# UNIVERSITY OF HOUSTON


## INDE 6334 – Predictive Data Analytics

## Topic: "Analyzing and Predicting Water Quality for Safe Consumption Using Machine Learning"

**Submitted By:**

**Madhukumar Gopal (2354612)**

**Under the Guidance of**
**Prof. Dr. Ying Lin**

# Contents

## Abstract

Clean and safe drinking water is paramount to public health, yet in most locations worldwide, monitoring and preserving water quality is a huge challenge. Water testing through traditional methods, regardless of their accuracy, typically takes a lot of time and resources and therefore is not suitable for large-scale or real-time testing. Machine learning methods have been applied here to develop a predictive model that is able to determine the potability of water based on measurable physicochemical properties.

The **dataset** used, which was from **Kaggle**, contains the features like pH, hardness, solids, sulfate, conductivity, organic carbon, trihalomethanes, and turbidity, and each sample was marked as potable or not potable. An entire end-to-end data science pipeline was followed, comprising data preprocessing, imputation of missing values, exploratory data analysis, and model validation. Different supervised classification techniques were applied, viz., Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and XGBoost.

The models were evaluated on accuracy, precision, recall, F1-score, and ROC-AUC. Gradient Boosting and Random Forest were the best-performing models among the models compared for predictability and generalizability. The outcomes suggest that we can categorize water quality by using machine learning models and present a scalable approach for the early identification of hazardous water and prompt intervention.

This project deals with the real-world application of AI for water quality monitoring and demonstrates how data-driven technology can be actualized to assist in achieving sustainable development goals by ensuring clean drinking water through smart, automated treatment.

## 1. Introduction

Safe and pure drinking water is one of the basic requirements of human existence and public health. Yet, a large percentage of the population worldwide still lacks access to drinking water because of natural contamination, industrial pollution, and inadequate infrastructure. Monitoring water quality has been based traditionally on chemical and microbial laboratory analysis. Although accurate, such analysis is time-consuming, costly, and not well-suited for extension to large or remote settings.

In the last several years, machine learning and data science have emerged to provide powerful tools to automate water quality analysis. Machine learning algorithms take advantage of available datasets to learn the relationships and correlations between potability status and water quality parameters, allowing for fast and low-cost classification of water samples. Machine learning therefore offers a pragmatic solution to augment conventional approaches and facilitate timely decision-making for environmental agencies, municipalities, and health departments.

The main task of this project is to create a predictive model that classifies a water sample as potable (drinking) or not, given a data set of actual water quality analyses. The data set, which is retrieved from Kaggle, includes physicochemical properties like pH, hardness, solids, sulfate, conductivity, organic carbon, trihalomethanes, and turbidity, with a binary target variable of potability.

The goal of this project is to **implement a sequence of supervised classification algorithms, namely Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and XGBoost**. This project proceeds with a systematic flow starting from exploratory data analysis (EDA), missing value handling, normalization of data, and feature correlation study. Both model training and testing are done based on 80:20 train-test splitting, and all the classifiers' performances are assessed based on accuracy, precision, recall, F1-score, and ROC-AUC.

Therefore, the project not only aims to determine the most precise model for potability classification but also to deduce knowledge about the most important factors determining water safety. Finally, this research demonstrates how machine learning can be employed to assist in maintaining water safety, especially where laboratory analysis is scarce, and support general activities in public health protection, sustainability, and technological innovation in environmental monitoring.

## 2. Problem Statement

The safety of drinking water is an important public health issue, particularly in areas where laboratory testing is not easily available or is available only on a limited scale. Conventional water quality analysis methods, although dependable, are not always feasible with their expense, sophistication, and man-hours need. Hence, increasing interest is being taken in exploiting machine learning with the aim of automating and scaling potability analysis of water founded on easily measurable chemical and physical parameters.

This research explores the potential for the use of machine learning algorithms in the prediction of the safety of a water sample for drinking using only its physicochemical characteristics. It aims to develop an effective classifier that can achieve fast, affordable, and reliable water quality testing.

The research questions informing us about this project are:

1. Can machine learning effectively predict whether water samples are potable based on physicochemical parameters?
2. Which features have the greatest influence on water potability?
3. How do different machine learning models compare in performance and accuracy when classifying potable vs. non-potable water?

Previous research, such as **H. Gao (2022)**, used algorithms such as Logistic Regression and K-Nearest Neighbors (KNN) and obtained a similarly low accuracy of **61.52%.** In this project, we will attempt to boost prediction performance by employing more advanced classification models, improved data preprocessing techniques, and thorough model testing with parameters such as accuracy, precision, recall, F1-score, and ROC-AUC. In addition to model building, the project will also attempt to close the gap between data science and everyday usability by deploying the best model as an interactive live web application through Streamlit Cloud. The rollout is aimed at making the solution accessible for use by non-technical individuals and demonstrating its viability for real-time water safety evaluation.

### 2.1 Project Workflow

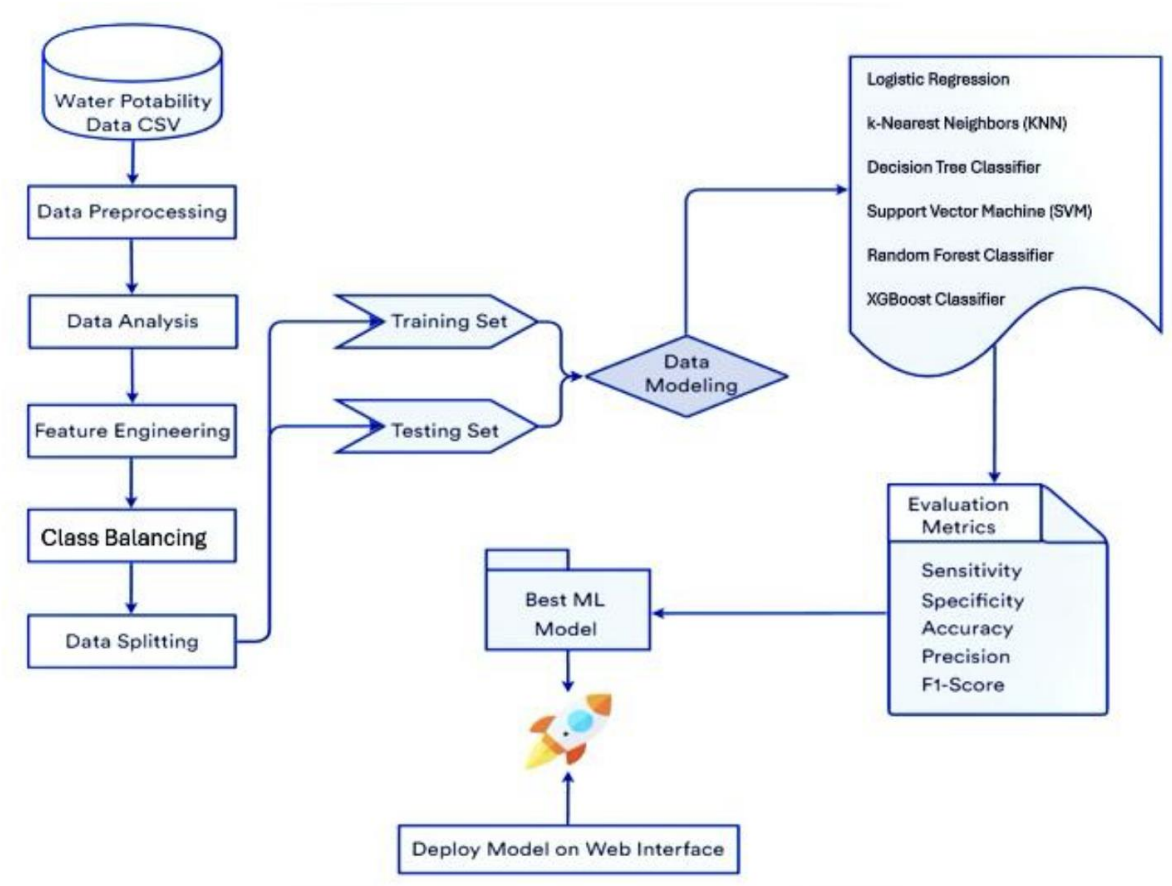
The project followed a structured machine learning pipeline, starting with dataset acquisition and culminating in deployment as a user-facing web application.

The key stages included:

1. **Data Collection & Cleaning**: Sourced from Kaggle, the dataset was cleaned and checked for missing values, which were imputed using median-based strategies.

