Design of a Machine Learning-based Water Quality Classification Tool for On-site Colorimetric Analysis

Brodeth, Van Jersey Paolo
Pascual, Ken Leonard

Technological Institute of the Philippines
Quezon City

November 2025

Approval Sheet

This design project entitled **"Design of a Machine Learning-based Water Quality Classification Tool for On-site Colorimetric Analysis"** prepared by **Van Jersey Paolo Brodeth and Ken Leonard Pascual** of the Computer Engineering Department, was examined and evaluated by the members of the Student Design Evaluation Panel and is hereby recommended for approval.

ENGR. JI HAN C. GANG
Adviser

Student Design Evaluation Panel:

ENGR. JIMLORD M. QUEJADO
Panel Member

ENGR. LLOYD ALDRIN T. PORNOBI
Panel Member

ENGR. ROMAN M. RICHARD
Lead Panel

ENGR. ROMAN M. RICHARD
Program Chair

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES

Quezon City

Major (Capstone) Design Experience Information
SOFTWARE DESIGN PROJECT

1st Semester, SY 2025–2026

Student/Team **Team 5**	Brodeth, Van Jersey Paolo Pascual, Ken Leonard
Project Title	Design of a Machine Learning-based Water Quality Classification Tool for On-site Colorimetric Analysis
Project Concentration Area	Machine Learning, Water Classification, Software Design
Design Objectives	The project aims to develop a water quality classification system powered by machine learning algorithms that enable automated classification of water samples based on DENR guidelines and location-based result management using colorimetry data as a primary analytical input. Specifically, the project aims to: 1. Develop a web-based application that: a. Processes and classifies water sample quality with focus on chemical contamination and water usability using machine learning algorithms. b. Outputs water classification and analytical interpretations based on colorimetry data gathered c. Saves and organizes the output for each specified testing location. 2. Test and evaluate the system's accuracy.
Constraints	
Safety (Misclassification Rate)	The safety constraint is focused on minimizing errors to the system that can prove fatal or dangerous to its users. In the context of the aforementioned project, misclassification rate will be the quantifiable metric to measure how safe the system is as a tool. Misclassification rate is calculated by dividing the number of incorrect predictions by the total number of predictions, and can also be calculated as the difference between 100 percent accuracy and the actual accuracy of the system.

Performance (Inference Time)	Performance constraint focuses on the system's ability to produce a classification result quickly after receiving input data. Inference Time defines the duration required for the model to process input and output a prediction. The metric is typically measured in milliseconds or seconds and is obtained using time profiling tools such as Python's time module during benchmark tests. The constraint is related to the metric because long inference times delay decision making, reducing the functionality and responsiveness of the system. Therefore, the model with the lowest inference time is preferred for the final design.
Manufacturability (Training Time)	Manufacturability is focused on how fast and efficient the system can be developed, tested, and deployed. This is especially crucial when designing systems that are meant to be included for mass production. In the context of the aforementioned project, training time will be the quantifiable metric to assess how fast the system's machine learning model can be trained and tested before being deployed.
Efficiency (Storage Consumption)	Efficiency constraint ensures the system operates with minimal computational resources and runs fast for large inputs. In the project's context, storage consumption is used to determine the amount of storage space a machine learning model uses to be utilized for its applications. Lower storage consumption effectively means that the model uses less storage space, which can be useful in low-storage scenarios. Thus, the design with the lowest storage consumption will be the preferred design
Compatibility (Maintainability Index Score)	Compatibility constraint focuses on the system's ease of long-term maintenance and upgrade. Maintainability Index Score quantifies software maintainability through metrics such as code complexity, documentation, and ease of modification, usually calculated by static analysis tools (e.g., radon in Python). The metric is obtained by running these tools on the codebase to generate a score, with higher scores indicating better maintainability. Alternatively, python libraries such as radon can compute these metrics. Poor maintainability complicates future development and updates, risking compatibility with evolving requirements. Therefore, a higher maintainability index score indicates a more compatible and sustainable system.
Other constraints: These constraints do not affect each design; therefore, these were not included in selecting the best design.	
Sustainability	Sustainability involves reducing the environmental footprint of the system and promoting its long-term usability. Designs with high sustainability minimize adverse impacts on natural resources and are built for extended operational lifespans, supporting ecological preservation and responsible management of water quality over time.

Public Health	Public health pertains to how the system affects the well-being of users, stakeholders, and communities interacting with it. Ensuring the design supports good health practices means preventing negative effects on the physical or mental health of users and ultimately safeguarding public safety when providing water quality information.
Welfare	Welfare measures how the solution enhances people's lives. This design provides clients with accessible tools and services that empower them to manage water quality, thus contributing to better living conditions and improving users' overall quality of life
Social	Social constraint considers how the software fosters connections between users and stakeholders within the project ecosystem. By supporting collaborative features and effective communication, the system builds strong relationships between clients and their customers, resulting in a more cohesive user experience.
Standards	
ISO 8601 (Date and Time Format)	ISO 8601 is a globally recognized standard for formatting dates and times in a way that eliminates confusion. It uses an ordered structure, starting from the largest unit (year) down to the smallest (second), such as YYYY-MM-DDTHH:mm:ssZ. This ensures all timestamps attached to water sample data, sensor readings, and predictions are standardized, which is essential for reliable chronological analysis, model training, and interpretation across systems *(See Appendix C, p. 53)*
ANSI/IEEE 1012 (Software Verification and Validation)	ANSI/IEEE 1012 provides a structured framework for software verification and validation (V&V) throughout the lifecycle of a software product. It details processes such as analysis, reviews, testing, and evaluation to ensure the software satisfies user requirements and its intended function, whether the software is being developed, maintained, or reused. This provides a framework for rigorously testing, validating, and documenting ML models and their pipelines, improving both reliability and regulatory compliance *(See Appendix D, p. 54)*
WHO Guidelines for Drinking Water Quality	The World Health Organization's Guidelines for Drinking-Water Quality set forth comprehensive recommendations using risk management approaches to ensure water safety from source to consumer. These guidelines establish health-based targets, advocate for water safety plans, and require independent monitoring. Parameters covered include microbiological, chemical, and physical aspects of water to safeguard public health. This establishes threshold values and risk-based classification logic for ML model labeling (e.g., safe/unsafe for drinking), ensuring health significance

	(See Appendix E, pp. 55 - 58)
Philippine National Standards for Drinking Water (PNSDW)	The Philippine National Standards for Drinking Water define maximum limits for microbiological, physical, chemical, and radiological constituents in water, aiming to protect public health. The standards specify how water quality should be monitored and establish protocols for emergency situations, sampling, and responsibilities of providers and stakeholders. It was used to create ground truth labels in training data, define output categories, and evaluate model outputs according to national standards *(See Appendix F, pp. 59 - 60)*
DAO 2016-08 Water Quality Guidelines and General Effluent Standards	DAO 2016-08 outlines water quality guidelines and general effluent standards, classifying water bodies and setting beneficial use categories (such as drinking, recreation, and fisheries). It serves as a regulatory basis for preserving water quality across fresh, marine, and groundwater and provides actionable thresholds for intervention, control, and abatement of water pollution across the Philippines. It was used as the basis for water classification. *(See Appendix G, pp. 61 - 62)*

Abstract

List of Tables

List of figures

List of abbreviation

Definition of terms

TABLE OF CONTENTS

CHAPTER 1: THE PROJECT AND ITS BACKGROUND

This chapter provides an overview of the project, its background, and the factors that influenced its development. It outlines the project's goals and constraints, identifies the target client and their requirements, discusses relevant engineering standards, and details the design process used.

1.1 The Problem

Access to safe drinking water is a major concern for hikers and campers who often depend on natural sources like streams and lakes, which can easily be contaminated. With people spending time outdoors and the difficulty of checking water quality in remote areas, there is a clear need for reliable, portable testing tools. While various waterborne threats exist, chemical contaminants represent significant and often underestimated health risks to outdoor enthusiasts.

Chemical contaminants such as phosphate and nitrate pose substantial health hazards when present in freshwater sources. These substances are particularly concerning because they are not visually detectable, and their health effects may manifest over time or in response to short-term exposure (Lin et al., 2022; US EPA, 2015). Elevated levels of nitrate and nitrite in drinking water can cause methemoglobinemia, a potentially life-threatening condition in infants under six months of age that reduces the blood's capacity to carry oxygen (U.S. EPA, 2025; Minnesota Department of Health, 2025). Recent epidemiological research has identified associations between long-term nitrate exposure at concentrations below the regulatory standard and increased risks of colorectal cancer, thyroid problems, and adverse pregnancy outcomes (Picetti et al., 2022; Ward et al., 2018; Knobeloch et al., 2000; Morales-Suarez-Varela et al., 1995).

For hikers and campers, particularly those engaging in extended outdoor activities, the risks associated with chemical contamination are compounded by several factors. First, outdoor enthusiasts may consume untreated water from remote sources without knowledge of potential chemical pollutants present in their watershed. Second, the remote nature of many hiking and camping destinations often means limited access to medical assistance, making even seemingly mild health issues more dangerous. Third, some outdoor enthusiasts may repeatedly use the same water sources during multiple trips, leading to cumulative exposure to chemical contaminants. Chemical exposure through drinking water can lead to a variety of short- and long-term health effects, from neurological damage and organ failure to developmental and reproductive effects, and in severe cases, cancer (U.S. EPA, 2025; Lin et al., 2022).

In the Philippines, where outdoor recreation is substantially popular as a recreational activity, water quality concerns are usually ignored since supervised trips involving tour guides or locals traditionally mitigate many contamination issues by guiding tourists, hikers, and campers to water sources that have been tested by the Department of Environment and Natural Resources' Environmental Management Bureau or local authorities to be safe for consumption. However, this protection is unavailable for independent outdoor enthusiasts.

Despite ongoing government initiatives to classify and monitor rivers, lakes, and other freshwater sources in the Philippines, data reporting and sampling frequency remains slow and inconsistent, with most comprehensive government monitoring data last publicly updated in 2019–2021 (Water Environment Partnership in Asia, 2025). This creates significant gaps between official water quality

assessments and the actual present-day state of community water resources, especially in remote or non-prioritized areas where monitoring is much less frequent. Furthermore, 43% of the country's rivers and 56% of its major water bodies are polluted, primarily due to industrial discharge, agricultural runoff, and inadequate domestic sewage treatment (Filipenco, 2024).

Regular and frequent water quality assessments are crucial since rivers and other freshwater bodies across the country continue to deteriorate in terms of quality due to pollution from domestic, agricultural, and industrial waste (Water Environment Partnership in Asia, 2025). Industrial activities release pollutants such as heavy metals, oils, and hazardous chemicals that can accumulate in the environment (Water Pollution in the Philippines, 2024). Agricultural pollution, accounting for 37% of the country's water pollution, introduces pesticide and fertilizer runoff containing nitrates and other chemical contaminants (Water Pollution in the Philippines, 2024). These findings highlight how sources of water in remote areas that are neither maintained and monitored on a regular basis can pose a major health risk not just for unwary locals, but hikers and campers as well.

Addressing the limitations of traditional water quality monitoring requires the creation of a versatile, user-friendly platform that non-specialists can easily operate in the field. While laboratory analysis remains the most accurate method for water quality assessment, it is often impractical for on-site testing, especially in remote or resource-constrained environments (Srivastava et al., 2018). Similarly, commercial chemical test kits, though portable, typically require technical handling and expertise, limiting their accessibility for general outdoor users.

Existing studies highlight the significant potential of smartphone-integrated sensing technologies, which utilize built-in cameras coupled with machine learning algorithms to deliver rapid and affordable environmental diagnostics (Niu et al., 2022; Doğan et al., 2022). Smartphone-based colorimetric analysis represents a promising approach, as color-based detection methods are inherently suited to measuring chemical contaminants through their optical properties (Kılıç et al., 2018; Doğan et al., 2022). However, many existing solutions focus on single water quality parameters such as turbidity or pH and face challenges related to sensor integration and calibration consistency, which can affect their analytical precision (Zhang et al., 2024; Alhaqi, 2025; Kaur et al., 2024).

Developing an analytical system capable of simultaneously measuring several key chemical indicators would vastly improve the ability of hikers, campers, and other outdoor enthusiasts to independently and confidently evaluate the safety of natural water sources in real time. This approach not only empowers users with immediate, evidence-based information but also bridges crucial gaps left by infrequent or inconsistent official water quality monitoring efforts.

1.2 The Client

The client for the project is Danielle Dolom, who is part of a group of mountaineers that hikes a few times a year. Trekking a few times a year means she frequently encounters natural water sources like streams and springs in remote settings, where immediate and reliable water quality assessment is crucial for her safety and resource management.

The client aims to utilize a tool that can help identify safe water sources during their trips, even in cases wherein locals and tour guides are not present to guide them. The tool must be capable of identifying whether the water can be used for drinking, safe to use for other purposes, or unsafe to use at all. Additionally, the tool needs to be reliable in its results, easy to comprehend even for non-technical individuals, and be capable of recording and storing the gathered data to be shared or used as a future resource.

Table 1.1. Client and Engineering Requirements / Considerations

Client Requirements / Considerations	Engineering Requirements / Considerations
The system must be able to classify water samples accurately	The system can classify water samples with at least **95% accuracy**
The system must be able to quickly show results	The system can be trained to show results within 10 minutes
The system must be able to classify water samples to determine what the water source is safe to use for	The system can classify water samples into three categories: Safe to drink after treatment (Class A), Only safe for non-drinking uses (Class B-C), and not safe for consumption (Class D)

1.3 The Project

The project aims to develop a water quality classification system that leverages a machine learning-based web platform to automate the classification and potability analysis of freshwater sources with an emphasis on chemical contamination detection. Users input numeric colorimetry data captured from colorimetric water tests which utilize reagent-based test kits to report concentration values derived from measured color intensity.

After gathering the numerical data, a machine learning model predicts and classifies the water sample's quality based on the guidelines for classification established by the Department of Environment and Natural Resources (2016) for fresh water, which categorizes freshwater bodies such as lakes, rivers, and streams according to beneficial use. This classification guides the evaluation of water safety for different purposes by assessing chemical quality indicators consistent with the guidelines stated by the Department of Environment and Natural Resources (2016). Additionally, the system features a web application that displays test results and archives data from past analyses, supporting ongoing environmental monitoring and informed decision-making.

1.4 Project Objectives

The project aims to develop a water quality classification system powered by machine learning algorithms that enable automated classification of water samples based on DENR guidelines and location-based result management using colorimetry data as a primary analytical input.

Specifically, the project aims to:

1. Develop a web-based application that:
    a. Processes and classifies water sample quality with focus on chemical contamination and water usability using machine learning algorithms.
    b. Outputs water classification and analytical interpretations based on colorimetry data gathered
    c. Saves and organizes the output for each specified testing location.
2. Test and evaluate the system's accuracy.

1.5 Scope and Delimitations

The scope of this project includes the design and implementation of a water quality classification system using machine learning. The system will utilize smartphone-based image capture with an integrated web application for outdoor enthusiasts such as hikers and campers to evaluate and classify the water quality of natural water sources in remote settings.

Since the system is limited to a smartphone camera for its image capture input, a color reference standard will also be utilized, which is a known and consistent color patch within the captured image to enable color calibration across different environmental conditions.Color calibration is necessary since the project is focused in colorimetric analysis as basis for its water quality classification system.

Furthermore, the classification system used in this project is solely based on the Department of Environment and Natural Resources (2016) classification guidelines for fresh water, which categorizes water bodies based on their intended beneficial use, but does not directly equate to the potability of water. This means that while the classification system provides guidance for different uses, it does not guarantee that the water is safe for direct human consumption without conventional treatment methods.

