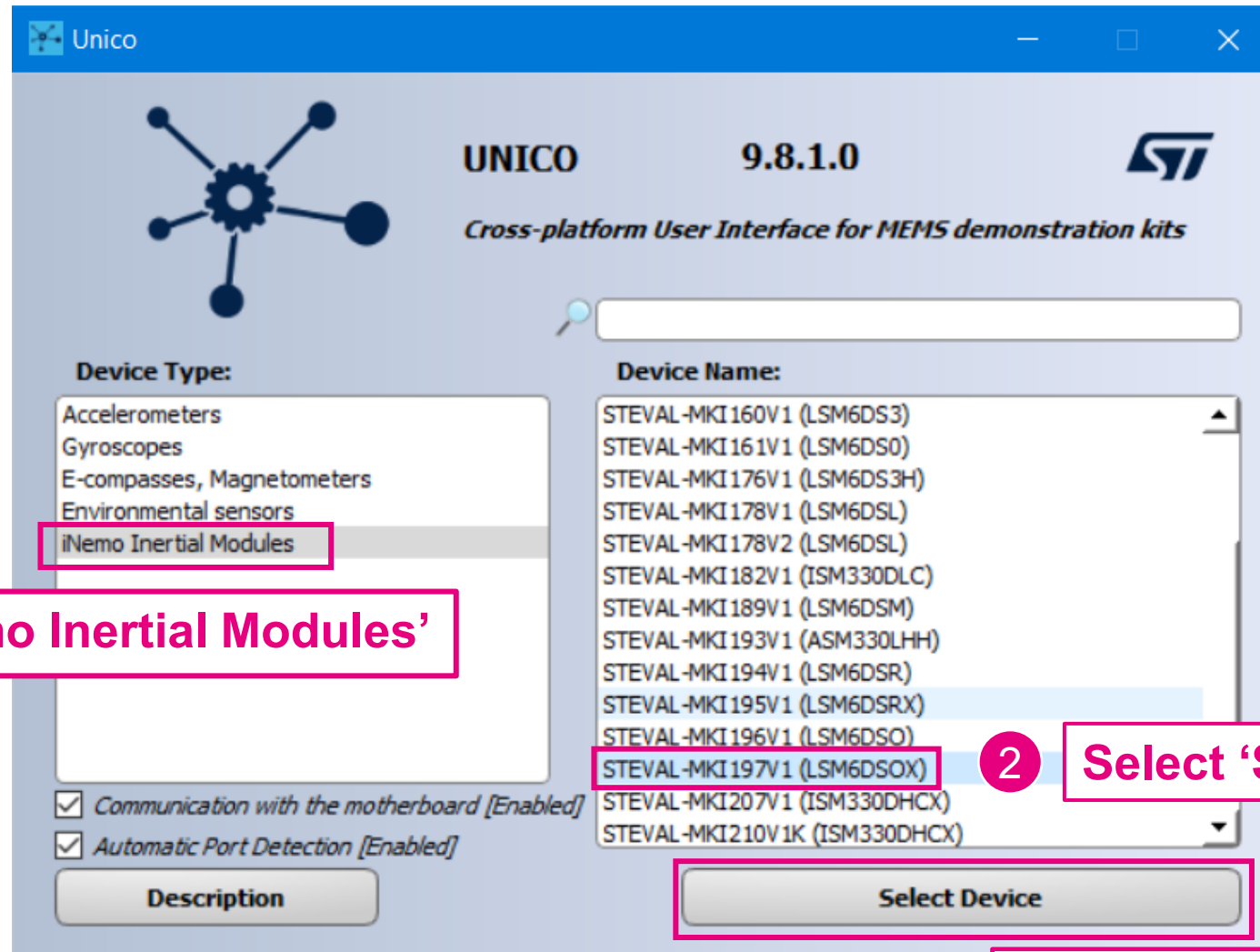


# STEP 1- (a): Connect the board to the PC



- 1 Connect ProfiMEMS board with LSM6DSOX sensor adapter board to PC USB port

# STEP 1-(b): Execute Unico-GUI tool



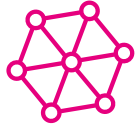
1 Select 'iNemo Inertial Modules'

2 Select 'STEVAL-MKI197V1'

3 Press 'Select Device' button



Log collection



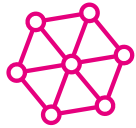
# STEP 2: Set ODR and Full scale

The screenshot shows the UNICO configuration tool interface. The 'Options' tab is selected. The interface is annotated with four numbered callouts:

- 1** Open 'Options' tap: Points to the 'Options' tab in the top navigation bar.
- 2** Set accelerometer ODR 26Hz and FS  $\pm 2g$ : Points to the 'Accelerometer Output Data Rate' (set to 26 Hz) and 'Accelerometer Full Scale' (set to 2 g) dropdown menus.
- 3** Set Gyro ODR Power Down: Points to the 'Gyroscope Output Data Rate' dropdown menu, which is set to 'Power-Down'.
- 4** Press 'Start' button: Points to the 'Start' button in the top toolbar.

Other visible settings include: Easy Configuration (button), Gyroscope Sleep Mode (Sleep mode Disabled), Gyroscope Full Scale (dropdown), Gyroscope High-Performance Mode (High-Performance Enabled), and C code generation (button).

