ENSET
ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT TECHNIQUE DE MOHAMMEDIA
UNIVERSITÉ HASSAN II DE CASABLANCA
المدرسة العليا لأساتذة التعليم التقني المحمدية

Aqua Sentinel

Department of Mathematics and Computer Science

Master of Distributed Systems and Artificial intelligence

A Machine Learning Project on

Harnessing Machine Learning for Web-Based Water Monitoring

Conducted by:

KHALSS Yassine

BARIK Zakariae

Supervised by:

Pr. HAMIDA Soufiane

2023 - 2024

Table of contents

Table of figures

Table of Tables

Introduction

Water, as we all know, is an indispensable resource crucial for the sustenance of all living beings on our planet. Its significance extends far beyond mere hydration, encompassing essential functions in agriculture, industry, and ecosystem stability. However, the quality of water, the deciding factor of its usability, has undergone a significant evolution over the past few decades, caused by industrialization, urbanization, and agricultural intensification.

Amongst all the complexities of modern existence, the preservation of water quality emerges as an imperative task. With exponentially growing populations and increasing anthropogenic activities, the pressure on water bodies intensifies, leading to a myriad of pollutants infiltrating our once pristine water sources. Furthermore, the filtration of already non suitable water is generally a costly endeavor that requires considerable resources and infrastructure. Thus, forecasting water quality becomes not just a matter of environmental concern, but a fundamental necessity for public health, economic prosperity, and ecological balance.

Forecasting water quality involves predicting the fluctuation of a water system's health at a given point in time based on current observations. Constant assessment of water quality is critical for strategic water management and regulation, in fact, considerable improvements can be made to water pollution averting measures by forecasting forthcoming changes in water usability levels. Thus, in the modern society, it is imperative to explore solutions for real-time monitoring of water quality in order to be able to assess its fluctuation in advance.

Yet, monitoring water quality faces several challenges, including the complexity of water systems, variability in data, limited spatial and temporal coverage, and the need for real-time assessment. Traditional monitoring methods often struggle to provide comprehensive and timely information, leading to gaps in data and hindering effective management strategies. Take the WQI (water quality index) for instance, it is a metric frequently utilized to quantify water quality based on a set of characteristics at a given location and time. Although undoubtedly convenient and useful, the computation of the WQI is intensive and is often influenced by mistakes [1]. As a result, providing a method to accurately and efficiently predict the WQI would be of great benefit. For this reason, it's a good idea to consider machine learning as potential solution. Machine learning offers promising solutions by enabling the analysis of large volumes of data from diverse sources, identifying patterns and trends, and predicting water quality parameters. By leveraging machine learning algorithms, we can enhance the efficiency and accuracy of water quality monitoring, enabling early detection of pollution events, proactive management of water resources, and better protection of ecosystems and public health.

1. Project Background

1.1. Problem statement

Water, an essential resource for sustaining life, highlights the critical need for continuous monitoring to safeguard its quality and ensure its availability for current and future generations. Effective water quality monitoring enables the timely detection of degradation, allowing for prompt interventions to mitigate potential threats to public health and the environment. However, traditional laboratory-based methods, while commonly utilized, pose significant challenges in terms of cost, time efficiency, and the ability to provide real-time data. As such, there is a pressing need to explore alternative approaches that can overcome these limitations and enhance the accuracy and efficiency of water quality monitoring systems.

One major challenge in water quality monitoring is the handling of partial data, which often arises due to factors such as equipment malfunctions, sampling limitations, or gaps in data collection. Conventional methods reliant on complete datasets may struggle to provide accurate assessments in the presence of missing or incomplete information. Moreover, inconsistencies in measurements can further complicate the computation of critical metrics such as the water quality index (WQI), undermining the reliability of monitoring efforts [1]. Addressing these challenges requires innovative solutions capable of effectively handling partial data and ensuring the integrity of water quality assessments even in the face of incomplete information.

In addition to addressing partial data challenges, there is a need to optimize resource allocation in water quality monitoring initiatives. The substantial resources required for rectifying data inconsistencies and maintaining traditional monitoring infrastructure can strain budgets and divert attention from other critical areas of concern. Leveraging advanced technologies such as machine learning (ML) presents an opportunity to enhance resource efficiency and streamline monitoring processes. By automating data analysis, predicting missing values, and optimizing sampling strategies, ML can enable more cost-effective and sustainable approaches to water quality monitoring, ultimately improving the management and allocation of resources in this vital domain.

1.2. Objectives

- **Develop a robust and efficient methodology for real-time monitoring of water quality utilizing machine learning algorithms.** This objective aims to leverage advanced computational techniques to enable continuous monitoring of key water quality parameters and facilitate prompt detection of any deviations from desired standards.

- **Implement a machine learning-based approach for predicting real-time water quality index (WQI) values using a set of measured parameters.** By employing predictive modeling techniques, this objective seeks to provide accurate and timely assessments of water quality, empowering specialists to make informed decisions regarding water resource management and conservation efforts.

- **Provide a suite of tools designed to visualize water physiochemical parameters over time in a clear and user-friendly manner.** This objective aims to enhance data interpretation and analysis by offering intuitive visualization tools that enable users to identify trends, anomalies, and patterns in water quality data with ease.

- **Develop a comprehensive platform that enables users to access and track the history of measurements and predictions over time.** This objective aims to create a centralized hub where users can securely store, retrieve, and analyze historical water quality data, facilitating data-driven decision-making and promoting transparency and accountability in water resource management practices.

1.3. Scope

In order to effectively execute the project and concentrate on essential objectives, it is crucial to define a realistic scope. Ideally, real-time monitoring of water quality would entail utilizing live feeds of water physiochemical parameters captured by sensors deployed across various locations. However, due to budget constraints and resource limitations, this approach is not feasible within the scope of this project. Instead, we will utilize an available dataset comprising measurements of several physiochemical parameters from multiple Indian water bodies. Additionally, we will assign timestamps to these measurements to simulate continuous data capture, akin to sensor-based monitoring. While this approach may not fully replicate real-world conditions, it will serve as a proof of concept for our objectives. We acknowledge that this method may have limitations regarding accuracy and correspondence to real-life scenarios, but it provides a practical foundation for demonstrating the efficacy of our proposed methodologies.

1.4. Expected result

Aligned with the defined objectives and scope of the project, our aim is to develop a web application that streamlines the process of water monitoring. The resulting application, named "**Aqua Sentinel**", is envisioned as a modern and comprehensive tool that leverages machine learning algorithms to provide real-time predictions of the water quality index (WQI). Beyond this primary functionality, Aqua Sentinel will offer a range of features to support users in monitoring and analyzing water quality data effectively.