2. **Exploratory Data Analysis (EDA)**: Statistical summaries, visualizations, and correlation matrices were used to understand feature distributions and identify relationships.
3. **Feature Engineering**: Data normalization and scaling were performed to improve model performance and convergence.
4. **Model Development**: Multiple classification models (Logistic Regression, Decision Tree, Random Forest, SVM, KNN) were trained and evaluated.
5. **Model Evaluation**: Models were compared using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.
6. **Deployment**: The best-performing model was integrated into a web application using Streamlit, allowing real-time potability prediction.
7. **Version Control & Code Access**: The complete project, including the Streamlit app and notebook, is available on GitHub for reproducibility and collaboration.

- 🔗 GitHub Repository: [https://github.com/punnasurya2000/water-potability-streamlit](https://github.com/punnasurya2000/water-potability-streamlit)



## 3. Literature Survey

In recent times, machine learning has become an effective instrument for forecasting water quality, providing a data-centered option to conventional laboratory water testing techniques. Numerous researchers have investigated different classification algorithms to determine if a water sample is suitable for consumption based on its chemical attributes. Initial research concentrated on linear models such as Logistic Regression, which are straightforward to understand but frequently do not adequately represent the intricate and non-linear trends found in environmental data. As discipline evolved, scientists moved towards more sophisticated models. Support Vector Machines (SVMs) became well-known for their capacity to manage high-dimensional data and non-linear correlations by utilizing kernel functions. For instance, research indicates that SVMs surpass linear models in differentiating between safe and unsafe water, particularly when the data features lack strong correlation.

Additional enhancements were observed with ensemble techniques such as Random Forest and XGBoost, which merge several decision trees to boost accuracy and minimize overfitting. These models have proven to be especially effective with imbalanced datasets—often seen in water quality issues—by more effectively identifying instances of the minority class. For example, scholars utilizing Random Forests on the UCI Water Quality dataset noted enhanced precision and recall when compared to single-tree models, whereas XGBoost delivered even stronger performance using gradient boosting methods. Certain research has also highlighted the significance of feature engineering, normalization, and cross-validation to enhance model generalization.

In general, the literature indicates that although simpler models establish a baseline, non-linear and ensemble methods consistently deliver higher predictive accuracy. This influenced the model selection process in this project, during which various algorithms were assessed and contrasted to identify the most efficient method for predicting water safety using actual data.

## 3. Methodology

### 3.1 Dataset Overview

The project data is being downloaded from Kaggle and provides physicochemical properties of the water samples in the form of binary classification as 1 for potable and 0 for not potable. The prime usage of the dataset is to provide quantitative features that will be used in training machine learning algorithms to distinguish between the quality of water being safe or dangerous.

Each entry in the dataset is a stand-alone water sample, characterized by nine salient features:

| Feature | Description |
|---|---|
| **ph** | Indicates the acidity or alkalinity of water |
| **Hardness** | Measures the concentration of calcium and magnesium ions (mg/L) |
| **Solids** | Total dissolved solids in the water (ppm) |
| **Chloramines** | Chlorine-based disinfectant used in water treatment (ppm) |
| **Sulfate** | Concentration of sulfate ions (mg/L) |
| **Conductivity** | Ability of water to conduct electricity (μS/cm) |
| **Organic_carbon** | Amount of organic compounds in the sample (ppm) |
| **Trihalomethanes** | Chemical compounds formed during chlorination (μg/L) |
| **Turbidity** | Clarity of water; measures light scattering (NTU) |
| **Potability** | Target variable; 1 = safe to drink, 0 = not safe |

The dataset consists of **3,276** samples in total. Of these, about **61%** have been labeled as non-potable, and **39%** have been labeled as potable, indicating a minimal class imbalance that will need to be considered while training and testing the model.

### Missing Values

During inspection, three columns—ph, Sulfate, and Trihalomethanes—were found to contain missing values. These were handled with median imputation, which is less outlier-sensitive than mean imputation and preserves the distributional properties of data.

**Class Distribution**

The water was assigned as safe (1) or unsafe (0) for drinking based on a binary variable Potability:

- **Non-potable (0)**: ~61% of total samples
- **Potable (1)**: ~39% of total samples

This class imbalance was handled with **class weight balancing** and **stratified splitting** during model training.

## Feature Characteristics

All features are numeric and widely vary in magnitude. Accordingly, data normalization was done before passing the features to some algorithms (like Logistic Regression and SVM) that are sensitive to feature magnitudes.

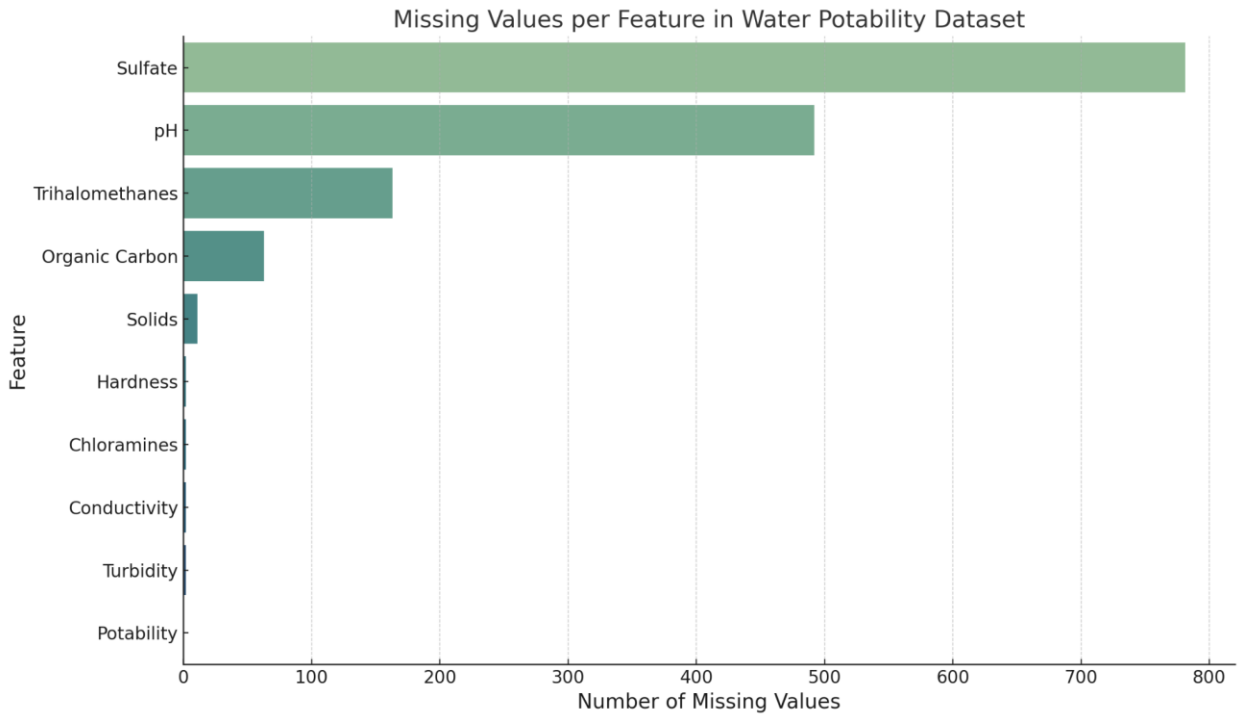
This data is a solid foundation for predictive modeling, with potential to explore relationships and patterns among numerous chemical attributes and water safety results.

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| ph | 2784.0 | 7.080766 | 1.594605 | 0.000000 | 6.092665 | 7.035894 | 8.062251 | 14.000000 |
| Hardness | 3274.0 | 196.359915 | 32.887464 | 47.432000 | 176.844221 | 196.928061 | 216.671731 | 323.124000 |
| Solids | 3265.0 | 22015.793088 | 8777.206464 | 320.942611 | 15658.086280 | 20933.512750 | 27336.962620 | 61227.196010 |
| Chloramines | 3274.0 | 10.189087 | 175.472523 | -4.502117 | 6.129569 | 7.131929 | 8.115665 | 10047.050500 |
| Sulfate | 2495.0 | 333.775777 | 41.416840 | 129.000000 | 307.699498 | 333.073546 | 359.950170 | 481.030642 |
| Conductivity | 3274.0 | 426.206621 | 80.810979 | 181.483754 | 365.763185 | 421.884968 | 481.763341 | 753.342620 |
| Organic_carbon | 3213.0 | 50.531962 | 1672.222564 | 2.200000 | 12.079207 | 14.225917 | 16.563116 | 91456.654130 |
| Trihalomethanes | 3113.0 | 66.395074 | 16.177464 | 0.738000 | 55.835966 | 66.621027 | 77.339918 | 124.000000 |
| Turbidity | 3274.0 | 3.966995 | 0.780551 | 1.450000 | 3.439880 | 3.955122 | 4.500544 | 6.739000 |
| Potability | 3276.0 | 0.103175 | 0.304233 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |

### 3.2 Data Preprocessing

| **Features** | Missing Values | Implications |
|---|---|---|
| Sulfate | 781 | Has the highest missing data (~24% of total samples). As sulfate influences water taste and chemical composition, imputing it carefully is essential. |
| pH | 492 | A crucial indicator of acidity or alkalinity; missing in ~15% of records. This needs careful handling as pH is vital for potability assessment. |
| Organic Carbon | 63 | Organic carbon levels can indicate |
| Solids | 11 | Minor missing data. Solids affect taste and clarity; still worth imputing. |
| Hardness, Chloramines, Conductivity, Turbidity | 2 each | Hardness, Chloramines, Conductivity, Turbidity |
| Potability | 0 | No missing values in the target variable, which is excellent for supervised learning. |

Sulfate and pH have the highest missing values, requiring careful imputation due to their impact on water safety. Other features like Organic Carbon and Solids have minor gaps but still influence quality. Potability has no missing data, making the dataset ideal for supervised learning.




