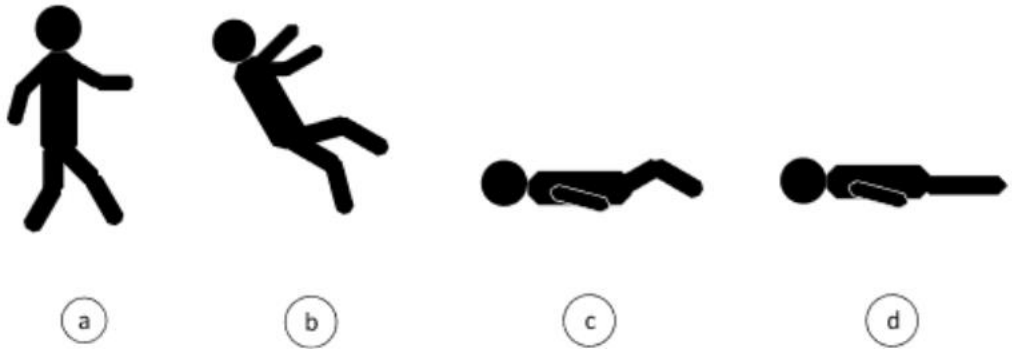
Fall Detection

ATSAMD21G18 and BMI160 6DOF IMU sensor

Objective:

Designing an intelligent wearable device utilizing the **ATSAMD21G18** Machine Learning Evaluation Kit in conjunction with the MPLAB X Machine Learning Development Suite for fall detection



The ML Development Suite streamlines the development of machine learning solutions specifically tailored for application and deployment on Microchip's microcontrollers and microprocessors. This manual offers a comprehensive walkthrough, demonstrating the steps to gather 6-axis IMU data from the ATSAMD21G18 Machine Learning Evaluation Kit, transmit the acquired data to the ML Model Builder, construct a personalized model capable of accurately classifying the input data, and ultimately deploy the model back onto the ATSAMD21G18(SAM-IoT WG Development Board)

Materials:

Hardware:

SAM-IoT WG Development Board

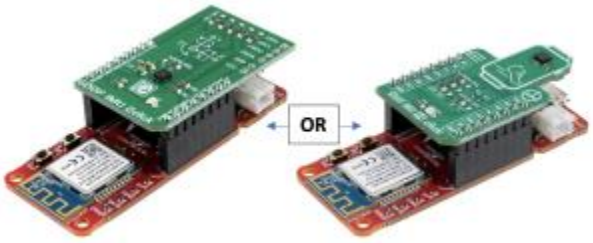
6DOF IMU 17 CLICK

Software Tools:

- MPLAB X IDE (Integrated Development Environment) (v6.05 or later)
- XC 32 Compiler
- MPLAB ML Development Suite
    - ML Data Collector
    - ML Model Builder

Connection Diagram

The IMU click board is connected to the mikroBUS™ socket. Check the pin labels when connecting the boards to ensure that they are connected properly.



Procedure

Before getting started, be sure to install all the necessary software tools. We will work within MPLAB X IDE and build the project with the MPLAB XC32 compiler. Both MPLAB DV and the Machine Learning Plugin are available within the plugin update center in MPLAB X IDE. You will also need to sign up for myMicrochip account to access MPLAB X Machine Learning Development Suite.

Data Collection

The initial phase of constructing a machine-learning model involves the acquisition of data. This dataset serves as the foundation for training the ML model, enabling it to discern various types of activities. In this particular instance, the model will be trained to identify two different activities.

Exercise Files:

- The firmware and dataset used in this tutorial and other MPLAB X project files can be found in the latest GitHub release.
- For those interested in the IMU data collection firmware and usage instructions for the Curiosity Nano, they are available for download from the SAMD21 Data Logger firmware.

Data Collection Overview

In this section, the recommended approach for gathering data samples to build the exercise classifier model is discussed.

To begin the data collection process, it is essential to identify a suitable sensor configuration for your specific application. This will involve determining the optimal placement of the sensor, selecting an appropriate installation method, and configuring the signal processing parameters, such as the sample rate and sensitivity.

Sensor Sampling Configuration

The sensor sampling parameters are summarized below:

- Sensor: 3-axis Accelerometer + 3-axis Gyrometer
- Sample Rate / Frequency Range: 100 Hz (~40 Hz 3 dB cutoff)
- Accelerometer Full Scale Range: +/-2G (most sensitive setting)
- Gyrometer Full Scale Range: +/-125 DPS (most sensitive setting)

Keep in mind that the configuration above was derived from analysis for a specific activity, so it may not be optimal for different setups.

Data Collection Protocol

To proceed with the data collection process, the next essential step involves creating a protocol that outlines the necessary procedures for collecting data. This entails determining the number of samples to be collected, the metadata parameters to be captured and other relevant parameters that will govern the data collection procedure. The protocol will serve as a guide for executing the data collection process and ensure that the data collected is consistent and accurate.

Data Collection: Metadata

Initially, it is imperative to address metadata, as it significantly contributes to contextualizing our data. In this application, our metadata comprises solely the dataset, which is partitioned into training and testing subsets.

Data Collection: Sampling Method

At this stage, it is necessary to decide the data sampling process for our application. This involves determining the number of samples to be collected and defining the necessary steps for measuring the data. In the context of this example application, the methodology can be summarized in the following steps:

- Record and label segments for each of the following activities: Fall activity and ADL

How to Configure, Compile and Flash

This document explains the steps involved in configuring the data logger firmware build, compiling it and flashing it to the SAMD21 IoT device. Follow the instructions below:

1. Connect the Curiosity board to your PC using a USB cable.
2. Install the MPLAB X IDE and XC32 compiler on your PC. These are essential tools for loading the data logger project and programming the SAMD21-IoT board.

3. Open the firmware project folder in MPLAB X.

4. Choose the appropriate MPLAB X Project Configuration for your sensor based on the table provided below.

Select the data streaming format you want by setting the DATA_STREAMER_FORMAT macro in firmware/src/app_config.h to the appropriate value as summarized in the table below.

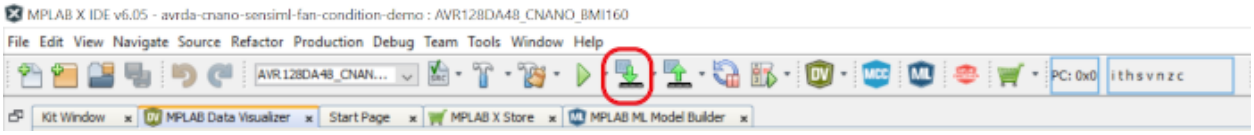
Figure 1-18. Selecting the type of data streamer format.

Streaming Format	app_config.h Configuration Value
ASCII text Format	#define DATA_STREAMER_FORMAT DATA_STREAMER_FORMAT_ASCII
MPLAB Data Visualizer	#define DATA_STREAMER_FORMAT DATA_STREAMER_FORMAT_MDV
Disable all Data Streaming	#define DATA_STREAMER_FORMAT DATA_STREAMER_FORMAT_NONE

1. Modify high level sensor parameters like sample rate (SNSR_SAMPLE_RATE), accelerometer range (SNSR_ACCEL_RANGE) and others by changing the macro values defined in firmware/src/app_config.h. See the inline comments for further details.

When you are satisfied with your configuration, click the Make and Program Device icon button in the toolbar (see image below for reference).

Figure 1-19. Programming the device



Firmware Operation

The data streamer firmware will output sensor data over the UART port with the following UART settings:

- Baud rate 115200
- Data bits 8
- Stop bits 1
- Parity None

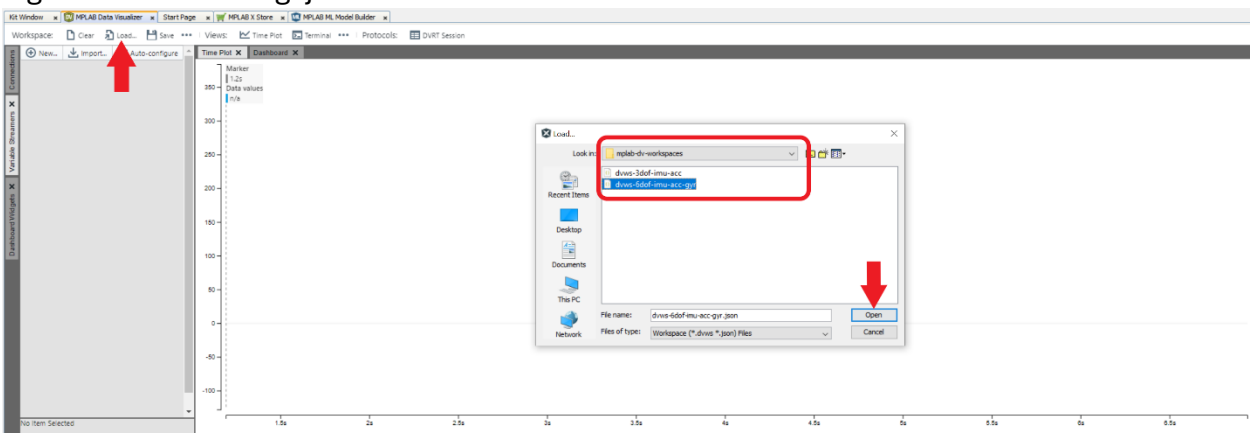
Usage with the MPLAB Data Visualizer and Machine Learning Plugins

This project can be used to generate firmware for streaming data to the MPLAB Data Visualizer plugin by setting the DATA_STREAMER_FORMAT macro to DATA_STREAMER_FORMAT_MDV as described above. When the firmware is flashed, follow the steps below to set up Data Visualizer.

