Feedback/Comment

Table of Contents

[Feb 10, 2023 1](#_Toc131761866)

[Mar 3, 2023 3](#_Toc131761867)

[Apr 7, 2023 5](#_Toc131761868)

# Feb 10, 2023

**Overall**

1. Keep naming conventions consistent between modules teams.
2. **Folder structure:** consider separating drafts, ready for review, validated, … etc and make sure the folder names are self-explanatory.
3. If there are dependencies, considering numbering them in sequence.
4. **Communication:** set expectations
   1. Summarize questions in a central place (for example, in this log) so that we don’t miss any of your questions.
   2. Please be explicit and let us know which one(s) you’d like us to review. In this case, it seems that “Final Modules” would contain the ones for review or will be published?
5. Consider adding table of contents to live scripts for easier navigation.
6. Where available, text description is very well written. Great job!
7. Consider adding graphics and equations where applicable, to make modules more self-serve and visually easier to digest.

**Module\_Team1**

Please avoid punctuations in file names

**GatherData\_forvalidation\_validated\_needs\_revision**

1. You can also show import tool here as a way to bring data interactively into MATLAB or you can link documentation.
2. Yes, there is a way to webscrub in MATLAB, you can use “webread” <https://www.mathworks.com/help/matlab/ref/webread.html>
3. Rightjoin – are you asking about merging tables, if yes, you can do that <https://www.mathworks.com/help/matlab/ref/table.join.html>
4. You can include doc links
5. You can use “summary” function to show overall summary of the data

* <https://www.mathworks.com/help/matlab/ref/table.summary.html>

Other suggestions:

* Add table of contents to live scripts
* You can add additional resources links such as MATLAB Onramp, MATLAB Fundamentals etc. and point out relevant sections.
* Maybe timetable and datastore would be nice to mention
  + <https://www.mathworks.com/help/matlab/ref/timetable.html>
  + <https://www.mathworks.com/help/matlab/datastore.html>

**Clean.mlx**

* Are you planning to add any comments to this script?
* It will be also nice to show preprocessing interactively using apps or at Import Tool when you are importing data.
* Good use of logical indexing

**Module\_Team2**

**Feature\_Engineering\_Module**

* Check spelling – there are spelling mistakes
* Maybe you can link this: <https://www.mathworks.com/discovery/feature-extraction.html>
* Feature hashing plot – investigating (maybe Jianghao can comment)
* Text reflects Python (under Feature Collection and Generation) – needs revision
* Are you looking for chi square goodness of fit test? (maybe Jianghao can comment) <https://www.mathworks.com/help/stats/chi2gof.html>

**Module\_Team2\_splitdata**

<https://www.mathworks.com/help/matlab/matlab_prog/split-data-into-groups-and-calculate-statistics.html>

<https://www.mathworks.com/help/matlab/ref/matlab.io.datastore.imagedatastore.spliteachlabel.html>

* You can also set split ratios when you are training the models
* Cvpartition is good

**Module\_Team3**

Feature engineering:

**Question from the team:** “How can we get the K scores in matlab? Dr. Hedengren uses a python package, sklearn.feature\_selection to get k scores and then he plots those scores for each feature. We cannot find an equivalent for this in MATLAB.”

**Answer:** K-score is not a terminology. In this case, I think it’s choosing the top K highest scoring features. K is a number that falls in 1:total number of features. You can decide on what test you run (i.e. a scoring function) to calculate a score, and then sort() and return the indices of the K highest scores.

**Final Modules**

Stats\_Module.mlx – need a title

Visualize\_validated.mlx – need more text description/comments in the main part of the script

# Mar 3, 2023

**Overall:**

* Great work!!
* Table of content needs to come after titles (and possibly after your initial problem description)
* Okay to reference non-built-in functions as long as they are included in the repo
* I’m not able to run any code because for many, data is not available
* Many functions work with the native “table” data type, i.e. no need to covert to array for functions to work. Go through the script and refer to documentation to optimize the scripts. Some examples are provided below
* How are we doing on the overall progress?
* For each check-in meeting, would you provide an update to the files reviewed in the previous meeting?
* Do you at some point need to use GPUs to train a neural network? How and where do you get access?
* Creating data with labels – you mentioned that was done in Python, can you say more? (is it possible to do this all in MATLAB?)

