INTERNET OF THINGS AND MACHINE LEARNING

A SUMMER TRAINING REPORT

Submitted by

JATIN GAUTAM

Enrollment Number: 03414802818

Electronics and Communication Engineering

Under the supervision of

Mr. Pranav Pai Vernekar

CEO, Bolt IoT

And

Mr. Vinayak Joshi

Mentor, Bolt IoT

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

ROHINI, NEW DELHI

FINAL PROJECT: Capstone Project

Email alert system for some thresholds temperature and Z-score analysis to detect the anomaly.

Project Objectives :

1. Build the circuit for temperature monitoring system, using the Bolt and LM35 sensor.

2. Create a product on the Bolt Cloud, to monitor the data from the LM35, and link it to your Bolt.

3. Write the product code, required to run the polynomial regression algorithm on the data sent by Bolt.

4. Keep the temperature monitoring circuit inside your fridge with the door of the fridge closed, and let the system record the temperature readings for about 2 hours.

5. Using the reading that you received in the 2 hours, set boundaries for the temperature within the fridge.

6. Write a python code which will fetch the temperature data, every 10 seconds, and send out an email alert, if the temperature goes beyond the temperature thresholds you decided on in Objective "E".

7. Modify the python code, to also do a Z-score analysis and print the line "Someone has opened the fridge door" when an anomaly is detected.

8. Tune the Z-score analysis code, such that, it detects an anomaly when someone opens the door of the fridge.

Equipment's Used :

1. BOLT IOT WIFI MODULE

2. Temperature Sensor (LM35 Sensor)

3.Male/Female Jumper Wires

4. LM35 Sensor

5.USB Cable

Software Apps and Online Services :

1.BOLT IOT BOLT CLOUD

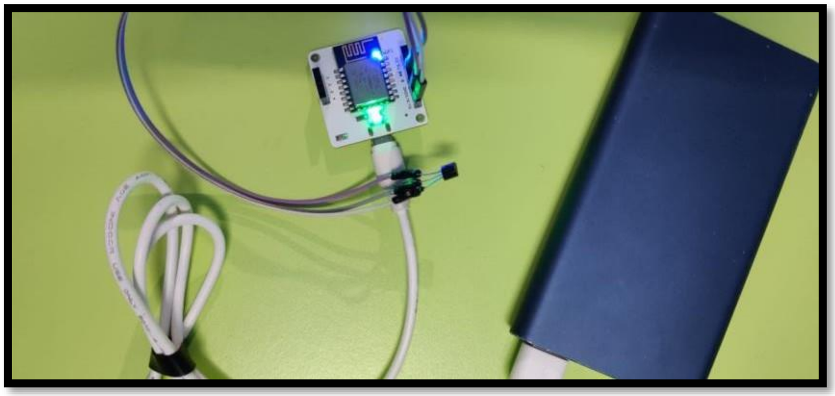
2.BOLT IOT Android App

3. MAILGUN

4.Snappy Ubuntu Core

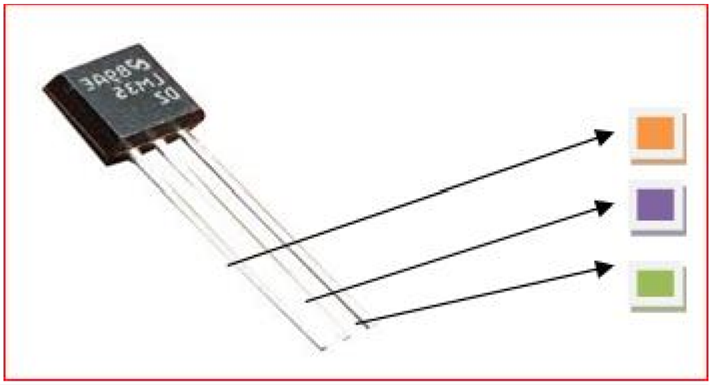
CIRCUIT CONNECTIONS

For circuit connections we are using BOLT IOT WIFI MODULE , LM35 Sensor , Male/Female Jumper Wires , USB Cable and Power Bank for supplying the current.

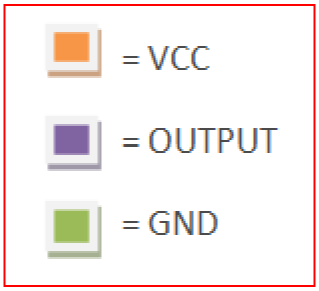


LM35 SENSOR

LM35 Sensor has three terminals - supply, output and ground. It is a temperature monitoring sensor and it can be used in Pharmaceutical companies to maintain the temperature of the medicines .



Where,



🟧 = VCC

🟪 = OUTPUT

🟩 = GND

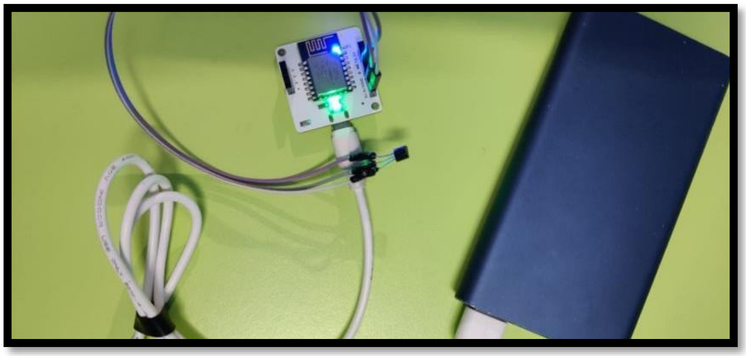
1. VCC pin of the LM35 connects to 5V of the BOLT WIFI MODULE.

2. Output pin of the LM35 connects to A0 (Analog input pin) of the BOLT WIFI MODULE.

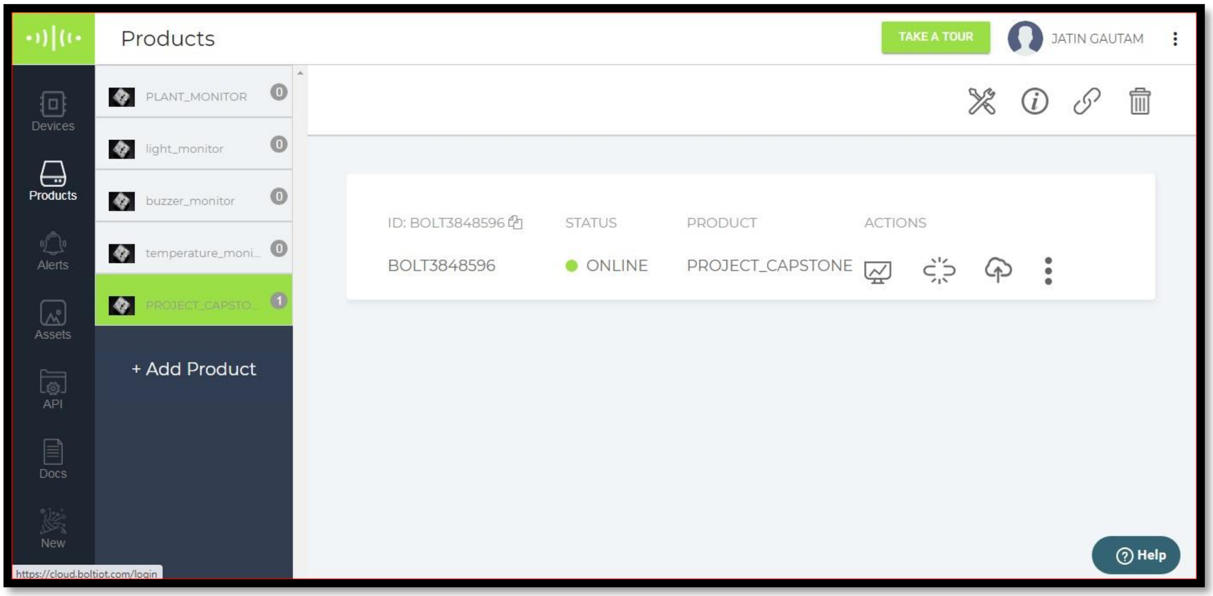
3.GND pin of the LM35 connects to the GND of the BOLT WIFI MODULE.

