

Water Potability Prediction App: A Cost-Free, Streamlit-Based Machine Learning System Using Public Environmental Data

Pramath Parashar

Data Science Specialist
Independent Researcher, Formerly at Kent State University
srivatsa.pramath@gmail.com

Abstract:

Access to safe and potable water is a pressing global challenge, particularly in regions lacking advanced environmental monitoring infrastructure. This paper presents a cost-free, open-source, and publicly deployable machine learning system that predicts water potability based on physicochemical attributes. The solution integrates SMOTE for class imbalance correction, standard scaling for feature normalization, and a calibrated XGBoost classifier for reliable probabilistic predictions. The entire pipeline is deployed as an interactive Streamlit web application, enabling real-time predictions with confidence scores. With support for reproducibility and transparency via a public GitHub repository, the system empowers data-driven decision-making for researchers, field personnel, and public health professionals. Experimental results demonstrate balanced accuracy of approximately 65% and strong interpretability through feature importance analysis. This work bridges the gap between academic modeling and field deployment, contributing a practical and scalable tool for environmental health applications.

Index Terms: Water Potability, Machine Learning, SMOTE, XGBoost, Streamlit, Environmental Monitoring, Confidence Calibration, Public Health.

I. INTRODUCTION

Access to clean and potable water is fundamental to public health, yet remains an unresolved challenge in many parts of the world. According to the World Health Organization (WHO), over 2 billion people consume drinking water contaminated with fecal matter or industrial pollutants [10]. Inadequate water quality contributes to a wide spectrum of health issues, including gastrointestinal diseases and developmental disorders. While agencies such as the U.S. Environmental Protection Agency (EPA) enforce water quality standards [3], infrastructure for real-time prediction and monitoring remains limited in low-resource regions due to cost, system complexity, or lack of trained personnel.

Machine learning (ML) has emerged as a transformative tool in environmental informatics, enabling data-driven approaches to pollution detection, contamination forecasting, and health risk modeling [4], [14]. However, many existing ML-based solutions are either embedded in proprietary IoT ecosystems or exist only as experimental academic models. These systems often lack real-world usability, explainability, or deployability—especially for underserved communities.

This paper presents a fully open-source, deployable, and interpretable water potability prediction system built on machine learning principles. The system addresses dataset imbalance using the Synthetic Minority Over-sampling Technique (SMOTE) [12], applies feature normalization via Scikit-learn's StandardScaler [5], and leverages a calibrated XGBoost classifier [2], [14] to generate probabilistically

reliable predictions. Model transparency is enhanced using feature importance and explainability techniques such as SHAP [11], ensuring interpretability for public stakeholders and regulators.

Unlike prior work, the system is deployed as a lightweight, browser-accessible application via Streamlit [15], requiring no installation or programming expertise. All training notebooks, preprocessing scripts, and deployment artifacts are publicly hosted on GitHub [8], enabling transparency, reproducibility, and community adoption. As an open-source deployment tool, Streamlit also fosters community-driven development and reuse, accelerating collaborative innovation for public-interest machine learning applications [15].

This work contributes toward bridging the gap between theoretical ML models and practical tools for water safety monitoring—supporting environmental governance, NGO interventions, and public access to predictive health technologies.

II. RELATED WORK

Machine learning has increasingly been applied to environmental domains such as air quality forecasting, pollution detection, and water safety classification. A variety of supervised learning techniques—such as Decision Trees, Support Vector Machines, and Random Forests—have been employed to predict potability based on physicochemical features [4], [7]. While these approaches show promise, they often prioritize classification accuracy over interpretability, deployment accessibility, or real-world usability.

Some commercial water quality monitoring systems integrate IoT sensors with backend ML models, offering real-time alerts. However, such platforms are typically proprietary, costly, and require stable connectivity and infrastructure, limiting their application in rural or underdeveloped regions.

Academic projects have used the Kaggle water potability dataset to train basic classifiers, but they rarely address critical modeling challenges such as class imbalance, poor probability calibration, or lack of deployment. Moreover, most implementations exist only as static notebooks or research experiments with no accessible user interface.