**Imbalanced\_Validated**

1. Line 2 & 51 & 154: make sure data is included in the final submission, I suppose in the same Git repo works.
2. Plot data: scatter() and many other MATLAB functions work with the “table” data structure. Take a look and consider avoiding loops: <https://www.mathworks.com/help/matlab/ref/scatter.html>
   1. Search for “tbl” and try relevant examples
3. SMOTE: consider using this one instead: [https://www.mathworks.com/matlabcentral/fileexchange/75168-oversampling-imbalanced-data-smote-related-algorithms](https://www.mathworks.com/matlabcentral/fileexchange/75168-oversampling-imbalanced-data-smote-related-algorithms?s_tid=FX_rc2_behav)

It’s also okay to include the functions directly in your repo as long as they are properly referenced

1. Separate “Activity” from the rest of the code, so that it’s clear all other is lecture, while activity is homework (it also seems that the code here is the solution – make sure you also have a corresponding work file)

**Linear Regression Validated**

1. Two examples demonstrate multiple Python methods for (1) univariate linear regression and (2) multiple linear regression. Should be MATLAB I suppose?
2. Maybe good to break down and blend text description and code together. I will explain when we meet.
3. I like how things are broken down in the original Python code, i.e. solution file shows how you can write everything using NumPy to how you can directly call a built-in function. Please consider doing something similar here.
4. Okay to include supporting functions. Great that you include it at the end. Perhaps make it clear that they are supplementary by bolding and renaming

**Polymer\_Melt\_Flowrate\_Validated**

Great work. No major feedback.

1. Condense table of content,
2. KNN – okay to use custom function, just be sure to include the actual function in the repo
3. Feature selection
   1. Feature selection in general: <https://www.mathworks.com/help/stats/feature-selection.html>
   2. fscchi2 works with table. Have you tried that? It may solve the problem you asked in the other document.
4. fitlm works with table (line 99 – 103)

**Scale data for machine learning**

Great problem description and script.

1. Add table of content (after problem description)
2. Separate activity from example, and include work/solution copies for activity

**Sonar Detection**

* Script is incomplete, I will review when it’s done
* **Also for case studies in general, make sure Ayan has a pass at it as she’s the chemical engineering expert.**

# Apr 7, 2023

**Bit\_Classification\_Ready4Validation**

**Neural Network**

Does training time go up for each larger value of the hyperparameter?

Make sure to spell MATLAB right, all capital.

Nice job in explaining interactive workflow.

For training times you indicated, please include computer specs (whether training was done on a cpu, gpu etc.)

In Python version of bit classification, there is reflections etc. in the data augmentation step. You did that for concrete example, but I didn’t see it for the bit classification. Please add preprocessing.

**Before Concrete Activity:**

Python code has sections about Hyperparameter tuning and validation accuracy. I didn’t see those in the Live Script. Please add them. You can also mention “Experiment Manager” for Hyperparameter tuning.

**DrawClassification\_Ready4Validation**

In Python version, user specifies the points by drawing them. We mentioned using [getpts](https://www.mathworks.com/help/images/ref/getpts.html) function for that. Did you try it? Didn’t it work as you expected? Is that why use used saved data set?

It is nice that you mentioned classification learner app. You can also mention how different models can be trained in parallel using this app.

**SoilClassification\_Ready4Validation**

Make sure to spell MATLAB correctly (should be all capital).

Maybe you can also mention that different CNNs you use take different input sizes, that’s why you are rescaling your images to a certain size using augmentedImageDatastore.

You can get the arrows straight in the network image.

Nice that you mentioned Deep Network Designer and good use of screenshots.

“Ignore warning” -> what was the warning, maybe you can tell why it doesn’t matter?

**Soil\_Classification\_Ready4Validation**

1. Consider referring to MATLAB documentation when applicable, for example: <https://www.mathworks.com/help/deeplearning/ug/introduction-to-convolutional-neural-networks.html>
2. Line 12: Great use of datastore! Consider using **filesep** instead of / or \ so that the code works on all OS

train = imageDatastore(['train',filesep], 'IncludeSubfolders',true

1. Text above line 18: Add hyperlinks to deep network designer (DND) documentation: <https://www.mathworks.com/help/deeplearning/gs/get-started-with-deep-network-designer.html>
2. After line 43: Include a line of code to open DND

deepNetworkDesigner

1. Can you explain line 51 - 60?
2. Section model testing can be shortened, see example: <https://www.mathworks.com/help/deeplearning/ref/seriesnetwork.classify.html>
3. Line 100 – 137: are there ways to vectorize, i.e. avoiding loops?

**Draw\_Classification\_Ready4Validation**

1. Link is broken (For more information on the different classifier methods, go to this link.)
2. Would it help if you have a summary table at the end of the script to compare the results from different models? Also are the models introduced earlier in other modules? It may be useful to link to their definitions as you use them in the script.
3. Objective can be expanded and more complete.
4. Towards the end, consider including some text on the key outcome of the script.

**Bit\_Clasification\_Ready4Validation**

1. Line 55 – 56, I suppose you can remove the output arguments and just leave them as

rmdir('train','s')

rmdir('test','s')