SCHEMATICS:

A. Build the circuit for temperature monitoring system, using the Bolt and LM35 sensor.



B. Create a product on the Bolt Cloud, to monitor the data from the LM35, and link it to your Bolt.

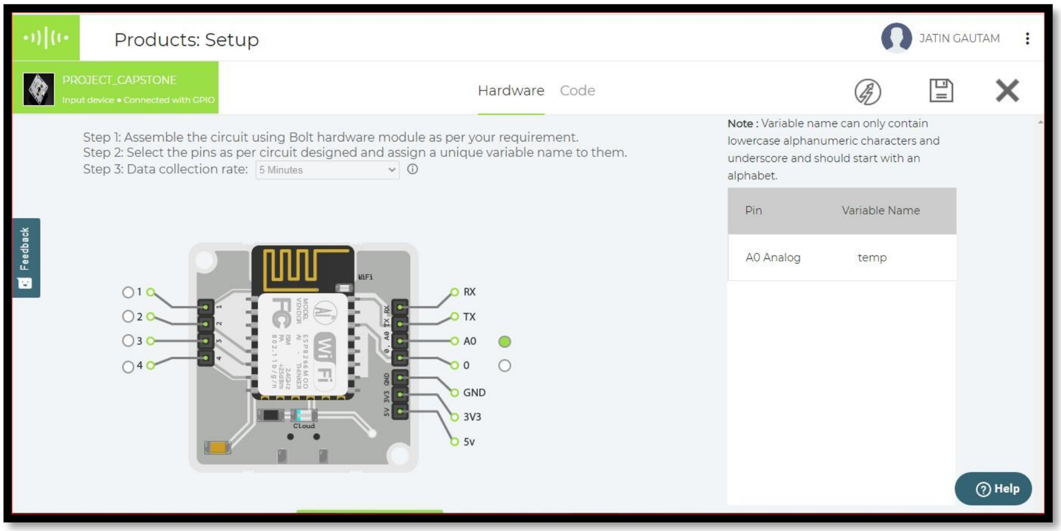


ID: BOLT3848596	STATUS	PRODUCT	ACTIONS
BOLT3848596	● ONLINE	PROJECT_CAPSTONE	

In the above image, we are creating a new Product named as Project_Capstone..Than we are linking this product to BOLT WIFI MODULE.

C. Write the product code, required to run the polynomial regression algorithm on the data sent by the Bolt.

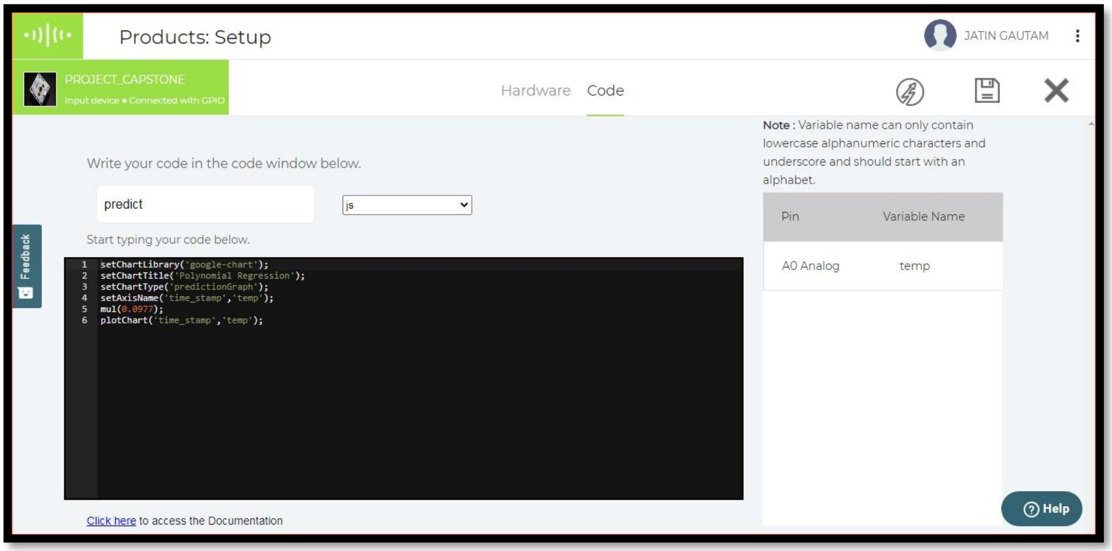
By taking the effective measures , Mr. Ram managed to satisfy the first condition set by the Government. Using the prediction data, he was able to take early action, whenever the graph predicted that the temperature would be maintained within the -33 and -30 degrees Celsius range for longer than 20 minutes.



Step 1: Assemble the circuit using Bolt hardware module as per your requirement.
Step 2: Select the pins as per circuit designed and assign a unique variable name to them.
Step 3: Data collection rate: 5 Minutes

Note : Variable name can only contain lowercase alphanumeric characters and underscore and should start with an alphabet.

Pin	Variable Name
A0 Analog	temp



Write your code in the code window below.

predict   js

Start typing your code below.

```
setChartLibrary('google-chart');
setChartTitle('Polynomial Regression');
setChartType('predictionGraph');
setAxisName('time_stamp','temp');
mul(0.0977);
plotChart('time_stamp','temp');
```

Click here to access the Documentation

Note : Variable name can only contain lowercase alphanumeric characters and underscore and should start with an alphabet.

Pin	Variable Name
A0 Analog	temp

Similarly, In the above image shown we are writing the code to run the Polynomial Regression on the data sent by the BOLT.

CODE:

```
setChartLibrary('google-chart');

setChartTitle('Polynomial Regression');

setChartType('predictionGraph');

setAxisName('time_stamp','temp');

mul(0.0977);

plotChart('time_stamp','temp');
```

Polynomial Regression:

Polynomial Regression is a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modeled as an nth degree polynomial.

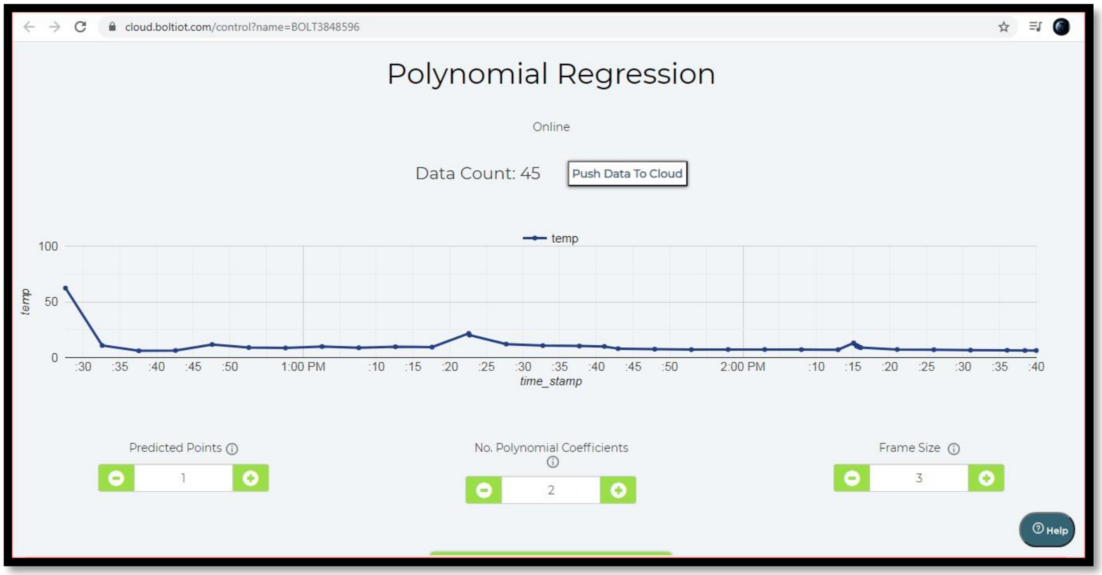
Parameters used in Polynomial Regression:

