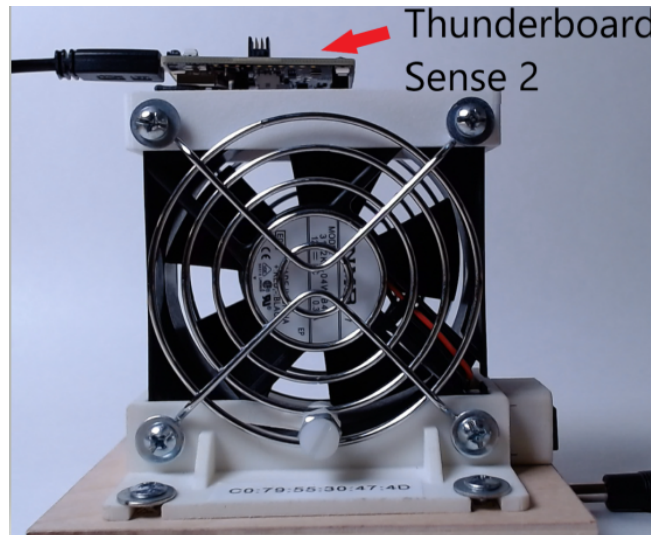


Overview

In this tutorial, we are going to use the [SensiML Analytics Toolkit](#) to build a predictive maintenance application that will run entirely on the SiLabs Thunderboard Sense 2.



Objective

1. Demonstrate how to collect and annotate a high-quality dataset of machine state for the fan using the SensiML Data Capture Lab.
2. Build a data pipeline to extract features in real-time on your target device.
3. Train a Classification model using SensiML AutoML.
4. Convert the model into a Knowledge Pack and flash it to our target embedded device.
5. Perform live validation of the Knowledge Pack running on-device using the SensiML TestApp

Prerequisites

The Hardware

- [Silicon Labs Thunderboard Sense 2](#)
- Fan (any fan will work)
- Mounting Equipment (sturdy double-sided sticky tape or mounting screws)

The Software

- [SensiML Data Capture Lab](#)
- SiLabs Thunderboard Sense 2 [data collection firmware](#)
- [SensiML Open Gateway](#) (optional: used to record video, view model results)
- [Silicon Labs Simplicity Studio 5](#) (optional: for building data collection firmware)
- [Silicon Labs Platform Applications](#) (optional: for building data collection firmware)

Data Sets

- Condition Monitoring Data Capture Lab [project](#),
- Condition Monitoring Captured [Sensor Data](#)

Create SensiML Community Edition account

- <https://sensiml.com/plans/community-edition/>

Capturing Sensor Data

For every machine learning project, the quality of the final product depends on the quality of your curated data set. Time series sensor data, unlike image and audio, are often unique to the application as the combination of sensor placement, sensor type, and event type greatly affects the type of data created. Because of this, you will be unlikely to have a relevant dataset already available, meaning you will need to collect and annotate your own dataset.

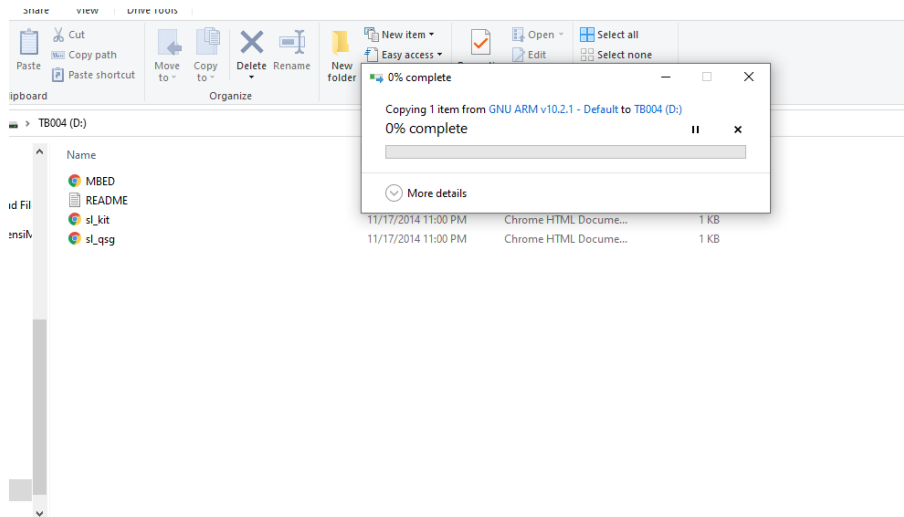
To help you to build a data set for your application we have created the SensiML [Data Capture Lab](#), which we are going to use for collecting and annotating data for different fan states.

Data Collection Firmware

In order to collect sensor data, you will need to flash the Thunderboard Sense 2 with data collection firmware. For this tutorial, we have already built the data collection firmware which you can flash to the device by copying it into the folder.

[Data Collection Firmware](#)

Download the above file and copy it to the thunderboard folder on your machine. The device will reboot and load the new firmware.

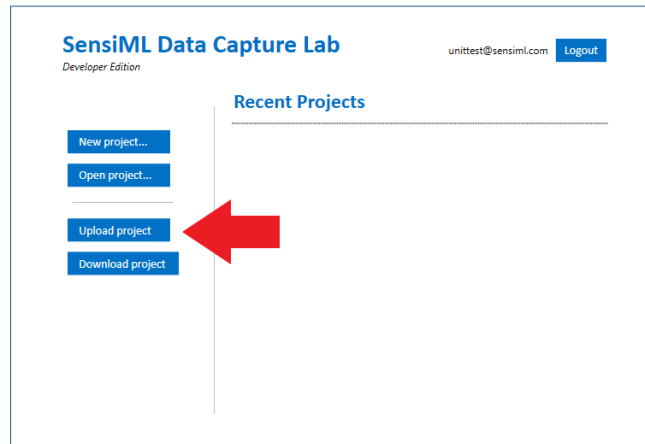


Detailed instructions for building and flashing the firmware can be found [here](#). You can follow those steps to modify the firmware to add different sensors or change other parameters such as the sample rate and sensor ranges

