# Project Title

*A project report submitted to ICT Academy of*

*Kerala in partial fulfillment of the requirements*

*for the certification of*

## CERTIFIED SPECIALIST

## IN

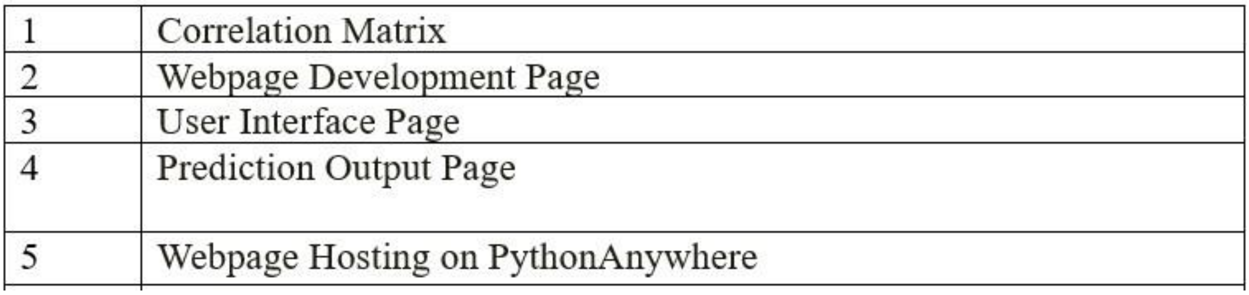
## DATA SCIENCE & ANALYTICS

Submitted by

**Aishwarya Narayanan**
**Aswini**

ICT ACADEMY OF KERALA
**THIRUVANANTHAPURAM, KERALA, INDIA**
**January 2024**

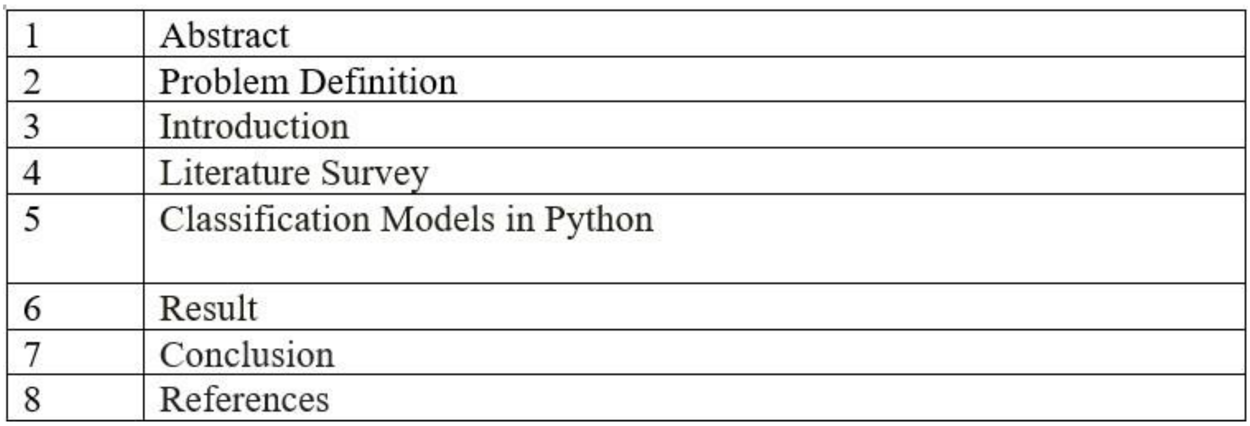
# List of Figures



# List of Abbreviations

Abstract,Problem Definition,Introduction,Literature Survey,Classification Models,Results,Prediction,Output,Conclusion,References

# Table of Contents



# Abstract

## Environmental Pollution Analysis

***Introduction***

This ‘Data Science and Analytics’ project focuses on the analysis of water quality in an urban environment, leveraging a simulated dataset comprising of 21 attributes. The dataset is meticulously crafted to facilitate binary classification, distinguishing between safe and not safe water conditions based on predefined threshold values for various water quality components.

***Dataset Overview***

**Attributes**:

20 water quality components such as aluminium, ammonia, arsenic, etc.

'is_safe' - Binary class attribute indicating water safety (0 - not safe, 1 - safe).

***Objective***

The primary aim is to create an effective binary classification model that can accurately determine the safety of water based on its composition.

***Conclusion***

The project contributes to understanding the interplay of various water quality components and their impact on safety. The resulting classification model serves as a valuable tool for educational contexts, empowering learners to apply data science techniques to real-world scenarios.

# Problem Definition

## 1.1 Overview

Water is a precious and essential resource that sustains life on Earth. Access to clean and safe water is fundamental to human health, economic development, and the well-being of ecosystems. As the global population continues to grow and human activities intensify, the importance of ensuring water quality becomes paramount. This essay explores the significance of water quality, the challenges it faces, and the measures required to maintain and improve the quality of this vital resource.

