<u>Studio: 1.Arduino Nano Accelerometer</u>
In this studio, you will set up the Arduino and explore using the built-in accelerometer for data collection.

Install and open the Arduino IDE to have an interface with the Nano board. An empty sketch should open with the IDE on startup.

1. What are the two default functions that appear when opening a new sketch in the Arduino IDE?
2. In which function should you put code to run once? In which function should you put code that should run repeatedly?

- Connect your Arduino Nano to your local computer via USB.
- In the top menu bar of your local computer, navigate to Tools > Board > Board Manager and select Arduino Mbed OS Boards.
- Select install.
- Exit the Board Manager.
- Navigate to Tools > Board > Arduino > Arduino Mbed OS Nano Boards and then select the board connected to the computer.
- Navigate to Tools > Port and then select the port connected to the board you are using. The Nano should now be connected to the Arduino IDE.
- Navigate to File > Examples > 0.1.Basics > Blink

3. What does the line digitalWrite(LED_BUILTIN, HIGH); do in the code?
4. What code, in the setup() function, allows the previous line to run correctly?

- Click on the icon on the side of the Arduino IDE screen that looks like standing books. The words “Library Manager” should appear if you hover over the icon.
- In the Arduino Library Manager, search for the Arduino_LSM9DS1 library and install it.
- Navigate to File > Examples > Arduino_LSM9DS1 > SimpleAccelerometer

5. What sensor is used in this code and has the property readAcceleration. What does the abbreviation stand for?

- Click the right pointing arrow at the top of the screen to upload the script to the Arduino Nano board.

6. How many seconds did it take to compile and upload the script to the board?

- Click the magnifying glass in the top right corner to open the serial monitor. Click the peak icon to the left of the magnifying glass to open the serial plotter. Data values should appear in the serial monitor and in the serial plotter.

7. Approximately what values are the three axes?

- Move the Arduino Nano to see how the values change.
- Orient the Arduino in front of you with the long side of the Nano closest to you and the USB port on the left side.

8. Rotate the board upwards, toward yourself, so that the shield is now standing upright. What are the values of the three axes?
9. Rotate the board so that the USB port is facing straight down. What are the values of the three axes?
10. Take a screenshot of the serial plotter after moving the board in a quick motion. Attach the plot here.
11. Based on your answers to problem 7, 8, and 9, as well as any further exploration necessary, draw a picture showing the orientation of all three axes of the Arduino's built-in accelerometer. Include which side is positive and which is negative for each axis.

Studio: 2.Bluetooth with Arduino Nano

In this studio, you will learn how to connect two Arduino Nano boards via their built-in Bluetooth and how this enables data collection.

Please complete the Arduino Nano Accelerometer studio before working through this one.

- Open the Arduino IDE and navigate to the Library Manager.
- Search for and install the ArduinoBLE and Arduino_APDS9960 libraries.
- Create two new Arduino sketches, one for the central device that will be connected to the computer and one for the peripheral device that will be battery powered.
- Copy the central and peripheral code from the Arduino Bluetooth documentation into your scripts.

1. Compile and upload the central code to one of your Arduino Nanos. How many seconds did this process take?
2. What does the central device print in the serial monitor while it is waiting to connect to the peripheral? You may need to reset the Nano to see this message if you don't initially have the serial monitor open when uploading and compiling.

- Compile and upload the peripheral code on the second Nano. Once this process is done, disconnect the Nano from the USB and connect it to a battery. If you have the TinyML kit, the shield has a green adapter that allows cables to easily connect the battery to the GND and VIN pins.
- Reconnect the first Nano to the computer via USB. This should be the Nano that was programmed with the central script.

3. Open the serial monitor. What do you see? Did the two devices find each other? Record any observations here. Note that this is sometimes a challenging process since the Arduino Bluetooth library can be unreliable. Therefore, a successful connection might require being in an area away from other devices.

Once Bluetooth connection is established between the Nanos, it is important to get the data from the Arduino IDE to the local computer. This can be done using a Python script and the Python serial package.

