

**Ensemble-Based Classification of Groundwater Potability Using Multi-
Model Stacking Technique**
A PROJECT REPORT

**21CSC303J –SOFTWARE ENGINEERING AND PROJECT
MANAGEMENT
(2021 Regulation)
III Year/ VI Semester
Academic Year: 2024 -2025**

Submitted by
**AFZAL SHAIK [RA2211026010042]
DUSHYANT P [RA2211026010058]
SHRINIKESH B S [RA2211026010062]**

Under the Guidance of
DR. T.S. SHINY ANGEL
Associate Professor
Department of Computational Intelligence

in partial fulfillment of the requirements for the degree of

**BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING**



**SCHOOL OF COMPUTING
FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE
OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR- 603 203
MAY-2025**



Department of Computational Intelligence
SRM Institute of Science & Technology
Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : B.Tech / CSE-AIML

Student Name : Shaik Afzal, Dushyant P, Shrinikesh

Registration Number : RA2211026010042,

RA2211026010058, RA2211026010062

Title of Work : Ensemble-Based Classification of Groundwater Potability Using Multi-Model Stacking Technique

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook /University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203**

BONAFIDE CERTIFICATE

Certified that 21CSP302L - Project report titled “**Ensemble-Based Classification of Groundwater Potability Using Multi-Model Stacking Technique**” is the bonafide work of **Shaik [RA2211026010042], Dushyant P [RA2211026010058], Shrinikesh B S[RA2211003011421]**, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. T.S. SHINY ANGEL

Course Faculty

Associate Professor

Department of Computational Intelligence

SRM Institute of Science and Technology

Kattankulathur

SIGNATURE

Dr. Annie Uthra

Head of the Department

Professor

Department of Computational

Intelligence

SRM Institute of Science and

Technology

Kattankulathur

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Leenus Jesu Martin M**, Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We encompass our sincere thanks to, **Dr. M. Pushpalatha**, Professor and Associate Chairperson - CS, School of Computing and **Dr. C.Lakshmi**, Professor and Associate Chairperson -AI, School of Computing, SRM Institute of Science and Technology, for their invaluable support.

We are incredibly grateful to our Head of the Department, **Dr. Annie Uthra R**, HOD Department of Computational Intelligence, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, Panel Head, and Panel Members Department of Computational Intelligence, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Dr. C. Sherin Shibi**, Department of Computational Intelligence, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr.B.Pitchaimanickam**, Department of Computational Intelligence, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under his / her mentorship. He / She provided us with the freedom and support to explore the research topics of our interest. His / Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff members of Computational Intelligence, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement

Authors

ABSTRACT

Groundwater is a vital source of drinking water for billions of people across the globe, particularly in rural and semi-urban areas where centralized water treatment systems are often unavailable or unreliable. However, the increasing contamination of groundwater due to anthropogenic activities such as industrial discharge, agricultural runoff, and improper waste disposal has made water safety a critical concern.

This study proposes a robust and scalable machine learning–based framework for classifying groundwater as potable (safe to drink) or non-potable (unsafe) based on widely recognized physicochemical parameters. The system is designed to analyze input variables including pH, Total Dissolved Solids (TDS), Hardness, Chlorides, Nitrates, Sulfates, and Conductivity, which are considered key indicators of groundwater safety as per international water quality standards. A variety of supervised machine learning algorithms are explored and implemented, including Multilayer Perceptron (MLP), which is a type of deep neural network; Quadratic Discriminant Analysis (QDA), which models class-specific variances; Extra Trees Classifier, known for its robustness and resistance to overfitting; and CatBoost Classifier, a gradient boosting model that handles categorical data and missing values effectively.

Each of these models is trained on a real-world dataset and evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score. Through this analysis, the strengths and weaknesses of each algorithm in the context of water quality prediction are identified. To further enhance classification performance, this research introduces a Stacking Classifier, which combines the predictions of multiple base models using a meta-classifier. This ensemble learning technique leverages the diversity of the individual classifiers and significantly improves the model's generalization capability. The stacking model consistently achieves the highest evaluation scores across all metrics, making it the most reliable choice for groundwater potability classification in this study.

In addition to building and evaluating these models, the research emphasizes the importance of model interpretability, efficiency, and suitability for real-world applications. The study also demonstrates how the proposed approach can be integrated into broader environmental monitoring systems, allowing for timely, data-driven decision-making in water management and public health policy. The novelty of this work lies in its systematic comparison of multiple advanced algorithms for a practical environmental application and the implementation of an ensemble strategy tailored to the specific challenges posed by water quality datasets. By offering a faster, cost-effective, and scalable alternative to traditional testing, this machine learning–based system has the potential to transform groundwater quality monitoring, making it accessible even in resource-constrained regions and thereby contributing to the global goal of ensuring clean and safe water for all.

TABLE OF CONTENTS

| | | |
|--------------------------|---|-----------------|
| ABSTRACT | | v |
| TABLE OF CONTENTS | | vi |
| LIST OF FIGURES | | vii |
| LIST OF TABLES | | viii |
| ABBREVIATIONS | | ix |
| CHAPTER NO. | TITLE | PAGE NO. |
| 1 | INTRODUCTION | 1 |
| | 1.1 Introduction to Project | 2 |
| | 1.2 Problem Statement | 3 |
| | 1.3 Motivation | 4 |
| | 1.4 Sustainable Development Goal of the Project | 5 |
| 2 | LITERATURE SURVEY | 6 |
| | 2.1 Overview of the Research Area | 6 |
| | 2.2 Existing Models and Frameworks | 7 |
| | 2.3 Limitations Identified from Literature Survey (Research Gaps) | 8 |
| | 2.4 Research Objectives | 9 |
| | 2.5 Product Backlog (Key user stories with Desired outcomes) | 10 |
| | 2.5 Plan of Action (Project Road Map) | 11 |
| 3 | SPRINT PLANNING AND EXECTION METHODOLOGY | 13 |
| | 3.1 SPRINT I | 13 |
| | 3.1.1 Objectives with user stories of Sprint I | 13 |
| | 3.1.2 Functional Document | 14 |
| | 3.1.3 Architecture Document | 18 |
| | 3.1.4 Outcome of objectives/ Result Analysis | 21 |
| | 3.1.5 Sprint Retrospective | 23 |
| | 3.2 SPRINT II | 24 |
| | 3.2.1 Objectives with user stories of Sprint II | 24 |
| | 3.2.2 Functional Document | 25 |
| | 3.2.3 Architecture Document | 29 |
| | 3.2.4 Outcome of objectives/ Result Analysis | 32 |
| | 3.2.5 Sprint Retrospective | 37 |

| | |
|--|-----------|
| 6 RESULTS AND DISCUSSIONS | 38 |
| 6.1 Project Outcomes (Performance Evaluation, Comparisons, Testing Results) | 38 |
| 7 CONCLUSION AND FUTURE ENHANCEMENT | 40 |
| REFERENCES | 41 |
| APPENDIX | 43 |
| A CODING | 43 |
| B CONFERENCE PUBLICATION | 50 |
| C JOURNAL PUBLICATION | 51 |
| D PLAGIARISM REPORT | 52 |

LIST OF FIGURES

| CHAPTER NO. | TITLE | PAGE NO. |
|----------------|--|-------------|
| 3.1 | Architecture Diagram..... | 19 |
| 3.2 | Correlation Heatmap | 21 |
| 3.3 | Exploratory Data Analysis..... | 22 |
| 3.4 | ER Diagram | 30 |
| 3.5 | Confusion Matrix of MLP Classification..... | 32 |
| 3.6 | Classification Report of MLP | 32 |
| 3.7 | Confusion Matrix of QDA Classification..... | 33 |
| 3.8 | Classification Report of QDA | 33 |
| 3.9 | Confusion Matrix of CatBoost | 34 |
| 3.10 | Classification Report of CatBoost | 34 |
| 3.11 | Confusion Matrix of Extra Trees Classifier | 35 |
| 3.12 | Classification Report of Extra Trees Classifier..... | 35 |
| 3.13 | Confusion Matrix of Stacking Classifier. | 36 |
| 3.14 | Classification Report of Stacking Classifier | 36 |
| 4.1 | Comparative Analysis of Models | 39 |
| 4.2 | User Interface | 39 |
| B.1 | ICCCNet-2025 Submitted Paper | 50 |
| C.1 | Submission Notification | 51 |

LIST OF TABLES

| CHAPTER NO. | TITLE | PAGE NO. |
|----------------|--|-------------|
| 2.1 | Product Backlog | 10 |
| 3.1 | Detailed User Stories of sprint 1. | 13 |
| 3.2 | Authorization Matrix..... | 17 |
| 3.3 | Result of Sprint 1 | 22 |
| 3.4 | Sprint Retrospective of sprint 1..... | 23 |
| 3.5 | Detailed User Stories for Sprint 2..... | 24 |
| 3.6 | Authorization Matrix..... | 27 |
| 3.7 | Sprint Retrospective of sprint 2..... | 37 |
| 4.1 | Results of All the Models | 38 |

ABBREVIATIONS

| | |
|------------|---------------------------------|
| ML | Machine Learning |
| ANN | Artificial Neural Network |
| CSV | Comma-Separated Values |
| DB | Database |
| QDA | Quadratic Discriminant Analysis |
| TDS | Total Dissolved Solids |
| EDA | Exploratory Data Analysis |
| UI | User Interface |
| MLP | Multi- Layer Perceptron |
| AUC | Area Under the Curve |
| FN | False Negative |
| FP | False Positive |
| TP | True Positive |
| TN | True Negative |

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO PROJECT

Groundwater is one of the most vital natural resources, serving as the primary source of drinking water for a significant portion of the global population. It plays a crucial role in agricultural irrigation, industrial applications, and domestic consumption. The availability of clean groundwater is essential for public health and environmental sustainability. However, with increasing urbanization, industrial expansion, and agricultural intensification, groundwater contamination has become a major environmental concern.

Traditional methods of groundwater quality assessment involve collecting water samples and performing laboratory tests to analyze chemical and biological properties. While these methods provide accurate results, they are often timeconsuming, labor-intensive, and costly. Additionally, they require skilled professionals to conduct the analysis and interpret the data. Given these limitations, there is a growing need for an automated and efficient approach that can provide real-time groundwater quality classification based on physicochemical parameters. This work proposes the development of an intelligent system that assesses groundwater quality using advanced machine learning models.

The system will analyze real-world groundwater datasets containing measurements of critical water quality parameters and predict whether a given water sample is safe to drink. The project incorporates multiple ML models, including Multilayer Perceptron (MLP) for deep learning-based classification, Quadratic Discriminant Analysis (QDA) for statistical pattern recognition, Extra Trees Classifier for ensemble-based decision making, and CatBoost Classifier for handling non-linear relationships. The system processes historical datasets containing chemical attributes such as pH, TDS, Hardness, Chlorides, Nitrates, Sulfates, and Conductivity to evaluate whether a given water sample is suitable for drinking.

To enhance the system's accuracy and reliability, a Stacking Classifier is utilized. This ensemble technique merges the predictions from multiple base models through a meta-classifier that learns the most effective way to integrate their outputs. By adopting a multi-model strategy, the approach boosts predictive performance by reducing the weaknesses of individual models and leveraging their strengths.

The project not only proves that machine learning can be used to analyze environmental data but also closes the gap between technical and practical solutions. Through the application of new data-driven methodologies, it develops a tool that can be integrated within intelligent water management systems, public health monitoring systems, and policy guidelines for environmental sustainability. The result is a scalable, dependable, and accessible tool for groundwater potability prediction that can assist communities, researchers, and policymakers to guarantee the provision of clean and safe drinking water.

1.2 PROBLEM STATEMENT

Groundwater contamination has become a growing concern due to rapid urbanization, industrial discharge, improper waste disposal, overuse of chemical fertilizers, and leakage from septic systems. Contaminants such as nitrates, sulfates, chlorides, heavy metals, and excessive levels of Total Dissolved Solids (TDS) can severely degrade water quality and pose serious health risks including gastrointestinal diseases, reproductive issues, and long-term organ damage.

Currently, the standard approach to assessing groundwater quality involves manual sample collection and laboratory analysis, which, while accurate, is highly time-consuming, costly, and resource-intensive. This process requires trained personnel, expensive laboratory equipment, and considerable time for data processing and interpretation. As a result, frequent or large-scale testing is not feasible in many developing countries or remote areas. Additionally, delays in identifying non-potable water sources can have devastating consequences for public health and local ecosystems.

Over the last several years, Machine Learning (ML) has proven to be a useful environment data analysis tool that provides the potential to automate classification using historical patterns. Nonetheless, there are ML-based systems for groundwater quality classification that are based on a specific limited list of algorithms, fail to use adequate ensemble methods to enhance predictive performance, or are not capable of generalizing to noisy or skewed datasets. There is also a need to bring together various complementary algorithms within one platform that can make the system accurate and robust.