1.6 Design Constraints

Safety (Misclassification Rate)
The safety constraint is focused on minimizing errors to the system that can prove fatal or dangerous to its users. In the context of the aforementioned project, misclassification rate will be the quantifiable metric to measure how safe the system is as a tool. Misclassification rate is calculated by dividing the number of incorrect predictions by the total number of predictions, and can also be calculated as the difference between 100 percent accuracy and the actual accuracy of the system.

$$misclassification\ rate = \#\ Ip / \#\ Tp \quad \textit{Equation 1.6.1}$$

$$misclassification\ rate = 1 - accuracy \quad \textit{Equation 1.6.2}$$

Where in:
 # Ip - the number of false positive and false negative combined predictions
 # Tp - The number of total predictions in the system
Modules numpy and confusion_matrix is used to obtain the misclassification rate and accuracy

These calculations can be obtained by using Python libraries such as numpy and sklearn metrics such as confusion matrices. A higher misclassification rate means that the system cannot accurately classify the water samples based on its purpose, which can lead to risks such as consumption of contaminated water. Thus, the design that has a lower misclassification rate is much more preferable

Performance (Inference Time)
Performance constraint focuses on the system's ability to produce a classification result quickly after receiving input data. Inference Time defines the duration required for the model to process input and output a prediction. The metric is typically measured in milliseconds or seconds and is obtained using time profiling tools such as Python's time module during benchmark tests. The constraint is related to the metric because long inference times delay decision making, reducing the functionality and responsiveness of the system. Therefore, the model with the lowest inference time is preferred for the final design.

$$inference\ time = end\ time - start\ time \quad \textit{Equation 1.6.3}$$

Where in:
End time - The time where the system finished in making predictions
Start time - The time where the system receives the input
Modules numpy and time are used to obtain the end time, start time, and inference time

Efficiency (Storage Consumption)
Efficiency constraint ensures the system operates with minimal computational resources and runs fast for large inputs. In the project's context, storage consumption is used to determine the amount of storage space a machine learning model uses to be utilized for its applications. Lower storage consumption effectively means that the model uses less storage space, which can be useful in low-storage scenarios. Thus, the design with the lowest storage consumption will be the preferred design

Manufacturability (Training Time)
Manufacturability is focused on how fast and efficient the system can be developed, tested, and deployed. This is especially crucial when designing systems that are meant to be included for mass production. In the context of the aforementioned project, training time will be the quantifiable metric to assess how fast the system's machine learning model can be trained and tested before being deployed. Training time can be calculated with the following formula:

$$training\ time = end\ time - start\ time \quad \textit{Equation 1.6.4}$$

Where in:
End time - The time where the system finished training a model
Start time - The time where the system start training a model
Modules numpy and time are used to obtain the end time, start time, and training time

Lower training time means faster deployment cycles, meaning that the design with the lowest training time will be the preferred design.

Compatibility (Maintainability Index Score)
Compatibility constraint focuses on the system's ease of long-term maintenance and upgrade. Maintainability Index Score quantifies software maintainability through metrics such as code complexity, documentation, and ease of modification, usually calculated by static analysis tools (e.g., radon in Python). The metric is obtained by running these tools on the codebase to generate a score, with higher scores indicating better maintainability. Alternatively, python libraries such as radon can compute these metrics. Poor maintainability complicates future development and updates, risking compatibility with evolving requirements. Therefore, a higher maintainability index score indicates a more compatible and sustainable system.

$$MI = \max\left[0, 100\,\frac{171 - 5.2\ln V - 0.23G - 16.2\ln L + 50\sin(\sqrt{2.4C})}{171}\right]$$

Equation 1.6.5

Where in:
V = Halstead Volume
G = total Cyclomatic Complexity
L = number of Source Line of Code
C = percent of comment lines, converted to radians

Other constraints:
These constraints do not affect each design; therefore, these were not included in selecting the best design.

Sustainability
Sustainability involves reducing the environmental footprint of the system and promoting its long-term usability. Designs with high sustainability minimize adverse impacts on natural resources and are built for extended operational lifespans, supporting ecological preservation and responsible management of water quality over time.

Public Health
Public health pertains to how the system affects the well-being of users, stakeholders, and communities interacting with it. Ensuring the design supports good health practices means preventing negative effects on the physical or mental health of users and ultimately safeguarding public safety when providing water quality information.

Welfare
Welfare measures how the solution enhances people's lives. This design provides clients with accessible tools and services that empower them to manage water quality, thus contributing to better living conditions and improving users' overall quality of life

Social
Social constraint considers how the software fosters connections between users and stakeholders within the project ecosystem. By supporting collaborative features and effective communication, the system builds strong relationships between clients and their customers, resulting in a more cohesive user experience.

Global
Global constraint addresses the solution's ability to reach users worldwide and maintain inclusivity. The design is capable of operating on low-bandwidth networks and complies with international standards to ensure connectivity and accessibility for a diverse and global user base.

Cultural
Cultural constraint focuses on the software's respect for users' traditions, beliefs, and values. By including localized messages, imagery, and options, the system avoids offending cultural groups and ensures appropriate engagement across different regions and backgrounds.

1.7 Engineering Standards
The engineering standards serve as the foundation for the overall design and functionality of the project. To ensure that all specifications and requirements are carried out in compliance with these standards, the project adheres to the following guidelines:

ISO 8601 (Date and Time Format)
ISO 8601 is a globally recognized standard for formatting dates and times in a way that eliminates confusion. It uses an ordered structure, starting from the largest unit (year) down to the smallest (second), such as YYYY-MM-DDTHH:mm:ssZ. This format standardizes date and time representation, making information exchange clearer and more compatible across different platforms and regions.

ANSI/IEEE 1012 (Software Verification and Validation)
ANSI/IEEE 1012 provides a structured framework for software verification and validation (V&V) throughout the lifecycle of a software product. It details processes such as analysis, reviews, testing, and evaluation to ensure the software satisfies user requirements and its intended function, whether the software is being developed, maintained, or reused.

WHO Guidelines for Drinking Water Quality
The World Health Organization's Guidelines for Drinking-Water Quality set forth comprehensive recommendations using risk management approaches to ensure water safety from source to consumer. These guidelines establish health-based targets, advocate for water safety plans, and require independent monitoring. Parameters covered include microbiological, chemical, and physical aspects of water to safeguard public health.

Philippine National Standards for Drinking Water (PNSDW)
The Philippine National Standards for Drinking Water define maximum limits for microbiological, physical, chemical, and radiological constituents in water, aiming to protect public health. The standards specify how water quality should be monitored and establish protocols for emergency situations, sampling, and responsibilities of providers and stakeholders.

DAO 2016-08 Water Quality Guidelines and General Effluent Standards

DAO 2016-08 outlines water quality guidelines and general effluent standards, classifying water bodies and setting beneficial use categories (such as drinking, recreation, and fisheries). It serves as a regulatory basis for preserving water quality across fresh, marine, and groundwater and provides actionable thresholds for intervention, control, and abatement of water pollution across the Philippines.

1.8 Engineering Design Process

The Engineering Design Process is a method applied in developing solutions to problems involving the gathering of needs, testing, and refining of designs. The cycle steps involve problem definition, idea development, prototype development, and solution research to ensure that it meets the standards.

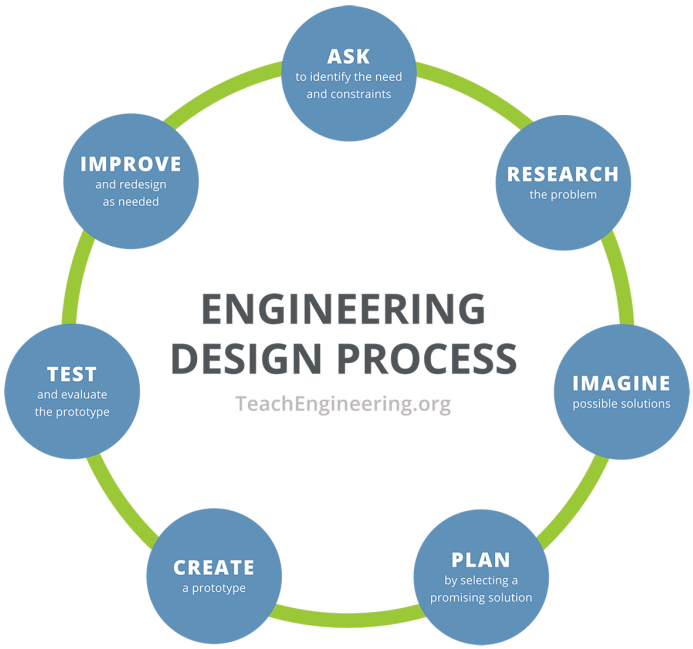


Figure 1.1 The Engineering Design Process (TeachEngineering, 2023)

The diagram presents the Engineering Design Process, an organized process that guides the solutions to problems through this cycle of steps to develop and refine innovative solutions

1.8.1 Ask: Identifying the Need and Constraints

To identify the need and constraints, the team has conducted research regarding past studies pertaining to portable water quality monitoring. After careful consideration, a client was contacted to be interviewed, giving insights about the problems experienced by hikers and campers when it comes to water consumption, and how water quality assessment plays a role in it. It was identified that without local guides, hikers and campers cannot be

confidently sure about the potability of water from natural water bodies in the remote areas they visit, prompting the need for independent

1.8.2 Research the Problem

According to a study, access to safe drinking water is a major concern for hikers and campers who often depend on natural sources like streams and lakes, which can easily be contaminated (United States Environmental Protection Agency, 2024). With people spending time outdoors and the difficulty of checking water quality in remote areas, there is a clear need for reliable, portable testing tools. While various waterborne threats exist, chemical contaminants represent significant and often underestimated health risks to outdoor enthusiasts (EPA, 2024).

1.8.3 Imagine: Develop Possible Solution

Several potential solutions were considered to address the identified needs. These include, but are not limited to measuring clarity/turbidity, identifying the pH level of water, measuring microbes, dedicated software application for analysis and prediction. Each option was evaluated based on usability, cost and scalability.

1.8.4 Plan: Select a Promising Solution

After comparing the alternatives, using a software application for analysis while pairing it up with several different reagents as the most practical and scalable solution. It offers real time analysis and prediction using the chemical reaction of reagents to the water.

1.8.5 Create: Build a Prototype

A working prototype was developed featuring two integrated interfaces:

- **Data Input Module:** Allows users to input data
- **Results and Reporting Module:** Displays the simple water quality report, the numerical parameter readings, the risk classification, and, most importantly, the **step-by-step purification guidance**.

1.8.6 Test and Evaluate the Prototype

The prototype will be tested using sample water examined by the software using the several reagents. Evaluation criteria include:

- Accuracy: Reliability of the analysis and step by step purification guidance with at least **95% accuracy**.
- Efficiency: Speed and responsiveness of the system.

1.8.7 Improve: Redesign as Needed

Based on testing results and user feedback, the system will be refined. Improvements may include UI enhancements, backend optimizations, and expanded features to better meet user needs and operational goals. Feedback will be collected and used to refine the features, simplify the interface, and improve the system overall.

CHAPTER 2: PROJECT DESIGN

This chapter includes the conceptual phase of a project design. It includes a data flow diagram to visualize the system's flow, along with engineering principles to develop efficient solutions. It provides an analysis of existing solutions, comparing them to the project requirements. Additionally, it outlines the hardware and software elements crucial for system design. The evaluation of alternative designs assesses their constraints, leading to the selection of the best design.

2.1 Description of the Design Solution

2.1.1 General Description

The water classification system is a solution that leverages colorimetric testing and machine learning to assess water quality. Users start by collecting a water sample and adding a chemical reagent, causing the sample to undergo a color change according to the substances present. This color change is quantified through a colorimetric reading, translating it into numeric concentration values. These values are then processed by a machine learning model, which evaluates water quality and automatically classifies the sample. The system stores the classification and makes it accessible to users, enabling quick, evidence-based water safety decisions in the field.

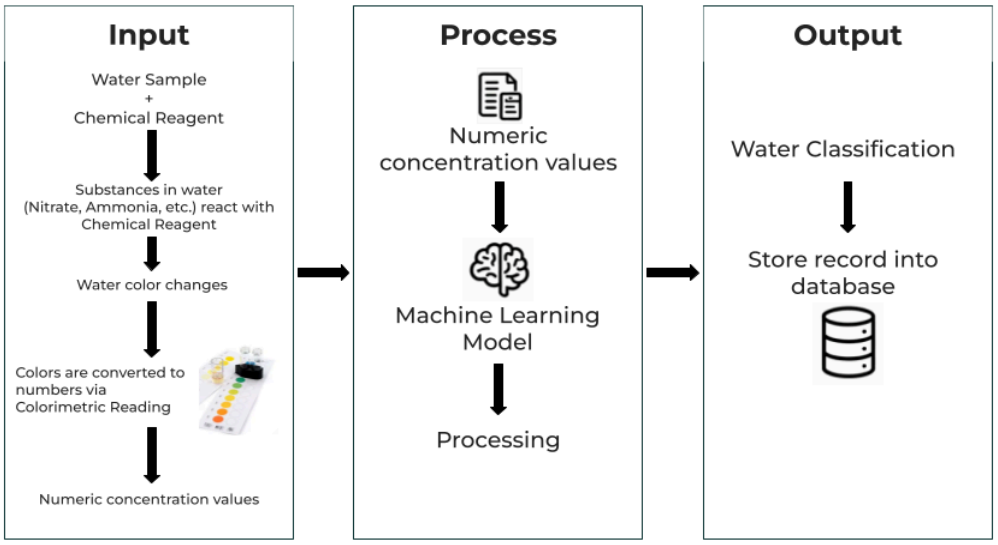


Figure 2.1 Illustrative Diagram of the System

The diagram illustrates the end-to-end workflow for water sample classification. It begins with the input of a water sample, which is mixed with a chemical reagent to trigger visible color changes in response to specific contaminants (such as nitrates or ammonia). Colorimetric analysis is used to convert these color changes into precise numeric concentration values. These numerical values serve as the input for a machine learning model, which processes the data to classify water safety. The resulting classification is then stored as part of a file record. Afterwards, users can access and interpret the file to understand the status of the water sample.

2.1.2 Engineering Principles Involved

This section introduces the fundamental concepts, methods, and best practices of engineering principles that will serve as the framework for the design of the system. This ensures that both the technical and user aspects of the system are well-engineered and maintain high standards of quality and performance.

The three designs revolve around computer engineering concepts, namely: machine learning, software engineering, and the KDD framework.

Software Engineering
Software engineering provides the structured development methodology for the system, including the software development lifecycle, and utilization of modular design principles. This modular approach enables for efficient processes of creating and modifying layers of the design without disrupting the entire system (Chiaramonte, 2024). This framework supports quality assurance, streamlined maintenance, continuous testing throughout development, as well as future scalability. (Barghoth et al., 2020)

Machine Learning and Predictive Modeling
Machine learning is a subset of artificial intelligences focused on algorithms that can learn through patterns found in data and make predictions and inferences based on those patterns (IBM, 2021). Supervised machine learning algorithms are used for classification tasks, while unsupervised algorithms can be used for clustering.

Knowledge Discovery in Databases
Knowledge discovery in databases enables systematic extraction of meaningful patterns and insights from accumulated records. The KDD process comprises multiples stages that aim to transform raw data into actionable information that supports data-driven decisions. KDD emphasizes the discovery of valid, novel, and understandable patterns in data that contribute to informed strategies and interventions. (Shu & Ye, 2022)

2.1.3 Prior Art Analysis
Prior studies, products, and patents have existed for the purpose of monitoring, analyzing, and classifying water quality. These methods either employ colorimetry which uses reagents that react with water, object detection, or IoT devices.

A study by Kılıç et al. (2018) employs single-image-references colorimetric water quality detection using a smartphone and machine learning algorithms to predict the concentration value given an image input from the smartphone. It leverages distance-based analysis to match the input color to a measured concentration level. This eliminates the need for understanding what concentration value each color implies, at the cost of accuracy. Furthermore, the study is more inclined with using portable devices to make the input process easier for the users.

Another product is called the Kactoily water tester, a commercially available product that can be integrated with devices to measure distinct water quality parameters through electrochemical measurements and optical analysis, rather than colorimetric

reagent-based reactions (Kactoily, 2025). This product, despite its features and market availability, is also relatively expensive compared to colorimetric alternatives.

Sophisticated water quality monitoring systems used in facilities are not considered to be consumer-viable solutions. The US6444172B2 patent filed by Hitachi Ltd. is an innovative approach to distributed, real-time water quality monitoring specifically designed for drinking water distribution systems. Its capability to measure water quality degradation across distribution channels while providing real-time data is what sets it apart from other alternatives.

Another solution that can be used for water quality monitoring and classification is the usage of visual language models (VLMs) such as Moondream. However, the model is currently trained for visual recognition via object detection, which may not be useful for contamination threats that are seen beyond the camera lens.

Finally, a sophisticated on–site solution currently exists, albeit on its early stages of consumer adoption. The WaterScope water testing platform is a portable platform that includes incubation and imaging units to assess water quality through both bacterial and chemical assays. Furthermore, it employs machine learning principles for its advanced image classification and on-board inferences. However, it is currently incapable of storing results, and is quite bulky for its intended application.

The features of the aforementioned studies and projects are summarized in the prior art analysis matrix. Given that information, the proposed system for this project aims to bridge the gap between these technologies by employing a low-cost approach that is user-friendly through its interface and implementation of machine learning algorithms, as well as capability to store information for future reference.

Table 2.1.3 Prior Art Analysis Matrix

Design	**Features**				
	Low-cost / Consumer-viable solution	**Smartphone-compatible, web-based Interface**	**ML/AI Water Classification**	**Modular Assays (Usage of Chemical Reagents)**	**Capability to store analysis and results**
SIR-based Smartphone colorimetry	X			X	
Kactoily 7-in-1 Water Tester	X				X

Design	Features				
	Low-cost / Consumer-viable solution	Smartphone-compatible, web-based Interface	ML/AI Water Classification	Modular Assays (Usage of Chemical Reagents)	Capability to store analysis and results
Hitachi Patent on Water Quality Monitoring Tool					X
Moondream 2 AI VLM	X	X	X		X
WaterScope testing platform	X	X	X	X	
PROJECT	X	X	X	X	X

2.2 General System Architecture

The general system architecture discusses the fundamental concepts implemented in the design. This includes the software elements involved in the project and the methods in which they were utilized in the development process.