1. Prediction points

2. Number of polynomial coefficients

3. Frame Size

$$data(t) = (C_n * t^n) + (C_{n-1} * t^{n-1}) + (C_{n-2} * t^{n-2}) + \ldots\ldots + (C_1 * t^1) + C_0$$


cloud.boltiot.com/control?name=BOLT3848596

Polynomial Regression

Online

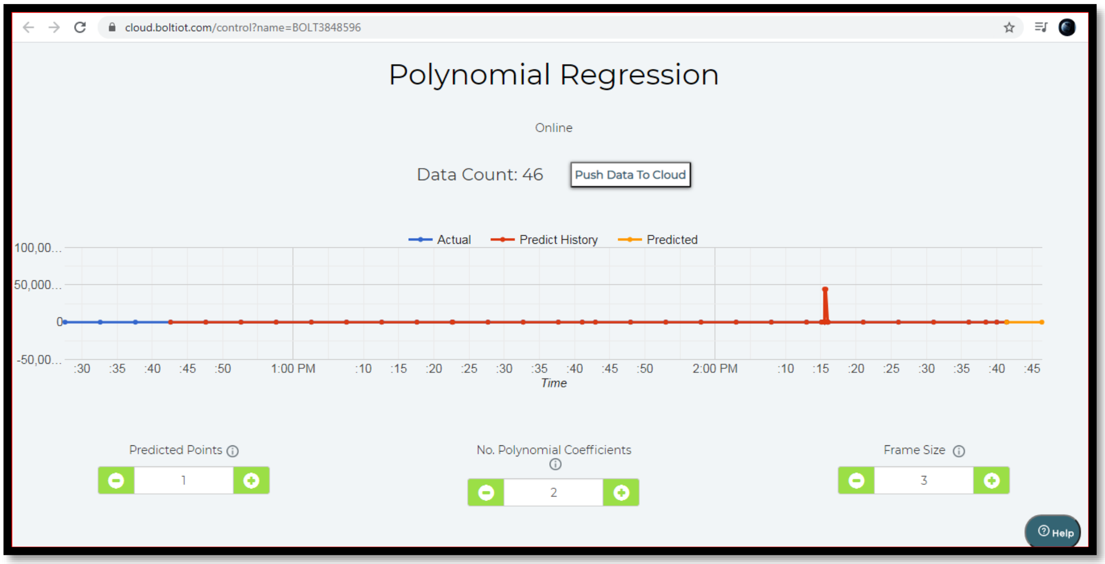
Data Count: 45

Predicted Points: 1

No. Polynomial Coefficients: 2

Frame Size: 3

After applying Polynomial Regression:


cloud.boltiot.com/control?name=BOLT3848596

Polynomial Regression

Online

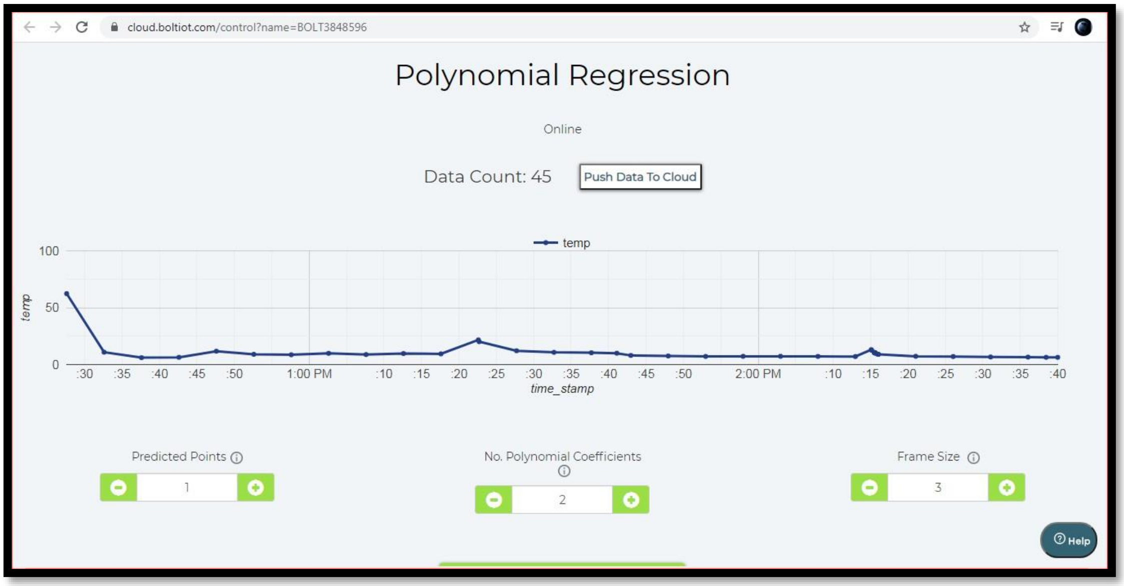
Data Count: 46

Predicted Points: 1

No. Polynomial Coefficients: 2

Frame Size: 3

D. Keep the temperature monitoring circuit inside your fridge with the door of the fridge closed, and let the system record the temperature readings for about 2 hours.



