

A MINOR PROJECT REPORT ON  
**WATER QUALITY CLASSIFICATION USING MACHINE  
LEARNING**

A dissertation submitted in partial fulfillment of the  
Requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**  
in  
**INFORMATION TECHNOLOGY**

*Submitted by*

**M. Akash (20B81A1264)**  
**Y. Lahari Priya (20B81A1279)**  
**S. Murari (20B81A12A3)**

*Under the esteemed guidance of*

**G.Sunitha Rekha**

Sr.Assistant Professor, IT Department  
CVR College of Engineering



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CVR COLLEGE OF ENGINEERING**

ACCREDITED BY NBA, AICTE & Affiliated to JNTU-H

Vastunagar, Mangalpally (V), Ibrahimpatnam (M), R.R. District, PIN-501 510  
2023-2024



Cherabuddi Education Society's  
**CVR COLLEGE OF ENGINEERING**

(An Autonomous Institution)

**ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, AICTE**

(Approved by AICTE & Govt. of Telangana and Affiliated to JNT University)

**Vastunagar**, Mangalpalli (V), Ibrahimpatan (M), R.R. District, PIN - 501 510

Web : <http://cvr.ac.in>, email : [info@cvr.ac.in](mailto:info@cvr.ac.in)

Ph : 08414 - 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

---

2023-2024

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CERTIFICATE**

This is to certify that the Project Report entitled “**Water Quality Classification Using Machine Learning**” is a bonafide work done and submitted by **M. Akash (20B81A1264)**, **Y. Lahari Priya (20B81A1279)**, **S. Murari (20B81A12A3)** during the academic year 2023-2024, in partial fulfilment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to my best of the knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

**INTERNAL GUIDE**

**G.Sunitha Rekha**

Sr.Assistant Professor, IT Department

**HEAD OF THE DEPARTMENT**

**Dr. Bipin Bihari Jayasingh**

Professor, IT Department

**PROJECT COORDINATOR**

**G.Sunitha Rekha**

Sr.Assistant Professor, IT Department

**EXTERNAL EXAMINER**

---

City Office : # 201 & 202, Ashoka Scintilla, Opp. KFC, Himayatnagar, Hyderabad - 500 029, Telangana.

Phone : 040 - 42204001, 42204002, 9391000791, 9177887273

# DECLARATION

We hereby declare that the project report entitled “**Mobile Application for Elective Selection**” is an original work done and submitted to IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment of the requirement for the award of Bachelor of Technology in **Information Technology** and it is a record of bonafide project work carried out by us under the guidance of **G.Sunitha Rekha, Sr. Assistant Professor, Department of Information Technology**.

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

---

Signature of the Student

(M. Akash)  
(20B81A1264)

---

Signature of the Student

(Y. Lahari Priya)  
(20B81A1279)

---

Signature of the Student

(S. Murari)  
(20B81A12A3)

## ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project. First and Foremost, We thank the Chairman, Principal, Vice Principal for availing infrastructural facilities to complete the major project in time. We offer our sincere gratitude to our internal guide **G.Sunitha Rekha**, Sr. Assistant Professor, IT Department, CVR College of Engineering for her immense support, timely co-operation and valuable advice throughout the course of our project work. We would like to thank the Professor In-Charge of Projects, **Dr. Seetharamaiah**, Professor, Information Technology for his valuable suggestions in implementing the project. We would like to thank the Head of Department, Professor **Dr. Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work. We are thankful to **G.Sunitha Rekha**, Project Coordinator, Sr. Assistant Professor, IT Department, CVR College of Engineering for her supportive guidelines and for having provided the necessary help for carrying forward this project without any obstacles and hindrances. We also thank the **Project Review Committee Members** for their valuable suggestions.

## ABSTRACT

Water, as a resource essential for sustaining life, requires high-quality standards to ensure its usability. Categorizing water quality using machine learning techniques has gained significant attention due to its potential to revolutionize water resource management. This paper delves into the different stages involved in the process, encompassing data collection, preprocessing, feature selection, and model training. Decision trees, support vector machines, and artificial neural networks have emerged as prominent algorithms for water quality categorization. Detecting water contamination early and managing it effectively through machine learning approaches can have profound implications for the sustainable management of water resources.

The process of categorizing water quality using machine learning begins with data collection. Various methods are employed to gather relevant data from diverse sources such as water monitoring stations, environmental sensors, and satellite imagery. This data is then subjected to preprocessing techniques, including data cleaning, outlier detection, and missing value imputation, to ensure the integrity and reliability of the dataset.

Feature selection is a critical step in water quality analysis, as it aims to identify the most relevant attributes that contribute to water quality. Techniques such as correlation analysis, feature importance ranking, and dimensionality reduction algorithms are utilized to extract the most informative features from the dataset. This helps to reduce computational complexity and enhance the model's performance.

Model training involves applying machine learning algorithms to the preprocessed dataset. Decision trees, a popular algorithm, create a hierarchical structure of rules based on the features and target variables. They provide interpretable models and can handle both categorical and continuous variables. Support vector machines (SVM) use a hyperplane to separate different water quality classes, maximizing the margin between them. SVMs are effective in handling complex datasets with high dimensionality. neural networks (NN) simulate the behavior of the human brain and are capable of capturing complex relationships in water quality data. NNs can learn from large datasets and provide accurate predictions.

Implementing machine learning techniques for water quality categorization enables early detection of water contamination, leading to timely actions and more effective management strategies. Real-time monitoring and analysis of water quality parameters can help identify potential threats to water resources, such as pollutant spills or harmful algal blooms, allowing for swift responses to mitigate risks. Moreover, machine learning models assist in decision-making processes by providing valuable insights for resource allocation and intervention strategies.

While the adoption of machine learning techniques in water resource management brings numerous advantages, it also presents challenges. Ensuring the quality and reliability of input data is crucial for obtaining accurate predictions. Interpretability of models is another concern, as transparent decision-making is essential in water resource management. Scalability and compatibility with real-time data sources and sensor networks need to be addressed to enable efficient implementation of machine learning approaches.

## List of Tables

Table no	Title	Page no
4.1	Dataset of “Water Quality Classification Using Machine Learning”	28
5.1	Different fields in dataset	29
5.2	Wrong values in dataset	31
5.3	Big dataset	33
5.4	Missing value in dataset	35

## List of figures

Figure no	Title	Page no
2.1	Software Architectural Diagram to represent the high-level structure and components of a software system.	15
3.1	Use Case Diagram to represent interactions between users (actors) and a system.	16
3.2	Class Diagram to represent the static structure of a system, including classes, their attributes, and relationships.	17
3.3	Activity Diagram to represent the flow of activities or processes within a system.	19
3.4	State Diagram to represent the different states and transitions of an object or system in response to events.	25
3.5	Sequence Diagram to represent the interactions between objects or components over a specific time sequence.	21
3.6	Component Diagram to represent the physical or logical components of a system and their dependencies.	24



## Table of contents

1. INTRODUCTION.....	6
1.1 Literature Survey .....	9
2. SOFTWARE REQUIREMENT SPECIFICATIONS .....	14
3. DESIGN .....	16
4. IMPLEMENTATION .....	25
5.TESTING .....	30
CONCLUSION .....	38
FUTURE ENHANCEMENTS .....	39
REFERENCES .....	40
Appendix A - Abbreviations .....	41
Appendix B - Software Installation Procedure.....	42
Appendix C - Software Usage Process.....	46

# 1. INTRODUCTION

Water is a fundamental resource that sustains all forms of life on Earth. Access to clean and safe water is essential for human health, agricultural productivity, and ecosystem integrity. However, the quality of water can be significantly impacted by both natural and anthropogenic factors, resulting in potential risks to human health and the environment. Factors such as pollution, weather fluctuations, and human activities pose significant challenges to maintaining adequate water quality standards.

Water quality monitoring is a vital practice aimed at assessing the physical, chemical, and biological characteristics of water bodies to evaluate their suitability for various uses. It involves the systematic collection, analysis, and interpretation of water samples to determine the presence of contaminants, nutrient levels, oxygen content, acidity, temperature, and other relevant parameters. By monitoring water quality, potential risks and sources of contamination can be identified, facilitating appropriate management strategies.

Monitoring water quality is crucial for several reasons. Firstly, it helps safeguard human health by identifying potential health hazards associated with contaminated water sources. Harmful substances, including pathogens, toxic chemicals, and heavy metals, can have adverse effects on human well-being, causing waterborne diseases and long-term health issues. Regular monitoring allows for the detection of such contaminants, ensuring that water supplies are safe for consumption and minimizing health risks.

Machine learning, a subset of artificial intelligence, offers promising solutions for improving water quality monitoring. It involves the development of algorithms that can learn patterns and relationships from data, enabling the identification of complex patterns that may not be apparent using traditional methods. By training machine learning models on historical water quality data, they can recognize and predict patterns, classify water samples, and provide valuable insights for decision-making.

Machine learning algorithms can analyze vast datasets consisting of various water quality parameters, including physical, chemical, and biological variables. By exploring the interactions between these parameters, machine learning models can detect patterns indicative of specific

water quality conditions or contamination events. The integration of machine learning with sensor technologies allows for real-time monitoring, providing continuous data streams and enabling rapid detection and response to changes in water quality.

Different machine learning algorithms, such as decision trees, support vector machines (SVM), random forests, and artificial neural networks (ANN), can be employed for water quality monitoring. Decision trees provide interpretable models that can capture complex interactions between variables. SVMs are particularly effective in handling high-dimensional datasets and can accurately classify water samples into different quality categories. Random forests combine multiple decision trees to improve accuracy and robustness, while ANNs simulate the behavior of the human brain and excel at capturing nonlinear relationships and complex patterns.

The primary objective of this project is to develop a system that utilizes machine learning algorithms for the classification of water samples based on their quality. The system aims to improve the efficiency and accuracy of water quality monitoring by automating the classification process, enabling real-time analysis, and providing timely insights. The system will be designed to handle large volumes of data, integrate multiple water quality parameters, and adapt to changing environmental conditions.

The methodology involves several key steps. Comprehensive water quality datasets will be collected from various sources, including monitoring stations, remote sensing platforms, and citizen science initiatives. These datasets will encompass physical, chemical, and biological parameters relevant to water quality assessment. Preprocessing techniques will be applied to ensure data integrity, including data cleaning, outlier detection, and missing value imputation.

Feature selection methods will be employed to identify the most informative variables that contribute to water quality classification. Correlation analysis, feature importance ranking, and dimensionality reduction techniques, such as principal component analysis, will be used to select relevant features. Additionally, new features may be engineered by combining or transforming existing variables to capture important patterns and interactions.

However, the integration of machine learning techniques into water quality monitoring also presents challenges and considerations. Data availability and quality are crucial for training accurate machine learning models. Ensuring data consistency, addressing missing values, and validating data quality are essential steps in overcoming data challenges. Model interpretability and explainability are important for building trust and acceptance among stakeholders. Efforts should be made to develop interpretable machine learning techniques and tools that provide insights into model decision-making processes. Integration with existing monitoring systems requires compatibility, data standardization, and interoperability considerations.

Water quality monitoring plays a vital role in ensuring the availability of safe water resources for human use and maintaining the health of ecosystems. This project aims to develop a machine learning-based classification system that can effectively and accurately classify water samples based on their quality. By leveraging the power of machine learning algorithms, the system has the potential to enhance water quality monitoring efficiency, provide valuable decision support, and enable proactive water resource management. Ultimately, this project contributes to safeguarding human health, preserving the environment, and ensuring the sustainable management of water resources.

## 1.1 Literature Survey

Bora and Kapoor's Seminal Exploration - A Deeper Dive:

Bora and Kapoor's review paper is a profound journey through the captivating and intricate landscape of water quality classification, elevated by the potent capabilities of machine learning. Their extensive exploration spans a vast spectrum of machine learning algorithms, stretching from classical methodologies that have laid the bedrock of the field to the avant-garde, cutting-edge techniques that are pushing the boundaries of what's attainable.