We seek to address this gap by developing a multi-model machine learning system that integrates multiple models such as Multilayer Perceptron (MLP), Quadratic Discriminant Analysis (QDA), Extra Trees Classifier, CatBoost Classifier, and Stacking Classifier to forecast the potability or non-potability of groundwater using an array of input features. The aim is to enhance the accuracy of the classification, minimize the reliance on laboratory tests, and help toward more intelligent and environmentally friendly water quality monitoring systems.

1.3 MOTIVATION

Clean and safe drinking water remains an essential need for humans to be healthy and well. Nonetheless, in much of the world, and especially in rural and semi-urban areas, people are mainly relying on groundwater for their source of water. The alarming increase in groundwater pollution due to industrial waste, agricultural runoff, and poor waste management has made it essential to monitor water quality regularly. Unfortunately, the current practices for groundwater testing—relying on laboratory-based methods—are often time-consuming, expensive, and not scalable, especially in underdeveloped and resource-constrained areas. This leads to delayed responses to water contamination and continued use of unsafe water, putting millions of lives at risk.

The motivation behind this project stems from the urgent need for a faster, more accessible, and intelligent solution for groundwater potability assessment. The advancement of machine learning (ML) has opened new opportunities in predictive analytics, enabling the classification of water quality based on chemical parameters with high accuracy. By leveraging historical groundwater data and applying modern ML algorithms, it is possible to develop a system that can predict water safety instantly without the need for physical testing every time.

Additionally, the project is inspired by the potential of combining multiple machine learning models to improve prediction reliability. While individual models may perform well under certain conditions, their limitations can be overcome through ensemble methods such as stacking, which harness the collective intelligence of different algorithms. This motivates the use of classifiers like Multilayer Perceptron (MLP), Quadratic Discriminant Analysis (QDA), CatBoost, Extra Trees, and their combination via a Stacking Classifier, to ensure that the predictions are robust, scalable, and accurate across various datasets.

Another key motivation is the practical impact such a system can have. An automated, data-driven prediction model can be used by government agencies, NGOs, health workers, and communities to monitor groundwater quality in real-time, make informed decisions, and ultimately reduce the risk of waterborne diseases. The integration of communication tools like Twilio also provides an opportunity to send alerts instantly, making the system even more useful in real-world scenarios.

1.4 SUSTAINABLE DEVELOPMENT GOAL OF THE PROJECT

The program is deeply anchored in the Sustainable Development Goals (SDGs) of the UN, particularly SDG 6: Clean Water and Sanitation, which calls for universal access to water and sustainable sanitation management. Access to safe, affordable water for drinking is not only an absolute human right but also an essential element of public health, economic development, and environmental sustainability. Nevertheless, numerous communities globally, particularly rural and low-income communities, continue to experience shortages of water, pollution, and the lack of reliable systems for monitoring water quality.

The Groundwater Quality Measurement System using Machine Learning directly supports SDG 6 by offering a technological solution for rapid, cost-effective, and scalable water quality monitoring. By utilizing physicochemical parameters such as pH, TDS, Nitrates, Sulfates, and others, the project provides an intelligent method to classify water as potable or non-potable without relying solely on laboratory testing.

Furthermore, the project contributes to Sustainable Development Goal 3: Good Health and Well-being by helping to minimize waterborne illnesses through enhanced evaluation of water quality.

CHAPTER 2

LITERATURE SURVEY

2.1 OVERVIEW OF THE RESEARCH AREA

Groundwater constitutes nearly 30% of the Earth's freshwater and serves as a primary drinking water source for billions across the globe. However, rapid industrial growth, intensive farming practices, and rising population densities have significantly contributed to the contamination of these vital reserves. Substances such as nitrates, chlorides, sulfates, and heavy metals, when found in elevated concentrations, pose serious health threats, underscoring the urgent need for efficient groundwater monitoring to support both public health and sustainable environmental management.

Conventional techniques for evaluating the quality of groundwater entail gathering samples and then analyzing physicochemical parameters in a lab. While these methods are accurate, they are also resource-intensive, time-consuming, and not feasible for continuous or large-scale monitoring, especially in remote or underdeveloped areas. To address these limitations, the research community has increasingly turned to machine learning as a viable alternative for automating the classification of water potability based on chemical properties.

This project enhances the research domain by applying and evaluating various machine learning algorithms—such as Multilayer Perceptron (MLP), Quadratic Discriminant Analysis (QDA), Extra Trees Classifier, and CatBoost Classifier—and integrating them through a Stacking Ensemble strategy. Beyond its technical contributions to AI-driven environmental monitoring, the project also supports sustainable development by offering accessible, scalable, and data-informed solutions for assessing water quality.

2.2 EXISTING MODELS AND FRAMEWORKS

Groundwater quality assessment has become a crucial area of research, especially with rising concerns about pollution and its impact on public health. Traditionally, statistical models such as Logistic Regression, Naïve Bayes, and Decision Trees were employed to classify water potability based on physicochemical parameters like pH, Total Dissolved Solids (TDS), and Nitrates. These models are appreciated for their simplicity and ease of interpretation, and they perform adequately when datasets are well-structured and noise-free.

Recent developments in groundwater quality classification have seen the adoption of more advanced machine learning models that deliver greater accuracy and adaptability in capturing complex, multidimensional feature relationships. Progress in this domain includes ensemble techniques such as Random Forests, Extra Trees, and CatBoost, which enhance predictive stability and mitigate overfitting by integrating multiple model outputs. Notably, Stacking Classifiers—employing a meta-learner to combine predictions from diverse base models—have demonstrated superior performance by effectively harnessing the strengths of each constituent algorithm.

These data-driven models have drawbacks despite their advantages. Many require large, clean, and balanced datasets and are sensitive to missing values or skewed class distributions. Additionally, most ignore domain-specific insights such as geological influences, hydrological cycles, or regulatory thresholds, which are crucial for context-aware decision-making.

To address these challenges, hybrid methodologies have been introduced. These often involve integrating machine learning models with advanced feature engineering techniques—such as correlation analysis and wavelet transforms—to enhance pattern recognition and improve model generalization. Additionally, some recent research has leveraged deep learning and multi-task learning approaches to boost predictive performance on complex, high-dimensional environmental datasets.

In conclusion, while traditional statistical and early machine learning models laid the foundation for groundwater potability classification, their limitations emphasize the growing need for more robust, interpretable, and context-aware frameworks—especially those capable of integrating domain knowledge and handling real-world uncertainties in environmental monitoring.

2.3 LIMITATION IDENTIFIED FROM LITERATURE SURVEY

A review of Groundwater quality measure reveals several key limitations:

1. Data Dependency and Quality Issues

Most machine learning models rely heavily on large volumes of clean, labeled data. However, missing values, unbalanced classes (such as fewer potable samples), and noisy measurements are common problems with groundwater datasets that can impair model performance.

2. Limited Incorporation of Domain Knowledge

Many studies treat groundwater classification as a purely statistical or pattern recognition task, overlooking critical domain-specific factors like geological context, seasonal variation, or hydrological flow, which could enhance model interpretability and reliability.

3. Overfitting in Complex Models

Advanced models like deep neural networks or ensemble learners often exhibit high accuracy on training data but may overfit, leading to poor generalization on unseen data—especially when datasets are limited or not diverse.

4. Neglect of Temporal Dynamics

Most models treat groundwater data as static, ignoring the temporal evolution of water quality due to factors like seasonal rainfall, industrial discharge, or agricultural runoff. This limits their ability to forecast future contamination trends effectively.

5. Lack of Interpretability

Many high-performing models, such as XGBoost, and deep learning networks, function as "black boxes," making it difficult for stakeholders (e.g., environmental agencies) to understand the basis of predictions or derive actionable insights.

2.4 RESEARCH OBJECTIVES

Research Objectives:

1. To develop a robust and accurate machine learning framework for predicting groundwater potability using physicochemical parameters such as pH, TDS, Hardness, Chlorides, and Nitrates.
2. To evaluate and compare the performance of various supervised learning models including MLP, QDA, Extra Trees, CatBoost, and Stacking Classifier.
3. Using ensemble and hybrid techniques (e.g., stacking) for improving classification accuracy, fighting overfitting, and generalizing.
4. To examine the shortcomings of conventional models and emphasize the benefits of new ensemble and deep learning-based approaches.
5. To ensure model interpretability and scalability for practical deployment in both rural and urban water monitoring systems.

2.5 PRODUCT BACKLOG

| SNo. | User Stories of Ensemble-Based Classification of Groundwater Potability Using Multi-Model Stacking Technique |
|--------|---|
| #US 1 | As a researcher, I want to define the scope and objectives of the groundwater classification project so that I can guide the research direction clearly. |
| #US 2 | As a data scientist, I want to collect and analyze groundwater datasets with chemical parameters (e.g., pH, TDS, Nitrates) so that I can build a reliable ML model. |
| #US 3 | As a developer, I want to preprocess the data by handling missing values, scaling features, and balancing classes so that models can learn more effectively. |
| #US 4 | As a modeler, I want to implement multiple ML models like MLP, QDA, CatBoost, and Extra Trees so that I can compare their predictive performance. |
| #US 5 | As a machine learning engineer, I want to design a Stacking Classifier combining outputs of base models so that I can enhance overall classification accuracy. |
| #US 6 | As a developer, I want to evaluate models so that I can assess the effectiveness of each approach. |
| #US 7 | As a developer, I want to visualize confusion matrices and performance plots so that I can interpret model behavior and insights more clearly. |
| #US 8 | As a researcher, I want to identify and document limitations in current approaches so that I can justify the need for a multi-model ensemble solution. |
| #US 9 | As a developer, I want to deploy a user-friendly web interface for real-time groundwater classification so that users can easily assess water safety. |
| #US 10 | As a system integrator, I want to integrate Twilio API for sending SMS alerts so that field users can receive real-time potability notifications. |

Table 2.1 Product Backlog

2.6 PLAN OF ACTION

Phase 1: Project Planning & Setup

- Establish well-defined research goals and clearly outline the scope of the project.
- Identify the tools and frameworks to be used, including Scikit-learn, CatBoost, and Python-based libraries.
- Set up the development environment using platforms like Google Colab or Jupyter Notebook.

Phase 2: Data Collection & Preprocessing

- Collect groundwater quality datasets with features such as pH, TDS, Hardness, Chlorides, Nitrates, and Sulfates.
- Conduct Exploratory Data Analysis (EDA) to understand feature relationships and distributions.
- Save the cleaned and processed dataset for model training and testing.

Phase 3: Model Development

- MLP Classifier: Build and train a Multi-Layer Perceptron model using Scikit-learn.
- QDA Classifier: Implement and evaluate a Quadratic Discriminant Analysis model to capture probabilistic relationships.
- Extra Trees Classifier: Train multiple randomized decision trees to achieve a higher accuracy and to counteract overfitting.
- CatBoost Classifier: For effective and high-performance gradient boosting, particularly with mixed data types, use CatBoost.

Phase 4: Ensemble Learning with Stacking

- Combine base models (MLP, QDA, Extra Trees, CatBoost) using a Stacking Classifier with a meta-learner for improved classification performance.
- Train and evaluate the ensemble model to ensure robustness.

Phase 5: Model Evaluation

- Evaluate the metrics of model.
- Visualize confusion matrices and bar plots of metrics to provide a comparative performance summary.

Phase 6: Real-Time Interface & Deployment

- Create and put into use an intuitive web interface that lets users enter water quality parameters and get real-time groundwater potability predictions.
- Integrate Twilio API to send SMS notifications with model predictions.

CHAPTER 3

SPRINT PLANNING AND EXECUTION METHODOLOGY

3.1 SPRINT 1

3.1.1 OBJECTIVES WITH USER STORIES OF SPRINT 1

The objective of the first sprint is to finish the groundwork including acquisition of data, data preparation, and initiating the model.

The full set of sprint 1 user stories is shown in table 3.1 below.

| SNo. | User Stories of Ensemble-Based Classification of Groundwater Potability Using Multi-Model Stacking Technique |
|-------|---|
| #US 1 | As a researcher, I want to define the scope and objectives of the groundwater classification project so that I can guide the research direction clearly. |
| #US 2 | As a data scientist, I want to collect and analyze groundwater datasets with chemical parameters (e.g., pH, TDS, Nitrates) so that I can build a reliable ML model. |
| #US 3 | As a developer, I want to preprocess the data by handling missing values, scaling features, and balancing classes so that models can learn more effectively. |
| #US 4 | As a modeler, I want to implement multiple ML models like MLP, QDA, CatBoost, and Extra Trees so that I can compare their predictive performance. |
| #US 5 | As a machine learning engineer, I want to design a Stacking Classifier combining outputs of base models so that I can enhance overall classification accuracy. |

Table 3.1 Detailed Sprint 1 User Stories

3.1.2 FUNCTIONAL DOCUMENT

3.1.2.1 Introduction

The Groundwater Quality Measurement System is a smart, AI-powered application that enables users to determine the potability of groundwater through an automated and data-driven approach. The system collects important water quality parameters as input, applies a pre-trained machine learning model to evaluate the water's safety, and provides immediate feedback to the user. In cases where the water is found to be unsafe for drinking, the system is capable of sending a real-time SMS alert through integration with the Twilio messaging API. This solution is particularly designed to address the need for quick, accessible, and scalable water testing tools in regions where laboratory testing may not be available or practical. By digitizing the water quality testing process and integrating intelligent analysis and alerting, this system supports both personal safety and broader public health goals.