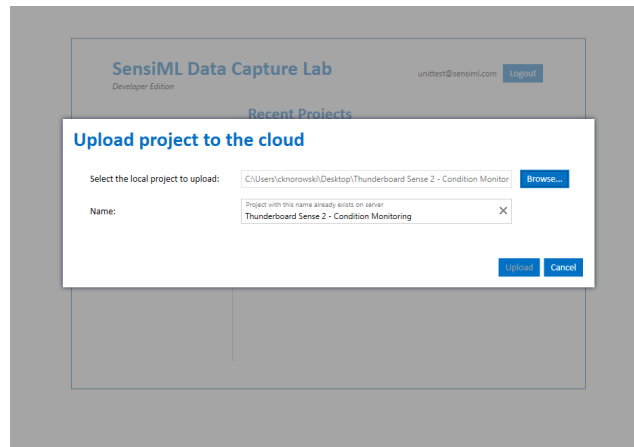
Connecting to the Firmware

To begin data collection, we will use the SensiML Data Capture Lab. We have created an initial project for you to get started quickly. First:

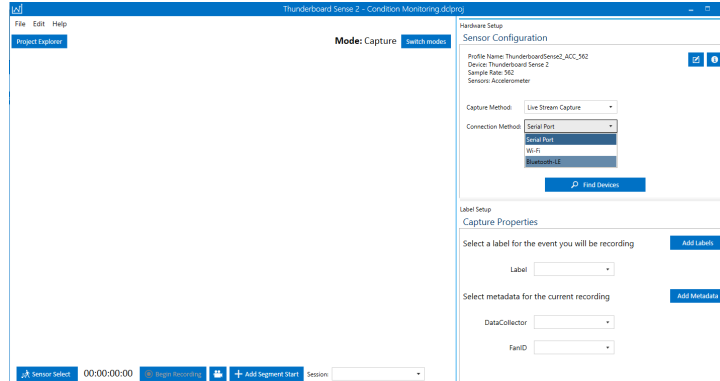
1. Download this [project](#), and unzip it.
2. Open the Data Capture Lab and select Upload Project



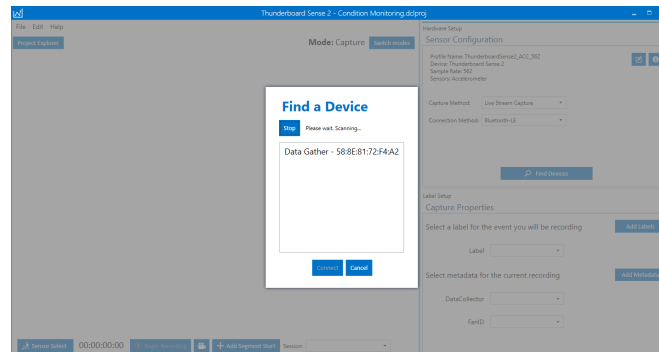
3. Select browse and find the .dclproj file inside of the Thunderboard Sense 2-Condition Monitoring Project folder you unzipped.
4. Click the **Upload** button to upload the project to your account. You are now ready to start collecting your own Sensor Data.



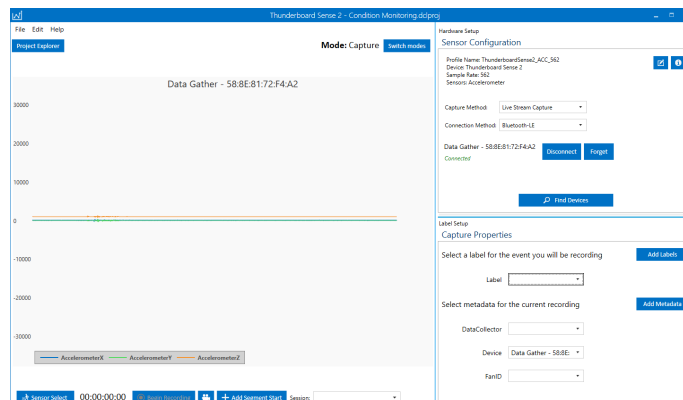
5. The Data Capture Lab opens to the Label Explore Screen, switch to the Capture Mode Screen by clicking the **Switch Mode** button and selecting **Capture** mode.



6. Now that you are in **Capture** mode we can connect to the sensor to record data.
7. First, switch the connection method to **Bluetooth LE**
8. Next, click the **Find Devices** button and then the **Scan** button. After it scans it will display a list of devices. Select the one corresponding to your board and click the **Connect** button.



9. Now you are ready to begin recording sensor data



Capturing environmental context

The data capture lab uses Labels to identify regions of sensor data as belonging to a specific class. For example, we would put a label around the region of sensor data that corresponds to the fan event **On**. Before we collect any data:

1. Identify which type of data you are going to collect.
2. Select the option from the **Label** in the dropdown menu.

Note: If you are going to be recording multiple types of events, I typically use the label **Unknown**. The file that is recorded will start with the name of the **Label** that you select.

In addition to labels, it is also important to capture information about the environment. Capturing this contextual information enables you to build highly tailored models. Additionally, it makes it possible to diagnose why a model might be failing for subsets of your data.

For example, in this fan state dataset, we captured the contextual properties **data collector** and **Fan ID**. This metadata can be useful in building models for subtypes of data as well as debugging problems with model accuracy.

You can capture the contextual information in the Data Capture Lab using the metadata properties. Metadata is applied to the entire captured file, so when you are creating your data collection strategy think carefully about what information you may need. Metadata can be created as a selectable dropdown or text field.

Recording data

The data collection firmware streams Accelerometer data at a sample rate of 562Hz using is Bluetooth LE radio. The captured data will be saved locally to your computer as well as be synced up to the SensiML Cloud. This allows other members of your team who have permission to see and label your new captured file. Alternatively, if you already have a data collection method for your device, the Data Capture Lab can import CSV and WAV files directly.