*Importance of Water Quality:*

Water quality refers to the chemical, physical, and biological characteristics of water that determine its suitability for various uses. Clean and safe water is essential for drinking, agriculture, industrial processes, and maintaining aquatic ecosystems. Contaminated water poses significant risks to human health, leading to waterborne diseases and environmental degradation. Therefore, ensuring water quality is crucial for promoting public health, supporting sustainable development, and preserving biodiversity.

## 1.2 Problem Statement

### **Problem Statement:**

In contemporary society, ensuring the safety of water sources is a critical challenge, considering the increasing threats posed by industrial discharges, agricultural runoff, and urbanization. The presence of various contaminants, such as barium, ammonia, chlorine, and other minerals, has a profound impact on water quality. The need for an accurate and efficient water quality analysis system that classifies water as safe or unsafe based on the concentrations of these specific minerals is evident.

Developing a robust classification model that utilizes advanced analytical techniques to assess and categorize water quality is essential for safeguarding public health and promoting sustainable water management practices. This problem statement seeks to address the pressing need for a comprehensive water quality analysis system that focuses on specific minerals known to have significant implications for human and environmental well-being.

## Introduction

The dataset at hand represents a comprehensive collection of water quality measurements, encompassing a wide array of chemical parameters critical for assessing the safety and potability of water. With a total of 7999 entries and 21 columns, each entry corresponds to a specific water sample, while the columns capture diverse attributes associated with the composition of water.

The dataset includes various quantitative measures, expressed as floating-point numbers, and two categorical variables. The quantitative features span a spectrum of contaminants and essential elements, such as aluminium, arsenic, barium, cadmium, chloramine, chromium, copper, fluoride, bacteria, viruses, lead, nitrates, nitrites, mercury, perchlorate, radium, selenium, silver, uranium, and more.

Notably, the presence of these substances is critical in determining the overall quality of water, and their concentrations can have significant implications for human health and environmental sustainability. The inclusion of both organic and inorganic elements, as well as microorganisms, reflects the dataset's comprehensiveness in addressing the multifaceted nature of water quality assessment.

Two columns, 'ammonia' and 'is_safe,' are of object data type, indicating potential categorical or string values. The 'ammonia' column may represent levels of ammonia in the water, while 'is_safe' likely indicates whether the water sample meets safety standards, possibly denoted as 'safe' or 'unsafe.'

This dataset serves as a valuable resource for researchers, environmental scientists, and policymakers involved in water quality analysis. By leveraging the information contained within, stakeholders can gain insights into the distribution of contaminants, trends in water quality, and potential correlations among different parameters. Effective utilization of this dataset holds the promise of advancing our understanding of water quality dynamics and, consequently, contributes to informed decision-making for the protection and sustainable management of water resources.

# Literature Survey

Water quality analysis plays a pivotal role in safeguarding public health, protecting ecosystems, and ensuring sustainable water resource management. With the increasing complexity and volume of water quality data, the application of classification techniques has become essential for efficient analysis and decision-making. This literature survey explores the current state of research on classification methodologies employed in water quality analysis, shedding light on the advancements, challenges, and emerging trends in this critical field.

## 3.1 Water Quality Analysis and Classification Datasets

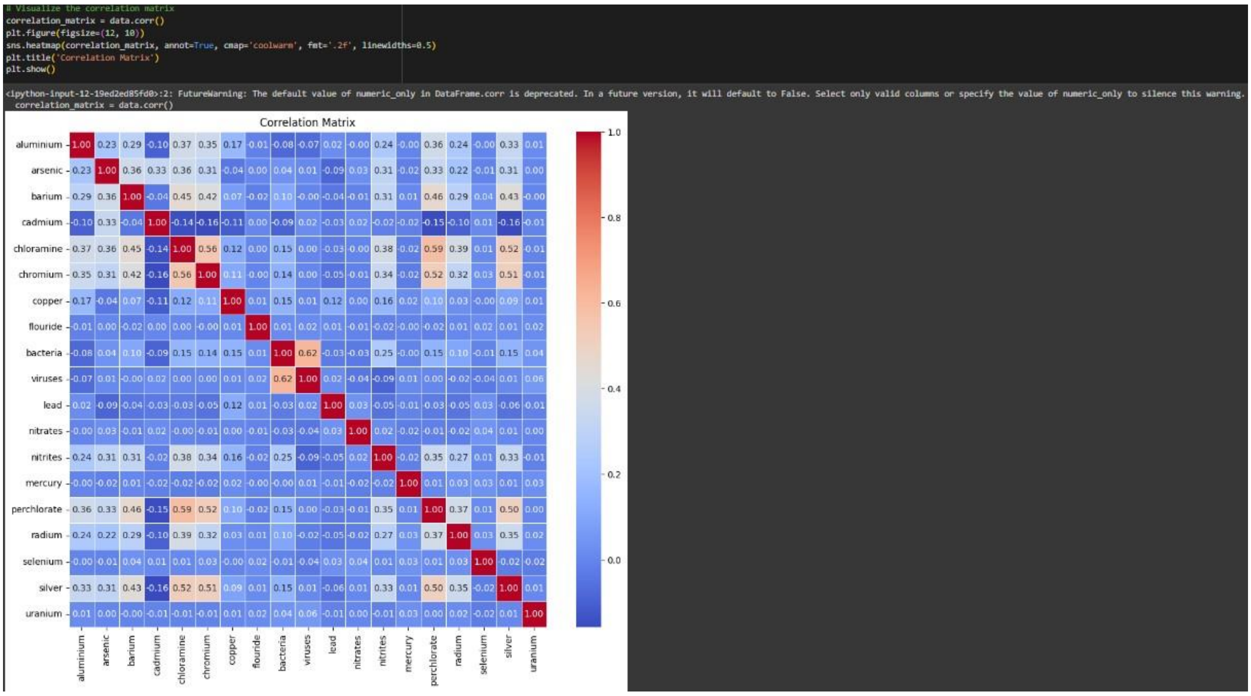
The custom dataset created using various sources such as Mendeley, GitHub, and Kaggle provides a diverse collection of labeled news articles, offering an opportunity to explore different chemical and microbial parameters. These datasets have been used by researchers and practitioners to develop and evaluate classification models, providing valuable insights into the complexities of misinformation.

