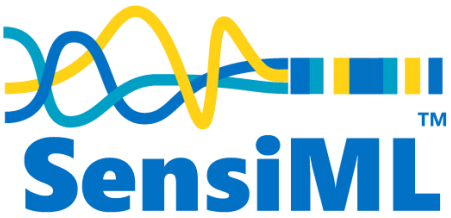


Free Trial - Quick Start Guide

Building Your First App – Version 1.07

Contents

1.0 Build Your First App 4
Software Overview: What's included with SensiML 4
Analytics Studio Tutorials 4
2.0 Quick Start Projects 5
Slide Demo Project 5
3.0 Fundamentals 6
4.0 Getting Started with the Data Capture Lab 7
Projects 8
Opening the Quick Start Projects 8
5.0 Exploring Your Project Data 9
Viewing your capture files 9
Segments 10
Sessions 10
Adding Segments to a Session 11
6.0 Data Capture Lab - More Features 13
Capture Mode 13
Custom Data Sets 13
Import Files from External Sources 13
Auto-Session Algorithms 14
Media Player 14
7.0 Getting Started with Analytics Studio 15
Jupyter Notebooks 15
Create a model: SensiML Dashboard and Advanced Model Building 16
SensiML Dashboard 16
Advanced Model Building 16
8.0 Building a Model with SensiML Dashboard 17
Loading the Dashboard Widget 17
Data Exploration: Setting up your query 18
Model Building: Setting Up Your AutoSense Pipeline 18
Seeds 19
Exploring Model Results 20

Test model 21
Confusion Matrix 22
Feature Summary 22
Model Summary 22
Model Rehydration 22
Generating a Knowledge Pack 23
HW Platform 23
Format 23
Classification Output 24
Debug Option 24
9.0 Validating Your Results 25
10.0 Where to go next? 26
Data Depot 26
Upgrade Your Subscription 26
Advanced Model Building 26

1.0 Build Your First App

Prerequisites: This guide assumes you have:

- Registered a team or account with SensiML (Click here if you do not have an account)
- Installed the SensiML **Data Capture Lab** and **Analytics Studio** (link):

https://sensiml.com/download/

The goal of this guide is to provide a step by step guide on creating your first basic sensor application using the SensiML Toolkit. You will learn how to create a sensor application and have it running on a real sensor device.

Software Overview: What's included with SensiML

The SensiML Toolkit is a software suite that makes it easy for a software developer to take simple sensors and turn them into advanced event detectors.

The toolkit comes with three applications:

- **Data Capture Lab (Windows)** – The DCL is a tool that helps you capture, organize, and label raw data from the sensor and transform it into the events you want to detect.
  Analytics Studio – The Analytics Studio is a python-based tool for filtering and optimizing your labeled sensor data through machine learning algorithms. It generates a device optimized SensiML Knowledge Pack (event detection algorithm) ready to be flashed onto your sensor of choice.
- **SensiML TestApp** – The SensiML TestApp allows you to see the real-time event classifications from the SensiML Knowledge Pack running on your sensor over Bluetooth-LE. It comes in a Windows or Android version that can be downloaded from the Windows/Google Play stores.

Analytics Studio Tutorials

Once you have finished this Quick Start Guide, you can find more advanced tutorials on how to use the Analytics Studio at the link below:

Analytics Studio Tutorials

2.0 Quick Start Projects

The free version of the SensiML Toolkit comes with several data sets that allow you to explore the features in the SensiML Toolkit. In this guide, we are going to reference a specific example called the **Slide Demo**.

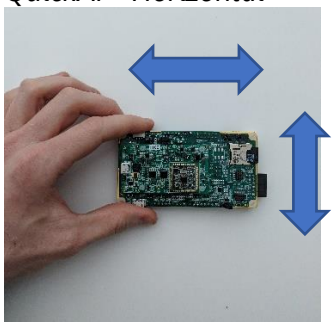
Slide Demo Project

The **Slide Demo** has three continuous states. This demo is used to simulate a motor status. To make this more accessible for the demo we hold the sensor in our hand and make the actions, but in the real world you would attach the sensor to a motor.

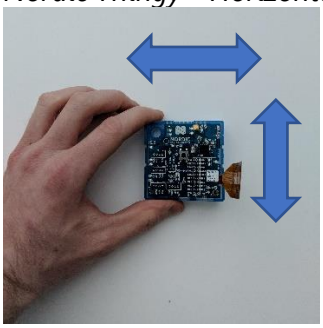
Events of interest:

- **Stationary** – Board is resting on a desk surface
- **Horizontal** – Board is slid back and forth (**left to right** or **forward to back**) on a flat surface in a repetitive rhythmic motion to simulate a part moving back and forth

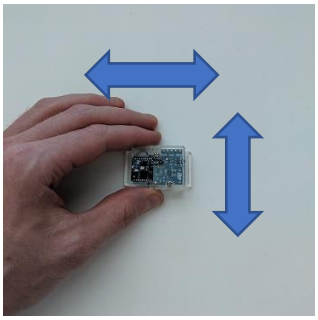
QuickAI - Horizontal



Nordic Thingy - Horizontal

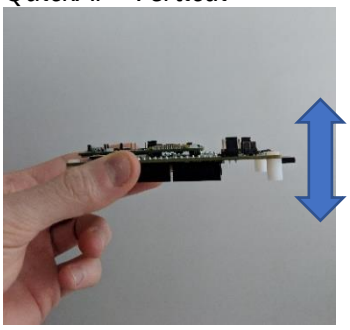


SensorTile - Horizontal

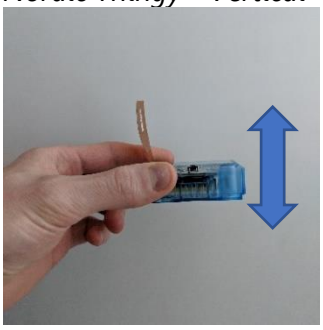


- **Vertical** – Board is lifted in an **up and down** motion in the air in repetitive rhythmic motion to simulate a part being lifted up and down

