Quick-Start Guide

1. To run the app, cd to the ***ml-scheduler*** directory and run ***docker compose up --build***

   Note: *If you would like to try modifying the code and saving it, the containers are set up to detect the change and re-load with your new changes in place.*

2. The six unit tests in *api/tests* will run and try to pass before the image is built.

3. Two services will spin up from separate containers, ***api*** and ***scheduler***

   a. The API will use rf-1-base.joblib as a base model, with version 1.0 to start.

   b. Every interval, the scheduler will first conduct checks and measure metrics.

      i. If they are satisfactory it would also update the model to a randomly picked model in the *scheduled_tasks/retrained_models* folder. This is done by sending a POST request to the API at *http://api:8000/update_model* with a Base64-encoded form of the model.

      ii. The scheduled re-training interval can be adjusted in *scheduled_tasks/scheduler_service.py* with the constant RETRAINING_SEC_INTERVAL.

To send sample POST requests for prediction, there are two methods to do so:

1. Method 1: After the container is on, navigate to the SwaggerUI Docs on the browser at http://localhost:8000/docs

   a. Go to the ***/predict*** request and click “Try it out”. Click “Execute” on the sample request body. Scroll down to the resulting response body. It should return:

   ```
   {
     "species": <prediction>,
     "model": <name_of_the_model_that_generated_the_prediction>
   }
   ```

2. Method 2: After the container is on, copy the following sample command and run it in another terminal window. It should return the same response as Method 1, and it should also log ***"POST /predict HTTP/1.1" 200 OK*** in the API container.

   ```
   curl -X 'POST' \
     'http://localhost:8000/predict' \
     -H 'accept: application/json' \
     -H 'Content-Type: application/json' \
     -d '{
     "sepallength": 1.0,
     "sepalwidth": 1.0,
     "petallength": 1.0,
     "petalwidth": 1.0
   }'
   ```