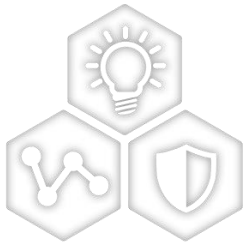


AIML1001



A Leading Provider of Smart, Connected and Secure Embedded Control Solutions



SMART | CONNECTED | SECURE

Ruby Chen

2024/9/26

Class Objectives

- **Before this class:**
 - Basic C programming language design skills
- **When you finish this course, you will be able to:**
 - The basic information to implement AI/ML at the edge.
 - Be familiar with MPLAB Machine Learning Model Builder.
 - Enable to configure and deploy model on the small processors.
- **You won't be able to learn:**
 - Making you an Artificial Intelligence/ML expert.
 - Build a Large language model.

Agenda

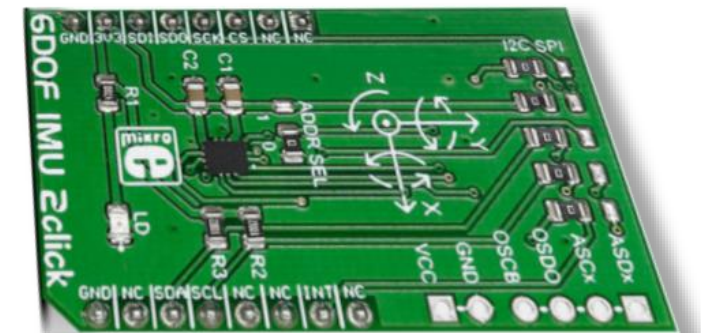
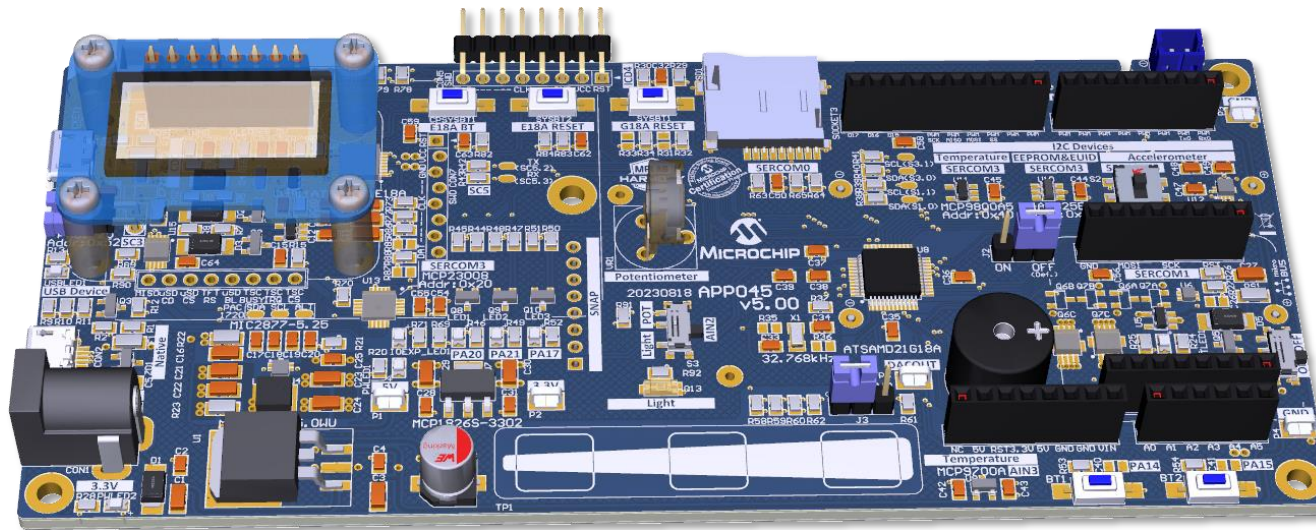
- Artificial Intelligence and Machine Learning
- Introduction of Edge AI
 - Lab0: Initial Setup
- Data Collection
 - MPLAB® Data Visualizer
 - Lab1: Collect Data by MPLAB Data Visualizer
- Organize: label and prepare data
 - MPLAB® Machine Learning Development Suite
 - Lab2: Organize: Label and Prepare data
- Analyze the data & Introduce ML model
 - Lab3: Analyze the data & Build and Test model
- Explore & Test model
 - Lab4: Explore & test model
- Deploy the model
 - Lab5: Export & Deploy the Model
- Large Language Models

Software Prerequisite and Installation

- **MPLAB® X IDE v6.20** <https://www.microchip.com/mplab/mplab-x-ide>
- **MPLAB® XC32 v4.35** <https://www.microchip.com/mplab/compilers>
- **SAMD21 DFP v3.6.144** (Device Family Pack download from X IDE Tools Packs)
- **CMSIS v5.8.0** (Common Microcontroller Software Interface)
- [Install MPLAB® ML Model Builder v1.1.0](#)
- [Install MPLAB® ML Data Collector v1.2](#)
- **Apply “MyMicrochip” account:**
<https://www.microchip.com/myMicrochip/#/>
- **TeraTerm**

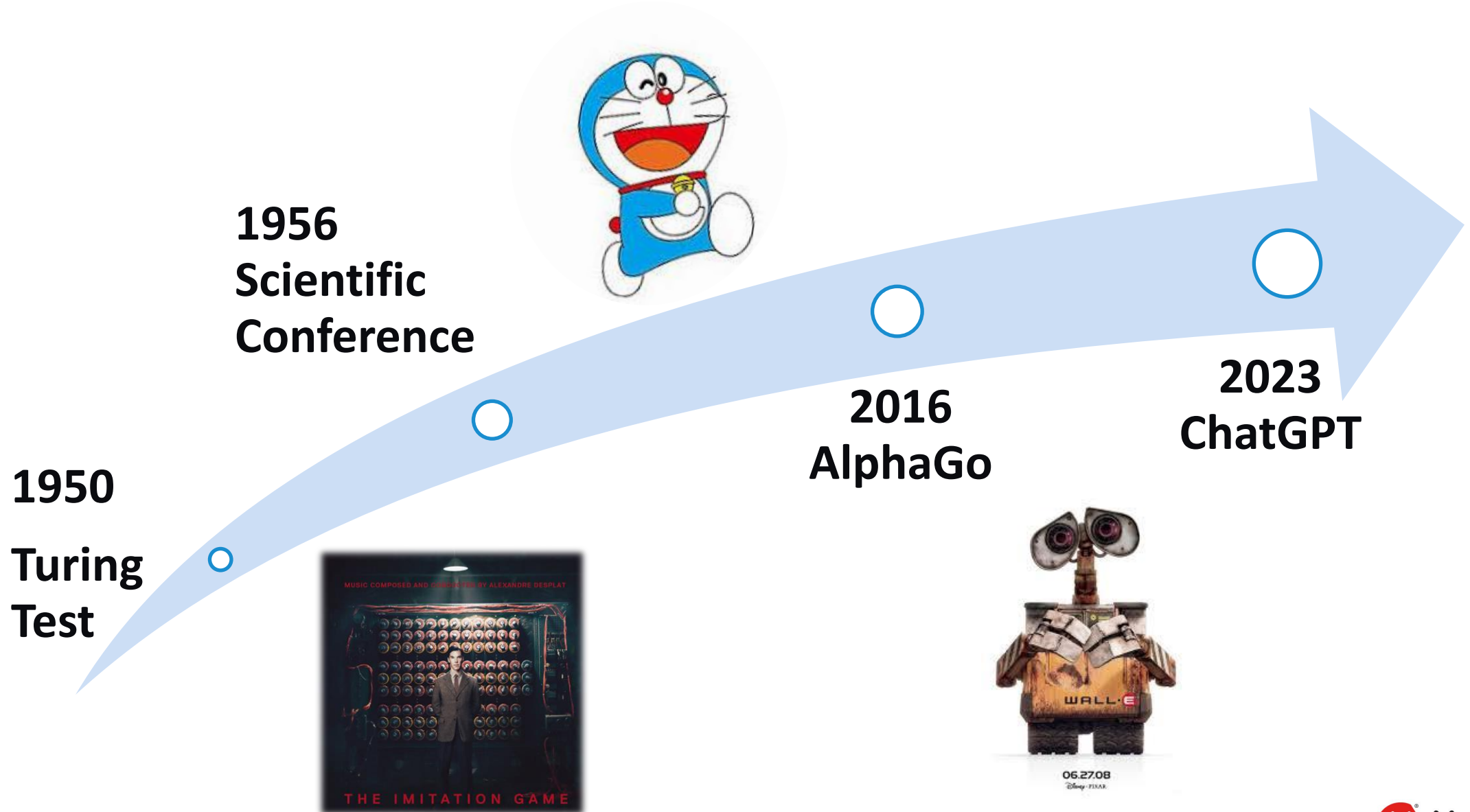
Hardware Prerequisite

- APP045 EV Board with SAMD21
- 6DOF IMU 2 Click board
- Micro USB transmission cable
- Debugger



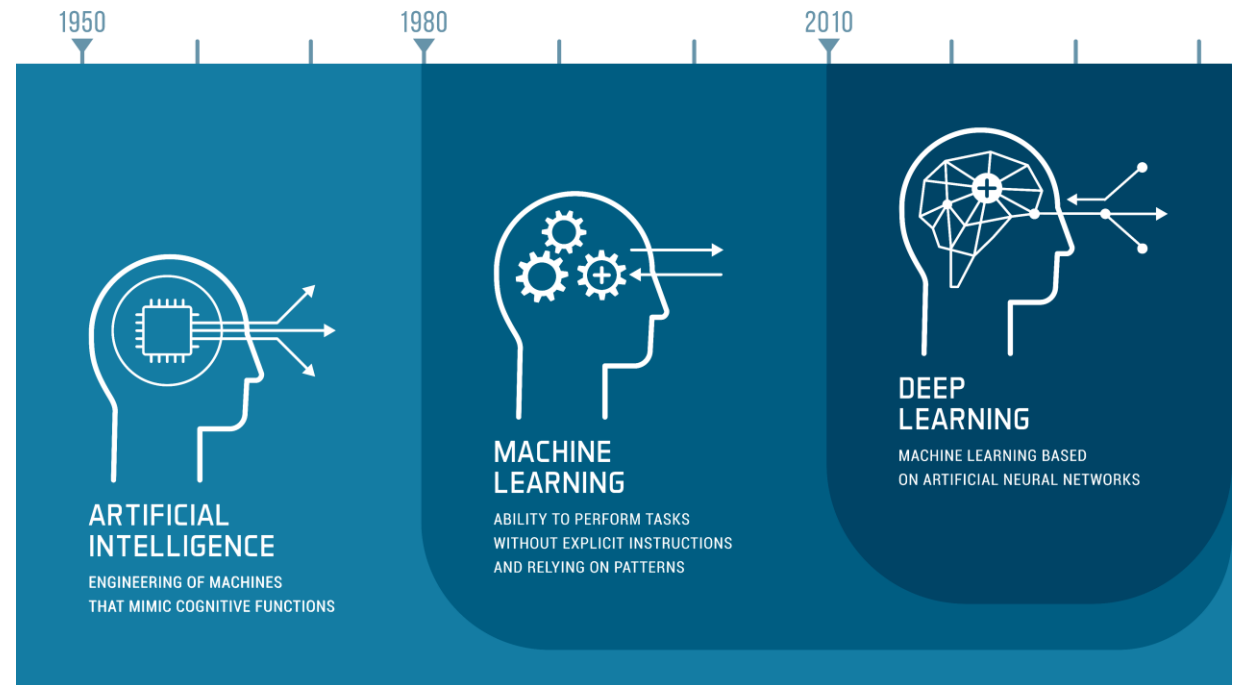
Introduction of Machine Learning

Artificial Intelligence and Machine Learning



AI, ML and DL

- **AI (Artificial Intelligence):** technology that enables computers and machines to simulate human intelligence and problem-solving capabilities.
- **ML (Machine Learning):** a type of artificial intelligence (AI) focused on building computer systems that learn from data.
- **DL (Deep Learning):** a subset of machine learning that uses multi-layered neural networks, called deep neural networks, to simulate the complex decision-making power of the human brain.



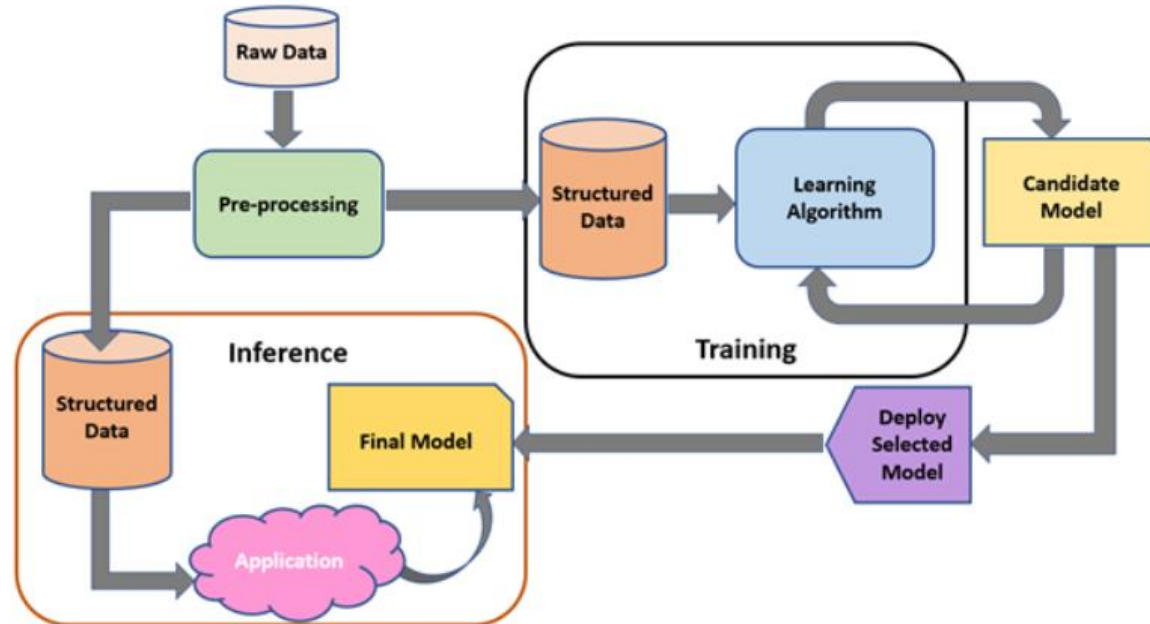
Why is it so popular now?

- **Advancements in computational power**
 - Vastly parallel operations
 - Virtual computing and Data Center
- **Large Datasets**
 - Example data
- **Algorithms**
 - New algorithms
 - Accessibility
- **Automation**
 - 'Intelligence' and automation
 - Sustainability



Machine Learning

- Machine learning (ML) is a technique of using computers to predict things based on past observations.
- A set of algorithmic methods that discovers patterns from seemingly unrelated data, providing you with important information to facilitate decision making.



4 kinds of Machine Learning

Supervised Learning

Give label and data

Give the standard answer while training.

Un-supervised Learning

Unlabeled data

Provide examples. Let computer find the underlying rules in the examples.

Semi-supervised Learning

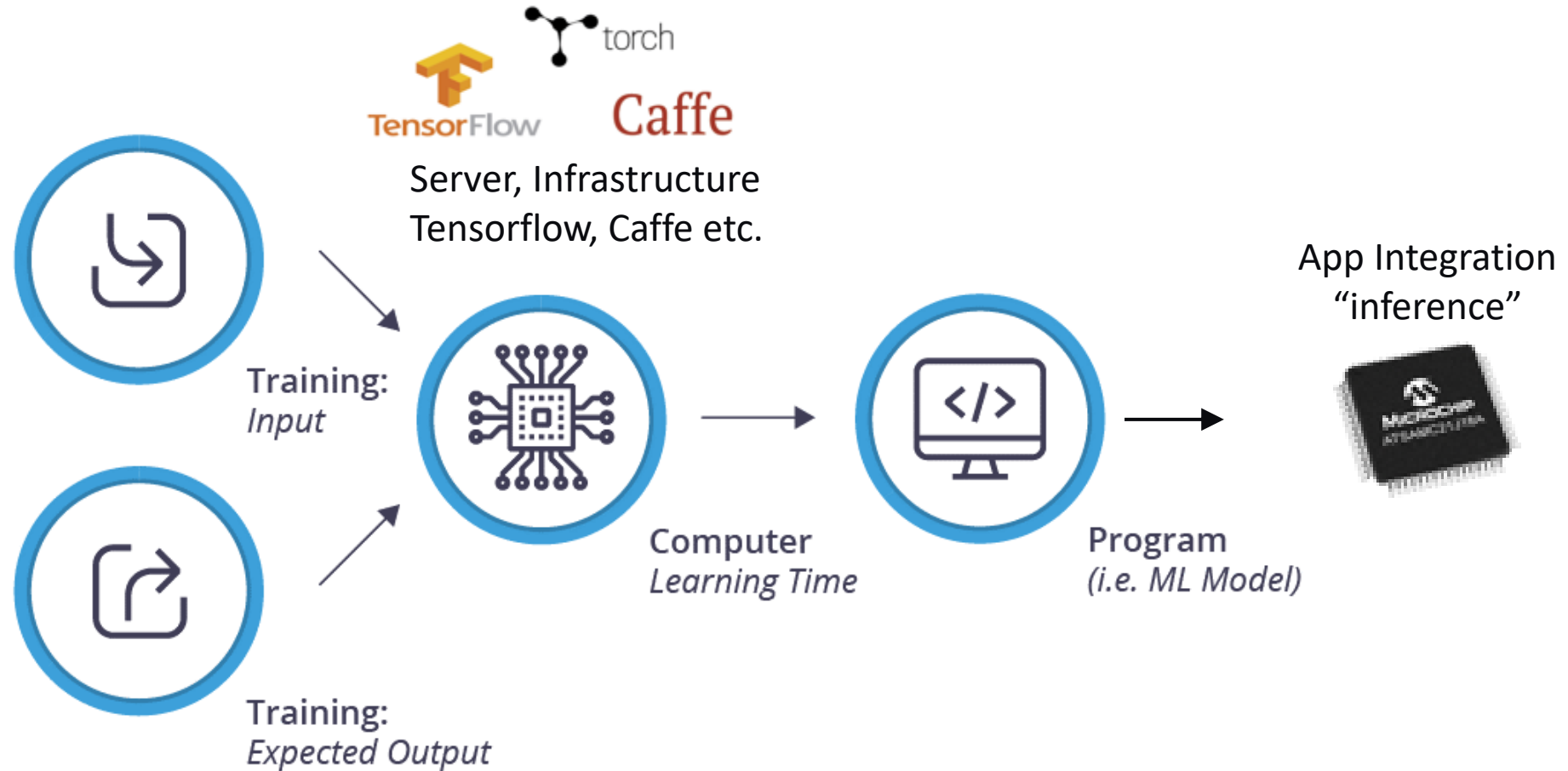
Small amount of labeled data
+
Large amount of unlabeled data

Reinforcement Learning

Unlabeled data

Tell computer whether the steps are correct, the machine will modify by environment and experience.

How does Supervised Learning work?



Edge AI

Edge Computing vs. Cloud Computing

- **Cloud Computing:** Transfer the data to the server on the Internet, and use the high-performance advantages of the server for calculation and processing.
- **Edge Computing:** The data is calculated directly on the local side. It can overcome the problem of data transmission delay and facilitate instant judgment and decision-making.



Why AI at the Edge?

- **Edge AI:** Execute the application of artificial intelligence algorithms on edge computing nodes such as embedded devices and sensors.
- **Edge is the location of Data Processing near the source of Data Generation**



Privacy and Security

Data stays on the device, gets processed locally.



Reduce Power Consumption

Stay operational for longer periods of time.



Real Time Operation

Process data real-time on the edge device, without having to wait for a response back from the cloud.

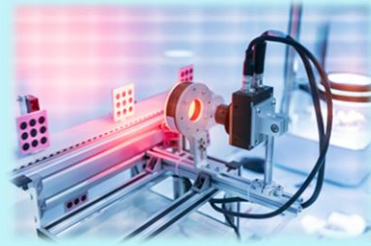


Reduce Cost

Save on compute costs, storage, and less communication.

ML Edge Applications

Smart Embedded Vision



Enhanced quality and productivity at factories

Machine assistance for faster, better



Surveillance / security with granular, real-time monitoring and response

Smart Human Machine Interface (HMI)



Local processing for voice interactive appliances



Real time, interactive gesture recognition

Smart Predictive Maintenance



Detect ageing / environment driven degradation

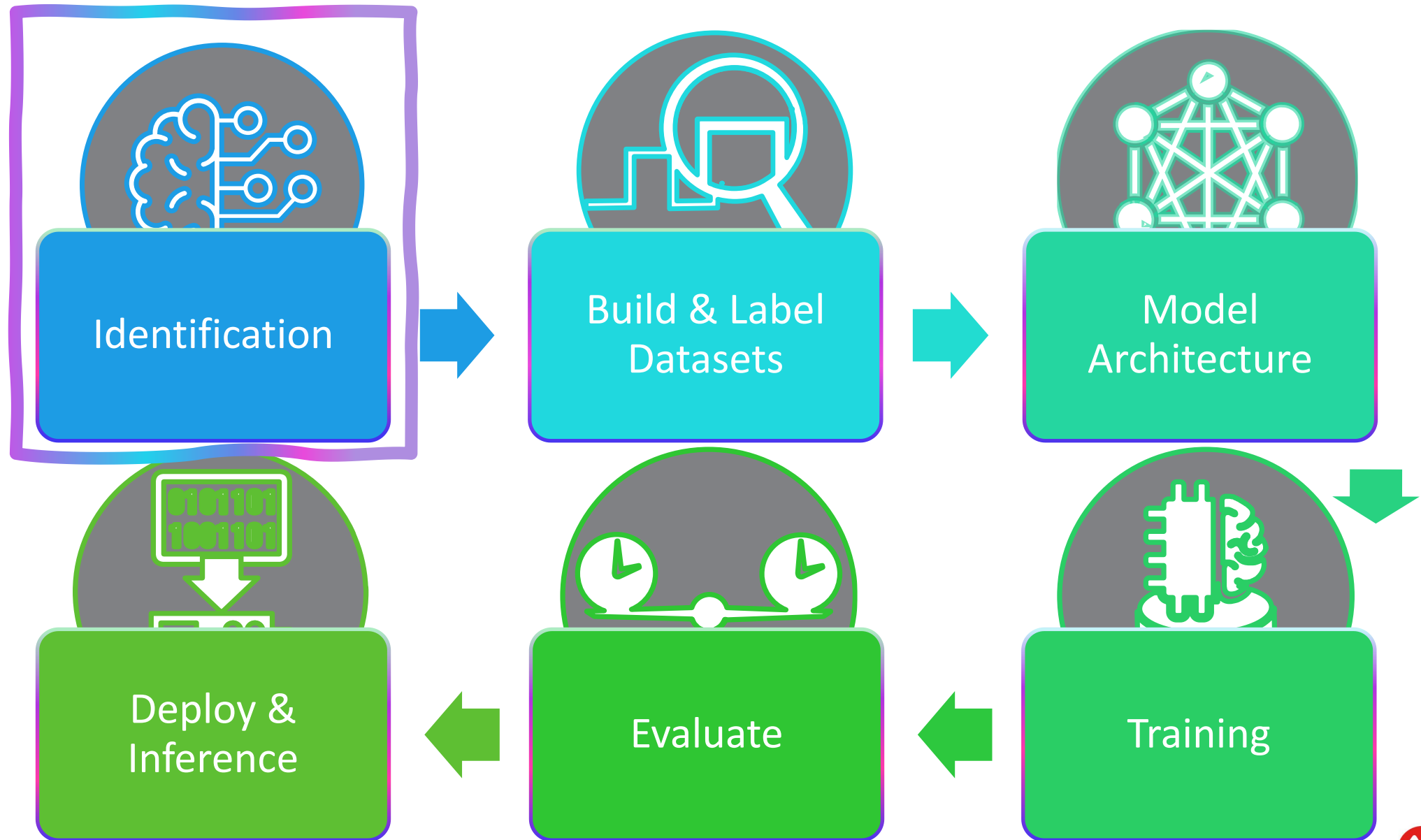


Predict and prevent system failures



Water / hazardous leak detection and alarm

Machine Learning workflow



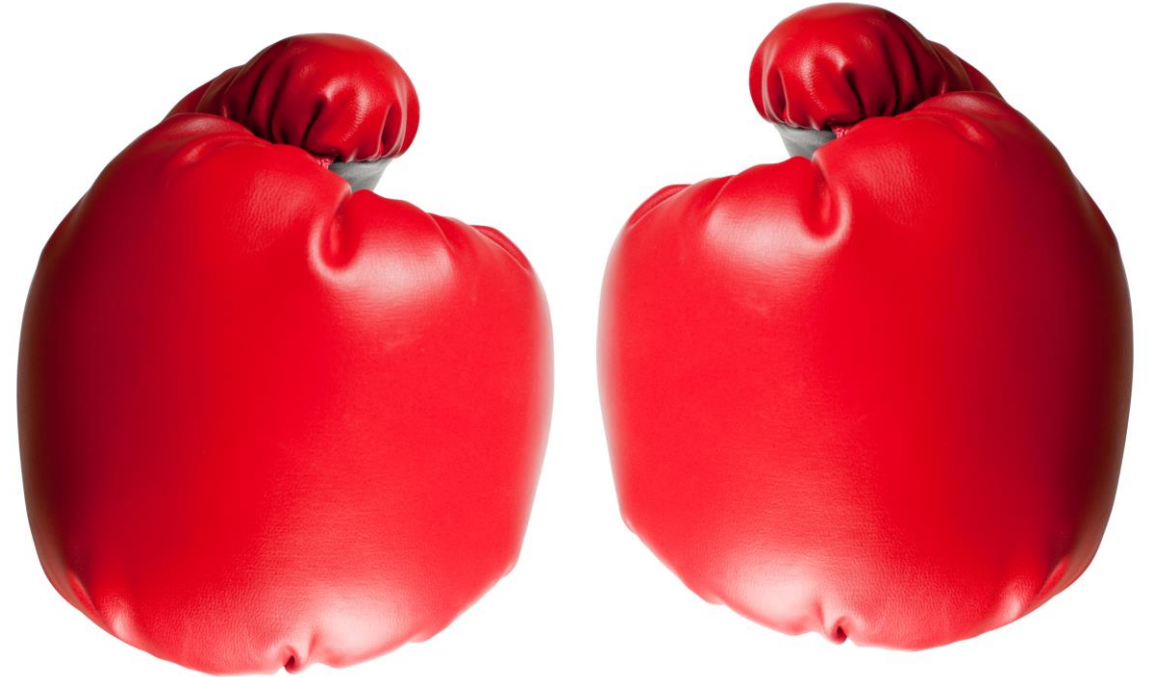
What is the goal of today?

- Build a boxing recognition model and deploy it on the APP045 development board with 6-axis G-sensor.
- The model should recognize 4 different actions, and which is:
 - Straight
 - Righthook
 - Uppercut
 - Freeze



Problem Identification

- **Detect boxing action posture**
 - Straight, Uppercut, Right Hook, Freeze.
- **Traditional approach(es)**
 - Camera, Computer vision
 - Dynamic system, Hand tracking
- **Let's try Machine Learning**
 - Does it offer any benefits?
 - Development speed?
 - Accuracy?
 - Code size?



Lab 0 Initial Setup

Lab 0 Initial Setup

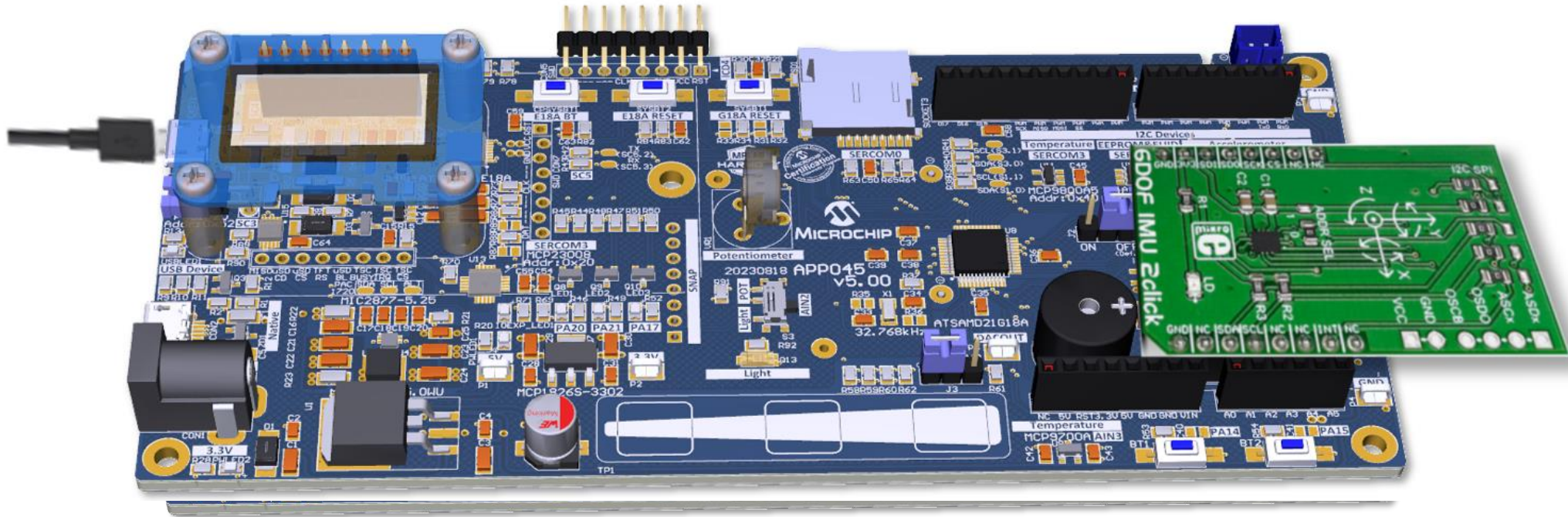
- **Verify the hardware setup and open the project.**
 - APP045 EV Board with SAMD21
 - 6DOF IMU 2 Click board
- **Check the project configuration and inspect the operation of the supplied sampling program.**

Let's go!

Lab 0 Initial Setup

Step 1

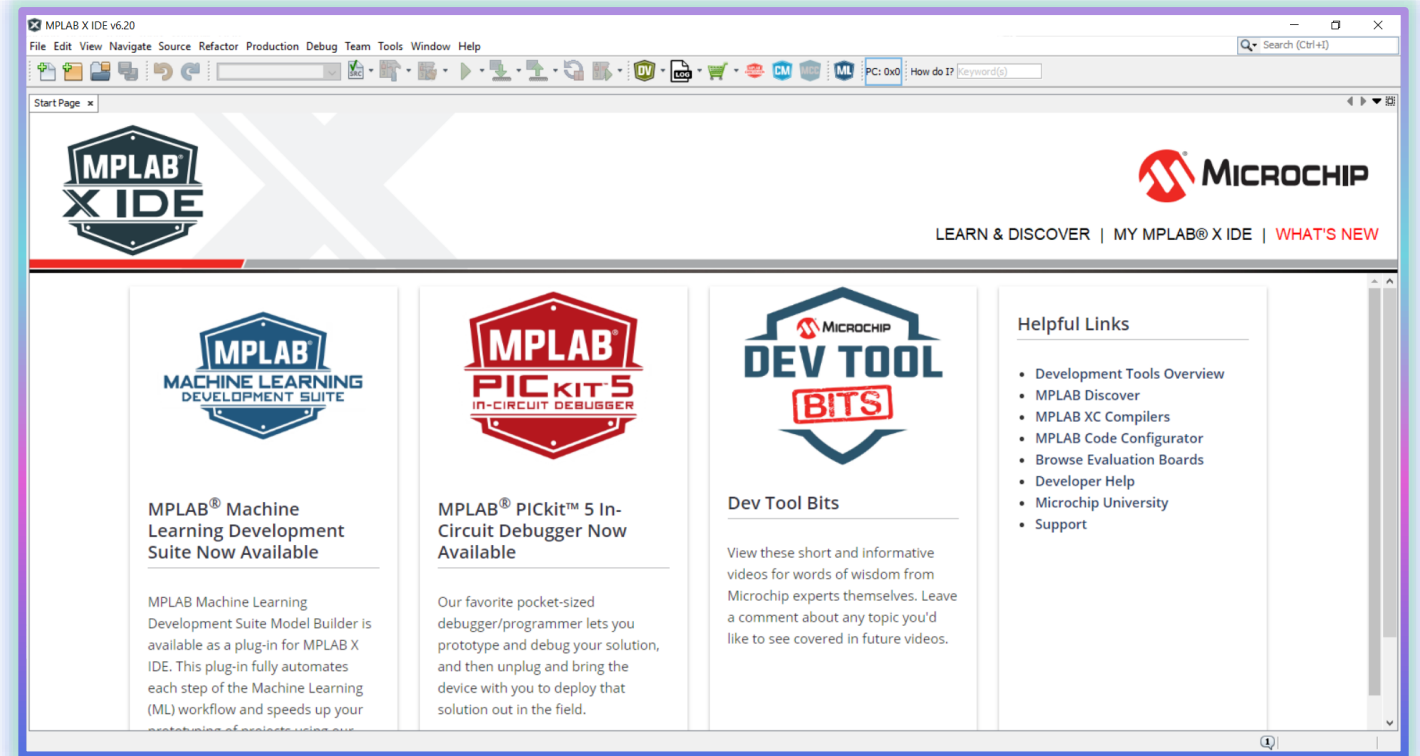
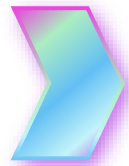
- Assemble BMI160 click board on APP045 with mikroBUS socket.
- Connect the APP045 to the PC.
- Connect Debugger



Lab 0 Initial Setup

Step 2

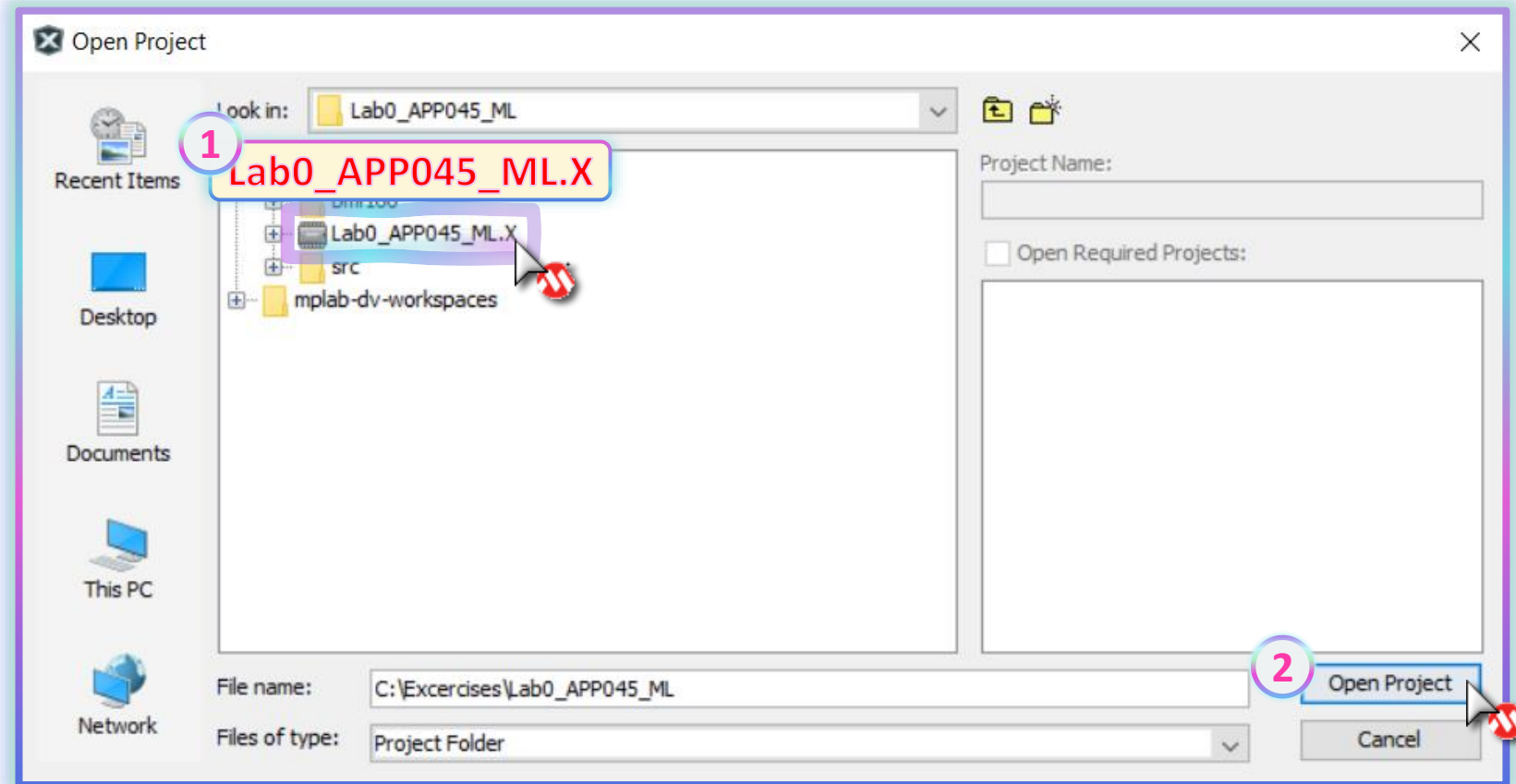
- Open MPLAB X IDE.



Lab 0 Initial Setup

Step 3

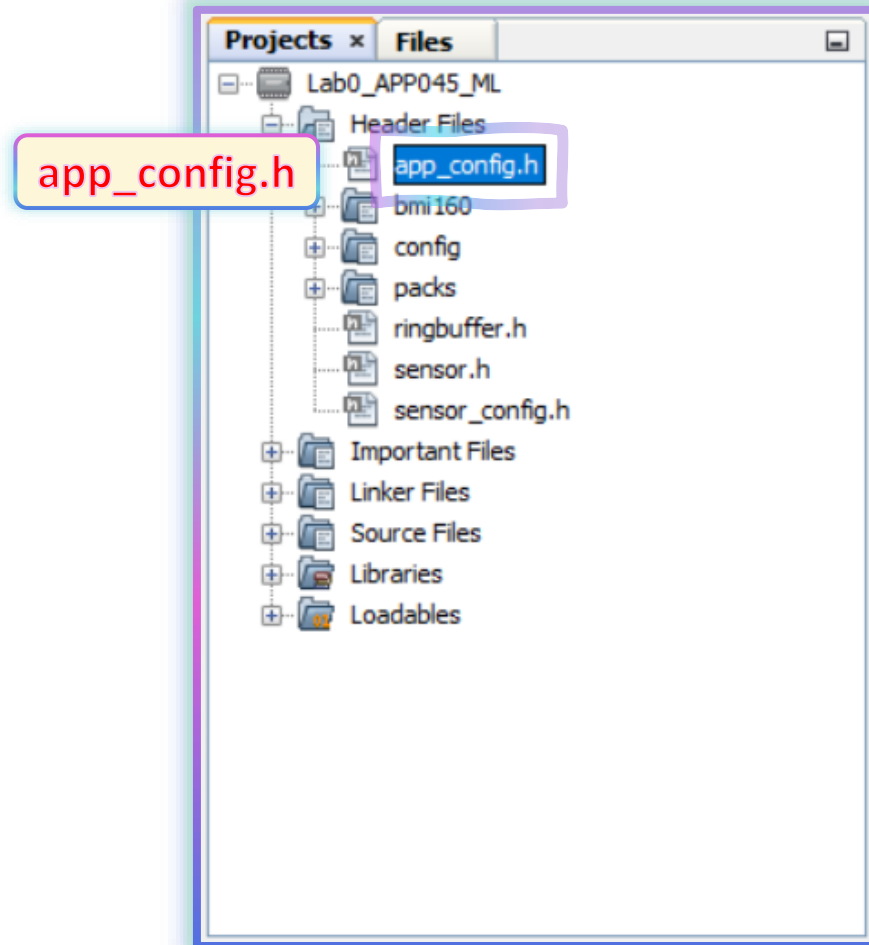
- Click  icon or File -> Open Project
- Open project **Lab0_APP045_ML**



Lab 0 Initial Setup

Step 4

- Projects > Header Files > app_config.h



Lab 0 Initial Setup

Step 5

Make sure the Data Streaming Format is MDV.

- Line 60

```
#define DATA_STREAMER_FORMAT DATA_STREAMER_FORMAT_MDV
```

```
58 // Data streaming formatting selection
59 #ifndef DATA_STREAMER_FORMAT
60 #define DATA_STREAMER_FORMAT DATA_STREAMER_FORMAT_MDV
61 #endif
62
```

Lab 0 Initial Setup

Step 6

- Line 69 Check **Sample Rate** is **100**.

```
#define SNSR_SAMPLE_RATE    100
```

- Line 73 Check the **Accelerometer range** is **16**.

```
#define SNSR_ACCEL_RANGE    16
```

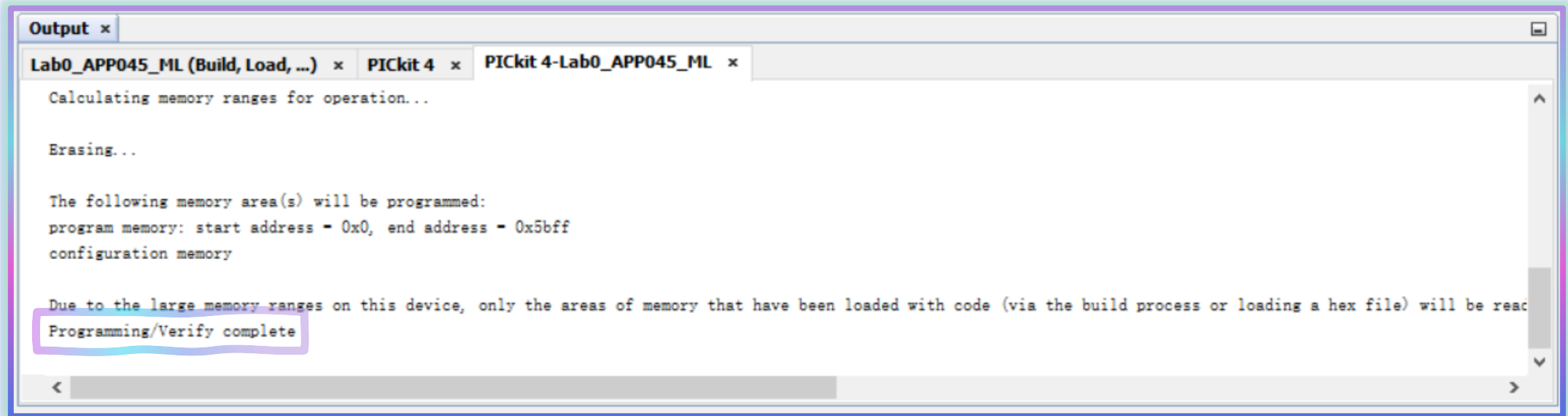
- Line 77 Check the **Gyro range** is **2000**.

```
#define SNSR_GYRO_RANGE    2000
```

Lab 0 Initial Setup

Step 7

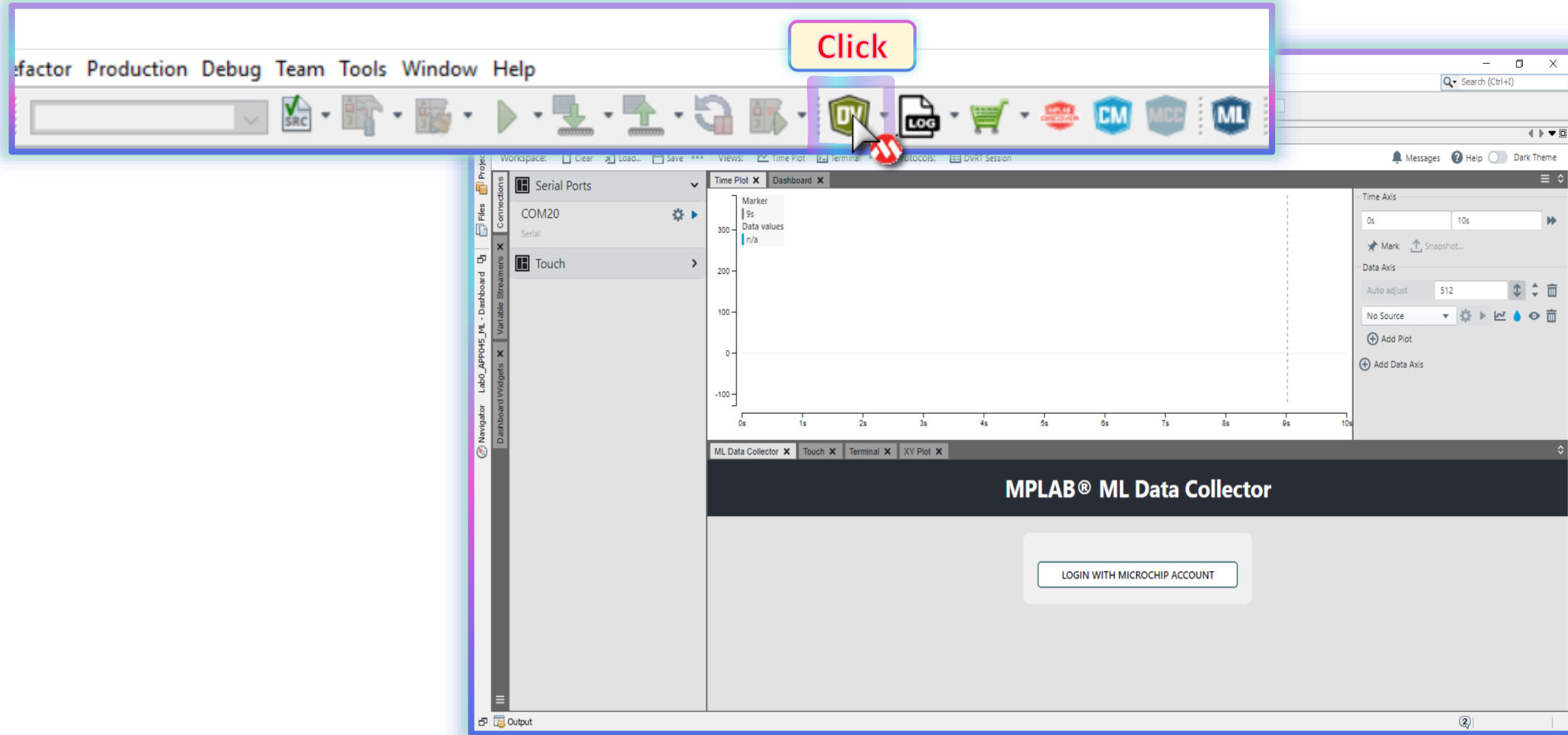
- Clean and Build Project
- Click  icon to Program the Project.
- Output Window will show *Programming/Verify Complete.*



Lab 0 Initial Setup

Step 8

- Open **Data Visualizer**. Click icon 

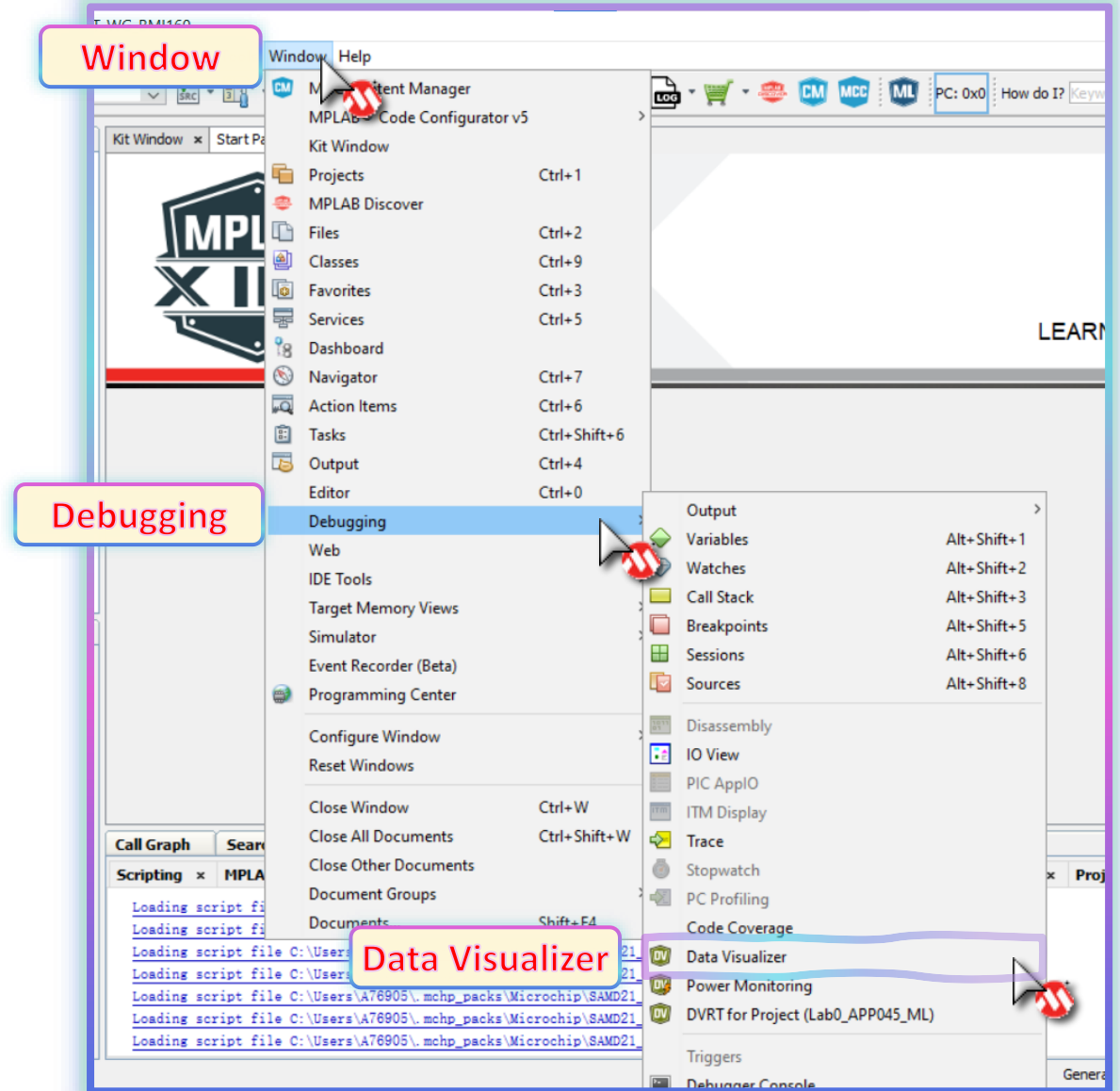


Lab 0 Initial Setup

Step 8

OR

- Open Data Visualizer.
Select
Window ->
Debugging ->
Data Visualizer.

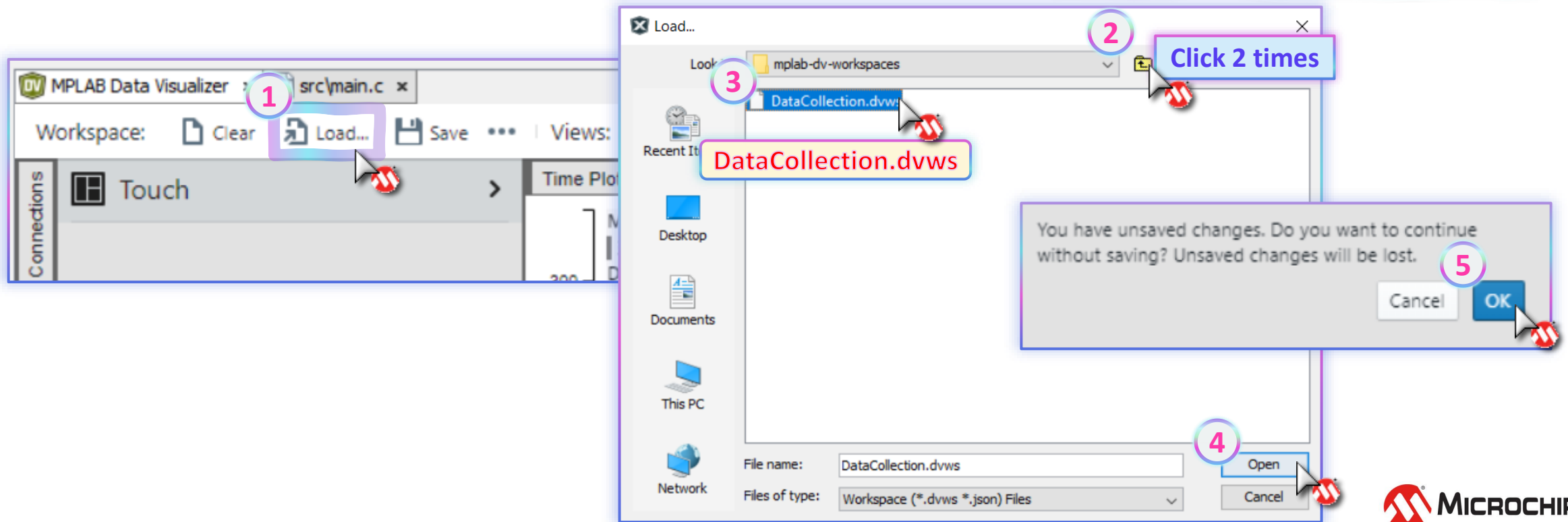


Lab 0 Initial Setup

Step 9

Load the DV workspace from the project.

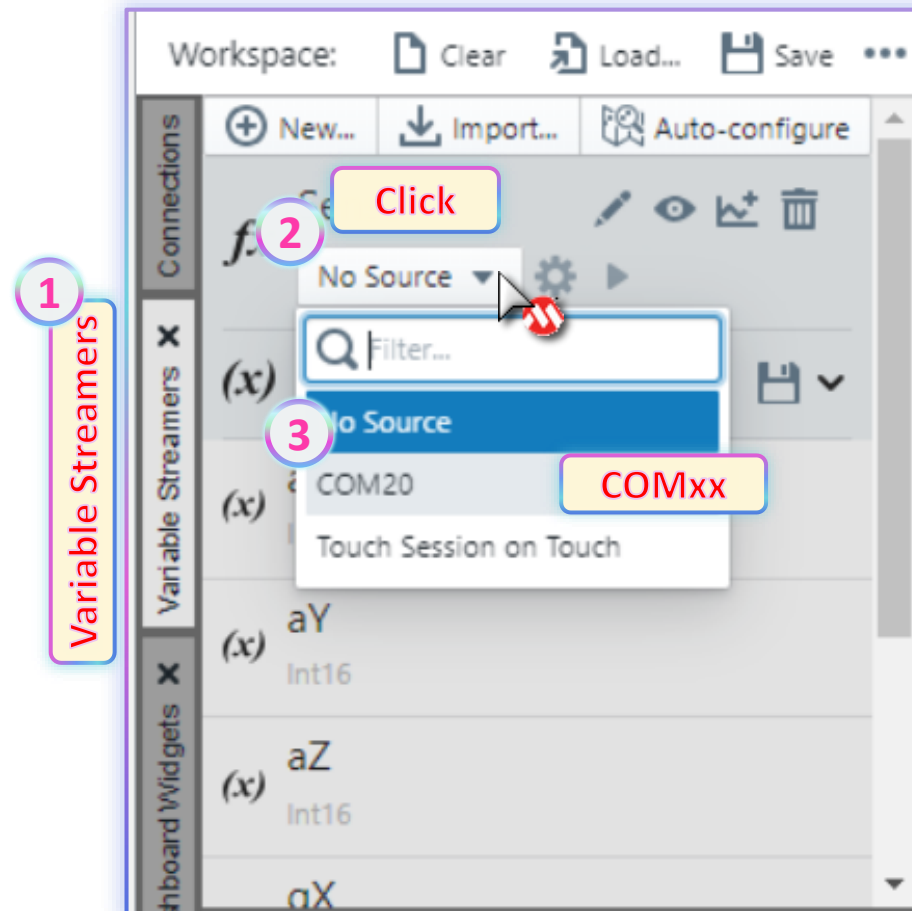
- Click  Load... icon.
- Lab0_APP045_ML/mplab-dv-workspaces/**DataCollection.dvws**



Lab 0 Initial Setup

Step 10

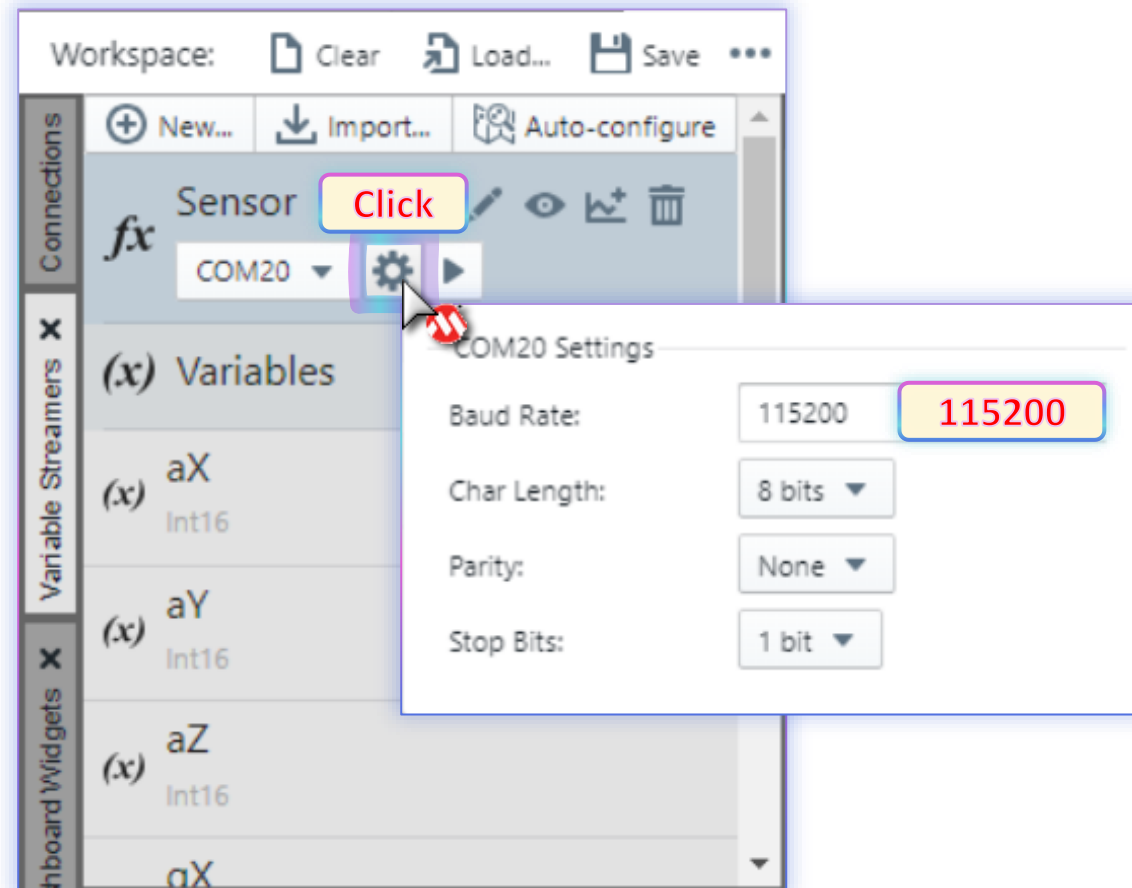
- Variable Streamers > No Source > COMxx



Lab 0 Initial Setup

Step 11

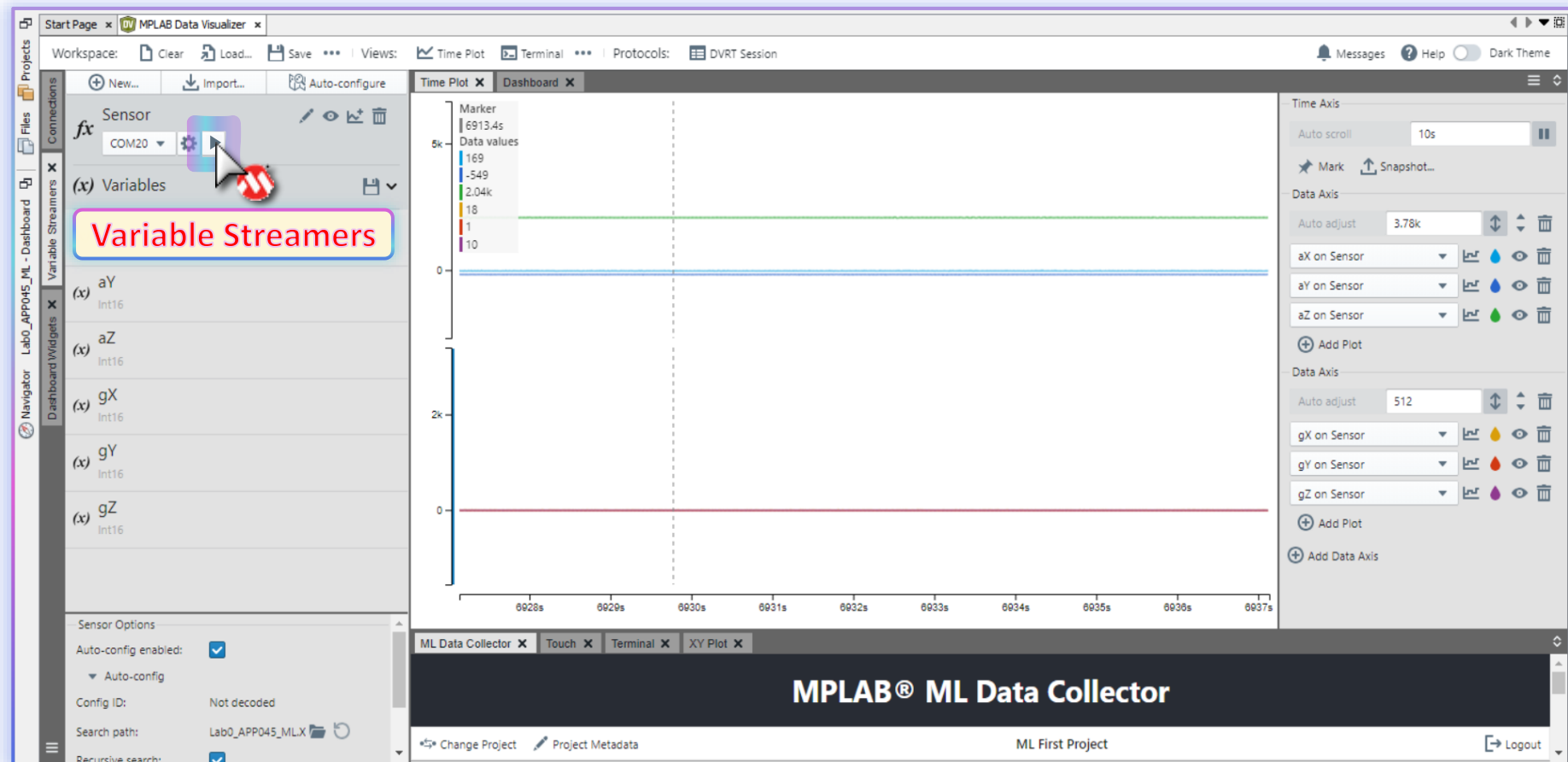
- Sensor> ⚙️ (Source Option) > Baud Rate: 115200



Lab 0 Initial Setup

Step 12

- Start Variable Streamers
- Click ► icon



Lab 0 Initial Setup

Result

- Time Plot Graphic Window will show the raw data.



Time Axis

70677.1s 10s

Mark Snapshot...