Aqua Sentinel's interface will prioritize simplicity, modernity, and intuitiveness, ensuring that users can easily navigate and utilize its various tools and functionalities. The application will provide users with the ability to visualize the evolution of different water physiochemical parameters over time through interactive charts and graphs. Additionally, Aqua Sentinel will offer statistical information on these parameters, empowering users with detailed insights into water quality trends and patterns.

2. Materials and Methodology

When employing machine learning for water quality prediction, two prominent methodologies emerge. The first approach involves the direct prediction of the water quality index (WQI), essentially framing the task as a regression problem. Conversely, the second approach focuses on predicting the water quality class, requiring the interpretation of the WQI as a classification task. For this project, we have opted to pursue the former approach, where our objective is to predict the WQI directly and subsequently interpret the obtained results.

2.1. Proposed methodology

The project workflow begins with the exploration and preprocessing of the data to ensure its quality and readiness for analysis. Once the data is prepared, a suitable machine learning algorithm is carefully selected and employed to train a model. Following this, cross-validation techniques are utilized to assess the performance of the trained model and determine if it has achieved its predefined objectives. If the model meets the desired criteria, it proceeds to the testing and evaluation phase. However, if the model falls short of expectations, further refinement is undertaken by fine-tuning its hyperparameters to optimize its performance. For clarity, a flow chart of the process is given below.

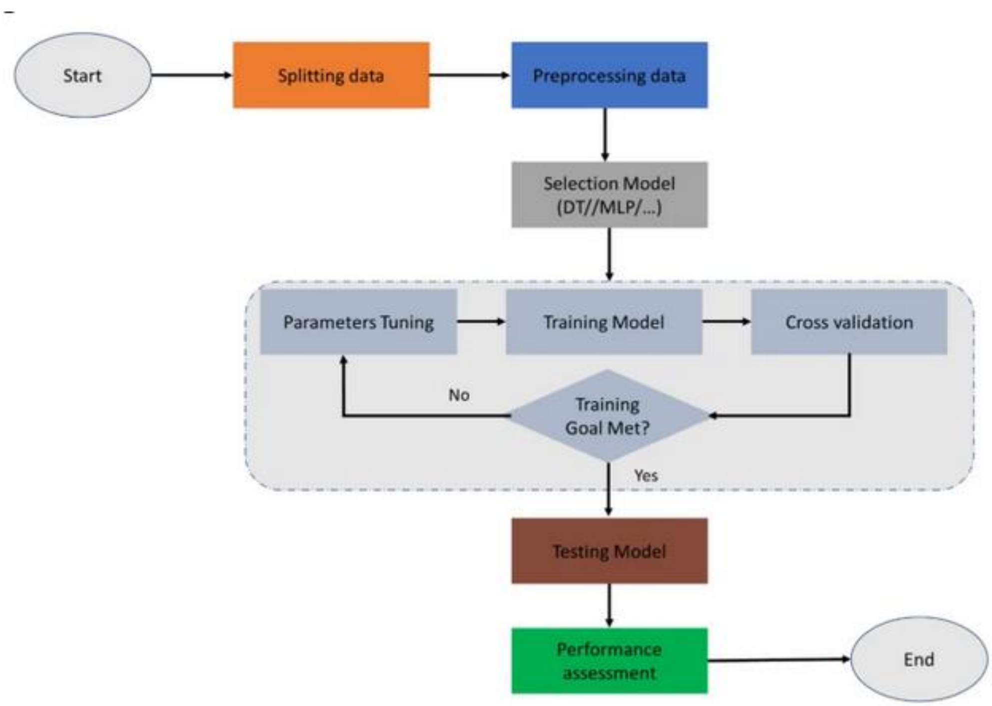


Figure 1: Project workflow flowchart [1] [2]

In this work, four of the common Regression algorithms, namely, KNN, DT, SVR and MLP are trained, tested and evaluated in order to ascertain to best one at predicting the WQI.

2.2. Regression models for predicting WQI

In this section, we will be giving a brief overview of the selected algorithms.

2.2.1. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a versatile machine learning algorithm used for classification and regression tasks. During training, KNN memorizes the entire dataset. For prediction, it calculates the distance between the new data point and all training points, then selects the K nearest neighbors. In regression, it averages their target values. The value of K, determining the number of neighbors, is crucial and often chosen empirically. The distance is calculated using the following equation:

$$D = \sqrt{(x_1 - x_2) + (y_1 - y_2)}$$

Where x_1, x_2, $y_1$and y_2 are parameters for data input.

While KNN is simple and intuitive, it can be computationally expensive with large datasets due to distance calculations for every prediction.

2.2.2. Decision Tree (DT)

Decision Tree is a versatile machine learning algorithm used for both classification and regression tasks. During training, the algorithm recursively splits the data into subsets based on the feature that provides the best separation according to a certain criterion, such as Gini impurity or information gain. This process continues until a stopping criterion is met, such as reaching a maximum depth or when further splits do not improve the model's performance. For prediction, the algorithm traverses the tree from the root node to a leaf node, where it assigns the majority class (for classification) or the average target value (for regression) of the training instances in that node. Decision trees are easy to interpret and understand, and they can handle both numerical and categorical data. However, they are prone to overfitting, especially with complex or noisy datasets, and may not generalize well to unseen data.

2.2.3. Support Vector Regression (SVR)

Support Vector Regression (SVR) is a machine learning algorithm designed for regression tasks. It extends the Support Vector Machine (SVM) algorithm, aiming to find a function that best fits the training data while maintaining a maximum margin of tolerance. During training, SVR identifies support vectors, which are data points closest to the hyperplane defining the margin. It optimizes to minimize the error between predicted and actual target values, with a user-defined tolerance ε. For prediction, SVR calculates the distance of new data points from the hyperplane and predicts their target values accordingly. SVR can handle both linear and nonlinear relationships between features and target variables by utilizing different kernel functions, such as Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid. While effective for capturing complex patterns and outliers, SVR's performance depends on proper tuning of hyperparameters like C and kernel parameters to prevent overfitting. Overall, SVR is widely used in regression tasks across various domains.

2.2.4. Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron (MLP) is a versatile neural network used for classification and regression tasks. It consists of input, hidden, and output layers, with neurons interconnected by weighted edges. During training, MLP adjusts weights using backpropagation to minimize prediction errors. Non-linear activation functions like sigmoid or ReLU introduce complexity. Output layer activation depends on the task. Techniques like dropout or L2 regularization prevent overfitting. Hyperparameters like layer size and learning rate require careful tuning. MLPs excel at learning complex patterns but can be computationally intensive. Despite this, they find widespread use in various domains.

