

VSA GROUP OF INSTITUTIONS

PROJECT TITLE

**WATER QUALITY ANALYSIS**

# Team members

Name Reg No

K.KADHAR FERRASH 623121104001

M.S.KRISHNA KUMAR 623121104003

P.PREMKUMAR 623121104006

A.REGANRAJA 623121104007

Under by

M.LAVANYAPRABHA.,AP/CSE

**AI & ADS**

Analyzing water quality with AI and ads (Advanced Driver Assistance Systems) is a broad and complex topic, and creating a full project on it might be beyond the scope of a mini-project. However, I can provide you with a simplified example of a water quality analysis using AI and some code snippets to get you started. Please note that this is a basic illustration and real-world applications will require more data, advanced models, and domain-specific knowledge.

**Project Overview**

In this mini-project, we will create a simple AI model to predict water quality using some common water quality parameters. The AI model will be trained on a small dataset, and we'll use Python and popular libraries such as scikit-learn for machine learning.

**Requirements**

1. Python

2. Jupyter Notebook (optional)

3. scikit-learn

4. Pandas

5. Matplotlib

**Code**

Here's a step-by-step outline of the project:

**1. Data Collection:**

You need a dataset with water quality parameters (e.g., pH, turbidity, temperature) and corresponding labels (e.g., water quality index).

**2. Data Preprocessing:**

1. Load the dataset.
2. Clean and preprocess the data.
3. Split the data into training and testing sets.

**3. Feature Selection:**

Choose relevant features for water quality prediction. You might need domain knowledge for this.

**4. Machine Learning Model:**

1. Train a machine learning model (e.g., Decision Tree, Random Forest, or Gradient Boosting) using scikit-learn.
2. Use the selected features to train the model.

**5. Model Evaluation:**

1. Evaluate the model using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or R-squared.
2. Visualize the model's performance.

**6. Prediction:**

Use the trained model to predict water quality based on input feature values.

**Sample Code:**

Here's a simplified code outline for a Decision Tree Regressor model using scikit-learn:

**PYTHON :**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeRegressor

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

# Load the dataset (replace 'data.csv' with your dataset)

data = pd.read\_csv('data.csv')

# Define features (X) and labels (y)

X = data[['pH', 'turbidity', 'temperature']]

y = data['water\_quality\_index']

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train a Decision Tree Regressor

model = DecisionTreeRegressor()

model.fit(X\_train, y\_train)

# Make predictions

y\_pred = model.predict(X\_test)

# Model evaluation

mae = mean\_absolute\_error(y\_test, y\_pred)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f'MAE: {mae}')

print(f'MSE: {mse}')

print(f'R-squared: {r2}')

Remember that for a real-world application, you'd need a more extensive dataset, data from various sources (e.g., sensors or lab tests), and possibly more advanced machine learning models. Also, ads are typically not directly related to water quality analysis, so you might want to clarify their role in your project.

**DAC**

Water quality analysis using Digital-to-Analog Converters (DACs) is an uncommon approach. DACs are typically used to convert digital signals to analog signals, and they are not directly related to water quality analysis. However, if you are interested in controlling some analog equipment or sensors using a microcontroller, you can use a DAC for this purpose.

Here's a simple example of a mini-project where you can use a DAC to control a water quality measurement sensor (e.g., a pH sensor) and collect analog data for analysis. This project assumes you have a microcontroller with a DAC and analog input capability (e.g., Arduino or Raspberry Pi).

**Project Overview**

In this mini-project, we will use a DAC to control the voltage supplied to a water quality sensor (e.g., pH sensor) and collect analog data for analysis.

**Requirements:**

1. Microcontroller (e.g., Arduino, Raspberry Pi)

2. DAC (if not built into the microcontroller)

3. Water quality sensor (e.g., pH sensor)

4. Appropriate libraries for your microcontroller

**Code:**

Here's an outline of the project:

1. **Setup DAC:**

* If your microcontroller has a built-in DAC, you can set the DAC voltage level to control the sensor. If not, you'll need an external DAC module.

2. **Connect Sensor:**

* Connect the water quality sensor (e.g., pH sensor) to an analog input pin on your microcontroller.