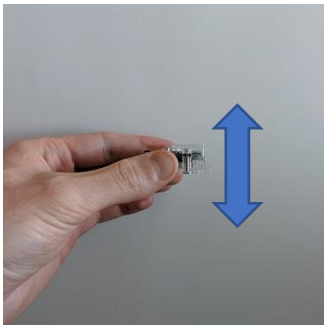
QuickAI - Vertical



Nordic Thingy - Vertical

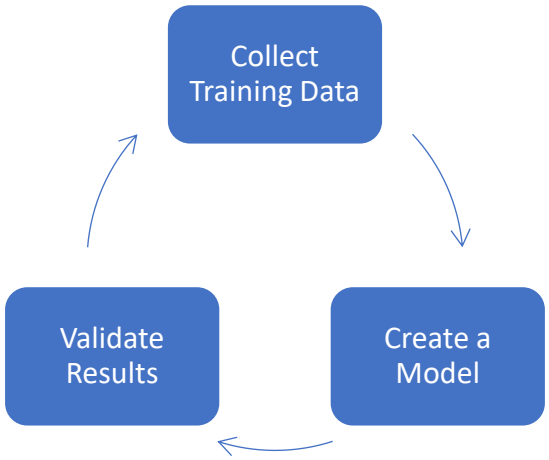


SensorTile - Vertical



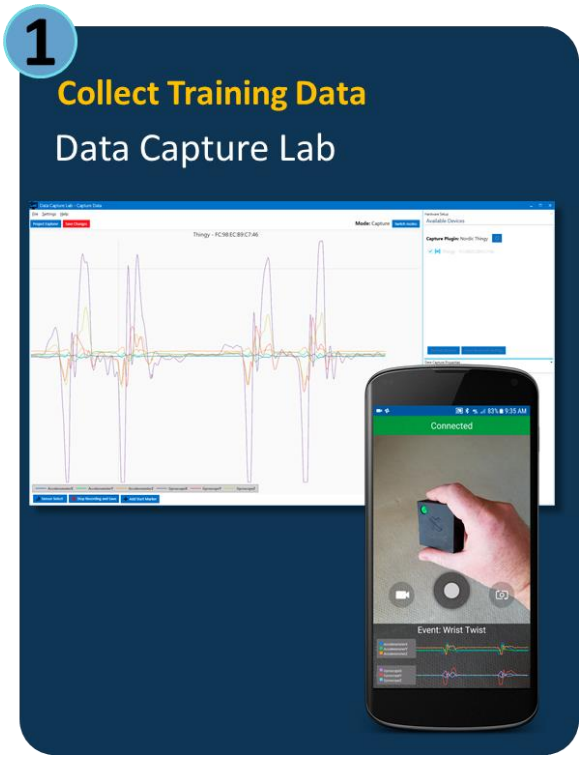
3.0 Fundamentals

Before you ever open the SensiML Toolkit, let's take a moment to think about the problem we are trying to solve. We need to build a smart algorithm that fits on a low-powered device. The SensiML Toolkit takes a 'train by example' approach to do this. This can be broken down into 3 steps.

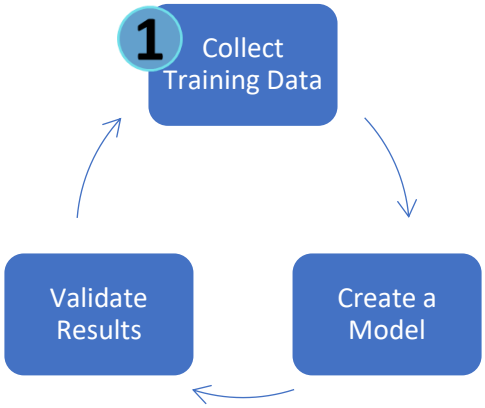


4.0 Getting Started with the Data Capture Lab

The first step of creating a sensor application is going to be collecting and labeling raw sensor data into events of interest through the **Data Capture Lab**.



PC and Android Version

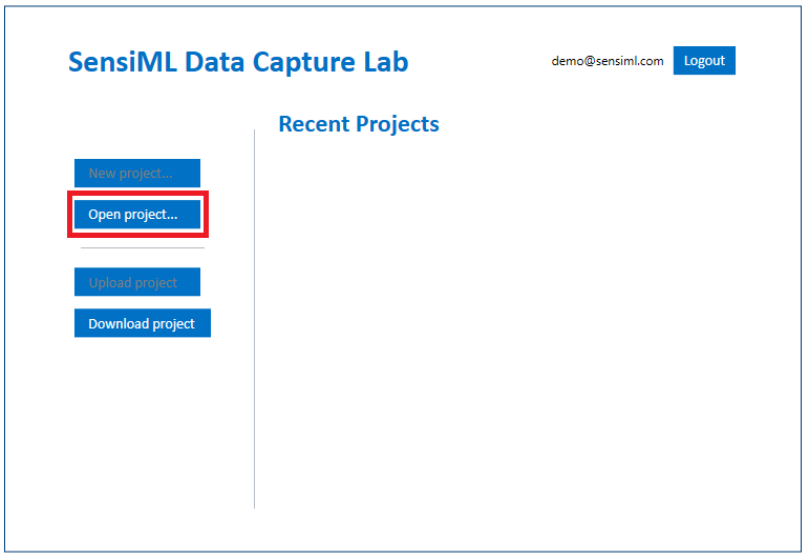


Projects

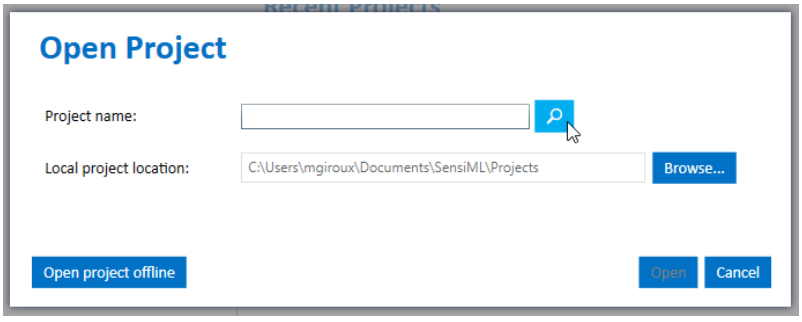
A project contains all your collected sensor data and algorithms. Your sensor data is saved to your local computer and synced with the cloud. Only your team has access to your data. This allows you to have multiple people collecting/labeling data on a project at the same time.

Opening the Quick Start Projects

1. Open the Data Capture Lab and **Log in** to your account
2. Click **Open project…**



3. Click the **search icon** and you will see the projects that come with the free version of the SensiML Toolkit



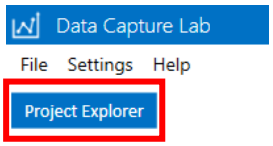
Optional: You can change the local project directory location where the project will be stored on your computer

4. Open one of the versions of the Slide Demo *(QuickAI, Nordic Thingy, SensorTile)*

5.0 Exploring Your Project Data

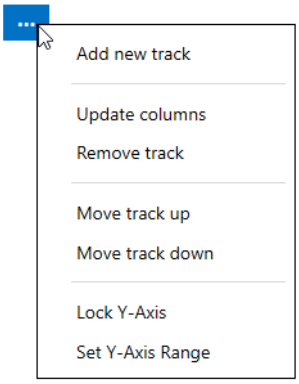
Viewing your capture files

1. To open the sensor data files in a project, use the **Project Explorer**. This can be found in the upper left-hand corner of the DCL.