Labeling Sensor Data

Now that we have collected example sensor data, it is time to annotate the dataset by applying labels.

1. Open the **project explorer** and select a file that was captured with fan **On** data. This will open the file into Label explore mode.
2. You will see a single graph that has all of the **Accelerometer** data.
3. Right-click on the graph to create a segment around your event of interest.
4. Next, click the edit button to select the label that is associated with that event.

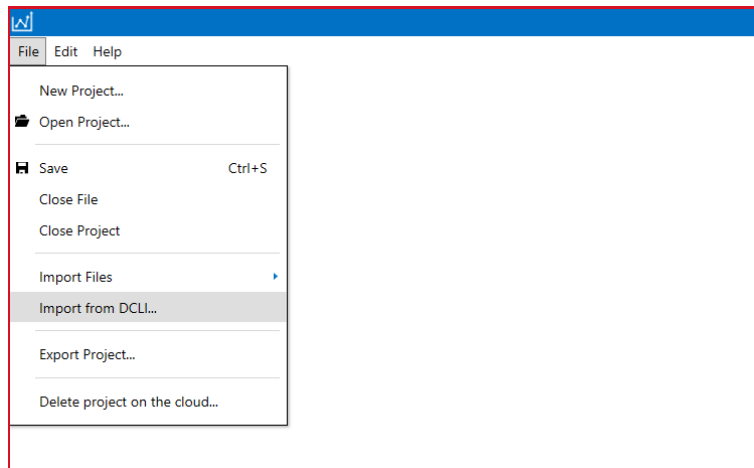
Repeat this process for the other files in your project until you have labeled them all.

Note: If you have collected your data set through other means, you can also import CSV or WAV files into the Data Capture Lab.

Note: For those of you that are not able to collect your own sensor data. You can download the sensor data from my project and import it into yours.

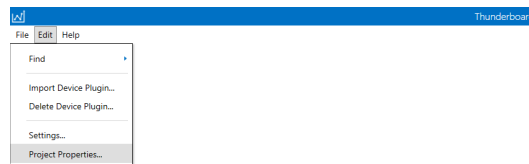
[Download Sensor Data](#)

Unzip the file, go to Import from DCLI and select the .dcli file to import the extra sensor data.

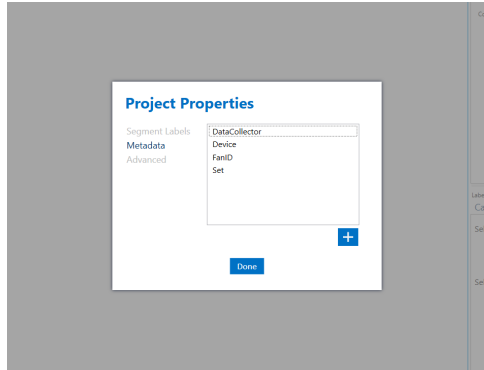


Finally, let's add a metadata property to the project called **Set** with two metadata labels **Train** and **Test**.

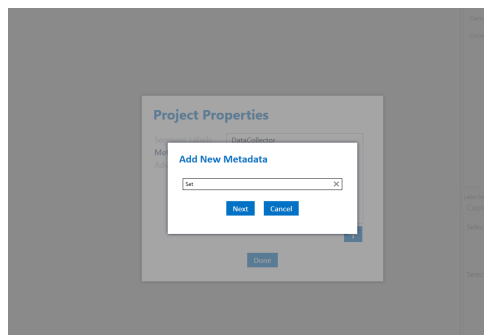
1. Click **Edit** in the toolbar and select **Project Properties**



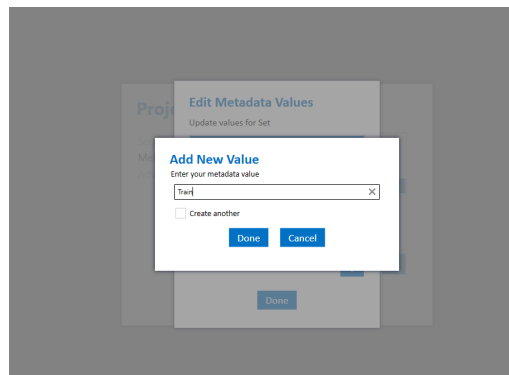
2. Click on the **Metadata** and then click the **+** button to add a new Metadata Property



3. Enter **Set** into the text field and click the **Next** button.

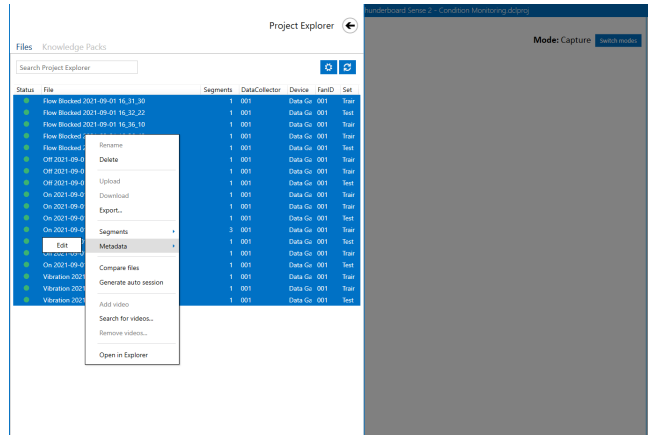


4. Click the **add** button and enter the name **Train**. Do that again for the label **Test**.

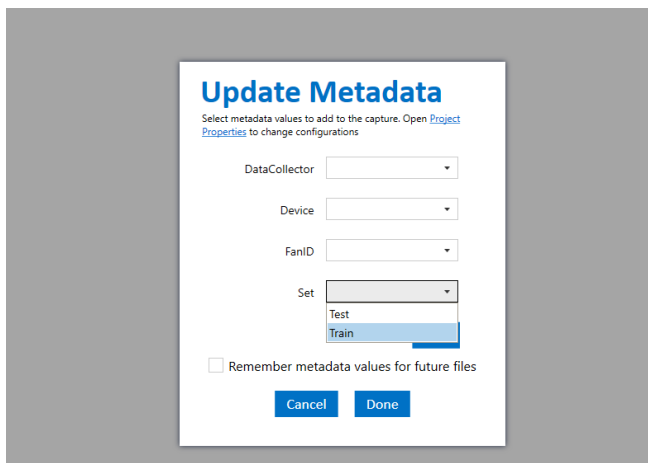


Now we will update the metadata labels for our project so that we can split the data into train and test sets. For this project, we will select one file from each of our classes to become part of our test set. Open the **project explorer** tab again.