2.2.1 Software Elements

The software elements used in this project are covered in this section. This includes the Python libraries used for analysis and data modeling, as well as other software used in the development process.

Pandas

This library is used for data handling throughout the project, including loading datasets from CSV files, merging multiple data sources, cleaning missing or inconsistent values, and constructing the final feature tables that are passed into the machine learning models

Numpy

This library underpins the numerical side of the project, providing arrays and mathematical routines that support vectorized operations, feature transformations, and the numerical computations required by the learning algorithms.

Scikit-learn

This library provides the workflow utilities for data modeling, including splitting the data into training and test sets, scaling or normalizing features, and computing performance metrics such as accuracy, confusion matrices, and ROC-AUC to objectively assess each trained model.

Joblib
This library is utilized to serialize and store trained models, preprocessing objects such as scalers, and other large Python objects so that they can be efficiently loaded later for evaluation or integration into the user interface without retraining

Radon
This library is used to compute software-quality metrics, in particular the maintainability index and cyclomatic complexity of the project's source code, providing a quantitative assessment of code complexity and guiding refactoring decisions.

A. . Application Software
The design currently utilizes a command–line interface to input colorimetric data from the user. Then, the results are displayed and then stored into a CSV file, which contains data such as the input values, the time that the data was processed, as well as the classification result.

A working mock-up design was also developed, to showcase the user interface. The mock-up was implemented through streamlit, and works by uploading the prediction log generated by the CLI interface as its output.

```
(CPE312_Pascual) PS C:\Users\Leon\Documents\Github\CPE025A\python\lightgbm> python PredictionDraft_LightGBM.py
Please input the following water colorimetry concentration values:
Ammonia (mg/l): 0.05
pH (ph units): 8
Nitrate (mg/l): 2
C:\Users\Leon\miniconda3\envs\CPE312_Pascual\Lib\site-packages\sklearn\utils\validation.py:2749: UserWarning: X
  warnings.warn(
Predicted water class: A3
Prediction saved to predictions_log.csv
Current file size: 108 bytes
```

Figure 2.2.1.1. Sample CLI Input and Output for Water Classification

Ammonia (mg/l)	pH (ph units)	Nitrate (mg/l)	Timestamp	Predicted_Class
0.05	8	2	2025-11-24T10:22:57	A3
2	9	7	2025-11-24T10:23:52	C3
90	5	90	2025-12-06T11:08:30	C4

Figure 2.2.1.2. CSV file Containing Past Water Classification Records

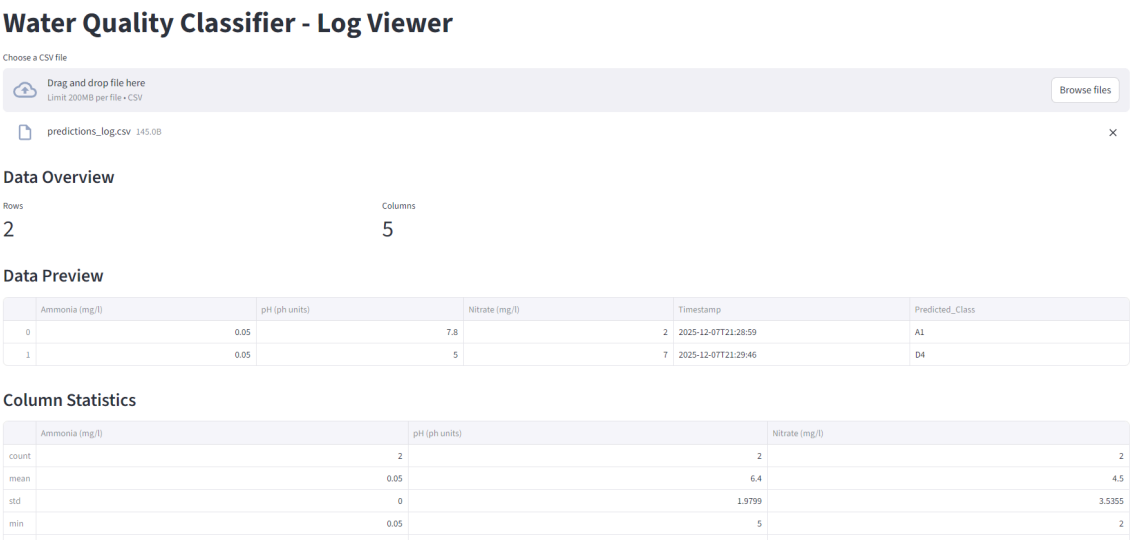


Figure 2.2.1.3. Concept UI for Water Quality Classification Output Display

B. Key Algorithms Used

Gradient boosting was the key algorithm used in all the designs. CatBoost, LightGBM, and XGBoost are all gradient boosting libraries with varying performance specializations. Gradient boosting is a tree-based algorithm which works by adding many simple decision trees one after another, wherein each new tree tries to fix the mistakes of the model built so far. It uses gradients of a loss function to decide what each new tree should learn. The final prediction is the sum of the contributions from all trees, which makes the whole algorithm an ensemble learning method.

2.2.2 System Algorithm

This section discusses the structural design and core components implemented in the system, as visualized through the Level 0 and Level 1 Data Flow Diagrams. The discussion surrounds the following architectural elements and how they are integrated within the development process. It details the system's input, processing, and output mechanisms, illustrating the flow from raw data collection to the generation of the water classification results

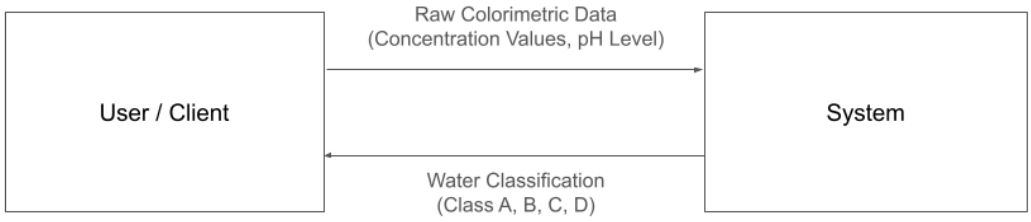


Figure 2.2.2.1 . Level 0 Data Flow Diagram

The Level 0 Data Flow Diagram shows the overall flow of interaction between the user and the system. It specifies the primary input and primary output of the two entities, which basically describes how raw colorimetric data is processed to return a water classification.

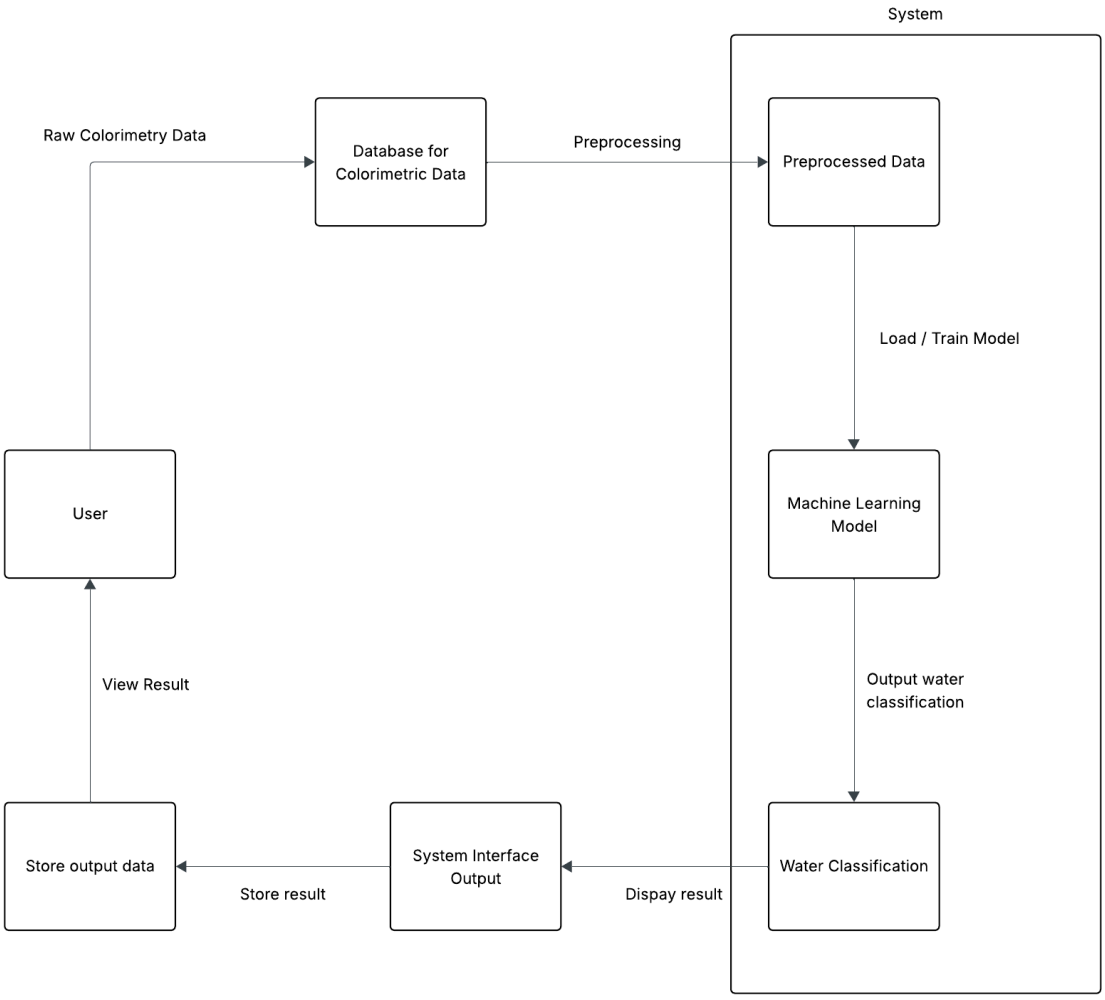


Figure 2.2.2.2. Level 1 Data Flow Diagram

The Level 1 Data Flow Diagram provides further detail about the overall flow of interaction between the user and the system. It specifies how raw colorimetric data is processed and cleaned before being fed into the machine learning to be interpreted or used for training. Afterwards, the machine learning model outputs the input data's classification and stores it, which can then be viewed by the user.

2.2.3 Data, Datasets, and Processing

This section details the data to be used in the project's development. This includes the steps for collecting, validating, and utilizing the data gathered from datasets. In addition to that, this section describes the processes of cleaning, processing, and transforming the data to achieve the project's objectives and comply with client requirements while ensuring accuracy and reliability with the results.

a. Datasets

The dataset used in training the machine learning models is named "A Comprehensive Surface Water Quality Monitoring Dataset (1940-2023): 2.82Million Record Resource for Empirical and

ML-Based Research", which contains data about concentration levels of different substances in water (Karim et al., 2025). The dataset is available in CSV data format, and contains water quality readings from five countries, namely: United States, Canada, Ireland, and China. Water quality was classified based on eight features, namely: Ammonia, BOD (Biochemical Oxygen Demand), DO (Dissolved Oxygen), Orthophosphate, Temperature, Nitrogen, Nitrate, and pH levels. The data also includes the date when each sample was collected, as well as the waterbody type it came from. For classification, the Canadian Council of Ministers of the Environment Water Quality Index (CCME WQI) model was used in the dataset for calculating the CCME values, which was then used to classify the samples.

	Country	Area	Waterbody Type	Date	Ammonia (mg/l)	Biochemical Oxygen Demand (mg/l)	Dissolved Oxygen (mg/l)	Orthophosphate (mg/l)	pH (ph units)	Temperature (cel)	Nitrogen (mg/l)	Nitrate (mg/l)	CCME_Values	CCME_WQI
27967	Ireland	Cork Harbour, Moy Killala, YELLOW (KNOCK)_020	River	22-06-2021	0.042	1.0	11.684103	0.025	8.0	14.10	0.56	0.55	100.0	Excellent
27968	Ireland	Cork Harbour, Moy Killala, YELLOW (KNOCK)_020	River	07-10-2021	0.020	1.3	4.068000	0.019	7.6	13.50	0.51	0.51	100.0	Excellent
27969	Ireland	Cork Harbour, Moy Killala, YELLOW (KNOCK)_020	River	30-11-2021	0.047	1.0	4.972500	0.017	7.9	11.15	0.56	0.55	100.0	Excellent
27970	Ireland	Cork Harbour, Moy Killala, YELLOW (KNOCK)_020	River	08-02-2022	0.039	1.1	11.684103	0.013	7.9	8.30	0.77	0.76	100.0	Excellent
27971	Ireland	Cork Harbour, Moy Killala, YELLOW (KNOCK)_020	River	07-04-2022	0.030	1.0	11.684103	0.025	8.1	6.00	0.57	0.57	100.0	Excellent
27972	Ireland	Cork Harbour, Moy Killala, YELLOW (KNOCK)_020	River	21-06-2022	0.024	1.4	4.293000	0.025	7.9	16.20	0.37	0.37	100.0	Excellent

Figure 2.2.3.1. Raw Initial Dataset

b. Data Processing Scheme and Algorithms

Preprocessing the dataset included standardization of the date column to separate the year, month, and day into their own respective columns. Afterwards, data points that do not correspond to freshwater body types were removed. Additionally, only the data points recorded since the year 2000 were included for development. Afterwards, the resulting dataframe was checked for duplicate rows and missing values.

Afterwards the dataframe was checked for outliers, and the columns that will be relevant for the project were selected. Three features were selected, namely: Ammonia, pH, and Nitrate levels. Based on these columns, the data points were identified as potential outliers based on the interquartile range, with the lower and upper bounds being defined as follows:

$$IQR = Q3 - Q1 \quad \text{Equation No. 2.2.3.1}$$

$$Lower\ Bound = Q1 - IQR \quad \text{Equation No. 2.2.3.2}$$

$$Upper\ Bound = Q3 + IQR \quad \text{Equation No. 2.2.3.3}$$

Wherein:

IQR - Interquartile Range; Q3 - 3rd Quartile; Q1 - 1st Quartile

Outliers within the dataframe were filtered out to keep only the data points that lie between the lower and upper bound. This procedure of filtering was done with the use of the three features aforementioned. The resulting dataframe had 1,674,462 data points, from 2,827,977 data points from the original dataframe.

	Ammonia (mg/l)	pH (ph units)	Nitrate (mg/l)
26	0.05152	8.3700	9.73940
28	0.07728	8.0167	8.72119
29	0.09016	7.7900	9.51805
30	0.10304	8.1583	8.63265
31	0.10304	7.7900	8.76546

Figure 2.2.3.2. Preprocessed Dataframe

After preprocessing, reclassification through K-means clustering was done, as the original dataset had five classifications, and based on the Department of Environment and Natural Resources (2016a) guidelines, the data points can be classified into four classes instead, ranging from Class A to Class D, with Class D considered to be of lowest quality.

Upon clustering, it was found that the inertia when only using four clusters was high, which implies that the data points for each cluster were so far apart from each other in terms of similarity based on their features. Using the elbow method, the optimal number of clusters were determined, as shown in Figure 2.2.3.4. While 12 clusters was considered optimal, 16 clusters were used instead due to a significantly lower inertia.

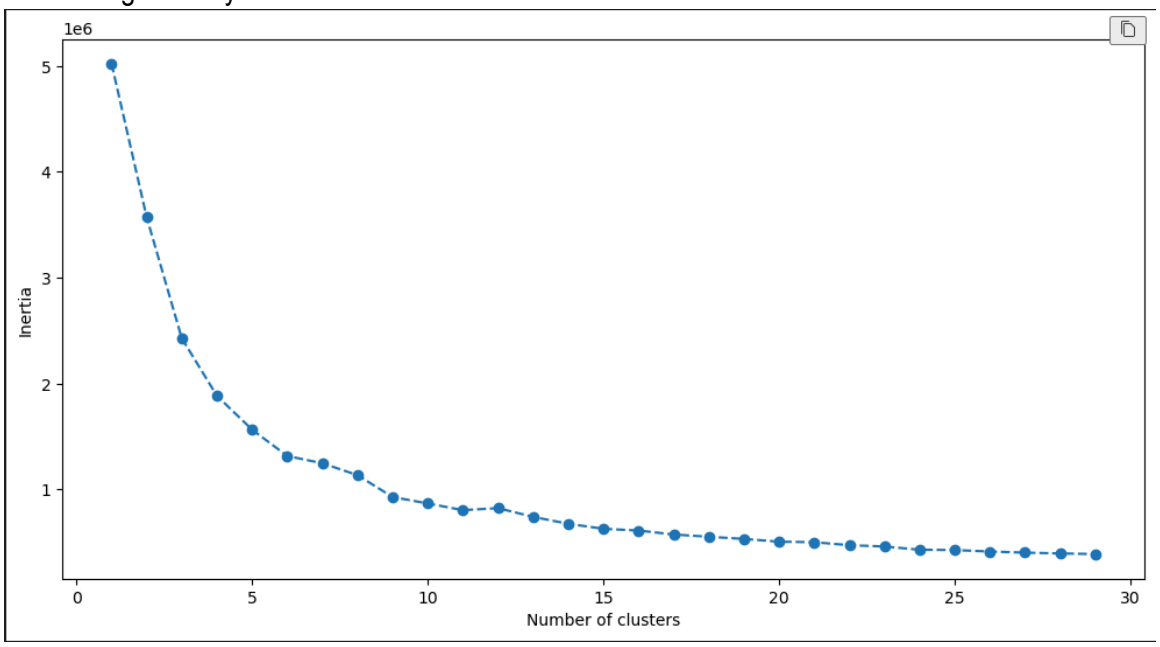


Figure 2.2.3.4. Elbow Method Results.