**Outlier Detection and Treatment**

The boxplots of all the numerical features in the dataset are shown below. These plots were used to visually detect **outliers**—data points that deviate significantly from the rest. Outliers can distort statistical summaries and reduce the accuracy of machine learning models, especially those that rely on distance-based metrics or assume data normality.

**Key Takeaways from the Boxplots**

**Highly Outliered Features:**

- **Chloramines**: Contains extreme outliers reaching up to 10,000 ppm, far higher than expected.
- **Organic Carbon**: Includes outliers greater than 80,000 ppm, suggesting substantial contamination in some samples.
- **Solids**: Displays a wide range with outliers near 60,000 ppm.

These extreme values point to the existence of heavily contaminated or anomalous water samples.

**Moderate Outliers:**

- Trihalomethanes, Sulfate, and Conductivity show moderate outliers despite being within a reasonable spread.

These features are chemically important and require careful treatment to prevent misleading the model.

**Minimum or No Significant Outliers:**

- pH, Hardness, and Turbidity show compact, normally distributed data with few or no major outliers.

These features are more stable and contribute to model robustness.

**Outlier Treatment Strategy**

To handle these extreme values without compromising data integrity, the **Interquartile Range (IQR) capping** method was used:

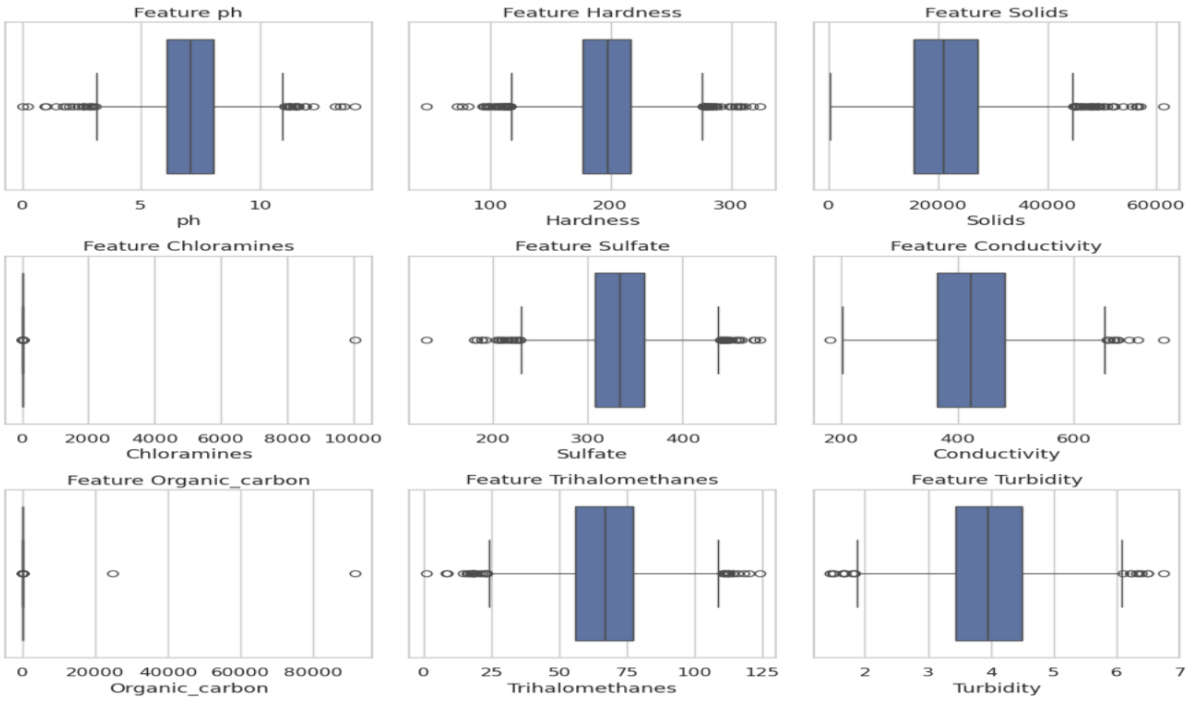
- Any value below Q1 - 1.5 × IQR or above Q3 + 1.5 × IQR was capped at the boundary thresholds.
- This technique preserves the shape of the distribution and avoids data loss that occurs with row deletion.
- Capping is especially important in imbalanced datasets, where retaining every sample is crucial for learning minority class patterns.

**Why It Matters**

- **Improved Model Stability**: Prevents the model from being influenced by outlier-driven skewness, especially for SVM and Logistic Regression.

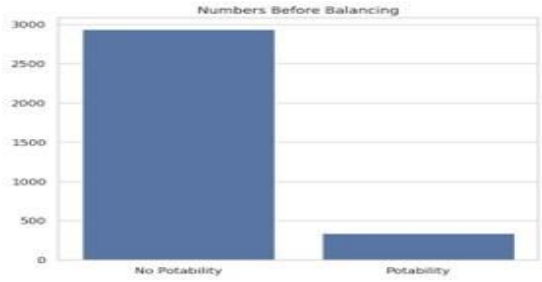
- **Avoid Overfitting**: Prevents the model from fitting noise and extreme cases.

- **Fair Feature Scaling**: Makes feature scaling (e.g., with StandardScaler) more effective by keeping features on a similar scale.

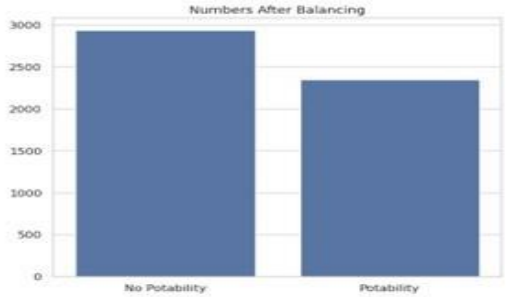



**Class Imbalance and SMOTE**

The dataset initially exhibited a significant imbalance in the target variable, with approximately **90% of the samples labeled as non-potable** and only **10% as potable**, as shown in the left-hand chart. This imbalance can lead machine learning models to favor the majority class, resulting in poor performance when identifying potable water samples.




To resolve this, we applied **SMOTE (Synthetic Minority Over-sampling Technique)**—a method that generates synthetic examples for the minority class instead of duplicating existing samples. The post-balancing distribution, shown in the right-hand chart, illustrates a much more equitable class ratio.




**Benefits of SMOTE**

- Improves model fairness and class recall
- Reduces bias toward the dominant class
- Preserves all original data while enhancing minority representation

This preprocessing step was essential to ensure that the model could accurately predict both potable and non-potable water samples.

## 4. Exploratory Data Analysis (EDA)

### 4.1 Distribution of Features

To get an idea about the nature of the input features, we have drawn histograms with KDE (Kernel Density Estimate) curves for all the numerical features. This allowed us to see the shape, spread, and skewness of each feature.

**1. Characteristics Approaching a Normal Distribution:**
The following are some of the characteristics that exhibit a very close to symmetric, bell-shaped distribution, indicating normal distribution:

- Hardness, Conductivity, Turbidity, Chloramines, Organic Carbon

These characteristics possess stable and steady values for different water samples and therefore are good candidates for being used by algorithms that make Gaussian input assumptions.