Bottom status bar: STMicroelectronics | Demonstration kit = STEVAL-MKI197V1 (LSM6DSOX) | Board = ProfiMEMSTool | Firmware Version = V3.6.26 | Unico Version = 9.8.1.0



# STEP 3- (a): Get the log file

1 Open 'Load/Save' tap

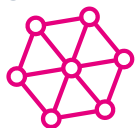
2 Specify the save location and file name

3 Check only 'Acceleration'

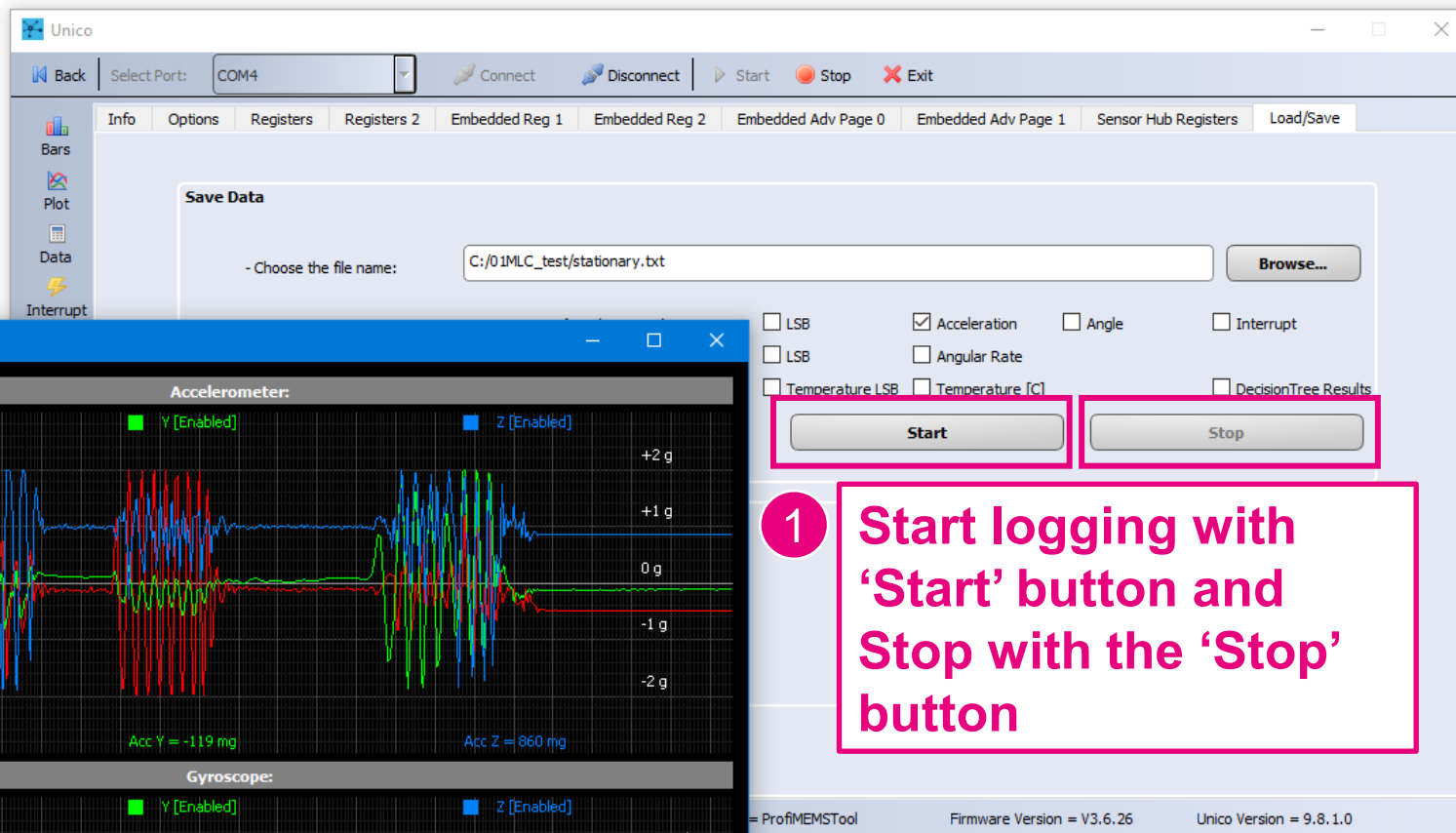
STMicroelectronics Demonstration kit = STEVAL-MK1197V1 (LSM6DSOX) Board = ProfMEMSTool Firmware Version = V3.6.26 Unico Version = 9.8.1.0



Log collection



# STEP 3- (b): Get the log file



**1** Start logging with 'Start' button and Stop with the 'Stop' button

**2** Get a log file for each state  
 State #1: Face-up  
 State #2: Face-down  
 State #3: Shaking



Labeling



# STEP 4: MLC settings and decision tree creation

The screenshot shows the Unico software interface. The window title is "Unico". The top menu bar includes "Back", "Select Port: COM4", "Connect", "Disconnect", "Start", "Stop", and "Exit". The main content area displays the UNICO logo and the text "Cross-platform User Interface for MEMS demonstration kits". Below this, it shows "GUI Version: 9.8.1.0" and "Firmware Version: V3.6.26". The selected demonstration kit is "STEVAL-MKI197V1 (LSM6DSOX)" and the board is "ProfIMEMSTool". The ST logo and "life.augmented" are in the bottom right corner. A status bar at the bottom shows "Demonstration kit = STEVAL-MKI197V1 (LSM6DSOX)", "Board = ProfIMEMSTool", "Firmware Version = V3.6.26", and "Unico Version = 9.8.1.0".

On the left sidebar, the "MLC" icon is highlighted with a red box and a pink circle containing the number "1". Below this, a pink box contains the text "Launch MLC tool".



Labeling



# STEP 4: MLC settings and decision tree creation

Machine Learning Core

Data Patterns Configuration

Load Data Pattern

**Browse...** Advanced Options...

1

Press 'Browse' button

Current Data Pattern

*Set Class (Label):*  **LOAD**

Data Patterns Loaded **CLEAR**

MLC Tool



Labeling



# STEP 4: MLC settings and decision tree creation

Machine Learning Core

Open File

This PC > Windows (C:) > 01MLC\_test

Search 01MLC\_test

Name	Date modified	Type
SCAN_16S		
OneDrive - STMicroelectroni		
This PC		
3D Objects		
Desktop		
Documents		
Downloads		
Music		
Pictures		
Videos		
Windows (C:)		
common (\tkyna003.tky.st.cor		

1

Select log file

File name: 1\_Log1\_FaceUp.txt

Data Pattern File (\*.txt \*.csv \*.tsv)