1. Connect the SAMD21board to your computer, open up MPLAB X, then open the Data Visualizer plugin.

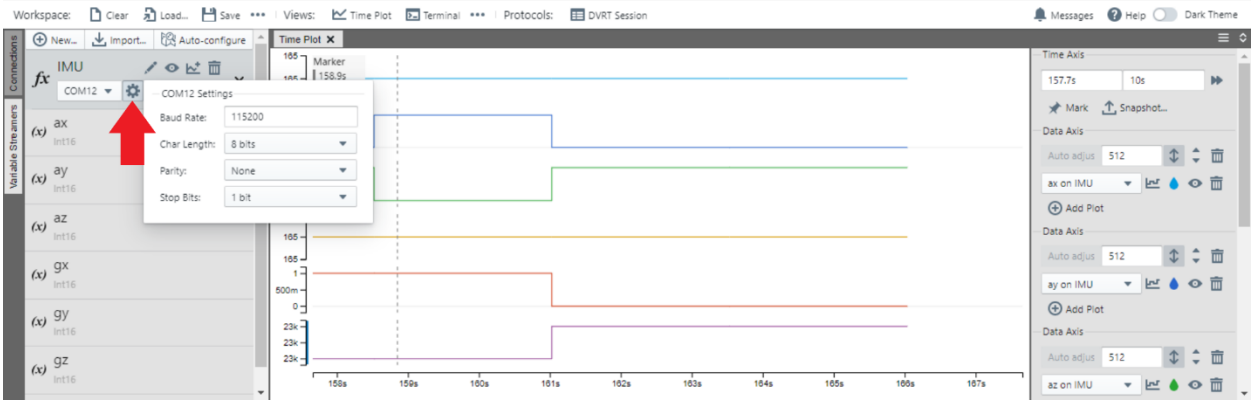
2. Click the **Load Workspace** button as highlighted in the image below. Select one of the workspace files included in this repository - located under the mplab-dv-workspaces folder - whose name most closely describes your sensor configuration; you can always modify the configuration when it is loaded if needed.

Figure 1-20. Loading .json file to variable streamers.



1. Next, select the Serial/CDC Connection that corresponds to the SAMD21 board, adjust the baud rate to 115200, then click **Apply**.

Figure 1-21. Configuring the COM Port settings.



1. Use the play button on the Serial/CDC Connection to connect to the serial port. When the connection is made, the SAMD21 is available for use with the variable streamer.

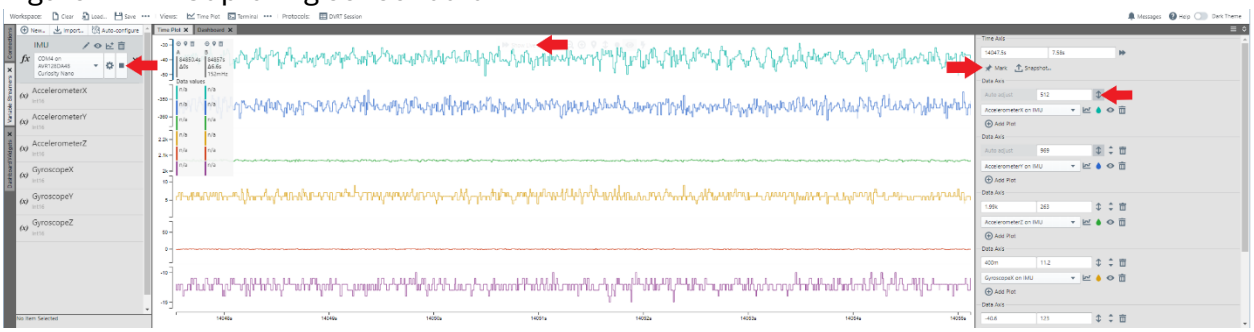
2. Switch to the **Variable Streamers** tab, then select the Serial/CDC Connection from the dropdown menu as the input data source for the IMU variable streamer (see image below for reference); this will enable parsing of the SAMD21 data stream. You may delete or add variables here if your sensor configuration differs from the pre-loaded ones.

3. The IMU data is now available in the time plot. Double click anywhere within the time plot to start/stop scrolling of the time axis.

4. Click the up down arrow button to auto set the time scale.

5. Click the mark icon button, then the snapshot icon button to snip the part of the time plot as data to save.

Figure 1-22. Capturing sensor data.



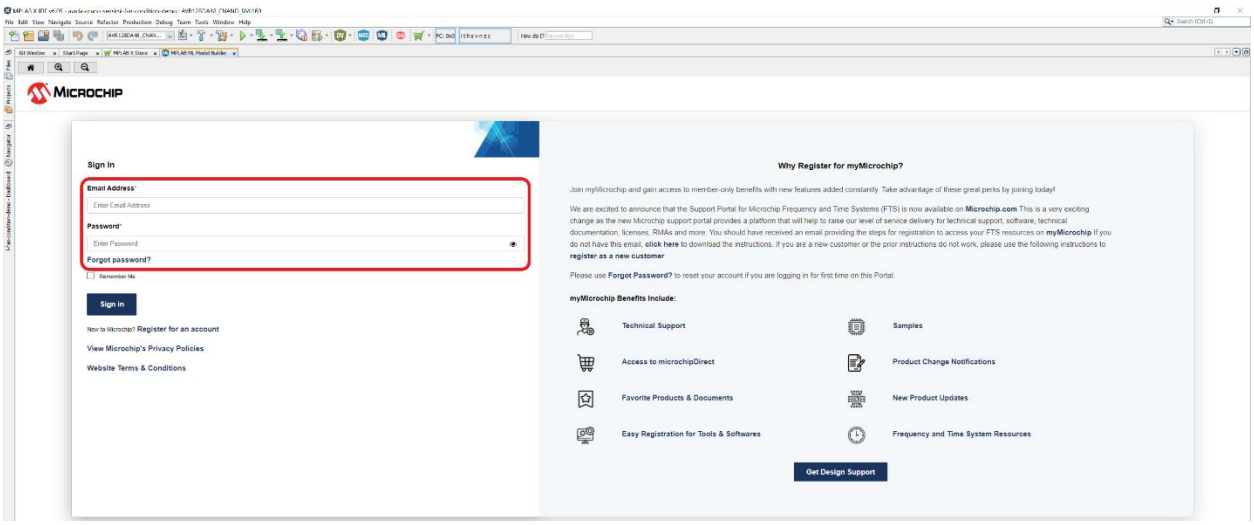
Visit the *Machine Learning Plugin* page to learn more about using the Data Visualizer plugin to export your data for machine learning applications.

Model Development

Once the initial dataset for your application has been gathered, the next step is to proceed to the ML Model Builder to create your classifier model. To generate your classifier model, proceed to MPLAB ML Development Suite.