- Analyze the following pseudocode for reading data from the Arduino serial communication. Note that implementing this requires having the Arudino serial monitor closed since one serial port cannot be used for two purposes at once.

```
import serial, csv
```

```
2 arduino = serial(port, baud)
3    if arduino.isOpen():
4         print(‘connected’)
5    else:
6         print(‘failed to connect’)
7    with open(‘data.csv’, ‘w’) as file:
8         wr = csv.writer(file)
9         wr.writerow([‘time’,’x’,’y’,’z’])
10        while(true):
11             reading = arduino.readline()
12             data = reading.split(‘,’)
13             wr.writerow(data)
```

4. What purpose does line 2 serve?
5. On which line does the code write the data to a CSV file?
6. What happens if the serial communication fails?
7. Based on your answer to question 3 (what the serial monitor prints when collecting data from the given Bluetooth scripts), what additional changes might be needed in order to enable using a Python script, similar to the pseudocode provided, to produce CSV files with usable accelerometer data?

Studio: 3.Designing the Hardware

In this studio, you will explore different ways to design the hardware for the peripheral, as well as tips for producing a quality product.

This project will require designing hardware to hold a functioning peripheral Arduino Nano on the user's arm. There are many considerations that go into designing hardware, including the dimensions of all included parts, any necessities for the individual parts (i.e. the battery for the Nano), and how the design will look and feel on the user's arm.

1. List at least 3 parts that are necessary for designing a mechanism to attach a peripheral Nano to the user's arm for data collection. Keep in mind small parts that might be helpful as well, including a switch for the battery.
2. Of the list of parts from question 1, and any other necessary parts you think of, what is most likely the limiting factor in the design? What part is contributing to this limitation?
3. List at least 3 design aspects that are important when considering that the hardware will be used on the user's arm.
4. The Arduino has built-in sensors, as we explored in previous studios. Because of this, what is important about the orientation of the board inside the hardware?

Look at the following component options for the final hardware design. Note that other similar parts could be used for the project.

- Adafruit perfboard
- MPD Battery holder
- Generic 9V battery
- Arduino Nano 33 BLE Sense
- Switch

5. What is the widest component of those in the list above?
6. What is the longest component of those in the list above?
7. What is the tallest component of those in the list above?
8. Draw a template design for the layout of a box that would hold all the components in the list. Try to draw to scale and label dimensions. Keep in mind the considerations from your answer to question 3. How could this box be made/bought?
9. On the internet, find jumper cables, wires, a button, and a strap that could be used inside a box with the dimensions in question 8.
10. What methods could be used to attach all of the listed parts together?
11. When soldering parts together, it is best to use male-female headers, such as the ones from Phoenix Enterprises or OCR via Amazon. Why might this be important?

Studio: 4.Data Collection

In this studio, you will explore various techniques for data collection, as well as the pros and cons for each.

There are many different ways to collect data. When considering options, it is best to put everything out on the table: ideas cost nothing. Below is a list of initial ideas for collecting data to track boxing punches.

- Press a button when the user punches.
- Press a button when the user starts and ends the punch.
- Collect raw accelerometer readings.
- Filter raw accelerometer readings.
- Take continuous pictures and measure the delta between sequential images, in addition to accelerometer data.
- Take the delta of continuous pictures with accelerometer data and a microphone.

1. For each of the above data collection ideas, what is the advantage to using this method for tracking/recognizing boxing punches?
2. What are some initial ideas for collecting data for a speech recognition device? Why would they be advantageous to use for recognition?

For implementation, you have to consider the expected development cost vs. possible gain.

3. For each of the above data collection ideas, what is the disadvantage to using this method for tracking/recognizing boxing punches?
4. Based on the analysis in (1) of the advantages and in (3) of the disadvantages of each collection idea, which method would you choose to collect data? Provide justification for your choice.

For the design lab, filtered accelerometer data and images will be used for data collection. However, you can explore other methods, as outlined above, for future projects and/or continued work on this lab.

After choosing the data collection method, a Design of Experiment (D.O.E.) is helpful to organize aspects such as how much and what kind of data to collect. Read the following article from the American Society of Quality.

5. Using the guide from the American Society of Quality and any other resources/examples you find, create a short D.O.E for the design lab, a boxing training app that recognizes when a punch occurs and what punch type was performed: uppercut, straight, hook.