The clusters were then used to label the data points based on their features, with the intention of ranking the cluster with the lowest average ammonia and nitrate levels and lowest pH variability to be labeled as the highest class (class A1). This preference is based on studies indicating that water contaminated with high levels of ammonia and nitrate pose serious health risks (Suaebu et al., 2025), while lower variability in pH was preferred to indicate that the pH levels of each of the data points within clusters are not far apart from each other . 16 clusters would mean that for each water quality class, their would be 4 subclasses. To label the data points, each cluster's details were summarized, primarily to determine the mean Ammonia and Nitrate concentration levels, and the standard deviation of the pH levels. Based on the aforementioned preferences, a composite score was computed, based on the following formula:

$$score = Nitrate\ mean - Ammonia\ mean - pH\ std$$ Equation No. xx

cluster	pH_std	Ammonia_mean	Nitrate_mean	score
1	0.070663	0.041362	1.047175	-1.159200
4	0.115417	0.056083	1.135499	-1.306999
8	0.080833	0.046046	1.314016	-1.440895
14	0.076815	0.051962	1.349588	-1.478365
9	0.122387	0.366066	2.805336	-3.293789
7	0.072822	0.087181	4.278805	-4.438808
13	0.115371	0.089248	4.380934	-4.585553
0	0.074765	0.071987	4.445383	-4.592135
10	0.080686	0.058146	4.542783	-4.681614
11	0.129915	0.536051	4.412280	-5.078247
12	0.145242	0.490943	4.734382	-5.370568
2	0.102108	0.629339	4.731246	-5.462693
6	0.090227	0.066449	6.491260	-6.647935
15	0.106357	0.061430	7.691113	-7.858901
5	0.128749	0.111813	7.881060	-8.121623
3	0.100816	0.090566	8.826008	-9.017390

Figure 2.2.3.5. Clusters Ranked Based on Composite Score

The clusters are then ranked based on the composite score, and the data points within each cluster are then labeled with their respective classes. Afterwards, the resulting dataframe was exported into a parquet file, which can then be loaded and prepared for training.

	Ammonia (mg/l)	pH (ph units)	Nitrate (mg/l)	cluster	class_label
0	0.051520	8.370000	9.739400	15	D2
1	0.077280	8.016700	8.721190	3	D4
7	0.094024	7.790000	3.023641	7	B2
8	0.014039	8.327270	5.836820	10	C1
9	0.046368	7.619440	5.825932	6	D1
10	0.097888	8.337500	1.651271	8	A3
13	0.055813	7.790000	2.331555	1	A1
18	0.069552	8.001700	0.447127	14	A4
22	0.303066	7.790000	0.451997	9	B1
28	0.455093	7.880000	6.654135	12	C3
38	0.009789	7.447200	0.690612	4	A2
50	0.108836	7.896460	3.877167	0	B4
59	0.274344	7.290000	5.836820	13	B3
67	0.482485	7.420800	5.304874	11	C2
79	0.601633	7.796667	6.302567	2	C4
95	0.137478	7.669167	7.639811	5	D3

Figure 2.2.3.6. Preprocessed Dataset

2.3 Design Alternatives

2.3.1 Rationale for Design Alternatives

The three design alternatives used for this project were CatBoost, XGBoost, and LightGBM, all of which were gradient boosting models with different strengths and weaknesses in terms of accomplishing the tasks to reach the design objectives as well as adhere to client requirements. CatBoost is a gradient boosting model optimized for classification even with less resources, while XGBoost and LightGBM are built to be efficient in training with large amounts of data.

2.3.1 Design Alternative 1: LightGBM

A. Engineering Principles of Alternative

Gradient boosting is an ensemble machine learning technique that builds a strong predictive model by sequentially combining multiple weak learners (Clark & Lee, 2025). At its core, gradient boosting minimizes a loss function through iterative optimization. Rather than reweighting samples as in AdaBoost, each new tree is trained to predict the negative gradient of the loss function with respect to the predictions of the current model. This gradient represents the direction and magnitude of errors that the existing model makes. The predictions from the new tree are then added to the ensemble, scaled by a learning rate parameter that controls the contribution strength of each tree.

By focusing on the remaining errors at each iteration, gradient boosting creates a powerful composite model where the final prediction is the sum of contributions from all trees.

The loss function flexibility is a key strength of gradient boosting, since the framework can handle regression, binary classification, and multiclass problems by simply changing the loss function (mean squared error for regression, log-loss for classification). However, this sequential nature makes training computationally expensive, and without proper regularization, gradient boosting models can overfit, particularly on noisy datasets. (Belyadi & Haghighat, 2021)

B. Architecture of Design Alternative

The defining architectural characteristic of LightGBM is its leaf-wise (best-first) tree growth strategy, which contrasts sharply with the level-wise approach used by XGBoost. For a fixed number of leaves, leaf-wise algorithms achieve lower loss than level-wise algorithms because they concentrate modeling capacity where error reduction is most significant. This results in deeper, narrower, more asymmetric trees that can capture complex patterns with fewer total nodes, leading to faster convergence, which means that fewer trees are needed for equivalent accuracy.

LightGBM incorporates a critical efficiency innovation called Gradient-based One-Side Sampling (GOSS). GOSS accelerates training by retaining all samples with large gradient magnitudes, which correspond to data points the model currently predicts poorly and therefore contain the most useful information for updating the model. It then randomly downsamples the samples with small gradients, as these well-predicted points contribute less to determining the best tree splits. By focusing computation on the most informative samples and applying proper weighting to the remaining ones, GOSS significantly reduces training time while preserving model accuracy on large datasets.

LightGBM employs histogram-based discretization rather than pre-sorting continuous features, reducing memory usage and computational complexity. Features are binned into discrete ranges, and only gradient statistics within each bin are stored, dramatically reducing memory overhead while maintaining split-finding quality.

C. Constraints

Safety

Safety measures how often the system misclassifies each data instance. This can affect how accurate the system classifies the samples. As such, a misclassification rate of more than 5% means that the system is deemed insufficient to be used.

Metric	Percentage (%)
Misclassification Rate	0.598 %

Performance

Performance measures how fast the system is when predicting and classifying a new instance of data. This can affect how fast the system returns an output to the user.

Metric	Time (s)
Inference Time	8.01422

Manufacturability
Manufacturability measures how fast the system can be utilized from scratch. This involves the amount of time that the system's model takes to train before being introduced to a new instance of data. A slow training time would be detrimental in updating the model in future stages of development.

Metric	Time (s)
Training Time	22.84536

Compatibility
Compatibility measures how clean the algorithm of the model is used to predict and display results. A compatibility score of less than 50 means that the system is deemed insufficient for usage.

Metric	Score
Maintainability Index	78.32318594

Efficiency
Efficiency constraint ensures the system operates with minimal computational resources and runs fast for large inputs. In the project's context, storage consumption is used to determine the amount of storage space a machine learning model uses upon being stored. Higher values mean that the model will take up a lot of space within the system.

Metric	Size (MB)
Storage Consumption	21.4742

2.3.2 Design Alternative B: CatBoost

A. Engineering Principles of Alternative
Gradient boosting is an ensemble machine learning technique that builds a strong predictive model by sequentially combining multiple weak learners (Clark & Lee, 2025). At its core, gradient boosting minimizes a loss function through iterative optimization. Rather than reweighting samples as in AdaBoost, each new tree is trained to predict the negative gradient of the loss function with respect to the predictions of the current model. This gradient represents the direction and magnitude of errors that the existing model makes. The predictions from the new tree are then added to the ensemble, scaled by a learning rate parameter that controls the contribution strength of each tree.

By focusing on the remaining errors at each iteration, gradient boosting creates a powerful composite model where the final prediction is the sum of contributions from all trees.

The loss function flexibility is a key strength of gradient boosting, since the framework can handle regression, binary classification, and multiclass problems by simply changing the loss function (mean squared error for regression, log-loss for classification). However, this sequential nature makes training computationally expensive, and without proper regularization, gradient boosting models can overfit, particularly on noisy datasets. (Belyadi & Haghighat, 2021)

B. Architecture of Design Alternative

CatBoost implements ordered boosting, a permutation-driven alternative to classical gradient boosting that addresses a subtle statistical issue: prediction shift. In standard boosting, each tree is trained to predict residuals computed using the same dataset used to determine the tree structure, introducing optimistic bias. Ordered boosting mitigates this by using different permutations of the training data during different phases of tree construction, ensuring that residual calculations and tree structure decisions operate on independent data subsets, leading to more unbiased gradient estimates and improved generalization

A distinctive feature of CatBoost is its ability to achieve strong performance with minimal manual hyperparameter tuning. The library provides sensible defaults and robust automatic mechanisms for most parameters, making it more accessible for practitioners without extensive machine learning expertise.

C. Constraints

Safety

Safety measures how often the system misclassifies each data instance. This can affect how accurate the system classifies the samples. As such, a misclassification rate of more than 5% means that the system is deemed insufficient to be used.

Metric	Percentage (%)
Misclassification Rate	0.702 %

Performance

Performance measures how fast the system is when predicting and classifying a new instance of data. This can affect how fast the system returns an output to the user.

Metric	Time (s)
Inference Time	0.22848

Manufacturability

Manufacturability measures how fast the system can be utilized from scratch. This involves the amount of time that the system's model takes to train before being introduced to a new instance of

data. A slow training time would be detrimental in updating the model in future stages of development.

Metric	Time (s)
Training Time	112.0747

Compatibility
Compatibility measures how clean the algorithm of the model is used to predict and display results. A compatibility score of less than 50 means that the system is deemed insufficient for usage.

Metric	Score
Maintainability Index	78.10722314

Efficiency
Efficiency constraint ensures the system operates with minimal computational resources and runs fast for large inputs. In the project's context, storage consumption is used to determine the amount of storage space a machine learning model uses upon being stored. Higher values mean that the model will take up a lot of space within the system.

Metric	Size (MB)
Storage Consumption	6.6758

2.3.3 Design Alternative C: XGBoost

A. Engineering Principles of Alternative
Gradient boosting is an ensemble machine learning technique that builds a strong predictive model by sequentially combining multiple weak learners (Clark & Lee, 2025). At its core, gradient boosting minimizes a loss function through iterative optimization. Rather than reweighting samples as in AdaBoost, each new tree is trained to predict the negative gradient of the loss function with respect to the predictions of the current model. This gradient represents the direction and magnitude of errors that the existing model makes. The predictions from the new tree are then added to the ensemble, scaled by a learning rate parameter that controls the contribution strength of each tree. By focusing on the remaining errors at each iteration, gradient boosting creates a powerful composite model where the final prediction is the sum of contributions from all trees.

The loss function flexibility is a key strength of gradient boosting, since the framework can handle regression, binary classification, and multiclass problems by simply changing the loss function (mean squared error for regression, log-loss for classification). However, this sequential nature makes training computationally expensive, and without proper regularization, gradient boosting models can overfit, particularly on noisy datasets. (Belyadi & Haghighat, 2021)

B. Architecture of Design Alternative

XGBoost employs a level-wise (depth-first, breadth-first) tree growth strategy. At each iteration, it evaluates all possible splits at the current tree depth for every feature. Then, it calculates information gain for each potential split by computing how much the split would reduce the objective function. Then, it selects the best split at each node based on the maximum gain. It advances to the next level or iteration only after all nodes at the current depth have been processed.

This systematic, level-by-level approach ensures trees remain balanced with uniform leaf depths, making them more interpretable and preventing pathological tree shapes. The level-wise strategy allows parallelization within each tree level, wherein multiple splits at the same depth can be evaluated simultaneously across processors, providing computational efficiency on multi-core systems.

C. Constraints

Safety

Safety measures how often the system misclassifies each data instance. This can affect how accurate the system classifies the samples. As such, a misclassification rate of more than 5% means that the system is deemed insufficient to be used.

Metric	Percentage (%)
Misclassification Rate	0.772%

Performance

Performance measures how fast the system is when predicting and classifying a new instance of data. This can affect how fast the system returns an output to the user.

Metric	Time (s)
Inference Time	1.1537

Manufacturability

Manufacturability measures how fast the system can be utilized from scratch. This involves the amount of time that the system's model takes to train before being introduced to a new instance of data. A slow training time would be detrimental in updating the model in future stages of development.

Metric	Time (s)
Training Time	21.80616

Compatibility
Compatibility measures how clean the algorithm of the model is used to predict and display results. A compatibility score of less than 50 means that the system is deemed insufficient for usage.

Metric	Score
Maintainability Index	77.10482265

Efficiency
Efficiency constraint ensures the system operates with minimal computational resources and runs fast for large inputs. In the project's context, storage consumption is used to determine the amount of storage space a machine learning model uses upon being stored. Higher values mean that the model will take up a lot of space within the system.

Metric	Size (MB)
Storage Consumption	6.6122

2.4 Standards Involved in the Design

Table 2.4.1. Summary of Standards Involved in the Alternatives

Standard	Brief Description	DESIGNS		
		DESIGN A	DESIGN B	DESIGN C
ISO 8601 (Date and Time Format)	International standard for unambiguous date and time representation (e.g., YYYY-MM-DDTHH:mm :ssZ) in logical, hierarchical order.	Ensures all timestamps attached to water sample data, sensor readings, and predictions are standardized, which is essential for reliable chronological analysis, model training, and interpretation across systems		
ANSI/IEEE 1012 (Software Verification and Validation)	Standard for establishing systematic verification and validation processes to ensure software quality and fitness for purpose.	Provides a framework for rigorously testing, validating, and documenting ML models and their pipelines, improving both reliability and regulatory compliance		

WHO Guidelines for Drinking Water Quality	Global reference for health-based water quality targets, covering chemical, microbiological, and physical properties of drinking water.	Establishes threshold values and risk-based classification logic for ML model labeling (e.g., safe/unsafe for drinking), ensuring health significance
Philippine National Standards for Drinking Water (PNSDW)	Sets national criteria for physical, chemical, microbiological, and radiological elements in drinking water for public health protection.	Used to create ground truth labels in training data, define output categories, and evaluate model outputs according to national standards
DAO 2016-08 Water Quality Guidelines and General Effluent Standards	Regulatory framework specifying water quality thresholds and effluent discharge standards in the Philippines.	Informs feature selection, thresholding, and output interpretability for models predicting compliance or classifying water samples by their beneficial use, including legal acceptability

Each standard in the table plays a unique supporting role in building and running machine learning models for water quality prediction and classification that are accurate and compliant with existing safety standards. ISO 8601 brings order to time-stamped data, ANSI/IEEE 1012 ensures software correctness, and the various water quality standards (WHO, PNSDW, DAO 2016-08) set the health and legal targets the model's predictions must meet. Collectively, they establish the data formats, safety requirements, validation processes, and regulatory thresholds needed to make model results both accurate and actionable, addressing both public health and compliance in water management.

CHAPTER 3: DESIGN TRADEOFFS

This section discusses design constraints, trade-offs, and normalization of scores for the project. It outlines the quantitative metrics and constraints discussed in the previous sections. Trade-off options are also highlighted in this section to choose the best design for the project.

3.1 Summary of Constraints

Table 3.1 summarizes the scores and values garnered from testing the designs.

Table 3.1 Summary of Design Constraints

Designs	Constraints				
	Safety (Misclassification Rate)	Performance (Inference Time)	Manufacturability (Training Time)	Compatibility (Maintainability Index Score)	Efficiency (Storage Consumption)
LightGBM	0.00598	8.01422	22.84536	78.32318594	21.4742
CatBoost	0.00702	0.22848	112.0747	78.10722314	6.6758
XGBoost	0.00772	1.1537	21.80616	77.10482265	6.6122

3.2 Trade-offs

Table 3.2 Preference and Importance of Constraints

Constraints	Preference	Importance (raw)	% Importance
Safety	Minimization	10	25.00%
Performance	Minimization	9	22.50%
Manufacturability	Minimization	8	20.00%

Compatibility	Maximization	7	17.50%
Efficiency	Minimization	6	15.00%

Pareto multi-criteria decision making is applied in tradeoff-analysis to evaluate the system's three design alternatives. The alternative is scored depending on how well each alternative meets each requirement. Each constraint of the alternatives is given a scale of 1 through 10 on the basis of its value to calculate the score.