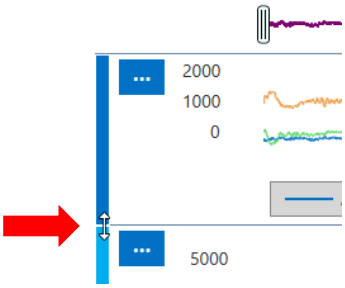


2. **Double-Click** on a file name to open the file
3. Depending on the project you have loaded, the file will open and look similar to this

4. You will notice the DCL has split the data into two tracks. These tracks can be customized to display as many sensor columns as you need. *(Optional)* Click on the track menu options to add new tracks, update columns, or remove tracks. This is a powerful feature that allows you to visualize your data easier.



5. *(Optional)* Click and drag the area between tracks to adjust a track height



Segments

Segments are how we label an **event** in the DCL. Open any of the provided files and you will see one or more segments in the sensor data graph depending on what project/file you opened. The pair of blue and red lines defines where our event is located and is called a **segment**. See screenshot below:

Sessions

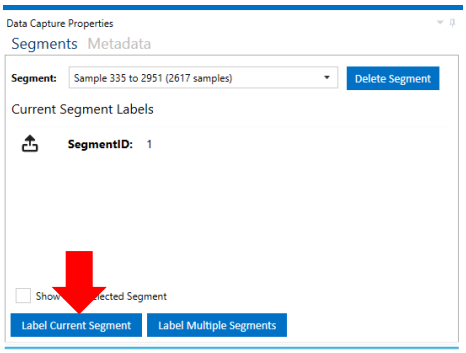
A session separates your labeled events (segments) into a group. Every segment is inside a session. Sessions allow you to work on multiple branches of the same dataset. Sessions can be useful when collaborating with a team or if you want to experiment with different segmentation methods on the same data set. Note - The free trial of the SensiML Toolkit does not allow you to create new sessions.

Adding Segments to a Session

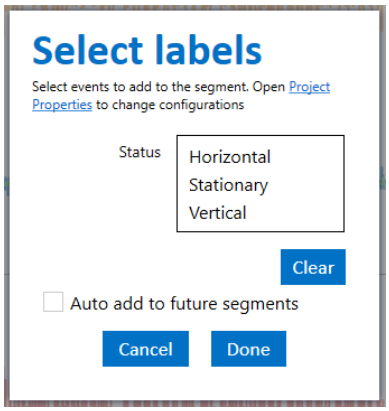
1. Select a **manual** session to work in. In our quick start projects, we have (**M) My Training Session** already created

2. Open a file
3. Next, identify your event of interest in the graph. To do this, move your mouse onto the graph of your sensor data and **right-click + drag** your mouse over the area that you want to label as an event. This will place a new **segment** in the file.

4. After creating the segment, you label this segment by clicking **Label Current Segment**.



5. Next select the label and click **Done**

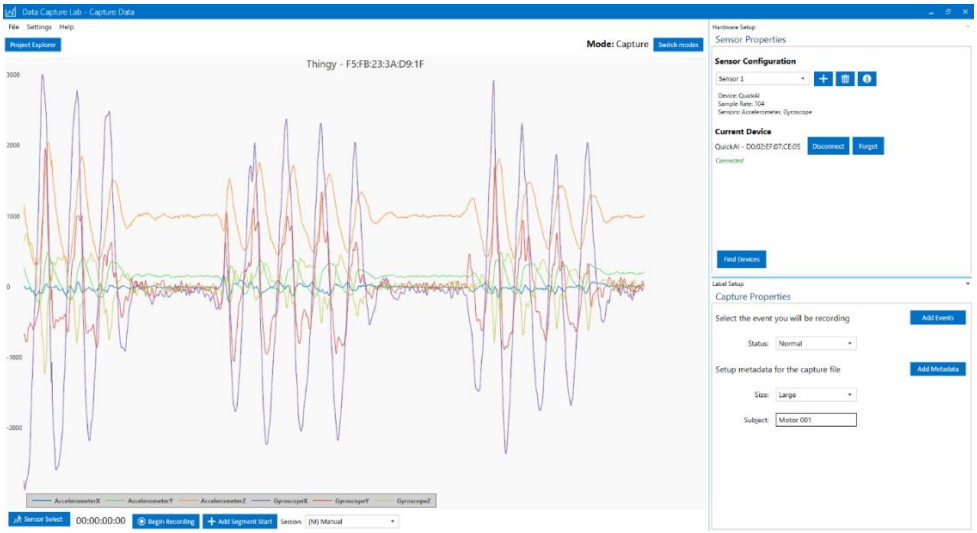


6.0 Data Capture Lab - More Features

The free version of the SensiML Toolkit comes with a few simple data sets that allow you to check out some of the features that make it so useful. Upgrading to a paid subscription enables the ability to create/modify your own data sets and unlocks several other features listed below:

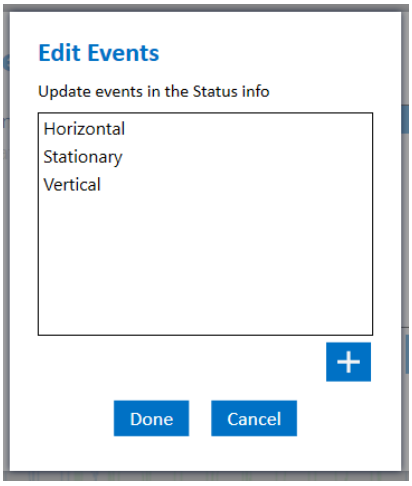
Capture Mode

Capture live sensor data in real time from SensiML supported devices



Custom Data Sets

The Slide Demo is a simple example to get you started. When you create your own project, you can setup your own events/metadata through the **Project Properties** screen



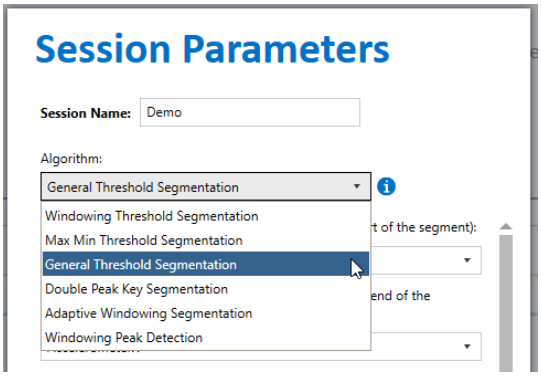
Import Files from External Sources

Have you already collected your own data sets? The Data Capture Lab can import it into our file format! See more about the file format at the link:

Importing External Sensor Data into the Data Capture Lab

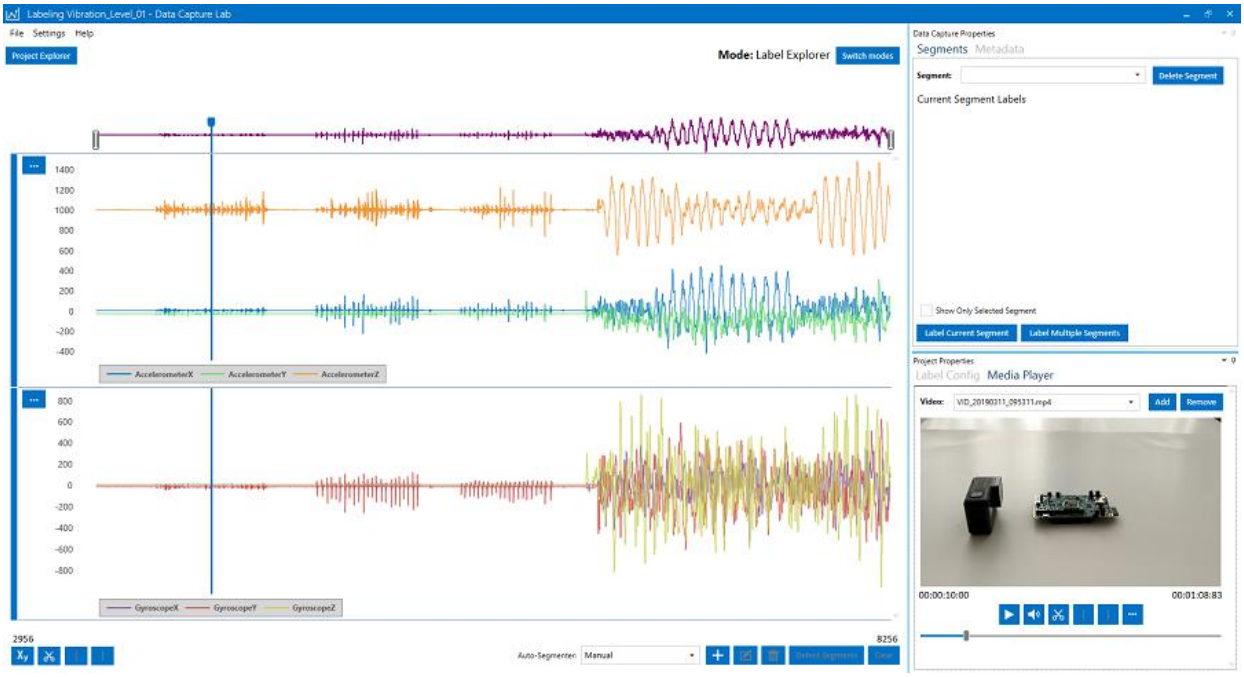
Auto-Session Algorithms

Use the real device-side algorithms on your data set to preview the results before deploying to your hardware



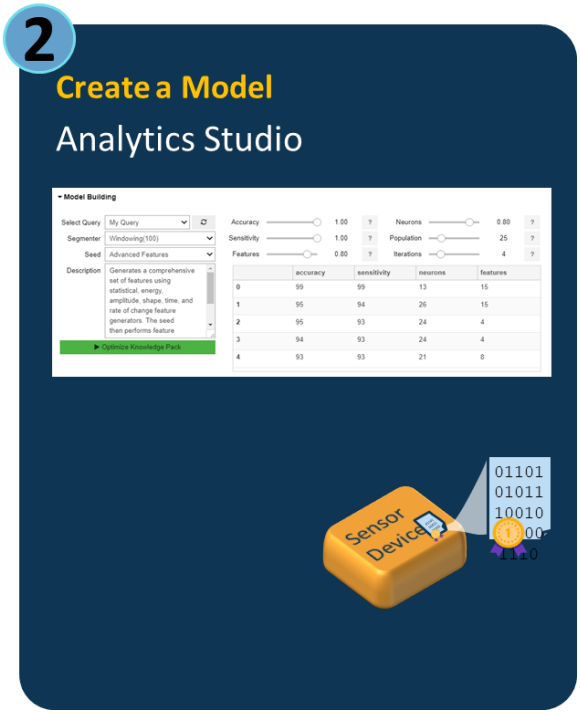
Media Player

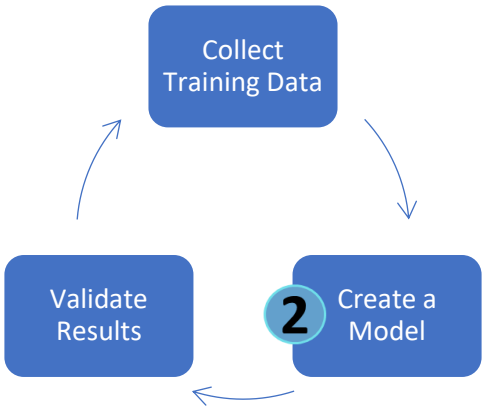
The Data Capture Lab can play WAV, MP4, and MOV files with your sensor data to make identifying your events of interest easier



7.0 Getting Started with Analytics Studio

Next it is time to use the **Analytics Studio** to build your machine learning algorithm for detecting events.





The Analytics Studio is a python-based tool for filtering and optimizing your labeled sensor data through machine learning algorithms. It generates a device optimized SensiML Knowledge Pack (event detection algorithm) ready to be flashed onto your device of choice. The most powerful part of the Knowledge Pack is that it will be detecting events on the low memory sensor without ever requiring a connection to the cloud.

Jupyter Notebooks

The Analytics Studio is a tool based on jupyter notebooks. If you have not used jupyter notebooks before the following keyboard shortcuts will be useful.

- **Run a cell** - Shift + Enter
- **Auto-complete** - Press tab at any time while typing a function/command and the Analytics Studio will give you all available options

Create a model: SensiML Dashboard and Advanced Model Building

The Analytics Studio can give you complex machine learning tools to build a model that detects your events. We created a **Dashboard** widget that abstracts the complexities of machine learning algorithms and translates them to a user-friendly interface. If you want to dive deeper into the underlying algorithms that the Dashboard uses, we also give you the tools to create and modify them on your own.

SensiML Dashboard

The SensiML **Dashboard** widget uses advanced machine learning to build a model that gives you control of the features you want in your device. For example, if you build an algorithm that detects your events with 100% accuracy, the device might cost more money. But by tweaking parameters through our **Dashboard** widget you might find you can get an algorithm that cuts your device cost in half, while still getting 98% accuracy. We make this easy by emulating the cost of putting an algorithm on a device and being able to tell you if it will fit. This is a powerful concept that can save you a lot of time and money.