Data Axis for Accelerometer

Auto adjust 512

aX on Sensor

aY on Sensor

aZ on Sensor

Add Plot

Data Axis for Gyroscope

Auto adjust 512

gX on Sensor

gY on Sensor

gZ on Sensor

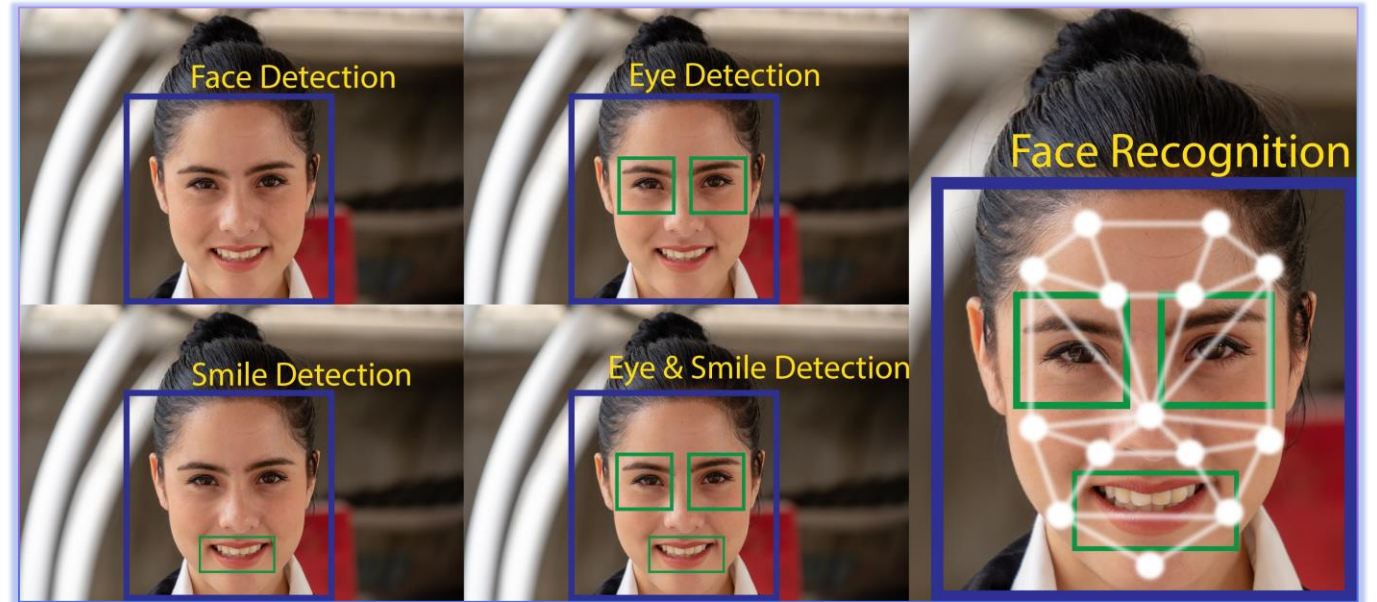
Add Plot

Add Data Axis

Data Collection

Select and Collect dataset

- **Data is the basis of model training.**
 - Quantity
 - Quality
 - Diversity
 - Performance evaluation
 - Ethical
- **Where to get the data?**
 - Acquire it
 - Buy it
 - Build it



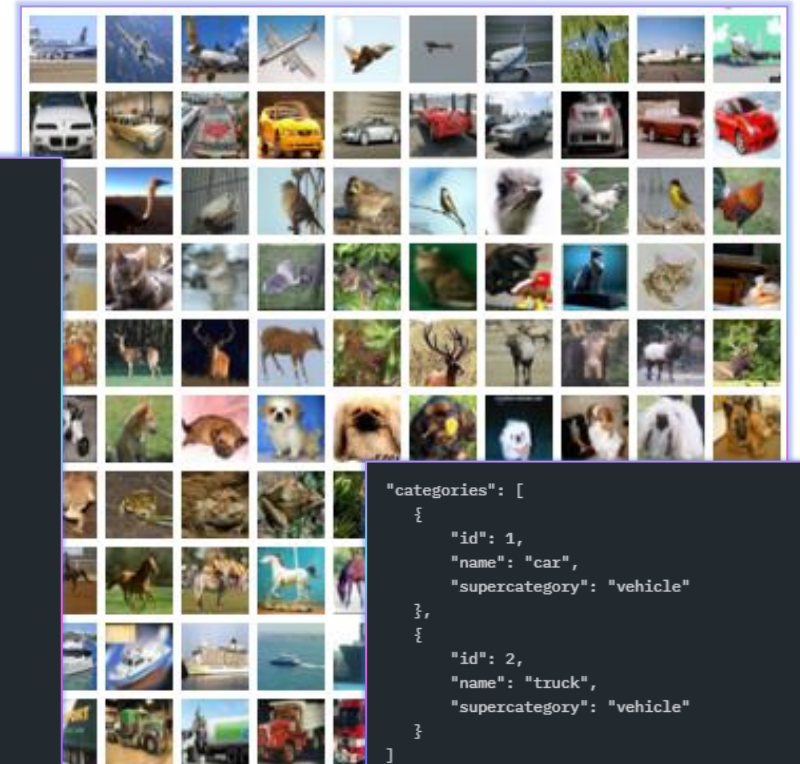
Example MNIST

- Collection of handwritten digits widely used for training and testing in the field of machine learning and computer vision
- **70,000 images**
 - 60,000 training samples
 - 10,000 testing samples
- **Image Properties**
 - Grayscale images (0-255 levels)
 - Size of 28x28 pixels
- **Labels**
 - Each image is labelled with the digit it represents (0 through 9)
- **Benchmark for evaluating the performance of image processing systems**
- **Availability**
 - The dataset can be freely downloaded for use with various data science and machine learning libraries
- **Useful for creating Convolutional Neural Networks**



Example MS COCO

- The Microsoft Common Objects in Context – Object detection, segmentation, and caption. Pre-divided into training, validation, and test sets
- 80 object categories with a wide range from everyday scenes.
 - Diverse and Complex Scenes
 - Closely resemble real-world scenarios
- **Detailed annotations**
 - Object bounding boxes
 - Segmentation masks
 - Image captions
- **Over 200,000 labelled images**
- **Image Captions**
- **Updated yearly**
- **Free for researchers**



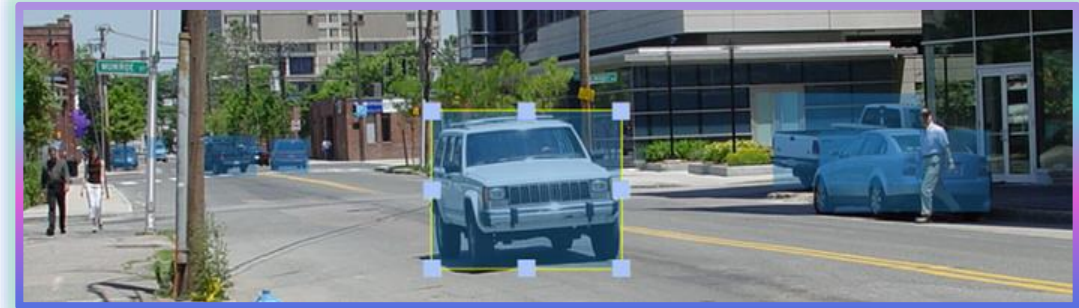
```
"annotations": [  
  {  
    "id": 1,  
    "image_id": 1,  
    "category_id": 1,  
    "segmentation": [  
      [  
        56.97,  
        56.97  
      ],  
      [  
        56.97,  
        56.97  
      ],  
      ...  
    ],  
    "area": 2351.45,  
    "bbox": [  
      56.97,  
      56.97,  
      56.97,  
      56.97  
    ],  
    "iscrowd": 0  
  },  
  ...  
]
```

```
"categories": [  
  {  
    "id": 1,  
    "name": "car",  
    "supercategory": "vehicle"  
  },  
  {  
    "id": 2,  
    "name": "truck",  
    "supercategory": "vehicle"  
  },  
  ...  
]
```


Obtaining Datasets

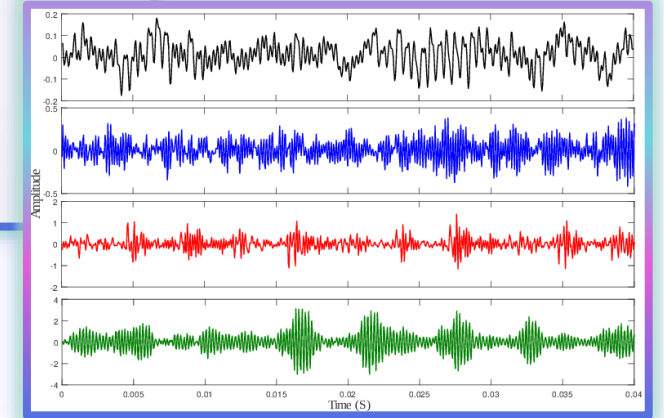
- **Limited number of *free* datasets**
 - Check licenses
- **Dataset Providers**
 - Kaggle (public datasets)
 - Subsidiary of Google
 - Companies that will create a dataset (for a fee)
 - Scale AI
 - MTurk (Amazon Mechanical Turk)
 - Crowdsourced marketplace for data labelling
 - Most focus on image analysis or Generative AI
- **Create your own**
 - Sometimes only viable option
 - New products/fields-of-play
 - Where data confidentiality is important
 - You know your data
 - *But you can be too close to it*

kaggle

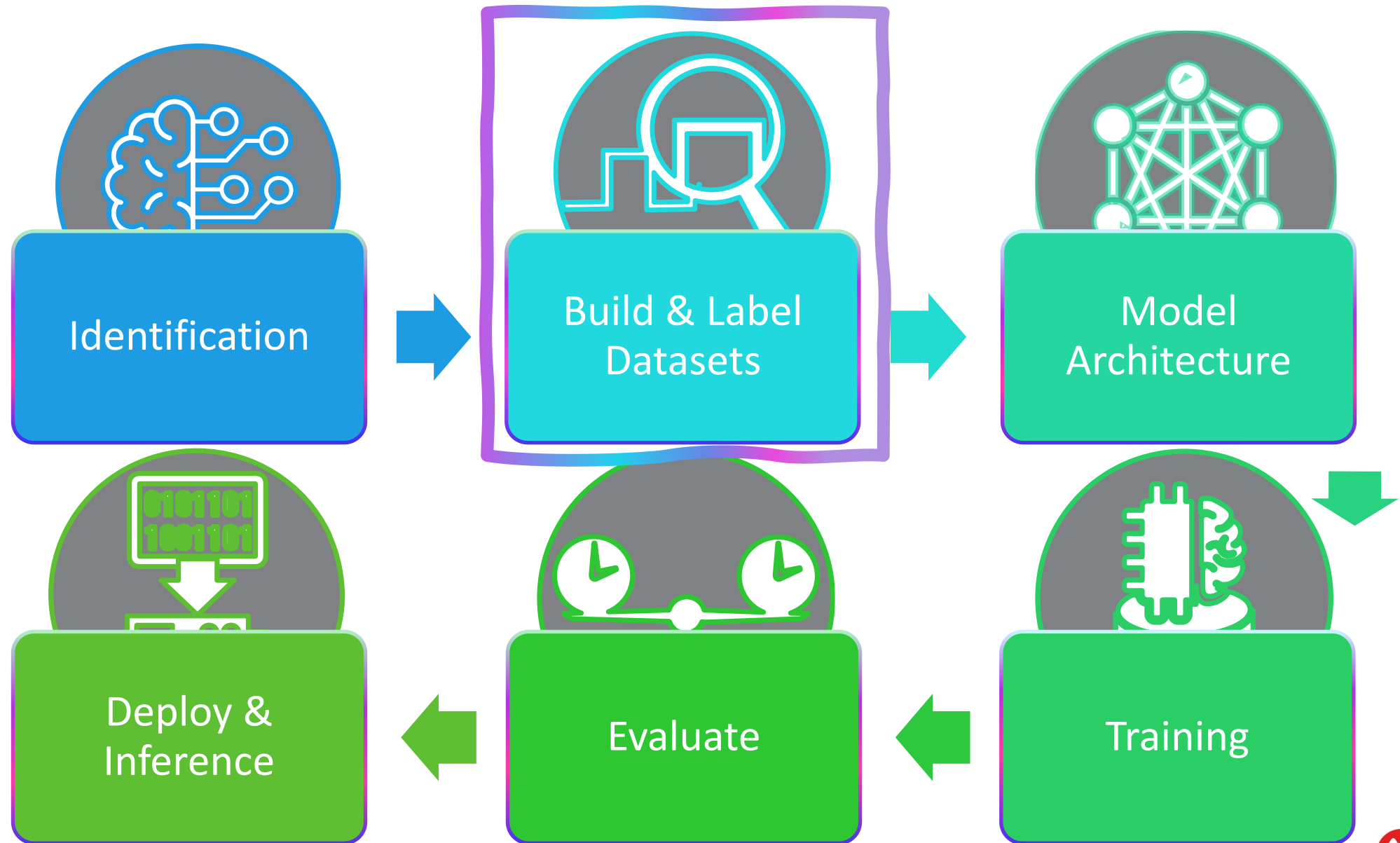


Vibration Analysis

- **Bearing Data Center (CWRU)**
 - Vibration sensor samples at 12kHz
 - EDM Artificial Damage
- **Machinery Failure Data**
 - Extensive set of vibration readings
 - Common failure modes for broad range of machines
- **NASA Acoustics and Vibration Database**
 - Great for building rockets



Machine Learning workflow



MPLAB® Data Visualizer

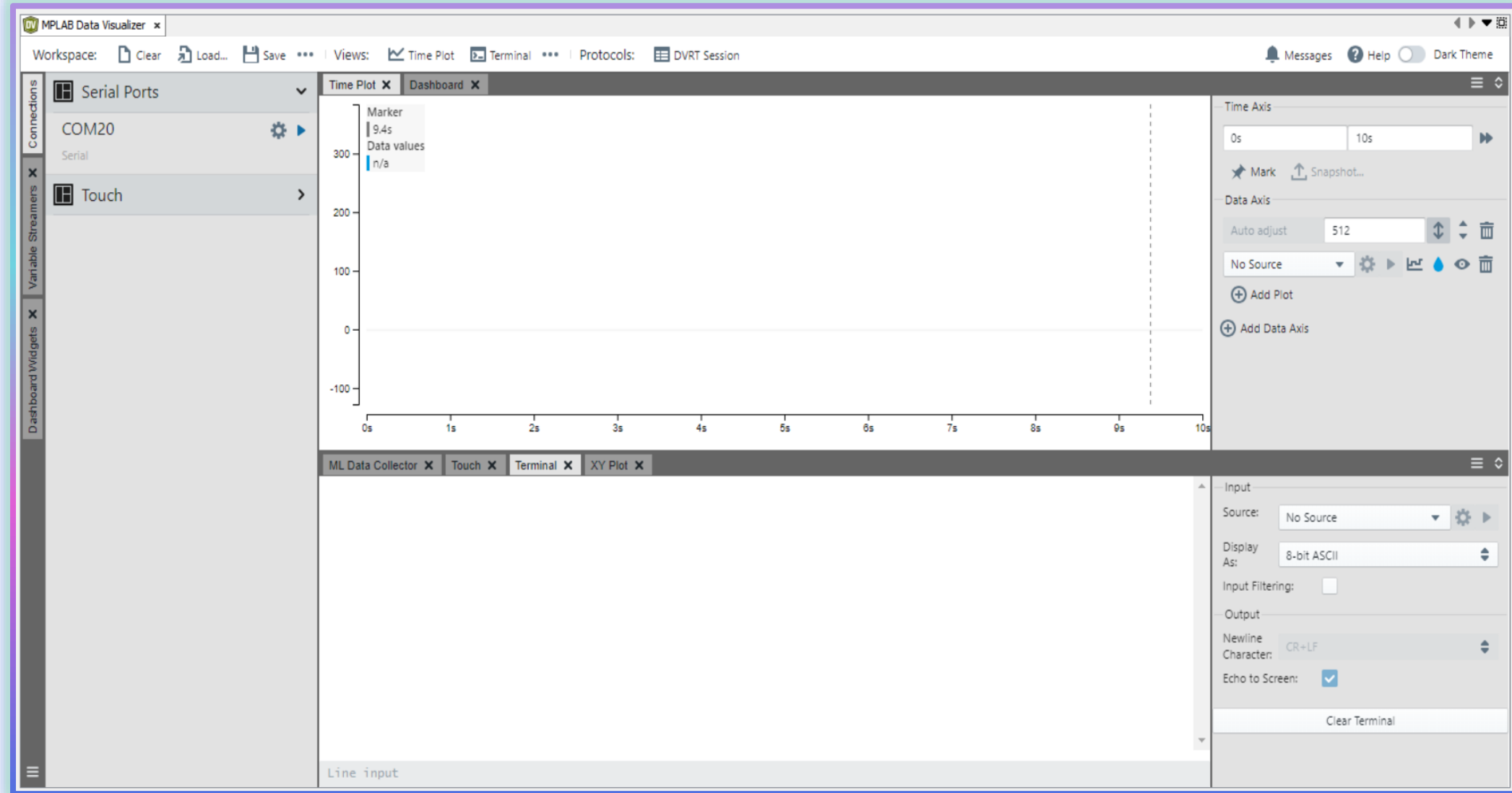
MPLAB Data Visualizer

- **MPLAB® Data Visualizer is a free debugging tool that graphically displays run-time variables in an embedded application.**
- **Available as a plug-in for MPLAB X Integrated Development Environment (IDE) or a stand-alone debugging tool, it can receive data from various sources such as the Embedded Debugger Data Gateway Interface (DGI) and COM ports.**



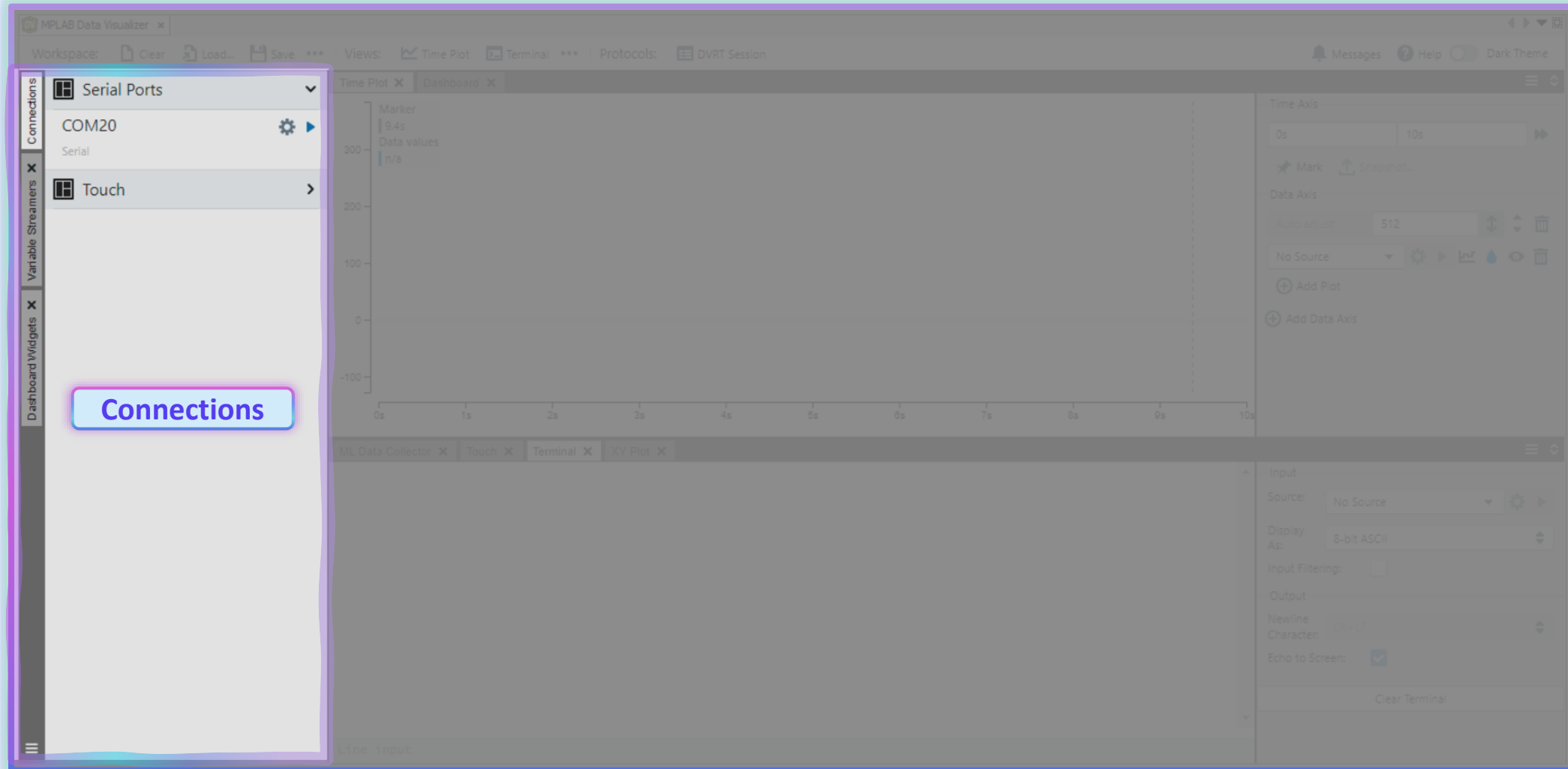
[View Data Stream format for DV](#)

Interface of DV



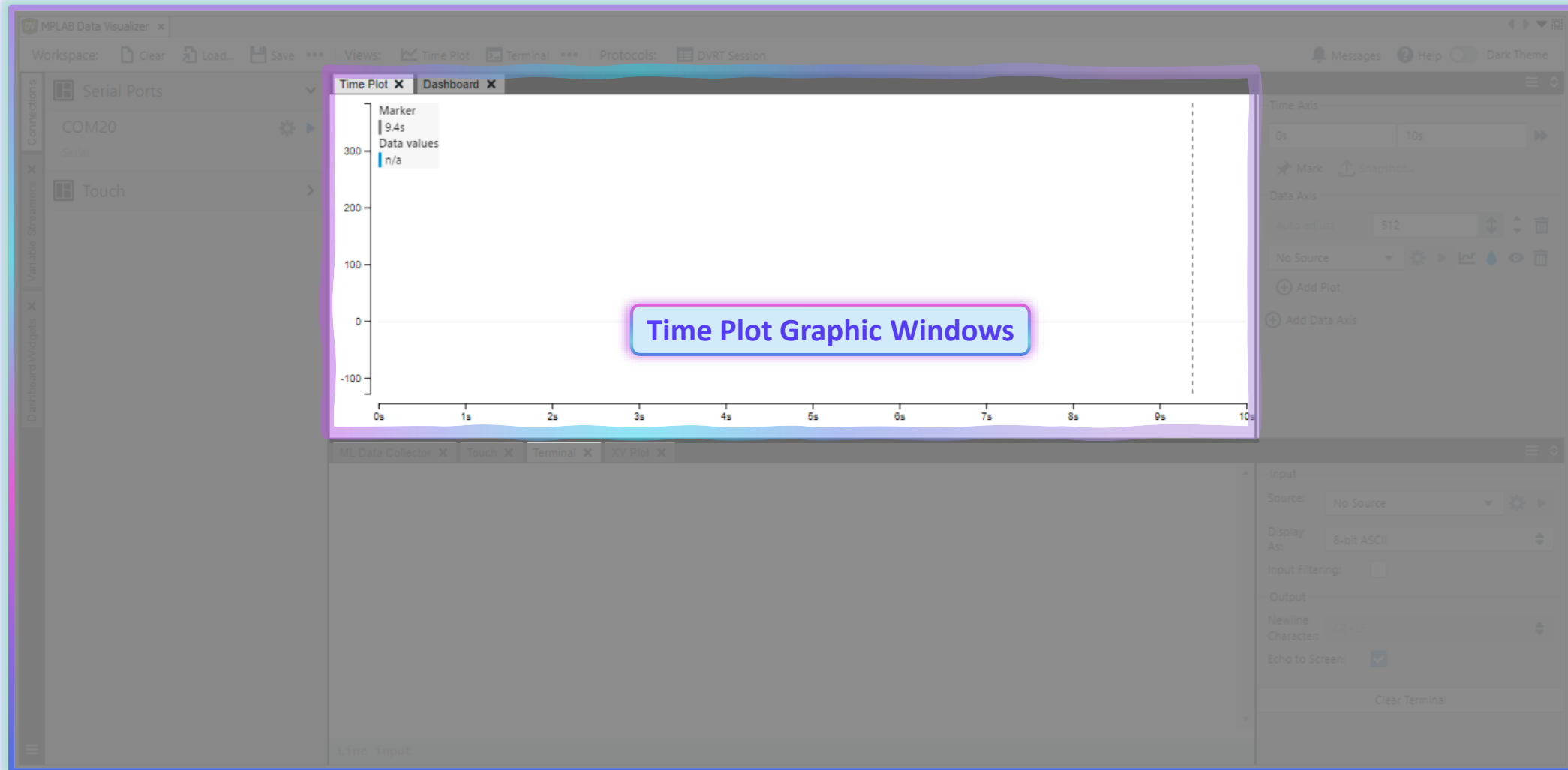
Interface of DV

- Connections



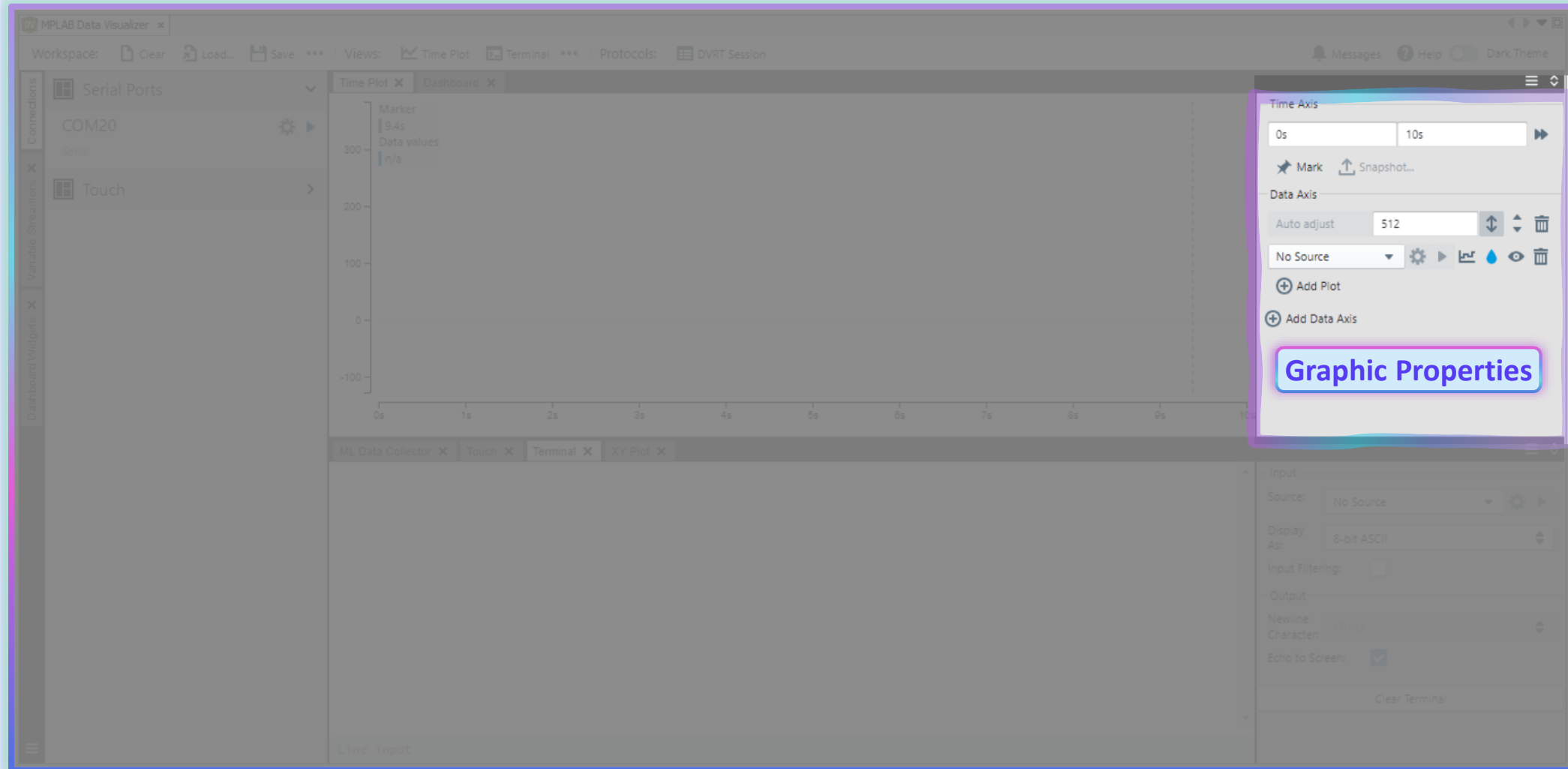
Interface of DV

- Time Plot Graphic Windows



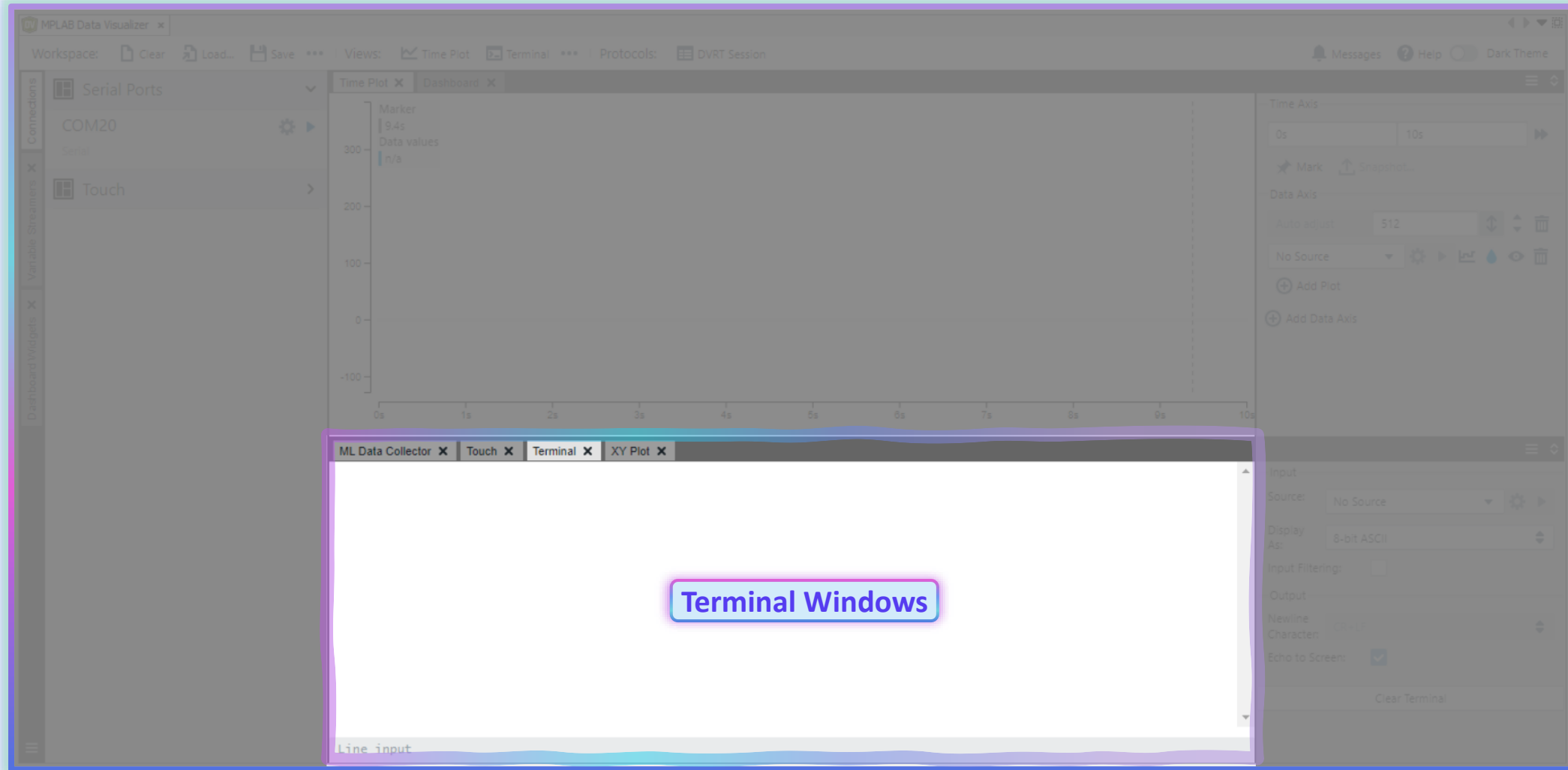
Interface of DV

- **Graphic Properties**



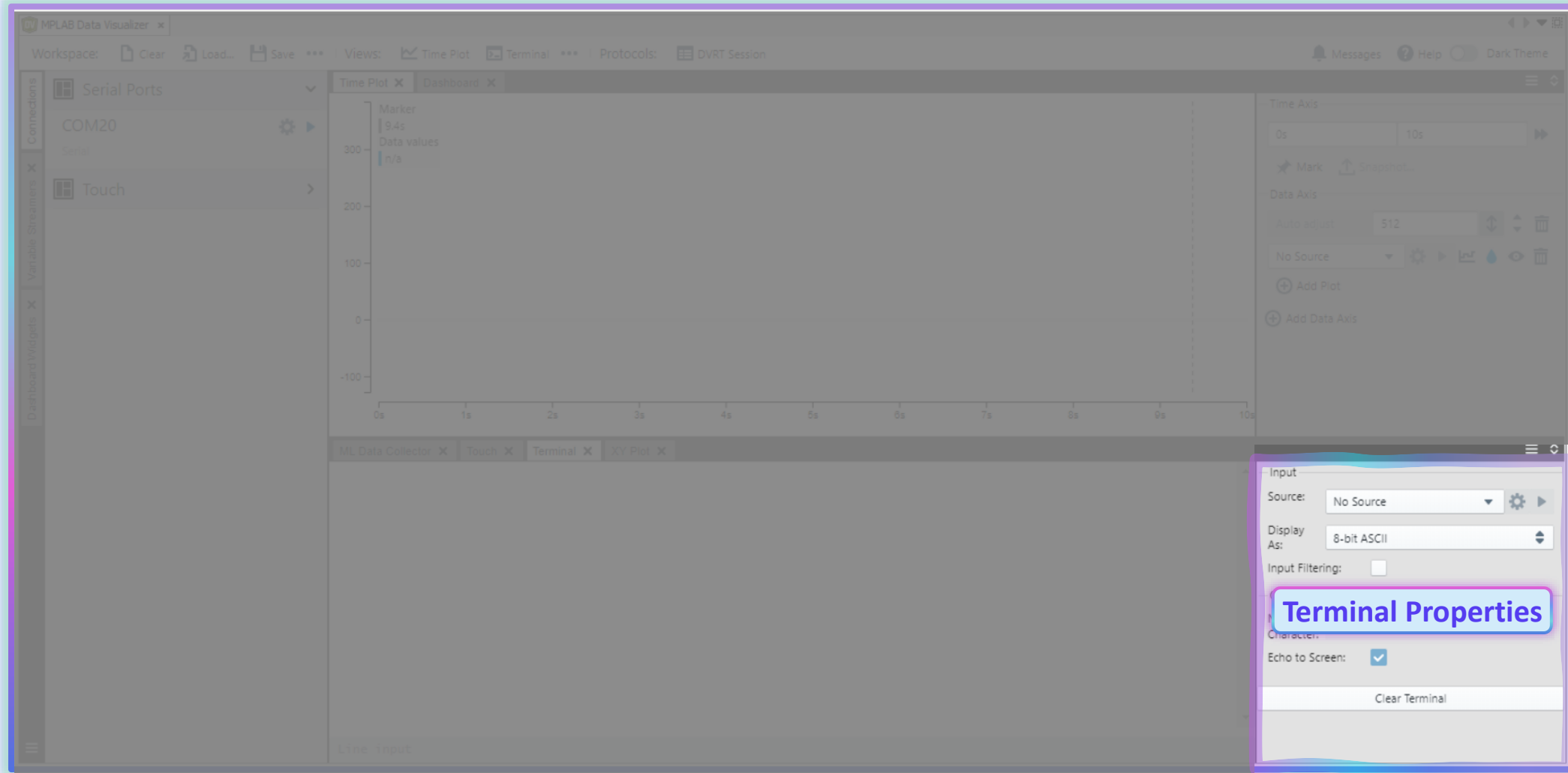
Interface of DV

- Terminal Windows



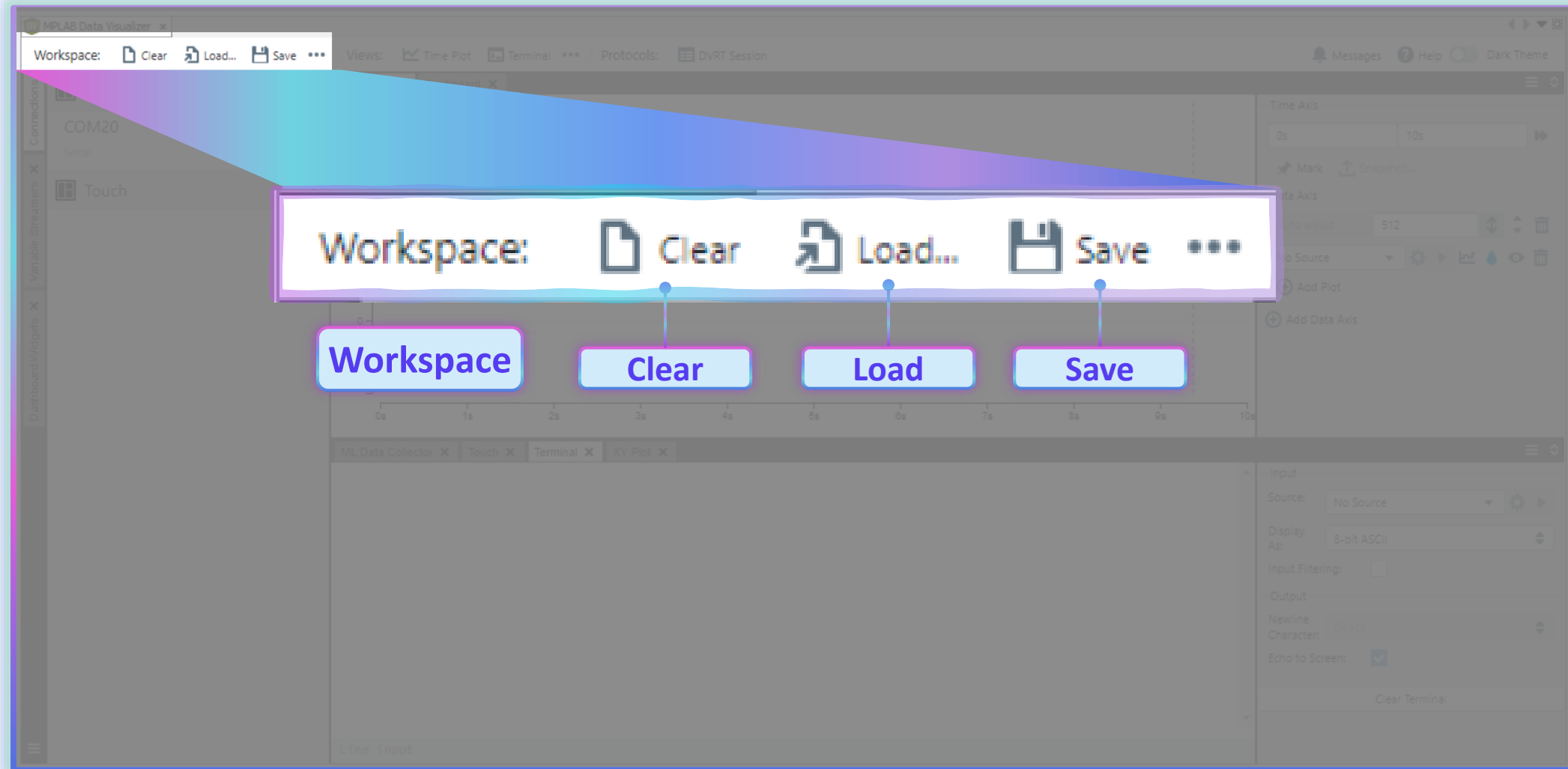
Interface of DV

- Terminal Properties



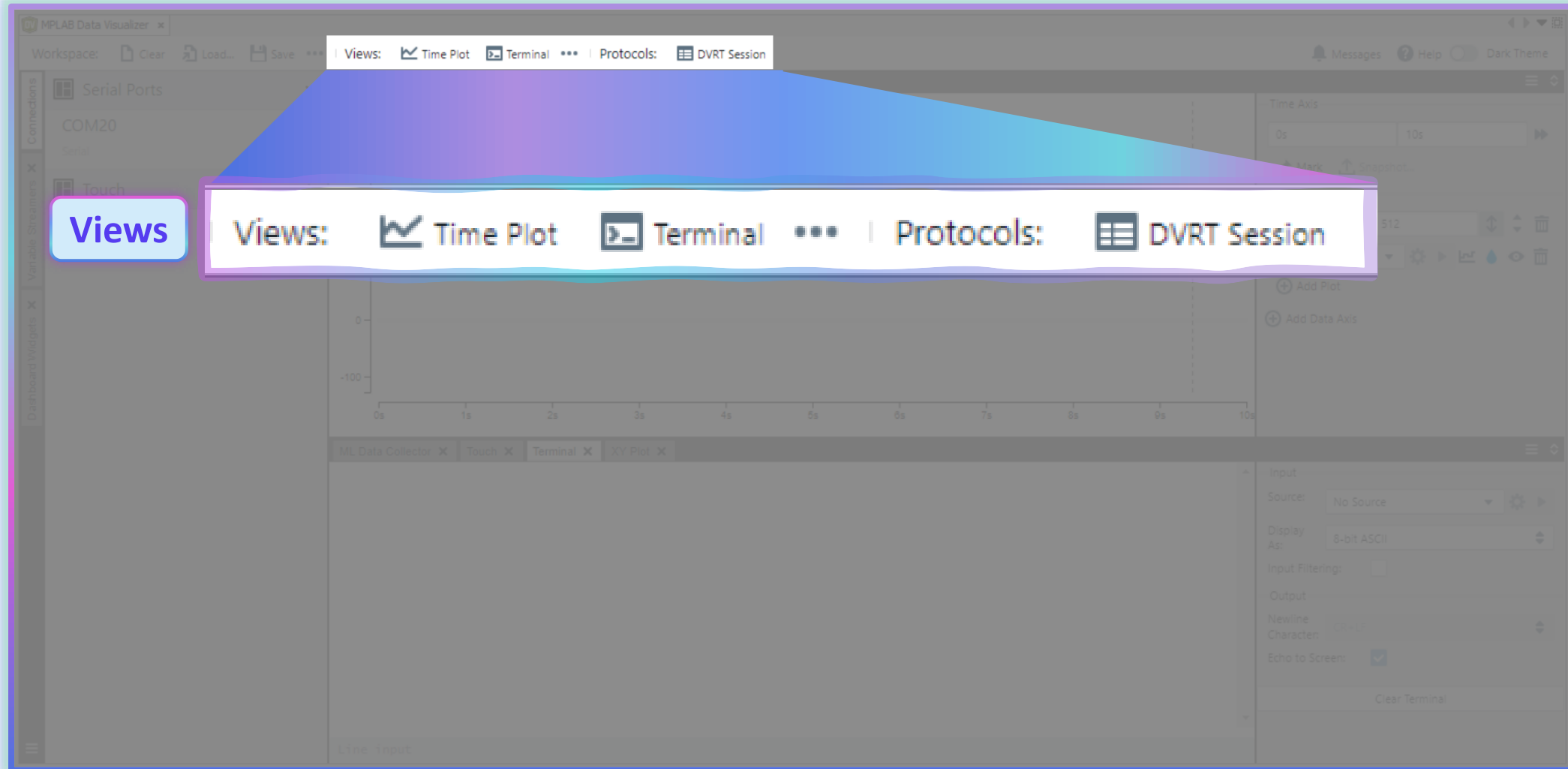
Interface of DV

- **Workspace**



Interface of DV

- Views



Interface of DV



Show the information of all cursors. Including bandwidth, Time period, and Frequency

Need two cursor to show the window.

Click to show/hide the Axis.

Lab1: Collect Data by MPLAB Data Visualizer

Lab1 Collect Data by DV

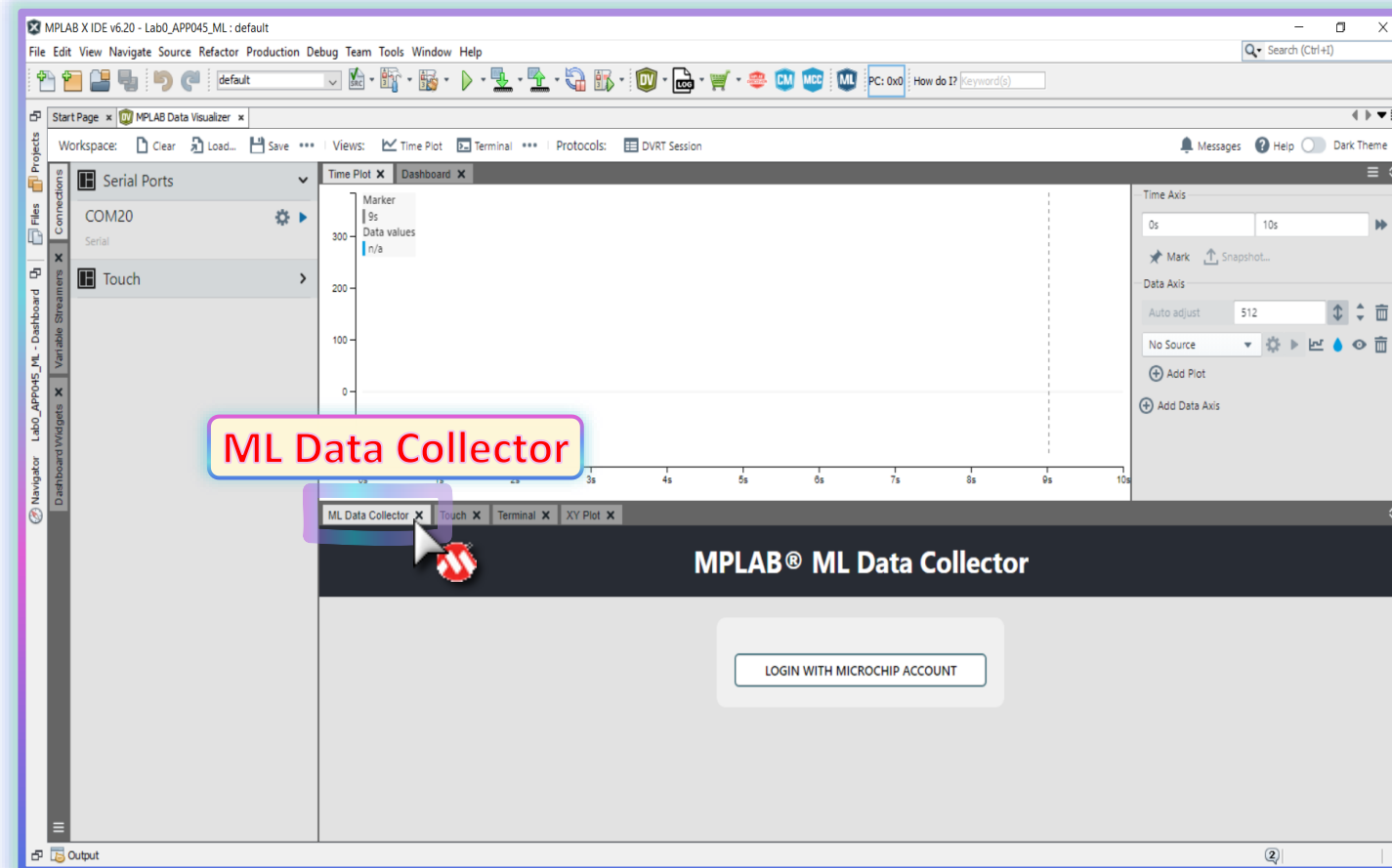
- **Use DV to collect data and save as the Dataset (CSV files) in two different ways.**
 1. Upload and save Data by ML Data Collector.
 2. Snapshot Data by DV and save the file in local. (Optional)
- **Build at least 2 CSV files for 3 different actions.**
 - Straight1.csv, Straight2.csv
 - UpperCut1.csv, UpperCut2.csv
 - Righthook1.csv, Righthook2.csv

Let's go!

Lab1 Collect Data by DV

Step 1

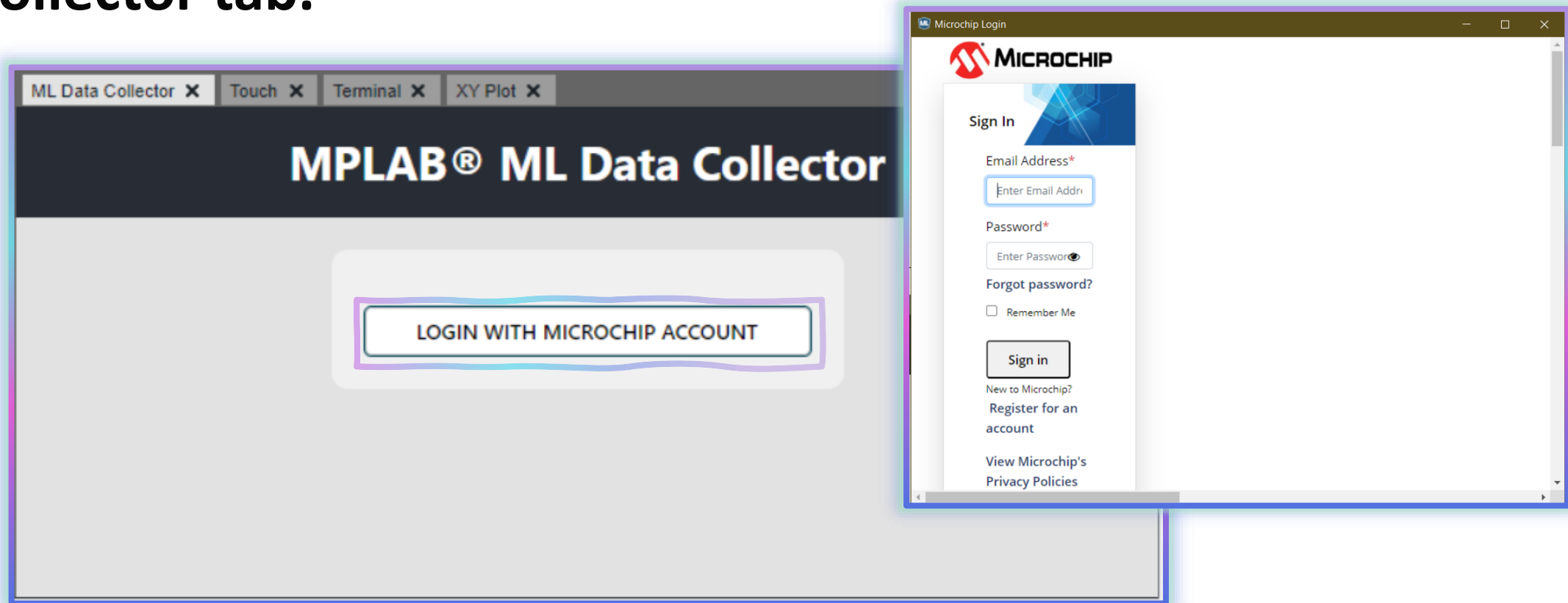
- Choose **ML Data Collector** tab in the below window.



Lab1 Collect Data by DV

Step 2

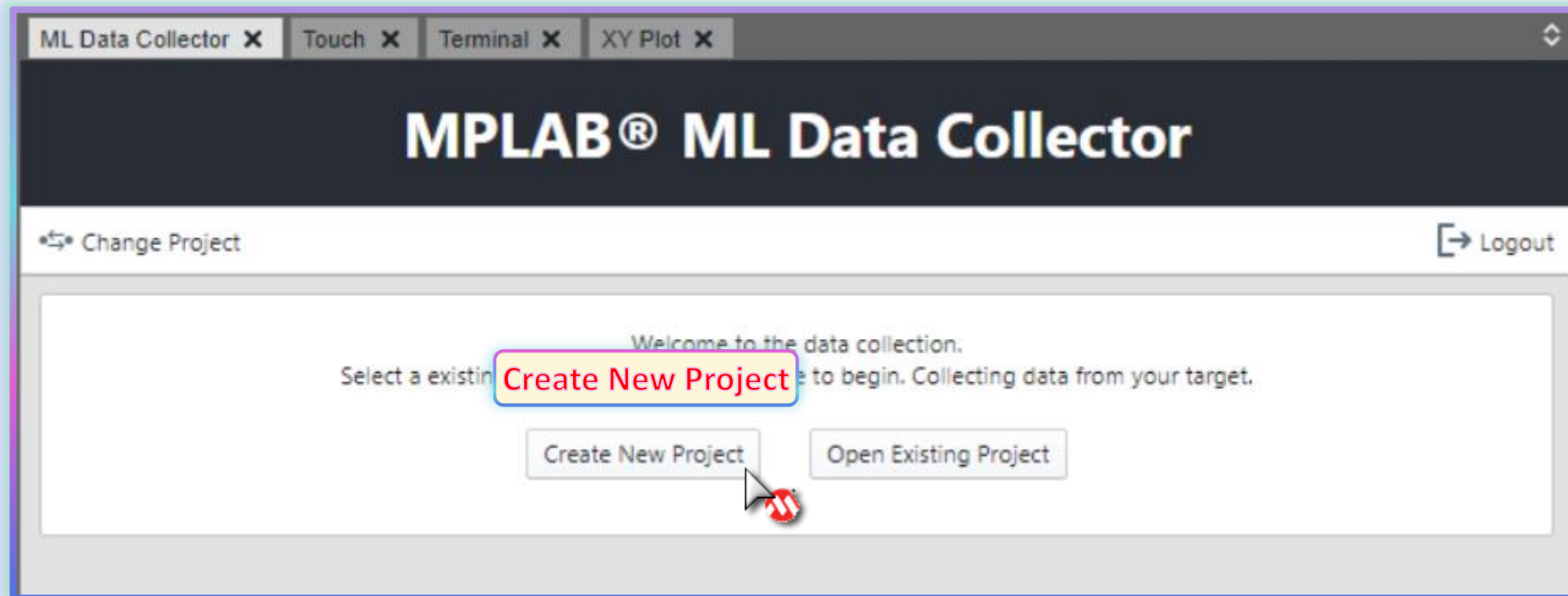
- Log in with your **myMicrochip** credentials in the MPLAB ML Data Collector tab.



Lab1 Collect Data by DV

Step 3

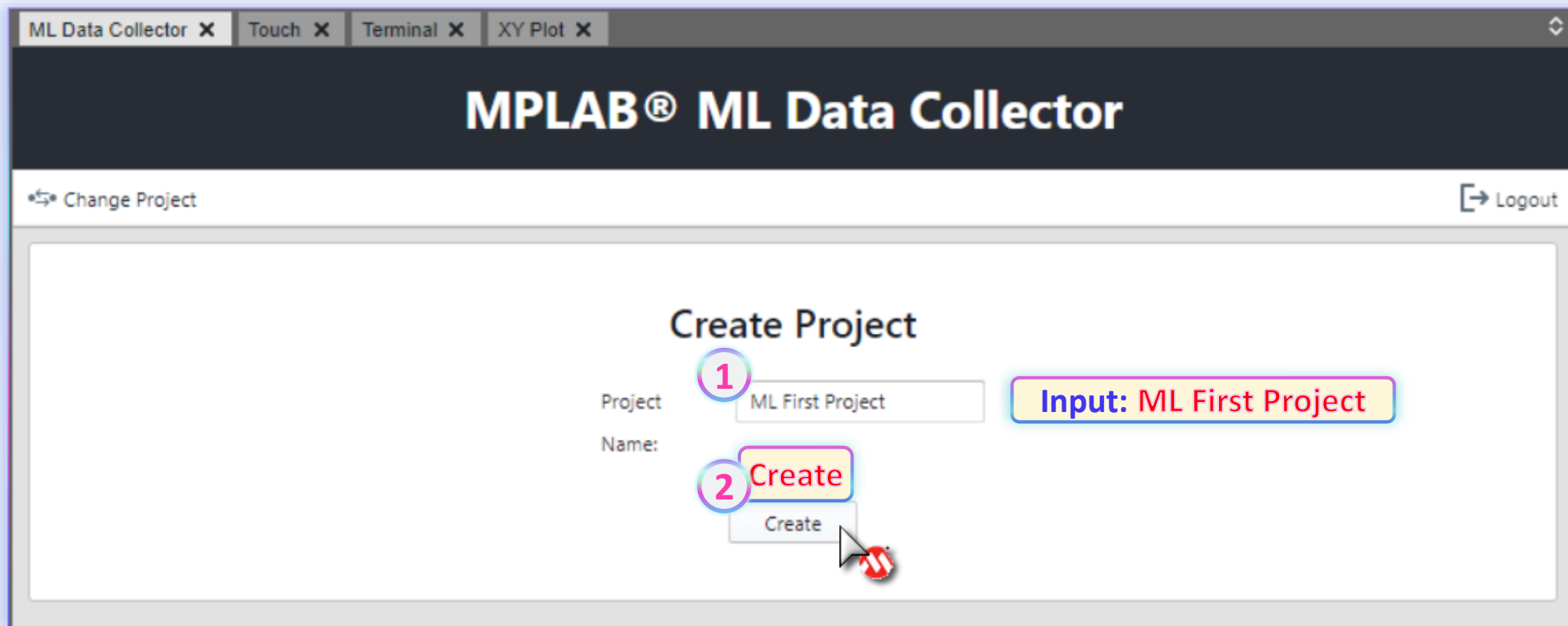
- Click the **Create New Project** button.



Lab1 Collect Data by DV

Step 4

- Input a name for your project.
Project Name: **ML First Project**



The screenshot shows the MPLAB ML Data Collector web interface. At the top, there are tabs for 'ML Data Collector', 'Touch', 'Terminal', and 'XY Plot'. The main header reads 'MPLAB® ML Data Collector'. Below the header, there is a 'Change Project' link and a 'Logout' button. The central area is titled 'Create Project'. It contains a form with two fields: 'Project Name' and 'Name'. The 'Project Name' field is highlighted with a pink circle and the number '1', and it contains the text 'ML First Project'. To the right of this field is a yellow box with the text 'Input: ML First Project'. The 'Name' field is highlighted with a pink circle and the number '2', and it contains the text 'Create'. Below the 'Name' field is a 'Create' button, which is also highlighted with a pink circle and the number '2'. A mouse cursor is pointing at the 'Create' button. The interface is clean and modern, with a dark header and a light main area.