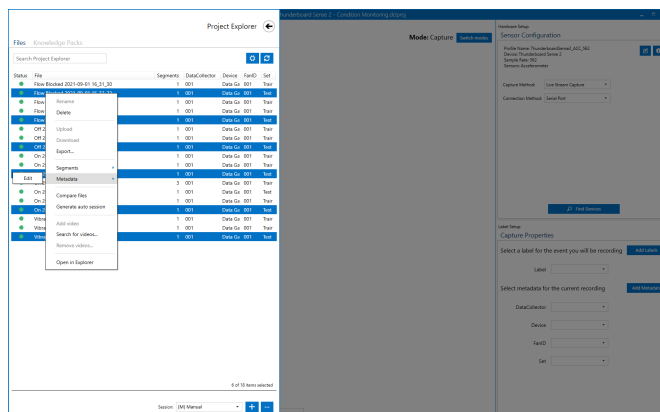
1. Select all of the files in the project explorer (click on and hit **ctrl+a**)
2. Then **right-click** and navigate to **Metadata->Edit**



3. Select **Train** from the **Set** dropdown.



4. Then Select one file from each of the event types (**ctrl+click**)
5. **right-click** and then **Metadata->Edit**
6. Set the selected **Set** to **Test**

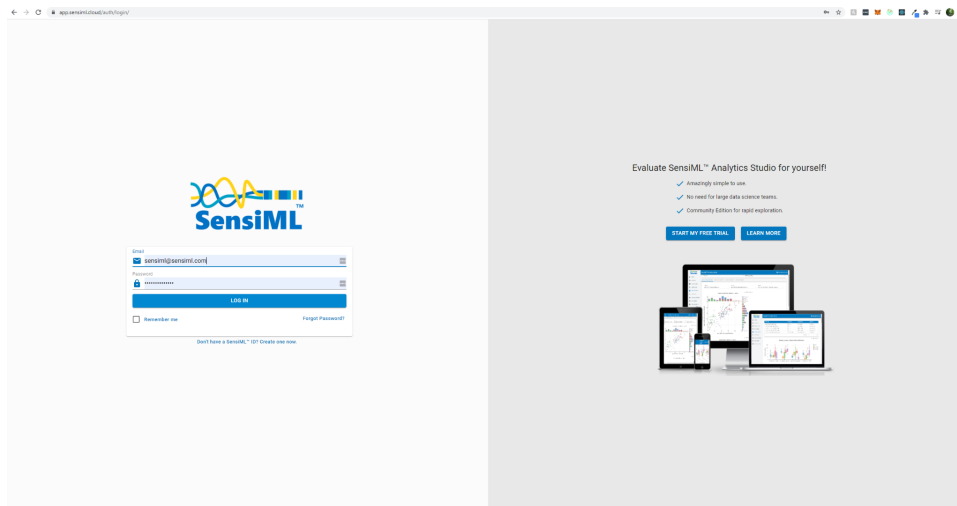


Now you know how to edit metadata for the files. We will use the **Set** label to create a model only from the **Train** data. Then test our models against the files in the **Test** set. This will allow us to better identify when the model is under/overfitting and if we need to collect more training data.

This concludes the data labeling part of the tutorial. We will take a break here for you to walk through these steps and ask any questions.

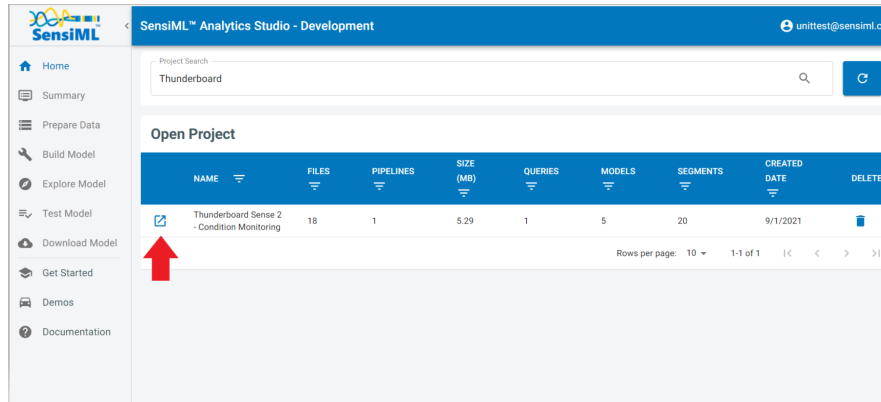
Building a Model

The SensiML Analytic Studio is where you can create a query to pull data into your model, build models using AutoML, validate model accuracy against held out test data and finally download your model as firmware code for the target device. For the next part of the tutorial, you will need to log into [Analytic Studio \(https://app.sensiml.cloud\)](https://app.sensiml.cloud)



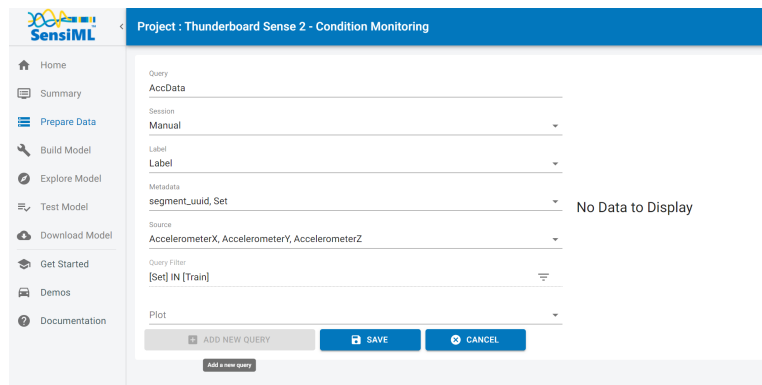
After logging in, open the **Thunderboard Sense 2 - Condition Monitoring** Project by clicking the open icon.