Data Count: 45

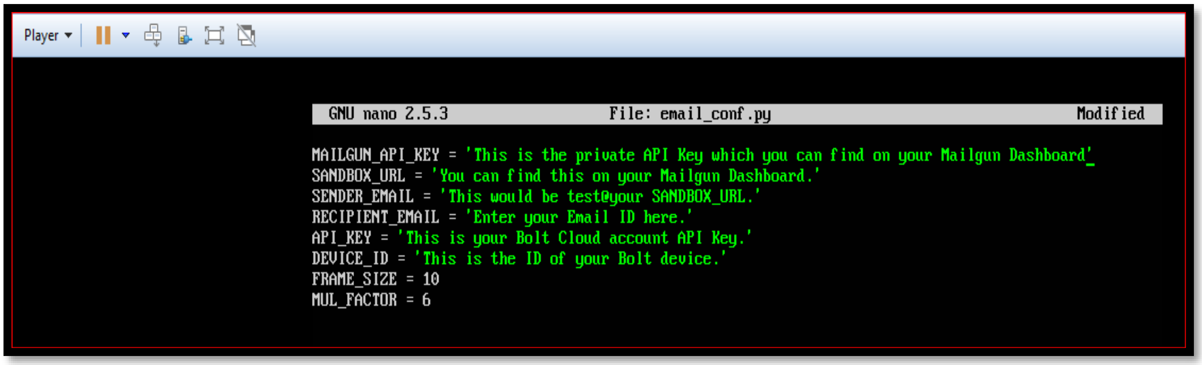
Predicted Points: 1

No. Polynomial Coefficients: 2

Frame Size: 3

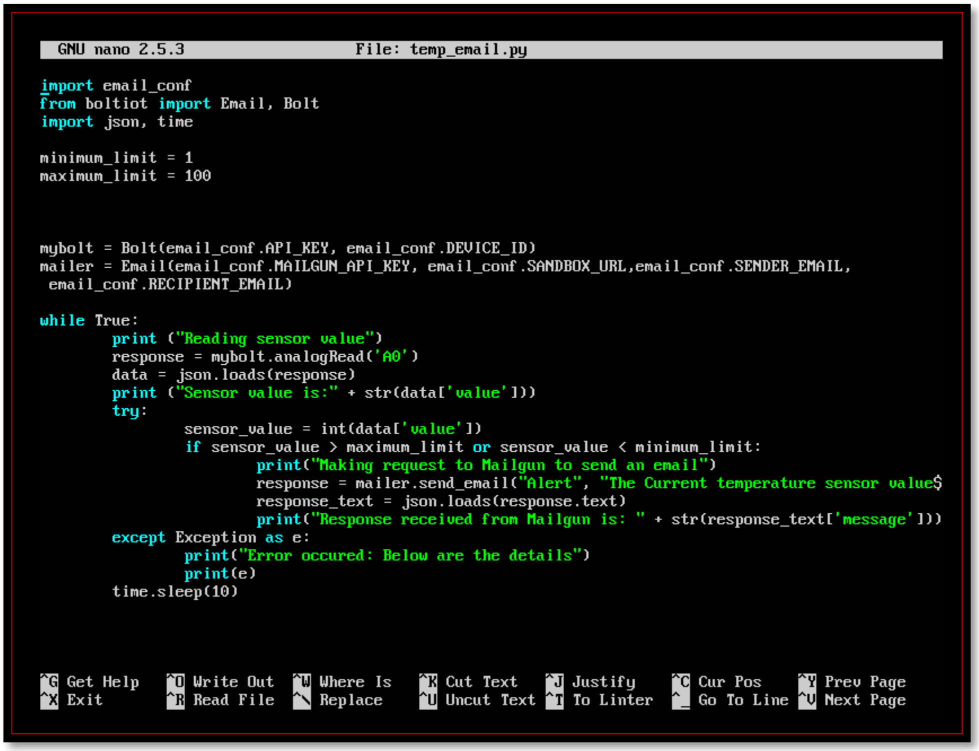
E. Using the reading that you received in the 2 hours, set boundaries for the temperature within the fridge

STEP 1: Create a filename sudo nano email_conf.py to store the credentials.



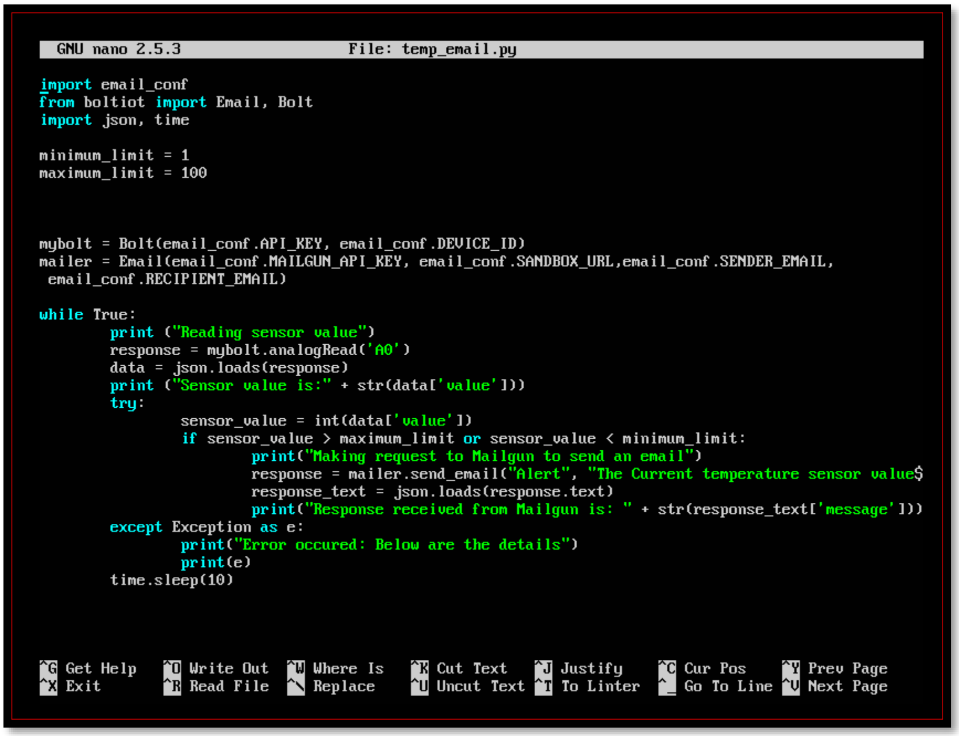
```
GNU nano 2.5.3                    File: email_conf.py                        Modified

MAILGUN_API_KEY = 'This is the private API Key which you can find on your Mailgun Dashboard'
SANDBOX_URL = 'You can find this on your Mailgun Dashboard.'
SENDER_EMAIL = 'This would be test@your SANDBOX_URL.'
RECIPIENT_EMAIL = 'Enter your Email ID here.'
API_KEY = 'This is your Bolt Cloud account API Key.'
DEVICE_ID = 'This is the ID of your Bolt device.'
FRAME_SIZE = 10
MUL_FACTOR = 6
```

STEP 2: Create another filename sudo nano temp_email.py for sending emails when temperature crosses its threshold.



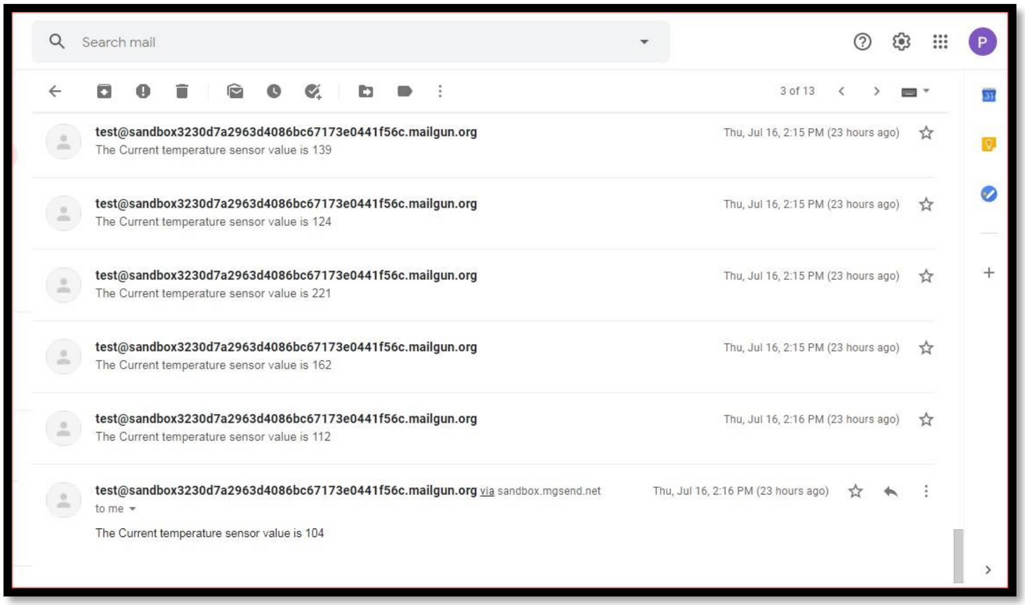
```
import email_conf
from boltiot import Email, Bolt
import json, time

minimum_limit = 1
maximum_limit = 100



mybolt = Bolt(email_conf.API_KEY, email_conf.DEVICE_ID)
mailer = Email(email_conf.MAILGUN_API_KEY, email_conf.SANDBOX_URL,email_conf.SENDER_EMAIL,
 email_conf.RECIPIENT_EMAIL)

while True:
        print ("Reading sensor value")
        response = mybolt.analogRead('A0')
        data = json.loads(response)
        print ("Sensor value is:" + str(data['value']))
        try:
                sensor_value = int(data['value'])
                if sensor_value > maximum_limit or sensor_value < minimum_limit:
                        print("Making request to Mailgun to send an email")
                        response = mailer.send_email("Alert", "The Current temperature sensor value$
                        response_text = json.loads(response.text)
                        print("Response received from Mailgun is: " + str(response_text['message']))
        except Exception as e:
                print("Error occured: Below are the details")
                print(e)
        time.sleep(10)
```