Lab1 Collect Data by DV

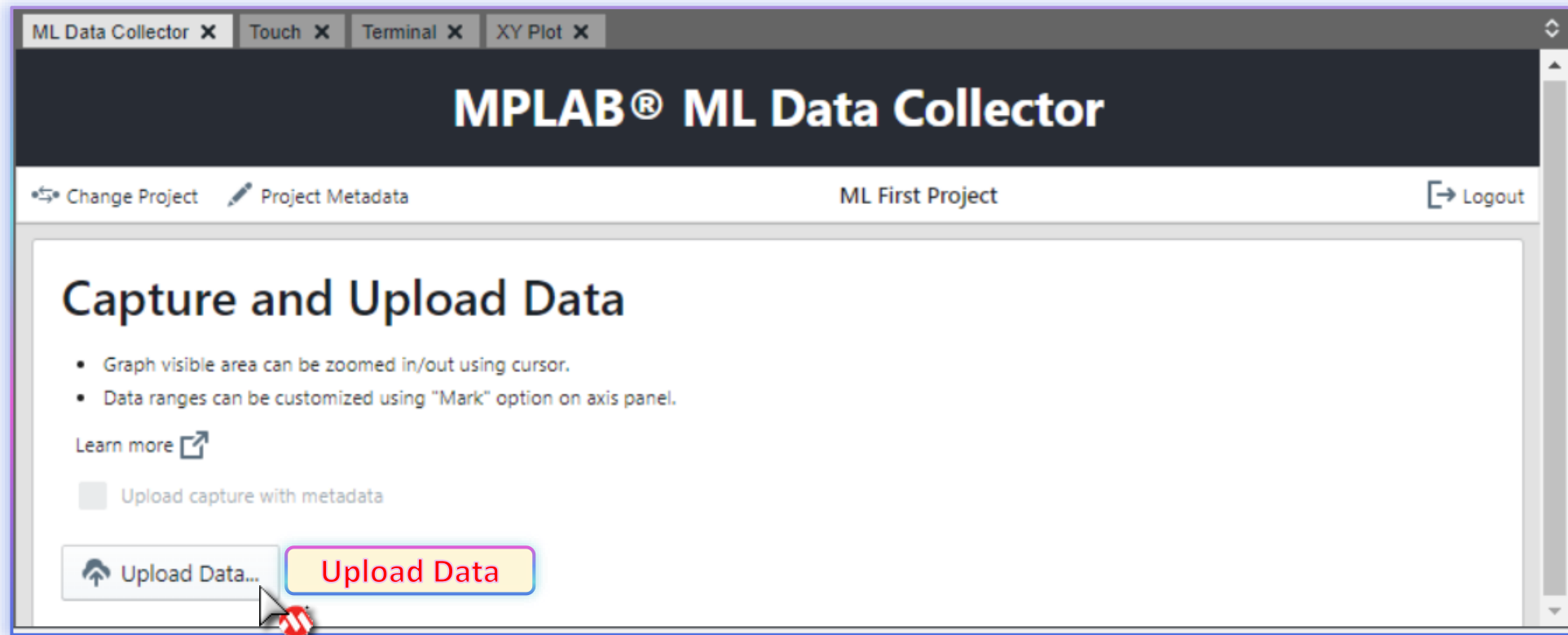
Step 5

- Two ways to capture and Upload Data.
 1. Upload and save Data by ML Data Collector. The data will be uploaded to **the project in your MyMicrochip account**.
 2. Snapshot Data by DV and save the file **in local**. Must upload the data to your project in your MyMicrochip account by yourself. (Optional)

Lab1 Collect Data by DV

Step 6

- Upload and save Data by ML Data Collector
- Click **Upload Data**



Lab1 Collect Data by DV

Step 7

- Profile Name: **Boxing ML**
- Device Name: **BMI160**
- Sample Rate (Hz): **100Hz**

ML Data Collector x Touch x Terminal x XY Plot x

MPLAB® ML Data Collector

Change Project Project Metadata ML First Project

Create Sensor Configuration

Profile Name:

Device Name:

Sample Rate (Hz):

Sensor Names :

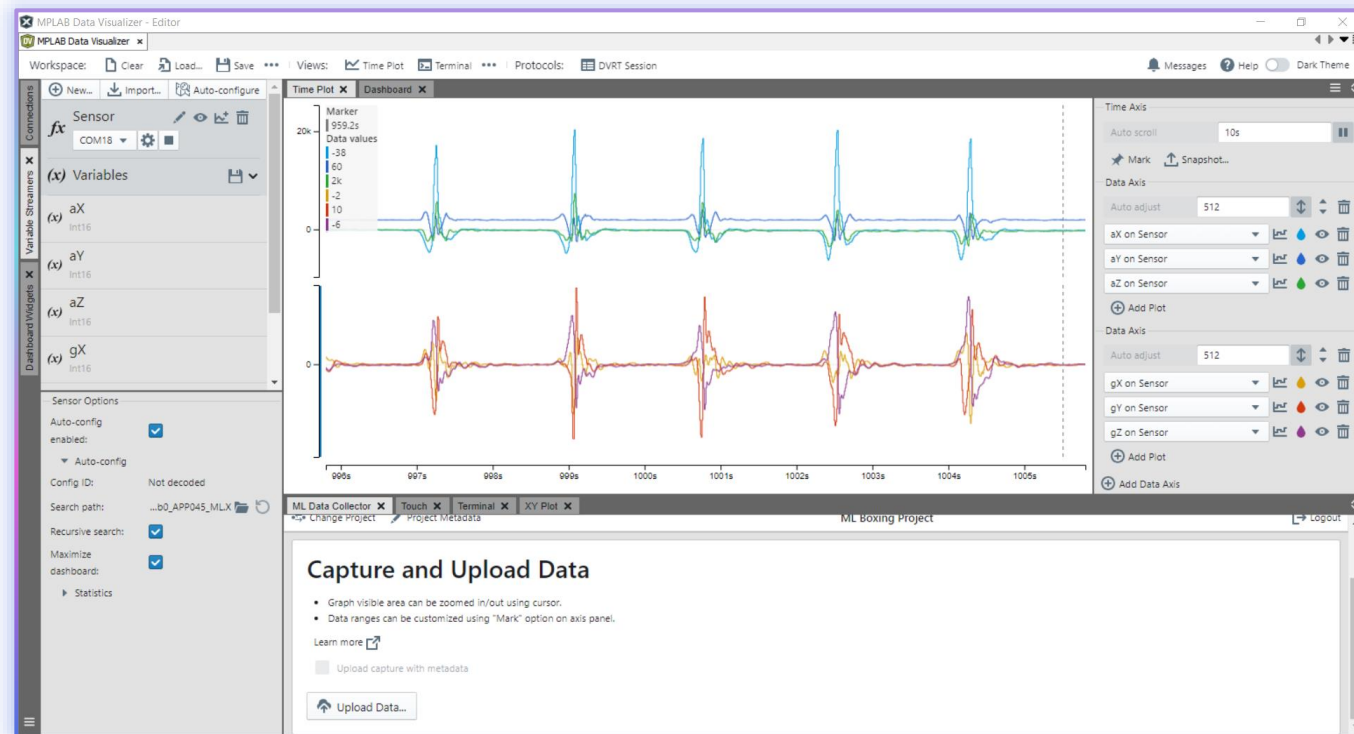
- ☐ aX
- ☐ aY
- ☐ aZ
- ☐ gX
- ☐ gY
- ☐ gZ

Back Create Create

Lab1 Collect Data by DV

Step 8

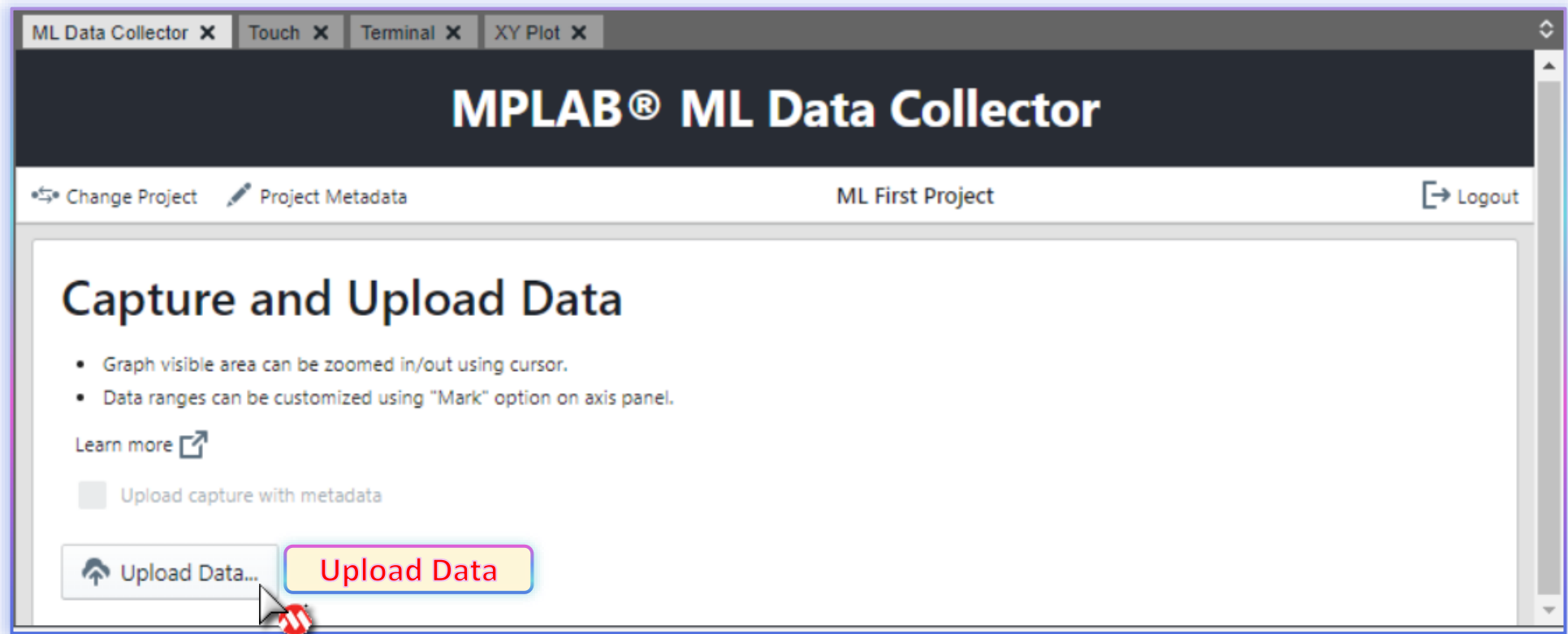
- Observe the data sent by g-sensor.
- Start moving your body and do some **straight** action. **Remember the time interval while you start the action.**



Lab1 Collect Data by DV

Step 9

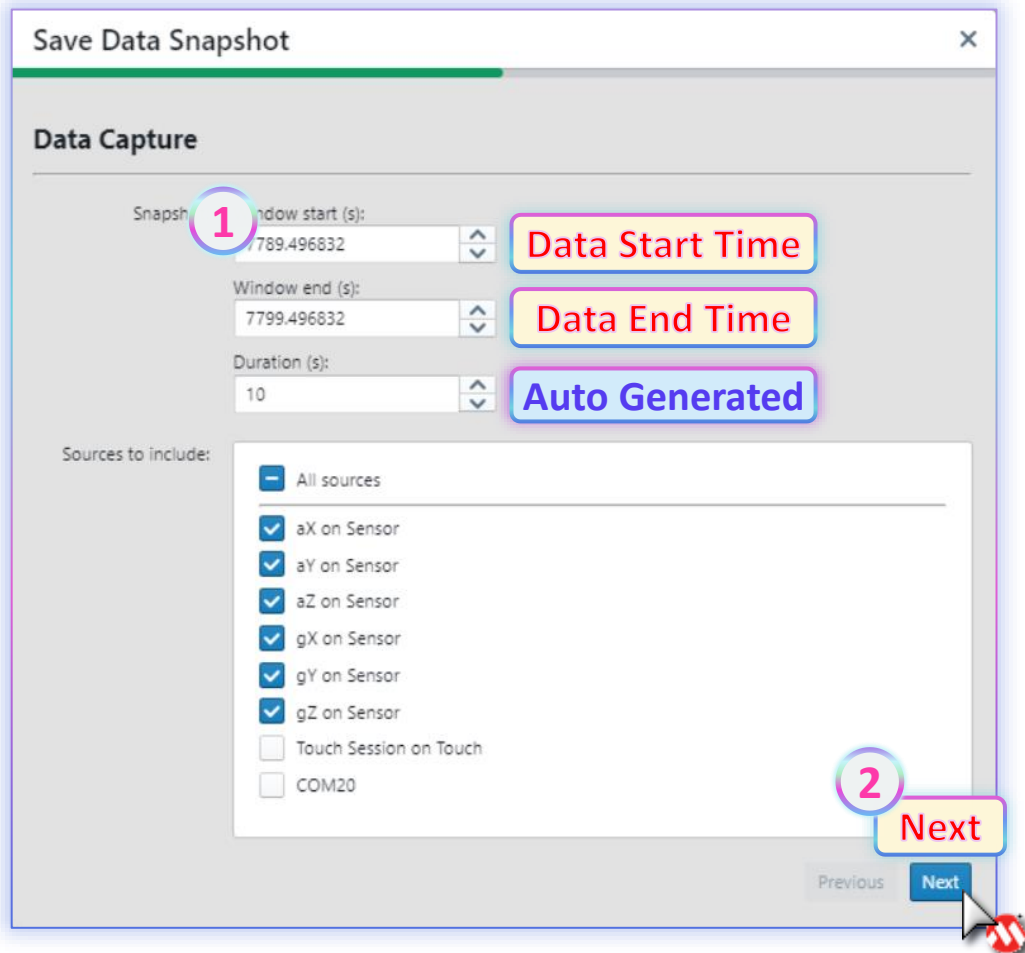
- Capture and Upload Data



Lab1 Collect Data by DV

Step 10

- Enter the **time period** of data to take snapshot



Save Data Snapshot

Data Capture

Snapshot window start (s): 7789.496832 **Data Start Time**

Window end (s): 7799.496832 **Data End Time**

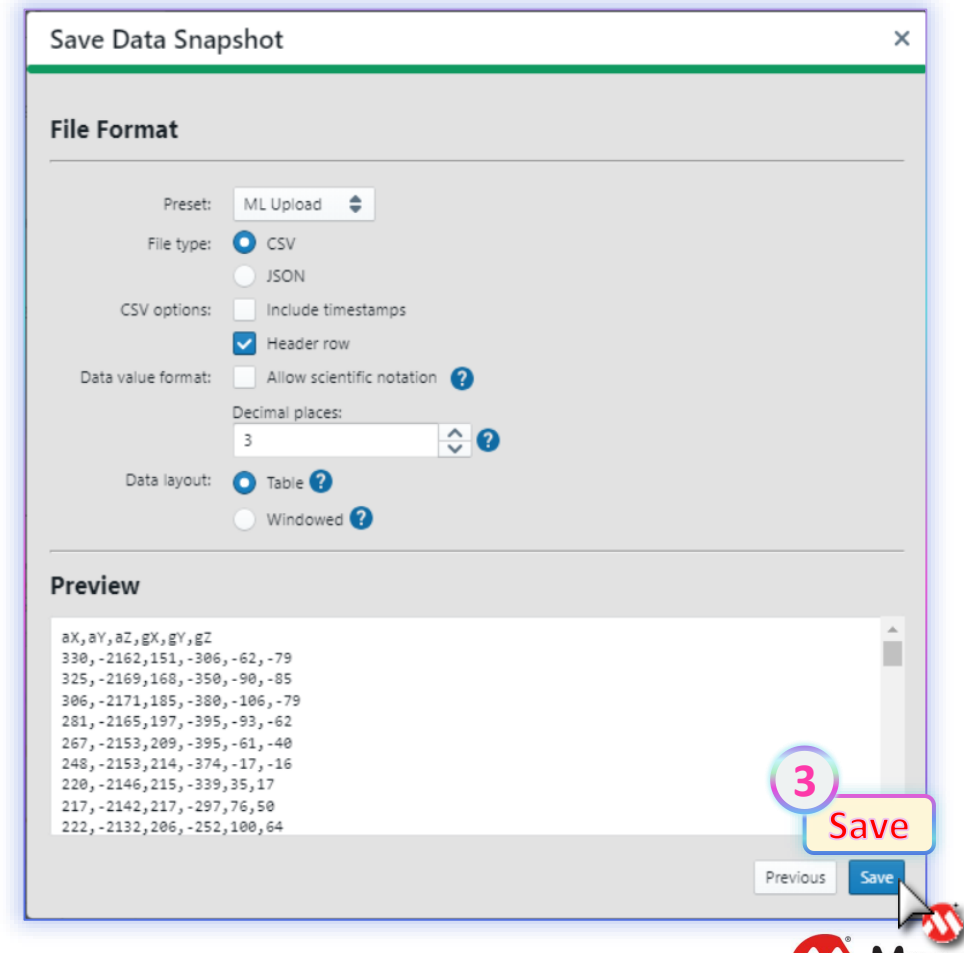
Duration (s): 10 **Auto Generated**

Sources to include:

- ☐ All sources
- ☒ aX on Sensor
- ☒ aY on Sensor
- ☒ aZ on Sensor
- ☒ gX on Sensor
- ☒ gY on Sensor
- ☒ gZ on Sensor
- ☐ Touch Session on Touch
- ☐ COM20

2 **Next**

Previous Next



Save Data Snapshot

File Format

Preset: ML Upload

File type: ☒ CSV ☐ JSON

CSV options: ☐ Include timestamps ☒ Header row

Data value format: ☐ Allow scientific notation

Decimal places: 3

Data layout: ☒ Table ☐ Windowed

Preview

```
aX, aY, aZ, gX, gY, gZ
330, -2162, 151, -306, -62, -79
325, -2169, 168, -350, -90, -85
306, -2171, 185, -380, -106, -79
281, -2165, 197, -395, -93, -62
267, -2153, 209, -395, -61, -40
248, -2153, 214, -374, -17, -16
220, -2146, 215, -339, 35, 17
217, -2142, 217, -297, 76, 50
222, -2132, 206, -252, 100, 64
```

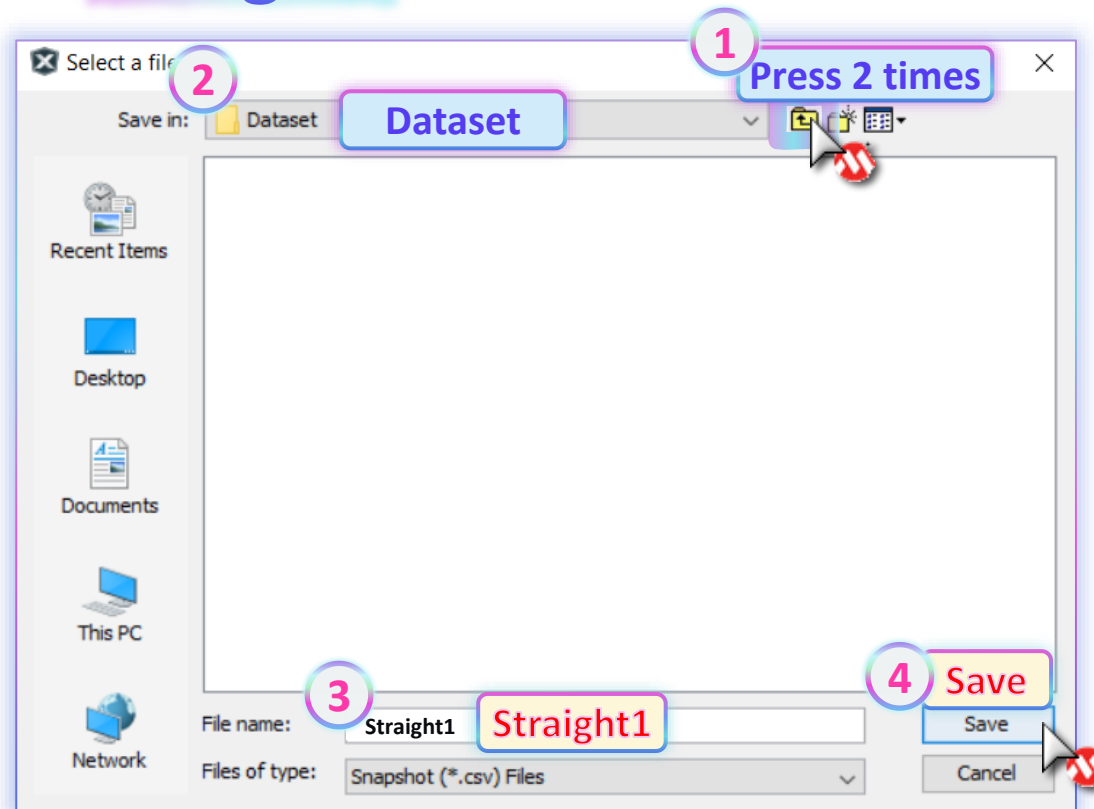
3 **Save**

Previous Save

Lab1 Collect Data by DV

Step 11

- Save the .csv files in a folder. In this project, I create a folder named “**Dataset**” under Lab0_APP045_ML.
- Name the file as **Straight1**.



Lab1 Collect Data by DV

Step 12

- Show File uploaded successfully after the dataset uploaded successfully to the cloud under your myMicrochip account.



- Repeat step 12 to step 17.

Build, Upload and Save all the data file.

- Straight1.csv, Straight2.csv
- UpperCut1.csv, UpperCut2.csv
- Righthook1.csv, Righthook2.csv

Lab1 Collect Data by DV

Precautions

- Before starting capture data, you can first observe the results of different actions G-sensor.
- Ensure **uniformity** and **quality** in executing every action.
- It is *recommended* that the length of each data file should not longer then **30** seconds while today's training.
- It is *recommended* perform **one movement per second**, and rest for about one second between each movement.
- Proper data organization and naming rules will facilitate data retrieval and analysis.

Lab1 Collect Data by DV

Step 13

- Two ways to capture and Upload Data.
 1. Upload and save Data by ML Data Collector. The data will be uploaded to **the project in your MyMicrochip account**.
 2.  Snapshot Data by DV and save the file **in local**. Must upload the data to your project in your MyMicrochip account by yourself. (Optional)

Lab1 Collect Data by DV

Step 14 (Optional)

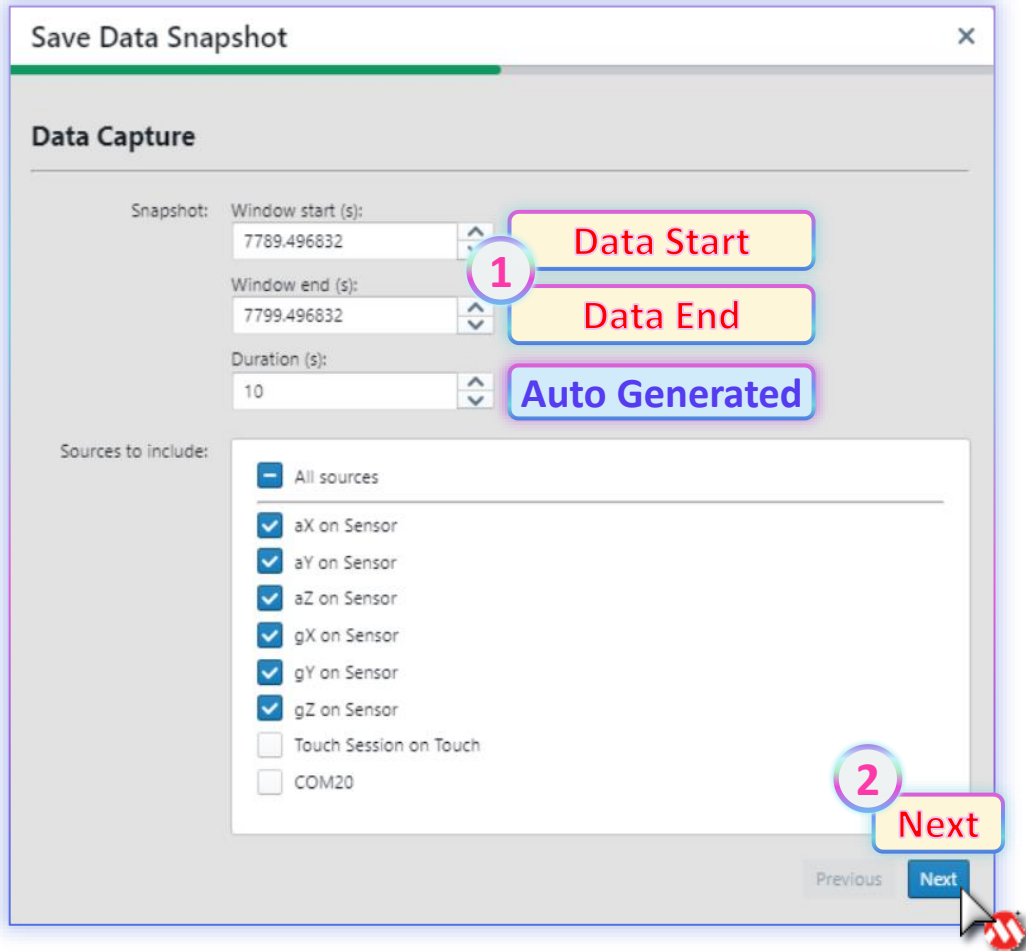
- Click snapshot in the upper right corner.



Lab1 Collect Data by DV

Step 15(Optional)

- Enter the **time period** of data to take snapshot



Save Data Snapshot

Data Capture

Snapshot: Window start (s): 7789.496832 **1 Data Start**

Window end (s): 7799.496832 **Data End**

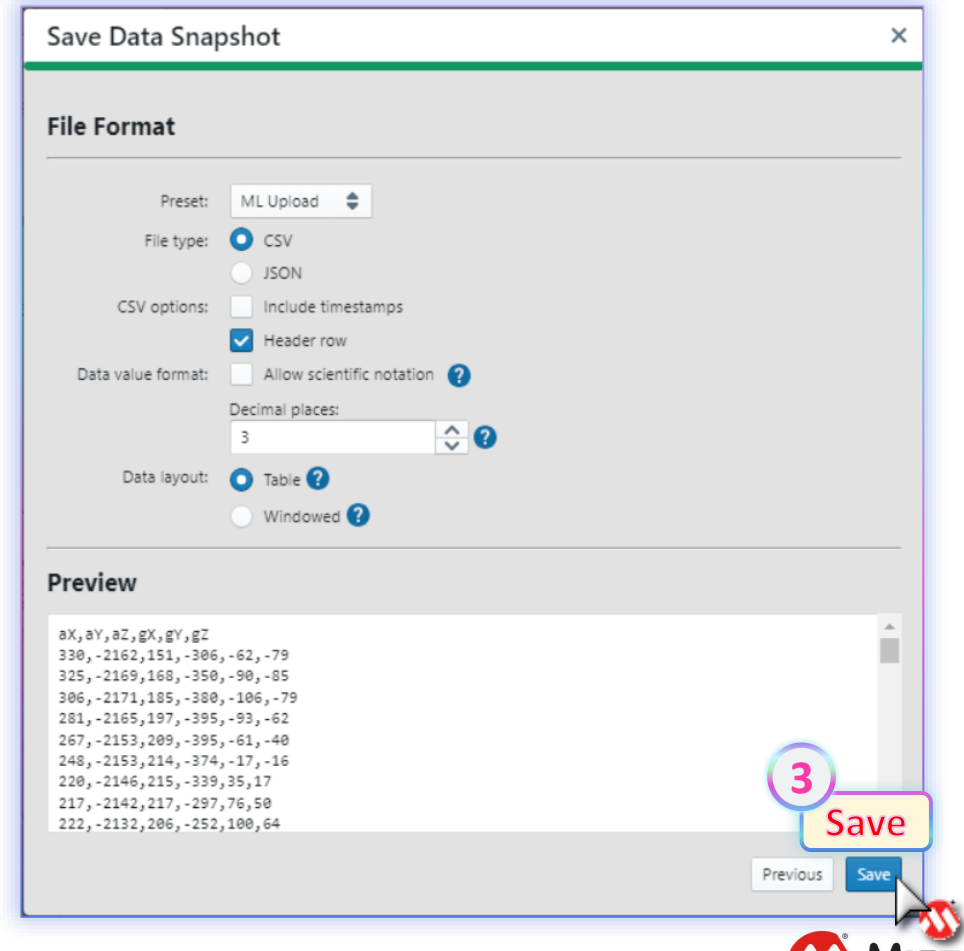
Duration (s): 10 **Auto Generated**

Sources to include:

- ☐ All sources
- ☒ aX on Sensor
- ☒ aY on Sensor
- ☒ aZ on Sensor
- ☒ gX on Sensor
- ☒ gY on Sensor
- ☒ gZ on Sensor
- ☐ Touch Session on Touch
- ☐ COM20

2 Next

Previous Next



Save Data Snapshot

File Format

Preset: ML Upload

File type: ☒ CSV ☐ JSON

CSV options: ☐ Include timestamps ☒ Header row

Data value format: ☐ Allow scientific notation

Decimal places: 3

Data layout: ☒ Table ☐ Windowed

3 Save

Previous Save

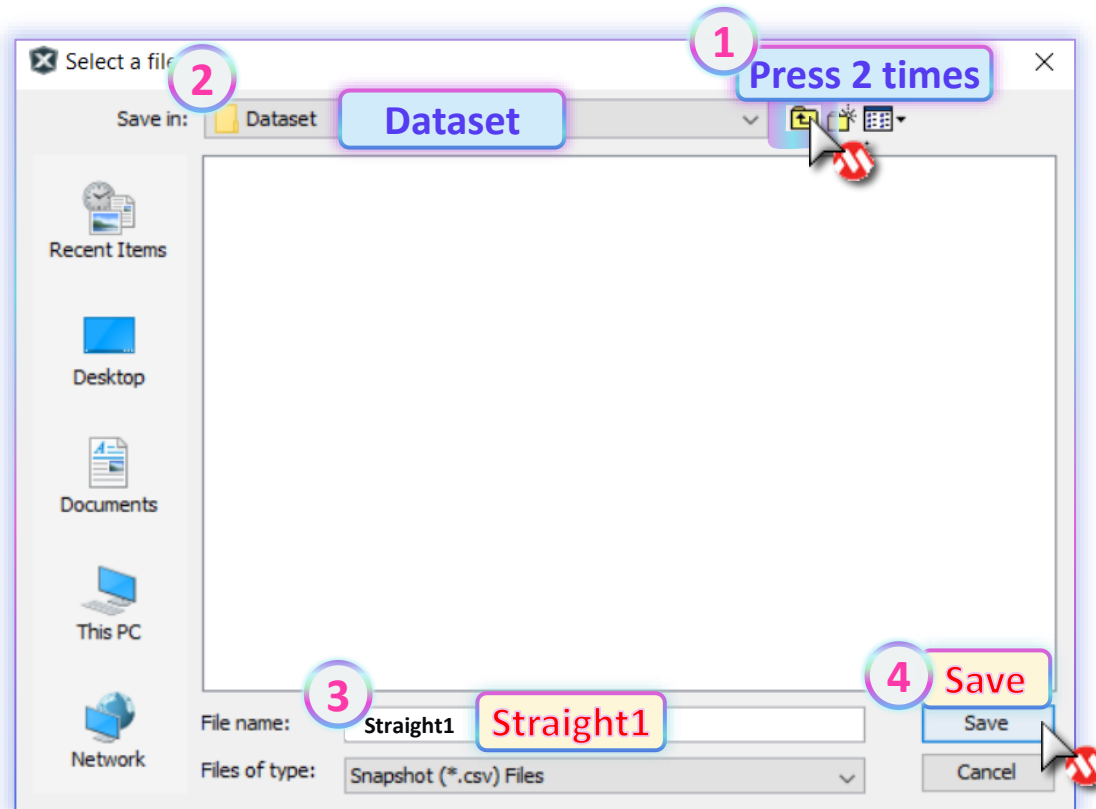
Preview

```
aX,aY,aZ,gX,gY,gZ
330,-2162,151,-306,-62,-79
325,-2169,168,-350,-90,-85
306,-2171,185,-380,-106,-79
281,-2165,197,-395,-93,-62
267,-2153,209,-395,-61,-40
248,-2153,214,-374,-17,-16
220,-2146,215,-339,35,17
217,-2142,217,-297,76,50
222,-2132,206,-252,100,64
```

Lab1 Collect Data by DV

Step 16(Optional)

- Also save the file under folder “Dataset”, and name the .csv file **Straightx** then save.



Organize: Label and Prepare Data

The importance of data

- **Google's Research Director Peter Norvig has been quoted many times on the value of more data in analytics:**
 - “We don't have better algorithms. We have more data.”
 - “Simple models and a lot of data trump more elaborate models based on fewer data.”



Data must be organized - Categorize events

- Where in the sample has it been selected?

- What *thing* has been detected in the picture?

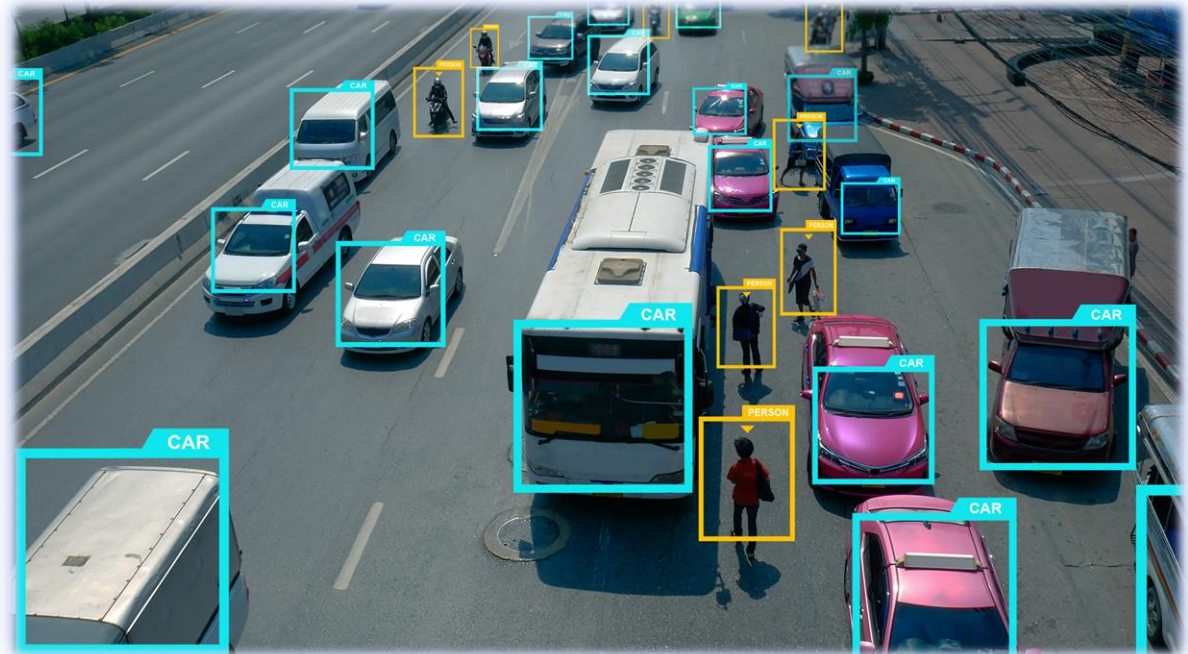
- Bicycle, Person, Bus

- Vibration Classification?

- Knock, Blocked, Normal

- Detect light levels?

- Blink, Steady, Brightness, Off



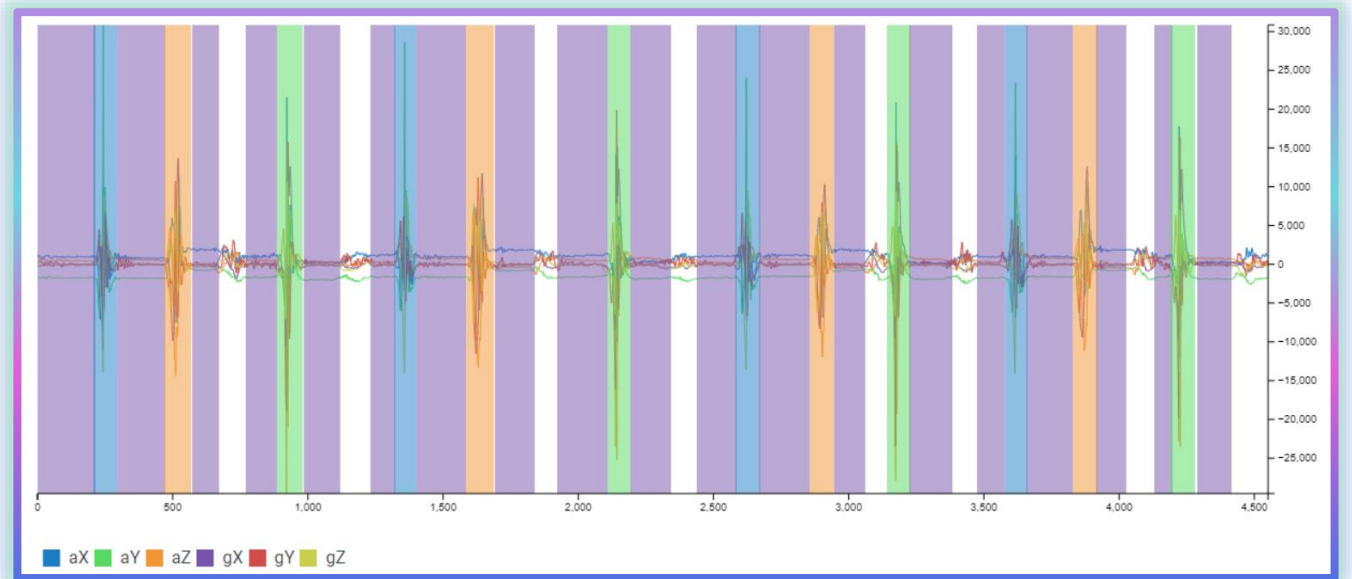
Segment Label and Metadata

- **Segment Label:** Describe the types of our events.
 - The outputs from the classifier.
- **Metadata:** Data about the data. Provide the description of the dataset and define custom information that can be used in training.
 - Data quality management
 - Feature engineering
 - Model training and evaluation
 - Model tracking and version management



Example Data from our Lab

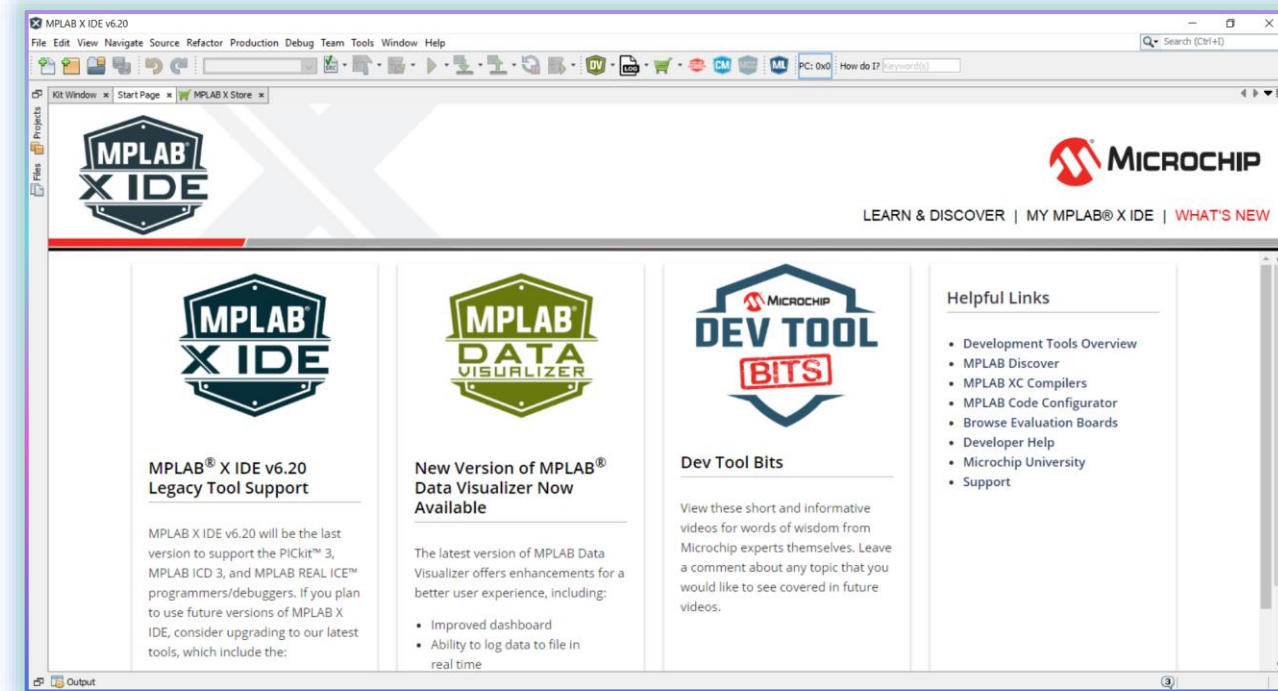
- **Boxing demo**
 - Accelerometer (AX, AY, AZ)
 - Gyroscope (GX, GY, GZ)
- **Multiple Events**
 - Each signal has 'quiet' spells
 - Amplitude, spectrum vary
- **Metadata**
 - Dominant hand
 - Gender



MPLAB® Machine Learning Development Suite

MPLAB ML Development Suite

- The MPLAB ML Model Builder plug-in is an application that filters and optimizes your labeled sensor data through machine learning algorithms. It generates a model (Knowledge Pack) ready to be flashed into the firmware of your device of choice.



MPLAB ML Development Suite

- Create compact supervised and anomaly-detection algorithms that can run on edge based MCUs and MPUs
- Automate each step of the ML flow process.
 - Data preparation
 - Label and filter data
 - Feature extraction
 - Training
 - Validation
 - Deployment
- **Generate AutoML AutoML-powered code**
 - Create a prototype in days or weeks (or 3 hours)
- **Fine-tune aspects of your algorithm**
 - Advanced users can develop with Google Colab or Jupiter Notebooks
- **Firmware generation for 8-, 16- and 32-bit XC compilers**
 - Library and Source Code export options



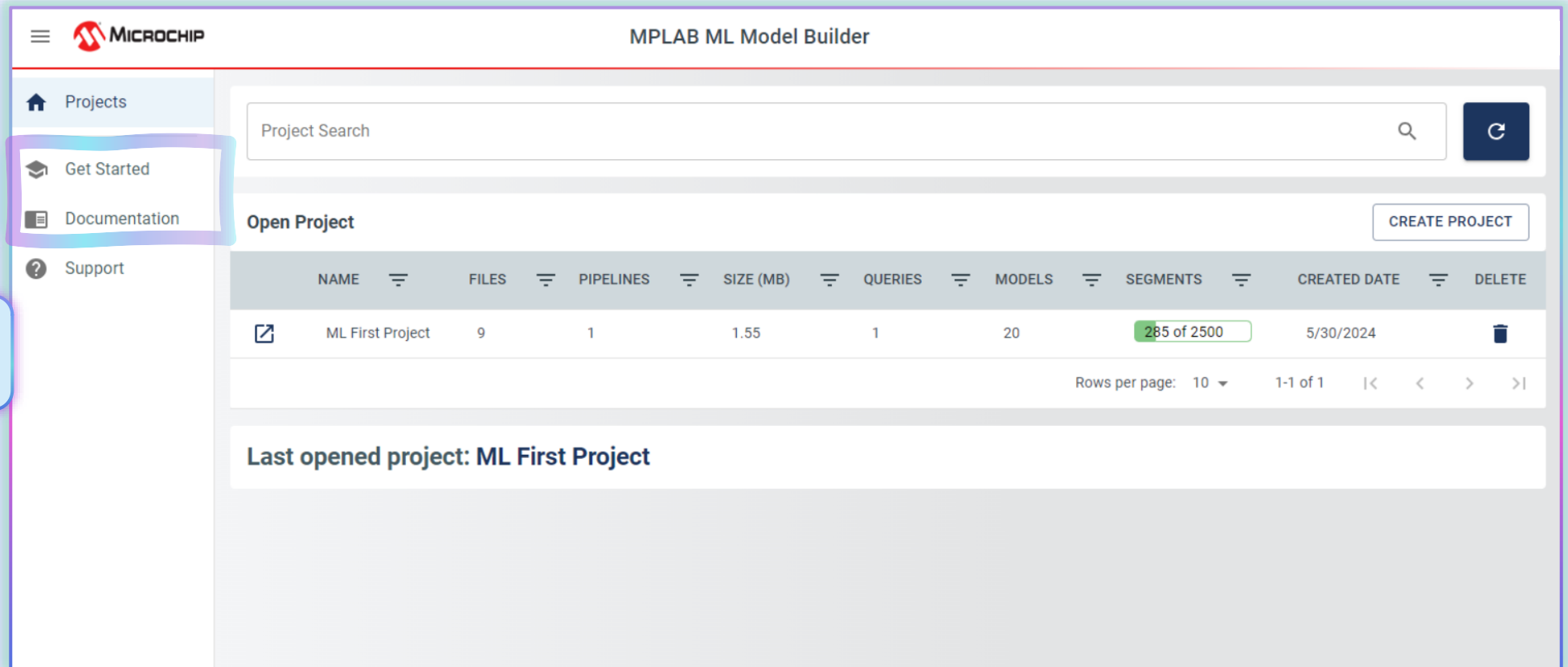
Licensing Information for MPLAB ML

Licensing Information for MPLAB Machine Learning Development Suite

	Evaluation License	Standard License	Pro License
Price	Free	\$29/month	\$1990/year
Supported Silicon	8-, 16- and 32-bit MCUs	8-, 16- and 32-bit MCUs	8-, 16- and 32-bit MCUs
Data	1 GB	Up to 10 GB	Up to 10 GB
Labels	2500	Unlimited	Unlimited
Output	Library	Library	Source Code
AutoML CPU Time	5H/month	10H/month	250H
Application	Evaluation Only	Production	Production

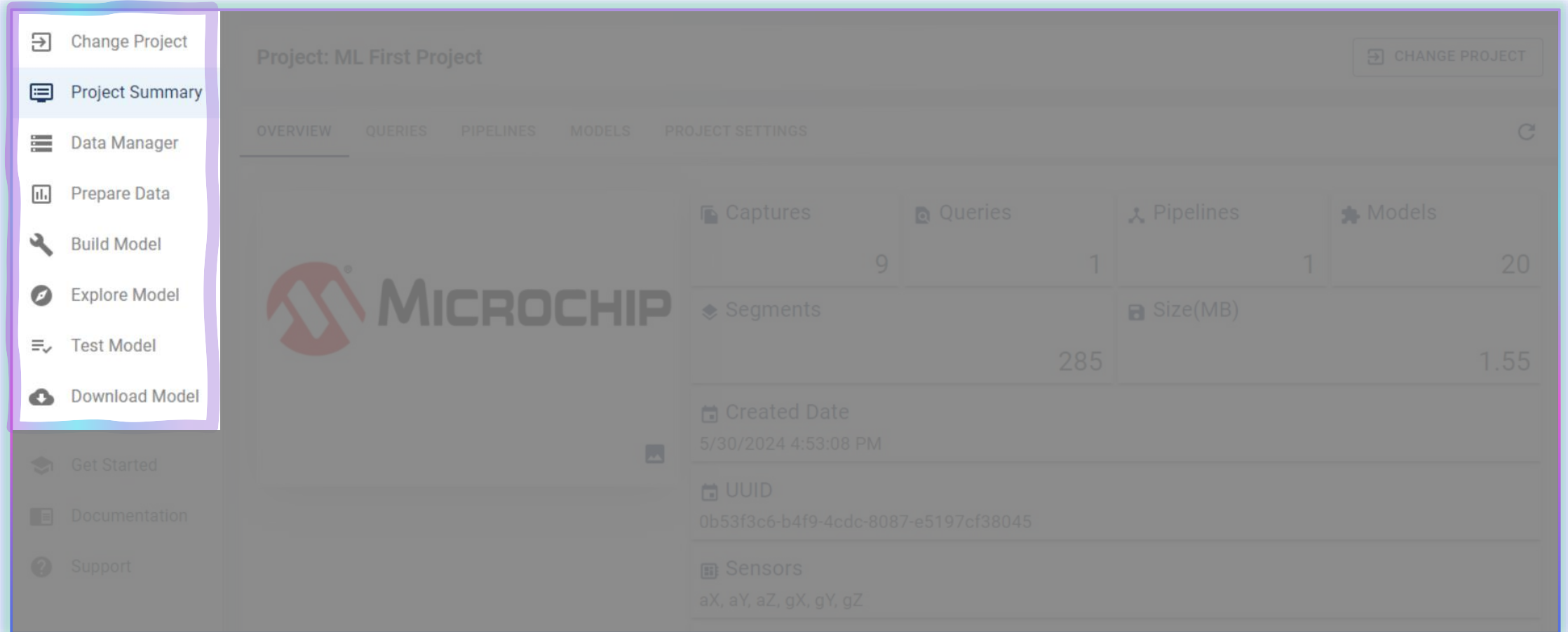
The Workspace of ML Development suite

- Home Page of ML Development suite



The Workspace of ML Development suite

- Each step of the ML flow process.



The Workspace of ML Development suite

- The Overview, Queries, Pipelines, Models, and Project Settings of the project.

The screenshot displays the Microchip ML Development suite workspace. On the left is a sidebar with navigation options: Change Project, Project Summary, Data Manager, Prepare Data, Build Model, Explore Model, Test Model, Download Model, Get Started, Documentation, and Support. The main area is titled 'Project: ML - First Project' and features a 'CHANGE PROJECT' button. Below the title are tabs for Overview, Queries, Pipelines, Models, and Project Settings. The Overview tab is active, showing a summary of project components: 9 Captures, 1 Query, 1 Pipeline, and 20 Models. It also displays 285 Segments and a total size of 1.55 MB. The interface includes the Microchip logo and a 'Created Date' of 5/30/2024 4:53:08 PM. The UUID is 0b53f3c6-b4f9-4cdc-8087-e5197cf38045, and the Sensors are listed as aX, aY, aZ, gX, gY, gZ.

Component	Count
Captures	9
Queries	1
Pipelines	1
Models	20

Segment	Value
Segments	285
Size(MB)	1.55

Created Date
5/30/2024 4:53:08 PM

UUID
0b53f3c6-b4f9-4cdc-8087-e5197cf38045

Sensors
aX, aY, aZ, gX, gY, gZ

Lab2:

Organize - Label and Prepare Data

Lab2 Label and Prepare Data

- **View the project on ML development suite.**
- **Add 4 labels in the project.**
 - Straight
 - Uppercut
 - Righthook
 - Freeze
- **Label all the data.**

Let's go!

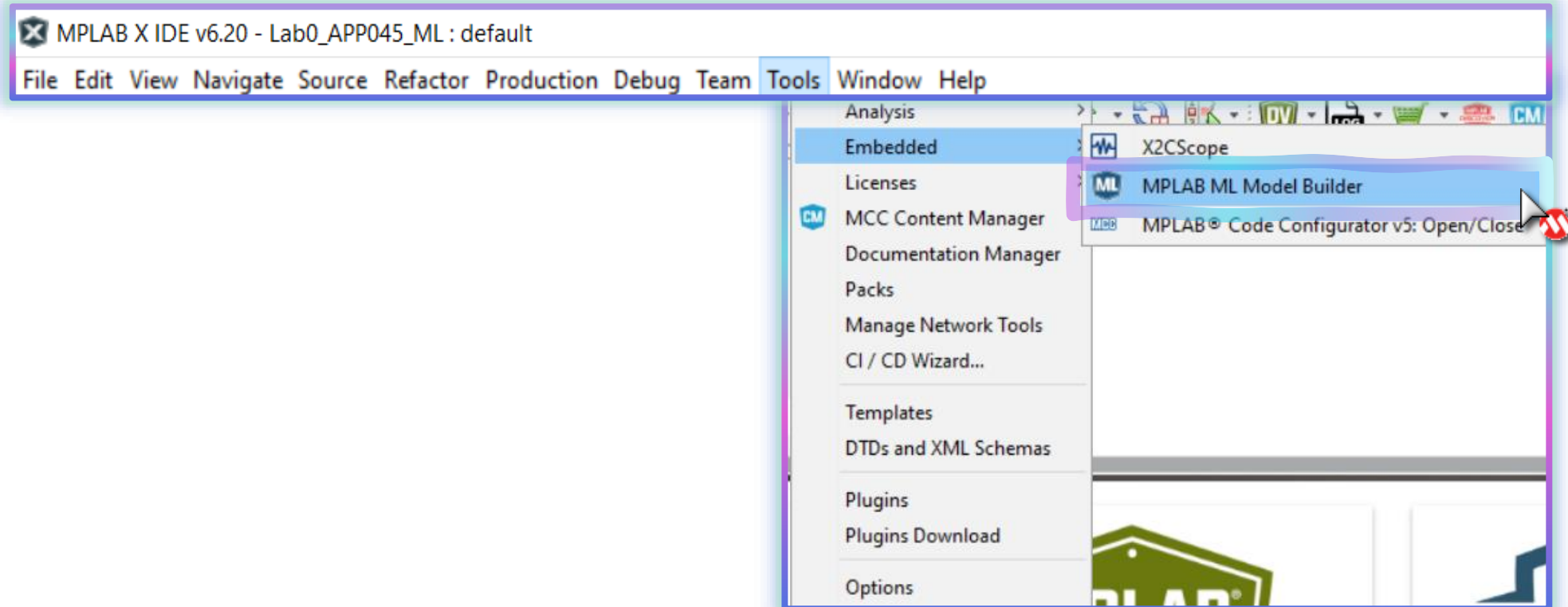
Lab2 Label and Prepare Data

Step 1

- Click  icon

Or

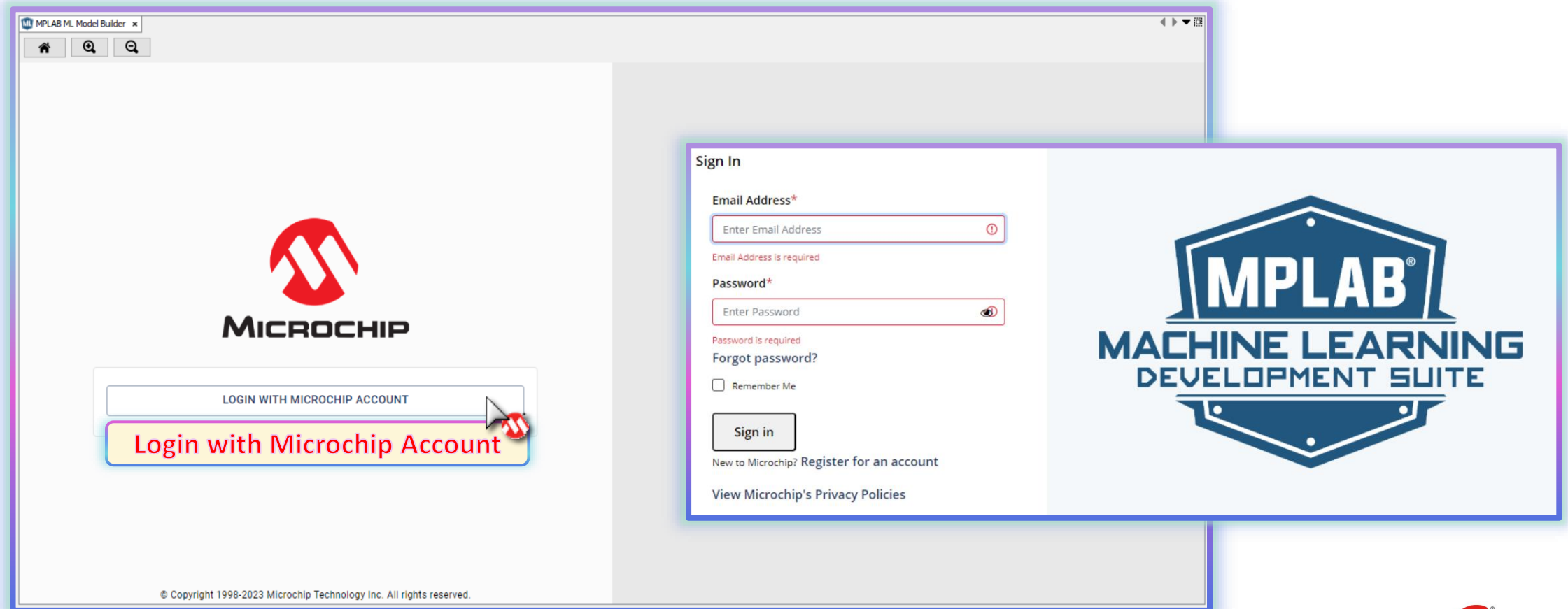
- Tools > Embedded > **MPLAB ML Model Builder**



Lab2 Label and Prepare Data

Step 2

- Log in with your **myMicrochip** account.



The screenshot displays the MPLAB ML Model Builder interface. On the left, the Microchip logo is prominent, with a button labeled "Login with Microchip Account" highlighted by a red box and a cursor. On the right, a "Sign In" modal is open, containing the following elements:

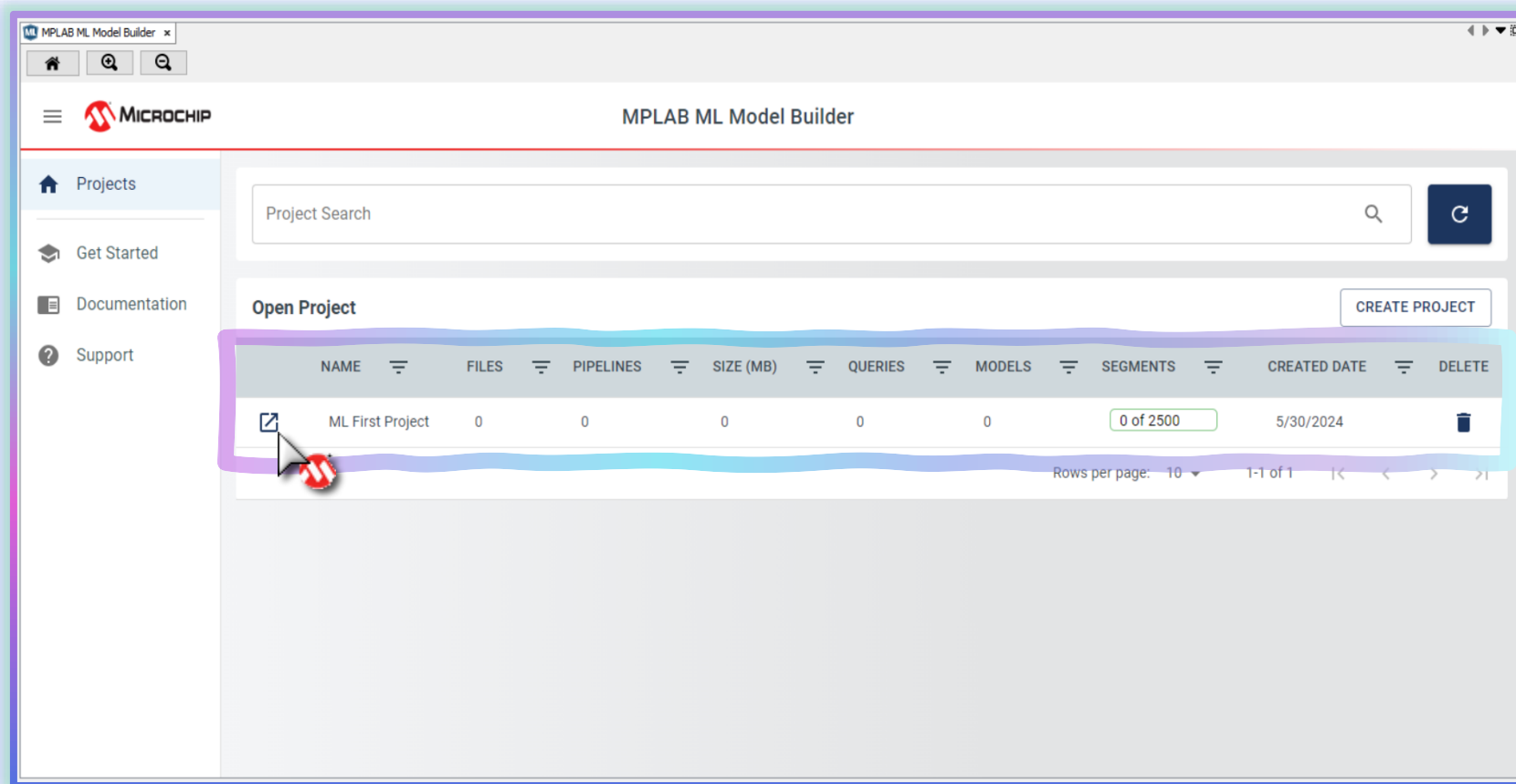
- Email Address***: A text input field with the placeholder "Enter Email Address" and a red error message "Email Address is required" below it.
- Password***: A text input field with the placeholder "Enter Password" and a red error message "Password is required" below it.
- Forgot password?**: A link for password recovery.
- Remember Me**: A checkbox.
- Sign in**: A button to submit the login credentials.
- New to Microchip? Register for an account**: A link for new users.
- View Microchip's Privacy Policies**: A link to the privacy policy.

The background of the modal features the "MPLAB MACHINE LEARNING DEVELOPMENT SUITE" logo. At the bottom of the main window, the copyright notice "© Copyright 1998-2023 Microchip Technology Inc. All rights reserved." is visible.

Lab2 Label and Prepare Data

Step 3

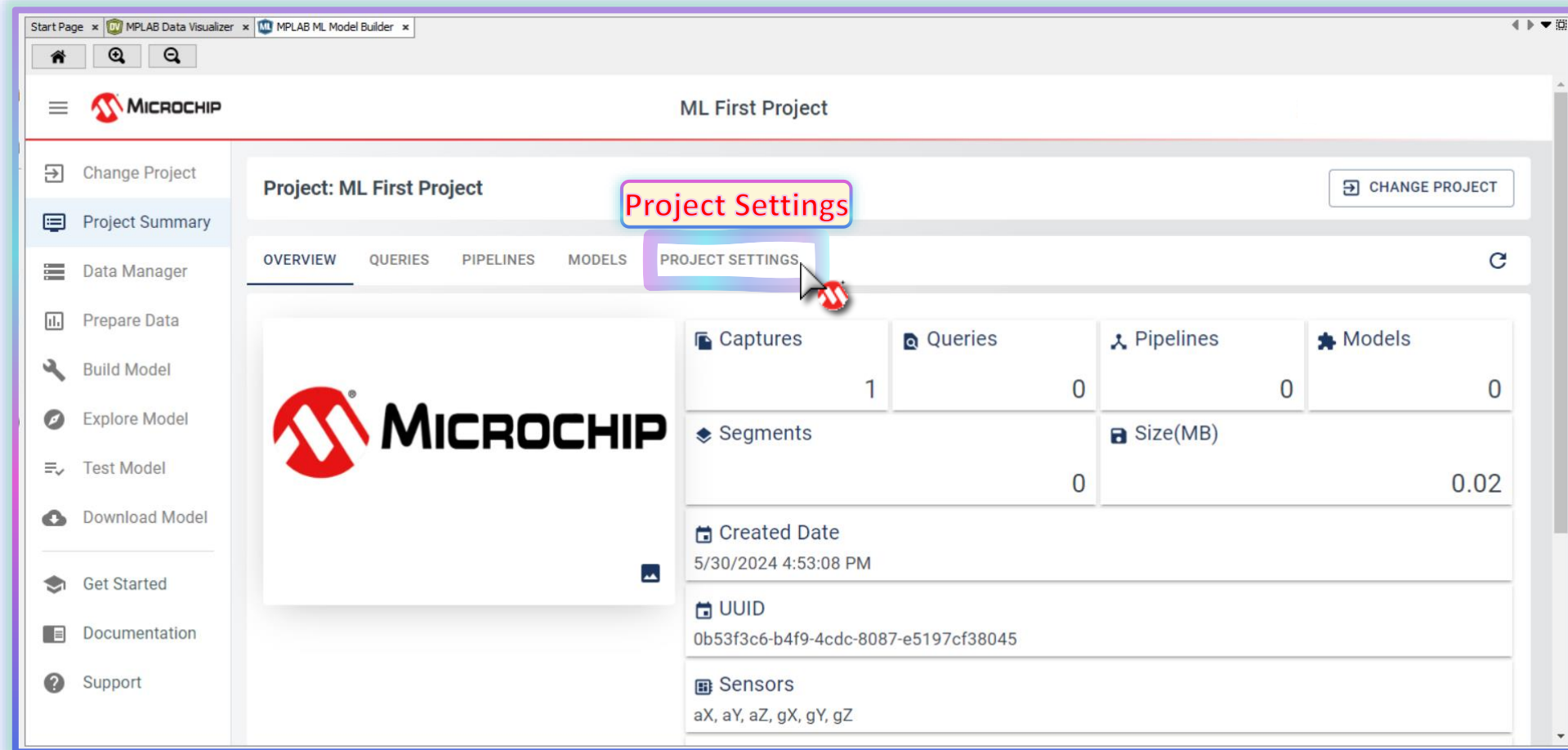
- The project we built will show on the welcome page.
- Click  icon to open **ML First Project**.



Lab2 Label and Prepare Data

Step 4

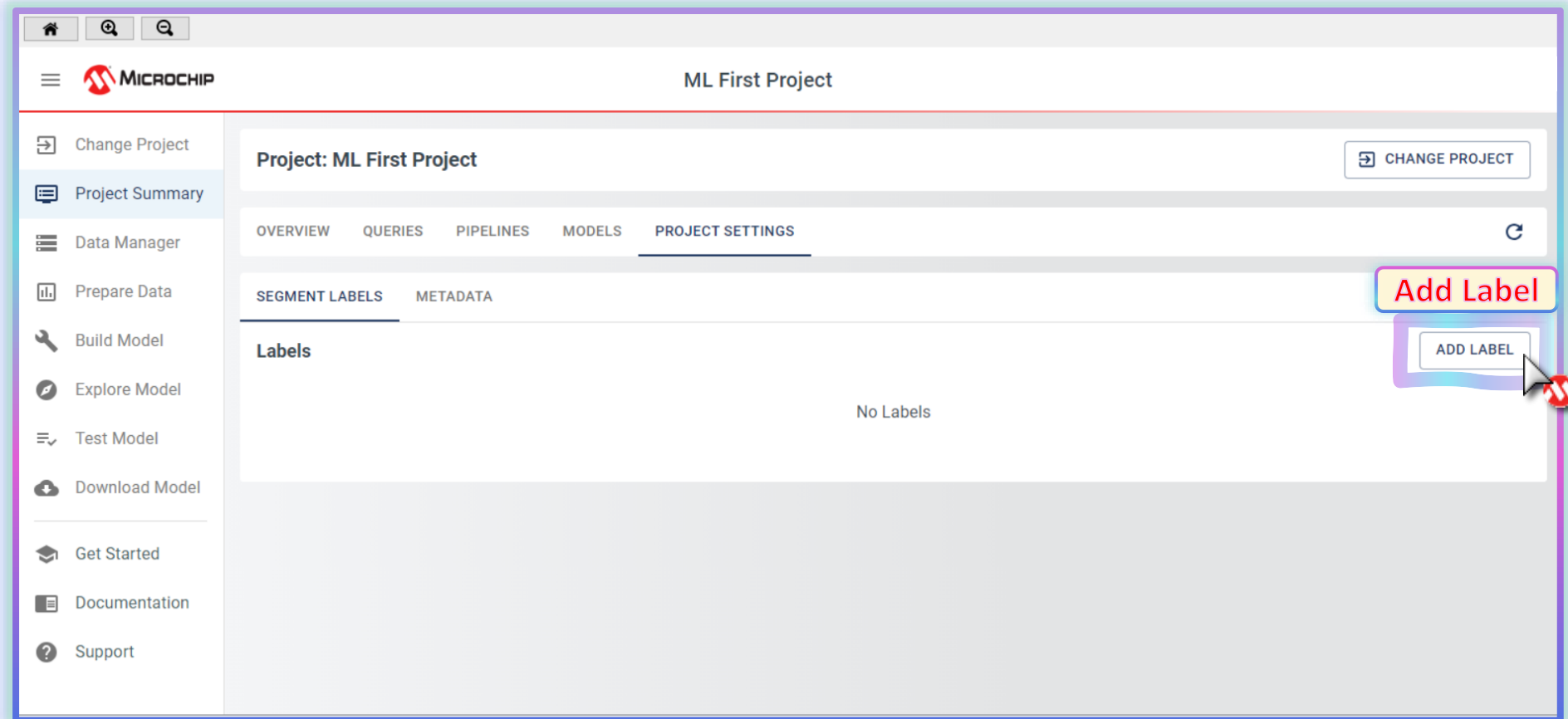
- Click **Project Settings**.