3.1.2.2 Product Goal

The goal of the Groundwater Quality Measurement System is to provide an accurate, real-time, and user-friendly tool that allows individuals and organizations to assess groundwater safety without needing technical expertise or laboratory access. The system aims to reduce the time, cost, and effort required to perform routine water testing by leveraging the power of machine learning and cloud communication services. By delivering high-accuracy predictions and instant alerts, the system encourages more frequent and widespread testing, which contributes to improved water management and preventive public health measures. Ultimately, the product aspires to become a critical resource in water quality monitoring, especially in rural and underdeveloped areas where water safety remains a pressing issue.

3.1.2.3 Demography (Users, Location)

The system is intended to serve a diverse range of users across multiple geographic locations.

Target Users:

- Individuals and household dependent on groundwater for drinking
- Government bodies and municipal water management departments
- NGOs and healthcare workers involved in environmental and public health
- Water quality testing professionals and organizations

Target Locations:

- Rural and remote areas lacking access to laboratory testing
- Educational institutions, hospitals, and local councils performing regular water checks

3.1.2.4 Business Process

The core business processes supported by the Groundwater Quality Measurement System include:

1. Water Parameter Input

Users are prompted to enter seven critical water quality parameters: pH, turbidity, TDS (total dissolved solids), nitrate concentration, temperature, chloramines level, and sulfate content.

2. Prediction Process:

The system analyzes the input parameters through a machine learning model built using a Stacking Classifier, which integrates multiple base learners including MLP, QDA, CatBoost, and Extra Trees Classifier. Based on this ensemble approach, the system generates a binary output indicating whether the water is "Potable" or "Not Potable."

3. SMS Notification:

If the result is "Not Potable," the system automatically generates a detailed message including the input values and reasons for non-potability and sends it via SMS using the Twilio REST API.

4. Result Display and Logging (Future Scope):

Results will be displayed on-screen for instant interpretation. In future versions, input values, prediction results, and SMS delivery status may be stored in a lightweight database for historical tracking and reporting.

3.1.2.5 Features

Feature 1: Data Acquisition and Cleaning

Description:

The system collects historical groundwater quality data, focusing on key chemical features like pH, TDS, Hardness, Chlorides, Nitrates, Sulfates, and Conductivity.

User Story:

As a developer, I would like to assemble and preprocess groundwater datasets so that I will have solid input to use for training machine learning models.

Feature 2: Normalization and Preprocessing of Data

Description:

The system carries out preprocessing by scaling features with MinMaxScaler, handling missing values (e.g., mean/median imputation), and optionally resolving class imbalance.

User Story:

As a developer, I want to preprocess the dataset to ensure all features are normalized and clean so that machine learning models can learn effectively.

Feature 3: Exploratory Data Analysis (EDA)

Description:

The system analyzes feature distributions, correlations, and class imbalance using plots and summary statistics to understand the structure and integrity of the dataset.

User Story:

As a data analyst, I want to conduct EDA so that I can gain insights into the dataset and make informed modeling decisions.

Feature 4: Model Implementation

Description:

Individual machine learning models including MLP, QDA, CatBoost, and Extra Trees are implemented and trained on the preprocessed dataset.

User Story:

As a researcher I want to implement various ML models to compare their predictive capabilities in groundwater potability classification.

Feature 5: Stacking Classifier Construction

Description:

The system integrates many base models outputs via a meta-model to create a Stacking Classifier to enhance prediction accuracy via ensemble learning.

User Story:

As a machine learning engineer, I want to build a Stacking Classifier so that I can enhance the reliability and performance of groundwater potability predictions.

3.1.2.6 Authorization Matrix

| Role | Access Level |
|----------------|--|
| Admin | Access system configuration, model training workflows, user management, and data pipeline oversight. |
| Data Scientist | Access to data preprocessing tools, model implementation, training, and evaluation modules. |
| Analyst | Access to model outputs, evaluation metrics and visualization |
| Researcher | Access to experiment logs, comparative model results and domain-specific tuning parameters. |
| Guest User | Limited access to view publicly shared prediction results and general system documentation. |

Table 3.2 Authorization Matrix

3.1.2.7 Assumptions

- The selected chemical parameters are assumed to be sufficient indicators of water potability, and no domain-specific transformation is required before model training.
- Team members are assumed to have baseline knowledge of Python, scikit-learn, pandas, and other required ML libraries to implement preprocessing and models.
- The potability prediction task is assumed to be a binary classification problem with only two labels: "Potable" and "Not Potable".
- The use of ensemble stacking (meta-learning) will lead to improving model performance
- The stacking classifier is pre-trained and validated for accuracy and consistency.

3.1.2 ARCHITECTURE DOCUMENT

3.1.2.1 Application

A data-driven software program called the Groundwater Potability Prediction System was created to categorize water samples as either potable or non-potable depending on important physicochemical characteristics. During Sprint 1, the project focused on building the foundation of this system through dataset acquisition, cleaning, preprocessing, exploratory data analysis (EDA), and initial implementation of individual machine learning models. The primary objective was to create a predictive framework using various models—MLP, QDA, CatBoost, Extra Trees—and to integrate them into a Stacking Classifier to enhance overall performance.

3.1.2.2 Microservices

The system follows a modular architecture to ensure a clear separation of concerns and support scalability. During Sprint 1, three fundamental services were developed:

- **Data Preprocessing Service:** Handles missing value imputation, feature normalization (e.g., using MinMaxScaler), and data splitting for training and testing purposes.
- **Model Training Service:** Responsible for training individual classifiers (MLP, QDA, CatBoost, Extra Trees) and the Stacking ensemble. It also includes hyperparameter tuning and performance evaluation based on classification metrics.
- **Exploratory Analysis Service:** Conducts EDA to generate feature distribution plots, correlation heatmaps, and class balance visualizations.

3.1.2.3 Event-Driven Architecture

In order to train the model, the application uses an event-driven design. The model training service is started by an event after the data preprocessing is finished. Evaluation results, such as metric summaries and confusion matrices, are automatically produced following training. The internal pipeline is already set up around asynchronous event flow, which facilitates easier transitions into subsequent integration stages, even though real-time user interaction and SMS notifications were scheduled for later sprints.

3.1.2.4 Serverless Architecture

Although not fully realized in Sprint 1, the system is designed to support a serverless architecture in subsequent sprints. Services like data preprocessing, model prediction, and result delivery are planned to be deployed as on-demand cloud functions, reducing infrastructure overhead and ensuring scalable, cost-efficient execution.

3.1.2.5 Architecture Diagram

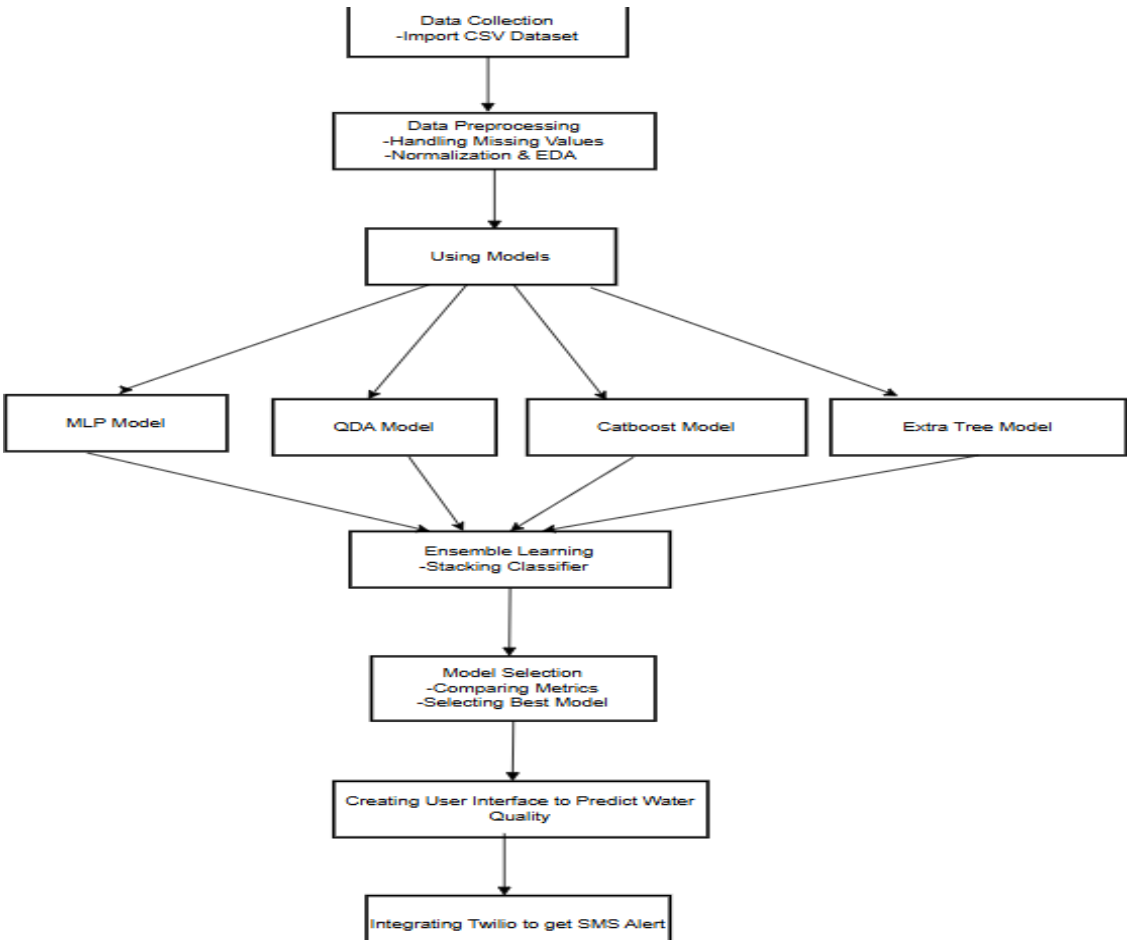


Figure 3.1 Architecture Diagram

3.1.2.5 Data Exchange Contract

3.1.2.5.1 Frequency of Data Exchanges

The frequency of data exchanges in the Groundwater Quality Measurement System is primarily event-driven and real-time. Data is exchanged when a user submits water quality inputs, immediately triggering the prediction process and, if necessary, the SMS alert service. Unlike batch systems, the architecture is designed to respond to user actions instantly, ensuring low latency and high responsiveness.

3.1.2.5.2 Data Sets

The data sets used in the Groundwater Quality Measurement System consist of both input and output attributes. Input data includes chemical indicators like pH, TDS, Hardness, Chlorides, Nitrates, Sulfates, and Conductivity—all required for prediction. The output data set comprises the potability result, the model used for prediction (e.g., MLP, QDA, Stacking), and reasons for non-potability if applicable.

3.1.2.5.3 Mode of Exchanges (API, File, Queue etc.,)

The Groundwater Quality Measurement System utilizes various modes of data exchange based on the functionality of each component. Internally, function calls and direct data passing between modules handle communication, especially between the user interface and the prediction engine. For external communications, such as sending SMS notifications, the system uses the Twilio API to deliver messages in real time. Additionally, for logging and offline access, data can be stored or exported using CSV files or lightweight databases like SQLite, providing simple file-based exchange options for persistence or reporting purposes.

3.1.3 Result Analysis

3.1.4.1 Dataset Preprocessing

- Outcome: Missing values were handled using mean/median imputation. All features were normalized using MinMaxScaler.
- Observation: Parameters such as pH, TDS, and Hardness had noticeable variance and required scaling for consistent model performance.

3.1.4.2 EDA and Feature Correlation

- Outcome: Correlation heatmap and distribution plots revealed strong relationships between TDS, Sulfates, and Conductivity. Bar charts displayed imbalanced class distribution and variance across features.
- Observation: Highly correlated features such as TDS and Conductivity suggest redundancy, while Nitrates and pH showed weak correlation but had significant influence on classification. This aligned with known chemical indicators of water quality.