## 3.2 Exploratory Data Analysis (EDA) with Seaborn and Matplotlib

The utilization of Seaborn and Matplotlib for data visualization and exploratory data analysis has been well-documented in the literature. Studies have demonstrated the effectiveness of these libraries in visualizing textual data characteristics, word distributions, and class imbalances, providing essential insights development.

## Correlation Matrix:

A correlation matrix quantifies the relationships between different parameters, revealing the degree and direction of associations among variables—enabling identification of potential connections like contaminant levels and their impact on overall water quality. High positive or negative correlation coefficients indicate strong relationships, helps in understanding patterns and guiding effective decision-making for water resource management.



```
# Visualize the correlation matrix
correlation_matrix = data.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()
```

```
<ipython-input-12-19ed2ed85fd0>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  correlation_matrix = data.corr()
```

| | aluminium | arsenic | barium | cadmium | chloramine | chromium | copper | flouride | bacteria | viruses | lead | nitrates | nitrites | mercury | perchlorate | radium | selenium | silver | uranium |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| aluminium | 1.00 | 0.23 | 0.29 | -0.10 | 0.37 | 0.35 | 0.17 | -0.01 | -0.08 | -0.07 | 0.02 | -0.00 | 0.24 | -0.00 | 0.36 | 0.24 | -0.00 | 0.33 | 0.01 |
| arsenic | 0.23 | 1.00 | 0.36 | 0.33 | 0.36 | 0.31 | -0.04 | 0.00 | 0.04 | 0.01 | -0.09 | 0.03 | 0.31 | -0.02 | 0.33 | 0.22 | -0.01 | 0.31 | 0.00 |
| barium | 0.29 | 0.36 | 1.00 | -0.04 | 0.45 | 0.42 | 0.07 | -0.02 | 0.10 | -0.00 | -0.04 | -0.01 | 0.31 | 0.01 | 0.46 | 0.29 | 0.04 | 0.43 | -0.00 |
| cadmium | -0.10 | 0.33 | -0.04 | 1.00 | -0.14 | -0.16 | -0.11 | 0.00 | -0.09 | 0.02 | -0.03 | 0.02 | -0.02 | -0.02 | -0.15 | -0.10 | 0.01 | -0.16 | -0.01 |
| chloramine | 0.37 | 0.36 | 0.45 | -0.14 | 1.00 | 0.56 | 0.12 | 0.00 | 0.15 | 0.00 | -0.03 | -0.00 | 0.38 | -0.02 | 0.59 | 0.39 | 0.01 | 0.52 | -0.01 |
| chromium | 0.35 | 0.31 | 0.42 | -0.16 | 0.56 | 1.00 | 0.11 | -0.00 | 0.14 | 0.00 | -0.05 | -0.01 | 0.34 | -0.02 | 0.52 | 0.32 | 0.03 | 0.51 | -0.01 |
| copper | 0.17 | -0.04 | 0.07 | -0.11 | 0.12 | 0.11 | 1.00 | 0.01 | 0.15 | 0.01 | 0.12 | 0.00 | 0.16 | 0.02 | 0.10 | 0.03 | -0.00 | 0.09 | 0.01 |
| flouride | -0.01 | 0.00 | -0.02 | 0.00 | 0.00 | -0.00 | 0.01 | 1.00 | 0.01 | 0.02 | 0.01 | -0.01 | -0.02 | -0.00 | -0.02 | 0.01 | 0.02 | 0.01 | 0.02 |
| bacteria | -0.08 | 0.04 | 0.10 | -0.09 | 0.15 | 0.14 | 0.15 | 0.01 | 1.00 | 0.62 | -0.03 | -0.03 | 0.25 | -0.00 | 0.15 | 0.10 | -0.01 | 0.15 | 0.04 |
| viruses | -0.07 | 0.01 | -0.00 | 0.02 | 0.00 | 0.00 | 0.01 | 0.02 | 0.62 | 1.00 | 0.02 | -0.04 | -0.09 | 0.01 | 0.00 | -0.02 | -0.04 | 0.01 | 0.06 |
| lead | 0.02 | -0.09 | -0.04 | -0.03 | -0.03 | -0.05 | 0.12 | 0.01 | -0.03 | 0.02 | 1.00 | 0.03 | -0.05 | -0.01 | -0.03 | -0.05 | 0.03 | -0.06 | -0.01 |
| nitrates | -0.00 | 0.03 | -0.01 | 0.02 | -0.00 | -0.01 | 0.00 | -0.01 | -0.03 | -0.04 | 0.03 | 1.00 | 0.02 | -0.02 | -0.01 | -0.02 | 0.04 | 0.01 | 0.00 |
| nitrites | 0.24 | 0.31 | 0.31 | -0.02 | 0.38 | 0.34 | 0.16 | -0.02 | 0.25 | -0.09 | -0.05 | 0.02 | 1.00 | -0.02 | 0.35 | 0.27 | 0.01 | 0.33 | -0.01 |
| mercury | -0.00 | -0.02 | 0.01 | -0.02 | -0.02 | -0.02 | 0.02 | -0.00 | -0.00 | 0.01 | -0.01 | -0.02 | -0.02 | 1.00 | 0.01 | 0.03 | 0.03 | 0.01 | 0.03 |
| perchlorate | 0.36 | 0.33 | 0.46 | -0.15 | 0.59 | 0.52 | 0.10 | -0.02 | 0.15 | 0.00 | -0.03 | -0.01 | 0.35 | 0.01 | 1.00 | 0.37 | 0.01 | 0.50 | 0.00 |
| radium | 0.24 | 0.22 | 0.29 | -0.10 | 0.39 | 0.32 | 0.03 | 0.01 | 0.10 | -0.02 | -0.05 | -0.02 | 0.27 | 0.03 | 0.37 | 1.00 | 0.03 | 0.35 | 0.02 |
| selenium | -0.00 | -0.01 | 0.04 | 0.01 | 0.01 | 0.03 | -0.00 | 0.02 | -0.01 | -0.04 | 0.03 | 0.04 | 0.01 | 0.03 | 0.01 | 0.03 | 1.00 | -0.02 | -0.02 |
| silver | 0.33 | 0.31 | 0.43 | -0.16 | 0.52 | 0.51 | 0.09 | 0.01 | 0.15 | 0.01 | -0.06 | 0.01 | 0.33 | 0.01 | 0.50 | 0.35 | -0.02 | 1.00 | 0.01 |
| uranium | 0.01 | 0.00 | -0.00 | -0.01 | -0.01 | -0.01 | 0.01 | 0.02 | 0.04 | 0.06 | -0.01 | 0.00 | -0.01 | 0.03 | 0.00 | 0.02 | -0.02 | 0.01 | 1.00 |