Studio: 5.Data Cleaning and Preprocessing

In this studio, you will explore a few techniques for analyzing, preprocessing, and cleaning the data before using it to train and test models.

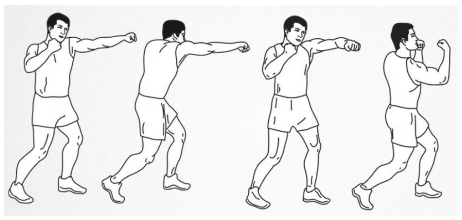
Raw data can come in many different forms. Sometimes, this causes issues when training models that lead to overfitting, underfitting, samples that don't match their classification, and other challenges.

Look at data_2_13-4_30.pdf, which is a PDF of CSV data collected via an Arduino and serial communication script.

1. Are all of the samples equally spaced based on the timestamp? Why might this be the case?
2. What are the first three timestamp delays between samples [in ms] using the first four samples?
3. Are all of the samples equally spaced based on the received time? Why might this be the case?
4. For the first three samples, how long did the transmission take [in ms]?

There is some noise at the beginning of this data file from when the Bluetooth and data collection were initializing. For further analysis in this studio, start from about 10 samples (rows) from the top of the file.

5. At what timestamp do you first notice significant change in the accelerations?
6. What are the rows before and after the significant change?
7. Which axis had the largest change in acceleration at this sample?



Straight Straight Hook Uppercut

8. This data was collected with the Nano on the user's right arm. The board was attached to the shield so the USB adapter was facing inward toward the user's body. Based on your answer to question 7, which punch type is this data most likely from? Punch types used for this lab are shown above.

9. Go through the file and highlight the sections of data samples that look like they most likely correspond to a punch, based on how we analyzed the data in questions 5-8. How many punches does it look like this sample collected?

Questions 1-9 walked through a process of manual analysis that was used, initially, to ensure the data was feasible and consisted of recognizable punches. For practical purposes, it is helpful to have computer software aid with this process when dealing with large amounts of data.

10. List at least 2 ways that you could perform this data analysis using a computer.

This analysis will split the total collected data into segments of punches and background movements for use in the model training. Once the data is in this format, further data preprocessing and cleaning can be done using filters and hyperparameters inside the machine learning platform. This is explored further in the Machine Learning with Edge Impulse Studio.

Studio: 6.Machine Learning with Edge Impulse

In this studio, you will explore Edge Impulse, a development platform for machine learning on edge devices, and how this tool can be used to design and deploy machine learning models for various purposes.

Before beginning the studio, explore the Edge Impulse page to learn more about their mission and product features.

1. What are the four emphasized sections on the Product page?
2. As it is described on the Product page, what is flexible data ingestion and how could this be helpful in designing a product?

On the Product page, scroll down to the button “Launch Studio” in teal or visit the site. Click “Let’s build your first model in 5 minutes!” This does not require an Edge Impulse account. Work through the 10 steps.

3. What keyword or sentence did you give the model?
4. What signal processing/learning blocks were added to the demo project automatically for you on step 5?
5. What was the learning rate for the default neural network in step 8?
6. What three data specifications are shown on the “Congratulations!” screen after testing the model?
7. What is the accuracy and loss of the initiation model training? Attach a picture of the final confusion matrix. What label was misclassified the largest percent of the time? What label was it misclassified as?

Navigate to the Data Acquisition tab to look at the collected and imported dataset for this model.

8. How much total data was assigned as training data for the model (in minutes and seconds)? How much of this training data is your chosen word or sentence (in minutes and seconds)?
9. What is the training/testing split that was automatically generated by Edge Impulse when adding data?
10. Add a filter to the collected data by clicking the icon. Filter for only samples with your word or sentence and select one of the samples. Attach a picture of the raw data preview that is displayed when selecting the sample.
11. How much total data was assigned to the test set (in minutes and seconds)? How much of this test data is your chosen word or sentence (in minutes and seconds)?

In the top right corner of the screen, select “Get Started” and create an Edge Impulse account.