2.3. Dataset

2.3.1. Overview

The dataset utilized in this study comprises data collected from lakes and rivers across various locations in India spanning the period from 2005 to 2014. Collected by the government of India, this dataset aims to ensure the water's suitability for drinking purposes. It encompasses 1991 instances and encompasses 7 distinct features [1]. These features include:

- **Dissolved Oxygen**, which measures the level of oxygen dissolved in the water critical for supporting aquatic life;
- **pH**, indicating the water's acidity or alkalinity level;
- **Conductivity**, evaluating its electrical conductivity and dissolved solids presence;
- **Biological Oxygen Demand (BOD),** gauging the oxygen absorbed by microorganisms, revealing organic contamination extent;
- **Nitrate**, examining nitrate ion concentration, potentially indicating fertilizer or sewage pollution;
- **Fecal Coliform**, reflecting fecal pollution presence through coliform bacteria detection;
- **Total Coliform**, representing the total coliform bacteria count, encompassing both fecal and non-fecal sources.

2.3.2. Preprocessing

In order to ensure the dataset's quality and suitability for machine learning applications, a series of preprocessing steps were conducted. Initially, all data types were verified and converted into numerical formats to facilitate analysis and modeling. Addressing missing numerical values, a **median imputation** strategy was employed to replace them, ensuring the integrity of the dataset while preserving its statistical properties. Furthermore, to enhance the robustness of the dataset, outliers were identified and removed using the **Z-score method**, mitigating their potential impact on the analysis and model performance. Finally, measurements resulting in a negative water quality index (WQI) were also identified and dropped from the dataset.

2.4. Water quality index

The Water Quality Index (WQI) condenses multiple water quality parameters into a single numerical value, simplifying assessment and enabling comparative analysis across different locations and time periods. It aids decision-making in water resource management, pollution control, and public health by providing valuable information for prioritizing actions and allocating resources effectively.

The WQI is computed using the following equation:

$$WQI = \frac{\sum_{i=1}^{N} q_i \times w_i}{\sum_{i=1}^{N} w_i}$$

Where N represents the number of parameters, q_i represents the quality rating scale for the parameter i and w_i reprenests the unit weight for the parameter i.

q_i is calculated using the following equation:

$$q_i = 100 \times \left(\frac{v_i - v_{id}}{s_i - v_{id}}\right)$$

Where v_i represents the estimated value for the parameter i, v_{id} represents and ideal value for the parameter I while the water is pure, and s_i, represents a standard value for the parameter i.

The unit weight w_i is computed using the following equation:

$$w_i = \frac{k}{s_i}$$

Where k represents the constant of proportionality and computed using the equation:

$$k = \frac{1}{\sum_{i=1}^{N} s_i}$$

The following table presents the calculated weight unit, the standard value and the ideal for each of the dataset's features used the compute the Water Quality Index.

Feature Name	**Unit Weight**	**Standard Value**	**Ideal Value**
Dissolved oxygen	0.2213	10	14.6
pH	0.2604	8.5	7
Conductivity	0.0022	1000	0
Biological oxygen	0.4426	5	0
Nitrate	0.0492	45	0
Fecal coliform	0.0221	100	0
Total coliform	0.0022	1000	0

Table 1: The calculated weight unit, standard value and ideal value of the dataset's features

The distribution of the calculated WQI for our dataset is as follows:

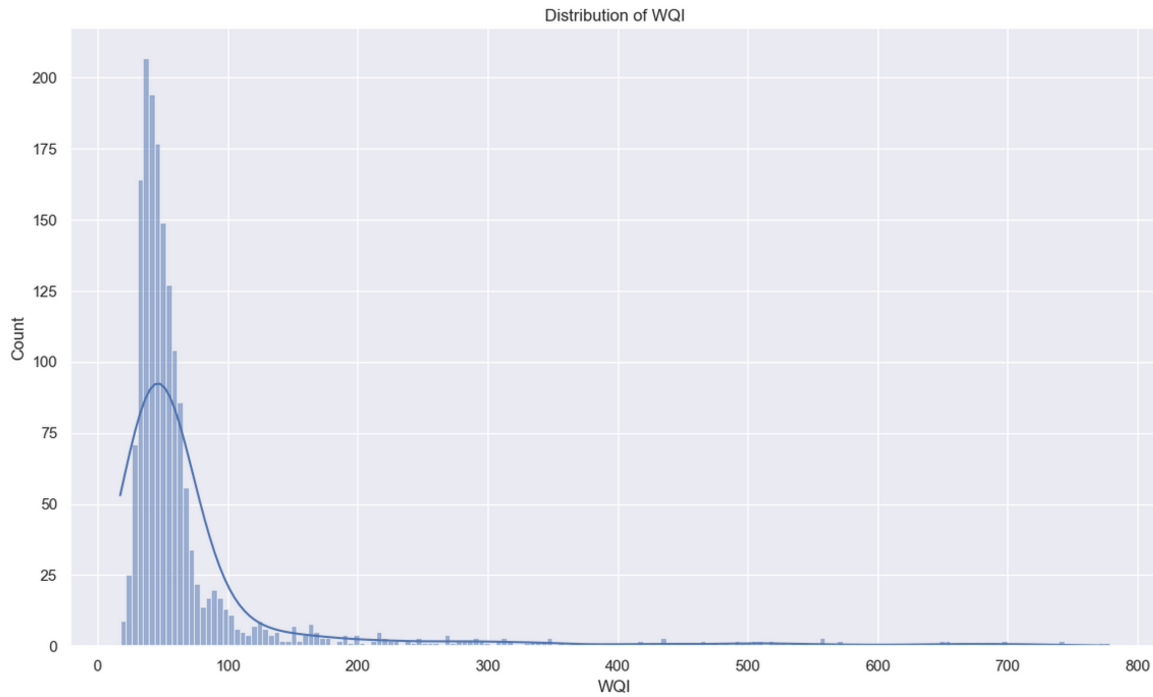


Figure 2: The distribution of WQI values

The calculated Water quality indices once interpreted give the following numbers and ration of corresponding water quality classes:

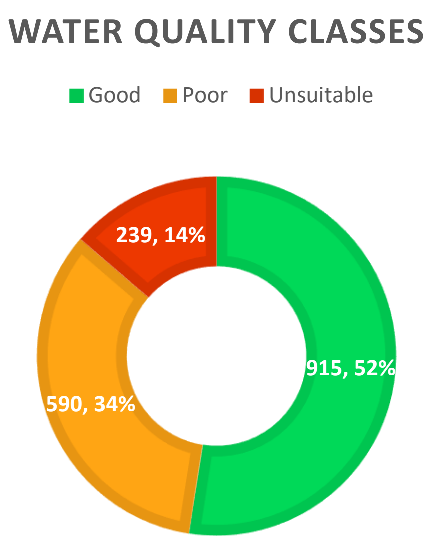


Figure 3: WQC values and ratios

2.5. Model performance evaluation

Performance evaluation of the regression models is conducted using several criteria, including **Mean Absolute Error (MAE)**, **Median Absolute Error (MedAE)**, **Mean Square Error (MSE)**, and **Coefficient of Determination (R2)**. These metrics provide insights into the effectiveness of the regression models in predicting water quality index (WQI). Additionally, akin to the accuracy metric used in classification tasks, the predicted WQI is interpreted to determine the corresponding water quality class, which is then compared to the actual class. To differentiate this metric from the accuracy typically associated with classification, it is termed as **Interpreted Accuracy (IAccu)**. By assessing both traditional regression metrics and the interpreted accuracy, a comprehensive understanding of the regression model's performance in predicting water quality is attained.

The MSE is calculated using Equation 1:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_{real_i} - y_{pred_i}\right)^2$$

Equation 1

The MAE is calculated using Equation 2:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_{real_i} - y_{pred_i}\right|$$

Equation 2

The MedAE is calculated using the following Equation 3:

$$MedAE = median\left(\left|y_{real_1} - y_{pred_1}\right|, \ldots\ldots, \left|y_{real_N} - y_{pred_N}\right|\right)$$

Equation 3

The IAccu, much like the standard Accuracy metric is calculated using Equation 4:

$$IAccu = \frac{TP + TN}{TP + FP + FN + TN}$$

Equation 4

3. Results

The experiments are conducted using **Jupyter Notebook** version 6.4.6, which facilitates the execution and creation of Python scripts. Jupyter Notebook is renowned for its user-friendly interface and is extensively utilized as an open-source platform for implementing and executing AI and ML models. Additionally, we will be utilizing **sklearn**, a popular Python library for machine learning, in our experiments.

The table below summarizes the tuning parameters explored for each regression model, along with the specific parameter values that yielded the highest performance during the tuning phase (Best parameters). These optimal parameters are pivotal in refining the models to ensure precise regression predictions.

Model	Tuning parameters	Best parameters
MLP	**hidden_layer_sizes :** [(50,), (100,), (50, 50), (100, 50)] **activation :** [relu, tanh, logistic] **solver :** [adam, sgd] **alpha :** [0.0001, 0.001, 0.01] **learning_rate :** [constant, adaptive]	**activation :** relu **alpha :** 0.01 **hidden_layer_sizes :** (50,) **learning_rate :** adaptive **solver :** sgd
SVR	**C :** [1,2,3,4,5] **kernel :** [sigmoid, linear, ploy, rbf] **epsilon :** [0.1, 0.01, 0.001]	**C :** 5, **epsilon :** 0.001 **kernel :** linear
DT	**max_depth :** [from 1 to 3], **random_state :** [from 1 to 50]	**max_depth :** 14, **random_state :** 47
KNN	**n_neighbors :** [from 1 to 50] **weights :** [uniform, distance]	**n_neighbors :** 2 **weights :** 'distance'

Table 2: Tested and best parameters for regression models using Grid Search method

Table 3 provides an overview of the performance metrics of various regression models obtained through the grid search method. These models encompass the KNN regressor model, DT regressor model, SVR model, and the proposed MLP regressor model. Notably, the proposed SVR model emerges as the top performer among these models. It achieves exceptional results, boasting a Mean Absolute Error (MAE) of 7.04 × 10-4, Mean Squared Error (MSE) of 5.43×10-7, Median Absolute Error (MedAE) of 7.69×10-4, and an R-squared (R2) value of 99%. Furthermore, it attains a perfect Interpreted Accuracy score (IAccu) of 1.0, highlighting its superior predictive capabilities compared to other regression models.

Model	MSE	MAE	MedAE	R^2	IAccu
MLP	3.13×10^{-2}	0.11	7.6×10^{-2}	0.99	1.0
SVR	$\mathbf{5.43\times10^{-7}}$	$\mathbf{7.04\times10^{-4}}$	$\mathbf{7.69\times10^{-4}}$	**0.99**	**1.0**
DT	351.43	6.79	2.54	0.98	0.91
KNN	1175.98	11.65	3.56	0.94	0.83

Table 3: Performance of the four regression models

Figure 2 presents scatter plots comparing test values and their predictions for each of the four regression models. Each point on the scatter plot represents a test value-prediction pair. The point's position shows how the model's prediction compares to the actual test value. Points away from the red perfect prediction line show differences between predictions and actual values. Points above overestimate, below underestimate.