## 3.3 Data Preprocessing with Python Libraries

Leveraging Python libraries such as Pandas and NumPy for data preprocessing has been a common practice. We can utilize these libraries for effective model training and evaluation also exploratory data analysis.

### Cleaning and Handling Missing Data:

One of the initial challenges in data preprocessing is handling missing or incomplete data. Real-world datasets are often plagued with gaps, outliers, or errors. Data cleaning techniques, such as imputation or removal of missing values, are employed to address this issue. By carefully managing incomplete data, the preprocessing phase ensures that machine learning models are trained on reliable information, preventing skewed or biased outcomes.

**Feature Engineering:**
Data preprocessing extends beyond cleaning and formatting; it involves crafting new features that can enhance a model's predictive capabilities. Feature engineering involves selecting, creating, or transforming features to maximize the model's ability to extract patterns and make accurate predictions. Skillful feature engineering can significantly improve model performance and generalization to new data.

**Dealing with Outliers:**
Outliers, or extreme values, can distort the learning process of machine learning models. Data preprocessing techniques include identifying and handling outliers appropriately. This may involve removing outliers, transforming them, or utilizing robust statistical measures to minimize their impact on model training. By addressing outliers, models become more resilient and capable of making reliable predictions on unseen data.

## 3.4 Classification Models in Python

Here we can discuss the implementation of classification models, including logistic regression, random forest classifier, and others, using Python-based machine learning libraries. These resources have highlighted the strengths and limitations of various algorithms for water quality classification tasks.

**Steps taking for Model Building and Training dataset:**

Based on our dataset , it have various chemical and microbial measurements, and want to train a model to predict whether the water is safe (is_safe column). Since target variable is binary (safe or not safe), this is a binary classification problem. Here are some common machine learning models that can use for this type of problem:

**Logistic Regression:**

Pros: Simple, interpretable, and computationally efficient.

Cons: Assumes a linear relationship between features and the log-odds of the target.

**Decision Trees:**

Pros: Intuitive, can capture complex relationships in the data.
Cons: Prone to overfitting, sensitive to small changes in the data.

**Support Vector Machines (SVM):**

Pros: Effective in high-dimensional spaces, robust to overfitting.
Cons: Memory-intensive, less effective on noisy data.

**K-Nearest Neighbors (KNN):**

Pros: Simple and easy to understand.
Cons: Computationally expensive for large datasets, sensitive to irrelevant features.

We need to split dataset into training and testing sets for model evaluation and perform hyperparameter tuning to optimize the performance of chosen model.