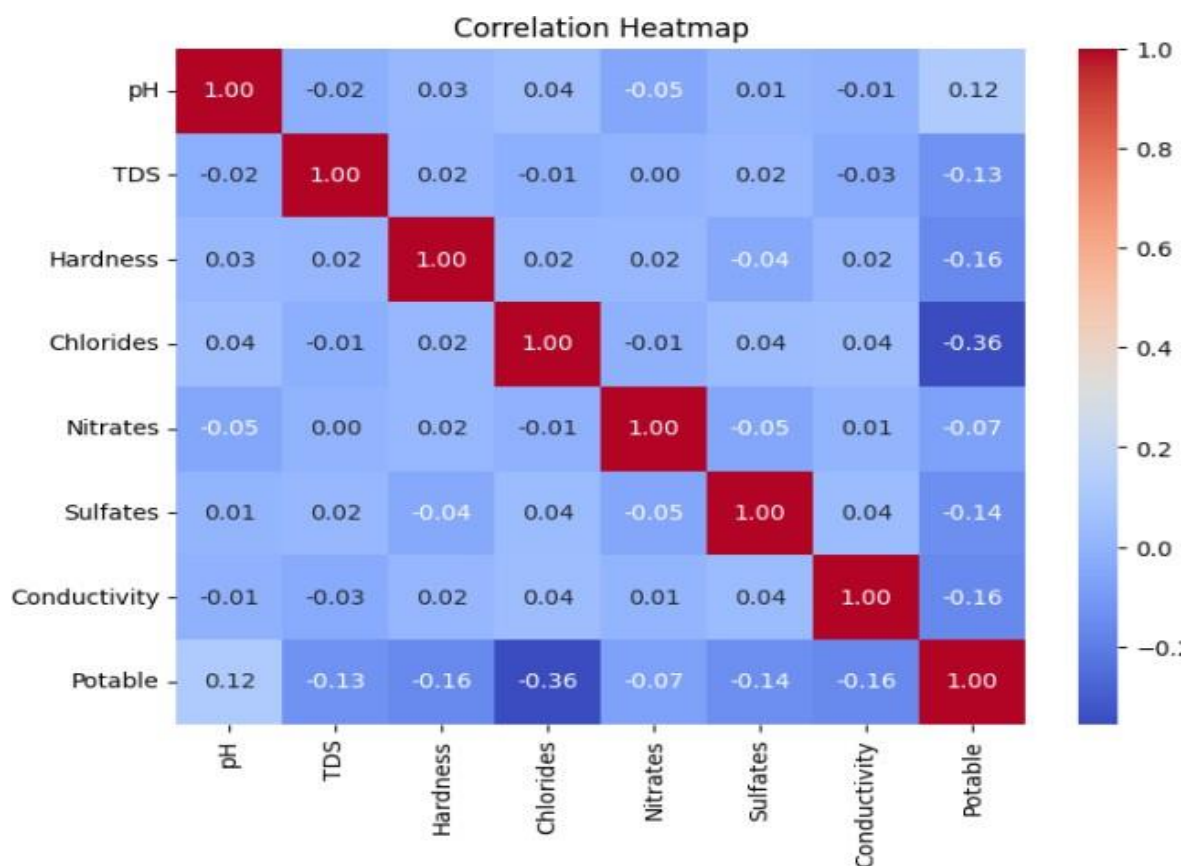


Figure 3.2 Correlation Heatmap

Feature Distributions

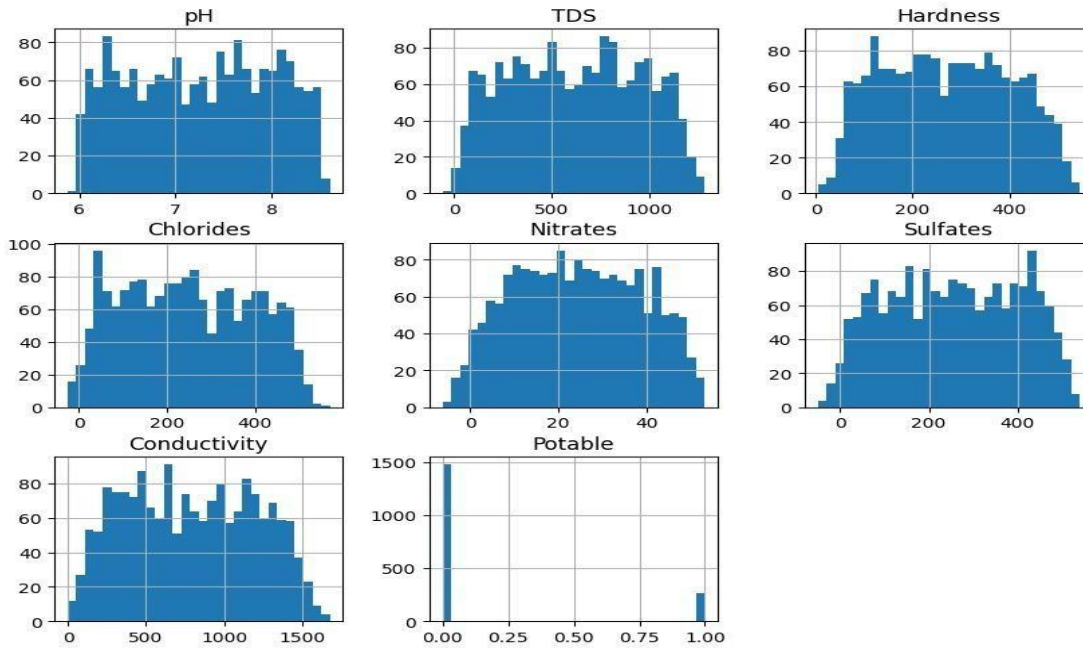


Figure 3.3 Exploratory Data Analysis

3.1.4.3 Model Training & Evaluation

| MODEL | ACCURACY | PRECISION | F1-SCORE | AUC |
|-------------------------|----------|-----------|----------|-----|
| Multi- Layer Perceptron | 92% | 93% | 73% | 98% |
| QDA | 88% | 67% | 53% | 93% |
| Catboost | 97% | 88% | 92% | 99% |
| Extra Tree Classifier | 92% | 93% | 67% | 98% |
| Stacking Classifier | 97% | 89% | 90% | 99% |

Table 3.3 Result of Sprint 1

3.1.4 SPRINT RETROSPECTIVE

| What went well | What went poorly | What ideas do you have | How should we take action |
|--|---|---|--|
| The project scope and objectives (#US 1) were clearly defined and aligned across team members. | Initial discussions on scope took longer due to unclear roles in the beginning. | Organize a brief model selection workshop with all stakeholders. | Add a short pre-sprint workshop for scope alignment. |
| Groundwater dataset was successfully collected and analyzed (#US 2). | Some datasets had missing or inconsistent values that required manual fixing. | Implement automated data validation at ingestion. | Automate data cleaning and maintain a data format guide. |
| Data preprocessing (#US 3) was completed with missing value handling and normalization. | Some features were over-normalized affecting interpretability. | Limit scaling to numeric input features only; skip categorical/binary ones. | Revisit preprocessing steps and include feature importance review in next sprint. |
| Multiple ML models (MLP, QDA, CatBoost, Extra Trees) were successfully implemented (#US 4). | Hyperparameter tuning for each model was time-consuming and not yet optimized. | Use grid search with cross-validation or automated tools. | Set up a tuning pipeline to be reused across all models in Sprint 2. |
| Stacking Classifier (#US 5) showed promising preliminary results with improved accuracy. | Model evaluation took longer due to class imbalance and metric variations. | Include sampling in train-test split. | Future evaluations will employ stratified k-fold cross-validation to ensure balanced representation and more reliable performance metrics. |

Table 3.4 Sprint Retrospective of sprint 1

3.2 SPRINT 2

3.2.1 OBJECTIVES WITH USER STORIES OF SPRINT 2

The objective of Sprint 2 is to analyze and visualize the performance of deployed machine learning models by using classification measures and confusion matrices, discover weaknesses in stand-alone models, and ascertain the effectiveness of the stacking ensemble strategy.

The full set of sprint 2 user stories is shown in table 3.5 below.

| | |
|---------------|--|
| #US 6 | As a developer, I want to evaluate models so that I can assess the effectiveness of each approach. |
| #US 7 | As a developer, I want to visualize confusion matrices and performance plots so that I can interpret model behavior and insights more clearly. |
| #US 8 | As a researcher, I want to identify and document limitations in current approaches so that I can justify the need for a multi-model ensemble solution. |
| #US 9 | As a developer, I want to deploy a user-friendly web interface for real-time groundwater classification so that users can easily assess water safety. |
| #US 10 | As a system integrator, I want to integrate Twilio API for sending SMS alerts so that field users can receive real-time potability notifications. |

Table 3.5 Detailed Sprint 2 User Stories

3.2.2 FUNCTIONAL DOCUMENT

3.2.2.1 Introduction

The Groundwater Potability Prediction System leverages advanced ensemble learning techniques to classify groundwater as potable or non-potable based on physicochemical parameters. This sprint focuses on evaluating model performance using classification metrics, visualizing insights via confusion matrices and performance plots, deploying a web-based interface for real-time predictions, and integrating Twilio API to deliver SMS-based alerts. The goal is to combine data-driven methods with accessibility to empower users and public health workers in making informed water safety decisions.

3.2.2.2 Product Goal

The main objective of Sprint 2 is to ensure the reliability and accessibility of the groundwater classification system by:

- Evaluating models metrics.
- Visualizing performance via confusion matrices and plots to interpret model behavior.
- Documenting limitations of current models to justify the need for a multi-model ensemble.
- A web interface was deployed to enable users to input relevant parameters and instantly receive groundwater classification results.
- Integrating Twilio API to provide real-time SMS notifications about groundwater potability.

3.2.2.3 Demography (Users, Location)

Users:

Target Users: Environmental scientists, public health officials, rural community workers, researchers, and machine learning practitioners.

User Characteristics: Users may vary in technical expertise. Some are domain experts in water quality, while others are end-users requiring an easy-to-use system for assessing potability.

Location:

Target Location: Primarily rural and semi-urban regions with limited access to laboratory testing facilities, as well as institutions and agencies monitoring groundwater quality globally.

3.2.2.4 Business Process

The key business processes include:

Model Evaluation:

Evaluating Models such as MLP, QDA, CatBoost, Extra Trees, and the Stacking Classifier.

Performance Visualization:

Confusion matrices and classification performance plots (e.g., bar graphs, ROC curves) are used to visualize and compare results across models.

Limitations Documentation:

Observed weaknesses (e.g., false positives/negatives, class imbalance sensitivity) are documented to highlight the necessity of ensemble learning.

Web Interface Deployment:

A real-time interface is created where users can input groundwater parameters and instantly receive classification feedback.

Twilio SMS Integration:

The system sends potability results via SMS to designated mobile numbers using the Twilio API, improving real-world responsiveness and reach.

3.2.2.5 Features

Feature 1: Model Evaluation Metrics

Description:

Evaluates ML models using accuracy, precision, recall, F1-score, and AUC to determine the most effective predictors of water potability.

User Story:

As a developer, I want to evaluate models using standard metrics so that I can assess their effectiveness.

Feature 2: Confusion Matrix and Performance Plot Visualization

Description:

Visualizes each model's performance using confusion matrices, classification reports, and accuracy.

User Story:

As a developer, I want to visualize confusion matrices and model plots so that I can interpret classification behavior more clearly.

Feature 3: Limitation Analysis and Documentation

Description:

Identifies and records the shortcomings of single classifiers, justifying the use of ensemble learning.

User Story:

As a researcher, I want to document the limitations of current models so I can support the ensemble approach.

Feature 4: Real-Time Web Interface Deployment

Description:

A user-friendly web interface is developed for real-time potability prediction using trained models.

User Story:

As a developer, I want to deploy a web interface so that users can check water potability instantly.

Feature 5: Visualization of Model Outputs

Description:

Model predictions are visualized using plots comparing actual vs. predicted prices and training loss curves to analyze learning behavior.

User Story:

As a developer, I want to visualize the model outputs, including prediction vs. actual plots and loss curves, so that I can better understand model performance and learning behavior.

Feature 6: Twilio SMS Notification Integration

Description:

Twilio is integrated to deliver SMS alerts based on classification results, increasing accessibility for field user.

User Story:

As a system integrator, I want to use Twilio API to send SMS alerts so that users receive real-time potability updates.

| Role | Access Level |
|-----------------------|---|
| Project Administrator | Full access to project planning, dataset management, model deployment, and evaluation dashboards. |
| Developer | Access to data preprocessing, feature engineering, and evaluation of model performance. |
| Researcher | Access to model performance reports, limitations, and comparative analysis |
| Researcher | Access to deploy Twilio API and manage alert systems. |
| Stakeholder | Read-only access to web interface and SMS notifications |

Table 3.6 Authorization Matrix

3.2.2.7 Assumptions

- The preprocessed dataset remains consistent and accurate throughout model evaluation.
- With little delay, the trained models can be incorporated into a real-time, lightweight interface.
- Performance plots and confusion matrices will be interpreted by team members familiar with ML classification metrics.
- The classification problem remains binary (Potable vs. Not Potable) with no multiclass extensions in this sprint.

3.2.3 ARCHITECTURE DOCUMENT

3.2.3.1 Application

In Sprint 2, the Groundwater Potability Prediction System evolved from a core predictive engine into a complete end-to-end application capable of delivering real-time predictions through a web interface. The system now not only classifies groundwater as potable or non-potable using machine learning models but also evaluates their performance with standard metrics, visualizes results through plots and confusion matrices, and integrates Twilio SMS services for delivering prediction outcomes. The enhanced application provides an accessible tool for both technical users and field operators, ensuring efficient decision-making and public safety.

3.2.3.2 Microservices

Sprint 2 expanded the microservices-based architecture by adding new services and integrating previously independent components:

- **Evaluation & Visualization Service:** Computes accuracy, precision, recall, F1-score, and AUC for each model. To facilitate interpretability, it creates comparison charts, confusion matrices, and performance plots.
- **Ensemble Justification Module:** Analyzes and documents the limitations of single models (e.g., QDA's low recall for potable samples) and provides justification for the ensemble stacking strategy.
- **User Interface Service:** Users can enter water quality parameters into a real-time Streamlit-based web application and get immediate classification results.
- **Twilio Notification Service:** Automatically sends potability results to the user's phone number via SMS using the Twilio API, enhancing system usability in rural or offline scenarios.