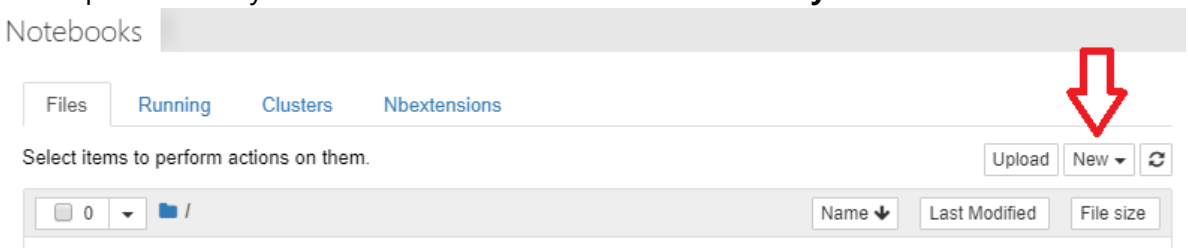
Advanced Model Building

If you have experience with machine learning you might want to dive deeper into the functions that were used to generate a model in the **Dashboard** or skip the **Dashboard** all together and customize your own functions in the model. We give you the tools for this! See Advanced Model Building for how to get started.

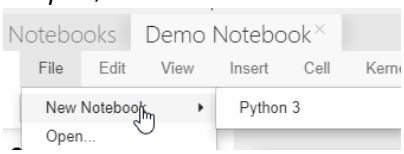
8.0 Building a Model with SensiML Dashboard

Loading the Dashboard Widget

1. Open the Analytics Studio and click **New** -> **Notebook: Python 3**.



(If you already have a notebook open, click File -> New Notebook -> Python 3)



2. This should create an Untitled.ipynb notebook and open it for you. *(If it does not, click the Refresh icon and open it from the list)*

3. After you open your notebook, enter the following command into the cell and click **Run (SHIFT + Enter)**

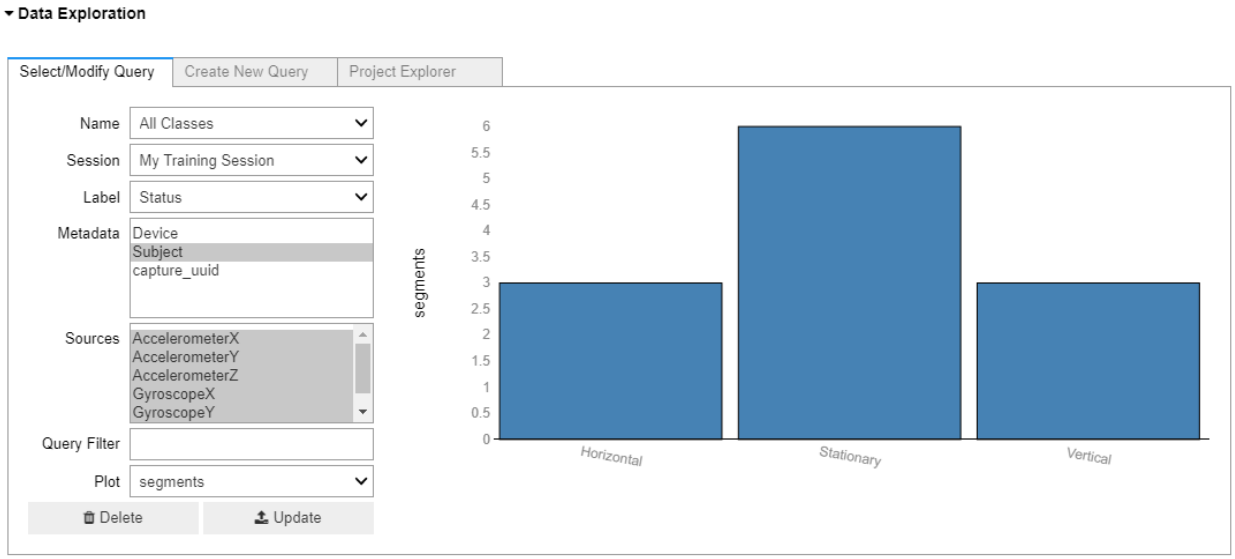
```
from sensiml import *
d = DashBoard()
```

4. This loads the Dashboard widget. **Login** and select the **Slide Demo** project
5. Enter the name **My Pipeline** for your pipeline and click **Add**

Data Exploration: Setting up your query

The **query** step is used to select your sensor data from your project. If you need to filter out certain parts of your sensor data, you would do it here. In the free version of the SensiML Toolkit you cannot create new queries. But this is a great tool for getting more insight to your data.

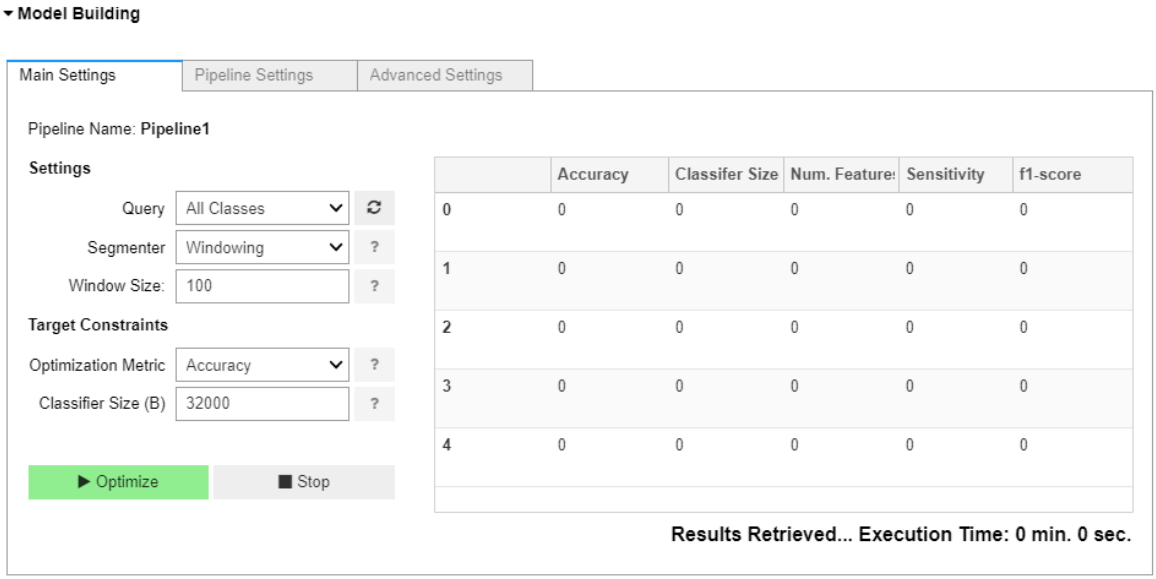
1. Open the **Data Exploration** tab in the Dashboard widget
2. Select the pre-made query: **All Classes**
3. The graph will look similar to this:



The X-Axis shows the events in the query and the Y-Axis shows how many events you selected

Model Building: Setting Up Your AutoSense Pipeline

The Model Building step is where the machine learning happens. Let's look at a screenshot and dive a little deeper.



1. Click the **refresh** Icon to populate the query from above and select **All Classes** from the query list.
2. Select your **Segmenter**
    a. Slide Demo: **Windowing.** Set size to **100**. The Windowing segmenter works well with continuous events. *100 refers to the window size in samples, so by picking Windowing(100) on 100hz data we have a 1 second window size, meaning every 1 second you will get a new classification.*

Seeds

The AutoSense Pipeline starts with a seed, you can think of the seed like a template for how the Model Building widget should build an algorithm.