Lab2 Label and Prepare Data

Step 5

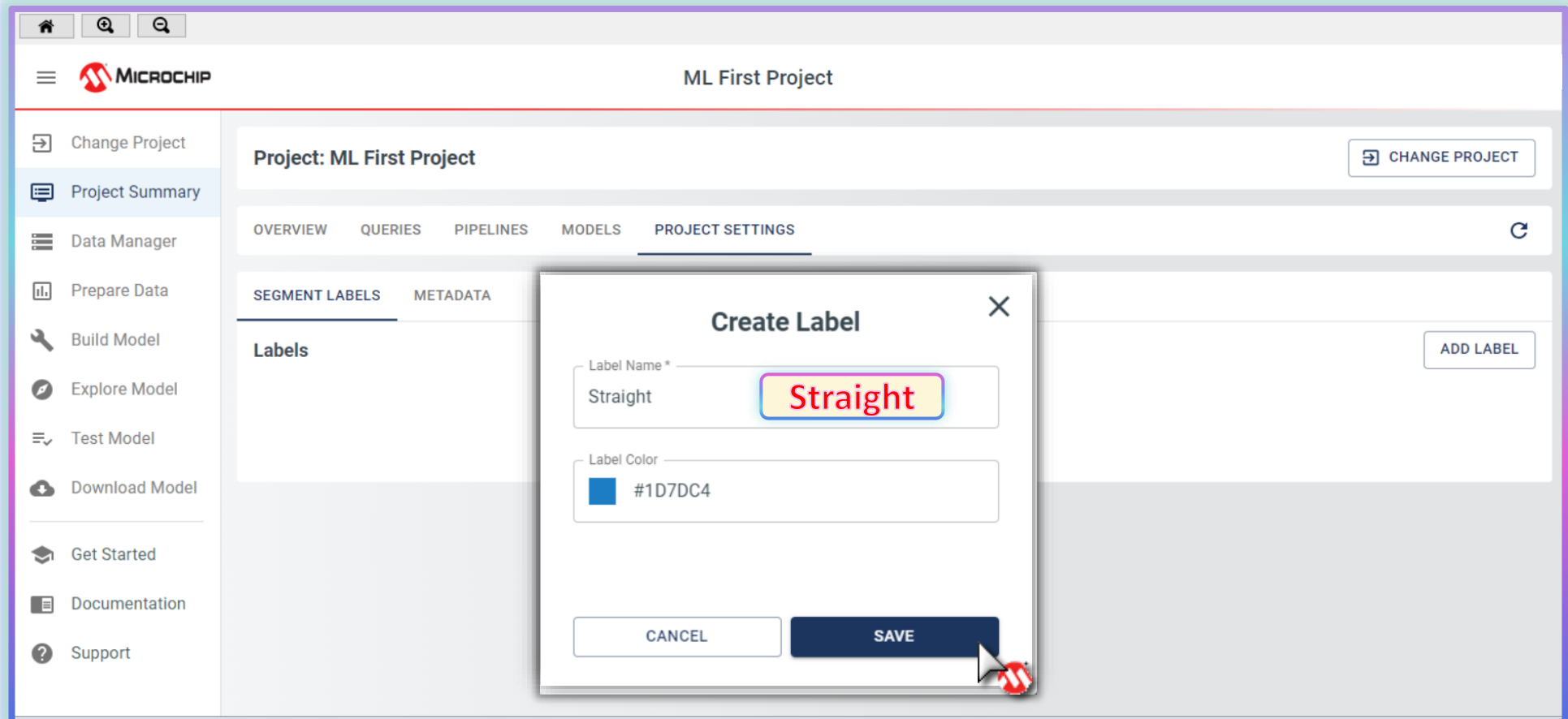
- Add Label Name.



Lab2 Label and Prepare Data

Step 6

- Add Label Name.



Lab2 Label and Prepare Data

Step 9

- Repeat the **step 5 to step 6** to add segment label. Make sure that the Label color for each label is different.
 - Straight
 - Uppercut
 - Righthook
 - Freeze
- (Optional) Repeat the step 7 to step 8 to add Metadata.

Lab2 Label and Prepare Data

Result

- Should show **4 Label** with different color after finished step 9.

Project: ML First Project

CHANGE PROJECT

OVERVIEWQUERIESPIPELINESMODELSPROJECT SETTINGS

SEGMENT LABELSMETADATA

Labels

ADD LABEL

NAME	CREATED DATE	UPDATED DATE	EDIT	DELETE
Straight	7/15/2024 5:01:34 PM	7/15/2024 5:01:34 PM		
Uppercut	7/15/2024 5:01:50 PM	7/15/2024 5:01:50 PM		
Righthook	7/15/2024 5:02:01 PM	7/15/2024 5:02:01 PM		
Freeze	7/16/2024 2:09:45 PM	7/16/2024 2:09:45 PM		

Straight

Uppercut

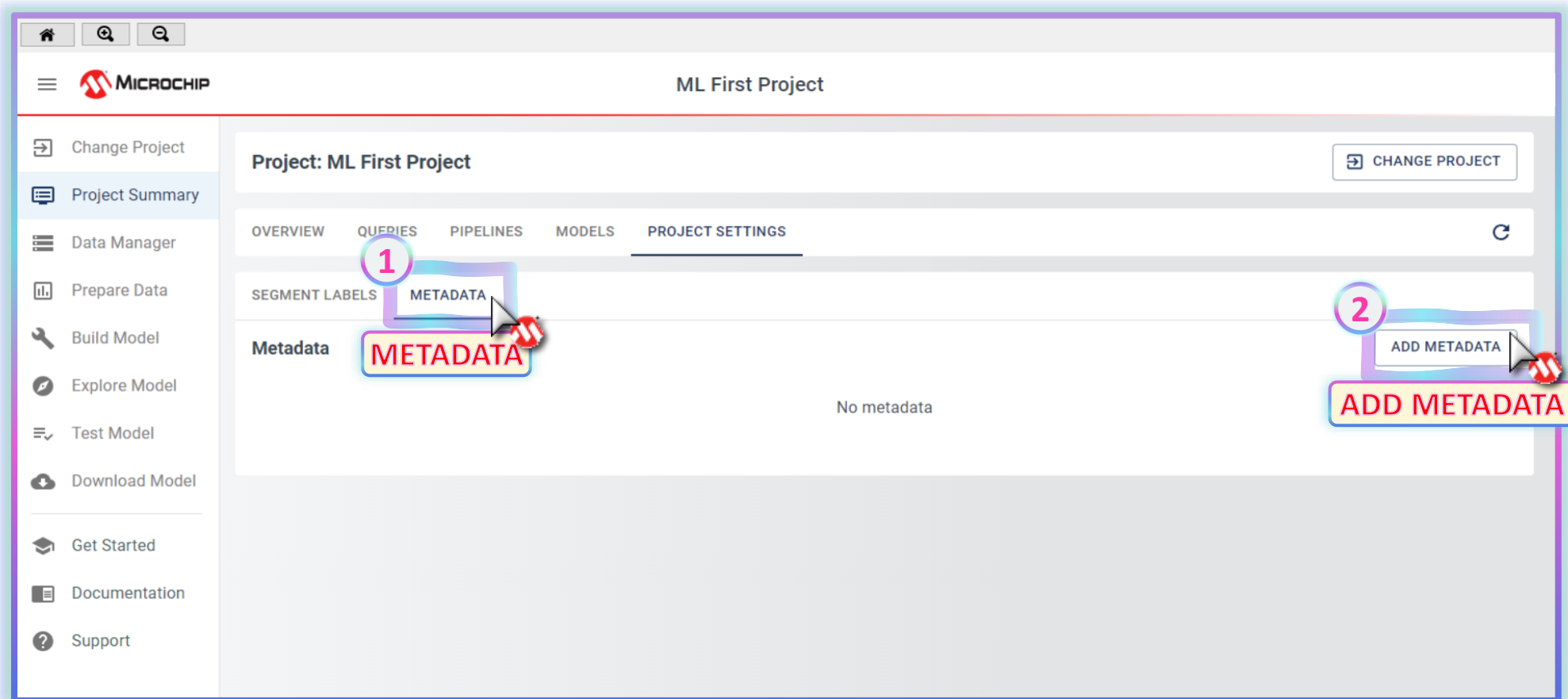
Righthook

Freeze

Lab2 Label and Prepare Data

Step 7 (Optional)

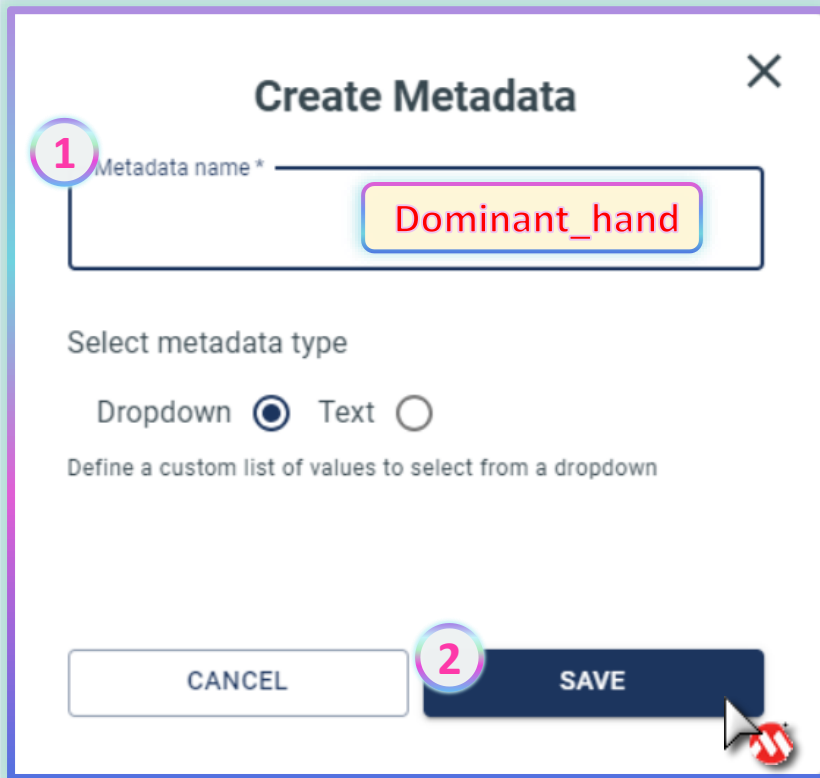
- Add Metadata.



Lab2 Label and Prepare Data

Step 8 (Optional)

- Add Metadata and Metadata values.



Create Metadata [X]

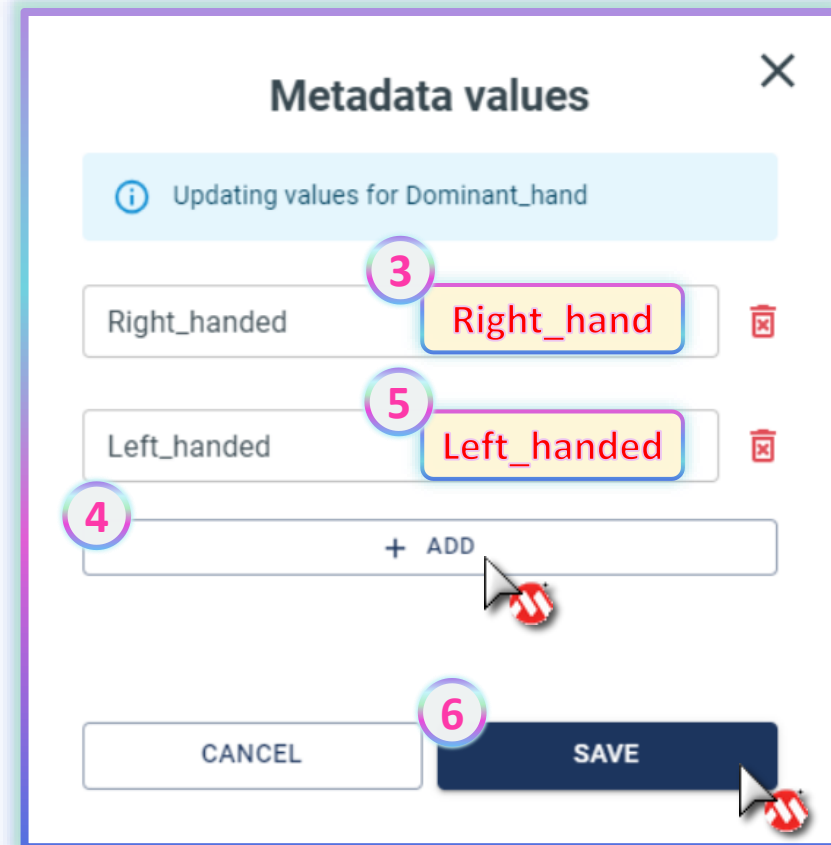
1 Metadata name *

Select metadata type

Dropdown ☒ Text ☐

Define a custom list of values to select from a dropdown

CANCEL 2 SAVE



Metadata values [X]

i Updating values for Dominant_hand

3 Right_handed [X]

5 Left_handed [X]

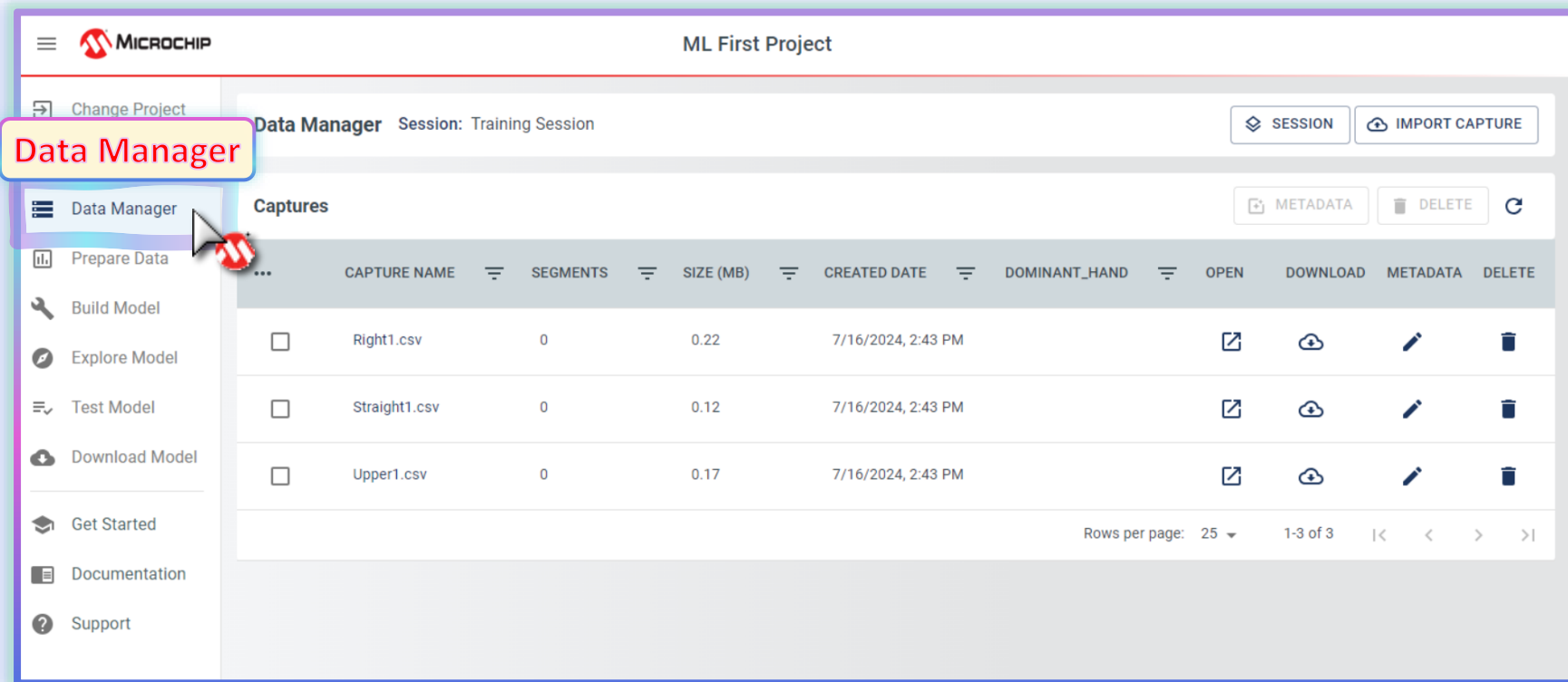
4 + ADD

CANCEL 6 SAVE

Lab2 Label and Prepare Data

Step 10

- Click **Data Manager**. The collected data files can be found in the Data Manager page.



The screenshot shows the Microchip ML First Project interface. The left sidebar contains a menu with the following items: Change Project, Data Manager (highlighted with a red box), Prepare Data, Build Model, Explore Model, Test Model, Download Model, Get Started, Documentation, and Support. The main area is titled 'Data Manager' and 'Session: Training Session'. It features a table of captures with the following columns: CAPTURE NAME, SEGMENTS, SIZE (MB), CREATED DATE, DOMINANT_HAND, OPEN, DOWNLOAD, METADATA, and DELETE. The table contains three rows of data:

	CAPTURE NAME	SEGMENTS	SIZE (MB)	CREATED DATE	DOMINANT_HAND	OPEN	DOWNLOAD	METADATA	DELETE
☐	Right1.csv	0	0.22	7/16/2024, 2:43 PM					
☐	Straight1.csv	0	0.12	7/16/2024, 2:43 PM					
☐	Upper1.csv	0	0.17	7/16/2024, 2:43 PM					

At the bottom right of the table, it says 'Rows per page: 25' and '1-3 of 3'.

Lab2 Label and Prepare Data

Step 11

- Import the file that have captured before.
- Click **IMPORT CAPTURE**

Microchip ML First Project

IMPORT CAPTURE

Data Manager Session: Training Session

Captures

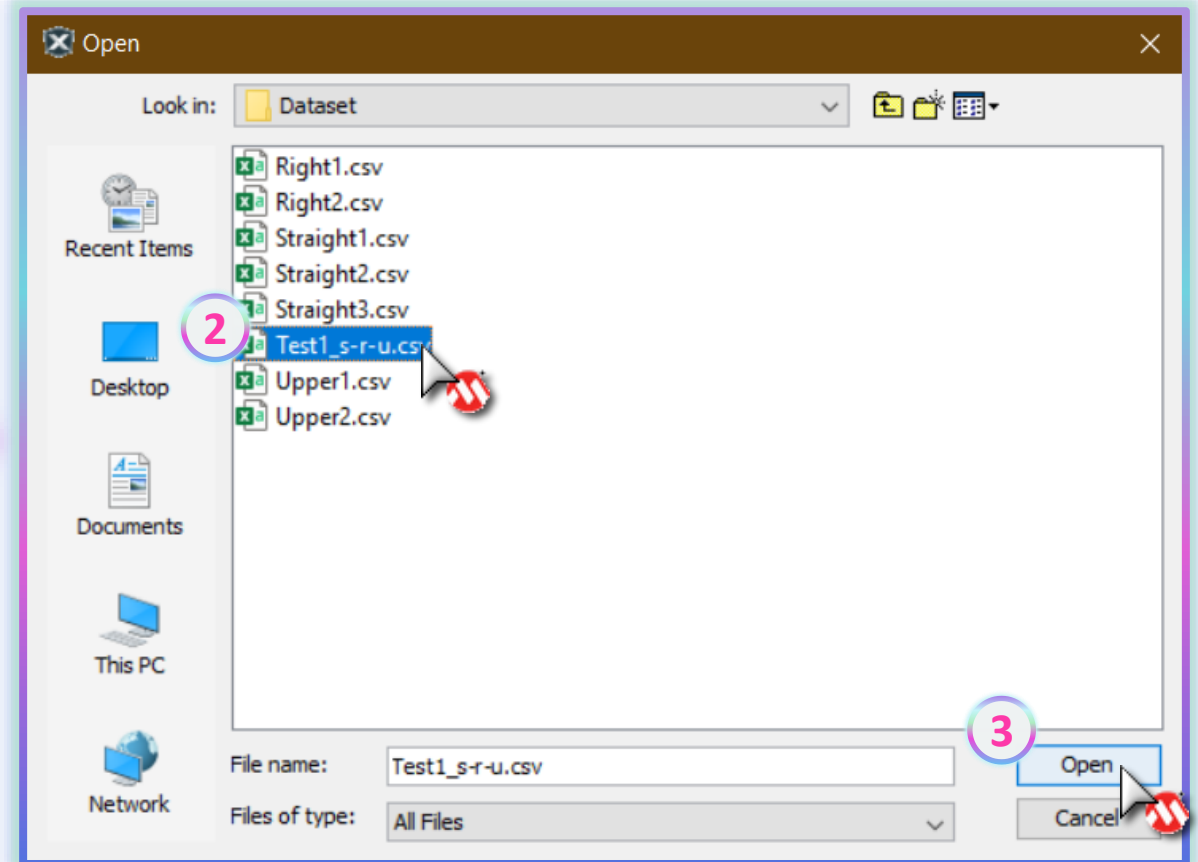
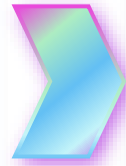
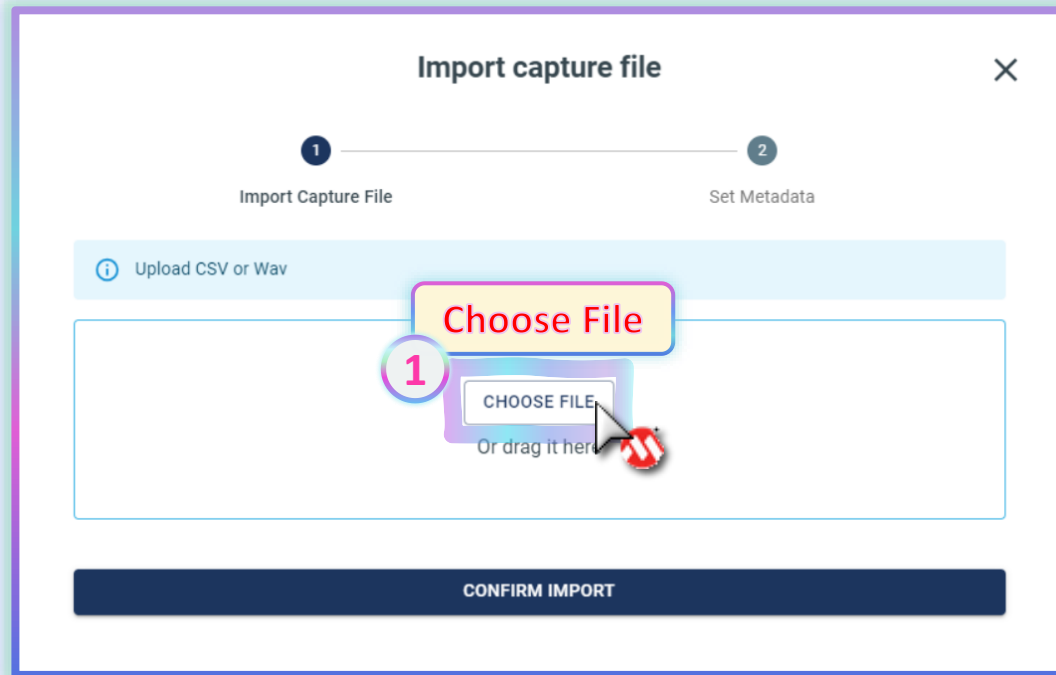
...	CAPTURE NAME	SEGMENTS	SIZE (MB)	CREATED DATE	DOMINANT_HAND	OPEN	DOWNLOAD	METADATA	DELETE
☐	Right1.csv	0	0.22	7/16/2024, 2:43 PM					
☐	Straight1.csv	0	0.12	7/16/2024, 2:43 PM					
☐	Upper1.csv	0	0.17	7/16/2024, 2:43 PM					

Rows per page: 25 1-3 of 3 |< < > >|

Lab2 Label and Prepare Data

Step 12

- Choose the file you would like to import.



Lab2 Label and Prepare Data

Step 13 (Optional)

- Confirm Import -> Set the **metadata**. (Leave it blank if doesn't want to set metadata)

Import capture file

1 Import Capture File 2 Set Metadata

Upload CSV or Wav

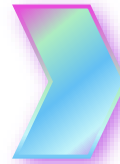
Test2_s-r-u.csv

CHOOSE FILE

Or drag it here

1 Confirm import

CONFIRM IMPORT



Import capture file

Import Capture File Set Metadata

Files uploaded successfully. Now you can set metadata.

Sensor Configuration

Dominant_hand

2 Set Metadata

SET METADATA

Lab2 Label and Prepare Data

Step 14

- Let's start labeling our data, which also means define segments.
- Open **Straight1.csv** data.

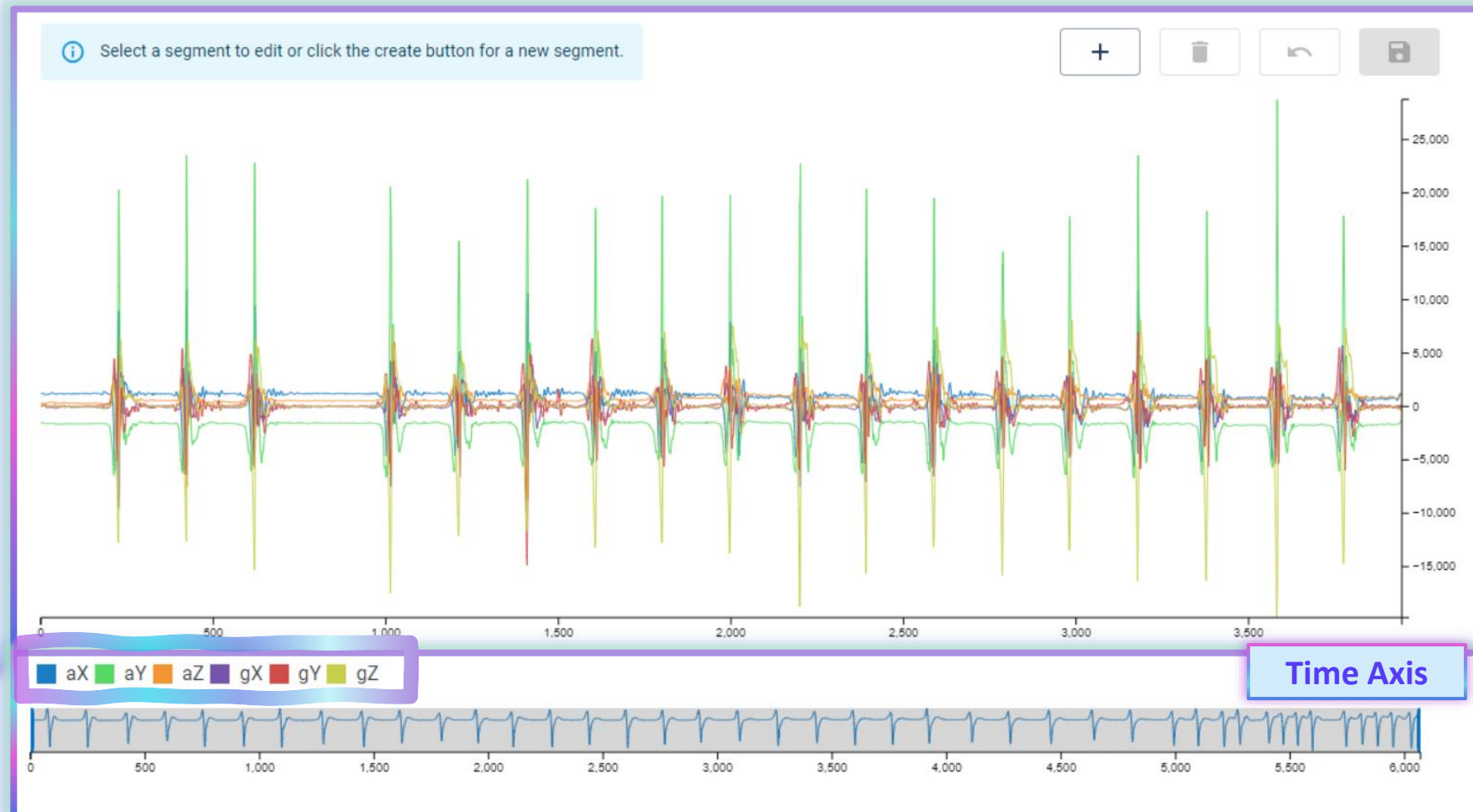
The screenshot displays the Microchip ML First Project Data Manager interface. The left sidebar contains navigation options: Change Project, Project Summary, Data Manager (selected), Prepare Data, Build Model, Explore Model, Test, Download Model, Get Started, Documentation, and Support. The main area shows the Data Manager for the Training Session. It lists three captures: Right1.csv, Straight1.csv, and Upper1.csv. The 'Straight1.csv' row is highlighted with a blue box, and a red circle with a white 'X' is placed over the 'OPEN' icon for this row. The table columns are: CAPTURE NAME, SEGMENTS, SIZE (MB), CREATED DATE, DOMINANT_HAND, OPEN, DOWNLOAD, METADATA, and DELETE. The bottom right shows 'Rows per page: 25' and '1-3 of 3'.

...	CAPTURE NAME	SEGMENTS	SIZE (MB)	CREATED DATE	DOMINANT_HAND	OPEN	DOWNLOAD	METADATA	DELETE
☐	Right1.csv	0	0.22	7/16/2024, 2:43 PM					
☐	Straight1.csv	0	0.12	7/16/2024, 2:43 PM					
☐	Upper1.csv	0	0.17	7/16/2024, 2:43 PM					

Lab2 Label and Prepare Data

Step 15

- **Segment:** Define where an event is located in your sensor data file.

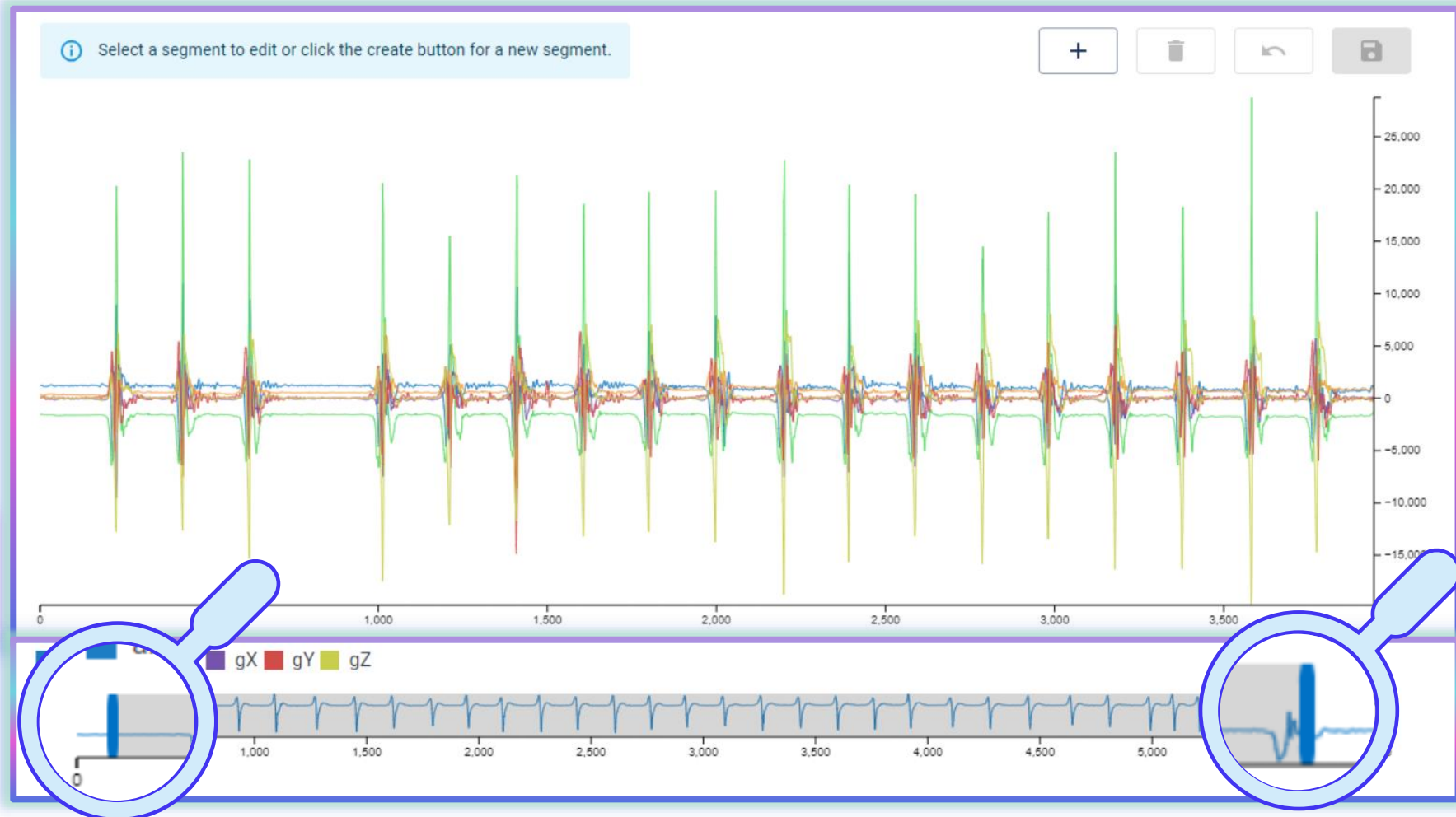


Click to show
or hide the
selected
raw data.

Lab2 Label and Prepare Data

Step 16

- You may zoom in to view the detail of the file.

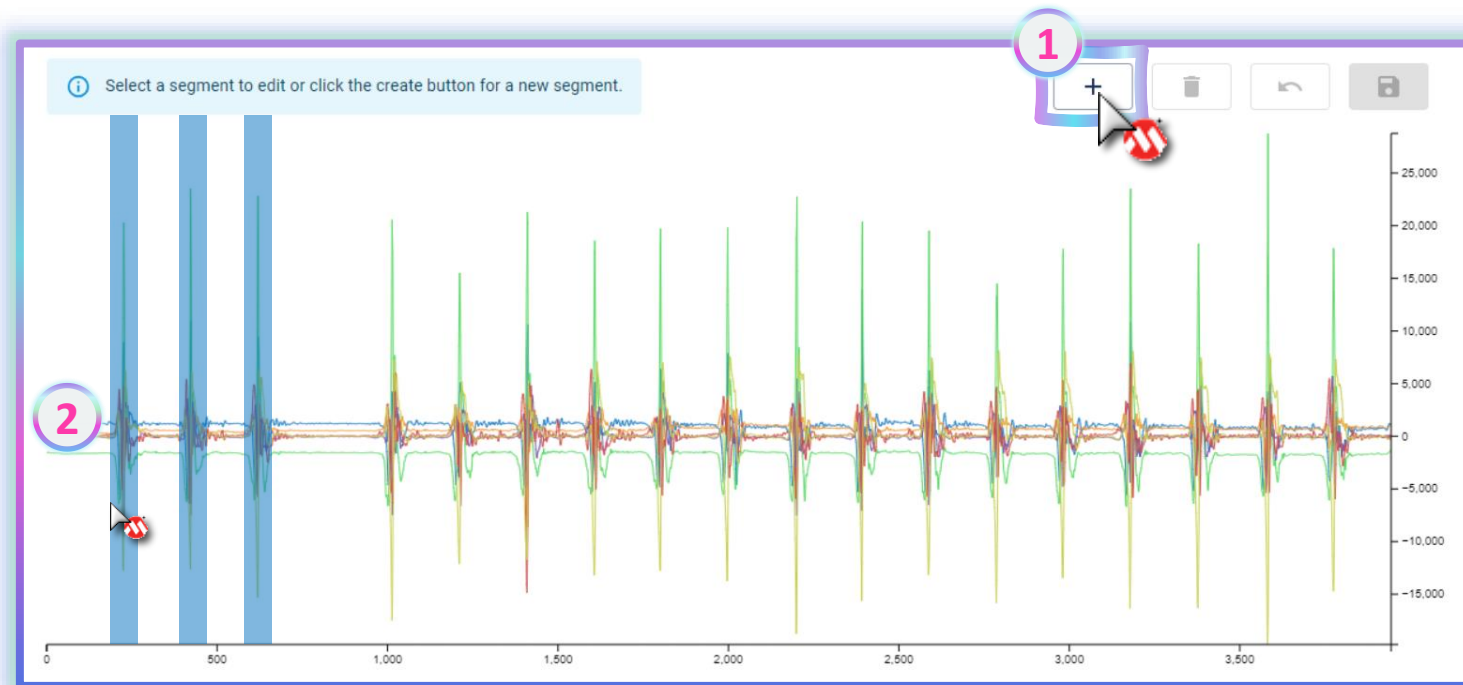


Slide it to
Zoom in the view

Lab2 Label and Prepare Data

Step 17

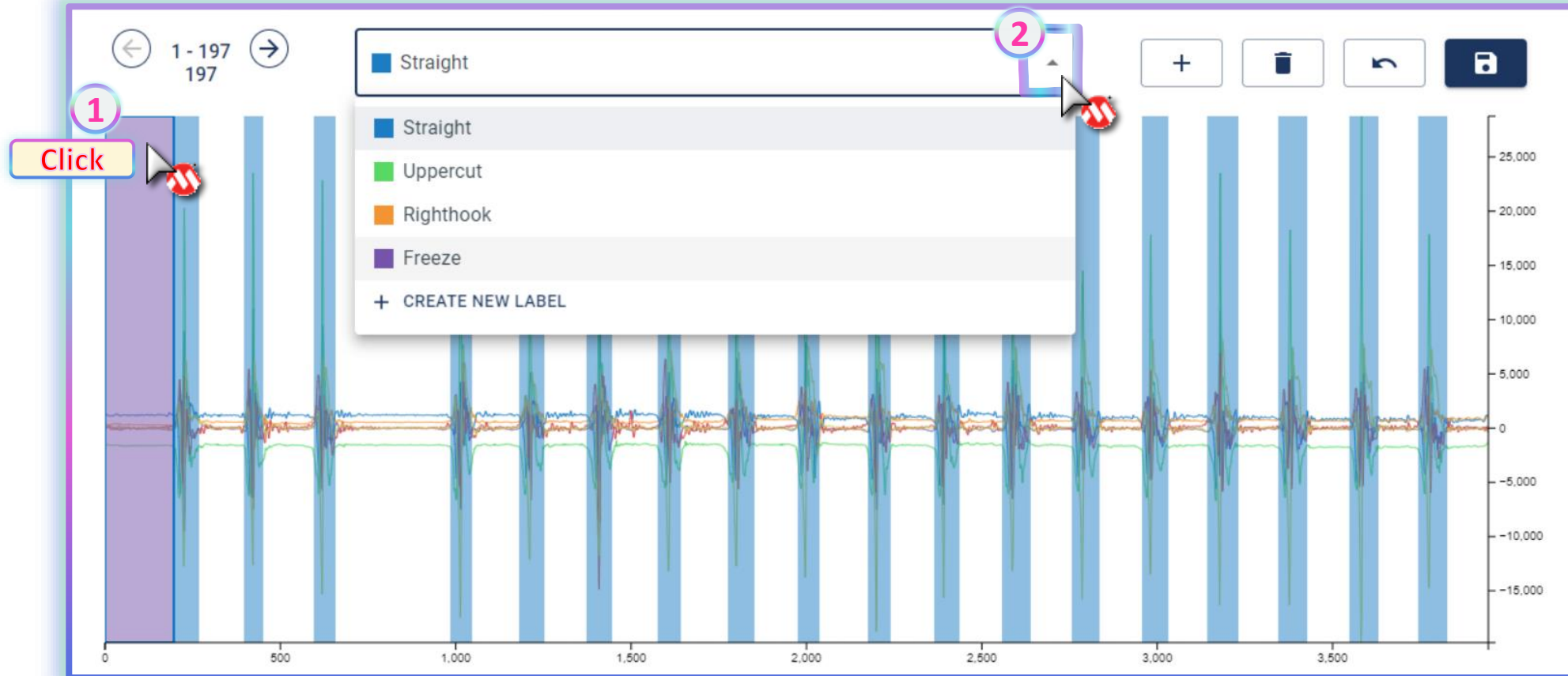
- Click  to add segment.
- Move the cursor over the chart, then click and drag the area that will be marked as an event. This will place a new segment in the file.



Lab2 Label and Prepare Data

Step 18

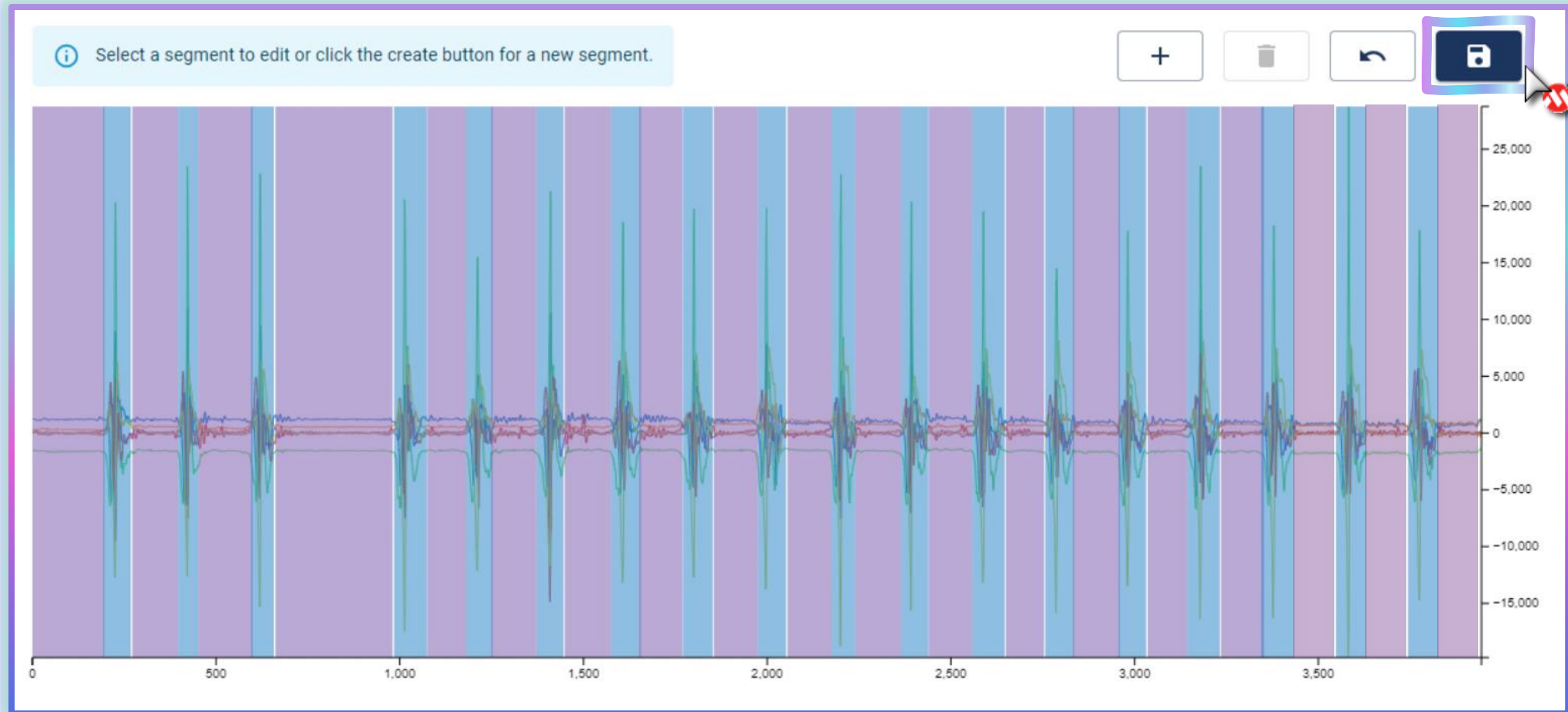
- Click the segment first, then choose the label for the corresponding selection segment.



Lab2 Label and Prepare Data

Step 19

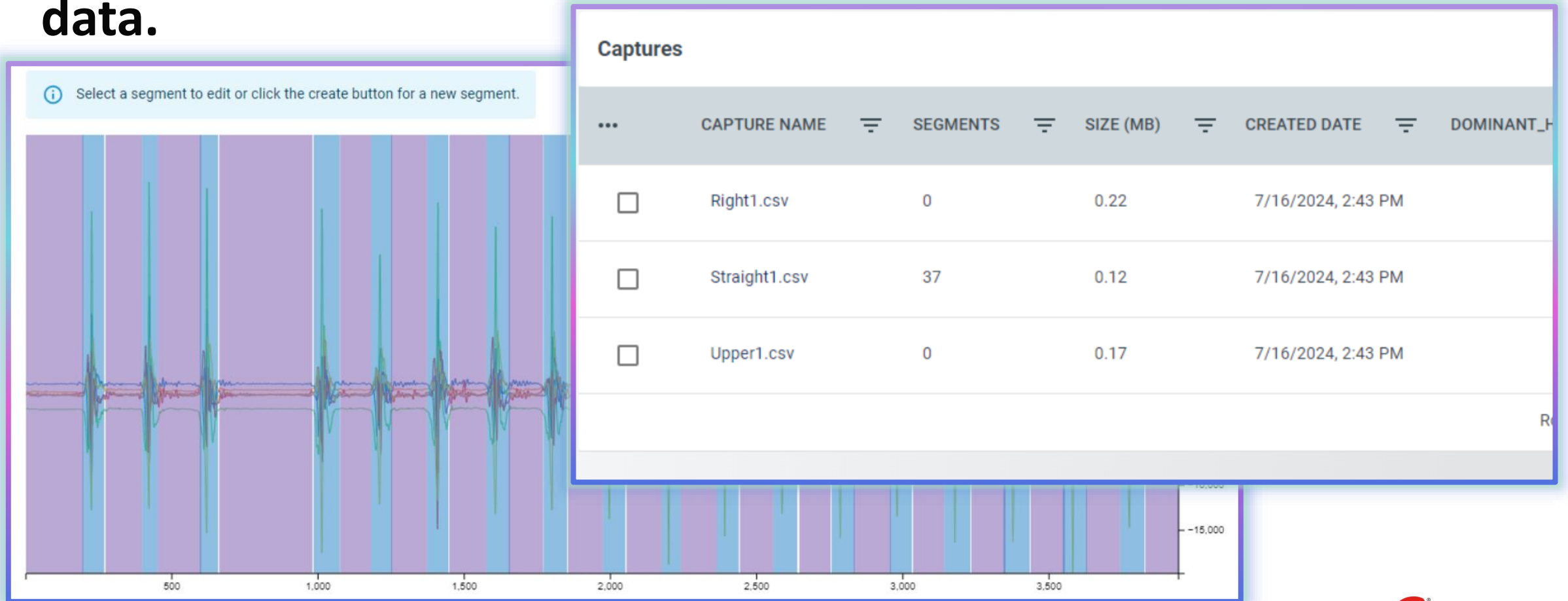
- Finish labeling the segment in each dataset. Click  to save.



Lab2 Label and Prepare Data

Result

- The data manager will show the number of segments in each data.



Lab2 Label and Prepare Data

Precautions

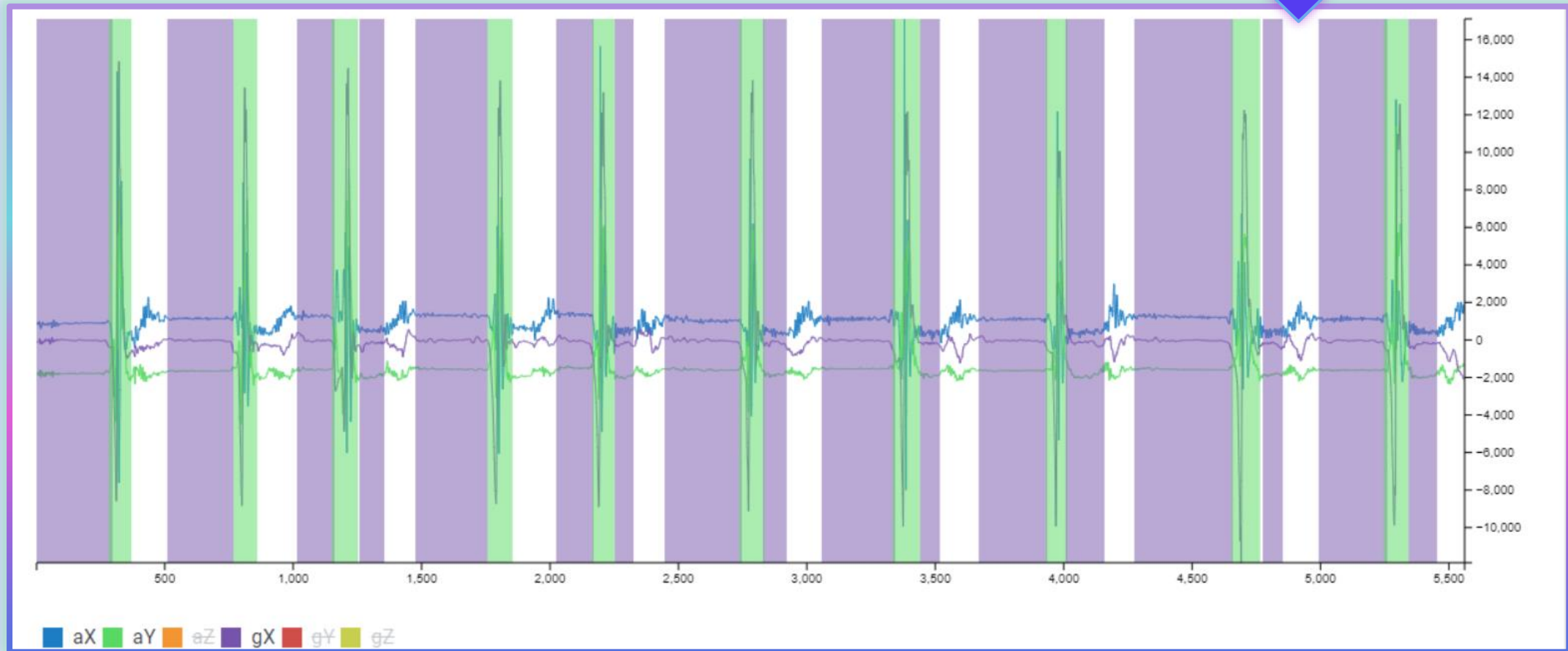
- “Label the data” this step may be the most important but also most boring work during the ML training.
- Observe the aX, aY, aZ, gX, gY, and gZ raw data first, then start labeling data, even slightly differences may impact the training result.
- If the event doesn't belong to any label, keep it empty to avoid confusing training.
- Repeat **step 17 to step 19** to label all the segment and file.

Lab2 Label and Prepare Data

Precautions

- Take **Uppercut1.csv** as example.

Why keep this unlabeled?
Is it also an uppercut?



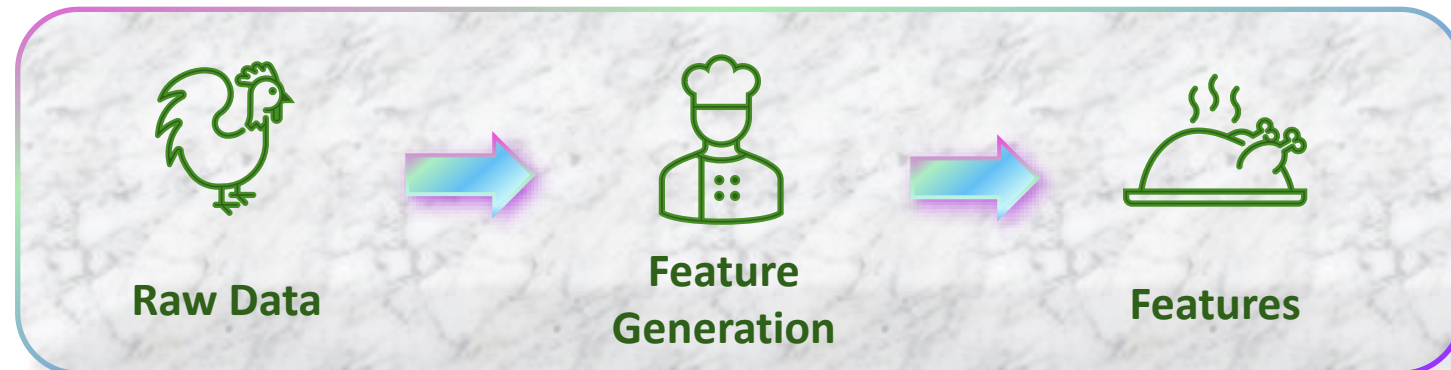
Analyze the data & ML model

Querying the data

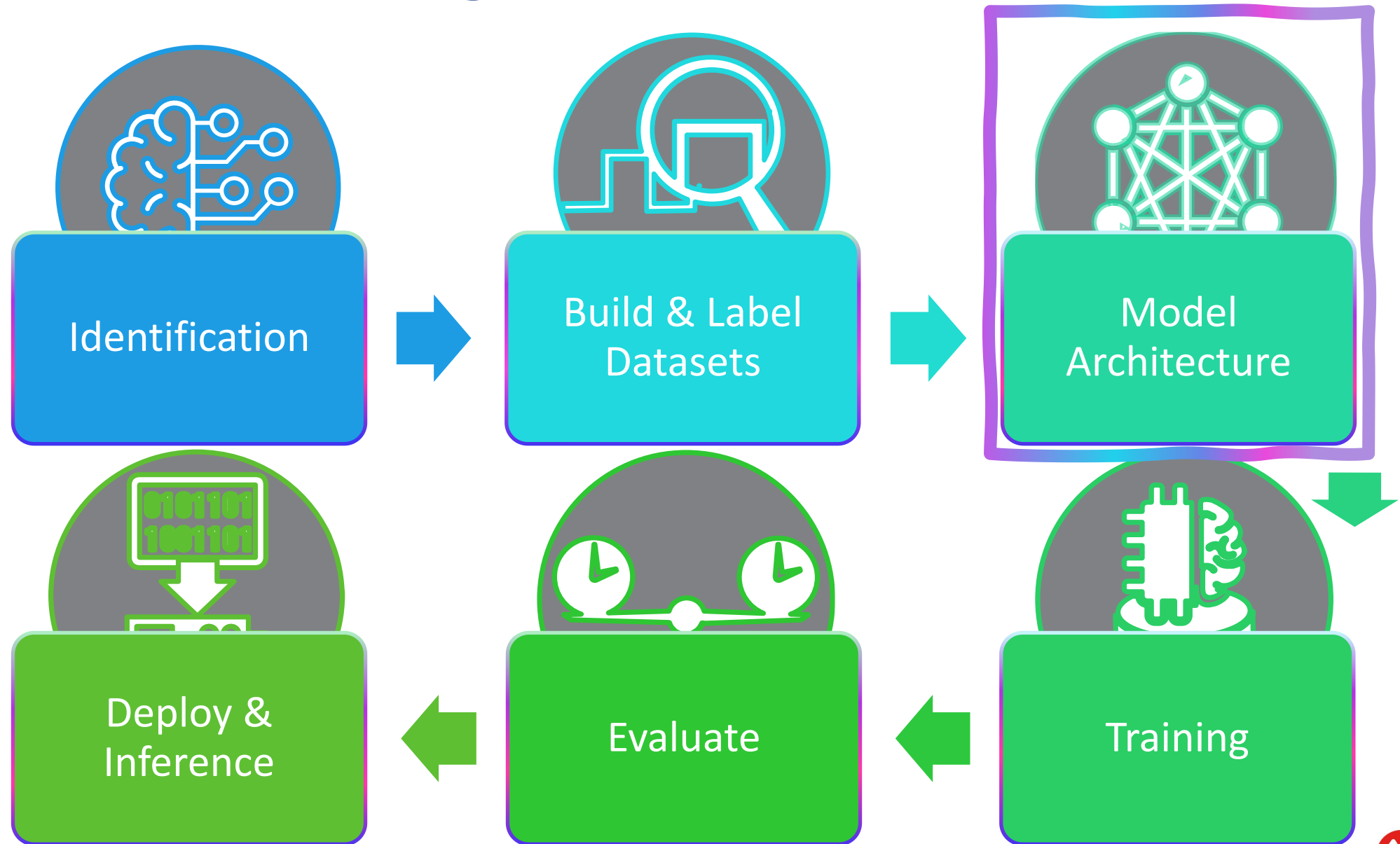
- **Querying data:** The process of extracting data from a database, repository, or other source for use in training a machine learning model.
 - Only a subset may be required
 - Metadata comes in useful
 - Only select samples from right-handed
 - Or samples taken by male
 - Create models better tuned for certain scenarios or markets

Feature Generation

- “More data beats clever algorithms, but better data beats more data,” said Peter Norvig.
- The process of converting raw data into a format that a machine learning model can understand. May include extracting information from the raw data, creating new features, or transforming the data to improve the performance of the model.
 - Which sensor data to use
 - Generate features
 - Select features
 - Classification

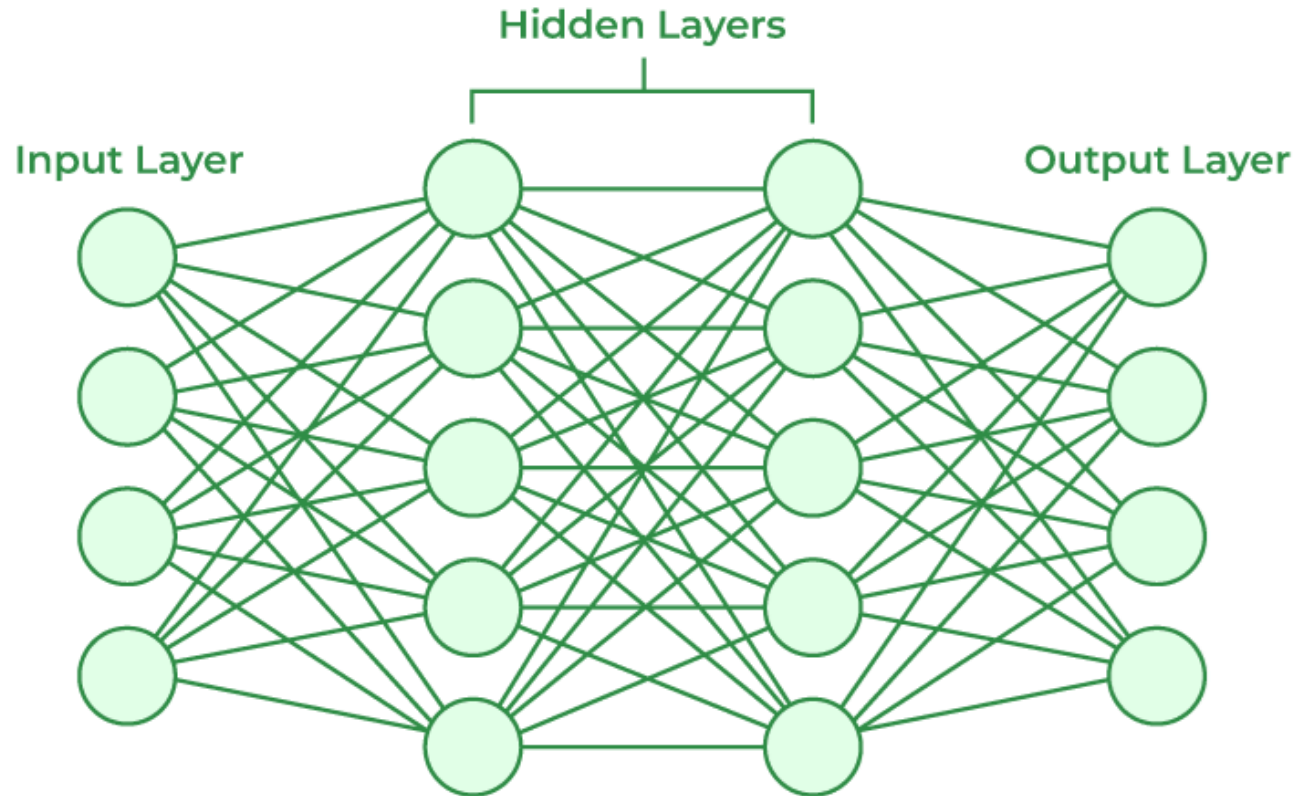


Machine Learning workflow



AI/ML Models

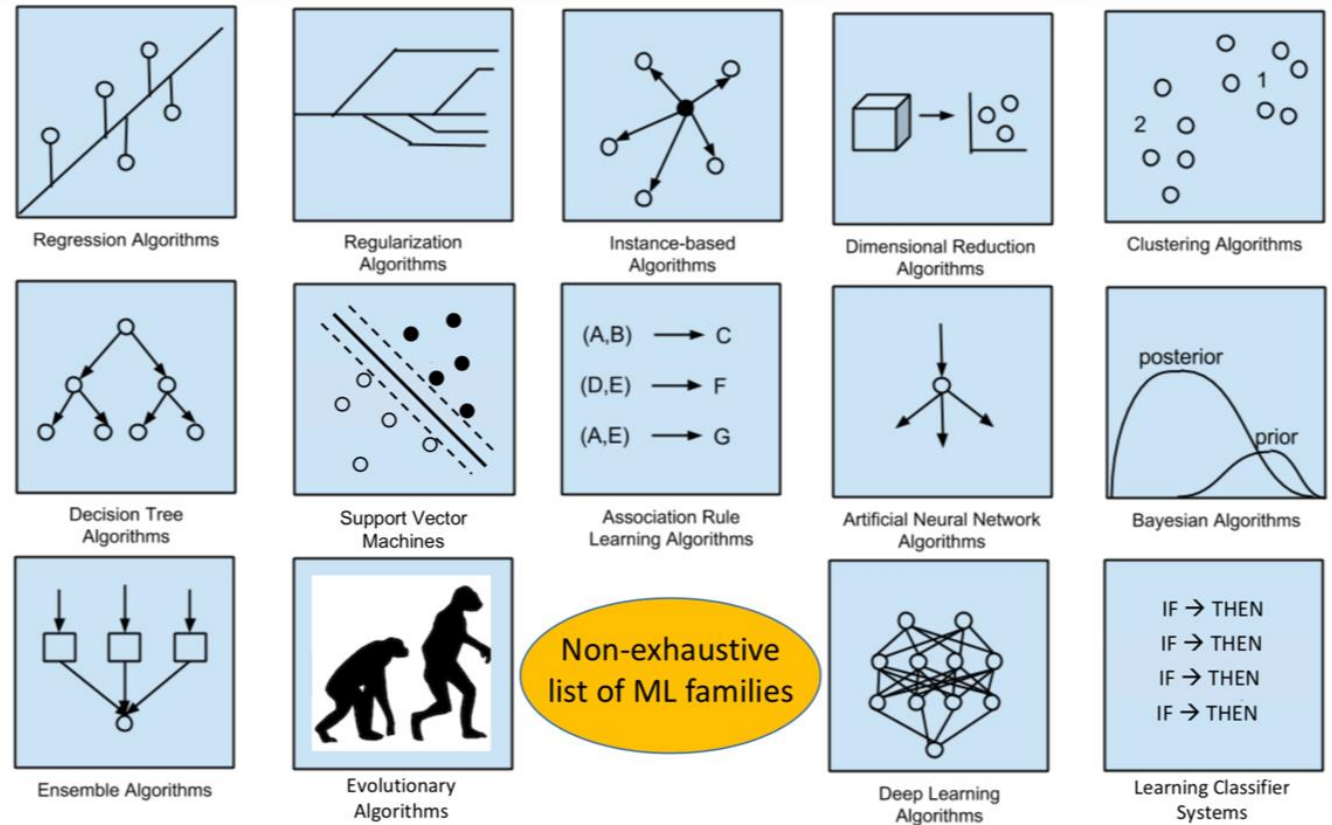
- How does an AI/ML model look like in your mind?



The answer is.....Yes and no.
Not all AIML models are Neural Networks.