F. Write a python code which will fetch the temperature data, every 10 seconds, and send out an email alert, if the temperature goes beyond the temperature thresholds you decided on in objective "E".



```
import email_conf
from boltiot import Email, Bolt
import json, time

minimum_limit = 1
maximum_limit = 100



mybolt = Bolt(email_conf.API_KEY, email_conf.DEVICE_ID)
mailer = Email(email_conf.MAILGUN_API_KEY, email_conf.SANDBOX_URL,email_conf.SENDER_EMAIL,
 email_conf.RECIPIENT_EMAIL)

while True:
        print ("Reading sensor value")
        response = mybolt.analogRead('A0')
        data = json.loads(response)
        print ("Sensor value is:" + str(data['value']))
        try:
                sensor_value = int(data['value'])
                if sensor_value > maximum_limit or sensor_value < minimum_limit:
                        print("Making request to Mailgun to send an email")
                        response = mailer.send_email("Alert", "The Current temperature sensor value$
                        response_text = json.loads(response.text)
                        print("Response received from Mailgun is: " + str(response_text['message']))
        except Exception as e:
                print("Error occured: Below are the details")
                print(e)
        time.sleep(10)
```

- **Sending an email alert when temperature crosses its threshold.**



test@sandbox3230d7a2963d4086bc67173e0441f56c.mailgun.org Thu, Jul 16, 2:15 PM (23 hours ago)
The Current temperature sensor value is 139

test@sandbox3230d7a2963d4086bc67173e0441f56c.mailgun.org Thu, Jul 16, 2:15 PM (23 hours ago)
The Current temperature sensor value is 124

test@sandbox3230d7a2963d4086bc67173e0441f56c.mailgun.org Thu, Jul 16, 2:15 PM (23 hours ago)
The Current temperature sensor value is 221

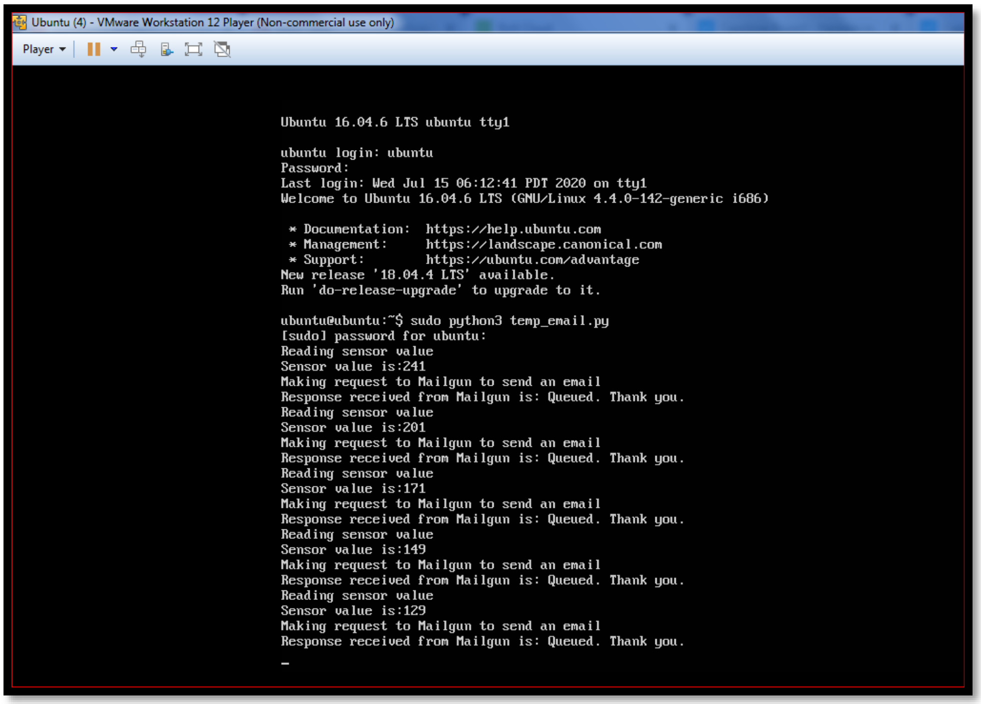
test@sandbox3230d7a2963d4086bc67173e0441f56c.mailgun.org Thu, Jul 16, 2:15 PM (23 hours ago)
The Current temperature sensor value is 162

test@sandbox3230d7a2963d4086bc67173e0441f56c.mailgun.org Thu, Jul 16, 2:16 PM (23 hours ago)
The Current temperature sensor value is 112

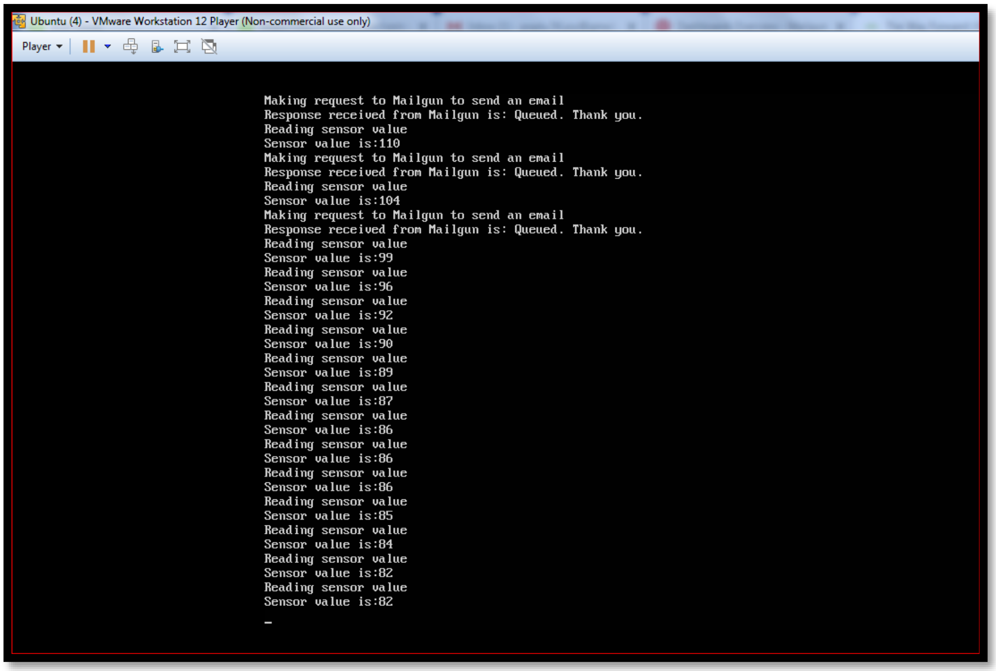
test@sandbox3230d7a2963d4086bc67173e0441f56c.mailgun.org via sandbox.mgsend.net Thu, Jul 16, 2:16 PM (23 hours ago)
to me