3. Select the **Pipeline Setttings** tab and change the **Seed** to **Advanced Features**

You might be wondering when you should choose a given seed. Listed below are some examples

Basic Features:

- You are wondering where to start
- You want execution to be as quick as possible
- You want simple, easy-to-interpret features

Advanced Features:

- You tried "Basic Features" and didn't get a good model
- You don't mind if execution takes a while
- You want the best possible features, even if they are complex

Downsampled Features:

- You are creating a gesture recognition application

Histogram Features:

- You are creating a motor vibration application

Custom Seed:

- You tried the other seeds and didn't get a good model
- You want to build your own pipeline and use the genetic algorithm to find the best number of features, best number of neurons, and other model-related parameters

No Feature Generation:

- You do not want to generate any features, only test the ones you have made offline (Note: resulting knowledgepacks will not have a feature extraction algorithm, so will not operate on a device; intended for testing only)

4. Leave the other options as the default values
5. Click **Optimize** and the Analytics Studio will automatically build you a model to detect your events
6. You'll notice that after clicking **Optimize** a status will appear below the dashboard widget that tells you when the pipeline is complete. Depending on the amount of sensor data, the seed, and parameters you selected this could take some time to build an algorithm.
7. Once the pipeline is complete it will display 5 models on the right side of the Model Building widget. We provide an estimate of the classifier size in bytes and the number of features. In general, the more features a model has, the larger the latency required for classification. You can use this information to select a model that supports your device's resources while providing the level of accuracy your application needs. Keeping in mind that typically, there is an accuracy vs resource usage trade off, where the more resources you allocate to modeling, the higher accuracy of a model that can be built.

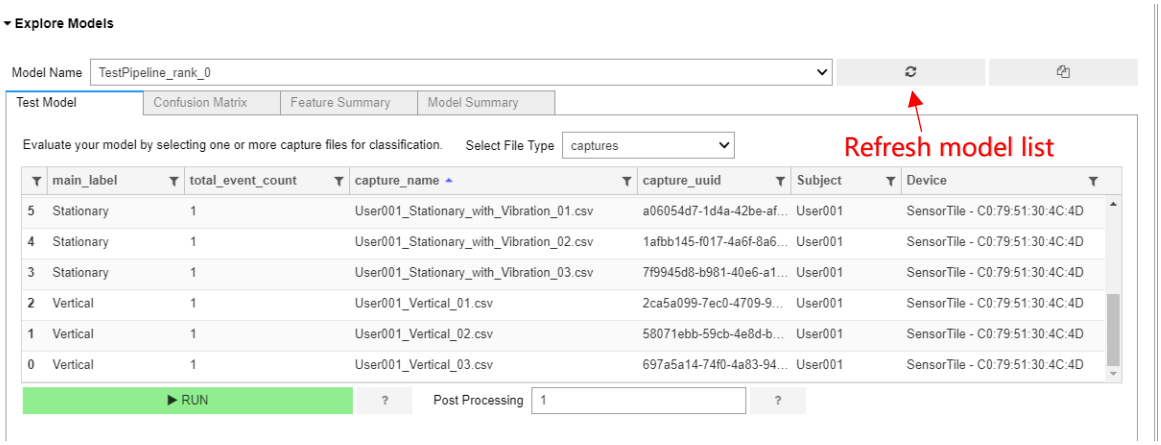
	Accuracy	Classifier Size (B)	Num. Features	Sensitivity	f1-score
0	100	980	3	100	100
1	100	1620	3	100	100
2	100	1833	3	100	100
3	100	1835	3	100	100
4	100	2123	3	100	100

Automation Pipeline Completed.

Exploring Model Results

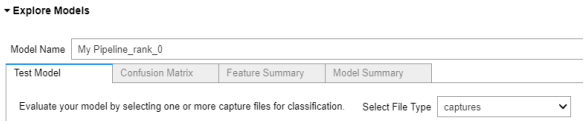
In the **Explore Models Widget** you can get more information about the models that were generated. In the **Test Model** tab, you can validate your model against captured data. The **Confusion Matrix** tab shows the averaged confusion matrix for the validation data. The **Feature Summary** tab shows which feature extractors and which sensors were used to feed into the model. The **Model Summary** tab describes the classifier, classifier parameters and training algorithm that was used to generate the final.

To refresh your model list after running a pipeline click the **refresh** button.

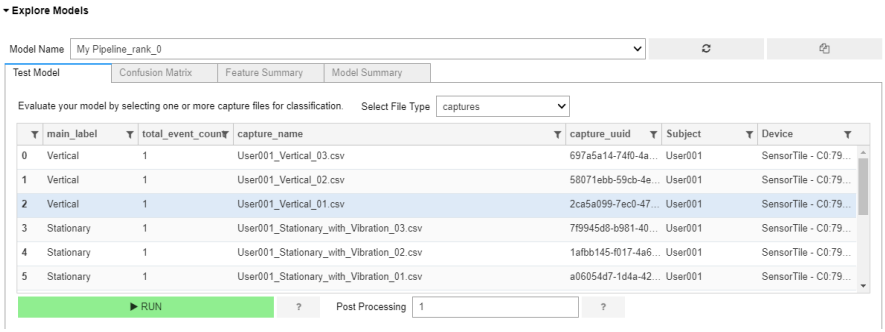


Test model

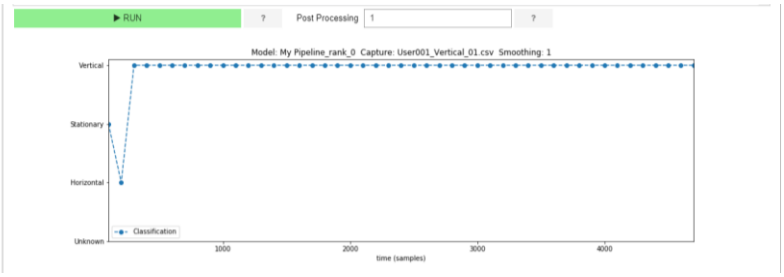
In the **Test Model** tab, select the file type you would like to validate against. Captures are files uploaded through the DCL. Data files are files uploaded through the Analytic Studio. For this project select captures as the file type.



Next, select one or more of the capture files to validate the model against. To validate the model, click the Run button, which will emulate the model and validate it against the test data.

