Chapter 7: SensiML

After completing this chapter, you will understand how to use the partner machine learning solution from SensiML to develop and train ML models for use with Infineon MCUs. You will work with solutions using inputs from motion and audio sensors.

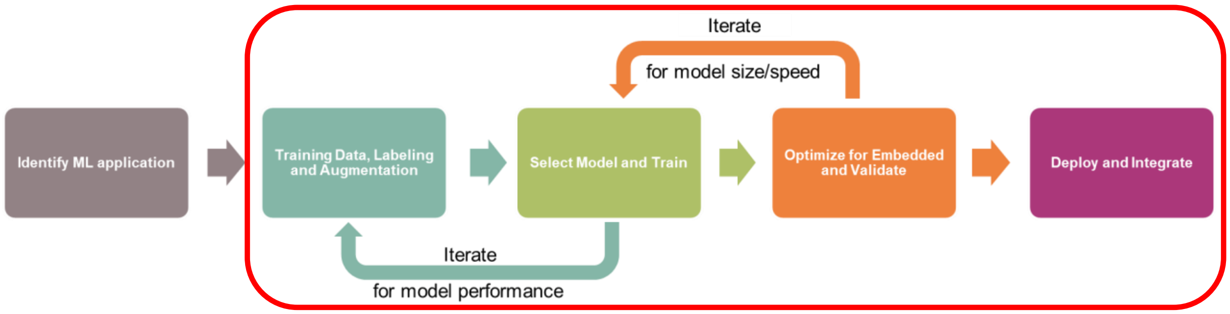
Table of contents

Document conventions

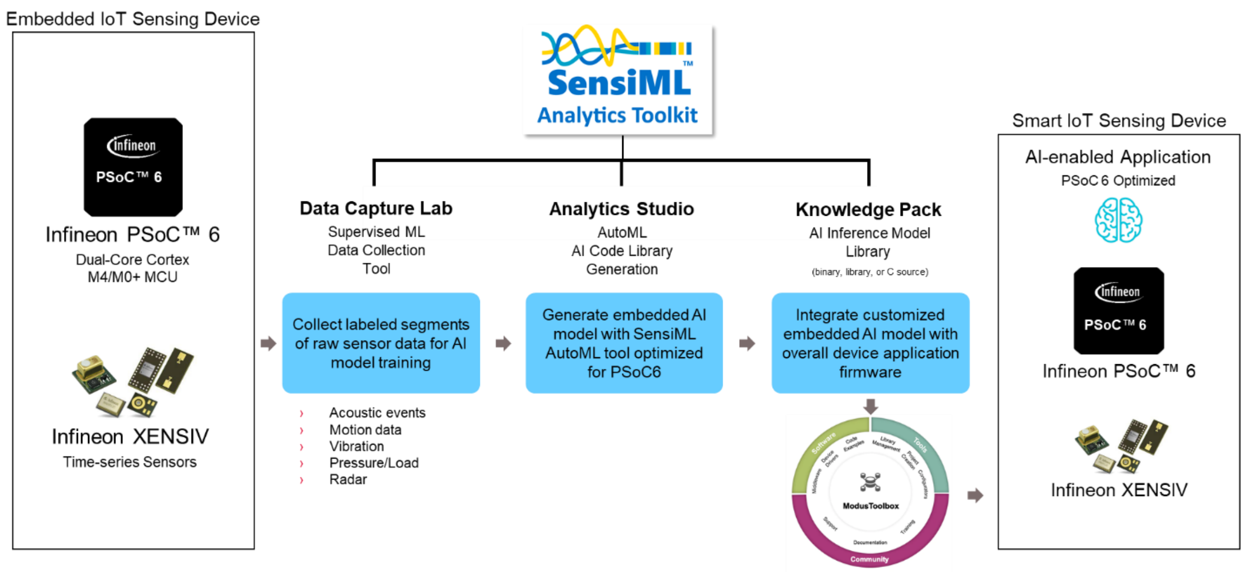
Convention	Usage	Example
`Courier New`	Displays code and text commands	`CY_ISR_PROTO(MyISR);` `make build`
Italics	Displays file names and paths	*sourcefile.hex*
[**bracketed, bold**]	Displays keyboard commands in procedures	[**Enter**] or [**Ctrl**] [**C**]
Menu > Selection	Represents menu paths	**File > New Project > Clone**
Bold	Displays GUI commands, menu paths and selections, and icon names in procedures	Click the **Debugger** icon, and then click **Next**.

7.1 Overview

In this chapter we will be using one of Infineon's Machine Learning partners - SensiML (https://sensiml.com) - to help facilitate and automate the steps of capturing raw data from a sensor, labeling the data, generating and training a machine learning model, generating embedded code optimized for a PSoC™ 6 MCU and validating the model on the target hardware.

