EMERGING TECHNOLOGIES I

MIS 284N | Fall 2021

Project Milestone 5

Overview: In the fifth milestone, we will provide a trained model that will predict a "goal" step count based on the day's weather. This model will run on the Raspberry Pi, using the weather information retrieved from the Android device via MQTT. Instead of the Raspberry Pi sending back the (private!) step count to the Android device, it will now just send back an indicator as to whether the goal was achieved or not. The main task here is to understand the provided model and how to preprocess the inputs for using the model to obtain the step count prediction. The following steps walk you through the system integration for your prototype step predictor and evaluator.

Step 1: Obtain the weather data from the openweathermap (Milestone 4).

Step 2: Send the weather data to the Raspberry Pi via MQTT; relative to Milestone 4, you will need to be sure to send *all* of the data expected by the model prediction.

Step 3: In python, on the Raspberry Pi, write linear regression code that uses the model weights provided below to convert input weather data into a step count prediction. The three pieces of weather data you will use for the provided model are: daily high temperature, daily low temperature, and daily humidity level. We have four weights for the model; the first W0 is for the offset; the other three, in order, are for the high, low, and humidity values. These weights (based on your provided step input) are:

W0 = 718.91565187

W1 = -18.87648056 (max in F)

W2 = -25.9855236 (min in F)

W3 = -27.87443884 (humidity in percentage)

For completeness, the model is:

$$\text{Steps} = W0 + W1 * \text{Max} + W2 * \text{Min} + W3 * \text{Humidity}$$

For this step, use the current day's weather data (i.e., the current weather from openweathermap) to predict the number of steps the user should have walked (given the current data), given the measured weather for that day. Print this to the terminal on the Raspberry Pi.

Step 4: Do the same for tomorrow. Use the upcoming day's weather *forecast* from openweathermap to communicate to the user how many steps he or she *should* expect walk that day. Print this to the terminal on the Raspberry Pi.

Step 5: Design your system so that the microbit communicates steps in a controlled fashion to the Raspberry Pi, and the Raspberry Pi program stores the step count in a variable. You will have to design how the step count is reliably communicated from the microbit and reset daily, given how you expect to

support user interaction. That is, you are also designing the UX (*user experience*) and communicating it to your user as intuitively as possible.

Step 6: Once you have a "reliable" number for steps-per-day in the Raspberry Pi, in the python program, compare the actual steps to the predicted steps (for both actual and forecast data). For instance, you might print, at the Raspberry Pi, both the (actual/predicted) steps for the current day plus the goal steps for the following day.

Step 7: Refactor your Android program so that when you click on a UI button to check on whether the user has achieved the "Goal", the Android should (a) retrieve the weather data from openweathermap; (b) prompt the user switch the wireless networks; (c) upon the user's command acknowledging that the network has been changed, send the weather to Raspberry Pi via MQTT. The Raspberry Pi should respond to your Android program with something like 'Goal achieved, keep going!' or 'Work harder', and this message should be displayed to the user. It is entirely up to you to design the constraints in the UX, but you will be graded on how easy your system is to use without assistance. In particular, for full credit, you will not decide the steps in the demo for this step. Instead, the TA will try to walk through it herself, simply telling you, via zoom, what to do next. To support her in this effort, you can provide her not more than 1/2 page of instructions, and she should be able to figure out what to do (e.g., if it's easy to forget to do things in the right order, and we don't have proper instructions or receive proper prompts, this will affect your grade).

Step 8: [Optional: but required for an A- grade] Design a way to communicate the information that results from Step 6 to the user, via the microbit in addition to reporting it on the screen of the Android device.

Step 9: [Optional: but required for an A grade] Integrate the network switching into the app. That is, the app should directly allow the user to launch the wireless settings page to change the network.

What to submit

Via Canvas, submit a writeup that answers the following questions: (1) challenges you faced in each step; (2) the trade-offs between this device to some actual health app (apart from accuracy and prediction); (3) your (brief) thoughts on, if you were to sell this product, knowing the disadvantages, how would you possibly commercialize it; and (4) at least one advantage and disadvantage of our prediction model?

During or before office hours on Thursday, December 2, demonstrate Step 7 (and 8 and 9, if applicable). [Note, because of the Thanksgiving holiday, this demo is late, and you will likely be demoing Milestone 6 at the same time.]