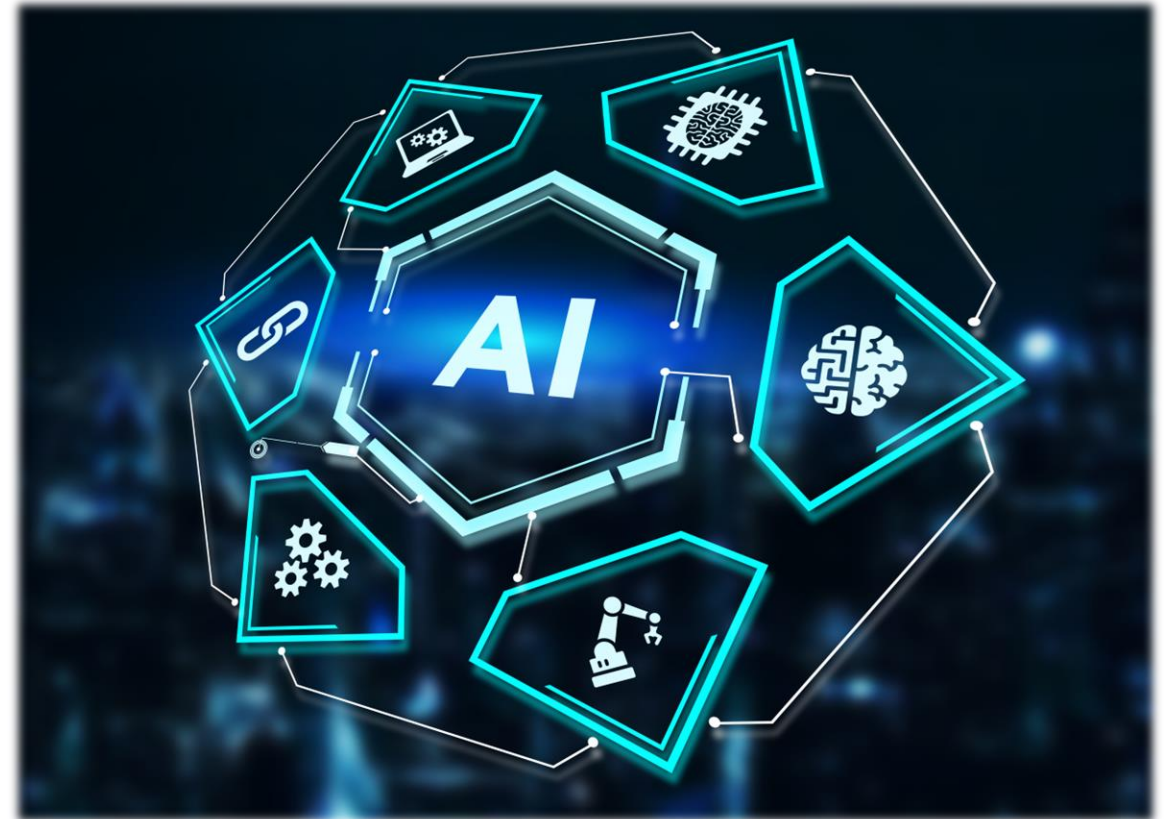
ML Models

- Not all models are neural networks
- To build an application we need to know what type of model to use.
 - Depends on application space
 - Hire a PhD
 - Try using AutoML



Common ML model

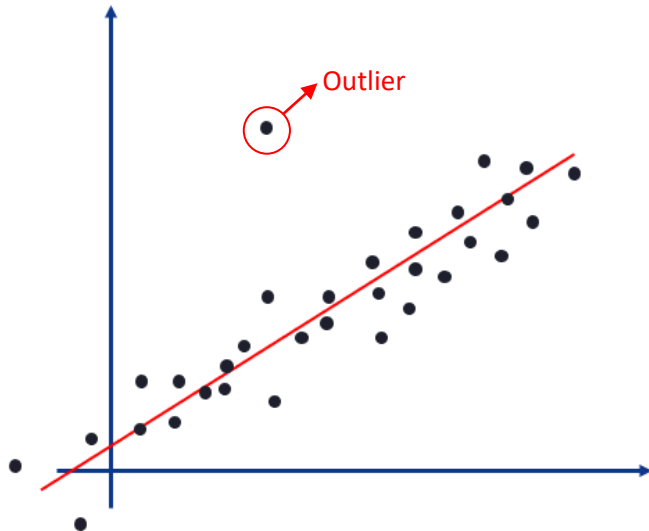
- **Supervised Learning**
 - Linear/Logistic Regression
 - Decision Tree
 - Support Vector Machine
- **Unsupervised Learning**
 - K-means Clustering
 - Hierarchical Clustering
- **Deep Learning**
 - CNN
 - RNN



Supervised Learning - Linear/Logistic Regression

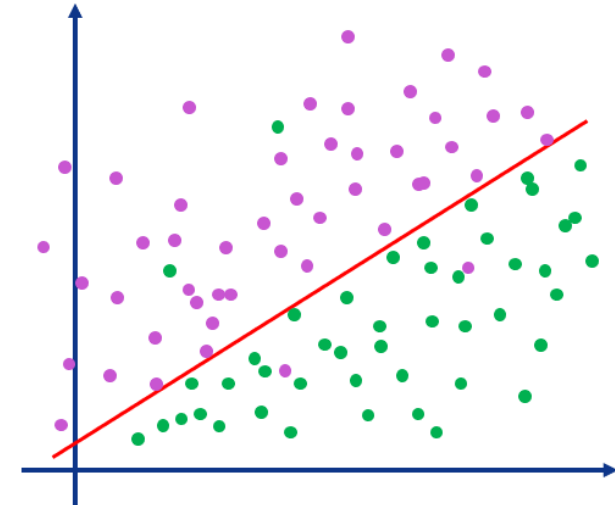
- **Linear Regression**

- Find the relationship between independent variables and dependent variables.
- Example:
 - Optimize pricing strategies
 - Demand analysis



- **Logistic Regression**

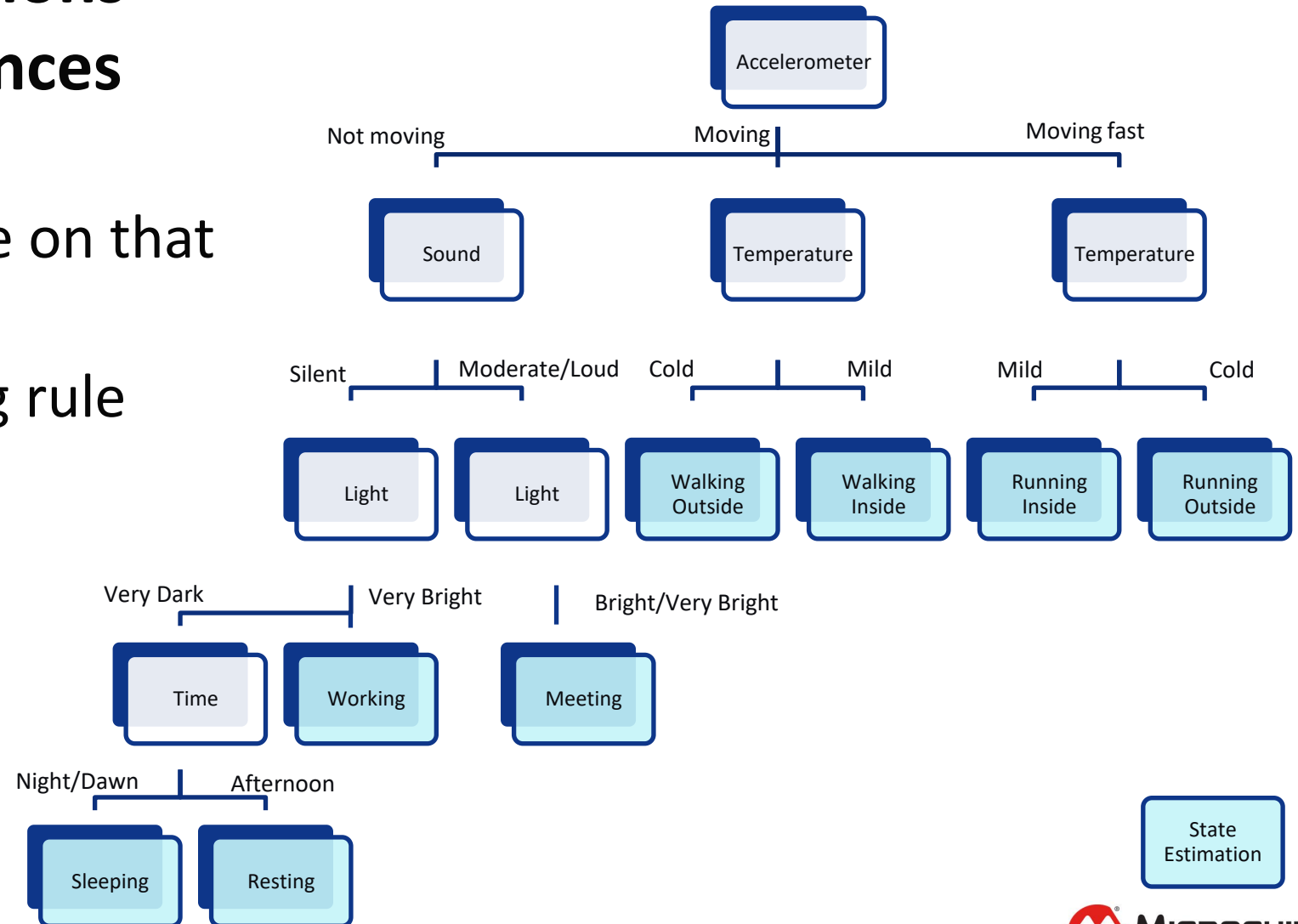
- The output of logistic regression is binary, either 1 or 0.
- Example:
 - Whether there is fraud based on transaction behavior patterns
 - Whether a tumor is malignant



Supervised Learning - Decision Tree

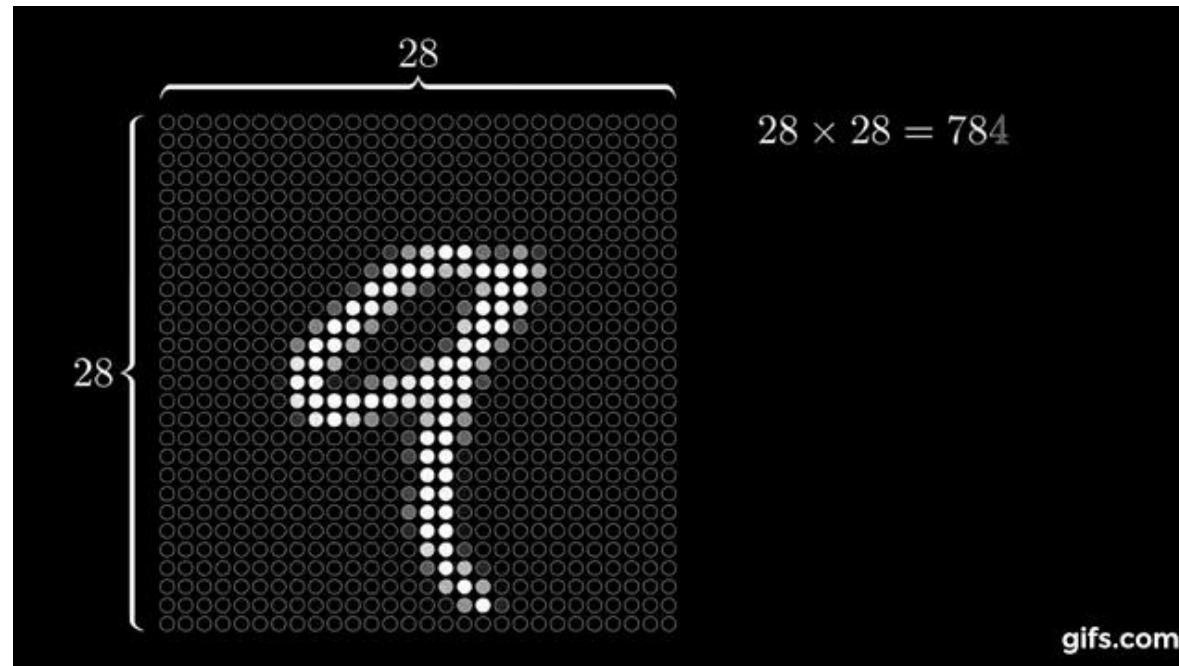
- **Tree-like graph of decisions and possible consequences**
 - Each node is a feature
 - Branching is decision rule on that feature
 - Training requires splitting rule generation and pruning
- **Example:**
 - Provide a decision-making framework

Example Movement Sensor



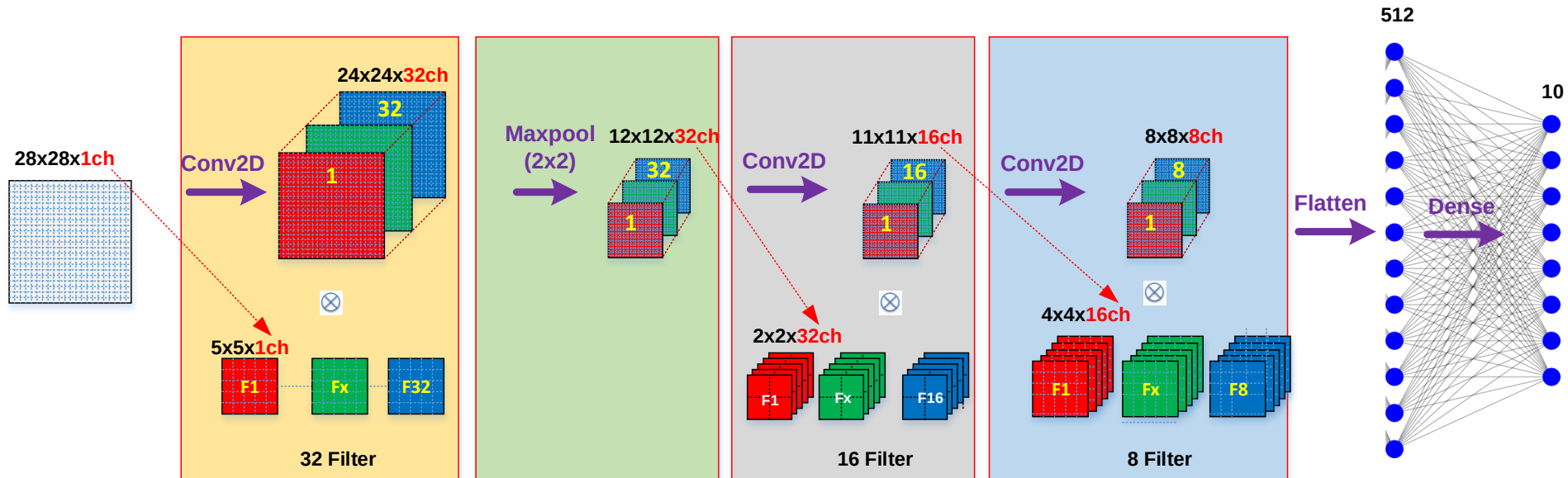
Deep Learning - Artificial Neural Network

- **Fully connected layers of neurons**
 - Best for tabular data, time series forecast, basic classification.
- **Touch example**
 - Touch sensor with resolution of 28 X 28 means 784 pieces of data to feed into the network with 10 desired outputs.



Deep Learning – Convolution Neural Network

- **Convolutions applied to local input regions**
 - The architecture most commonly used in image recognition.
 - Best for spatially related data
- **Example:**
 - Image Classification, Edge Detection



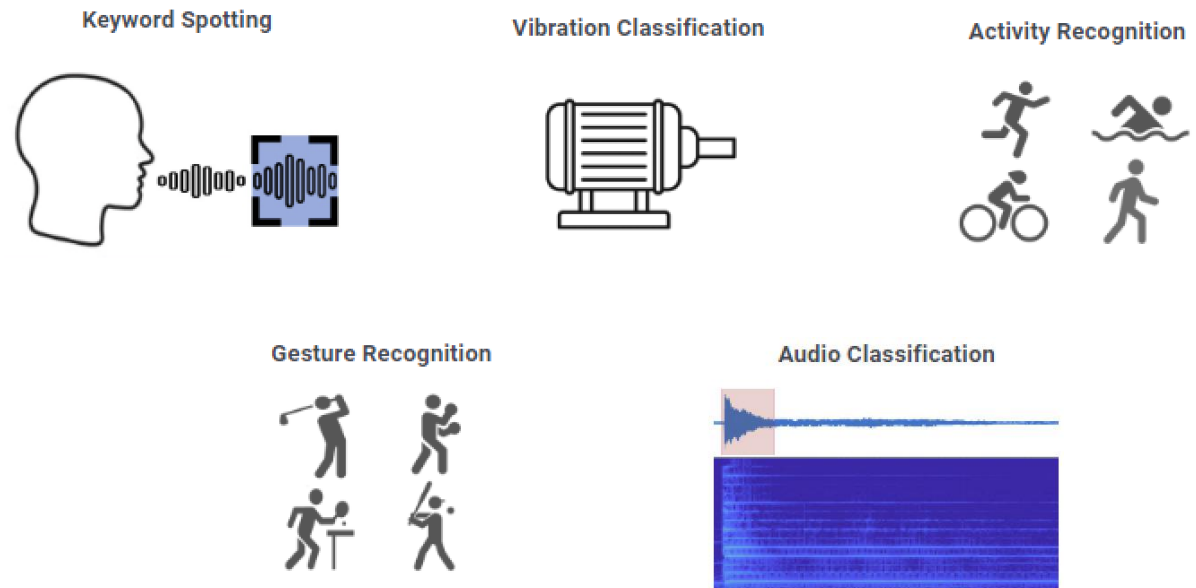
AutoML

- **Automated machine learning, AutoML is the process of automating the time-consuming, iterative tasks of machine learning model development.**
 - Implement ML solutions without extensive programming knowledge
 - Save time and resources
 - Apply data science best practices
 - Provide agile problem-solving



Select the Model

- Depending upon the problem space different models may be more appropriate
 - Certain architectures suit different problems
 - YOLO (You only look once) well suited to object detection using deep convolutional neural network.
 - CNN can be used for vibration analysis but others like Bayesian networks might be better.
- **MPLAB® ML Suite Templates**
 - Keyword Spotting
 - Vibration Classification
 - Activity Recognition
 - Gesture Recognition
 - Audio Classification



Lab3:

Analyze the data & Build Model

Lab3 Analyze the data & Build Model

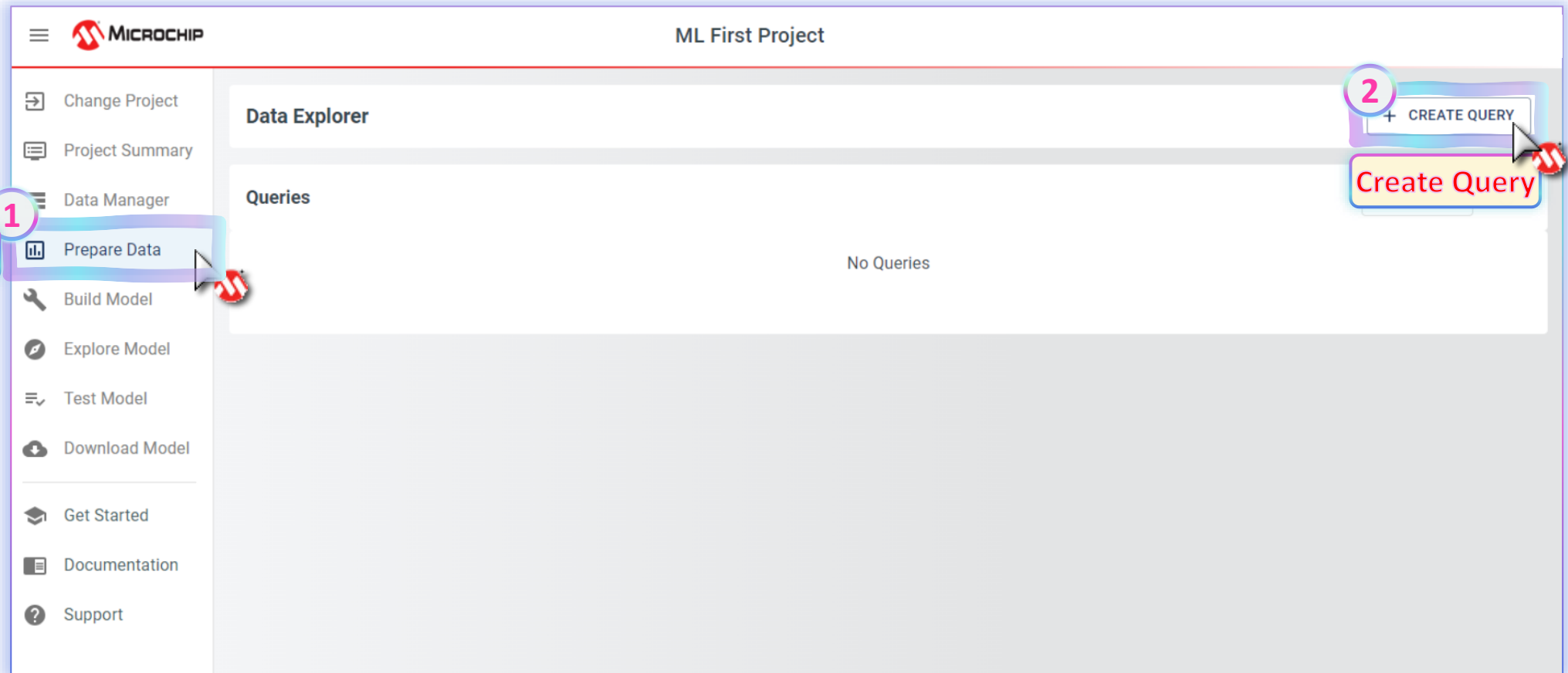
- Build the query of previous segment.
- Select the proper model for your project.
- Configure the parameters of the model.
- Start training!

Let's go!

Lab3 Analyze the data & Build Model

Step 1

- Click Prepare Data -> Create Query



Lab3 Analyze the data & Build Model

Step 2

- Click to open the list and select.

The screenshot shows a 'Create Query' form with several fields. To the left of the form, there are four colored boxes with text, each pointing to a specific field in the form:

- A red box labeled 'All Classes' points to the 'Query Name' field.
- A yellow box labeled 'Training Session' points to the 'Session' dropdown menu.
- A red box labeled 'Label' points to the 'Label' dropdown menu.
- A blue box labeled 'Segment_uuid' points to the 'Metadata' dropdown menu.

The form itself has the following fields:

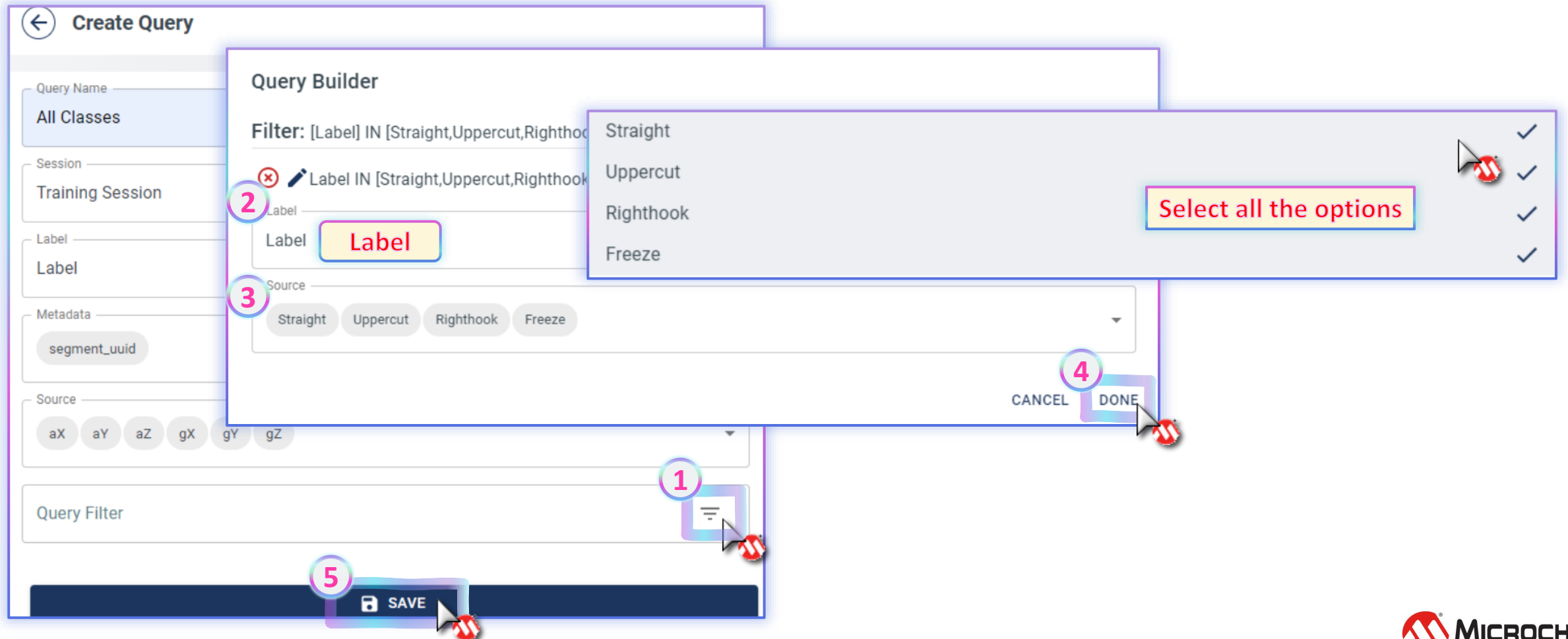
- Query Name:** A text input field containing 'All Classes'.
- Session:** A dropdown menu with 'Training Session' selected.
- Label:** A dropdown menu with 'Label' selected.
- Metadata:** A dropdown menu with 'segment_uuid' selected.
- Source:** A dropdown menu with a list of options: 'aX', 'aY', 'aZ', 'gX', 'gY', and 'gZ'.
- Query Filter:** A text input field with a filter icon on the right.

At the bottom of the form is a dark blue bar with a white 'SAVE' button.

Lab3 Analyze the data & Build Model

Step 3

- Click  to open the list



Create Query

Query Name
All Classes

Session
Training Session

Label
Label

Metadata
segment_uid

Source
aX aY aZ gX gY gZ

Query Filter

Query Builder

Filter: [Label] IN [Straight, Uppercut, Righthook, Freeze]

Label IN [Straight, Uppercut, Righthook, Freeze]

Source
Straight Uppercut Righthook Freeze

1 

2 

3 

4 

5 

Select all the options

CANCEL DONE

Lab3 Analyze the data & Build Model

Step 4

 **Create Query**

Query Name

All Classes

Session

Training Session

Label

Label

Metadata

segment_uuid

Source

aX aY aZ gX gY gZ

Query Filter

[Label] IN [Straight,Uppercut,Righthook,Freeze]

 **SAVE**

Lab3 Analyze the data & Build Model

Step 5

- Click  to build Cache

Queries

DELETE 

...	NAME	COLUMNS	FILTER	SEGMENTS	LAST MODIFIED	CREATED DATE	UUID	STATUS	REBUILD
☐	All Classes	aX aY aZ gX gY gZ	[Label] IN [Straight,Uppercut,Righthook,Freeze]		7/17/2024 5:23:21 PM	7/17/2024 5:23:21 PM	c66e7cd8- 376c-4a4e- 87f5- 41b3fcc66c5	NOT BUILT	

Rows per page: 25 1-1 of 1 |< < > >|



Queries

DELETE 

...	NAME	COLUMNS	FILTER	SEGMENTS	LAST MODIFIED	CREATED DATE	UUID	STATUS	REBUILD
☐	All Classes	aX aY aZ gX gY gZ	[Label] IN [Straight,Uppercut,Righthook,Freeze]	113	7/17/2024 5:23:43 PM	7/17/2024 5:23:21 PM	c66e7cd8- 376c-4a4e- 87f5- 41b3fcc66c5	CACHED	

Rows per page: 25 1-1 of 1 |< < > >|

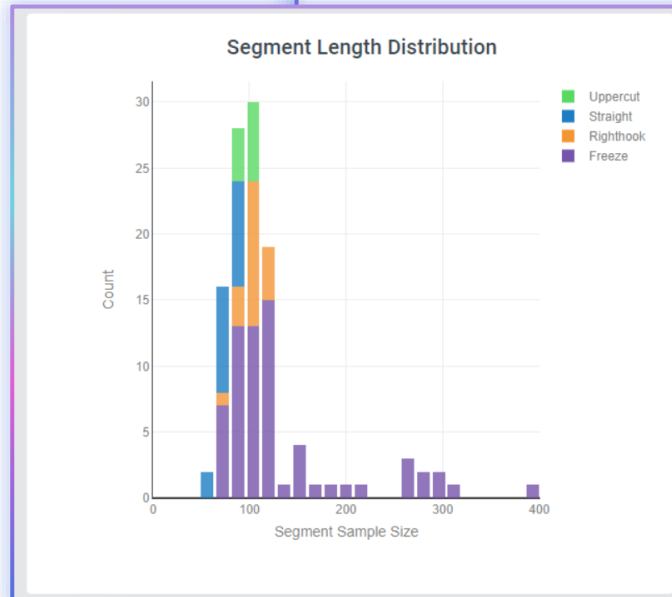
Lab3 Analyze the data & Build Model

Step 6

- Click **All classes** to see details

Queries

NAME	COLUMNS	FILTER	SEGMENTS	LAST MODIFIED	CREATED DATE	UUID	STATUS	REBUILD
All Classes	aX aY aZ gX gY gZ	[Label] IN [Straight,Uppercut,Righthook,Freeze]	113	7/17/2024 5:23:43 PM	7/17/2024 5:23:21 PM	c66e7cd8-376c-4a4e-87f5-41b3fcc66c5	✓ CACHED	



Lab3 Analyze the data & Build Model

Step 7

- Click Build Model

Build Model 1

+ CREATE PIPELINE ⬇️ IMPORT PIPELINE

Create Pipeline

Pipelines are individual workspaces where you can build machine learning models against your labeled data sets.

- You will be able to select the components of the pipeline including digital signal processing, feature extraction, sampling techniques and the Machine Learning algorithms.
- Pipelines create a cache at each step so you can quickly iterate your models.

CREATE PIPELINE

Keyword Spotting

Templates to do keyword spotting for human voice

SELECT TEMPLATE

Vibration Classification

This is a template for using machine learning to classify machine states based on vibrations sensors.

SELECT TEMPLATE

Activity Recognition

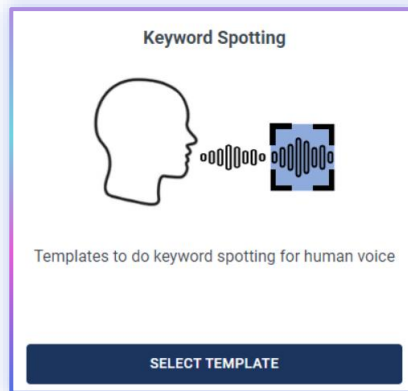
Gesture Recognition

Audio Classification

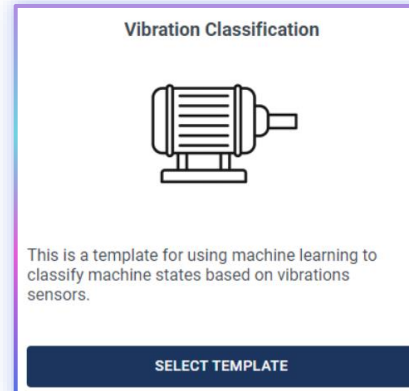
Lab3 Analyze the data & Build Model

Step 8

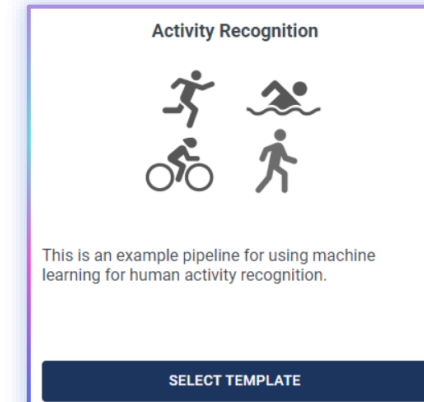
- MPLAB® ML Suite Templates, choose **Gesture Recognition**.



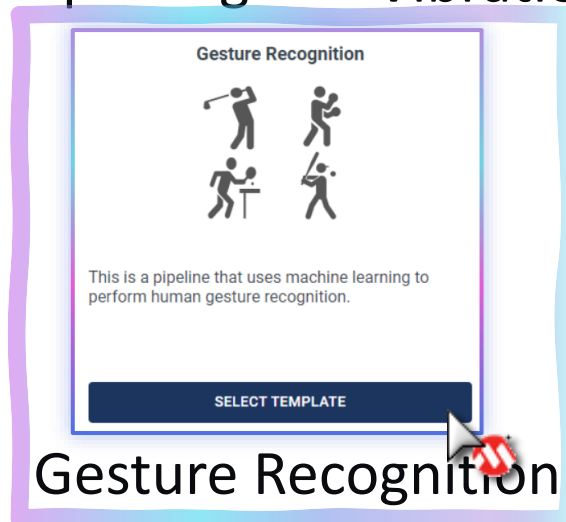
Keyword Spotting



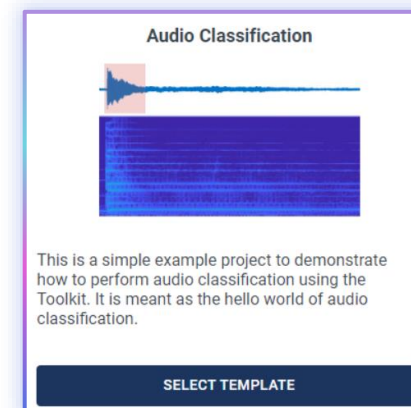
Vibration Classification



Activity Recognition



Gesture Recognition

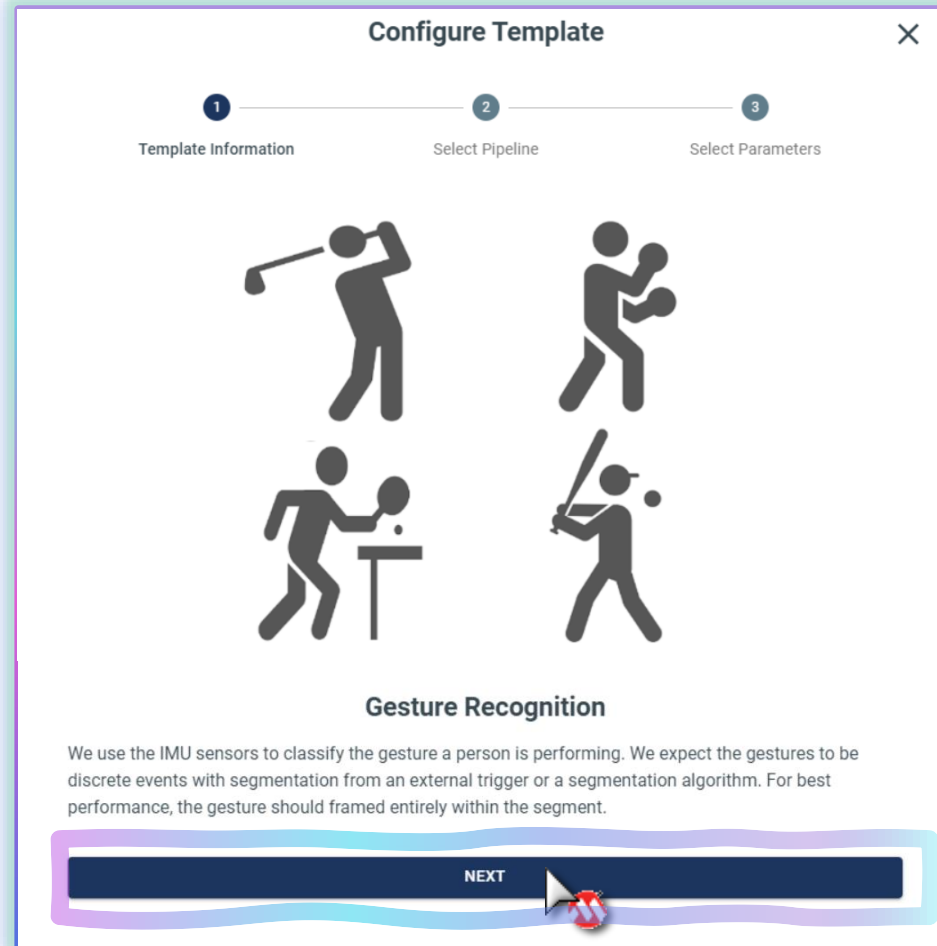


Audio Classification

Lab3 Analyze the data & Build Model

Step 9

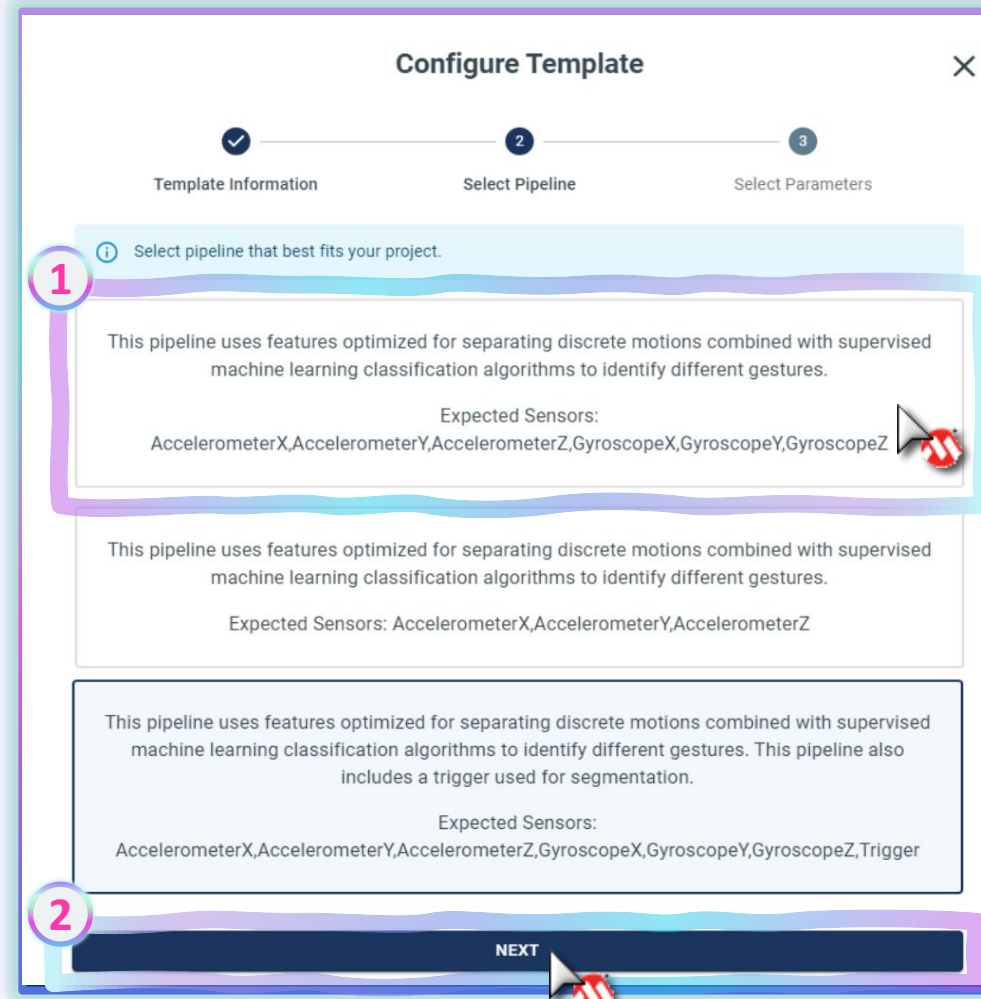
- Slide to bottom and click next.



Lab3 Analyze the data & Build Model

Step 10

- It will auto select the pipeline that best fits.
- Choose the 1st one.



Lab3 Analyze the data & Build Model

Step 11

- The name of model.
- Choose the Query.

Boxing Recognition

Select the query
you create

Configure Template

✓

✓

3

Template InformationSelect PipelineSelect Parameters

New name

Boxing Recognition

Query

All Classes

CREATE PIPELINE

Lab3 Analyze the data & Build Model

Step 12

- Fill in all the sensor Columns.

Name

All Classes

Label Column

Label

Columns

aX

aY

aZ

gX

gY

gZ

Metadata Columns

segment_uuid

Session

Training Session

The imported pipeline and the query All Classes have different sensor columns. To import this pipeline, please, select how to replace them.

Sensor Columns

AccelerometerX

→

aX

aX

×

AccelerometerY

→

aY

aY

×

AccelerometerZ

→

aZ

aZ

×

GyroscopeX

→

gX

gX

×

GyroscopeY

→

gY

gY

×

GyroscopeZ

→

gZ

gZ

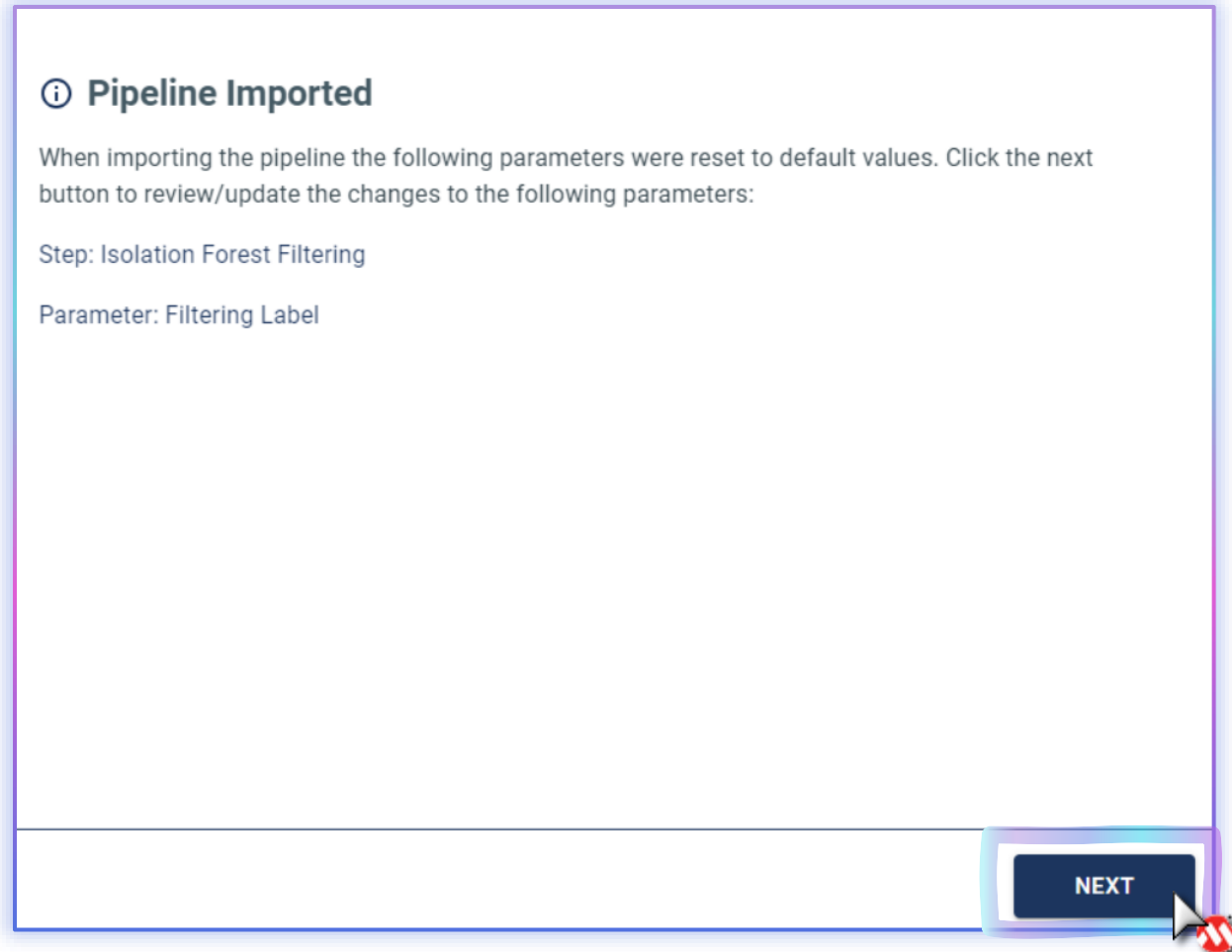
×

CREATE PIPELINE

Lab3 Analyze the data & Build Model

Step 13

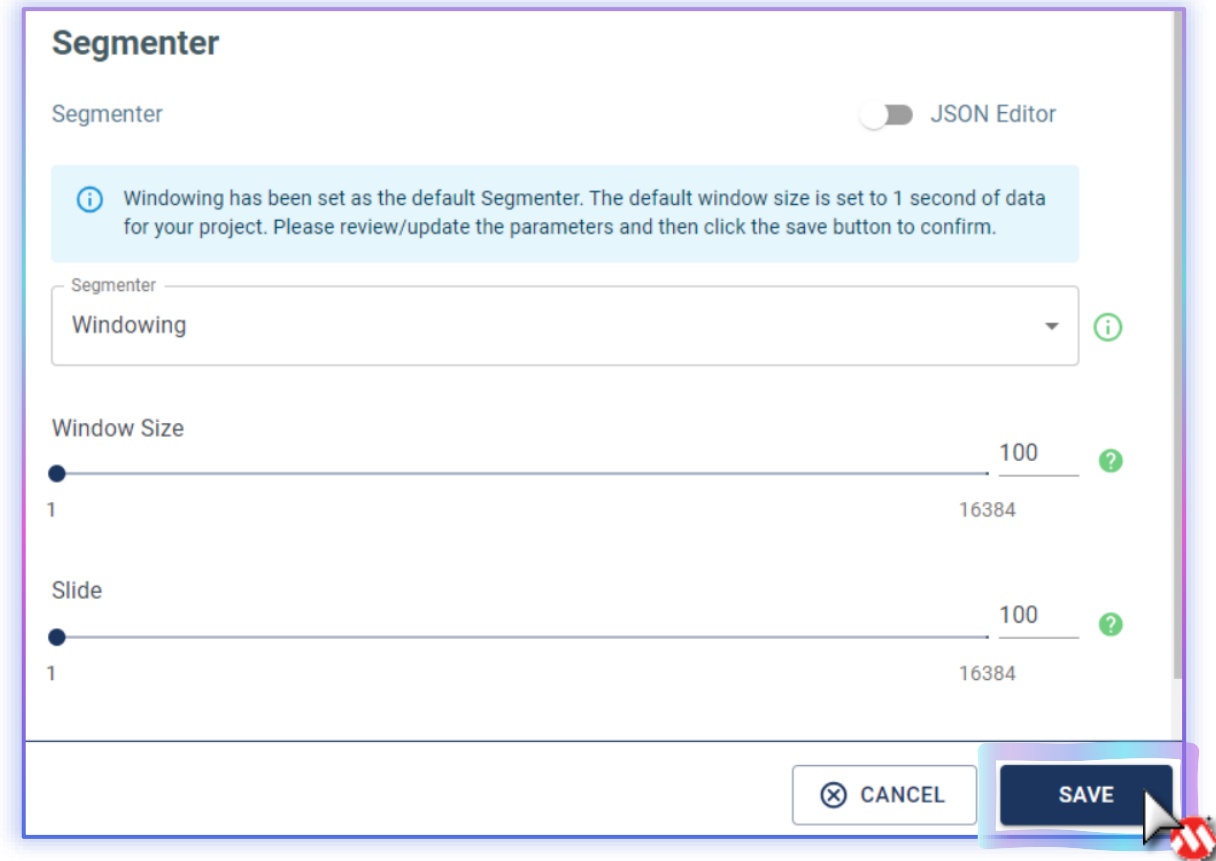
- Click next.



Lab3 Analyze the data & Build Model

Step 14

- **Adjust and confirm the segmenter.**
- **Window Size:** “100” refers to the window size in samples.
 - By picking “Windowing” and “100” on 100 hz data, there is a 1 second window size.
Which means every 1 second you will get a new classification.
- **Slide:** Control the amount of overlap between this signal and the last.



The screenshot shows a dialog box titled "Segementer". At the top right, there is a toggle switch for "JSON Editor" which is currently turned off. Below this, a light blue information box contains the text: "Windowing has been set as the default Segmenter. The default window size is set to 1 second of data for your project. Please review/update the parameters and then click the save button to confirm." The main area of the dialog has two sections. The first section, labeled "Segmenter", features a dropdown menu with "Windowing" selected and a green information icon to its right. The second section, labeled "Window Size", has a slider control. The slider is positioned at 100, with a green question mark icon to its right. The slider's range is from 1 to 16384. Below the "Window Size" section is a "Slide" section, which also has a slider control positioned at 100 with a green question mark icon to its right, and a range from 1 to 16384. At the bottom right of the dialog, there are two buttons: "CANCEL" and "SAVE". A mouse cursor is pointing at the "SAVE" button.

Lab3 Analyze the data & Build Model

Step 15

- Adjust and confirm the outlier filter.

Outlier Filter

Sampler JSON Editor

Review the values for the parameter: Filtering Label

Sampler
Isolation Forest Filtering

Filtering Label
Straight Uppercut Righthook Freeze

Outliers Fraction
0.01 0.05 1

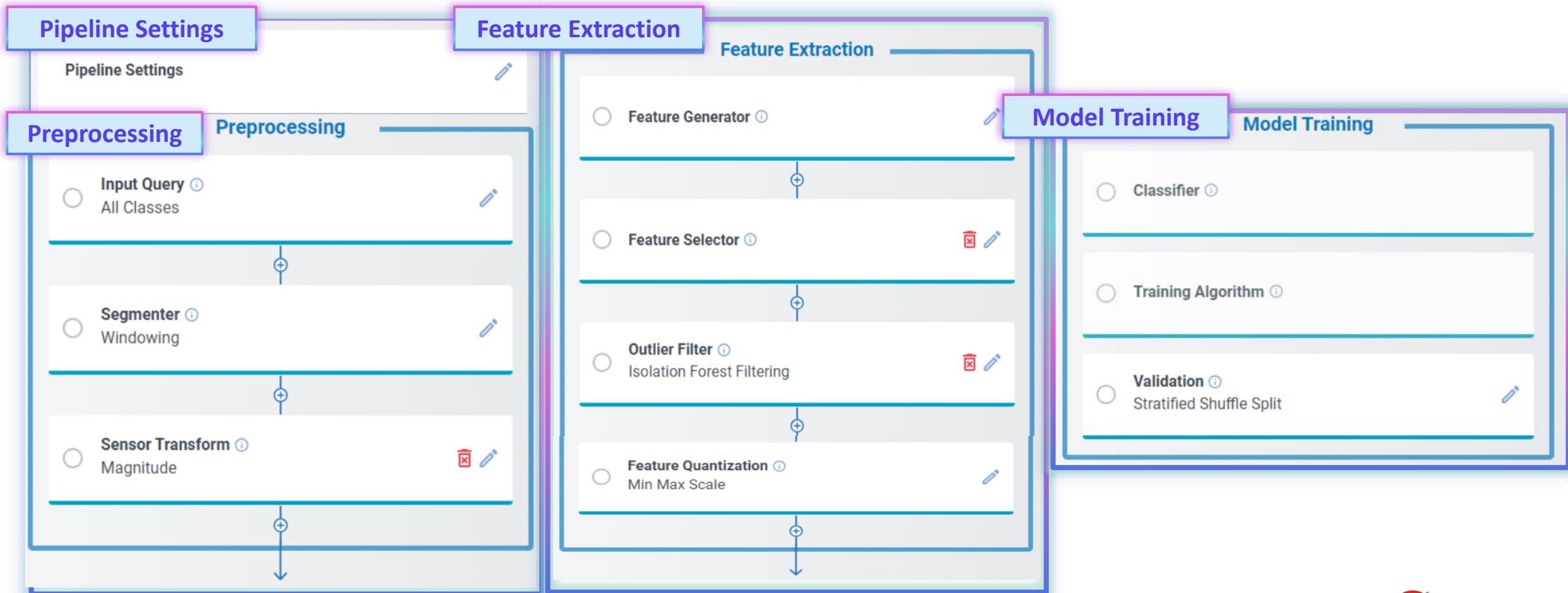
Assign Unknown

CANCEL SAVE

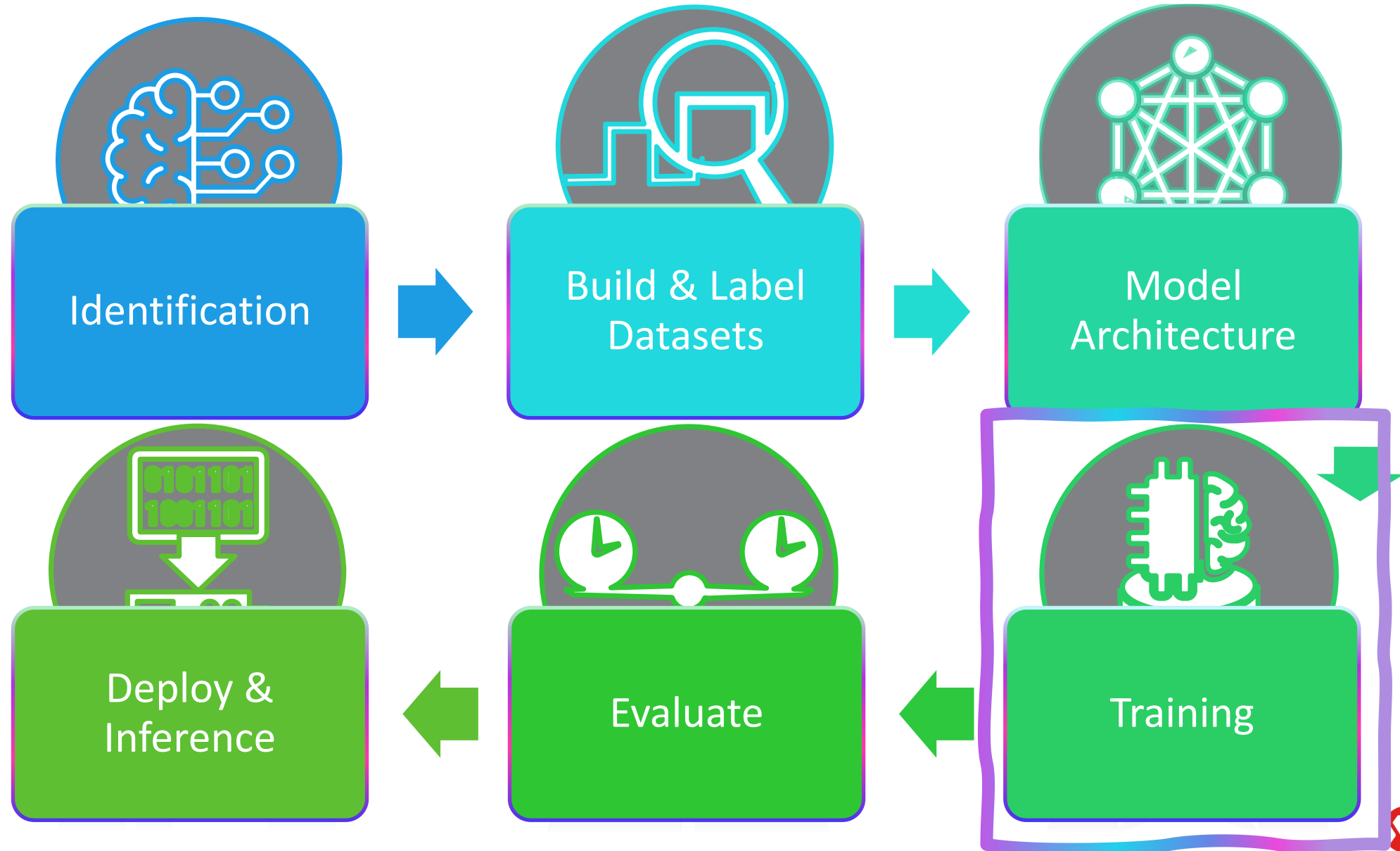
Lab3 Analyze the data & Build Model

Step 16

- Pipeline overview. Click  to edit.



Machine Learning workflow



Lab3 Analyze the data & Build Model

Step 17

- Start running pipeline.

The screenshot displays the ML Studio interface for a pipeline named 'Pipeline: Boxing Recognition'. The 'Preprocessing' section is visible, containing three steps: 'Input Query' (All Classes), 'Segmenter' (Windowing), and 'Sensor Transform' (Magnitude). A red callout box labeled 'Run Pipeline' points to the 'RUN PIPELINE' button. The 'LOGS' tab is selected, showing the following status:

```

RUNNING
✓ Optimizing Query...
✓ Query is optimized...
✓ AutoML Pipeline submitted to queue
10.0/10.0 CPU hours remaining. See our plans for more information at
https://www.microchip.com/en-us/tools-resources/develop/mplab-machine-learning-development-suite.

```

Lab3 Analyze the data & Build Model

Result

- Training Complete! 5 different models generated.

RESULTLOGS ✓TRAINING SUMMARY ✓

✓ COMPLETED
Population: 1
✓ Status: Running, Time: 0:05:03, Iteration 2/2, Step: 5/8, Name: Feature Selection, Population: 1
✓ Status: Running, Time: 0:05:08, Iteration 2/2, Step: 5/8, Name: Feature Selection, Population: 1
✓ Status: Running, Time: 0:05:13, Iteration 2/2, Step: 6/8, Name: Isolation Forest Filtering, Population: 1
✓ Status: Running, Time: 0:05:18, Iteration 2/2, Step: 7/8, Name: Min Max Scaling, Population: 1
✓ Status: Running, Time: 0:05:23, Iteration 2/2, Step: 8/8, Name: Model Training, Population: 16
✓ Status: Running, Time: 0:05:28, Iteration 2/2, Step: 8/8, Name: Model Training, Population: 16
✓ Status: Running, Time: 0:05:33, Iteration 2/2, Step: 8/8, Name: Model Training, Population: 16
✓ Status: Running, Time: 0:05:38, Iteration 2/2, Step: 8/8, Name: Model Training, Population: 16

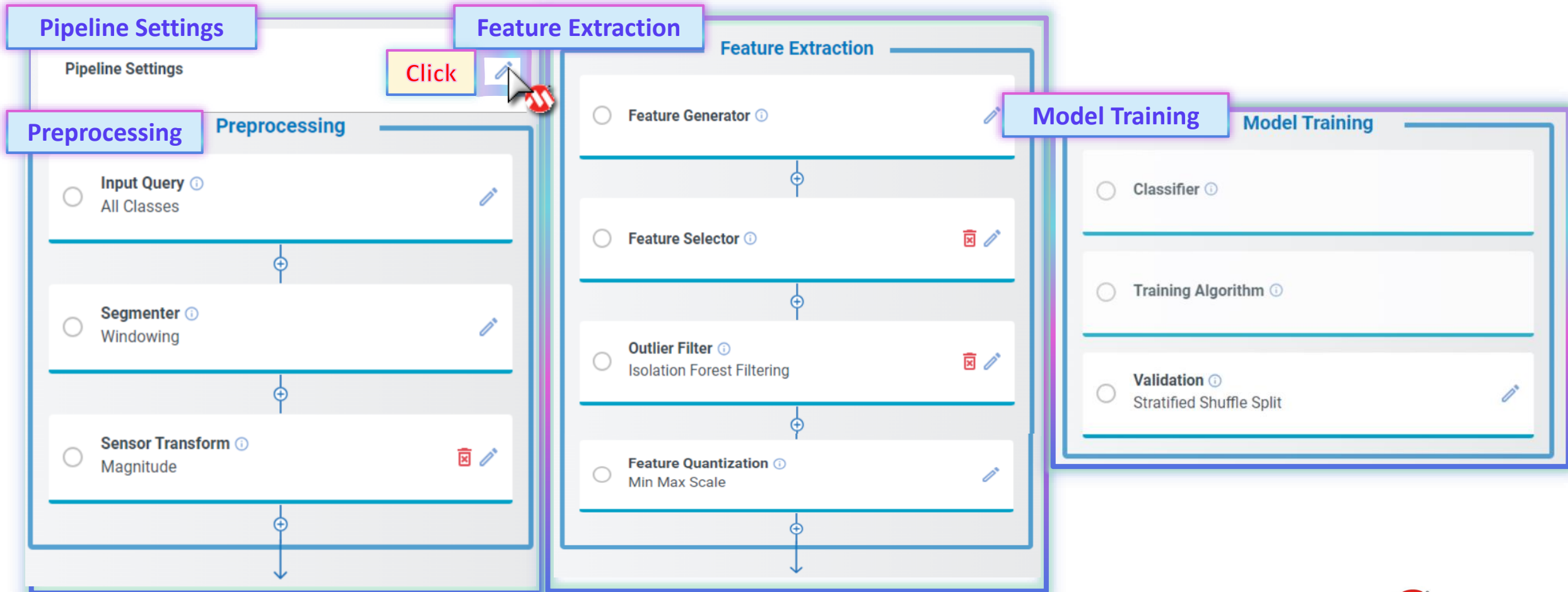
RESULTLOGS ✓TRAINING SUMMARY ✓

MODEL NAME	ACCURACY	CLASSIFIER SIZE(B)	NUM. FEATURES	SENSITIVITY	F1-SCORE
Boxing_Recognition_rank_0	100	1376	2	100	100
Boxing_Recognition_rank_1	100	1096	3	100	100
Boxing_Recognition_rank_2	100	193	8	100	100
Boxing_Recognition_rank_3	100	239	8	100	100
Boxing_Recognition_rank_4	100	232	12	100	100

Lab3 Analyze the data & Build Model

Step 18

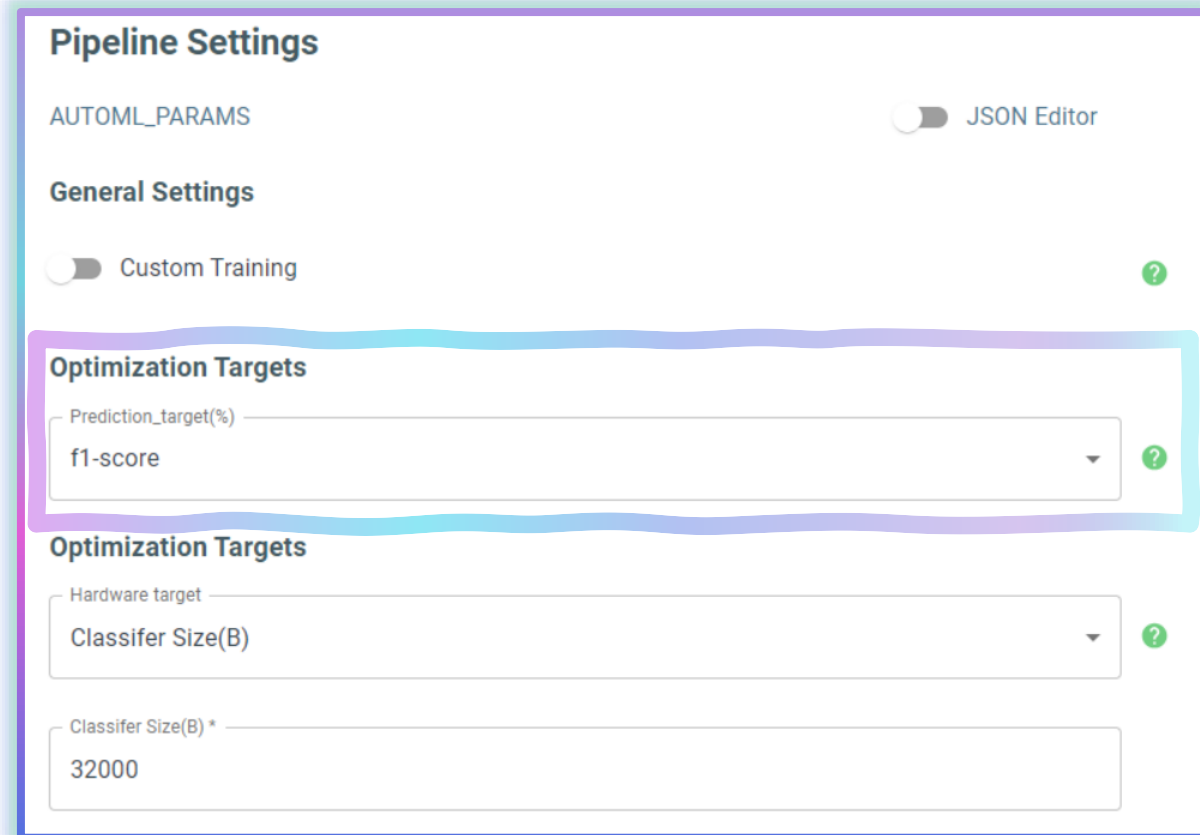
- Pipeline overview. Click  to edit.



Lab3 Analyze the data & Build Model

Pipeline Settings

- Pipeline Settings Involves defining the execution order of individual steps and the passing of parameters between them.
- Prediction Target:
 - F1-score
 - Accuracy
 - Sensitivity



The screenshot displays the 'Pipeline Settings' interface. At the top, there is a 'JSON Editor' toggle switch. Below this, the 'General Settings' section includes a 'Custom Training' toggle switch. The 'Optimization Targets' section is highlighted with a blue border and contains two dropdown menus: 'Prediction_target(%)' set to 'f1-score' and 'Hardware target' set to 'Classifier Size(B)'. A text input field below these dropdowns shows '32000'. Green question mark icons are visible next to the dropdown menus.

Pipeline Settings

AUTOML_PARAMS ☐ JSON Editor

General Settings

☐ Custom Training ?

