National Textile University, Faisalabad



Department of Computer Science

|  |  |
| --- | --- |
| Name: | Muhammad Hassaan Raza |
| Class: | BSAI-6th |
| Registration No: | 22-NTU-CS-1362 |
| Assignment | Lab13-Lab/Home tasks |
| Course Code: | AIE-3079 |
| Course Name: | Internet of Things Fundamentals |
| Submitted To: | Mr. Nasir Mehmood |
| Submission Date: | 20-May-2025 |

Lab 13 report

# Task 1: Arduino MQTT Publisher for DHT Sensor

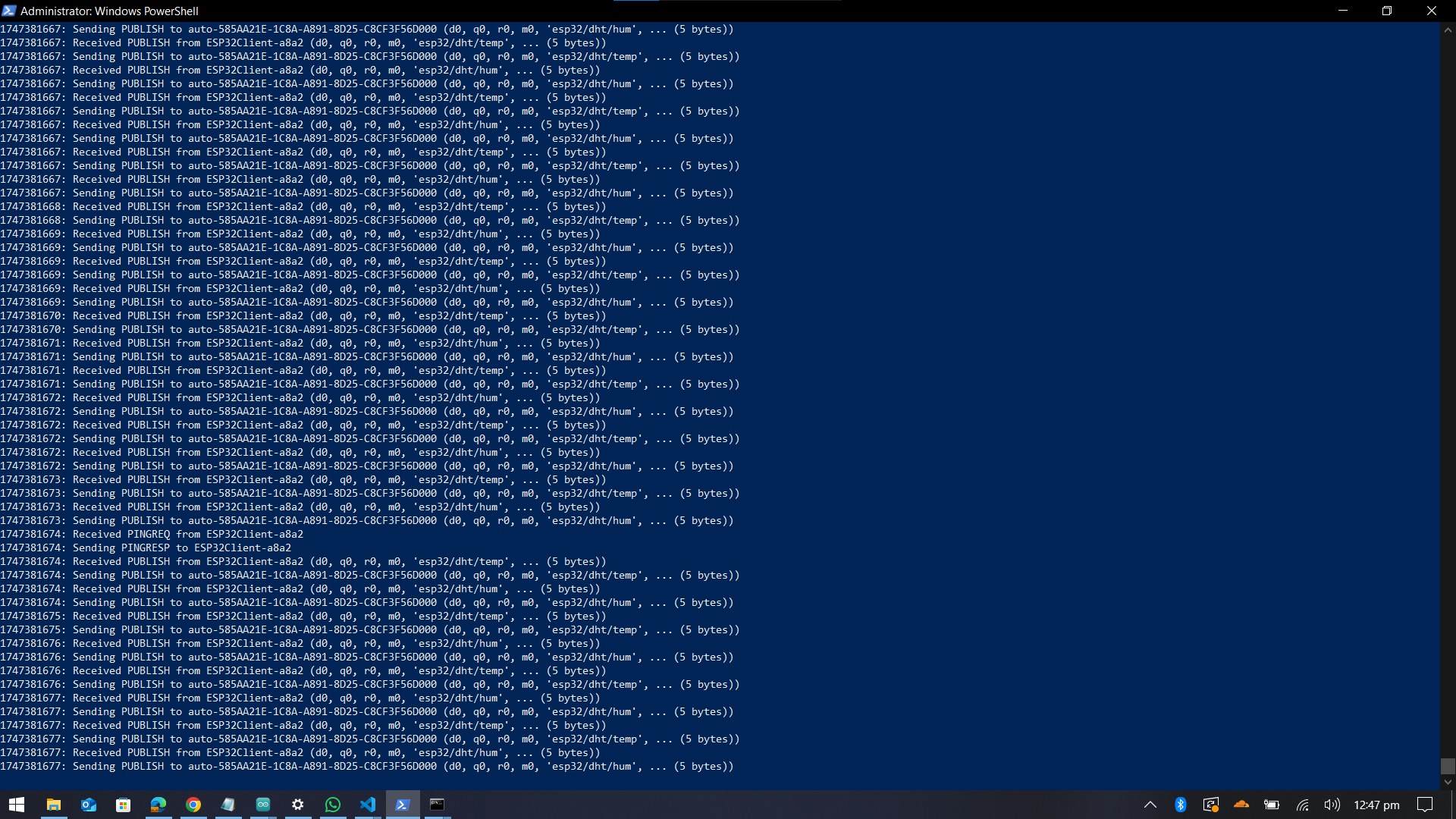
**What I Did:**  
I used Arduino code to send temperature and humidity data from a DHT11 sensor to an MQTT broker (Mosquitto). The ESP32 connects to Wi-Fi and publishes data every second.