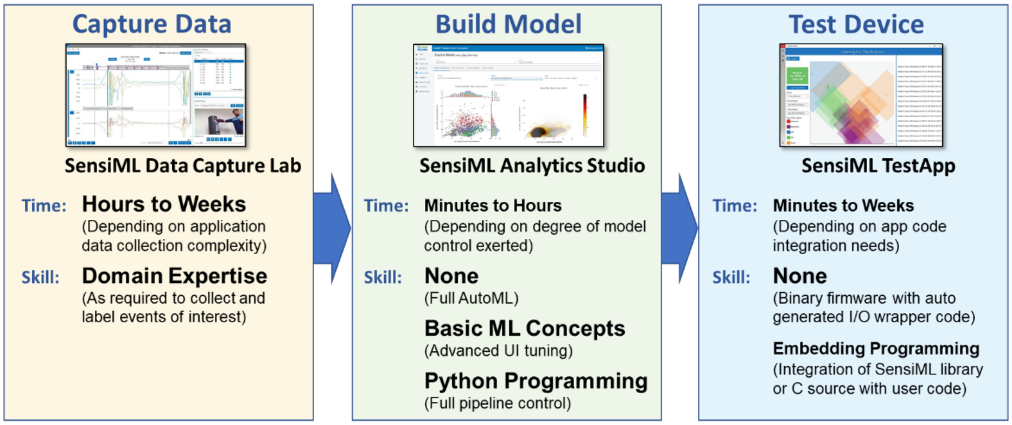


The SensiML software consists of the SensiML Analytics Toolkit, which is broken into three parts. Data from the PSoC™ 6 and its attached sensor(s) is captured using Data Capture Lab. That tool also provides methods to label the data once captured. When the dataset is ready, Analytics Studio is used to generate and train a model that is optimized for PSoC™ 6. Finally, the Knowledge Pack is used to test the newly created model on the target hardware using a SensiML code example inside ModusToolbox™.



7.2 Workflow

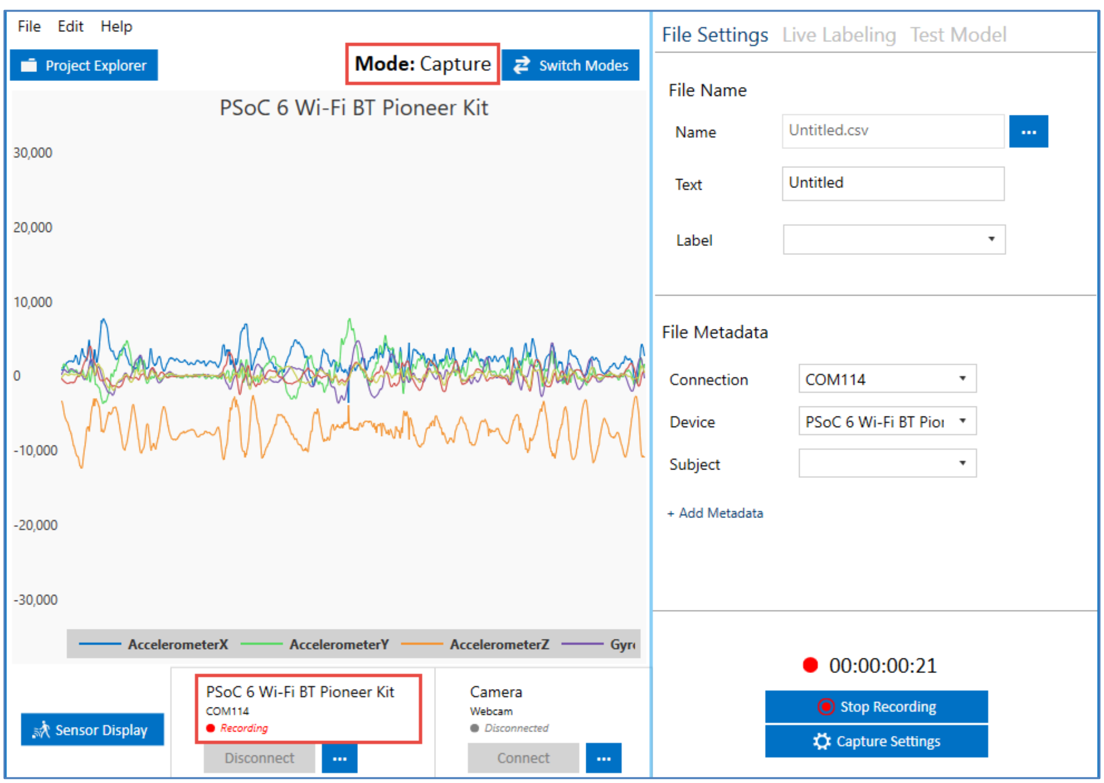
Now we'll go into more detail about the workflow in SensiML and the skills required at each step.