**2. Features Displaying Mild Right Skew:**
Some features display a mild right skew, i.e., nearly all values are lower, with an extended tail of high values:

- Solids, Sulfate, Trihalomethanes

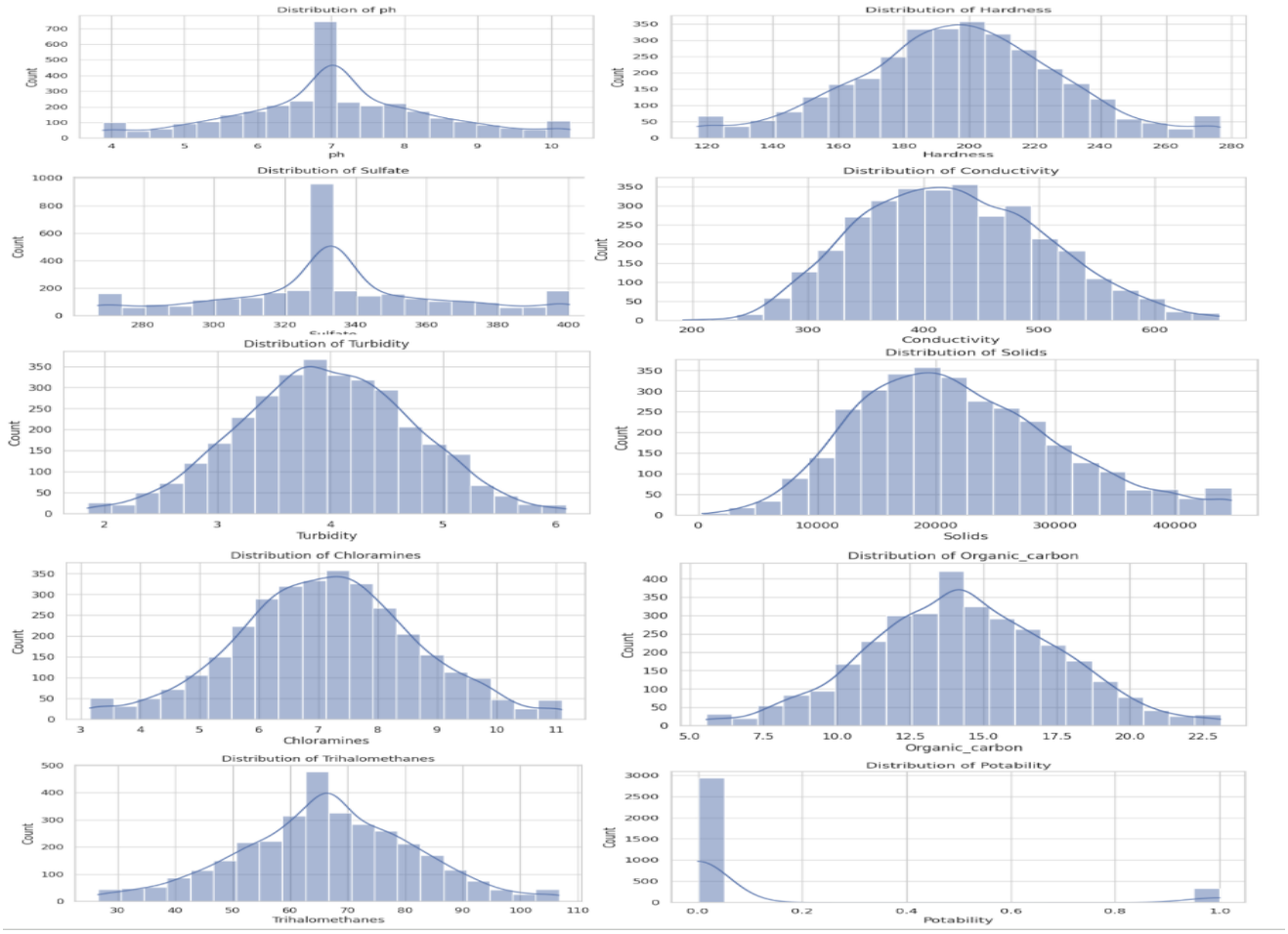
This implies that most samples have a medium level, but some contain much higher levels, and this could affect model performance if not scaled or transformed.

**3. Features with Concentration Peaks or Sharp Concentration Zones:**

- **pH**: Displays a dominant peak at 7, the optimum level of neutrality for drinking water.
- **Sulfate and Trihalomethanes**: Expose sharp clustering, likely due to them sharing the same water treatment processes or origins.

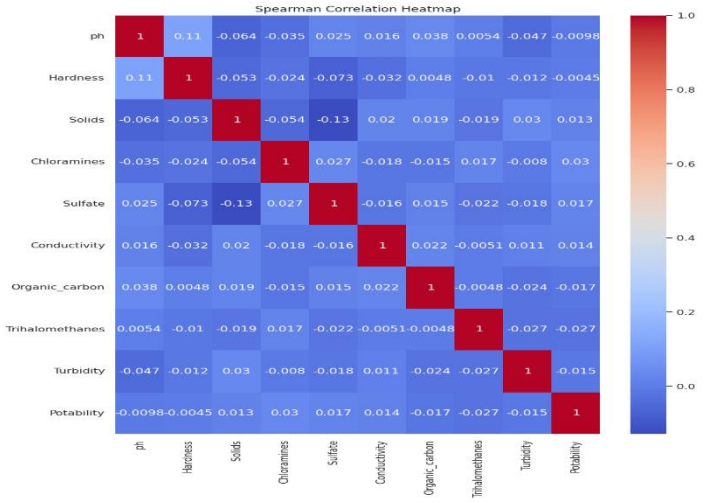
**4. Target Variable – Potability:**

The class distribution of the Potability feature is highly imbalanced. Most of the water samples are non-potable (class 0) with an overwhelming majority, and few are potable (class 1). This confirms the need for class-balancing techniques such as SMOTE before model training to avoid majority class bias.




### 4.2 Correlation Analysis

For understanding the linear as well as monotonic correlations among numeric features, a Spearman correlation heatmap was visualized. Unlike Pearson correlation, Spearman estimates rank-based correlation and is therefore more capable of taking care of non-linear correlations, as well as non-normal distributed features.



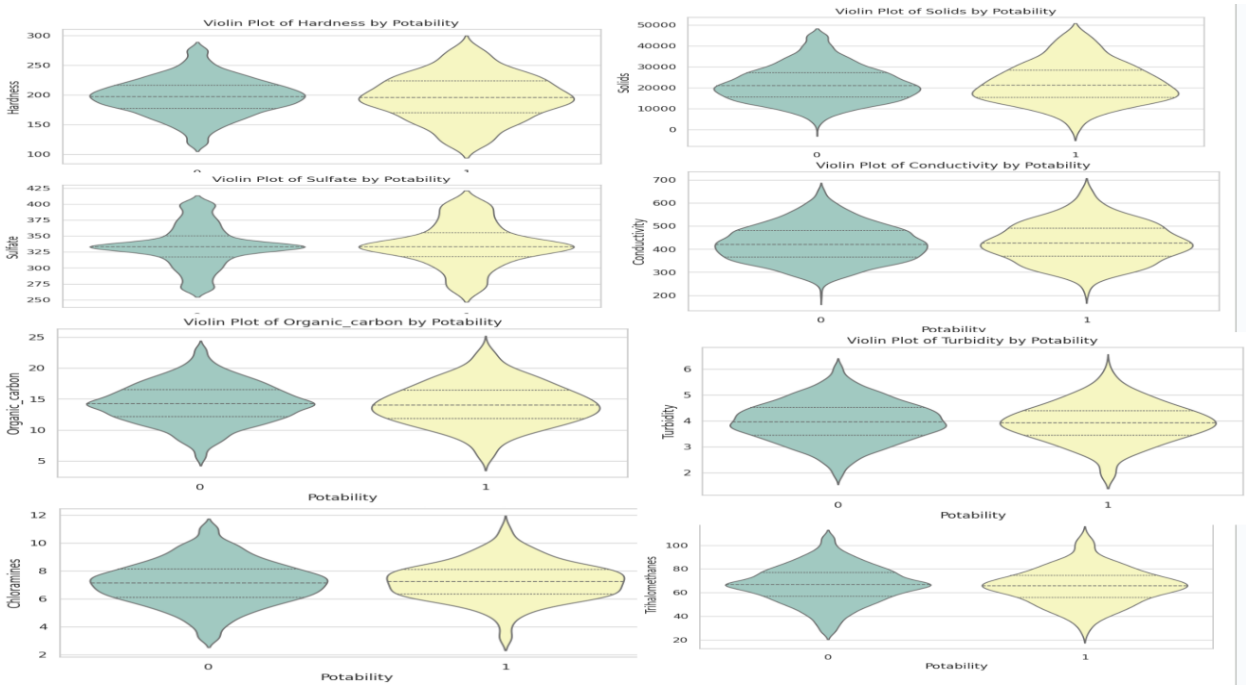

We observe the following from the heatmap:

- Most features loosely correlate with each other with correlation values in the vicinity of 0.
- This shows a lack of multicollinearity between features, i.e., the features are largely independent, and all contribute individually to the prediction model.