This will set the **Thunderboard Sense 2 - Condition Monitoring** as the active project in the data capture lab.



Prepare Data

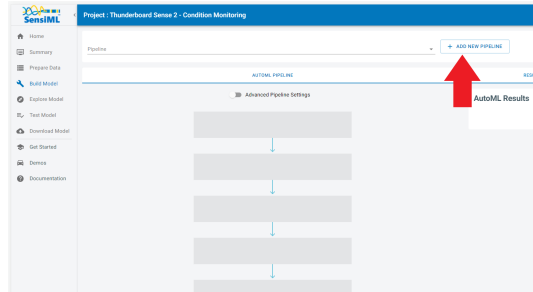
Next, we can create a query that will generate the input data for our modeling. Click on the **Prepare Data** tab on the left, and fill in the fields as shown in the image below.



Once you have filled in the fields, click save which will create the query that can then be used to build a model. You will select this query on the next screen. After you save the query, you will also see a graph that contains statistics about the size of the dataset that is going to be used.

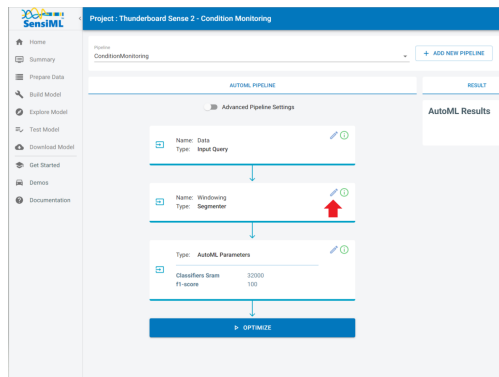
Setting up a pipeline

Now that we have created a query we can go to the build model screen. Click on the **Build Model** tab in the navbar on the left side. This will bring you to the model building screen. At this point, you have not yet created a pipeline. Click on the **+ Add New Pipeline** button and enter the name of your pipeline.

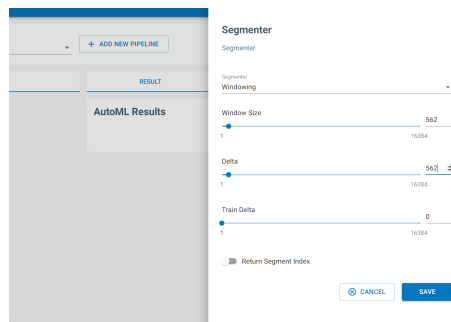


A default pipeline will be created with a set of predefined parameters. To change any parameters you can hit the edit button. We need to change the size of the Windowing segmenter which is set by default to classify every 250 samples of data. Which for us would be about every half second as we are sampling at 562Hz.

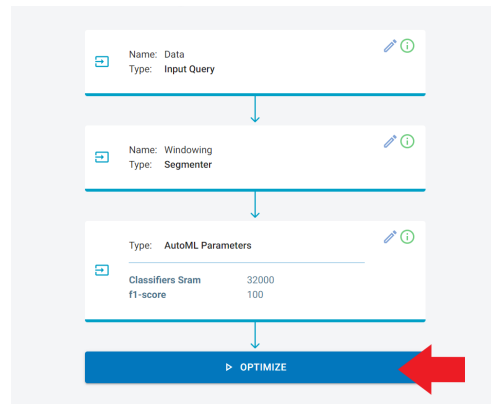
Click on the edit icon on the **Windowing** card.



This will open up an edit screen, in the edit screen set the **Window Size** and the **Delta** values to 562, which will give us a classification every 1 second with no overlap between data. After you change the values click the **Save** button to save your changes.



Next, click the **Optimize** button to start the search for the best model.