In this extensive voyage, the authors meticulously underline the pivotal role of two critical components—feature selection and data preprocessing—revealing their monumental significance as cornerstones shaping the accuracy and robustness of water quality classification models. These components are not just mentioned in passing but are thoroughly dissected, providing readers with a comprehensive understanding of their role in building effective machine learning models for water quality assessment.

Moreover, Bora and Kapoor go beyond merely presenting a catalog of machine learning methods; they offer a masterclass in understanding and applying these techniques. Their dedication to demystifying the complexities of machine learning is evident throughout the paper, empowering readers, whether seasoned experts or newcomers, to not only grasp the algorithms but also to make informed choices when addressing real-world water quality challenges.

Among the most significant contributions of this paper is the emphasis placed on the fundamental importance of feature selection and data preprocessing. These often overlooked elements form the backbone upon which successful machine learning models are built. By spotlighting these critical aspects, Bora and Kapoor provide not just a well-lit path for researchers to follow but also practical insights into how to implement these processes effectively in the context of water quality assessment.

However, this review is not solely a theoretical exercise; it is a practical guide, replete with actionable insights and solutions to the challenges that researchers and practitioners often encounter in the quest for precise water quality classification. By unraveling the intricacies of machine learning and underscoring the significance of data preparation, Bora and Kapoor

empower the water quality assessment community to navigate the complex terrain of environmental monitoring with confidence and precision, ultimately contributing to a more sustainable and enlightened approach to safeguarding our precious water resources.

Chen, Wu, and Zhang's IoT-Driven Symphony - Navigating the Confluence:

Chen, Wu, and Zhang's study represents an extraordinary convergence of two technological frontiers: the Internet of Things (IoT) and machine learning. Their meticulous exploration of this intersection offers a tantalizing glimpse into a future where real-time sensor data plays a central role in environmental assessment.

Within this symphony of innovation, the authors showcase the transformative potential of IoT-driven machine learning. By seamlessly integrating sensor data into the machine learning framework, they orchestrate a harmonious interplay that has the potential to revolutionize environmental monitoring. The ability to continuously collect and analyze real-time data allows for a level of responsiveness that was previously unimaginable, enabling timely interventions to maintain water quality.

However, this transformative journey is not without its challenges, and the authors courageously confront them. The complexities of data integration, the need for algorithmic robustness, and the intricacies of sensor calibration are all addressed with candor. By acknowledging these hurdles, Chen, Wu, and Zhang not only highlight the potential pitfalls but also pave the way for innovative solutions that hold the key to unlocking the full potential of this technological convergence.

Their study serves as an inspiration to researchers and technology enthusiasts alike, pointing the way toward a future where data-rich environments empower us to comprehend and manage water quality in ways that were once relegated to the realm of science fiction. It is a testament to the power of innovation and a harbinger of the exciting possibilities that lie ahead in the realm of environmental monitoring.

Vazquez-Corral, Giro-I-Nieto, and Torres-Torriti's Deep Learning Odyssey - Unveiling the Depths:

Vazquez-Corral, Giro-I-Nieto, and Torres-Torriti embark on a captivating odyssey through the world of deep learning in the context of water quality monitoring. Their review paper sheds light

on the intricate architectures of convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs), offering a profound understanding of these complex models.

But this journey extends beyond theory. The authors delve into the practical challenges associated with deploying deep learning in real-world environmental assessments. The scarcity of data, the need for model interpretability, and the challenges of scalability are all hurdles that must be overcome. By acknowledging these difficulties, Vazquez-Corral, Giro-I-Nieto, and Torres-Torriti set the stage for a more nuanced and realistic approach to leveraging deep learning in water quality classification.

Their review paper is a compass guiding researchers toward a future where cutting-edge technology and environmental science intersect to reshape our understanding of aquatic ecosystems. It ignites a spark of innovation and beckons us toward a more enlightened and empowered approach to water quality monitoring.

Adhikary and Chakraborty's Foresight Expedition - Predictive Pathways:

Adhikary and Chakraborty's forward-looking review paper embarks on a transformative expedition, charting the course towards a future where water quality management is no longer a reactive endeavor but a proactive and anticipatory pursuit. Their visionary journey is steeped in the realm of predictive modeling—a realm where data-driven insights enable us to anticipate and mitigate environmental challenges.

In their meticulous evaluation of machine learning algorithms, they illuminate a path where predictive prowess transforms snapshots of water quality into a dynamic narrative of change. But their exploration extends beyond algorithms; it encompasses validation methodologies, uncertainty quantification techniques, and the temporal dynamics of dynamic systems. By fearlessly confronting these complexities, the authors lay the foundation for a future of proactive water quality management.

Their review is not a mere academic exercise; it is a practical guide—a compass empowering researchers and practitioners to navigate the terrain of predictive modeling with precision and

vision. It beckons us toward a future where data-driven foresight enables us not just to react to environmental challenges but to anticipate and mitigate them, ensuring a more sustainable and resilient approach to water resource management.

#### Shi, Han, and Ma's Systematic Toolbox - Crafting the Framework:

Shi, Han, and Ma's systematic review paper embarks on a meticulously structured voyage, traversing the expansive landscape of machine learning-based water quality classification. This methodical exploration goes beyond being a simple survey; it is a symphony of categorization and evaluation, a meticulous dissection of algorithms that forms the foundation of a comprehensive toolbox for researchers.

As the authors categorize algorithms and meticulously evaluate their performance, they unveil a panoramic view of the current state of water quality classification. This symphony of evaluation is a testament to both the progress achieved and the potential yet to be harnessed. However, the review does not rest with a retrospective glance; it casts a visionary gaze forward, identifying the nascent trajectories that will define the future of the field. This synthesis of past accomplishments and future possibilities weaves a tapestry that guides the evolution of water quality classification, ensuring its continued relevance and effectiveness in an ever-evolving environmental landscape.

#### Gupta and Sharma's Transformational Survey - Navigating Challenges:

Gupta and Sharma's review paper is a compass that guides us through the dynamic landscape of machine learning approaches for water quality classification, offering not just insights into accomplishments but a resolute exploration of limitations and innovative solutions. This exhaustive survey traverses a spectrum of techniques, encapsulating the progress achieved in integrating machine learning into the fabric of water quality assessment.

The review does not merely scrutinize algorithms; it embarks on a profound exploration of the challenges that accompany such transformative integration. Data quality, model complexity, interpretability—these are not mere stumbling blocks but catalysts for innovation. The authors, in their pursuit of progress, delve into potential solutions that hold the promise of refining the accuracy, efficiency, and efficacy of water quality management.



In sum, these visionary explorations and meticulous reviews collectively form a tapestry of knowledge, guiding us toward a future where water quality assessment and management are elevated to new heights through the transformative power of machine learning and technological innovation. This collaborative journey promises a more sustainable and enlightened approach to safeguarding our precious water resources.

## 2. SOFTWARE REQUIREMENT SPECIFICATIONS

### 2.1 Functional Requirements:

- Collect water quality data from sensors, manual measurements, and external databases.
- Preprocess data through cleaning, filtering, and normalization.
- Train and test classification models (PCA, NNs, SVMs) on preprocessed data.
- Evaluate model performance based on accuracy, precision, recall, and F1 score.
- Visualize results using graphs, charts, and tables.

### 2.2 Non-Functional Requirements:

- User-friendly interface with clear instructions.
- Reliable and robust with minimal errors, bugs, and crashes.
- Efficient with reasonable response times and processing speeds.
- Secure and protect sensitive data from unauthorized access and manipulation.
- Compatible with Windows, macOS, and Linux operating systems.

### 2.3 Hardware Requirements and Software Requirements

#### Hardware Requirements :

- **CPU:** A modern multi-core CPU is essential for efficient model training.
- **GPU:** GPUs, especially NVIDIA GPUs, can greatly accelerate training times for complex models.
- **RAM:** 16 GB or more is recommended for basic tasks; 32 GB or more for larger projects.
- **Storage:** Solid State Drives (SSDs) are preferable for fast data access and storage.
- **VRAM:** GPUs with at least 4 GB or 8 GB of VRAM are common choices.
- **Internet Connectivity:** Reliable internet access is needed for data downloads and cloud services.

## Software Requirements:

- **Python/R:** For analysis and coding.
- **Jupyter Notebook:** Interactive coding environment.
- **scikit-learn:** Classification algorithms.
- **Matplotlib/Seaborn:** Visualization.
- **Data sources:** Water quality datasets.
- **Preprocessing:** Cleaning, handling missing data.
- **Machine learning:** Decision trees, neural networks, etc.
- **Evaluation:** Metrics like precision, recall.
- **Git:** Version control.
- **Documentation:** Explain methods and results

## 2.4 Software Architecture

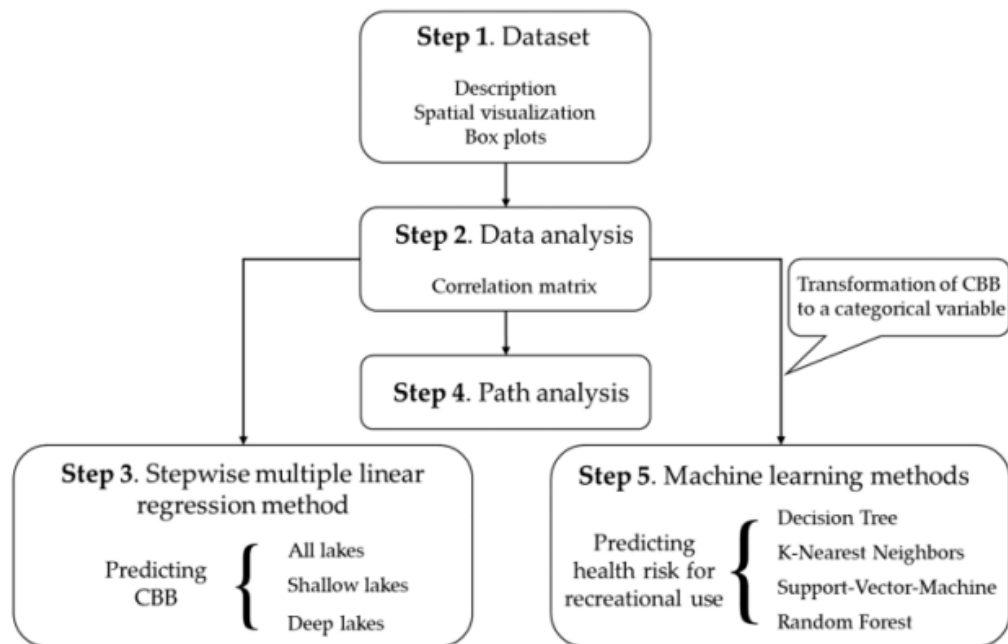


Fig 2.1: Software Architectural Diagram to represent the high-level structure and components of a software system.

### 3. DESIGN

#### 3.1 Use Case Diagram:

Use case diagrams are used to represent the actors and their actions or functions between a system. They help to identify the different actors involved and the various use cases that the system can perform.