3. **Data Collection:**

* Use the DAC to control the voltage supplied to the sensor.
* Read analog data from the sensor.

4. **Data Analysis:**

* Analyze the analog data collected from the sensor.
* Calculate water quality parameters (e.g., pH level) based on the sensor's response.

**Sample Code (Arduino):**

Here's a simple code example for Arduino using a hypothetical DAC and pH sensor:

**Arduino:**

#include <DACLibrary.h>

const int dacPin = A0; // Replace with the actual DAC pin

const int sensorPin = A1; // Replace with the actual sensor pin

float voltage;

void setup() {

// Initialize DAC if using an external DAC module

// Initialize serial communication

Serial.begin(9600);

}

void loop() {

// Generate an analog voltage using the DAC

voltage = 2.5; // Set the desired voltage (for example, 2.5V)

DAC.write(dacPin, voltage); // Write the voltage to the DAC pin

// Read the analog value from the sensor

int sensorValue = analogRead(sensorPin);

// Perform data analysis

// Convert analog data to water quality parameters (e.g., pH)

// Print the results

Serial.print("Voltage: ");

Serial.print(voltage);

Serial.print("V, Sensor Value: ");

Serial.println(sensorValue);

delay(1000); // Delay for data stability

}

Please note that the specific code may vary depending on your microcontroller and the water quality sensor you are using. You will need to refer to the documentation and libraries for your hardware. This is a basic example to get you started with the concept of using a DAC to control a sensor and collect data for water quality analysis.

**IOT**

Water quality analysis using the Internet of Things (IoT) is a practical and relevant application. In this mini-project, I'll outline a basic example of how to monitor and analyze water quality using IoT devices. We'll use a simple setup with a microcontroller (such as an Arduino or ESP8266), a water quality sensor, and an IoT platform (ThingSpeak) to store and visualize the data.

**Project Overview:**

In this mini-project, we'll set up an IoT-based water quality monitoring system using a microcontroller, a water quality sensor (e.g., pH, turbidity, or conductivity sensor), and the ThingSpeak platform to log and visualize the data.

**Requirements:**

1. Microcontroller (e.g., Arduino, ESP8266, or ESP32)

2. Water quality sensor (compatible with your microcontroller)

3. Internet connection (Wi-Fi or Ethernet)

4. ThingSpeak account (IoT platform for data storage and visualization)

**Code:**

Here's an outline of the project:

1. **Setup Hardware:**

* Connect the water quality sensor to the microcontroller following the sensor's datasheet.
* Ensure your microcontroller is connected to the internet (e.g., via Wi-Fi or Ethernet shield).

2. **Arduino Code:**

* Write code for your microcontroller to read data from the water quality sensor.
* Send the data to ThingSpeak using the ThingSpeak API.

**Sample Arduino Code:**

Here's a simplified example using an Arduino with a pH sensor and the ThingSpeak platform. You can modify this code to suit your specific sensor and needs.

**Arduino**

#include <Wire.h>

#include <Adafruit\_ADS1015.h>

#include <ThingSpeak.h> // Library for ThingSpeak

#define pHsensorPin A0

#define SECRET\_CH\_ID 1234567 // Replace with your ThingSpeak Channel ID

#define SECRET\_API\_KEY "YOUR\_API\_KEY" // Replace with your ThingSpeak API Key

Adafruit\_ADS1115 ads; // Create an instance of the ADS1115 ADC

void setup() {

Serial.begin(115200);

ThingSpeak.begin(client); // Initialize ThingSpeak

// Initialize the ADC

ads.begin();

}

void loop() {

float pHValue = readpH(); // Read pH value from the sensor

Serial.println("pH Value: " + String(pHValue, 2));

// Send data to ThingSpeak

ThingSpeak.setField(1, pHValue); // Set the field to store pH data

int status = ThingSpeak.writeFields(SECRET\_CH\_ID, SECRET\_API\_KEY);

if (status == 200) {

Serial.println("Data sent to ThingSpeak successfully.");

} else {

Serial.println("Error sending data to ThingSpeak. HTTP error code: " + String(status));

}

delay(60000); // Send data every 60 seconds (adjust as needed)

}

float readpH() {

// Replace this with code to read data from your water quality sensor

// This is just a placeholder.