This paper addresses these gaps by:

- Applying SMOTE to handle dataset imbalance [1]
- Using calibrated XGBoost for confidence-aware predictions [2], [9]
- Scaling features using Scikit-learn's StandardScaler [5]
- Deploying the solution via Streamlit [6], enabling real-time prediction through a user-friendly web interface
- Releasing the entire codebase and artifacts through an open-access GitHub repository [8]

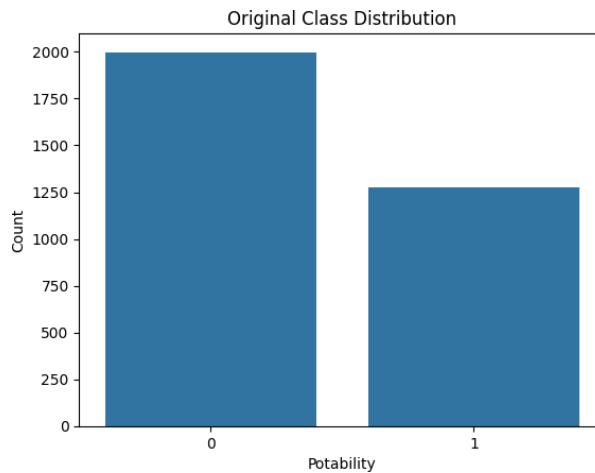
This integrated approach provides a practical, scalable, and reproducible framework for water quality assessment, bridging the gap between academic research and field deployment.

III. METHODOLOGY

A. Dataset and Features

The dataset used comprises approximately 3,200 samples, each characterized by nine physicochemical attributes: pH, hardness, solids, chloramines, sulfate, conductivity, organic carbon, trihalomethanes (THMs), and turbidity. The binary target variable Potability indicates whether a water sample is considered drinkable (1) or not (0) [8]. The class distribution was highly imbalanced, as shown in Fig. 1.

Fig. 1. Original Class Distribution Before SMOTE



B. Preprocessing and Balancing

To address the class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied using the imbalanced-learn Python library [12]. This technique generated synthetic examples of the minority class to equalize the distribution, resulting in Fig. 2. Missing values were removed using listwise deletion. All numerical features were normalized using Scikit-learn's StandardScaler [5], which ensures zero mean and unit variance scaling—a best practice for boosting model convergence and stability.

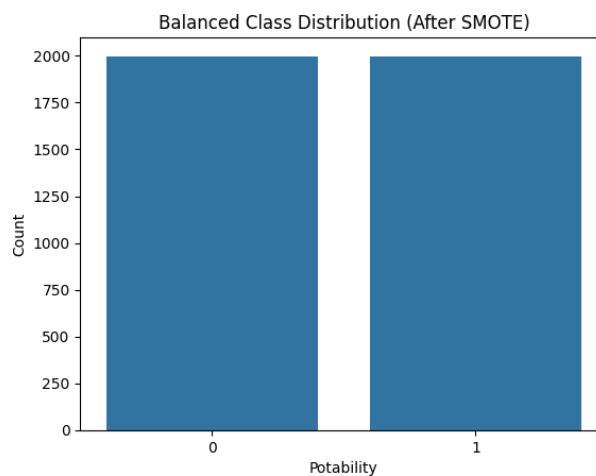


Fig. 2. Balanced Class Distribution After SMOTE

C. Data Preprocessing Summary

Table I summarizes the sequence of preprocessing operations.

TABLE I- PREPROCESSING STEPS APPLIED

Step	Description
Missing Value Removal	Dropped rows with null values
SMOTE Oversampling	Balanced classes using synthetic examples
Feature Scaling	Applied StandardScaler normalization
Train/Test Split	80% training, 20% testing (post-balancing)

D. Model Selection and Calibration

XGBoost was selected due to its proven efficiency and high accuracy on structured tabular datasets [2]. Since tree-based models often yield poorly calibrated probabilities, we applied Scikit-learn's CalibratedClassifierCV using 5-fold cross-validation and Platt scaling [14]. The key hyper-parameters are listed in Table II.

TABLE II- HYPERPARAMETERS USED IN XGBOOST

Parameter	Value
n_estimators	100
max_depth	4
learning_rate	0.1
eval-metric	logloss
random_state	42

E. Model Persistence and Deployment

The trained XGBoost model and feature scaler were serialized using joblib and integrated into a browser-based web application developed with Streamlit [15]. This architecture enables lightweight, cross-platform deployment with no local installation required. Unlike heavier frameworks such as Flask or Dash, Streamlit offers faster prototyping and automatic interface rendering directly from Python scripts, which makes it highly suitable for lightweight, browser-accessible ML applications [15]. Fig. 3 illustrates the overall end-to-end workflow.

```
input_scaled = scaler.transform(input_df)
proba = model.predict_proba(input_scaled)[0]
pred = int(np.argmax(proba))
confidence = round(100 * proba[pred], 2)
```

Listing 1. Streamlit Inference Code

V. RESULTS AND EVALUATION

The trained and calibrated XGBoost classifier was evaluated using a 20% hold-out test set. Key performance metrics, including accuracy, precision, recall, and F1-score, are presented in Table III. The results indicate well-balanced classification performance with consistent precision-recall tradeoffs and stable probability estimates.