Optimization Targets

Prediction_target(%)
f1-score ?

Optimization Targets

Hardware target
Classifier Size(B) ?

Classifier Size(B) *
32000

Lab3 Analyze the data & Build Model

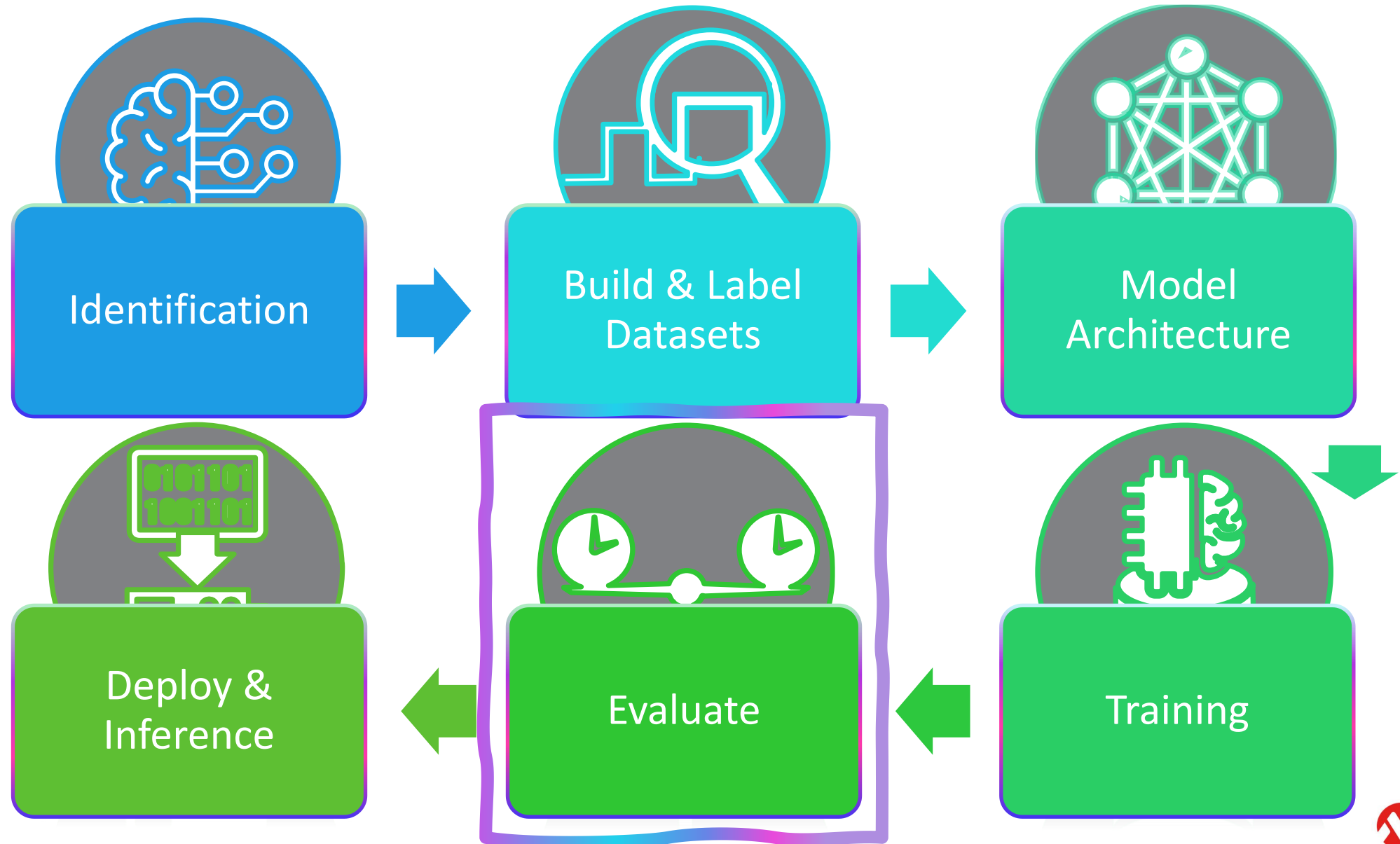
Pipeline Settings

- **Pipeline Settings** Involves defining the execution order of individual steps and the passing of parameters between them.
 - **Iteration:** Refer to the number of times an algorithm runs or cycles through the data.
 - **Population Size:** Defines how large the initial population is.

The screenshot displays a 'Search Settings' interface with three components:

- Iterations:** A horizontal slider with a range from 1 to 15. The current value is set to 2. A green question mark icon is located to the right of the slider.
- Population Size:** A horizontal slider with a range from 10 to 200. The current value is set to 40. A green question mark icon is located to the right of the slider.
- Anomaly Detection:** A toggle switch that is currently turned off, with a green question mark icon to its right.

Machine Learning workflow



Explore & Test model

How good is the model?

- **5 models generated**
 - rank_4 will usually be the best
 - May not be the smallest or fastest
 - Model tested against inputs
 - Using stratified split
- **Confusion Matrix**
 - An important summary
 - Compare actual against predicted

Model: light_pipeline_rank_4

← CHAN

CONFUSION MATRIX FEATURE VISUALIZATION FEATURE EMBEDDING FEATURE SUMMARY MODEL SUMMARY PIPELINE SUMMARY KNOWLEDGE PACK SUMMARY

	ambient	torch_blink	torch_high	torch_low	torch_mid	UNK	Support	Sensitivity(%)
ambient	9	0	0	0	0	0	9	100.00
torch_blink	0	8	0	0	1	0	9	88.89
torch_high	0	0	8	0	0	0	8	100.00
torch_low	0	0	0	9	0	0	9	100.00
torch_mid	0	0	0	0	9	0	9	100.00
Predicted	9	8	8	9	10	0	44	
Pos_Predic(%)	100.00	100.00	100.00	100.00	90.00		Acc(%)	97.73

	ambient	torch_blink	torch_high	torch_low	torch_mid	UNK	Support	Sensitivity(%)
ambient	44	0	0	0	0	0	44	100.00
torch_blink	0	19	0	0	25	0	44	43.18
torch_high	0	0	42	0	0	0	42	100.00
torch_low	0	0	0	44	0	0	44	100.00
torch_mid	0	0	0	0	44	0	44	100.00
Predicted	44	19	42	44	69	0	218	
Pos_Predic(%)	100.00	100.00	100.00	100.00	63.77		Acc(%)	88.53

	ambient	torch_blink	torch_high	torch_low	torch_mid	UNK	Support	Sensitivity(%)
ambient	9	0	0	0	0	0	9	100.00
torch_blink	0	8	0	0	1	0	9	88.89
torch_high	0	0	8	0	0	0	8	100.00
torch_low	0	0	0	9	0	0	9	100.00
torch_mid	0	0	0	0	9	0	9	100.00
Predicted	9	8	8	9	10	0	44	

	ambient	torch_blink	torch_high	torch_low	torch_mid	UNK	Support	Sensitivity(%)
ambient	35	0	0	0	0	0	35	100.00
torch_blink	0	31	0	0	4	0	35	88.57
torch_high	0	0	34	0	0	0	34	100.00
torch_low	0	0	0	35	0	0	35	100.00
torch_mid	0	0	0	0	35	0	35	100.00
Predicted	35	31	34	35	39	0	174	

Confusion Matrix

- The Confusion Matrix tab shows the averaged confusion matrix for the validation data.

Pos_Predic(%):

$$\frac{TP}{TP + FP} \times 100\%$$

Sensitivity (Recall)(%):

$$\frac{TP}{TP + FN} \times 100\%$$

Accuracy(%):

$$\frac{TP + TN}{TP + FN + FP + TN} \times 100\%$$

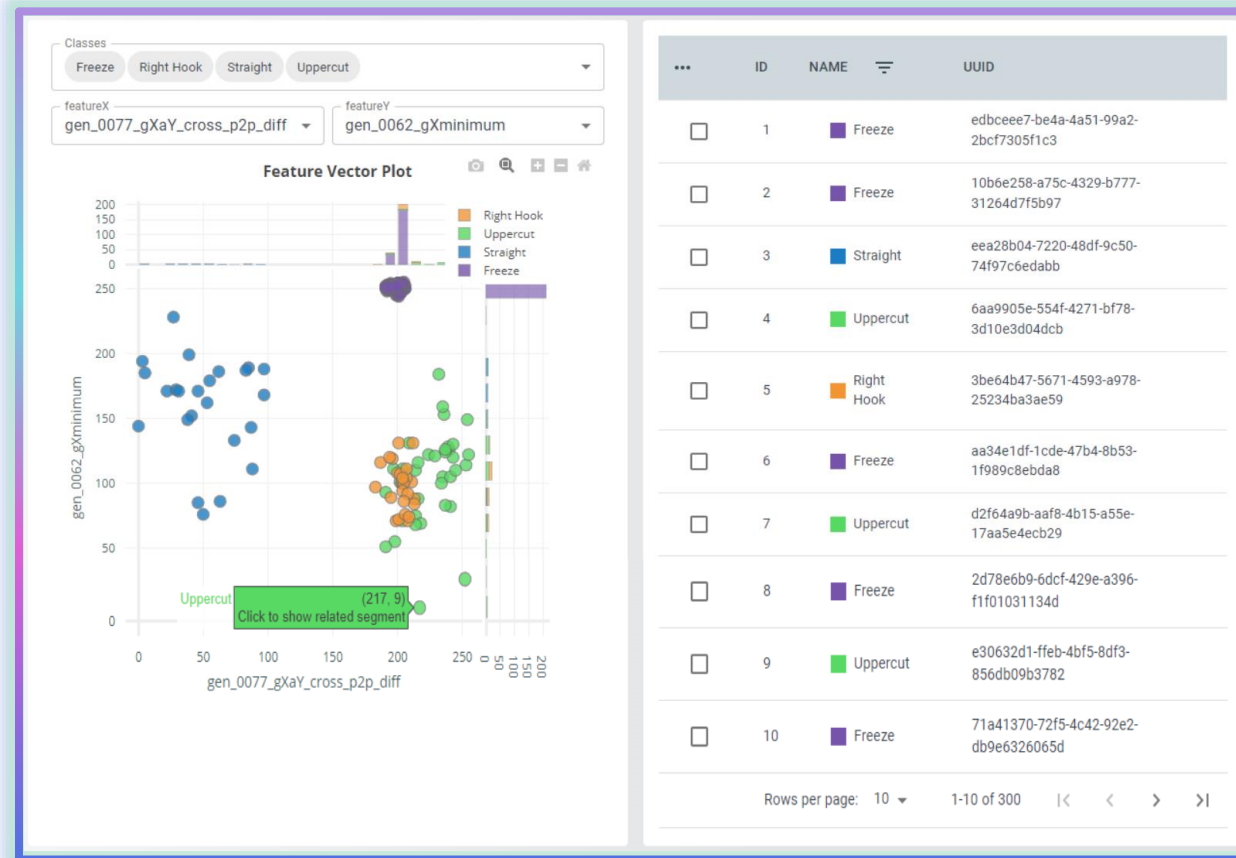
What's the prediction? Positive/Negative

Is the prediction correct? True/False

Confusion Matrix		Predicted	
		Yes	No
Actual	True	TP (True Positive)	FN (False Negative)
	False	FP (False Positive)	TN (True Negative)

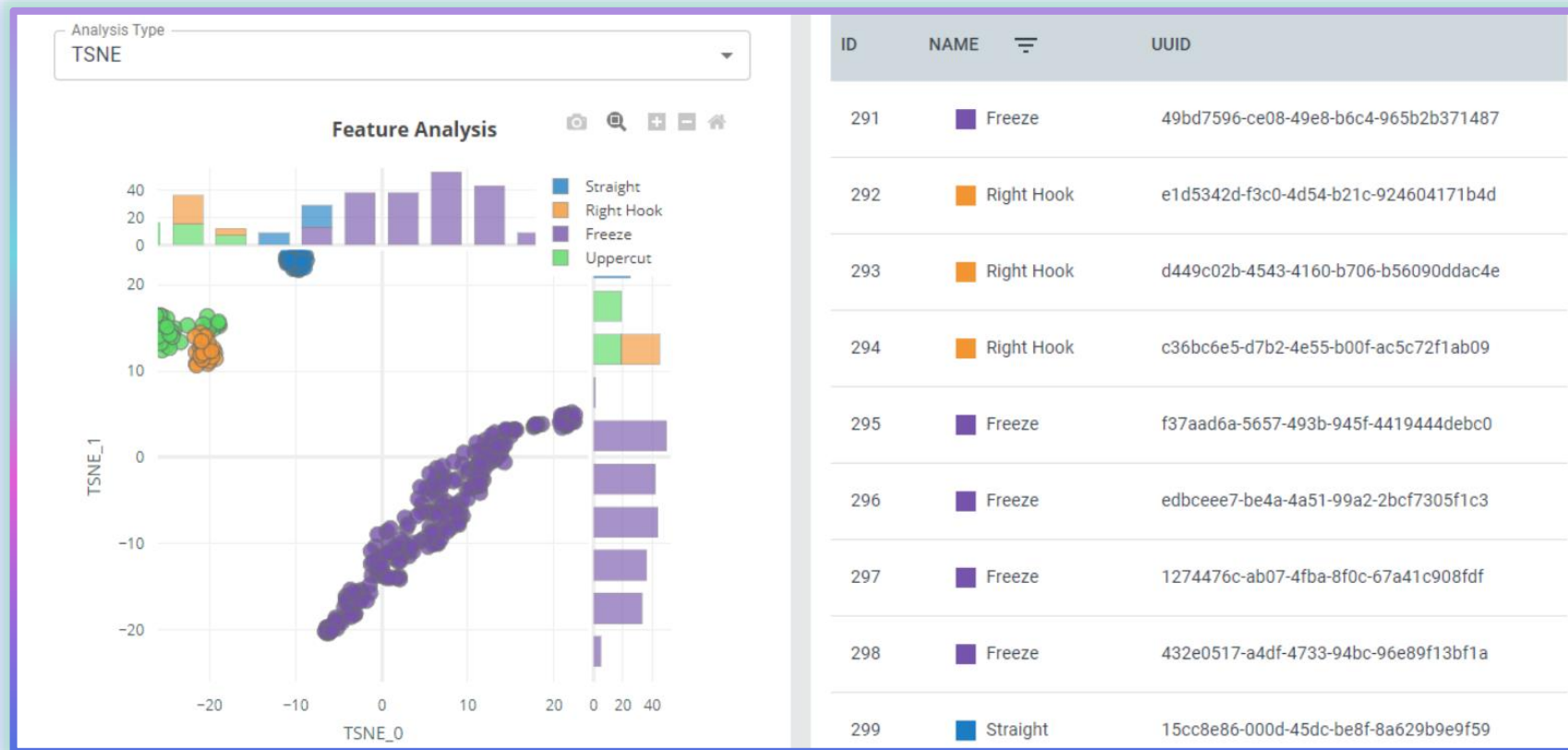
Feature Visualization

- Provide feature visualization as a 2-D comparison plot and feature insights chart.
- How well features are clustered according to classification.



Feature Embedding

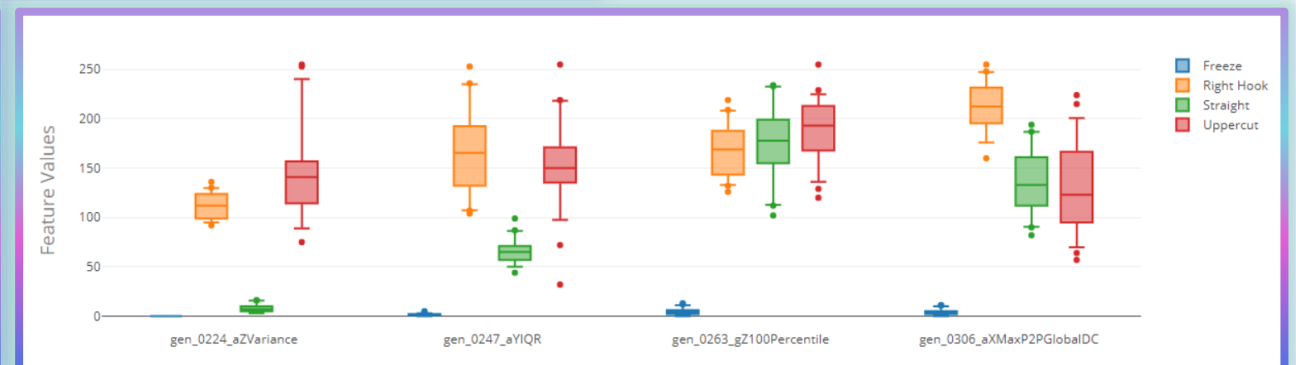
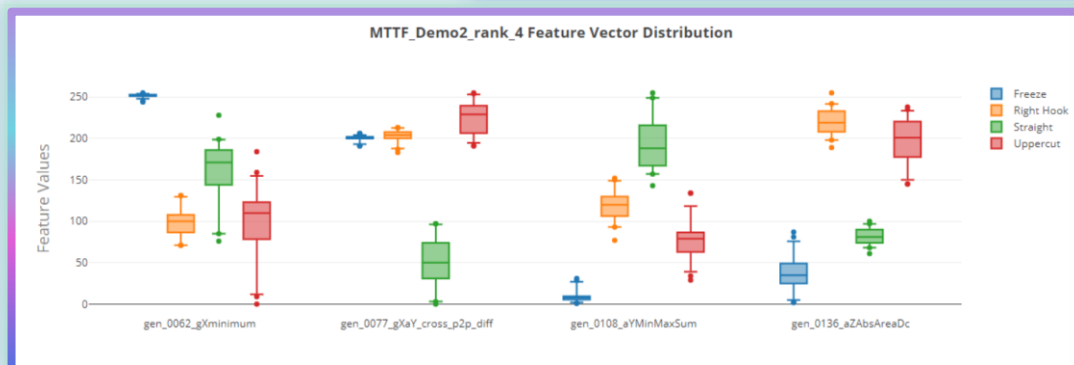
- The Feature Embedding tab provides three different embedding algorithms: UMAP, PCA, and TSNE.



Feature Summary

- Indication of what features are actually used to classify the results. Contain information about the features that are used during the feature extraction step of the Knowledge Pack.

FEATURE	CATEGORY	GENERATOR	SENSORS
gen_0062_gXminimum	Statistical	Minimum	gX
gen_0077_gXaY_cross_p2p_diff	Column Fusion	Two Column Peak To Peak Difference	gXaY
gen_0108_aYMinMaxSum	Amplitude	Global Min Max Sum	aY
gen_0136_aZAbsAreaDc	Area	Absolute Area of Low Frequency	aZ
gen_0224_aZVariance	Statistical	Variance	aZ
gen_0247_aYIQR	Statistical	Interquartile Range	aY
gen_0263_gZ100Percentile	Statistical	100th Percentile	gZ
gen_0306_aXMaxP2PGlobalDC	Amplitude	Global Peak to Peak of Low Frequency	aX



Model Summary

- The Model Summary tab describes the classifier, classifier parameters, and training algorithm that was used to generate the final model.

CLASSIFIER	UUID	OPTIMIZER	NEURONS	CLASSIFICATION_MODE	DISTANCE_MODE
PME	d6027a4e-5937-4547-b9fd-0dcf0381855f	RBF with Neuron Allocation Optimization	14	KNN	LSUP

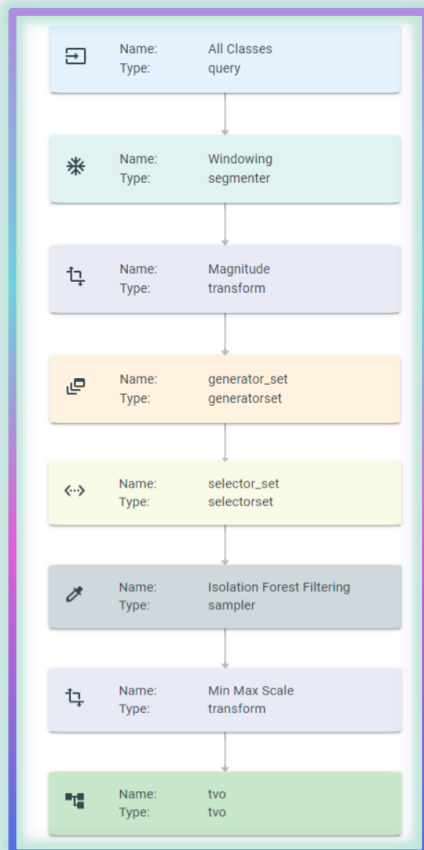
1-1 of 1

CATEGORY	CATEGORY ID	AIF	PATTERN
Uppercut	4	62	126,237,72,201,149,154,129,104
Right Hook	2	43	119,196,129,207,124,130,194,160
Freeze	1	100	251,200,8,73,0,3,5,9
Straight	3	100	171,22,215,79,5,87,233,90
Uppercut	4	37	68,214,81,161,97,120,211,174
Uppercut	4	59	107,199,134,214,134,203,210,95
Right Hook	2	41	94,204,130,222,122,163,187,244
Right Hook	2	53	84,213,100,198,110,107,188,208
Uppercut	4	58	126,240,80,217,162,144,197,102
Uppercut	4	78	111,197,134,228,149,153,139,125
Right Hook	2	55	92,208,100,200,102,160,174,203
Uppercut	4	76	131,209,67,145,75,108,229,224
Uppercut	4	100	55,198,51,145,89,32,163,159
Right Hook	2	100	71,199,95,224,113,253,149,176

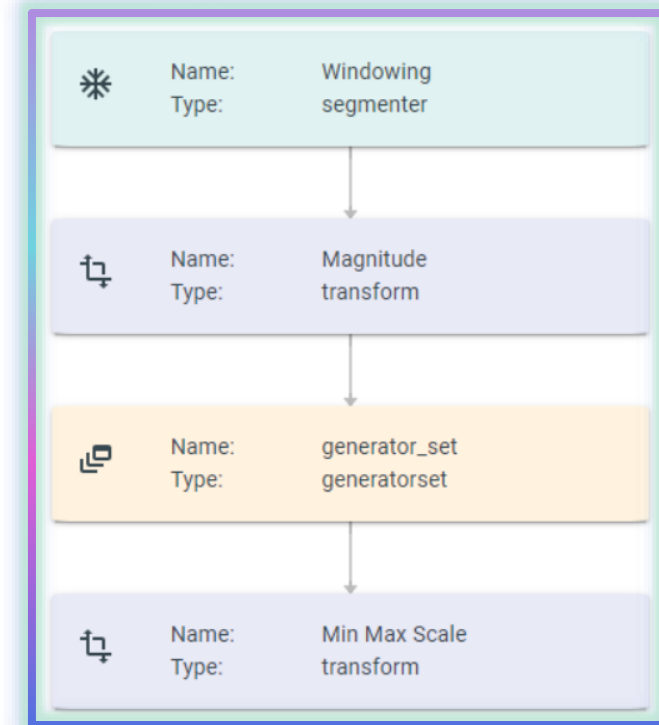
1-14 of 14

Pipeline & Knowledge Pack Summary

- Pipeline summary consists of data input, signal conditioning, signal preprocessing, feature extraction, *sampling*, and *model training*.



- Knowledge Packs summary consists of data input, signal conditioning, signal preprocessing, feature extraction, and *classification*.



Lab4: Explore & Test model

Lab4 Explore & Test model

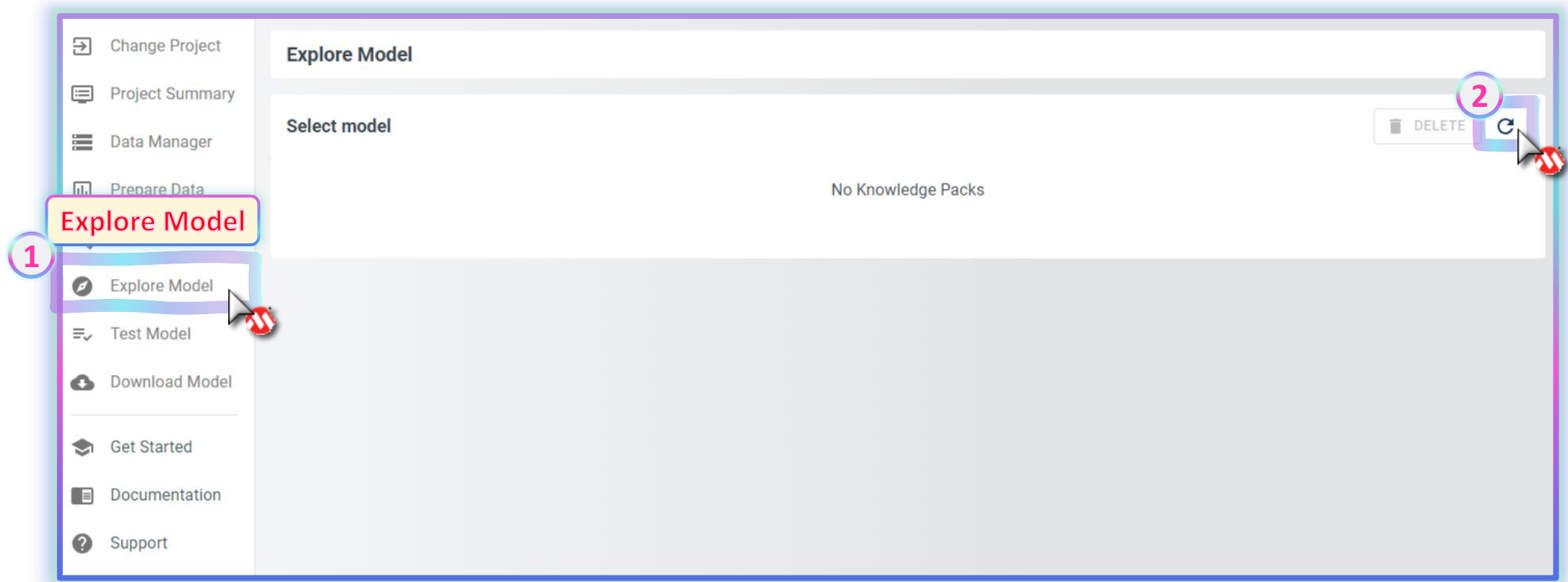
- **Observe 5 different models that the pipeline trained.**
- **View the detail of the model by exploring the model.**
- **Select the data to test the model.**

Let's go!

Lab4 Explore & Test model

Step 1

- Click Explore model and  to reload.



Lab4 Explore & Test model

Step 2

- The details of 5 model will show on the window.
- Click **rank_4** model to explore it. (Usually rank_4 is the best.)

Select model DELETE

...	NAME	CLASSIFIER	ACCURACY	MODEL SIZE (BYTES)	FEATURE COUNT	PIPELINE	CREATED DATE	UUID	EXPLORE	TEST	DOWNLOAD	RENAME	DELETE
☐	Boxing_Recognition_rank_4	PME	100.00 %	259	12	Boxing Recognition	7/18/2024 3:53:03 PM	9faa0712-69c7-4d53-971f-a3914588a878					
☐	Boxing_Recognition_rank_3	PME	100.00 %	193	8	Boxing Recognition	7/18/2024 3:53:03 PM	c0c01323-e35a-41b8-a187-6b9dcf3bcc43					
☐	Boxing_Recognition_rank_2	PME	100.00 %	216	8	Boxing Recognition	7/18/2024 3:53:03 PM	44661fed-3f22-488e-b39e-71ba2666b1aa					
☐	Boxing_Recognition_rank_1	PME	100.00 %	1348	3	Boxing Recognition	7/18/2024 3:53:03 PM	acceb015-efbb-429e-b955-f4597c2302b1					
☐	Boxing_Recognition_rank_0	Decision Tree Ensemble	100.00 %	1352	2	Boxing Recognition	7/18/2024 3:53:03 PM	d80ddabc-0e07-43c0-9751-0b500df91c6e					

Lab4 Explore & Test model

Step 3

- The window will show confusion matrix, feature visualization, feature embedding...etc of the model.

CONFUSION MATRIX FEATURE VISUALIZATION FEATURE EMBEDDING FEATURE SUMMARY MODEL SUMMARY PIPELINE SUMMARY KNOWLEDGE PAC >							
Validation							
	Freeze	Right Hook	Straight	Uppercut	UNK	Support	Sensitivity(%)
Freeze	43	0	0	0	0	43	100.00
Right Hook	0	5	0	0	0	5	100.00
Straight	0	0	5	0	0	5	100.00
Uppercut	0	0	0	8	0	8	100.00
Predicted	43	5	5	8	0	61	
Pos_Predic(%)	100.00	100.00	100.00	100.00		Acc(%)	100.00

Training							
	Freeze	Right Hook	Straight	Uppercut	UNK	Support	Sensitivity(%)
Freeze	215	0	0	0	0	215	100.00
Right Hook	0	26	0	0	0	26	100.00
Straight	0	0	25	0	0	25	100.00
Uppercut	0	0	0	39	0	39	100.00
Predicted	215	26	25	39	0	305	
Pos_Predic(%)	100.00	100.00	100.00	100.00		Acc(%)	100.00

Fold 0 - train							
	Freeze	Right Hook	Straight	Uppercut	UNK	Support	Sensitivity(%)
Freeze	172	0	0	0	0	172	100.00
Right Hook	0	21	0	0	0	21	100.00
Straight	0	0	20	0	0	20	100.00
Uppercut	0	0	0	31	0	31	100.00
Predicted	172	21	20	31	0	244	
Pos_Predic(%)	100.00	100.00	100.00	100.00		Acc(%)	100.00

Fold 0 - validation							
	Freeze	Right Hook	Straight	Uppercut	UNK	Support	Sensitivity(%)
Freeze	43	0	0	0	0	43	100.00
Right Hook	0	5	0	0	0	5	100.00
Straight	0	0	5	0	0	5	100.00
Uppercut	0	0	0	8	0	8	100.00
Predicted	43	5	5	8	0	61	
Pos_Predic(%)	100.00	100.00	100.00	100.00		Acc(%)	100.00

Lab4 Explore & Test model

Step 4

- Click Test model.

The screenshot shows the Microchip Studio interface for testing a model. The left sidebar contains a list of options: Change Project, Project Summary, Data Manager, Prepare Data, Build Model, Test Model (highlighted with a red circle and the number 1), Download Model, Get Started, Documentation, and Support. The main area displays the 'Model: Boxing_Recognition_rank_4' page. At the top, there is a 'Session' dropdown menu set to 'Training Session' and a 'CHANGE MODEL' button. Below this, a light blue box contains the instruction: 'Evaluate your model by selecting one or more capture files for classification.' The 'Captures' table lists three files: Straight1.csv, Upper1.csv, and Right1.csv. Each row includes checkboxes for selection, the file name, accuracy, segments, size, results, created date, capture UUID, and capture configuration UUID. At the bottom of the table, there are buttons for 'RECOGNIZE', 'SUMMARIZE', 'STOP', and a refresh icon.

	NAME	ACCURACY	SEGMENTS	SIZE(MB)	RESULTS	CREATED	CAPTURE_UUID	CAPTURE_CONFIGURATION_UUID	DC
☐	Straight1.csv		37	0.12		2024-07-16T06:43:20.252Z	564450f1-8e51-4572-b3a3-722e00d6a8e6		
☐	Upper1.csv		27	0.17		2024-07-16T06:43:20.961Z	7ab77451-9556-4ebe-b460-85222fd37493		
☐	Right1.csv		49	0.22		2024-07-16T06:43:21.675Z	a3d470d2-bb9f-4749-a7e7-65c7e6f7ee53		

Lab4 Explore & Test model

Step 5

- Select ✓ the files which want to test, may be multiple.
- Click Recognize

Session: Training Session

Evaluate your model by selecting one or more capture files for classification.

Recognize

2

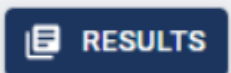
1

Captures

...	NAME	ACCURACY	SEGMENTS	SIZE(MB)	RESULTS	CREATED	CAPTURE_UUID	CAPTURE_CONFIGURATION_UUID	DC
✓	Right1.csv		49	0.22		2024-07-16T06:43:21.675Z	a3d470d2-bb9f-4749-a7e7-65c7e6f7ee53	22113093-e1ac-42ef-ac40-0e95a9b9acc3	R
✓	Right2.csv		0	0.20		2024-07-22T09:41:07.030Z	792f7714-f3e3-437e-9116-af979dd66a3a	22113093-e1ac-42ef-ac40-0e95a9b9acc3	R
☑	Straight1.csv		37	0.12		2024-07-16T06:43:20.252Z	564450f1-8e51-4572-b3a3-722e00d6a8e6	22113093-e1ac-42ef-ac40-0e95a9b9acc3	R
							c16e90e1-f502-		

Lab4 Explore & Test model

Step 6

- The result will be shown in the result field.
- Click  to view details.

Captures

▶ RECOGNIZE

SUMMARIZE

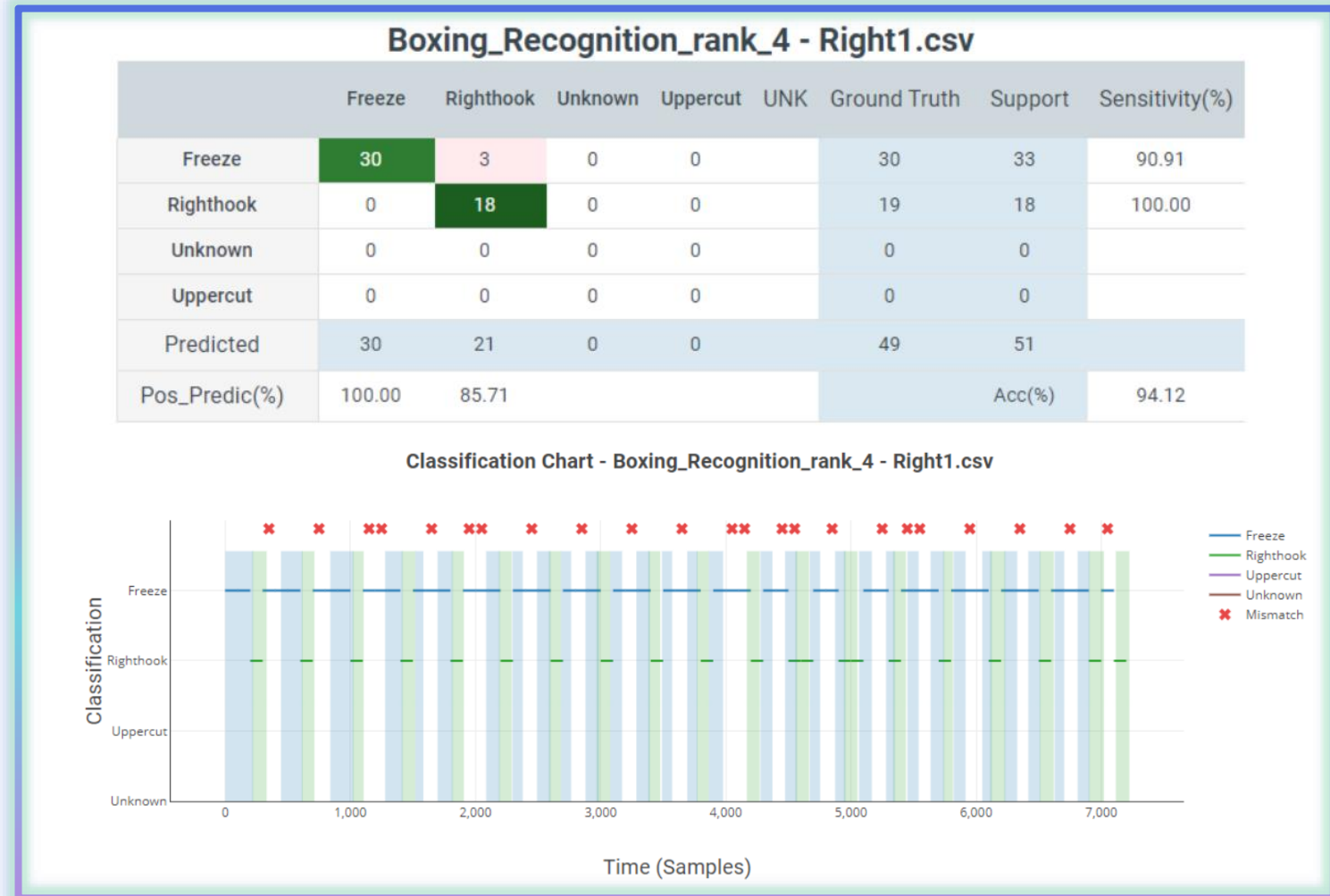
STOP

...	NAME	ACCURACY	SEGMENTS	SIZE(MB)	RESULTS	CREATED	CAPTURE_UUID	CAPTURE_CONFIGURATION_UUID
☒	Right1.csv	94.1	49	0.22	RESULTS	2024-07-16T06:43:21.675Z	a3d470d2-bb9f-4749-a7e7-65c7e6f7ee53	22113093-e1ac-42ef-ac40-0e95a9b9acc3
☐	Right2.csv		32	0.20		2024-07-16T09:41:07.030Z	792f7714-f3e3-437e-9116-af979dd66a3a	22113093-e1ac-42ef-ac40-0e95a9b9acc3
☐	Straight1.csv		37	0.12		2024-07-16T06:43:20.252Z	564450f1-8e51-4572-b3a3-722e00d6a8e6	22113093-e1ac-42ef-ac40-0e95a9b9acc3
☒	Straight2.csv	63.5	43	0.19	RESULTS	2024-07-22T09:41:07.631Z	c16e90e1-f502-4bb3-bbda-46eefb385009	22113093-e1ac-42ef-ac40-0e95a9b9acc3
☐	Straight3.csv		38	0.15		2024-07-22T09:41:08.231Z	29cd7292-3ce1-4143-8442-df9bbe7defac	22113093-e1ac-42ef-ac40-0e95a9b9acc3

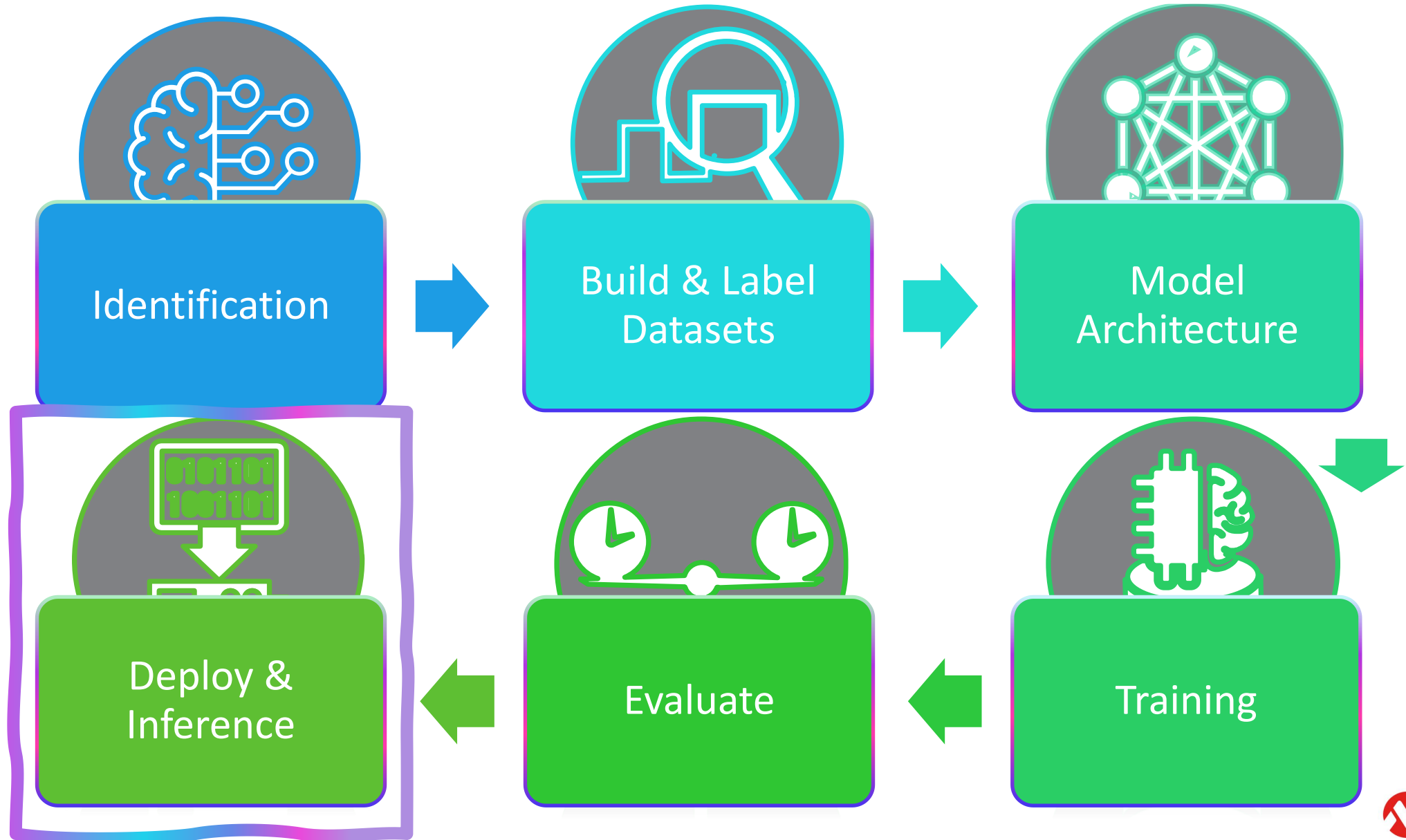
Lab4 Explore & Test model

Step 7

- Slide to the bottom to view detail.



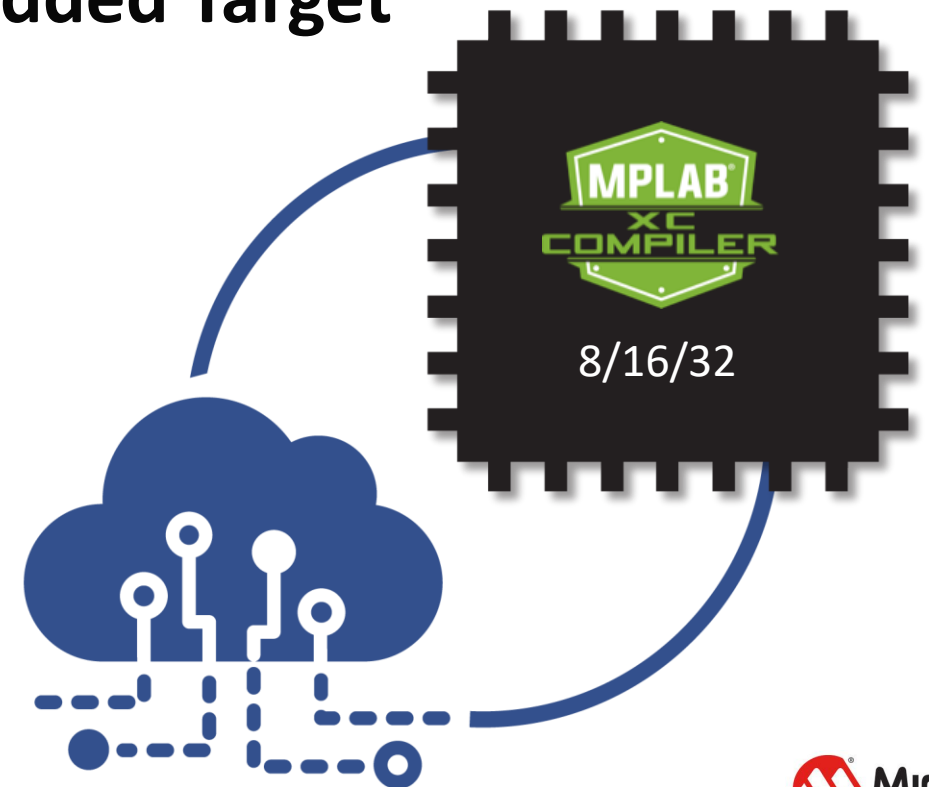
Machine Learning workflow



Deploy the model

Model Deployment

- **The model exists in the cloud**
 - Mathematical filters
 - Transformations
 - Python code
- **Needs to be converted to run on Embedded Target**
 - Converting precision
 - Doubles->char or int
 - Possibly involving regeneration
 - Convert language
 - Python to 'C' library
 - Account for architecture (8/16/32bit)
- **Implement calls to model**
 - Integrate it into your code



Silicon Platforms for ML Edge

Smart Embedded Vision

High performance video

- Real time / near real time
- High frame rate / large image size
- Multiple input streams
- Simultaneous multiple DNNs

Input sensors / data

Camera: Up to 4 interfaces

MIPI, parallel, HDMI

Mic: Multiple audio streams

Additional sensors

PolarFire
/ PF SoC
FPGA

Smart HMI

- Visual / audio / touch / gesture
- Recognition and interaction
- Single stream

Input sensors / data

Camera: MIPI, parallel, USB

Mic: Multiple audio streams

Touch surface

SAMA7
MPU

Smart Predictive Maintenance

- Time series / linear sensor data
- Simultaneous, multiple sensor
- Variety of sensor type supported

Input sensors / data

IMU (Inertial Measurement Unit)

Temp, Current, Pressure

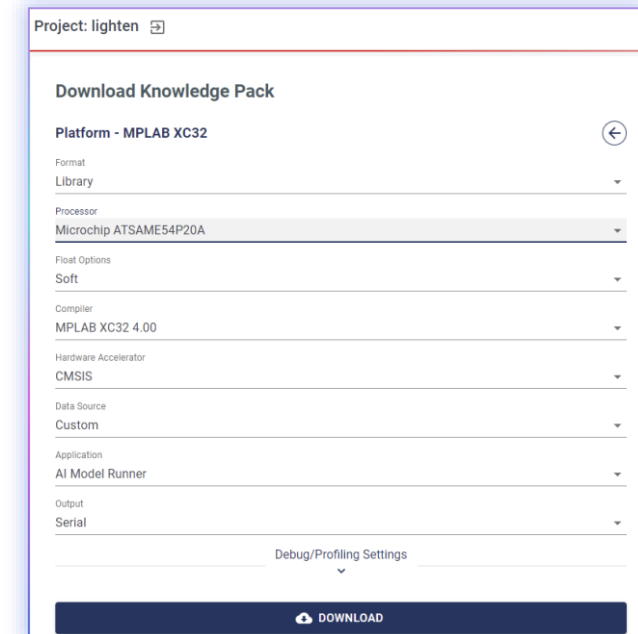
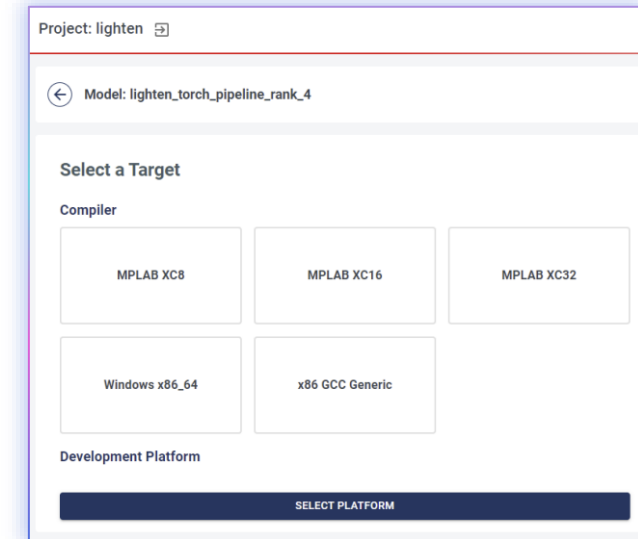
Mic: Audio Stream

SAMD21
MCU

AVR
MCU

Deploy the model

- **Select the optimal pipeline**
 - That achieves your desired performance
 - Verify the results with test data
 - Take more samples
- **Select your compiler target**
 - Select the processor
- **Create and download the knowledge pack**
 - zip file
 - Library (libmplabml.a)
 - Header files (6)
 - Source output is possible
 - Separate License



Deploy the model

- To interface with a Knowledge Pack, use the files that get downloaded with your Knowledge Pack.
 - Kb.h
 - sml_recognition_run.c
- **For this demo only 3 function calls needed**
 - kb_model_init(): Before any other functions are called, must initialize the Knowledge Pack.
 - kb_reset_model(): Called after every run of the model
 - kb_run_model(): Add new data to the model and classify it

Lab5 Export & Deploy the Model

Lab5 Export & Deploy the Model

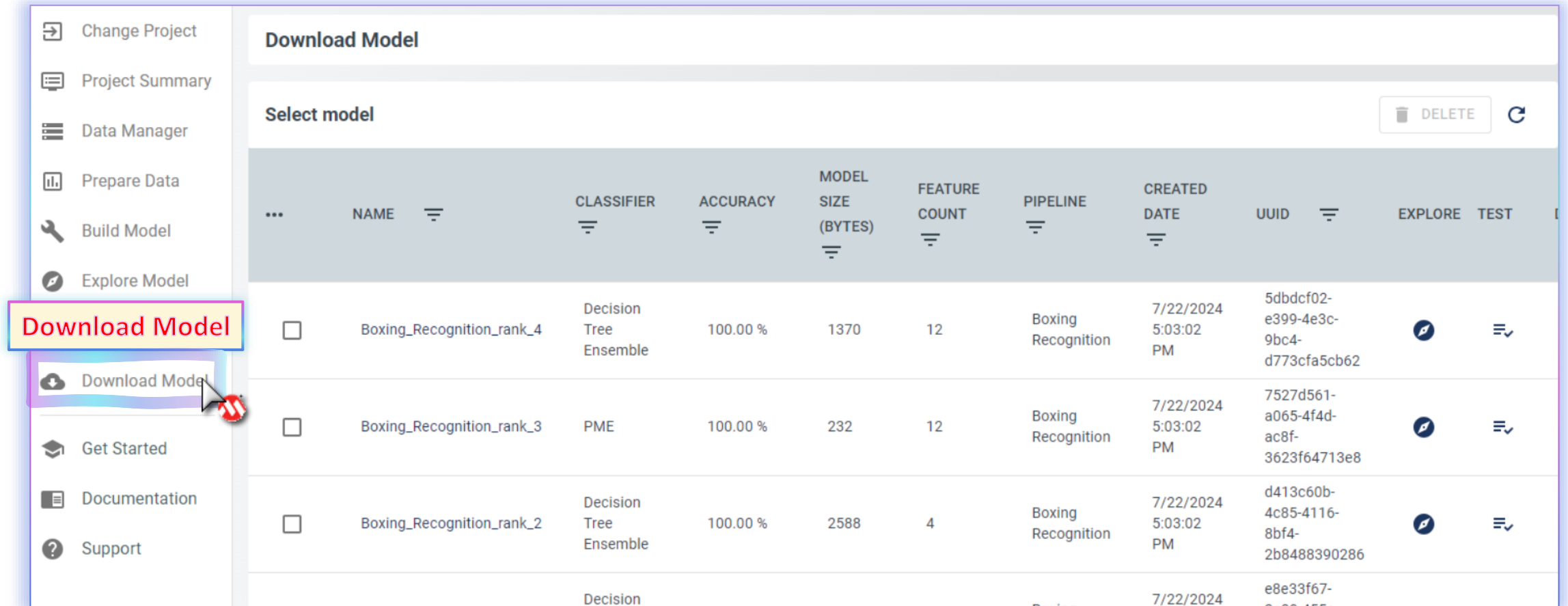
- **Convert the trained model to run on a SAMD21(XC32).**
- **Download the model and Integrate this into the project.**
- **Compile, program and test the model.**
 - Detects different actions.

Let's go!

Lab5 Export & Deploy the Model

Step 1

- Select Download Model



The screenshot displays the 'Download Model' interface. On the left, a sidebar contains navigation options: Change Project, Project Summary, Data Manager, Prepare Data, Build Model, Explore Model, Download Model (highlighted with a red box), Get Started, Documentation, and Support. The main area is titled 'Download Model' and contains a table of models. The table has columns for NAME, CLASSIFIER, ACCURACY, MODEL SIZE (BYTES), FEATURE COUNT, PIPELINE, CREATED DATE, UUID, EXPLORE, and TEST. Three models are listed, all with 100.00 % accuracy. The 'Download Model' button in the table header is highlighted with a red circle.

...	NAME	CLASSIFIER	ACCURACY	MODEL SIZE (BYTES)	FEATURE COUNT	PIPELINE	CREATED DATE	UUID	EXPLORE	TEST
☐	Boxing_Recognition_rank_4	Decision Tree Ensemble	100.00 %	1370	12	Boxing Recognition	7/22/2024 5:03:02 PM	5dbdcf02-e399-4e3c-9bc4-d773cfa5cb62		
☐	Boxing_Recognition_rank_3	PME	100.00 %	232	12	Boxing Recognition	7/22/2024 5:03:02 PM	7527d561-a065-4f4d-ac8f-3623f64713e8		
☐	Boxing_Recognition_rank_2	Decision Tree Ensemble	100.00 %	2588	4	Boxing Recognition	7/22/2024 5:03:02 PM	d413c60b-4c85-4116-8bf4-2b8488390286		
		Decision				Boxing	7/22/2024	e8e33f67-2e80-455e		

Lab5 Export & Deploy the Model

Step 2

- Choose the model you want to download and click .

Select model

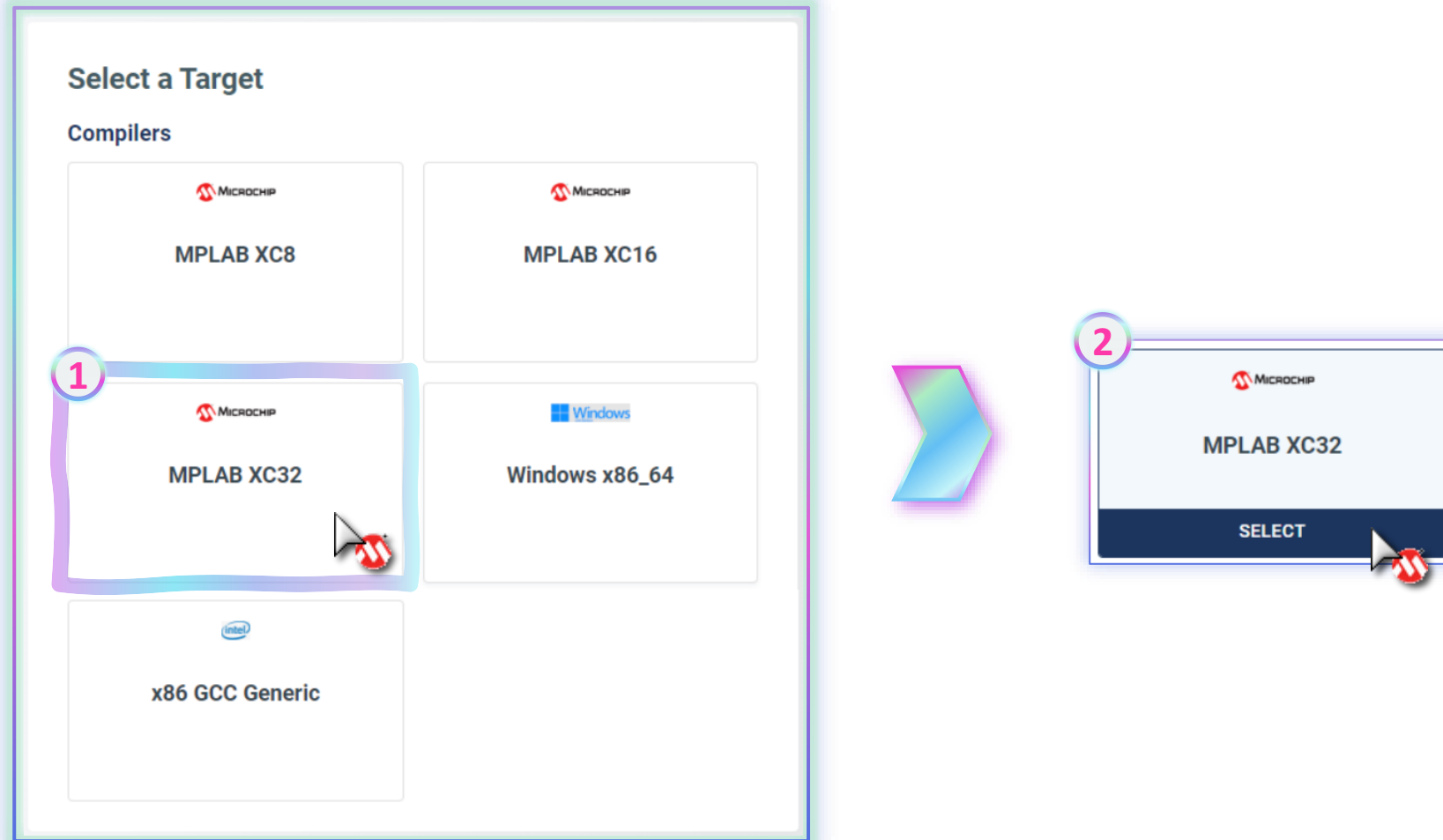
DELETE

...	NAME	CLASSIFIER	ACCURACY	MODEL SIZE (BYTES)	FEATURE COUNT	PIPELINE	CREATED DATE	UUID	EXPLORE	TEST	DOWNLOAD	RENAME	DELETE
☐	Boxing_Recognition_rank_4	Decision Tree Ensemble	100.00 %	1370	12	Boxing Recognition	7/22/2024 5:03:02 PM	5dbdcf02-e399-4e3c-9bc4-d773cfa5cb62					
☐	Boxing_Recognition_rank_3	PME	100.00 %	232	12	Boxing Recognition	7/22/2024 5:03:02 PM	7527d561-a065-4f4d-ac8f-3623f64713e8					
☐	Boxing_Recognition_rank_2	Decision Tree Ensemble	100.00 %	2588	4	Boxing Recognition	7/22/2024 5:03:02 PM	d413c60b-4c85-4116-8bf4-2b8488390286					

Lab5 Export & Deploy the Model

Step 3

- Select MPLAB XC32.



Lab5 Export & Deploy the Model

Step 4

- Configure the knowledge pack ->Download

The screenshot shows the 'Download Knowledge Pack' interface. On the left, a vertical navigation bar contains five items: 'Library' (blue), 'ATSAMD21G18A' (yellow), 'Library' (blue), 'Custom' (yellow), and 'AI Model Runner' (blue). Below these are 'Serial' (blue) and 'Debug/Profiling Settings' (grey). A red circle with the number '1' highlights the 'Platform - MPLAB XC32' dropdown menu. A red circle with the number '2' highlights the 'DOWNLOAD' button at the bottom. The main content area has a 'Platform' section with fields for Name (MPLAB XC32), Manufacturer (Microchip), Description (Compile libraries for Microchip 32 bit processors.), and Resources (Firmware Documentation). Below this is a 'Class Map' section with three buttons: '1 - Freeze', '2 - Righthook', and '3 - Uppercut'. The 'Knowledge Pack Resource Estimates' section includes 'Estimated Memory Usage' (SRAM Used: 2108 Bytes, Stack Size: 1114 Bytes, Flash Used: 6480 Bytes) and 'Estimated Latency' (Feature Extraction Latency: 2.154 ms (172300), Total Latency: 2.15375 ms (172300)). Each estimate has a green question mark icon to its right.

Download Knowledge Pack

1 Platform - MPLAB XC32

Format: Library

Processor: ATSAMD21G18A

Compiler: MPLAB XC32 4.35

Data Source: Custom

Application: AI Model Runner

Output: Serial

Debug/Profiling Settings

2 DOWNLOAD

Platform

Name: MPLAB XC32
Manufacturer: Microchip
Description: Compile libraries for Microchip 32 bit processors.
Resources: [Firmware Documentation](#)

Class Map:

1 - Freeze 2 - Righthook 3 - Uppercut

Knowledge Pack Resource Estimates

Estimated Memory Usage

SRAM Used:	2108 Bytes	?
Stack Size:	1114 Bytes	?
Flash Used:	6480 Bytes	?

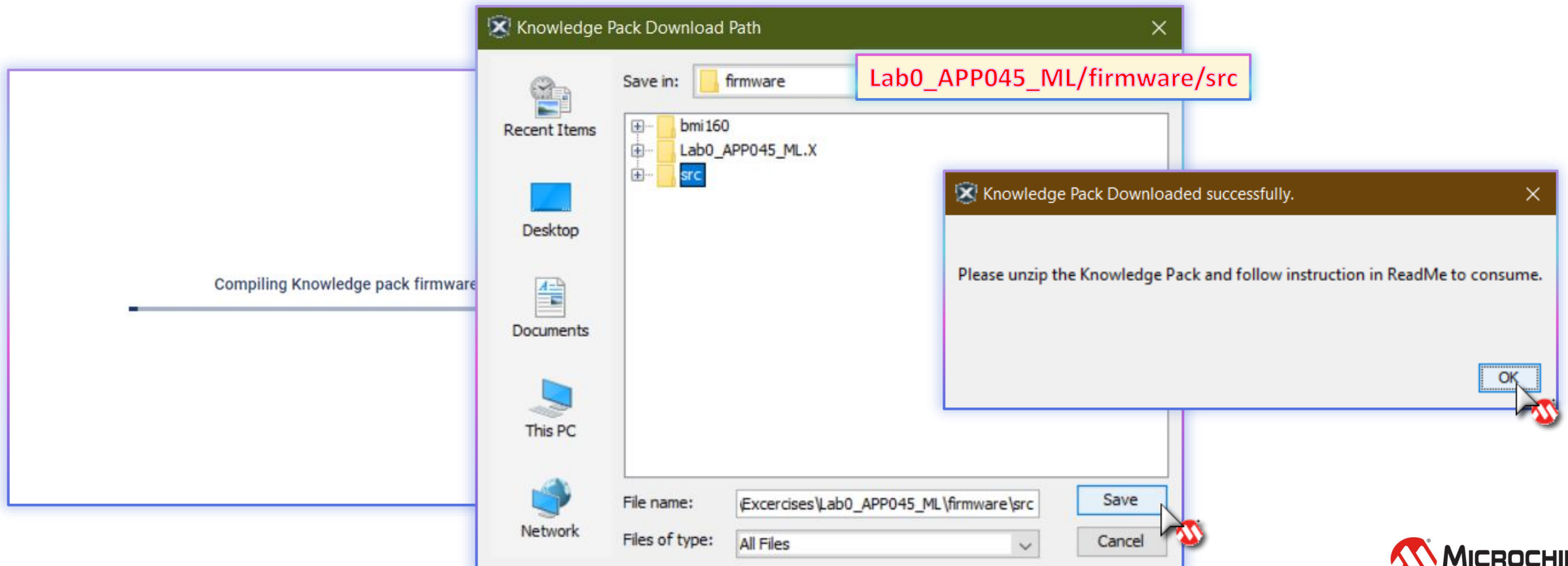
Estimated Latency

Feature Extraction Latency:	2.154 ms (172300)	?
Total Latency:	2.15375 ms (172300)	?