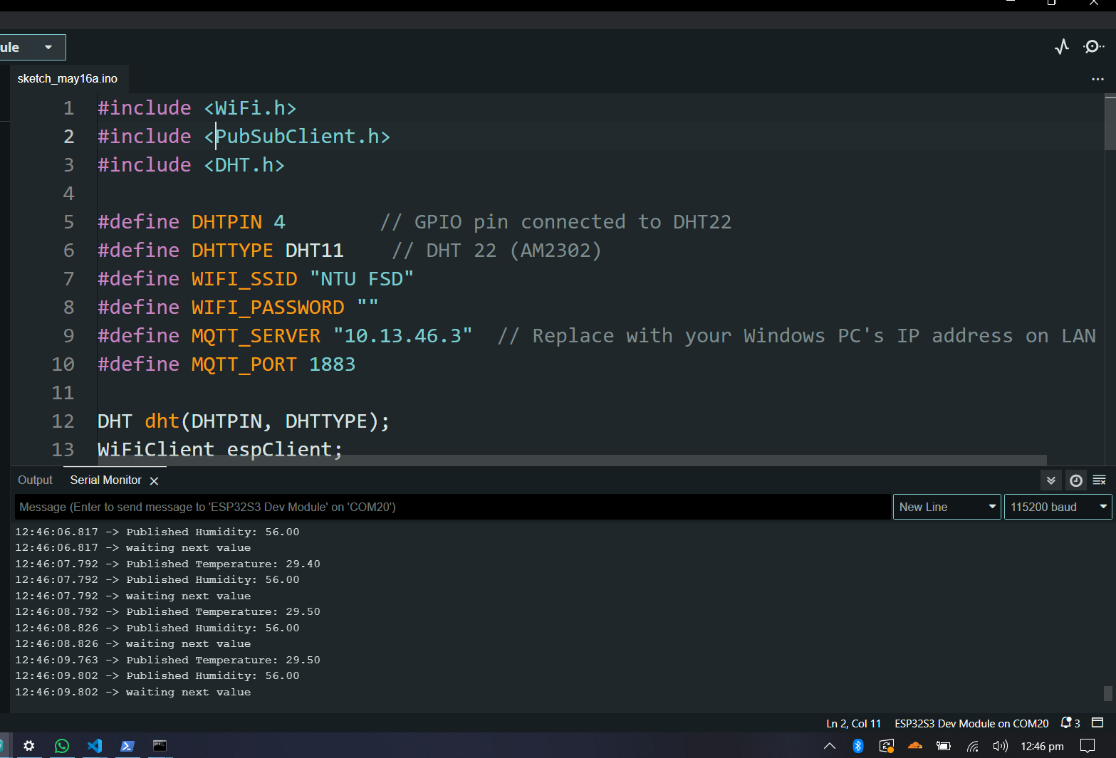
**Results:**

* The Serial Monitor showed live updates like:  
  Published Temperature: 29.40°C  
  Published Humidity: 56.00%
* The data was sent successfully to the topic esp32/dht/temp and esp32/dht/hum.

**Problems Faced:**

* Sometimes, the sensor failed to read values, but the code handled it by printing an error.