- Hardness and pH have weak positive correlation (≈ 0.11), while Sulfate and Solids have weak negative correlation (≈ -0.13).
- The target variable Potability has very low correlation with all features, which means that no feature can alone strongly predict the potability.
- This independence benefits machine learning models because it implies that every feature has the potential to capture various pieces of information about water quality.

### 4.3 Multivariate Analysis

- Multivariate visualizations (pair plots) showed overlapping distributions, indicating the need for complex models.



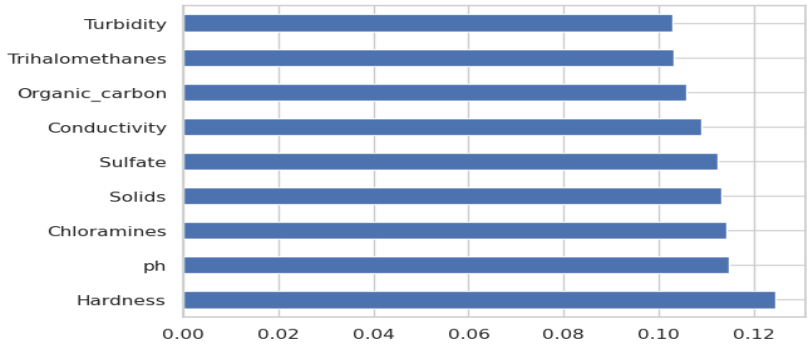

- Most features show overlapping distributions, making **linear separation difficult**.
- This highlights the need for **complex models and feature combinations** to improve prediction accuracy.
- Most features like **pH, Hardness, and Solids** overlap in both potable and non-potable water.
- No single feature can decide potability on its own.
- Potability depends on a **combination of features**, not just one.

## 5. Case Study: Feature Engineering & Selection

Using an ExtraTreesClassifier, we assessed the importance of each feature.

Results showed:

- **Top feature**: Hardness
- Other influential features: Chloramines, Solids, and Sulfate




No single feature could predict potability alone, emphasizing the importance of combining multiple features.

## 6. Model Development

To evaluate the predictive potential of various algorithms, we trained six supervised machine learning models. Each model underwent rigorous hyperparameter tuning using **GridSearchCV** to ensure optimal performance. The evaluation focused on both accuracy and the Area Under the ROC Curve (AUC), which together give a holistic picture of model effectiveness.

### 6.1 Logistic Regression

Logistic Regression is a linear classifier utilized for predicting binary results based on a collection of independent variables. It represents the likelihood that a specific input is part of a certain class by utilizing the sigmoid (logistic) function, which converts predicted values to a scale from 0 to 1. The model acquires the ideal weights for each feature by reducing the log-loss (cross-entropy) throughout training. Although Logistic Regression is straightforward and easy to interpret, it presumes a linear link between the input variables and the target's log-odds, which limits its effectiveness on datasets exhibiting intricate, non-linear patterns. Techniques for regularization like L1 (Lasso) and L2 (Ridge) are frequently employed to avoid overfitting and enhance generalization.

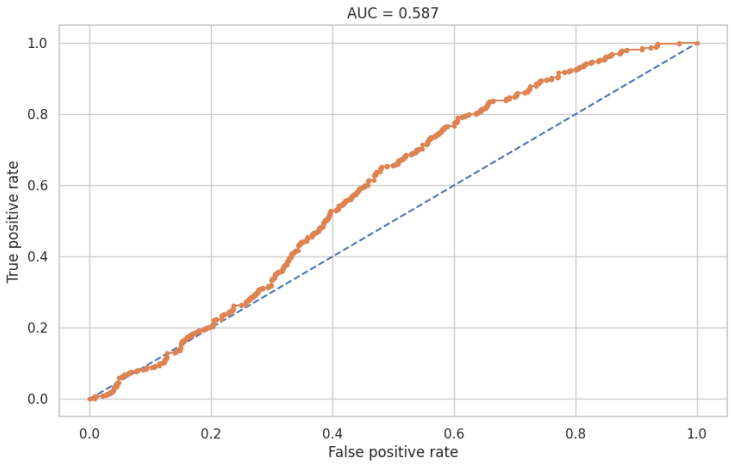
**Result**

The Logistic Regression model was implemented using scikit-learn's LogisticRegression and optimized with a Grid Search across parameters such as penalty (l1, l2), regularization strength C, and class_weight (balanced, None), using 10-fold cross-validation.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.64 | 0.56 | 0.60 | 601 |
| 1 | 0.50 | 0.59 | 0.54 | 457 |
| accuracy | | | 0.57 | 1058 |
| macro avg | 0.57 | 0.57 | 0.57 | 1058 |
| weighted avg | 0.58 | 0.57 | 0.57 | 1058 |

The model achieved an accuracy of **57%,** with a precision of **0.64 for non-potable water (class 0) and 0.50 for potable (class 1), and a macro-averaged F1-score of 0.57.**

The ROC AUC score was 0.587, indicating weak discriminative power. These results suggest that Logistic Regression, being a linear model, struggled with this dataset likely due to non-linear relationships and weak feature-target correlations, as highlighted in the correlation analysis. Although class balancing was attempted, the model was limited by its inability to model complex decision boundaries, making it useful primarily as a baseline for comparison with more expressive models.

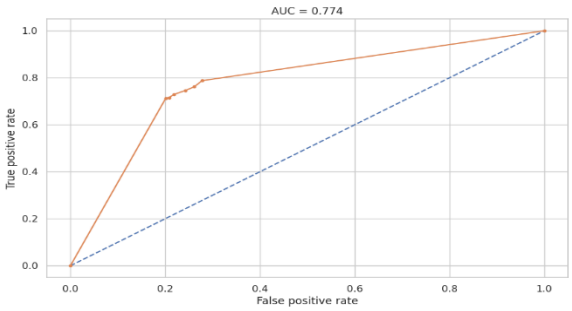



### 6.2 Decision Tree Classifier

Decision Tree is a supervised learning method that employs a tree-shaped framework to represent choices based on feature values. It repeatedly divides the dataset into smaller subsets based on criteria that enhance the distinction between classes. These divisions are established using metrics such as Gini Impurity or Information Gain (derived from Entropy), with the goal of producing uniform branches. Every internal node signifies a feature-oriented decision rule, whereas leaf nodes represent final class labels. Decision Trees are non-parametric, which signifies that they do not assume anything about the distribution of the data and can successfully represent non-linear relationships. Nonetheless, they are susceptible to overfitting, particularly when the tree becomes excessively deep, necessitating pruning or restricting tree depth for better generalization.

**Result:**

The Decision Tree Classifier demonstrated a reasonable performance, achieving an overall accuracy of **76%, with F1-scores of 0.79 for class 0 (non-potable) and 0.72 for class 1 (potable),**




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.79 | 0.78 | 0.79 | 601 |
| 1 | 0.72 | 0.73 | 0.72 | 457 |
| accuracy | | | 0.76 | 1058 |
| macro avg | 0.75 | 0.76 | 0.75 | 1058 |
| weighted avg | 0.76 | 0.76 | 0.76 | 1058 |

**resulting in a macro-average F1-score of 0.75.** The ROC AUC score of 0.774 reflects good yet not outstanding discriminatory ability. Decision Trees are straightforward and interpretable models capable of capturing non-linear relationships and interactions between features without the necessity for feature scaling. Nonetheless, they are susceptible to overfitting, particularly when the tree grows deep and is excessively customized to the training data. This could clarify why the model's ability to generalize was not as good as that of ensemble methods such as Random Forest.

Nevertheless, the Decision Tree model offers significant insights into the importance of features and decision boundaries, and functions effectively as an independent model when simplicity and clarity are essential.