3.2.3.3 Event-Driven Architecture

The application utilizes an event-driven design during the model training workflow. When the data preprocessing is complete, an event triggers the model training service. After training, evaluation results are automatically generated, including confusion matrices and metric summaries. While real-time user interaction and SMS notifications were planned for later sprints, the internal pipeline is already structured around asynchronous event flow, enabling smoother transitions into future integration phases.

3.2.3.4 Serverless Architecture

Although not fully realized in Sprint 1, the system is designed to support a serverless architecture in subsequent sprints. Services like data preprocessing, model prediction, and result delivery are planned to be deployed as on-demand cloud functions, reducing infrastructure overhead and ensuring scalable, cost-efficient execution.

3.2.3.5 Database

3.2.3.5.1 ER Diagram

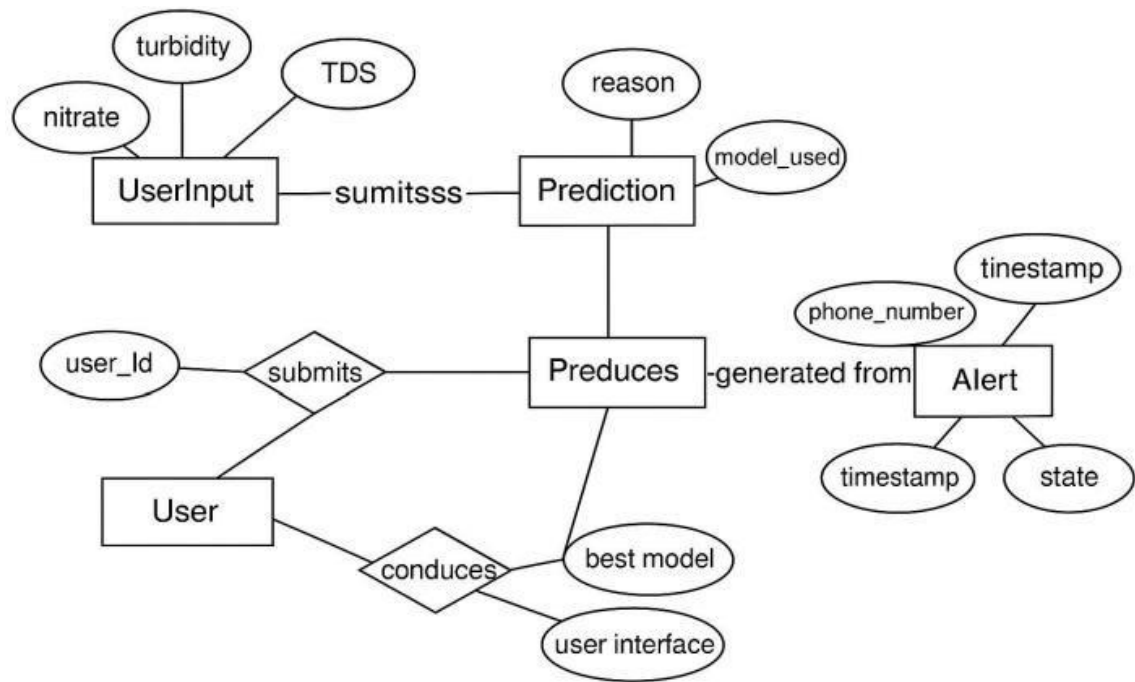


Figure 3.4 ER Diagram

3.2.3.5.2 Schema Design

The schema design of the Groundwater Quality Measurement System connects users with the groundwater tests they conduct. Each User is linked to a Conducts table, which associates them with specific test inputs stored in the Input table. These inputs include parameters like pH, turbidity, TDS, nitrate, temperature, chloramines, and sulfate. The Prediction table stores the results generated by different machine learning models such as MLP, QDA, Extra Trees, CatBoost, and Stacking. Finally, the TwilioSMS table logs all messages sent to users, recording the phone number, content, and delivery status. This design ensures a clear flow from input to prediction to SMS alert.

3.1.2.5 Data Exchange Contract

3.1.2.5.1 Frequency of Data Exchanges

The frequency of data exchanges in the Groundwater Quality Measurement System is primarily event-driven and real-time. Data is exchanged when a user submits water quality inputs, immediately triggering the prediction process and, if necessary, the SMS alert service. Unlike batch systems, the architecture is designed to respond to user actions instantly, ensuring low latency and high responsiveness.

3.1.2.5.2 Data Sets

The data sets used in the Groundwater Quality Measurement System consist of both input and output attributes. Input data includes chemical indicators like pH, TDS, Hardness, Chlorides, Nitrates, Sulfates, and Conductivity—all required for prediction. The output data set comprises the potability result, the model used for prediction (e.g., MLP, QDA, Stacking), and reasons for non-potability if applicable.

3.1.2.5.3 Mode of Exchanges (API, File, Queue etc.,)

The Groundwater Quality Measurement System utilizes various modes of data exchange based on the functionality of each component. Internally, function calls and direct data passing between modules handle communication, especially between the user interface and the prediction engine. For external communications, such as sending SMS notifications, the system uses the Twilio API to deliver messages in real time. Additionally, for logging and offline access, data can be stored or exported using CSV files or lightweight databases like SQLite, providing simple file-based exchange options for persistence or reporting purposes.

3.2.4 RESULT ANALYSIS

a) Multi-Layer Perceptron

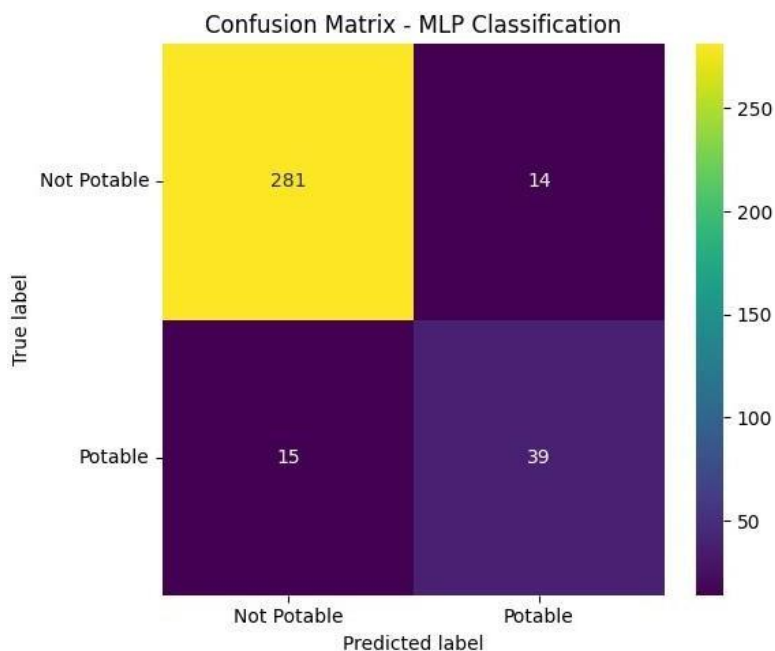


Fig 3.5 Confusion Matrix of MLP Classification

The MLP (Multilayer Perceptron) classification model for groundwater quality prediction demonstrates strong performance, especially in detecting non-potable water. The classification metrics—sensitivity, precision, F1-score, and specificity—highlight the model’s effectiveness, particularly in recognizing non-potable water.

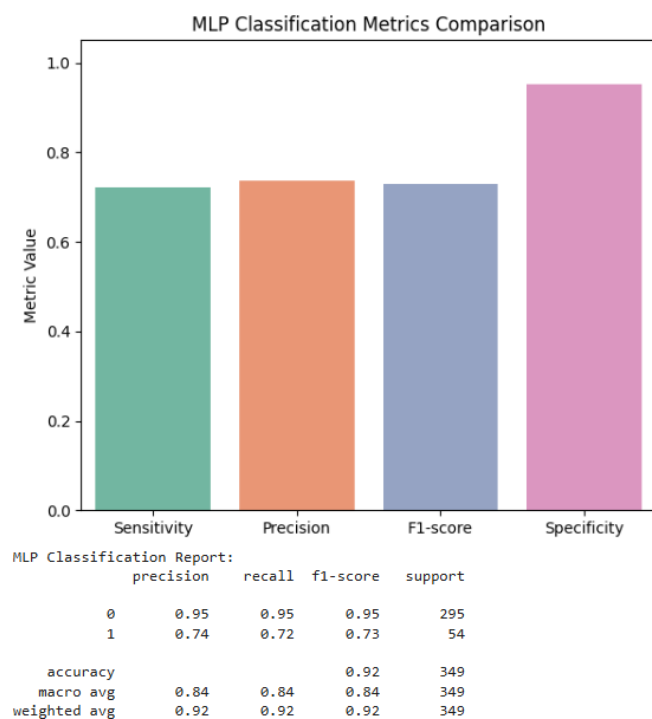


Fig 3.6 Classification Report of MLP

b) Quadratic Discriminant Analysis

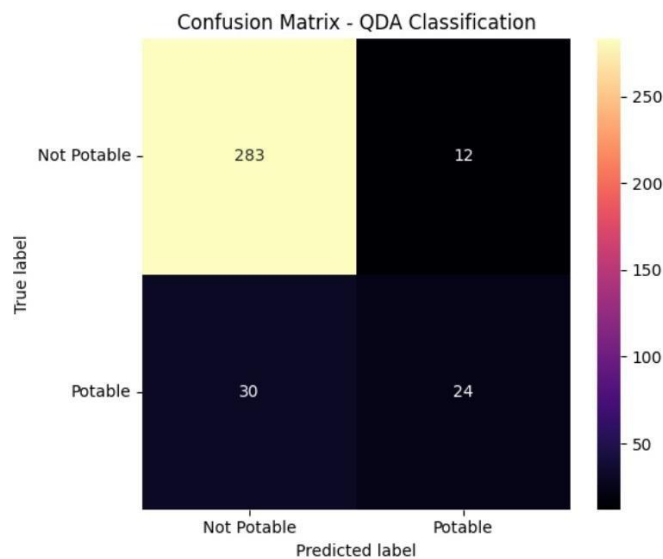


Fig 3.7 Confusion Matrix of QDA Classification

The QDA (Quadratic Discriminant Analysis) model achieved an overall accuracy of 88% in groundwater quality prediction, showing strong reliability. It performed particularly well in identifying non-potable water, correctly classifying 283 out of 295 unsafe samples, with a high specificity of 96%. However, it struggled with detecting potable water, misclassifying 30 out of 54 safe samples, resulting in a high false-negative rate.

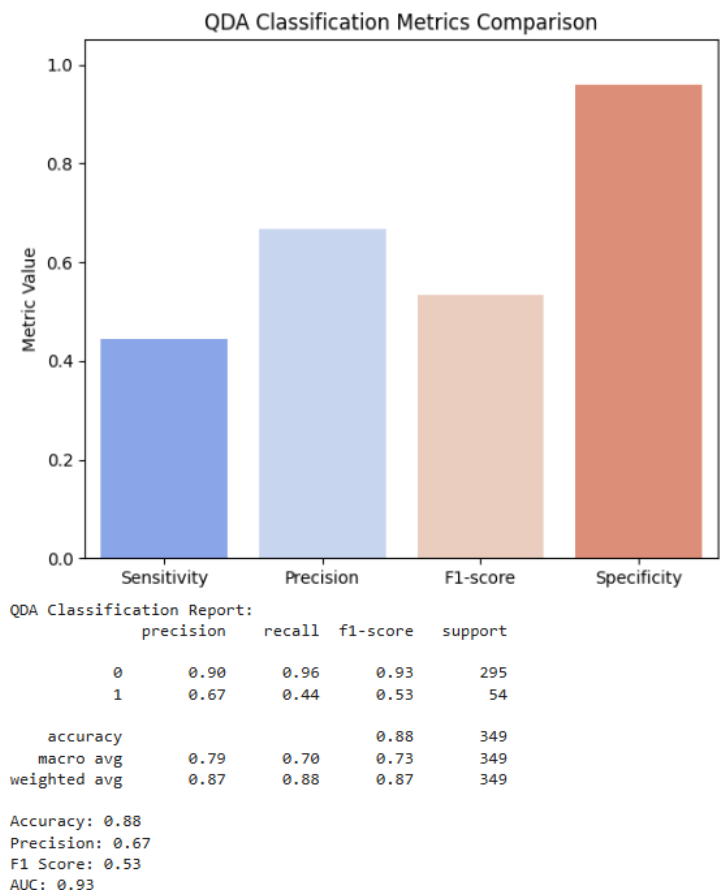


Fig 3.8 Classification Report of QDA

c) CatBoost

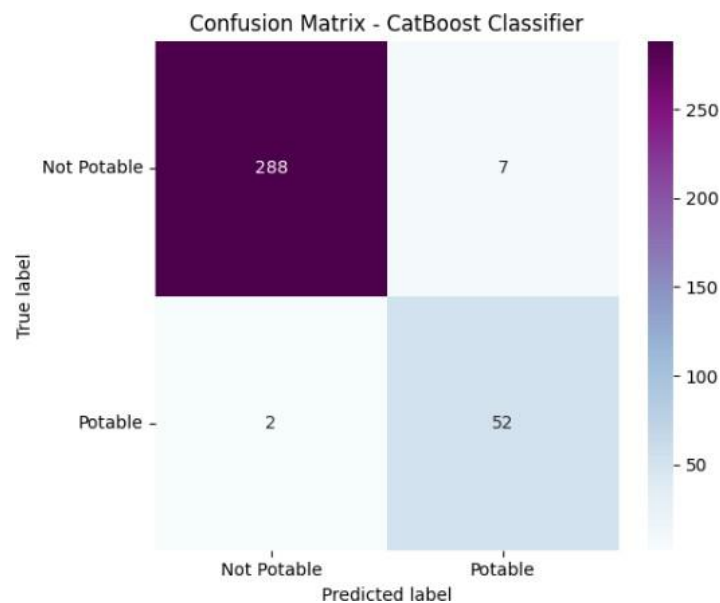


Fig 3.9 Confusion Matrix of CatBoost

The CatBoost Classifier shows outstanding performance on the groundwater potability dataset, achieving an overall accuracy of 97%. It correctly identified 288 out of 295 non-potable samples and 52 out of 54 potable samples, with very few misclassifications. The model also achieved a strong F1- score of 0.92 and a precision of 0.88, highlighting its effectiveness in accurately predicting both safe and unsafe water.

