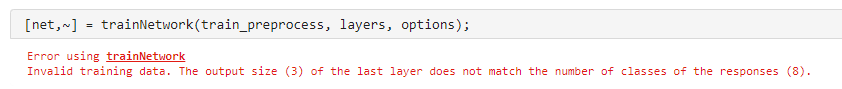
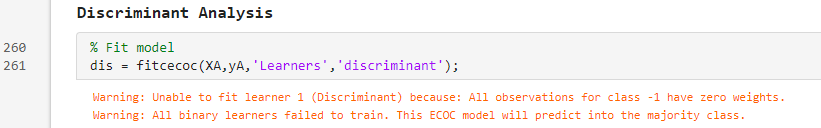
**Additive Manufacturing**

* Nice use of Table of Contents, perhaps you can add data set either under data cleaning section or make a subtitle for it too. It is in the intro part, but at the very bottom. It may be hard to see. Since in the next part, you read the data, it is not super critical, but would make the text easier to follow.
* Great use of Live Tasks (if we haven’t used Live Tasks before, maybe you can add a line about how you added the Live Task or some general information about Live Tasks)
* Nice explanation above line 57
* Nice explanation above line 72
* In bar chart, y-label (I think it should be z score) is missing. Please add the label.
* Good interpretation after line 138
* Figure under line 188 doesn’t have labels. Please add labels.
* Jon-Looks good

**Bit Classification**

* Good that you gave computer specs
* If test data is used as validation data, how will we do test? Will we give non-drill data? Maybe you can explain that in the text before line 1
* Another advantage of using augmented image datastore is you can resize all the images to match the input size of the network.
* Above line 29, by “larger values” which values do you mean? Perhaps, you can say something like “Based on the size of the dataset, size of the network and hyperparameters chosen, training time of the network may vary. We trained this network locally, but if more compute resources are needed, these computations could be performed using cluster or cloud computing or using local NVIDIA GPUs if these are available for use. Deep Learning Toolbox automatically detects and uses compatible GPUs if they are locally available.
* You can separate build network into 2 part, programmatic and interactive approach. Make it clear that you are building the same network interactively in the 2. Approach. You can also mention that Deep Network Designer allows you to automatically generate code.
* Nice labels for true values and predicted values!
* Seems like what can be improved section is the same as bit classification. Maybe you can say similar suggestions as above apply here too. Either way is fine.
* Jon- I get an error, 

**Classification Overview**

* Are we using scikit learn data set or MATLAB MNIST dataset? Wording is confusing since you mention scikit learn first. Seems like we are using MNIST, I’d say remove the text about scikit learn data set to avoid confusion.
* If you copied any code from the documentation, instead of copying, use open example command from documentation to show the original example, then explain what you modified. If your code is original, you can include MNIST related doc examples as further reading.
* You can break SVM into 2 parts and mention apps under interactive approach to make it clear and keep your code under programmatic approach.
* Prediction logical arrays may be confusing to the audience, write a bit more description/comment. What does prediction logical array mean? If it is 0, does it mean the prediction was wrong? Maybe printing true value and prediction value and labeling them would make it more clear. (line 113 and earlier)
* You can also define your definition of accuracy to be clear. Is it 0.71% accuracy or 71% accuracy?
* Are commented lines from 257-259 needed? If not, you can delete them to keep things simpler
* Jon- line 36, what is the first classifier? Maybe comment on this
* Jon- when selecting the options for which data set to use, I get a warning for the “moons” data .
* Also for the “circles” and “moons” data set, it only picks out one of the regions

**Cyber\_Security**

* Good mention of interactive methods and good interpretation of the results.
* Jon-good

**Draw\_Classification**

* **Jon- explain what is happening in this module at the beginning. Explain the data set and what is going to take place**

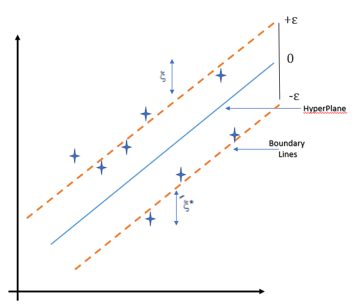
**Hand\_Tracking\_modified.mlx**

* **Jon- for the “Method 2: Hand Tracking using Foreground Detection” what should it be doing? I see a black and white video.**

**Neural\_Network\_Regressor.mlx**

* **Jon- I found something strange here. I noticed the MSE for the tanh and cosine activation functions were the same, that certainly should not be the case. I found 2 major things:**
  + **The mse function in MATLAB needs to have the same dimension vectors, otherwise it just uses the first element.**
  + **Scaling data here is actually a bad thing to do, simply because so that might be something to discuss in that module, i.e. scaling data is not always good**

**Support\_Vector.mlx**

* **Maybe add an image such as this:**

**As long as it’s not copyrighted... 😊**

**reinforcement\_learning.mlx**

* **Should we be able to see the cart-pole “learning. ” Probably not I guess**