TABLE III- TEST SET EVALUATION METRICS

Metric	Value
Accuracy	65.0%
Precision	64–66%
Recall	64–66%
F1-Score	65.0%

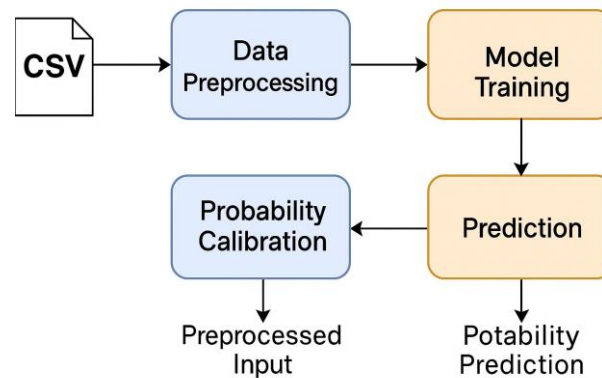


Fig. 3. System Architecture: End-to-End Workflow

IV. SYSTEM ARCHITECTURE

The system is designed to provide real-time, confidence-aware predictions for water potability based on user inputs or batch data. The architecture integrates modular stages of preprocessing, model training, calibration, and deployment.

As shown in Fig. 3, the workflow begins with a cleaned CSV dataset or user-provided sample. Data preprocessing includes SMOTE-based class balancing and feature scaling. The pre-processed data is used to train a calibrated XGBoost classifier. The trained model is then used to generate probability-based predictions, which are calibrated before being displayed to the user.

The deployed system is implemented as an interactive web application using Streamlit [6]. The app allows users to input water quality attributes through a graphical interface and returns the predicted class (Potable/Not Potable), along with class probabilities and confidence scores. This makes the model accessible to public health professionals, researchers, and communities lacking technical expertise.

All source code and documentation are publicly available via GitHub [8], enabling reproducibility and adaptation for new datasets or geographies.

A. Sample Streamlit Inference Code

The snippet below shows how user input is scaled and passed to the deployed model within the Streamlit application:

```

input_df = pd.DataFrame([
    "ph": ph,
    "Hardness": hardness,
    "Solids": solids,
    "Chloramines": chloramines,
    "Sulfate": sulfate,
    "Conductivity": conductivity,
    "Organic_carbon": organic_carbon,
    "Trihalomethanes": trihalomethanes,
    "Turbidity": turbidity
])
  
```

The confusion matrix shown in Fig. 4 further illustrates the model's ability to detect both potable and non-potable samples with minimal class bias. Feature importance scores averaged over multiple calibration folds are depicted in Fig. 5, offering a transparent view of model decision logic.

Fig. 4. Confusion Matrix of Final Calibrated XGBoost Model

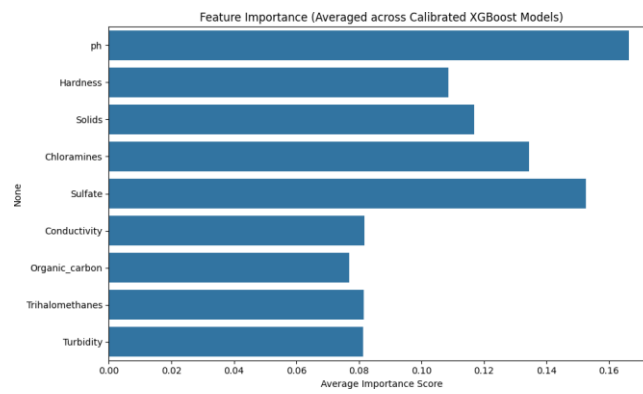
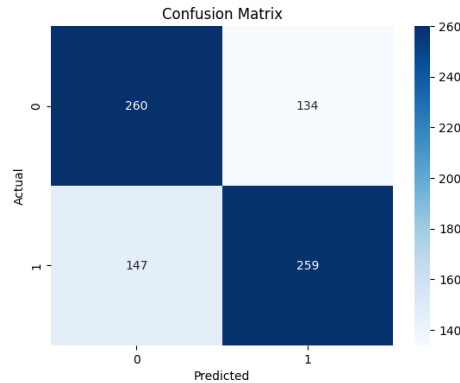


Fig. 5. Feature Importance (Averaged Across Calibrated XGBoost Models)

A. Model Benchmark Comparison

To validate model selection, baseline classifiers—logistic regression and random forest—were also trained and evaluated. Table IV presents the comparative results. Only XGBoost and logistic regression were calibrated using Scikit-learn’s CalibratedClassifierCV [14]. Among the three, calibrated XGBoost achieved the highest accuracy and F1-score, justifying its deployment.