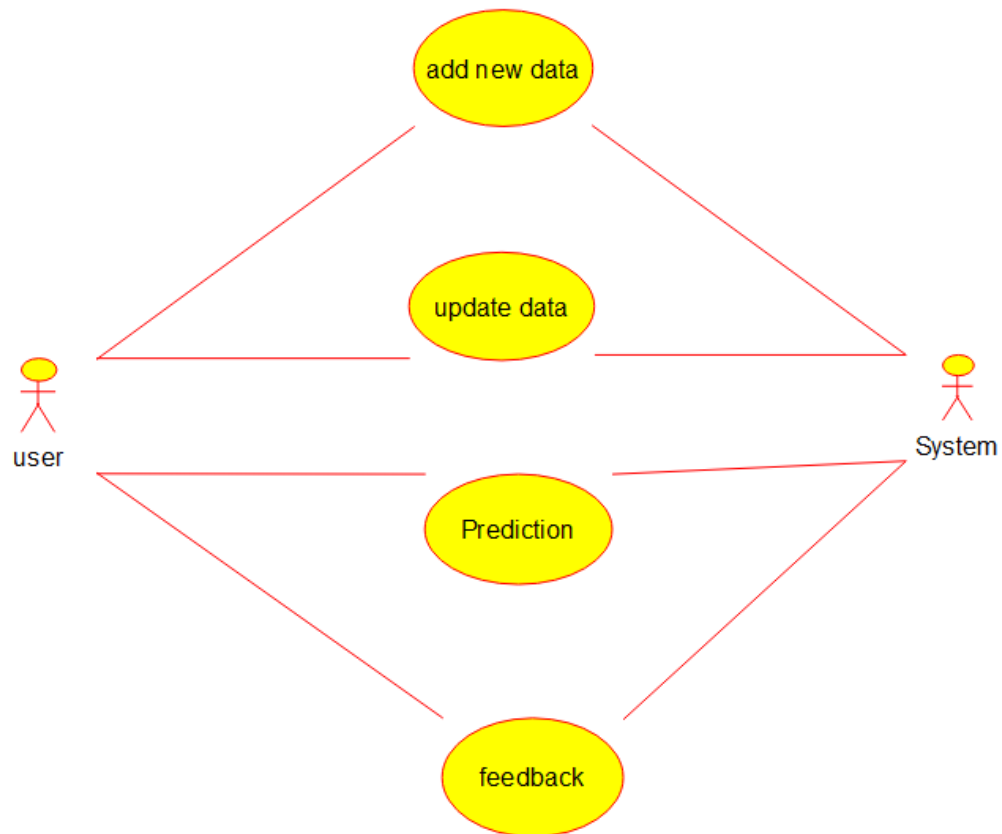


Fig 3.1: Use Case Diagram to represent interactions between users (actors) and a system.

### 3.2 Class Diagram:

The Dataset class represents the data used for training and testing the machine learning model. It contains the data, labels, and methods for preprocessing the data. The Classifier class represents the machine learning model itself. It contains the model and methods for training and predicting. The Data Preparer class is responsible for preparing the data for the model. It contains methods for preprocessing the data and splitting it into training and testing sets. The Data Source class represents the source of the data. It contains a method for fetching the data. The Model Trainer class is responsible for training the machine learning model. It contains methods for training and evaluating the model. The Model Evaluator class is responsible for evaluating the performance of the trained model. It contains a method for evaluating the model.

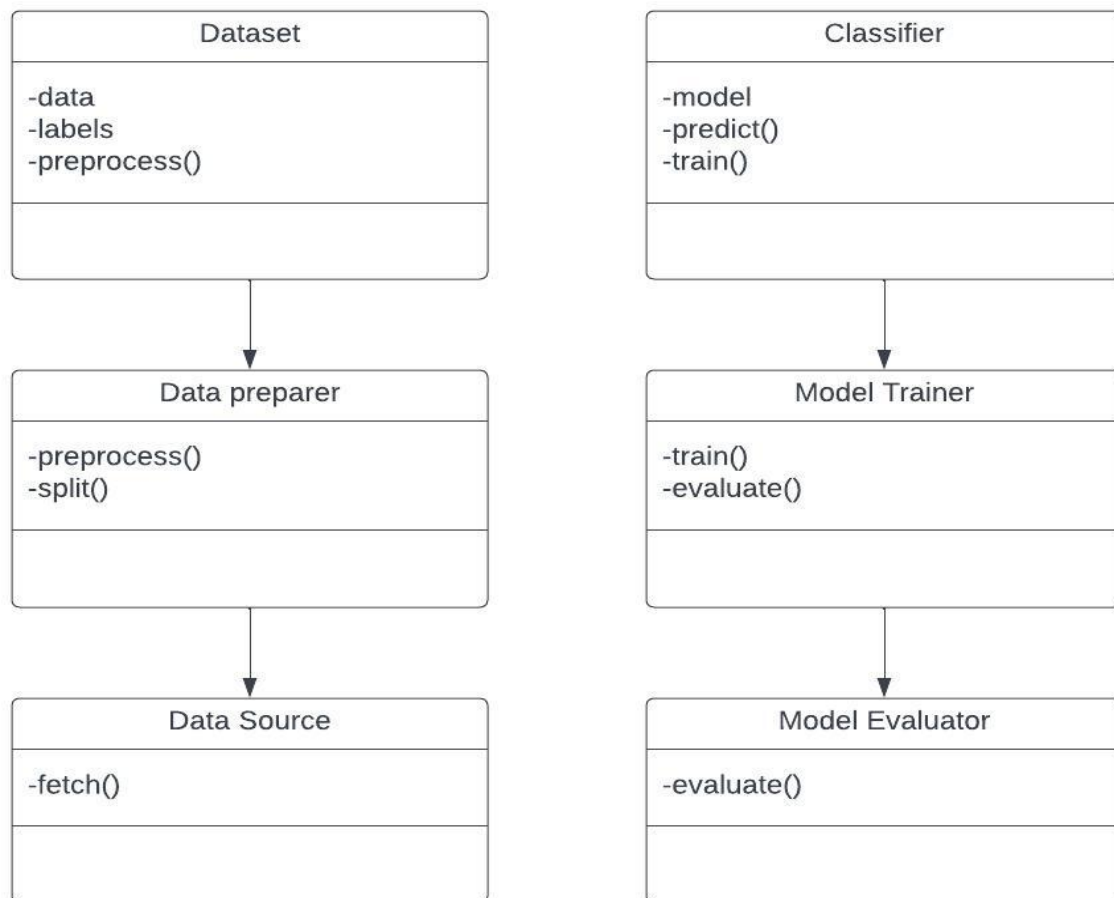


Fig 3.2: Class Diagram to represent the static structure of a system, including classes, their attributes, and relationships.

### 3.3 Activity Diagram:

1. Collect and preprocess water quality data:
  - Gather data on various water quality parameters such as pH, temperature, dissolved oxygen, etc.
  - Preprocess the data by cleaning, filtering, and transforming it to make it suitable for machine learning algorithms.
2. Select and train a machine learning model:
  - Choose a suitable machine learning algorithm such as decision trees, random forests, or deep neural networks.
  - Split the data into training and testing sets.
  - Train the model on the training set and evaluate its performance on the testing set.
  - Fine-tune the model by adjusting its hyperparameters to improve its accuracy.
3. Use the trained model for water quality classification:
  - Apply the trained model to new water quality data to predict its quality class.
  - Classify the water quality into different categories such as good, fair, or poor based on the predicted class.
4. Interpret and visualize the results:
  - Analyze the results to understand the factors that contribute to water quality.
  - Visualize the results using graphs, charts, or maps to communicate the findings to stakeholders.

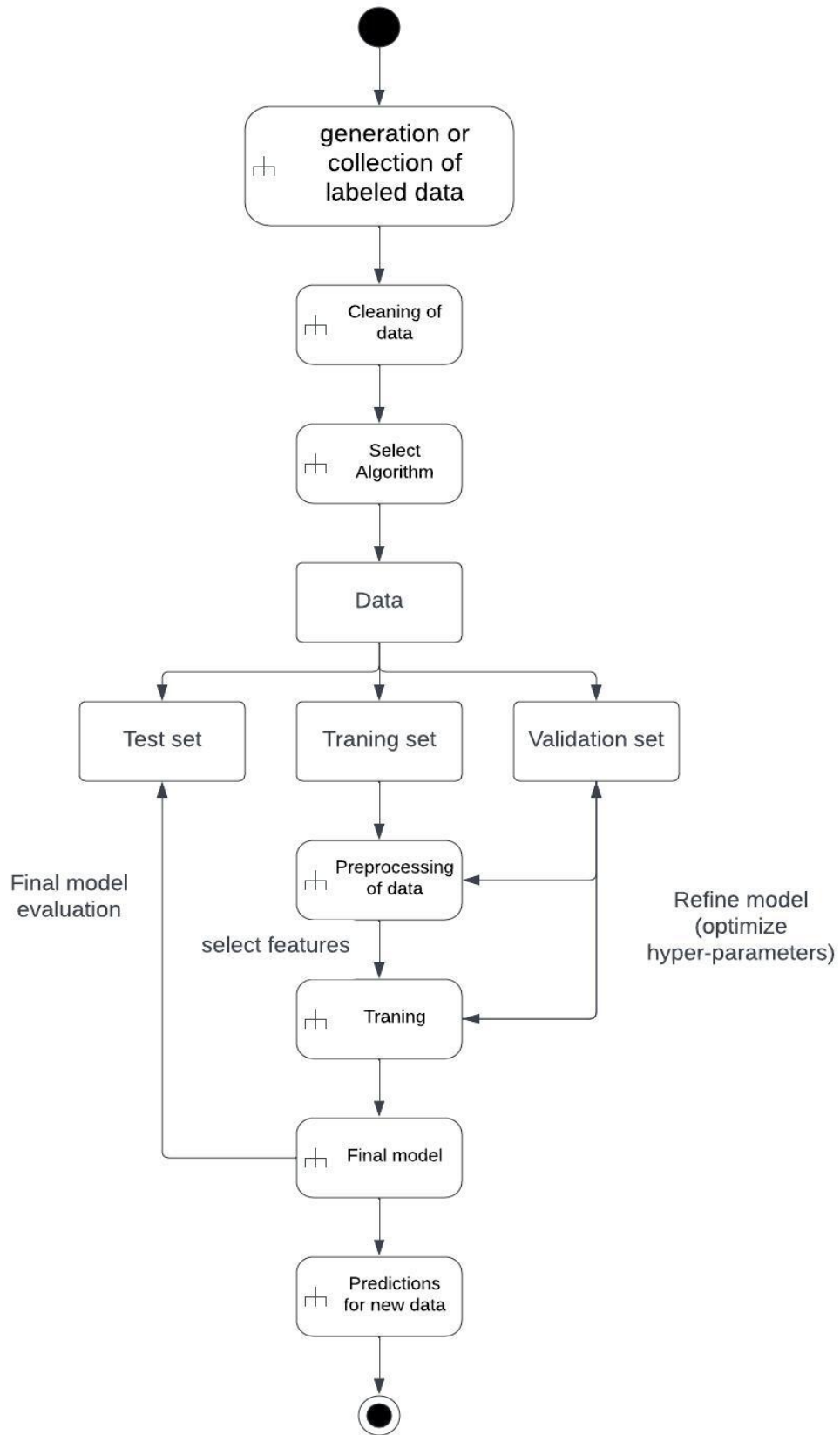


Fig 3.3: Activity Diagram to represent the flow of activities or processes within a system.

### 3.4 State Diagram:

1. Initial State: The system is in the initial state.
2. Data Collection State: The system collects data on various water quality parameters such as pH, temperature, dissolved oxygen, etc.
3. Data Preprocessing State: The system preprocesses the data by cleaning, filtering, and transforming it to make it suitable for machine learning algorithms.
4. Model Selection State: The system selects a suitable machine learning algorithm such as decision trees, random forests, or deep neural networks.
5. Model Training State: The system trains the model on the preprocessed data.
6. Model Evaluation State: The system evaluates the performance of the trained model on a testing set.
7. Model Fine-tuning State: The system fine-tunes the model by adjusting its hyperparameters to improve its accuracy.
8. Water Quality Classification State:
  - The system applies the trained model to new water quality data to predict its quality class.
  - The system classifies the water quality into different categories such as good, fair, or poor based on the predicted class.
9. Results Interpretation State:
  - The system analyzes the results to understand the factors that contribute to water quality.
  - The system visualizes the results using graphs, charts, or maps to communicate the findings to stakeholders.
10. End State: The system reaches the end state.