The Current temperature sensor value is 104

READINGS :


Ubuntu (4) - VMware Workstation 12 Player (Non-commercial use only)
Player

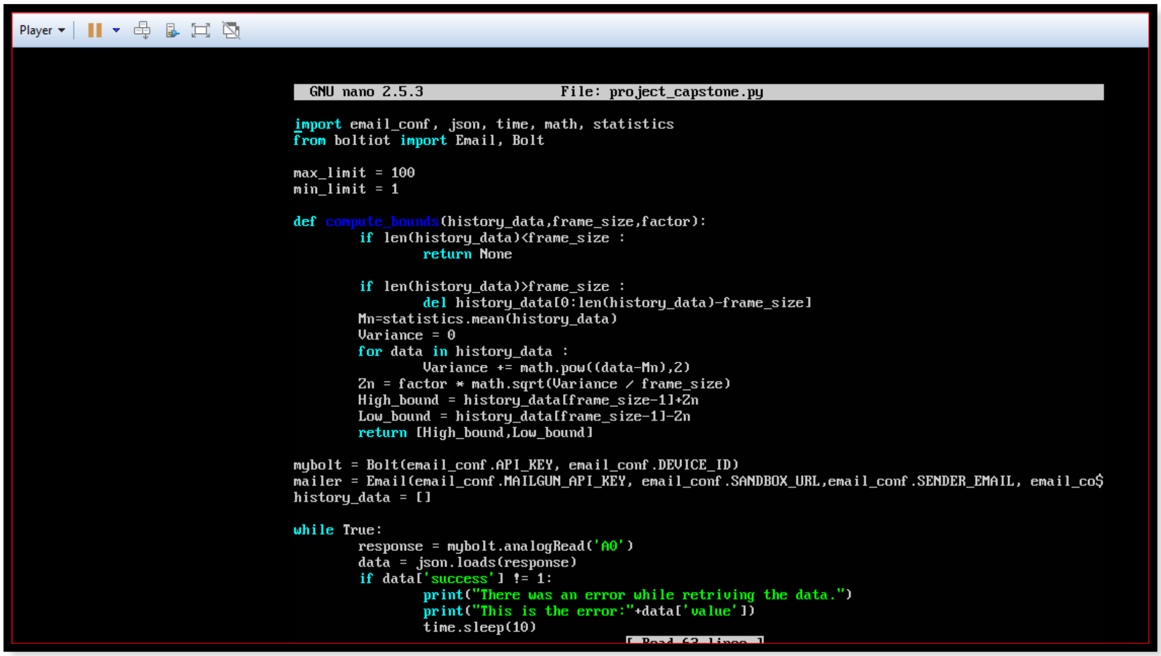
```
Ubuntu 16.04.6 LTS ubuntu tty1

ubuntu login: ubuntu
Password:
Last login: Wed Jul 15 06:12:41 PDT 2020 on tty1
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-142-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

ubuntu@ubuntu:~$ sudo python3 temp_email.py
[sudo] password for ubuntu:
Reading sensor value
Sensor value is:241
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
Reading sensor value
Sensor value is:201
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
Reading sensor value
Sensor value is:171
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
Reading sensor value
Sensor value is:149
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
Reading sensor value
Sensor value is:129
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
_
```


Ubuntu (4) - VMware Workstation 12 Player (Non-commercial use only)
Player

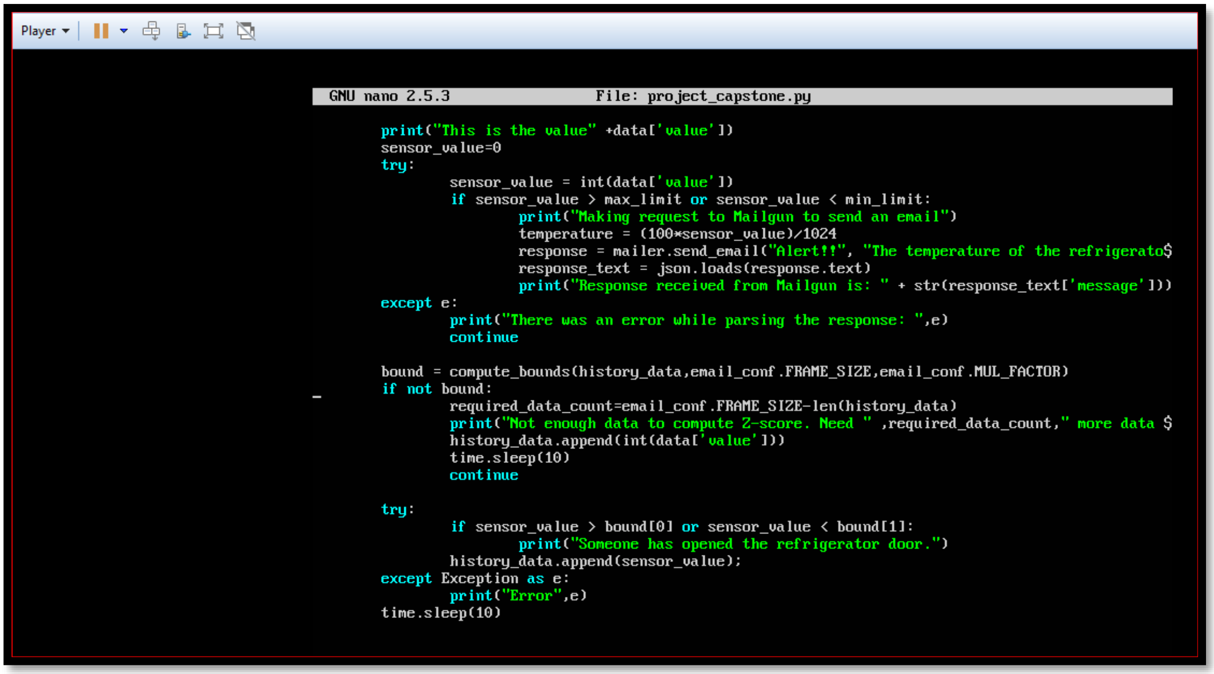
```
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
Reading sensor value
Sensor value is:110
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
Reading sensor value
Sensor value is:104
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
Reading sensor value
Sensor value is:99
Reading sensor value
Sensor value is:96
Reading sensor value
Sensor value is:92
Reading sensor value
Sensor value is:90
Reading sensor value
Sensor value is:89
Reading sensor value
Sensor value is:87
Reading sensor value
Sensor value is:86
Reading sensor value
Sensor value is:86
Reading sensor value
Sensor value is:86
Reading sensor value
Sensor value is:85
Reading sensor value
Sensor value is:84
Reading sensor value
Sensor value is:82
Reading sensor value
Sensor value is:82
_
```

G. Modify the python code, to also do a Z-score analysis and print the line "Someone has opened the fridge door" when an anomaly is detected.

STEP 1: For writing the final code for Z-score analysis ,create another filename sudo nano project_capstone.py .