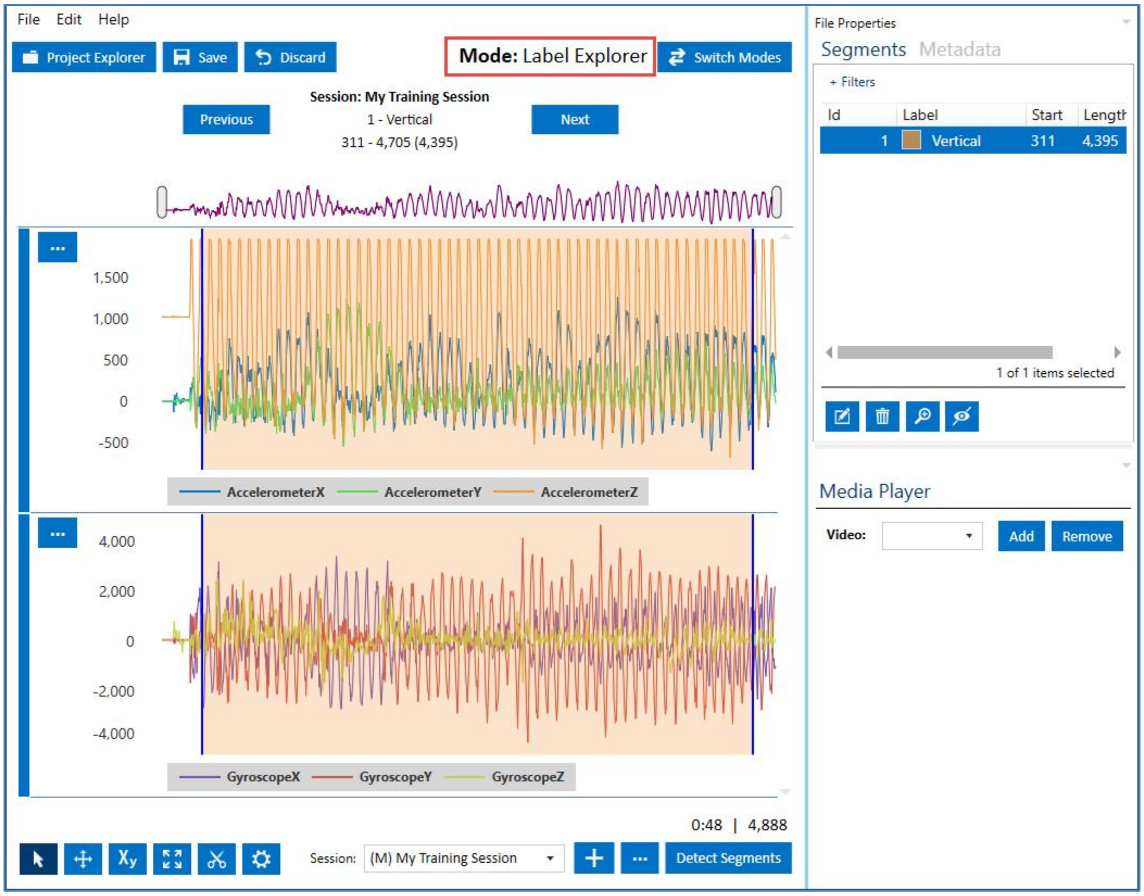


7.2.1 Capture and label data

As you learned earlier, getting a good dataset is one of the largest factors in developing a successful machine learning solution. With SensiML, the Data Capture lab can be used to quickly collect data directly from your sensor of choice and label it. As you will see, automated segmentation of input data can make it much easier to label large datasets. A code example provided in ModusToolbox™ allows you to program the PSoC™ 6 device and capture data from audio or motion sensors without requiring you to write any firmware. Data collection from additional sensors such as pressure and radar sensors will be added in the future.

You can even use your webcam to capture video while you perform the actions so that you can see what the motions look like for each sample.





The person(s) responsible for capturing data will need some expertise in the end application. That is, they will need to understand how the sensor(s) will be used in the real world and what the valid range of inputs looks like in order to ensure that the dataset contains sufficient information to create a good ML model.

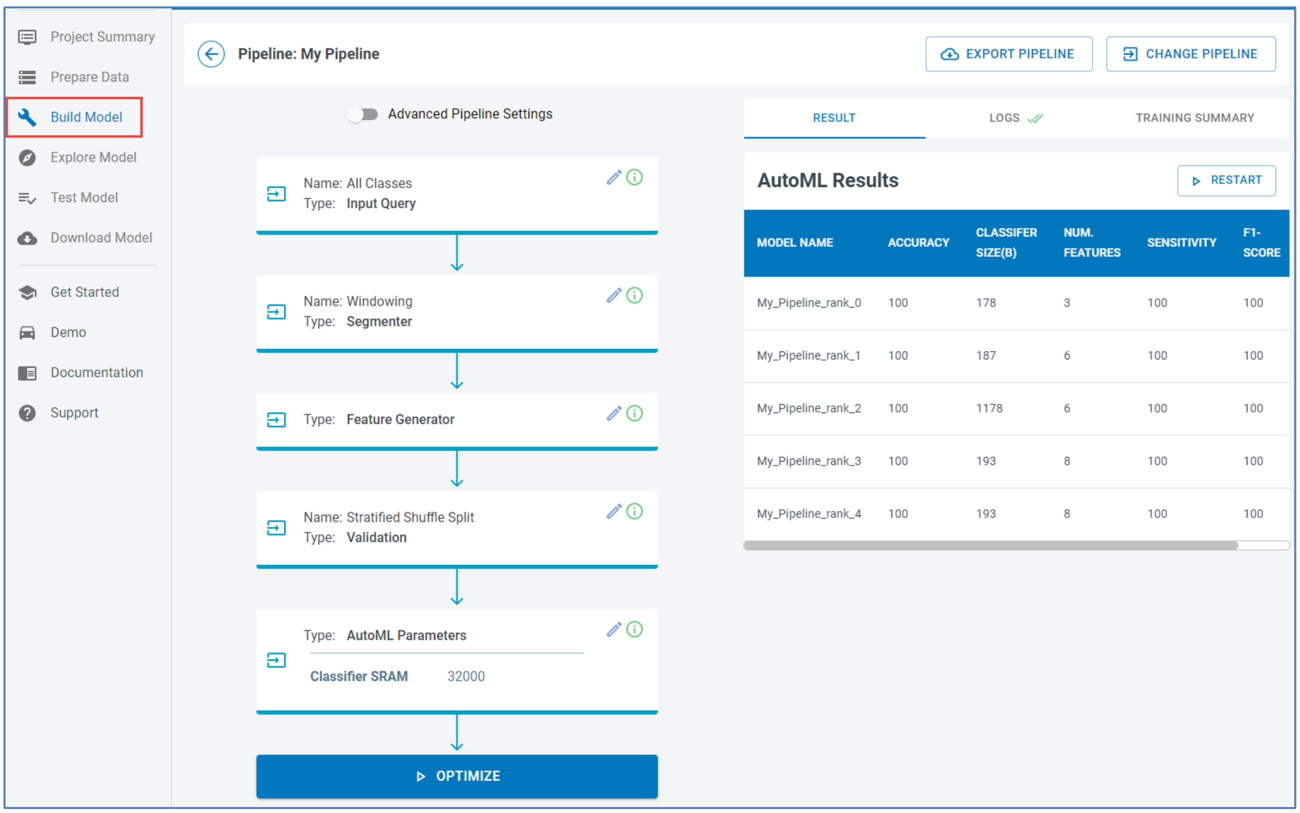
Once you have finished collecting data, it will be available on the cloud via your SensiML account and you can move on to building the model.