Each constraint will be normalized based on the ranking. The formula to get minimization rank is as follows::

$$PC_{norm} = 9 \times \left(\frac{Max\,Value - PC_{raw}}{Max\,Value - Min\,Value} \right) + 1 \qquad \text{Equation No. 3.2.1}$$

Where:

$PC_{norm} = normalized\ value\ of\ criteria$

$PC_{raw} = raw\ value\ of\ the\ criteria\ to\ be\ normalized$

$Min_{raw} = smallest\ possible\ value\ of\ the\ criteria$

$Max_{raw} = largest\ possible\ value\ of\ the\ criteria$

And, the formula used to compute for maximization rank is as follows:

$$PC_{norm} = 9 \times \left(\frac{PC_{raw} - Min\,Value}{Max\,Value - Min\,Value} \right) + 1 \qquad \text{Equation No. 3.2.2}$$

Where:

$PC_{norm} = normalized\ value\ of\ criteria$

$PC_{raw} = raw\ value\ of\ the\ criteria\ to\ be\ normalized$

$Min_{raw} = smallest\ possible\ value\ of\ the\ criteria$

$Max_{raw} = largest\ possible\ value\ of\ the\ criteria$

Given that each constraint is given a level of importance, the normalized scores can be used with it to determine the best design among the three design alternatives. The formula for getting the percentage of importance is as follows:

$$\%_{cn} = \frac{Importance\ (raw)}{sum(Importance(raw))} x\ 100 \qquad \text{Equation No. 3.2.3}$$

$\%_{cn} = percentage\ importance\ of\ the\ constraint$

3.2.1 Tradeoff 1: Safety (Misclassification Rate)

Table 3.2.1 Evaluation of Three Design Alternatives based on Safety

Design	Safety (Misclassification Rate)
LightGBM	0.00598
CatBoost	0.00702
XGBoost	0.00772

The scores in Table 3.2.1 will be subject to normalization detailed in the succeeding sections. It can be seen that all three design alternatives were able to achieve a misclassification rate less than 0.05 or 5 percent, demonstrating their accuracy in predicting and classifying water samples.

3.2.1.1 Design 1: Normalization of Safety (Misclassification Rate)

$$Minimization = 9 \times \left(\frac{0.00772 - 0.00598}{0.00772 - 0.00598}\right) + 1 = 10$$

3.2.1.2 Design 2: Normalization of Safety (Misclassification Rate)

$$Minimization = 9 \times \left(\frac{0.00772 - 0.00702}{0.0194 - 0.01745}\right) + 1 = 4.620689655$$

3.2.1.3 Design 3: Normalization of Safety (Misclassification Rate)

$$Minimization = 9 \times \left(\frac{0.00772 - 0.00702}{0.0194 - 0.01745}\right) + 1 = 1$$

3.2.2 Tradeoff 2: Performance (Inference Time)

Table 3.2.2 Evaluation of Three Design Alternatives based on Performance

Design	Performance (Inference Time)
LightGBM	8.01422
CatBoost	0.22848
XGBoost	1.1537

The scores in Table 3.2.2 will be subject to normalization detailed in the succeeding sections.

3.2.2.1 Design 1: Normalization of Performance (Inference Time)

$$Minimization = 9 \times \left(\frac{8.01422 - 8.01422}{8.01422 - 0.22848}\right) + 1 = 1$$

3.2.2.2 Design 2: Normalization of Performance (Inference Time)

$$Minimization = 9 \times \left(\frac{8.01422 - 0.22848}{8.01422 - 0.22848}\right) + 1 = 10$$

3.2.2.3 Design 3: Normalization of Performance (Inference Time)

$$Minimization = 9 \times \left(\frac{8.01422 - 1.1537}{8.01422 - 0.22848}\right) + 1 = 8.930483165$$

3.2.3 Tradeoff 3: Manufacturability (Training Time)

Table 3.2.3 Evaluation of Three Design Alternatives based on Manufacturability

Design	Manufacturability (Training Time)
LightGBM	22.84536
CatBoost	112.0747
XGBoost	21.80616

The scores in Table 3.2.3 will be subject to normalization detailed in the succeeding sections. The training times of the three designs indicate their capability to produce results or classify incoming data in less than ten minutes. This also implies that the designs can be trained as quick as possible to be used in the field at a much faster rate.

3.2.3.1 Design 1: Normalization of Manufacturability (Training Time)

$$Minimization = 9 \times \left(\frac{112.0747 - 22.84536}{112.0747 - 21.80616}\right) + 1 = 9.896389152$$

3.2.3.2 Design 2: Normalization of Manufacturability (Training Time)

$$Minimization = 9 \times \left(\frac{112.0747 - 112.0747}{112.0747 - 21.80616}\right) + 1 = 1$$

3.2.3.3 Design 3: Normalization of Manufacturability (Training Time)

$$Minimization = 9 \times \left(\frac{112.0747 - 21.80616}{112.0747 - 21.80616}\right) + 1 = 10$$

3.2.4 Tradeoff 4: Compatibility (Maintainability Index Score)

Table 3.2.4 Evaluation of Three Design Alternatives based on Compatibility

Design	Compatibility (Maintainability Index Score)
LightGBM	78.32318594
CatBoost	78.10722314

XGBoost	77.10482265

The scores in Table 3.2.4 will be subject to normalization detailed in the succeeding sections.

3.2.4.1 Design 1: Normalization of Compatibility (Maintainability Index Score)

$$Maximization = 9 \times \left(\frac{78.32318594 - 77.10482265}{78.32318594 - 77.10482265} \right) + 1 = 1$$

3.2.4.2 Design 2: Normalization of Compatibility (Maintainability Index Score)

$$Maximization = 9 \times \left(\frac{78.10722314 - 77.10482265}{78.32318594 - 77.10482265} \right) + 1 = 8.404691616$$

3.2.4.3 Design 3: Normalization of Compatibility (Maintainability Index Score)

$$Maximization = 9 \times \left(\frac{77.10482265 - 77.10482265}{78.32318594 - 77.10482265} \right) + 1 = 1$$

3.2.5 Tradeoff 5: Efficiency (Storage Consumption)

Table 3.2.5 Evaluation of Three Design Alternatives based on Efficiency

Design	Efficiency (Storage Consumption)
LightGBM	21.4742
CatBoost	6.6758
XGBoost	6.6122

The scores in Table 3.2.5 will be subject to normalization detailed in the succeeding sections.

3.2.5.1 Design 1: Normalization of Efficiency (Storage Consumption)

$$Minimization = 9 \times \left(\frac{21.4742 - 21.4742}{21.4742 - 6.6122} \right) + 1 = 1$$

3.2.5.2 Design 2: Normalization of Efficiency (Storage Consumption)

$$Minimization = 9 \times \left(\frac{21.4742 - 6.6758}{21.4742 - 6.6122} \right) + 1 = 9.961485668$$

3.2.5.3 Design 3: Normalization of Efficiency (Storage Consumption)

$$Minimization = 9 \times \left(\frac{21.4742 - 6.6122}{21.4742 - 6.6122} \right) + 1 = 10$$

3.3 Summary of the Normalized Values of the Three Designs

Table 3.3 Summary of the Normalized Values of the Three Designs

Designs	Constraints				
	Safety (Misclassification Rate)	Performance (Inference Time)	Manufacturability (Training Time)	Compatibility (Maintainability Index Score)	Efficiency (Storage Consumption)
LightGBM	10	1	9.896389152	10	1
CatBoost	4.620689655	10	1	8.404691616	9.961485668
XGBoost	1	8.930483165	10	1	10

3.4 Designers Raw Ranking for the Three Designs

Table 3.4 Designers Raw Ranking for the Three Designs

Decision Criteria	Criterion's Importance		Ability to Satisfy Criterion		
	Scale (0-10)	Percentage (%)	LightGBM	CatBoost	XGBoost
Safety (Misclassification Rate)	10	25.00%	10	4.620689655	1
Performance (Inference Time)	9	22.50%	1	10	8.930483165
Manufacturability (Training Time)	8	20.00%	9.896389152	1	10
Compatibility (Maintainability Index Score)	7	17.50%	10	8.404691616	1
Efficiency (Storage Consumption)	6	15.00%	1	9.961485668	10
TOTAL	40	100%	6.60427783	6.570216297	5.934358712

3.5 Sensitivity Analysis

The score values of each alternative was determined through sensitivity analysis, taking into account 120 distinct combinations of the five constraints. It facilitates the display of the best design based on the level of importance of the criteria or constraint. To carry out the sensitivity analysis, the scores of each alternative design across 120 combinations of the constraints' criterion of relevance has to be determined. The computation of the scores was made possible by utilizing a Python notebook (see Appendix I5). The notebook produced a CSV file containing 120 scores for every other design. Then, a radar chart was used to visualize the results of the sensitivity analysis.

Figure 3.5. Illustrates the sensitivity analysis for the three designs. The blue line represents the first design, LightGBM, which shows fluctuating performance, but dominating in some instances and in most instances, closely comparable to the design represented by the red line. The red line represents the second design, CatBoost, which shows consistent performance across all instances, but is surpassed by the first design in some instances. Finally, the green line represents the third design, XGBoost, which shows the lowest performance in most instances, but marginally surpasses the other two designs in a few instances. Overall, it can be inferred that the three designs, particularly Designs 1 and 2, have varying strengths and weaknesses given the constraints, and depending on the constraint that is most values, any of these designs can be te best design.

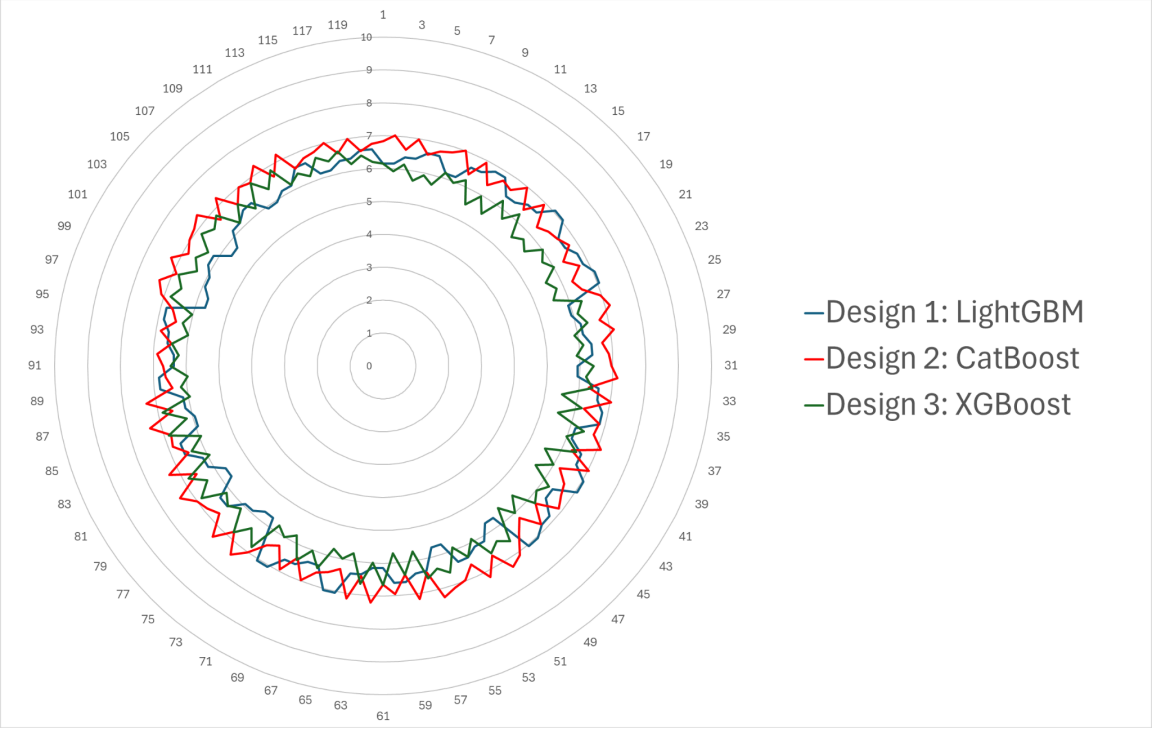


Figure 3.5. Results of Sensitivity Analysis

From the sensitivity analysis, it can be concluded that Design 2 emerged as the winning design in most instances. Upon getting the average score of each design across 120 instances or combinations, Design 2 came out on top, with an average score of 6.797373383, narrowly edging out Designs 1 and 3, which averaged 6.37927783 and 6.186096633, respectively. This shows that

the CatBoost algorithm is the best fit for on-site water classification via colorimetry, considering the constraints of Safety (Misclassification Rate), Performance (Inference Time), Manufacturability (Training Time), Compatibility (Maintainability Index Score), and Efficiency (Storage Consumption).

3.6 Influence of the Design Tradeoffs in the Final Design

Evaluating the different tradeoffs in each design helped in determining the most optimal design that can attain the project's objectives and adhere with the client's requirements given the constraints. The computations of the tradeoff show CatBoost as the winning model for the design.

References

Alhaqi, M. (2025). Smartphone camera-based prediction of water pH using multispectral image simulation and machine learning models. *Journal of Information Systems Engineering and Management*, *10*, 324–333. https://doi.org/10.52783/jisem.v10i17s.2729

Barghoth, M. E., Salah, A., & Ismail, M. A. (2020). A Comprehensive Software Project Management Framework. *Journal of Computer and Communications*, *08*(03), 86–102. https://doi.org/10.4236/jcc.2020.83009

Belyadi, H., & Haghighat, A. (2021). Supervised learning. *Machine Learning Guide for Oil and Gas Using Python*, 169–295. https://doi.org/10.1016/b978-0-12-821929-4.00004-4

Chiaramonte, M. (2024, September 20). *Developing modular software: Top strategies and best practices - vFunction*. VFunction. https://vfunction.com/blog/modular-software/

Clark, B., & Lee, F. (2025, April 7). *What is Gradient Boosting?* Ibm.com. https://www.ibm.com/think/topics/gradient-boosting

Dabrowska, A., Lewis, G. R., Minaleshewa Atlabachew, Salter, S. J., Henderson, C., Ji, C., Ehlers, A., Stirling, J., Mower, S., Allen, L., Lay, E., Stuart, K., Appavou, L., Bowman, R., Zhao, T., Patel, N., Patto, A., Holmes, M. A., Baumberg, J. J., & Mahdi, S. (2024). Expanding access to water quality monitoring with the open-source WaterScope testing platform. *Npj Clean Water*, *7*(1). https://doi.org/10.1038/s41545-024-00357-y

Department of Environment and Natural Resources. (2016a). *DAO 2016-08 Water Quality Guidelines and General Effluent Standards*. https://emb.gov.ph/wp-content/uploads/2019/04/DAO-2016-08_WATER-QUALITY-GUIDELINES-AND-GENERAL-EFFLUENT-STANDARDS.pdf

Department of Environment and Natural Resources. (2016b). *Water Quality Guidelines and General Effluent Standards of 2016*. https://pab.emb.gov.ph/wp-content/uploads/2017/07/DAO-2016-08-WQG-and-GES.