- Create a new project, name the project, and make sure the “Developer” type is selected.
- On the next screen, choose “Accelerometer Data” and then select “Let’s Get Started.”
- Navigate to the Data Acquisition page and then the Upload Data tab.
- Unzip “edge-impulse-studio-export.zip”. Select “Choose Files”. Navigate to where you have unzipped all 80 data JSON files and select all 80 at once. Check that it says “80 files” next to the “Choose Files” button. The “Upload into Category” should be “Automatically split between training and testing” and “Label” should be “Infer from filename.”
- Select “Begin Upload.” In the “Upload Output” display on the right, if there are any files that failed to upload, retry the uploading process again.

Still within the Data Aquisition page, navigate to the Training Data tab.

12. What are the four labels in this dataset?
13. How many minutes of total training data is in this dataset? Does it look approximately equally distributed among the four labels? What implication could this have on training the model?
14. Click on a sample to visualize the raw data on the right side of the screen. Attach a picture of the raw data of one sample in each label.

Navigate to the Testing data tab.

15. How many minutes of total testing data is in this dataset? Does it look approximately equally distributed among the four labels?

Move to the Impulse Design page. There should be a Time Series Data block already, but no processing block or learning block.

16. Click “Add a processing block” and click “Add” next to the Spectral Analysis option. What are the three default input axes?

- Click “Add a learning block” and click “Add” to add the Classification option.
- Select “Save Impulse” on the far right under the Output Features block.
- Navigate to the Spectral Features subpage.
- Add a low pass filter by changing the “Type” of Filter to low. You can keep the other parameters as their default or adjust them. The right side of the screen shows what the data will look like after the filter is applied, based on the parameters set on the left side.

17. Did you change any additional parameters for the Filter or Analysis? If so, list what you changed. You can always go back later and continue making adjustments after viewing the performance of the model.

- Click “Save parameters”.
- Check the box for “Calculate feature importance” and then click “Generate features”.

18. Attach a picture of the Feature Explorer block on the right, showing the data clusters.
19. Looking at the Feature Importance, what was determined to be the most important feature for this dataset?

- Navigate to the Classifier subpage.
- Change the number of training cycles to 100 and then scroll to the bottom and click “Start training”. Like the Spectral Features subpage, you can adjust any other parameters on this page to see how they adjust the training process. You can also return to this later and continue making adjustments.

20. If you made additional adjustments, list them here.
21. What is the accuracy and loss for the resulting trained model?

If you want to make additional adjustments and explore the parameters in the Spectral Features and Classifier subpages, you can go to the Retrain Model page when you are done to retrain the model in one step.

- Navigate to the Model Testing page and click the green “Classify all” button.

22. What was the model testing accuracy result?

- Navigate to the Versioning page and select “Store your current project version” in the top right corner. Select the green “Store version”.
- Once the output displays “Job Completed,” close the popup window. You should see the stored version in the log on the page.
- Navigate to the Deployment tab.
- Scroll down to Build Firmware and select Arduino Nano 33 BLE Sense.
- Continue scrolling to Select Optimizations. Select the orange “Analyze optimizations” to determine if any optimizations would improve the model performance on the device without sacrificing accuracy.
- Once the optimizer has completed, select “Build” at the bottom of the screen.
- When the build completes, there should be a tutorial video that appears to show you how to flash the binary for deployment. You do not need to do this for this studio.

23. From the video, how do you flash the binary?
24. What command do you run in the terminal/command prompt?

Submit the downloaded zip file with your firmware deployment along with your answers to the above questions.

Studio: 7.Model Performance
In this studio, you will explore overfitting and dataset imbalance, two challenges in modeling that affect model performance.

Before beginning this studio, read through the Edge Impulse documents about increasing model performance to learn more about overfitting and imbalance.

Please complete the Machine Learning with Edge Impulse studio before working through this one.

- On your Edge Impulse dashboard, create a new project and title it “Overfitting”. Make sure the project type is still marked “Developer”.
- Choose Accelerometer Data when prompted and then select “Let’s get started.”
- Unzip model-performance-studio-export.zip to access the data for this studio. The data uploader in Edge Impulse accepts CBOR (Concise Binary Object Representation) files, which is what is provided in the model-performance-studio-export.zip.
- Navigate to the Data Acquisition page and go to Upload Data.
- Select “Choose Files” and find the CBOR file samples inside the “training” folder. Select all 98 samples in the folder to upload. Change “Upload into category” to “training”. Leave “Label” as “Infer from filename” and select “Begin Upload”.
- Once it has finished uploading, repeat the process to upload the 15 files from the “testing” folder, making sure to change “Upload into category” to “testing”.
- Navigate to the Training Data tab.