7.2.2 Build a model

Once the dataset is complete, it's time to create and train a model. For these steps you will use Analytics Studio, which is a Web-based application. The high-level steps are:

1. Open the project containing the sensor data you collected
2. Create a query to select the sensor data to use
3. Set up a pipeline to create and train a set of models
4. Run the pipeline on SensiML's servers

The set of instructions and the configuration used to create and train the model is called a "pipeline" in the SensiML ecosystem. SensiML Analytics Studio has several methods that can be used to set up and run a pipeline. The simplest is to use full AutoML. Using this method, the tool tries out different machine learning models on your dataset and trains each one of them to determine which model provides the best solution. For this method, very little machine learning knowledge is required. The tool does everything for you.



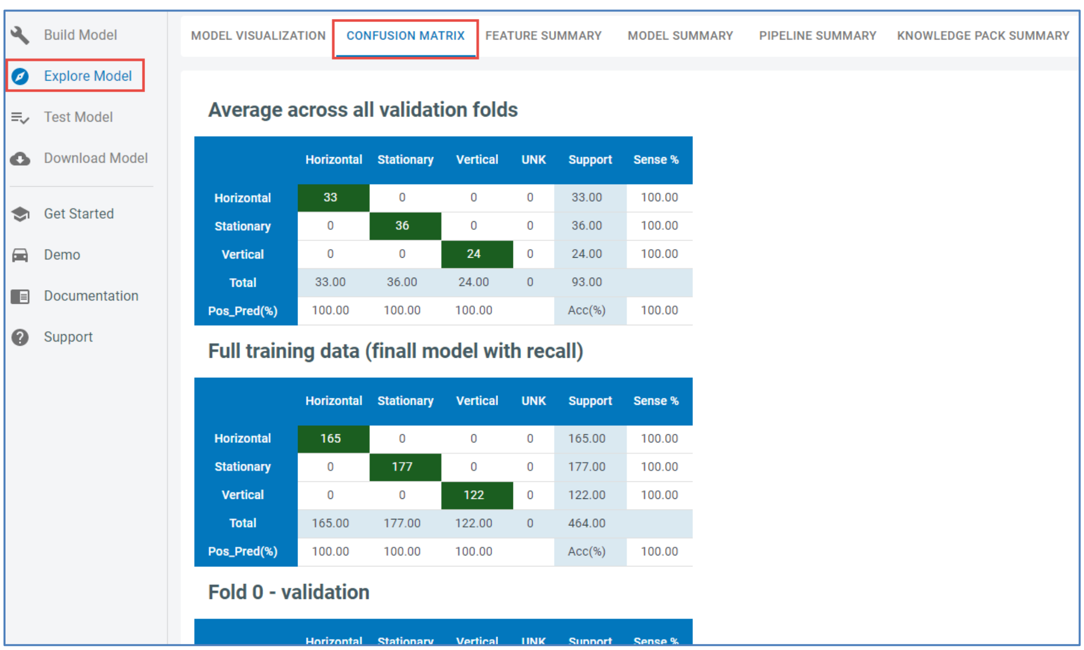
If you want a bit more control, you can use the advanced tuning UI. This allows you to use basic understanding of machine learning concepts to fine-tune the pipeline to potentially get even better results.

Finally, if you want full control of the Analytics Studio pipeline, you can do that too. With this method, you use Python commands to control exactly how the model is generated and trained.

7.2.3 Analyze the model

When you use the AutoML flow, it will produce several possible models for you. Once the training is done, you will want to look at each model to see how it performs and how much memory it will consume. You should select the best model for your purposes based on this analysis.

In Analytics Studio, you can look at charts that show how well the model distinguishes between data points, how well it performs against the test data, what features are used, and what layers are used, etc.



Average across all validation folds

	Horizontal	Stationary	Vertical	UNK	Support	Sense %
Horizontal	33	0	0	0	33.00	100.00
Stationary	0	36	0	0	36.00	100.00
Vertical	0	0	24	0	24.00	100.00
Total	33.00	36.00	24.00	0	93.00	
Pos_Pred(%)	100.00	100.00	100.00		Acc(%)	100.00

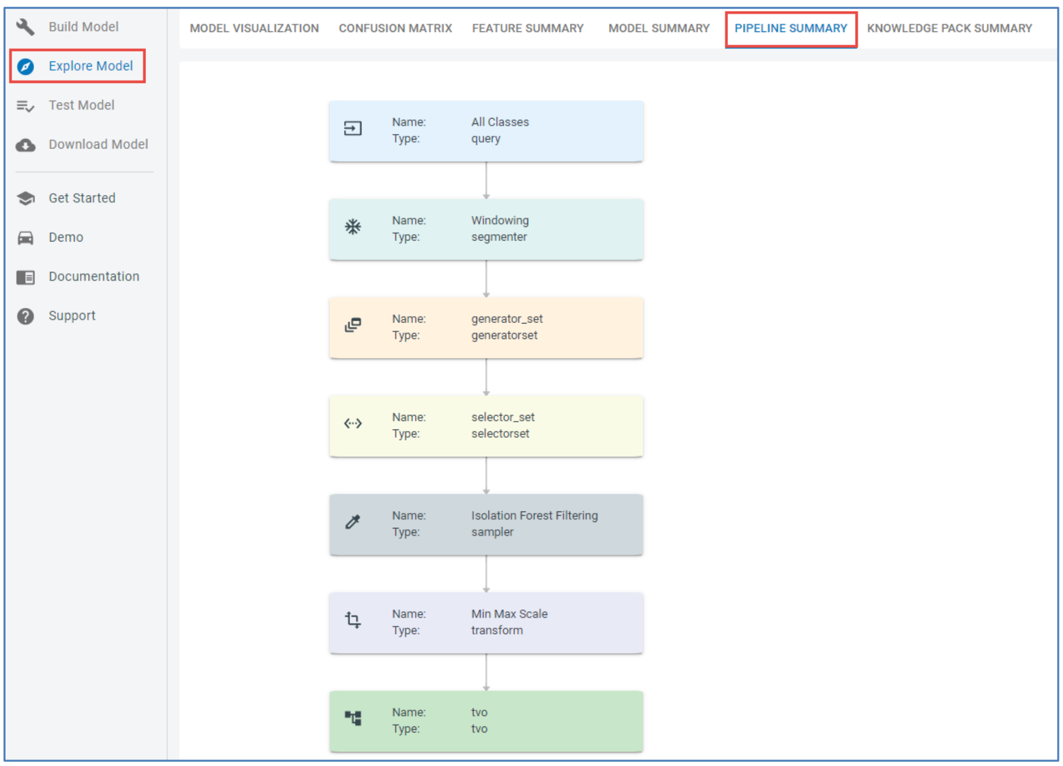
Full training data (finall model with recall)