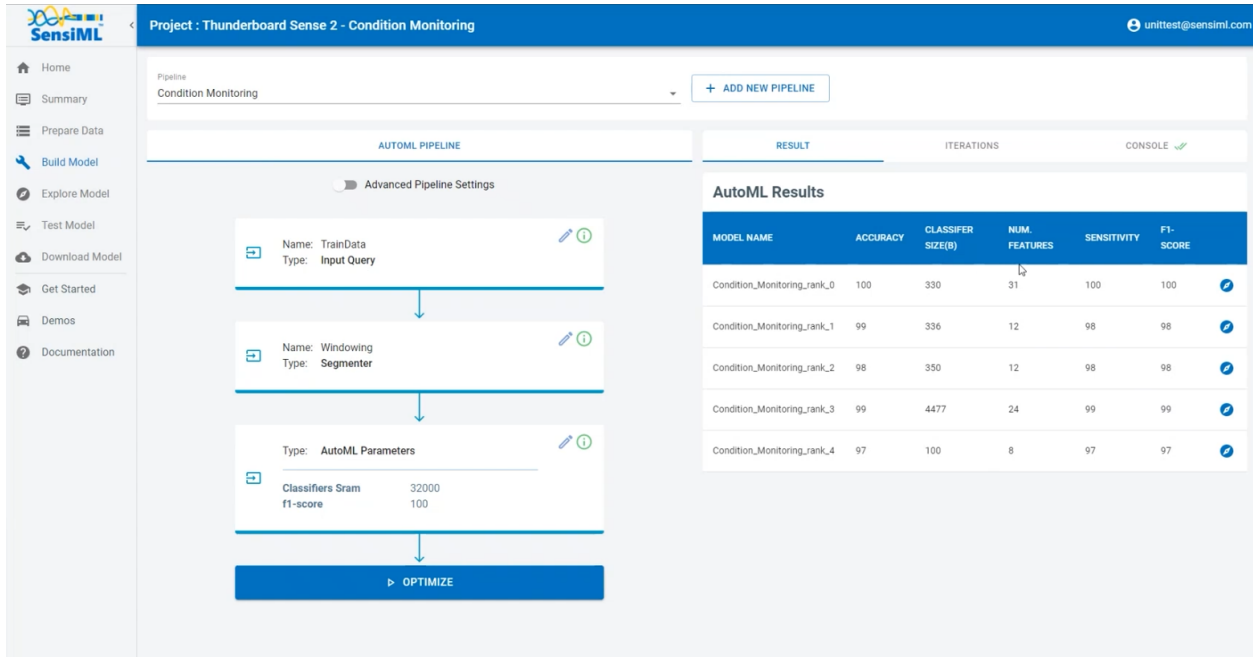


The SensiML Cloud will now start searching for the best model that meets your constraints.

Training the model with AutoML

AutoML is used to create a set of models within the desired statistical (accuracy, f1-score, sensitivity, etc.) and classifier size. As the algorithm iterates each optimization step, it narrows down the searching space to find a desired number of models. The optimization terminates when the desired model is found or the number of iterations reaches the max number of iterations.

We take advantage of dynamic programming and optimizations for training algorithms to speed up the computation. This makes it possible to search for large parameter spaces quickly and efficiently. The results are ranked by the fitness score which considers the model's statistical and hardware parameters.



Once the models have been generated, you can explore the details of the top five candidate models in the explore models tab. In this tab, there are visualizations and information about the models including, features, confusion matrix, model hyperparameters, and the Knowledge Pack training and inference pipeline.



Validating the model against the Test Set

Before you flash the model to the device, you can test the model using the **Test Model** tab. You can test against any of the captured data files. To do this:

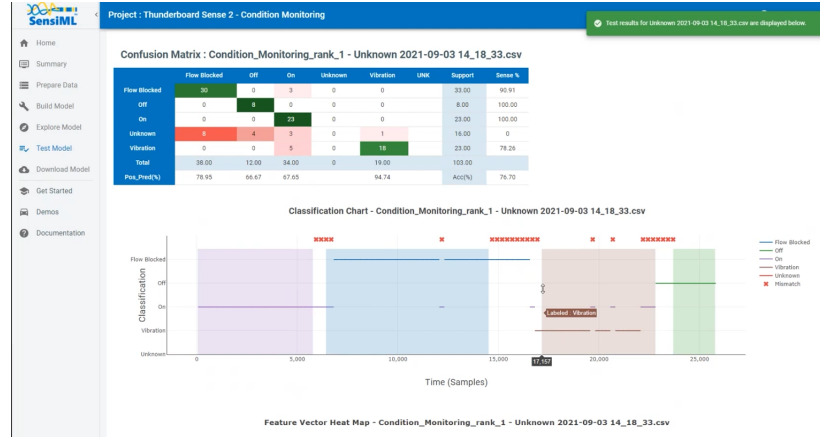
1. Go to the Explore Model tab of the Analytic Studio.
2. Select the pipeline you built the model with.
3. Select the model you want to test.
4. Add a filter to the **Set** column to only include **Test** data.
5. Click on the three dots menu and the top left->**Select All**
6. Click the **Compute Accuracy** button to classify the selected capture files.

CAPTURE NAME	ACCURACY	TOTAL EVENT COUNT	CAPTURE UID	RESULTS	SIZE (MB)	DATA COLLECTOR	DEVICE	FANID	SET
On 2021-09-03 14_09_42.csv		1	02957f2f-a958-4445-9f43-60288c79a367		0.44	001	Data Gather - 58BE8172F4A2	001	Test
Unknown 2021-09-03 14_18_33.csv		4	06ae2104-518f-4b92-b088-d789df91b065		0.44	001	Data Gather - 58BE8172F4A2	001	Test
Off 2021-09-03 14_07_34.csv		1	0a3a986-037e-4e60-bf25-9894fc296f80		0.45	001	Data Gather - 58BE8172F4A2	001	Test
Vibration 2021-09-03 14_17_31.csv		2	20c348b-91c0-4445-844b-7368f9102058		0.45	001	Data Gather - 58BE8172F4A2	001	Test
Off 2021-09-03 14_08_43.csv		1	27f0ee44-e0f0-4303-80b-bbc46cc90e5		0.45	001	Data Gather - 58BE8172F4A2	001	Test
Flow Blocked 2021-09-03 14_16_36.csv		1	6f02036e-e334-4601-8782-a61e5212c059		0.36	001	Data Gather - 58BE8172F4A2	001	Test
Off 2021-09-03 13_53_24.csv		1	430c848c-c48f-4743-b0ea-68594908d21		0.33	001	Data Gather - 58BE8172F4A2	001	Test

The model will be compiled in the SensiML Cloud and the output of the model will be returned. The graph shows the segment start and segment classified for all the detected events.

CAPTURE NAME	ACCURACY	TOTAL EVENT COUNT	CAPTURE UID	RESULTS	SIZE (MB)	DATA COLLECTOR	DEVICE	FANID	SET
On 2021-09-03 14_09_42.csv	99.3	1	02957f2f-a958-4445-9f43-60288c79a367	RESULTS	0.44	001	Data Gather - 58BE8172F4A2	001	Test
Unknown 2021-09-03 14_18_33.csv	76.7	4	06ae2104-518f-4b92-b088-d789df91b065	RESULTS	0.44	001	Data Gather - 58BE8172F4A2	001	Test
Off 2021-09-03 14_07_34.csv	97.1	1	0a3a986-037e-4e60-bf25-9894fc296f80	RESULTS	0.45	001	Data Gather - 58BE8172F4A2	001	Test
Vibration 2021-09-03 14_17_31.csv	92.3	2	20c348b-91c0-4445-844b-7368f9102058	RESULTS	0.45	001	Data Gather - 58BE8172F4A2	001	Test
Off 2021-09-03 14_08_43.csv	99.3	1	27f0ee44-e0f0-4303-80b-bbc46cc90e5	RESULTS	0.45	001	Data Gather - 58BE8172F4A2	001	Test
Flow Blocked 2021-09-03 14_16_36.csv	83.3	1	6f02036e-e334-4601-8782-a61e5212c059	RESULTS	0.36	001	Data Gather - 58BE8172F4A2	001	Test
Off 2021-09-03 13_53_24.csv	96	1	430c848c-c48f-4743-b0ea-68594908d21	RESULTS	0.33	001	Data Gather - 58BE8172F4A2	001	Test

You can click on the **Results** button in the table to see more detailed information for how each of the captures were classified by the selected model.