Department of Health. (2017). *Philippine National Standards for Drinking Water of 2017*. https://leadservlab.com/2020/ao2017-10_PNSDW.pdf

Doğan, V., Isık, T., & Horzum, N. (2022). A field-deployable water quality monitoring with machine learning-based smartphone colorimetry. *Analytical Methods*, *14*. https://doi.org/10.1039/D2AY00785A

EPA. (2024, December 12). *National Primary Drinking Water Regulations | US EPA*. US EPA. https://www.epa.gov/ground-water-and-drinking-water/national-primary-drinking-water-regulations

Filipenco, D. (2024, April 10). *Water pollution in the Philippines*. DevelopmentAid. https://www.developmentaid.org/news-stream/post/155108/water-pollution-in-the-philippines

Fukunaga, M., Ishihara, T., Saito, K., Yamada, K., Enoki, H., Mori, S., Miyake, R., Terayama, T., & Kanamaru, M. (2001, April 10). *Water quality meter and water quality monitoring system*. https://patents.google.com/patent/US6444172B2/en

IBM. (2021, September 22). *What Is Machine learning?* IBM; IBM. https://www.ibm.com/think/topics/machine-learning

IEEE. (2017). *IEEE Standard for System, Software, and Hardware Verification and Validation*. https://standards.ieee.org/ieee/1012/5609/

ISO. (n.d.). *ISO - ISO 8601 — Date and time format*. ISO. Retrieved December 6, 2025, from https://www.iso.org/iso-8601-date-and-time-format.html

Kactoily. (2025). *Kactoily 7-in-1 Portable Water Tester*. Kactoily. https://kactoily.com/products/kactoily-7-in-1-portable-water-tester

Karim, M. R., Syeed, M., Rahman, A., Rabbani, K. A., Fatema, K., Khan, R. H., Hossain, S., & Uddin, M. F. (2025). *A Comprehensive Surface Water Quality Monitoring Dataset (1940-2023): 2.82Million Record Resource for Empirical and ML-Based Research*. https://doi.org/10.6084/m9.figshare.27800394

Kılıç, V., Alankus, G., Horzum, N., Mutlu, A. Y., Bayram, A., & Solmaz, M. E. (2018). Single-Image-Referenced Colorimetric Water Quality Detection Using a Smartphone. *ACS Omega*, *3*(5), 5531–5536. https://doi.org/10.1021/acsomega.8b00625

Kleinjans, J. C., Albering, H. J., Marx, A., Maanen, van, Agen, B. van, F ten Hoor, Swaen, G. M., & Mertens, P. L. (1991). Nitrate contamination of drinking water: evaluation of genotoxic risk in human populations. *Environmental Health Perspectives*, *94*, 189–193. https://doi.org/10.1289/ehp.94-1567968

Knobeloch, L., Salna, B., Hogan, A., Postle, J., & Anderson, H. (2000). Blue babies and nitrate-contaminated well water. *Environmental Health Perspectives*, *108*(7), 675–678. https://doi.org/10.1289/ehp.00108675

Kunz, J., Lawinger, H., Miko, S., Gerdes, M., Thuneibat, M., Hannapel, E., & Roberts, V. (2024). Surveillance of waterborne disease outbreaks associated with drinking water - united states, 2015-2020. *Morbidity and Mortality Weekly Report. Surveillance Summaries (Washington, D.C. : 2002)*, *73*(4), 1–23. https://doi.org/10.15585/mmwr.ss7301a1

Lawinger, H., Hlavsa, M., Miko, S., Kunz, J., Thuneibat, M., Gerdes, M., Gleason, M., & Roberts, V. (2023). *Waterborne disease and outbreak surveillance system (WBDOSS) summary report, united states, 2021*. U.S. Department of Health and Human Services, Centers for Disease Control and Prevention.

https://www.cdc.gov/healthy-water-data/media/pdfs/2024/04/2021_Annual_Waterborne_Disease_Surveillance_Report.pdf

Lozano Wilches, L., Jantarakasem, C., Sioné, L., Templeton, M., & Mikolajczyk, K. (2022). Estimating water turbidity from a smartphone camera. In *The 33rd British Machine Vision Conference Proceedings*. https://bmvc2022.mpi-inf.mpg.de/0880.pdf

Minnesota Department of Health. (2017). *Nitrate in Drinking Water*. State.mn.us; Minnesota Department of Health. https://www.health.state.mn.us/communities/environment/water/contaminants/nitrate.html

Niu, X., Cheng, N., Du, D., & Lin, Y. (2022). *Smartphone-based sensors for on-site water quality monitoring* (pp. 331–348). https://doi.org/10.1142/9789811245770_0011

Picetti, R., Deeney, M., Pastorino, S., Miller, M. R., Shah, A., Leon, D. A., Dangour, A. D., & Green, R. (2022). Nitrate and nitrite contamination in drinking water and cancer risk: A systematic review with meta-analysis. *Environmental Research*, *210*, 112988–112988. https://doi.org/10.1016/j.envres.2022.112988

Pries, C. N. (1981). Reproduction effects of occupational exposures. *American Family Physician*, *24*(2), 161–165. https://pubmed.ncbi.nlm.nih.gov/7258079/

Shaofeng, L., Ran, Z., Cui, C., & Yixing, Y. (2011). *Study on the inactivation of cryptosporidium and giardia by chlorine dioxide in water*. 1819–1822. https://doi.org/10.1109/CDCIEM.2011.469

Shu, X., & Ye, Y. (2022). Knowledge Discovery: Methods from data mining and machine learning. *Social Science Research*, *110*, 102817. https://doi.org/10.1016/j.ssresearch.2022.102817

Srivastava, S., Vaddadi, S., & Sadistap, S. (2018). Smartphone-based System for water quality analysis. *Applied Water Science*, *8*. https://doi.org/10.1007/s13201-018-0780-0

Suaebu, S., Daud, A., Mallongi, A., Wahyu, A., Amiruddin, R., Wahiduddin, W., & Birawida, A. B. (2025). Health Risks Due to Exposure Nitrate (NO3) and Ammonia (NH3) in Local Communities Final Disposal of Waste in Makassar City. *International Journal of Environmental Impacts*, *8*(4), 825–835. https://doi.org/10.18280/ijei.080420

Tseng, C.-H., Lee, I-Hsuan., & Chen, Y.-C. (2019). Evaluation of hexavalent chromium concentration in water and its health risk with a system dynamics model. *Science of the Total Environment*, *669*, 103–111. https://doi.org/10.1016/j.scitotenv.2019.03.103

United States Environmental Protection Agency. (2024, July 8). *Drinking water*. US EPA. https://www.epa.gov/report-environment/drinking-water

US EPA. (2015, September 21). *Drinking Water Regulations | US EPA*. US EPA. https://www.epa.gov/dwreginfo/drinking-water-regulations

Ward, M. H., deKok, T. M., Levallois, P., Brender, J., Gulis, G., Nolan, B. T., & VanDerslice, J. (2005). Workgroup Report: Drinking-Water Nitrate and Health—Recent Findings and Research Needs. *Environmental Health Perspectives*, *113*(11), 1607–1614. https://doi.org/10.1289/ehp.8043

Ward, M., Jones, R., Brender, J., De Kok, T., Weyer, P., Nolan, B., Villanueva, C., & Van Breda, S. (2018). Drinking Water Nitrate and Human Health: An Updated Review. *International Journal of Environmental Research and Public Health*, *15*(7), 1557. https://doi.org/10.3390/ijerph15071557

Water Environment Partnership in Asia (WEPA). (2025). *2024 Outlook on Water Environmental Management in Asia* (pp. 129–131). https://wepa-db.net/wp-content/uploads/2025/05/WEPA_Outlook2024_EN_Web-a.pdf

World Health Organization. (2017). *Guidelines for Drinking Water Quality*. https://www.who.int/publications/i/item/9789241549950

Zhang, S., Wu, S., Chen, L., Guo, P., Jiang, X., Pan, H., & Li, Y. (2024). Multi-task water quality colorimetric detection method based on deep learning. *Sensors*, *24*, 22. https://doi.org/10.3390/s24227345

APPENDICES

Includes standards preview, certification from experts/clients, code snippets, patent reports, and other long and detailed documents.

APPENDIX A: LETTER TO CLIENT

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES

8 November 2025

TO : Ms. Danielle Dolom

Dear **Ms. Dolom**

This is to introduce the following Computer Engineering students of the Technological Institute of the Philippines, Quezon City:

Brodeth, Van Jersey Paolo P.
Pascual, Ken Leonard

Kindly extend your assistance to them in connection with their Project Study. It would be much appreciated if they would be allowed to gather data and conduct an interview in your good office.

We look forward to your favorable consideration of this request. Kindly accept our sincerest gratitude for the help that you shall extend to our students.

Thank you.

Sincerely yours,

ENGR. JI HAN C. GANG
Instructor, CPE025A - Software Design 1

Noted by:
ENGR. ROMAN M. RICHARD
Program Chair, CPE Department

APPENDIX B: Questionnaire for Acquiring Client Requirements

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES

27 October 2025

Good day!

This document serves as an interview questionnaire for the Project Study of the following Computer Engineering students of the Technological Institute of the Philippines, Quezon City:

Brodeth, Van Jersey Paolo P. | qvjppbrodeth@tip.edu.ph
Pascual, Ken Leonard | qkl-pascual@tip.edu.ph

As part of the aforementioned students' Project Study, this interview questionnaire is intended as a preliminary investigation report to identify existing challenges, inefficiencies, or unmet needs in your operations that may be addressed through the application of emerging technologies such as machine learning and computer vision. The gathered information will help the group determine and design a practical and relevant system that could improve certain processes through technological innovation and societal impact.

The next pages of this document contain the questionnaire itself. We ask for your kind consideration and cooperation in this interview.

Interview Questionnaire

Background and History

1. Could you tell us a brief history of your hiking experience, including typical locations and trip durations?

2. How often do you go on hiking or camping trips?

Current Water Usage Concerns

3. How do you currently source and treat water during your hiking and camping trips?

4. Have you ever faced any water quality issues while outdoors? If yes, please describe them.

5. What concerns do you have regarding water safety and quality when in the wilderness?

Awareness and Use of Water Monitoring Tools

6. Are you currently using any tools, devices, or methods to test or monitor water quality? If so, what are they?

7. What do you like or dislike about those current methods or tools?

Needs and Challenges with Water Quality Monitoring

8. What are the biggest challenges you face in ensuring safe drinking water while hiking or camping?

9. What features would be most important for you in a smart portable water quality monitoring tool?

Viability and Interest in a New Tool

10. Would you be interested in using a dedicated water quality monitoring tool designed specifically for hikers and campers?

11. What would make you trust and rely on such a device?

Additional Feedback

12. Are there other water-related safety or convenience issues you think should be addressed for outdoor enthusiasts?

13. Is there anything else you'd like to share about your experience or ideas for water quality monitoring when hiking or camping?

APPENDIX C: ISO 8601 (Date and Time Format)

What can ISO 8601 do for me?

When dates are represented with numbers they can be interpreted in different ways. For example, 01/05/22 could mean January 5, 2022, or May 1, 2022. On an individual level this uncertainty can be very frustrating, in a business context it can be very expensive. Organizing meetings and deliveries, writing contracts and buying airplane tickets can be very difficult when the date is unclear.

ISO 8601 tackles this uncertainty by setting out an internationally agreed way to represent dates:

YYYY-MM-DD

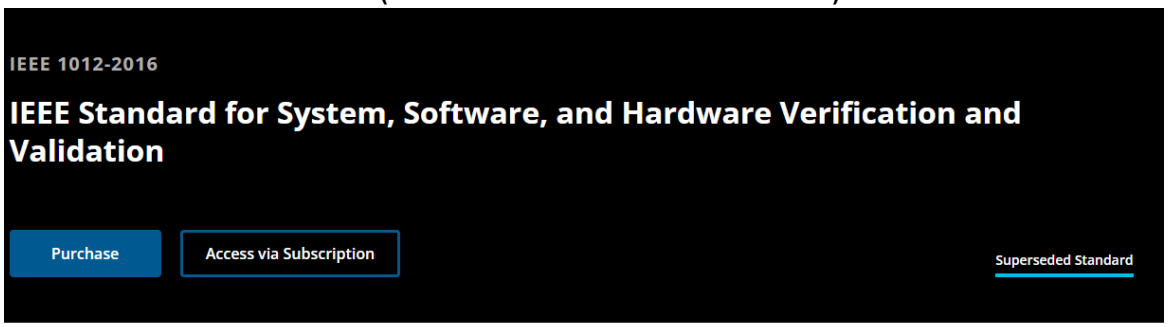
Therefore, the order of the elements used to express date and time in ISO 8601 is as follows: year, month, day, hour, minutes, seconds, and milliseconds.

For example, September 27, 2022 at 6 p.m. is represented as 2022-09-27 18:00:00.000.

ISO 8601 can be used by anyone who wants to use a standardized way of presenting:

- Date
- Time of day
- Coordinated Universal Time (UTC)
- Local time with offset to UTC
- Date and time
- Time intervals
- Recurring time intervals

APPENDIX D: ANSI/IEEE 1012 (Software Verification and Validation)



Verification and validation (V&V) processes are used to determine whether the development products of a given activity conform to the requirements of that activity and whether the product satisfies its intended use and user needs. V&V life cycle process requirements are specified for different integrity levels. The scope of V&V processes encompasses systems, software, and hardware, and it includes their interfaces. This standard applies to systems, software, and hardware being developed, maintained, or reused (legacy, commercial off-the-shelf [COTS], non-developmental items). The term software also includes firmware and microcode, and each of the terms system, software, and hardware includes documentation. V&V processes include the analysis, evaluation, review, inspection, assessment, and testing of products.

APPENDIX E: WHO Guidelines for Drinking Water Quality

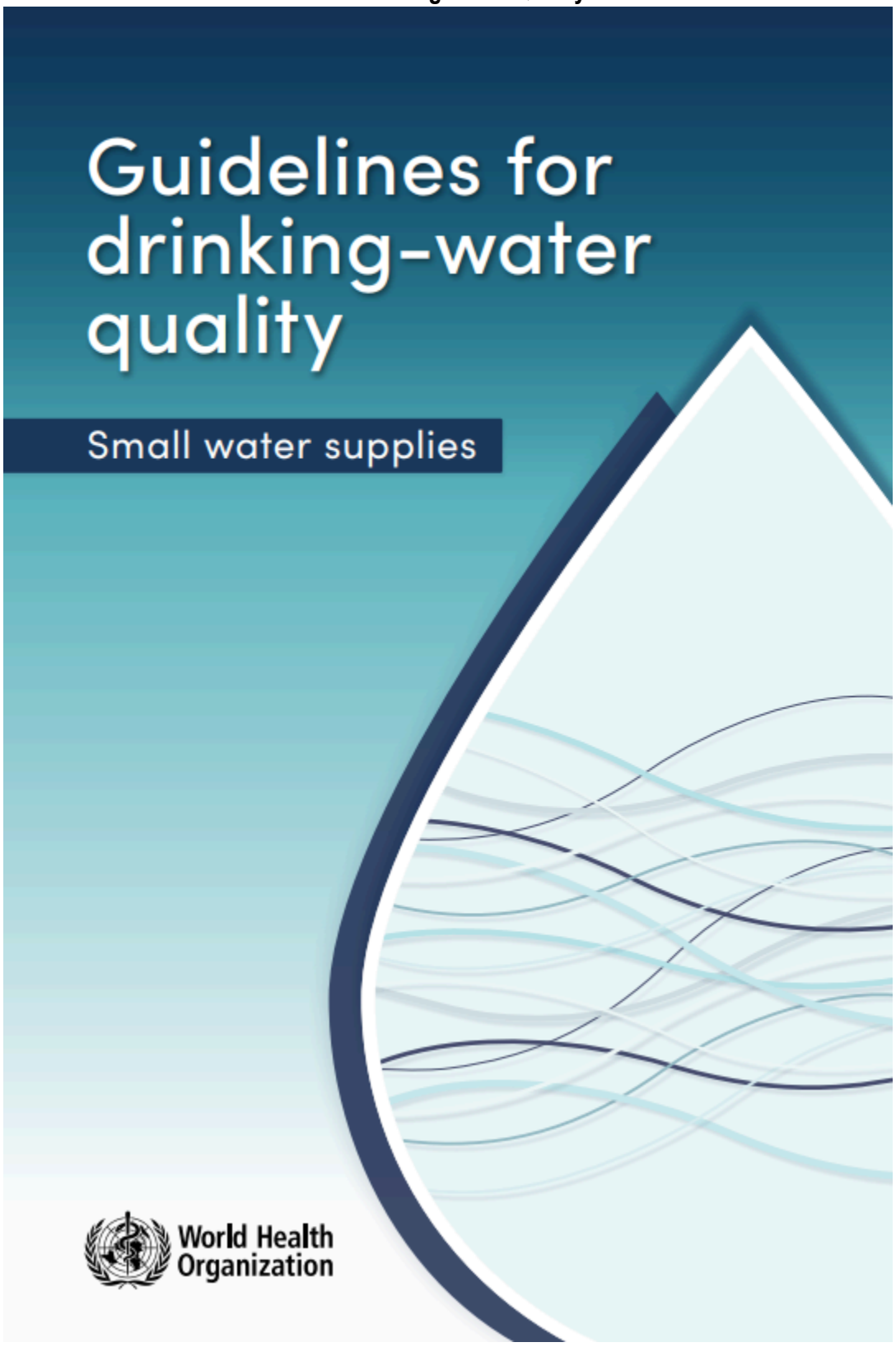


Table 3.1 • Priority parameters related to microbial safety

Critical parameters related to microbial safety (all supplies)		
Parameter[a]	**Significance for microbial water quality**	**Occurrence in drinking-water**
E. coli (or alternatively thermotolerant coliforms)	*E. coli* is excreted in large numbers in the faeces of humans and other warm-blooded animals. While most strains are non-pathogenic, certain strains can cause acute diarrhoea. *E. coli* is an important indicator of the presence of recent faecal contamination and associated pathogens (see Box 3.2).	Higher *E. coli* concentrations are expected in surface water and shallower groundwater sources (including those under the influence of surface water). Lower concentrations are typically found in deeper groundwater sources that are protected.
Free chlorine residual (if chlorinated)	Free chlorine residual provides an indication of microbial safety in terms of the efficacy of disinfection. Maintaining a residual throughout storage and distribution provides some protection against low-level microbial recontamination and growth, including as a result of user practices.	Added as a water treatment chemical for disinfection purposes.
Turbidity	Turbidity is caused by suspended or dissolved organic and inorganic materials. Where water treatment is applied, turbidity provides an indication of the effectiveness of particle removal processes and/or of conditions for effective disinfection (as high turbidity can interfere with disinfection processes, including chlorination). It also provides an indication of changes in source water quality and distribution network integrity, which can indicate vulnerability to microbial contamination.[b] (Acceptability issues are discussed at the end of this section.)	Higher turbidity levels are expected in surface water and shallower groundwater sources (including those under the influence of surface water), and turbidity levels tend to fluctuate with rainfall and snowmelt (e.g. seasonally). Lower levels of turbidity are typically found in deeper groundwater sources that are protected.
pH (if chlorinated)	pH is an important parameter in determining chlorination efficacy.	pH is naturally influenced by source water characteristics (including geology), and may be optimized by the addition of treatment chemicals. Contact with certain materials (e.g. cement-based storage tanks and pipes) may alter the pH.