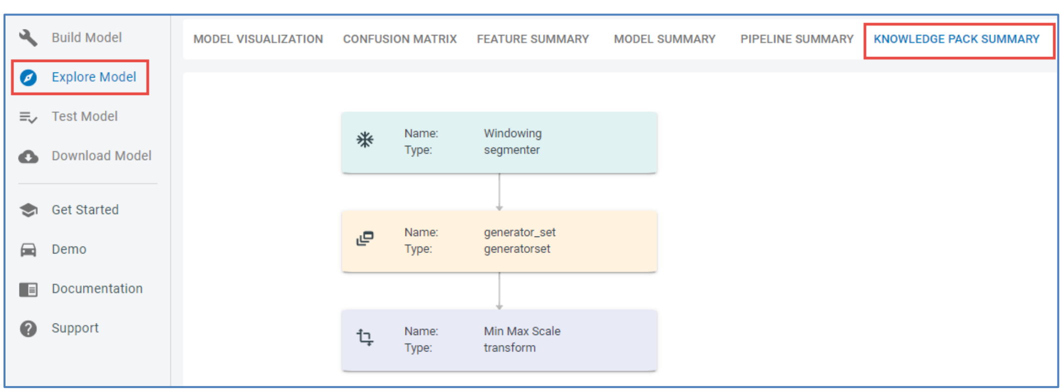
	Horizontal	Stationary	Vertical	UNK	Support	Sense %
Horizontal	165	0	0	0	165.00	100.00
Stationary	0	177	0	0	177.00	100.00
Vertical	0	0	122	0	122.00	100.00
Total	165.00	177.00	122.00	0	464.00	
Pos_Pred(%)	100.00	100.00	100.00		Acc(%)	100.00

Fold 0 - validation

	Horizontal	Stationary	Vertical	UNK	Support	Sense %


Build Model
Explore Model
Test Model
Download Model
Get Started
Demo
Documentation
Support
MODEL VISUALIZATION
CONFUSION MATRIX
FEATURE SUMMARY
MODEL SUMMARY
PIPELINE SUMMARY
KNOWLEDGE PACK SUMMARY
FEATURE
CATEGORY
GENERATOR
SENSORS
gen_0061_AccelerometerZmaximum Statistical Maximum AccelerometerZ
gen_0145_GyroscopeZmfe_000006 Frequency MFE GyroscopeZ
gen_0145_GyroscopeZmfe_000013 Frequency MFE GyroscopeZ
gen_0145_GyroscopeZmfe_000019 Frequency MFE GyroscopeZ
gen_0145_GyroscopeZmfe_000022 Frequency MFE GyroscopeZ
1-5 of 8
My_Pipeline_rank_4 Feature Vector Distribution
Feature Values
250
200
150
100
50
0
gen_0145_GyroscopeZmfe_000006
gen_0145_GyroscopeZmfe_000013
gen_0145_GyroscopeZmfe_000019
gen_0145_GyroscopeZmfe_000022
Horizontal
Stationary
Vertical


Build Model
Explore Model
Test Model
Download Model
Get Started
Demo
Documentation
Support
MODEL VISUALIZATION
CONFUSION MATRIX
FEATURE SUMMARY
MODEL SUMMARY
PIPELINE SUMMARY
KNOWLEDGE PACK SUMMARY
Name: All Classes
Type: query
Name: Windowing
Type: segmenter
Name: generator_set
Type: generatorset
Name: selector_set
Type: selectorset
Name: Isolation Forest Filtering
Type: sampler
Name: Min Max Scale
Type: transform
Name: tvo
Type: tvo


Build Model
Explore Model
Test Model
Download Model
Get Started
Demo
Documentation
Support
MODEL VISUALIZATION
CONFUSION MATRIX
FEATURE SUMMARY
MODEL SUMMARY
PIPELINE SUMMARY
KNOWLEDGE PACK SUMMARY
Name: Windowing
Type: segmenter
Name: generator_set
Type: generatorset
Name: Min Max Scale
Type: transform