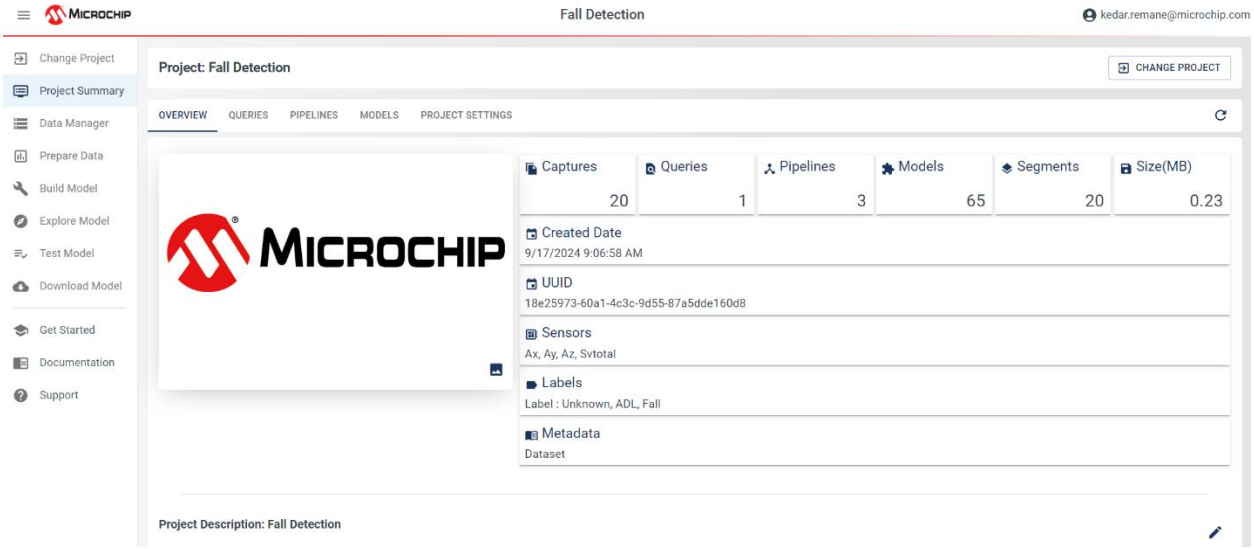
The process for accessing the Microchip ML Development Suite requires the following steps:

- Launch the MPLAB X IDE software.
- Next, launch the ML Model Builder tool.
- When the ML Model Builder page loads, you will be prompted to authenticate your account by entering your myMicrochip login credentials.



Upon successful login, users can create a new project by selecting the "CREATE PROJECT" tab, which prompts a pop-up window to specify a project name. To access a created project, simply click the "Open Project" icon button located before the project name.

Figure 1-29. Creating a new ML Project.

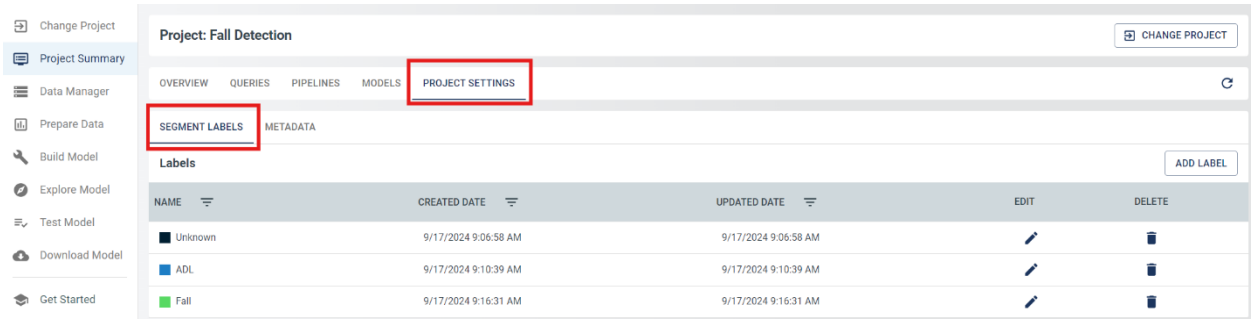


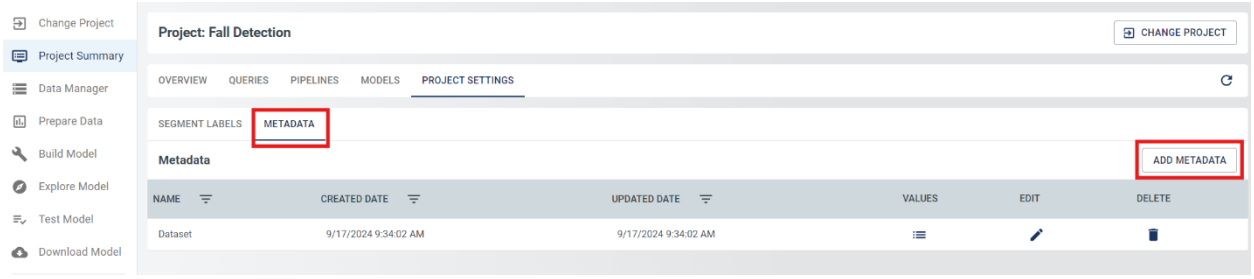
To create new labels and metadata fields for a project, follow these steps:

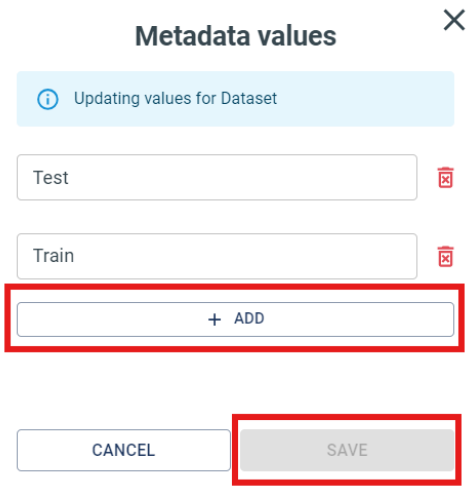
1. Navigate to the Project Summary page.
2. Access Project Settings.
3. Locate the "Segment Labels" and "Metadata" fields.

4. Select the "Add Label" and "Add Metadata" tabs positioned in the top right corner of the page.
5. Input the necessary details for the new label and metadata fields.
6. Save the information to generate the new label and metadata field.
7. Repeat steps 4-6 for each additional label or metadata field desired.

Figure 1-30. Configuring Project settings.

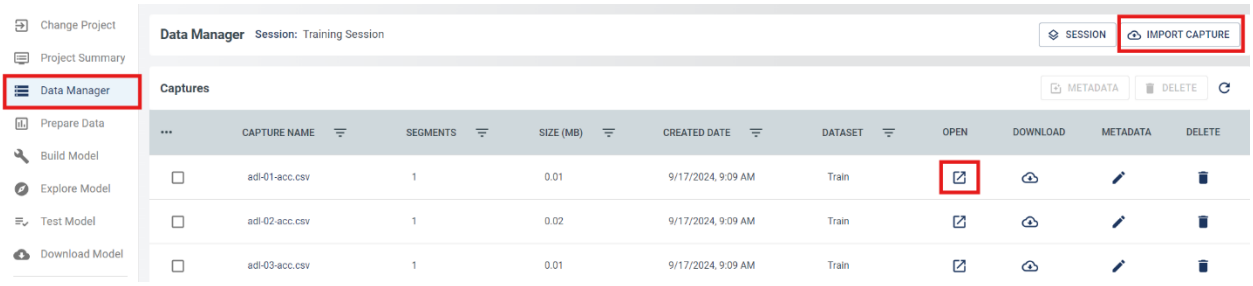






Next, to import necessary data files for the project, follow these steps:

1. Navigate to the **Data Manager** tab.
2. Locate the **IMPORT CAPTURE FILE** button, then click it.
3. Ensure that the data files to be imported are in either CSV or Wav format.
4. Select all the files required for the project.
5. Begin the import process.



Upon completion of the import process, the imported data files will be ready for utilization within the project. It is crucial to confirm that all data files are correctly formatted and labeled to guarantee precise analysis and interpretation of project outcomes.

To access the imported data files, click the icons corresponding to the desired file under the OPEN column. Once the desired file is open, navigate to the **CREATE (+)** tab to begin segmenting the data. To create segments, drag the segments axis to the desired locations within the data.