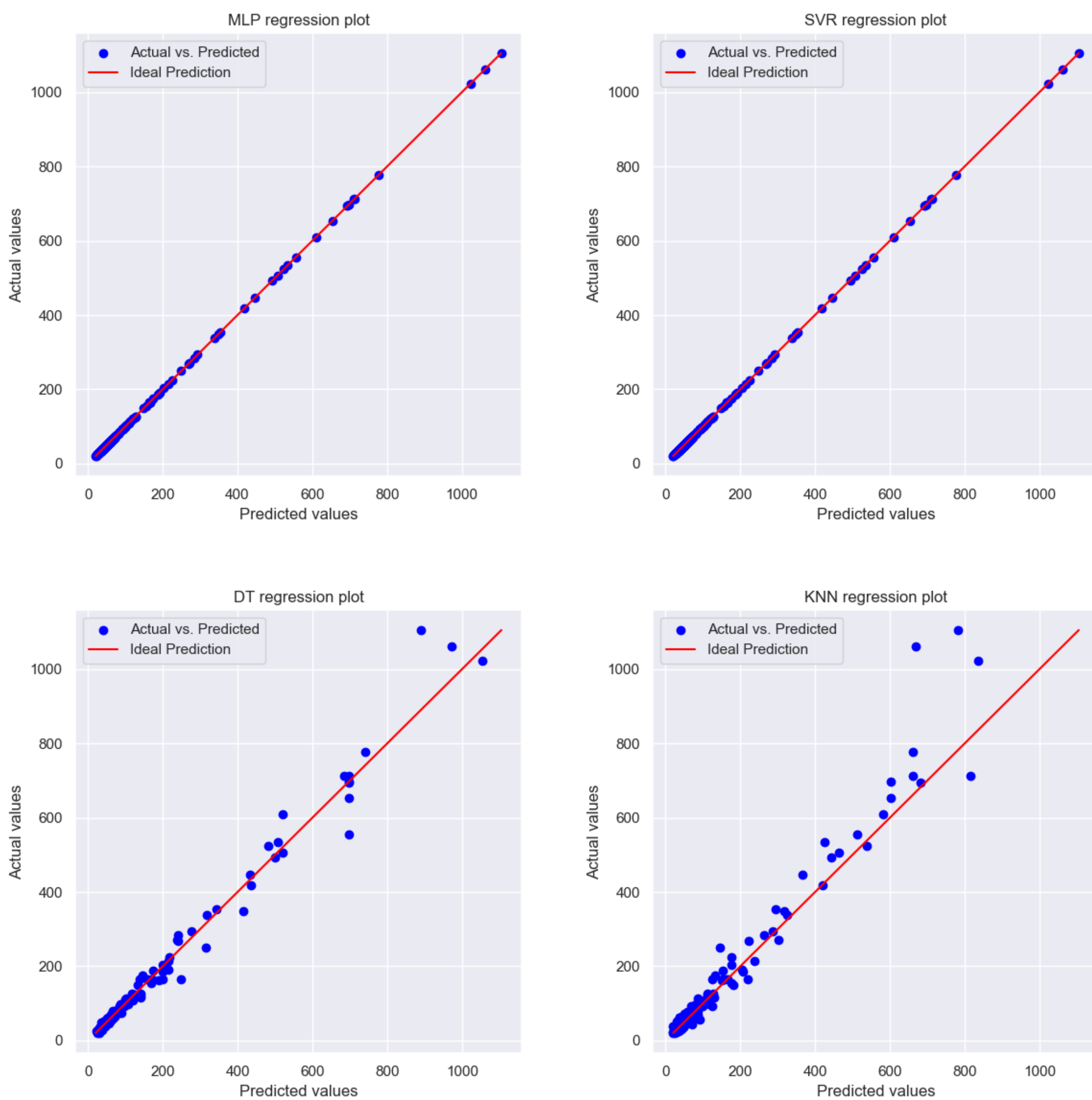


Figure 4: Regression plots for the four regression models

4. Implementation

4.1. Tools and technologies

In our project, we utilized a combination of **React** and **Flask** to develop our web application. React was employed for the front end, while Flask served as the backend, housing the Python code responsible for implementing our machine learning algorithms and managing the dataset.

4.1.1. React

Figure 5: React logo

React is a popular JavaScript library for building user interfaces, particularly for web applications. It was developed by Facebook and has gained widespread adoption due to its efficiency, simplicity, and component-based architecture. React provides a powerful and efficient way to build dynamic and interactive user interfaces for web applications. Its component-based architecture, virtual DOM, declarative syntax, and state management capabilities make it a popular choice for front-end development [2].

4.1.2. Flask

Figure 6: Flask logo

Flask is a lightweight and flexible web application framework for Python. It is designed to make it easy to build web applications quickly and with minimal boilerplate code. Flask is often referred to as a "micro" framework because it focuses on simplicity and does not enforce any particular way of structuring your application [3].

4.1.3. Tailwind

Figure 7: Tailwind css logo

Tailwind CSS is a utility-first CSS framework that revolutionizes the way developers approach styling web applications. Unlike traditional frameworks that rely on pre-defined components, Tailwind empowers developers to build custom designs by composing utility classes directly into their HTML markup. With a comprehensive set of utility classes covering everything from layout and typography to spacing and colors, Tailwind enables rapid prototyping and consistent styling across projects [4].

4.2. Application showcase

4.2.1. The Dashboard view

The Dashboard tab offers an interactive interface featuring five unique data visualization tools. These tools allow users to analyze and visualize fluctuations in water quality across time. Within this section, users can obtain a comprehensive summary of crucial physicochemical parameters, aiding in swift and intuitive evaluations of water condition.

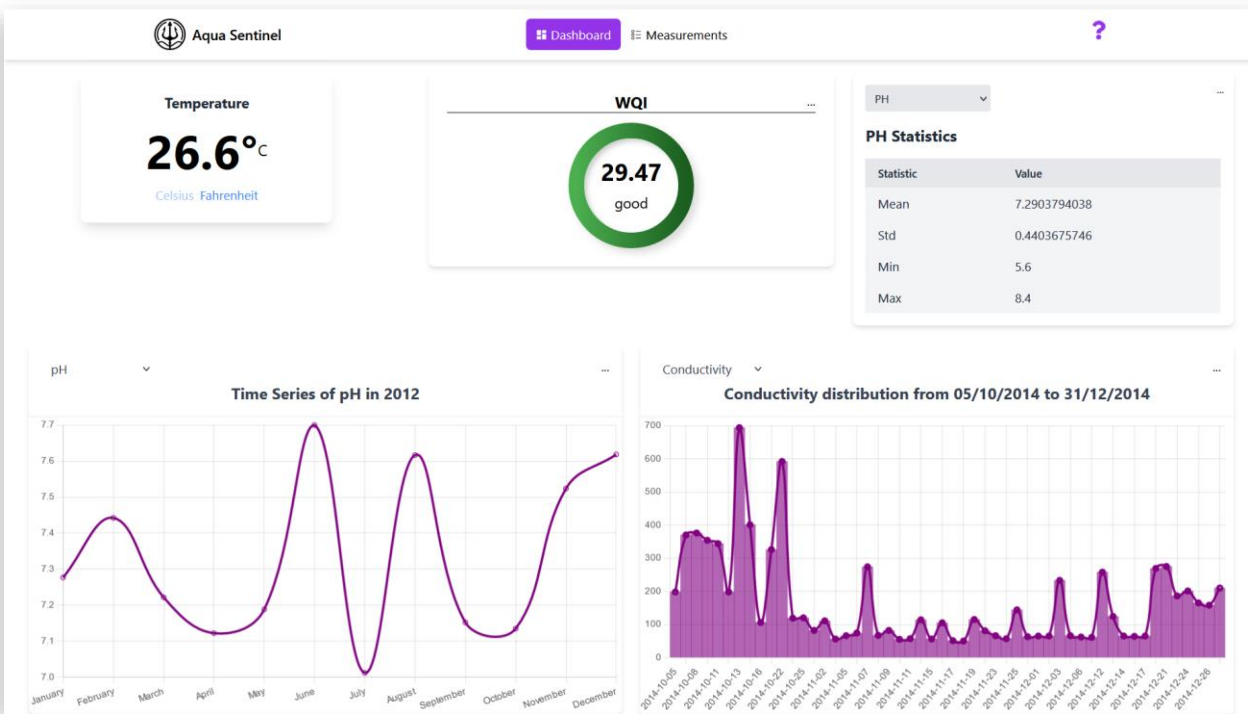


Figure 8: Preview of the dashboard tab of Aqua Sentinel

Every widget can be personalized to display different water parameters at specific time points and intervals. This functionality empowers users to finely adjust their analysis according to their monitoring needs.