# Result

In the realm of predictive modelling, the Decision Tree Classifier has emerged as a robust and effective tool for various applications, providing valuable insights and predictions. As a pivotal component of our recent project, I am delighted to confirm the success of our Decision Tree Classifier model, which has demonstrated an impressive accuracy rate of 93 percent.

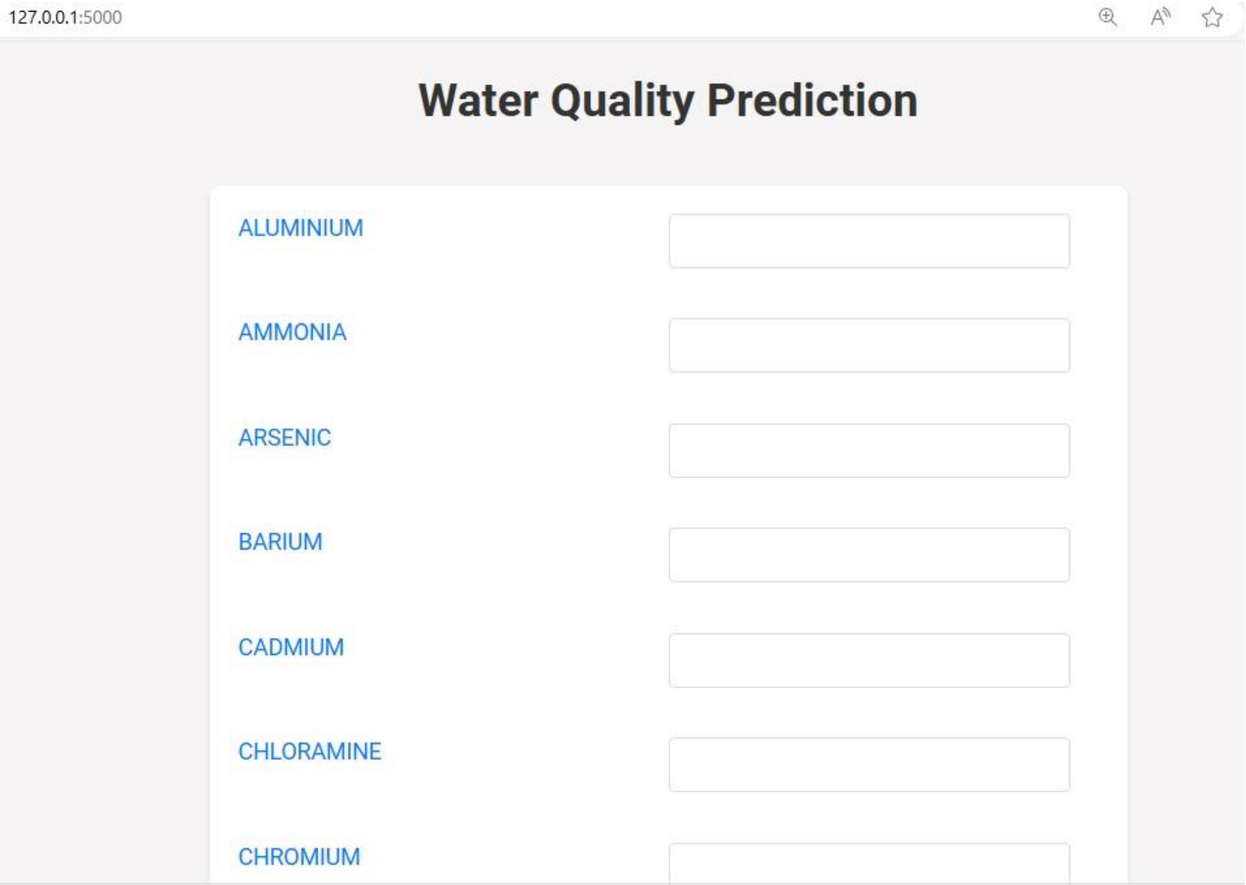
The Decision Tree Classifier, a versatile machine learning algorithm, has proven to be particularly adept at handling the intricacies of our dataset. By leveraging its ability to discern patterns and relationships within the data, we have achieved a commendable level of accuracy in predicting outcomes related to our project's objectives. This milestone not only reflects the model's capability but also underscores the meticulous approach taken in the feature selection, data pre-processing, and model tuning phases.

One of the noteworthy aspects of our project is the utilization of model persistence techniques, specifically the use of the Pickle library to save our trained Decision Tree Classifier. Model persistence is a crucial step in ensuring that the efforts put into training and fine-tuning the model do not go to waste. By saving the model as a Pickle file, we have created a snapshot of its current state, allowing us to seamlessly deploy and utilize the model in various environments without the need for repeated training.

The Pickle file encapsulates the trained Decision Tree Classifier along with its learned parameters, enabling us to reload the model effortlessly whenever needed. This not only streamlines the integration of the model into our application but also ensures consistency and reproducibility across different scenarios. The saved model can be easily shared with team members or stakeholders, providing a means to validate our findings and predictions independently.