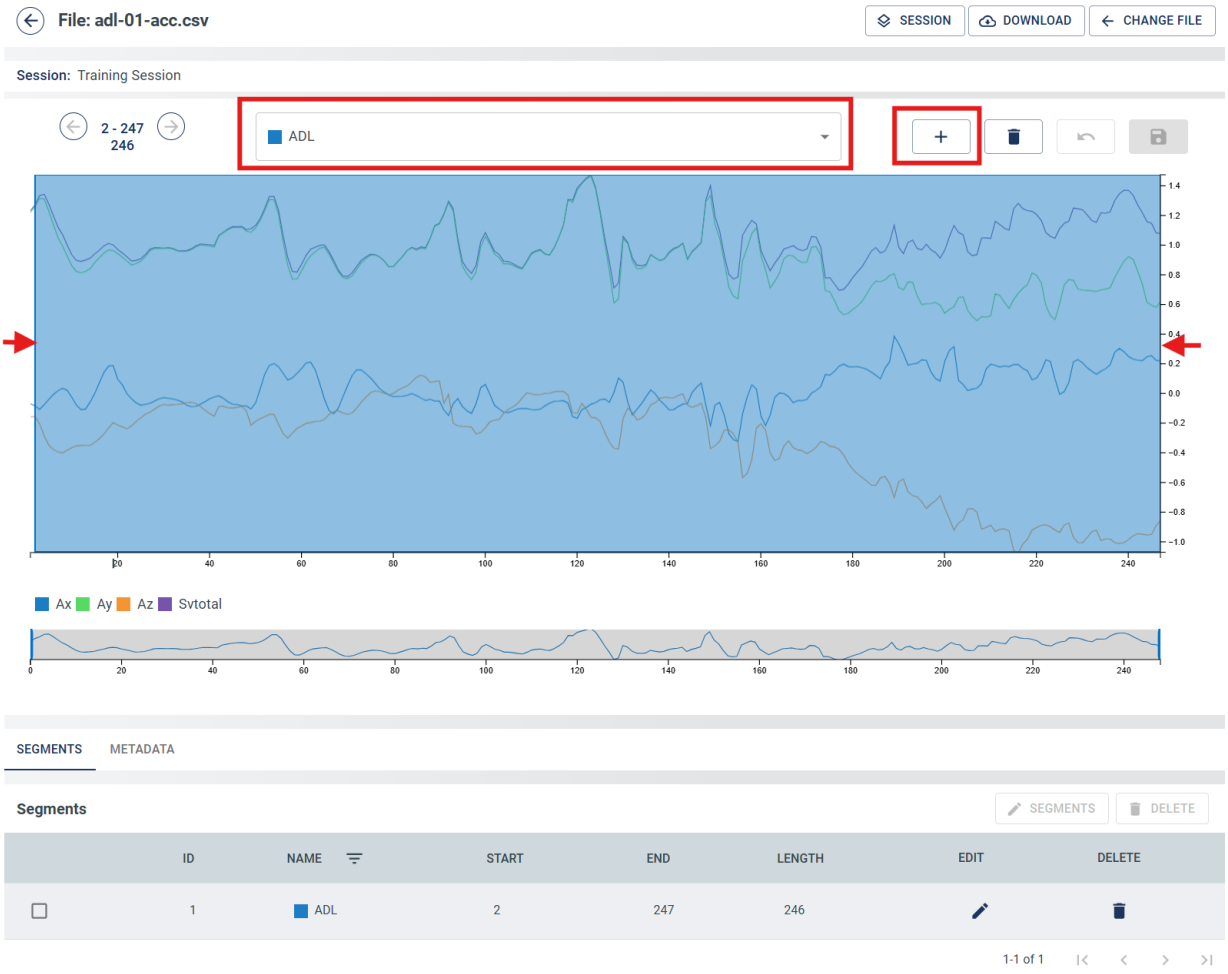


Figure 1-32. Creating and labelling segments inside data file.

It is, then, necessary to correctly categorize these segments using appropriate labels and metadata fields for ease of organization and future analysis.

To proceed, navigate to the Prepare Data view. From there, locate and click the **CREATE QUERY** button to initiate the data preparation process.

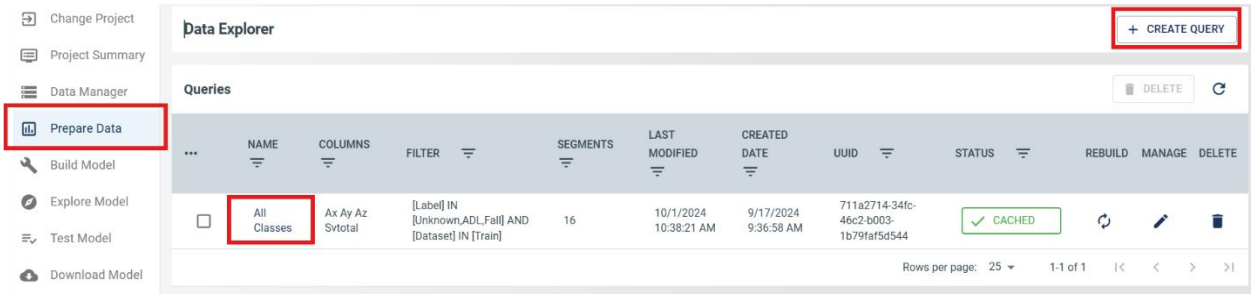
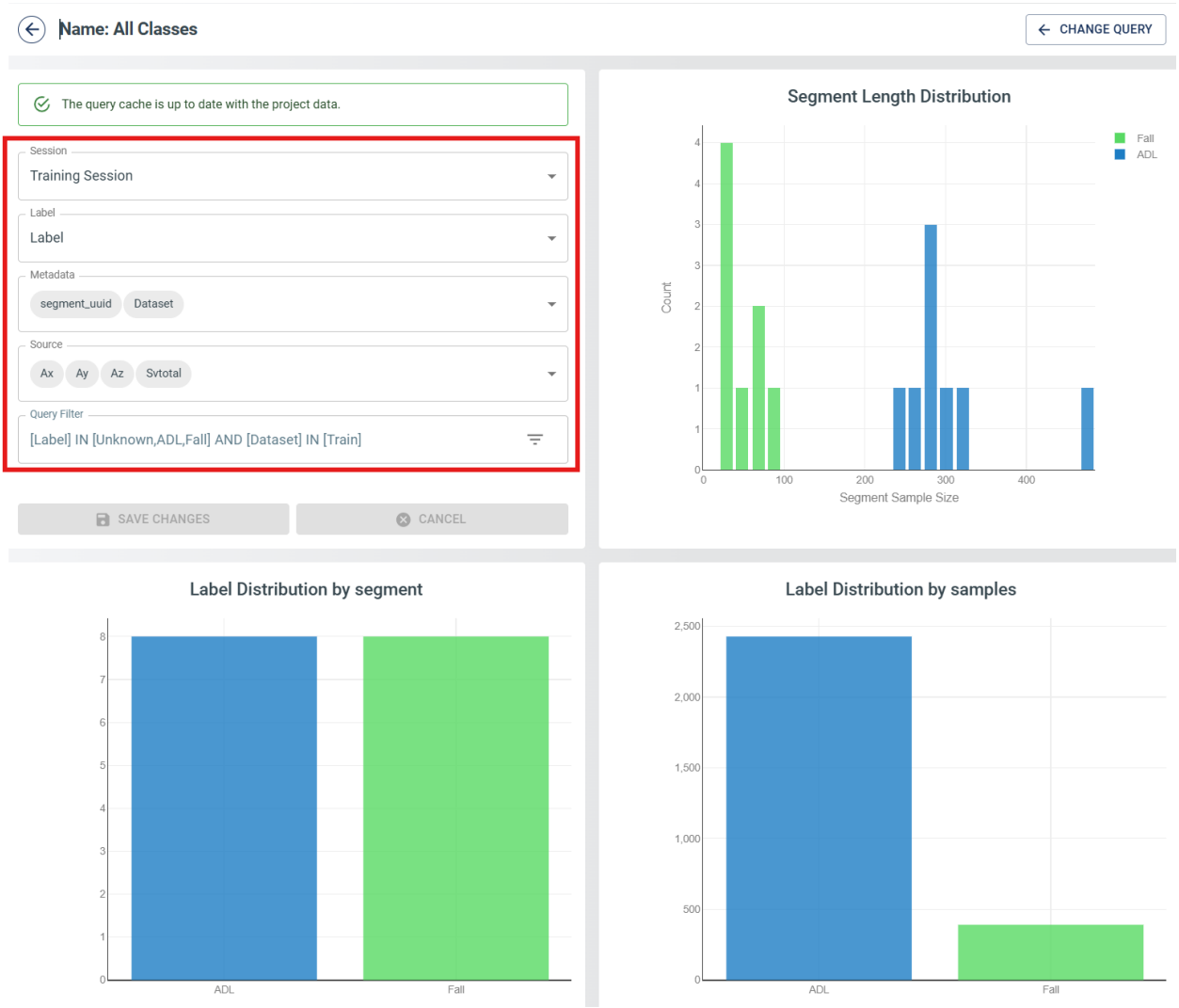


Figure 1-34. Creating new project query.

1. Fill out the block as displayed below.
2. Press the **CACHE** button to initiate the caching process.
3. Wait for the cache to build out completely.
4. Click the **SAVE** button when the cache is fully built.

Figure 1-35. Filling in details for the query.



Following these steps will result in the block saving cached information. Once the caching process is complete, details such as Segment Length Distribution, Label Distribution by segment, and Label Distribution by samples will be displayed.