1. What is the training/testing split for this project after uploading all the given data?
2. How many different labels are in this dataset? What are the labels? Which one has more data than the rest?
3. For the label with the most data, click on various samples to view the raw data plots. Find and attach two pictures of samples that seem to have very different X, Y, and Z acceleration patterns from each other.

- Navigate to the Impulse Design page.
- Add the Spectral Analysis processing block.
- Add the Classification learning block and save the impulse.
- On the Spectral Features page, add a low pass filter with a cutoff frequency of 2. Save the parameters and generate features.

4. List observations about the Feature Explorer output after the features are finished generating.

- Navigate to the Classifier page.

- Change the Number of training cycles to 100, the learning rate to 0.09, and leave the Validation test size at 20%. Do not check the “Autobalance dataset”. Start Training.

5. What is the resulting accuracy and loss? Attach a picture of the confusion matrix.

- Navigate to the Model Testing page. Select “Classify All”.

6. What is the model testing accuracy? Attach a picture of the confusion matrix. How does this compare to the training data classifications?
7. Why might the results be signifying the presence of overfitting?
8. What might be causing snake data to be classified with the wave label? Hint: look at how much data of each label is in the training and testing sets.
9. List at least three aspects of the dataset, dataset breakdown, or the above steps that could be adjusted to prevent overfitting if the model were to be retrained and tested. Refer to the document about overfitting in Edge Impulse.
10. In the current model, as it was trained and tested above, was the dataset imbalanced?
11. How could a dataset imbalance influence overfitting in the model?

Studio: 8.Combining Models

In this studio, you will explore combining models, as well as the advantages and disadvantages to the number of models used in a project.

The lab is designed to both recognize a punch and classify its type between straight, hook, and uppercut. This multi-level problem could be implemented in different ways, but the design lab focused on training two independent models and then deploying both.

1. What is one advantage and one disadvantage of having a single model for both punch recognition and classification?
2. What is one advantage and one disadvantage of having two models, one for punch recognition and the other for classification?

Please complete the Machine Learning with Edge Impulse studio before working through this one. To analyze the difference between one model and two models:

- Open Edge Impulse and create a new project.
- Unzip the combining-models-studio-export.zip file.
- Navigate to Data Acquisition and the CSV Wizard. Upload one of the data files from the zip to the CSV wizard.
- For Step 2 of the CSV wizard, click "Looks Good, next".
- For Step 3, select "Yes, this is time-series data".
- To answer the question "How is your time-series data formatted?", select the first option, indicating that each row contains a reading.
- Select "Yes, it's timestamp" and then "Time elapsed in milliseconds".
- Enter 125 ms for the timestamp override at the bottom of the screen. Edge Impulse should automatically detect 8 Hz. Then select "Great, let's look at your values".
- For Step 4, select No for "Do you have a column that contains the label?" and then uncheck "received". The columns that contain values should only be checked for accX, accY, and accZ. NOTE: the columns must be titled as accX, accY, and accZ for the live classification feature to work. When collecting data on your own, ensure that these column titles match if you plan on testing with live classification through Edge Impulse. Live classification will be discussed further in the Deploying Models studio.
- Click "Next, split up into samples".
- For Step 5, select unlimited for the sample length and then click "Finish wizard". This process gave Edge Impulse a template for how to treat the data files when uploading samples to the dataset.
- Click "upload some data" on the Step 6 screen.
- Upload all of the data from all four folders inside the zip, separately by folder, making sure to keep "automatically split between training and testing" selected. Enter the label

with the name of the folder. For example, enter the label "background" when uploading all 79 files from the background folder.

- Navigate to Impulse Design. Add a Spectral Analysis Processing Block and a Classification Learning Block and then save the impulse.
- Under Spectral Features, leave the default parameters and click "Save Parameters" at the bottom. Then Select "Generate features" on the next page.