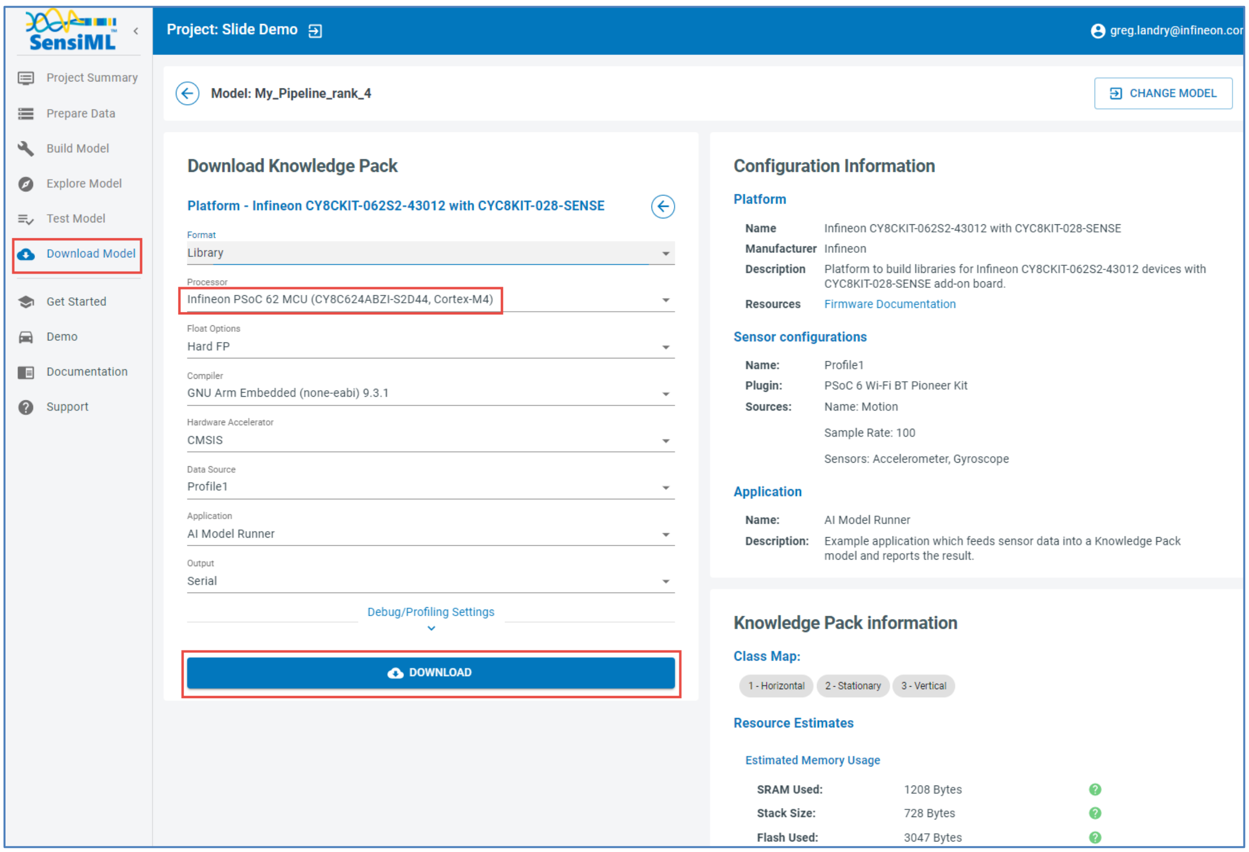
7.2.4 Test the model

Once you have decided on a model, you can use Analytics Studio to feed in new test datasets and see how well it performs on never-before-seen data.

In Data Capture Lab you can also run new test datasets against the model or you can stream data from the device in real-time using the same firmware that was used to collect the original datasets. With either option, you can use the new data to re-train the model for enhanced performance.

7.2.5 Test on the device

The final step to verify the model is to test on the target device to make sure it behaves as expected. You will download a "Knowledge Pack" which contains everything the model needs to run on a PSoC™ 6. The SensiML code example in ModusToolbox™ that was used to collect the data earlier can now be used to test the completed model from the Knowledge Pack. No special skills are required to test the model, however, once you want to deploy the model into your final end application, some embedded programming skills will be necessary.



7.2.6 Deployment

Once you have the final model, you will want to deploy it in your end application. You can either add your application's functionality to the code example that was used to verify the model in the previous step or you can integrate the model into another application. Either way, it is useful to review how the model is used in the code example.

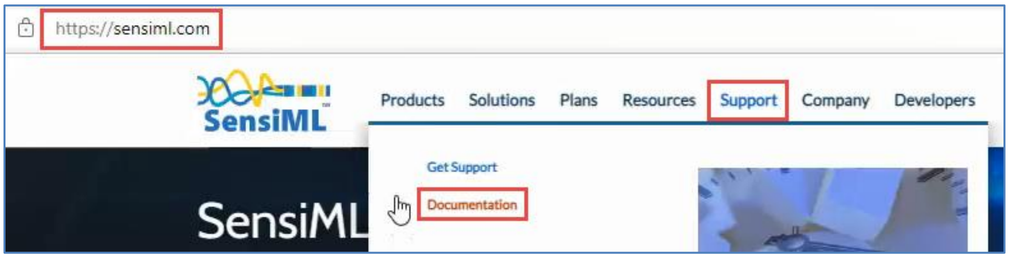
You can find documentation on the SensiML API used to interface with the model from the website. On the left panel, go to **User Documentation > Knowledge Packs / Model Firmware > Knowledge Pack Functions**.