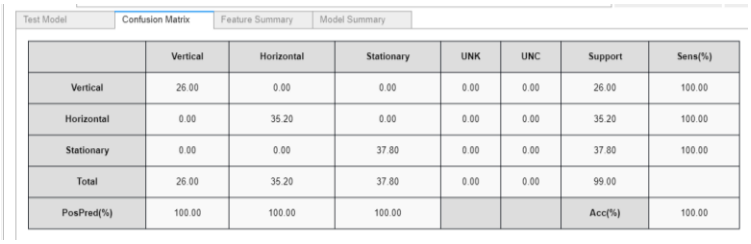


The results of the model validation are displayed below the widget. The Y-axis tells us the classification name and the X-axis tells us the sample number or time. Classifications are generated by the model at the interval you selected for windowing segmentation. These are represented by the circle marks on the graph. You can also apply post processing which will apply a majority vote algorithm at the specified width.



Confusion Matrix

The confusion matrix describes how well the model performed at recognizing each class. It also provides information about how the model misclassifies classes. The confusion matrix for models generated by the AutoSense pipeline are the created by averaging across the results of the validation data sets for each fold.

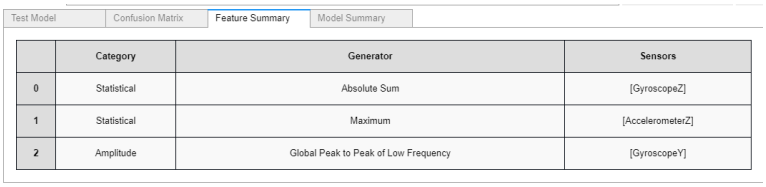


Test Model | Confusion Matrix | Feature Summary | Model Summary

	Vertical	Horizontal	Stationary	UNK	UNC	Support	Sens(%)
Vertical	26.00	0.00	0.00	0.00	0.00	26.00	100.00
Horizontal	0.00	35.20	0.00	0.00	0.00	35.20	100.00
Stationary	0.00	0.00	37.80	0.00	0.00	37.80	100.00
Total	26.00	35.20	37.80	0.00	0.00	99.00	
PosPred(%)	100.00	100.00	100.00			Acc(%)	100.00

Feature Summary

The **Feature Summary** tab contains information about the features that are used during the feature extraction step of the Knowledge Pack. This was a simple example project and only needed three feature extractors to generate a high accuracy model. You can see the **Category Type** of the feature generator in the first column, which describes the family type that a feature generator belongs too. The **Generator** column has the name of the feature extractor, which can be used to reference the feature generator when building custom pipelines. The **Sensors** column describes the sensors that were used as input into this particular feature extractor.

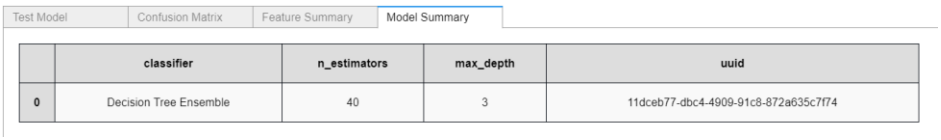


Test Model | Confusion Matrix | Feature Summary | Model Summary

	Category	Generator	Sensors
0	Statistical	Absolute Sum	[GyroscopeZ]
1	Statistical	Maximum	[AccelerometerZ]
2	Amplitude	Global Peak to Peak of Low Frequency	[GyroscopeY]

Model Summary

The **Model Summary** tab contains information about the classifier used in the model. The information here will be different depending on which classifier was used by the model. But in general, this will have the classifier name, the training algorithm used to train the model, along with any hyperparameters that were set for training. The final column is the **uuid** which is the unique identifier for this particular model.



Test Model | Confusion Matrix | Feature Summary | Model Summary

	classifier	n_estimators	max_depth	uuid
0	Decision Tree Ensemble	40	3	11dceb77-dbc4-4909-91c8-872a635c7f74

Model Rehydration

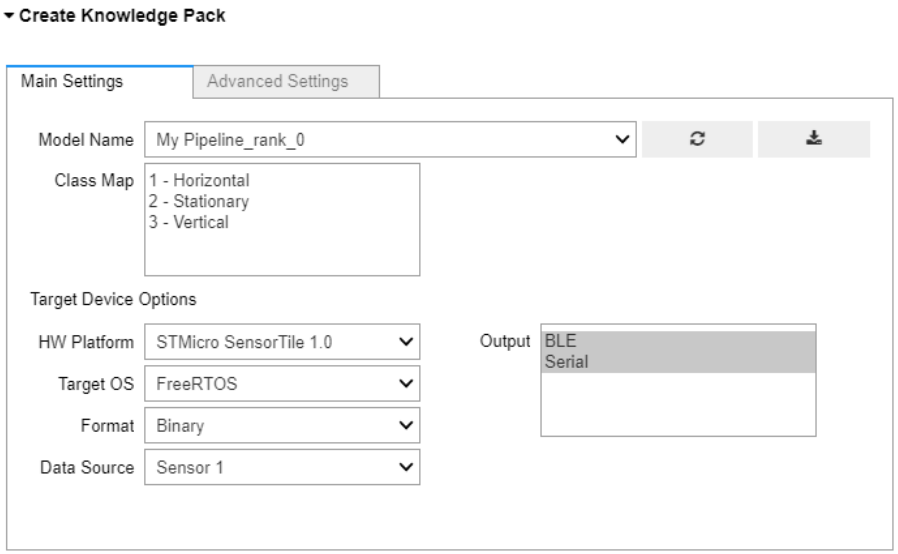
The **Model Rehydration** button allows you to "rehydrate" a model and see the code/algorithms that were used to generate this particular model.

When you click the **rehydrate button** it will print out the model in the log section of the DashBoard as well as copy it into your copy buffer. You can paste this into a new cell. From here you can modify the code and re-train your model hyperparameters tom improve the accuracy. For more information on the programmatic approach to model building see our tutorials for the Analytics Studio.

```
dsk.pipeline.reset()

dsk.pipeline.set_input_query('All Classes')

dsk.pipeline.add_transform('Windowing', params={'window_size':100,
                                'delta':100,
                                'train_delta':0})

dsk.pipeline.add_feature_generator([{'name':'Absolute Sum', 'params':{'columns':['GyroscopeZ']}},
                                    {'name':'Maximum', 'params':{'columns':['AccelerometerZ']}},
                                    {'name':'Global Peak to Peak of Low Frequency', 'params':{'smooth
                                'columns':['GyroscopeY']}}])
```

Generating a Knowledge Pack

A SensiML **Knowledge Pack** takes the event detection model you generated in the pipeline and transforms it into a binary or library file that can be run on your hardware device at the edge. Once the Knowledge Pack is on your hardware device, it starts outputting classification IDs that correspond to your events of interest. Let's take a look at the **Knowledge Pack widget** below.



The free version of the SensiML Toolkit does not allow you to download Knowledge Packs, but you can explore the various features in the widget.

HW Platform

SensiML Knowledge Packs are not locked to any specific hardware platform. This feature allows you to choose the platform where you will deploy your Knowledge Pack.

Format

We provide two formats for your knowledge pack; **Binary** and **Library**. The binary format will build a package that is ready to flash to your device and includes the predefined **output** methods listed in the widget (for our demo this will be **BLE** and **Serial**).