3. Attach a picture of the feature explorer. There should be four different classification categories.

- Navigate to the Classifier page and change the number of training cycles to 100. Leave the rest of the parameters as their defaults and then select "Start training".

4. What is the accuracy of the results? Attach a picture of the confusion matrix.

- Return to the Data Aquisition page. Select the checkbox symbol at the top of the Dataset section. Hovering over this symbol should show the words "select multiple items". Then check the box at the top, next to Sample Name, to select all the training data. Select the delete button.
- Switch to the test data tab and, again, delete all the data.
- A new prompt to upload data should appear. Upload all the data from all four folders again, but this time labeling the data from the background folder as "background" and labeling all other data as "punch".
- Return to the Create Impulse page and "save impulse"
- Return to the Spectral Features page and select "save parameters" and then "generate features".

5. Attach a picture of the new, binary feature explorer output.

- Return to the Classifier tab and start training. Ensure the number of training cycles is set to 100 and then start training.

6. What is the accuracy of the new results for just binary classification: punch vs. background? Attach a picture of the confusion matrix.

- Create a new Edge Impulse project and follow the same steps as above to set up the CSV wizard.
- Import data, in the same way, but only upload data from the hook, straight, and uppercut folders, labeling them with their corresponding folder name.

- In Impulse Design, add a Spectral Analysis Processing Block and Classification Learning Block and then save the impulse.
- In the Spectral Features page, leave all parameters as their defaults. Select “Save Parameters” and then “Generate Features”.

7. Attach a picture of the feature explorer output. There should only be three classification categories: hook, straight, and uppercut.

- On the Classifier tab, change the number of training cycles to 100 and “Start training”.

8. What is the accuracy of the new results with just punch classification? Attach a picture of the confusion matrix.
9. What conclusions can you draw from the results in questions 1-8?

Note: To optimize the classification performance, we will focus on using two models: one for binary classification between a punch and background movement, and the other for punch type classification between straight, hook, and uppercut.

<u>**Studio: 9.Deploying Models**</u>

In this studio, you will explore methods for testing the trained models, including using the Live Classification feature in Edge Impulse and deploying the models to Arduino Nano boards.

The Live Classification feature in Edge Impulse is useful for testing your designed model without the Arduino Nano. It differs from the Model Testing feature because it classifies data in real time rather than showing the results of the model classification on the set-aside testing dataset.

Please complete the Combining Models studio before working through this one.

- Open the project in Edge Impulse that was created in the Combining Models studio for binary classification: punch versus background. All of the training steps, up to the Classifier, should already be complete.
- Navigate to the Model Testing page and select the green "Classify All" button. This will use the trained model to classify the data in the test set that was set-aside during training.

1. Looking at the model testing results, what is the accuracy of the test? Attach a picture of the confusion matrix and the feature explorer that are produced.
2. What is the purpose of setting aside a test set of data while training the model?

- Navigate to the Devices page and select "Connect a new Device" in the top right corner.
- Select the box for "Connect your device or development board".
- In the search bar, search for Arduino Nano 33 BLE Sense and select it. Select the Button to view installation documents for this device. This is a useful guide to refer to if you run into problems, however, the next few steps will also walk you through connecting the device.

Setting up the Edge Impulse CLI:

- Confirm that you have access to your Edge Impulse username and password.
- Confirm that you have Python 3 installed on your local computer. You can run `python --version` in the terminal/command prompt to check the version.
- Confirm that you have Node.js version 14 or higher on your computer. You can run `node -v` to check the version. Note that this might also require updating npm on your computer if you later run into errors caused by npm and/or gyp.
- If your computer is running MacOS, Ubuntu, or Raspian OS, run the command `npm config get prefix` in the terminal/command prompt to check the directory that node is stored in. If this command returns `/usr/local`, run the following commands to change the directory.

```
mkdir ~/.npm-global
```

```
npm config set prefix '~/.npm-global'
echo 'export PATH=~/.npm-global/bin:$PATH' >> ~/.profile
```

- Install the Edge Impulse CLI by running the command `npm install -g edge-impulse-cli`. This should take a few minutes. If you run into errors about npm and/or gyp, run the following commands to update to the latest version of Node.js and then try to install the CLI again.

```
npm install -g n
sudo n latest
```

3. The CLI should now be installed on your local computer. Connect your Arduino Nano board to the computer via USB. Run the command `edge-impulse-daemon` in the terminal/command prompt. Right now, the daemon should fail to find the device, but what is the first line to print to the terminal/command prompt?