### 6.3 K-Nearest Neighbors (KNN)

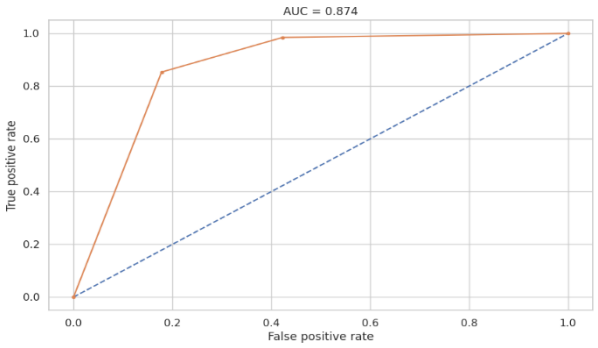
K-Nearest Neighbors is a non-parametric, instance-driven learning algorithm applied for tasks involving classification and regression. In classification, it determines the class of a new data point by locating the "k" nearest training examples in the feature space and designating the most frequent class among these. Distance metrics like Euclidean, Manhattan, or Minkowski are utilized to assess proximity. KNN lacks a clear training phase, categorizing it as a lazy learner; however, it may require significant computation during the prediction stage. Its effectiveness largely relies on the selection of 'k', scaling of features, and distribution of classes. KNN works especially well with locally organized and well-defined data, but it can have difficulties with high-dimensional or imbalanced data without prior preprocessing.

**Result**

The K-Nearest Neighbors (KNN) model was optimized through Grid Search to determine the ideal number of neighbors (n_neighbors ranging from 1 to 9), with validation performed using both 3-fold and 10-fold cross-validation methods. KNN demonstrated robust performance, attaining an **accuracy of 84%, an F1-score of 0.85 for non-potable water (class 0) and 0.82 for potable water (class 1), plus a macro-average F1-score of 0.83.**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.88 | 0.82 | 0.85 | 601 |
| 1 | 0.78 | 0.85 | 0.82 | 457 |
| accuracy | | | 0.84 | 1058 |
| macro avg | 0.83 | 0.84 | 0.83 | 1058 |
| weighted avg | 0.84 | 0.84 | 0.84 | 1058 |

The **ROC AUC score of 0.874** for the model further demonstrates its outstanding discriminative capability. These findings indicate that KNN successfully identified the class patterns in the dataset, probably because of the significant structure in the feature space and adequate class separability when utilizing distance-based learning. The high recall for class 1 (0.85) demonstrates its efficiency in recognizing safe drinking water samples. In general, KNN demonstrated strong performance, underscoring that even basic instance-based models can produce excellent results when the feature space is well-structured and appropriately adjusted.



### 6.4 Random Forest Classifier

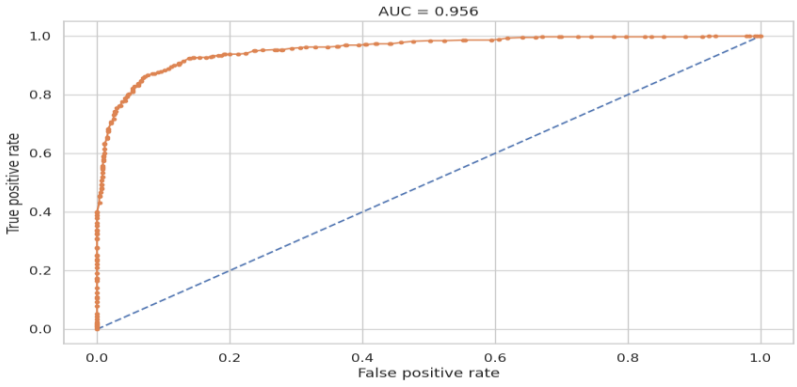
Random Forest is an ensemble learning method that constructs several decision trees and merges their predictions to enhance precision and reliability. Every tree is trained on a random portion of the data (through bootstrap sampling) and chooses a random selection of features at every split, encouraging diversity among the trees. In classification, every tree casts a vote for a class label, with the ultimate result being decided by majority voting. This collective method mitigates the chance of overfitting, a frequent problem in single decision trees, and enhances generalization. Random Forest can model intricate, non-linear connections and is very efficient for datasets that contain noise or have missing information. It also offers feature importance scores, rendering it valuable for interpretability in tasks involving feature selection.

**Results**

The Random Forest Classifier exhibited robust and steady performance on the water potability dataset, attaining an overall **accuracy of 90%, alongside balanced F1-scores of 0.91 for class 0 (non-potable) and 0.88 for class 1 (potable).** The model achieved a macro-average F1-score of 0.90 and a strong ROC AUC score of 0.956, demonstrating outstanding classification ability.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.90 | 0.93 | 0.91 | 601 |
| 1 | 0.90 | 0.86 | 0.88 | 457 |
| accuracy | | | 0.90 | 1058 |
| macro avg | 0.90 | 0.90 | 0.90 | 1058 |
| weighted avg | 0.90 | 0.90 | 0.90 | 1058 |

The curve illustrates the excellent performance of the Random Forest Classifier. The curve remains significantly above the diagonal reference line, signifying elevated true positive rates at different thresholds. The **AUC (Area Under Curve) score of 0.956** verifies the model's outstanding capability to differentiate between drinkable and undrinkable water samples. This elevated AUC indicates that Random Forest attained a nearly ideal equilibrium between sensitivity and specificity, positioning it as one of the most dependable models in the research. Its capacity to manage non-linear feature interactions and minimize overfitting via ensemble averaging played a major role in this impressive performance.



### 6.5 Support Vector Machine (SVM)

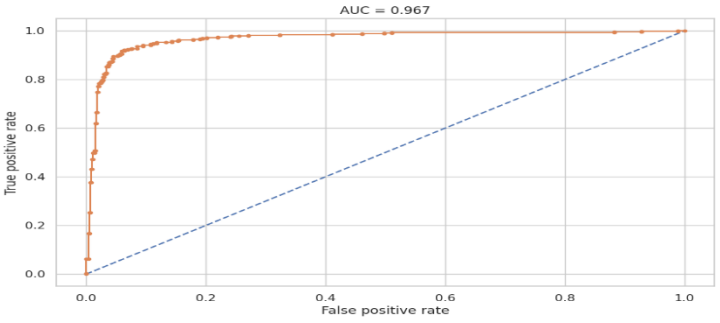
Support Vector Machine (SVM) is a robust supervised learning algorithm employed for classification purposes. It operates by identifying the ideal hyperplane that most effectively divides data points from various classes by maximizing the distance between the closest points (referred to as support vectors). When data cannot be linearly separated, SVM employs kernel functions—like the Radial Basis Function (RBF)—to transform data into a higher-dimensional space where linear separation becomes feasible. This capability to represent non-linear associations renders SVM very efficient for intricate datasets. It is also resilient to overfitting, particularly in high-dimensional environments, and shows good performance when the feature count exceeds the sample count.

**Result**

The Support Vector Classifier (SVC) was applied to the water potability dataset and delivered exceptional performance, achieving an **accuracy of 93%, with F1-scores of 0.94 for class 0 (non-potable) and 0.91 for class 1 (potable).** The model also attained a macro-average F1-score of 0.93 and a very high **AUC score of 0.967**, indicating superior discriminative ability. These results highlight SVM's strength in handling non-linear relationships with kernel functions, making it well-suited for datasets like this where linear models fail to separate the classes effectively.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.92 | 0.96 | 0.94 | 601 |
| 1 | 0.94 | 0.89 | 0.91 | 457 |
| accuracy | | | 0.93 | 1058 |
| macro avg | 0.93 | 0.92 | 0.93 | 1058 |
| weighted avg | 0.93 | 0.93 | 0.93 | 1058 |

The ROC curve for the Support Vector Machine demonstrates a remarkable ascent toward the top-left corner, signifying that the model is very successful at differentiating between drinkable and non-drinkable water. With an **AUC of 0.967,** the SVM model evidently shows robust predictive capabilities and very few false positives. This performance emphasizes the effectiveness of the SVM, particularly with the RBF kernel, in managing the intricate, non-linear characteristics of the data.



### 6.6 XGBoost Classifier

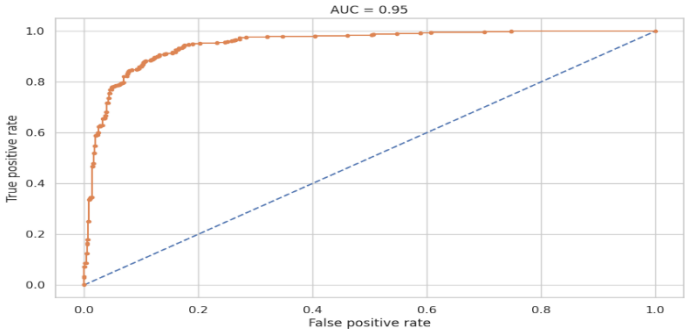
XGBoost (Extreme Gradient Boosting) is a powerful, scalable machine learning algorithm built on the gradient boosting framework. It creates a sequence of decision trees, with each subsequent tree addressing the mistakes of the earlier ones. In training, it reduces a regularized objective function that incorporates both prediction error and model complexity, aiding in the prevention of overfitting. XGBoost facilitates tree pruning, manages missing values, enables parallel processing, and incorporates regularization, resulting in high speed and accuracy. Its capacity to represent intricate, non-linear connections and its strong performance on structured/tabular data have made it a favored option for classification tasks in practical scenarios.