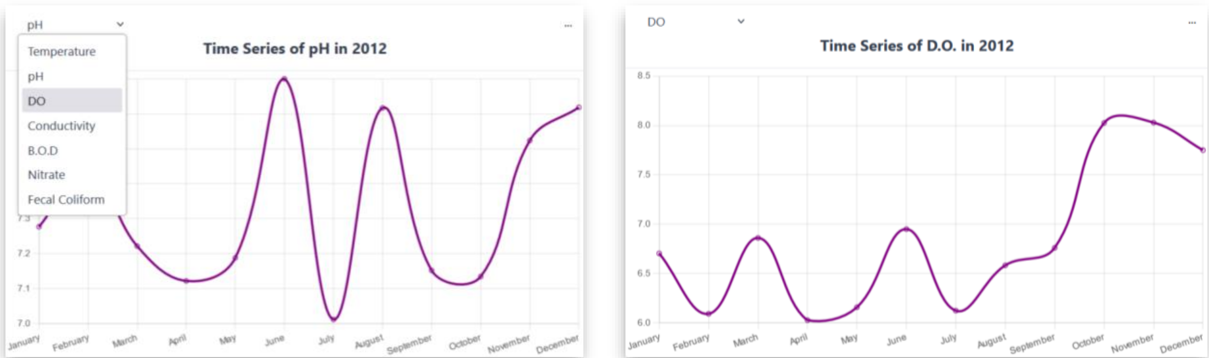
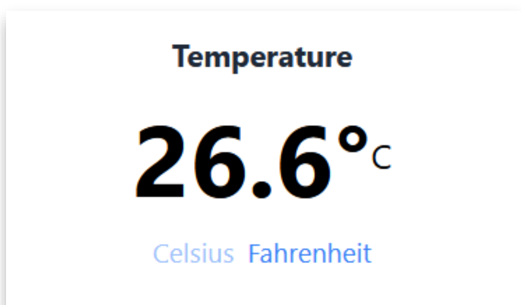


Figure 9: Demonstration of Customization

4.2.1.1. Temperature Widget



The Temperature widget provides users with live updates on the present water temperature in Celsius. Users can toggle between Celsius and Fahrenheit measurements according to their preference, as well as view temperatures at various time points, ensuring flexibility in observing temperature data.

Figure 10: Temperature Widget

4.2.1.2. WQI Widget

The WQI (Water Quality Index) widget provides users with an instant overview of the current water quality index. Employing Machine Learning algorithms, specifically SVR (Support Vector Regression), the WQI predicted rather than calculated. Users also have the ability to delve into historical WQI data by selecting a specific date from the drop-down menu located at the top right corner of the widget.

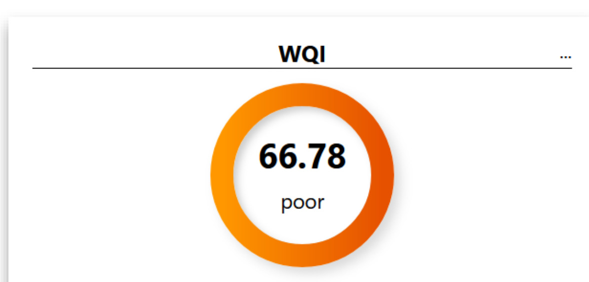
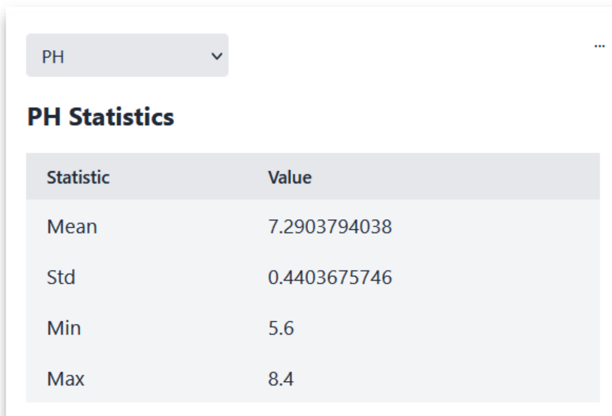


Figure 11: WQI Widget

4.2.1.3. Statistics Widget

The statistics widget offers users an interactive platform for effortless exploration of statistical data. Through a dropdown menu, users can select various statistical metrics, and the widget instantly updates to showcase the relevant data. Additionally, it provides a functionality to delve into detailed statistics for each metric via a pop-up window, fostering user engagement and facilitating thorough data analysis.



Statistic	Value
Mean	7.2903794038
Std	0.4403675746
Min	5.6
Max	8.4

Figure 12: Statistics Widget

4.2.1.4. Yearly Time Series Widget

The Yearly Time Series widget graphically illustrates the yearly progression of a chosen water parameter, allowing users to visualize trends and patterns over time. By selecting options from the drop-down menu situated at the top left of the widget, users can explore different water parameters, thus gaining a comprehensive perspective on temporal fluctuations in water quality.