Open Cancel



Labeling



# STEP 4: MLC settings and decision tree creation

The screenshot shows the 'Machine Learning Core' software interface. The 'Configuration' tab is active. Under 'Load Data Pattern', the file path 'C:/01MLC\_test/1\_Log1\_FaceUp.txt' is entered. Below this is a table titled 'Current Data Pattern' with columns for Sample [#], AccX, AccY, AccZ, GyrX, GyrY, GyrZ, ExtX, ExtY, and ExtZ. The table contains data for samples 1926 through 1931. A red circle with the number '2' is placed over the 'GyrX' column for sample 1931. Below the table, the 'Set Class (Label):' field contains 'face\_up', with a red circle and the number '1' next to it. A green 'LOAD' button is highlighted with a red box, and a red box with the text 'Press 'LOAD' button' is placed over it. Below the 'Set Class (Label):' field, a red box with the text 'Set label' is placed over the 'face\_up' text. At the bottom of the interface, there is a 'Data Patterns Loaded' section with a 'CLEAR' button.

Sample [#]	AccX	AccY	AccZ	GyrX	GyrY	GyrZ	ExtX	ExtY	ExtZ
1926	-34	392	926	0	0	0	0	0	0
1927	-63	376	902	0	0	0	0	0	0
1928	-47	411	965	0	0	0	0	0	0
1929	-38	376	903	0	0	0	0	0	0
1930	-15	390	939	0	0	0	0	0	0
1931	-1	396	941	0	0	0	0	0	0



Labeling



# STEP 4: MLC settings and decision tree creation

The screenshot shows the 'Machine Learning Core' software interface. It has two tabs: 'Data Patterns' and 'Configuration'. The 'Data Patterns' tab is active. It contains a 'Load Data Pattern' section with a text input field, a 'Browse...' button, and an 'Advanced Options...' button. Below this is the 'Current Data Pattern' section, which is a table with columns: Sample [#], AccX, AccY, AccZ, GyrX, GyrY, GyrZ, ExtX, ExtY, and ExtZ. Below the table is a 'Set Class (Label):' input field and a green 'LOAD' button. At the bottom is the 'Data Patterns Loaded' section, which contains a table with columns: Pattern [#], Samples [#], Result, and Location. A 'CLEAR' button is located to the right of this table. A pink box highlights the 'Data Patterns Loaded' table and a callout box below it.

Pattern [#]	Samples [#]	Result	Location
0	1932	face_up	C:/01MLC_test/1_Log1_FaceUp.txt
1	1759	face_down	C:/01MLC_test/1_Log2_FaceDown.txt
2	1369	shaking	C:/01MLC_test/1_Log3_Shaking.txt

**1** Label and load logs corresponding to all acquired states



Labeling



# STEP 4: MLC settings and decision tree creation

The screenshot shows the 'Machine Learning Core configuration' window. At the top, there are two tabs: 'Data Patterns' and 'Configuration'. The 'Configuration' tab is selected and highlighted with a pink box and a pink circle containing the number '1'. A pink callout box next to it says 'Open 'Configuration' tap'. Below the tabs, the 'Device' section is visible, with the text 'Select the device:' followed by a dropdown menu. The dropdown menu is open, and 'LSM6DSOX' is selected, highlighted with a pink box and a pink circle containing the number '2'. A pink callout box next to it says 'Select "LSM6DSOX"'. At the bottom of the window, there is a 'Reset' button on the left, a green progress bar at 100% in the center, and a 'Next' button on the right. The 'Next' button is highlighted with a pink box and a pink circle containing the number '3'. A pink callout box next to it says 'Press 'Next' button'.



Labeling



# STEP 4: MLC settings and decision tree creation

Machine Learning Core

Data Patterns Configuration

Machine Learning Core configuration

Device  
Select the device: LSM6DSOX

Machine Learning Core ODR  
Select the internal data rate for the Machine Learning Core: 26 Hz

1 Set MLC ODR to 26Hz

2 Press 'Next' button

Reset 100% Next



Labeling



# STEP 4: MLC settings and decision tree creation

The screenshot shows the 'Machine Learning Core configuration' window. It has two tabs: 'Data Patterns' and 'Configuration'. The 'Configuration' tab is active. The window contains several sections:

- Device:** A dropdown menu with 'LSM6DSOX' selected.
- Machine Learning Core ODR:** A dropdown menu with '26 Hz' selected.
- Inputs:** A dropdown menu with 'Accelerometer only' selected.
- Accelerometer:** Two dropdown menus: 'Full scale:' with '2 g' selected, and 'ODR:' with '26 Hz' selected.

At the bottom of the window, there is a 'Reset' button, a green progress bar at 100%, and a 'Next' button. Three numbered callouts are overlaid on the interface:

- 1 Select 'Accelerometer only'
- 2 Full scale:  $\pm 2g$   
ODR: 26Hz
- 3 Press 'Next' button



Labeling



# STEP 4: MLC settings and decision tree creation

Machine Learning Core configuration

Select the device: LSM6DSOX

Machine Learning Core ODR

Select the internal data rate for the Machine Learning Core: 26 Hz

Inputs

Select the Machine Learning Core inputs: Accelerometer only

Accelerometer

Full scale: 2 g

ODR: 26 Hz

Decision trees

Number of decision trees: 1

Reset

100%

Next

1

Set the number of decision trees to create → Normally one

2

Press 'Next' button



Labeling



# STEP 4: MLC settings and decision tree creation

Machine Learning Core configuration

Select the internal data rate for the Machine Learning Core: 26 Hz

**Inputs**

Select the Machine Learning Core inputs: Accelerometer only

**Accelerometer**

Full scale: 2 g

ODR: 26 Hz

**Decision trees**

Number of decision trees: 1

**Window length**

Number of samples for the window of interest: 26

Reset [Progress Bar: 100%] Next

1

Set "Window length" to 26

2

Press 'Next' button



Labeling



# STEP 4: MLC settings and decision tree creation

1 If you don't use a filter  
Leave as the default

2 Press 'Next'  
button



Labeling



# STEP 4: MLC settings and decision tree creation

The screenshot shows the 'Machine Learning Core configuration' window. It has two tabs: 'Data Patterns' and 'Configuration'. The 'Configuration' tab is active. Under 'Machine Learning Core configuration', there is a dropdown for 'Number of samples for the window of interest' set to 25. Below that is 'Filter configuration' with a dropdown set to 'End filters configuration'. The 'Features' section contains a table with the following rows:

Feature	Input	Enabled	Signed	Action
Mean	ACC_X	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Mean	ACC_Y	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Mean	ACC_Z	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Mean	ACC_V	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Mean	ACC_V2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Variance	ACC_X	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Variance	ACC_Y	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Variance	ACC_Z	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Variance	ACC_V	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Variance	ACC_V2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)

Annotations on the screenshot:

- 1. A pink box highlights the 'Enabled' checkbox for the 'Mean' feature with input 'ACC\_Z'. A callout box says '1 Enable Mean of ACC\_Z'.
- 2. A pink box highlights the 'Enabled' checkbox for the 'Variance' feature with input 'ACC\_V'. A callout box says '2 Enable Variance of ACC\_V'.
- 3. A pink box highlights the 'Next' button at the bottom right. A callout box says '3 Press 'Next' button'.



Labeling



# STEP 4: MLC settings and decision tree creation

Machine Learning Core configuration

Configure one filter: [End filters configuration]

Feature	Input	Enabled	Signed	
Variance	ACC_Z	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Variance	ACC_V	<input checked="" type="checkbox"/>		
Variance	ACC_V2	<input checked="" type="checkbox"/>		
Energy	ACC_X	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Energy	ACC_Y	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Energy	ACC_Z	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Energy	ACC_V	<input type="checkbox"/>		
Energy	ACC_V2	<input type="checkbox"/>		
Peak to Peak	ACC_X	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Peak to Peak	ACC_Y	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	(+)

Save ARFF file

ARFF file: C:/01MLC\_test/Epen\_features.arff [Browse]

Reset [Progress bar: 100%] Next

1 Save the input file containing the calculated features in ARFF format

2 Press 'Next' button



Labeling



# STEP 4: MLC settings and decision tree creation

The screenshot shows the 'Machine Learning Core' configuration window. It has two tabs: 'Data Patterns' and 'Configuration'. The 'Configuration' tab is active, showing a table of feature configurations. Below the table is a 'Save ARFF File' section with a text input field and a 'Browse' button. At the bottom is a 'Decision Tree #1 Results' section with three input fields for 'face\_up' (0), 'face\_down' (4), and 'shaking' (8). At the very bottom are a 'Reset' button, a green progress bar at 100%, and a 'Next' button. A vertical label 'MLC Tool' is on the right side of the window.

Feature	Input	Enabled	Signed	Action
Mean	ACC_Y	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Mean	ACC_Z	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Mean	ACC_V	<input type="checkbox"/>		
Mean	ACC_V2	<input type="checkbox"/>		
Variance	ACC_X	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Variance	ACC_Y	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Variance	ACC_Z	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(+)
Variance	ACC_V	<input checked="" type="checkbox"/>		
Variance	ACC_V2	<input type="checkbox"/>		

Save ARFF File:  Browse

Decision Tree #1 Results  
Insert the result values [from 0 to 15] for decision tree #1:

face\_up:       face\_down:       shaking:

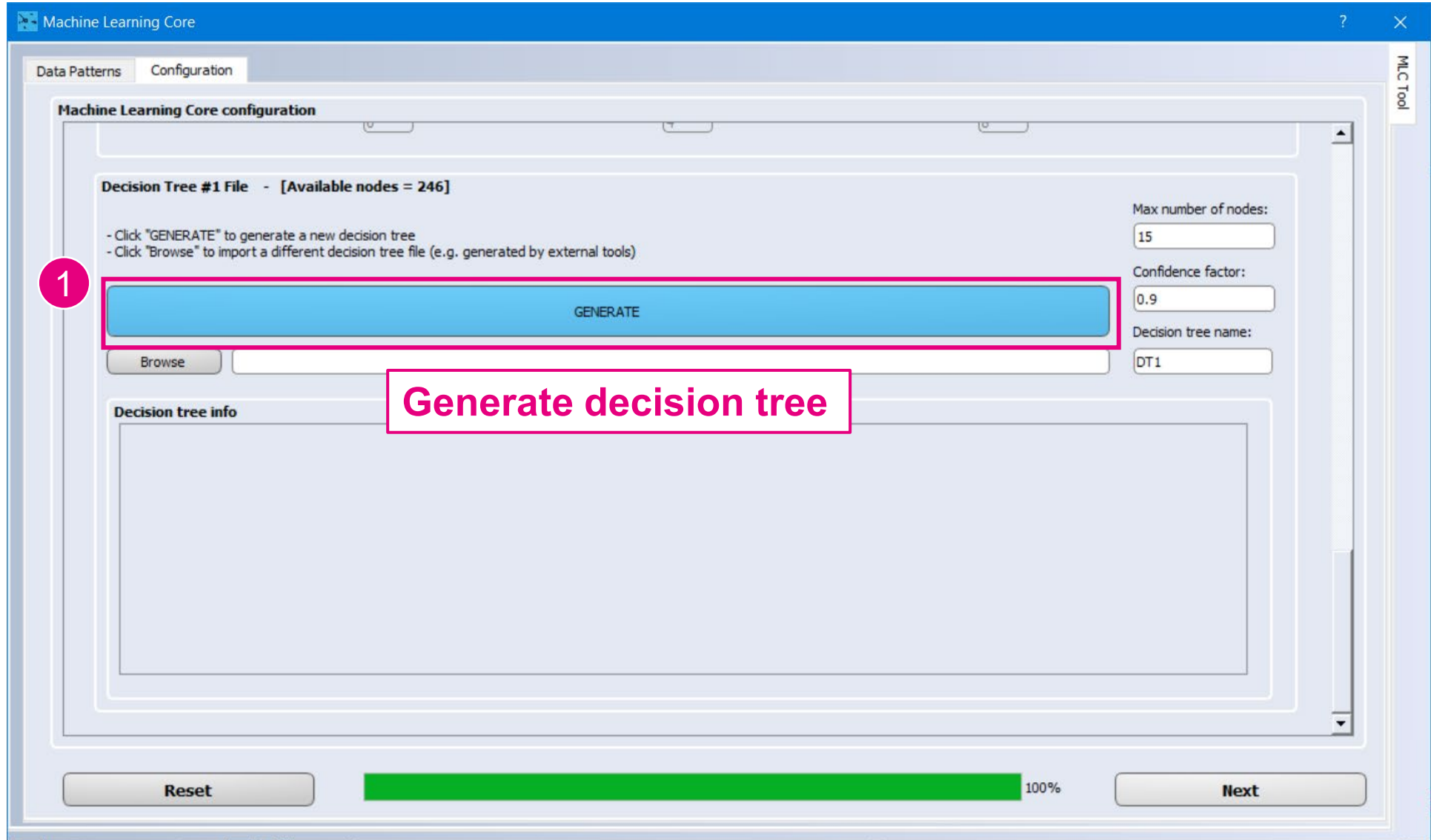
Reset      100%      Next

1 Set output code for each class

2 Press 'Next' button



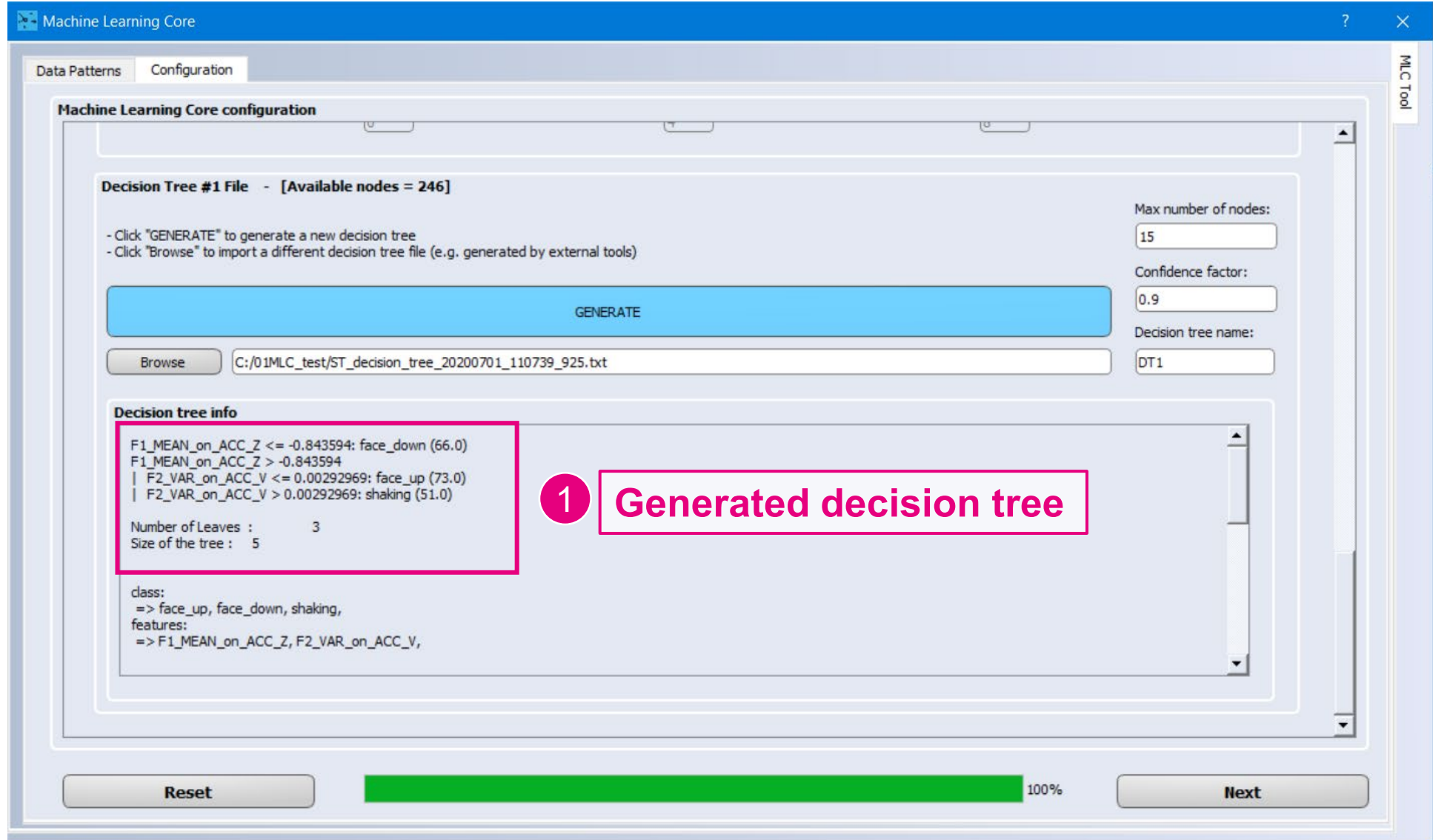
# STEP 4: MLC settings and decision tree creation



The screenshot shows the 'Machine Learning Core configuration' window. The 'Configuration' tab is active. Under 'Machine Learning Core configuration', there is a section for 'Decision Tree #1 File - [Available nodes = 246]'. It includes instructions: '- Click "GENERATE" to generate a new decision tree' and '- Click "Browse" to import a different decision tree file (e.g. generated by external tools)'. A red circle with the number '1' highlights the 'GENERATE' button. A red callout box with the text 'Generate decision tree' points to the 'GENERATE' button. To the right of the instructions are input fields for 'Max number of nodes' (15), 'Confidence factor' (0.9), and 'Decision tree name' (DT1). At the bottom, there is a 'Reset' button, a green progress bar at 100%, and a 'Next' button. The window title is 'Machine Learning Core' and the tab is 'MLC Tool'.



