Python script on the IoT sensors to send real-time water consumption data to the data-sharing platform.

```
import serial
import time

SERIAL_PORT = COM3
BAUD_RATE = 9600

# Thresholds for excess consumption and low moisture
WATER_CONSUMPTION_THRESHOLD = 150
SOIL_MOISTURE_THRESHOLD = 30

# Function to send an alert message
def send_alert(message):
    print(f'ALERT: {message}')

# Initialize the serial connection to the Arduino
ser = serial.Serial(SERIAL_PORT, BAUD_RATE)

try:
    while True:
        data = ser.readline().decode().strip()
        try:
            category, value = data.split(":")
            value = float(value)

            if category == "SoilMoisture":
                print(f'Soil Moisture Level: {value}%')
                if value < SOIL_MOISTURE_THRESHOLD:
                    alert_message = 'Low soil moisture level detected. Water your
garden.'
                    send_alert(alert_message)

            elif category == "Humidity":
                print(f'Humidity: {value}%')

            elif category == "WaterConsumed":
                print(f'Water Consumed: {value} liters')
                if value > WATER_CONSUMPTION_THRESHOLD:
                    alert_message = f'Excess water consumption detected: {value} liters.'
                    send_alert(alert_message)

        except ValueError:
            print(f'Invalid data received: {data}')

except KeyboardInterrupt:
    ser.close()
```

Explanation:

We read data from the Arduino over the serial connection, split it into categories (SoilMoisture, Humidity, WaterConsumed), and convert the values to floating-point numbers.

We check for low soil moisture and excess water consumption. If a condition is met, we call the send_alert function (which you can implement as needed) to generate an alert. The script prints the alert message to the console, but you can replace it with your preferred alerting mechanism.

We continuously read and process data until the script is interrupted.