[a] See Tables 3.4 and 3.5 for guideline or target values.

[b] See *Water quality and health – review of turbidity (24)*.

Sources: adapted from WHO documents *Developing drinking-water quality regulations and standards (15)* and the *GDWQ (6)*.

Table 3.2 continued • Priority chemical parameters

Priority chemicals (where applicable)[a]		
Parameter[b]	**Health significance[c]**	**Occurrence in drinking-water**
Nitrate	High nitrate and nitrite concentrations in drinking-water can give rise to blue-baby syndrome in bottle-fed infants, particularly where there is endemic diarrhoea in infants (e.g. from poor microbial quality of drinking-water).	Nitrate may be naturally occurring, although its presence in drinking-water is more often associated with agricultural activities (e.g. excessive use of fertilizers), or it may come from poorly sited and maintained latrines and septic tanks. Nitrate occurs widely throughout the world in both groundwater and surface water, and it presents a particular problem in shallow wells. Elevated concentrations of nitrite may occur in groundwater supplies under reducing conditions, or in piped supplies where there are high concentrations of free ammonia entering the distribution system (which can lead to nitrification). Nitrite is usually not present in significant concentrations except for these situations.

[a] A risk assessment should be conducted to determine if these parameters are likely to occur at concentrations of concern and, therefore, should be prioritized for compliance monitoring. This risk assessment should be revisited following any significant changes in circumstance that could affect the parameter's presence or concentration.

[b] See Tables 3.6 to 3.9 for guideline values.

[c] See the GDWQ *(6)* for more comprehensive information on health effects.

Sources: adapted from WHO documents *Developing drinking-water quality regulations and standards (15)* and the GDWQ *(6)*.

Table 3.3 • Considerations for determining compliance monitoring frequencies and locations

How often to monitor?	
What is the size of the population served?	Water supplies serving more consumers may warrant more frequent monitoring due to the potential for exposing larger populations to unsafe water.
Is the parameter likely to be present at concentrations of concern?	The frequency of monitoring should reflect the risk that a parameter will be present at a concentration of concern. Where a local risk assessment indicates that a parameter (prioritized for monitoring at a national or subnational level) is not expected to be present at a concentration of concern, only very occasional monitoring may be needed, and possibly no monitoring at all once successive sampling events have validated the low likelihood of occurrence.
How stable is the parameter?	Water quality parameters that can change rapidly should be tested at greater frequency than parameters that are more stable. For example, microbial indicators and chemical disinfectants (e.g. chlorine) should be tested more frequently than inorganic chemicals found in groundwater, such as arsenic or fluoride.
Are there seasonal variations?	The timing and frequency of monitoring should account for seasonal variations (including as a result of changes in climate), particularly for surface water and groundwater under the influence of surface water. For example, turbidity and microbial loading may be greater during the rainy season or periods of snowmelt, and nitrate concentrations may be higher during the season(s) of fertilizer application. Monitoring should be carried out when parameters are most likely to be present at concentrations of concern.

APPENDIX F: Philippine National Standards for Drinking Water (PNSDW)

Republic of the Philippines
Department of Health
OFFICE OF THE SECRETARY

JUN 2 3 2017

ADMINISTRATIVE ORDER
No. 2017 - 0010

SUBJECT: Philippine National Standards for Drinking Water of 2017

I. RATIONALE

The history of the Philippine National Standards for Drinking Water (PNSDW) started in the year 1963. It was based on the 1958 World Health Organization International Standard for Drinking Water and the 1962 United States Public Health Service Standards. The 1963 PNSDW edition was subsequently revised in 1978, 1993 and 2007.

Since the last revision of PNSDW in 2007, a number of issues and concerns from various stakeholders have emerged. Among these are: (i) experiences of water service providers in complying with the standards; (ii) publication of the fourth edition of the Guidelines for Drinking-Water Quality by the World Health Organization in 2011, which includes new parameters and an improved framework for drinking-water safety that should be considered in water quality monitoring, testing, and analysis; (iii) issuance of DOH Administrative Order Number 2014-0027, which requires all drinking-water service providers to develop and implement water safety plans; (iv) new scope and definitions of Sustainable Development Goal (SDG) water supply indicators; and (v) the need for water quality standards during emergency situations.

This led to the updating of the PNSDW of 2007 through the Inter-agency Technical Working Group (TWG), headed by the Department of Health (DOH) with support from the World Health Organization (WHO).

II. OBJECTIVES

This Administrative Order shall prescribe the standards and procedures on drinking-water quality to protect public/consumer's health.

III. SCOPE AND COVERAGE

The PNSDW of 2017 shall apply to all drinking-water service providers including government and private developers and operators, bulk water suppliers, water refilling station operators, and water vending machine operators; ice manufacturers; all food establishments, residential, commercial, industrial and institutional buildings that use/supply/serve drinking water; water testing laboratories; health and sanitation authorities; the general public and all others who are involved in determining the safety of public's drinking-water.

3. Standards for Other Modes of Distribution of Drinking-water

A. Drinking-water from refilling stations, vending machines, mobile tanks and bulk water supply shall be subject for initial and periodic examinations for microbiological, physical, chemical and radiological quality.

B. All standard values of mandatory parameters shall be applicable to product water from refilling stations and vending machines, except for the standard values of *pH* and total dissolved solids (TDS). The *pH* value shall be 5-7 while the TDS levels of product water shall not exceed 10 mg/L to validate the efficiency of reverse osmosis or distillation process.

C. Water from mobile tanks shall have chlorine residual (as free chlorine) of at least 0.5 mg/L but not to exceed to 1.50 mg/L at the point of delivery.

D. Bulk water supply shall maintain chlorine residual (as free chlorine) level between 0.3 mg/L to 1.5 mg/L or chlorine dioxide residual between 0.2 mg/L to 0.4 mg/L prior to distribution.

E. All water-refilling stations, vending machines, mobile tanks and bulk water supply shall comply with the standard minimum number of samples and frequency of sampling requirements. Refer to *Annex C*.

APPENDIX G: DAO 2016-08 Water Quality Guidelines and General Effluent Standards

Republic of the Philippines
Department of Environment and Natural Resources
Visayas Avenue, Diliman, Quezon City
Tel Nos. 929-6626 to 29; 929-6633 to 35
926-7041 to 43; 929-6252; 929-1669
Website: http://www.denr.gov.ph / E-mail: web@denrgov.ph

DENR Administrative Order
No. 2016 -08

MAY 24 2016

SUBJECT: **Water Quality Guidelines and General Effluent Standards of 2016**

Pursuant to Section 19e and 19f of Republic Act (RA) 9275, otherwise known as the Philippine Clean Water Act of 2004, and Executive Order 192 (Providing the Reorganization of the Department of Environment, Energy and Natural Resources; Renaming it as the Department of Environment and Natural Resources) dated 10 June 1987, the Department of Environment and Natural Resources (DENR) hereby adopts and promulgates these Water Quality Guidelines (WQG) and General Effluent Standards (GES).

SECTION 1.0 Basic Policy. It is the policy of the State to pursue a policy of economic growth in a manner consistent with the protection, preservation and revival of the quality of our fresh, brackish and marine waters.

SECTION 2.0 Objectives. This Administrative Order is issued to provide guidelines for the classification of water bodies in the country; determination of time trends and the evaluation of stages of deterioration/enhancement in water quality; evaluation of the need for taking actions in preventing, controlling, or abating water pollution; and designation of water quality management areas (WQMA). In addition, this Order is issued to set the General Effluent Standards (GES).

SECTION 3.0 Scope and Coverage. The WQG applies to all water bodies in the country: freshwaters, marine waters, and groundwater; and shall be used for classifying water bodies, determining time trends, evaluating stages of deterioration or enhancement in water quality, and as basis for taking positive actions in preventing, controlling, or abating water pollution. Moreover, this WQG shall be used in the process of designating WQMA.

The GES applies to all point sources of pollution, regardless of volume, that discharge to receiving body of water or land. The GES shall be used regardless of the industry category.

SECTION 4.0 Definition of Terms. For purposes of this Order, the following terms shall have the following meanings:

a) **"Annual Average"** means the sum of all values in one year divided by the number of values.

Table 1. Water Body Classification and Usage of Freshwater

Classification	Intended Beneficial Use
Class AA	Public Water Supply Class I – Intended primarily for waters having watersheds, which are uninhabited and/or otherwise declared as protected areas, and which require only approved disinfection to meet the latest PNSDW
Class A	Public Water Supply Class II – Intended as sources of water supply requiring conventional treatment (coagulation, sedimentation, filtration and disinfection) to meet the latest PNSDW
Class B	Recreational Water Class I – Intended for primary contact recreation (bathing, swimming, etc.)
Class C	1. Fishery Water for the propagation and growth of fish and other aquatic resources 2. Recreational Water Class II – For boating, fishing, or similar activities 3. For agriculture, irrigation, and livestock watering
Class D	Navigable waters

Table 3. Water Quality Guidelines for Primary Parameters

Parameter	Unit	Water Body Classification								
		AA	A	B	C	D	SA	SB	SC	SD
BOD	mg/L	1	3	5	7	15	n/a	n/a	n/a	n/a
Chloride	mg/L	250	250	250	350	400	n/a	n/a	n/a	n/a
Color	TCU	5	50	50	75	150	5	50	75	150
Dissolved Oxygen[a] (Minimum)	mg/L	5	5	5	5	2	6	6	5	2
Fecal Coliform	MPN/100mL	<1.1	<1.1	100	200	400	<1.1	100	200	400
Nitrate as NO_3-N	mg/L	7	7	7	7	15	10	10	10	15
pH (Range)		6.5-8.5	6.5-8.5	6.5-8.5	6.5-9.0	6.0-9.0	7.0-8.5	7.0-8.5	6.5-8.5	6.0-9.0
Phosphate	mg/L	<0.003	0.5	0.5	0.5	5	0.1	0.5	0.5	5
Temperature[b]	°C	26-30	26-30	26-30	25-31	25-32	26-30	26-30	25-31	25-32
Total Suspended Solids	mg/L	25	50	65	80	110	25	50	80	110

Table 4. Water Quality Guidelines for Secondary Parameters-Inorganics

Parameter	Unit	Water Body Classification								
		AA	A	B	C	D	SA	SB	SC	SD
Ammonia as NH_3-N	mg/L	0.05	0.05	0.05	0.05	0.75	0.04	0.05	0.05	0.75
Boron	mg/L	0.5	0.5	0.5	0.75	3	0.5	0.5	5	20
Fluoride	mg/L	1	1	1	1	2	1.5	1.5	1.5	3
Selenium	mg/L	0.01	0.01	0.01	0.02	0.04	0.01	0.01	0.1	0.2
Sulfate	mg/L	250	250	250	275	500	250	250	275	500

APPENDIX H: Proof of Concept

```
(CPE312_Pascual) PS C:\Users\Leon\Documents\Github\CPE025A\python\xgboost> python PredictionDraft_XGB.py
Please input the following water colorimetry concentration values:
Ammonia (mg/l): 1
pH (ph units): 7.8
Nitrate (mg/l): 2
Predicted water class: B1
Prediction saved to predictions_log.csv
```

Classification via CLI using XGBoost

```
(CPE312_Pascual) PS C:\Users\Leon\Documents\Github\CPE025A\python\lightgbm> python PredictionDraft_LightGBM.py
Please input the following water colorimetry concentration values:
Ammonia (mg/l): 0.05
pH (ph units): 8
Nitrate (mg/l): 2
C:\Users\Leon\miniconda3\envs\CPE312_Pascual\Lib\site-packages\sklearn\utils\validation.py:2749: UserWarning: X
  warnings.warn(
Predicted water class: A3
Prediction saved to predictions_log.csv
Current file size: 108 bytes
```

Classification via CLI using LightGBM

```
(CPE312_Pascual) PS C:\Users\Leon\Documents\Github\CPE025A\python\catboost> python PredictionDraft_CatBoost.py

Please input the following water colorimetry concentration values:

Ammonia (mg/l): 50
pH (ph units): 8
Nitrate (mg/l): 1
C:\Users\Leon\Documents\Github\CPE025A\python\catboost\PredictionDraft_CatBoost.py:45: DeprecationWarning: Conve
u extract a single element from your array before performing this operation. (Deprecated NumPy 1.25.)
  pred_cluster = int(pred_cluster_raw) if not isinstance(pred_cluster_raw, str) else pred_cluster_raw

Predicted water class: B1

Prediction saved to predictions_log.csv
Current file size: 108 bytes
```

Classification via CLI using CatBoost

APPENDIX I: Model Evaluation for Constraint Metrics

APPENDIX I1: Evaluation of Training Time, Inference Time, and Misclassification Rate

LIGHTGBM

Size	Train Time (s)	Pred Time (s)	Accuracy	Misclass Rate
13395	2.5929	7.6132	0.9875	0.0125
133956	6.2895	7.7470	0.9940	0.0060
669784	20.8494	6.8847	0.9965	0.0035
937698	30.3299	6.9660	0.9970	0.0030
1339569	58.9863	6.7624	0.9971	0.0029

XGBOOST

Size	Train Time (s)	Pred Time (s)	Accuracy	Misclass Rate
13395	0.8855	1.1171	0.9860	0.0140
133956	3.2297	1.0530	0.9932	0.0068
669784	24.1829	1.2224	0.9947	0.0053
937698	35.0554	1.0956	0.9951	0.0049
1339569	54.2305	1.3723	0.9954	0.0046

CATBOOST

Size	Train Time (s)	Pred Time (s)	Accuracy	Misclass Rate
13395	9.2534	0.2240	0.9879	0.0121
133956	27.1735	0.2039	0.9939	0.0061
669784	115.1454	0.1992	0.9952	0.0048
937698	160.1249	0.2178	0.9954	0.0046
1339569	218.9980	0.1899	0.9957	0.0043

		Accuracy based on dataset size					
Safety - Misclassification Rate		1%	10%	50%	70%	100%	Average
LightGBM	0.00598	0.9861	0.9939	0.9965	0.9967	0.9969	0.99402
CatBoost	0.00702	0.9867	0.9934	0.9946	0.995	0.9952	0.99298
XGBoost	0.00772	0.9852	0.9925	0.9945	0.9947	0.9945	0.99228
				0.99			
Performance - Inference Time		Inference time based on dataset size					
LightGBM	8.01422	7.458	10.0918	7.3908	7.6601	7.4704	8.01422
CatBoost	0.22848	0.202	0.2577	0.2615	0.2212	0.2	0.22848
XGBoost	1.1537	1.104	1.1416	1.142	1.2107	1.1702	1.1537
Manufacturability - Training Time		Training time based on dataset size					
LightGBM	22.84536	2.104	7.0447	25.8416	30.8859	48.3506	22.84536
CatBoost	112.0747	8.3548	30.4281	119.3633	163.6814	238.5459	112.0747
XGBoost	21.80616	0.6831	4.2222	25.1735	32.5275	46.4245	21.80616

Computation Table for Raw Scores in Safety, Performance, and Manufacturability Constraints

APPENDIX I2: Evaluation for Storage Consumption

```python
import joblib

for model_name, model_obj in [('LightGBM', lgbm), ('XGBoost', xgb), ('CatBoost', cb)]:
    filename = f"{model_name}_model.pkl"
    joblib.dump(model_obj, filename)
    file_size_mb = os.path.getsize(filename) / (1024 * 1024)
    print(f"{model_name}: {file_size_mb:.4f} MB")
    os.remove(filename)  # Clean up

print("\n" + "="*100)
print("ANALYSIS COMPLETE")
print("="*100)
```

```
LightGBM: 21.4742 MB
XGBoost: 6.6122 MB
CatBoost: 6.6758 MB
```