To proceed with building the pipeline, navigate to the Build Model module and, then locate the Create Pipeline tab

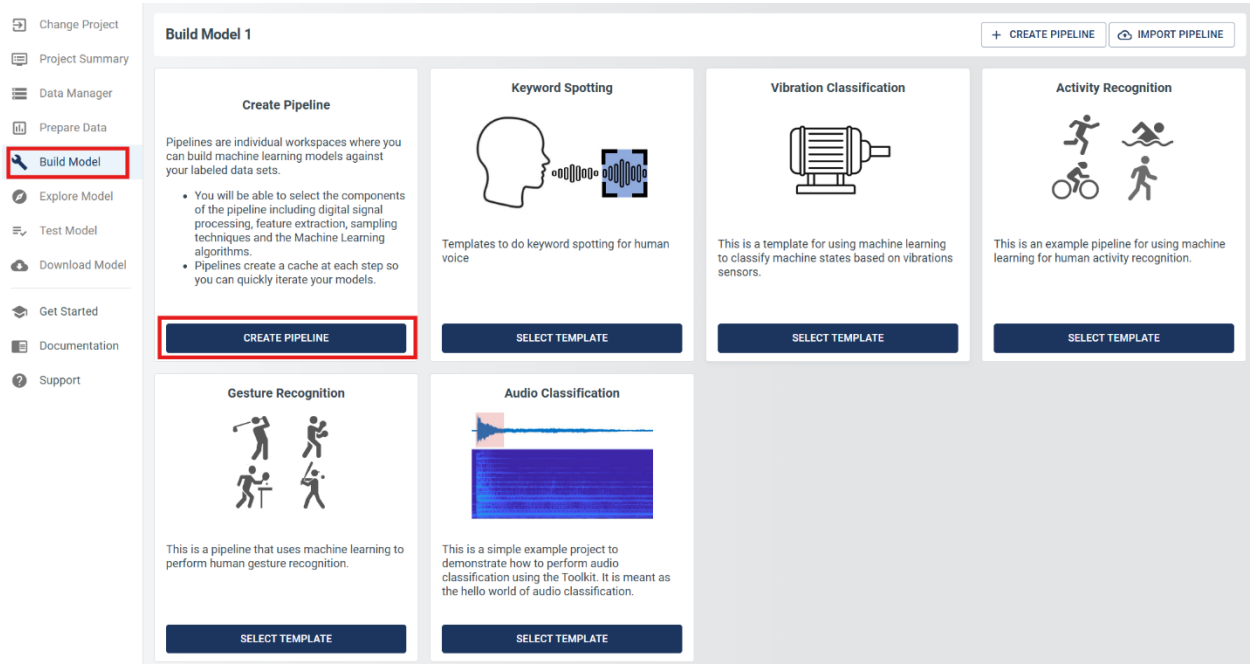
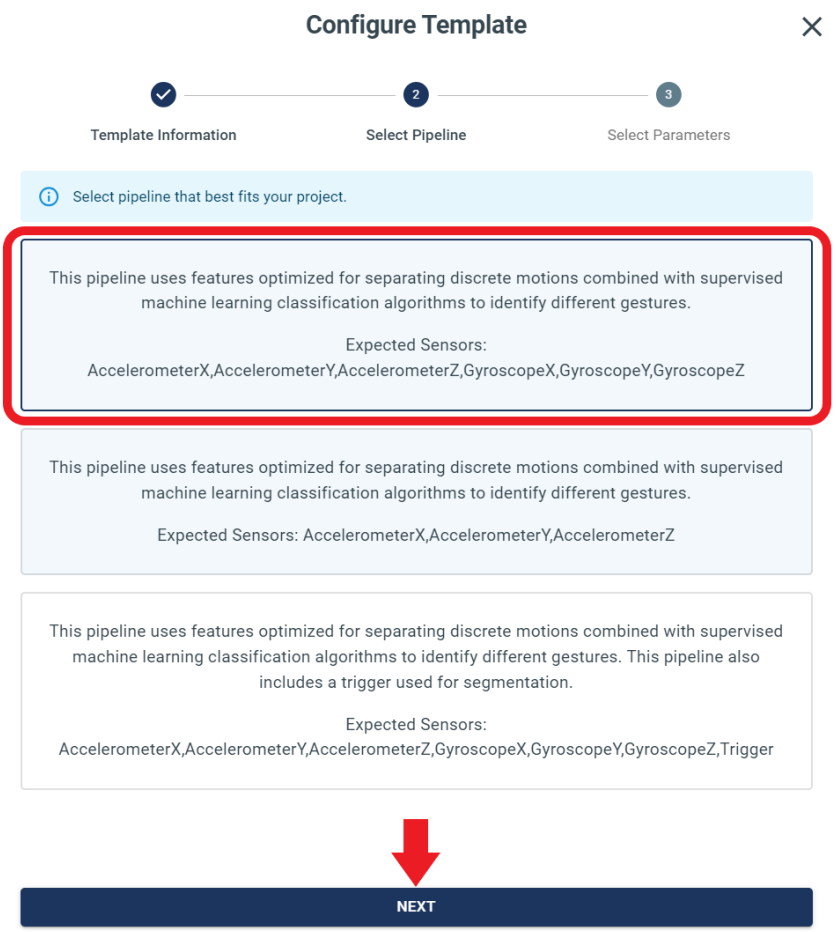


Figure 1-36. Creating a new pipeline.

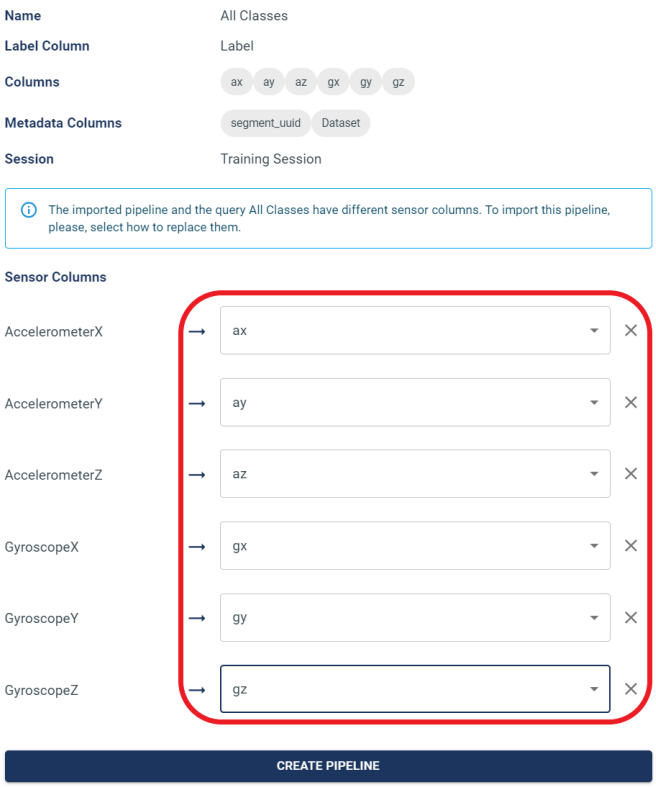
After clicking **CREATE NEW PIPELINE**, information regarding Pipeline displays. Click **NEXT** to proceed

The second step in configuring the template involves selecting the appropriate pipeline based on the sensors intended for inclusion. In this instance, six signals have been gathered, comprising three from each accelerometer and gyroscope, warranting the selection of the first template.



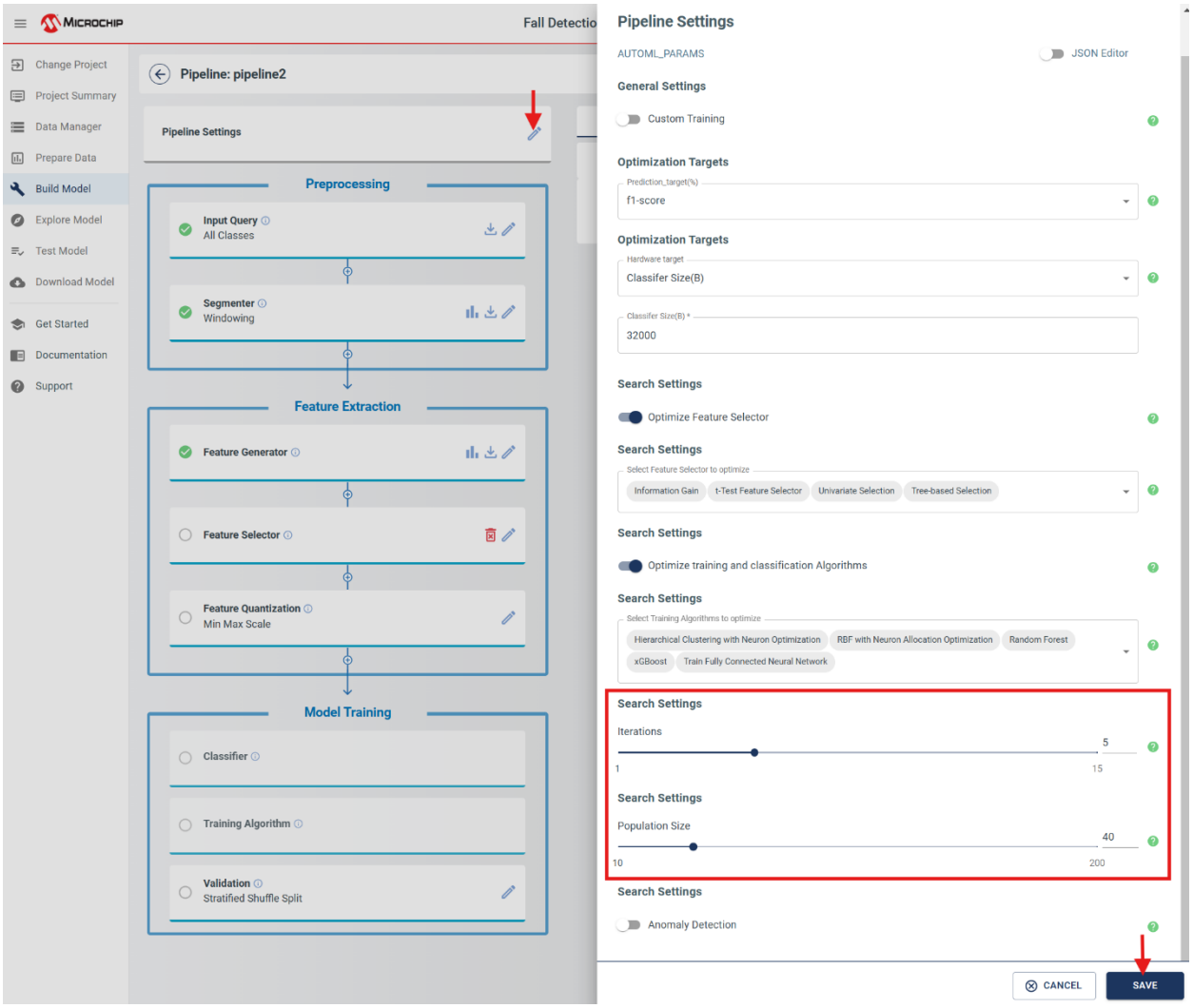
It is essential to appropriately link sensor data signals with their respective sensor columns in the template, replicating the process demonstrated below. Upon selecting the query, ensure to click the "CREATE PIPELINE" button to preserve your selection.