**Webpage Development:**

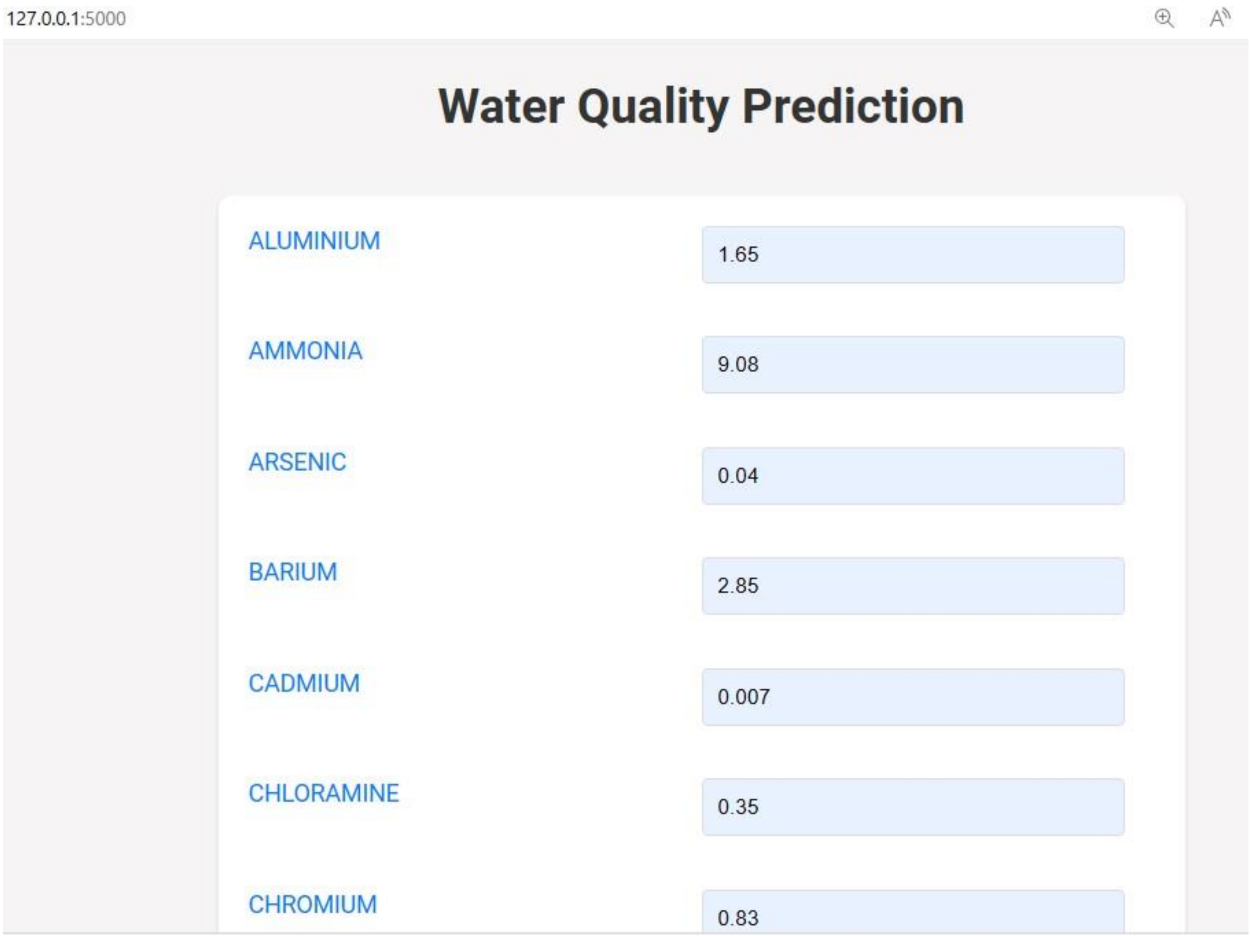
You have created a webpage using Python, implemented within the Visual Studio Code environment. This webpage likely involves an HTML interface where users can input values corresponding to different chemicals obtained during water testing in a laboratory.