Lab5 Export & Deploy the Model

Step 5

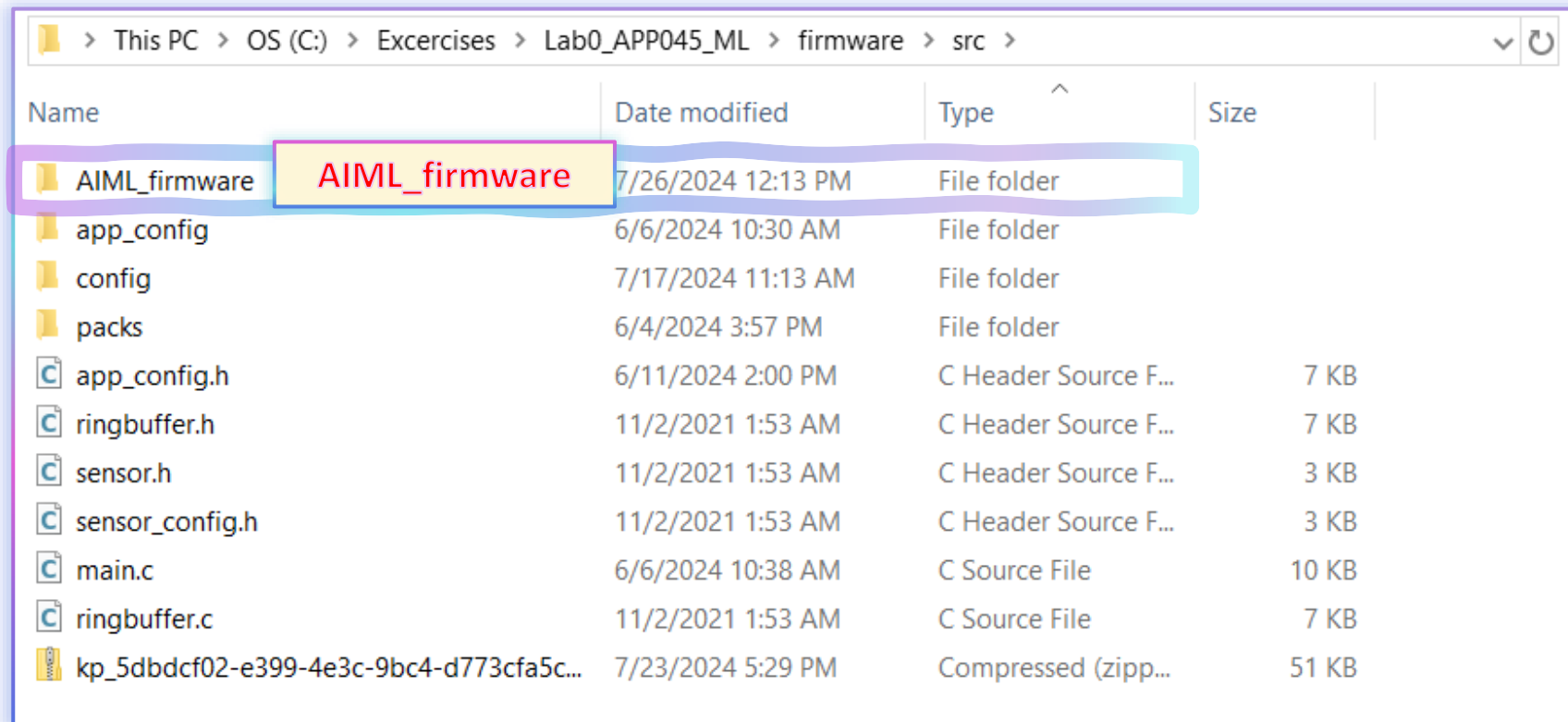
- Save the zip file under “**Lab0_APP045_ML /firmware/src**”.



Lab5 Export & Deploy the Model

Step 6

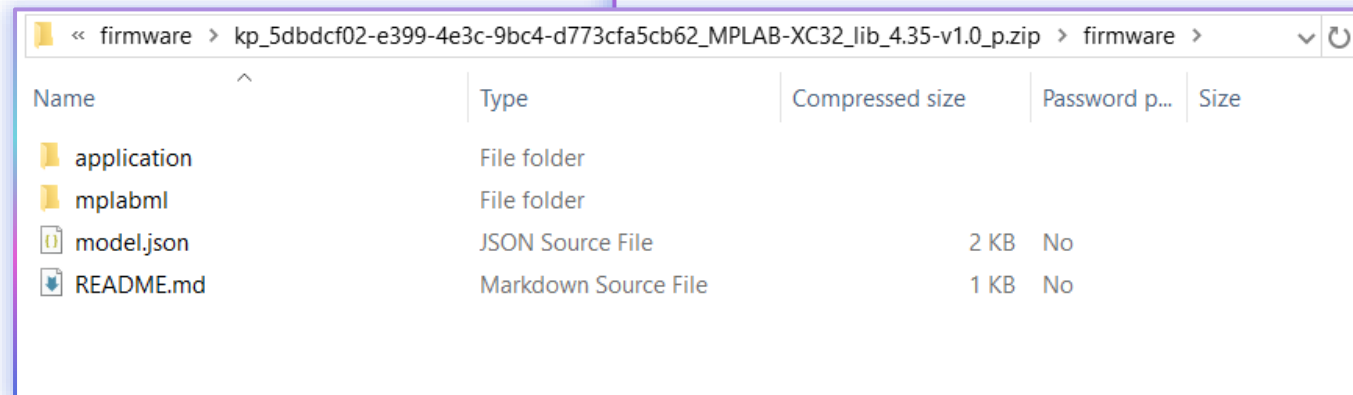
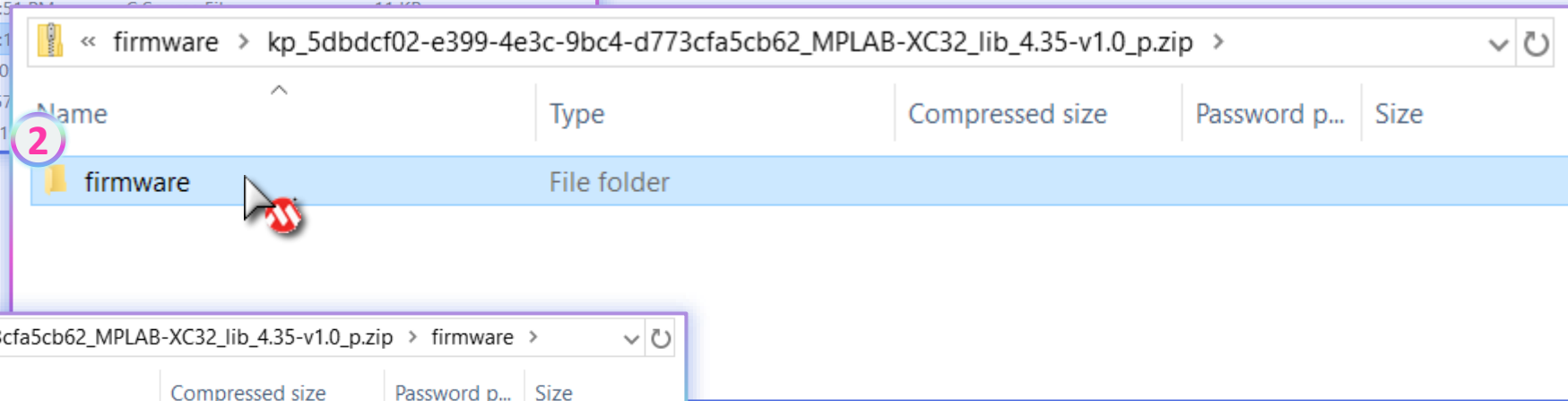
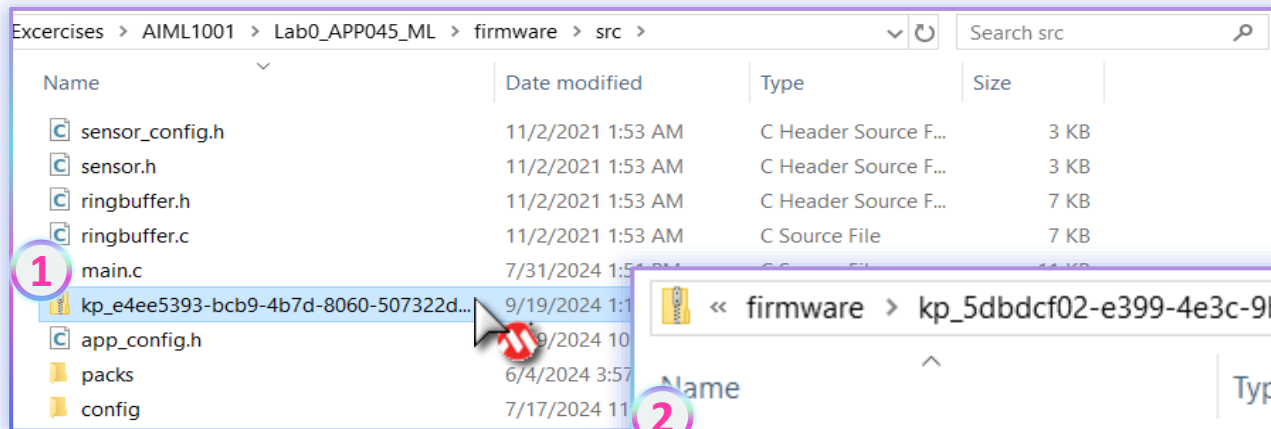
- Create a “**AIML_firmware**” folder “firmware/src” folder.



Lab5 Export & Deploy the Model

Step 7

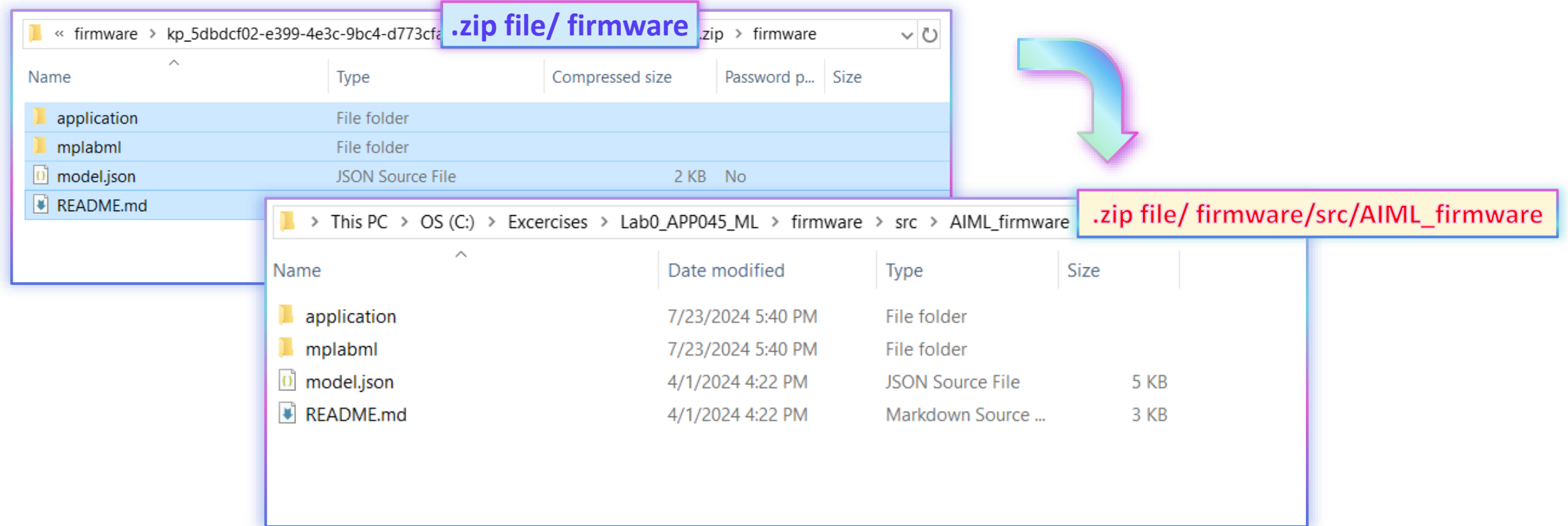
- Double click to open the .zip file, open the firmware folder.



Lab5 Export & Deploy the Model

Step 8

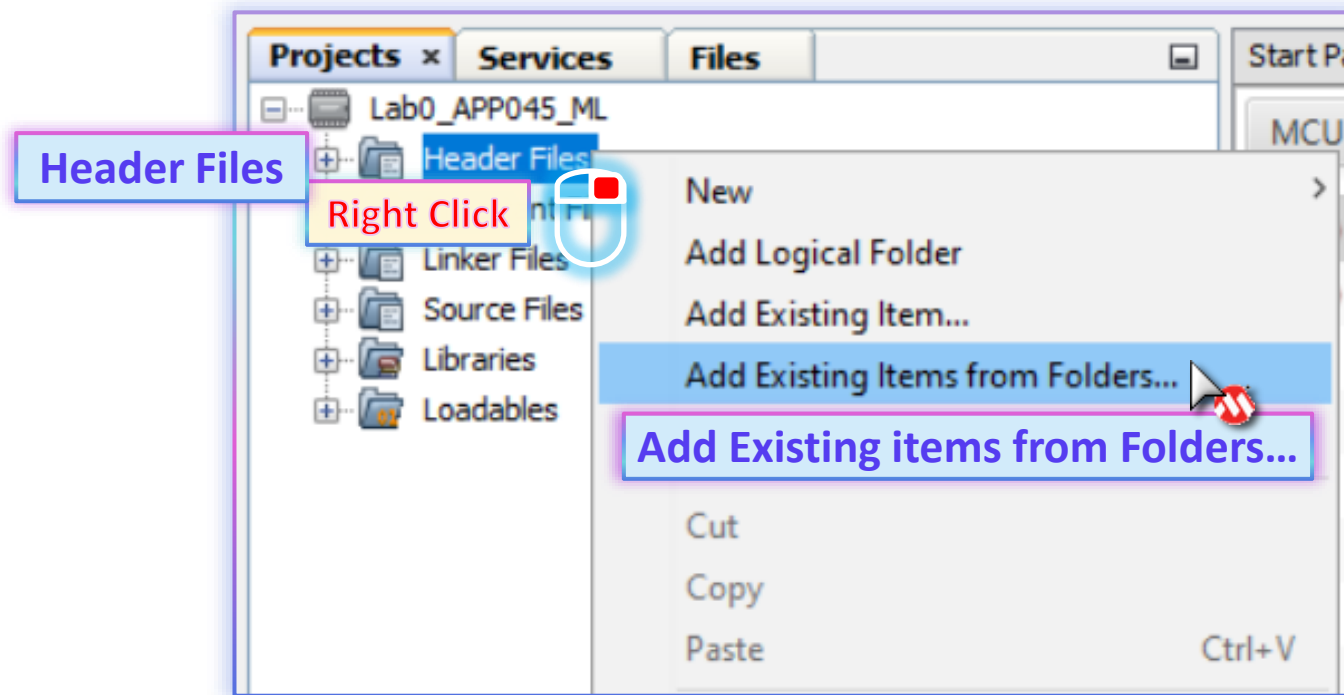
- Copy and paste all the files to “[firmware/src/AIML_firmware/](#)”.



Lab5 Export & Deploy the Model

Step 9

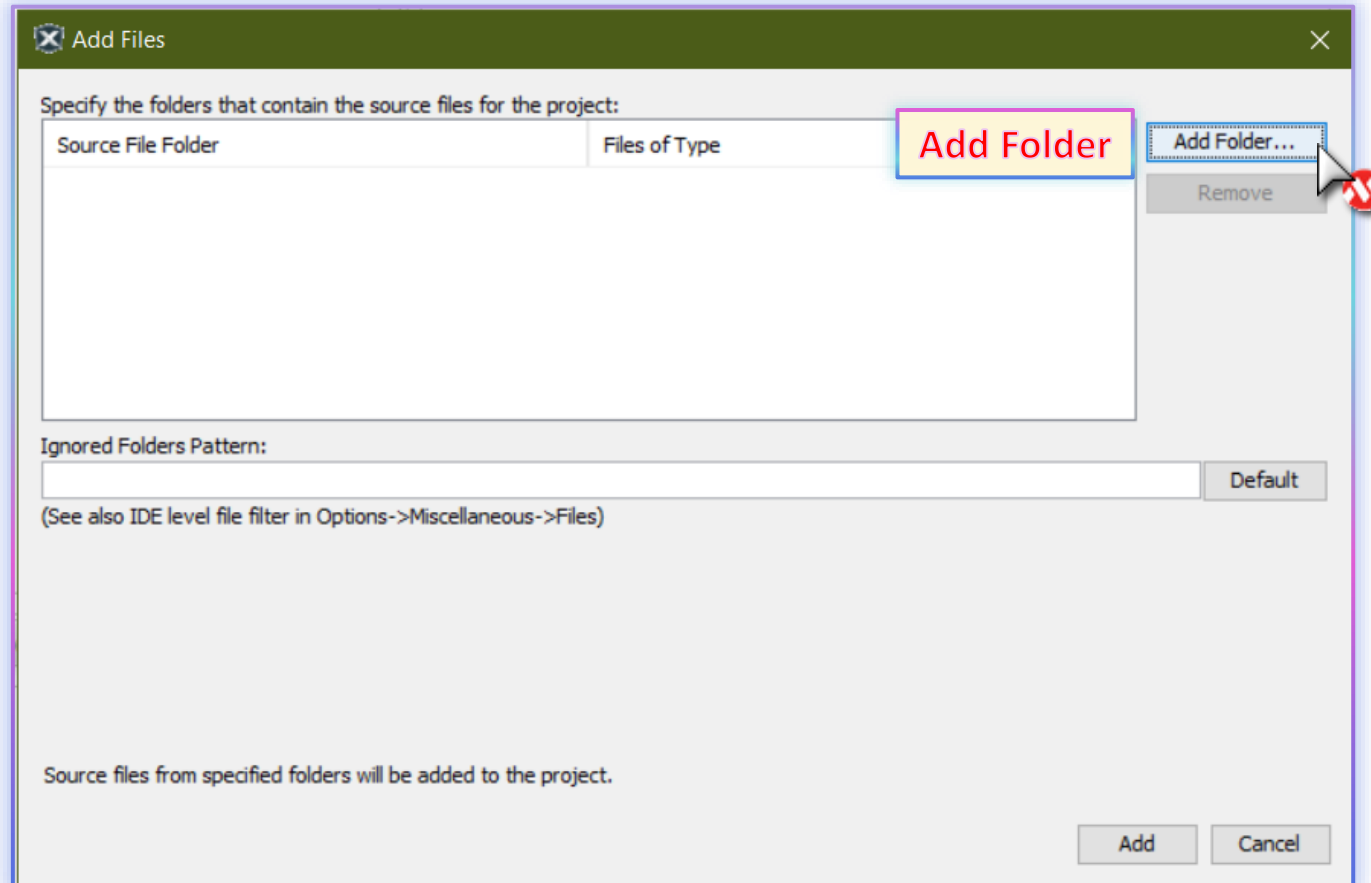
- Back to MPLAB X IDE. Add those file into the project.
- Right click **Header Files** -> **Add Existing Items from Folders.**



Lab5 Export & Deploy the Model

Step 10

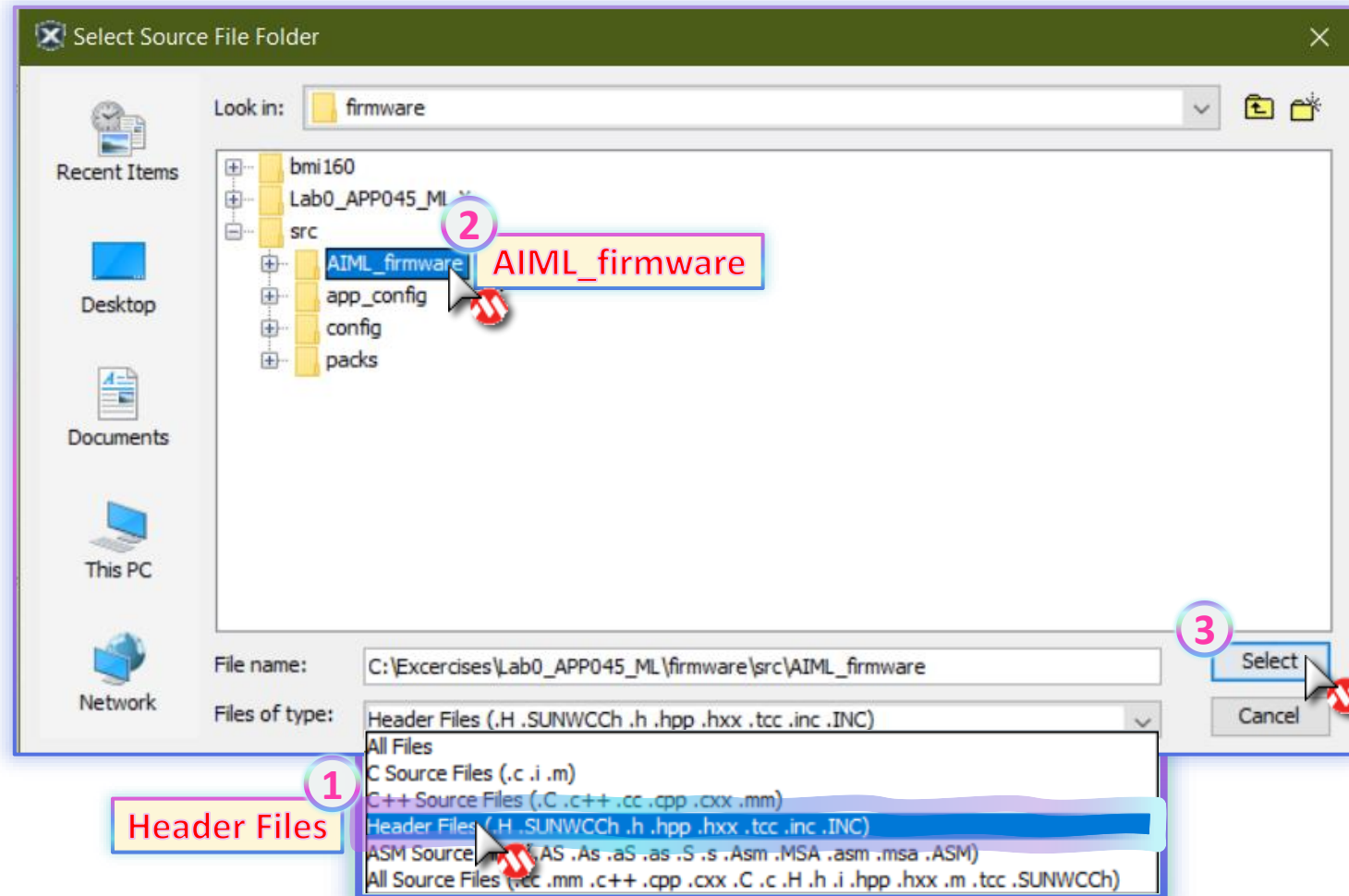
- Click Add Folder



Lab5 Export & Deploy the Model

Step 11

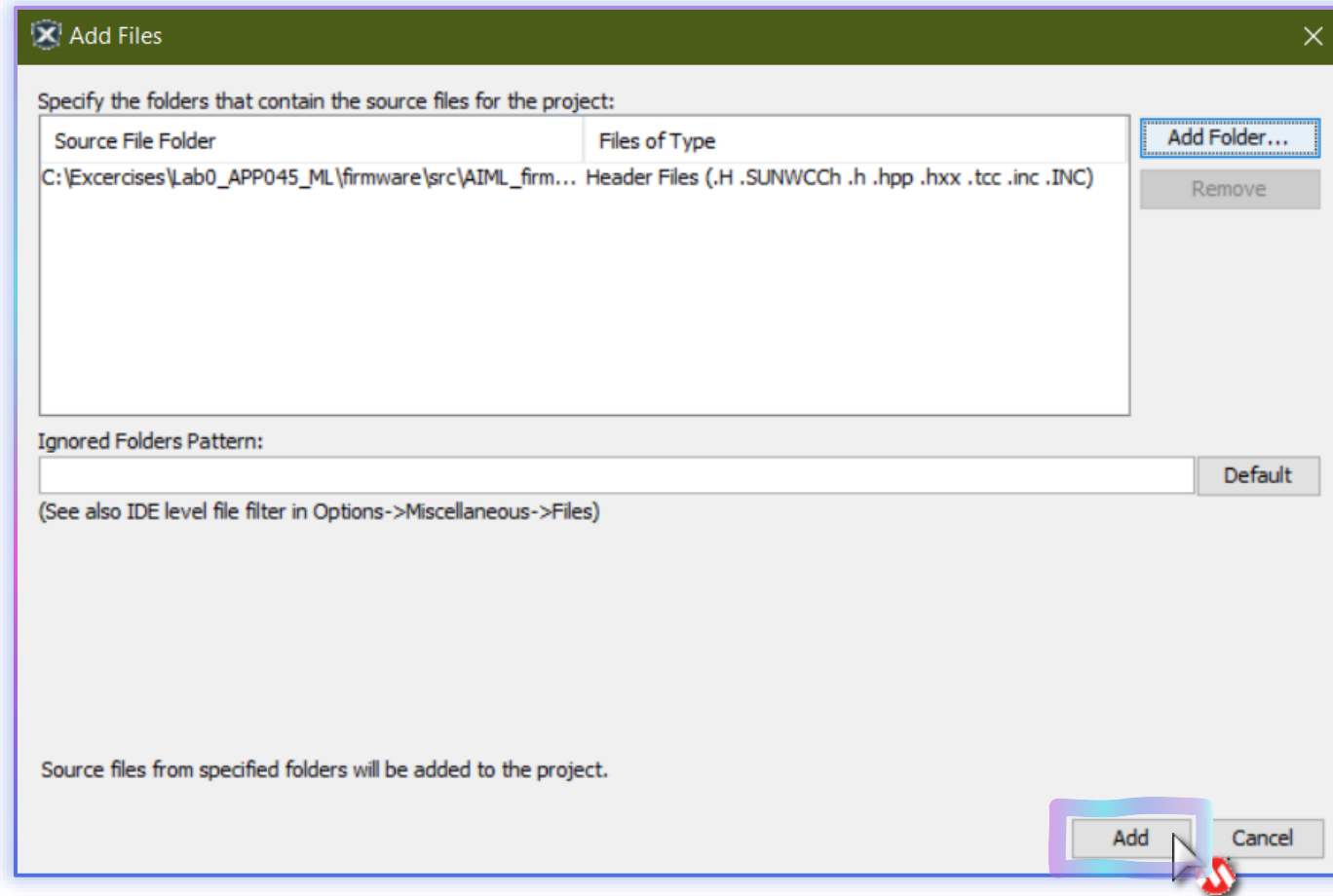
- Select folder “AIML_firmware.”



Lab5 Export & Deploy the Model

Step 12

- All the .h file will add in the project.



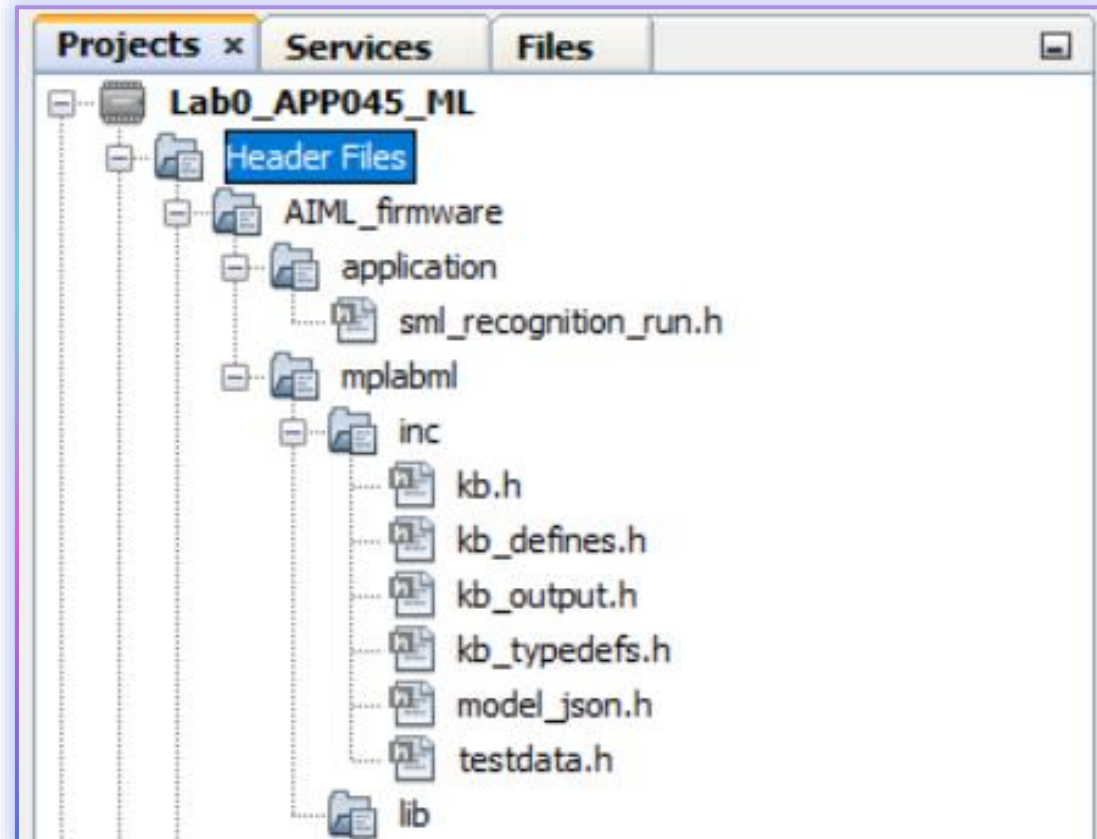
Lab5 Export & Deploy the Model

Step 13

- Check and make sure that all the .h files are added into the project.

AIML_firmware

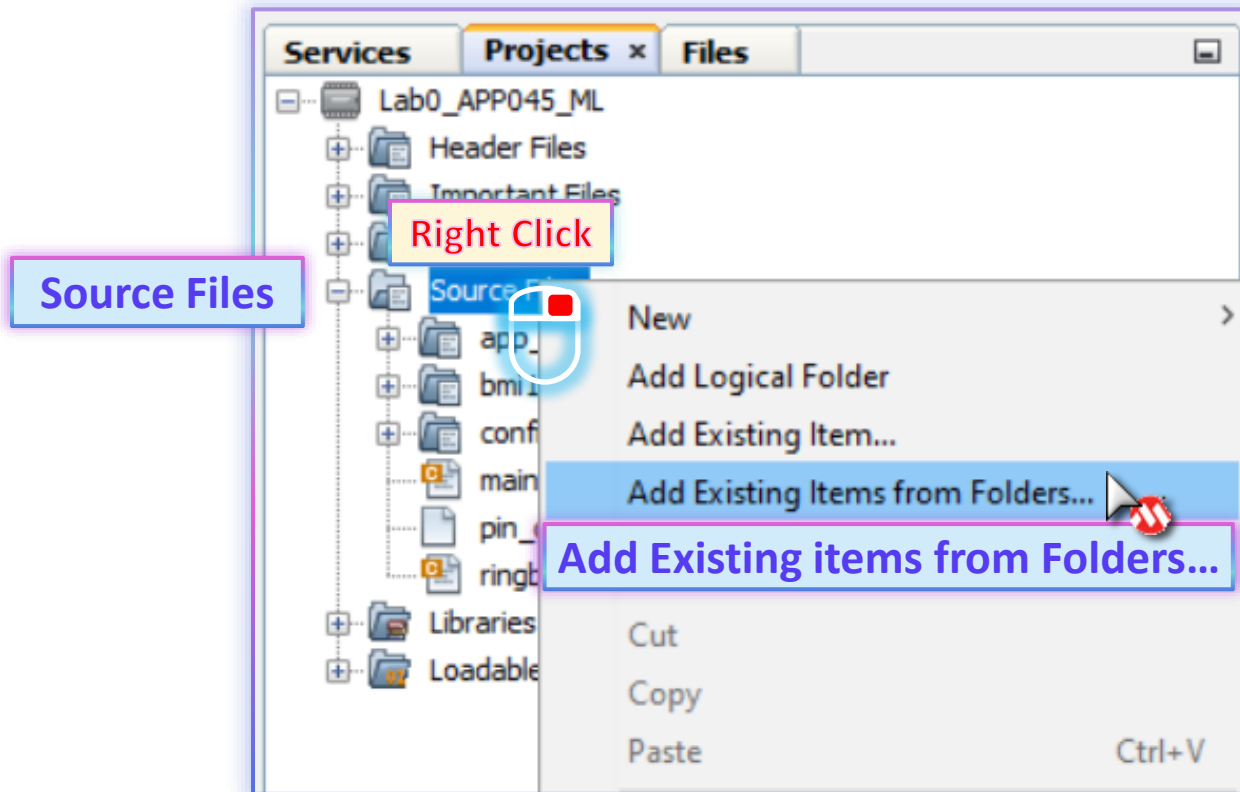
- Application
 - sml_recognition_run.h
- mplabml
 - Inc
 - kb.h
 - kb_defines.h
 - kb_output.h
 - Kb_typedefs.h
 - model_json.h
 - testdata.h
 - lib



Lab5 Export & Deploy the Model

Step 14

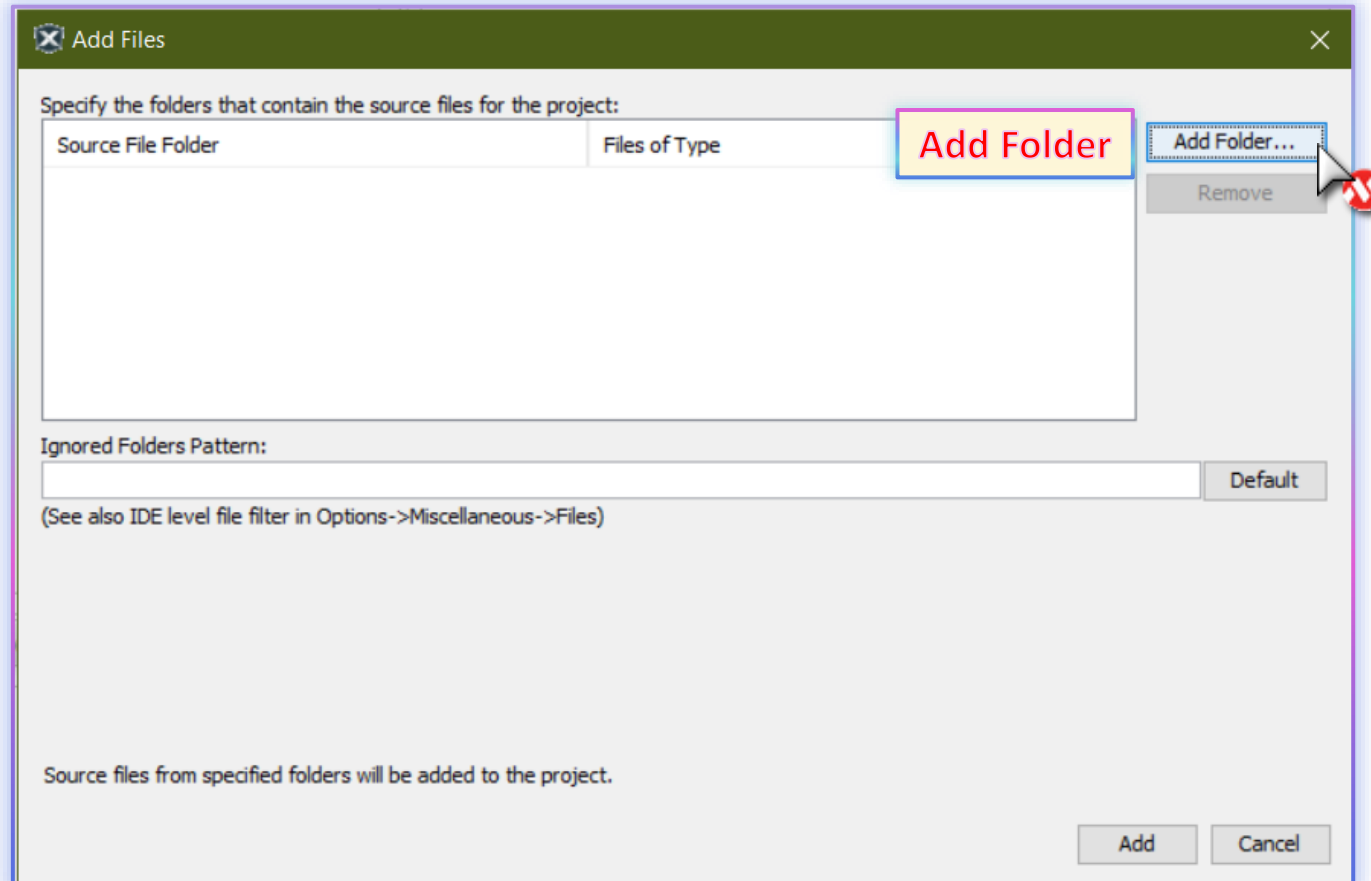
- Back to MPLAB X IDE. Add those file into the project.
- Right click **Source Files** -> **Add Existing Items from Folders.**



Lab5 Export & Deploy the Model

Step 15

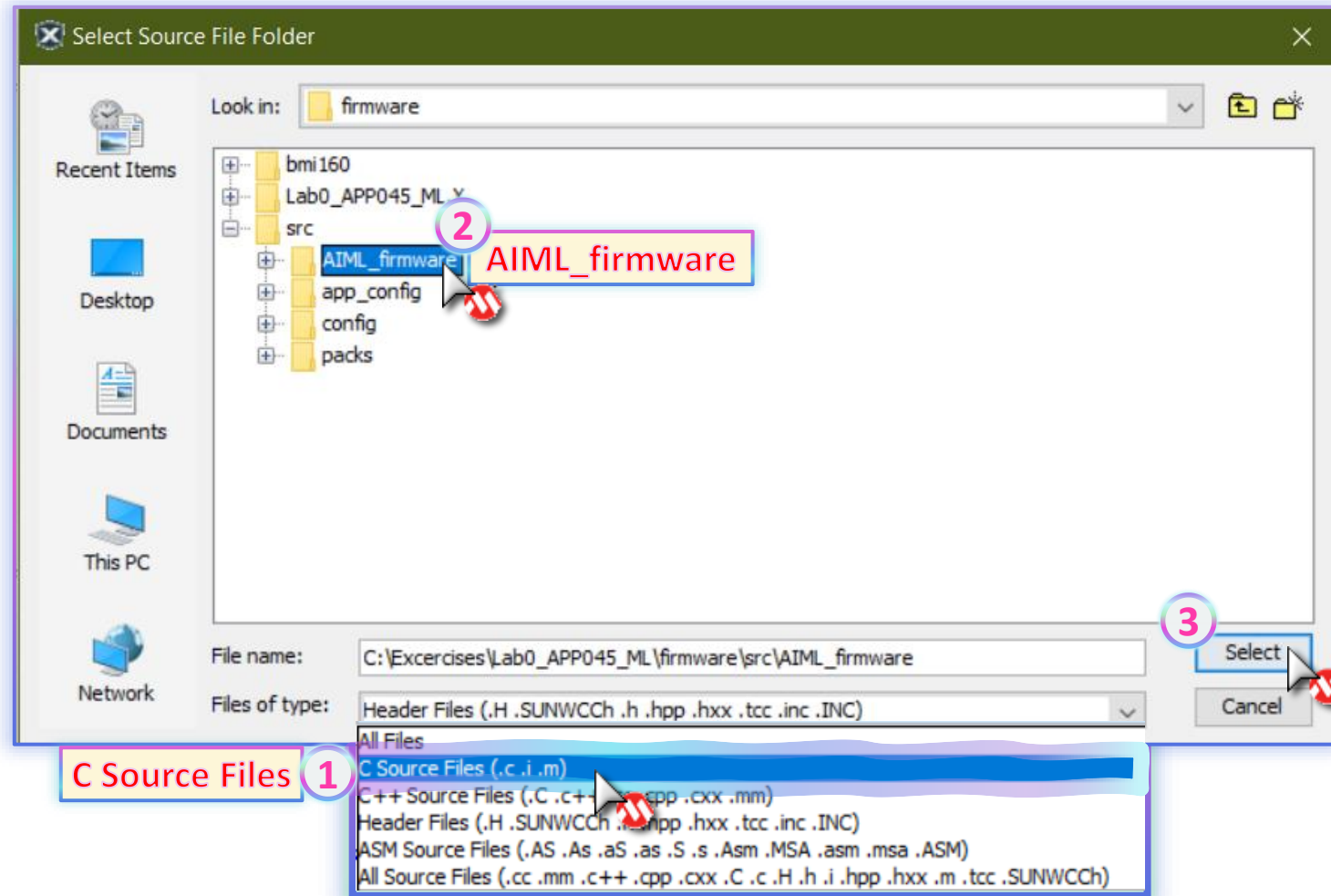
- Click Add Folder



Lab5 Export & Deploy the Model

Step 16

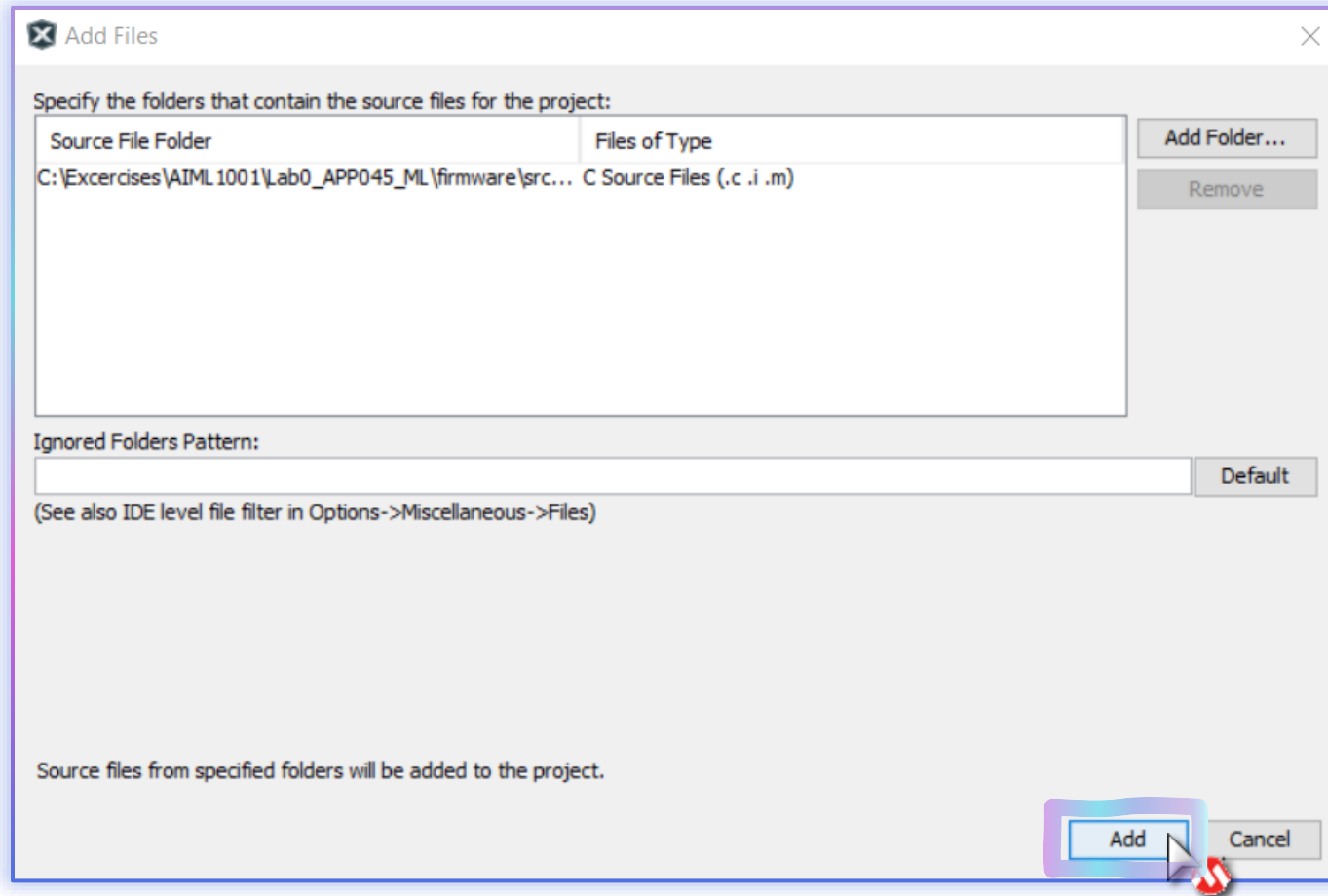
- Select folder “AIML_firmware.”



Lab5 Export & Deploy the Model

Step 17

- All the .c file will add in the project.



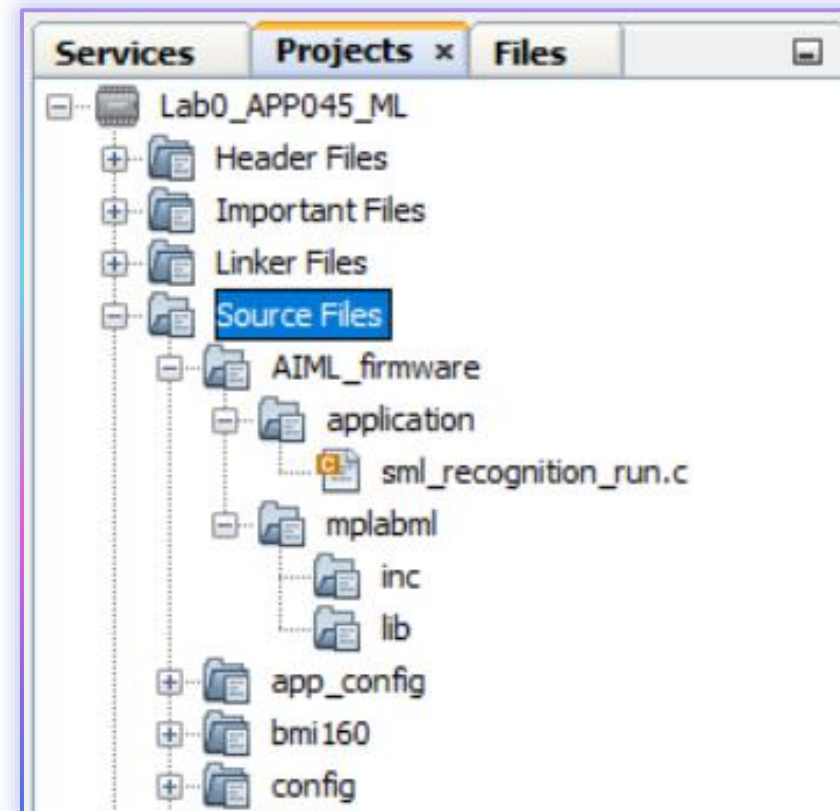
Lab5 Export & Deploy the Model

Step 18

- Check and make sure that all the .c files are added into the project.

AIML_firmware

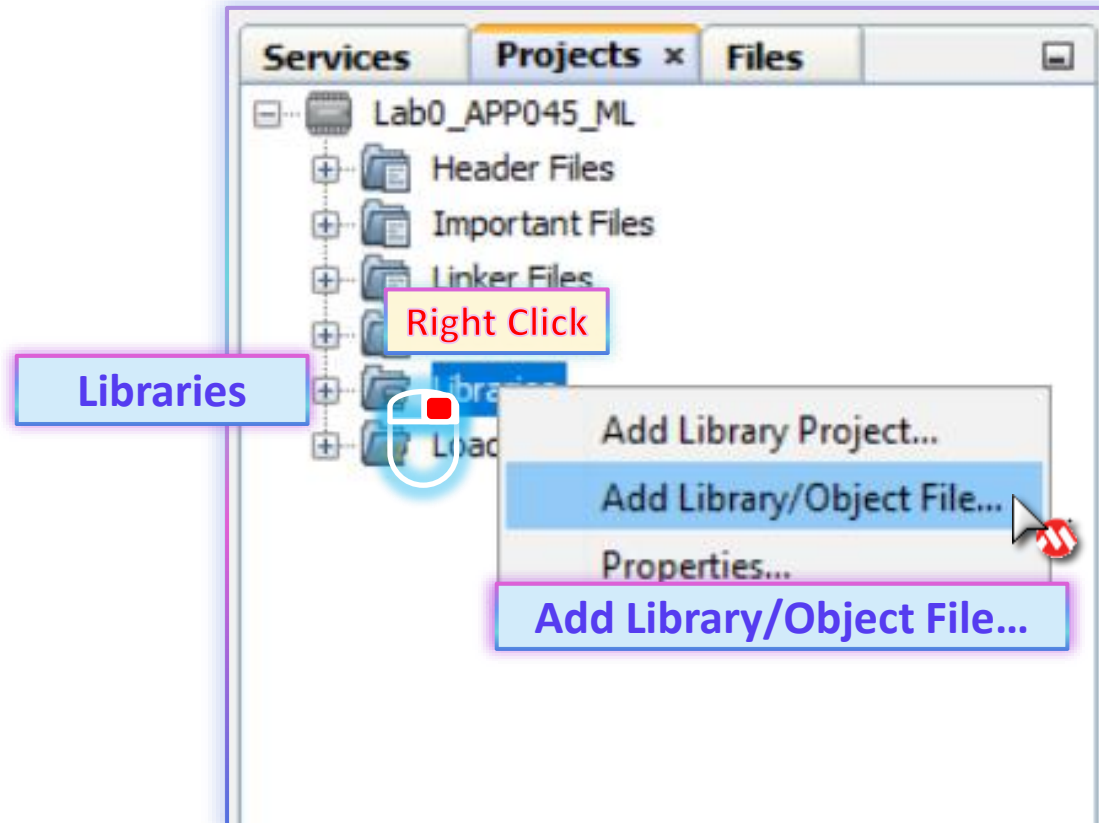
- Application
 - sml_recognition_run.c
- mplabml
 - Inc
 - lib



Lab5 Export & Deploy the Model

Step 19

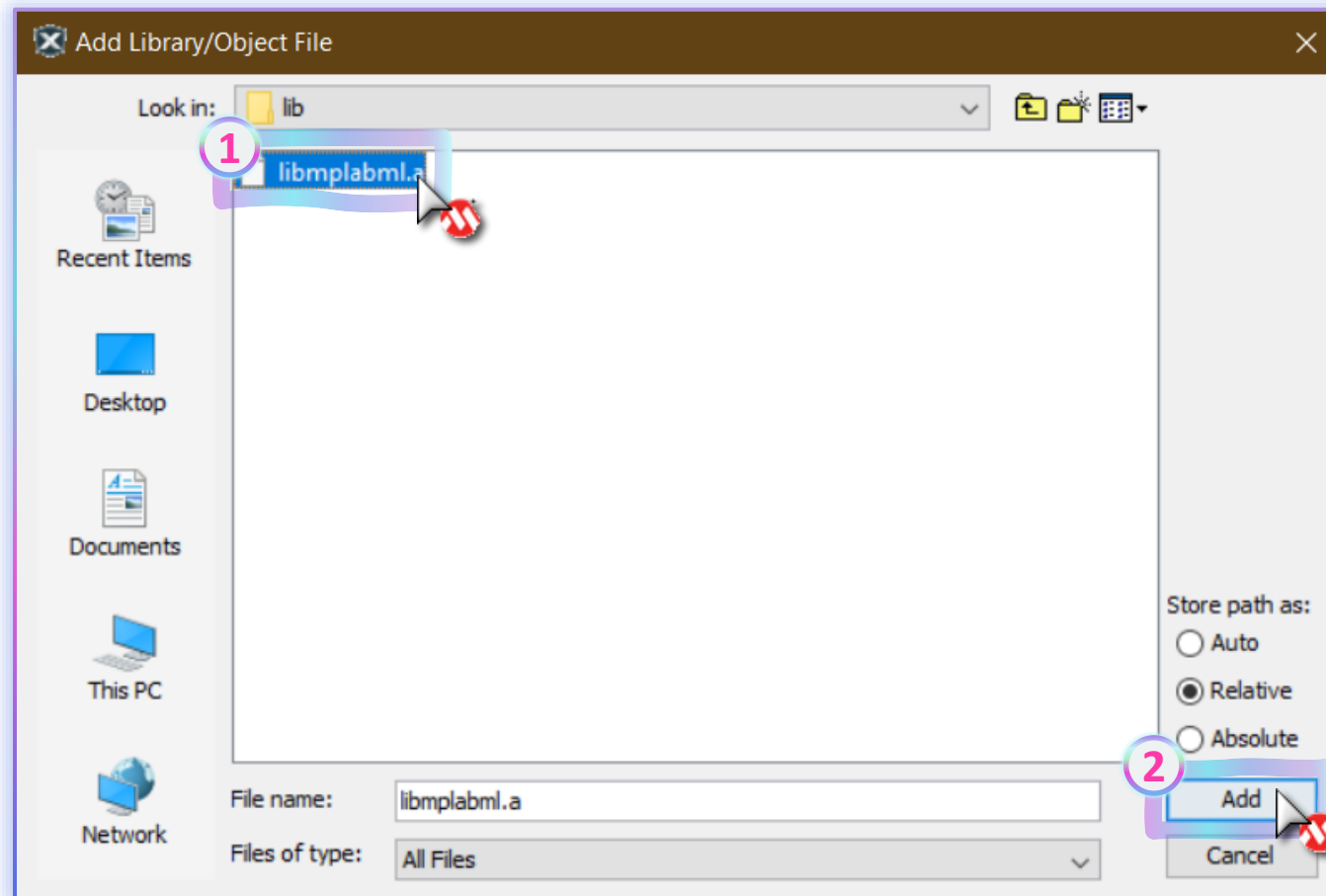
- Add **Libraries** in the project.
- Right click **Libraries** -> **Add Library/Object File...**



Lab5 Export & Deploy the Model

Step 20

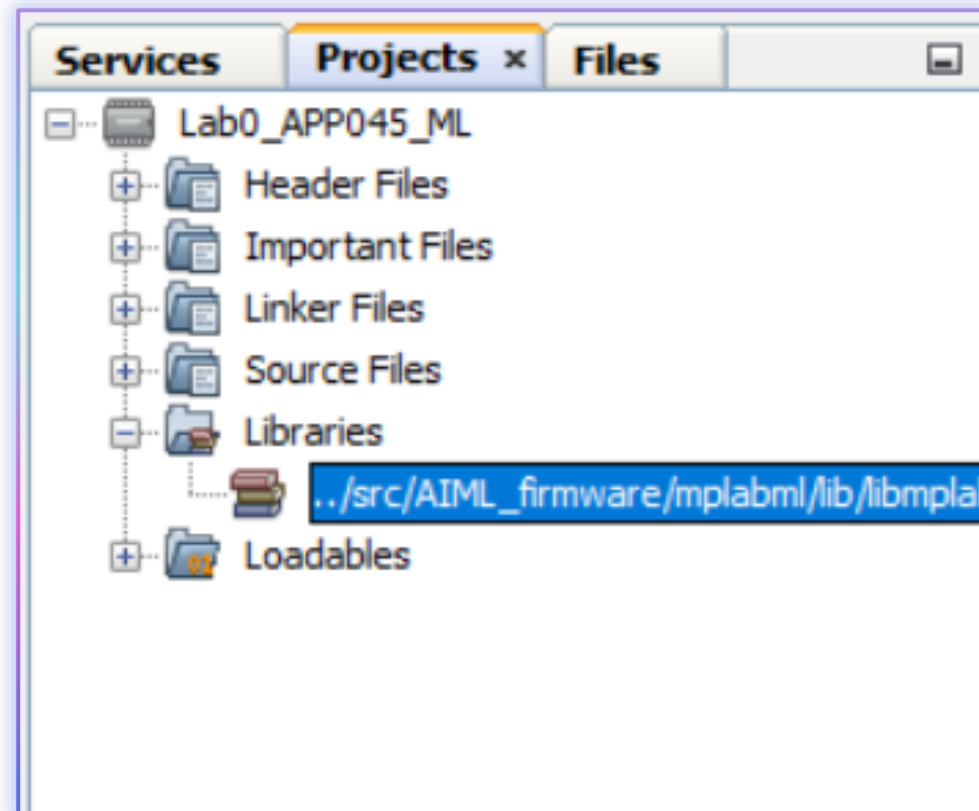
- Find “[AIML_firmware/mplabml/lib/libmplabml.a](#)”



Lab5 Export & Deploy the Model

Step 21

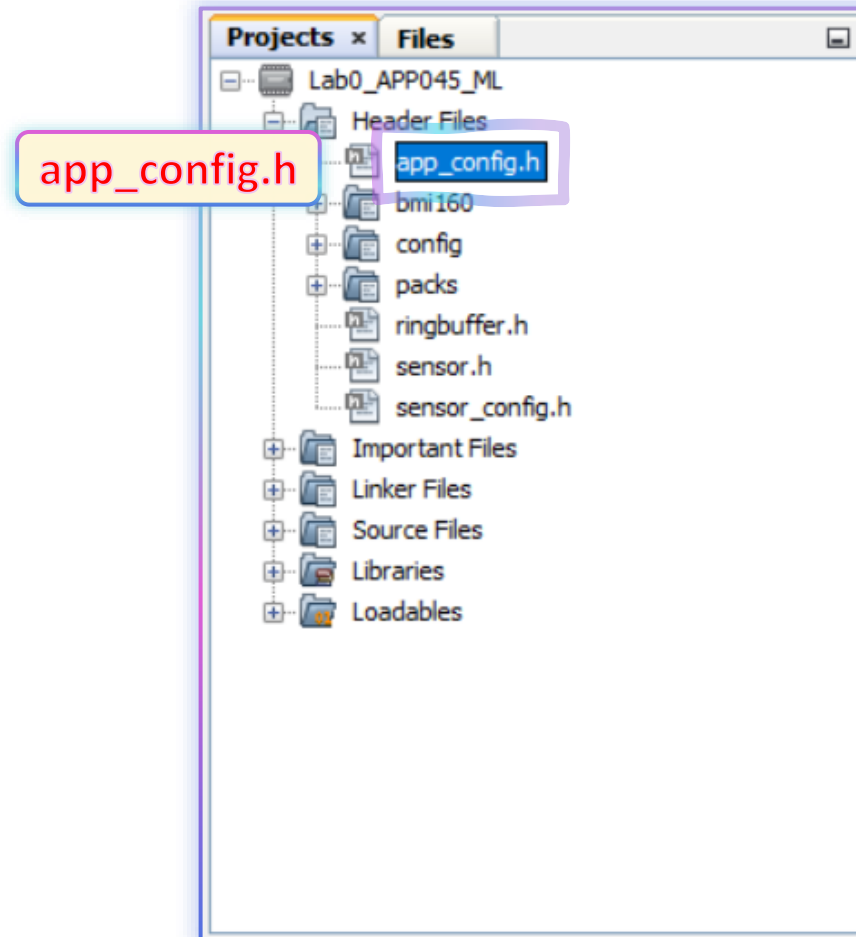
- Check and make sure the Library is added into the project.



Lab5 Export & Deploy the Model

Step 22

- Projects > Header Files > app_config.h



Lab5 Export & Deploy the Model

Step 23

Select the Data Streaming Formats

- Line 60

```
#define DATA_STREAMER_FORMAT DATA_STREAMER_FORMAT_NONE
```

```
58 | // Data streaming formatting selection
59 | #ifndef DATA_STREAMER_FORMAT
60 | #define DATA_STREAMER_FORMAT DATA_STREAMER_FORMAT_NONE
61 | #endif
62 |
```

Lab5 Export & Deploy the Model

Step 24

Change the file path in “`AIML_firmware/application/sml_recognition_run.h`” .

Line 3

`#include "kb.h"` to `#include "../mplabml/inc/kb.h"`

```
1  #ifndef __SML_RECOGNITION_RUN_H__
2  #define __SML_RECOGNITION_RUN_H__
3  #include "../mplabml/inc/kb.h"
4
5  int32_t sml_recognition_run(int16_t *data, int32_t num_sensors);
6
7  #endif // __SML_RECOGNITION_RUN_H__
8
```

Lab5 Export & Deploy the Model

Step 25

Change the file path in “`AIML_firmware/application/sml_recognition_run.c`” .

Line 1 & 2

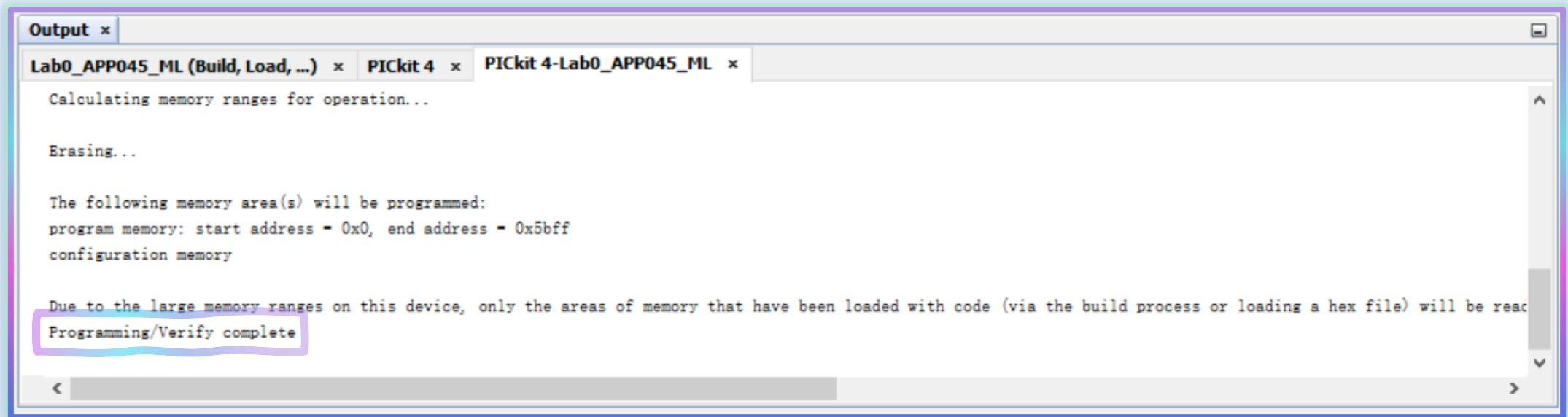
`#include "kb.h"` to `#include "../mplabml/inc/kb.h"`
`#include "kb_output.h"` to `#include "../mplabml/inc/kb_output.h"`

```
1  #include "../mplabml/inc/kb.h"
2  #include "../mplabml/inc/kb_output.h"
3  #include <string.h>
4  #ifdef SML_USE_TEST_DATA
5  #include "testdata.h"
6  int32_t td_index = 0;
7  #endif // SML_USE_TEST_DATA
8
```


Lab 0 Initial Setup

Step 25

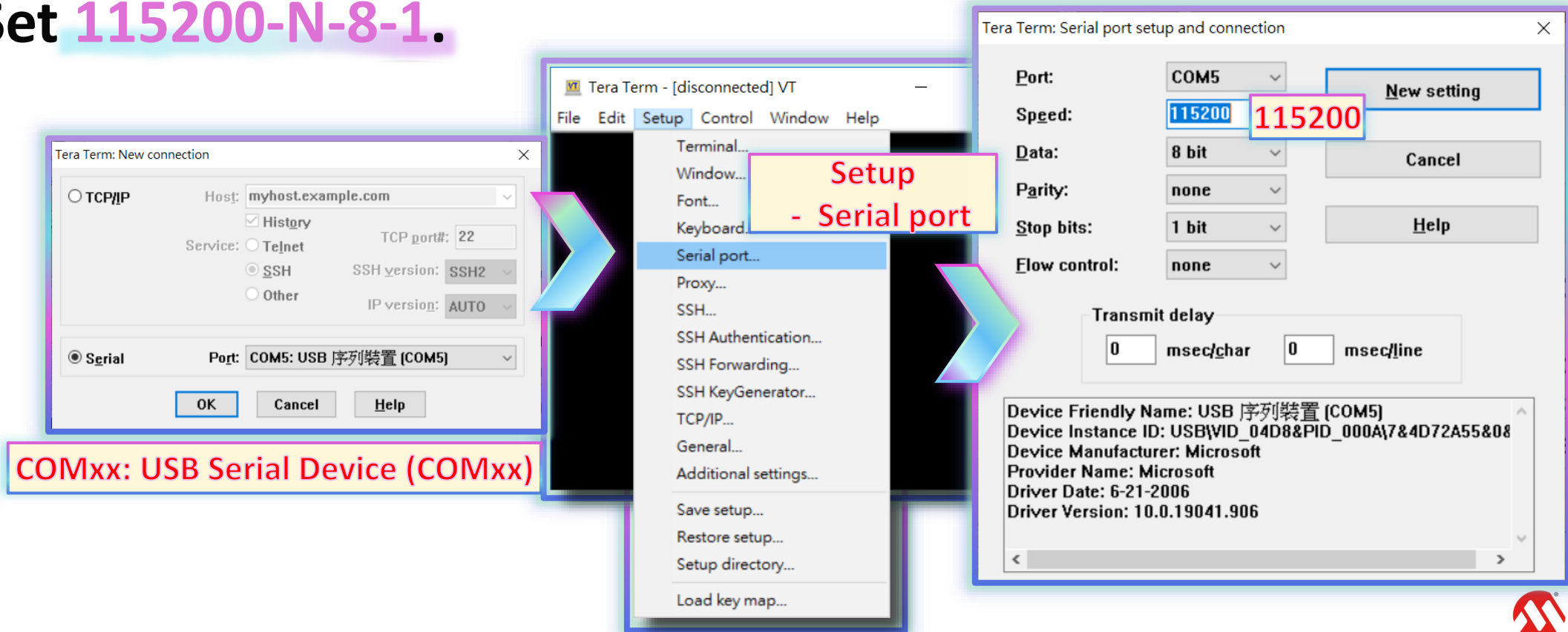
- Clean and Build Project
- Click  icon to Program the Project.
- Output Window will show *Programming/Verify Complete.*



Lab5 Export & Deploy the Model

Step 26

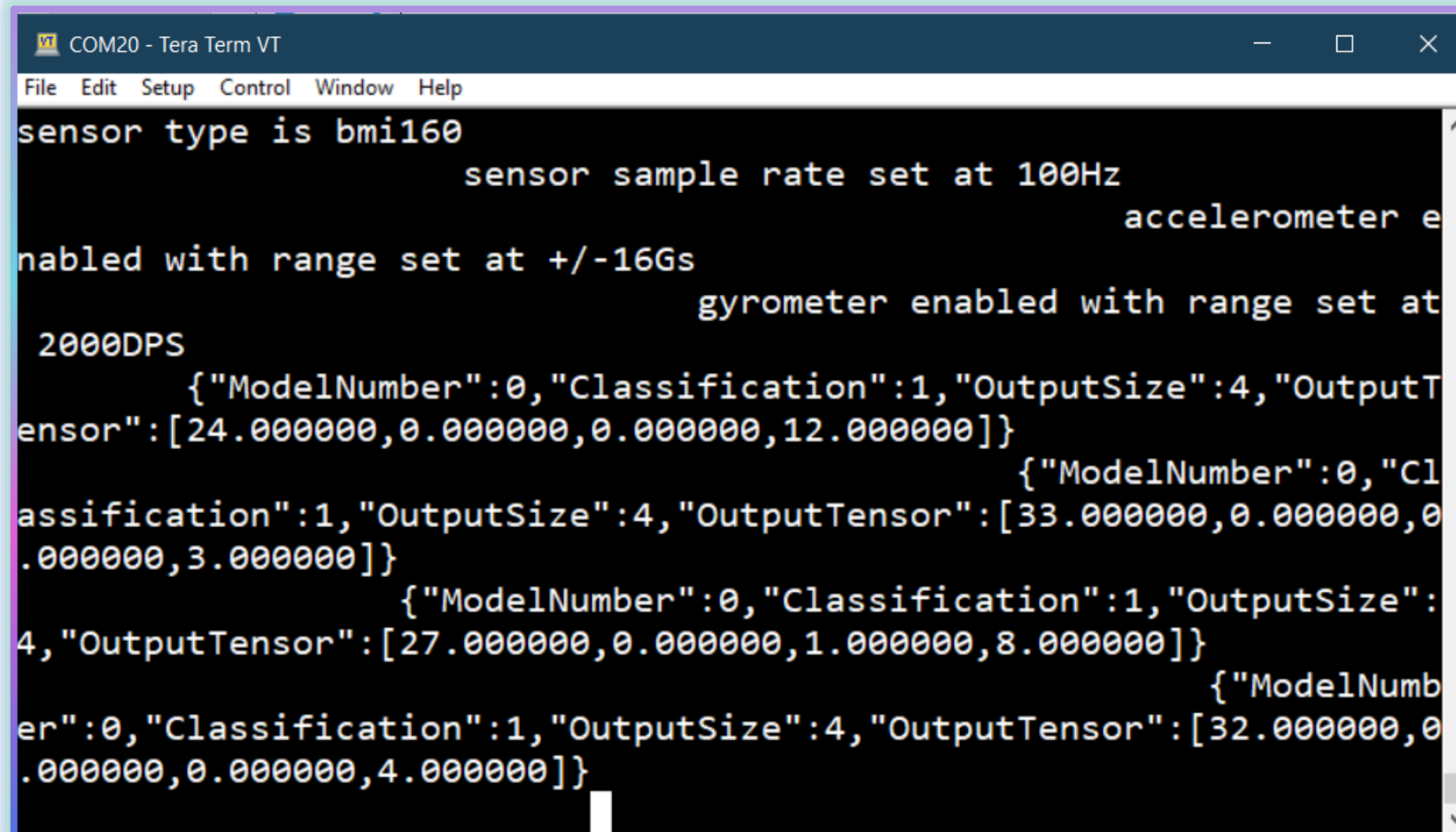
- Open Tera Term as the terminal console software.
- Select COMxx: **USB Serial Device (COMxx)**
- Set **115200-N-8-1**.



