MOBILE SENSING LEARNING

Mobile Sensing Video Assignment Four:

Learn It, Save It, Load It, Use It

You are to complete this assignment in **groups of two or three**. You are to complete the following (or get as far you can). Start with

1. the Xcode project (`InClassAssignment` branch of `HTTPSwiftExample`) and
2. python server (`turi_create_mongo_six` branch of the `tornado_bare` repository) from the video lecture.

If you are using an M1/M2 Mac be sure you have installed the correct versions using a terminal with Rosetta.

[1 points] Part One: Run the Server

3. Run MongoDB from your mac and then run `tornado_turi_create.py`
4. Look up the name or IP address of your mac (e.g., using `ifconfig |grep "inet "`)
5. Connect the iPhone to your server and navigate to "`http://(your IP):8000/Handlers`" using safari in iOS
   1. *If you can't connect, create a WiFi network from your mac and try again*
6. Take a screenshot **from the phone** and save it for uploading with the project.

[1.5 points] Part Two: Update the App

7. In the iOS `HTTPSwiftExample` App, change the name of the URL to point to your server
8. Update the User Interface and code so that the DSID can be selected by the user (you can use any UI element you want for selecting a DSID). That is, the code should now use the user selected DSID, rather than the default DSID. A nice UI choice for this is a stepper.
9. Run `HTTPSwiftExample` on the iPhone and perform the calibration procedure a few times in order to get some example data into the database, then train a model. Note that you must click "train model" from the interface in order for a model to be trained.
10. After clicking "train model," take a quick video of the example working with a predicted feature vector. **The classifier does not need to be very good**. It will likely be terrible… that's okay!

[2 points] Part Three: Change the Loading of the Classifier

All of the following changes should be made in the files `turihandlers.py` and, possibly, in `tornado_turi_create.py` (depending on your implementation choices):

11. Setup the `.clf` property to be a dictionary rather than just being the Turi model.
    1. In this dictionary the **key should be the DSID** and the **value is the turi model**. This enables you to have multiple models loaded, each with different DSIDs. That is, when training a model, update the code to save the DSID and classifier as a key/object pair in the `.clf` dictionary. In this way, all models can be loaded into the `.clf` dictionary (not just one model). As new models are trained, they will be added to the dictionary.

More steps on next page

MOBILE SENSING LEARNING

12. Update the code in `PredictOneFromDatasetId` to:
    1. **Change the loading of the classifier**: Check if the requested DSID from the POST request exists in the `.clf` dictionary. If it does not, then load the requested classifier and save it in the dictionary property as a new key/object pair. If a model has not been trained for that DSID, notify the app in some way. **Account for any errors that might occur in loading and saving the model.**
    2. Update the prediction function to use the `.clf` dictionary property for predicting a label from the uploaded feature vector.

[0.5 points] For Thought

13. Is the current method of saving the classifier blocking to the tornado IOLoop? Justify your response.
14. Would the models saved on one server be useable by another server if we migrated the saved documents in MongoDB? Justify your response.

What to turn in:

- Team member names, zipped Xcode project, zipped python code, screenshots/videos, and answers to "for thought" questions (written out in complete sentences with valid justification for answers).

This was the last flipped assignment for the course! Thanks for all your hard work in preparing for these throughout the semester.

Summary of Steps

- Run mongodb using are services
- Run tornado server from the Turi branches
    - `python tornado_turi_create.py`
- Get your ip address using `ifconfig |grep "inet "` and update the Swift code for the app to point to your server SERVER_URL
- If running, **add some calibration data to the server**
- Click Update Model button
- Receive all the predictions from the Server
- The terminal running tornado will show each time that a connection exists.

SMU | Bobby B. Lyle School of Engineering