# STEP 4: MLC settings and decision tree creation



The screenshot shows the 'Machine Learning Core configuration' window. The 'Configuration' tab is active. Under 'Machine Learning Core configuration', there is a section for 'Decision Tree #1 File' with a 'Max number of nodes' set to 15 and a 'Confidence factor' of 0.9. A 'GENERATE' button is visible. Below it, a 'Browse' button and a text field containing the file path 'C:/01MLC\_test/ST\_decision\_tree\_20200701\_110739\_925.txt' are shown. The 'Decision tree name' is set to 'DT1'. A 'Decision tree info' section is highlighted with a pink box, containing the following text:

```

F1_MEAN_on_ACC_Z <= -0.843594: face_down (66.0)
F1_MEAN_on_ACC_Z > -0.843594
| F2_VAR_on_ACC_V <= 0.00292969: face_up (73.0)
| F2_VAR_on_ACC_V > 0.00292969: shaking (51.0)

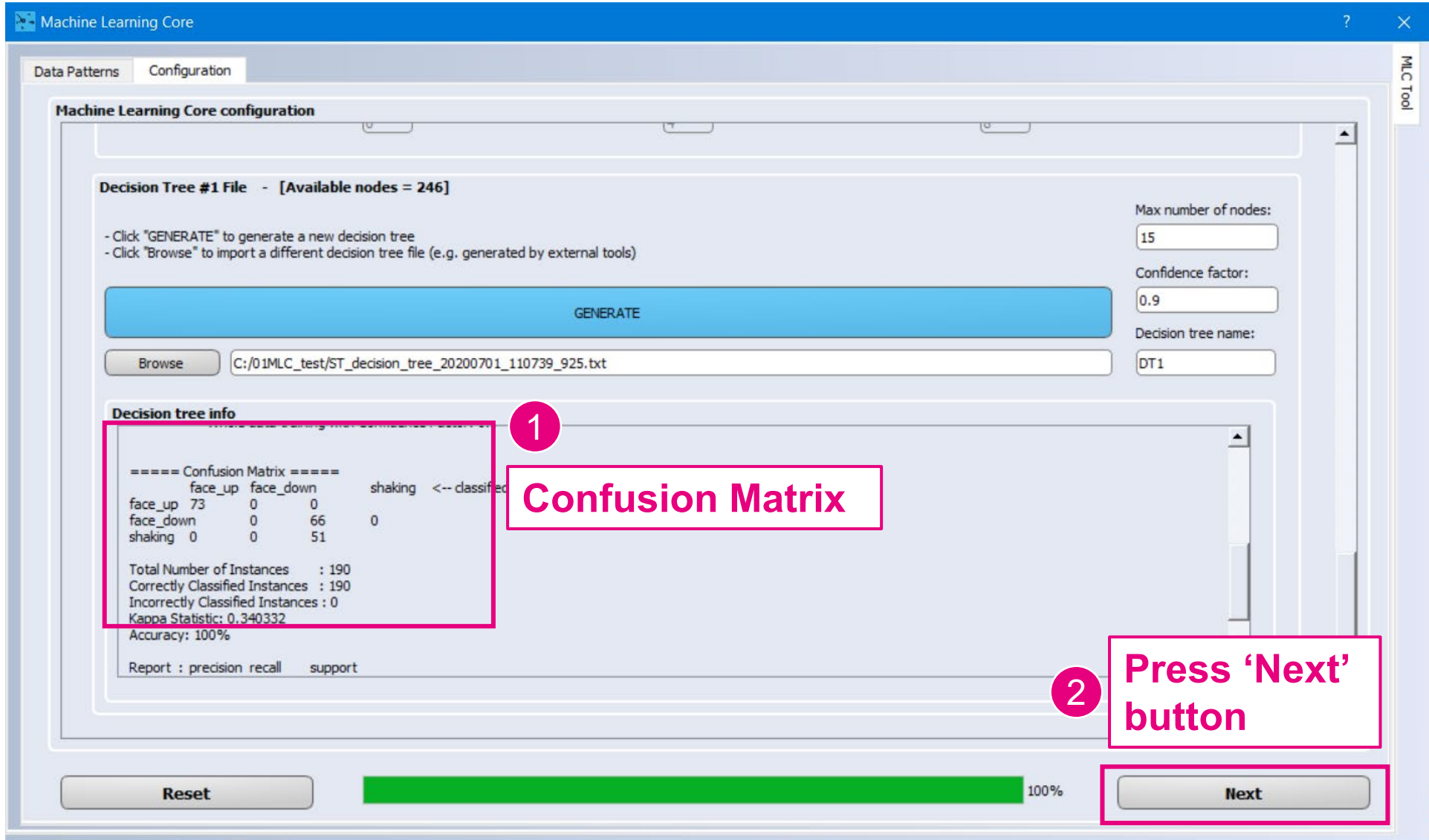
Number of Leaves :      3
Size of the tree :     5

class:
=> face_up, face_down, shaking,
features:
=> F1_MEAN_on_ACC_Z, F2_VAR_on_ACC_V,
    
```

A pink callout box with the number '1' and the text 'Generated decision tree' points to the decision tree info section. At the bottom of the window, there is a 'Reset' button, a green progress bar at 100%, and a 'Next' button.



# STEP 4: MLC settings and decision tree creation



**Machine Learning Core configuration**

**Decision Tree #1 File - [Available nodes = 246]**

- Click "GENERATE" to generate a new decision tree  
 - Click "Browse" to import a different decision tree file (e.g. generated by external tools)

Max number of nodes: 15  
 Confidence factor: 0.9  
 Decision tree name: DT1

GENERATE

Browse C:/01MLC\_test/ST\_decision\_tree\_20200701\_110739\_925.txt

**Decision tree info**

```

===== Confusion Matrix =====
      face_up  face_down  shaking  <-- classified
face_up  73      0      0
face_down  0      66      0
shaking  0      0      51

Total Number of Instances : 190
Correctly Classified Instances : 190
Incorrectly Classified Instances : 0
Kappa Statistic: 0.340332
Accuracy: 100%

Report : precision recall  support
  
```

1 **Confusion Matrix**

2 **Press 'Next' button**

Reset  100% Next



# STEP 4: MLC settings and decision tree creation

The screenshot shows the 'Machine Learning Core configuration' window. It includes a 'GENERATE' button, a file path for the decision tree, and a 'Decision tree info' section with a confusion matrix and classification statistics. A 'Meta-classifier' section is highlighted with a pink box, containing three input fields for end counter values. A 'Next' button is also highlighted with a pink box and a callout bubble.