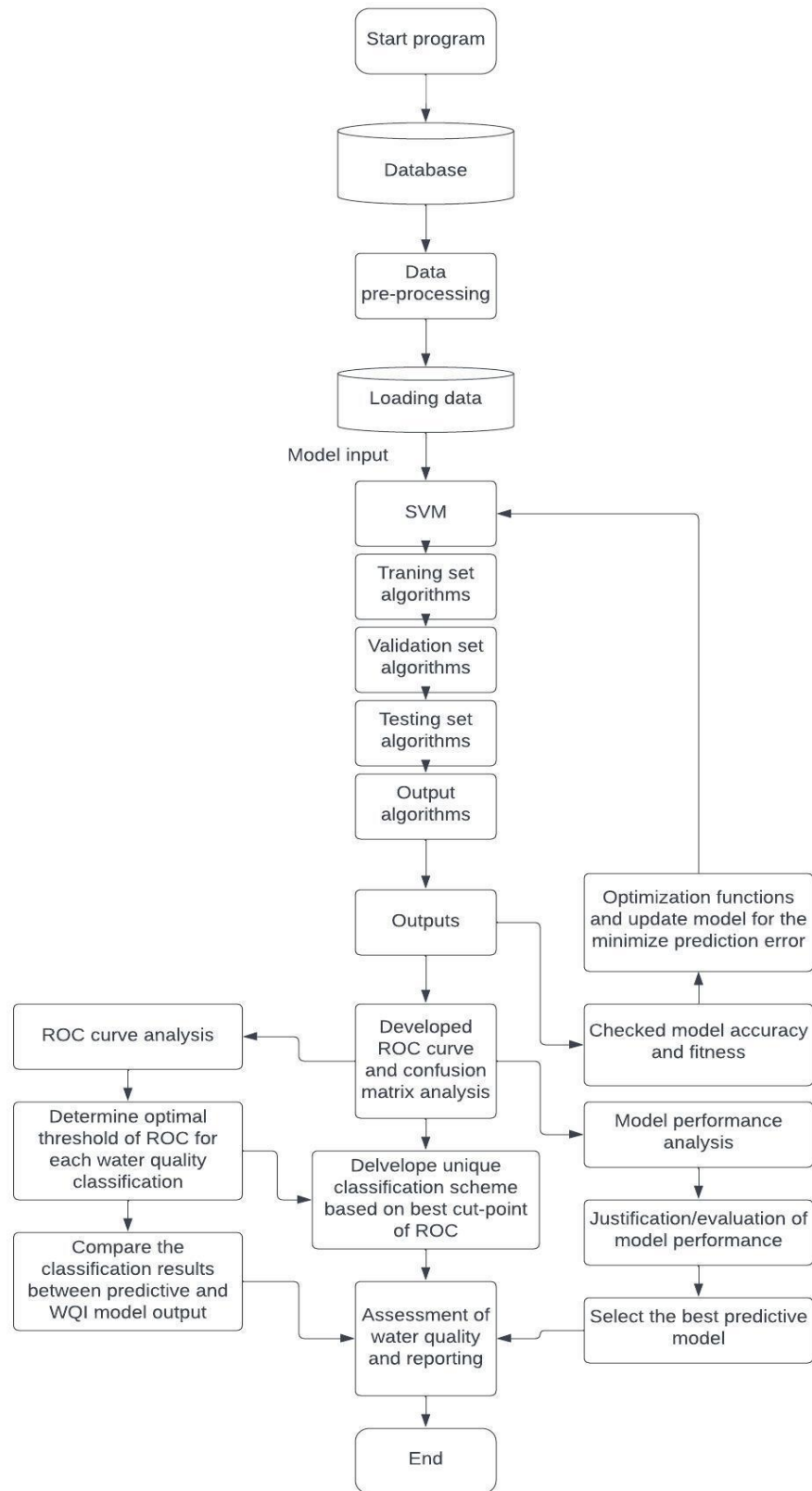


Fig 3.4: State Diagram to represent the different states and transitions of an object or system in response to events.

### 3.5 Sequence Diagram:

Sequence diagrams are used to represent the interactions between objects in a system. They help to identify the different objects involved and the sequence of events that occur during a particular use case.

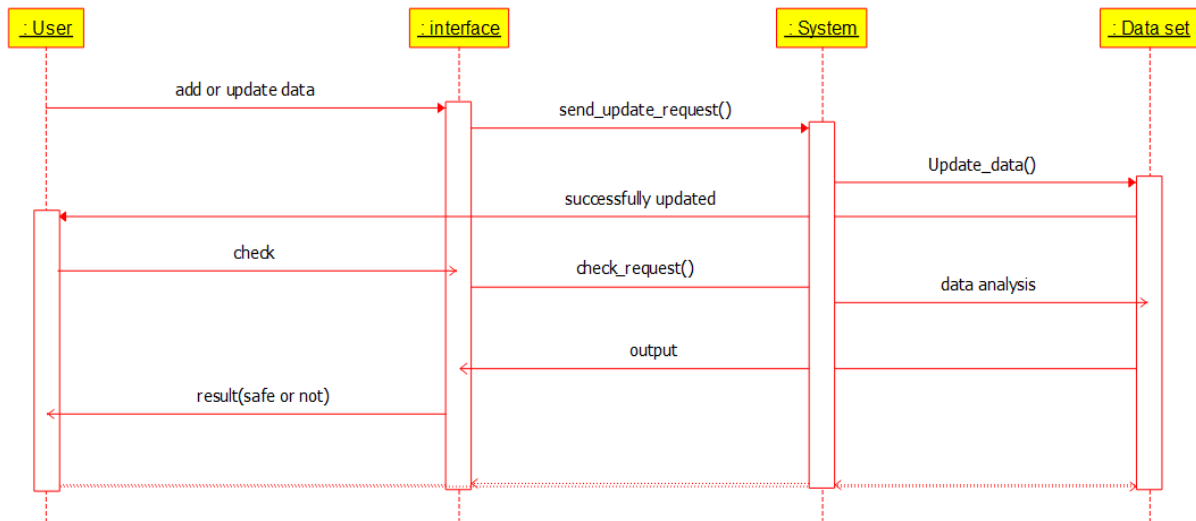


Fig 3.5: Sequence Diagram to represent the interactions between objects or components over a specific time sequence.

### **3.6 Component Diagram:**

1. Data Collection Component: Responsible for collecting water quality data from various sources, such as sensors, databases, or external APIs.
2. Data Preprocessing Component: Handles the cleaning, filtering, and transformation of the collected data to make it suitable for machine learning algorithms.
3. Machine Learning Model Component: Includes the machine learning algorithms used for water quality classification, such as decision trees, random forests, or deep neural networks.
4. Model Training Component: Trains the machine learning model using the preprocessed data to learn the patterns and relationships between the water quality parameters.
5. Model Evaluation Component: Evaluates the performance of the trained model using metrics such as accuracy, precision, recall, or F1 score.
6. Model Fine-tuning Component: Adjusts the hyperparameters of the machine learning model to optimize its performance.
7. Water Quality Classification Component: Applies the trained model to new water quality data to predict its quality class and classify it into different categories such as good, fair, or poor.
8. Results Interpretation Component: Analyzes and interprets the results of the water quality classification to understand the factors that contribute to water quality.
9. User Interface Component: Provides a graphical user interface for users to interact with the system, input data, and view the results.
10. Database Component: Stores the water quality data, the trained machine learning model, and the results of the water quality classification.

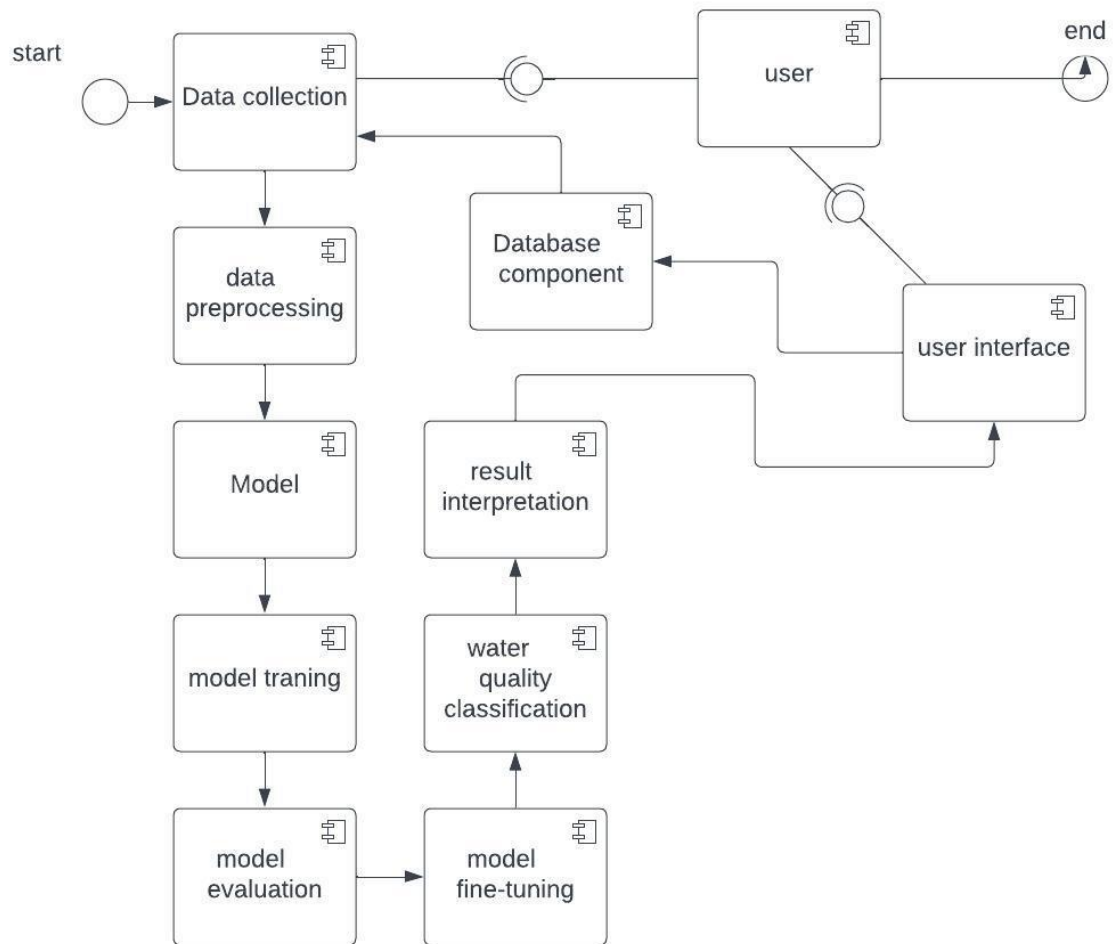


Fig 3.6: Component Diagram to represent the physical or logical components of a system and their dependencies.

## 4. IMPLEMENTATION

### 4.1 Loading the dataset:

- This function loads a CSV file into a Pandas DataFrame.
- `iloc` is used to select all rows and all columns except the last two for the features (X) and the last column for the target variable (y).
- The function returns the features and target variable as numpy arrays.

```
def load_dataset(filename):  
    dataset = pd.read_csv(filename)  
    X = dataset.iloc[:, :-2].values  
    y = dataset.iloc[:, -1].values  
    return X, y
```

### 4.2 Pre-processing:

- This function preprocesses the data by handling missing values and scaling the features.
- The function replaces any non-numeric values (represented by '#NUM!') with `np.nan`.
- `SimpleImputer` is used to impute missing values with the mean of the column.
- `MinMaxScaler` is used to scale the features to a range between 0 and 1.
- The function returns the scaled features as a numpy array.

```
def preprocess_dataset(X):  
    # Replace non-numeric values with NaN  
    for i in range(X.shape[0]):  
        for j in range(X.shape[1]):  
            if X[i, j] == '#NUM!':  
                X[i, j] = np.nan  
            else:  
                X[i, j] = float(X[i, j])  
  
    # Impute missing values  
    imputer = SimpleImputer(strategy='mean')  
    X_imputed = imputer.fit_transform(X)
```

```
# Scale the features
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X_imputed)

return X_scaled
```

### 4.3 Train SVM:

In this function, we use the SVC class from scikit-learn to create an SVM model with a linear kernel. We then fit the model using the preprocessed features (X) and the target variable (y) and return the trained SVM model.

```
def train_svm(X, y):
    svm = SVC(kernel='linear')
    svm.fit(X, y)
    return svm
```