**Result**

The XGBoost Classifier demonstrated impressive predictive capability, attaining an **accuracy of 88% and F1-scores of 0.90 for class 0 (non-potable) and 0.87 for class 1 (potable), with a macro-average F1-score of 0.88.**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.91 | 0.88 | 0.90 | 601 |
| 1 | 0.85 | 0.88 | 0.87 | 457 |
| accuracy | | | 0.88 | 1058 |
| macro avg | 0.88 | 0.88 | 0.88 | 1058 |
| weighted avg | 0.88 | 0.88 | 0.88 | 1058 |

The model achieved an impressive ROC **AUC score of 0.95**, showcasing its strong ability to differentiate between the two classes. XGBoost, a gradient boosting ensemble technique, is proficient in addressing both linear and non-linear relationships via iterative learning and efficient regularization. Its capacity to merge various weak learners (shallow trees) into a robust predictive model allows it to recognize intricate patterns and decrease both bias and variance. The balanced precision and recall for both classes indicate that the model effectively manages class imbalance and remains robust with various sample types. In general, XGBoost demonstrates itself as a very efficient model for this task, providing an excellent balance between accuracy, speed, and interpretability.




### 7.6 Deployment Using Streamlit Cloud

To ensure practical usability and accessibility of the developed machine learning model, the final solution was deployed using **Streamlit Cloud**. Streamlit is an open-source Python framework that enables the rapid creation and deployment of interactive web applications directly from Python scripts. This allowed the team to transform the potability prediction model into a functional tool accessible via a web browser.

The deployed application features a simple, intuitive interface that accepts user inputs for key physicochemical water quality parameters, including: pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic Carbon, Trihalomethanes, Turbidity.

Upon entering the values, the app predicts whether the water is **potable (safe to drink)** or **non-potable**, based on the classification model trained earlier. The model deployed was selected based on its superior performance metrics during the evaluation phase, ensuring reliable predictions.

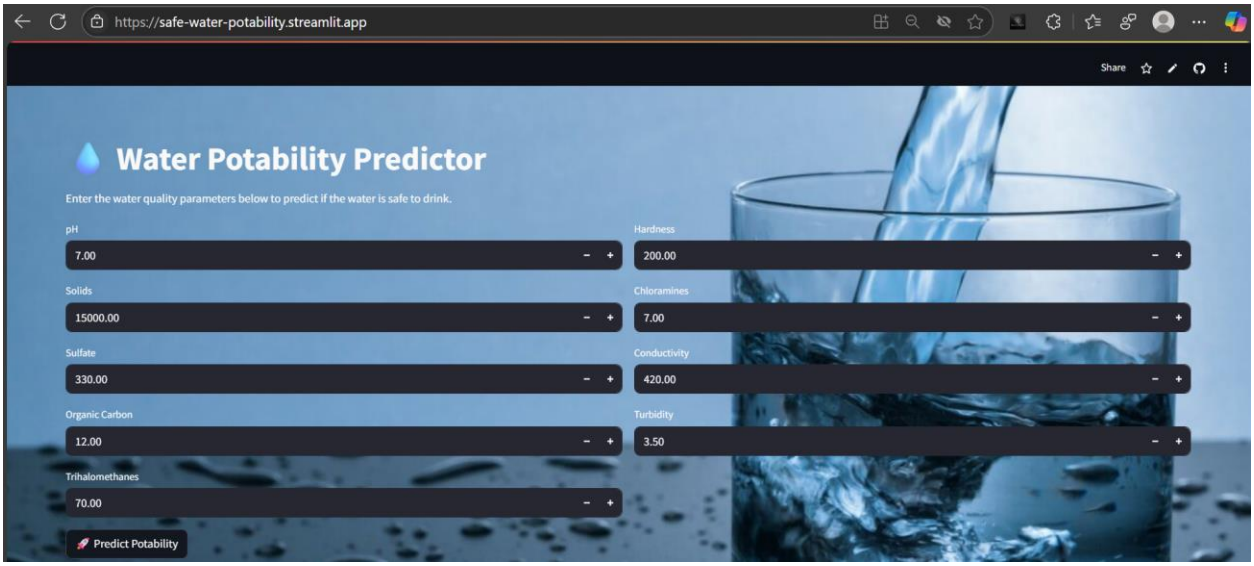
This deployment highlights the practical applicability of machine learning in environmental and public health domains. By offering real-time predictions in a user-friendly format, the Streamlit app can be used by researchers, water management agencies, or even individuals seeking quick water safety assessments.

The project and deployed model are fully open-source and available for review and enhancement. The source code repository contains the complete workflow, including data preprocessing, model training, and the Streamlit app code.

- 🌐 **Live App**: https://safe-water-potability.streamlit.app/

This deployment marks the final step in translating a data science project into a real-world application, demonstrating the end-to-end utility of machine learning for solving environmental challenges.

**User Interface** :




The deployed **Water Potability Predictor** features a clean and interactive user interface developed using **Streamlit**, chosen for its rapid prototyping capabilities and seamless model integration.

**Input**:

Users are prompted to input values for essential water quality parameters, including **pH, Solids, Sulfate, Organic Carbon, Hardness, Chloramines, Conductivity, Turbidity**, and **Trihalomethanes**, using intuitive sliders. Upon clicking the **"Predict Potability"** button, the system uses the trained **Support Vector Machine (SVM)** model—the best performer in terms of accuracy (**93%**) and AUC (**0.97**)—to generate a prediction.

The result is then displayed with a clear message:

- ✅ **"Water is safe to drink"** if the model predicts potability.
- ❌ **"Water is not potable"** if the water is deemed unsafe.

These outputs are visually supported by **tick or cross icons**, providing an immediate and user-friendly indication of water safety. This enhances usability, especially for non-technical users who may rely on the tool for practical decision-making. The interface's aesthetic background and streamlined design help communicate complex machine learning functionality in a simple, accessible manner, effectively bridging the gap between data science and real-world application in environmental health.

**7. Results & Discussion**

This section presents the results of six machine learning models trained to classify water samples as potable or non-potable based on physicochemical properties. Each model was evaluated using standard classification metrics: **Accuracy**, **Precision**, **Recall**, **F1-Score**, and **AUC (Area Under the ROC Curve)**.

| Model | Accuracy | Precision | Recall | F1-Score | AUC |
|---|---|---|---|---|---|
| Logistic Regression | 0.57 | 0.58 | 0.57 | 0.57 | 0.587 |
| Decision Tree | 0.76 | 0.75 | 0.76 | 0.75 | 0.774 |
| KNN | 0.84 | 0.84 | 0.84 | 0.84 | 0.874 |
| XGBoost | 0.88 | 0.88 | 0.88 | 0.88 | 0.95 |
| Random Forest | 0.9 | 0.9 | 0.9 | 0.9 | 0.956 |
| SVM | 0.93 | 0.93 | 0.93 | 0.93 | 0.967 |

**Support Vector Machine (SVM)**

SVM demonstrated the highest performance across all metrics, achieving 93% accuracy and an AUC of 0.967. This suggests that SVM was the most effective in separating the two classes (potable vs non-potable) based on the provided features. The model benefits from its ability to find optimal hyperplanes and handle non-linear relationships through kernel tricks.

**Random Forest**

Random Forest followed closely with 90% accuracy and an AUC of 0.956. Its robustness to noise and feature interactions made it a strong performer. It also provided insights into feature importance, which can be useful in explaining model behavior.

**XGBoost**

With 88% accuracy and an AUC of 0.950, XGBoost proved to be a competitive model, benefiting from its gradient boosting framework. It was effective at reducing bias and handling moderate imbalance without overfitting.

**K-Nearest Neighbors (KNN)**

KNN achieved 84% accuracy, which is commendable, especially considering its simplicity. However, being a distance-based model, it is sensitive to scaling and less suited for high-dimensional or imbalanced datasets.

**Decision Tree**

The Decision Tree classifier achieved 76% accuracy. Although it performed better than Logistic Regression, it tends to overfit on training data without pruning or ensemble strategies. It remains valuable for interpretability and understanding decision boundaries.

**Logistic Regression**

This model recorded the lowest performance, with only 57% accuracy and an AUC of 0.587. Its linear nature and inability to model complex relationships contributed to poor generalization. It also struggled with the class imbalance in the dataset.

**Key Observations:**

- **Non-linear models** like SVM, Random Forest, and XGBoost significantly outperformed linear models, highlighting the complexity in the decision boundaries of potable vs. non-potable water.