7.3 Getting Started

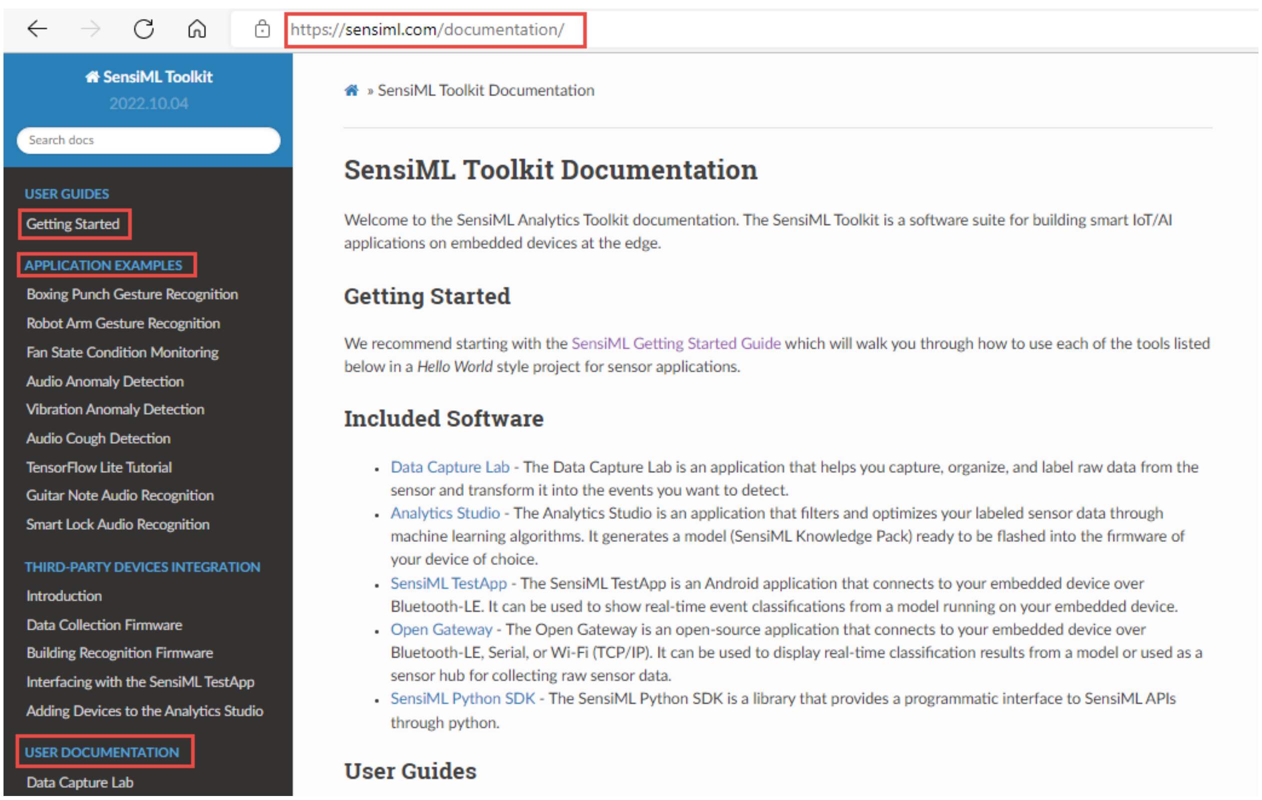
The SensiML web site has lots of useful documentation. Rather than repeating that information here, let's take advantage of their existing documentation. First, go to the SensiML main webpage:

https://sensiml.com

From that page, select **Support** and then **Documentation**.

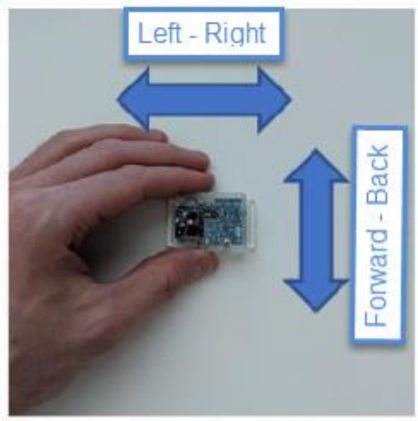


The documentation page includes links to the Getting Started Guide, lots of application examples, tool documentation, supported devices and compilers, and even directions on how to integrate other devices into SensiML.



The Getting Started Guide provides a tutorial that goes through the entire SensiML flow for an application that uses the motion sensor to detect movement of the kit. It will detect the following three movement conditions:

1. Stationary
2. Horizontal (Left – Right or Forward - Back)
3. Vertical (Up - Down)



When you do this yourself, you will follow along in the SensiML Getting Started Guide for detailed instructions of each step, but first, here is a high-level overview of what you will be doing:

7.3.1 Create an account

You will create a free SensiML evaluation account. This account never expires and supports many of SensiML's key tools and features limited to 3 hours/month of CPU time on SensiML's AutoML servers. Don't worry – that's plenty of time for completing the exercises in this chapter. For additional account options, visit: https://sensiml.com/plans/.

7.3.2 Create your first SensiML project

After you have an account and have installed Data Capture Lab, you can create a new SensiML project. In the first exercise, you will follow the instructions in the Getting Started Guide to create the motion sensor example described above.

7.4 Exercises

Exercise 1: Install SensiML tools

Note: If you already did the installation in Chapter 1 you can skip this exercise.

☐ 1. Go to the following page: https://sensiml.com/download

☐ 2. Download and install SensiML Data Capture Lab.

☐ 3. Download and install SensiML Open Gateway.

☐ 4. (Optional) Download and install SensiML Python SDK.

Note: SensiML tools are only available for Windows.

Note: The SensiML Python SDK is only needed for some of the advanced applications.

Note: If you install the Python SDK, run the pip command from modus-shell.

Note: SensiML Analytics Studio runs in the cloud, so no installation is required for that tool.

Exercise 2: Create a SensiML account and complete the getting started tutorial

In this exercise, you will try out the full SensiML flow by following the Getting Started Guide. It uses the full AutoML feature in Analytics Studio to create and train the model. Labeling is done using a manual labelling session.