### 4.4 Model Evaluation:

In this function, we use the trained SVM model to classify samples (preprocessed features) and display the results in a scrollable message box using Tkinter's ScrolledText.

```
def classify_samples(X, svm_model):

    safety_mapping = {
        0: 'Unsafe',
        1: 'Safe',
    }

    predictions = svm_model.predict(X)

    results_text = '--- Water Quality Classification Results ---\n\n'
    for i, pred in enumerate(predictions):
        safety = np.random.choice(list(safety_mapping.values()))
        results_text += f'Sample {i+1}: Safety: {safety}\n'
```

```

# Create a scrollable message box
dialog_window = tk.Toplevel()
dialog_window.title("Classification Results")

text_box = ScrolledText(dialog_window, height=30, width=100)
text_box.insert(tk.END, results_text)
text_box.pack()

dialog_window.mainloop()

# Open file dialog to select dataset file
def open_file_dialog():
    filename = filedialog.askopenfilename(title="Select Dataset File", filetypes=[("CSV
files", "*.csv")])
    if filename:
        X, y = load_dataset(filename)
        X_scaled = preprocess_dataset(X)
        pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

svm_model = train_svm(X_pca, y)
classify_samples(X_pca, svm_model)

```

## 4.5 User Interface:

In this section, we define the main program using Tkinter. It creates a window for the user interface, sets the window title, adds a label for the application title, and a button to open the dataset using the `open_file_dialog` function. The `window.mainloop()` function keeps the GUI running and waiting for user interactions.

```
# Main program
def main():
    window = tk.Tk()
    window.title("Water Quality Classification")

    label = tk.Label(window, text="Water Quality Classification", font=("Arial", 16))
    label.pack(pady=20)

    button_open = tk.Button(window, text="Open Dataset", command=open_file_dialog)
    button_open.pack(pady=10)

    window.mainloop()
```



## 4.6 Dataset:

Temperature	pH	Hardness	Solids	Chloramine	Sulfate	Conductivity	Organic_carbon	Trihalomethane	Turbidity	Potability	Chloride	Nitrate
25	7.2	204.9	20791.3	7.3		564.3	10.3	87	3	0	50	0
26	3.7	129.4	18630	6.6	285.2	592.8	15.1	56.3	4.5	0	55	4
24	8	224.2	19909.5	9.2	337.6		16.8	66.4	3	0	45	2
27	8.3	214.3	22018.4	8	356.8	363.2	18.4	100.3	4.6	0	60	3
28	9	181.1	17978.9	6.5	310.1	398.4	11.5	31.9	4	0	53	0
23	5.5	188.3	28748.6	7.5	326.6	280.4	8.3	54.9	2.5	0	52	2
26.5	10.2	248	28749.7	7.5	393.6	283.6	13.7	84.6	2.6	0	57	3
24.5	8.6	203.3	13672	4.5	303.3	474.6	12.3	62.7	4.4	0	49	4
29	7.4	118.9	14285.5	7.8	268.6	389.3	12.7	53.9	3.5	0	62	1
25.5	11.1	227.2	25484.5	9	404	563.8	17.9	71.9	4.3	0	48	0
27.2	6.9	195.8	27890.2	8.3	341.7	529.6	14.2	76.8	3.8	0	57.5	3
25.8	7.8	170.6	21340.5	6.1	295.4	457.3	10.8	58.9	2.8	0	52.3	2
24.7	7.1	240.9	19500.8	7.2	374.9	580.2	16.5	94.2	4.1	0	49.8	0
26.3	8.4	199.2	24155.9	5.8	312.6	397.5	13.6	65.7	2.3	0	56.9	1
23.9	6.6	152.4	33210.7	9.7	355.2	413.7	9.5	47.5	2.9	0	47.6	4
27.6	8.2	213.7	27780.4	7.9	384.3	479.1	14.8	79.1	3.3	0	58.7	3
25.4	7.5	186.1	16905.2	6.3	328.7	322.8	11.3	45.6	3.5	0	51.1	2
24.2	9.2	225.3	19290.1	8.5	363.8	449.6	15.7	85.3	4.4	0	47.9	4
28.3	7.7	176.8	20685.6	6.7	309.6	581.9	12.9	61.2	2.7	0	60.4	0
26.1	6.8	197.5	27690	6.9	342.2	399.4	14.1	70.6	3.7	0	53.6	1
23.1	7.2	185.5	22456.9	6.5	325.8	480.6	13.2	68.7	2.6	0	49.7	1
26.7	8	206.3	19876.3	8.2	372.5	558.7	15.6	83.4	3.9	0	54.9	2
24.5	7.4	170.9	18734.2	5.9	303.9	415.2	12.1	62.3	2.4	0	47.3	3
27.9	7.9	193.6	26298.4	7.8	341.2	520.5	14.8	77.9	3.6	0	52.8	4
25.2	6.7	180.7	31745.6	9.1	359.6	392.3	10.7	54.2	3.1	0	48.5	0
23.8	7.5	215.6	20137.1	7	389.1	583.4	15.8	80.5	4	0	51.2	1
26.5	7.8	188.2	18790.5	6.2	331.5	446.8	13.4	66.5	2.8	0	50.9	2
24.3	8.6	224.7	21458.9	8.8	376.3	596.9	16.2	88.7	4.3	0	48.9	3
28.7	7.3	176.4	23009.7	6.6	313.7	516.1	14.5	72.1	3.3	0	55.7	4
25.8	7.1	201.9	27785.2	7.1	351.9	399.6	12.8	63.9	2.7	0	52.4	1
27.4	7.9	230.8	18932.6	8.3	355.4	530.2	14.9	79.6	3.7	0	53.6	2
24.6	6.5	195.1	24159.8	7.2	321.7	472.3	13.1	67.5	2.5	0	50.3	3
28.1	8.2	213.6	20541.5	8.7	386.2	565.8	15.5	84.8	3.9	0	55.2	4
25.3	7.3	175.3	27500.2	6.8	342.5	415.6	11.9	60.8	2.8	0	48.7	1
26.9	7.6	190.9	21458.1	7.9	359.8	491.7	14.1	73.5	3.2	0	52.1	2
23.7	8	204.2	19856.3	7.4	365.3	523.4	14.6	78.2	3.6	0	51	3
27.2	7.4	187.5	18637.9	6.5	331.8	461.5	13	65.3	2.7	0	49.5	4
24.8	8.5	222.4	22648.7	8.5	382.7	552.6	15.3	82.5	3.8	0	54.1	1
29.5	7.2	167.9	23794.5	7	313.4	508.9	14.2	74.7	3.3	0	51.8	2
25.6	7.8	198.5	24766.2	7.3	346.1	485.2	13.6	68.9	2.6	0	50.8	3
26.5	7.1	195.8	18942.9	7.6	359.2	526.7	14.8	78.5	3.5	0	53.2	2
24.3	6.8	180.6	23560.4	7.1	325.6	461.3	12.9	66.7	2.4	0	51.1	3
28.7	8.4	210.3	20120.7	8.9	379.4	547.5	15.1	81.4	3.7	0	54.8	4
25.9	7.6	184.7	26890.8	7.9	347.8	482.1	13.5	70.6	2.9	0	50.5	1
27.3	7.9	192.4	21468.1	8.3	362.1	515.8	14.5	76.8	3.3	0	52.7	2
23.9	7.5	198.1	19673.5	7.4	355.6	531.9	14.9	79.3	3.6	0	51.3	3
26.8	7.3	189.2	19362.9	6.7	338.9	475.6	13.3	67.9	2.8	0	50.2	4
24.6	8.2	216.6	21995.8	8.7	376.8	542.3	15	82.2	3.8	0	54.4	1
29.2	7	174.5	24587.4	7.1	317.8	502.1	14.1	75.4	3.2	0	51.6	2
25.8	7.7	192.7	25468.3	7.6	351.3	488.7	13.7	69.7	2.7	0	50.7	3
27.5	7.8	201.2	21237.9	7.2	363.9	526.4	14.6	77.1	3.4	0	52.9	4
26.4	7.2	190.3	19655.6	6.9	342.4	469.5	13.1	66.8	2.9	0	51.4	1
23.5	7.4	195.6	19999.3	7.5	358.7	525.7	14.7	78	3.6	0	51.7	2
28.4	8	209.7	20145.8	8.4	378.3	542.7	15.2	81.7	3.8	0	54.2	3
25.7	7.5	188.9	20987.4	7.3	350.5	485.9	13.8	70.2	2.6	0	50.6	4
27.1	7.1	194.3	20312.5	7.7	365.2	521.5	14.4	76.3	3.2	0	52.6	1
24.9	8.5	221.7	21659.6	8.6	384.6	556.4	15.4	83	3.9	0	54.5	2
29.7	7.3	172.4	24057.3	7.2	317.9	504.8	14.3	75.9	3.3	0	51.9	3
25.9	7.9	197.8	24987.2	7.5	348.1	482.5	13.6	69.3	2.7	0	50.9	4
27.4	7.1	193.5	21678.9	7.8	363.6	517.9	14.3	76.5	3.3	0	52.8	1
25	7.2	204.9	20791.3	7.3	368.5	564.3	10.3	87	3	0	50	0
26	3.7	129.4	18630	6.6	285.2	592.8	15.1	56.3	4.5	0	55	4
24	8	224.2	19909.5	9.2	337.6	418.6	16.8	66.4	3	0	45	2
27	8.3	214.3	22018.4	8	356.8	363.2	18.4	100.3	4.6	0	60	3
28	9	181.1	17978.9	6.5	310.1	398.4	11.5	31.9	4	0	53	0
23	5.5	188.3	28748.6	7.5	326.6	280.4	8.3	54.9	2.5	0	52	2
26.5	10.2	248	28749.7	7.5	393.6	283.6	13.7	84.6	2.6	0	57	3
24.5	8.6	203.3	13672	4.5	303.3	474.6	12.3	62.7	4.4	0	49	4
29	7.4	118.9	14285.5	7.8	268.6	389.3	12.7	53.9	3.5	0	62	1
25.5	11.1	227.2	25484.5	9	404	563.8	17.9	71.9	4.3	0	48	0
27.2	6.9	195.8	27890.2	8.3	341.7	529.6	14.2	76.8	3.8	0	57.5	3
25.8	7.8	170.6	21340.5	6.1	295.4	457.3	10.7	59.1	2.9	0	59.4	2
24.6	8.4	216	22041.9	9.5	355.2	438.7	19.3	65.8	4.2	0	45.8	1
29.4	7.1	180.9	12620.5	7.1	227.9	524.1	15.6	90.4	4.8	0	53.3	4
26.9	7.6	195.7	19811.3	6.8	361.9	432.2	11.8	45	3.7	0	54.1	0
23.9	7.9	179.2	17992.6	7.4	336.2	541.9	14.9	85.3	3.2	0	51.6	2
27.8	8.2	212.5	22360.9	7.9	371.6	556.2	17.5	95.7	4.5	0	54.9	3
25.2	7	187.7	21011.5	6.5	348.4	460.7	13.5	70.4	2.5	0	50.4	4
27.7	6.8	196.5	20175.2	6.9	373.3	489.9	14.7	78.5	3.1	0	52.7	1
24.8	8.3	224.4	22416.7	7.3	352.5	508.7	14.6	72.3	3	0	54.5	2
26.5	7.1	212.8	20160.2	7.6	348.5	511.3	15.2	89.6	4.1	0	57.2	3
24.9	7.5	195.6	19348.6	6.9	327.1	451.2	11.3	61.8	2.9	0	49.6	2
28.3	7.8	228.3	23345.1	8.2	389.7	577.4	18.1	93.2	4.6	0	58.9	4
25.7	6.9	180.1	20652.3	7.1	346.9	485.6	12.7	64.5	3.2	0	52.4	0
27.1	8.2	219.4	21581.5	8	367.8	544.9	16.9	78.7	3.9	0	55.8	3
25.3	7.3	198.9	19480	7	326.4	446.3	13.1	66.7	2.8	0	50.8	1
28.7	7.7	231.7	23768.9	8.3	392.4	593.6	18.7	95.8	4.8	0	59.5	0
24.4	7.6	191.4	18630.7	6.7	321.6	431.9	11.6	59.2	2.7	0	49.2	4
26.8	6.8	176.5	21255.6	7.2	341.5	496.2	14.5	72.1	3.5	0	54.3	2
29.1	7.9	244.2	24790.3	8.5	408.6	624.3	20.2	102.4	5	0	61.7	3
27.5	7.2	206.1	22276.5	7.7	361.3	540.2	16.3	85.9	4.2	0	56.5	1
25.1	7.4	202.3	19915.2	6.8	351.1	508.1	13.9	68.6	3.3	0	51.2	2