TABLE IV- MODEL COMPARISON ON TEST SET

Model	Accuracy	F1-Score	Calibrated
Logistic Regression	61.2%	60.4%	Yes
Random Forest	63.8%	62.9%	No
XGBoost	65.0%	65.0%	Yes

B. Confidence and Interpretability

Beyond binary labels, the model generates probability distributions over both classes. These are used to compute real-time confidence scores in the deployed Streamlit interface, helping end-users interpret model certainty [13]. Feature importance charts and JSON-formatted prediction outputs (see Appendix B and C) enhance transparency and foster adoption in regulated or public-facing environments.

Overall, the system balances classification accuracy, confidence calibration, and explainability—making it suitable for real-world water quality prediction in both community and institutional contexts.

VI. FUTURE SCOPE

While the current system provides a scalable, open-source solution for water potability prediction, several enhancements are planned to extend its impact, usability, and scientific utility:

A. *Real-Time Data Integration*

Future versions may support real-time water quality data ingestion from public repositories such as EQuIS, as well as IoT-based sensor networks. This would enable dynamic monitoring of water sources and instant alert systems for contamination events.

B. *Geospatial Risk Mapping*

By integrating geolocation tags, the system can be extended to generate risk heatmaps, enabling public health departments to visualize potability status across regions. This would facilitate resource allocation and preventive action in high-risk zones.

C. *Mobile and Offline Access*

To enhance reach in rural or low-connectivity areas, the current Streamlit web app could be containerized as a Progressive Web App (PWA) for mobile devices. Offline prediction using pre-loaded models and data is also being explored.

D. *AutoML and Self-Updating Models*

Integrating AutoML tools such as AutoGluon or H2O.ai will allow non-technical users to retrain and deploy updated models based on new datasets. This ensures long-term adaptability without needing data science expertise.

E. *Explainability and Trust*

Incorporating explainability techniques such as SHAP and LIME helps end-users interpret which features influenced a classification decision. This capability is particularly valuable for policy makers and regulatory transparency.

F. *NGO and Government Integration*

Discussions are underway to adapt this tool for use in humanitarian and governmental water testing initiatives. The system could serve as a decision-support tool for environmental regulators and disaster response agencies.

G. *Multilingual and Accessibility Enhancements*

To increase adoption across global communities, future versions will support multilingual interfaces, screen reader compatibility, and local water quality thresholds.

VII. LIMITATIONS

While the Water Potability Prediction App demonstrates strong performance using calibrated XGBoost and SMOTE-based balancing, several limitations exist:

- **Dataset Size and Scope:** The model is trained on a relatively small dataset of ~3,200 samples. This may limit generalizability across different regions, climates, or seasonal variations.
- **Lack of Geolocation Context:** The current model uses only physicochemical properties without incorporating source location or regional context, which could influence water potability.
- **Synthetic Sampling Risks:** Although SMOTE helps balance the dataset, it may introduce synthetic

data points that do not correspond to real-world physical samples, possibly affecting precision.

- **Interpretability:** While the app includes confidence scores, deeper explainability (e.g., SHAP or LIME) is not integrated into the interface yet.
- **Real-Time Applicability:** The model is trained on historical data and does not currently ingest live sensor or IoT data, which limits deployment in real-time monitoring environments.

VIII. CONCLUSION

This study presents a practical, cost-free, and fully re-producible machine learning solution for predicting water potability using publicly available environmental data. By combining SMOTE for class balancing, XGBoost for accurate classification, and probability calibration for interpretability, the system achieves robust performance suitable for real-world applications.

The model is deployed through a lightweight Streamlit interface that enables users to input water quality measurements and receive real-time predictions along with confidence scores. This end-to-end workflow—from model training to deployment—ensures accessibility even in resource-constrained settings, contributing directly to public health initiatives and environmental decision-making. Streamlit's zero-setup, web-based interface enhances accessibility for NGOs, public health officers, and academic users by eliminating the need for infrastructure management or specialized coding skills [15].