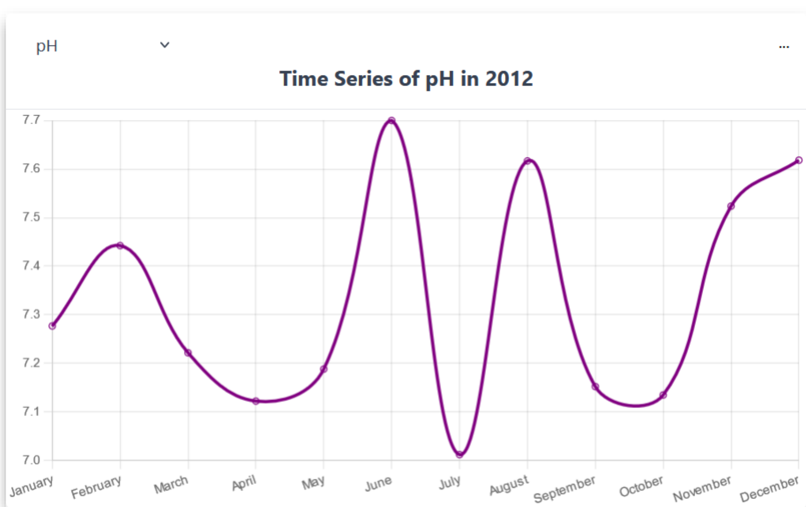


Figure 13: Yearly Time Series Widget

4.2.1.5. Time Series Widget

The Time Series widget showcases the progression of a designated water parameter within a user-selected interval, providing valuable insights into its fluctuations over time. Users can analyze trends across various water parameters by choosing options from the drop-down menu located at the top left of the widget. Moreover, users can refine their analysis by adjusting the displayed time interval through the drop-down menu at the top right, offering additional customization and control over the visualization.

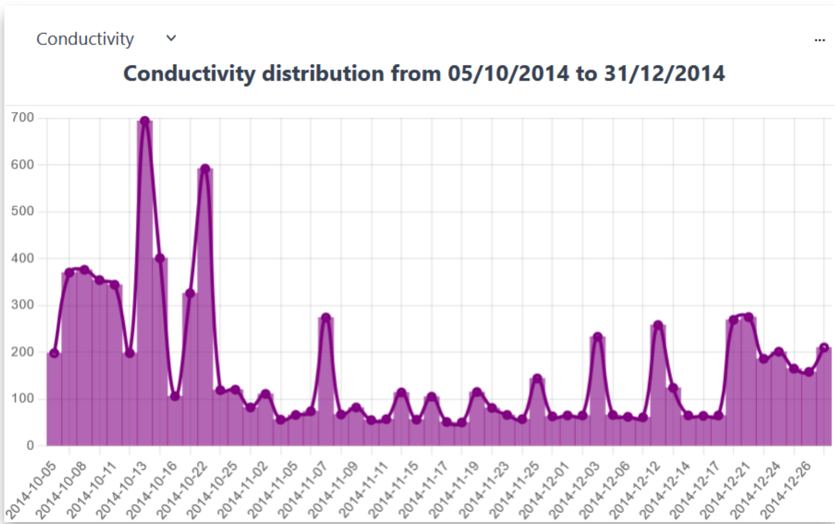
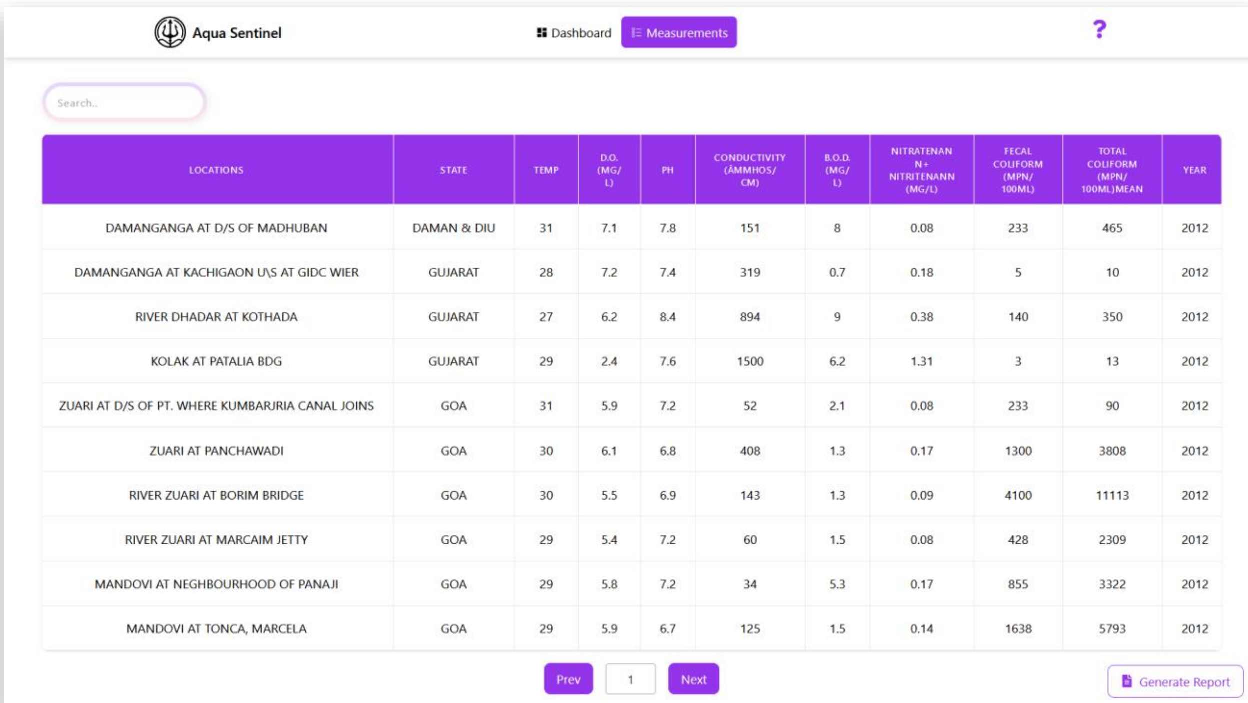


Figure 14: Time Series

4.2.2. The Measurements view

The Measurements tab in Aqua Sentinel provides users with an extensive list of precise measurements, providing specific data points for thorough examination and comparison. Users can seamlessly navigate historical data using timestamps or keywords, streamlining the retrieval of pertinent information. This feature empowers users to delve deeply into the dataset, uncovering trends and patterns across time. Additionally, users can opt to save the list of measurements in PDF format, simplifying the sharing and documentation of discoveries. With the Measurements tab, users gain access to detailed data and insights crucial for effectively supporting their water quality monitoring endeavors.