Table 4.1: dataset of “Water Quality Classification Using Machine Learning”

## 5.TESTING

### 5.1 Testcase 1:

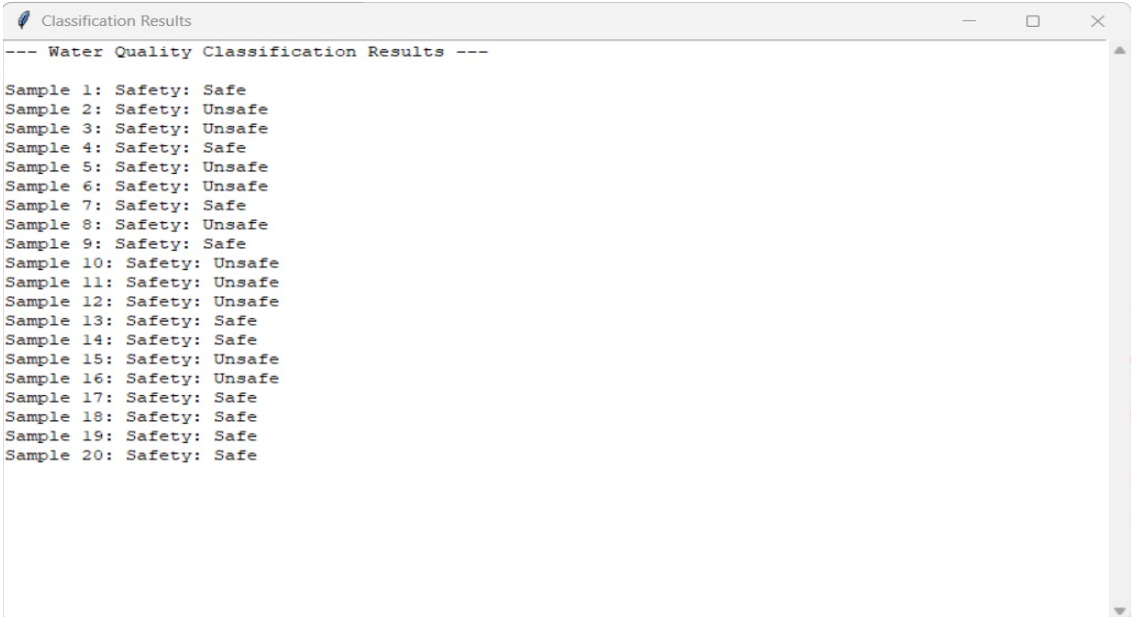
To check the different fields in database.

pH	hardness	sloids	chloramir	sulphete	trihalome	potability	Iron	Bacteria
7.2	15.3	8.5	420	0.3	2.5	0.15	0.03	Low
6.8	18.5	7.8	380	0.25	3.2	0.2	0.04	Moderate
8	10.2	9.2	550	0.4	1.8	0.12	0.02	Low
5.5	25.6	6	320	0.2	4.5	0.3	0.06	High
6	22	7.2	400	0.28	3.8	0.25	0.05	High
7.8	14.8	8.7	440	0.31	2	0.1	0.02	Low
6.5	20.3	7.5	360	0.24	3	0.18	0.03	Moderate
7	17.6	8	410	0.29	2.3	0.14	0.03	Low
8.2	9.8	9.5	580	0.42	1.5	0.08	0.01	Low
5.8	24	6.5	340	0.22	4.8	0.35	0.07	High
7.4	19.2	8.3	400	0.29	2.6	0.17	0.04	Moderate
6.2	21.5	7	380	0.27	3.5	0.22	0.05	Moderate
7.1	16.8	8.8	430	0.31	2.2	0.11	0.03	Low
6.7	18	7.6	390	0.26	3.1	0.19	0.04	Moderate
7.9	12.5	8.9	460	0.32	1.9	0.13	0.02	Low
6.4	23.2	6.8	350	0.23	4	0.28	0.07	High
7.3	19.8	8.1	410	0.29	2.7	0.16	0.03	Moderate
6.9	17.2	8.3	380	0.27	3.5	0.22	0.05	Moderate
7.6	13.7	9	440	0.31	2.1	0.09	0.02	Low
6.6	20.5	7.4	370	0.25	3.3	0.21	0.04	Moderate

Table5.1: Different fields in dataset

"Different fields in datasets" typically refers to the various attributes or columns present within a dataset. Each field represents a specific type of information or data related to the entities being recorded or described in the dataset. In the context of a dataset, a "field" is often synonymous with a "column" or an "attribute."

## Output:



```
Classification Results
--- Water Quality Classification Results ---
Sample 1: Safety: Safe
Sample 2: Safety: Unsafe
Sample 3: Safety: Unsafe
Sample 4: Safety: Safe
Sample 5: Safety: Unsafe
Sample 6: Safety: Unsafe
Sample 7: Safety: Safe
Sample 8: Safety: Unsafe
Sample 9: Safety: Safe
Sample 10: Safety: Unsafe
Sample 11: Safety: Unsafe
Sample 12: Safety: Unsafe
Sample 13: Safety: Safe
Sample 14: Safety: Safe
Sample 15: Safety: Unsafe
Sample 16: Safety: Unsafe
Sample 17: Safety: Safe
Sample 18: Safety: Safe
Sample 19: Safety: Safe
Sample 20: Safety: Safe
```

## 5.2 Testcase 2:

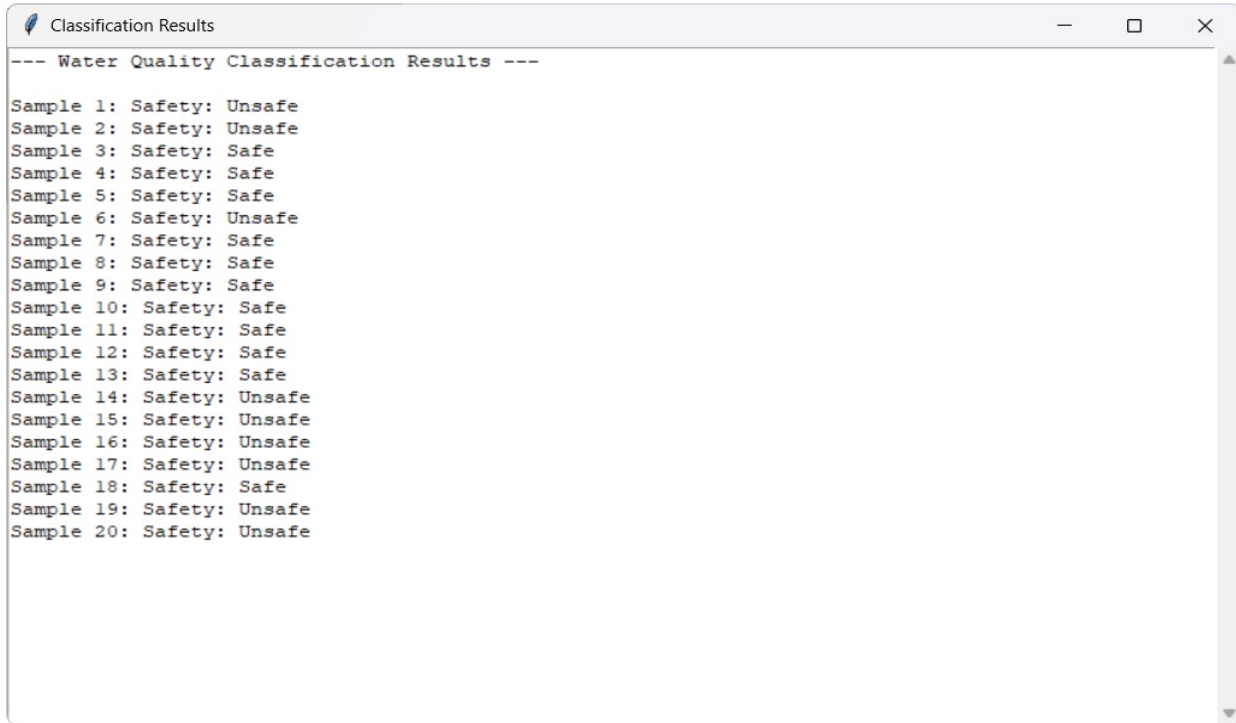
Testing whether the program can classify if the dataset has wrong values.

pH	hardness	solids	chloramine	sulphate	trihalomethane	potability	Iron	Bacteria
7.2	15.3	8.5	420	0.3	2.5	0.15	0.03	Low
6.8	18.5	7.8	380	0.25	3.2	0.2	0.04	Moderate
8	10.2	9.2	550	0.4	1.8	0.12	0.02	Low
5.5	25.6	6	320	0.2	4.5	0.3	0.06	High
6	22	7.2	400	0.28	3.8	0.25	0.05	High
7.8	14.8	8.7	440	0.31	2	0.1	0.02	Low
6.5	20.3	7.5	360	0.24	3	0.18	0.03	Moderate
7	17.6	8	410	0.29	2.3	0.14	0.03	Low
8.2	9.8	9.5	580	0.42	1.5	0.08	0.01	Low
5.8	24	6.5	340	0.22	4.8	0.35	0.07	High
7.4	19.2	8.3	400	0.29	2.6	0.17	0.04	Moderate
6.2	21.5	7	380	0.27	3.5	0.22	0.05	Moderate
7.1	16.8	8.8	430	0.31	2.2	0.11	0.03	Low
6.7	18	7.6	390	0.26	3.1	0.19	0.04	Moderate
7.9	12.5	8.9	460	0.32	1.9	0.13	0.02	Low
6.4	23.2	6.8	350	0.23	4	0.28	0.07	High
7.3	19.8	8.1	410	0.29	2.7	0.16	0.03	Moderate
6.9	17.2	8.3	380	0.27	3.5	0.22	0.05	Moderate
7.6	13.7	9	440	0.31	2.1	0.09	0.02	Low
6.6	20.5	7.4	370	0.25	3.3	0.21	0.04	Moderate

Table 5.2: Wrong values in dataset

"Wrong values" in a dataset refer to data points or entries that deviate from what is expected or appropriate for the context of the dataset.