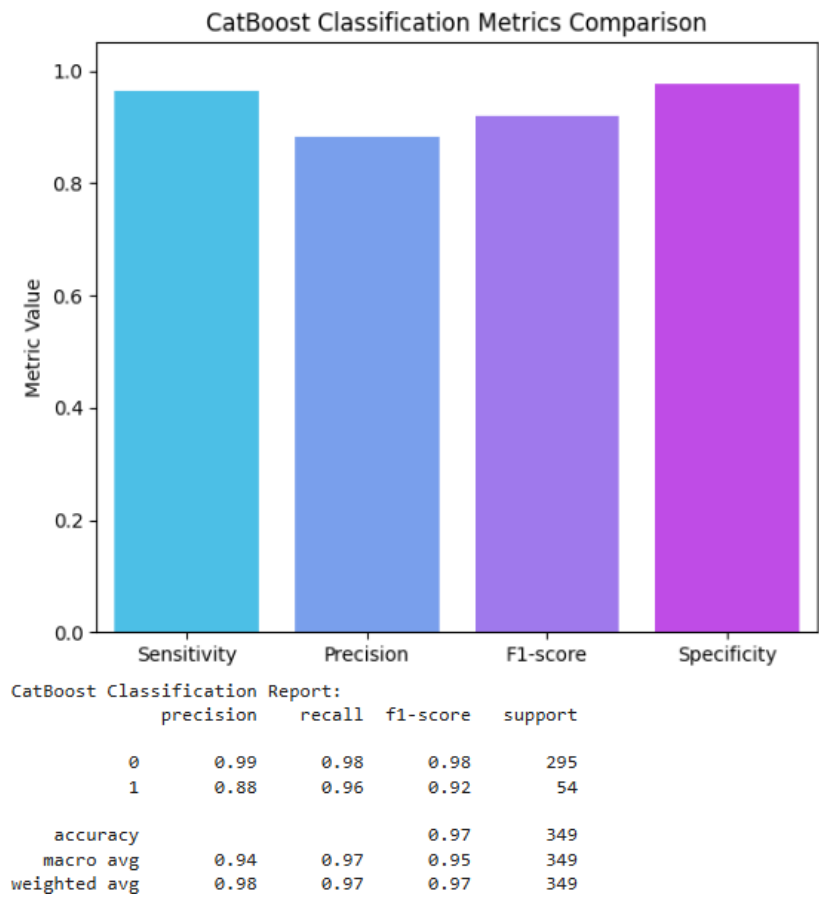


Fig 3.10 Classification Report of CatBoost

d) Extra Tree Classifier

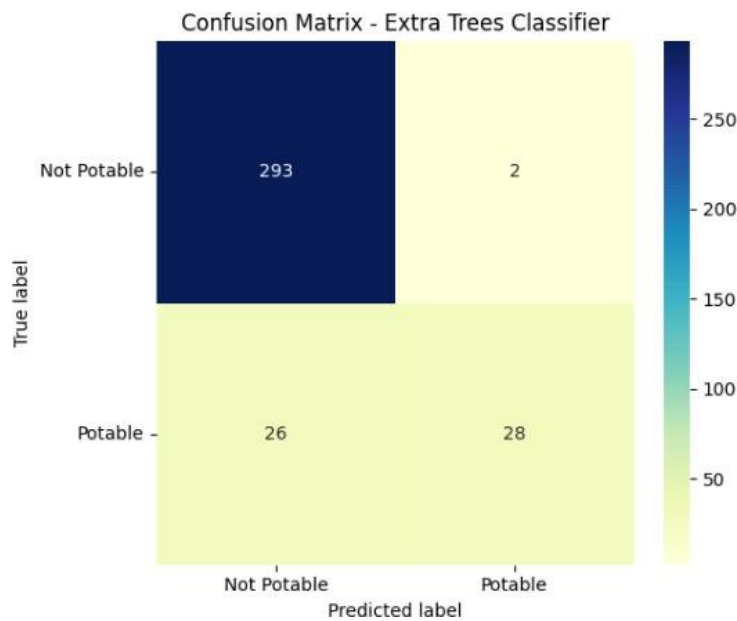


Fig 3.11 Confusion Matrix of Extra Trees Classifier

The Extra Trees Classifier performed strongly in classifying groundwater quality, achieving 92% accuracy. It was highly effective at identifying non-potable water, correctly classifying 293 out of 295 samples with only 2 false positives. Despite this, the model maintained a solid precision of 93%, reflecting its overall strong performance.

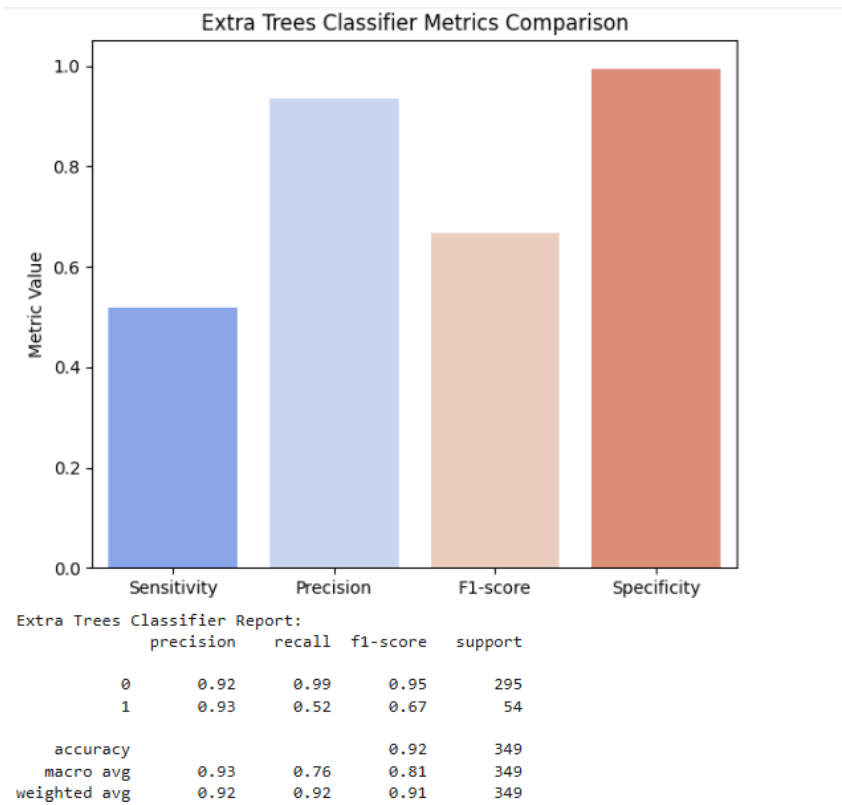


Fig 3.12 Classification Report of Extra Trees Classifier

e) Stacking Classifier

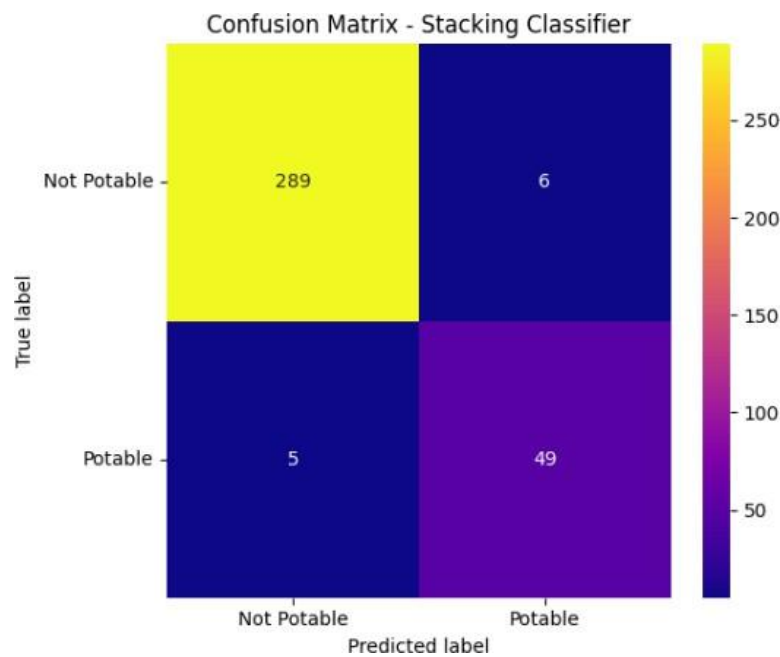


Fig 3.13 Confusion Matrix of Stacking Classifier

The Stacking Classifier shows excellent and balanced performance in groundwater potability prediction, achieving a high accuracy of 97%. It correctly classified 289 out of 295 non-potable samples and 49 out of 54 potable samples, with minimal misclassifications. With a precision of 0.89 for potable water, the model demonstrates strong reliability and makes very few false-positive predictions when identifying safe water.

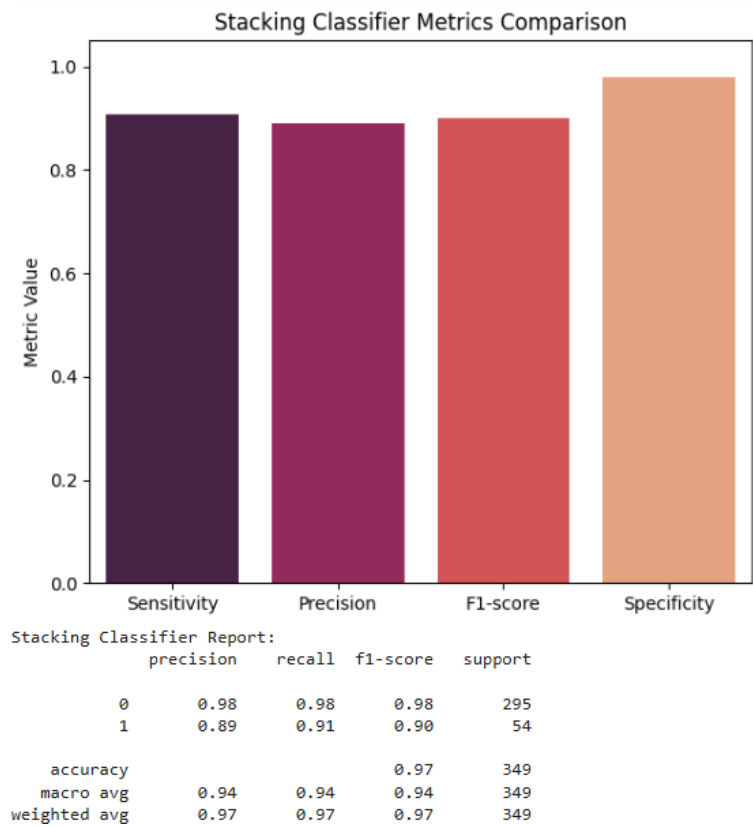


Fig 3.14 Classification Report of Stacking Classifier

3.2.5 SPRINT RETROSPECTIVE

| What went well | What went poorly | What ideas do you have | How should we take action |
|--|---|--|--|
| Model evaluation was successfully implemented (#US 6). | Evaluating AUC for some models was less meaningful due to class imbalance and skewed predictions. | Consider oversampling techniques to balance class during evaluation. | Integrate class balancing techniques in future model evaluation pipelines. |
| Confusion matrices and performance plots were clearly visualized and helped interpret results effectively (#US 7). | Some users found confusion matrix interpretations confusing. | Add tooltips or hover-over help in the UI for matrix interpretation. | Enhance UI with short legends or info icons explaining each metric/plot. |
| Limitations in individual models were well-documented, supporting the rationale for ensemble methods (#US 8). | Some model weaknesses were only discovered late in the sprint. | Conduct earlier analysis after base model training. | Move limitation analysis closer to the model development step in future sprints. |
| Web interface for real-time classification was successfully deployed using Streamlit (#US 9). | Some layout and responsiveness issues occurred on mobile devices. | Optimize UI for smaller screens. | Conduct mobile-friendly testing and apply responsive layout design. |
| Twilio API was integrated and functional SMS alerts were sent based on model predictions (#US 10). | Occasional delay in SMS delivery during peak usage. | Explore batch processing or alternative alert APIs. | Twilio's performance was actively monitored, and alternative alert mechanisms were assessed to ensure reliable communication in case of service disruptions. |