APPENDIX I3: Evaluation for Maintainability Index Score

```python
from radon.metrics import mi_visit
from radon.complexity import cc_visit

# Example: Read your model's code file
with open('catboost/PredictionDraft_CatBoost.py', 'r') as file:
    code = file.read()

# Calculate the Maintainability Index
mi_score = mi_visit(code, True)  # True gives the score on a 0-100 scale

print(f"CatBoost Maintainability Index Score: {mi_score}")
```

```
Maintainability Index Score: 78.1072231421837
```

```python
from radon.metrics import mi_visit
from radon.complexity import cc_visit

# Example: Read your model's code file
with open('lightgbm/PredictionDraft_LightGBM.py', 'r') as file:
    code = file.read()

# Calculate the Maintainability Index
mi_score = mi_visit(code, True)  # True gives the score on a 0-100 scale

print(f"LightGBM Maintainability Index Score: {mi_score}")
```

```
Maintainability Index Score: 78.32318593963109
```

```python
from radon.metrics import mi_visit
from radon.complexity import cc_visit

# Example: Read your model's code file
with open('xgboost/PredictionDraft_XGB.py', 'r') as file:
    code = file.read()

# Calculate the Maintainability Index
mi_score = mi_visit(code, True)  # True gives the score on a 0-100 scale

print(f"XGBoost Maintainability Index Score: {mi_score}")
```

```
Maintainability Index Score: 77.10482264626778
```

APPENDIX I4: Summary of Tradeoff Analysis (Spreadsheet)

Constraint	Metric	Constraint - Metric	Raw Scores LightGBM	CatBoost	XGBoost	Minimization or Maximization?
Safety	Misclassification Rate	Safety -	0.00598	0.00702	0.00772	Minimization
Performance	Inference Time	Performance -	8.01422	0.22848	1.1537	Minimization
Manufacturability	Training Time	Manufacturability -	22.84536	112.0747	21.80616	Minimization
Compatibility	Maintainability Index	Compatibility -	78.3231859	78.1072231	77.10482265	Maximization
Efficiency	Storage Consumption	Efficiency - Storage	21.4742	6.6758	6.6122	Minimization

Criterion Importance		Ability to satisfy criterion (scale from 1 to 10))		
Level of Importance	Percentage	LightGBM	CatBoost	XGBoost
10	25.00%	10	4.620689655	1
9	22.50%	1	10	8.930483165
8	20.00%	9.896389152	1	10
7	17.50%	10	8.404691616	1
6	15.00%	1	9.961485668	10
40	100.00%	6.60427783	6.570216297	5.934358712

APPENDIX I5: Computation for Sensitivity Analysis

```python
import pandas as pd
import numpy as np

# --- Data setup ---
# columns
index=["Safety", "Performance", "Manufacturability","Compatibility","Efficiency"]

# rows for each model (In respective order based on the index)
lightgbm = [0.00598, 8.01422, 22.84536, 78.32318594, 21.4742,  ]
catboost  = [0.00702, 0.22848, 112.0747, 78.10722314, 6.6758]
xgboost   = [0.00772, 1.1537, 21.80616, 77.10482265, 6.6122 ]

df = pd.DataFrame(
    data={
        "LightGBM": lightgbm,
        "CatBoost": catboost,
        "XGBoost": xgboost
    },
    index=index
)
```

```python
# --- Apply normalization per row (Minimization) ---
def normalize_min_row(row):
    max_val = row.max()
    min_val = row.min()
    return 9 * (max_val - row) / (max_val - min_val) + 1
```

```python
# --- Apply normalization per row (Maximization) ---
def normalize_max_row(row):
    max_val = row.max()
    min_val = row.min()
    return 9 * (row - min_val) / (max_val - min_val) + 1
```

```
# get the rows that will be minimized and maximized
min_rows = df.loc[['Safety', 'Manufacturability','Performance','Efficiency']]
max_rows = df.loc[['Compatibility']]

df_max = max_rows.apply(normalize_max_row, axis=1)
df_min = min_rows.apply(normalize_min_row, axis=1)

df_norm = pd.concat([df_min, df_max])
order = ["Safety", "Performance", "Efficiency", "Manufacturability", "Compatibility"]
df_norm = df_norm.reindex(order)

print("\n=== Normalized Data (1–10 scale, by row) ===")
df_norm
```

```
# --- Generate all permutations of LOI (10, 9, 8, 7, 6) ---
import itertools
# here is the LOIs, change it
LOI_base = [10, 9, 8, 7, 6]
all_combinations = list(itertools.permutations(LOI_base))

print(f"\nTotal LOI combinations: {len(all_combinations)}")
```

```python
# appending to a table

results = []
for i, combo in enumerate(all_combinations, start=1):
    # Convert LOI to percentage
    total = sum(combo)
    loi_percentages = [x / total for x in combo]
    loi_series = pd.Series(loi_percentages, index=df_norm.index)

    # Weighted calculation
    df_weighted = df_norm.mul(loi_series, axis=0)
    final_scores = df_weighted.sum()

    # Save to CSV
    filename = f"weighted_results_combo_{i}.csv"
    df_weighted.to_csv(filename, index=True)

    # Store results summary
    results.append({
        "Combo": i,
        "LightGBM": final_scores["LightGBM"],
        "CatBoost": final_scores["CatBoost"],
        "XGBoost": final_scores["XGBoost"],
        "LOI": combo,
    })
```

```python
# --- Save all final scores summary ---
df_results = pd.DataFrame(results)
df_results.to_csv("all_LOI_results_summary.csv", index=False)

print("\nAll LOI combinations processed and saved!")
print(df_results.head())
```

```
All LOI combinations processed and saved!
   Combo  LightGBM  CatBoost   XGBoost               LOI
0      1  6.156868  6.833173  6.159359  (10, 9, 8, 7, 6)
1      2  6.159458  7.018291  5.934359  (10, 9, 8, 6, 7)
2      3  6.379278  6.609136  6.159359  (10, 9, 7, 8, 6)
3      4  6.384458  6.979371  5.709359  (10, 9, 7, 6, 8)
4      5  6.604278  6.570216  5.934359  (10, 9, 6, 8, 7)
```

APPENDIX J: Reclassification of training data through Clustering (K-means)

Reclassification

The classification that will be used is based on the DENR Guidelines for Water Quality Management in the Philippines (DAO 2016-08). For freshwater analysis, we'll be using Classes A, B, C, and D. Clustering was used to help determine if 4 classes is enough to group the data points together.

Clustering via K-means

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data_scaled = scaler.fit_transform(df_cluster)

# statistics of scaled data
pd.DataFrame(data_scaled).describe().map(lambda x: f"{x:.2f}")
```

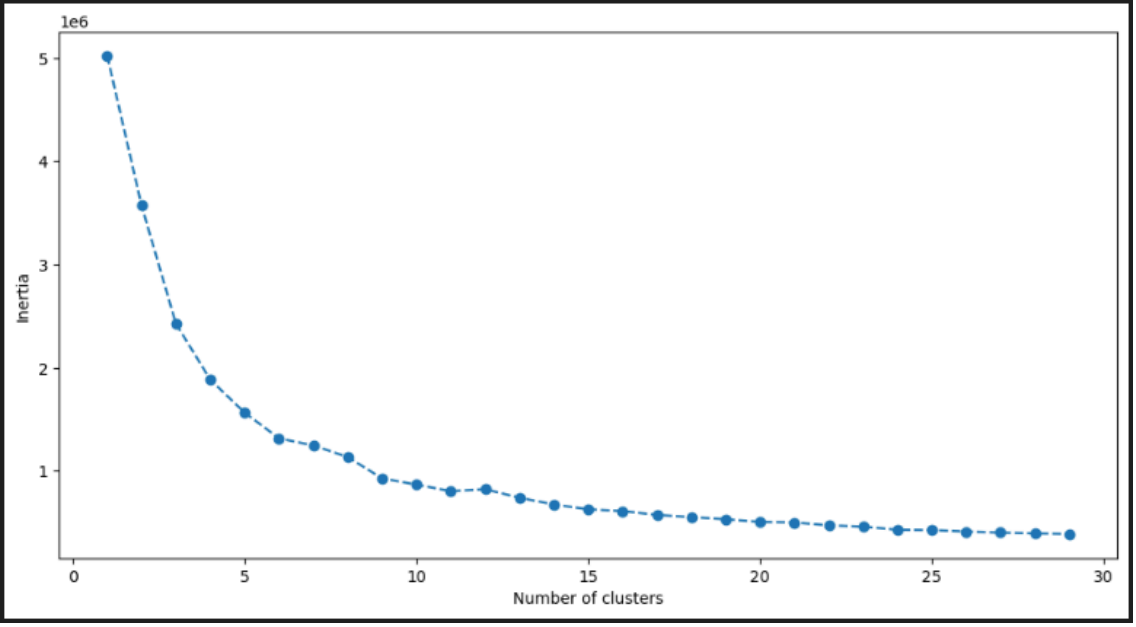
Python

	0	1	2
count	1674462.00	1674462.00	1674462.00
mean	0.00	-0.00	0.00
std	1.00	1.00	1.00
min	-0.73	-2.39	-1.50
25%	-0.57	-0.56	-0.96
50%	-0.49	-0.02	0.28
75%	0.00	0.70	0.36
max	4.44	2.26	2.63

```python
"""Use elbow method to determine the optimal number of clusters"""
# fitting multiple k-means algorithms and storing the values in an empty list
SSE = []
for cluster in range(1,30):
    kmeans = KMeans(n_clusters = cluster, init='k-means++')
    kmeans.fit(data_scaled)
    SSE.append(kmeans.inertia_)

# converting the results into a dataframe and plotting them
frame = pd.DataFrame({'Cluster':range(1,30), 'SSE':SSE})
plt.figure(figsize=(12,6))
plt.plot(frame['Cluster'], frame['SSE'], marker='o', linestyle='--')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
```

```
Text(0, 0.5, 'Inertia')
```



```
# Create cluster of n-clusters
kmeans = KMeans(n_clusters=4, init='k-means++')

# fit the k means algorithm on scaled data
kmeans.fit(data_scaled)
# inertia on the fitted data
kmeans.inertia_
```

```
1893390.3569604945
```

```
# Create cluster of n-clusters
kmeans = KMeans(n_clusters=12, init='k-means++')

# fit the k means algorithm on scaled data
kmeans.fit(data_scaled)
# inertia on the fitted data
kmeans.inertia_
```

```
762940.5503341694
```

12 clusters is considered optimal, but for a finer separation, 16 clusters will be used

```
# Create cluster of n-clusters
kmeans = KMeans(n_clusters=16, init='k-means++')

# fit the k means algorithm on scaled data
kmeans.fit(data_scaled)
# inertia on the fitted data
kmeans.inertia_
```

```
652238.8493761349
```

```
pred = kmeans.predict(data_scaled)
frame = pd.DataFrame(data_scaled)
frame['cluster'] = pred
frame['cluster'].value_counts()
```

```
cluster
1     291022
7     206493
0     143935
14    132150
4     129649
2      95875
13     90222
10     86625
6      82592
15     76581
8      75674
11     67376
9      60001
3      58718
12     43518
5      34031
Name: count, dtype: int64
```

Get the data used right before clustering

```
df_cluster = df_cluster.reset_index(drop=True)
df_cluster
```

	Ammonia (mg/l)	pH (ph units)	Nitrate (mg/l)
0	0.05152	8.3700	9.73940
1	0.07728	8.0167	8.72119
2	0.09016	7.7900	9.51805
3	0.10304	8.1583	8.63265
4	0.10304	7.7900	8.76546
...	...	...	...
1674457	0.02400	7.9000	0.37000
1674458	0.03800	7.9000	0.54000
1674459	0.03500	7.6000	0.79000
1674460	0.04600	8.0000	1.30000
1674461	0.02000	7.9000	1.30000

1674462 rows × 3 columns

Get cluster frame into the dataframe

```
#make a copy of the original dataframe before any filtering
df_original = df_cluster.copy()
df_original['cluster'] = frame['cluster'].values
```

```
#feature_names is in the code block for removing the columns earlier
cluster_stats_unscaled = df_original.groupby('cluster')[feature_names].describe()
cluster_stats_unscaled
```

Label data points based on cluster metrics

```python
cluster_stats_unscaled = df_original.groupby('cluster').agg({
    'pH (ph units)': ['std'],
    'Ammonia (mg/l)': ['mean'],
    'Nitrate (mg/l)': ['mean']
})

# Flatten the MultiIndex columns
cluster_stats_unscaled.columns = [
    'pH_std', 'Ammonia_mean', 'Nitrate_mean'
]

# Lower BOD, NH3, NO3 preferred; higher DO preferred; lower pH variance preferred.
cluster_stats_unscaled['score'] = (
# negative for lower is better
# positive for higher is better
    - cluster_stats_unscaled['Nitrate_mean']
    - cluster_stats_unscaled['Ammonia_mean']
    - cluster_stats_unscaled['pH_std']    # negative for more stable pH
)

cluster_means = cluster_stats_unscaled.sort_values('score', ascending=False)
labels = []
for idx, row in enumerate(cluster_means.itertuples()):
    if idx < 4:
        labels.append(f'A{idx+1}')
    elif idx < 8:
        labels.append(f'B{idx-3}')
    elif idx < 12:
        labels.append(f'C{idx-7}')
    else:
        labels.append(f'D{idx-11}')
cluster_means['label'] = labels
cluster_means

# Assuming your clusters are 0-15 and labels are in same order
cluster_label_map = dict(zip(cluster_means.index, cluster_means['label']))
df_original['class_label'] = df_original['cluster'].map(cluster_label_map)


df_original['class_label'].value_counts().sort_index()
```

```
cluster_means = cluster_stats_unscaled.sort_values('score', ascending=False)
cluster_means
```

cluster	pH_std	Ammonia_mean	Nitrate_mean	score
1	0.070663	0.041362	1.047175	-1.159200
4	0.115417	0.056083	1.135499	-1.306999
8	0.080833	0.046046	1.314016	-1.440895
14	0.076815	0.051962	1.349588	-1.478365
9	0.122387	0.366066	2.805336	-3.293789
7	0.072822	0.087181	4.278805	-4.438808
13	0.115371	0.089248	4.380934	-4.585553
0	0.074765	0.071987	4.445383	-4.592135
10	0.080686	0.058146	4.542783	-4.681614
11	0.129915	0.536051	4.412280	-5.078247
12	0.145242	0.490943	4.734382	-5.370568
2	0.102108	0.629339	4.731246	-5.462693
6	0.090227	0.066449	6.491260	-6.647935
15	0.106357	0.061430	7.691113	-7.858901
5	0.128749	0.111813	7.881060	-8.121623
3	0.100816	0.090566	8.826008	-9.017390

```
#take a look at the reclassified data
ordered_labels = ['A1', 'A2', 'A3', 'A4',
                  'B1', 'B2', 'B3', 'B4',
                  'C1', 'C2', 'C3', 'C4',
                  'D1', 'D2', 'D3', 'D4']

subset = df_original.groupby("class_label").head(1)
subset.set_index('class_label').reindex(ordered_labels).reset_index()
subset
```

	Ammonia (mg/l)	pH (ph units)	Nitrate (mg/l)	cluster	class_label
0	0.051520	8.370000	9.739400	15	D2
1	0.077280	8.016700	8.721190	3	D4
7	0.094024	7.790000	3.023641	7	B2
8	0.014039	8.327270	5.836820	10	C1
9	0.046368	7.619440	5.825932	6	D1
10	0.097888	8.337500	1.651271	8	A3
13	0.055813	7.790000	2.331555	1	A1
18	0.069552	8.001700	0.447127	14	A4
22	0.303066	7.790000	0.451997	9	B1
28	0.455093	7.880000	6.654135	12	C3
38	0.009789	7.447200	0.690612	4	A2
50	0.108836	7.896460	3.877167	0	B4
59	0.274344	7.290000	5.836820	13	B3
67	0.482485	7.420800	5.304874	11	C2
79	0.601633	7.796667	6.302567	2	C4
95	0.137478	7.669167	7.639811	5	D3

APPENDIX K: Gantt Chart

TEAM 5

Pascual, Ken Brodeth, Van Jersey Paolo

Project star **Fri, 8/1/2025**

Display wee **11**

GANTT CHART

TASK	ASSIGNED TO	PROGRESS	START	END
Client Identification and Topic Formulation				
Preliminary Investigation		100%	8/1/25	8/15/25
Review of Related Literature		90%	8/15/25	11/10/25
Client Consultation 1		90%	10/10/25	10/15/25
Client Consultation 2		90%	11/1/25	11/10/25
Development of Alternative Designs				
Data Gathering / Dataset		100%	10/13/25	10/30/25
Data Cleaning		100%	10/30/25	11/5/25
Design Draft		100%	11/5/25	11/24/25
Design Evaluation and Metrics		100%	11/24/25	11/26/25
Tradeoff and Sensitivity Analysis		100%	11/26/25	12/1/25
Documentation and Presentation				
Chapter 1		100%	8/16/25	11/17/25
Chapter 2		100%	11/10/25	12/6/25
Chapter 3		80%	11/16/25	12/6/25
Prelim Presentation		100%	9/1/25	9/5/25
Final Presentation		100%	11/10/25	12/7/25