All source code, documentation, and test cases are available in a public GitHub repository, promoting transparency, reproducibility, and collaboration.

Future work will extend this framework by integrating live data from IoT-based sensors, enabling continuous water monitoring in vulnerable communities. Additional plans include adapting the model to multi-regional datasets, supporting multilingual interfaces, and incorporating interpretability tools like SHAP to provide actionable insights for local policymakers and non-expert users.

This project exemplifies how accessible AI tools can bridge data science and societal impact, offering a replicable model for other environmental applications in the public domain.

APPENDIX A

USER INTERFACE SNAPSHOT

Fig. 6 shows the deployed Streamlit application interface, allowing users to input physicochemical values and obtain predictions with confidence scores in real time.

APPENDIX B

SAMPLE PREDICTION OUTPUT

Below is an example of the JSON response generated after submitting a test sample through the app interface. The output includes the binary classification labels along with calibrated probability scores.

```
1 {
2   "Not Potable": "28.27%",
3   "Potable": "71.73%",
4   "Confidence Score": "71.73%"
}
```

Listing 2. Sample App Output in JSON Format

Potable / Not Potable: These fields represent the model's calibrated probability estimates for each class label. The value with the higher percentage indicates the final prediction.

Confidence Score: This is a duplicate of the higher probability value for user convenience, displayed explicitly for clarity in the app interface.

APPENDIX C SAMPLE TEST CASES

Table V presents 10 representative water samples along with the full set of physicochemical input features used in prediction. Outputs are shown with class labels and calibrated confidence scores using the deployed Streamlit app powered by the XGBoost model.

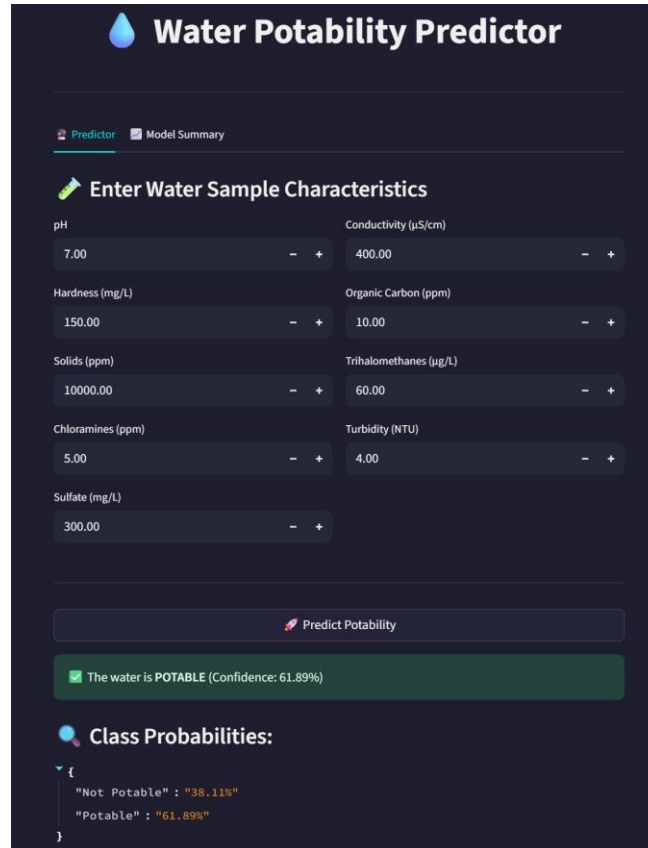


Fig. 6. Streamlit-Based Potability Prediction Web Interface

APPENDIX D SOURCE CODE REPOSITORY

All training scripts, model files, and deployment logic are available at:

For source code and documentation, visit: <https://github.com/pramathparashar/WaterQuality-Predictor>

The repository includes:

- Training pipeline notebook
- Final model and scaler (.pkl)
- Streamlit app source code
- Sample test cases and screenshots
-

APPENDIX E GLOSSARY OF KEY TERMS

- **SMOTE (Synthetic Minority Over-sampling Tech- nique)** – A data augmentation method used to balance imbalanced datasets by generating synthetic samples for the minority class.
- **CalibratedClassifierCV** – A scikit-learn wrapper that enhances the reliability of predicted probabilities by applying post-training calibration techniques like Platt scaling or isotonic regression.
- **Confidence Score** – The model's probabilistic estimate of the predicted class. In this paper, it refers to the calibrated probability that a sample is classified as potable.