**Machine Learning Core configuration**

GENERATE

Browse C:/01MLC\_test/ST\_decision\_tree\_20200608\_102941\_608.txt

Decision tree name: DT1

0.9

**Decision tree info**

```
==== Confusion Matrix =====
      Stationa Idle Writing <-- classified as
Stationa 24    0    0
Idle      0   31    0
Writing  0    0   26

Total Number of Instances : 81
Correctly Classified Instances : 81
Incorrectly Classified Instances : 0
Kappa Statistic: 0.337296
Accuracy: 100%

Report : precision recall support
Stationa 1    1    24
Idle      1    1    31
```

**Meta-classifier**  
Insert the end counter values for the decision tree (allowed values from 0 to 14):

#1: Stationary

#2: Idle

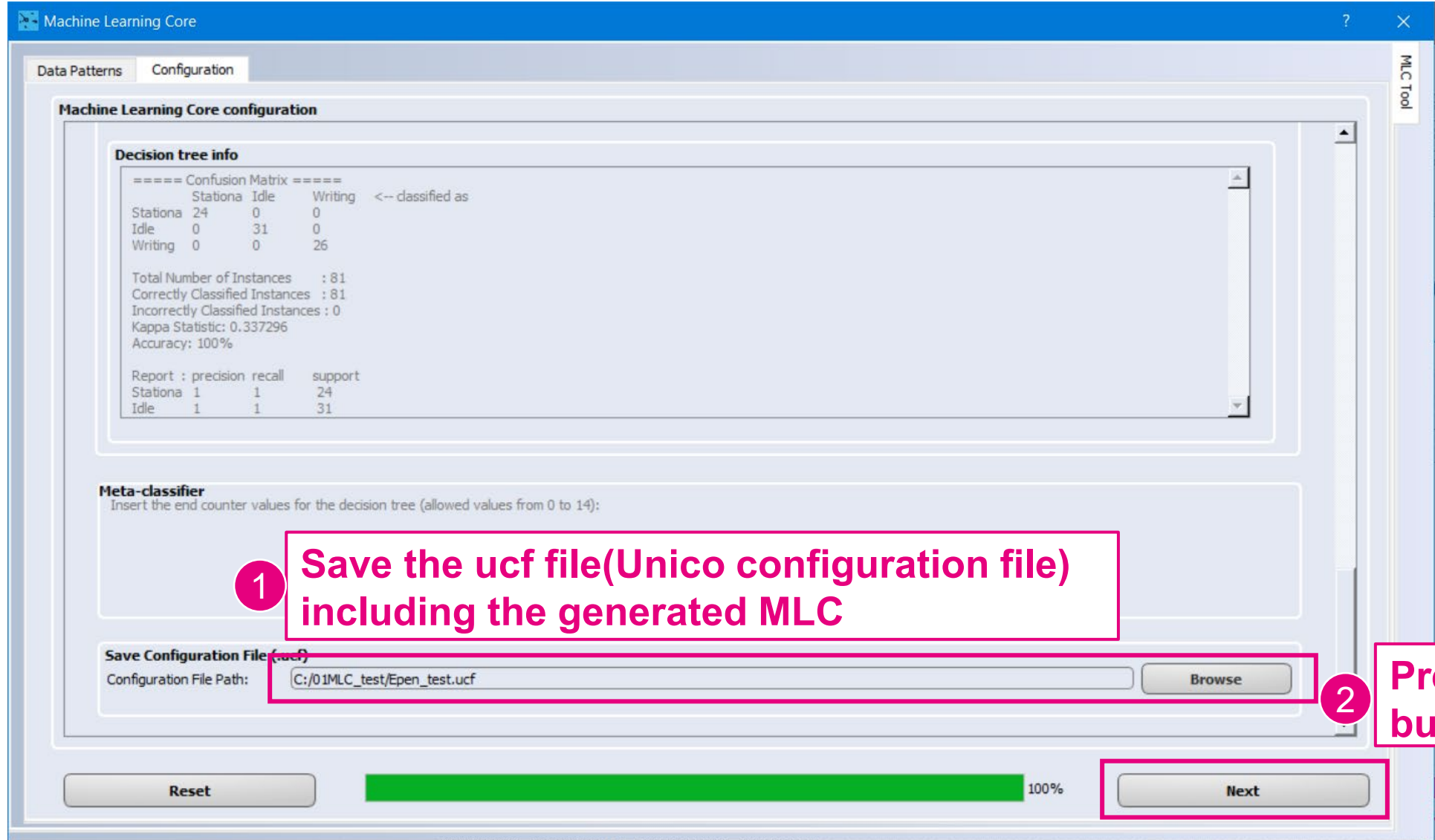
#3: Writing

Reset  100% **Next**

1 Press 'Next' button



# STEP 4: MLC settings and decision tree creation



**Machine Learning Core configuration**

**Decision tree info**

```

===== Confusion Matrix =====
      Stationa Idle   Writing  <-- classified as
Stationa 24     0     0
Idle     0     31     0
Writing  0     0     26

Total Number of Instances   : 81
Correctly Classified Instances : 81
Incorrectly Classified Instances : 0
Kappa Statistic: 0.337296
Accuracy: 100%

Report : precision recall   support
Stationa 1     1     24
Idle     1     1     31
  