Figure 1-37. Assigning sensor columns

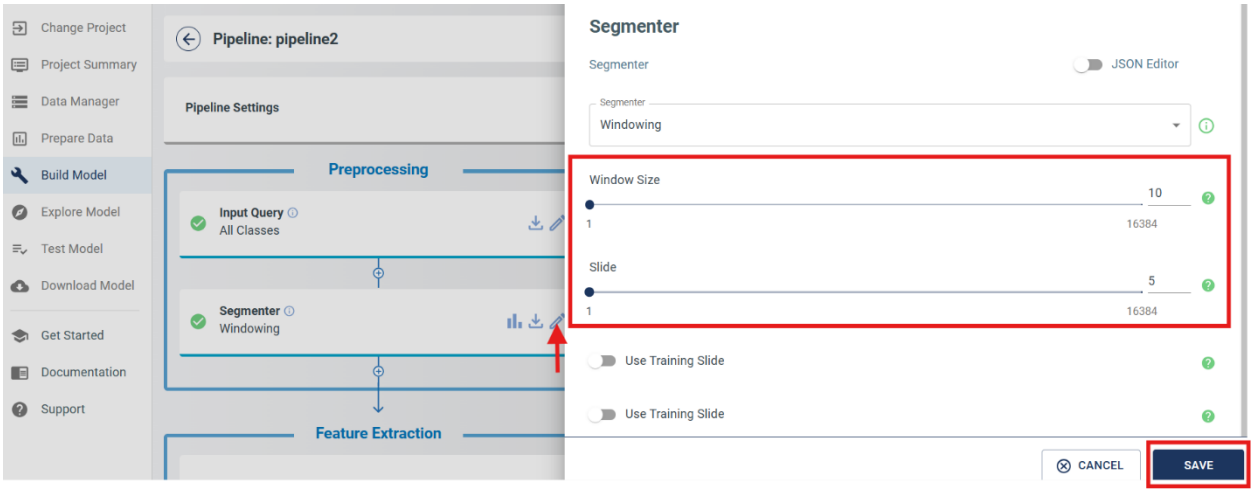


The pipeline will be created showing parameters that will be used while building the Model. Click **NEXT**.

To begin, access the pipeline settings by selecting the edit icon adjacent to it. By default, the system employs AutoML mode. However, users have the option to utilize custom training mode if necessary. For this demonstration, we will proceed with AutoML. Adjustments will be made solely to the search settings, specifically the number of iterations and population size, as indicated below.



Access the Segmenter module, which performs segmentation of the data. Set the parameters as depicted in the figure below to ensure the appropriate segmentation of the data. After setting the parameters, click the **SAVE** button to save the changes made.



Then, open the Feature Generator module. Select the features used to extract the information from the input sensor data. In this form, you can specify the types of feature generators you would like to use, the inputs into the feature generators and any parameters the feature generators take. Select the features that are shown below and click **NEXT** after selecting the required features.

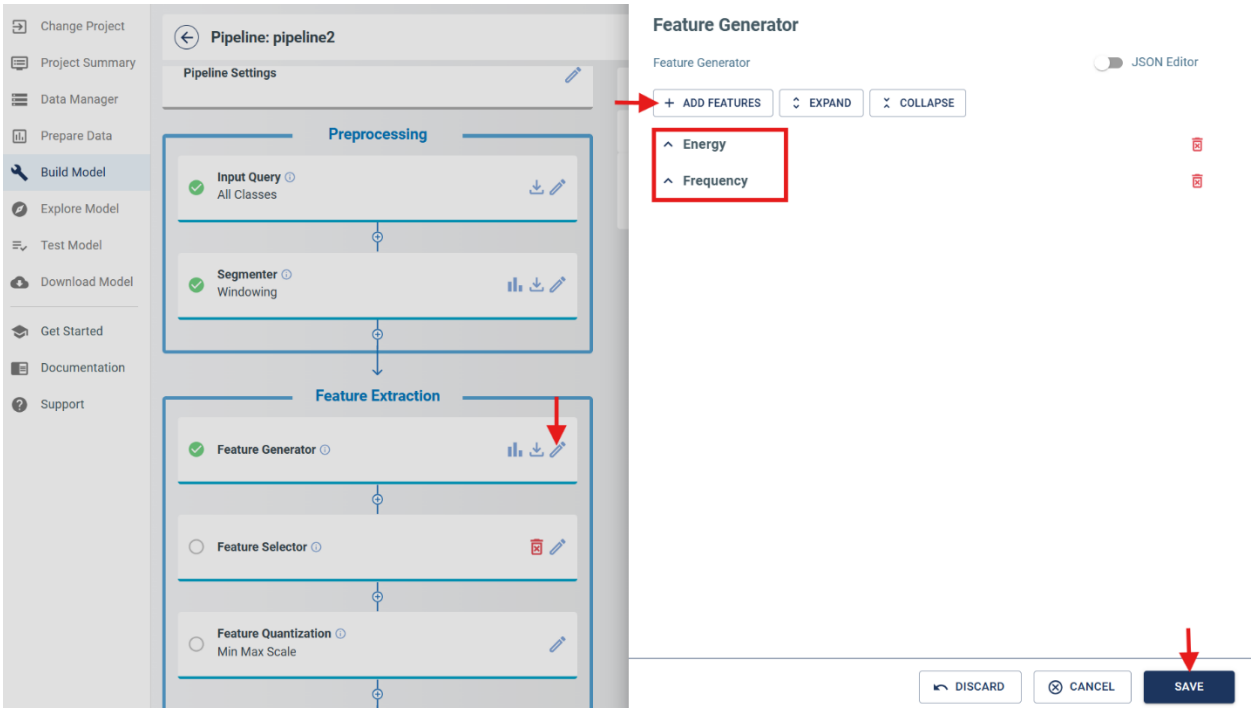


Figure 1-40. Selecting the Features for model building

Upon configuring all settings, activate the optimization process by selecting the "Run Pipeline" button located at the top.

Upon configuring all settings, activate the optimization process by selecting the "Run Pipeline" button located at the top.