TABLE V- COMPACT DISPLAY OF FEATURE-WISE TEST CASE PREDICTIONS

Case	Features (Split Line)	Prediction	Conf.
1	pH=7.0, Hardness=145, Solids=25000, Chloramines=7.5, Sulfate=375, Conductivity=450, Org. Carbon=10.2, THMs=60, Turb=4.0	Potable	71.73%
2	pH=5.9, Hardness=130, Solids=18000, Chloramines=6.1, Sulfate=330, Conductivity=400, Org. Carbon=9.8, THMs=100, Turb=4.8	Not Potable	54.47%
3	pH=8.1, Hardness=160, Solids=29000, Chloramines=8.0, Sulfate=420, Conductivity=470, Org. Carbon=11.1, THMs=80, Turb=3.2	Potable	96.71%
4	pH=6.2, Hardness=110, Solids=21000, Chloramines=6.5, Sulfate=200, Conductivity=410, Org. Carbon=9.0, THMs=35, Turb=5.1	Not Potable	90.24%
5	pH=7.5, Hardness=142, Solids=27000, Chloramines=7.3, Sulfate=380, Conductivity=455, Org. Carbon=10.5, THMs=65, Turb=4.5	Potable	80.91%
6	pH=6.0, Hardness=150, Solids=23000, Chloramines=6.7, Sulfate=250, Conductivity=430, Org. Carbon=9.7, THMs=45, Turb=3.8	Not Potable	60.22%
7	pH=8.4, Hardness=135, Solids=28000, Chloramines=8.5, Sulfate=410,	Potable	93.89%

8	Conductivity=480, Org. Carbon=10.8, THMs=70, Turb=2.9 pH=5.5, Hardness=125, Solids=20000, Chloramines=5.9, Sulfate=195, Conductivity=390, Org. Carbon=8.5, THMs=40, Turb=4.2	Not Potable	85.34%
9	pH=7.8, Hardness=148, Solids=26000, Chloramines=7.6, Sulfate=370, Conductivity=460, Org. Carbon=10.6, THMs=77, Turb=4.0	Potable	88.65%
10	pH=6.8, Hardness=132, Solids=24000, Chloramines=6.9, Sulfate=310, Conductivity=420, Org. Carbon=9.2, THMs=60, Turb=4.7	Potable	69.12%

- **XGBoost (Extreme Gradient Boosting)** – A fast, regularized gradient boosting algorithm widely used for high-performance classification and regression tasks on tabular data.
- **Streamlit** – An open-source Python framework used to deploy machine learning models through interactive web applications without the need for traditional front-end development.

REFERENCES:

1. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
2. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, San Francisco, CA, USA, 2016, pp. 785–794.
3. U.S. Environmental Protection Agency, "National Primary Drinking Water Regulations," 2023. [Online]. Available: <https://www.epa.gov/ground-water-and-drinking-water>
4. A. Ahmed, S. Mahmud, and M. H. Kabir, "Machine Learning Techniques for Water Quality Prediction: A Comparative Analysis," *IEEE Access*, vol. 8, pp. 97867–97879, 2020.
5. F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
6. Streamlit Inc., "Streamlit: Turn Python Scripts into Shareable Web Apps," 2023. [Online]. Available: <https://streamlit.io/>
7. A. Yadav and V. Batra, "Potability Prediction Using ML Models," in *Proc. 3rd Int. Conf. Adv. Comput. Commun. Technol.*, 2021, pp. 112–117.
8. P. Parashar, "WaterQuality-Predictor," GitHub Repository, 2025. [Online]. Available: <https://github.com/pramathparashar/WaterQuality-Predictor>
9. S. Niculescu-Mizil and R. Caruana, "Predicting Good Probabilities with Supervised Learning," in *Proc. 22nd Int. Conf. Machine Learning (ICML)*, 2005, pp. 625–632.

10. World Health Organization, “Guidelines for Drinking-water Quality,” WHO, 4th Edition, 2022. [Online]. Available: <https://www.who.int/publications/i/item/9789241549950>
11. S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in *Proc. NeurIPS*, 2017.
12. G. Lemaître, F. Nogueira, and C. K. Aridas, “Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning,” *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.
13. J. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.
14. A. Niculescu-Mizil and R. Caruana, “Predicting Good Probabilities with Supervised Learning,” in *Proc. ICML*, 2005, pp. 625–632.
15. Streamlit Inc., “Streamlit: Turn data scripts into shareable web apps,” 2023. [Online]. Available: <https://streamlit.io>