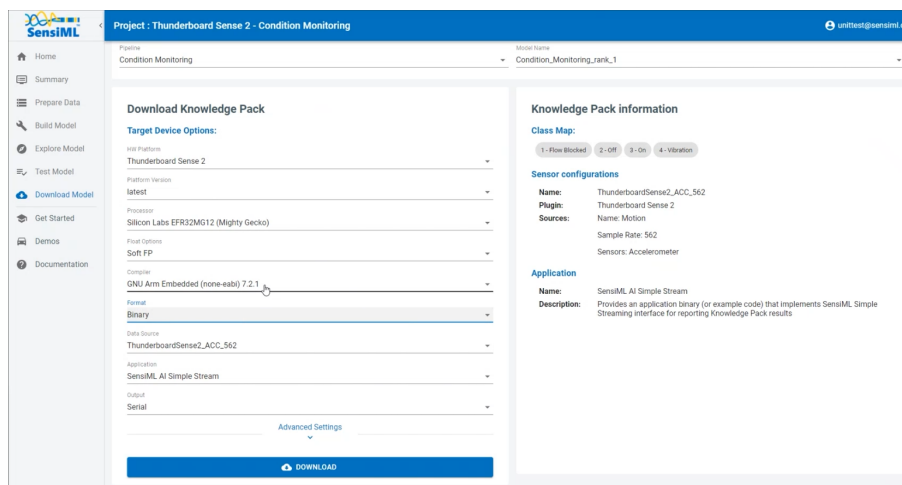
Deploying Knowledge Pack Firmware

Downloading the Knowledge Pack

Now that we have validated our model it is time for a live test. To build the firmware for the **Thunderboard Sense 2** go to the Download Model tab of the Analytic Studio.

If you are using the community edition of SensiML you can download the firmware binary or library for your device. For the full source code, see our [plans](#).

1. Go to the Download Model tab of the Analytic Studio
2. Select the pipeline and model you want to download
3. Select the HW platform **Thunderboard Sense 2**
4. Click Download and the model will be compiled and downloaded to your computer.
- 5.



6. Unzip the downloaded file and copy the binary to the device.
7. It will reflash and be boot running the new model firmware.

Viewing Results

To see the results of the model we will look at the output over the USB serial connection. The model's output results as 1,2,3..etc. In order to get the corresponding label value, you will need to map it using the model's class map. You can see the model class map in the knowledge pack information tab on the download model page.

Knowledge Pack information

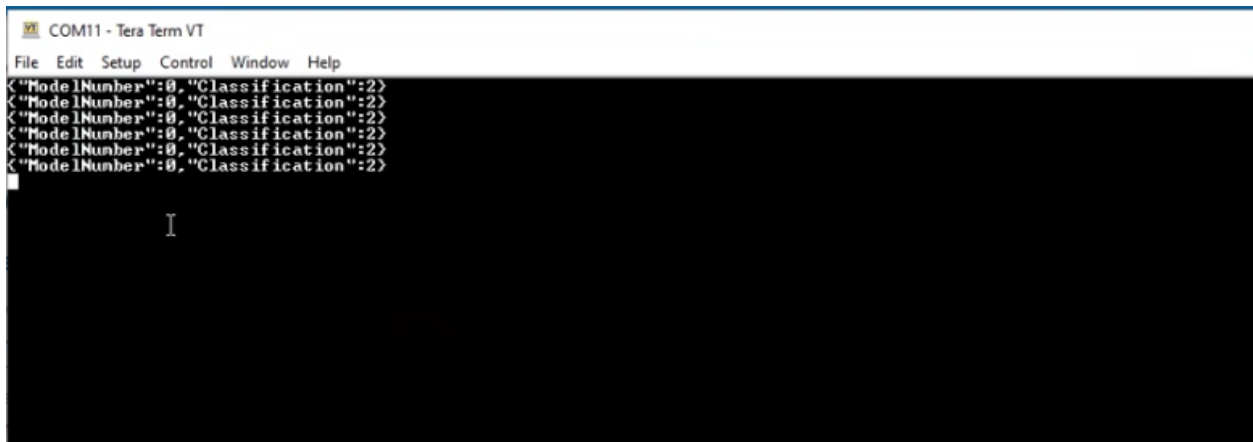
Class Map:

1 - Flow Blocked 2 - Off 3 - On 4 - Vibration

Sensor configurations

Name: ThunderboardSense2_ACC_562
Plugin: Thunderboard Sense 2
Sources: Name: Motion
Sample Rate: 562
Sensors: Accelerometer

If you open your favorite [teraterm](#) application for reading USB serial COM port, you will see the classifications being printed out.



```
COM11 - Tera Term VT
File Edit Setup Control Window Help
<"Mode lNumber":0,"Classification":2}>
<"Mode lNumber":0,"Classification":2}>
<"Mode lNumber":0,"Classification":2}>
<"Mode lNumber":0,"Classification":2}>
<"Mode lNumber":0,"Classification":2}>
<"Mode lNumber":0,"Classification":2}>
```

Viewing results in the SensiML Open Gateway

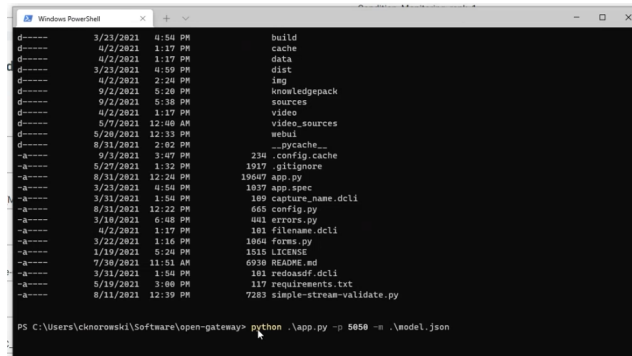
Another option is to use the SensiML [open gateway](#). When the model is downloaded, it also comes with a model.json file that contains all of this information. To install the open gateway

1. git clone <https://github.com/sensiml/open-gateway.git>
2. cd open-gateway
3. pip3 install -r requirements.txt

To start the application with the class-map included.