- **Class imbalance** negatively affected Logistic Regression and Decision Tree, reinforcing the need for proper resampling or class-weighting techniques.

- **Ensemble methods** (Random Forest, XGBoost) handled variance and overfitting better and achieved more stable results.

- **Model performance** improved substantially after applying **data preprocessing**, **scaling**, and **class balancing**, validating the importance of a well-defined pipeline.

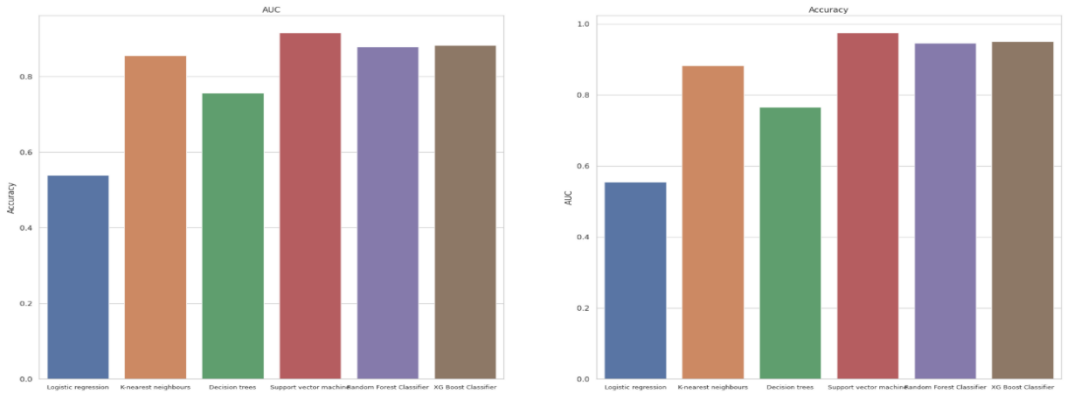
**Key Insights**:

- **SVM** was the best-performing model, achieving the highest accuracy (93%) and AUC (0.97), making it highly reliable for classifying water potability.
- **Random Forest** and **XGBoost** followed closely, offering strong, balanced predictions with excellent class separation.
- **KNN** provided solid performance and improved significantly with scaling and balancing techniques.
- **Logistic Regression** and **Decision Tree** showed limited performance, primarily due to overlapping feature distributions and class imbalance.

**Graphical Analysis of Evaluation Metrics**

To visually compare model performance, bar charts were generated for key evaluation metrics: **AUC**, **Accuracy**, and **ROC-AUC**.

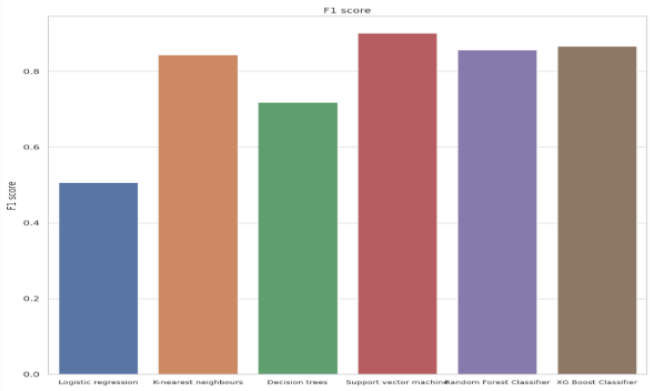
- The **Accuracy** chart clearly shows **SVM** outperforming all other models, with **Random Forest** and **XGBoost** closely behind. **Logistic Regression** lags significantly, validating its lower reliability in handling complex, nonlinear relationships.




- In the **AUC** chart, **SVM again leads**, closely followed by **XGBoost** and **Random Forest**, reflecting their superior ability to distinguish between potable and non-potable classes. **KNN** showed moderate performance, while **Decision Tree** and **Logistic Regression** were less effective in maintaining a strong balance between sensitivity and specificity.

- These visual comparisons support the tabular results and emphasize the **importance of advanced models and proper preprocessing** for achieving robust classification in water potability prediction.

Overall, models with more advanced architectures and ensemble techniques demonstrated superior generalization and classification capabilities, reinforcing the need for robust preprocessing and model tuning in predictive analytics.




## 8. Conclusion

The primary objective of this study was to predict water potability using various physicochemical parameters by applying multiple machine learning algorithms. A complete pipeline was implemented, involving data preprocessing, class balancing, feature scaling, and model evaluation to ensure optimal performance.

Six classification models were trained and compared. The key findings are summarized below:

- **Support Vector Machine (SVM)**
    - Achieved the highest performance with **93% accuracy** and an **AUC score of 0.967**
    - Demonstrated excellent ability to distinguish between potable and non-potable water
    - Best overall performer across all evaluation metrics
- **Random Forest**
    - Delivered strong and consistent results
    - Robust to overfitting and effective in handling noisy and imbalanced data
- **XGBoost**
    - Comparable to Random Forest in terms of performance
    - Particularly effective after parameter tuning, offering good generalization
- **K-Nearest Neighbors (KNN)**
    - Showed moderate performance
    - Sensitive to data scaling and distance-based feature relationships

- **Decision Tree**
    - Performed reasonably well but prone to overfitting
    - Simpler and interpretable, but less accurate compared to ensemble methods
- **Logistic Regression**
    - Lowest accuracy among all models
    - Performance limited due to overlapping feature distributions and class imbalance

These results confirm that **machine learning algorithms can effectively predict water potability**, especially when proper **data preprocessing, feature engineering, and class balancing** are applied.

The deployed model, integrated into a **Streamlit web application**, demonstrates the practical applicability of this solution for **real-time water quality assessment**. Such systems can aid environmental monitoring agencies and public health departments in early detection and safe water distribution.

Future enhancements may include integrating additional features (e.g., location, temperature, bacterial levels), applying deep learning techniques, and extending deployment to mobile and IoT-based platforms.Such models can assist governments, municipalities, and NGOs in early detection of unsafe water and in building real-time water monitoring systems.

## 9. Future Scope

While the current study demonstrates the feasibility of using machine learning to predict water potability based on physicochemical parameters, there are several avenues for enhancement and expansion in future work:

1. **Utilization of Larger and More Diverse Datasets:** The current dataset, though informative, is limited in scale and geographic diversity. Incorporating larger and region-specific datasets can significantly improve the generalizability of the model across different environmental and demographic conditions.

2. **Incorporation of Additional Features:** Future models can be enriched with more detailed parameters such as **temperature**, **geographic location**, and **bacterial content**. These additional features could provide a more holistic view of water quality and enhance model accuracy.

3. **Advanced Class Imbalance Handling:** The dataset exhibits a moderate imbalance between potable and non-potable samples. While basic techniques were used to address this, future iterations could apply more sophisticated methods such as ADASYN (Adaptive Synthetic Sampling), SMOTE with Tomek links, or cost-sensitive learning to better handle minority class predictions.

4. **Integration of Deep Learning Models:** Although classical machine learning algorithms performed reasonably well, the inclusion of deep learning models such as **Artificial Neural Networks (ANNs)** or **Convolutional Neural Networks (CNNs)** (if spatial data is added) may uncover deeper, non-linear patterns and improve predictive capabilities.

5. **Model Stacking and Ensemble Techniques:** Exploring ensemble strategies such as stacking, boosting, or bagging combinations could further improve performance by leveraging the strengths of multiple classifiers in a hybrid framework.

6. **Field Deployment and IoT Integration:** The deployed Streamlit app demonstrates the practicality of the model. Future scope includes integrating the model into **IoT-based water monitoring systems**, enabling real-time predictions and alerts in resource-constrained or remote areas.

## 10. References


1. H. Gao, Y. Li, H. Lu, and S. Zhu, "Water Potability Analysis and Prediction," *Highlights in Science, Engineering and Technology AMMSAC*, vol. 16, 2022.

2. Kaggle Dataset: https://www.kaggle.com/datasets/uom190346a/water-quality-and-potability/data

3. Scikit-learn Documentation: https://scikit-learn.org/stable/documentation.html

4. SMOTE – Synthetic Minority Over-sampling Technique: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

5. Towards Data Science. *Handling Missing Data Like a Pro*. https://towardsdatascience.com/handling-missing-data-like-a-pro-part-1-307b7b7b6324

6. Breiman, L. (2001). *Random Forests*. **Machine Learning**, 45(1), 5–32 https://doi.org/10.1023/A:1010933404324

7. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: Synthetic minority over-sampling technique*. *Journal of Artificial Intelligence Research*, 16, 321–357. https://doi.org/10.1613/jair.953