**Screenshot Attached:**



# Task 2: Store MQTT Data in InfluxDB

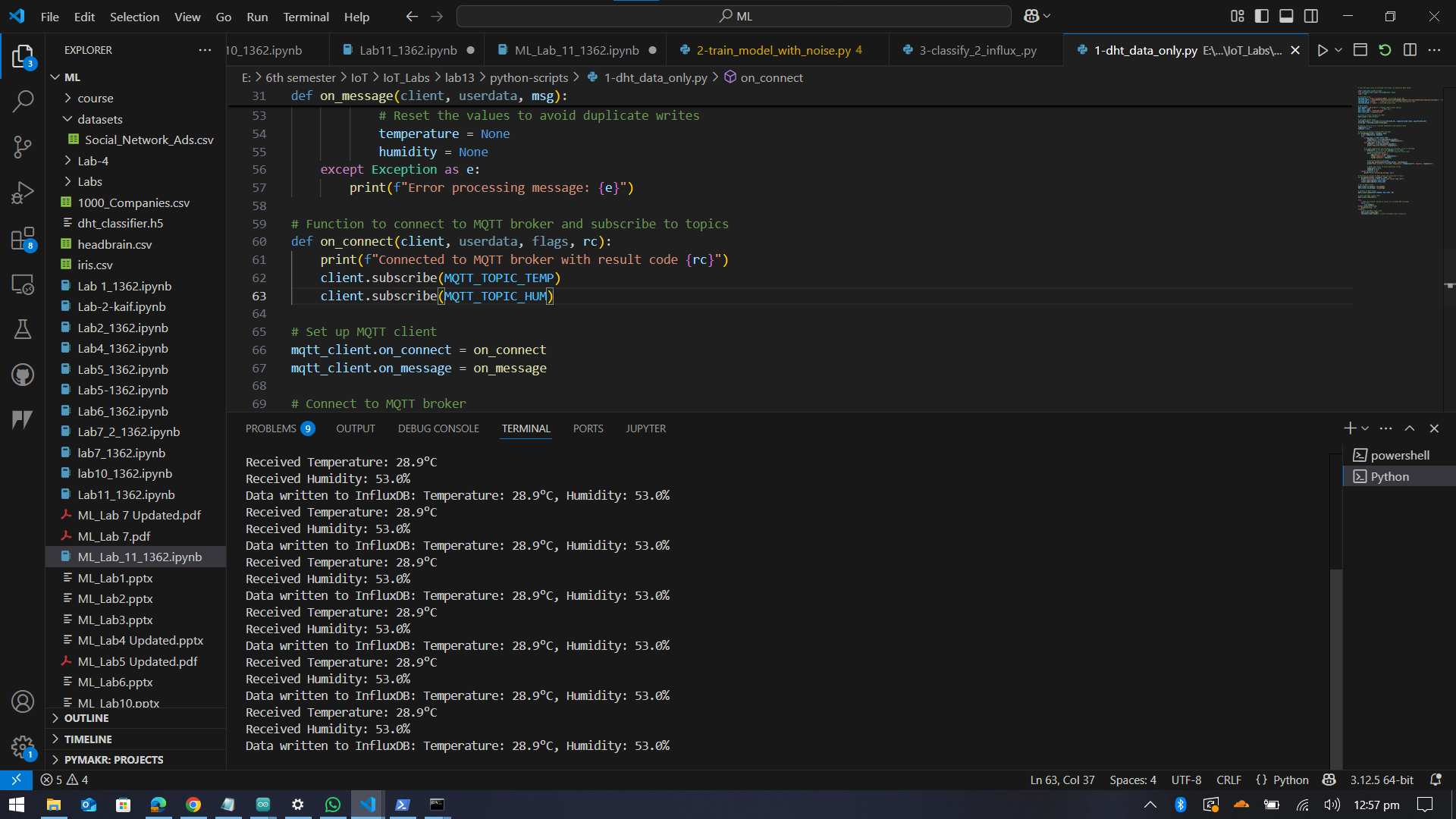
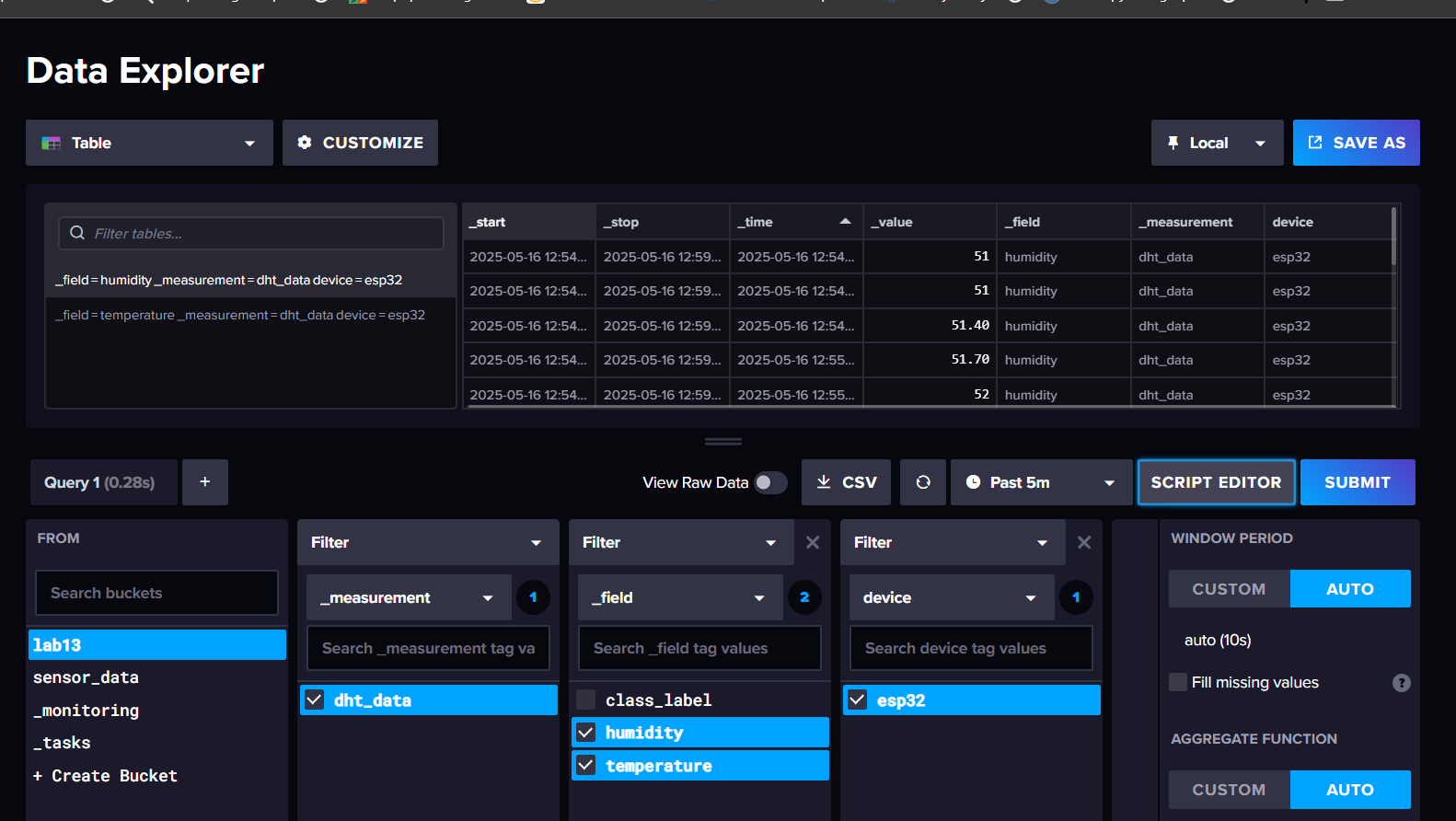
**What I Did:**  
I ran a Python script (1-dht\_data\_only.py) to subscribe to MQTT topics and save the sensor data to InfluxDB.

