Abhi Pote Shrestha
Senior Design 2
Professor Dow
July 26, 2022

Task 9 - Tests and Results

- This test involves testing the model to see if it accurately classifies new temperatures.

Explain why this test is important for verifying that the design (or a module of the design) will work properly.

- One of the main goals of this project involves detecting problematic temperatures. And this test checks to see if the statistical model that we built is accurate and can correctly classify temperatures that are outside the normal range.

Explain what the test is. This often involves diagrams, paragraphs of text, and a procedure (series of steps to be done).

- The statistical model is built using temperature data from the last 5 days. This data comes from the room alert monitor set up in Dobbs 202. The sensor we used for this test is the External Sensor 1.
- The metrics we have calculated are as follow:

Metric	Value
Median	81.86
Standard Deviation	1.794

Lower Bound	78.271
Upper Bound	85.449
90th percentile	83.12
10th percentile	78.296
75th percentile	82.76
25th percentile	80.42

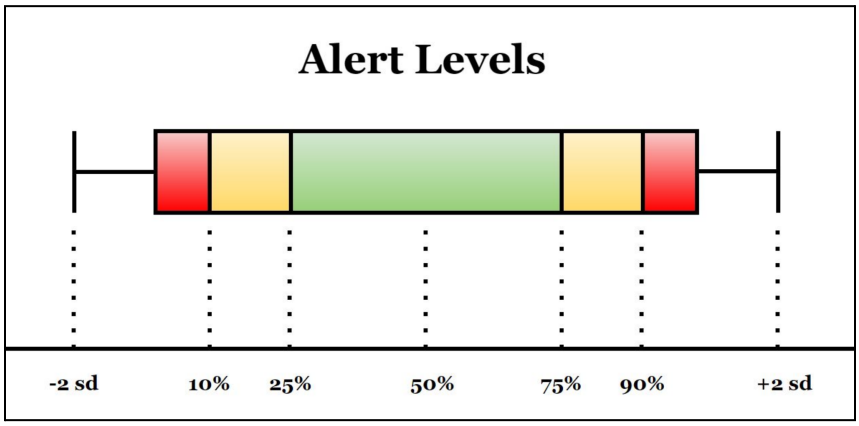

Figure 1: Alert levels - green, yellow, and red

What are the key variables, and will any be varied between trials.

- The key variable is the incoming temperature. We read temperatures for today (August 2) and checked to see where each temperature lied in the range calculated above. Based on where the temperature lied, we classify them through different alert levels:
    - Green alert - temperature is within 25th and 75th percentiles

- Yellow alert - temperature is within 10th and 90th percentiles, but outside the range of the green alert
- Red alert - temperature is beyond the 10th and 90th percentiles

Repeat at least 3 times for the same parameter values.

1. Test 1

- Temperature = 80.42 F
- Green alert because this temperature is within range for the green alert

2. Test 2

- Temperature = 82.94 F
- Yellow alert because this temperature is within range for the yellow alert

3. Test 3

- Temperature = 83.66 F
- Red alert because this temperature is beyond the 10th and 90th percentiles

```
Alert for 80.42 F - GREEN
Alert for 82.94 F - YELLOW
Alert for 83.66 F - RED
```

Figure 2: Output of code used for the testing module

Explain what is measured, when in the series of steps, and how measured.

- First, we collected temperature data for today (08/02) and selected three values that are not equal or close to each other.

- We took each temperature data and compared it with the percentile ranges of temperature data from the last 5 days.
- Depending on where the new data point falls, we classified it into 1 of 3 categories.
- We repeated this process for two other temperatures.
- Temperature used here is actual temperature collected through AVTech's room alert monitor.

Code snippets:

```
metrics = {}
metrics["median"] = np.percentile(temperatures, 50)
metrics["sd"] = np.std(temperatures)
metrics["lower_bound"] = metrics["median"] - 2 * metrics["sd"]
metrics["upper_bound"] = metrics["median"] + 2 * metrics["sd"]
metrics["ninety_pct"] = np.percentile(temperatures, 90)
metrics["ten_pct"] = np.percentile(temperatures, 10)
metrics["seventy_five_pct"] = np.percentile(temperatures, 75)
metrics["twenty_five_pct"] = np.percentile(temperatures, 25)
```

Code snippet 1: Calculate metrics for existing temperature data

```python
def classify_temp(temp, metrics):
    # unpacking metrics
    median = metrics["median"]
    sd = metrics["sd"]
    lower_bound = metrics["lower_bound"]
    upper_bound = metrics["upper_bound"]
    ninety_pct = metrics["ninety_pct"]
    ten_pct = metrics["ten_pct"]
    seventy_five_pct = metrics["seventy_five_pct"]
    twenty_five_pct = metrics["twenty_five_pct"]

    if temp < lower_bound or temp > upper_bound:
        return "Out of Bounds"
    elif temp < ten_pct or temp > ninety_pct:
        return "Red"
    elif temp < twenty_five_pct or temp > seventy_five_pct:
        return "Yellow"
    else:
        return "Green"
```

Code Snippet 2: Classify incoming temperature by determining the range it falls into

Summarize what the results mean. Do they show that this module should work as desired for the design? If not, suggest possible modifications to improve.

- The module works as expected. The statistical model accurately classifies actual temperature data into its expected category (red, yellow, or green). Even after using made up temperature values, such as 110F or 32F (not likely to be observed inside a room), the model classifies such extreme temperatures accurately.

```
Alert for 110 F - MAGENTA
Alert for 32 F - MAGENTA
```

Figure 3: Extreme temperatures such as 110F and 32F are classified with the color magenta which indicates that the temperature is out of bounds (beyond the minimum and maximum bounds set by the model)