☐ 1. Create the SensiML template firmware code example in ModusToolbox™. Name the application: **ch07_ex01_sensiml_motion**.

In the file *source/app_config.h*:

Set `SENSOR_SELECT_MODE` to `SENSOR_MOTION`.
Set `SENSOR_SCAN_RATE` is set to `MOTION_SAMPLE_RATE_100HZ`.
Verify `APPLICATION_RUNNING_MODE` is set to `DATA_CAPTURE_RUNNING_MODE`.

Note: Those settings will allow the Data Capture Lab tool to capture motion sensor data from the sensor on the kit.

☐ 2. In the *Makefile*, verify that `SHIELD_DATA_COLLECTION` is set to the correct value for the shield board you are using.

Note: If the silk screen on the back of the shield says "9-axis IMU" and "600-60608-01 REV03" it is v1. If the silk screen says "6-axis IMU" and "600-60608-01 REV04" it is v2.

☐ 3. If you are using a Mac, you may have to modify the following line in the *Makefile* to work with csh instead of bash. The necessary addition is a pair of single quotes.

Old: `PREBUILD+=sed -i 's/UINT8_C(0xD1)/UINT8_C(0xD8)/' bmi160/bmi160_defs.h;`
New: `PREBUILD+=sed -i  ''  's/UINT8_C(0xD1)/UINT8_C(0xD8)/' bmi160/bmi160_defs.h;`

☐ 4. Program the application to your kit.

☐ 5. Open a UART terminal to verify that data is streaming from the motion sensor. Once you have verified this, close the terminal so that Data Capture Lab will be able to access the port.

Note: The baud rate is 1000000.

6. Follow the SensiML Getting Started Guide at: https://sensiml.com/documentation/guides/getting-started/index.html (**sensiml.com > Support > Documentation > Getting Started**). Notes for some of the steps are provided below.

Note: ***Overview****: Data Capture Lab and Open Gateway are only available for Windows 10.*

Note: ***Data Capture Lab****: Create a new project rather than importing the DCL project. You will collect your own data using your kit and shield with the application that was programmed above.*

Note: ***Capturing Sensor Data:***

Use the same sample rate as the value set in the application for `SENSOR_SCAN_RATE`. *Otherwise the connection to the PSoC™ will fail.*

If you want to use a camera to capture video while performing the gestures, you need to install the SensiML Open Gateway application.

Note: ***Labeling Your Data****:*

Since each file only contains one gesture being repeated, you can add one large segment to each file during the time that you were performing the gesture. Be sure to exclude any time at the start and end when you were not performing the gesture.

Use the sliders above the charts to zoom in and out.

Note: ***Querying Data****: When creating a query, leave the Metadata field set to the default value of segment_uuid instead of Subject. Otherwise the query will not work.*

Note: ***Building a Model****:*

The pipeline creation screen is under the Build Model screen that you can select along the left pane.

When creating the pipeline, select the All Cases query that you set up in the previous step in the Query drop-down box.

Once you click Optimize, it will create and train the models, which will take a few minutes.

Note: ***Running a Model in the Data Capture Lab****:*

Try testing the model in Data Capture Lab using real time data before programming to the device.

If you open the Test Model tab in the right panel, click Connect and it says there are no packs available, do the following: click the refresh button, select the model, and then try to connect again.

Note: ***Running a model in On Your Embedded Device**:*

Test the model on the target device using Open Gateway.

To incorporate the model into the ModusToobox™ application and to configure the application to run the model instead of streaming raw sensor data, follow the instructions from Supported Devices > Infineon PSoC™ 6 > Download/Flash Model Firmware (except steps d and e below). The steps are:

a. Download the Knowledge Pack from Analytics Studio. Select "Infineon CY8CKIT-062S2-43012 with CY8CKIT-028-SENSE" as the development platform.

b. Unzip the Knowledge Pack file on your computer.

c. Copy the Knowledge Pack files from sensiml/lib and sensiml/inc to the application's source/lib directory. These will overwrite existing files.

d. Copy the Knowledge Pack files from sensiml/model.json to the application's source/lib directory. This will overwrite the existing file.

e. Do NOT copy the file from knowledgepack_project/app_config.h to the application's source directory. Instead, open source/app_config.h from the application and change the value of `APPLICATION_RUNNING_MODE` to `RECOGNITION_RUNNING_MODE`.

f. Clean the application before programming it to the device.

g. Open a UART terminal (baud 1000000) to make sure the application is outputting model information. Disconnect from the UART and then use Open Gateway to visualize the results from the model.

h. In Open Gateway, set the Device Mode to Recognition, Connection Type to Serial, Device ID to the appropriate COM port and baud rate to 1000000. Click the Connect to Device button. Once it is connected, select Test Mode from the left panel to see the model results in real-time. Select Home and click Disconnect Device when you are done.

Exercise 3: Run some application examples

In this exercise, you will look through the application examples provided by SensiML and try out one or more of the examples.

The various applications showcase different features of SensiML such as model creation using full AutoML (which you used in exercise 2), the advanced tuning UI for more control, and the SensiML Python SDK for complete control over the model creation. Some applications use the automatic labelling feature available in Data Capture Lab to simplify the process of creating labels for large amounts of data.

☐ 1. Review the list of application examples on the SensiML site (**sensiml.com > Support > Documentation > Application Examples**). Try out one or more examples to see how different features of SensiML are used. Two possibilities that may interest you are:

- The Guitar Note Audio Recognition example which uses the advanced tuning UI to create a model using TFLM.
- The Boxing Punch Recognition example which uses the SensiML Python SDK to exert full control over the model. It applies the automatic labelling feature in Data Capture Lab.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Published by
Infineon Technologies AG
81726 Munich, Germany