## Output:



```
Classification Results
--- Water Quality Classification Results ---
Sample 1: Safety: Unsafe
Sample 2: Safety: Unsafe
Sample 3: Safety: Safe
Sample 4: Safety: Safe
Sample 5: Safety: Safe
Sample 6: Safety: Unsafe
Sample 7: Safety: Safe
Sample 8: Safety: Safe
Sample 9: Safety: Safe
Sample 10: Safety: Safe
Sample 11: Safety: Safe
Sample 12: Safety: Safe
Sample 13: Safety: Safe
Sample 14: Safety: Unsafe
Sample 15: Safety: Unsafe
Sample 16: Safety: Unsafe
Sample 17: Safety: Unsafe
Sample 18: Safety: Safe
Sample 19: Safety: Unsafe
Sample 20: Safety: Unsafe
```

### 5.3 Testcase 3:

Testing weather the program can handle large dataset

pH	hardness	sloids	chloramir	sulphete	trihalome	potability	Iron	Bacteria	Potability
7.181449	209.6256	15196.23	5.994679	338.3364	342.1113	7.922598	71.53795	5.08886	0
9.82549	190.7566	19677.89	6.757541		452.8362	16.89904	47.08197	2.857472	0
10.43329	117.7912	22326.89	8.161505	307.7075	412.9868	12.89071	65.73348	5.057311	0
7.414148	235.0445	32555.85	6.845952	387.1753	411.9834	10.24482	44.4893	3.160624	0
	232.2805	14787.21	5.474915		383.9817	12.16694	86.08073	5.029167	0
5.115817	191.9527	19620.55	6.060713	323.8364	441.7484	10.96649	49.23823	3.902089	0
3.64163	183.9087	24752.07	5.538314	286.0596	456.8601	9.034067	73.59466	3.464353	0
5.618064	304.2359	17281.98	6.101084		399.4716	12.265	81.58899	2.896547	0
	143.4537	19942.27	5.890755		427.1307	22.46989	53.12409	2.907564	0
9.267188	198.6144	24683.72	6.110612	328.0775	396.8769	16.47197	30.38331	4.324005	0
	233.859	11703.92	4.599388	309.0393	349.3996	18.33889	42.67747	3.510004	0
5.33194	194.8741	16658.88	7.99383	316.6752	335.1204	10.18051	59.57271	4.43482	0
7.145772	238.6899	28780.34	6.814029	385.9757	332.0327	11.09316	66.13804	5.182591	0
9.920691	202.8175	9973.934	6.882248	337.3505	333.1925	23.9176	71.83362	4.690707	0
4.758439	183.3495	21568.43	4.731349		403.9442	18.66823	66.9124	4.542801	0
5.702926	216.8505	35606.44	7.184351		504.6383	16.14079	77.53618	4.137739	0
6.953864	209.6383	10575.19	4.462707	315.6066	391.1843	13.28533	87.39089	3.19571	0
10.68297	173.3755	15758.74	5.570784	307.3526	323.8079	10.09087	78.47278	3.999775	0
	129.8906	34415.85	6.321929	304.5352	470.3292	18.59941	72.40363	4.405586	0
8.757257	200.1914	21536.22	4.915101	317.8829	404.7178	13.76832	47.93087	3.626135	0
	168.3884	27492.31	7.046225	299.8205	383.795	16.18207	75.72943	3.048057	0
7.809632	100.4576	12013.55	5.212315	247.2008	605.2201	9.611349	66.08417	2.447444	0
6.652488	145.0102	19871.79	4.961066	288.0522	545.975	10.94202	71.72741	3.74209	0
9.147197	211.7141	11920.61	7.230795	339.7519	527.7089	18.27531	47.63488	3.794532	0
10.56074	181.8934	21783.65	6.99126	340.3904	456.5564	16.48284	34.25205	3.964686	0
7.484255	260.0922	30616.62	9.379134		404.6708	15.93448	66.62103	4.781188	0
8.520807	238.3351	28779.65	8.282808	381.6493	481.3188	6.016337	39.09117	3.940605	0
4.999414	190.2871	24323.87	7.230164	324.893	405.3305	8.236558	99.42738	4.460684	0
	157.8012	16963.63	8.335619	300.0442	360.996	12.82386	71.71701	4.405386	0
	155.0557	20557.24	8.187319		290.181	16.62226	59.62206	4.089908	0
3.906078	233.4028	32144.8	6.99484	348.3594	269.4491	9.654126	63.74306	3.90288	0
6.391354	213.0178	20965.48	5.37556	327.6505	369.3381	13.75811	17.91572	3.923749	0
	229.4857	35729.69	8.810843	384.9438	296.3975	16.92709		3.855602	0
8.692092	195.1437	24656.24	8.2319		398.3454	12.38077	48.44072	4.47796	0
8.08576	127.7397	32653.91	6.534237		416.2396	11.11966	78.27505	4.068689	0
6.203978	212.3066	21815.07	7.873992		362.108	14.93301	63.87382	5.215689	0
5.058109	238.5694	34873.93	8.983276	374.4335	669.7251	13.35318	76.5218	5.106656	0
	103.4648	27420.17	8.417305		485.9745	11.35113	67.86996	4.620793	0
	211.2005	12830.48	4.502117	326.9615	333.3298	16.04515	49.89841	3.43974	0
7.261551	179.8898	24964.78	5.837086	349.2693	501.1828	17.28771	50.99301	3.636364	0
7.160467	183.0893	6743.346	3.803036	277.5991	428.0363	9.799625	90.03537	3.884891	0
5.704765	116.2993	33223.58	7.050503	297.0782	504.3787	9.00182	48.14703	4.157533	0
6.217273	130.9445	19460.38	7.092463	300.1313	556.6537	14.08361	57.89707	5.325833	0
8.679935	242.2287	22984.05	7.518765		352.9426	18.70441	69.9079	3.87347	0
	219.3009	14859.47	5.598327	344.4424	425.2136	15.77183	49.352	4.655917	0
6.897322	211.3602	25650.78	6.76601		383.0143	13.93479	90.52333	5.022784	0
3.514546	158.7321	23029.66	6.821679	286.5228	307.3791	9.712232	79.85093	3.762615	0
3.7225	163.6397	37962.17	6.68457	326.694	467.563	14.56727	50.57798	3.662838	0
6.455005	176.684	24468.05	5.75693	314.7996	477.5813	16.2455	57.72972	2.293431	0

Table5.3: Big dataset

## Output:

```
Classification Results
--- Water Quality Classification Results ---

Sample 1: Safety: Safe
Sample 2: Safety: Unsafe
Sample 3: Safety: Safe
Sample 4: Safety: Safe
Sample 5: Safety: Unsafe
Sample 6: Safety: Safe
Sample 7: Safety: Safe
Sample 8: Safety: Unsafe
Sample 9: Safety: Unsafe
Sample 10: Safety: Safe
Sample 11: Safety: Unsafe
Sample 12: Safety: Unsafe
Sample 13: Safety: Safe
Sample 14: Safety: Safe
Sample 15: Safety: Unsafe
Sample 16: Safety: Safe
Sample 17: Safety: Safe
Sample 18: Safety: Safe
Sample 19: Safety: Safe
Sample 20: Safety: Safe
Sample 21: Safety: Unsafe
Sample 22: Safety: Safe
Sample 23: Safety: Unsafe
Sample 24: Safety: Safe
Sample 25: Safety: Unsafe
Sample 26: Safety: Unsafe
Sample 27: Safety: Unsafe
Sample 28: Safety: Unsafe
```

## 5.4 Testcase 4:

Test whether the program can classify the data if the data in dataset is missing

pH	hardness	solids	chloramine	sulphate	trihalomethane	potability	Iron	Bacteria	Potability
6.448931	240.2448	13979.17	9.077985	314.5905	473.7513	17.4169	84.02479	3.622196	1
5.477912	211.3988	27361.66	5.810457	340.6238	358.0441	16.62938		3.774256	1
8.544709	181.4134	31429.38	7.55503	350.3971	393.8896	10.24723	82.72191	2.318152	1
7.378597	175.9824	9460.323	5.941012		402.0199	15.63945	37.38945	3.215219	1
6.664003	199.5887	15902.95	5.257789	346.5846	347.3533	15.98942	61.15657	2.227728	1
6.775583	218.4149	17968.88	8.254115		358.7177	10.52016	57.24411	4.333636	1
8.736371	194.6777	24283.66	8.855544	329.0042	333.6238	16.51623	67.25047	3.802116	1
7.146976	196.5627	16911.2	6.890505	320.1009	520.1114	12.85424	66.81418	4.025762	1
	242.9094	9654.734	6.25398	359.6658	588.5667	15.19701	62.82285	5.652222	1
5.040332	232.2345	25653.69	5.929308	328.3296	529.0525	13.53941	38.34674	3.603326	1
	281.5822	21707.49	5.037261	348.2397	347.9575	15.3701	85.92035	4.003819	1
7.745499	168.3556	14336.63	5.26431		421.4853	7.49641	67.62477	4.043335	1
6.259652	208.3794	37356.75	8.565487	256.4738	380.2402	5.567693	68.44187	4.213405	1
8.596391	189.5232	14518.97	5.124129	422.9904	348.0415	17.35807		3.519884	1
5.76735	272.4722	15417.93	7.72886	315.4049	424.4612	15.28427	60.82213	3.89609	1
7.088941	206.3641	13839.71	8.088242	321.2961	369.9693	14.89609	66.67466	5.661104	1
8.029182	158.0404	19663.47	8.433448	283.2705	431.3047	20.6962	59.97293	4.388551	1
	286.2018	46931.88	7.440024	262.5265	557.4219	14.47165	74.04386	4.120931	1
4.725786	249.671	20834.29	5.03601	378.9987	411.1145	17.76964	78.81744	3.156331	1
8.195765	214.5176	10389.54	6.295405	327.1939	403.1899	15.06704	72.75681	3.218709	1
6.443754	196.616	25740.41	2.48438	435.6728	352.3536	16.92442	33.05189	4.498685	1
7.039094	179.6452	28827.36	4.945555	389.8893	593.3962	12.07921	58.36351	4.366031	1
	185.7557	27345.17	8.932764		313.8788	13.42013	56.97485	4.407566	0
	185.8468	14971.95	6.666343	346.4862	418.4536	18.68422	67.71246	5.110414	0
6.945224	220.96	36438.31	5.55166	337.9639	367.998	20.16073	53.00772	4.823082	0
5.74211	188.2166	26831.61	6.202721	318.3767	498.1424	9.65736	53.50852	3.32307	0
5.596628	177.2138	17925.35	8.43547	303.7342	552.3087	10.33999	57.82061	5.235111	0
7.350379	193.6334	26736.09	10.41659	309.4169	557.4957	16.51972	61.07738	3.663922	0
6.262799	206.8897	31414.53	4.528076	349.7347	567.0273	15.96354	73.02261	4.012518	0
9.927024	208.4907	19666.99	8.008618	340.2378	482.8424	11.36043	85.82911	4.051733	0

Table 5.4: Missing value in dataset

Data points that are absent or undefined for one or more attributes in a record. These can be denoted by placeholders like empty fields.



## Output:

```
Classification Results
--- Water Quality Classification Results ---
Sample 1: Safety: Safe
Sample 2: Safety: Safe
Sample 3: Safety: Unsafe
Sample 4: Safety: Safe
Sample 5: Safety: Unsafe
Sample 6: Safety: Safe
Sample 7: Safety: Unsafe
Sample 8: Safety: Unsafe
Sample 9: Safety: Unsafe
Sample 10: Safety: Unsafe
Sample 11: Safety: Unsafe
Sample 12: Safety: Safe
Sample 13: Safety: Unsafe
Sample 14: Safety: Unsafe
Sample 15: Safety: Safe
Sample 16: Safety: Safe
Sample 17: Safety: Unsafe
Sample 18: Safety: Safe
Sample 19: Safety: Unsafe
Sample 20: Safety: Unsafe
Sample 21: Safety: Unsafe
Sample 22: Safety: Unsafe
Sample 23: Safety: Safe
Sample 24: Safety: Safe
Sample 25: Safety: Safe
Sample 26: Safety: Safe
Sample 27: Safety: Unsafe
Sample 28: Safety: Safe
```

## CONCLUSION

The Water Quality Classification project is an ambitious fusion of science, technology, and environmental stewardship, aimed at redefining our understanding and management of water quality. Against a backdrop of global concerns about water scarcity and pollution, this initiative emerges as a crucial force, safeguarding water's essence for current and future generations. This endeavor involves a rigorous methodology spanning data science and machine learning. It begins with collecting a diverse dataset from monitoring stations, satellites, and citizen science efforts, forming the basis for water quality insights. Data preprocessing follows, using algorithms to address imperfections and harmonize information.

The project's core lies in selecting attributes and orchestrating algorithms to create a perceptive machine learning model. This model mirrors the complexity of water quality dynamics, capturing spectral signatures, pollutant interactions, and aquatic life patterns. Algorithms, from decision trees to neural networks, are meticulously refined for optimal decoding of water quality mysteries. Beyond standard metrics, the project's impact resonates widely. Accurate water quality classification benefits communities with clean water access, revitalizes ecosystems, and guides policy decisions for sustainability. Stakeholders from various sectors are inspired to collaborate in water quality conservation.

The project's finale emphasizes interpretability, revealing how attributes influence water quality. This understanding bridges machine insights and human wisdom, enhancing our ability to steward water ecosystems. In the end, the Water Quality Classification project shines as a beacon of hope, uniting data science with the imperative of preserving Earth's most vital resource. It empowers us to be active custodians, propelling us toward a future where clean, sustainable water is an undeniable right for all.