**Results:**

* The script printed confirmations like:  
  Received Temperature: 29.40°C  
  Data written to InfluxDB: Temperature: 29.40°C, Humidity: 56.00%
* Data was stored in InfluxDB under the measurement dht\_data with tags and fields.

**Problems Faced:**

* First, the script didn’t connect because of a wrong IP address. I fixed it by checking the broker address.

**screenshot Attached:**

# Task 3: Train Model with Noise

**What I Did:**  
I ran 2-train\_model\_with\_noise.py to create a synthetic dataset and train a neural network to classify temperature/humidity into 5 classes (e.g., "Hot and Dry").

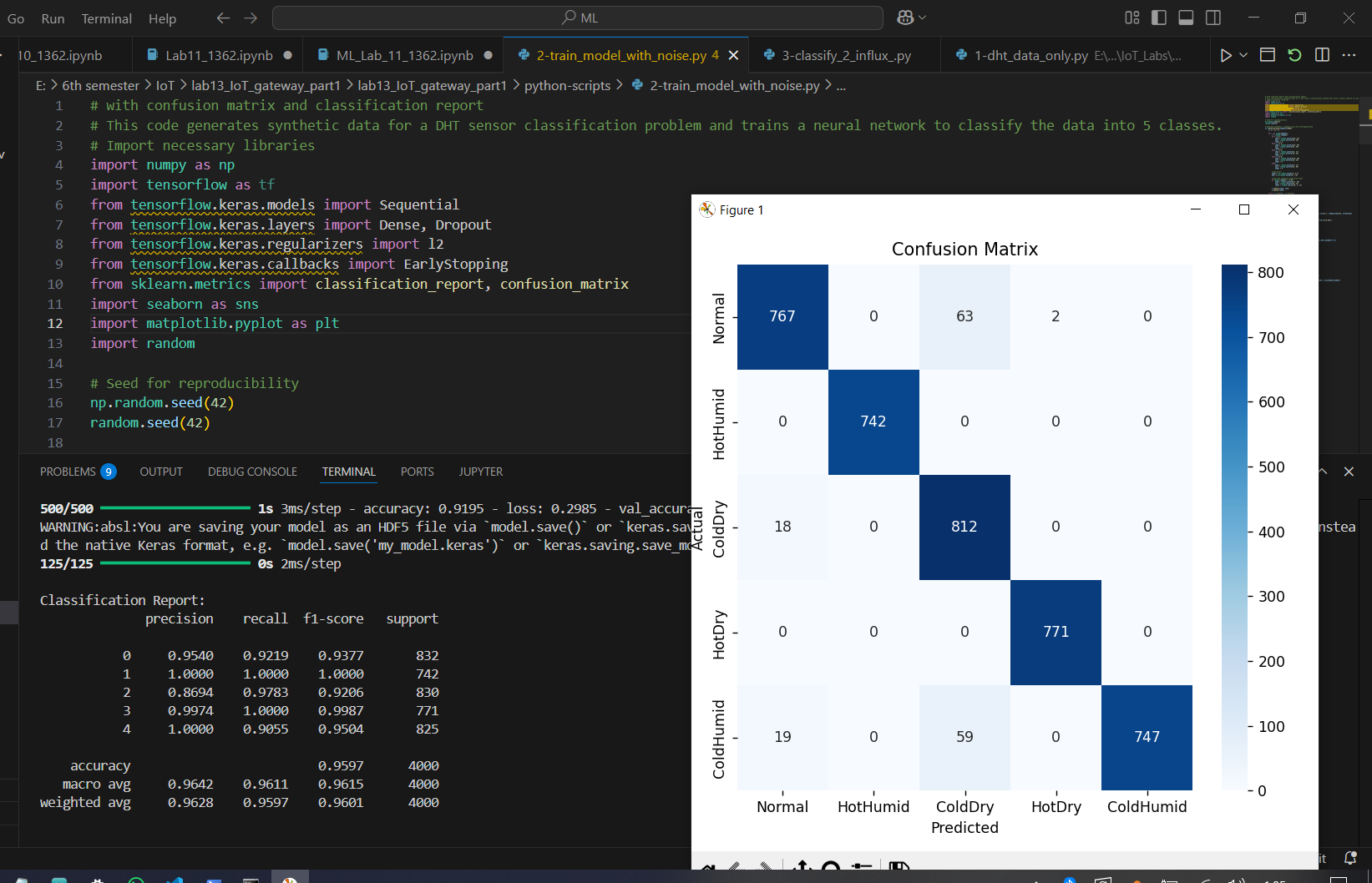
**Results:**

* The model achieved **~92% accuracy** .
* The confusion matrix showed good performance but some confusion between similar classes (e.g., "Normal" and "Cold and Humid").

**Problems Faced:**

* The training was slow on my laptop, but eventually it trained.

**Screenshots Attached:**



# Task 4: Classify and Store in InfluxDB

**What I Did:**  
I executed 3-classify\_2\_influx.py to read live sensor data, classify it using the trained model, and save everything (temperature, humidity, and class label) to InfluxDB.

**Results:**

* The console showed real-time updates:  
  🌡️ Received Temperature: 29.40°C  
  🔍 Predicted Class: Normal  
  ✅ Data saved to InfluxDB
* The InfluxDB dashboard displayed all fields: temperature, humidity, and class\_label.

**Problems Faced:**

* At first, the class labels didn’t appear in InfluxDB because of a typo in the field name. I fixed it by checking the script.

**Screenshots Attached:**