*More on **Library** format: If you have your own application and want to integrate a Knowledge Pack to build application specific non-AI features (LEDS, GPIO, ADC, DAC), then download the library format. This allows you to make calls directly to the Knowledge Pack API from your application. We will not be going into this during the quick start guide, but if you want to know more about library files click here.*

Classification Output

Output corresponds to how your events get broadcasted from the hardware device. BLE sends the events over Bluetooth Low Energy, LED will blink the LED light on the device a different color for each event, and serial allows you to plug your device into your PC over serial connection to send the events.

Debug Option

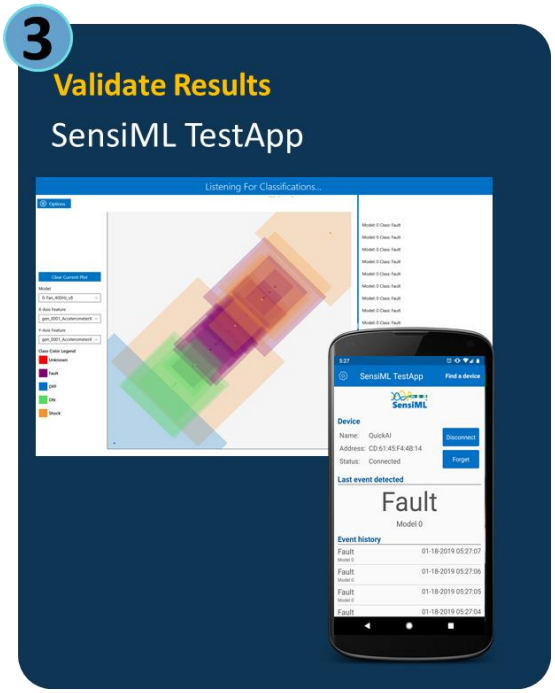
In the Advanced Settings tab, we provide a **Debug** option when generating a Knowledge Pack. When set to **True** the Knowledge Pack will log extra information like feature vectors, debug messages, and error messages over **serial** connection to help you debug the events of interest on a device.

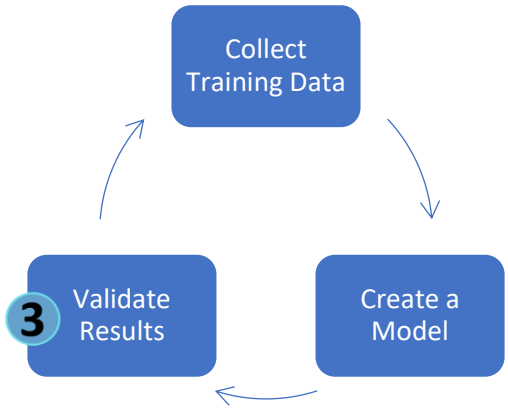
1. See the link below for how to view debug output:

   How to view debug output in a Knowledge Pack

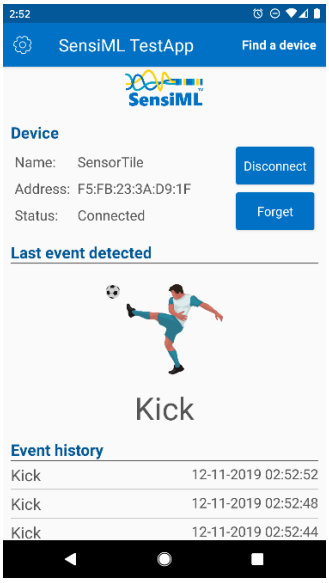
9.0 Validating Your Results

The easiest way to see the live event classification results of a Knowledge Pack running on your sensor is through the **SensiML TestApp**. The free trial version of the SensiML Toolkit does not have Knowledge Pack downloads enabled, so you will not be able to use the TestApp. However, once you upgrade your subscription you can use the TestApp to view classification results running on your device in real time. See screenshots of the TestApp below.

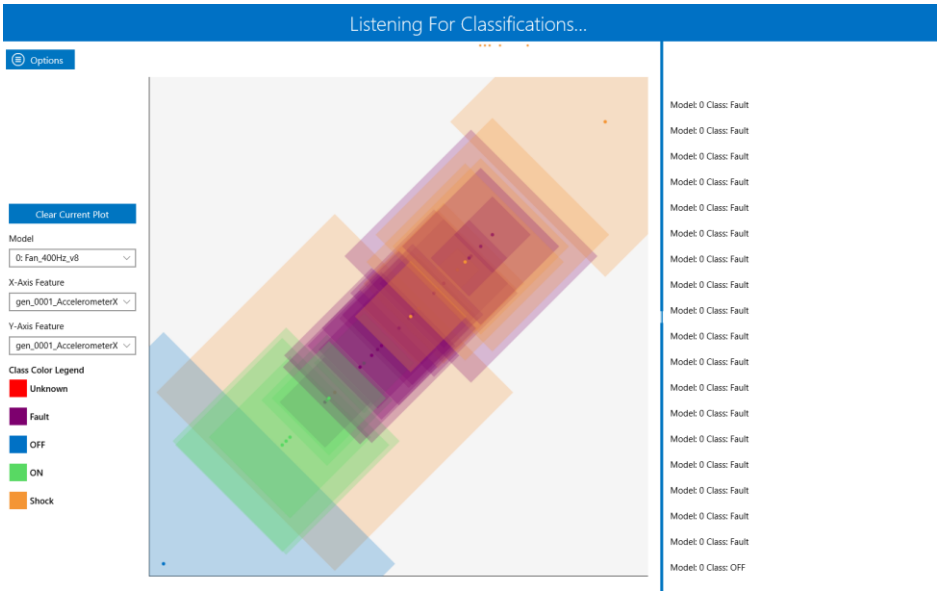




Android



Windows



10.0 Where to go next?

This guide walked you through the **Hello World** example of edge sensor detection. The purpose of the Free Trial is to show you the tools available to you in the SensiML Toolkit for when you want to build your own applications.

Data Depot

We provide a variety of more advanced datasets outside of this guide which you can find at the Data Depot link. Here you will find full curated datasets for industrial predictive maintenance, sports wearables, and more. These projects are meant to provide examples to help you get started with your own projects. We are constantly adding new datasets and welcome contributions.

Upgrade Your Subscription

The free trial version of the software is intended to give you a taste of what the SensiML Toolkit has to offer, but it has a lot of features disabled. See the products page for our subscription tiers and contact us to upgrade your subscription today.

Advanced Model Building

For data scientists we recommend looking into more of the advanced features within the SensiML Toolkit. In this guide we discussed the graphical DashBoard interface using AutoML, however, the Analytics Studio allows for precise control over the models that are generated. You can specify the features, transforms, training algorithm, and validation methods that go into your model. You can also specify different sampling, feature selection, and data augmentation techniques. See Advanced Model Building.