Lab5 Export & Deploy the Model

Result

- Show the classification result every 1 second (100 Mz).

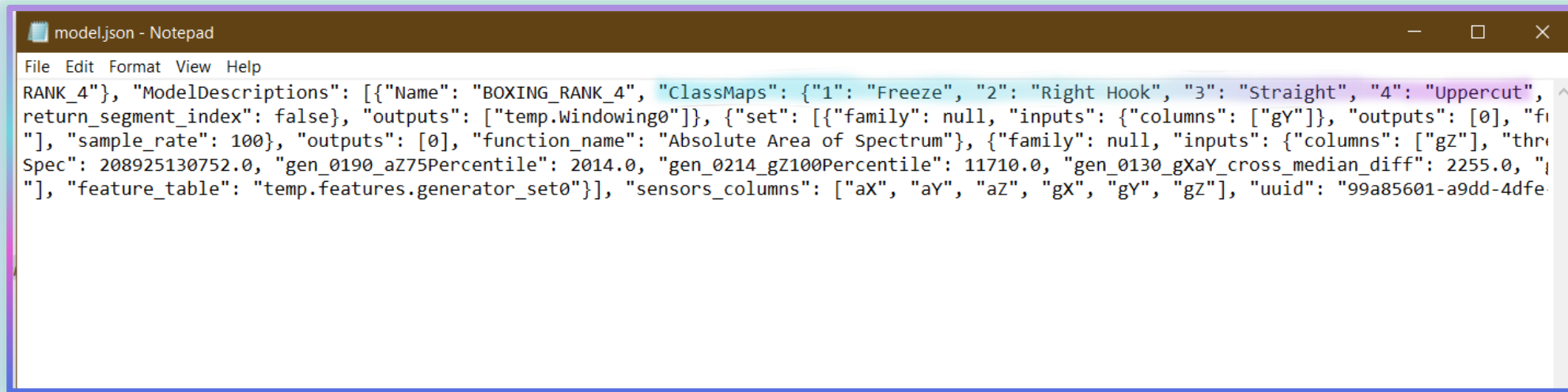


```
COM20 - Tera Term VT
File Edit Setup Control Window Help
sensor type is bmi160
                sensor sample rate set at 100Hz
                                accelerometer e
nabled with range set at +/-16Gs
                                gyrometer enabled with range set at
2000DPS
        {"ModelNumber":0,"Classification":1,"OutputSize":4,"OutputTensor":
[24.000000,0.000000,0.000000,12.000000]}
                                {"ModelNumber":0,"Cl
assification":1,"OutputSize":4,"OutputTensor":[33.000000,0.000000,0
.000000,3.000000]}
        {"ModelNumber":0,"Classification":1,"OutputSize":
4,"OutputTensor":[27.000000,0.000000,1.000000,8.000000]}
                                {"ModelNumber":0,"Classification":1,"OutputSize":4,"OutputTensor":[32.000000,0
.000000,0.000000,4.000000]}
```

Lab5 Export & Deploy the Model

Result

- What is the definition of the classification?
- Find .json file under “**AIML_firmware/model.json**”

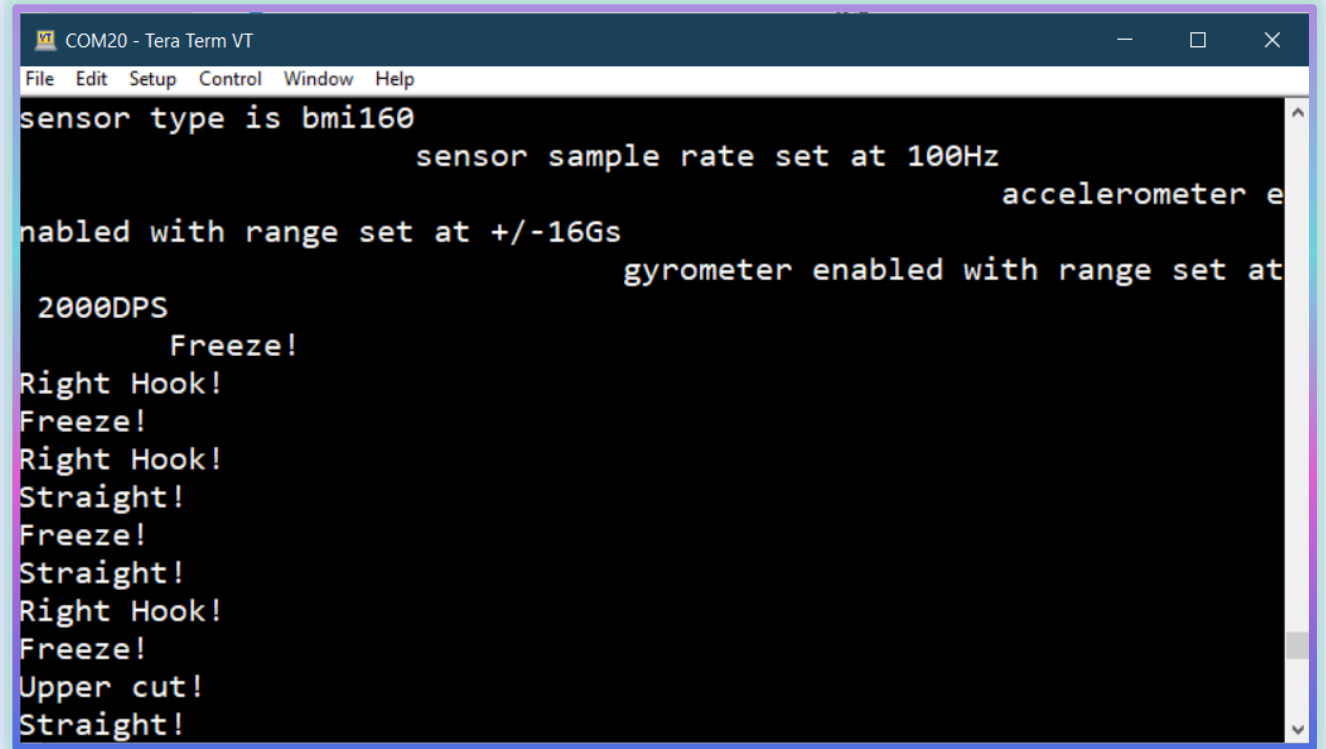


```
model.json - Notepad
File Edit Format View Help
RANK_4"}, "ModelDescriptions": [{"Name": "BOXING_RANK_4", "ClassMaps": {"1": "Freeze", "2": "Right Hook", "3": "Straight", "4": "Uppercut",
return_segment_index": false}, "outputs": ["temp.Windowing0"]}, {"set": [{"family": null, "inputs": {"columns": ["gY"]}, "outputs": [0], "fi
"], "sample_rate": 100}, "outputs": [0], "function_name": "Absolute Area of Spectrum"}, {"family": null, "inputs": {"columns": ["gZ"], "thre
Spec": 208925130752.0, "gen_0190_aZ75Percentile": 2014.0, "gen_0214_gZ100Percentile": 11710.0, "gen_0130_gXaY_cross_median_diff": 2255.0, "
"], "feature_table": "temp.features.generator_set0"}], "sensors_columns": ["aX", "aY", "aZ", "gX", "gY", "gZ"], "uuid": "99a85601-a9dd-4dfe
```

Lab5 Export & Deploy the Model

Result

- Show the result in a more humanity way.
 - “Main.c”
 - Uncomment Line 279 to 290.
 - “AIML_firmware/application/sml_recognition_run.c”
 - Comment Line 17.
- Compile and Program!

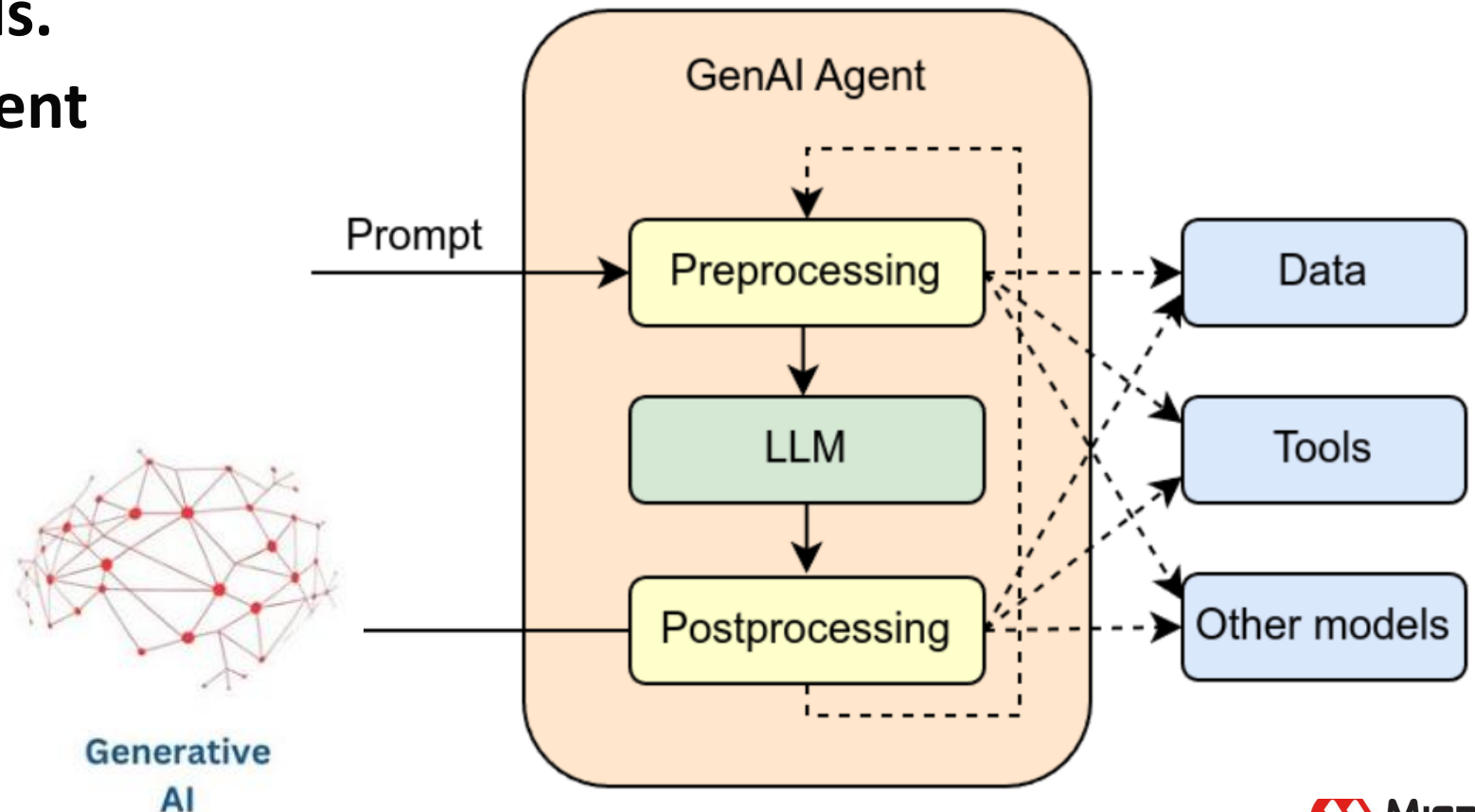


```
COM20 - Tera Term VT
File Edit Setup Control Window Help
sensor type is bmi160
                        sensor sample rate set at 100Hz
                        accelerometer e
nabled with range set at +/-16Gs
                        gyrometer enabled with range set at
2000DPS
                        Freeze!
Right Hook!
Freeze!
Right Hook!
Straight!
Freeze!
Straight!
Right Hook!
Freeze!
Upper cut!
Straight!
```

Large Language Models

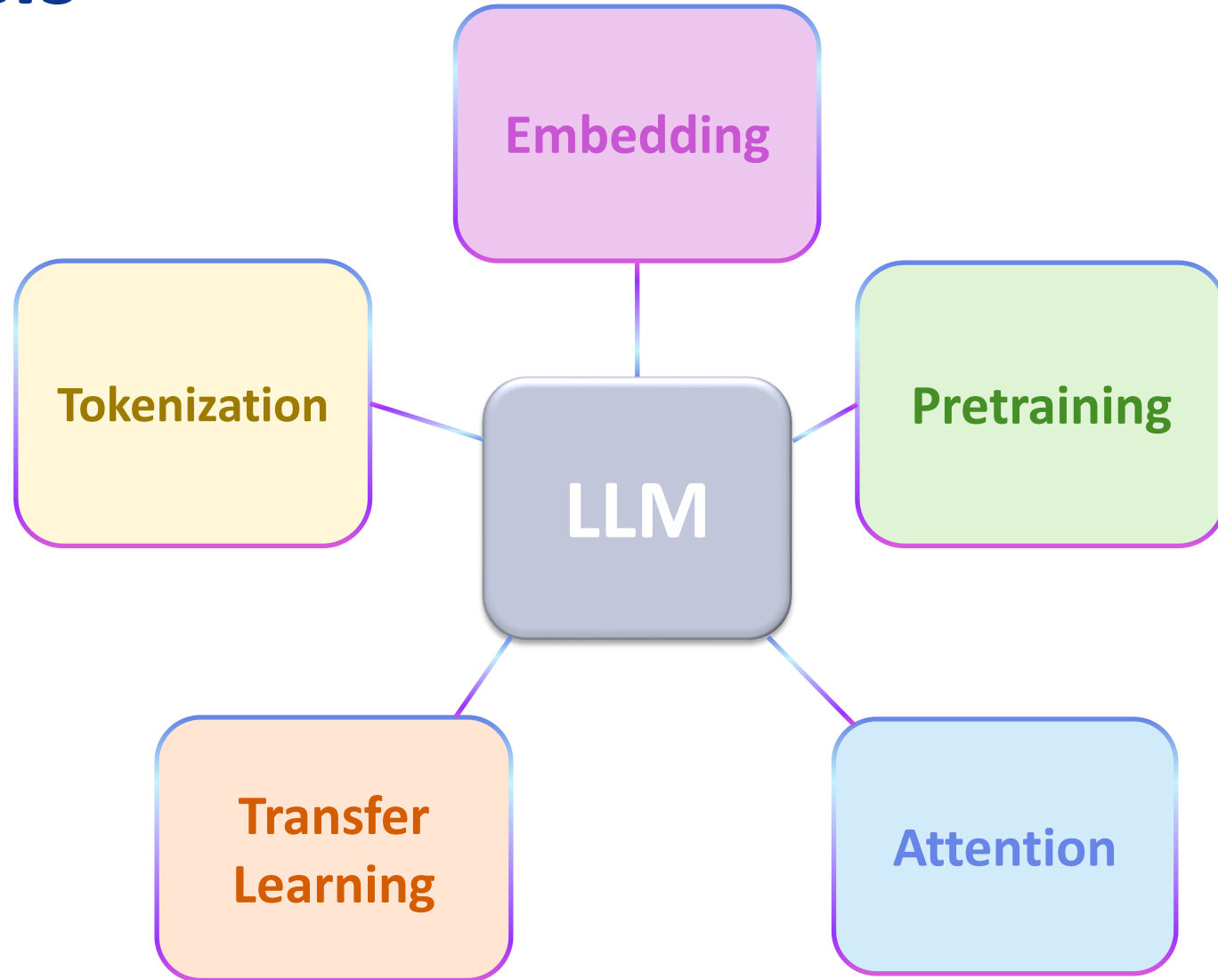
Generative AI

- G-AI models learn patterns and structures of input data and produce new data with similar characteristics.
- Input data such as instructions, images, etc. for G-AI is often prompted by Large Language Models.
- G-AI creates new content such as text, videos., pictures, audio, etc.



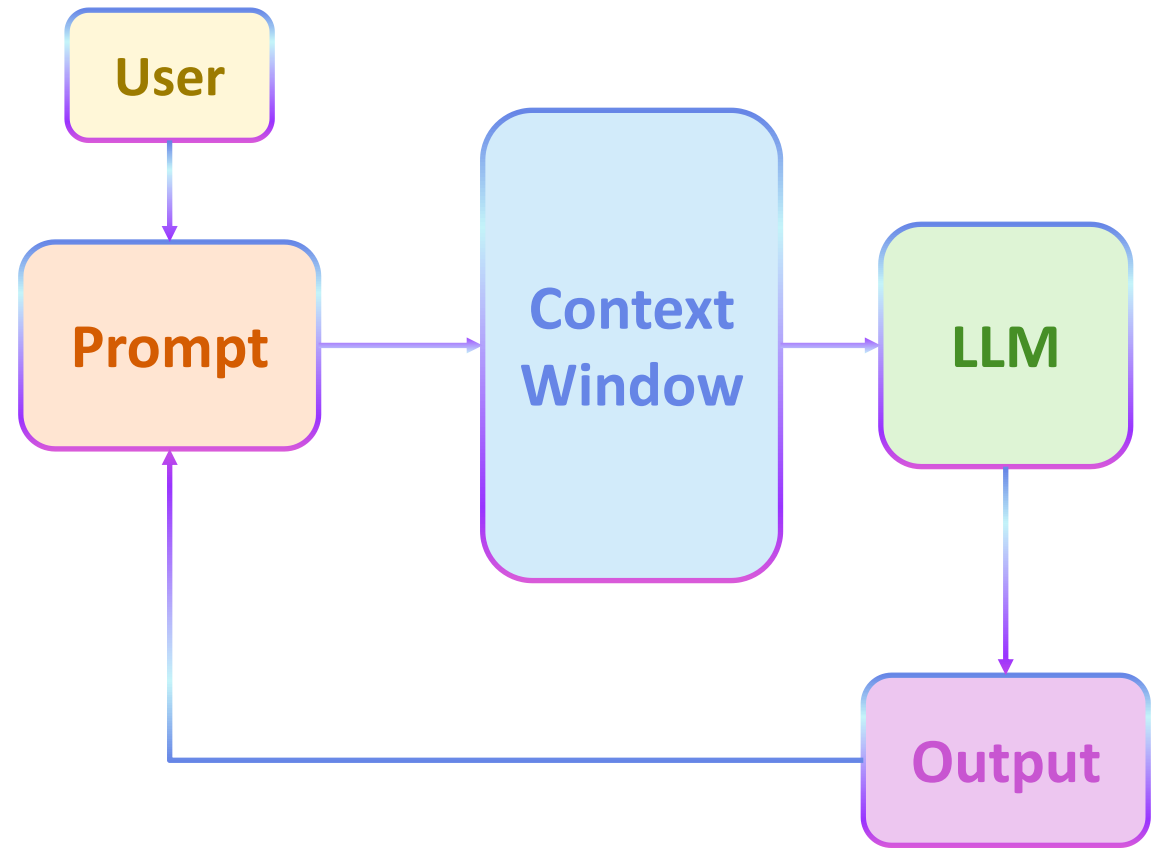
Large Language Models

- LLM's are transformer-based Deep Neural Networks(DNN)
- Transformer is a DL architecture developed by Google and based on multi-head attention mechanism.
- Transfer Learning
- Popular LLM:
 - OpenAI's ChatGPT
 - Microsoft's Copilot
 - Meta's LLaMA
 - Anthropic's Claude model
 - Mistral AI's models



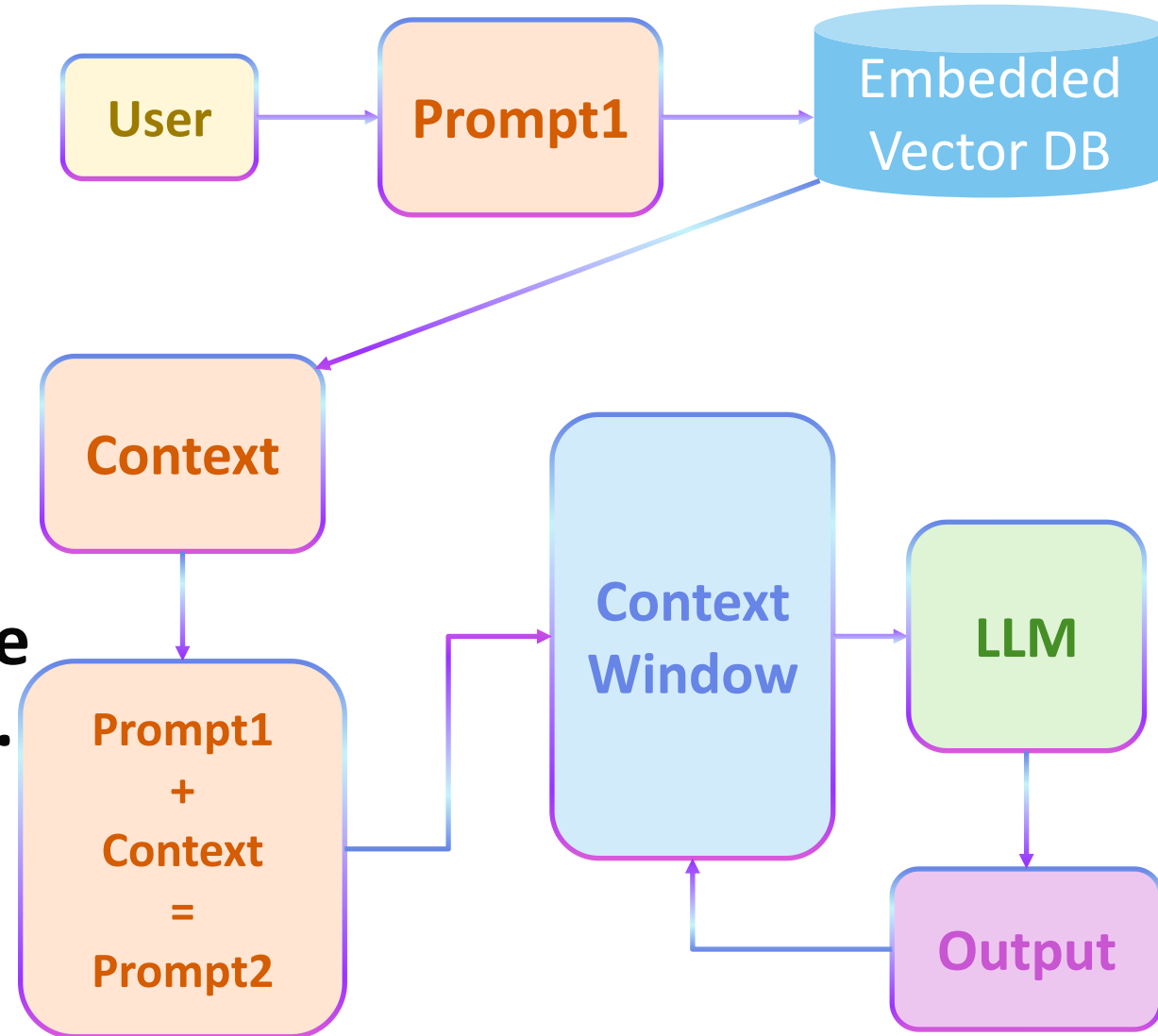
Interfacing with basic LLM prompt

- User initiate session with an initial prompt.
- Prompt goes to LLM context window.
- LLM uses prompt to generate output.
- Output is added to the context window along with additional prompts.



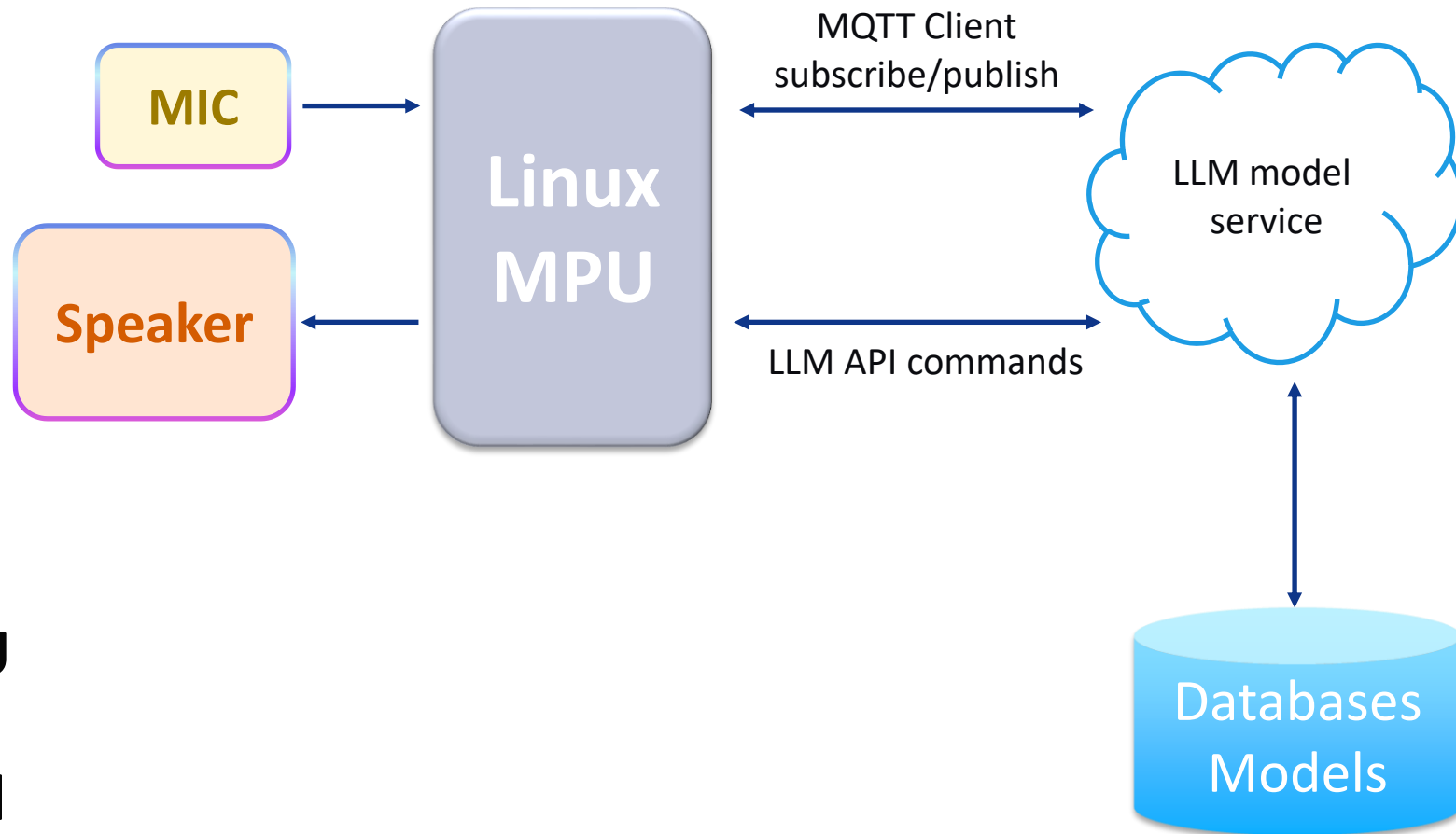
Retrieval Augmented Generation LLM prompt

- RAG provides a performance improvement as compared with LLM alone by reducing bad responses and focusing LLM on relevant data.
- Initiates prompt will search in the vector DB.
- The original prompt is added to the context and the expanded prompt.
- LLM generates output.
- Add New prompts until the output is satisfactory.



Connecting HW to LLM

- Use LLM API's to get answers and return natural language on hardware.
- HW uses inputs from the microphone to take in speech and write text then sends a text prompt to the LLM.
- The LLM responds to the text prompt, then the MPU uses text to speech the read the response out loud on the connected speaker



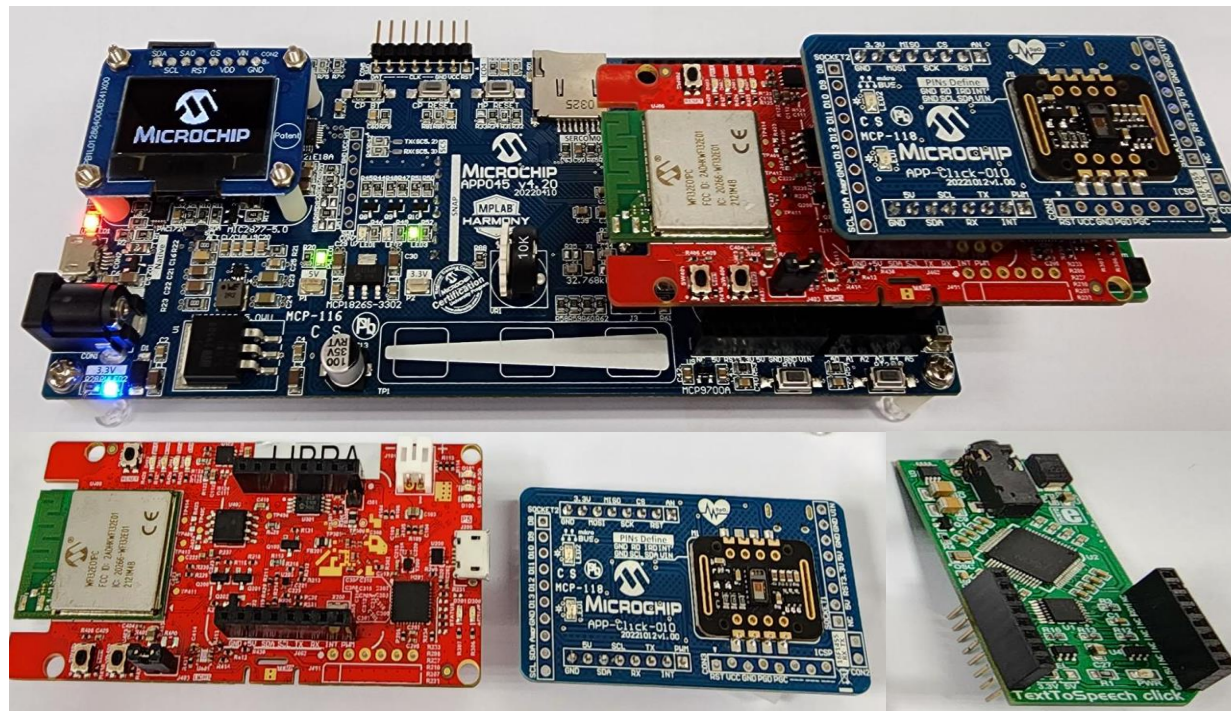
Architecture of ChatGPT in Microchip Healthcare Kits

User Query



I am xx years old,
my heart rate about
100 bpm and oxygen
saturation is 98%
SaO2.

Microchip Healthcare Kits



Prompt
Manager

Iterative Reasoning



Outputs



Summary

MPLAB® ML Suite

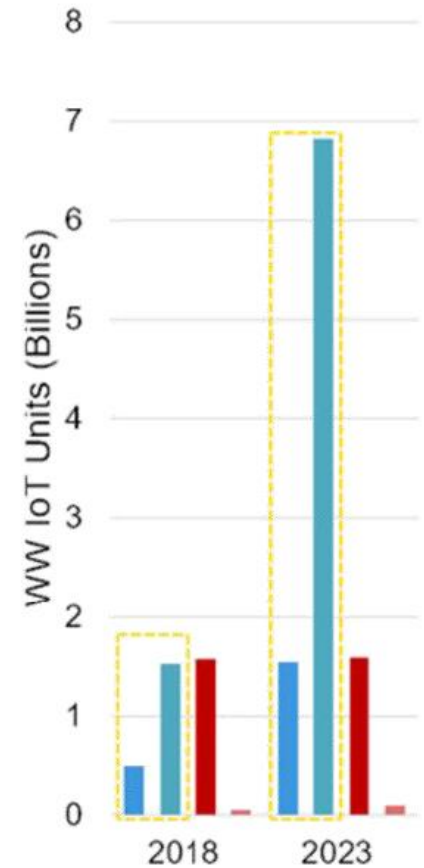
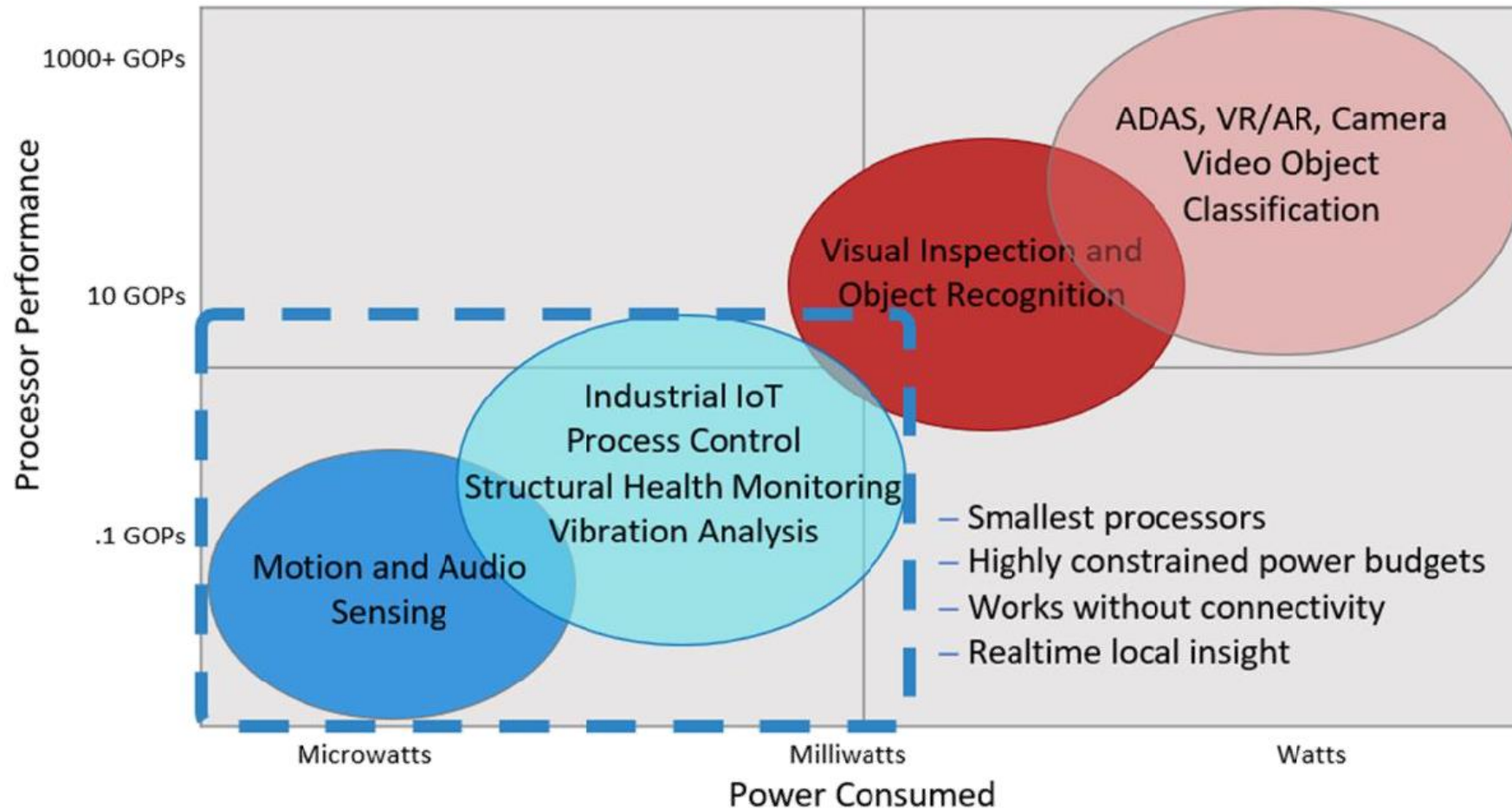
- **The ML Suite Plugin in MPLAB X is a plugin**
- **Behind this is a rich API based on SensiML**
 - Easily accessed from Python
 - Customize functions
 - Feature extraction
- **Can tuned by Jupyter Notebooks**
 - Web-based live notebooks
 - Code
 - Notes
 - Visualization



[Related document](#)



MPLAB® ML Suite Addresses Billions of Time-Series IoT Endpoints



Source: IHS 2017

Microchip Trademarks

<https://www.microchip.com/en-us/about/legal-information/microchip-trademarks>

Trademarks	Description/Appropriate Nouns	Notes
Adaptex "a" logo		Registered in other countries (not US)
Adaptex®	products, solutions, cables, software, drivers, adapters, controllers, series, family	
Adjacent Key Suppression™	technology	
AgileSwitch®	digital programmable gate driver, gate driver, drivers, family, device	US Only
AKS™	technology	
Analog-for-the-Digital Age™	AIPD family tagline	
Any Capacitor™	stable	
AnyIn™	feature, structure, capability	
AnyOut™	feature, capability	
Augmented Switching™	technology	
AVR®	devices, microcontrollers, MCUs, CPU, architecture, family	
AVR Logo		
AVR Freaks®	forum	
BesTime®	technology	
BitCloud®	SDK	
BlueSky™	firewall, subscription service, technology	
BodyCom™	technology, system, development kit	
ClockStudio™	application, software, program, utility	
ClockWorks®	Configurator, products, series, family, devices	US Only
CodeGuard™	security	
CryptoAuthentication™	development library, library, devices, ICs, family	
CryptoAutomotive™	technology, security ICs, development kits	
CryptoCompanion™	chip	
CryptoController™	solution, Trusted Platform Module (TPM)	
CryptoMemory®	cryptographic security ICs, ICs	

NOTE: *Always check with the Legal Department for the most current trademarks.

Trademarks	Description/Appropriate Nouns	Notes
PIC ³² Logo		
PICDEM™	demonstration board	
PICDEM.net™	Internet/Ethernet demonstration board	
PICKit™	starter kit, programmer, debugger, in-circuit debugger, development tool, tool, on-board	
picoPower®	technology	
PICSTART®	development programmer	
PICtail™	daughter board, board	
PolarFire®	FPGA, family, kits, devices, SoC, System-on-Chip	
Power MOS IV™		
Power MOS 7™		
Powermite 3®	package	US Only
PowerSmart™	digital control library designer, development suite	
Precision Edge®	family, product family	US Only
ProASIC®	FPGA, architecture, board, kit, family	US Only
ProASIC Plus®	FPGA, devices, family, architecture, switch	US Only
ProASIC Plus Logo		
Prochip Designer®	software suite	
PureSilicon™	oscillator, technology	
QMatrix™	sensor, devices	
QTouch®	sensor, devices, library, tools, development platform, project builder, Composer, Project Builder, board, solution, technology, Configurator	
Quiet-Wire®	enhanced EMI technology, family, technology	
Ripple Blocker™	technology, attenuators, product line, family of linear regulators	
REAL ICE™	In-circuit emulator	
RTAX™	FPGAs	
RTG4™	FPGAs	
SAM-BA®	software, application, monitor	

NOTE: *Always check with the Legal Department for the most current trademarks.

Trademarks	Description/Appropriate Nouns	Notes
CryptoRF®	transponder, devices	
dsPIC®	Digital Signal Controllers (DSCs)	
dsPIC Logo		
dsPICDEM™	demonstration board, development board	
dsPICDEM.net™	board	
Dynamic Averaged Matching™ (DAM™)	architecture	
ECAN™	technology, solution	
Espresso T1S™	device	
EtherGREEN™	platform, technology, family	
EtherSynch™	platform, technology, family	US Only
EyeOpen™	cable compensation	
Flashtec®	controller, products, family	US Only
flexPWR®	technology	
Frequency on Demand®		Registered in other countries (not US)
GestIC®	technology, sensing, controllers	
GridTime™	device, time server, GNSS time server	
HELDO®	family, regulators	
Hyper Speed Control®	family, architecture	US Only
HyperLight Load®	mode	US Only
ICSP™ or In-Circuit Serial Programming™	programming capability	
IdealBridge™	rectifier, dual MOSFET-bridge rectifier	
IGAT™	board, demonstration board	
IGLOO®	FPGAs, family, devices, series	
INICNet™	technology, sniffer	
Intelligent Paralleling™	technology	
IntelliMOS™	family, power stage family	US Only
Inter-Chip Connectivity™	technology	
JitterBlocker™	family, attenuators	
JukeBox®	technology, platform	

NOTE: *Always check with the Legal Department for the most current trademarks.

Trademarks	Description/Appropriate Nouns	Notes
SAM-ICE™	emulator	
SerGenuity®	sensors, products	
Serial Quad I/O™ or SQI™	family, flash device, product	
Silicon Storage Technology®		Japan only
simpleMAP™	protocol	
SimpliPHY™	family, products, devices	
SmartBuffer™	devices, ICs	
SmartFusion®	FPGA, evaluation kit, SoC, System-on-Chip,	US Only
SmartHLS™	IDE, compiler software, compiler, software	
SMART-I.S.™	technology	
SpyNIC®	device	
SST®		
SST Logo		
storClad™	encryption technology, technology	
SuperFlash®	technology, kit, macro, device, memory	
SuperSwitcher™	family, regulator, buck regulator	
SuperSwitcher II™	family	
Switchtec™	technology, family, switch(es), product line	China only
Symmcom®		
Symmetricon®		
SynchroPHY™	family, products, devices	
SyncServer®	server, instrument, clock, oscillator, device	
SyncWorld®	timing, Ecosystem Program	US only
Tachyon®	products, family, controllers, platform, technology	
The Embedded Control Solutions Company®	MarCom Use Only	
TimeCesium®	products, primary frequency reference	US only
TimeHub®	platform, system	US only
TimePictra®	software suite, platform, Synchronization Management System	US only

NOTE: *Always check with the Legal Department for the most current trademarks.

Trademarks	Description/Appropriate Nouns	Notes
JukeBox with design	Registered word mark JukeBox with a design 	Image is not registered; this is a graphic treatment with the word mark
KEELO®	security ICs, protocol, technology	
Kleer®	technology	
Kleer with design	Registered word mark Kleer with a design 	Image is not registered; this is a graphic treatment with the word mark
Knob-on-Display™/KoD™	technology, solution, HMI, user interface, touch controller, family, devices, Widget Configurator, Knob Designer, Position Wizard	
LANCheck®	online design review, online review, review	
Libero®	SoC Design Suite, tools, software, design files, IDE, designer	US registration
LinkMD®	cable diagnostics, technology, family, diagnostics engine	
MarginLink™	signal integrity testing	
maxCrypto™	controller-based encryption, encryption	
maxStylus™	solution	
maXTouch®	touchscreen controllers, technology, studio, family	
maxView™	storage manager, tools, drivers	
MediaLB®	bus, controller, device	
megaAVR®	devices, MCUs	
memBrain™	neuromorphic memory product, product, solution, technology	
Microchip Logo		
Microchip Name and Logo		
Microsemi Logo		
Microsemi®		

NOTE: *Always check with the Legal Department for the most current trademarks.

Trademarks	Description/Appropriate Nouns	Notes
TimeProvider®	series, server, clock, products, grandmaster, ePRTC, enhanced PRTC	US only
TimeSource®	Enhanced PRTC	
tinyAVR®	microcontroller, MCU	
Total Endurance™	software model	
Trusted Time™	technology	
TSHARC™	touch screen controller, controller	
Turing™	hardware security module, programmer	
UNIQ®	memory chips, bus, hardware port, communication protocol, firmware	
USBCheck™	design review	
VarSense™	technology	
VectorBox™	accelerator, software development kit, SDK, intellectual property, IP, solution, platform, tools, tool chain, tool flow	
Vectron®		
VeriPHY™	cable diagnostics algorithm, cable diagnostics software suite	
ViewSpan™	technology	
WiperLock™	technology	
XMEGA®	microcontroller, MCU, devices, Custom Logic (XCL), family, series	
XpressConnect™	retimer	
ZENA™	software, analyzer or wireless adapter	
ZL®		

NOTE: *Always check with the Legal Department for the most current trademarks.

Microchip Technology Inc. Servicemarks®

Trademarks	Description/Appropriate Nouns	Notes
SQTP™	Serial Quick Turn Programming capability	

NOTE: *Always check with the Legal Department for the most current servicemarks.

MICROCHIP RESERVES THE RIGHT TO CHANGE THESE GUIDELINES SOLELY AT ITS DISCRETION.

Trademarks	Description/Appropriate Nouns	Notes
MindI™	analog simulator, analog simulator tool	
MiWi™	wireless networking protocol, protocol, network, stack, software, applications, mesh, coordinator	
MOST®	technology, NetServices, evaluation kit, board, system, device, product, network, specification, platform, cooperation, design	
MOST Logo		
motorBench®	Development Suite	US Only
MPASM™	assembler	
MPF™	products, devices	
MPLAB®	Integrated Development Environment, In-circuit Debugger, In-Circuit Emulator, IDE, ICD, REAL ICE™ in-circuit emulator, starter kit, compiler, XC compiler, C compiler, Harmony, Integrated Programming Environment, IPE, Xpress, Code Configurator, development ecosystem, PICKit™, server, library, Snap, Connect Configurator, Network Creator, Analog Designer, Discover, MindI™, Analysis Tool Suite, Discover, Universal Device Programmer, Machine Learning Development Suite, PTG, Programmer-to-Go, Programmer-to-Go Mobil App, SIC Power Simulator	
MPLAB Certified Logo		
MPLIB™	object librarian or librarian	
MPLINK™	object linker or linker	
mSIC™	MOSFETs, diodes, products, gate drivers, modules, bare die, technology, solutions	
mTouch®	solution, sensing solution, evaluation kit, technology, development kit, library	US Only
MultiTRAK™	technology	
NetDetach™	technology	
Omniscient Code Generation™ (OCG)	compilation technology	
OptoLyzr®	tool, network analysis tool, Studio	
PIC®	microcontroller, MCU, device(s), assembler	

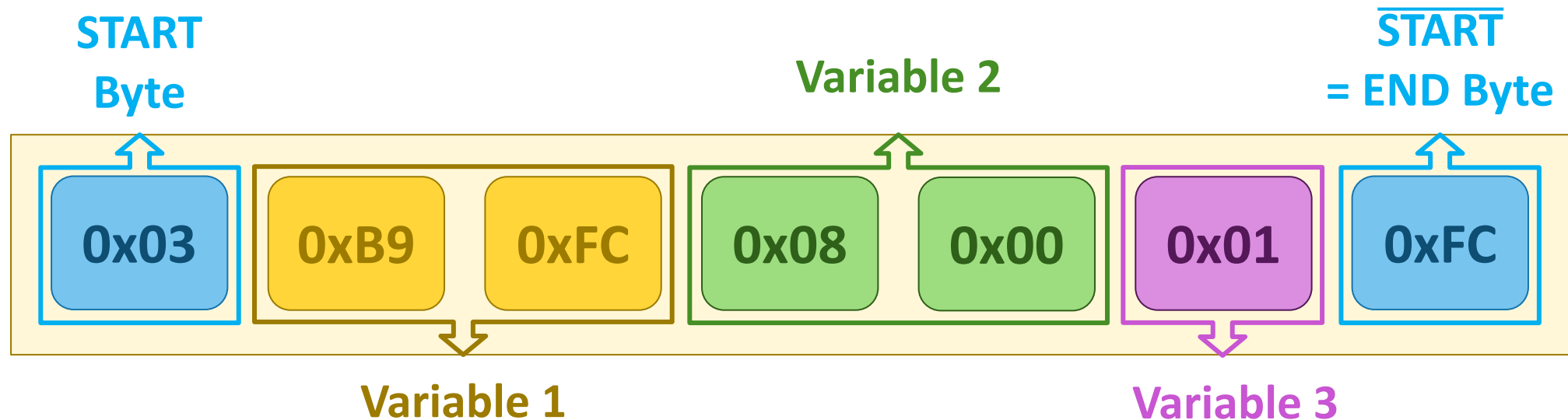
NOTE: *Always check with the Legal Department for the most current trademarks.



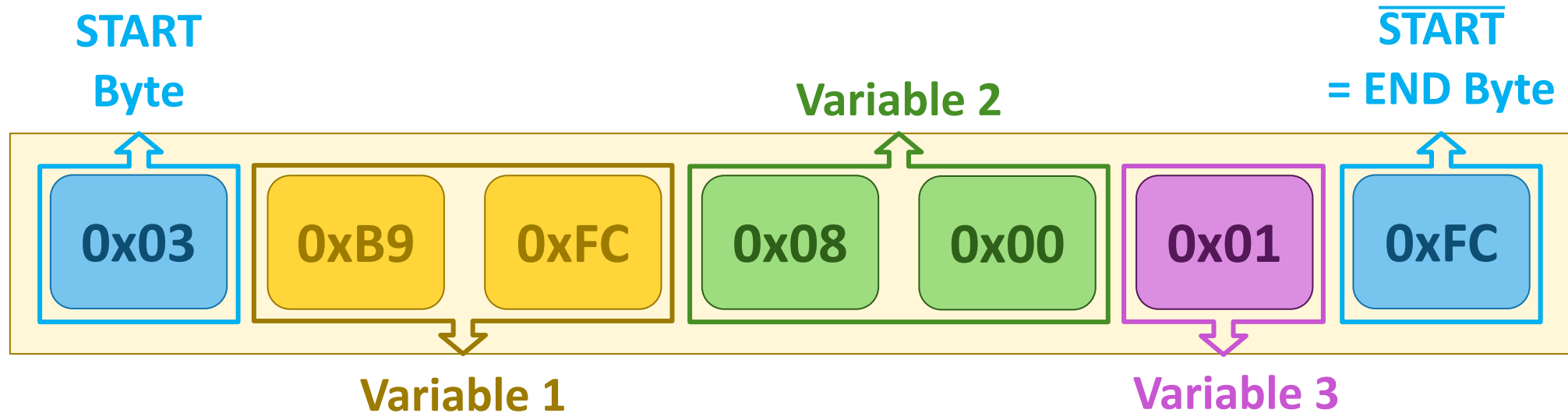
Appendix

Data Stream Format for DV

- The Data Stream Format is processed in the same order as the variables specified in the Variable Streamer.
 - Data must be given as little endian values.
 - The start byte can be arbitrary, but the end byte must be the inverse of the Start byte.



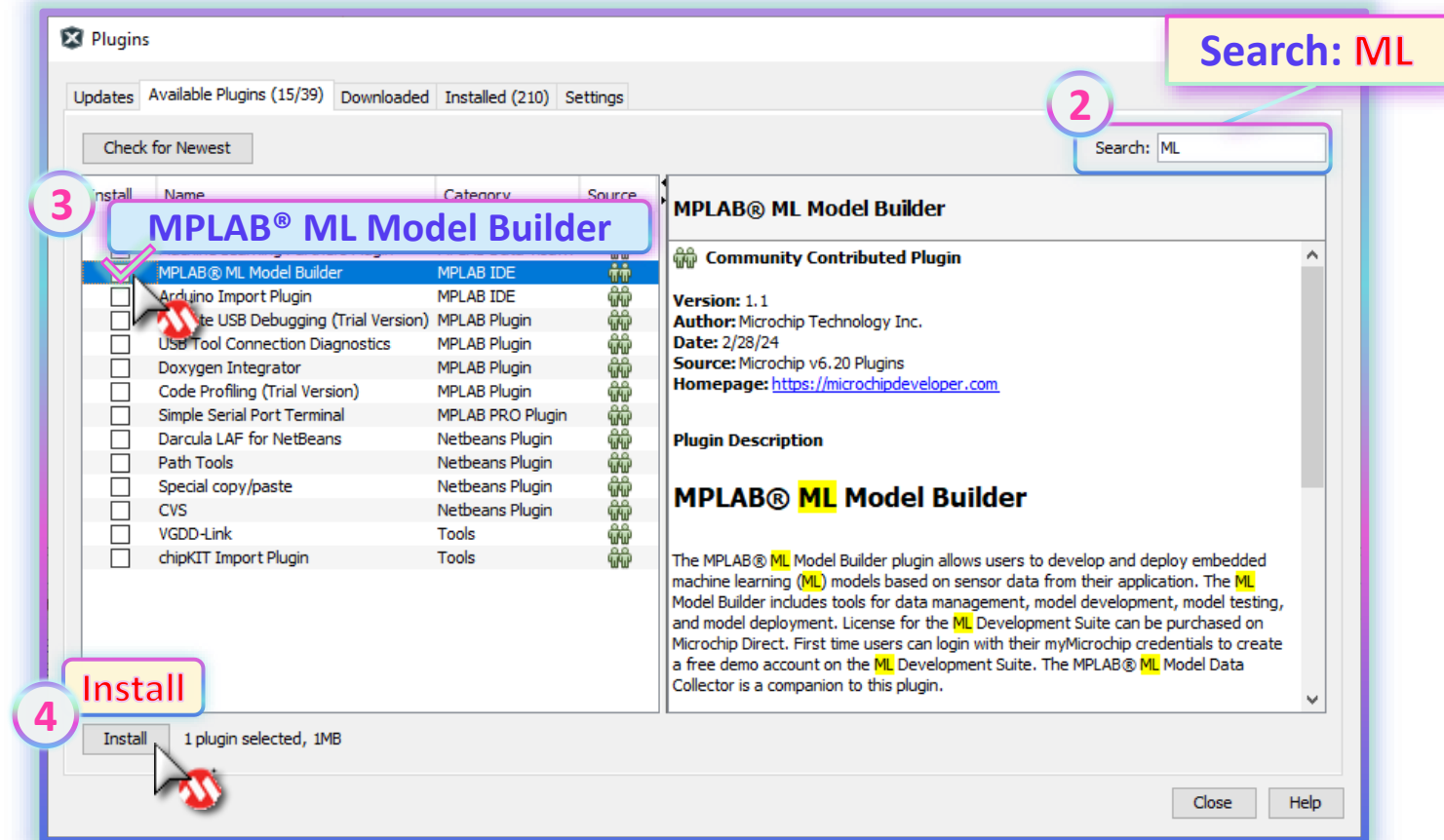
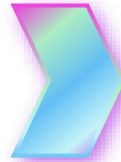
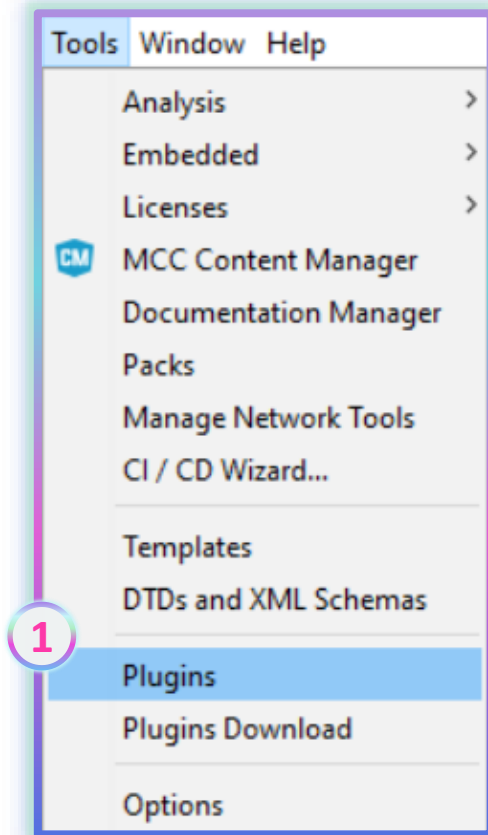
Data Stream Format for DV



```
printf("%c%c%c%c%c%c%c",  
    0x03,  
    variable[0] & 0xff, (variable[0]>>8) & 0xff,  
    variable[1] & 0xff, (variable[1]>>8) & 0xff,  
    LED_Get() & 0x01,  
    ~(0x03));
```

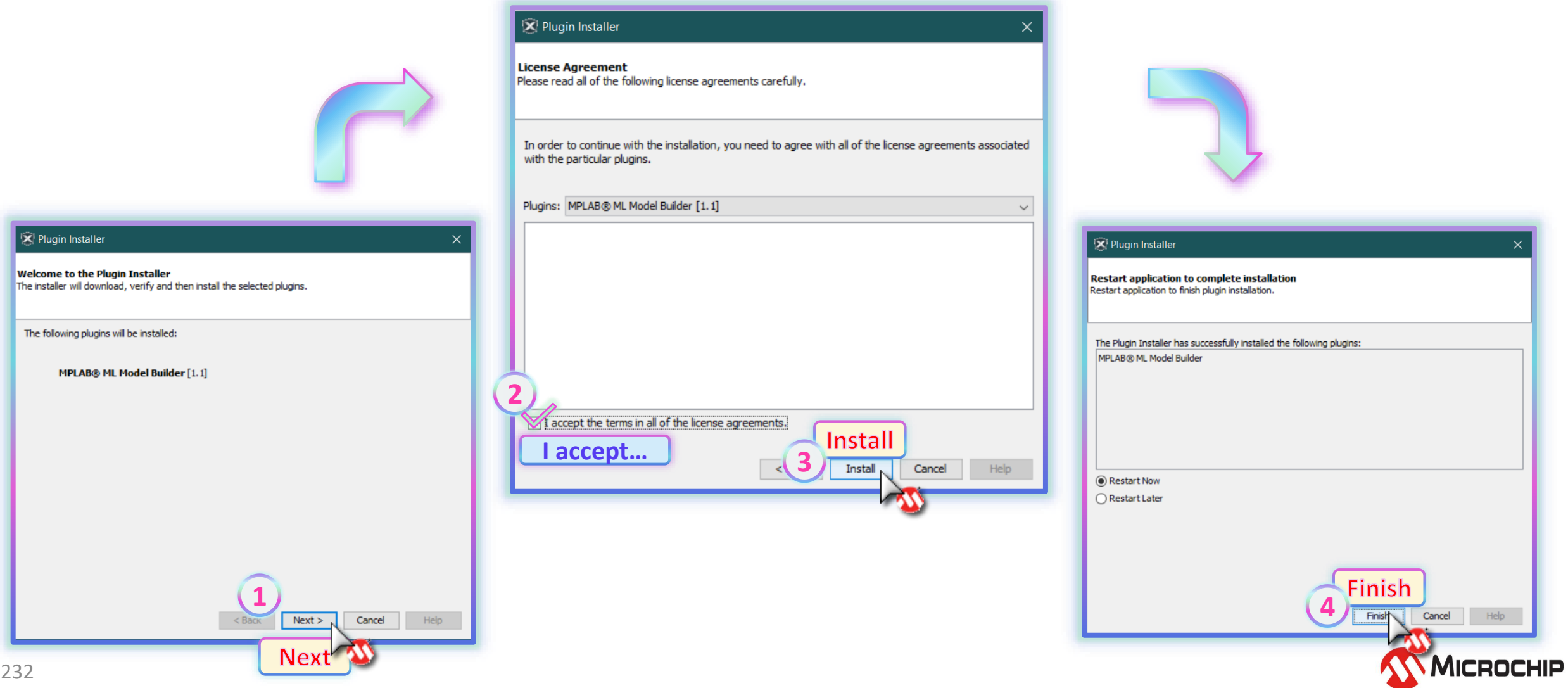
Install MPLAB® ML Model Builder

- MPLAB X IDE -> Tools -> Plugins -> Available Plugins -> MPLAB® ML Model Builder



Install MPLAB® ML Model Builder

- Next -> Install -> Restart the MPLAB X IDE.



Install MPLAB® ML Model Builder

- See  icon on the top after reopen MPLAB X IDE.



Install MPLAB® ML Data Collector

- MPLAB X IDE -> Tools -> Plugins -> Available Plugins -> MPLAB® ML Data Collector

The image shows a two-step process for installing the MPLAB ML Data Collector plugin in the MPLAB X IDE.

Step 1: The 'Tools' menu is open, and the 'Plugins' option is selected. A blue arrow points from this menu to the next screenshot.

Step 2: The 'Plugins' dialog box is shown. The 'Available Plugins (14/37)' tab is active. A search bar at the top right contains the text 'ML'. The 'MPLAB® ML Data Collector' plugin is highlighted in the list. The 'Install' button at the bottom left is clicked. The right pane shows the details for the selected plugin, including its version (1.2), author (Microchip Technology Inc.), date (6/14/24), source (Microchip v6.20 Plugins), and homepage (<https://microchipdeveloper.com>).

Numbered callouts indicate the sequence: 1 points to the 'Plugins' menu item; 2 points to the search bar; 3 points to the 'MPLAB® ML Data Collector' plugin in the list; 4 points to the 'Install' button.

Install MPLAB® ML Data Collector

- Next -> Install -> Restart the MPLAB X IDE.