LOCATIONS	STATE	TEMP	D.O. (MG/L)	PH	CONDUCTIVITY (ÂMMHOS/CM)	B.O.D. (MG/L)	NITRATENAN N+ NITRITENANN (MG/L)	FECAL COLIFORM (MPN/100ML)	TOTAL COLIFORM (MPN/100ML)MEAN	YEAR
DAMANGANGA AT D/S OF MADHUBAN	DAMAN & DIU	31	7.1	7.8	151	8	0.08	233	465	2012
DAMANGANGA AT KACHIGAON U\S AT GIDC WIER	GUJARAT	28	7.2	7.4	319	0.7	0.18	5	10	2012
RIVER DHADAR AT KOTHADA	GUJARAT	27	6.2	8.4	894	9	0.38	140	350	2012
KOLAK AT PATALIA BDG	GUJARAT	29	2.4	7.6	1500	6.2	1.31	3	13	2012
ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOINS	GOA	31	5.9	7.2	52	2.1	0.08	233	90	2012
ZUARI AT PANCHAWADI	GOA	30	6.1	6.8	408	1.3	0.17	1300	3808	2012
RIVER ZUARI AT BORIM BRIDGE	GOA	30	5.5	6.9	143	1.3	0.09	4100	11113	2012
RIVER ZUARI AT MARCAIM JETTY	GOA	29	5.4	7.2	60	1.5	0.08	428	2309	2012
MANDOVI AT NEGHBOURHOOD OF PANAJI	GOA	29	5.8	7.2	34	5.3	0.17	855	3322	2012
MANDOVI AT TONCA, MARCELA	GOA	29	5.9	6.7	125	1.5	0.14	1638	5793	2012

Figure 15: Preview of the measurements view of Aqua Sentinel

4.2.3. The Help section

The Help section of our website serves as an introduction to our platform, offering users an insight into our mission and objectives. It aims to provide clarity and guidance to users, ensuring they can efficiently navigate and utilize the web application. Through a quick starter guide, users can familiarize themselves with the various features and functionalities available, empowering them to make the most of their experience on our platform. Whether it's understanding our mission or getting started with the web app, the Help section is designed to support users every step of the way.

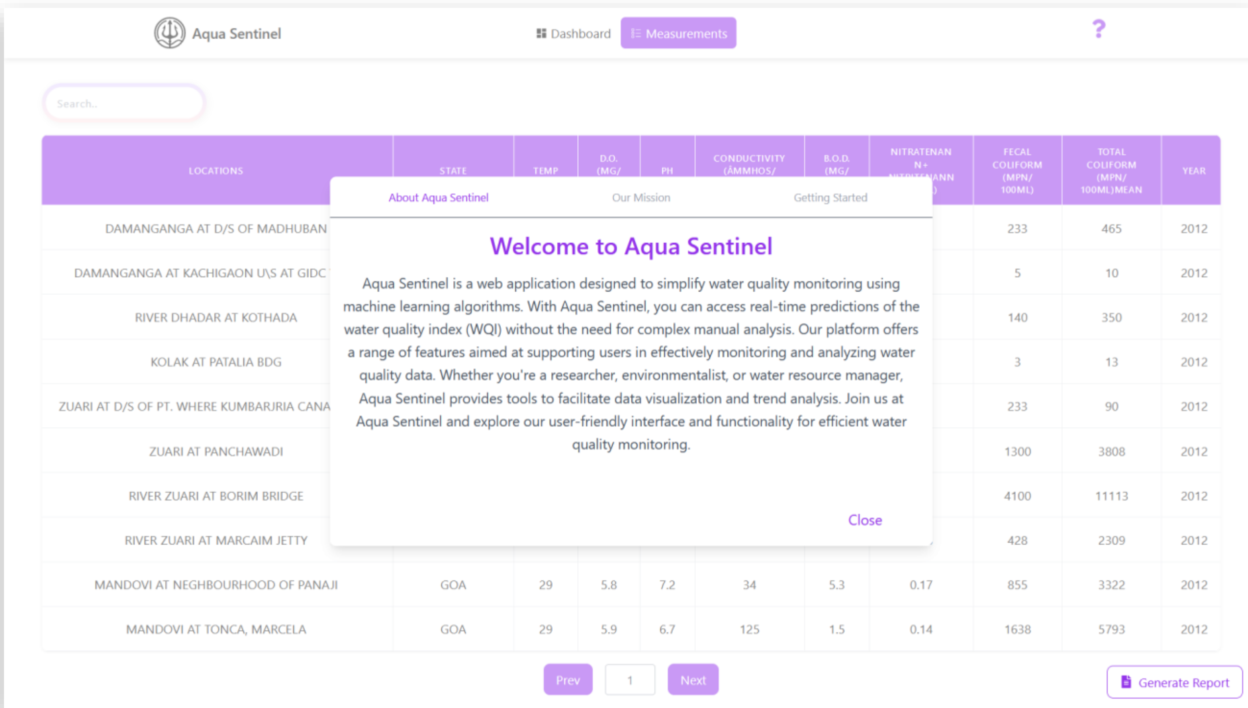


Figure 16: Preview of the Help Section of Aqua Sentinel

Conclusion and Horizons

In conclusion, this project has demonstrated the effectiveness of real-time water quality monitoring using machine learning algorithms. By leveraging data from diverse sources and employing advanced regression models, we have successfully developed a system capable of predicting water quality index (WQI) with high accuracy. The evaluation of various regression models, including KNN, Decision Tree, Support Vector Regression (SVR), and Multi-Layer Perceptron (MLP), revealed that the SVR model outperforms others, achieving remarkable results in terms of MAE, MSE, MedAE, R2, and IAccu. This indicates the potential of SVR in accurately predicting water quality in real-time scenarios. Furthermore, the implementation of a user-friendly web application, Aqua Sentinel, provides users with intuitive tools for visualizing and analyzing water physicochemical parameters over time, facilitating proactive management strategies. Overall, this project contributes to the advancement of water resource management and underscores the significance of integrating machine learning techniques in real-time water quality monitoring systems.

Looking ahead, there are several exciting prospects for expanding and refining this project. Firstly, delving into the prediction of water quality using satellite data offers the opportunity to gain insights into broader geographical trends and long-term environmental impacts. Incorporating alert systems that activate when irregularities in water quality occur can enable swift responses and proactive management strategies to address potential risks promptly. Additionally, enhancing the web application by introducing more widgets, tools, and customization options can empower users to visualize and analyze water quality data more comprehensively. Allowing users to personalize the interface by adding, removing, and rearranging elements further enhances the platform's flexibility and usability, catering to diverse user preferences and requirements. These avenues for development promise to advance the project's capabilities and contribute to more effective real-time water quality monitoring and management.

Webography

[1] A. M. E. E.-S. M. E.-k. A. I. F. M. T. &. Z. T. Mahmoud Y. Shams, "Springer Link," [Online]. Available: https://link.springer.com/article/10.1007/s11042-023-16737-4.

[2] "WikiPedia," [Online]. Available: https://en.wikipedia.org/wiki/React_(JavaScript_library).

[3] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Flask_(web_framework).

[4] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Tailwind_CSS.