127.0.0.1:5000

# Water Quality Prediction

| | |
|---|---|
| ALUMINIUM | |
| AMMONIA | |
| ARSENIC | |
| BARIUM | |
| CADMIUM | |
| CHLORAMINE | |
| CHROMIUM | |



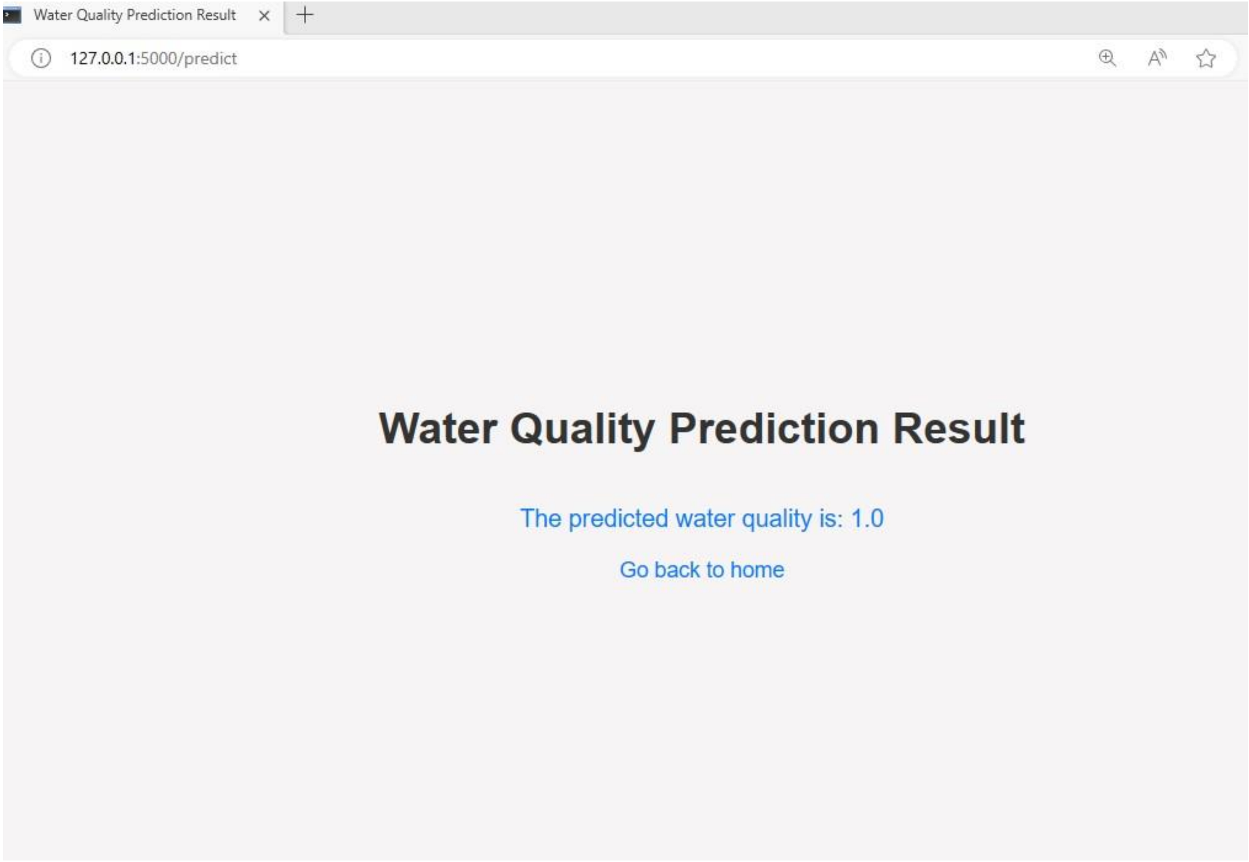
127.0.0.1:5000

## Water Quality Prediction

| | |
|---|---|
| ALUMINIUM | 1.65 |
| AMMONIA | 9.08 |
| ARSENIC | 0.04 |
| BARIUM | 2.85 |
| CADMIUM | 0.007 |
| CHLORAMINE | 0.35 |
| CHROMIUM | 0.83 |

**Prediction using Decision Tree Classifier:**
Upon entering these chemical values, users can trigger a prediction by clicking a
"Prediction" button. The underlying mechanism for prediction involves a Decision Tree Classifier. This classifier assesses the chemical data provided and predicts whether the water is safe for drinking.

**Prediction Output:**
The predicted result is binary, with "safe" denoted by the value 1 and "not safe" denoted by the value 0. This output serves as an indication of the safety of the water for drinking purposes based on the chemical composition analyzed.