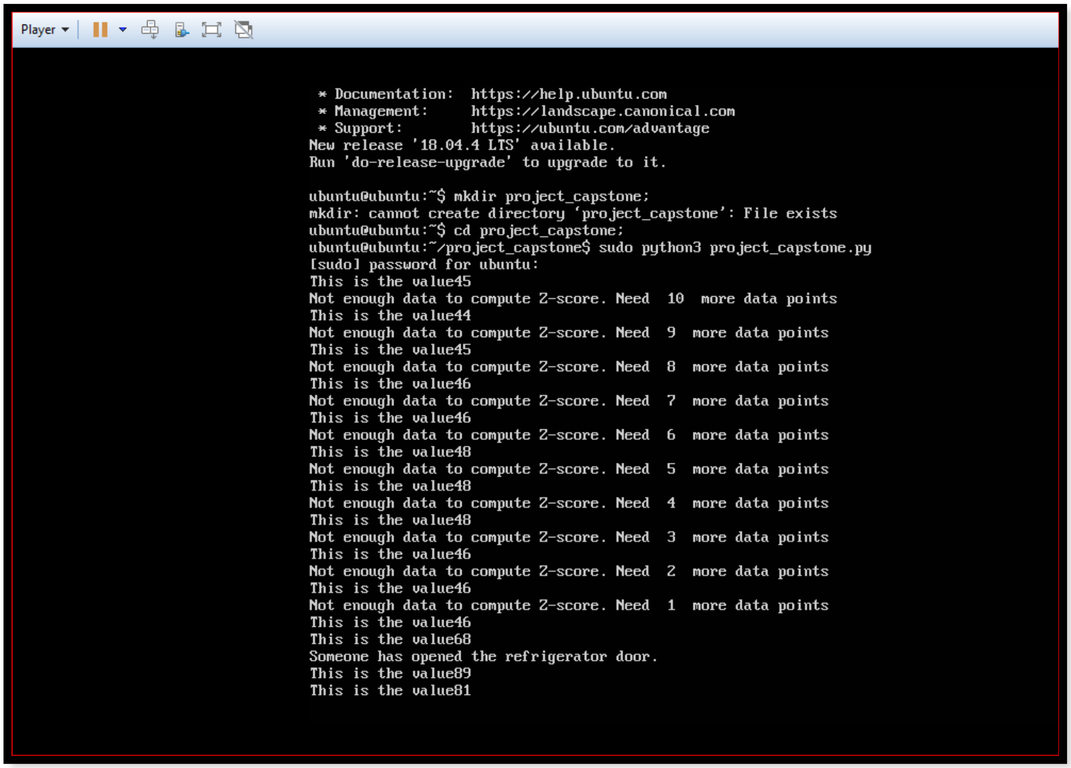


```
  GNU nano 2.5.3                    File: project_capstone.py

import email_conf, json, time, math, statistics
from boltiot import Email, Bolt

max_limit = 100
min_limit = 1

def compute_bounds(history_data,frame_size,factor):
        if len(history_data)<frame_size :
                return None

        if len(history_data)>frame_size :
                del history_data[0:len(history_data)-frame_size]
        Mn=statistics.mean(history_data)
        Variance = 0
        for data in history_data :
                Variance += math.pow((data-Mn),2)
        Zn = factor * math.sqrt(Variance / frame_size)
        High_bound = history_data[frame_size-1]+Zn
        Low_bound = history_data[frame_size-1]-Zn
        return [High_bound,Low_bound]

mybolt = Bolt(email_conf.API_KEY, email_conf.DEVICE_ID)
mailer = Email(email_conf.MAILGUN_API_KEY, email_conf.SANDBOX_URL,email_conf.SENDER_EMAIL, email_co$
history_data = []

while True:
        response = mybolt.analogRead('A0')
        data = json.loads(response)
        if data['success'] != 1:
                print("There was an error while retriving the data.")
                print("This is the error:"+data['value'])
                time.sleep(10)
                                          [ Read 63 lines ]
```

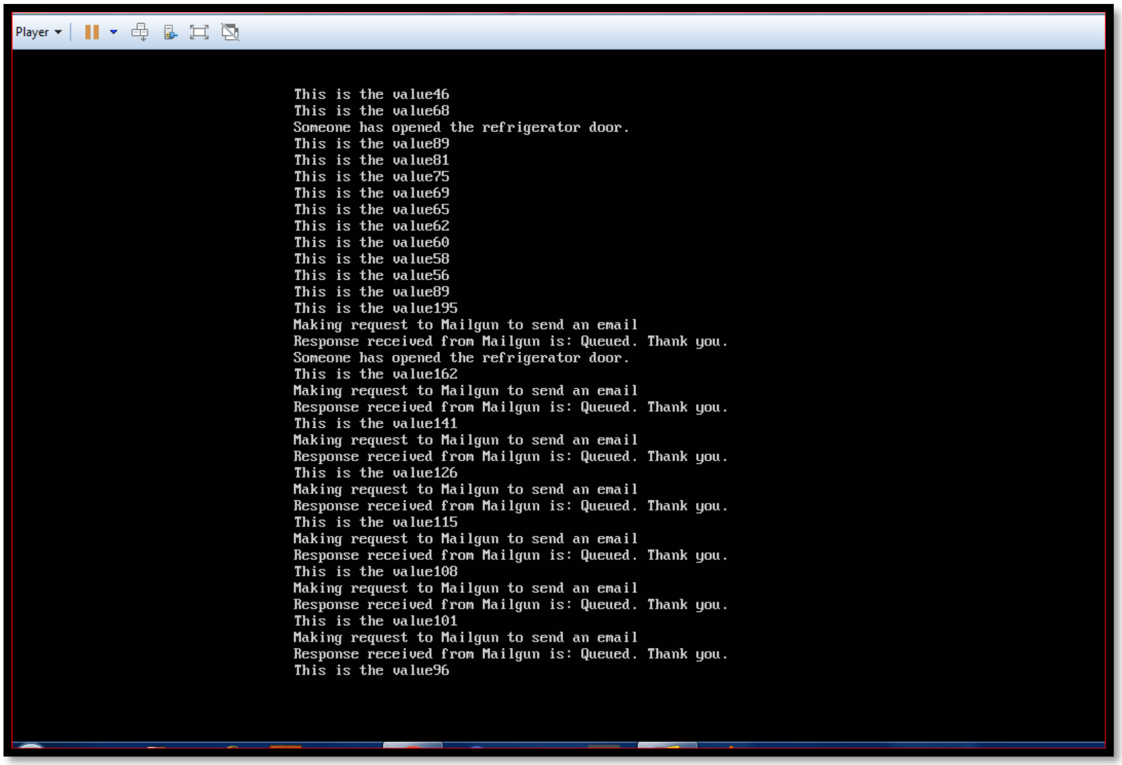


```
 GNU nano 2.5.3                 File: project_capstone.py

        print("This is the value" +data['value'])
        sensor_value=0
        try:
                sensor_value = int(data['value'])
                if sensor_value > max_limit or sensor_value < min_limit:
                        print("Making request to Mailgun to send an email")
                        temperature = (100*sensor_value)/1024
                        response = mailer.send_email("Alert!!", "The temperature of the refrigerato$
                        response_text = json.loads(response.text)
                        print("Response received from Mailgun is: " + str(response_text['message']))
        except e:
                print("There was an error while parsing the response: ",e)
                continue

        bound = compute_bounds(history_data,email_conf.FRAME_SIZE,email_conf.MUL_FACTOR)
        if not bound:
                required_data_count=email_conf.FRAME_SIZE-len(history_data)
                print("Not enough data to compute Z-score. Need " ,required_data_count," more data $
                history_data.append(int(data['value']))
                time.sleep(10)
                continue

        try:
                if sensor_value > bound[0] or sensor_value < bound[1]:
                        print("Someone has opened the refrigerator door.")
                history_data.append(sensor_value);
        except Exception as e:
                print("Error",e)
        time.sleep(10)
```

READINGS :