1. Copy the model.json file downloaded with the Knowledge Pack to the file to the home directory in the open gateway.
2. Start the application with the command

```
python3 app.py -m model.json
```

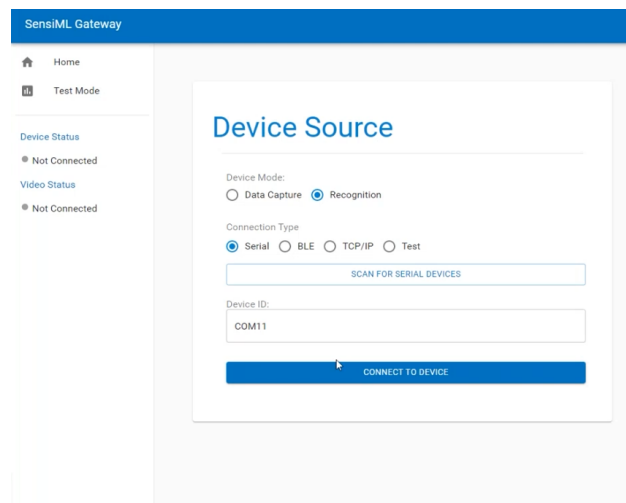


```
Windows PowerShell
d----- 3/23/2021 4:54 PM build
d----- 4/2/2021 1:17 PM cache
d----- 3/23/2021 4:59 PM data
d----- 4/2/2021 2:24 PM dist
d----- 9/2/2021 5:20 PM knowledgepack
d----- 9/2/2021 5:38 PM sources
d----- 4/2/2021 1:17 PM video
d----- 8/7/2021 12:40 AM video_sources
d----- 5/28/2021 12:33 PM webui
d----- 8/31/2021 2:42 PM _pycache_
-a----- 9/3/2021 3:47 PM 234 .config.cache
-a----- 5/27/2021 1:32 PM 1017 .gitignore
-a----- 8/31/2021 12:24 PM 19647 app.py
-a----- 3/23/2021 4:54 PM 1037 app.spec
-a----- 3/31/2021 1:54 PM 189 capture_name.dcll
-a----- 8/31/2021 12:22 PM 465 config.py
-a----- 3/18/2021 6:48 PM 441 errors.py
-a----- 4/2/2021 1:17 PM 181 filename.dcll
-a----- 3/22/2021 1:16 PM 1864 format.py
-a----- 1/19/2021 5:24 PM 1515 LICENSE
-a----- 7/30/2021 11:51 AM 6930 README.md
-a----- 3/31/2021 1:54 PM 101 redosass.dcll
-a----- 5/19/2021 3:40 PM 117 requirements.txt
-a----- 8/11/2021 12:39 PM 7283 simple-stream-validate.py

PS C:\Users\c1n0r0wsk1\Software\open-gateway> python .\app.py -p 5858 -m .\model.json
```

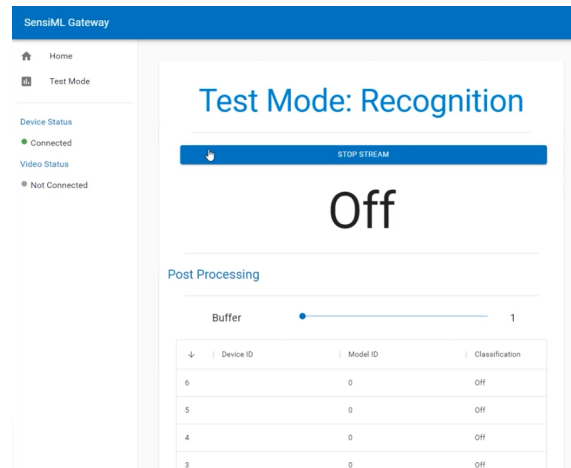
This will launch the open gateway application. You should see your browser open the UI. In the browser UI.

1. Select the **Recognition** radio button under Device Mode
2. Select **the Serial** radio button under Connection Type
3. Click the **Scan For Serial Devices** button
4. Select the COM port that your device is connected to
5. Click the **Connect To Device** button



Now the device is connected:

1. Switch to the **Test Mode** tab
2. Click the **Start Stream** button
3. Slide the buffer to 4, this will take a majority vote of the 4 previous classifications
4. The classification results will be now be displayed in the UI



Summary

We hope you enjoyed this tutorial using the SensiML Analytics Toolkit. In this tutorial we have covered how to:

1. Collect and annotate a high-quality data set using the **Thunderboard Sense 2**
2. Build a query as input to your model.
3. Use SensiML AutoML to search for an edge-optimized model.
4. Use the SensiML Analytic Studio to test the model offline.
5. Download Knowledge Pack firmware and test the model on the Thunderboard Sense 2

For more information about SensiML visit our website. To work with us to enable you to build your application, get in [touch with us](#).

SensiML

SensiML enables developers to quickly and easily create machine learning models that run locally on resource-constrained edge devices. SensiML SaaS provides an end-to-end solution from collecting and annotating a high-quality sensor time-series data set, to analyzing and building data models using AutoML, and finally generating firmware that will run on your target device