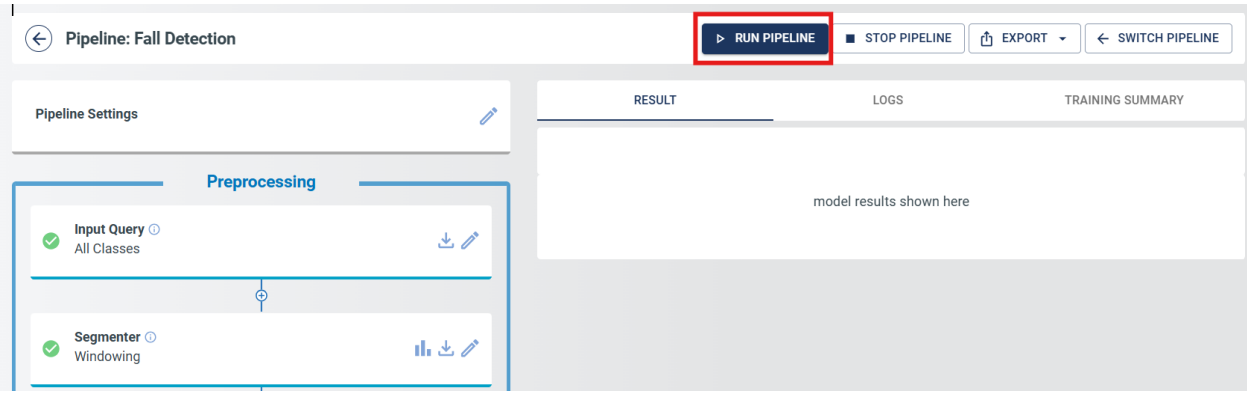
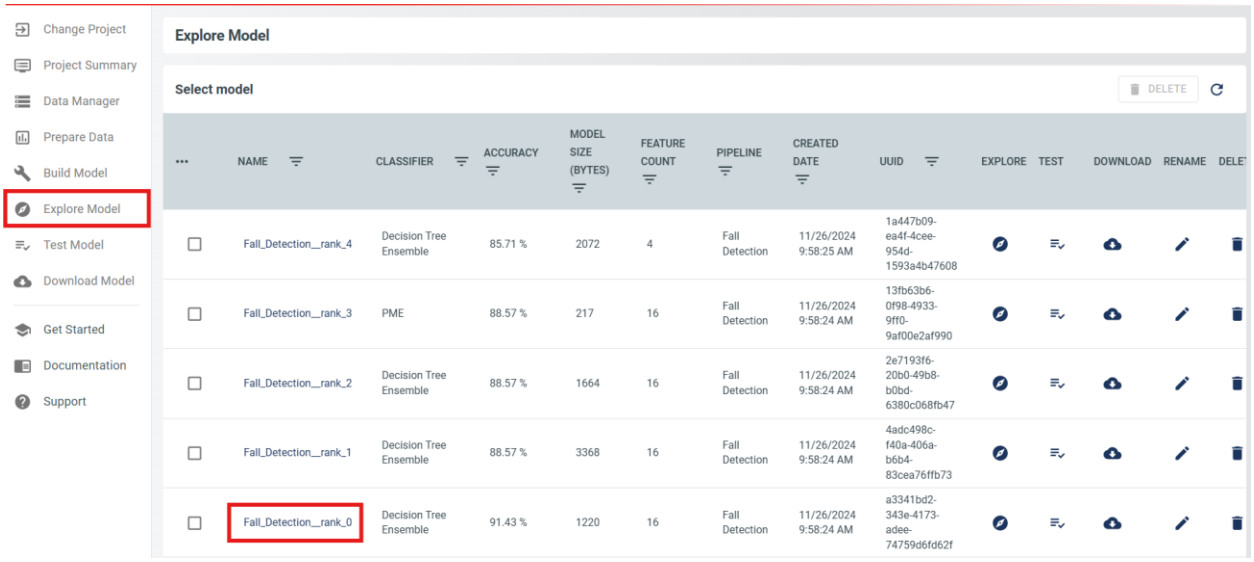


Figure 1-46. Starting the AutoML optimization process.

The AutoML techniques will automatically select the best features and machine learning algorithm for the gesture classification task based on the input data you provided. Note that this optimization process typically takes several minutes to complete.

After completing the process of Model Optimization, proceed to the Explore Model view. Within this view, the top five models are ranked and displayed based on their ability to produce optimal outcomes. To access additional information regarding each individual model, click the open model icon that is situated on the left-hand side of its respective name.

Figure 1-47. Exploring the model.



Users can access different output results via the options available at the top. These results include confusion matrices, graphs, and flowcharts, which can be explored for detailed analysis through modules such as confusion matrix, feature embedding, feature visualization, and different summaries.

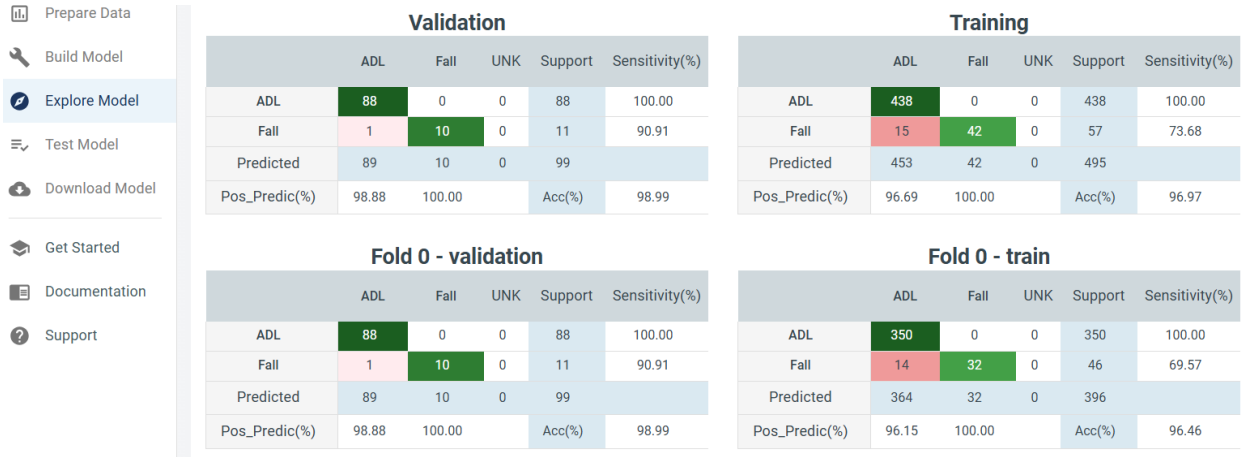


Figure 1-48. Validation output.

After confirming satisfactory model results, users can navigate to the Test Model view. To filter data and isolate test samples, follow these steps:

1. Choose the desired Session.
2. Click the upside-down triangle icon in the Fold column.
3. Select "Test" from the dropdown menu to filter and display only test samples.

To compute accuracy and generate a confusion matrix for the test samples, follow these steps:

1. Click the ellipsis (...) icon in the left-most column of the table.

2. Select "Select All" from the dropdown menu to include all test samples.

3. Click the "Recognize" button to compute accuracy metrics.

4. Once accuracy is computed, click "Summarize" to generate the confusion matrix.

5. Note that this process may take a few minutes. Upon completion, a table summarizing classification results will be presented.

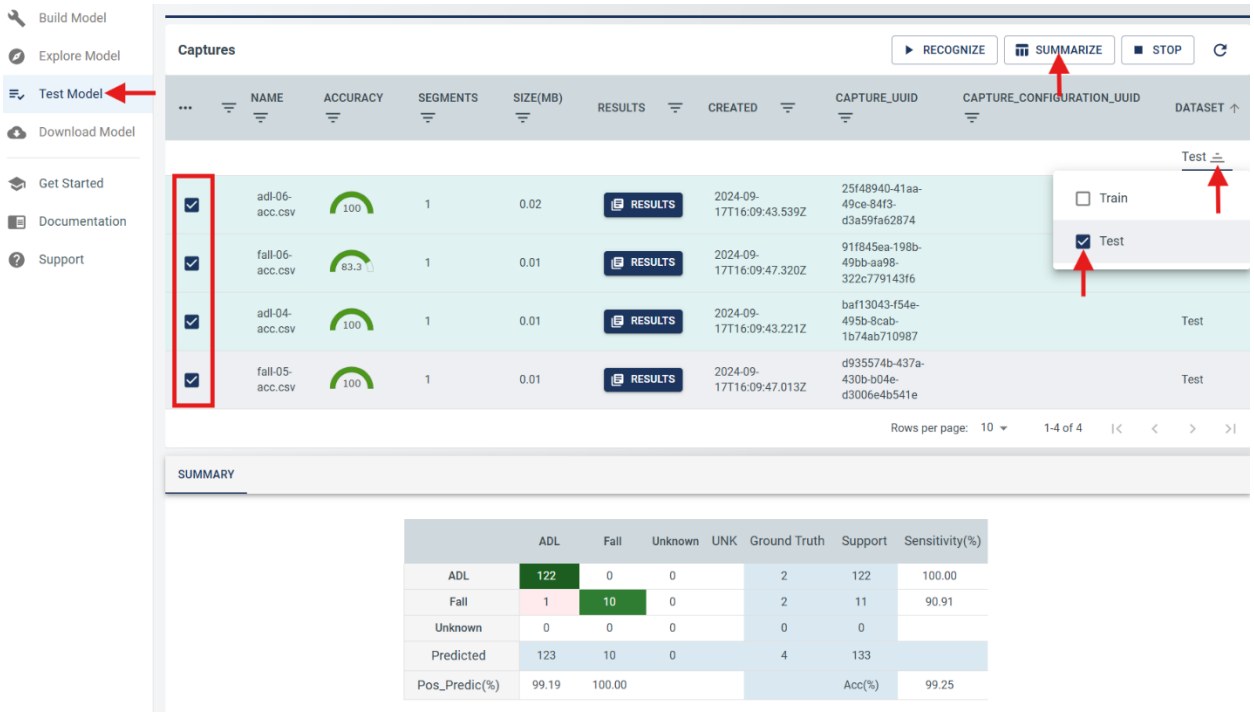


Figure 1-49. Testing the Model and Confusion matrix summary.