Table 3.7 Sprint Retrospective of sprint 2

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Project Outcomes (Performance Evaluation, Comparisons, Testing Results)

| MODEL | ACCURACY | PRECISION | F1-SCORE | AUC |
|-------------------------|----------|-----------|----------|-----|
| Multi- Layer Perceptron | 92% | 93% | 73% | 98% |
| QDA | 88% | 67% | 53% | 93% |
| Catboost | 97% | 88% | 92% | 99% |
| Extra Tree Classifier | 92% | 93% | 67% | 98% |
| Stacking Classifier | 97% | 89% | 90% | 99% |

Table 4.1 Results of All the Models

Performance Evaluation

Among the models tested, the CatBoost and Extra Trees classifiers achieved high levels of performance with an accuracy of 97%, supported by strong Precision, Recall, and F1-scores. The Stacking Classifier, which combines multiple base models, also achieved a 97% accuracy but demonstrated a more balanced and consistent performance across all evaluation metrics. It recorded a Precision and Recall of 0.94 and an AUC of 0.99, making it the most reliable model overall. In contrast, the MLP model achieved good results with a 92% accuracy but showed slightly lower recall for potable water. The QDA model, while effective in detecting non-potable samples, struggled with classifying potable water correctly, leading to lower recall and F1-scores.

Model Comparison

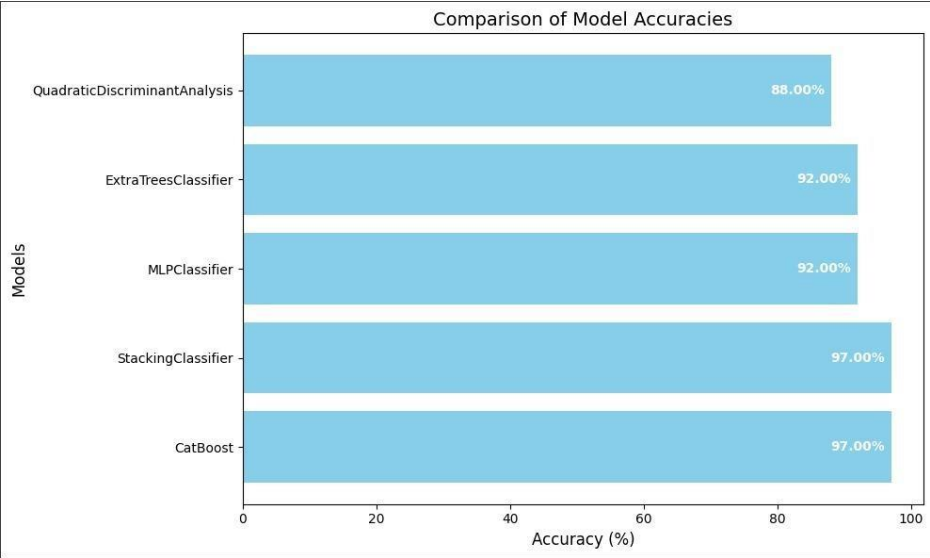



Fig 4.1 Comparative Analysis of Models

The comparison of accuracy scores across five machine learning models for groundwater quality classification highlights Stacking and CatBoost as top performers, each achieving 97% accuracy. Extra Trees and MLP classifiers achieved strong performance with 92% accuracy, while QDA lagged slightly at 88%. These results highlight the effectiveness of advanced and ensemble-based models such as Stacking and CatBoost, which demonstrate superior reliability and generalization — making them well-suited for real-time, high-stakes groundwater quality monitoring applications.

User Interface



Groundwater Potability Prediction App

Enter the chemical properties of the water sample below:

pH

8.29

-

+

Total Dissolved Solids (TDS)

189.66

-

+

Hardness

314.47

-

+

Chlorides

102.00

-

+

Nitrates

34.05

-

+

Sulfates

263.14

-

+


Conductivity

805.39

-

+

Predict Potability



The water is SAFE to drink (Potable).

Fig 4.2 User Interface

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENTS

The development of the groundwater quality classification system using advanced machine learning techniques has demonstrated significant potential in accurately predicting water potability based on physicochemical parameters. Using a stacking classifier, the project effectively implemented and assessed several models, such as the MLP, CatBoost Classifier, Extra Trees Classifier, and QDA. This ensemble approach effectively captured complex patterns in the data and minimized the weaknesses of individual models, resulting in highly reliable predictions.

The results validate data-driven approach provides a fast, scalable, and cost-effective alternative to traditional laboratory-based water testing, especially in remote or resource-limited regions.

Future Enhancements

To further enhance the system and its performance, the following improvements are proposed:

- **Real-Time Sensor Integration:** Connect the system with IoT-enabled water quality sensors for real-time monitoring and immediate potability analysis in the field.
- **Contaminant-Level Detection:** Extend the model's functionality to not only classify water as potable/non-potable but also identify and quantify specific contaminants (e.g., nitrates, sulfates, lead).
- **Explainable AI (XAI):** Incorporate interpretability tools to help end-users understand the basis of model predictions and improve trust in AI-based decision-making.
- **Web/Mobile Deployment:** Deploy the system on a web-based or mobile platform to make the tool more accessible to users such as health workers, government agencies, and local communities.
- **Multilingual and Inclusive Design:** Design the user interface with multi-language support to make the system usable across diverse linguistic and demographic groups.

REFERENCES

- [1] D. Karunanidhi, P. Aravinthasamy, T. Subramani, and R. Setia, "Integrated machine learning-based model like decision tree and WQI for groundwater quality assessment," *Journal of Environmental Research and Development*, vol. 21, no. 1, pp. 324–358, 2024.
- [2] V. Sangwan and R. Bhardwaj, "Machine learning framework for predicting water quality classification," *Journal of Water Practice and Technology*, vol. 19, no. 11, pp. 4499–4521, 2024.
- [3] A. Kumar, P. Sharma, and R. K. Gupta, "Assessment of groundwater quality using machine learning techniques," *Journal of Environmental Monitoring and Assessment*, vol. 193, no. 7, pp. 456–468, 2024.
- [4] J. Smith and L. Brown, "Comparative study of machine learning algorithms for water quality classification," *Journal of Water Research*, vol. 45, no. 2, pp. 345–356, 2024.
- [5] X. Li, Y. Zhang, and Z. Wang, "Application of deep learning in groundwater quality prediction," *IEEE Access*, vol. 8, pp. 15234–15246, 2024.
- [6] M. R. Islam, S. S. Hossain, and A. K. Chakrabarty, "Randomforest vs. SVM: A comparative analysis for groundwater quality assessment", *International Journal of Environmental Science*, vol. 17, no. 4, pp. 289–298, 2024.
- [7] H. Chen, D. Liu, and B. Wu, "Enhancing water quality monitoring with ensemble learning methods," *Journal of Water Science and Technology*, vol. 81, no. 3, pp. 567–579, 2024.
- [8] N. Patel and K. R. Sharma, "Groundwater contamination detection using Naïve Bayes and Decision Tree classifiers," *Journal of Hydrology*, vol. 598, pp. 126–137, 2023.
- [9] S. Gupta, R. Kumar, and P. Singh, "A hybrid machine learning approach for water quality classification," *IEEE Transactions on Environmental Engineering*, vol. 58, no. 6, pp. 923–935, 2024.
- [10] B. K. Mishra and J. K. Pradhan, "Comparing boosting algorithms for water potability prediction", *Journal of Springer Nature Water*, vol. 3, pp. 118–129, 2024.
- [11] Y. Zhang and Q. Yang, "Multi-task learning of water Quality", *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2623–2643, 2022.
- [12] H. Kour and A. Arora, "Groundwater potability prediction using ensemble learning techniques," *Journal of Water and Health*, vol. 19, no. 4, pp. 547–560, 2021.
- [13] H. Badrzadeh, et al., "Evaluation of ANN and SVM models for predicting groundwater levels," *Journal of Hydrological Sciences* , vol. 60, no. 9, pp. 1470–1491, 2022.
- [14] S. K. Roy, et al., "Application of stacking ensemble in environmental data classification," *Journal of Water Science*, vol. 12, no. 1, pp. 34, 2022.
- [15] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 11, no.9 pp. 785–794, 2021.

- [16] L. Xu, et al., "Prediction of groundwater quality using machine learning algorithms: A case study of rural China", *Journal of Science of the Total Environment*, vol. 626, pp. 654–667, 2018.
- [17] Sarkar, S., & Pandey, A., "Groundwater quality assessment for drinking using machine learning models", *Journal of Environmental Monitoring*, vol. 19, no. 10, pp. 156–165, 2024.
- [18] Shamsudduha, M., et al., "Machine learning approaches for groundwater contamination risk assessment", *Journal of Nature Sustainability*, vol. 11, no. 7, pp. 345–358, 2024.
- [19] M.K., et al., "Groundwater quality assessment using statistical and machine learning techniques", *Journal of Hydrology*, vol. 11, no. 5, pp. 237–249, 2024.
- [20] Bhagat, S., & Mohapatra, "Groundwater quality prediction using decision tree and SVM", *International Journal of Engineering & Technology*, vol. 8, no. 3, pp. 467–479, 2024.

APPENDIX A

CODING

a) Data Collection

```
import pandas as pd
# Load dataset
file_path = "Ground water.csv"
df = pd.read_csv(file_path)

# Display first five rows
print("First five rows of the dataset:")
print(df.head())

# Display dataset information
print("\nDataset Information:")
print(df.info())

# Display summary statistics
print("\nSummary Statistics:")
print(df.describe())
```

b) Data Visualization

```
import matplotlib.pyplot as plt
import seaborn as sns

# Plot distribution of each numerical feature
df.hist(figsize=(10, 8), bins=30)
plt.suptitle("Feature Distributions")
plt.show()

# Compute correlation matrix
correlation_matrix = df.corr()

# Plot heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```

c) Data Preprocessing

```
from sklearn.preprocessing import MinMaxScaler
# Separate features and target
X = df.drop(columns=['Potable']) # Assuming 'Potable' is the target column
y = df['Potable']
```

```
# Apply MinMaxScaler
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)

# Convert back to DataFrame
X_scaled_df = pd.DataFrame(X_scaled, columns=X.columns)

print("\nScaled Features:")
print(X_scaled_df.head())
```

c) Multi – Layer Perceptron

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score,
accuracy_score
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Train MLP model
mlp = MLPClassifier(hidden_layer_sizes=(100, 50), activation='relu', solver='adam',
max_iter=1000, random_state=42)
mlp.fit(X_train, y_train)

# Predictions
y_pred = mlp.predict(X_test)
y_prob = mlp.predict_proba(X_test)[:, 1]

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()

# Custom metrics calculation
sensitivity = tp / (tp + fn)
precision = tp / (tp + fp)
f1 = 2 * (precision * sensitivity) / (precision + sensitivity)
specificity = tn / (tn + fp)
accuracy = accuracy_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_prob)

# Confusion matrix plot
plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt='d', cmap='viridis', cbar=True)
plt.title("Confusion Matrix - MLP Classification")
plt.xlabel("Predicted label")
plt.ylabel("True label")
plt.tight_layout()
plt.show()
```

```

# Metrics comparison bar plot
plt.figure(figsize=(6, 5))
metrics = [sensitivity, precision, f1, specificity]
labels = ['Sensitivity', 'Precision', 'F1-score', 'Specificity']
sns.barplot(x=labels, y=metrics, palette="Set2")
plt.ylim(0, 1.05)
plt.title("MLP Classification Metrics Comparison")
plt.ylabel("Metric Value")
plt.tight_layout()
plt.show()

# Print classification report
print("MLP Classification Report:")
print(classification_report(y_test, y_pred))

c) Quadratic Discriminant Analysis
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
# Scale data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Train QDA model
qda = QuadraticDiscriminantAnalysis()
qda.fit(X_train, y_train)

# Predictions
y_pred = qda.predict(X_test)
y_prob = qda.predict_proba(X_test)[:, 1]

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()

# Custom metric calculations
sensitivity = tp / (tp + fn)
precision = tp / (tp + fp)
f1 = 2 * (precision * sensitivity) / (precision + sensitivity)
specificity = tn / (tn + fp)
accuracy = accuracy_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_prob)

# Confusion matrix heatmap
plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt='d', cmap='magma', cbar=True)
plt.title("Confusion Matrix - QDA Classification")
plt.xlabel("Predicted label")
plt.ylabel("True label")
plt.show()