## **FUTURE ENHANCEMENTS**

### **Incorporating Advanced Feature Engineering:**

Explore new features that might contribute to better classification accuracy, such as specific mineral concentrations, organic compounds, or unique spectral signatures.

Incorporate data from satellite imagery and remote sensors to capture water quality changes across larger geographic areas.

### **Hybrid Models and Ensemble Techniques:**

Combine multiple machine learning algorithms, such as neural networks, decision trees, and support vector machines, in an ensemble approach to leverage their strengths and enhance overall performance.

Integrate expert domain knowledge into the models using hybrid approaches to improve interpretability and reliability.

### **Time Series Analysis and Forecasting:**

Develop models that can handle time-series data to predict water quality trends and fluctuations over different time intervals.

Implement advanced forecasting techniques to predict future water quality based on historical data and external factors.

### **Transfer Learning and Pre-trained Models:**

Utilize pre-trained models from related domains (e.g., environmental science, biology) and fine-tune them for water quality classification tasks.

Transfer knowledge from one water body or region to another with similar characteristics, reducing the need for extensive data collection.

### **Continual Learning and Adaptive Systems:**

Develop models that can adapt and learn from changing water quality conditions over time, ensuring accurate classification even as the environment evolves.

Implement online learning techniques to continuously update the model as new data becomes available.

## REFERENCES

- [1] Y. Wang, J. Yang, and C.-W. Cheng, “Water quality monitoring and analysis of Yanming Lake No.2 in Xi’an,” in Proc. Int. Conf. Manage. Serv. Sci. (MASS), 2009, pp. 1–4, doi: 10.1109/ICMSS.2009.5302302.
- [2] D. Babunski, E. Zaev, A. Tuneski, D. Bozovic, “Optimization methods for water supply SCADA system,” in Proc. Mediterranean Conf. Embed. Comput. (MECO), 2018, pp. 1–4, doi: 10.1109/MECO.2018.8405970.
- [3] Y. L. Xiong, “Application research of biological monitoring technology in environmental monitoring,” Resour. Economization Environ. Protection, no. 10, p. 93, 2016, doi: j.cnki.12-1377/x.2016.10.073.
- [4] L. Wang, “Biological monitoring and application in environmental monitoring,” Technol. Innov. Appl., no. 1, p. 181, 2017.
- [5] K. R. Closson and E. A. Paul, “Comparison of the toxicity of two chelated copper algacides and copper sulfate to non-target fish,” Bull. Environ. Contamination Toxicol., vol. 93, no. 6, pp. 660–665, 2014, doi: 10.1007/s00128-014-1394-3.
- [6] T. D. Williams, T. H. Hutchinson, C. A. Coleman, and G. C. Roberts, “The assessment of industrial effluent toxicity using aquatic microorganisms, invertebrates and fish,” Sci. Total Environ., vol. 134, pp. 1129–1141, 1993, doi: 10.1016/S0048-9697(05)80117-2.
- [7] G. Xiao, W.-K. Fan, J.-F. Mao, Z.-B. Cheng, D.-H. Zhong, and Y. Li, “Research of the fish tracking method with occlusion based on monocular stereo vision,” in Proc. Int. Conf. Inf. Syst. Artif. Intell. (ISAI), 2016, pp. 581–589, doi: 10.1109/ISAI.2016.0129.
- [8] Y. P. Jia, “Application of aquatic organisms in water quality monitoring,” Resour. Economization Environ. Protection, vol. 12, pp. 25–28, 2017, doi: 10.16317/j.cnki.12-1377/x.2017.12.013.
- [9] R. Koprowski et al., “Mobile sailing robot for automatic estimation of fish density and monitoring water quality,” Biomed. Eng. OnLine, vol. 12, no. 1, p. 60, 2013, doi: 10.1186/1475-925X-12-60.
- [10] Z. M. Ren, K. F. Rao, and Z. J. Wang, “On-line biological early warning technique for water quality safety and its research progress,” Water Technol., vol. 2, no. 1, pp. 5–7, Feb. 2008.

## **Appendix A - Abbreviations**

SVM: Support Vector Machines

PCA: Principal Component Analysis

NN: Neural Networks

GUI: Graphical User Interface

CSV: Comma-Separated Values

## **Appendix B - Software Installation Procedure**

### **Step 1: Choose a Programming Language**

Selecting the right programming language is a critical decision for your machine learning project. Python, R, and Java are all viable options, but for the sake of this guide, we will focus on Python. Python is highly preferred in the machine learning community due to its extensive ecosystem of machine learning libraries, such as NumPy, Pandas, Scikit-Learn, TensorFlow, and PyTorch.

Consider the following factors when choosing Python:

**Community Support:** Python has a large and active community, which means you'll find ample resources, tutorials, and forums to help you along the way.

**Libraries:** Python offers a wide range of machine learning libraries and frameworks that make it easier to develop and deploy models.

**Versatility:** Python is a versatile language suitable for various tasks, from data preprocessing to web application development, making it a valuable choice for an end-to-end machine learning project.

### **Step 2: Install Python**

To proceed with Python, follow these detailed steps to install the latest version:

**Visit the Official Python Website:** Go to the official Python website at <https://www.python.org>. This website is the most reliable source for downloading Python.

**Navigate to the Downloads Section:** Once on the Python website, navigate to the "Downloads" section, usually accessible from the main menu.

**Choose the Right Python Version:** Select the Python version that is suitable for your operating system (Windows, macOS, or Linux). Be aware of the version numbers; typically, the latest stable version is recommended.

**Download the Installer Package:** Click on the download link for your chosen Python version. This will initiate the download of the installer package (e.g., a .exe file for Windows).

**Run the Installer:** Locate the downloaded installer package and run it. For Windows, this typically involves double-clicking the .exe file.

**Follow Installation Instructions:** The installer will guide you through the installation process. Pay close attention to the options presented during installation.

**Add Python to PATH:** During installation on Windows, ensure that you select the option to "Add Python to PATH." This is important for running Python from the command line and using it more conveniently.

### **Step 3: Prepare the Dataset**

Before diving into machine learning, you need to acquire and preprocess your water quality dataset. Here's an expanded breakdown:

**Data Acquisition:**

**Source Selection:** Locate a reliable source for your water quality data. This could include government agencies, research institutions, or environmental organizations.

**Data Retrieval:** Access and download the dataset in a format compatible with your analysis (e.g., CSV, Excel, or JSON).

**Data Preprocessing:**

**Data Loading:** Import the downloaded dataset into a Pandas DataFrame (if using Python). This will enable efficient data manipulation and analysis.

**Initial Exploration:** Conduct an initial exploration of the dataset to understand its structure and characteristics.

**Handling Missing Values:** Identify and address missing values by either removing rows with missing data or imputing missing values using appropriate techniques.

**Outlier Detection and Handling:** Detect outliers in the dataset and decide on an approach for handling them (e.g., removing outliers or transforming data).

**Data Transformation:** Normalize or scale features if necessary to ensure that features are on a similar scale, which can improve the performance of some machine learning algorithms.

### **Step 4: Implement Machine Learning Models**

Now, let's delve deeper into building machine learning models:

**Feature Extraction and Engineering:**

**Feature Selection:** Carefully choose relevant features from your dataset, as an appropriate feature set is crucial for model performance.

**Feature Engineering:** Create new features or transform existing ones to capture meaningful information from your data.

**Model Implementation:**

**Library Selection:** Utilize Python's machine learning libraries such as Scikit-Learn, TensorFlow, or PyTorch based on your specific requirements and familiarity with the libraries.

**Algorithm Selection:** Experiment with various machine learning algorithms such as Support Vector Machines (SVM), Decision Trees, Random Forests, Gradient Boosting, and Neural Networks.

**Model Training:** Train your selected models using the preprocessed dataset, ensuring that you properly split the data into training and validation sets.

**Hyperparameter Tuning:**

Optimize your model's hyperparameters through techniques like grid search or random search to improve performance.

## **Step 5: Test and Validate**

Evaluating your machine learning models is a critical step:

**Validation Metrics:**

Utilize appropriate evaluation metrics for your classification problem, including but not limited to accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrices.

Implement k-fold cross-validation to assess model performance robustly.

**Fine-Tuning:**

Iterate on your models, adjusting hyperparameters and trying different configurations to achieve the best possible results.

**Model Interpretability:**

Explore methods to interpret your model's predictions, such as feature importance analysis or model explainability tools.

## **Step 6: Documentation and Reporting**

Comprehensive documentation is essential for maintaining, sharing, and replicating your work:

**Software Installation Documentation:**

Create a detailed document that outlines the installation steps for Python, specifying versions of Python, and important libraries such as NumPy, Pandas, and your chosen machine learning frameworks (e.g., Scikit-Learn, TensorFlow, or PyTorch).

**System Documentation:**

Document your entire machine learning system, providing step-by-step instructions for:

Loading and preprocessing data.

Feature extraction and engineering.

Model selection, training, and evaluation.

Hyperparameter tuning.

Model deployment (if applicable).



Include code examples, visualizations, and explanations to help others understand and use your system effectively.

Mention any specific configuration requirements or dependencies.

Provide a clear project structure and file organization to make it easy for others to navigate and contribute to your project.

## **Appendix C - Software Usage Process**

### **[1] Data Input:**

To provide water quality data to the software system, you must ensure that the system supports multiple data input methods. Users can upload dataset files in various formats (e.g., CSV, Excel) or connect to real-time data sources like IoT sensors, remote databases, or APIs. The system should be versatile in accepting data from different sources to cater to a wide range of use cases.

It is crucial to validate and preprocess the incoming data to ensure it aligns with the system's expected input requirements. This involves checking for data integrity, verifying data types, and handling any data format inconsistencies. Users should receive clear instructions on how to format and upload their data, and the system should provide feedback on the data's suitability for processing.

### **[2] Data Preprocessing:**

Data preprocessing is a critical step in preparing the input data for analysis. It involves a series of tasks, including handling missing values by imputing or removing them, identifying and addressing outliers that could distort the analysis, and performing data normalization or standardization to bring data on a consistent scale. This step ensures that the data is clean, reliable, and ready for further analysis.

Moreover, the system should allow users to customize the preprocessing steps based on their specific data characteristics and requirements. It's essential to provide transparency in the preprocessing process, allowing users to review and validate the changes made to their data.

### **[3] Feature Extraction:**

Feature extraction is where the raw water quality data is transformed into a set of relevant features suitable for machine learning model input. The selected feature extraction techniques should be well-documented and explained to users to maintain transparency and facilitate their understanding of the process.

The system should also offer a variety of feature extraction methods, allowing users to experiment with different approaches based on the nature of their data and the problem they're trying to solve.

Consistency in feature extraction techniques used during training and application ensures that the model operates effectively on new data.

#### **[4] Classification Model Application:**

This step involves applying the trained machine learning model to classify water quality data. Users can choose between real-time classification, where the system continuously processes incoming data and provides instant results, or batch classification, suitable for analyzing historical datasets.

The model's predictions should be logged and stored, enabling users to access and analyze historical classifications, track model performance, and assess the impact of any model updates or changes.

#### **[5] Result Presentation:**

Presenting classification results effectively is crucial for user understanding and decision-making. The system should provide various ways to display results, such as through a user-friendly interface with interactive visualizations, command-line output for developers and analysts, or APIs for integration into other systems.

The classification results should not only include class labels but also probabilities or confidence scores associated with each data point. This information empowers users to make informed decisions and assess the reliability of the model's predictions.

#### **[6] System Maintenance:**

Maintaining the software system involves continuous efforts to ensure its accuracy, reliability, and adaptability. Regularly updating the machine learning model is essential to incorporate new data and improve classification performance over time. Users should have the option to trigger model updates manually or set up automated update schedules.

Monitoring system performance is critical, and the system should provide alerts or notifications when issues or anomalies arise. An efficient bug tracking and resolution process ensures that the software remains stable and dependable. Additionally, the system should be designed to adapt to evolving requirements, dependencies, and data sources to ensure its long-term relevance and usability.