```

**Meta-classifier**  
Insert the end counter values for the decision tree (allowed values from 0 to 14):

**Save Configuration File (.ucf)**  
Configuration File Path:

100%

1 Save the ucf file(Unico configuration file) including the generated MLC

2 Press 'Next' button



# STEP 4: MLC settings and decision tree creation

Machine Learning Core configuration

**Decision tree info**

```
==== Confusion Matrix ====
      Stationa Idle  Writing  <-- classified as
Stationa 24    0    0
Idle      0    31    0
Writing  0    0    26

Total Number of Instances : 81
Correctly Classified Instances : 81
Incorrectly Classified Instances : 0
Kappa Statistic: 0.337296
Accuracy: 100%

Report : precision recall support
Stationa 1 1 24
Idle 1 1 31
```

**Meta-classifier**  
Insert the end counter values for the decision tree (allowed values are 0-255)

#1: Stationary  #2: Idle  #3: Writing

**Save Configuration File (.ucf)**  
Configuration File Path:

100%

MLC setting is completed!



Operation  
check



# STEP 5: Check the operation of the created MLC

The screenshot shows the Unico software interface. The top bar includes a 'Select Port' dropdown set to 'COM4', and buttons for 'Connect', 'Disconnect', 'Start', 'Stop', and 'Exit'. The main area is divided into several tabs: 'Info', 'Options', 'Registers', 'Registers 2', 'Embedded Reg 1', 'Embedded Reg 2', 'Embedded Adv Page 0', 'Embedded Adv Page 1', 'Sensor Hub Registers', and 'Load/Save'. The 'Load/Save' tab is active, showing two sections: 'Save Data' and 'Load/Save Configuration'. The 'Save Data' section has a text input field for a file name, a 'Browse...' button, and several checkboxes for data to save: Accelerometer (LSB, Acceleration, Angle, Interrupt), Gyroscope (LSB, Angular Rate), and Temperature (LSB, Temperature [C], DecisionTree Results). There are 'Start' and 'Stop' buttons for logging. The 'Load/Save Configuration' section has instructions and 'Load' and 'Save' buttons. A red box highlights the 'Load' button with the text '1 Press 'Load' button'. The bottom status bar shows: 'STMicroelectronics', 'Demonstration kit = STEVAL-MKI197V1 (LSM6DSOX)', 'Board = ProfIMEMSTool', 'Firmware Version = V3.6.26', and 'Unico Version = 9.8.1.0'.



# STEP 5: Check the operation of the created MLC

1 Select the created .ucf file

2 Press 'Open' button



# STEP 5: Check the operation of the created MLC

The screenshot shows the Unico software interface. At the top, there is a toolbar with buttons for 'Back', 'Select Port: COM4', 'Connect', 'Disconnect', 'Start', 'Stop', and 'Exit'. The 'Start' button is highlighted with a pink box and a circled '1', with a callout box that says 'Press 'Start' button'. On the left sidebar, the 'Data' icon is highlighted with a pink box and a circled '2', with a callout box that says 'Open 'Data' window'. The main window displays the 'Save Data' section, which includes a 'Browse...' button and several checkboxes for data to save: Accelerometer (LSB, Acceleration, Angle, Interrupt), Gyroscope (LSB, Angular Rate), Temperature (LSB, Temperature [C]), and DecisionTree Results. Below this are 'Start' and 'Stop' buttons. The 'Load/Save Configuration' section has 'Load' and 'Save' buttons. The status bar at the bottom shows: STMicroelectronics, Demonstration kit = STEVAL-MKI197V1 (LSM6DSOX), Board = ProfMEMSTool, Firmware Version = V3.6.26, and Unico Version = 9.8.1.0.

# STEP 5: Check the operation of the created MLC

The screenshot shows the Unico software interface with the following data:

Accelerometer	X	Y	Z	Unit
Data read from the Accelerometer	-4899	12729	8443	mg
	0	0	0	dps
	0	0	0	dps
	0	0	0	dps

**Decision Tree results:**

1.	4	2.	0	3.	0	4.	0	5.	0	6.	0	7.	0
----	---	----	---	----	---	----	---	----	---	----	---	----	---

**Decision tree output**  
 0: Face-up  
 4: Face-down  
 8: Shaking

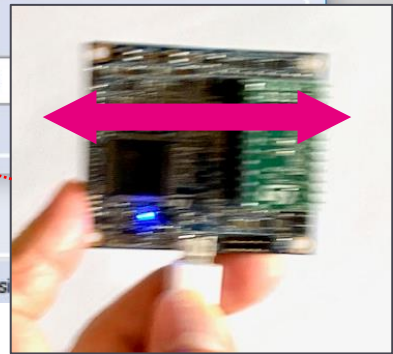
1



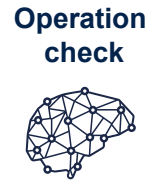
State #1: Face-up



State #2: Face-down



State #3: Shaking



# STEP 5: Check the operation of the created MLC

Unico

Back | Select Port: COM4 | Connect | Disconnect | Start | Stop | Exit

Info | Options | Registers | Registers 2 | Embedded Reg 1 | Embedded Reg 2 | Embedded Adv Page 0 | Embedded Adv Page 1 | Sensor Hub Registers | Load/Save

**Save Data**

- Choose the file name:  **Browse...**

- Check the data you want to save: *(Accelerometer)*  LSB  Acceleration  Angle  Interrupt  
*(Gyroscope)*  LSB  Angular Rate  
 Temperature LSB  Temperature [C]  DecisionTree Results

- Press Start for logging: **Start**

**Load/Save Configuration**

- Click Load for loading a configuration from a text file: *Loaded* **Load**

- Click Save for saving the current configuration to a text file: **Save**

1 **MLC result logging is also possible**

STMicroelectronics | Demonstration kit = STEVAL-MKI197V1 (LSM6DSOX) | Board = ProfiMEMSTool | Firmware Version = V3.6.26 | Unico Version = 9.8.1.0

# Q&A



# Thank you

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to [www.st.com/trademarks](http://www.st.com/trademarks).

All other product or service names are the property of their respective owners.



life.augmented