// You should use the appropriate method for your sensor.

float pHValue = random(4, 10); // Replace with actual reading

return pHValue;

}

* Please make sure to install the necessary libraries, configure your ThingSpeak account, and replace the placeholders with your actual ThingSpeak Channel ID and API Key. The `readpH` function should be replaced with the code to read data from your specific water quality sensor.
* This project provides a basic framework for an IoT-based water quality monitoring system. Depending on the specific sensor and requirements, you may need to adapt the code and hardware components accordingly.

**CAD**

Water quality analysis typically doesn't involve Computer-Aided Design (CAD), which is used for designing and modeling objects in 3D. Instead, water quality analysis usually involves collecting and analyzing data from various water quality parameters, such as pH, turbidity, and chemical concentrations. However, if you intend to create a CAD model of a water quality monitoring system or a related device for a mini-project, I can provide you with an example.

Here's a simplified mini-project idea that involves creating a 3D CAD model of a water quality monitoring station:

**Project Overview:**

Create a 3D CAD model of a water quality monitoring station. The station can include components like sensors, data loggers, and a protective housing. While this doesn't perform water quality analysis directly, it's a project relevant to the monitoring aspect of water quality analysis.

**Requirements:**

1. CAD software (e.g., AutoCAD, SolidWorks, Blender, Tinkercad, or Fusion 360)

2. Basic knowledge of CAD modeling

**CAD Modeling:**

1. **Design the Water Quality Monitoring Station:**

Create a 3D model of the monitoring station that includes all the necessary components such as sensors, data logger, power supply, and a protective housing. Consider the dimensions and arrangement of these components.

2. **Specify Sensor Locations**:

Within the model, specify where the water quality sensors would be placed. This demonstrates an understanding of the importance of sensor placement in real-world water quality analysis systems.

3. **Include Cable Management:**

Model the cable routing and management within the station, which is essential for sensor connectivity.

4. **Design for Environmental Protection:**

Ensure the model provides environmental protection for the sensitive electronic components. This may involve sealing to protect against water ingress.

5. **Add Detail:**

Add fine details like labels, buttons, or indicators to enhance the realism of the model.

6. **Render and Visualize:**

Use the rendering capabilities of your CAD software to create realistic images or animations of the water quality monitoring station. This can be useful for presentations and documentation.

**Code:**

* While there is no code involved in creating a CAD model, you can use CAD software to generate 3D models, make adjustments to the design, and create visual representations of your water quality monitoring station.
* Remember that this mini-project doesn't directly involve water quality analysis but focuses on creating a CAD model of a monitoring station. To incorporate actual water quality analysis, you would need additional components such as sensors and microcontrollers, as described in previous responses.

**CONLUSION**

* Pollution is the degradation of water quality brought on by human activities, which renders the water less usable than it was initially. To be deemed unpolluted, water does not need to be entirely pure.
* Waters found in nature are not entirely pure. Although almost a thousand times less concentrated, rainwater has dissolved salts in relative proportions that are similar to those in seawater. Rainwater is slightly more acidic than seawater because it has a higher relative amount of dissolved gases, especially carbon dioxide. River water is different from seawater and rainwater in terms of its composition; it contains suspended sediments and has a higher concentration of dissolved solids (TDS) than rainwater. According to the rocks it has travelled through, groundwater often has slightly higher TDS values than surface water and differs in composition.
* There are a wide variety of sources of pollution, including sewage from homes, farming, industry, mining , quarrying, and cooling. Pollutants come in many different forms, such as naturally occurring organic matter, living things, plant nutrients, organic and inorganic compounds, sediments, and heat.Water frequently needs to be treated before it is of a quality that is useful. The required quality varies on the intended use of the water; while the WHO, the EU, and certain individual nations set quality requirements for public water supplies, the standards for industrial water and irrigation water can differ.
* Sewage treatment seeks to, mostly through biological or settlement processes, reduce the quantity of organic and suspended solid material present, remove hazardous compounds, and eradicate harmful microorganisms. The residual sludge may be dumped at sea (though not in the EU), disposed of on farmland, disposed of in landfills, or burned. The effluent is discharged into rivers, lakes, or the sea