PYTHON CODE

```
import time
import serial
from pyfirmata import Arduino, util

# Setup for Arduino board using pyFirmata
board = Arduino('COM3')  # Replace COM3 with your Arduino port

# Start iterator to read analog pins
it = util.Iterator(board)
it.start()

# Define pins
LEVEL_PIN = board.analog[0]  # A0
RAIN_PIN = board.analog[1]   # A1
BUZZER_PIN = board.digital[13]  # Digital pin 13

# GSM module connected via serial
gsm_serial = serial.Serial('COM4', baudrate=9600, timeout=1)  # Replace with GSM serial port

# Enable reporting for analog inputs
LEVEL_PIN.enable_reporting()
RAIN_PIN.enable_reporting()

def read_sensor(pin):
    value = pin.read()
    if value is None:
        return 0
    return int(value * 1023)
```

```
def send_sms(message, number="+919380757402"):
    gsm_serial.write(b'AT+CMGF=1\r')
    time.sleep(1)
    gsm_serial.write(f'AT+CMGS="{number}"\r'.encode())
    time.sleep(1)
    gsm_serial.write((message + "\x1A").encode())
    time.sleep(8)

def make_call(number="+919380757402"):
    gsm_serial.write(f"ATD{number};\r".encode())
    time.sleep(20)
    gsm_serial.write(b"ATH\r")

def alert(message):
    BUZZER_PIN.write(1)
    send_sms(message)
    make_call()
    BUZZER_PIN.write(0)

def loop():
    while True:
        level_val = read_sensor(LEVEL_PIN)
        rain_val = read_sensor(RAIN_PIN)

        # DHT11 is handled on Arduino and sent via serial if needed (or added here if using GPIO)
        temperature = 25  # Dummy value or read from Arduino
        humidity = 60     # Dummy value or read from Arduino

        print(f"{level_val},{rain_val},{temperature},{humidity},", end='')
```

```
        if level_val > 500:
            print("1")
            alert("Water level is too high")
        elif rain_val <= 900:
            print("1")
            alert("Rain density is too high")
        else:
            print("0")
            BUZZER_PIN.write(0)

        time.sleep(2)

if __name__ == "__main__":
    try:
        loop()
    except KeyboardInterrupt:
        board.exit()
        gsm_serial.close()
```