To deploy your model, follow these steps:

1. Navigate to the Download Model tab.
2. Fill out the Knowledge Pack settings using the Pipeline, Model and Data Source you created in the previous steps.
3. Select the library output format.
4. Click the **Download** button.

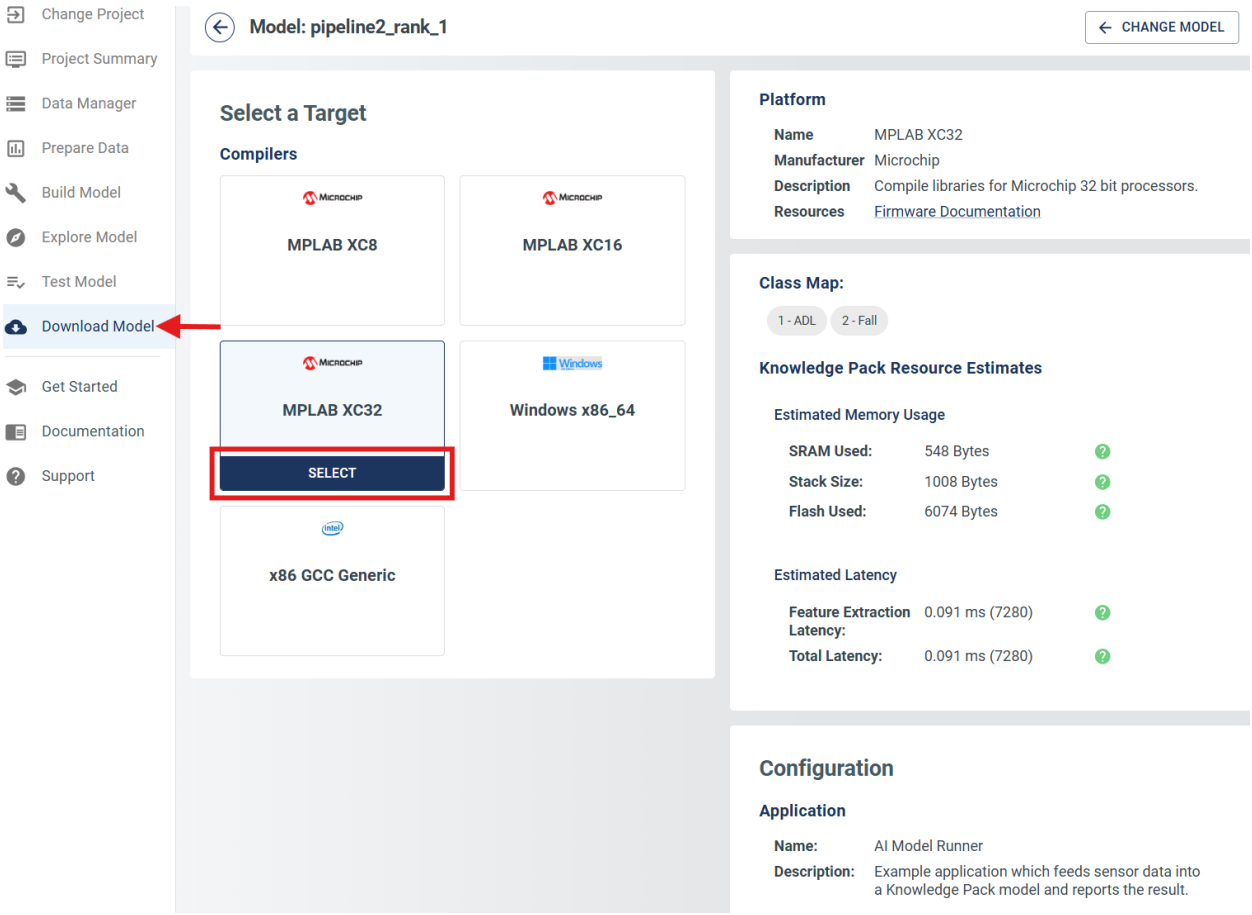


Figure 1-51. Downloading the model.

By completing these steps, you will be able to deploy your model and make it available for use. It is important to ensure that all necessary settings are correctly entered before

proceeding with the download.

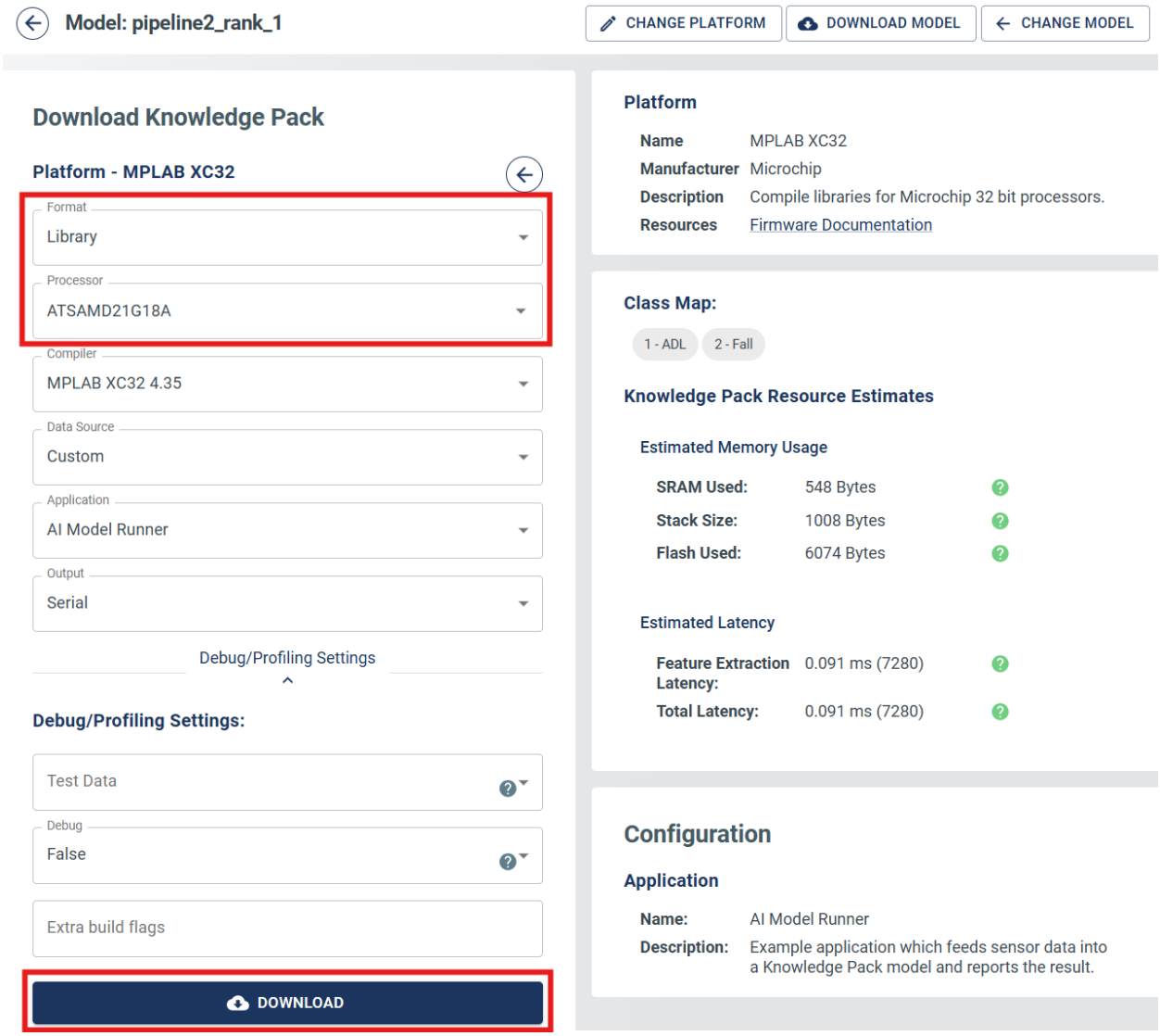


Figure 1-52. Selecting the parameters for downloading the model.

Knowledge Pack Integration

Integrate the Knowledge Pack APIs into your embedded device firmware by following the Knowledge Pack Integration.

Final Remarks

Upon completing this guide, you will possess a foundational knowledge of developing a Fall Detection application utilizing MPLAB ML Development Suite in conjunction with the SAMD21 and BMI160 6DOF IMU sensor ML evaluation kit. For additional insights into machine learning capabilities at Microchip, please refer to the resources available on Artificial Learning and Machine Learning.