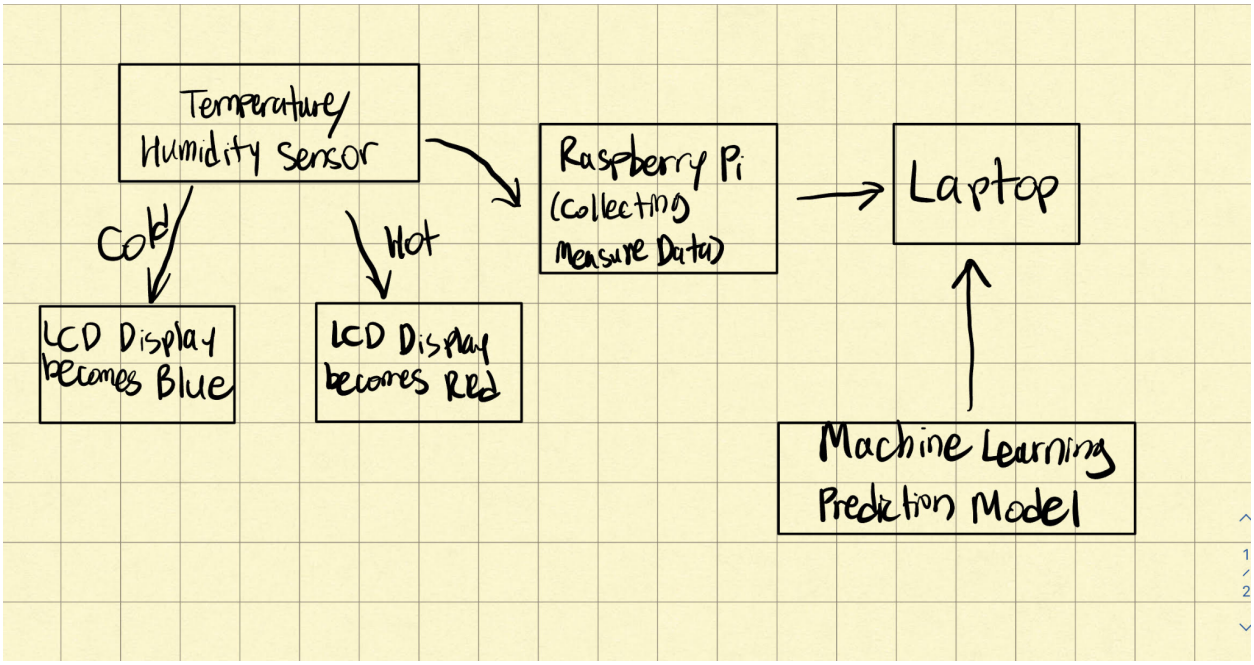
**Team Members: Carrie Lei, Ishraq Rahman**
**Spring 2023**

**EE250 Final Project: *Weather Stylist***

**Project Goal:**

Our project aims to bring together our knowledge from EE250 with our own personal flare for fashion. We created a "weather stylist" that reads in temperature and humidity data, then proceeds to offer outfit ideas for users corresponding to the weather. Given the limited data from testing only on the USC campus, we also utilized a larger database of Los Angeles climate measurements and incorporated machine learning on the set to create a prediction model based on weather features. In short, our creation is designed to offer fun, weather-appropriate outfit ideas *and* to predict the temperature given other weather parameters.

**Block Diagram:**




**Component Description:**

For our hardware component, our two nodes consist of our RaspberryPi and laptop. The RaspberryPi has a temperature sensor and LCD panel connected to it. The temperature sensor reads in the temperature and humidity, and subsequently, the LCD panel will print the sensor's readings and set the background to blue if the temperature falls below our 25 degrees Celsius threshold and to red if the temperature rises above.

The readings from the sensor are also printed on our terminal. Depending on the input readings, we will also suggest outfit ideas to further interact with the user. For example, womens' outfit ideas for warm weather are:

*"skirts, crop tops, sundresses, sunglasses, bucket hats, caps, tank tops, swimwear, sandals, denim jackets, cardigans, light jeans, denim shorts … Make sure you stay hydrated and have sunscreen on."*

For mens' outfit ideas in cold weather, we suggested:

*"wear coats, jackets, bombers, parkas, jeans and warm pants, boots, scarves, beanies, gloves, and hoodies … Make sure you stay warm, or else you'll freeze to death."*

For our Machine Learning model, we used a dataset found online that describes Los Angeles hourly weather over ~5 years between 2012 and 2017. The dataset can be found from Kaggle here, and I trimmed the data to just the last 3 months of 2017 which yielded ~2000 data points. The original dataset had ~50000 data points which was too large for my computer to run. We used data visualization techniques learned in class (namely matplotlib, numpy, and pandas) to plot the temperature over time and used the Linear Regression model from the sklearn library to create a multiple regression model that could predict temperature given user-inputted weather parameters. Specifically, we wanted to further the user interface portion by allowing users to enter values for humidity, pressure, windspeed, and month of the year; then the model would return the predicted temperature given those conditions.

Ideally, the hardware component of our system would have a lot of data points about temperature and humidity; but given the nature of this project, we used an online dataset to simulate how the readings could be used to predict future temperatures.

**Reflections:**
Through this project, we both felt that we were able to strengthen our hardware and software skills. We enjoyed getting the hands-on experience of piecing together our RaspberryPi and mounting the components because it provided a tangible aspect; seeing it work was a rewarding feeling. Moreover, we learned through a lot of trial and error how to use large databases and implement our own prediction models. Initially, our model wasn't able to be "fitted" to the original dataset because of the large volume of data points, so we quickly adapted and switched to a smaller set and used a multiple regression model rather than the initial Random Forest Classification model so that our computer had the memory capacity to run the program. While this process of trial and error was a tad frustrating, we got to read and try out multiple types of Machine Learning models along the way. Additionally, we think this project helped cement our understanding of how devices can connect and communicate with each other and its many day-to-day applications: a nice wrap-up of the EE250 course.

**X - Factor:**
We think the largest "X - Factor" to our project was that we really wanted to emphasize the user interaction process. Namely, they're able to receive an outfit idea given their location *and* enter the weather parameters to predict the temperature given those settings. When applied in the real world, think about: *how cool it would be to have a wearable device on your wrist per se, and the recorded temperature information from the device could be sent to your phone and generate outfit models (given additional UX design)?* This could save us a lot of time in the morning. Moreover, once the wearable device collects enough data with time (patterns in humidity, wind speed, pressure, and time of year), it can be used to even predict hourly temperature changes later in the day and suggest clothing changes accordingly.