- Install the Arduino CLI on your computer, if you don't already have it. On a MacOS or Linux computer, run the command `brew install arduino-cli`. On a Windows computer, follow these instructions to download the 64-bit binary from the Arduino Installation site.
- If you are running a Linux computer, run the command `sudo apt install screen`.

We now have all the dependencies for Live Classification. Download the latest Arduino firmware here or from the 9.deploying-models folder in the repository. This should be a zip file. Unzip it and find the script that corresponds to the operating system on your local computer.

4. What is the script called?

- Click the Arudino's reset button twice to start the bootloader. The built-in yellow LED should look like it's breathing. Right click on the script from question 4 and run it in the terminal/command prompt to start flashing the board with the new firmware.
- When the process has completed, press the Arduino reset button once more to launch the new firmware.

5. In a new terminal/command prompt window, run the command `edge-impulse-daemon` again. This time it should prompt you with questions. Answer the questions that it prompts. What are the questions it asks in order?

- Return to Edge Impulse where you have the Devices page open. You should now see a device connected to the project. It should have a green dot in the Remote Management column.

6. What is the device's MAC address?

- Navigate to the Live Classification page. You should see the question for Device auto-populate with the name that you assigned the connected Nano. If not, use the drop-down to select the connected Nano.
- Change the sensor to inertial.

7. Why does it make sense to use the "inertial" sensor?

- Change the sample length in ms. to 5000.
- Select "Start Sampling". You will have 5 seconds to collect live data. Stand still for a couple seconds and then perform a straight punch.

8. Attach a picture of the raw data and the spectral features. Additionally, include a picture of a section of the samples from the detailed results. The detailed results should show the percent certainty of its classification in each category for each sample.

- Feel free to continue sampling and explore what happens when sampling for longer.

Deploying the model, rather than live classification, will put the classification model in an Arduino library for use directly on the board.

- Navigate to the Deployment page. There will be two options: to create a library or build firmware. In the "Create Library" section, select Arduino Library.
- Scroll to the bottom of the screen. If there is a red/orange button for "analyze optimizations", select it. This will choose optimizations for your classifier. If there isn't, use the default optimizations.
- Select "Build".

9. Which optimization did the platform use for this library?

- Open an Arduino Sketch in the Arduino IDE. Navigate to Sketch > Include Library > Add .ZIP Library… and select the zip file that was downloaded when the build completed.
- Then, navigate to File > Examples > [name of project] to find example scripts using this custom library. If the new library doesn't show up as a custom library, try closing and reopening the Arduino IDE.

10. Submit your downloaded zip library with your answers to these questions.
11. Open the nano_ble33_sense_accelerometer_continuous example script. What #include statements are at the top of the script?
12. Upload the script to the Arduino Nano Board. Once it finishes, open the serial monitor. Leaving the board flat on a surface, what line does the model print to the serial monitor.
13. Holding the Nano, punch it straight towards the computer screen. What line does the model print to the serial monitor?

Studio: 10.Graphical User Interface
In this studio, you will explore ways to design a graphical user interface as a cohesive product for the machine learning project.

The design of a project is typically targeted at a particular problem for a customer/user. Because of this, it is helpful to have a single, cohesive product or deliverable that can be given to the customer rather than showing them several individual parts. Since there are many parts to this project, one helpful way of producing a deliverable is by creating a Graphical User Interface (GUI) that walks the user through all the steps, from data collection to deployment.

1. List at least three different frameworks for creating a GUI in Python.
2. What main components should be available in the GUI?
3. What are some methods for easily displaying all parts in question 2 on the GUI?
4. For this question, assume you have chosen to include the following components: Collect, Analyze, Model, Test. What are some important aspects/functionalities that should be included in each component?
5. Including features in the GUI will require additional Python packages other than just those provided by the main framework. Research some techniques for implementing the aspects you identified in question 4. List at least three Python packages that would be necessary/useful to import when creating the GUI.