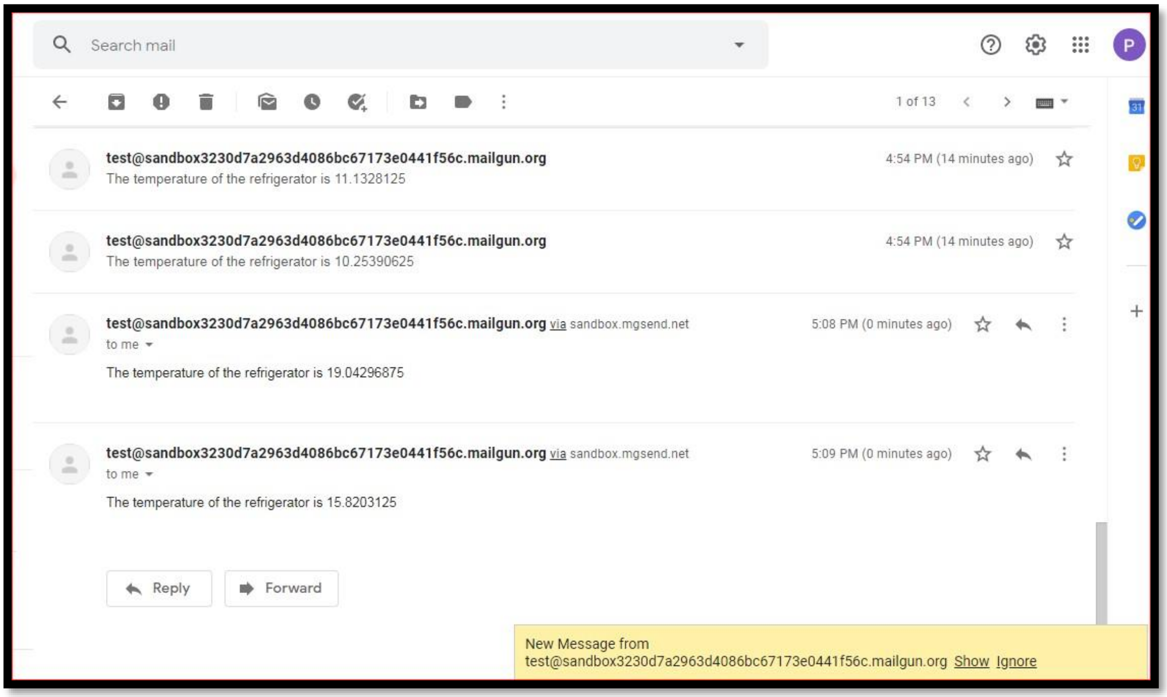


```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

ubuntu@ubuntu:~$ mkdir project_capstone;
mkdir: cannot create directory 'project_capstone': File exists
ubuntu@ubuntu:~$ cd project_capstone;
ubuntu@ubuntu:~/project_capstone$ sudo python3 project_capstone.py
[sudo] password for ubuntu:
This is the value45
Not enough data to compute Z-score. Need  10  more data points
This is the value44
Not enough data to compute Z-score. Need  9  more data points
This is the value45
Not enough data to compute Z-score. Need  8  more data points
This is the value46
Not enough data to compute Z-score. Need  7  more data points
This is the value46
Not enough data to compute Z-score. Need  6  more data points
This is the value48
Not enough data to compute Z-score. Need  5  more data points
This is the value48
Not enough data to compute Z-score. Need  4  more data points
This is the value48
Not enough data to compute Z-score. Need  3  more data points
This is the value46
Not enough data to compute Z-score. Need  2  more data points
This is the value46
Not enough data to compute Z-score. Need  1  more data points
This is the value46
This is the value68
Someone has opened the refrigerator door.
This is the value89
This is the value81
```



```
This is the value46
This is the value68
Someone has opened the refrigerator door.
This is the value89
This is the value81
This is the value75
This is the value69
This is the value65
This is the value62
This is the value60
This is the value58
This is the value56
This is the value89
This is the value195
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
Someone has opened the refrigerator door.
This is the value162
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
This is the value141
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
This is the value126
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
This is the value115
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
This is the value108
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
This is the value101
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
This is the value96
```

- **Sending email alert for the temperature of the refrigerator.**



test@sandbox3230d7a2963d4086bc67173e0441f56c.mailgun.org 4:54 PM (14 minutes ago)
The temperature of the refrigerator is 11.1328125

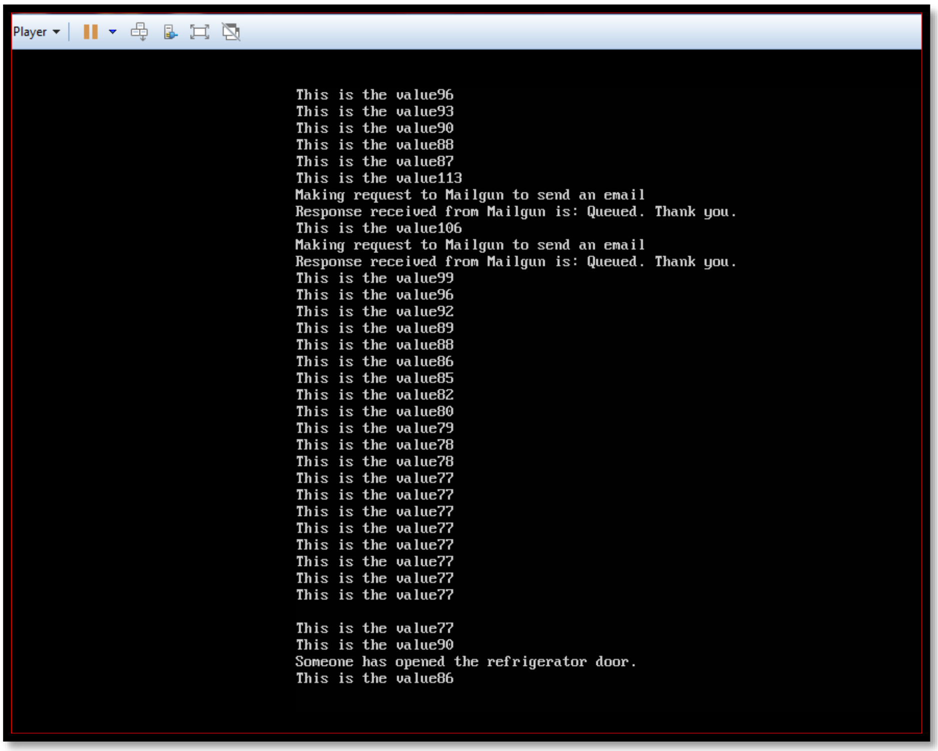
test@sandbox3230d7a2963d4086bc67173e0441f56c.mailgun.org 4:54 PM (14 minutes ago)
The temperature of the refrigerator is 10.25390625

test@sandbox3230d7a2963d4086bc67173e0441f56c.mailgun.org via sandbox.mgsend.net 5:08 PM (0 minutes ago)
to me
The temperature of the refrigerator is 19.04296875

test@sandbox3230d7a2963d4086bc67173e0441f56c.mailgun.org via sandbox.mgsend.net 5:09 PM (0 minutes ago)
to me
The temperature of the refrigerator is 15.8203125

New Message from
test@sandbox3230d7a2963d4086bc67173e0441f56c.mailgun.org Show Ignore

H. Tune the Z-score analysis code, such that, it detects an anomaly when someone opens the door of the fridge.



```
This is the value96
This is the value93
This is the value90
This is the value88
This is the value87
This is the value113
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
This is the value106
Making request to Mailgun to send an email
Response received from Mailgun is: Queued. Thank you.
This is the value99
This is the value96
This is the value92
This is the value89
This is the value88
This is the value86
This is the value85
This is the value82
This is the value80
This is the value79
This is the value78
This is the value78
This is the value77
This is the value77
This is the value77
This is the value77
This is the value77
This is the value77
This is the value77
This is the value77

This is the value77
This is the value90
Someone has opened the refrigerator door.
This is the value86
```