Module A

Create an iOS application using the HTTPSwiftExample that:

- Collects some form of low throughput (sampling rate > 1s) feature data for processing: audio, video, motion, or from the micro-controller
- Uploads labeled feature data to a server via HTTP POST requests
  - you can run the server from your laptop or mac mini
  - Alternatively you can use a virtual machine, AWS, or other cloud service
- Trains a model from the labeled data (*e.g.*, KNN, SVM, Random Forest, *etc.*)
- Requests predictions from the server by uploading unknown feature vectors
  - can be periodically or initiated by user
- Note that the server code given to you will automatically save any feature data you upload and train a machine learning model, given the correct POST/GET request commands

You should not need to update the server for any of the given functionality. However, the predictions from the server may not be sufficient without updating the training parameters or the type of model used. Verify the functionality of the application to the instructor during lab time or office hours (or scheduled via email).

You have a lot of free reign in this assignment to create something interesting and unique. Try to make this one iteration of the final project.

Module B:
Update the HTTPExample and the tornado web server to:

- Specify the type of model to use in the Machine Learning via the iOS POST request(s)
  - at least two different types of machine learning models (*e.g.*, SVM and KNN)
- Compare the efficacy of two or more different models
  - send parameters to use in the machine learning models from the phone (e.g., number of neighbors to use in KNN)

Exceptional Work: 7000 Level Students Choose ONE of the following:

- make the training of the model non-blocking to the tornado IOLoop
- implement authentication in tornado and in your iOS application
- Use CoreML to export your custom trained machine learning model and run the machine learning prediction locally on the iOS app (NOTE: the CoreML model must be exported from the data you create on your HTTPServer)
  - Also, the CoreML implementation should sufficiently different from the class example.