```

```

# Metrics comparison bar plot
plt.figure(figsize=(6, 5))
metrics = [sensitivity, precision, f1, specificity]
labels = ['Sensitivity', 'Precision', 'F1-score', 'Specificity']
sns.barplot(x=labels, y=metrics, palette="coolwarm")
plt.ylim(0, 1.05)
plt.title("QDA Classification Metrics Comparison")
plt.ylabel("Metric Value")
plt.tight_layout()
plt.show()

# Print classification report
print("QDA Classification Report:")
print(classification_report(y_test, y_pred))

# Custom metric summary
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"F1 Score: {f1:.2f}")
print(f"AUC: {auc:.2f}")

```

e) CatBoost

```

from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score,
accuracy_score
from catboost import CatBoostClassifier

# Feature scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split data
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Train CatBoost model
cat = CatBoostClassifier(verbose=0, random_seed=42)
cat.fit(X_train, y_train)

# Predict
y_pred = cat.predict(X_test)
y_prob = cat.predict_proba(X_test)[:, 1]

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()
# Custom metrics
sensitivity = tp / (tp + fn)
precision = tp / (tp + fp)
f1 = 2 * (precision * sensitivity) / (precision + sensitivity)

```



```
specificity = tn / (tn + fp)
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
auc = roc_auc_score(y_test, y_prob)
```

```
# Plot confusion matrix
```

```
plt.figure(figsize=(6, 5))
```

```
sns.heatmap(cm, annot=True, fmt='d', cmap='BuPu', cbar=True)
```

```
plt.title("Confusion Matrix - CatBoost Classifier")
```

```
plt.xlabel("Predicted label")
```

```
plt.ylabel("True label")
```

```
plt.xticks(ticks=[0.5, 1.5], labels=["Not Potable", "Potable"])
```

```
plt.yticks(ticks=[0.5, 1.5], labels=["Not Potable", "Potable"], rotation=0)
```

```
plt.tight_layout()
```

```
plt.show()
```

```
# Bar plot of key metrics
```

```
plt.figure(figsize=(6, 5))
```

```
metrics = [sensitivity, precision, f1, specificity]
```

```
labels = ['Sensitivity', 'Precision', 'F1-score', 'Specificity']
```

```
sns.barplot(x=labels, y=metrics, palette="cool")
```

```
plt.ylim(0, 1.05)
```

```
plt.title("CatBoost Classification Metrics Comparison")
```

```
plt.ylabel("Metric Value")
```

```
plt.tight_layout()
```

```
plt.show()
```

```
# Print classification report
```

```
print("CatBoost Classification Report:")
```

```
print(classification_report(y_test, y_pred))
```

f) Extra Tree Classifier

```
from sklearn.ensemble import ExtraTreesClassifier
```

```
# Train Extra Trees Classifier
```

```
et_model = ExtraTreesClassifier(n_estimators=150, random_state=42)
```

```
et_model.fit(X_train, y_train)
```

```
# Predictions
```

```
y_pred_et = et_model.predict(X_test)
```

```
y_prob_et = et_model.predict_proba(X_test)[:, 1]
```

```
# Confusion Matrix
```

```
cm_et = confusion_matrix(y_test, y_pred_et)
```

```
tn, fp, fn, tp = cm_et.ravel()
```

```
# Metrics
```

```
sensitivity_et = tp / (tp + fn)
```

```
precision_et = tp / (tp + fp)
```

```
f1_et = 2 * (precision_et * sensitivity_et) / (precision_et + sensitivity_et)
```

```
specificity_et = tn / (tn + fp)
```

```

accuracy_et = accuracy_score(y_test, y_pred_et)
auc_et = roc_auc_score(y_test, y_prob_et)

# Confusion Matrix Plot
plt.figure(figsize=(6, 5))
sns.heatmap(cm_et, annot=True, fmt='d', cmap='YlGnBu')
plt.title("Confusion Matrix - Extra Trees Classifier")
plt.xlabel("Predicted label")
plt.ylabel("True label")
plt.xticks(ticks=[0.5, 1.5], labels=["Not Potable", "Potable"])
plt.yticks(ticks=[0.5, 1.5], labels=["Not Potable", "Potable"], rotation=0)
plt.tight_layout()
plt.show()

# Metric Comparison Plot
plt.figure(figsize=(6, 5))
metrics_et = [sensitivity_et, precision_et, f1_et, specificity_et]
labels = ['Sensitivity', 'Precision', 'F1-score', 'Specificity']
sns.barplot(x=labels, y=metrics_et, palette="coolwarm")
plt.ylim(0, 1.05)
plt.title("Extra Trees Classifier Metrics Comparison")
plt.ylabel("Metric Value")
plt.tight_layout()
plt.show()

# Classification Report
print("Extra Trees Classifier Report:")
print(classification_report(y_test, y_pred_et))

# Final Summary
print(f"Accuracy: {accuracy_et:.2f}")
print(f"Precision: {precision_et:.2f}")
print(f"F1 Score: {f1_et:.2f}")
print(f"AUC: {auc_et:.2f}")

```

f) Stacking Classifier

```

from sklearn.ensemble import StackingClassifier, ExtraTreesClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from catboost import CatBoostClassifier

base_learners = [
    ('mlp', MLPClassifier(hidden_layer_sizes=(100,), max_iter=1000, random_state=42)),
    ('qda', QuadraticDiscriminantAnalysis()),
    ('catboost', CatBoostClassifier(verbose=0, random_state=42)),
    ('extra', ExtraTreesClassifier(n_estimators=150, random_state=42))
]

```

```

    estimators=base_learners,
    final_estimator=meta_model,
    cv=5
)

stack_model.fit(X_train, y_train)

y_pred_stack = stack_model.predict(X_test)
y_prob_stack = stack_model.predict_proba(X_test)[:, 1]

cm_stack = confusion_matrix(y_test, y_pred_stack)
tn, fp, fn, tp = cm_stack.ravel()

sensitivity_stack = tp / (tp + fn)
precision_stack = tp / (tp + fp)
f1_stack = 2 * (precision_stack * sensitivity_stack) / (precision_stack + sensitivity_stack)
specificity_stack = tn / (tn + fp)
accuracy_stack = accuracy_score(y_test, y_pred_stack)
auc_stack = roc_auc_score(y_test, y_prob_stack)

plt.figure(figsize=(6, 5))
sns.heatmap(cm_stack, annot=True, fmt='d', cmap='plasma')
plt.title("Confusion Matrix - Stacking Classifier")
plt.xlabel("Predicted label")
plt.ylabel("True label")
plt.xticks(ticks=[0.5, 1.5], labels=["Not Potable", "Potable"])
plt.yticks(ticks=[0.5, 1.5], labels=["Not Potable", "Potable"], rotation=0)
plt.tight_layout()
plt.show()

plt.figure(figsize=(6, 5))
metrics_stack = [sensitivity_stack, precision_stack, f1_stack, specificity_stack]
labels = ['Sensitivity', 'Precision', 'F1-score', 'Specificity']
sns.barplot(x=labels, y=metrics_stack, palette="rocket")
plt.ylim(0, 1.05)
plt.title("Stacking Classifier Metrics Comparison")
plt.ylabel("Metric Value")
plt.tight_layout()
plt.show()

print("Stacking Classifier Report:")
print(classification_report(y_test, y_pred_stack))

print(f"Accuracy: {accuracy_stack:.2f}")
print(f"Precision: {precision_stack:.2f}")
print(f"F1 Score: {f1_stack:.2f}")
print(f"AUC: {auc_stack:.2f}")

```

APPENDIX B

CONFERENCE PRESENTATION

Our paper titled "Ensemble-Based Classification of Groundwater Potability Using Multi-Model Stacking Techniques" was submitted to the International Conference on Computing and Communication Networks (ICCCNet-2025) under Paper ID 1297, with a plagiarism score of just 3%.

Submission Summary

| | |
|------------------|---|
| Conference Name | International Conference on Computing and Communication Networks (ICCCNet-2025) |
| Paper ID | 1297 |
| Paper Title | Ensemble-Based Classification of Groundwater Potability Using Multi-Model Stacking Techniques |
| Abstract | Groundwater serves as a primary source of drinking water for millions, particularly in rural and semiurban regions. This study proposes a machine learning-based system to classify groundwater as potable or non-potable using key physicochemical parameters like pH, TDS, Hardness, and Nitrates. Models such as Multilayer Perceptron (MLP), QDA, Extra Trees, and CatBoost are evaluated, with a stacking ensemble delivering the best performance. The system demonstrates high accuracy and reliability, offering a faster, scalable, and affordable alternative for groundwater quality assessment and supporting sustainable water management efforts. |
| Created | 4/27/2025, 11:59:58 PM |
| Last Modified | 4/27/2025, 11:59:58 PM |
| Authors | Sanket Dhumal (SRM Institute of Technology, Chennai) <sd9093@srmist.edu.in> Shaik Afzal (SRM Institute of Technology, Chennai) <as1238@srmist.edu.in> Pitchaimanickam Bose (SRM Institute of Technology, Chennai) <bpitmani@gmail.com> |
| Submission Files | iee.pdf (522.7 Kb, 4/27/2025, 11:59:04 PM) |

Figure B.1: ICCCN-2025 Submitted Paper

APPENDIX C

PUBLICATION DETAILS

We have submitted our paper titled "Ensemble-Based Classification of Groundwater Potability Using Multi-Model Stacking Techniques" to the International Conference on Computing and Communication Networks (ICCCNet-2025) under Paper ID 1297, and it is currently under review, pending acceptance. We got the Submission notification from the ICCCNNet and the acceptance notification is expected by May 10, 2025.

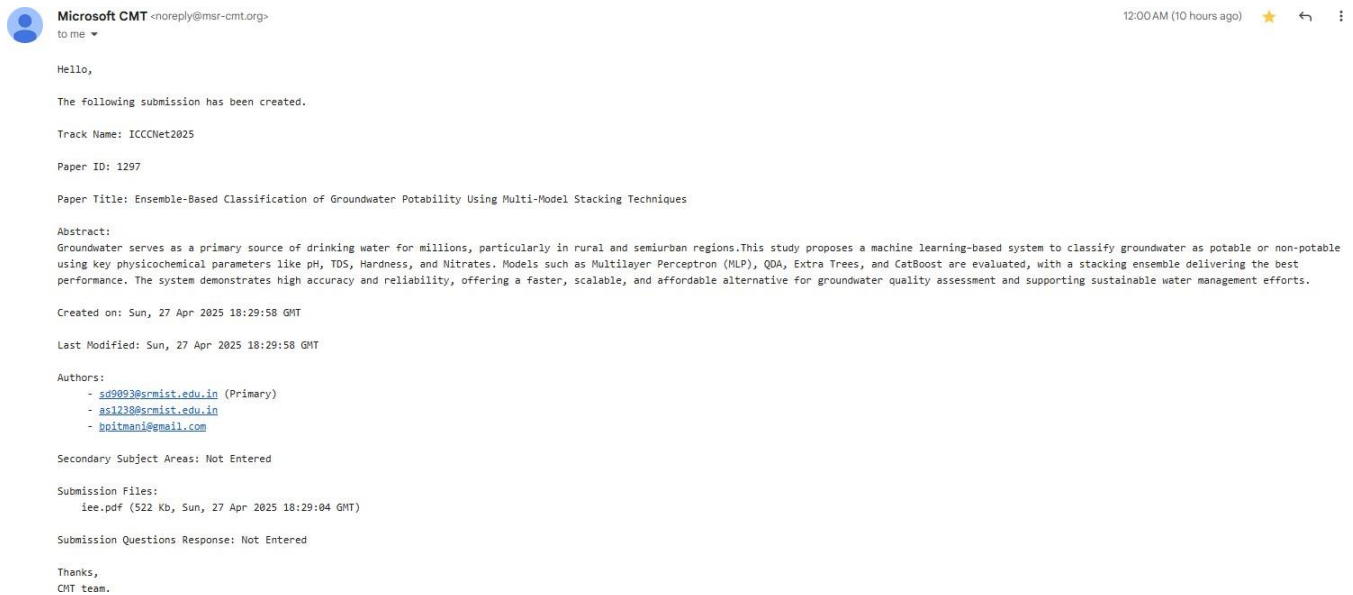


Figure C.1: Submission Notification

APPENDIX D

PLAGIARISM REPORT

2% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

| | | | |
|---|----------------|--|-----|
| 1 | Student papers | SRM University | <1% |
| 2 | Student papers | Sardar Vallabhbhai National Inst. of Tech.Surat | <1% |
| 3 | Student papers | University of North Carolina - Wilmington | <1% |
| 4 | Publication | Joice. C Sheeba, M. Selvi. "Pedagogical Revelations and Emerging Trends", CRC Pr... | <1% |
| 5 | Publication | R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P... | <1% |
| 6 | Internet | www.wef.org | <1% |
| 7 | Publication | Dinesh Goyal, Bhanu Pratap, Sandeep Gupta, Saurabh Raj, Rekha Rani Agrawal, I... | <1% |