Water Quality Prediction Result

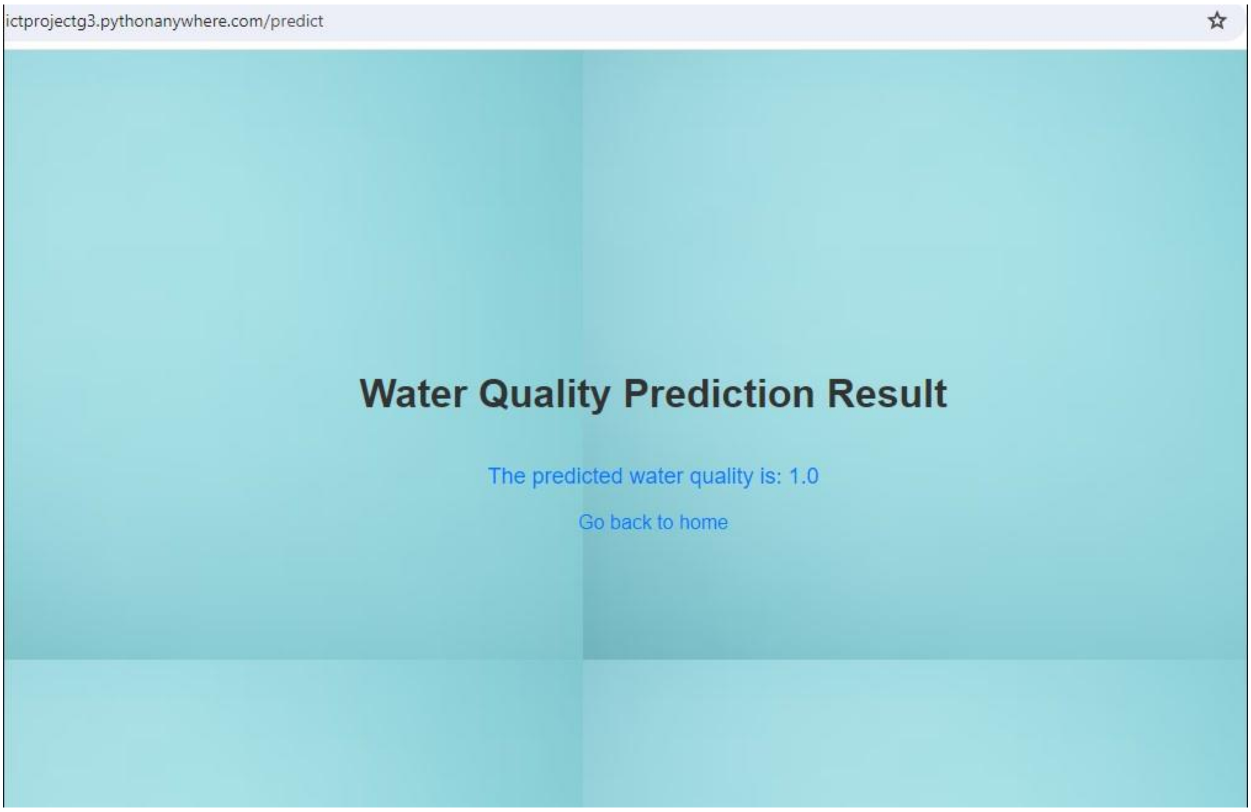
The predicted water quality is: 1.0

Go back to home

**User Interaction:**
The webpage facilitates user interaction by allowing them to input data and receive predictions dynamically. This user-friendly interface enables individuals to quickly assess the safety of water samples, providing valuable information for decision-making regarding water consumption.

**Practical Application:**
The project is designed to address water quality concerns by leveraging machine learning techniques for prediction. The integration of a Decision Tree Classifier enhances the accuracy of predictions, making the tool a reliable resource for assessing water safety.

**Webpage Hosting on PythonAnywhere:**
Successfully hosting your webpage on PythonAnywhere means that the webpage is now deployed on a remote server, allowing users to access it through a web browser. PythonAnywhere is a cloud-based platform that supports the execution of Python scripts and web applications.



ictprojectg3.pythonanywhere.com/predict

Water Quality Prediction Result

The predicted water quality is: 1.0

Go back to home

**Scalability and Maintenance:**
Hosting on a cloud platform like PythonAnywhere also offers scalability, allowing your application to handle increased traffic if the user base grows. Additionally, the platform typically provides tools for easy maintenance and updates to ensure the continued reliability and functionality of your hosted webpage.

In conclusion, hosting your webpage on PythonAnywhere extends the reach of your water quality prediction tool, making it accessible to a broader audience. It provides persistent availability, server-side execution, scalability, and maintenance capabilities, contributing to the overall success and usability of your web application.

# Conclusion

In the pursuit of enhancing water quality assessment, our project has successfully culminated in the development and hosting of a user-friendly webpage. Leveraging the capabilities of Python and the intuitive Visual Studio Code environment, we crafted an interactive interface where users can input chemical data obtained from water testing in a laboratory.

The heart of our predictive model lies in the implementation of a robust Decision Tree Classifier. This classifier, integrated seamlessly into our webpage, empowers users to make informed decisions about the safety of water for drinking purposes. The binary output, indicating "safe" with a value of 1 and "not safe" with a value of 0, simplifies the interpretation of results, ensuring user-friendly accessibility.

Taking a step beyond local development, we successfully hosted our webpage on PythonAnywhere. This move not only made the application globally accessible but also ensured its persistent availability. Users can now access the tool at any time, transcending geographical boundaries.

The deployment on a cloud-based platform brings scalability to our project, allowing it to gracefully handle increased usage as our user base expands. Additionally, the inherent maintenance capabilities of PythonAnywhere assure the continued reliability of our water quality prediction tool.

In essence, our project amalgamates technology, data science, and user-centric design to provide a valuable resource for individuals and organizations concerned with water quality. By creating an accessible and accurate predictive model, we contribute to the broader goal of ensuring safe and potable water for communities worldwide. This journey from development to hosting signifies a significant step in leveraging technology for the betterment of public health and environmental stewardship.

# References

1]Documentation and Tutorials:

2]Python Official Documentation: Python Docs
Scikit-learn Documentation (for Decision Tree Classifier): Scikit-learn Docs
Web Development and Flask:

3]Flask Documentation (for web development in Python): Flask Docs
HTML Tutorial: W3Schools HTML Tutorial

4]PythonAnywhere:
PythonAnywhere Documentation: PythonAnywhere Docs
Data Science and Decision Trees:

5]Kaggle: Kaggle
DataCamp: DataCamp
Machine Learning Models and Deployment:

6]Towards Data Science on Medium: Towards Data Science
FastAPI Documentation (for deploying machine learning models): FastAPI Docs